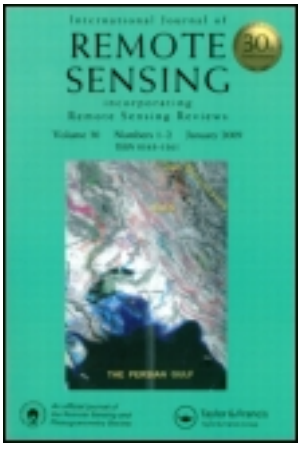

International Journal of Remote Sensing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tres20

An object-based analysis filtering algorithm for airborne laser scanning

Menglong Yan [a b], Thomas Blaschke [c], Yu Liu [b] & Lun Wu [b]

[a] Key Laboratory of Spatial Information Processing and Application System Technology, Chinese Academy of Sciences, Beijing, China

[b] Institute of Remote Sensing and Geographical Information Systems, Peking University, Beijing, China

[c] Z_GIS - Centre for Geoinformatics and Department for Geography and Geology, University of Salzburg, A-5020, Salzburg, Austria

Version of record first published: 25 Jun 2012

To cite this article: Menglong Yan, Thomas Blaschke, Yu Liu & Lun Wu (2012): An object-based analysis filtering algorithm for airborne laser scanning, International Journal of Remote Sensing, 33:22, 7099-7116

To link to this article: http://dx.doi.org/10.1080/01431161.2012.699694

International Journal of Remote Sensing
Vol. 33, No. 22, 20 November 2012, 7099–7116

An object-based analysis filtering algorithm for airborne laser scanning

MENGLONG YAN*†‡, THOMAS BLASCHKE§, YU LIU‡ and LUN WU‡
†Key Laboratory of Spatial Information Processing and Application System Technology, Chinese Academy of Sciences, Beijing, China
‡Institute of Remote Sensing and Geographical Information Systems, Peking University, Beijing, China
§Z_GIS – Centre for Geoinformatics and Department for Geography and Geology, University of Salzburg, A-5020 Salzburg, Austria

(*Received 14 July 2011; in final form 28 February 2012*)

Ground filtering is a key process to derive digital terrain models from airborne laser scanning data. Although many methods have been developed to tackle the filtering problem, it has not been fully solved so far. Current algorithms mainly focus on neighbourhood-based or directional filtering approaches. A new object-based analysis (OBA) method is proposed in this article. First, a grid index algorithm accelerates access to unorganized cloud points. Then, a segmentation algorithm is deployed based on the index, and objects are obtained. A filtering logic that utilizes the objects' characteristics is designed. Following this, the performance of the method is comprehensively tested using publicly available International Society for Photogrammetry and Remote Sensing (ISPRS) test data sets for nine urban and six rural regions, and the results are compared to those of eight other algorithms. The OBA method implemented in this article reveals good results without scene-wise optimization of the parameters, and it ranks third or fourth in most of the cases.

1. Introduction

Airborne laser scanning (ALS) is a technology for the quick acquisition of digital terrain models (DTMs) and digital surface models (DSMs) (Ackermann 1999, Sithole and Vosselman 2004). The method is also utilized for extracting features such as buildings (Sohn and Dowman 2007) and vegetation (Wagner *et al*. 2008).

As is well known, ground and non-ground back echoes are confusingly mixed in raw ALS data. When aiming to differentiate topographic information, the basic task of processing ALS data is to distinguish bare ground points from object points. Due to the complexity of terrain, a full automation of the point filtering process is not possible (Sithole and Vosselman 2004). Hyyppä *et al*. (2004) report from practical tests that laser scanning data even include low points under the ground level. These may be real returns or falsely interpreted elevation values from strong backscatters. Therefore, the typical procedures in existing approaches require a lot of human interaction, which is usually labour-intensive and time-consuming. Many research articles focus on filtering algorithms (Kraus and Pfeifer 1998, Pfeifer *et al*. 1999, Axelsson 2000, Vosselman 2000, Roggero 2001, Sithole 2001, Brovelli *et al*. 2002, Elmqvist 2002,

*Corresponding author. Email: yanmenglong@gmail.com

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online © 2012 Taylor & Francis
http://www.tandfonline.com
http://dx.doi.org/10.1080/01431161.2012.699694

Sohn and Dowman 2002, Wack and Wimmer 2002, Shan and Sampath 2005, Meng *et al*. 2009). Several studies have compared the performance of different algorithms (Sithole and Vosselman 2004, Zhang and Whitman 2005, Meng *et al*. 2010). Filtering algorithms filter the raw ALS point cloud or operate on a grid elevation generated by interpolations of points (Sithole and Vosselman 2004, Meng *et al*. 2009), which can be considered a DSM (Lloyd and Atkinson 2002). Most existing algorithms work on the assumption that the natural ground changes gradually and, as a result, that (1) the height difference between neighbouring ALS points on the ground is small and (2) the probability that a point accepted as a non-ground point increases with the height difference increases between two neighbouring points.

Kraus and Pfeifer (1998) proposed an iterative linear prediction method, which distinguishes between ground points and non-ground points by providing the points with different weights in iterations. Axelsson (2000) developed an adaptive triangulated irregular network (TIN) model, in which points below a given threshold and within a given distance of the nearest triangle nodes are accepted in the triangle. The model is adaptive and the threshold is changed adaptively between the iterations. Vosselman (2000) implemented a height difference mechanism in such a way that a point is accepted as a ground point if the height difference between the point and its neighbourhood is not beyond a given threshold. Roggero (2001) utilized a local linear regression method in order to find the initial bare ground surface, where points are classified as ground points or non-ground points based on their distance from the initial bare ground surface. Sohn and Dowman (2002) proposed a TIN model, in which the TIN is created in an iterative process, and the lowest point within the respective triangle is accepted as a bare ground point, while the remaining points are considered as object points. Brovelli *et al*. (2002) assumed that the points within a closed boundary are considered as object points. The algorithm finds the edge and connects the edges as a boundary. Wack and Wimmer (2002) interpolated a raster using ALS points while subsequently detecting elements of objects with Laplacian of Gaussian (LoG) operations in a hierarchical approach. Shan and Sampath (2005) proposed a one-dimensional and bi-directional labelling algorithm to identify ground and non-ground points in an urban area. Both the slope criterion and height criterion are used in this procedure. Meng *et al*. (2009) developed a multi-directional filtering algorithm to combine the advantages of directional information and neighbourhood information.

Most of these algorithms operate at a global scale. This means that the operation is the same no matter whether the scene is simple or complex. Further, when using global-scale algorithms, each ALS point is treated as an individual object, which means that most of the existing algorithms do not take full advantage of neighbourhood information. For instance, in linear prediction methods, i.e. slope-based or height-based models, filtering algorithms are applied throughout the scene with a global threshold. In some methods, the threshold is changed adaptively during the filtering procedure (Axelsson 2000). The morphological methods analyse the cloud points based on regions; however, usually the analysis window size and the related threshold should be chosen carefully or changed gradually between iterations (Zhang *et al*. 2003). In addition, the morphological algorithms usually operate on grid data derived from ALS points using interpolation; thus, errors may be introduced into the procedure (Zhang *et al*. 2003, Sithole and Vosselman 2004, Meng *et al*. 2009).

In this article, a filtering algorithm that uses object-based analysis (OBA) is proposed. First, this algorithm extracts objects from the ALS point cloud through

segmentation. Further, a topology relationship of these objects is built and the points are analysed based on the respective objects instead of using a global scale. The algorithm utilizes the structural information of high-resolution point clouds, including topology relationship information and object information such as object size, etc. Finally, the performance of the algorithm is tested using publicly available ISPRS test data sets.

2. Methodology

2.1 *OBA process framework*

With the development of earth observation techniques, such as IKONOS, QuickBird or GeoEye-1, the resolution of remote-sensing images has been improved dramatically. This improvement of resolution has rendered the conventional pixel-by-pixel image analysis unsatisfactory (Li 1996, Guo *et al*. 2007, Liu *et al*. 2008, Blaschke 2010). Object-based image analysis (OBIA) has been a hot research field in the past decade. OBIA methods derive objects through image segmentation and utilize many features of the resulting objects, including size, topology and shape information. Nowadays, commercial ALS equipment is capable of delivering high-density point clouds, where the points can represent relatively small structures such as buildings and terrain forms such as terraces or levees. When targeting such information, it may be beneficial to identify existing objects, such as buildings, trees, etc. In this article, we use the term OBA, and we will carry out ALS filtering based on OBA.

First, the original point clouds are organized by creating a grid index. Local minimum height value points are selected as the flag points for local grid cells. Subsequently, the segmentation and the analysis procedures are carried out based on the grid index. In accordance with the literature, we hypothesize that the grid index will significantly accelerate the process of finding neighbourhood points.

Second, a segmentation algorithm is developed, which begins from the flag data points. Several key procedures are implemented in this process, including (1) seed point generation, (2) initializing the segmentation, (3) invalid object absorption and merging, (4) creating boundaries of derived objects and (5) creating a topological relationship between these objects. The final results are objects consisting of ALS points. Third, several parameters are extracted from the derived objects and are used for the subsequent filtering of these objects. In this way, objects are identified by their features, and are eliminated from the background, which is considered as a non-terrain object. After this procedure, the remaining points in the grid are filtered based on local flag points by using a local slope method. The details will be discussed in §2.4.

2.2 *Grid index for multi-scale analysis*

Many existing algorithms operate on discrete point clouds or grid data derived by interpolation of cloud points (Lloyd and Atkinson 2002, Sithole and Vosselman 2004, Meng *et al*. 2009). Many articles point out that interpolation will cause errors in the derived grid data (Sithole and Vosselman 2004, Meng *et al*. 2009). Instead of interpolation methods, a suitable data organization method must be utilized in order to handle the discrete points. Several three-dimensional indexing methods exist such as the Octree and kd-tree data structures.

We developed the following grid index in order to organize the original point cloud; it is used to find the neighbouring points of a given laser point. The structure is demonstrated in figure 1. When laser points are loaded to memory, they are organized as an array based on their IDs ('data array' in figure 1), and using the storage structure it is possible to index each point by its ID with the complexity of O(1).

The grid index is like a matrix ('grid index matrix' in figure 1) and, for each cell of the matrix, there is a cell array containing the IDs of all laser points that fall into the cell. The grid position (x_{id}, y_{id}) of a given point (X, Y, Z) can be calculated using equation (1). For instance, in figure 1 it is assumed that four points, a, b, c and d, are located in the cell, marked by a red cross. In this case, there is a corresponding array in the 'grid index matrix', in which the four points are organized in a cell array by their IDs ('cell array' in figure 1). Then for each grid cell, the original points from the 'data array' are indexed by their IDs effectively. For instance from the cell array that contains the IDs of points a, b, c and d, we could index the original laser points by their IDs with the complexity of O(1):

$$x_{\mathrm{id}} = \frac{X - X_{\mathrm{min}}}{X_{\mathrm{size}}}, \quad y_{\mathrm{id}} = \frac{Y - Y_{\mathrm{min}}}{Y_{\mathrm{size}}}. \tag{1}$$

In equation (1), (X_{min}, Y_{min}) and (X_{max}, Y_{max}) describe the extent of the ALS data set. It can be derived from the file header, for instance from the header of a LAS format file. X_{size} and Y_{size} are the width and height of a grid cell, respectively, defined by

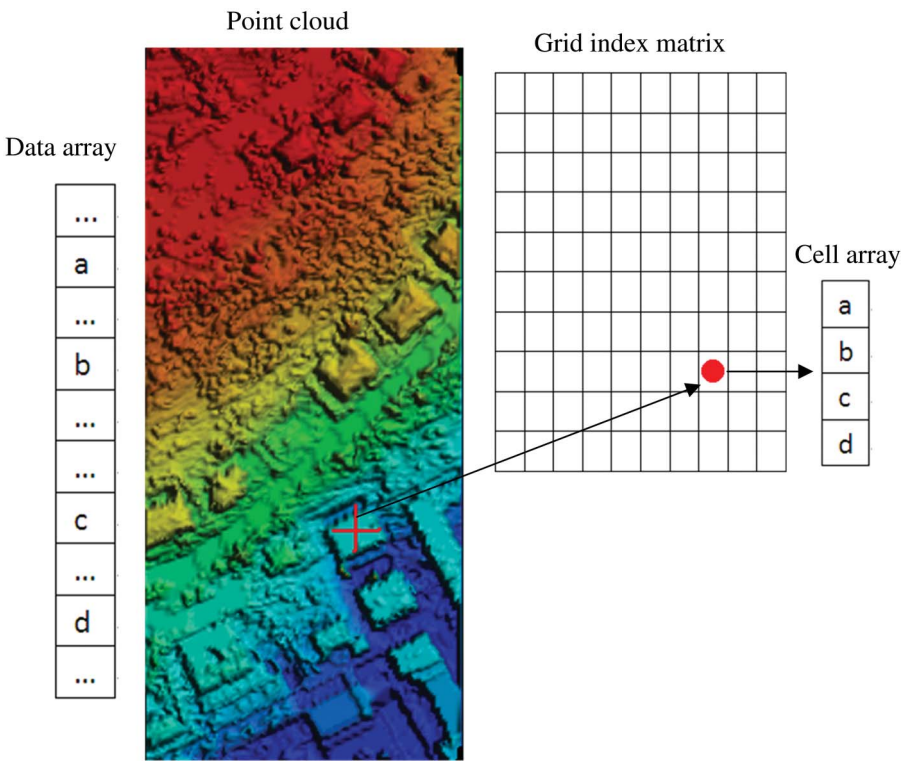

Figure 1. Demonstration of the grid index.

the user. In addition, the width (W) and height (H) of the grid can be derived from equation (2):

$$W = \text{floor}\left(\frac{X_{\max} - X_{\min}}{X_{\text{size}}}\right) + 1, \quad H = \text{floor}\left(\frac{Y_{\max} - Y_{\min}}{Y_{\text{size}}}\right) + 1. \tag{2}$$

After these calculations, each grid cell is composed of an array containing point IDs. If no ALS points fall into the corresponding cell, the cell remains empty. For the point ID array of each cell, the point with the lowest Z value is chosen to represent the cell. If the array is empty, the related cell is considered to be empty. It is emphasized that no interpolation is conducted in this process, and we use the original information of laser points for segmentation.

2.3 *Segmentation algorithm*

Segmentation is an important topic in image analysis. It can be defined as a partitioning process of an image into homogeneous and non-overlapping regions that are later identified as objects (Cheng *et al*. 2001). Many algorithms have been developed in the field of pattern recognition as well as in remote-sensing image analysis (Benz *et al*. 2004, Mitra *et al*. 2004, Li and Xiao 2007, Saha and Bandyopadhyay 2010).

The plethora of segmentation algorithms can be grouped into four categories (Blaschke 2010): point-based, edge-based, region-based and combined. In this article, the main purpose of the segmentation step is to derive partitions whose height values change gradually. These regions are considered to be relatively homogeneous in this respect. For this purpose, we propose an improved region-growing algorithm.

In our algorithm, we first build a grid index for neighbourhood searching, in which only the local point of z-min value is used for the segmentation of each grid cell instead of all the ALS points. We assume that the segmentation process is accelerated by using the strategy of the grid index. Then by using the local point of z-min value for each grid cell, we search for seed points of a given size in each region, which is used for segmentation. We use slope as the neighbour point accepting criterion, and the searching direction is the four-neighbour breadth-first search.

2.3.1 Generating seed points. Finding seed points is a crucial task for many existing filtering algorithms, e.g. for the TIN model proposed by Axelsson (2000). In the OBA approach, we consider the seed points to be local minimal height values. First, we filter out the 'low points' (points under the ground level), and then, for a given grid size C, the entire study area is partitioned into R regions. Subsequently, seed points S are generated according to equations (3) and (4):

$$R = \text{floor}\left(\frac{X_{\max} - X_{\min}}{C} + 1.0\right) \times \text{floor}\left(\frac{Y_{\max} - Y_{\min}}{C} + 1.0\right), \tag{3}$$

$$S_i = \min(\{Z_j | j \in R_i\}). \tag{4}$$

2.3.2 Segmentation based on grid index. In §2.2, a data matrix was generated and filled with points of local minimum z values. If S stands for the collection of seed

points while **M** stands for the matrix of the point identification, then the segmentation procedure can be demonstrated by the following pseudo-code:

```
M ← 0
id ← 0
obj_hashtable ← new hashtable()
foreach(s in S) {
    if(M [s]!= 0) {
        id ← id +1
        obj ← new OBJ(id)
        M [s] ← id
        obj.accept(s)
        queue.check_enter(s, s.get_4neighbour_points())
        while (queue is not empty) {
            p ← queue.exit()
            obj.accept(p)
            M [p] ← id
            queue.check_enter(p, s.get_4neighbour_points())
        }
        objs_hashtable.put(id, obj)
    }
}
Foreach(s in left) {
    if(M [s]!= 0) {
        id←id +1
        obj← new OBJ(id)
        M [s] ← id
        obj.accept(s)
        queue. check_enter (s, s.get_4neighbour_points())
        while (queue is not empty) {
            p ← queue.exit()
            obj.accept(p)
            M [p] ← id
            queue.check_enter (p, s.get_4neighbour_points())
        }
        objs_hashtable.put(id, obj)
    }
}
```

The segmentation procedure begins from these seed points. First, a new object is created with an incremental id. If a point p is absorbed as part of the object, it will be labelled with the ID of the corresponding object. Then the four-neighbourhood points of p are put into a queue after a criterion check. Each object is derived when its queue is empty. Then the segmentation is carried out on the remaining points that have no object ID. In this procedure, 'check_enter' is used to check whether a neighbour point p_1 (x_1, y_1, z_1) of an accepted point p (x_p, y_p, z_p) should be accepted as part of the respective object. The check function is defined in equation (5):

$$\theta = \operatorname{atan}\left(\frac{|\Delta z|}{\sqrt{(\Delta x)^2 + (\Delta y)^2}}\right) = \operatorname{atan}\left(\frac{|z_1 - z_p|}{\sqrt{(x_1 - x_p)^2 + (y_1 + y_p)^2}}\right). \tag{5}$$

In addition, an appropriate threshold for the angle θ_{th} is required to check whether a point should be absorbed. If θ is smaller than the given θ_{th}, the point will be accepted, otherwise refused.

With a suitable threshold angle θ_{th} defined by the user, an initial segmentation is conducted. After the initial segmentation, objects are stored in a hash table as described in the previous pseudo-code. Then, invalid objects are identified. To do this, an algorithm is developed to derive topologic relationships from any of the objects in the hash table by checking the identification matrix **M** as described in the previous pseudo-code. The procedure can be described as follows:

```
foreach(obj in objs_hashtable) {
    foreach(pt in obj.all_pts) {
        bdFlag ← false
        foreach(npt in pt.get_8neighbour_points()) {
            if(obj.id!= M[npt]) {
                obj.addTop(M[npt])
                bdFlag ← true
            }
        }
        if (bdFlag) {
            obj.boundary_pts.add(pt)
        }
    }
}
```

In this procedure, for each point p of the current object with ID1, we get its eight neighbour points set npts if there is a point of npts whose id is not ID1, and then we assume that the point p is a boundary point of the current object with ID1. In addition, the topologic relationships of each object are organized by means of a hash set, which helps to avoid multiple entries (http://www.sgi.com/tech/stl/hash_set.html).

The grid data derived in §2.2 are used in the segmentation procedure. Each grid cell contains an array of IDs of ALS points that lie within the respective cell. However, the array of some cells may be empty if no ALS points fall into the cell. This is caused by the low density of ALS points or too small a cell size (X_{size}, Y_{size}). Due to this problem, after the initial segmentation of the grid data, many invalid objects (just holes) may exist and an appropriate algorithm is designed to eliminate these invalid objects. These objects are usually small in size and are identified with an invalid object tag in the identification matrix **M** if the respective grid index cells are empty. Then, after the creation of topologic relationships, these invalid objects are absorbed by the adjacent objects. To guarantee consistency, the topologic relationships are updated each time these invalid objects are absorbed.

2.4 *Object-based filtering*

As a result of this segmentation procedure, homogeneous regions are created and region-based analyses can be carried out. According to the OBIA theory (Liu *et al*. 2008, Blaschke 2010), particular features are derived from the objects and their respective geometries. Three levels of features can be obtained from the segmented parts (Liu *et al*. 2008). Level one features derived from single objects include area, perimeter, shape, etc. (Liu *et al*. 2010). Level two features address spatial relations between

two objects, such as embeddedness, proximity and adjacency. Level three features are composed of spatial patterns in which more than two objects are involved.

In our approach, four major object characteristics are used. (1) The information whether an object is derived from a seed point. As described in §2.3.1, seeds are generated by a local minimum search for each local search window. It assumes that if an object originates from a seed point, it is a ground object and the remaining objects are considered as uncertain objects. (2) Topology information: (a) an object is more likely to be a non-ground object if it has the highest mean height value compared to its neighbouring objects based on the derived topological relationship, while (b) an object is more likely to be a ground object if it has the lowest mean height value compared to its neighbours. (3) Sharp boundary points. The local maximum z-differential value is calculated based on the four-neighbourhood case as defined in equation (6). Then, a special edge detector is defined in equation (7). In equation (6), LMHD stands for local maximum height differential matrix, N is the symbol for the four-neighbourhood case, i stands for the current cell position and j stands for the neighbourhood cell position of i. If the boundary point collection of an uncertain object contains sharp boundary points, it is classified as non-ground object. (4) Size information: objects that are too small are classified as tiny non-ground objects. A flow chart is provided in figure 2, which illustrates the analysis steps:

$$\mathrm{LMHD}(i) = \max(\{\forall j \in N,\ Z(i) - Z(j)\}), \tag{6}$$

$$\mathrm{EDGE}(i) = \begin{cases} 0 & \mathrm{LMHD}(i) \leq 0 \\ \mathrm{LMHD}(i) & \mathrm{LMHD}(i) > 0\ . \end{cases} \tag{7}$$

After these filtering steps, an additional process is carried out to identify the remaining points in the grid index. Case 1: if an object is identified as a non-ground object, then all the remaining points within the object are classified as non-ground points. Case 2: if an object is identified as a ground object, then the remaining points within the object are filtered using the nearest ground point based on equation (5) using the local threshold of slope defined by the user.

2.5 *Computation performance analysis*

By using the grid index, there are $W \times H$ points (point with the lowest Z value of each cell of grid index matrix) for segmentation. We assume that the total number of laser points is N, and that the complexity for building the grid index is $O\,(N)$. When generating seed points, the complexity is $O\,(W \times H)$. In this respect, our segmentation algorithm is similar to typical region-growing algorithms; according to Shih (2010), the computational complexity is $O\,((W \times H)\ \lg\,(W \times H))$. For the second step of building topologic relationships and defining boundary points for each object, the complexity is $O\,(W \times H)$.

Assuming that there are M objects after segmentation, then the complexity for object-based filtering process is $O\,(M)$. Usually, M is much less than N, and therefore the total time complexity for the OBA algorithm is $O\,(N + W \times H + (W \times H)\lg(W \times H) + W \times H + M) \approx O\,((W \times H)\lg(W \times H))$.

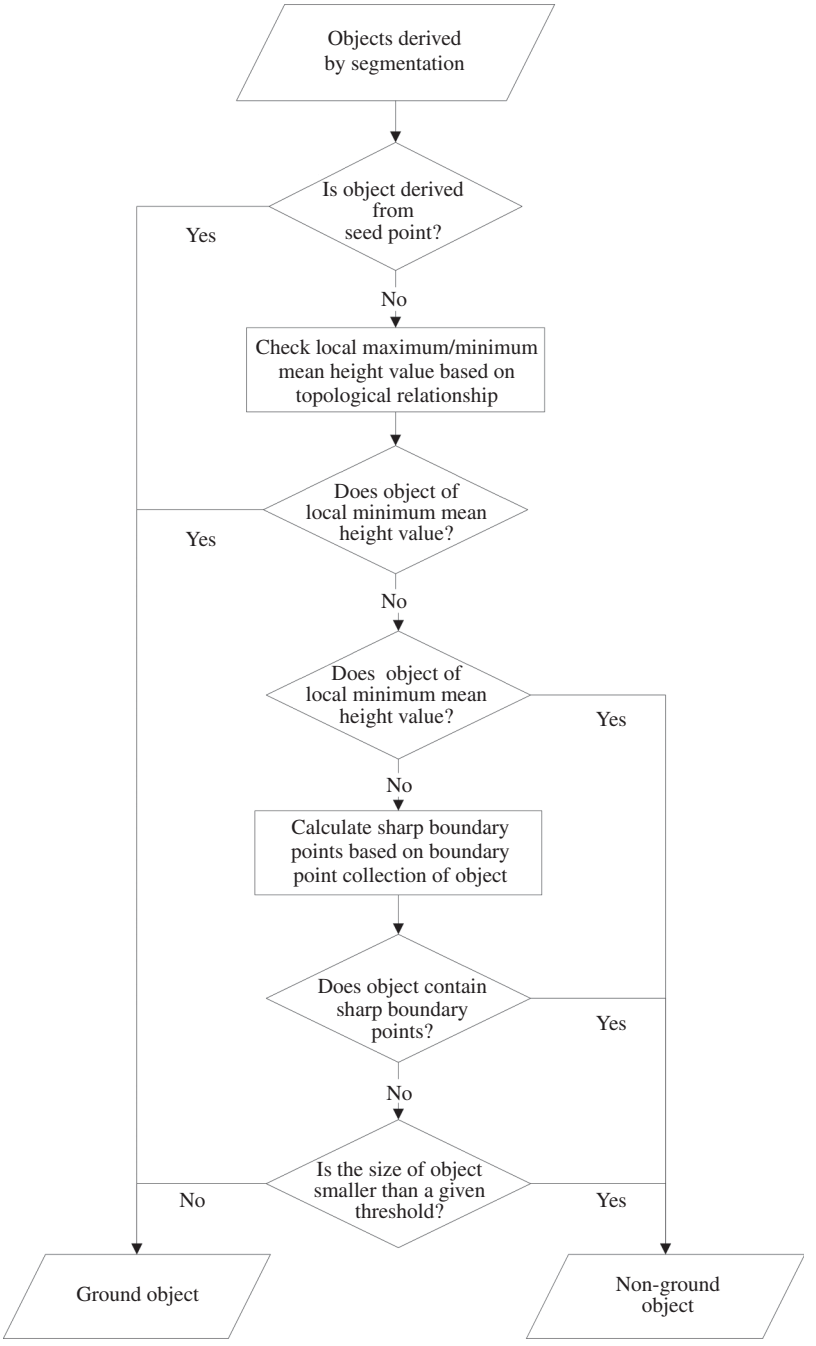

Figure 2. Flow chart of the filtering logic.

3. Analysis of the ISPRS test data sets

The algorithm is implemented in C++ language. In order to test the performance of the algorithm absolutely and relatively against other algorithms, experiments are

Table 1. Study site features of 15 data sets from Sithole and Vosselman (2003).

Site name	Data set name	Point spacing (m)	Special features
Urban site 1	samp 11, samp 12	1–1.5	Steep slopes, mixture of vegetation and buildings on hillside, buildings on hillside, data gaps
Urban site 2	samp 21, samp 22, samp 23, samp 24	1–1.5	Large buildings, irregularly shaped buildings, road with bridge and small tunnel, data gaps
Urban site 3	samp 31	1–1.5	Densely packed buildings with vegetation between them, building with eccentric roof, open space with mixture of low and high features, data gaps
Urban site 4	samp 41, samp 42	1–1.5	Railway station with trains (low density of terrain points), data gaps
Rural site 5	samp 51, samp 52, samp 53, samp 54	2–3.5	Steep slopes with vegetation, quarry, vegetation on riverbank, data gaps
Rural site 6	samp 61	2–3.5	Large buildings, road with embankment, data gaps
Rural site 7	samp 71	2–3.5	Bridge, underpass, road with embankments, data gaps

carried out using publicly available ISPRS test data sets. The ISPRS data sets can be downloaded from the ISPRS website of Commission III, Working Group 3 (http://www.itc.nl/isprswgIII-3/filtertest/index.html). They cover seven sites and consist of 15 individual data sets. The study sites are described in table 1, which is modified from Sithole and Vosselman (2003). Site 8 is excluded here due to the lack of reference data.

From the ISPRS filtering report (Sithole and Vosselman 2003), Error Type I, Error Type II and Error Total information are used to verify the performance of an algorithm. Error Type I (eI), Error Type II (eII) and Error Total (eT) are defined in equations (8)–(10). In these equations, a is the number of ground points that have been correctly identified as ground points; b is the number of ground points that have been incorrectly identified as non-ground points; c is the number of non-ground points that have been incorrectly identified as ground points; d is the number of non-ground points that have been correctly identified as non-ground points; and e is the total number of points tested. Although it is suggested by Sithole and Vosselman (2003) that filtering should aim to minimize Error Type I because Error Type II is easier to edit manually, most of the existing algorithms seem to pay more attention to minimizing Error Type II. For the best possible comparison to the existing algorithms, we therefore try to minimize Error Type II.

$$\mathrm{eI} = \frac{b}{a+b}, \tag{8}$$

$$\mathrm{eII} = \frac{c}{c+d}, \tag{9}$$

$$\mathrm{eT} = \frac{b+c}{e}. \tag{10}$$

Table 2. Parameters for segmentation and filtering.

Data set name	Index size	Average number of points in each index cell	Window size to find seeds (m)	Slope threshold for segmentation (°)	Local slope threshold (°)
samp 11	2	3.72	50	30	10
samp 12	2	3.85	50	30	10
samp 21	2	3.63	50	30	10
samp 22	2	3.86	50	30	10
samp 23	2	3.34	50	30	10
samp 24	2	3.38	50	30	10
samp 31	2	4.09	50	30	10
samp 41	2	2.43	50	30	10
samp 42	2	3.68	50	30	10
samp 51	3	1.61	50	45	15
samp 52	3	1.47	50	45	15
samp 53	3	1.52	50	45	15
samp 54	3	1.55	50	45	15
samp 61	3	1.41	50	45	15
samp 71	3	1.61	50	45	15

All parameters such as sizes and thresholds used in these tests are illustrated in table 2. The index size parameter is used to build the grid index, which is set according to the density of ALS point clouds. Then, the average number of points in each index cell can be calculated. The seed points are found according to the given window size defined by the user. Usually, the size of a window should exceed the size of the largest buildings within the study area. The segmentation is carried out based on the slope threshold. This local slope threshold is used to filter the remaining points after the initial filtering of objects through the segmentation process. We do not optimize this parameter for every site. It is kept constant in order to test the stability of our algorithm. Errors of type I, type II and type Total are calculated and the results are illustrated in table 3 and figures 3–5.

Table 3. Error Type I, Error Type II and Error Total.

Data set name	Error Type I (%)	Error Type II (%)	Error Total (%)
samp 11	28.87	5.71	20.62
samp 12	11.43	1.62	7.12
samp 21	7.14	4.96	6.76
samp 22	15.8	2.11	11.98
samp 23	27.92	2.31	16.59
samp 24	28.44	3.43	22.47
samp 31	5.56	1.88	4.02
samp 41	27.27	0.85	14.02
samp 42	9.39	3.2	5.18
samp 51	6.77	6.4	6.69
samp 52	11.63	7.65	11.2
samp 53	19.08	4.08	18.5
samp 54	14.67	2.0	7.85
samp 61	11.72	0.84	11.35
samp 71	16.05	3.13	14.59

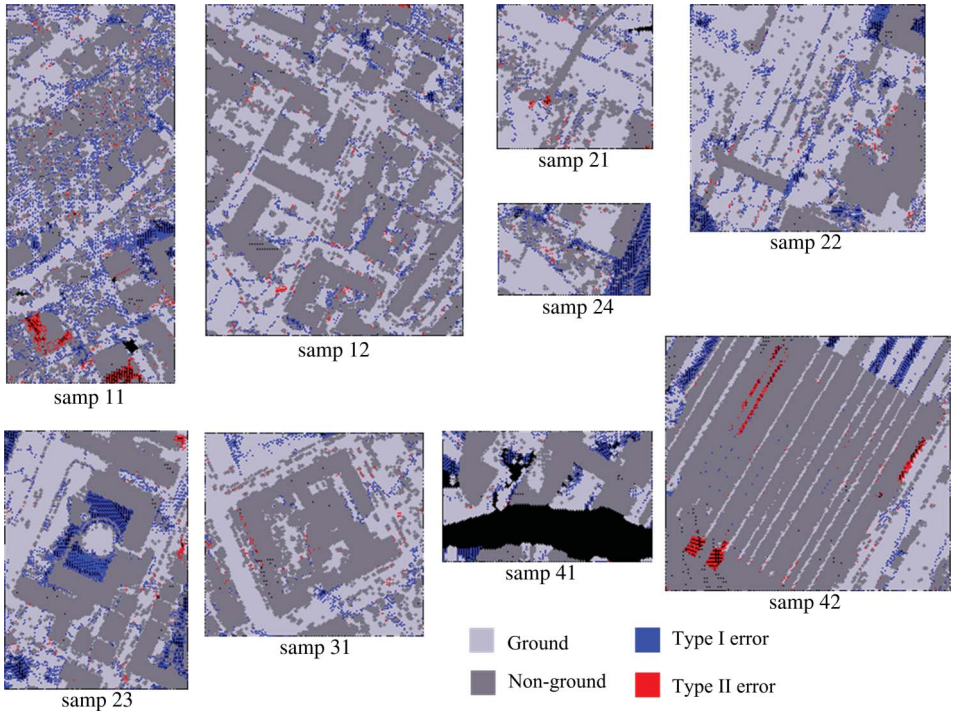

Figure 3. Error distribution for urban sites 1–4 displayed on the same scale.

4. Results

The nine urban test sites are relatively flat; however, they are composed of a complex mixture of land use and many buildings (figure 3, samp 11, samp 12, samp 21–samp 24, samp 31, samp 41 and samp 42). The spacing of the laser points in these nine urban regions is about 1 m. In our algorithm, the size of the grid index is set to 2 m. This results in typically three to four points for each grid index cell. In order to test the robustness of our algorithm, we apply the same slope segmentation threshold for all of these regions. As it can be observed in figure 3, most of the filtering results in these sites are very satisfactory. Large buildings in samp 12, samp 22, samp 31, samp 41 and samp 42 are filtered out. In samp 11, samp 23, samp 24 and samp 41, Error Type II needs to be improved. This means that in these regions, several ground points are classified as non-ground points. Remarkable is samp 11, where the scene is complex with steep slopes and a mixture of vegetation and buildings on hillsides. The comparison to the eight existing algorithms reveals better results. Problems arise for two samples: in samp 23 and samp 41, large buildings exist; however, in our algorithm, some ground areas are rejected. We interpret this to be due to the rule of the sharp boundary points.

For the six rural regions (figure 4, samp 51–samp 54, samp 61 and samp 71), the density of laser points is much lower than that of the nine urban regions. In order to be able to index enough points in each grid index cell, we apply a 3 m grid size for the grid index in each of these samples. There are about 1.5 points for each grid index cell. As seen in table 3 and figure 4, most areas are well classified. This is especially true for samp 53, samp 54 and samp 61. Although there are many break lines in samp 53, the OBA algorithm still seems to be appropriate to be applied to this region.

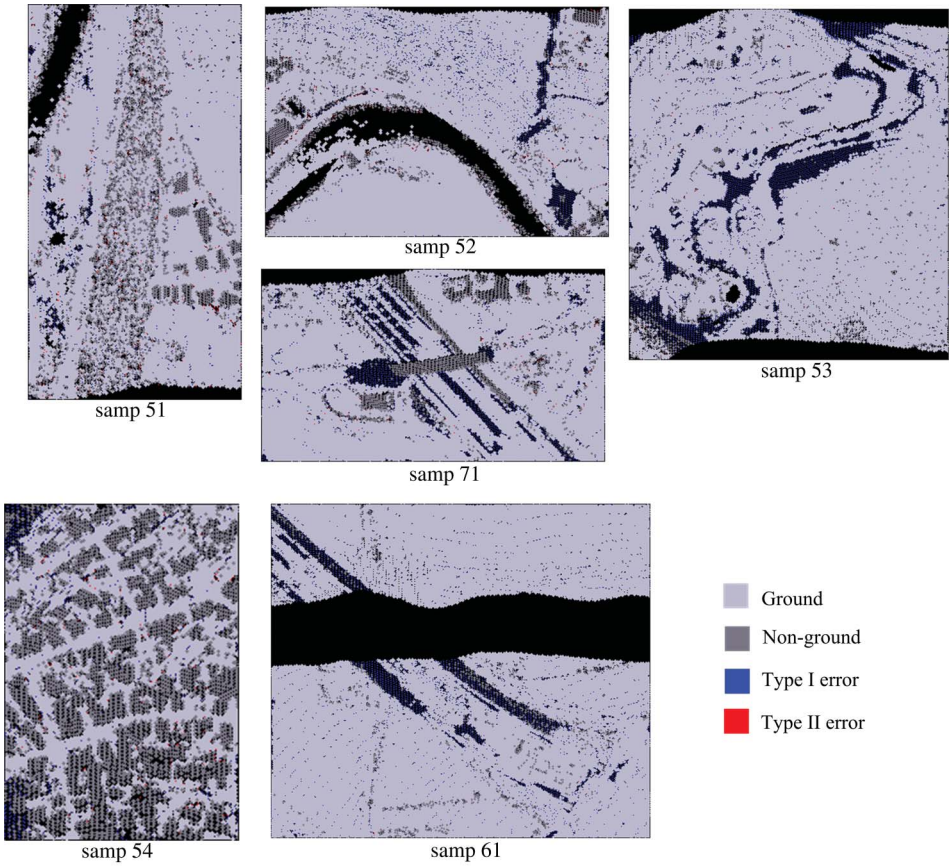

Figure 4. Error distribution for rural sites 5–7 displayed on the same scale.

As mentioned earlier, the results produced by the eight existing algorithms for these test sites are documented in Sithole and Vosselman (2003). We compare our new algorithm to these existing algorithms with regard to three different aspects: Error Type I, Error Type II and Error Total. As can be concluded from figures 5–7, the OBA method implemented in this article seems to achieve good results without parameter optimization. Figure 5 demonstrates that Error Type I is generally very different between all nine algorithms (the eight existing algorithms plus the OBA algorithm). All algorithms encountered problems with samp 41 and samp 53, while for all other samples, the errors are very diverse. Figure 5 also reveals that the algorithm of Axelsson (2000) performs best in most cases, while the OBA algorithm typically ranks third or fourth. Figure 6 shows that the algorithm of Roggero (2001) achieves good results for samp 22, samp 23 and samp 24, and especially for samp 21. On the contrary, the errors of the Axelsson algorithm change for these particular samples compared to Error Type I. The OBA algorithm ranks third or fourth in most of the cases. However, the results of samp 11, samp 21 and samp 52 need to be improved.

With regard to Error Type II, shown in figure 6, the results need to be improved for several of the test data sets, for instance for samp 21 and samp 52. However, we

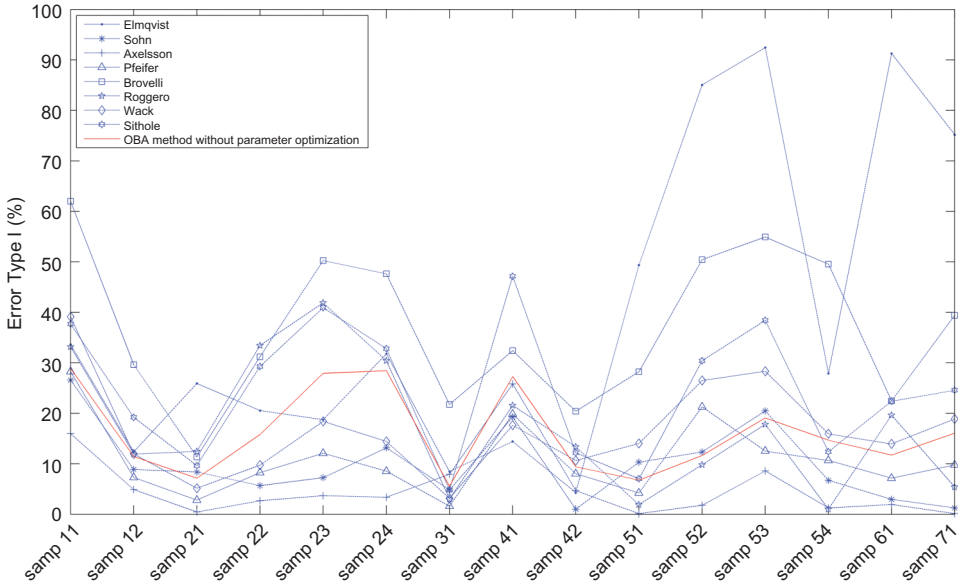

Figure 5. Comparison between existing algorithms and OBA methods of Error Type I.

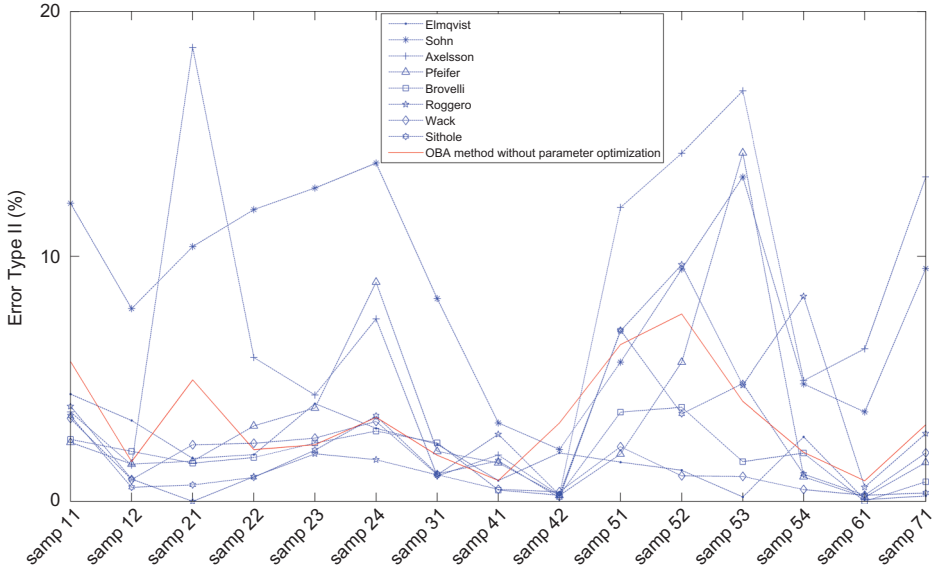

Figure 6. Comparison between existing algorithms and OBA methods of Error Type II.

do not apply any scene-wise optimization of the parameters for our algorithm. If we were to optimize the parameters for each data set separately, even better results would be achieved. We assume that it is usually difficult to find optimized parameters for practical DTM generation without reference data sets and without having particular applications in mind. Finding optimized parameters is usually labour-intensive and time-consuming. In this test, we use only two different settings for the pre-defined

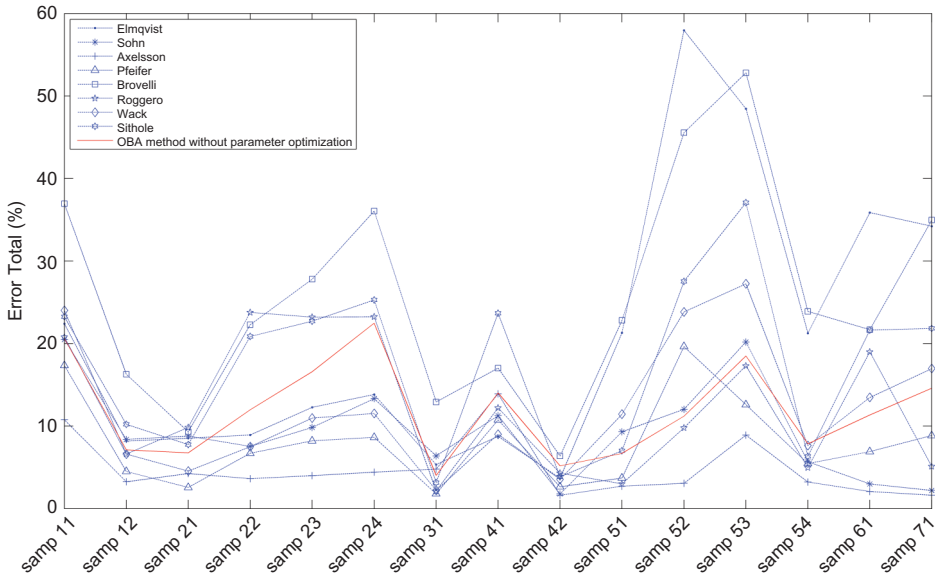

Figure 7. Comparison between existing algorithms and OBA methods of Error Total.

urban and rural test data sets. In summary, we can state that the OBA algorithm delivers very stable results in respect to Error Type I, Error Type II and Error Total (see figures 5–7).

5. Conclusions

Filtering is a very important task in ALS data processing. Although several methods tackle this problem, we conclude that it has not been fully solved. Most of the existing algorithms use a global scale and are in need of iteration. We developed a new filtering method based on OBA, which can utilize information derived from objects rather than from individual points. From the tests conducted using the ISPRS data sets, we conclude that the OBA algorithm is stable and yields good results without parameter optimization. With increasing laser point density, more and more structural information can be revealed and used to identify objects. We conclude that with a further increase in point densities, the OBA method becomes even more advantageous.

We tested our algorithm on various ISPRS data sets and compared it to eight existing algorithms. An analysis of the results revealed several advantages of our algorithm. First, based on the grid index, our algorithm could tackle high-density cloud points, for instance, the Riegl laser equipment, which could obtain the point cloud with a high density of 40 points m^{-2} (http://www.riegl.com/nc/products/airborne-scanning/). With our algorithm, we were able to process the cloud points at different resolution levels. Second, the new algorithm managed to utilize the information of derived objects, including topology, size, boundary and other shape information; therefore, it can derive more information from original point clouds than the methods based on points. Third, our algorithm can obtain more results from original point clouds, not only ground points or DTM. For instance, as illustrated in figure 8, the OBA algorithm can derive the boundaries of buildings, which could be used for building

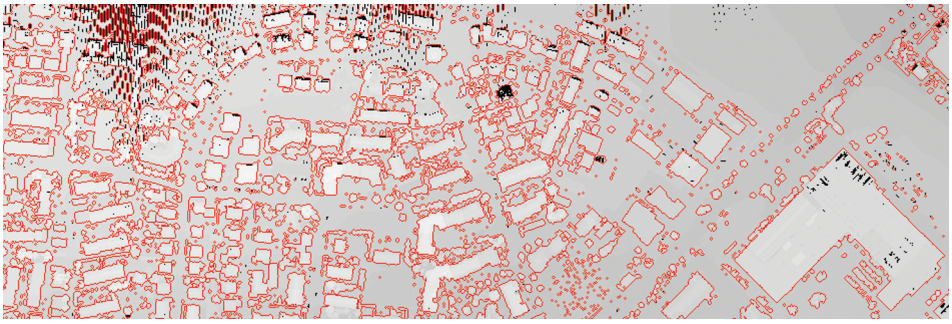

Figure 8. Building boundary extraction based on OBA method.

extraction. Finally, the OBA algorithm produces stable results in most of the cases without parameter optimization and iteration. The results reveal the potential of the OBA method for the filtering of ALS point clouds.

References

Ackermann, F., 1999, Airborne laser scanning – present status and future expectation. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**, pp. 64–67.

Axelsson, P., 2000, DEM generation from laser scanner data using adaptive TIN models. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 16–23 July 2000, Amsterdam, The Netherlands. Available online at: http://www.isprs.org/publications/archives.aspx

Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I. and Heynen, M., 2004, Multiresolution object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing*, **58**, pp. 239–258.

Blaschke, T., 2010, Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, **65**, pp. 2–16.

Brovelli, M.A., Cannata, M. and Longoni, U.M., 2002, Managing and processing LiDAR data within GRASS. In *Proceedings of the Open Source GIS-GRASS Users Conference*, 11–13 September 2002, Trento, Italy.

Cheng, H.D., Jiang, X.H., Sun, Y. and Wang, J.L., 2001, Color image segmentation: advances and prospects. *Pattern Recognition*, **34**, pp. 2259–2281.

Elmqvist, M., 2002, Ground surface estimation from airborne laser scanner data using active shape models. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 9–13 September 2002, Graz, Austria. Available online at: http://www.isprs.org/publications/archives.aspx

Guo, Q., Kelly, M., Gong, P. and Liu, D., 2007, An object-based classification approach in mapping tree mortality using high spatial resolution imagery. *GIScience and Remote Sensing*, **44**, pp. 1–24.

Hyyppä, J., Hyyppä, H., Litkey, P., Yu, X., Haggren, H., Roennholm, P., Pyysalo, U., Pitkaenen, J. and Maltamo, M., 2004, Algorithms and methods of airborne laser-scanning for forest measurements. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3–6 October 2004, Freiburg, Germany. Available online at: http://www.isprs.org/publications/archives.aspx

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

Li, P.J. and Xiao, X.B., 2007, Multispectral image segmentation by a multichannel watershed-based approach. *International Journal of Remote Sensing*, **28**, pp. 4429–4452.

Li, X., 1996, A method to improve classification with shape information. *International Journal of Remote Sensing*, **17**, pp. 1473–1481.

Liu, H.X., Wang, L., Sherman, D., Gao, Y. and Wu, Q.S., 2010, An object-based conceptual framework and computational method for representing and analyzing coastal morphological changes. *International Journal of Geographical Information Science*, **24**, pp. 1015–1041.

Liu, Y., Guo, Q.H. and Kelly, M., 2008, A framework of region-based spatial relations for non-overlapping features and its application in object based image analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, **63**, pp. 461–475.

Lloyd, C.D. and Atkinson, P.M., 2002, Deriving DSMs from LiDAR data with kriging. *International Journal of Remote Sensing*, **23**, pp. 2519–2524.

Meng, X., Currit, N. and Zhao, K., 2010, Ground filtering algorithms for airborne LiDAR data: a review of critical issues. *Remote Sensing*, **2**, pp. 833–860.

Meng, X., Wang, L., Silván-Cárdenas, J.L. and Currit, N., 2009, A multidirectional ground filtering algorithm for airborne LIDAR. *ISPRS Journal of Photogrammetry and Remote Sensing*, **64**, pp. 117–124.

Mitra, P., Shankar, B.U. and Pal, S.K., 2004, Segmentation of multispectral remote sensing images using active support vector machines. *Pattern Recognition Letters*, **25**, pp. 1067–1074.

Pfeifer, N., Reiter, T., Briese, C. and Rieger, W., 1999, Interpolation of high quality ground models from laser scanner data in forested areas. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 9–11 November 1999, La Jolla, CA. Available online at: http://www.isprs.org/publications/archives.aspx

Roggero, M., 2001, Airborne laser scanning-clustering in raw data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 22–24 October 2001, Annapolis, MD. Available online at: http://www.isprs.org/publications/archives.aspx

Saha, S. and Bandyopadhyay, S., 2010, Application of a multi seed-based clustering technique for automatic satellite image segmentation. *IEEE Geoscience and Remote Sensing Letters*, **7**, pp. 306–308.

Shan, J. and Sampath, A., 2005, Urban DEM generation from raw LIDAR data: a labeling algorithm and its performance. *Photogrammetric Engineering and Remote Sensing*, **71**, pp. 217–226.

Shih, F.Y., 2010, *Image Processing and Pattern Recognition: Fundamentals and Techniques*, pp. 157–158 (New Jersey: Wiley-IEEE Press).

Sithole, G., 2001, Filtering of laser altimetry data using a slope adaptive filter. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 22–24 October 2001, Annapolis, MD. Available online at: http://www.isprs.org/publications/archives.aspx.

Sithole, G. and Vosselman, G., 2003, Report: ISPRS comparison of filters. Available online at: http://www.itc.nl/isprswgiii-3/filtertest/Report05082003.pdf (accessed 5 April 2011).

Sithole, G. and 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**, pp. 85–101.

Sohn, G. and Dowman, I.J., 2002, Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 9–13 September 2002, Graz, Austria. Available online at: http://www.isprs.org/publications/archives.aspx

Sohn, G. and Dowman, I.J., 2007, Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, **62**, pp. 43–63.

Vosselman, G., 2000, Slope based filtering of laser altimetry data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 16–23 July 2000,

Amsterdam, The Netherlands. Available online at: http://www.isprs.org/publications/archives.aspx

Wack, R. and Wimmer, A., 2002, Digital terrain models from airborne laser scanner data – a grid based approach. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 9–13 September 2002, Graz, Austria. Available online at: http://www.isprs.org/publications/archives.aspx

Wagner, W., Hollaus, M., Briese, C. and Ducic, V., 2008, 3D vegetation mapping using small-footprint full-waveform airborne laser scanners. *International Journal of Remote Sensing*, **29**, pp. 1433–1452.

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

Zhang, K. and Whitman, D., 2005, Comparison of three algorithms for filtering airborne LIDAR data. *Photogrammetric Engineering and Remote Sensing*, **71**, pp. 313–324.