Automatic Weighted Splines Filter (AWSF): A New Algorithm for Extracting Terrain Measurements From Raw LiDAR Point Clouds

Ziad Abdeldayem

***Abstract*—Airborne raw light detection and ranging (LiDAR) measurements are georeferenced three-dimensional coordinates of ground surface, including all natural and man-made features. Extracting terrain surface measurements from raw LiDAR data is referred to as "filtering." Many filtering algorithms have been published, indicating the difficulty of the task; however, none performs equally well on all kinds of landscapes. This article presents a new algorithm, automatic weighted splines filter (AWSF), to extract the terrain points from raw LiDAR measurements. The mathematical model of the AWSF algorithm utilizes both the cubic smoothing splines (to interpolate and fit raw LiDAR data) and z-shaped function (to estimate the weight value of each point). The AWSF algorithm performance is compared against 14 filtering algorithms published between 1998 and 2019, as well as one unpublished (proprietary) algorithm designed by the world-leading company in processing remote sensing data, Harris Geospatial Solutions. Diverse landscape scenarios are used, ranging from open fields, rural land, and urban areas to dense forests on mountains where most of the filtering algorithms struggle as these areas represent the most difficult and challenging landscapes because the canopy prevents LiDAR pulses from reaching the ground surface. A total of 19 samples were tested; the results clearly show that the filtered terrain measurements are accurate and that the performance of the AWSF algorithm is stable for all the LiDAR data samples in comparison with the other filtering approaches.**

***Index Terms*—Algorithm, digital terrain model (DTM), filtering, forest, light detection and ranging (LiDAR), smoothing splines, urban.**

I. Introduction

HIGH-RESOLUTION digital terrain models (DTMs) can be generated from raw airborne light detection and ranging (LiDAR) data [1]. LiDAR technology is an active remote sensing system that calculates the distance between the laser scanner and the ground surface by measuring the two-way travel time of the emitted laser pulses [2], [3]. Raw LiDAR point clouds are very dense, irregularly distributed, and geo-referenced three-dimensional (3-D) measurements [4], [5].

LiDAR points could represent any natural or man-made feature on the ground surface, such as vehicles, trees, buildings, roads, power lines, etc. [6]. Removing all the off-terrain measurements (nonground points) to extract only the terrain measurements (ground points) from raw LiDAR data is referred to as "filtering" [7]–[9]. However, filtering LiDAR data in dense forests are still a challenge as the canopy prevents the emitted pulses from reaching the ground surface. Therefore, specialized approaches have been developed specifically for this purpose such as [10]–[14]. Several approaches for filtering raw LiDAR point clouds have been presented in the last two decades [15]. They can be categorized into three main groups: morphology-based, segmentation-based, and interpolation-based methods.

The first category uses the morphological processes such as erosion, dilation, and opening to filter raw LiDAR data [16]–[19]. The concept of these approaches is simple; it is based on comparing the elevation values within specific predefined horizontal distance (window) [20]. The morphological opening process can remove the measurements, which have higher elevations but retain the lower elevation measurements. This process is repeated using different horizontal distance values that should start from a small value and increase gradually through the iterative steps [21], [22]. Liu and Lim [23] presented a voxel-based multiscale morphological filtering method that applies a convexity constraint to LiDAR measurements. Then, the morphological operations are applied with a moving window to classify the data into terrain and off-terrain points after applying Otsu segmentation with a threshold value as a last step.

The segmentation-based group of filters deals with LiDAR point clouds as a raster image (interpolated LiDAR measurements). The whole raster LiDAR data are segmented and processed after identifying the segmentation parameters such as scale and weight for each layer [20]. These methods try to identify the segments of measurements raised higher than the surrounding segments, and then classifying the high segments as off-terrain measurements and the low ones as terrain measurements [24]. To improve the quality of the resulting data, additional parameters such as curvature and height difference, low and high vegetation, and smooth and planar surfaces can be used [25]. However, the segmentation-based methods are efficient for urban areas, but they struggle and fail in rural and forested areas due to the difficulty in identifying a distinct segment of measurements [26].

Manuscript received March 4, 2019; revised June 12, 2019, September 7, 2019, and October 16, 2019; accepted October 19, 2019.

The author is with the Department of Civil Engineering, Liverpool John Moores University, Liverpool L3 5UG, U.K. (e-mail: z.abdeldayem@ljmu.ac.uk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2019.2950600

The third category is based on interpolating the ground surface by using the whole LiDAR dataset [6]. Usually, the interpolation-based methods work on predicting the terrain surface by using a linear function to interpolate the data and a weight value could be assigned to the measurements [27]. Brovelli *et al.* [28] used bicubic and bilinear functions to interpolate the data, the residuals' histograms are computed as a step for estimating the threshold value. In addition, the gradient magnitude of the surface is calculated for improving the performance of the method. Kraus and Pfeifer [11] utilized the linear prediction for interpolating the randomly distributed LiDAR data using some ground control points. Then, the weights of the data points are calculated to enhance the classification of the terrain and nonterrain measurements. Chen *et al.* [29] presented a method that starts by identifying the last-return LiDAR measurements. Then, the prediction of the terrain surface is done iteratively.

Some filters are designed to process only the forested areas [3], [29], [30], and others to filter the urban landscapes containing large buildings [2], [31]. Silva *et al.* [32] compared the performance of some airborne LiDAR data filtering algorithms in forested areas. The algorithms are weighted linear least-squares, multiscale curvature classification, progressive morphological filter, and progressive triangulated irregular network. Nevertheless, there is a group of filtering approaches that does not belong to any of the abovementioned categories, for example, the triangulated irregular network-based methods [33]–[35] and the statistical-based methods [36]–[38]. Hui *et al.* [39] suggested that a filtering process starts by removing the outlier points, then estimating the ground surface by fitting the data using a quadratic polynomial function. After that, the method assumes that LiDAR data represent a mixture of Gaussian models, which can be used for refining the classification of the data into terrain and off-terrain measurements. Although the algorithm endeavored to automate the process of extracting the ground points, its performance was moderate in comparison with the other published work. Cai *et al.* [40] filtered the data through three main stages. In the first stage, the algorithm generates an initial ground surface estimation by applying a cloth simulation for identifying some ground control points. Then, the parameters' values are estimated using statistical analysis. Finally, the progressive TIN densification process is applied for extracting the final terrain measurements. Cai *et al.* [41] presented a model transfer-based filtering methodology that combines transfer learning theory with active learning for identifying the terrain points. It requires a training dataset as a core for this method. Bayram *et al.* [42] exploited weighted graph representations for the analysis of LiDAR data. This method considered the data as irregularly distributed signals and applied an iterative graph signal filter to remove the off-terrain measurements from the raw data.

A new algorithm for filtering raw LiDAR data is presented in this article. The proposed automatic weighted splines filter (AWSF) belongs to the interpolation-based filters. Unlike the other interpolation-based methods, which mostly use linear prediction or bicubic functions to interpolate the data, the AWSF has exploited, for the first time, the cubic smoothing splines function, which is very effective for such noisy data, as the smoothing parameter controls the fitted curve and leads to reserving the terrain measurements efficiently. Also, the z-shaped function is used to vigorously assign a weight value for each measurement. More significantly, the AWSF is designed to process raw LiDAR data with different densities and diverse landscapes that range from open fields, roads, railroads, power lines, buildings, rivers, vegetated areas to heavily forested areas. The detailed methodology is described in part 2 of this article, whereas parts 3 and 4 present the results and conclusions of the research.

II. Automatic Weighted Splines Filter (AWSF)

The proposed filter is based on processing raw LiDAR data by using cubic smoothing splines and z-shaped functions. A mathematical background about the smoothing spline interpolation is presented in the following section. After that, the AWSF processing stages are explained.

A. Cubic Smoothing Splines Interpolation

A cubic spline interpolation function $f(x)$ is a series of different cubic polynomials on each interval $[\xi_i, \xi_{i+1}]$, such that the derivatives of $f(x)$ are continuous up to the second order. Let the coordinates of the ith measurement be $(x_i, y_i, z_i),\ i = 1, 2, \ldots, n$, where n is the number of the interpolated measurements. The form of a cubic spline $f(x)$ on each interval is [43]

$$z_i = f_i\left(x\right) = a_i + b_i\left(x - x_i\right) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{1}$$

where $a_i,\ b_i,\ c_i,$ and d_i are the coefficients that can be determined from the definition of cubic splines. Continuity and interpolation require that $f_i(x_{i+1}) = f_{i+1}(x_{i+1})$. Also, at each interval node of a cubic spline, the first and second derivatives should have the same values; $f'_i(x_{i+1}) = f'_{i+1}(x_{i+1})$ and $f''_i(x_{i+1}) = f''_{i+1}(x_{i+1})$. In addition to the abovementioned equation, we can benefit from the properties of the "natural spline" function by considering the second derivatives at the end nodes equal to zero; $f''_1(x_1) = f''_{n-1}(x_{n-1}) = 0$. All the previous conditions and properties will help to determine the values of the coefficients $a_i,\ b_i,\ c_i,$ and d_i. After some manipulation, we get [43], [44]

$$\begin{aligned} t_{i-1}c_{i-1} + 2\left(t_{i-1} + t_i\right)c_i\ + t_ic_{i+1} \\ = 3\left(\frac{(z_{i+1} - z_i)}{t_i} - \frac{(z_i - z_{i-1})}{t_{i-1}}\right),\ i = 2,\ 3, \ldots, n-2 \end{aligned} \tag{2}$$

where $t_i = x_{i+1} - x_i$. Once the coefficients c_i are determined, from (2), the remaining coefficients could be found simply as follows [43], [44]:

$$\begin{aligned} a_i &= z_i \\ b_i &= \frac{(z_{i+1} - z_i)}{t_i} - \frac{\left(2c_i + c_{i+1}\right)t_i}{3} \\ d_i &= \frac{(c_{i+1} - c_i)}{3t_i}. \end{aligned} \tag{3}$$

There are stable mathematical methods for determining the coefficients $a_i,\ b_i,\ c_i,$ and d_i and solving such systems. For

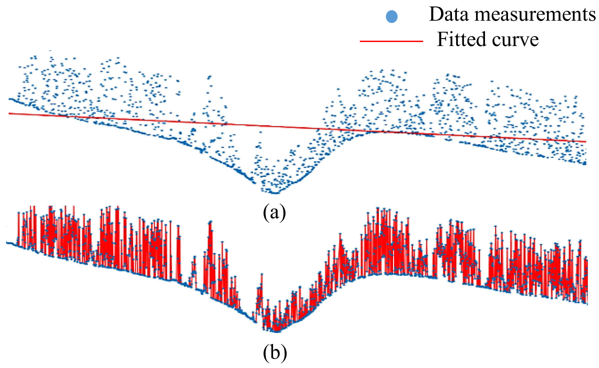

Fig. 1. Data measurements and the fitted curve with the smoothing parameter value of (a) $\alpha = 0$ and (b) $\alpha = 1$.

more details, see [45]–[47]. For some applications, it might be required not to interpolate the data measurements exactly, but to create kind of smooth spline, which aims to pass close to the measurements but not necessarily through them. Cubic smoothing spline functions represent the solution, by minimizing [47]

$$\alpha \sum_{i=1}^{n} w_i (z_i - f(x_i))^2 + (1-\alpha) \int_{x_1}^{x_n} \{f''(x)\}^2 dx \tag{4}$$

where w_i is the weight for the ith measurement, and α is the smoothing parameter.

Criterion (4) consists of two terms, the first one measures how closely the function fits the data points, and the second one measures the smoothness of the fitted function. The smoothing parameter value ranges between 0 and 1 [47]. When $\alpha = 0$, the resultant fit would be a straight line due to integrating the second derivative of a cubic function, whereas choosing $\alpha = 1$ leads to a cubic spline interpolation, as illustrated in Fig. 1, because the second part of (4), which smooths the resultant curve, is eliminated. Minimization of (4) could produce a smooth fitted curve that follows the general trend of the measurements but does not interpolate them exactly. The AWSF algorithm exploits this significant property of the cubic smoothing splines function to help in filtering raw LiDAR data.

B. *AWSF Algorithm*

The AWSF algorithm introduces an automatic filter for identifying and removing the off-terrain measurements from raw LiDAR data. MATLAB codes have been developed to implement all the processing stages described in the following sections.

1) Raw LiDAR Data Gridding: Raw LiDAR point clouds can be described as a random set of measurements covering the scanned area. Therefore, the first step of the filtering process would be converting these randomly distributed data into a uniform grid [18]. To generate the grid, a transformation between the irregular raw LiDAR measurements and the regularly spaced grid is applied. The dimensions of the grid—*m* rows and *l* columns—can be determined as follows [18]:

$$m = \left(\frac{y_{\max} - y_{\min}}{G_s}\right) + 1$$
$$l = \left(\frac{x_{\max} - x_{\min}}{G_s}\right) + 1 \tag{5}$$

where G_s is the grid cell size (spatial resolution). The selection of the G_s value depends on LiDAR data density. The denser the raw data, the smaller the grid cell size. The value assigned to each grid cell is the coordinates of the LiDAR measurement that falls into that cell. However, because of the irregular distribution of the measurements, some grid cells have no LiDAR measurements, while other cells have more than one. In the latter case, when there is more than one LiDAR measurement, the AWSF algorithm assigns the grid cell the measurement with the lowest elevation $(z - \text{coordinate})$ as a first attempt to establish the terrain surface. Consequently, the created 2-D array $A\{m,\ l\}$ contains either empty cells or the coordinates $(x,\ y,\ \text{and}\ z)$ of the LiDAR points.

2) Data Normalization, Fitting, and Residual Computation: The aim of the normalization is to improve the accuracy of the numerical measurements before creating the fitted curve. To normalize a set of data, the measurements should be centered at zero mean and scaled to the standard deviation value of the measurements as follows [48]:

$$Z_{\text{normalized}} = \frac{z_i - \mu_z}{\sigma_z} \tag{6}$$

where $Z = z_1,\ z_2, \ldots, z_n$, μ_z is the mean value of Z, and σ_z is the standard deviation of Z. For each row of the LiDAR dataset A, (6) is applied to normalize the data. After that, Criterion (4) is applied to fit (X, Z) data using a cubic smoothing spline function, where $Z = z_1,\ z_2, \ldots, z_n$. At this stage of processing, it is important to determine the smoothing parameter value α, which varies from 0 to 1. For the purpose of LiDAR data filtering, it is desired that the fitted curve passes close to the data measurements; $\alpha \approx 1$. Many tests were conducted to find the optimal value of the smoothing parameter, it is found that the α value should range from 0.9999 (for forested areas) to 0.99 (for all other areas). The fitted curve follows the general trend of the terrain and off-terrain LiDAR measurements without necessarily passing through each point. The difference between the elevation of each LiDAR measurement $z_{i\,(l)}$ and the elevation of the fitted curve $z_{i\,(f)}$ is called the residual value v_i. The residuals for all LiDAR measurements are computed using (7); these values will then be used to assign a weight to each LiDAR measurement [11] as

$$v_i = z_{i(l)} - z_{i(f)} \tag{7}$$

Fig. 2 shows a sample LiDAR data, the fitted curve, and the residual values. Obviously, LiDAR measurements with negative residual values lie below the fitted curve, and hence, more likely to be terrain measurements, whereas, on the other hand, points with positive residual values lie above the fitted curve, and therefore, more likely to be off-terrain measurements.

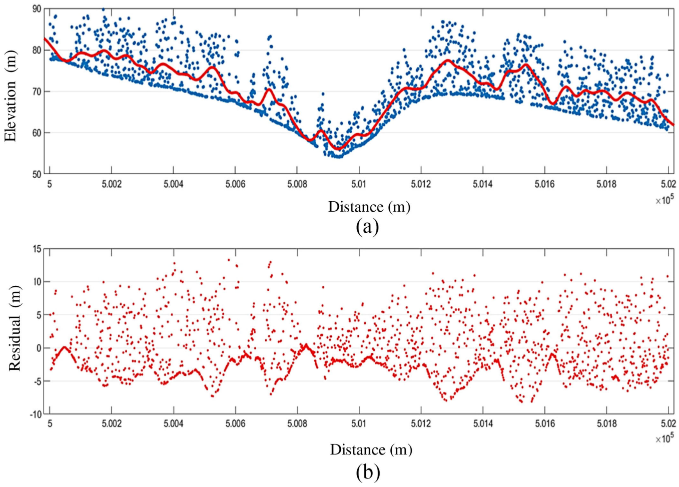

Fig. 2. Fitting of raw LiDAR data using cubic smoothing splines curve and computing the residuals. (a) Dots represent the LiDAR measurements (elevation versus distance), and the solid line is the fitted curve. (b) Residual values.

3) Computing the Weights by Using z-Shaped Function: In the first iteration, the curve is fitted with equal weights for all the z_i values. A z-shaped function is used to estimate the weights w_i as follows [49]:

$$w_i = \begin{cases} 1, & \upsilon_i < \sigma \\ 1 - 2\left(\dfrac{\upsilon_i - \sigma}{\tau - \sigma}\right)^2, & \sigma \le \upsilon_i \le (\sigma + \tau)/2 \\ 2\left(\dfrac{\tau - \upsilon_i}{\tau - \sigma}\right), & (\sigma + \tau)/2 \ \le \upsilon_i \le \tau \\ 0, & \upsilon_i \ge \tau \end{cases} \tag{8}$$

where σ and τ are the shift and threshold values, respectively. σ is the negative value of the standard deviation of the residuals υ_i. The measurements that have a residual value less than σ will be assigned a large weight value of 1 as they are more likely to be classified as terrain measurements.

4) Removing Off-Terrain Measurements: The threshold value τ is used for determining the candidate measurements to be removed. All the LiDAR measurements with residual values υ_i more than τ would be considered as off- terrain measurements and removed. The initial threshold value τ_o should be a small value so that the algorithm can remove as much as possible of the off-terrain measurements. For forested areas $\tau_o = 0.25$ and for all other areas $\tau_o = 0.50$. The weights w_i, which are computed by applying the z-shaped function described in (8), will then be used in a second iteration. LiDAR measurements having large negative residual values will be allocated large weight values, and consequently they attract the fitted curve toward them, whereas measurements that have positive or small negative residuals will be given smaller weight values to minimize their effect on the fitted curve. Fig. 3 illustrates that if the measurement has a residual value less than σ, then its weight will be 1 and, therefore, has more influence on the computations of the curve fitting. On the other hand, if the measurement has a residual value more than τ, then its weight will be 0, and therefore, it will be removed. The weights increase gradually from 0 to 1 when the residual values decrease from τ to σ. The previous processing steps are repeated for each row of the raw LiDAR dataset A.

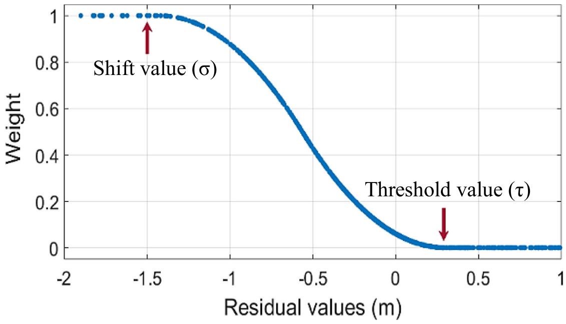

Fig. 3. Z-shaped weight function.

Algorithm 1: Removing Negative Blunders.

```
For each row of $A$
    Determine the threshold value $\tau_b = -3\sigma_r$
    Normalise the data using (6)
    Fit the data using cubic smoothing splines function;
    Criterion (4)
    Compute the residual values $\upsilon_i$ using (7)
    If      $\upsilon_i < \tau_b$
        Classify the LiDAR measurement as "negative
         blunder"
    Else
        Classify the LiDAR measurement as "terrain"
    End
        Remove all the measurements classified as negative
        blunder
End
```

5) Removing Blunders: Generally, there are two kinds of blunders in LiDAR data, positive blunders and negative blunders. Commonly, positive blunders emerge from laser returns from birds, aircraft, etc., and negative blunders come from pulses that are reflected several times or malfunction of a laser rangefinder [50]. Positive blunders can be removed by completing the first iteration over all the rows of A. However, negative blunders still exist. Nie *et al.* [51] suggested a three-step method to remove the negative blunders. In the first and second steps, the method tries to automatically remove the blunders; after that it requires manual intervention to remove the rest of the measurements, which cannot be removed automatically. The AWSF introduces an automatic approach to overcome the issue of manual editing by classifying the measurement, which has a residual value υ_i less than $-3\sigma_r$ (σ_r is the standard deviation of the residuals) as a negative blunder. In other words, if a point lies below the fitted curve by more than $3\sigma_r$, it will be classified as a negative blunder and removed from the data, as illustrated in Algorithm 1.

6) Iterations and Terrain Measurements Extraction: The weight w_i of each retained measurement is used in the subsequent iterations to compute a new fitted curve. To improve the results, the columns of the LiDAR dataset A could be filtered

```
Algorithm 2: Raw LiDAR Data Filtering Using the AWSF Algorithm.
Input data: Raw LiDAR data, $R$
Results: Filtered LiDAR data; terrain_points.txt (ASCII file)
Begin
    Load R
    Determine the grid cell size; $G_s$
    Create an empty grid; $A\{m, l\}$
    For each grid cell of $A$
        Cell value = $\min(z)$
    End
    Determine the smoothing parameter and the initial
    threshold values
    If the scanned area is forested area
        $\alpha = 0.9999$ & $\tau_o = 0.25$
    Else
        $\alpha = 0.99$ & $\tau_o = 0.50$
    End
    For each row/column of $A$
        Normalise the data using (6)
        Fit the data using cubic smoothing splines curve;
        Criterion (4)
        Compute the residuals' values:
                $v_i = z_{i(l)} - z_{i(f)}$
        Compute the weights' values $w_i$ using (8)
        If    $w_i = 0$
            Classify the LiDAR measurement as
            "off-terrain"
        Else
            Classify the LiDAR measurement as
            "terrain"
        End
        Remove all the measurements that are classified as
        off-terrain points from $R$
        Remove negative blunders using Algorithm 1
        -only at the second iteration.
        Repeat the process using the recommended threshold
        Values: $\tau_j = [7,6,5,4,3,2,1]$
    End
    Output data: Filtered LiDAR measurements; terrain_points.txt.
End
```

by applying the previous steps. After filtering all the rows and columns for the first iteration, τ values should be decreased gradually from large to small in the following iterations. It is preferable to start the second iteration using large τ value to stay in the safe range and not to remove any terrain measurement. After implementing many tests on raw LiDAR data with different densities covering various kinds of areas, it was clear that starting the second iteration with a threshold value of 7 (or greater), is sufficient. Therefore, the following values are used for all the study areas $\tau = [7, 6, 5, 4, 3, 2, 1]$. Finally, as illustrated in Algorithm 2, all the previous processing stages should be applied for each value of τ. Fig. 4 illustrates the iterating process to remove the off-terrain LiDAR measurements and extract the terrain points. The preserved measurements are gathered in one file and classified as terrain measurements. Algorithm 2 illustrates the main stages of raw LiDAR data filtering using the AWSF algorithm. Basically, after loading the raw LiDAR data, it is required to provide the grid cell size and the type of the scanned area (forest/other).

Therefore, there is no need to define the smoothing parameter and the threshold values as they are predefined and fixed along all the processing stages. However, the algorithm allows the user to test a sample data of the area of interest and adjust the parameter values easily to enhance the results.

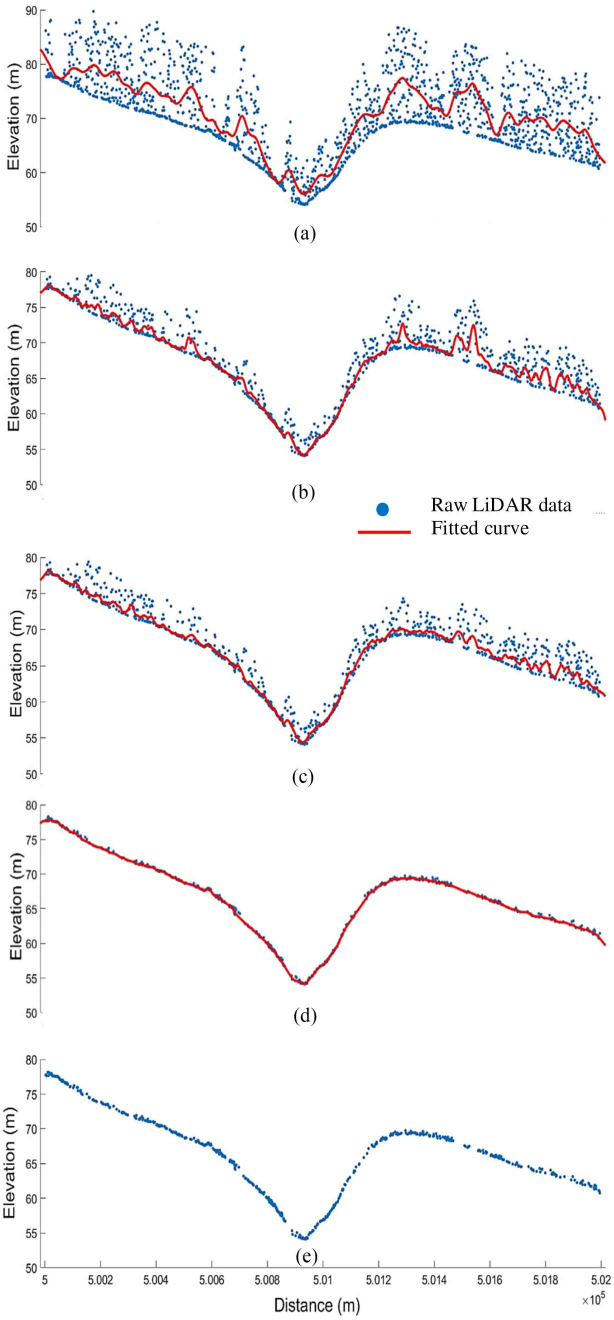

Fig. 4. Iterations and removal of off-terrain measurements to extract the terrain measurements. (a) First iteration: LiDAR data measurements (the dots) and fitted curve (the solid line). (b) Second iteration after removing the negative blunders and all the measurements having the weight 0. (c) Third iteration. (d) Last iteration. (e) Reserved measurements, i.e., the terrain points. Note: the sample data used in this figure are for a forested area (Southwick, U.K.). In this case, the off-terrain measurements represent the trees. The same processing stages are applied for any other area (urban, rural, etc.).

III. Experiments and Results

The performance of the AWSF algorithm is tested using two different LiDAR datasets. The first one is the data provided by ISPRS commission 3/WG3, which includes 15 subsamples [8].

TABLE I
CHARACTERISTICS OF THE ISPRS DATASET SUBSAMPLES, SOURCE [55]

Area & Data Density	Subsample number		Features of interest
Urban 0.67 points/m²	Sample 1	Sub. 11	Vegetation and buildings on steep slopes
		Sub. 12	Small objects (cars)
	Sample 2	Sub. 21	Narrow bridge
		Sub. 22	Bridge and gangway
		Sub. 23	Complex and large buildings
		Sub. 24	Ramp
	Sample 3	Sub. 31	Disconnected terrain
	Sample 4	Sub. 41	Clump of negative blunders
		Sub. 42	Elongated objects
Rural 0.18 points/m²	Sample 5	Sub. 51	Vegetation on slope
		Sub. 52	Low vegetation
		Sub. 53	Discontinuity preservation
		Sub. 54	Low resolution buildings
	Sample 6	Sub. 61	Sharp ridge, ditches
	Sample 7	Sub. 71	Bridge

The error values of the filtered data, using the AWSF algorithm, were computed and compared against corresponding values (the same 15 subsamples) from other algorithms, which have been published between 1998 and 2019. The second dataset consists of four samples in heavily forested areas that represent a challenge for most of the filtering algorithms. DTMs were generated from the filtered data and compared pixel-by-pixel against very accurate reference DTMs. Both of the raw LiDAR data and reference DTMs were obtained through EDINA digimap LiDAR data service [52]. Additionally, the performance of the AWSF was compared against professional software designed by the world-leading company in processing remote sensing data, Harris Geospatial Solutions.

A. *ISPRS Dataset*

Seven samples of raw LiDAR data were collected to test the filtering algorithms presented by researchers from all over the world. These samples were divided into 15 subsamples based on the feature type within each subsample. The reference data for each subsample were extracted manually, and each measurement was labeled either terrain or off-terrain. Samples 1, 2, 3, and 4 represent urban areas with a raw LiDAR data density of 0.67 points/m², whereas samples 5, 6, and 7 cover rural areas with less density of the raw data; 0.18 points/m². Table I illustrates the type of feature included in each subsample.

The following equations were used to calculate two error types; the first type occurs when a filtering algorithm classifies genuine terrain measurements as off-terrain measurements. On the other hand, classifying off-terrain measurements as terrain measurements is the second type of error [53] calculated as

$$\text{Type 1 error}\,(\%) = \frac{N_g}{t_g} \times 100$$

$$\text{Type 2 error}\,(\%) = \frac{N_n}{t_n} \times 100 \qquad (9)$$

where t_g and t_n are the total number of terrain and off-terrain measurements, respectively. N_g is the number of the removed terrain measurements, and N_n is the number of the preserved off-terrain measurements [53]. Note that the higher the Type 2 error, the more the preserved off-terrain measurements. Hence, when a filtering algorithm fails in identifying and removing the off-terrain points, the generated DTM will be very badly affected. Therefore, most filters focus on minimizing this type of error rather than Type 1 error (removing some terrain measurements in error), as the elevations of the removed measurements can be estimated through the DTM generation process [54]. Therefore, the AWSF algorithm aims to minimize Type 2 error to avoid the consequences of reserving high percentages of off-terrain points, which are much more serious than those of Type 1 error. Tables II and III list the results of filtering the ISPRS LiDAR dataset using the AWSF algorithm compared against the other 14 algorithms' results.

The average Type 1 error using the AWSF method is 21.12%, whereas the average Type 1 error using the other filters is 15.45% and ranges from 1.84% to 39.43%. It is noticeable that Type 1 error values vary from one subsample to another. The reason behind these variations is the complexity or simplicity of the features contained in each subsample. For example, subsample 53 contains a quarry, which is a complex feature for any filtering algorithm aims to avoid leaving off-terrain points without filtering. The Type 1 error value has dropped from 53.63% for subsample 53% to 9.56% for subsample 54 as the main features in the previous subsample are small buildings, which might be considered simple and easy to identify by the filtering algorithms. Yet, the elevation values of the removed terrain points could be estimated straightforwardly throughout interpolating the surrounding values for generating the DTM. Obviously, nonfiltered off-terrain points (trees, buildings, etc.) will reduce the accuracy of the generated DTM, as they attract the surrounding values (terrain points) toward them (top of a tree, building's roof, etc.). More importantly, the results show very high accuracy in removing the off-terrain measurements from the raw data. The average Type 2 error using the AWSF algorithm is 1.44%, whereas the average Type 2 error value using the other filters is 5.72% and ranges from 1.58% to 16.11%. Also, the presented algorithm has succeeded in achieving a maximum Type 2 error value of 2.82%, which is the best (lowest value) in comparison with the other algorithms. The average value of the maximum Type 2 error of all the 14 algorithms is 17.63%. Table III illustrates clearly that the maximum Type 2 error of these algorithms ranges from 3.38% to 48.49%. In other terms, the presented filtering method produced the lowest percentage of the nonfiltered LiDAR measurements. This will significantly improve the quality of the generated DTM from the filtered data.

The error value for each subsample could be considered as an important indicator for the method's ability to deal with specific types of feature (big building, bridge, vegetation, slopes, etc.), it is imperative to evaluate the ability of any filtering algorithm for dealing with all kinds of areas and features.

Nevertheless, it is vital for judging any filtering algorithm to look at the average error value for all the subsamples as an indicator for the ability of the method to filter different kinds of areas while maintaining a low value of the error.

TABLE II
COMPARISON OF TYPE 1 ERROR VALUES (%) OBTAINED BY THE AWSF ALGORITHM AGAINST 14 FILTERING ALGORITHMS PUBLISHED DURING THE PAST TWO DECADES (1998–2019)

Subsample	Elmqvist [55]	Sohn [35], [55]	Axelsson [55]	Pfeifer [12], [55]	Brovelli [28], [55]	Roggero [55]	Wack [55]	Sithole [17], [55]	Hu [14]	Nie [51]	Pingel [23]	Liu [23]	Hui [39]	Cai [40]	AWSF
Sub. 11	33.63	26.56	15.96	28.26	62.00	33.16	39.12	37.69	6.32	37.24	7.88	31.83	19.94	7.51	41.26
Sub. 12	12.36	8.87	4.89	7.29	29.63	11.92	11.94	19.19	1.75	11.86	2.57	11.57	8.55	4.68	8.05
Sub. 21	25.91	8.38	0.46	2.81	11.35	12.46	5.15	9.64	0.36	6.20	0.26	9.25	1.79	16.27	6.26
Sub. 22	20.55	5.68	2.68	8.25	31.19	33.43	9.73	29.29	1.67	20.81	2.57	28.50	2.90	2.22	18.01
Sub. 23	18.74	7.25	3.69	12.08	50.25	41.88	18.40	40.92	4.73	35.63	3.21	17.57	2.23	3.48	22.08
Sub. 24	31.80	13.17	3.38	8.54	47.63	30.43	14.41	32.79	2.19	32.58	2.25	17.30	19.44	3.13	20.27
Sub. 31	8.47	4.81	7.91	1.60	21.75	3.03	3.15	4.85	0.50	2.02	0.39	12.41	0.21	12.74	5.12
Sub. 41	14.42	19.25	25.81	19.85	32.41	21.55	17.63	47.13	2.95	52.03	3.64	32.01	12.28	25.56	23.51
Sub. 42	4.30	1.01	4.68	8.02	20.40	13.37	10.65	12.18	0.39	6.69	0.27	13.06	11.65	9.71	11.40
Sub. 51	49.34	10.33	0.13	4.21	28.23	1.90	14.03	7.03	0.53	2.90	0.59	10.57	5.56	0.07	13.49
Sub. 52	85.05	12.34	1.78	21.27	50.43	9.80	26.49	30.41	1.07	16.14	3.09	21.38	12.05	0.98	39.97
Sub. 53	92.45	20.48	8.58	12.53	54.93	17.81	28.33	38.41	1.63	20.22	1.18	18.44	26.22	2.72	53.63
Sub. 54	27.91	6.72	1.25	10.66	49.54	1.01	15.93	12.38	2.33	6.76	2.51	7.46	8.46	1.16	9.56
Sub. 61	91.28	2.96	1.94	7.15	22.45	19.64	13.94	22.39	0.27	8.17	0.51	7.36	26.56	0.39	19.66
Sub. 71	75.19	1.26	0.14	9.78	39.41	5.41	18.88	24.57	0.93	5.24	0.99	17.05	5.84	0.28	24.49
Max	**92.45**	**26.56**	**25.81**	**28.26**	**62.00**	**41.88**	**39.12**	**47.13**	**6.32**	**52.03**	**7.88**	**32.01**	**26.56**	**25.56**	**53.63**
Min	**4.30**	**1.01**	**0.13**	**1.60**	**11.35**	**1.01**	**3.15**	**4.85**	**0.27**	**2.02**	**0.26**	**7.36**	**0.21**	**0.07**	**5.12**
Average	**39.43**	**9.94**	**5.55**	**10.82**	**36.77**	**17.12**	**16.52**	**24.59**	**1.84**	**17.63**	**2.13**	**17.05**	**10.91**	**6.06**	**21.12**

TABLE III
COMPARISON OF TYPE 2 ERROR VALUES (%) OBTAINED BY THE AWSF ALGORITHM AGAINST 14 FILTERING ALGORITHMS PUBLISHED DURING THE PAST TWO DECADES (1998–2019)

Subsample	Elmqvist [55]	Sohn [35], [55]	Axelsson [55]	Pfeifer [12], [55]	Brovelli [28], [55]	Roggero [55]	Wack [55]	Sithole [17], [55]	Hu [14]	Nie [51]	Pingel [23]	Liu [23]	Hui [39]	Cai [40]	AWSF
Sub. 11	4.38	12.17	3.65	2.41	2.53	3.88	3.38	3.49	10.98	1.34	8.81	3.75	13.30	27.98	1.68
Sub. 12	3.30	7.87	1.48	1.52	2.04	0.91	0.89	0.57	3.45	1.05	3.30	2.14	5.68	13.21	2.14
Sub. 21	1.75	10.40	18.53	1.64	1.56	0.00	2.31	0.67	3.03	4.49	4.07	1.53	5.23	6.87	0.90
Sub. 22	1.91	11.91	5.87	3.08	1.79	1.01	2.37	0.98	6.69	3.60	5.07	6.11	30.45	8.73	1.89
Sub. 23	3.99	12.79	4.34	3.81	2.38	1.94	2.58	2.09	4.07	1.60	6.17	4.18	16.27	14.13	2.82
Sub. 24	2.98	13.81	7.45	8.95	2.87	1.70	3.26	3.48	8.07	15.42	6.90	3.50	6.80	48.49	2.71
Sub. 31	2.33	8.28	1.03	2.04	2.39	1.08	1.09	1.12	1.37	2.41	1.52	3.43	14.57	0.91	0.96
Sub. 41	0.85	3.20	1.89	1.57	0.46	2.74	0.49	1.65	8.86	0.32	8.17	6.02	10.63	0.35	1.98
Sub. 42	1.98	2.12	0.26	0.24	0.25	0.26	0.39	0.15	0.87	1.26	1.98	1.76	1.02	2.71	1.37
Sub. 51	1.60	5.68	12.00	1.93	3.64	6.96	2.23	6.99	7.46	2.78	4.44	6.47	7.22	15.81	1.15
Sub. 52	1.27	9.48	14.21	5.68	3.84	9.66	1.04	3.57	14.90	2.96	10.08	6.14	8.64	35.90	1.86
Sub. 53	0.18	13.24	16.76	14.23	1.62	4.74	1.02	4.81	38.75	0.72	31.97	6.91	20.52	33.05	0.28
Sub. 54	2.63	4.79	4.93	1.00	1.97	8.37	0.48	1.11	2.36	1.77	2.05	3.37	3.07	3.81	1.27
Sub. 61	0.07	3.65	6.23	0.17	0.00	0.58	0.25	0.25	16.75	2.07	10.70	3.73	7.97	13.93	0.16
Sub. 71	0.22	9.51	13.25	1.59	0.80	2.77	1.98	0.34	5.93	0.79	6.84	2.20	9.94	15.71	0.45
Max	**4.38**	**13.81**	**18.53**	**14.23**	**3.84**	**9.66**	**3.38**	**6.99**	**38.75**	**15.42**	**31.97**	**6.91**	**30.45**	**48.49**	**2.82**
Min	**0.07**	**2.12**	**0.26**	**0.17**	**0.00**	**0.00**	**0.25**	**0.15**	**0.87**	**0.32**	**1.52**	**1.53**	**1.02**	**0.35**	**0.16**
Average	**1.96**	**8.59**	**7.46**	**3.32**	**1.88**	**3.11**	**1.58**	**2.08**	**8.90**	**2.84**	**7.47**	**4.08**	**10.75**	**16.11**	**1.44**

Moreover, the maximum Type 2 error value indicates whether the method's performance is stable with all areas or not. In other words, if a method performs well with some samples, but has a high error value with other samples, that indicates an unstable performance. The AWSF method's highest value of Type 2 error is 2.82% (Subsample 23), Table III lists that Nie's method has achieved the lowest value; 1.60% for this subsample. However, the value of Type 2 error using Nie's method increases significantly to 15.42% for Subsample 24, whereas the AWSF maintains a stable performance with Type 2 error value of 2.71% for Subsample 24. In addition, the values of Type 1 error for Subsamples 23 and 24 are better using the AWSF method, as more than 12% of the terrain points are preserved in comparison with Nie's method. Likewise, the maximum value of Type 1 error using the AWSF algorithm is 53.63% for Subsample 53. The lowest Type 1 error for this subsample 1.18% is achieved by applying Pingel's method. However, by comparing the values of Type 2 error for Subsample 53, it is clear that the AWSF method has minimized this value to 0.28%, whereas Pingel's method has classified 31.97% of the off-terrain measurements as terrain measurements, this will reduce the quality of the generated DTM significantly. Additionally, Table III lists that Elmqvist's method has achieved the lowest Type 2 error value for Subsample 53, which is 0.18%, but the method removed 92.45% of the terrain points (large Type 1 error).

Visual comparisons of the planimetric distribution of the LiDAR points before and after the filtering process have been conducted. Fig. 5 shows that raw LiDAR data may have some areas without measurements due to either the absorption of the laser pulses by water surfaces or the unscanned areas between

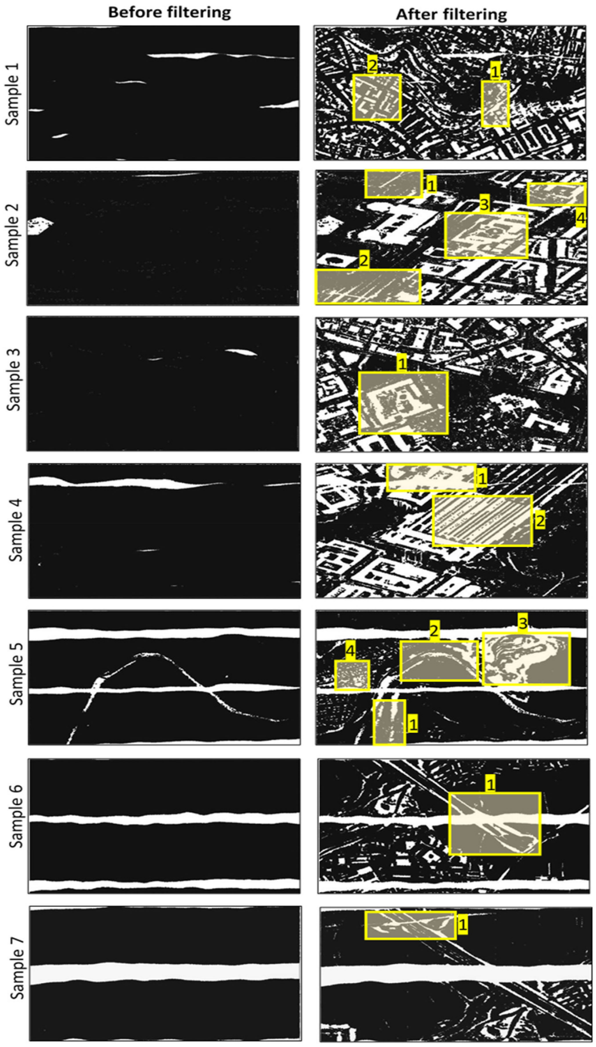

Fig. 5. Planimetric distribution of the raw LiDAR measurements (before filtering) showing the covered areas and some data gaps, and the reserved LiDAR measurements (after filtering) showing the removed points after applying the AWSF algorithm. The location and number of each subsample are highlighted in yellow.

the flight strips [55]. These gaps in raw data may affect the performance of the filter. However, the significance of the visual comparisons comes from displaying the distribution of the removed and reserved LiDAR measurements for each ISPRS sample.

As illustrated in Fig. 5, the AWSF algorithm removed the off-terrain features such as buildings and reserved the ground points that will help in producing an accurate DTM as their distribution covers the whole terrain surface excluding the filtered buildings and other nonground features. The digital surface models (DSMs) of the raw LiDAR data along with the filtered data (DTMs) for all the ISPRS dataset subsamples are shown in Fig. 6.

The statistical and visual results confirm that the AWSF method was successful in filtering raw LiDAR data in complex land covers and difficult scenarios, such as subsample 11 (different features on steep slope), subsample 31 (disconnected terrain), subsample 41 (negative blunders), subsamples 51 and 52 (low vegetation on slopes), and subsample 71 (bridge). Nevertheless, minimizing Type 1 error along with maintaining a very low value for Type 2 error is still a challenge for filtering algorithms. To overcome this issue, further research is needed to either improve the performance of some of the current filtering algorithms or utilize the state-of-the-art machine learning/deep learning methods for enhancing the classification of raw LiDAR data points by reducing both Type 1 and Type 2 errors if possible. In the following section, the generated DTMs will be compared pixel-by-pixel against an accurate reference DTMs to examine the performance of the AWSF algorithm qualitatively.

B. *U.K. Forests Dataset*

Fig. 7 shows four densely forested areas in the U.K., selected to test the AWSF algorithm performance in various types of forested zones. The specifications of the forests are stated in Table IV, they range from gentle slopes to steep mountainous areas with different tree types and different LiDAR data densities. Although airborne laser scanning technologies have advanced a great deal in the past two decades, raw LiDAR data classification in forested areas is still a very difficult task [3]. Generally, filtering raw LiDAR data for extracting the terrain surface is very useful for the forestry industry. For example, it helps in estimating the wood content in the forest after determining some forest parameters such as canopy height and trees density [10].

The AWSF algorithm is applied to process and filter the abovementioned forested areas. Mostly, the off-terrain measurements represent the laser pulses reflected by trees. The algorithm is designed to deal with each LiDAR measurement and classify it into either terrain measurement (ground point) or off-terrain measurement (tree point). After filtering all the data samples, DTMs are generated from the reserved terrain points and compared pixel-by-pixel against very accurate reference DTMs. The reference data are produced by the UK Environment Agency, its reported absolute vertical accuracy is ±0.05 m [56]. The error value is calculated for each point using

$$\mathrm{Error}_{(i)} = Z_{(i)\text{Reference DTM}} - Z_{(i)\text{Filtered LiDAR data}}. \quad (10)$$

The accuracy of the AWSF algorithm is estimated by calculating the root-mean-square error (RMSE), minimum error, maximum error, and the mean error for each sample of the forested areas [3] as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\mathrm{Error}_{(i)}\right)^2}{n-1}}. \quad (11)$$

Another very important comparison has been made; the four forested areas were filtered using professional software designed by the world-leading company in processing remote sensing data, Harris Geospatial Solutions. Part of the ENVI suite of image analysis tools, the ENVI LiDAR tool contains a set of advanced options to filter such forested areas. The filtering algorithm is proprietary to the Company and not published. Table V lists the statistical results of filtering the forested dataset samples using both the AWSF algorithm and ENVI LiDAR tool. The average value of the RMSE using the AWSF algorithm

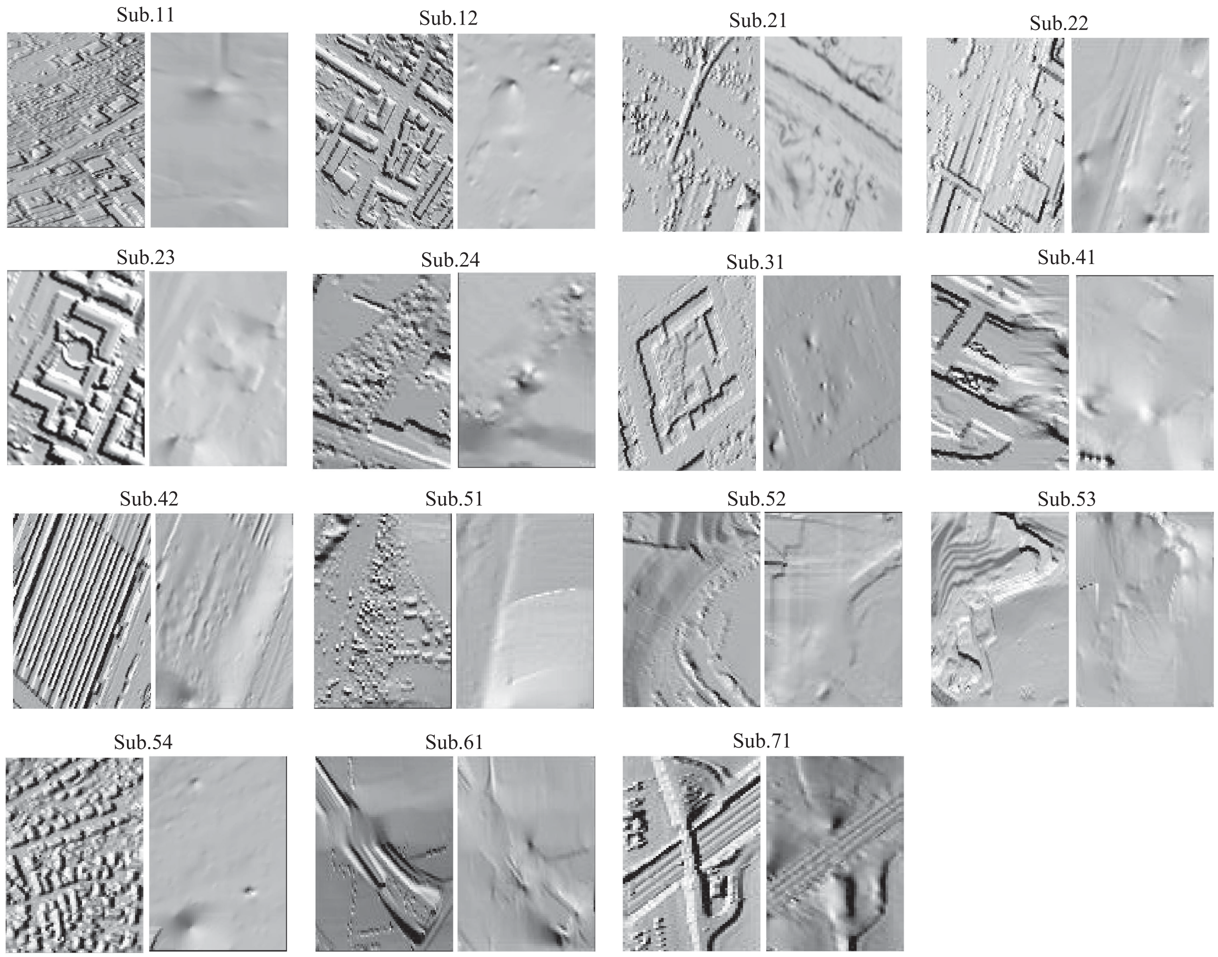

Fig. 6. DSMs and DTMs for all the 14 subsamples. The DSMs represent the raw LiDAR data (before filtering), and the DTMs represent the filtered data (after applying the AWSF algorithm).

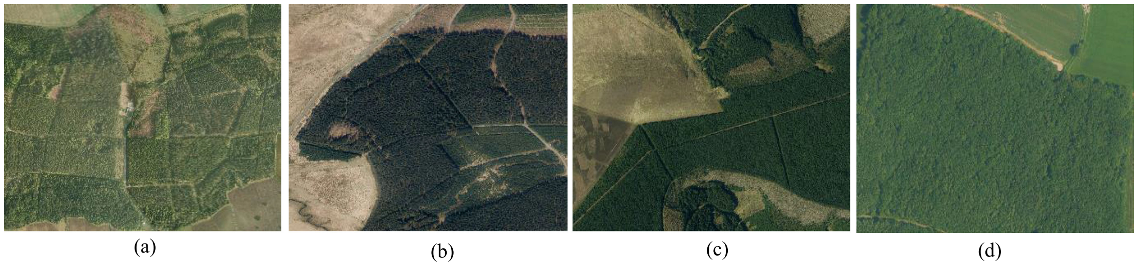

Fig. 7. U.K. dataset: four heavily forested study areas. (a) Southwick forest (England). (b) Dartmoor forest (England). (c) Drumotchy forest (Scotland). (d) Caerwent forest (Wales).

TABLE IV
U.K. Forested Areas Characteristics

Forest name	Location	Area (km²)	LiDAR data density (point/m²)	Elevation range (m)	Characteristics
Southwick	England	3	1	33-78	Oak trees – gentle sloped terrain
Dartmoor	England	1	1	338-442	Conifer trees
Drumtochy	Scotland	1	1	241-411	Pine trees - steep mountainous terrain
Caerwent	Wales	1	4	44-141	Dense LiDAR data with bare ground spaces

and ENVI algorithm are 0.32 and 0.56 m, respectively. These results confirm the excellent performance of the AWSF method as the RMSE values represent the most important indicator of the results' accuracy.

Fig. 8 compares the RMSE values using both filtering algorithms. For the Southwick forest sample, with oak trees on gentle sloped topography, both AWSF and ENVI algorithms have achieved nearly the same accuracy. However, with more difficult landscapes like Dartmoor and Caerwent forests, as the difference in elevation in these areas is about 100 m with

TABLE V
ACCURACY ESTIMATION OF THE RESULTED DTMS USING THE AWSF ALGORITHM AND ENVI LIDAR SOFTWARE

Forest name	RMSE (m)		Mean error (m)		Minimum error (m)		Maximum error (m)	
	AWSF	ENVI	AWSF	ENVI	AWSF	ENVI	AWSF	ENVI
Southwick	0.22	0.20	0.19	-0.11	-1.39	-6.75	1.98	1.03
Dartmoor	0.19	0.41	-0.01	0.08	-2.56	-14.90	1.13	6.00
Drumtochy	0.48	0.88	-0.46	0.70	-3.90	-6.13	5.70	10.95
Caerwent	0.39	0.76	-0.79	1.07	-2.86	-3.36	2.49	8.62
Average	0.32	0.56	-0.27	0.44	-2.68	-7.79	2.83	6.65

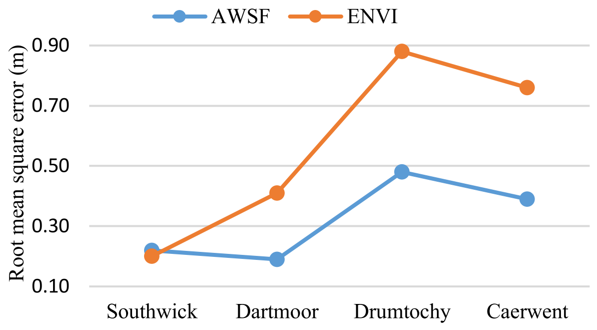

Fig. 8. RMSE values of the four filtered forested areas using both the AWSF algorithm and ENVI LiDAR tool.

very dense LiDAR data in Caerwent, AWSF has maintained as good a performance as the RMSE values, which are 0.19 and 0.48 m for Dartmoor and Caerwent, respectively. Whereas, using ENVI's algorithm, the RMSE values are 0.41 and 0.76 m for Dartmoor and Caerwent, respectively. The final forest sample, Drumtochy, represents one of the most challenging areas for most of the filtering algorithms. A heavy forest on steep terrain can easily mislead any algorithm and affect its performance. Yet, the proposed algorithm succeeded in minimizing the error value of the filtered data, as the RMSE is 0.48 m, which is of high accuracy in comparison with ENVI's algorithm results, where the RMSE is 0.88 m. Generally, the statistical results presented in Table V show the accurate filtering process of the AWSF algorithm.

The above mentioned results for both datasets indicate the robust and stable performance of the proposed algorithm in different scenarios. More specifically, the algorithm processed various densities of raw LiDAR data, starting from low density of 0.18 points/m^2 to very high density of 4 points/m^2, and various land cover, urban, rural, and forested areas, to a high degree of accuracy in comparison with the other algorithms.

C. Processing Time of Both Datasets

Processing LiDAR data is a time-consuming task due to the multistage process. This is the case not only for the AWSF algorithm but also for most of the filtering approaches that aim to maximize the reliability of the resulted data. Obviously, the processing time depends on the number of LiDAR measurements to be filtered, grid cell size, data density, etc. The AWSF algorithm's code is written and implemented using MATLAB R2019a software on a Windows 10 operating system. The hardware used is an Intel i5 processor (@3.5 Ghz) and 8 GB of RAM.

TABLE VI
PROCESSING TIME FOR EACH SAMPLE OF BOTH DATASETS USING THE AWSF ALGORITHM

Dataset	Sample	Area (km^2)	Grid cell size (m)	Number of points	Processing time (seconds)
ISPRS dataset	Sample 1	0.77	2	683,204	34.37
	Sample 2	0.26	2	243,400	26.56
	Sample 3	0.2	2	188,514	17.18
	Sample 4	0.3	2	259,028	24.67
	Sample 5	1.97	6	314,160	56.67
	Sample 6	1.71	6	275,128	50.7
	Sample 7	1.29	6	196,436	43.01
UK dataset	Southwick	3	2	3,004,739	121.02
	Dartmoor	1	2	1,001,043	62.79
	Drumtochy	1	2	1,013,529	79.3
	Caerwent	1	1	3,999,920	193.11

Table VI summarizes the time required for filtering all the ISPRS data samples and the U.K. forests samples using the AWSF algorithm. The grid cell size used for the ISPRS data in urban and rural areas is 2 m and 6 m, respectively, and depends on the density of the raw data. For the U.K. dataset, the grid cell size used is 2 m for Southwick, Dartmoor, and Drumtochy samples. Whereas for the very dense forest sample (Caerwent), the grid cell size is set to be 1 m. In addition to the grid cell size, the geographical extent and the amount of raw LiDAR data affect the processing time. For example, the Caerwent forest sample area is 1 km^2, which is the same for both Dartmoor and Drumotchy. However, the processing time differs due to the different densities of raw LiDAR data, as the number of input points in Caerwent is four times more than the other two samples. On the other hand, the time required for filtering a similar amount of raw data for sample 2 and sample 6 is 26.56 and 50.70 s, respectively. Taking into account the difference in the grid cell size between both samples (2 m for sample 2 and 6 m for sample 6) that should reduce the processing time, but it is doubled due to the significant difference in areas. Sample 2 covers a small area of 0.26 km^2, whereas sample 6 covers 1.71 km^2. Although the AWSF method has focused on improving the quality of the filtered LiDAR data, the processing time is sufficient, as it ranges from 17 s for small areas to 3.21 min for the very dense sample (Caerwent, 4 points/m^2), which covers a heavily forested area.

IV. CONCLUSION

This article presents a new algorithm for filtering raw LiDAR data to extract only the terrain measurements. The core of the proposed algorithm, AWSF, is based on interpolating raw LiDAR data using cubic smoothing splines with a predefined smoothing parameter and threshold values. Moreover, the AWSF algorithm presented, as one of its processing stages, an effective automatic approach to remove the negative blunders versus the previous manual methods [51]. After the first iteration, the AWSF allocates a weight value for each point of the raw LiDAR data. The algorithm utilizes the z-shaped function as a weight function to improve the classification process and help the algorithm in making an accurate judgment at a point level.

The performance of the algorithm is tested against two different datasets. The first one contains 15 subsamples in urban and rural areas. The accuracy of the filtered LiDAR measurements is assessed by calculating the percentages of classifying the terrain and off-terrain LiDAR measurements incorrectly. The AWSF succeeded in minimizing the percentage of the reserved off-terrain points, which affect the generated DTMs badly [1], [54]. The average percentage of the reserved off-terrain points (over all samples) using the AWSF is 1.44%, which is the best (lowest) when compared against all the other 14 filtering algorithms published between 1998 and 2019; the mean of the average percentage (over all samples) of all the other algorithms is 4.47% with the lowest average being 1.96%. As heavily forested areas on steep terrain represent the most challenging scenario for all filtering algorithms, the second dataset contains four different types of dense forests. The forested areas are filtered not only using the AWSF algorithm, but also using ENVI LiDAR, which is a professional remote sensing software designed by the world-leading company, Harris Geospatial Solutions. Very accurate DTMs are used as a reference data. The filtered LiDAR data are then compared pixel-by-pixel against the reference DTMs to calculate the error value for each point and estimate the accuracy (RMSE). The accuracies of the resultant DTMs of the most difficult forest landscape (steep mountainous terrain) using both the AWSF algorithm and ENVI software are 0.48 m and 0.88 m, respectively. The abovementioned results show clearly the robustness and reliability of the proposed AWSF algorithm for raw LiDAR data filtering.

Acknowledgment

The author would like to specially thank EDINA digimap LiDAR data service for providing the required data for the forested areas. They would also like thank S. Dunn from The Forestry Commission for his help in identifying the forested areas in the U.K.

References

[1] D. Mongus and B. Zalik, “Parameter-free ground filtering of LiDAR data for automatic DTM generation,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 67, no. 1, pp. 1–12, 2012.

[2] J. Shan and S. Aparajithan, “Urban DEM generation from raw LiDAR data: A labeling algorithm and its performance Jie,” *Photogrammetric Eng. Remote Sens.*, vol. 71, no. 2, pp. 217–226, 2005.

[3] A. S. Maguya, V. Junttila, and T. Kauranne, “Adaptive algorithm for large scale DTM interpolation from LiDAR data for forestry applications in steep forested terrain,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 85, pp. 74–83, 2013.

[4] Y. Quan, J. Song, X. Guo, Q. Miao, and Y. Yang, “Filtering LiDAR data based on adjacent triangle of triangulated irregular network,” *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 11051–11063, 2017.

[5] Y. B. Yang, N. N. Zhang, and X. L. Li, “Adaptive slope filtering for airborne light detection and ranging data in urban areas based on region growing rule,” *Surv Rev.*, vol. 6265, no. June, pp. 1–8, 2016.

[6] D. Tovari and N. Pfeifer, “Segmentation based robust interpolation—A new approach to laser data filtering,” in *Proc. Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.*, 2005, vol. 36-3/W19, pp. 79–84.

[7] Z. Hui, Y. Hu, Y. Z. Yevenyo, and X. Yu, “An improved morphological algorithm for filtering airborne LiDAR point cloud based on multi-level kriging interpolation,” *Remote Sens.*, vol. 8, no. 1, pp. 12–16, 2016.

[8] A. H. Ozcan and C. Unsalan, “LiDAR point data filtering and DTM generation using empirical mode decomposition,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 1, pp. 360–371, Jan. 2017.

[9] C. Serifoglu, O. Gungor, and V. Yilmaz, “Performance evaluation of different ground filtering algorithms for UAV-based point clouds,” *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLI-B1, no. July, pp. 245–251, 2016.

[10] A. S. Maguya, V. Junttila, and T. Kauranne, “Algorithm for extracting digital terrain models under forest canopy from airborne LiDAR data,” *Remote Sens.*, vol. 6, no. 7, pp. 6524–6548, 2014.

[11] K. Kraus and N. Pfeifer, “Determination of terrain models in wood areas with airborne laser scanner data,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 53, no. 4, pp. 193–203, 1998.

[12] N. Pfeifer, T. Reiter, C. Briese, and W. Rieger, “Interpolation of high quality ground models from laser scanner data in forested areas,” *Int. Arch. Photogrammetry Remote Sens.*, vol. 32, no. 3/W14, pp. 31–36, 1999.

[13] A. Kobler, N. Pfeifer, and P. Ogrinc, “Using redundancy in aerial LiDAR point cloud to generate DTM in steep forested relief,” in *Proc. Workshop 3D Remote Sens. Forestry*, Vienna, 14–15 Feb., 2006, pp. 1–6.

[14] H. Hu *et al.*, “An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 92, pp. 98–111, 2014.

[15] G. Forlani and C. Nardinocchi, “Adaptive filtering of aerial laser scanning data,” *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XXXVI, no. 3 / W52, pp. 130–135, 2007.

[16] G. Vosselman, “Slope based filtering of laser altimetry data,” *Int. Arch. Photogrammetry Remote Sens.*, vol. 33, no. Part B3/2, pp. 678–684, 2000.

[17] G. Sithole, “Filtering of laser altimetry data using a slope adaptive filter,” *Int. Arch. Photogrammetry Remote Sens.*, vol. 34, no. 3(WG4), pp. 203–210, 2001.

[18] K. Zhang, S. C. Chen, D. Whitman, M. L. Shyu, J. Yan, and C. Zhang, “A progressive morphological filter for removing nonground measurements from airborne LiDAR data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 872–882, Apr. 2003.

[19] C. Chen, Y. Li, N. Zhao, J. Guo, and G. Liu, “A fast and robust interpolation filter for airborne LiDAR point clouds,” *PLoS ONE*, vol. 12, no. 5, pp. 1–20, 2016.

[20] Z. Chen, B. Gao, and B. Devereux, “State-of-the-art: DTM generation using airborne LiDAR data,” *Sensors*, vol. 17, no. 1, p. 150, 2017.

[21] Q. Chen, P. Gong, D. Baldocchi, and G. Xie, “Filtering airborne laser scanning data with morphological methods,” *Photogrammetric Eng. Remote Sens.*, vol. 73, no. 2, pp. 175–185, 2007.

[22] J. Kilian, N. Haala, and M. Englich, “Capture and evaluation of airborne laser scanner data,” *Int. Arch. Photogrammetry Remote Sens.*, vol. XXXI, no. Part B3, pp. 383–388, 1996.

[23] L. Liu and S. Lim, “A voxel-based multiscale morphological airborne LiDAR filtering algorithm for digital elevation models for forest regions,” *Meas., J. Int. Meas. Confederation*, vol. 123, pp. 135–144, 2018.

[24] G. Sithole and G. Vosselman, “Filtering of airborne laser scanner data based on segmented point clouds,” *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 3/W3-4, pp. 66–71, 2005.

[25] S. Filin, “Surface clustering from airborne laser scanning data,” *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 34, pp. 119–124, 2002.

[26] A. Kobler, N. Pfeifer, P. Ogrinc, L. Todorovski, K. Oštir, and S. Džeroski, “Repetitive interpolation: A robust algorithm for DTM generation from aerial laser scanner data in forested terrain,” *Remote Sens. Environ.*, vol. 108, no. 1, pp. 9–23, 2007.

[27] W. Schickler and A. Thorpe, “Surface estimation based on LIDAR,” in *Proc. ASPRS Annu. Conf.*, 2001, pp. 23–27.

[28] M. Brovelli, M. Cannata, and U. Longoni, “Managing and processing LIDAR data within GRASS,” in *Proc. Open Source GIS—GRASS Users Conf.*, pp. 11–13, 2002.

[29] H. Chen, M. Cheng, J. Li, and Y. Liu, “An iterative terrain recovery approach to automated DTM generation from airborne LiDAR point clouds,” *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XXXIX-B4, pp. 363–368, 2012.

[30] W. Zhang *et al.*, “An easy-to-use airborne LiDAR data filtering method based on cloth simulation,” *Remote Sens.*, vol. 8, no. 6, pp. 1–22, 2016.

[31] J. Susaki, “Adaptive slope filtering of airborne LiDAR data in urban areas for digital terrain model (DTM) generation,” *Remote Sens.*, vol. 4, no. 6, pp. 1804–1819, 2012.

[32] C. A. Silva, C. Klauberg, Â. M. K. Hentz, A. P. D. Corte, U. Ribeiro, and V. Liesenberg, “Comparing the performance of ground filtering algorithms for terrain modeling in a forest environment using airborne LiDAR data,” *Floresta e Ambiente*, vol. 25, no. 2, pp. 1–10, 2018.

[33] Q. Chen, H. Wang, H. Zhang, M. Sun, and X. Liu, “A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas,” *Remote Sens.*, vol. 8, no. 71, pp. 1–22, 2016.

[34] H. Guan, J. Li, Y. Yu, L. Zhong, and Z. Ji, "DEM generation from LiDAR data in wooded mountain areas by cross-section-plane analysis," *Int. J. Remote Sens.*, vol. 35, no. 3, pp. 927–948, 2014.

[35] G. Sohn and I. Dowman, "Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion," *Int. Arch. Photogrammetry Remote Sens.*, vol. 24, no. 3, pp. 336–344, 2002.

[36] B. Yunfei *et al.*, "Classification of LiDAR point cloud and generation of DTM," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XXXVII, pp. 313–318, 2002.

[37] M. Bartels and H. Wei, "Threshold-free object and ground point separation in LiDAR data," *Pattern Recognit. Lett.*, vol. 31, no. 10, pp. 1089–1099, 2010.

[38] M. Bartels and H. Wei, "Segmentation of LiDAR data using measures of distribution," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XXXVI, pt. 7, pp. 1–6, 2006.

[39] Z. Hui, D. Li, S. Jin, Y. Y. Ziggah, L. Wang, and Y. Hu, "Automatic DTM extraction from airborne LiDAR based on expectation-maximization," *Opt. Laser Technol.*, vol. 112, pp. 43–55, 2019.

[40] S. Cai *et al.*, "Filtering airborne LiDAR data through complementary cloth simulation and progressive TIN densification filters," *Remote Sens.*, vol. 11, no. 9, p. 1037, 2019.

[41] Z. Cai, H. Ma, and L. Zhang, "Model transfer-based filtering for airborne LiDAR data with emphasis on active learning optimization," *Remote Sens. Lett.*, vol. 9, no. 2, pp. 111–120, 2018.

[42] E. Bayram, P. Frossard, E. Vural, and A. Alatan, "Analysis of airborne LiDAR point clouds with spectral graph filtering," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1284–1288, Aug. 2018.

[43] E. J. Wegman and I. W. Wright, "Splines in statistics," *J. Amer. Statistical Assoc.*, vol. 78, no. 382, pp. 351–365, 1983.

[44] D. Culpin, "Calculation of cubic smoothing splines for equally spaced data," *Numerische Mathematik*, vol. 48, no. 6, pp. 627–638, 1986.

[45] T. Ramsay, "Spline smoothing over difficult regions," *J. Roy. Statistical Soc. Series B, Statistical Methodol.*, vol. 64, no. 2, pp. 307–319, 2002.

[46] B. W. Silverman, "Some aspects of the spline smoothing approach to non-parametric regression curve fitting," *J. Roy. Statistical Soc.*, vol. 74, no. 1, pp. 1–52, 1985.

[47] C. De Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer, 2001, pp. 207–240.

[48] D. Jin and S. Lin, *Advances in Mechanical and Electronic Engineering*. New York, NY, USA: Springer, 2012, pp. 177–182.

[49] S. P. Simonovic and N. Nirupama, "A spatial multi-objective decision-making under uncertainty for water resources management," *J. Hydroinformat.*, vol. 7, no. 2, pp. 117–133, 2005.

[50] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk, "Morphology for matrix data: Ordering versus PDE-based approach," *Image Vis. Comput.*, vol. 25, no. 4, pp. 496–511, 2007.

[51] S. Nie, C. Wang, P. Dong, X. Xi, S. Luo, and H. Qin, "A revised progressive TIN densification for filtering airborne LiDAR data," *Meas., J. Int. Meas. Confederation*, vol. 104, pp. 70–77, 2017.

[52] EDINA, *LiDAR Data Download*. Edinburgh, U.K.: University of Edinburgh, 2017. [Online]. Available: http://edina.ac.uk/digimap

[53] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 59, no. 1–2, pp. 85–101, 2004.

[54] S. Lin and H. Wu, "Sweep-line-based filtering of airborne laser scanning data," *Proc. SPIE - Int. Soc. Op. Eng.*, vol. 6419, 2006, Art. no. 64191S.

[55] G. Sithole and G. Vosselman, "Report: ISPRS comparison of filters," ISPRS Commission III, Working Group 3 & Dept. Geodesy, Faculty of Civil Eng. Geosci. Delft Univ. of Technology, Delft, The Netherlands, 2003.

[56] Environment Agency, "Environment agency LiDAR data," Environ. Agency LiDAR Data, Bristol, U.K., 2016, pp. 1–16.

Ziad Abdeldayem received the Ph.D. degree in civil engineering from Cairo University, Egypt, in 2009.

He is currently a Senior Lecturer with Liverpool John Moores University, Liverpool, U.K. His research interests are related to the applications of GIS, remote sensing and spatial data processing and analysis in mapping, digital infrastructure, sustainable built environment, Earth sciences, archaeology, environmental studies, forestry, geography, etc.