Remote Sens. **2013**, *5*, 3749-3775; doi:10.3390/rs5083749

OPEN ACCESS

Remote Sensing

ISSN 2072-4292

www.mdpi.com/journal/remotesensing

Article

SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas

Jixian Zhang, Xiangguo Lin * and Xiaogang Ning

Key Laboratory of Mapping from Space, Chinese Academy of Surveying and Mapping,
Lianhuachixi Road No. 28, Beijing 100830, China; E-Mails: zhangjx@casm.ac.cn (J.Z.);
ningxg@casm.ac.cn (X.N.)

* Author to whom correspondence should be addressed; E-Mail: linxiangguo@gmail.com;
Tel.: +86-10-6388-0577; Fax: +86-10-6388-0535.

Received: 4 May 2013; in revised form: 17 July 2013 / Accepted: 18 July 2013 / Published: 31 July 2013

Abstract: Object-based point cloud analysis (OBPA) is useful for information extraction from airborne LiDAR point clouds. An object-based classification method is proposed for classifying the airborne LiDAR point clouds in urban areas herein. In the process of classification, the surface growing algorithm is employed to make clustering of the point clouds without outliers, thirteen features of the geometry, radiometry, topology and echo characteristics are calculated, a support vector machine (SVM) is utilized to classify the segments, and connected component analysis for 3D point clouds is proposed to optimize the original classification results. Three datasets with different point densities and complexities are employed to test our method. Experiments suggest that the proposed method is capable of making a classification of the urban point clouds with the overall classification accuracy larger than 92.34% and the Kappa coefficient larger than 0.8638, and the classification accuracy is promoted with the increasing of the point density, which is meaningful for various types of applications.

Keyword: airborne LiDAR; object-based classification; point clouds; segmentation; SVM

1. Introduction

Recent years have seen the emergence of airborne LiDAR (Light Detection And Ranging), also termed as airborne laser scanning (ALS), as a leading technology for the extraction of information about physical surfaces [1]. Compared with other techniques such as interferometric SAR and photogrammetry, ALS has the advantage of acquiring dense, discrete, detailed and accurate 3D point coverage over both the objects and ground surfaces directly. As a result, ALS is not only commonly used for generation of high spatial resolution digital terrain model (DTM; [2]), but also has become an important tool for a range of applications such as reconstruction of the 3D buildings models [3] and 3D road models [4], parameter estimation for forestry [5,6], *etc.* Theoretically, the detailed depiction of objects and ground surfaces by the dense point cloud may indicate a directness or simplicity in which objects and related information can be retrieved from the data. However, similar to the existing data sources such as aerial or satellite optical imagery, extensive manual post-processing is still required to extract accurate terrain or semantic information from the LiDAR point clouds [7], and thus underlining the necessity for research of the automatic methods in this area. Point cloud classification is one of the hot topics in the field of ALS data processing, because classification is a precondition of many applications such as building reconstruction or vegetation modeling or flood modeling [8].Point cloud classification implies the separation of points into vegetation, building or ground classes, and each of these classes implies the knowledge of its nature. Lots of point cloud classification methods have been proposed, and there are different classification systems for the existing methods based on different criteria.

According to the types of data sources employed, the existing methods can be categorized into points-based classification, full-waveform-data-based classification and data-fusion-based classification. Points-based classification only makes use of the available point clouds as the sole data source in classification, as done in [8]. Full-waveform-data-based classification makes use of the wave-derived features besides the ones from point clouds in classification [9]. For example, Mallet *et al.* [10] performed analysis of the full waveform data using support vector machines (SVM), and Rutzinger *et al.* [11] extract the vegetations from the full waveform data. Alternative, fusion of ALS data and other types of data sources are also employed to make land cover or land use classification [12]. For example, Secord and Zakhor [13] proposed an approach to detecting trees in registered and fused aerial image and LiDAR data, and SVM is utilized to classify the segmented images. Rottensteiner *et al.* [14] detected buildings by fusion of ALS data and multi-spectral images. Of course, full waveform data and fused data can simplify the task about classification, but these kinds of data also have some defects. On the one hand, other types of data sources are often not available. On the other hand, the slow acquisition of other data can hinder the fast data processing. Thus, only ALS point cloud is considered in this paper.

Based on the strategies for solving the classification problem, the existing methods can be categorized into hierarchical classification and simultaneous classification. The hierarchical classification consists of two sub-steps, filtering and objects classification [15]. In the first sub-step, a separation of points into terrain and object measurements is performed. Various types of filtering methods have been reviewed in Liu's survey [16], including the progressive TIN densification [17,18], hierarchic robust interpolation [19,20], *etc*. In the second sub-step, object measurements are further separated into buildings points, vegetation points, vehicle points, powerline points *etc*. For example, Darmawati [21] separated the building points from the vegetation points. Many hierarchical classification methods have been proposed. For example, Meng *et al*. [22] presented a morphological building detecting method to identify buildings by gradually removing non-building pixels, first ground ones and second vegetation ones. Sometimes hierarchical classification focuses on specific object types. For instance, Yao *et al.* [23] extract vehicles from ALS data for transportation monitoring. In hierarchical classification, the filtering sub-step not only performs a binary discrimination but also derives features for points or groups of points (e.g., segments, facets, *etc*.).The features are then combined in a discrimination function to yield the class membership of a point or a group of points. The objects classification sub-step performs this discrimination in a feature space by partitioning the feature space. This partitioning leads to a general solution that comes with its own some problems (e.g., choosing an appropriate classifier). What matters is the assumptions/rules made in a classification and here the advantage of a classical classification approach is that few assumptions are necessary (e.g., selection of features). However, the weakness is that hierarchical classification approaches adapt (because of the feature space partitioning) to the geometric and radiometric characteristics of the point cloud, and therefore the classification rules are not necessarily transferrable from one point cloud to another. To sum up, an advantage of hierarchical classification methods is they also attempt discover general rules that can be used to classify point clouds of many different kinds of landscapes, but a disadvantage is they make many assumptions. On the contrary, the simultaneous classification separates the point clouds into ground measurements, building measurements, vegetation measurements at the same time. For example, Oude Elberink and Mass [24] performed an unsupervised classification of the ALS data using anisotropic height texture features, which suggests that anisotropic operations have the potential to discriminate between orientated (with contrast at the edges in only one direction) and non-orientated (with contrast in horizontal, diagonal, as well in vertical directions) objects.

According to the basic element employed in classification, existing methods can be categorized into two groups, point-based methods and segment-based methods.

- Point-based classification methods. Point-based classification is performed by analyzing the features of one single laser point. Taking a discrete point of the surface and comparing its features to another point, the obtainable information is restricted in terms of the nature of the sampled surface. Most of the filtering algorithms belong to the point-based classifications, such as the labeling algorithm [25] and multi-scale curvature classification LiDAR algorithm [26]. Additionally, Petzold and Axelsson [27] presented a point cloud classification method in which it is found the reflectance of the laser echo is a good feature to discriminate the neighboring objects. However, the number of point features is limited, which allows only simple classifications for this kind of methods.
- Segment-based classification. Segment-based classification is to first segment the data, and then perform a classification based on those segments. A special case of segment-based classification is the segment-based filtering. The rationale behind such algorithms is that segments of objects

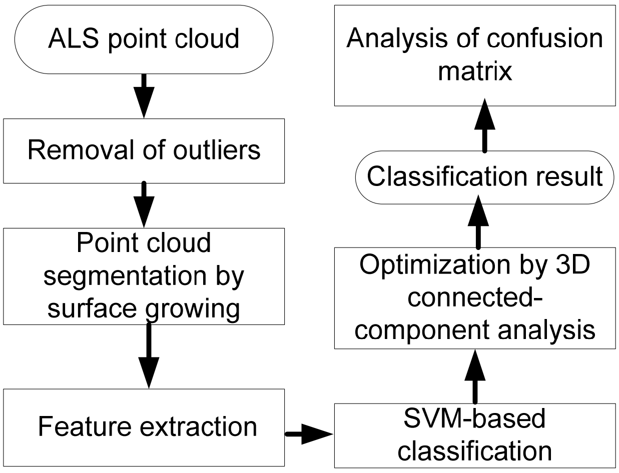
are situated higher than the ground segments [28,29]. The segment-based filters are typically designed for urban areas where many step edges can be found in the data. In the segment-based classification, Sithole [30] designed a four step procedure to classify the ALS data, in which it detects ‘macro objects’, bridges, ‘micro objects’ and the man-made and natural objects. The segments are classified depending on their geometrical relation. Golovinskiy *et al.* [31] design a system for recognizing objects in 3D point clouds obtained by mobile laser scanning in urban environments, and the system is decomposed into four steps: locating, segmenting, characterizing, and classifying clusters of 3D points. Shen *et al.* [29] proposed a segment-based classification method for ALS data with multiple returns, in which a rule-based classification is performed on the segmented point clouds. A special segment-based classification application can be seen in Barsi *et al.* [32], where vehicles on the highway are classified in ALS data by a clustering method.

After reviewing the existing methods, we found that the segment-based and simultaneous classification is becoming more and more popular, which should attribute to the point cloud segmentation. Practice has shown that the variety of 3Dobjects and the massive amount of points require introducing some level of organization into the ALS point clouds before the extraction of information can become effective [1,18], and such organization involves aggregating points with similar features into segments, usually in the form of surfaces, which will increase the classification accuracy and reduce the uncertainties in classification. As a result, an object-based point cloud analysis (OBPA) concept is proposed to classify the ALS data [11]. OBPA is designed to maintain the full resolution and information in ALS data and to avoid any rasterization and conversion to a 2.5D model, and it is composed of four steps, including segmentation, feature calculation, classification, and parameter estimation within the original point cloud [11]. In this sense, OBPA is very similar to the object-based imagery analysis [33] by eCognition Software in the field of remote sensing. With this background, an object-based classification method for airborne point clouds is proposed in this article, and it is composed of four core steps. Particularly, surface growing algorithm is employed to segment the ALS data, four types of features are calculated from the segments, SVM is selected as classifier to perform the classification of the segments, and connected-component analysis for point clouds is employed to sieve the small noises in the original classification results. Note that the urban environments are focused on herein, because the point cloud segmentation method employed prefers the objects with planar surfaces, which exist widely in urban environments. The main contribution of this article consists of two parts. The first one is the integration of surface growing algorithm and SVM for a segment-based classification strategy, and the second one is the proposal of a connected-component analysis for sieving the small noises in classified 3D point clouds.

2. The Proposed Method

The proposed method is composed of six sub-steps, including outlier removal, point cloud segmentation, features calculation, SVM-based classification, optimization of the classification result, and performance evaluation, as shown in Figure 1.

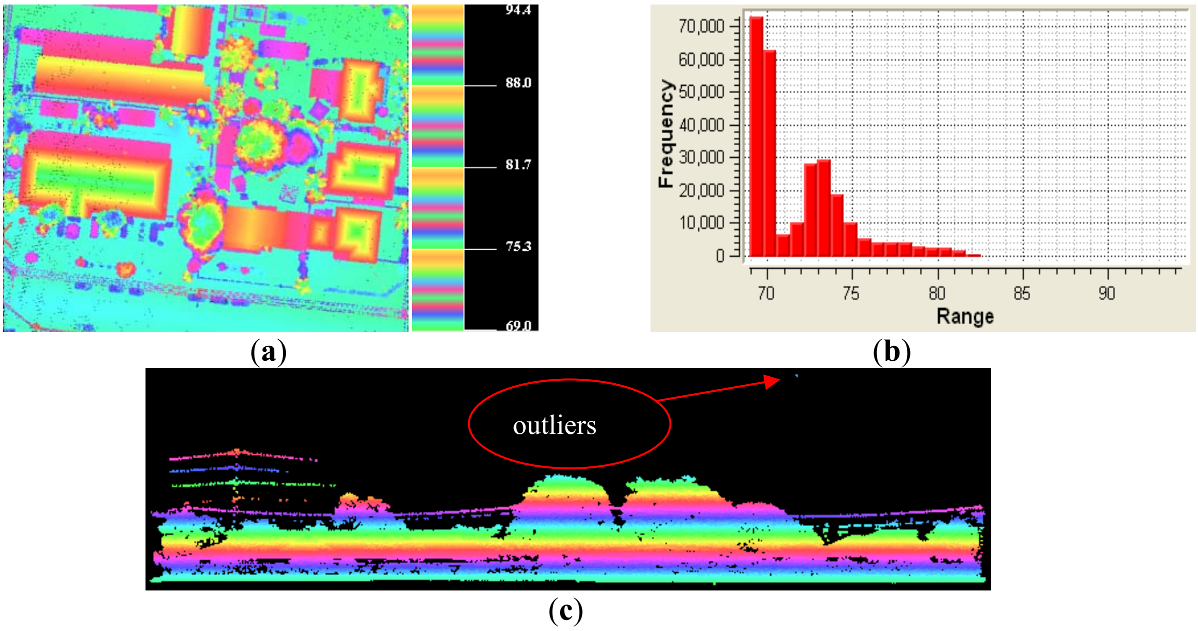
Figure 1. Flow chart of the proposed classification method


ALS point cloud
Removal of outliers
Point cloud segmentation by surface growing
Feature extraction
SVM-based classification
Optimization by 3D connected-component analysis
Classification result
Analysis of confusion matrix

2.1. Outlier Removal

Generally, original airborne LiDAR data contains outliers, as shown in Figure 2a,c. Outlier elimination is one of the key steps in the pre-processing of airborne LIDAR point clouds, because it has a significant influence on the subsequent classification operation. In this article, the method proposed by Silván-Cárdenas and Wang [34] is used to remove the outliers. The result of removing outliers from the point cloud in Figure 2a is displayed in Figure 2d. The results suggest that there is no obvious outlier in the outputs.

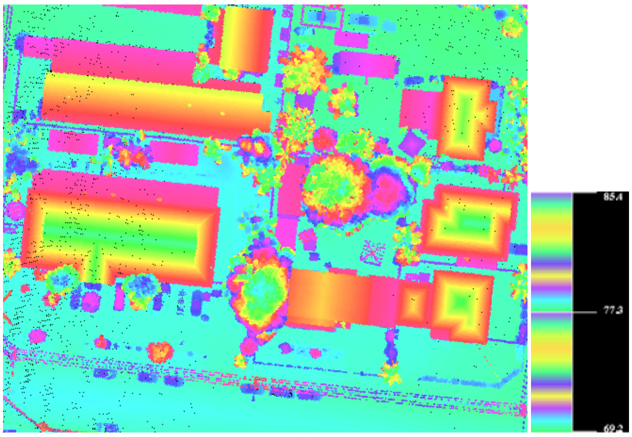
Figure 2. Outlier removal of a point cloud. (**a**) Original point clouds with outliers. (**b**) Elevation histogram of the point cloud in (a). (**c**) Side view of the point cloud in (a). (**d**) The resulted point cloud after outlier removal. Note there are 265266 measurements in the data of (a), and 265260 measurements in the data of (d).



(**a**) (**b**)

(**c**)

Figure 2. *Cont.*



(**d**)

2.2. Segmentation by Surface Growing

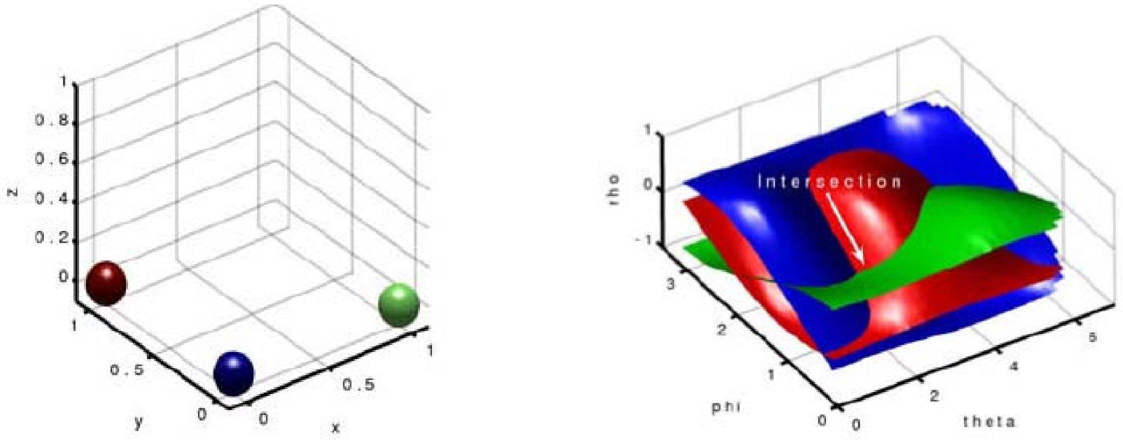
Segmentation is often the first step in extracting information from a point cloud [35]. Similar to image segmentation, the point cloud segmentation refers to the operation that will separate points into different groups based on characteristics without knowledge of what they really are. The most common segmentations of point clouds are those that group points that fit to the same plane, smooth surface, sphere or cylinder *etc*. There are a variety of airborne LiDAR point cloud segmentation methods, such as slope adaptive neighborhood method [1], mean-shift-based method [36], octree-based split-and-merge method [37].

Among the methods, the surface growing algorithm proposed by Vosselman *et al.* [38] is selected. The surface growing algorithm can be considered as an extension of the well-known region growing method into three dimensions. For processing point clouds, surface growing algorithm can be formulated as a method of grouping nearby points based on some homogeneity criterion, such as planarity or smoothness of the surface defined by the grouped points [35]. The surface growing algorithm is composed of two steps: seed detection and growing. Particularly, the seed detection is performed by testing if a minimum number of nearby points can be fitted to a plane, and 3D Hough transformation [39] is employed to finish this task, as shown in Figure 3. The growing can be based on one or more the following criteria:

- Proximity of points. Only points within certain distance to a seed surface, or the neighboring points of a seed surface, can be added to this seed surface.
- Globally planar. For this criterion, a plane equation is determined by fitting a plane through all surface points in this seed surface. Points can only be added if the perpendicular distance to the plane is below some a threshold.

The surface growing algorithm is not only capable of grouping the point dataset into planar faces, as shown in Figure 4b,c, but also into smooth surfaces, as shown in Figure 4f. Visual analysis of Figure 4b tells that, after the surface growing for plane detection, each segment should represent one object with semantics such as a roof, a wall, a road segment and a ground plane. At the same time, visual analysis of Figure 4f tells that, after the surface growing for smooth surface detection, each segment should represent one object with semantics such as a building, a smooth ground, but a building segment may mix with a ground segment when a smooth connection exists. This mixture is called undersegmentation. Thus, only the surface growing for plane detection is adopted in this article to avoid the undersegmentation. Figure 4b,c also suggest the above segmentation derives new features for the point clouds. For example, the sizes/areas of segments can reflect the types of objects. Particularly, the sizes of segments belonging to trees, powerlines and vehicles are usually smaller than those belonging to buildings and ground surfaces. Moreover, the building segments are often more rectangular than the ground segments. In a word, the point cloud segmentation will derive more features than the original data owns.

Figure 3. Voting in the 3D Hough space for plane detection (from [39]). (**a**) Three discrete coplanar points. (**b**) 3D Hough space where the surfaces, corresponding to the above three points, intersect at a point.



(**a**) (**b**)

The above segmentation will derive several 2D geometric properties, including boundary/contour, area, perimeter, and minimal oriented bounding box (MOBB). In order to derive the 2D geometric features, the points should be transformed into a 2D space from the 3D space. Thus, this can be done by projecting the points of a segment onto the best-fit plane and rotating the plane to be horizontal (as shown in Figure 5). Particularly, the best-fit plane of a segment is calculated using the principal component analysis (PCA), which is derived from the theory of the least-squares estimation [39]. Moreover, for each point $\mathrm{p}(\mathrm{x}, \mathrm{y}, \mathrm{z})$, the projected point p' on the best-fit plane can be calculated using the following formula:

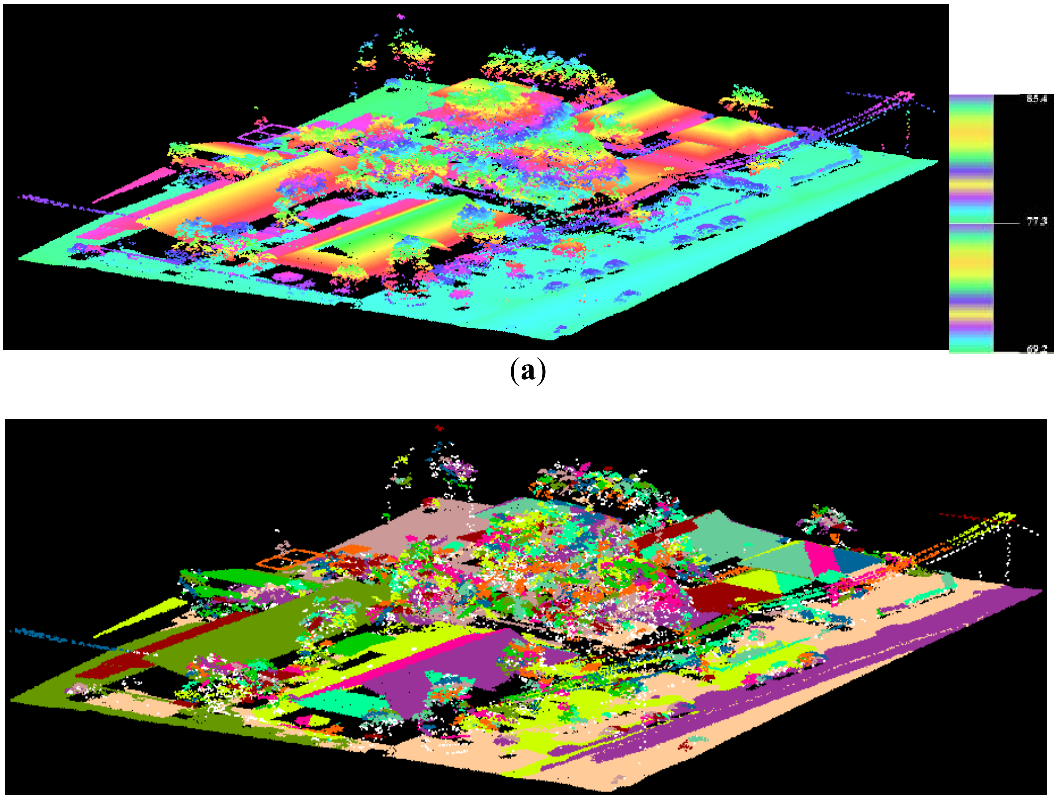
$$\mathbf{p}' = \mathbf{p} - \frac{\mathbf{a}x + \mathbf{b}y + \mathbf{c}z + \mathbf{d}'}{\mathbf{a}^2 + \mathbf{b}^2 + \mathbf{c}^2} \cdot \mathbf{n} \tag{1}$$

where $\mathrm{n} = (\mathrm{a}, \mathrm{b}, \mathrm{c})$ is the normal vector of the best-fit plane and it is a unit vector; d' is the perpendicular distance from the origin to the plane.

Once the transformation is finished, the following geometric properties can be derived as follows:

- Concave boundary: The boundary which closely encompasses the point cluster could be determined by finding the alpha shape of the points [40], as shown in Figure 6,b. Note, the points on the boundary also can derive the boundary of the segment in 3D space because each transformed point has the same point number as the original laser point, as shown in Figure 6,d. The boundaries of the segmented point clouds in Figure 4,b are displayed in Figure 4,g, which suggests that the boundaries of the segments reflect the geometric features of the segments.
- Area: The covered area of the point cluster can be computed using the connected concave boundary.
- MOBB: The MOBB is calculated from the concave polygon formed by the above concave boundary, as shown in Figure 6c.

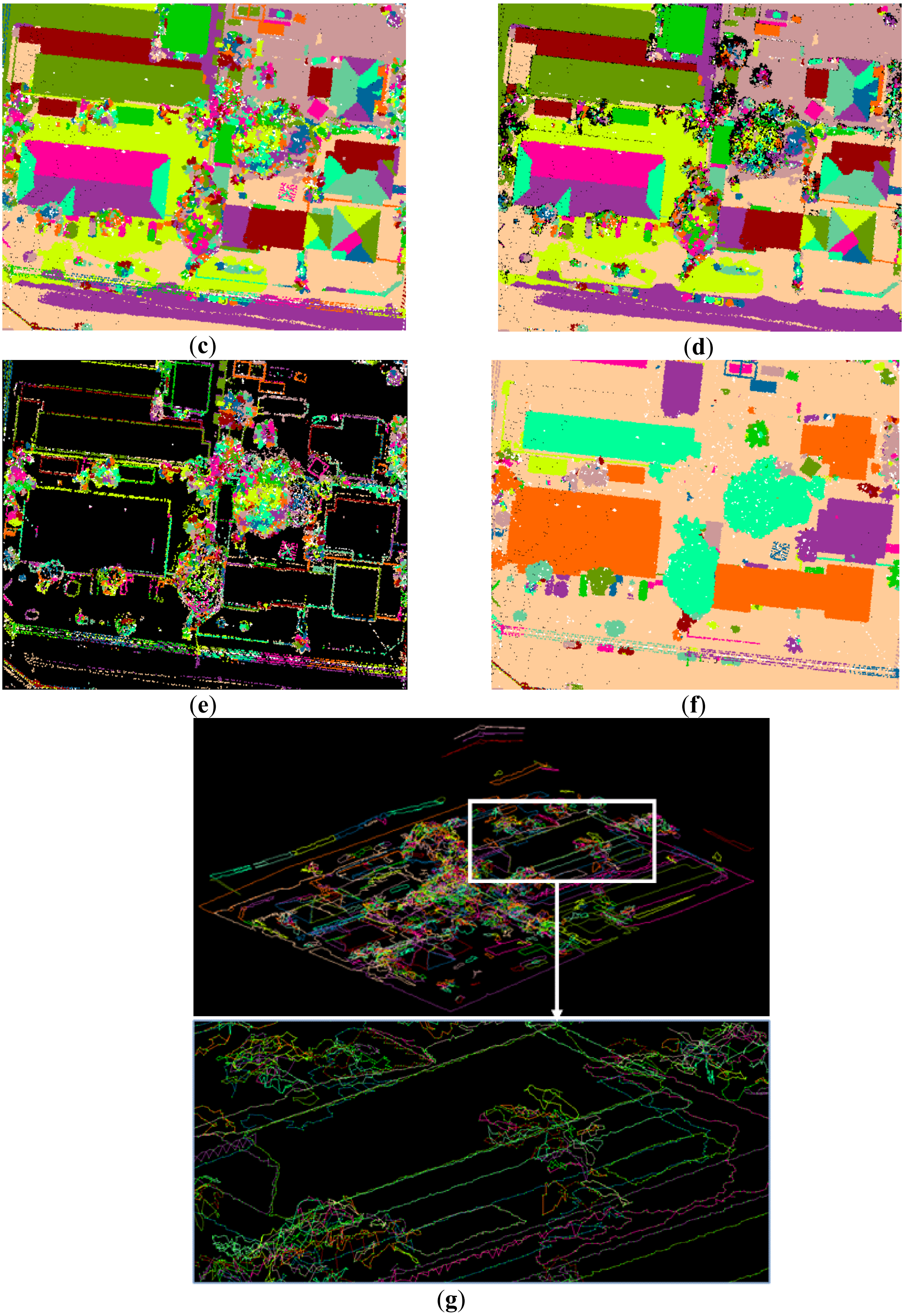
Figure 4. Point cloud segmentation by surface growing algorithm, and 3D boundary extraction for the segments. (**a**) Perspective view of the airborne laser scanning **(**ALS) point cloud of Figure 2d, colored by elevation value. (**b**) Perspective view of segmented point cloud in (a) colored by labeling number. (**c**) Top view of the segmented point cloud. (**d**) Top view of single returns. (**e**) Top view of multiple returns. (**f**) Top view of point cloud after smooth segmentation. (**g**) The 3D concave boundaries and boundary points of the segments in (b), in which the boundaries are colored by the polygon number, and boundary points are colored in green.



(**a**)

(**b**)

Figure 4. *Cont.*

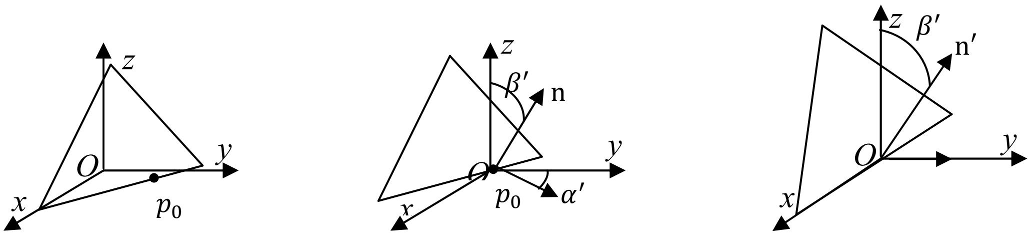


(**c**) (**d**)

(**e**) (**f**)

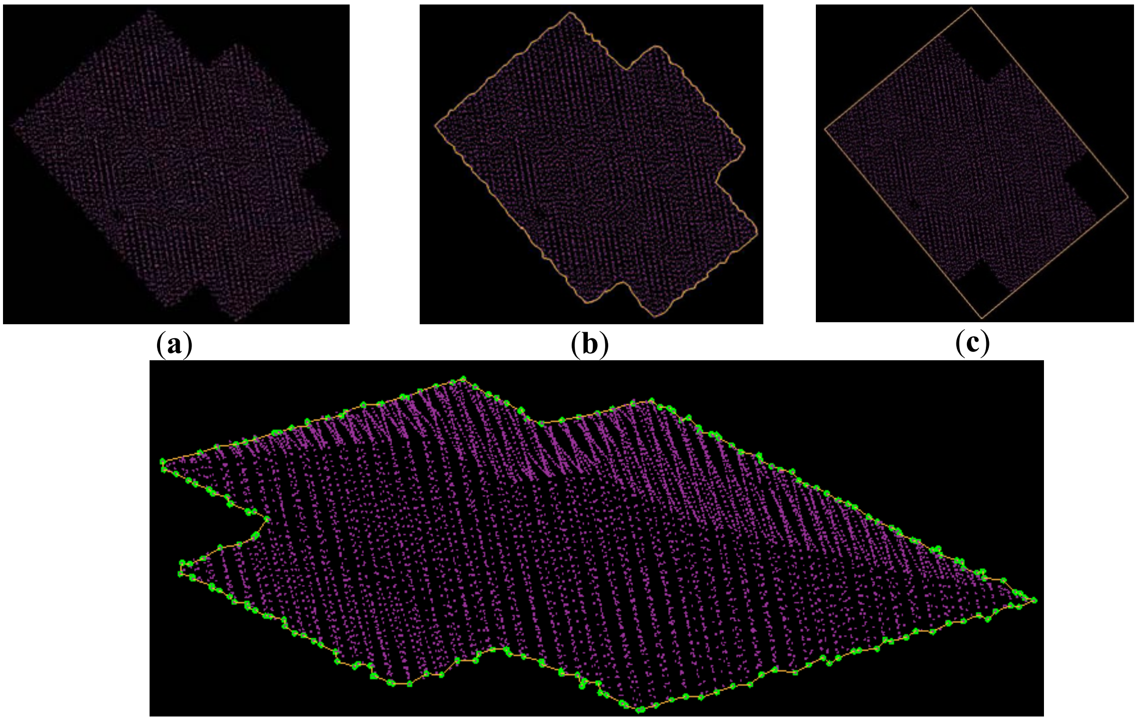
(**g**)

Figure 5. Transformation of points of one segment to the 2D space of the best-fit plane. (**a**) finding a point p₀ on the plane, (**b**) translating the coordinate system to make the origin coincide withp₀, and the two rotation angles, α and β , involved, (**c**) rotating the coordinate system to make the *z* axis coincide with the normal vector.



(**a**) (**b**) (**c**)

Figure 6. Calculation of 2D geometric properties of a cluster of points. (**a**) The transformed points into horizontal plane. (**b**) Concave boundary of the points. (**c**) MOBB (minimal oriented bounding box) of the points. (**d**) 3D boundary of the points, and the boundary points are colored in green.



(**a**) (**b**) (**c**)

(**d**)

2.3. Features Calculation

The employed features refer to geometry, echo, radiometry and topology.

(1) Geometrical features of the segments contain eight ones, including size/area (Section 2.2), orientation, rectangularity, elongatedness, compactness, average point-ness, average curve-ness and average surface-ness. The latter seven features are defined as below.

Orientation is defined as the angle between the best-fit plane (Section 2.2) and the horizontal plane, as shown in Figure 5a.Rectangularity is calculated as the ratio of the segment area and its MOBB area, elolngatedness is calculated as the ratio of the length of the short edge of MOBB and the length of the long edge of MOBB, and compactness can be calculated as the ratio of the segment area and square perimeter.

The point-ness, curve-ness (see Figure 7c) and surface-ness reflect the local distribution of the 3-D points. The distribution is captured by the decomposition into principal components of the covariance matrix of the 3-D points computed in a local neighborhood, the support region. The local neighborhood can be defined as the k nearest neighbors (KNN) in 3D space with k-d tree [41], and kdefines the scale of the features. The symmetric covariance matrix for a set of k 3-D points $\{X_i\} = \{(x_i, y_i, z_i)^\mathrm{T}\}$ with $\bar{X} = \sum_{i=0}^{k} X_i$ is defined in $\frac{1}{k}\sum(X_i - \bar{X})\,(X_i - \bar{X})^\mathrm{T}$.

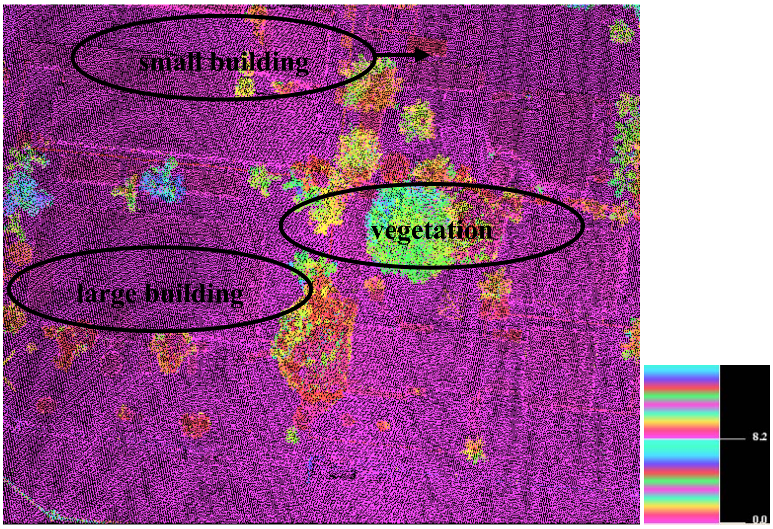
The matrix is decomposed into principal components ordered by decreasing eigenvalues. e_0,e_1 and e_2 are the eigenvectors corresponding respectively to the three eigenvalues $\lambda_0, \lambda_1, \lambda_2$ where $\lambda_0 \geq \lambda_1 \geq \lambda_2$. We use a linear combination of the eigenvalues as done by Lalonde *et al.* [42], see Equation 2, to represent the three saliencies, named as point-ness, curve-ness and surface-ness.

$$\begin{bmatrix} \text{point} - \text{ness} \\ \text{curve} - \text{ness} \\ \text{surface} - \text{ness} \end{bmatrix} = \begin{bmatrix} \lambda_2/\lambda_0 \\ (\lambda_0 - \lambda_1)/\lambda_0 \\ (\lambda_1 - \lambda_2)/\lambda_0 \end{bmatrix} \tag{2}$$

For each segment, the average of point-ness, average of curve-ness and average of surface-ness are selected as three features.

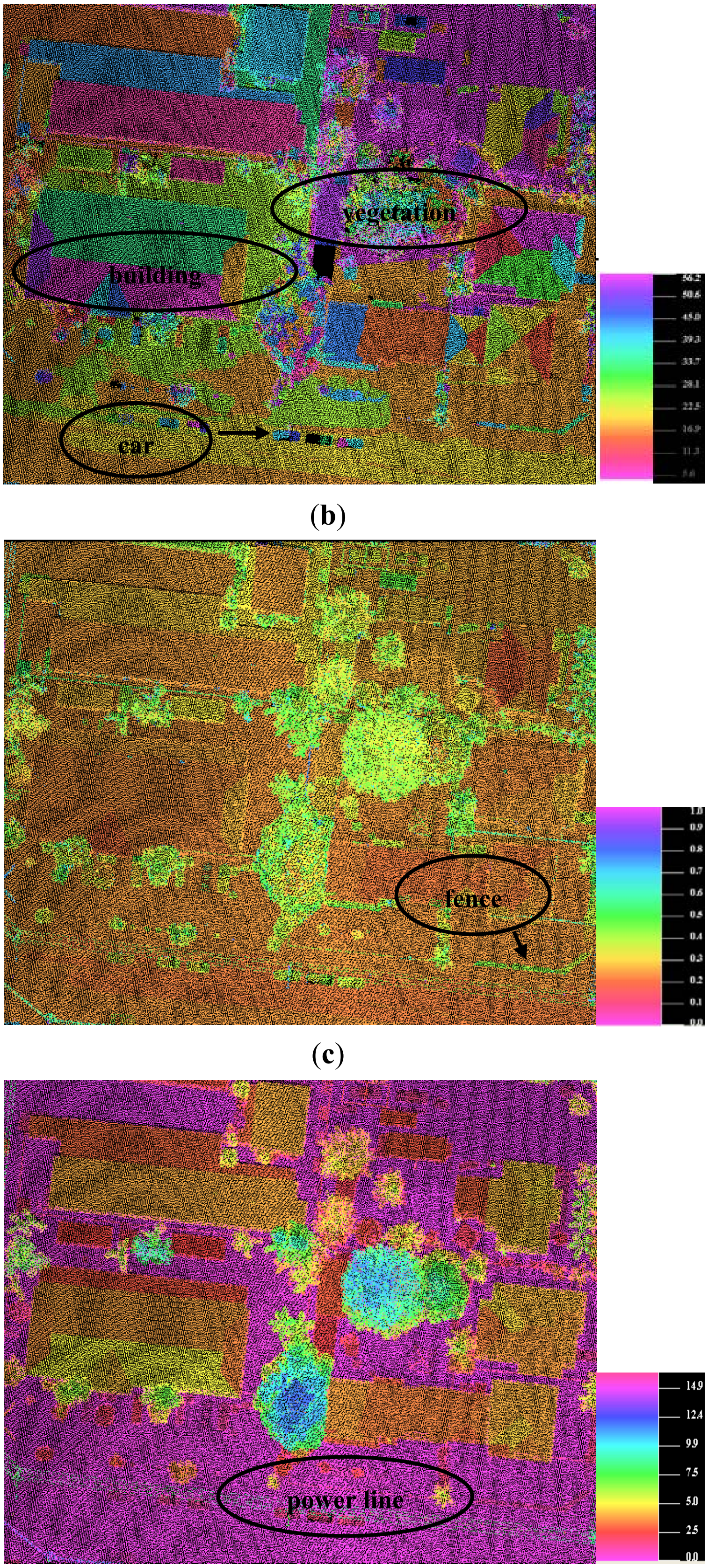
(2) Echo features of the segments contain two ones, including the average of elevation differences between the first and last echoes (see Figure 7a), and the proportion of multiple echoes. The latter is defined as follows.

Figure 7. Some features of the segments for classification. (**a**) Average of elevation differences between first and last echoes. (**b**) Average intensity. (**c**) Average curve-ness. (**d**) Average of elevation differences between boundary points and their lowest points.



(**a**)

Figure 7. *Cont.*



(**b**)

(**c**)

(**d**)

Currently, most of the ALS systems are capable of recording more than one echo, up to five echoes, as shown in Figure 4d,e. Particularly, the multi-pulse ALS system is capable of recording both singlereturns/echoes (as shown in Figure 4d) and multiple returns(as shown in Figure 4e), and the difference between the above two types of echoes is that whether a laser pulse would occur multiple reflections. The proportion of multiple echoes is defined as the ratio of multiple returns to the number of points in the segment [21].

(3) Radiometrical feature refers to the average intensity.

Most ALS systems are able to measure and record the backscattered energy of the emitted laser pulse. Since the measured intensity depends on the sensor to object distance, the intensity values have to be normalized by an average distance. The normalized intensity strongly differs for most the construction materials and vegetation surfaces [43], as shown in Figure 7b. Note the intensity is also relevant to the wavelength of the laser. For example, vegetations often have high reflectance compared to the other types of objects when the near-infrared laser is employed. Thus, average intensity of the points in a segment is selected as one of the features for classification.

(4) Topological features of a segment contain two ones, including average of elevation differences between boundary points and their lowest points, and average of slopes between boundary points and their lowest points.

Most filters assume that segments higher than their neighbors are objects, and the analysis of neighboring segments is based on their topological relation. The features derived from the analysis of the topological relationships are called topological features in this paper. Two topological features are contained, including elevation difference and slope. For determination of the above two features, the 3D contour lines and the points on the contours of each segment have to be extracted, as shown in Figure 6d. For each point on the 3D contour lines, find the lowest point with a fixed radius, R, in 2D space, and note that the points belonging to the current segment should be excluded when the lowest point is searched. Calculate the elevation difference and slope between the boundary point and the lowest point. The average of elevation difference and slope of the boundary points is regarded as the elevation difference and the slope of the segment, and they are input for the subsequent classification. As shown in Figure 7d, the feature about the elevation difference is useful to distinguish the object points/segments from the terrain points/segments.

Totally, there are thirteen features derived for the subsequent classification.

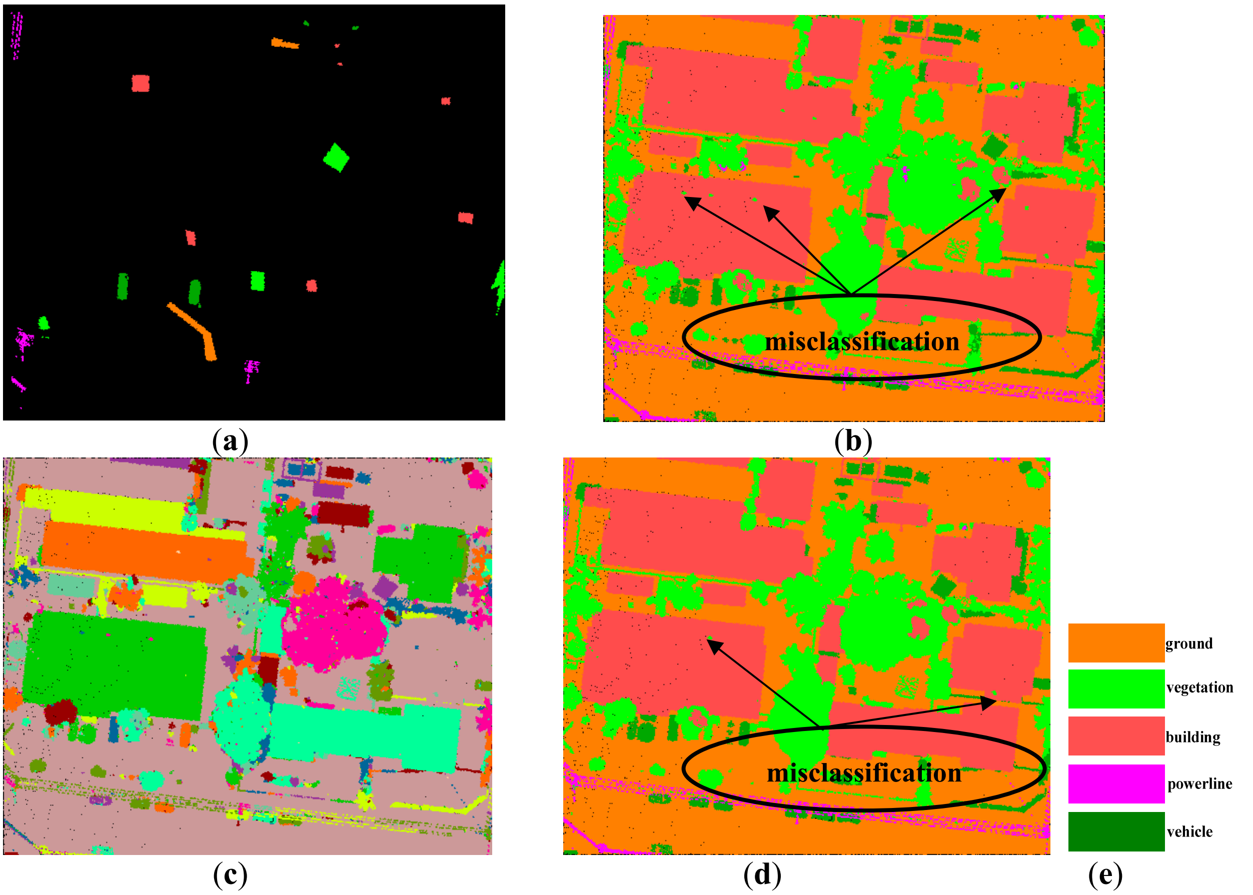
2.4. SVM-Based Classification

SVM is a mathematical tool, which is based on the structural risk minimization principle, and it tries to find a hyperplane in high dimensional feature space to solve some linearly inseparable problems [44]. The SVM has been widely applied within the remote sensing community to multi-spectral and hyperspectral imagery analysis. The kernel function is introduced into the SVM so that the original input space can be transformed non-linearly into a higher dimensional feature space where linear methods may be applied. A Gaussian radial basis function (RBF) is used in this study since it has been proved effective in many classification problems:

$$\mathbf{K}(\boldsymbol{x}_{\mathrm{i}} \cdot \boldsymbol{x}) = \mathbf{exp}\left(-\boldsymbol{\gamma}\left|\left|\boldsymbol{x}_{\mathrm{i}} - \boldsymbol{x}\right|\right|^{2}\right) \quad (3)$$

where γ is the RBF kernel parameter. Based on the training data in Figure 8a, the classification result of the point cloud in Figure 2d with the input features, as shown in Figure 8b, is obtained using the SVM, in which the γ is set 0.2 and C is set 100. The classification result suggests that most of the points are correctly classified by the SVM classifier, despite that some of them are misclassified.

Figure 8. Support vector machine (SVM)-based classification and optimization of the result. (**a**) Training samples. (**b**) Classification result using SVM. (**c**) Connected component analysis for the point cloud. (**d**) Final classification after sieving. (**e**) Legend of (a), (b) and (d).



2.5. Post-Processing

Figure 8b shows that, in the original classification result, some micro-objects such as dormers and chimneys on building roofs are likely to be misclassified, because they are similar to the vegetation in their features about size, proportion of multiple echoes. Thus, the classification result needs refinement if the small segments are reclassified into nearby larger segments. This refinement needs connected-component labeling. Traditionally, connected-component labeling is utilized in computer vision to detect connected regions in binary digital images, although color images and data with higher-dimensionality can also be processed [45]. A new connected component analysis for point clouds (CCAFPC) is proposed to finish the task. Fixed distance neighbors (FDN) is selected to search a local neighborhood of points around each point for CCAFPC. The advantage of FDN is the range of the neighborhood will be limited to the immediate local area. The distance metric used is Euclidean. Search for FDN can be optimized using different space partitioning strategies like k-d tree. With the help of k-d tree, CCAFPC is realized as follows:

(1) Specify a distance threshold, $\mathrm{d_{search}}$.

(2) Set all 3D points as unlabeled, insert them into a stack, and building the k-d tree for the point cloud with class attribute.

(3) If all the points have been already labeled go to step (7).Otherwise, select the first point without label in the stack as the current seed, build an empty list of potential seed points, and insert the current seed into the list.

(4) Select the neighboring points of the current seed using FDN. The points that satisfy having the same class attribute as the current seed, add them to the list of potential seed points.

(5) If the number of members in the potential seed point list is larger than one, set the current seed to the next available seed, and go to step (4). Otherwise, go to step (6).

(6) Give the points in the list a new label, clear the list, and go to step (3).

(7) Finish the task of labeling.

The result of CCAPFC for the classification result in Figure 8b is displayed in Figure 8c.

As a result, the small segments can be removed by the subsequent iterative process:

(1) Specify a minimum number of points threshold, $\mathrm{N_{search}}$.

(2) Perform CCAPFC.

(3) Calculate the number of points in each connected component, if the number of points of each component is larger than $\mathrm{N_{search}}$, finish the task.Otherwise, go to next step.

(4) Find a connected component without being processed, and the number of points in the component is no more than $\mathrm{N_{search}}$. Select the connect component as the current region, and go the next step.

(5) For each point in the current region, select the neighboring points of the current point using FDN. For each neighboring point, if the neighboring point has a different component label as the current region, add the component label of neighboring point into the region adjacency graph (RAG). Note that the component label could insert into the RAG only if it was not existed in the RAG.

(6) Label the state of the current region as being processed. Find the component with maximum number of points among the components in the RAG. If the maximum number is larger than $\mathrm{N_{search}}$, reclassify the points of current region into the class, which the component with maximum number belongs to, and the points of current region are relabeled as the label of the component with maximum number. Go to step (4).

(7) Finish this iteration. If there is no change of reclassifying happens, finish the task. Otherwise, go to step (3).

The refined classification of initial classification in Figure 8b is displayed in Figure 8d, and $\mathrm{d_{search}}$ is set to 0.5 as default value, and $\mathrm{N_{search}}$ is set to 50 as default value. As the result shows, most of the small regions can be removed after the above post-processing is performed, but some small regions with the number of points larger than 50 can't be correctly reclassified.

3. Experiments and Performance Analysis

A prototype system for object-based point clouds classification is developed in VC++6.0 IDE plus OpenGL library under Windows XP OS to verify our approach. Particularly, surface growing algorithm is implemented from scratch, and the triangulation of the ALS points was done by integrating an existing implementation of a 2D Delaunay triangulator called Triangle [46]. For the implementation of SVM, the software package LIBSVM by Chang and Lin [47] was adapted. Moreover, for the implementation of k-d tree, the software package ANN by [48] was adapted.

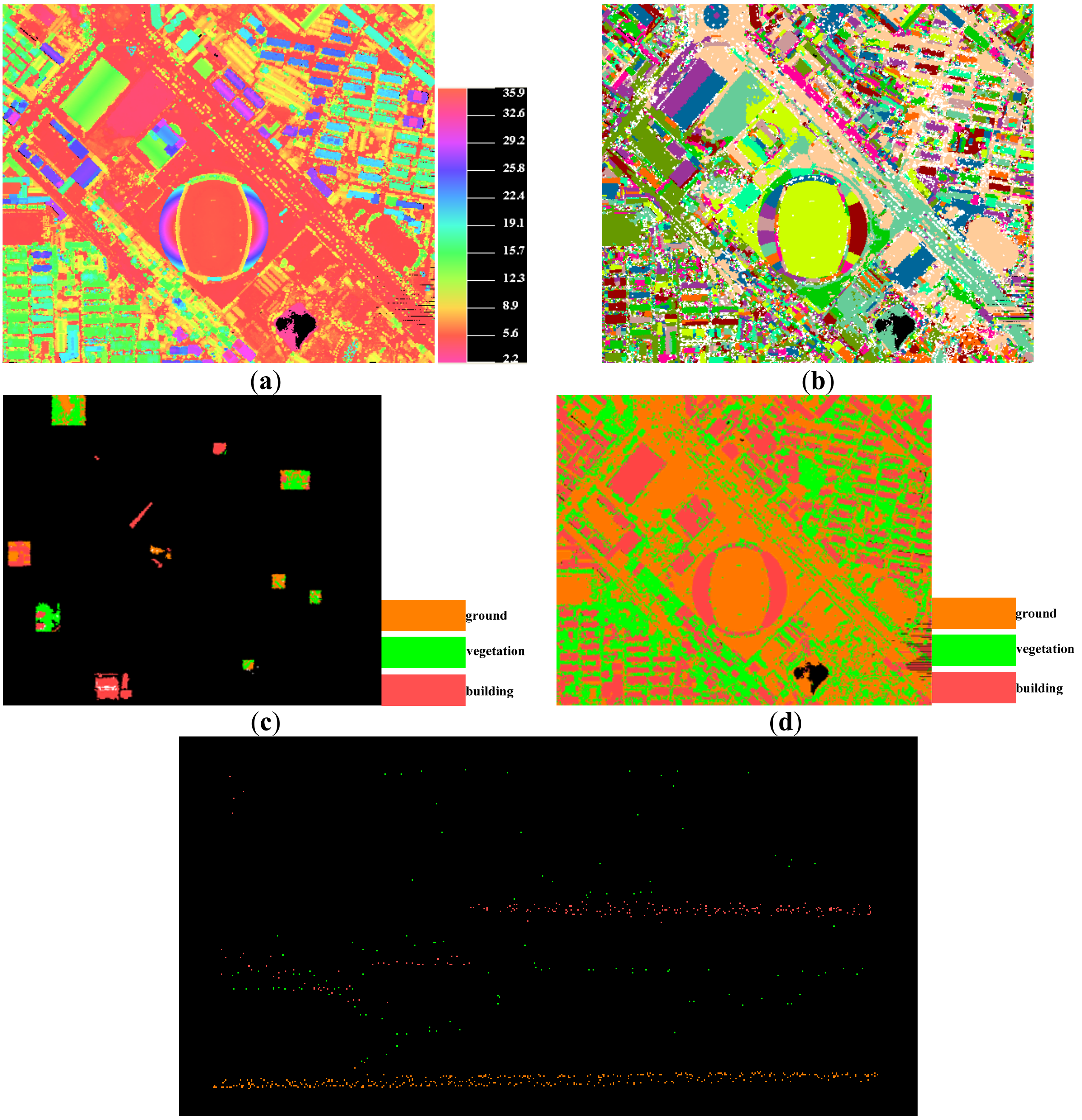
3.1. The Testing Data

Three ALS point cloud datasets are used to test our approach. The first data is near a town center of Shenyang City in northeast China, as shown in Figure 9a. The data was scanned by the Leica Geosytems ALS-50 system in September 2008. This site covers an area of 809 × 673 m with a point density of 1.2 points per m^2. Totally, there are 543082 measurements after outlier removal. The maximum number of returns of per pulse reaches to 2. The elevation ranges from 2.2 to 35.9 m. The ground surface is flat, but the buildings are complex. The heights, sizes and shapes of the buildings vary very much, however most of the building roofs are rectangular except the stadium in the center of the data, as shown in Figure 9a. Moreover, there are some trees around the buildings and on the roadsides, and most of the trees are small and low.

The second data is over the town of Enschede in the Netherlands with 1000.00 m width and 731.69 m length, as shown in Figure 10a. The dataset was recorded with the FLI-MAP 400 system with approximately 1.6 points per m^2, and maximum number of returns of per pulse reaches to four. Totally, there are 1,034,317 measurements after outlier removal. The elevation ranges from 26.9 to 75.9 m for this data. Moreover, the situations of the ground, buildings, and trees are very similar to the ones of the first data.

The third data is over a residential area of Melbourne, Australia, with 160.39 m width and 412.23 m length, as shown in Figure 11a. The dataset is with a point density approximately 40.0 points per m^2, and maximum number of returns of per pulse reaches to 4. Totally, there are 1,701,949 measurements after outlier removal. The elevation ranges from 64.2to 88.1 m for this data. This landscape is quite different from the former two datasets. Particularly, the ground surface is flat in the top part of the data, but there is a slope in the bottom part. The main artificial objects contain houses, powerlines, vehicles and a highway. Furthermore, the houses have difference configuration types such as flat roof, gable roof, hip roof, double sloped hip roof *etc*. Moreover, there are some powerlines across the residential areas, and they are the highest objects in this site. Some of the powerlines are higher than the vegetation below them, but some of them are not higher than the adjacent vegetation, as shown in Figure 11e. Additionally, there is a highway across the data, but it is attached on the ground. There are some vehicles around the houses and on the highway. At last, there are many trees around the houses, and their sizes and heights of trees vary very much.

Figure 9. Point cloud classification of the first scene. (**a**) The point cloud without outliers, colored by height. (**b**) Segmented point cloud, colored by label number. (**c**) Training samples. (**d**) Final classification result by our method. (**e**) Side view of the points in the rectangular region marked in (d), where mini-structure points and wall points are misclassified.



(**a**)

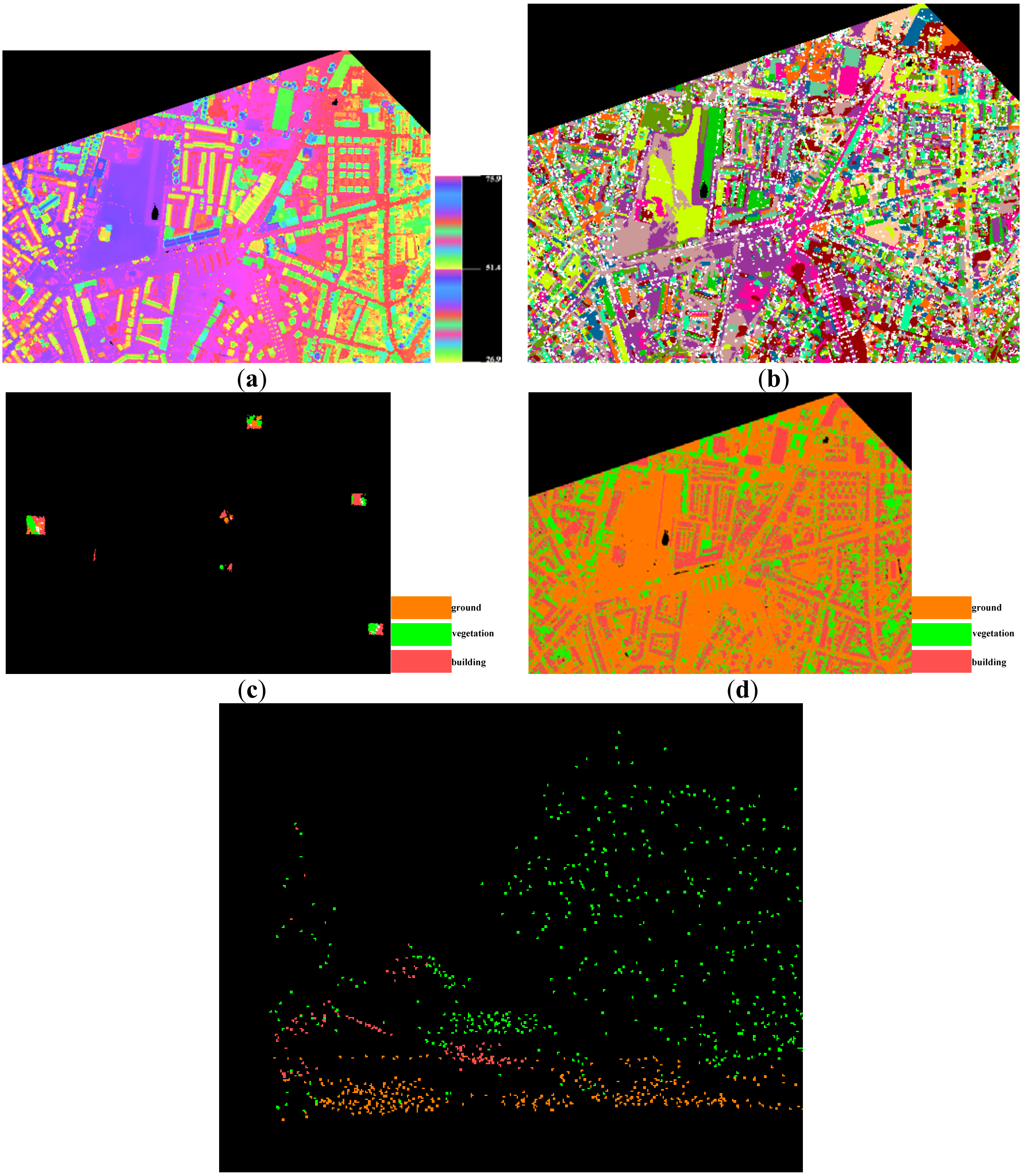
(**b**)

(**c**)

(**d**)

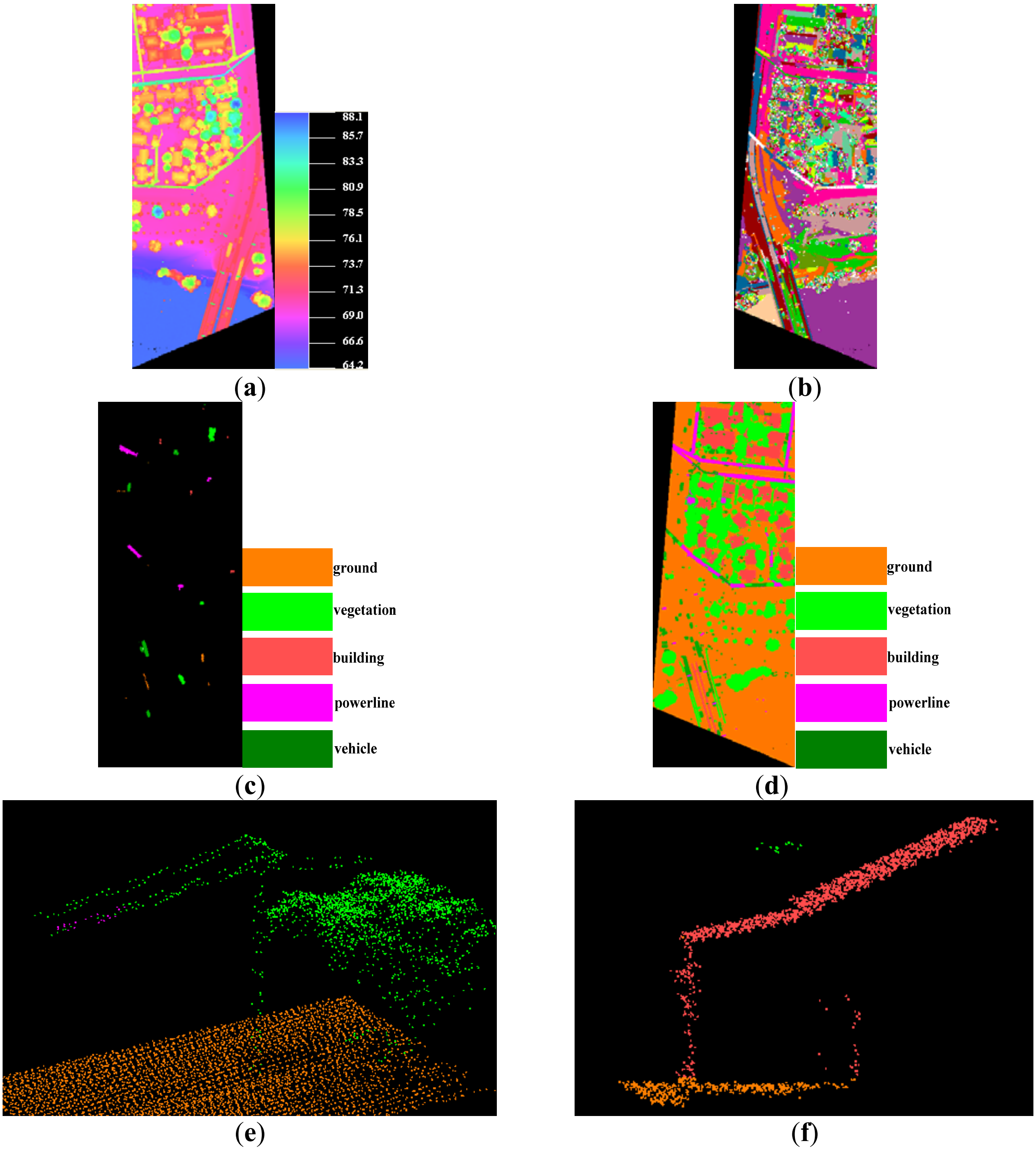
(**e**)

Figure 10. Point cloud classification of the second scene. (**a**)Point cloud without outliers, colored by height. (**b**) Segmented point cloud, colored by label number. (**c**) Training samples. (**d**) Final classification result by our method. (**e**) Side view of the points in the rectangular region marked in (d), where small buildings, below the trees, are misclassified.



(e)

Figure 11. Point cloud classification of the third scene. (**a**) The point cloud without outliers, colored by height. (**b**) Segmented point cloud, colored by label number. (**c**) Training samples. (**d**) Final classification result by our method. (**e**) Powerline measurements are misclassified as vegetation. (**f**) Mini-structure points are misclassified as vegetation.



3.2. Experimental Results

After removing the outliers, the three point clouds are segmented, and then the features are calculated. As mentioned above, there are total thirteen features, including size, orientation, rectangularity, elongatedness, compactness, average point-ness, average curve-ness and average surface-ness, average of elevation differences between the first and last echoes, and the proportion of multiple echoes, average intensity, average of elevation differences between boundary points and their lowest points, and average of slopes between boundary points and their lowest points. However, the intensity feature is not contained for the former two datasets, because the radiometric property is not available in the original data. In the sampling process, three classes of samples are selected for the former two datasets, and five classes of samples are selected for the last dataset. Note that the highway is regarded as the ground surface in the third testing data, and part of it is sampled. In the process of SVM-based classification, (C,γ) is set to (100.0, 0.33), (100.0, 0.33) and (100.0, 0.20) for the three datasets, respectively. Note that the determination of parameters C and$\gamma$is solved by cross validation and grid search on the training data set. At last, in the stage of post-processing, $\mathrm{d}_{\mathrm{search}}$ is set the default value, 0.5; and $\mathrm{N}_{\mathrm{search}}$is set to 5, 5 and 100 for the three datasets, respectively. The experimental results of three testing data are shown in Figures 9d, 10d and 11d, respectively.

A visual inspection tells us that our proposed method works well on the three testing datasets. In the first and second scenes, ground measurements are well separated from the non-ground/object measurements, as shown in Figures 9d and 10d. Furthermore, most of the building points and vegetation points are correctly detected despite there is some misclassification. Especially, wall points and the points belonging to the small structures on the building roofs are subject to be misclassified as vegetation points due to their similarities, as shown in Figure 9e; the vegetation points are also subject to be misclassified as building points or ground points if they are very close to the buildings or the Earth ground. The former kind of misclassified points are generally far away from the building segments, thus, they can't be relabeled by our proposed post-processing method. In the third scene, as shown in Figure 11d, ground measurements are also well identified, and most of building points, vegetation points and vehicle points are also well divided up. However, most of the power line measurements are not well detected, because some power line measurements, close the vegetation, are subject to be misclassified as vegetation points, as shown in Figure 11e. In contrast, some vegetation points are also misclassified as powerline. Moreover, there is also some misclassification between vegetation points and building points. As shown in Figure 11f, similar to the case in Figure 10e, some building measurements are also labeled wrongly as vegetations. In a word, our method is not only capable of well distinguishing the ground points from the object points, but also capable of correctly detecting most of the building points, vehicle points and vegetation points.

3.3. Evaluation

In order to quantitatively evaluate our proposed algorithm, we perform a semi-automatic classification of the above three datasets based on the TerraSolid software. Moreover, manual correction is performed to guarantee the quality of classification. The outputs of human-guided classification are regarded as reference data. In the reference data, the wall points and the dormer points are labeled as building points. Thus, a confusion matrix is derived by comparing the automatic classification results and manual classification results, as shown in Table 1.

Table 1. Confusion matrixes of the three testing scenes and their Kappa coefficients.

Class		Ground (Points)	Vegetation (Points)	Building (Points)	Vehicle (Points)	Powerline (Points)	Total (Points)	User Accuracy (%)
Ground (points)	Scene1	315,256	1,549	5,042	-	-	321,847	97.95
	Scene2	684,904	15,495	8,985	-	-	709,384	96.55
	Scene3	1,152,469	752	463	1	0	1,153,685	99.89
Vegetation (points)	Scene1	2,871	65,211	8,770	-	-	76,852	84.85
	Scene2	1,934	62,423	3,050	-	-	67,407	92.61
	Scene3	1,642	343,751	2,596	1,474	2057	351,520	97.79
Building(points)	Scene1	807	22,583	120,993	-	-	144,383	83.80
	Scene2	4,841	16,250	236,435	-	-	257,526	91.81
	Scene3	37	1,569	172,088	522	0	174,216	98.78
Vehicle (points)	Scene3	319	507	317	13,225	0	14,368	92.04
Powerline (points)	Scene3	0	1,268	0	0	6874	8,142	84.24
Total (points)	Scene1	318,934	89,343	134,805	-	-	543,082	-
	Scene2	691,679	94,168	248,470	-	-	1,034,317	
	Scene3	1,154,467	347,865	175,464	15,222	8931	1,701,949	
Producer accuracy (%)	Scene1	98.85	72.99	89.75	-	-	-	
	Scene2	99.02	66.28	95.15	-	-		
	Scene3	99.83	98.82	98.08	86.88	76.97		
Overall accuracy (%)	Scene1				92.34			
	Scene2				95.11			
	Scene3				99.20			
Kappa	Scene1				0.8638			
	Scene2				0.8972			
	Scene3				0.9837			

The statistics suggest that our proposed method has high classification accuracy in the three testing datasets. Particularly, the overall classification accuracy is larger than 92.34%, and the Kappa coefficient is not lower than 0.8638. Additionally, most classes obtain high classification accuracy. For example, user accuracy is higher than 96.55% for the ground class, 83.80% for building class, and 92.04% for the vehicle class. As far as producer accuracy is concerned, it is higher than 98.85% for the ground class, 89.75% for building class, and 86.88% for the vehicle class. Moreover, the statistics in Table 1 also show that the classification accuracy is promoted with the increase of the point density. As mentioned above, the point density of the three testing data is 1.2 points per m^2, 1.6 points per m^2, and 40.0 points per m^2, respectively. Correspondingly, the overall accuracy is 92.34%, 95.11% and 99.20%, respectively; Kappa coefficient is 0.8638, 0.8972, and 0.9837 for the three scenes. Both overall accuracy and Kappa coefficient have increasing trends with the increasing of the point densities. At last, the statistics in Table 1 tell us that different classes have quietly different classification accuracies. On average, the producer accuracy is 99.23% for ground class, 94.32% for building class, 92.04% for the vehicle class, 84.24% for the power line class, and 79.36% for vegetation class. Obviously, there is a large difference between the ground class and the vegetation class in the producer accuracy. Overall, our method obtains a desired accuracy in point cloud classification if just few classes are preset.

4. Uncertainties, Errors, and Accuracies

The above experiments suggest that our proposed classification method obtains good results. However, there are still some errors in the classification. Particularly, there are a few misclassified points between the vegetation class and building class. For example, there is 11.41% of vegetation points being misclassified as building points in the first scene, and 15.64% of building points being misclassified as vegetation points in the first scene. Similar errors also happen in the second and third scene. Generally, several concurred circumstances may bring about the vegetation points are subject to be misclassified as the building points: (1) some buildings are as low as vegetation, thus the height difference between the vegetation and the building is not significant enough, as illustrated in Figure 10e; (2) the vegetation canopies are dense enough so that the laser pulse is not capable of penetrating them. On the other hand, the following factors may lead to the building points are subject to be misclassified as the vegetation points: (1) the building roofs are not large or flat enough; (2) the building façade points are not dense and co-planar enough; (3) the micro objects on the roofs are very similar to the vegetation in their features, as shown in Figure 11f. Moreover, misclassification also often occurs between the vegetation class and the powerline class, especially when they are close enough. In the third scene, there is 15.76% of powerline measurements wrongly regarded as the vegetation class, as shown in Figure 11e. In a short, these kinds of errors derive from their similarities in feature space, however, there are lots of uncertainties.

After analyzing the classification results and the statistics in Table 1, we can get the following conclusions:

(1) Compared to the object points, the ground/terrain points are more likely to be correctly detected by our proposed approach, which is meaningful for the filtering and generation of DTM. Among the three scenes, the ground measurements are scarcely misclassified into other classes, and the producer accuracy and the user accuracy is higher than 96.55% and 98.85% respectively for the ground class. In all of the three tests, the ground class obtains higher classification accuracy than the vegetation and building class. On average, the producer accuracy of ground class is 6.38% higher than the vegetation class, and the user accuracy of ground class is 19.87% higher than the vegetation class. Similar statistics are 6.66% and 4.91% for the ground class compared to the building class. This should contribute to the following factors. First, the ground segments are usually larger than the other types of segments. Second, the sizes of the ground segments trend to distribute into a fixed range. Third, the ground surfaces seldom yield multiple echoes. Fourth, the ground surfaces usually are horizontal in the urban areas, and they are easy to be clustered into one same segment in point cloud segmentation. Last but not least, the ground surfaces may be the lowest part in a local neighborhood. The above factors lead the ground class having lower complexity than the other types of objects.

(2) Compared to the other types of object points, the building measurements are more likely to be correctly identified by our proposed approach. Among the three tests, the building class gets higher

classification accuracy than the vegetation, the vehicle and the powerline. For example, the user accuracy of building class is approximately 15% higher than the vegetation class on average. This should contribute to the following factors. First, most of the building roofs are planes, and they are often very large. Thus, the building measurements trend to be clustered into the same and large segment after point cloud segmentation. Second, a building roof trends to produce single returns rather than the multiple returns, which yields a difference from the trees. As a result, the building class is easier to be separated from the vegetation class.

(3) The wall points and the points, belonging to the small objects on the building roofs, are more likely to be wrongly regarded as vegetation points, as shown in Figure 9e and Figure 11f. The misclassification comes from that the wall points and the mini-structures on the building roofs often have similar features as the trees. Both of them are small in size, and the mini-structures also produce multiple echoes if they are high enough. The above reasons lead to the low classification accuracy for the vegetation class, as low as approximate 66% for the producer accuracy in the second scene. Fortunately, once misclassification occurs, some of the errors may be removed by our post-processing method, because these kinds segments sometimes are far away from the building segments, as shown in Figure 9e.

(4) The powerline may be not suitable to be recognized by our proposed method, as shown in Figure 11. In the third scene, the powerline class has low classification accuracy. The user accuracy and the producer accuracy is 84.24% and 76.97%, respectively. The low classification accuracy is derived from the following factors. Generally, the powerline in the urban areas is not far away enough from the adjacent trees. Moreover, the power line also trends to produce multiple echoes same as the trees. As a result, a lot of powerline measurements are subjective to be misclassified as vegetations in the urban areas.

(5) Errors also trend to occur where there is a mixture of different classes of points. As shown in Figure 10e, the small and low building measurements around the trees are misclassified as vegetations. Other similar errors often occur. The mixture could produce similar features such as same height, similar proportion of multiple echoes *etc*. Mixture increases the complexity of a landscape, which is a difficult problem to be solved.

(6) Point density of a point cloud has a significant influence on the classification accuracy. As illustrated above, our proposed classification method trend to correctly detect the objects with a larger size if the point density is fixed. On contrary, if the point density of a point cloud is increased, the classification accuracy is also promoted. In the three test dataset, the point cloud density is getting larger and larger. Correspondingly, the overall classification accuracy and the kappa coefficient are also getting higher and higher. Moreover, in the former two scenes, only three classes of points, including ground, building and vegetation, are obviously distinguished, because the point density is low for the two data, and it is insufficient for detecting small objects. However, in the third scene, five classes of points, including vehicle and powerline plus the former three classes are capable of being detected. Furthermore, most of the vehicles are corrected recognized, but the vehicles can’t be recognized by the human beings in the former two data. With the increasing of the point density, the

number of points in each face is also increased, which leads to a larger segment in point cloud segmentation. Thus, the classification accuracy is increased for the whole dataset.

5. Conclusions

Airborne LiDAR point cloud classification is meaningful for various applications, but it is a challenging task due to its complexity. In this article, classification is based upon object-based point cloud analysis, which defines the general two steps: clustering and classification. Particularly, surface growing algorithm is employed to make a clustering, and SVM is used to classify the points in a segment-wise style. There are two main contributions of the paper: (1) the use of SVM to classify surface growing segmented objects (OBPA approach); (2) the proposed connected-component analysis for 3D point clouds. The experimental results suggest that our method is capable of making a good coarse urban scene classification. Particularly, most of the ground points and building points are correctly identified, with producer accuracy higher than 98.13% and 91.46%, respectively. Moreover, most of vegetation measurements are also correctly recognized by our method, and its user accuracy is higher than66.28% in the three tests. Furthermore, our method is capable of detecting the vehicles if the point cloud density is large enough. However, our method often fails to extract the powerlines due to their similarity to the vegetation. In a word, the experiments show that OBPA approach has lot of advantages in urban scene classification, and it is capable of separating the point clouds with a sound accuracy, which is meaningful to the filtering and other kinds of application. However, the proposed method still has some limitations such as much misclassification of building and vegetation, time-consuming feature calculation. The future work will also focus on improvement of the surface growing algorithm, adoption of conditional random fields for topological analysis to increase the accuracy of classification.

Acknowledgments

We thank the three anonymous reviewers for their constructive suggestions. This research was funded by: (1) Project for Young Scientist Fund sponsored by the Natural Science Foundations of China (NSFC) under Grant 41001280; (2) the National High Technology Research and Development Program ("863 Program") under Grant 2011AA120405-01; (3) the Scientific and Technological Project for National Basic Surveying and Mapping with title Method Research, Software Development and Application Demonstration for Object-oriented Mobile Laser Scanning Point Clouds Classification and 3D Reconstruction of Facades. The Enschede data set is kindly provided by Faculty of ITC, Twente University; and the Melbourne data set is kindly provided by ROAMES Research Team at Ergon Energy in Australia.

Conflict of Interest

The authors declare no conflict of interest.

References and Notes

1. Filin, S.; Pfeifer, N. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS J. Photogramm*. **2006**, *60*, 71–80.
2. Sithole, G.; Vosselman, G. Experimental comparison of filter algorithms for bare earth extraction from airborne laser scanning point clouds. *ISPRS J. Photogramm*. **2004**, *59*, 85–101.
3. Maas, H.G.; Vosselman, G. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS J. Photogramm*. **1999**, *54*, 153–163.
4. Oude Elberink, S.; Vosselman, G. 3D information extraction from laser point clouds covering complex road junctions. *Photogramm. Rec*. **2009**, *24*, 23–36.
5. Hyyppä, J.; Schardt, M.; Haggrén, H.; Koch, B.; Lohr, U.; Scherrer, H.U.; Paananen, R.; Luukkonen, H.; Ziegler, M.; Hyyppä H., *et al*. HIGH-SCAN: The first European-wide attempt to derive single tree information from laser scanner data. *Photogram. J. Finl*. **2001**, *17*, 58–68.
6. Ussyshkin, V.; Theriault, L. Airborne Lidar: Advances in discrete return technology for 3D vegetation mapping. *Remote Sens.* **2011**, *3*, 416–434.
7. Zhang, J. Multi-source remote sensing data fusion: status and trends. *Int. J. Image Data Fusion* **2010**, *1*, 5–24.
8. Antonarakis, A.S.; Richards, K.S., Brasington, J. Object-based land cover classification using airborne LiDAR. *Remote. Sens. Environ*. **2008**,*112*, 2988–2998.
9. Fricker, G.A.; Saatchi, S.S.; Meyer, V.; Gillespie, T.W.; Sheng, Y. Application of semi-automated filter to improve waveform LiDAR sub-canopy elevation model. *Remote Sens.* **2012**, *4*, 1494–1518.
10. Mallet, C.; Bretar, F.; Soergel, U. Analysis of full waveform LIDAR data for classification of urban areas. *Photogramm. Fernerkun*. **2008**, *5*, 337–349.
11. Rutzinger, M.; Höfle, B.; Hollaus, M.; Pfeifer, N. Object-Based point cloud analysis of full-waveform airborne laser scanning data for urban vegetation classification. *Sensors* **2008**, *8*, 4505–4528.
12. Huang, X.; Zhang, L.P.; Gong, W. Information fusion of aerial images and LIDAR data in urban areas: vector-stacking, re-classification and post-processing approaches. *Int. J. Remote. Sens.* **2011**, *32*, 69–84.
13. Secord, J.; Zakhor, A. Tree detection in urban regions using aerial LiDAR and image data. *IEEE. Geosci Remote Sens.* **2007**, *4*, 196–200.
14. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubik, K. Building detection by fusion of airborne laser scanner data and multi-spectral images: performance evaluation and sensitivity analysis. *ISPRS J. Photogramm*. **2007**, *62*, 135-149.
15. Zhu, X.K.; Toutin, T. Land cover classificationusing airborne LiDAR products in Beauport, Québec, Canada. *Int. J. Image Data Fusion* **2012**, doi: 10.1080/19479832.2012.734339.
16. Liu, X. Airborne LiDAR for DEM generation: some critical issues. *Progr. Phys. Geogr.* **2008**, *32*, 31–49.
17. Axelsson, P.E. DEM generation from laser scanner data using adaptive TIN models. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2000**, *32*, 110–117.

18. Zhang, J.X.; Lin, X.G. Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS J. Photogramm.* **2013**, *81*, 44–59.
19. Kraus, K.; Pfeifer, N. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm.***1998**, *53*, 193–203.
20. Briese, C.; Pfeifer, N.; Dorninger, P. Applications of the robust interpolation for DTM determination. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 55–61.
21. Darmawati, A.T. Utilization of Multiple Echo Information for Classification of Airborne Laser Scanning Data. M.Sc. Thesis, International Institute for Geo-information Science and Observation, Enschede, The Netherlands, 2008.
22. Meng, X.; Wang, L.; Currit, N. Morphology-based building detection from airborne Lidar data. *Photogramm. Eng. Remote Sensing* **2009**, *75*, 437–442.
23. Yao, W.; Hinz, S.; Stilla, U. Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis. *ISPRS J. Photogramm.* **2011**, *66*, 260–271.
24. Oude Elberink, S.; Mass, H.G. The use of anisotropic height texture measurements for the segmentation of ariborne laser scanner data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2000**, *33*, 678–684.
25. Shan, J.; Sampath, A. Urban DEM generation from raw Lidar data: a labeling algorithm and its performance. *Photogramm. Eng. Remote Sensing* **2005**, *71*, 217–226.
26. Tinkham, W.T.; Huang, H.; Smith, A.M.S.; Shrestha, R.; Falkowski, M.J.; Hudak, A.T.; Link, T.E.; Glenn, N.F.; Marks, D.G. A Comparison of two open source LiDAR surface classification algorithms. *Remote Sens*. **2011**, *3*, 638–649.
27. Petzold, B.; Axelsson, P.Result of the OEEPE WG on Laser data acquisition. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2000**, *33*, 718-723.
28. Brovelli, M.A.; Cannata, M.; Longoni, U.M. Managing and Processing LIDAR Data within GRASS. In Proceedings of Open Source GIS: GRASS Users Conference, Trento, Italy, 11–13 September 2002.
29. Shen, J.; Liu, J.P.; Lin, X.G.; Zhao, R. Object-based classification of airborne light detection and ranging point clouds in human settlements. *Sensor Lett*. **2012**, *10*, 221–229.
30. Sithole, G. Segmentation and Classification of Airborne Laser Scanner Data. Ph.D Thesis, Netherlands Commission of Geodesy, Delft, The Netherlands, 2005.
31. Golovinskiy, A.; Kim, V.G.; Funkhouser, T. Shape-based Recognition of 3D Point Clouds in Urban Environments. In Proceedings of the 12th International Conference on Computer Vision (ICCV09), Kyoto, Japan, 27 September–4 October 2009; pp. 2154–2161.
32. Barsi, Á. Object detection using neural self-organization. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *35*, 366–371.
33. Benz, U.C.; Peter, H.; Gregor, W.; Lingenfelder, I.; Heynen, M. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS J. Photogramm.* **2004**, *58*, 239–258.

34. Silván-Cárdenas, J.L.; Wang, L. A multi-resolution approach for filtering LiDAR altimetry data. *ISPRS J. Photogramm.***2006**, *61*, 11–22.
35. Vosselman, G.; Klein, R. Visualization and Structuring of Point Clouds. In *Airborne and Terrestrial Laser Scanning*, 1st ed.; Vosselman, G., Maas, H.G., Eds.; Whittles Publising: Dunbeath, UK, 2010; pp. 43–79.
36. Melzer, T. Non-parametric segmentation of ALS point clouds using mean shift. *J. Appl. Geod.* **2007**, *1*, 159–170.
37. Wang, M.; Tseng, Y.H. Automatic segmentation of LiDAR data into coplanar point clusters using an octree-based split-and-merge algorithm. *Photogramm. Eng. Remote Sensing* **2010**, *76*, 407–420.
38. Vosselman, G.; Gorte, B.G.H.; Sithole, G.; Rabbani, T. Recognizing structure in laser scanner point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *36*, 33–38.
39. Rabbani, T. Automatic Reconstruction of Industrial Installations Using Point Clouds and Iimages. Ph.D. Thesis, Netherlands Commission of Geodesy, Delft, The Netherlands, 2006.
40. Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. On the shape of a set of points in the plane. *IEEE T. Inform. Theory* **1983**, *29*, 551–559.
41. Arya, S.; Mount, D.M.; Netanyahu, N.S.; Silverman, R.; Wu, A.Y. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* **1998**, *45*, 891–923.
42. Lalonde, J.-F.; Vandapel, N.; Huber, D.; Hebert, M. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field. Robot.* **2006**, *23*, 839–861.
43. Hug, C.; Wehr, A. Detecting and identifying topographic objects in imaging laser altimeter data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **1997**, *32*, 19–26.
44. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer-Verlag: New York, NY, USA, 1995.
45. Samet, H.; Tamminen, M. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE T. Pattern Anal.* **1988**, *10*, 579–586.
46. A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator. Available online: http://www.cs.cmu.edu/~quake/triangle.html. (accessed on 11 October, 2011).
47. Chang, C.C.; Lin, C.J. *LIBSVM: A Library for Support Vector Machines*; 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed on 15 March 2010).
48. ANN: A Library for Approximate Nearest Neighbor Searching. Available online: http://www.cs.umd.edu/~mount/ANN/ (accessed on 15 March 2010).