Eigen-feature Analysis of Weighted Covariance Matrices for LiDAR Point Cloud Classification

Chao-Hung Lin[1], Jyun-Yuan Chen[1], Po-Lin Su[1], and Chung-Hao Chen[2]

[1]Department of Geomatics, National Cheng Kung University, Taiwan

[2] Department of Electrical and Computer Engineering, Old Dominion University, USA

E-mail: linhung@mail.ncku.edu.tw

Abstract

Light Detection and Ranging (LiDAR) sensors with the ability to acquire high spatial resolution and accurate 3D data over a large area are increasingly utilized in the fields of remote sensing and surveying. The classification of airborne LiDAR data is a fundamental and critical step in related applications. The features used in the separation of different objects are important for successful classification. Eigen-features from a covariance matrix of a point set with the sample mean are commonly used geometric features that can describe the local geometric characteristics of a point cloud and indicate whether the local geometry is linear, planar, or spherical. However, eigen-features calculated by the principal component analysis of a covariance matrix are sensitive to LiDAR data with inherent noise and incomplete shapes because of the non-robust statistical analysis. To obtain reliable eigen-features from LiDAR data and to improve classification accuracy, we introduce a method of analyzing local geometric characteristics of a point cloud by using a weighted covariance matrix with a geometric median rather than the standard covariance matrix and the sample mean which are sensitive to point distribution. Each point in the neighborhood of a point is assigned a weight to represent its spatial contribution in the weighted principal component analysis and to estimate the geometric median which can be regarded as a localized center of a shape. In the

experiments, qualitative and quantitative analyses on airborne LiDAR data and simulated point clouds show a clear improvement of the proposed method compared with the standard eigen-features. The classification accuracy is improved by 1.6% to 4.5% using a supervised classifier.

Keywords: point cloud classification, weighted covariance matrix, eigen-feature

1. Introduction

Classification is the process of converting raw data into meaningful, useful, and understandable information (Mountrakis et al., 2011). Although researchers have proposed many classification methods, LiDAR point cloud classification is not fully solved because of the similar characteristics possessed by different objects. Therefore, many researchers devoted their efforts in automatic point cloud classification. In related studies, LiDAR is used to separate ground surface from non-ground objects (Lohmann et al., 2000). Non-ground objects are further classified into buildings (Axelsson, 1999), roads (Choi et al., 2008), and vegetation (Cobby et al., 2003). Several researchers also combine LiDAR data with multispectral images in their classification (Bork and Su, 2007; Secord and Zakhor, 2007), and LiDAR intensity is used in feature extraction and land cover classification (Flood, 2001; Hui et al., 2008).

In the classification of LiDAR data, various techniques were proposed, including unsupervised classification (Haala and Brenner, 1999; Vosselman, 2000) and supervised classification, such as Bayesian networks (Stassopoulou et al., 2000), decision trees (Antonarakis et al., 2008), and support vector machines (SVMs) (Charaniya et al., 2004; Lodha et al., 2006; Secord and Zakhor, 2007; Samadzadegan et al, 2010; Mallet et al., 2011). In this study, the SVM classifier is used to evaluate the generated eigen-features. In SVMs, each point is transformed into a

high-dimensional space which is composed of a set of defined features. These features are derived from the LiDAR point clouds. Some features are defined based on waveform considerations (Gross et al., 2007; Wagner et al., 2008; Rutzinger et al., 2008), and some features are related to the geometric information of point clouds, such as surface slopes, point heights, and local geometric shapes (Charaniya et al., 2004; Lodha et al., 2006; Secord and Zakhor, 2007; Samadzadegan et al, 2010; Mallet et al., 2011). Among these features, eigen-features computed from eigenvalues of the covariance matrix of a local point set are the commonly used 3D geometric descriptors. Eigen-features and eigenvalues can be efficiently obtained by principal component analysis (PCA) (Gross et al., 2007; Mallet et al., 2011; Brodu and Lague, 2012; Gressin et al., 2013), which is a mathematical tool widely used in numerous studies such as point cloud segmentation and reconstruction (Sampath et al., 2010). Although PCA is a powerful tool in revealing geometric information of 3D point data, the method suffers from difficulties caused by outliers and the incomplete surface sampling nature of airborne LiDAR point clouds because of the non-robust statistical analysis. In this study, a eigen-feature analysis based on weighted covariance matrix and geometric median is introduced to alleviate these difficulties and to extract reliable eigen-features from point clouds. The geometric median is known as a robust center of an arbitrary point set and has the prominent property of sampling insensitivity (Daszykowski et al., 2007). To obtain the geometric median and the weighed covariance matrix, each point in a local point set is assigned a weight to represent its spatial contribution, i.e., the surface occupied by that point, in the weighted PCA. A possible solution in calculating the area of a point occupied surface is through the use of Voronoi diagram. The area of a point occupied surface can be defined as the area of a Voronoi cell in a projection plane. However, this definition suffers computational sensitivity in calculating the projection plane and constructing

the Voronoi diagram for unorganized and noisy point clouds. From the observation that the area of a point occupied surface is inversely proportional to the point density, the point weight is defined as a function of point density that is computationally much more efficient. The obtained weights are then used with the geometric median in calculating the covariance matrix and in extracting the eigen-features.

The main contribution of this study is the introduction of a method to extract eigen-features from a local point set by using the weighted PCA with the geometric median, which has better performance in classification accuracy then that using the standard eigen-features. The remainder of this paper is organized as follows. Section 2 introduces the proposed approaches. Section 3 discusses the experimental results, and Section 4 presents the conclusions and future work.

2. Methodology

This paper introduces an improved eigen-feature analysis for LiDAR point cloud classification. SVM classification and eigen-feature analysis are briefly introduced in Sections 2.1 and 2.2, respectively. The proposed approaches on the weighted covariance matrix and eigen-feature analysis are then described in Section 2.3.

2.1 SVM classification

SVMs were first introduced by Vapnik (1995) for classification and have recently become an intensive research topic. SVMs were originally designed for two-class linear classifications. The basic idea is to determine the maximal margin of the input samples, where margin means the minimal distance from the separating hyperplane to the closest samples of classes. The middle of the margin is the optimal separating hyperplane, and the samples closest to the hyperplane are called support vectors. Given that the determined hyperplane cannot separate the samples well, the

linear SVM is extended to a non-linear SVM by transforming the problem into a feature space using a set of nonlinear basis functions, where the samples are separated as clearly as possible. In doing so, the algorithm avoids the process of determining the optimal separating hyperplane in the feature space. A kernel representation is used instead, in which the solution is written as a weighted sum of the values of a kernel function evaluated at the support vectors. In addition, only a few training samples are required in the SVM algorithm. These properties make SVMs suited to high-dimensional classification problems in the field of remote sensing.

Given a set of training points $A = \{x_k, y_k\}_{k=1}^{n}$, where *n* represents the number of points, x_i is the training sample, and y_i is the corresponding class label. The decision function can be found by solving the following optimization equation:

$$\arg\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \quad (1)$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \text{and } 0 \le \alpha_i \le C, \quad (2)$$

where α_i is the Lagrange coefficient, *C* is the parameter that controls the trade-off between the training error and the margin, and $K(x_i, x_j)$ is the transformation kernel function. The effectiveness of SVM is dependent on the margin parameter *C*, the selection of the kernel function, and the parameters in the kernel function. The design and selection of a kernel is difficult without sufficient priori knowledge of the classified targets in the point clouds (Mallet et al., 2011). Therefore, the Gaussian kernel that contains only one parameter is adopted, that is,

$$K(x_i, x_j) = \exp\left(-k \left\| x_i - x_j \right\|^2\right) \text{ for } k > 0. \quad (3)$$

The best combination of the parameters *C* and k is generally decided through cross validation, that is, the parameters with the best cross-validation accuracy are selected. The final SVM model is trained on a training dataset by using the selected

parameters. The model is then used in testing and classifying point clouds.

The SVM classifier assigns a label to each point based on its feature representation. The features are computed from the LiDAR point cloud. Some features are related to point heights, echo numbers, and local geometric characteristics under spatial considerations. The others are driven based on waveform considerations. The traditional multiple-pulse LiDAR data are tested in this study. Therefore, the spatial features, including height-based, echo-based, and eigen-based features, are used in the experiments.

2.2 Geometric eigen-features

The PCA is an orthogonal transformation technique used to convert a set of points with possibly correlated variables to another set of points with linearly uncorrelated variables called principal components in which the first principal component has the largest variance and each successive component is orthogonal to the preceding components (Jolliffe, 1986). Given a 3D point set $\mathbf{P}=\{(x_i,y_i,z_i)\}_{i=1}^{n}$ within a neighborhood ball of diameter *r*, an efficient method to compute the principal components of the point set **P** is to diagonalize the covariance matrix of **P**. In matrix form, the covariance matrix of **P** is written as

$$\mathbf{C}(\mathbf{P})=\frac{\sum_{p_i\in\mathbf{P}} w_i (p_i-\bar{p})^T (p_i-\bar{p})}{\sum_i w_i}, \tag{4}$$

where $\bar{p}$ represents the mean of the points, that is, $\bar{p}=\sum_{i=1}^{n} p_i / n$, where n represents the number of points in **P**. w_i is the weight of point p_i, and generally, $w_i=1$. The eigenvectors and eigenvalues of the covariance matrix are then computed by using matrix diagonalization technique, that is, $\mathbf{V}^{-1}\mathbf{C}\mathbf{V}=\mathbf{D}$, where **D** is a diagonal matrix containing the eigenvalues $\{\lambda_1,\lambda_2,\lambda_3\}$ of **C**, and **V** is an orthogonal matrix that contains the corresponding eigenvectors. The obtained eigenvalues are

greater than or equal to zero, that is, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, because the covariance matrix is a symmetric semi-positive definite matrix. In geometry, the eigenvalues relate with an ellipsoid that represents the local geometric structure of a point set (Gumhold et al., 2001). $\lambda_1 >> \lambda_2, \lambda_3$ represents a stick-like ellipsoid, meaning a linear structure such as building edges. $\lambda_1 \cong \lambda_2 >> \lambda_3$ indicates a flat ellipsoid, representing a planar structure. $\lambda_1 \cong \lambda_2 \cong \lambda_3$ corresponds to a volumetric structure such as corners of buildings. Some combinations of these eigenvalues provide discriminating geometric features, especially for the point clouds in urban areas. Following the definitions in (Gross and Thoennessen, 2006; Demantke et al., 2011), the eigen-features of linearity A_λ, planarity P_λ, and sphericity S_λ are defined as $A_\lambda = (\lambda_1 - \lambda_2)/\lambda_1$, $P_\lambda = (\lambda_2 - \lambda_3)/\lambda_1$, and $S_\lambda = \lambda_3/\lambda_1$. The linearity feature can be used to detect line structures; the planarity feature has the ability to discriminate planar structures; and the sphericity feature allows the exhibition of 3D structures.

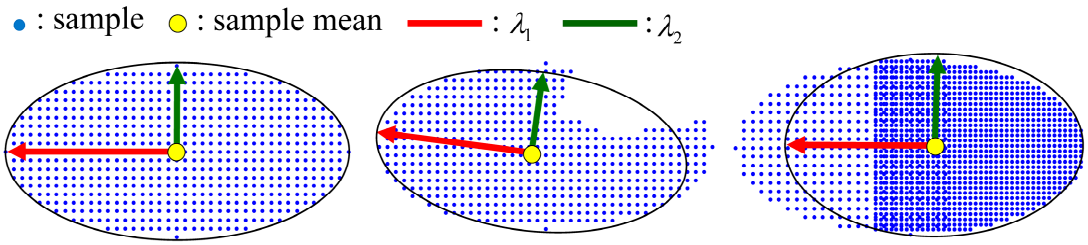

Fig. 1. Results of PCA on point datasets with uniform sampling (top), incomplete surface sampling (middle), and different sampling densities (bottom).

While PCA obtains mathematically optimal principal components, it should be noted that this method is sensitive to point distribution, especially for data with outliers, missing information, and different sampling densities (Stanimirova et al., 2007; Kriegel et al., 2008). Outliers are points that exhibit far different positions compared with the majority of the dataset. Therefore, the obtained principal components are unable to describe properly the majority of the data. Similarly, missing information and different sampling densities make PCA difficult to accurately describe the geometric shape of data. As shown in Fig. 1, the PCA on point sets with

missing information and different point densities generate inaccurate eigenvalues and eigen-features. A possible solution to this problem is to use the center of a neighborhood ball as the mean in PCA calculation. This approach is efficient and insensitive to point density. However, the smallest eigenvalue is generally larger than that of PCA using sample mean for non-planar data sets (Fig. 2). In addition, the smallest eigenvalue may be inaccurate near object edges because of missing information. Another solution is using ellipsoid fitting instead of PCA. This approach can solve the problem of missing information for point sets with ellipsoidal shapes; however, the approach suffers from difficulties caused by point distribution as shown in Fig. 2.

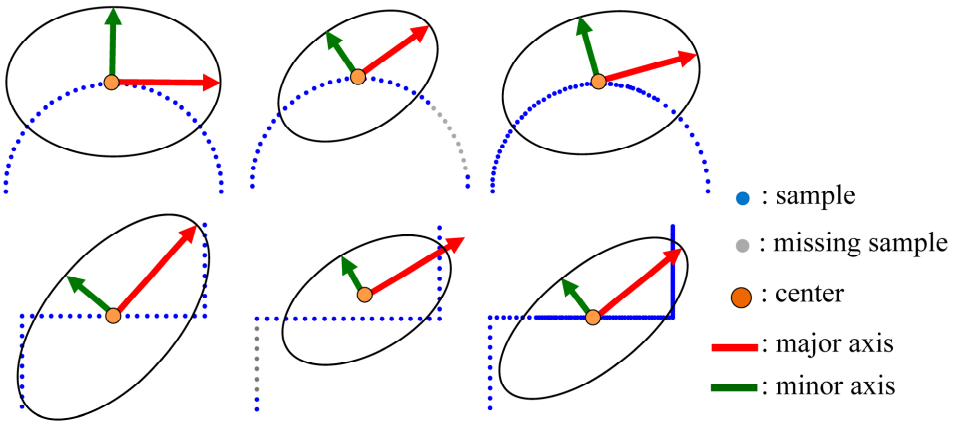

Fig. 2. Fitting results on point datasets with uniform sampling (left), incomplete surface sampling (middle), and different sampling densities (right). Top: PCA (using the center of a neighborhood ball as the mean); bottom: ellipsoid fitting.

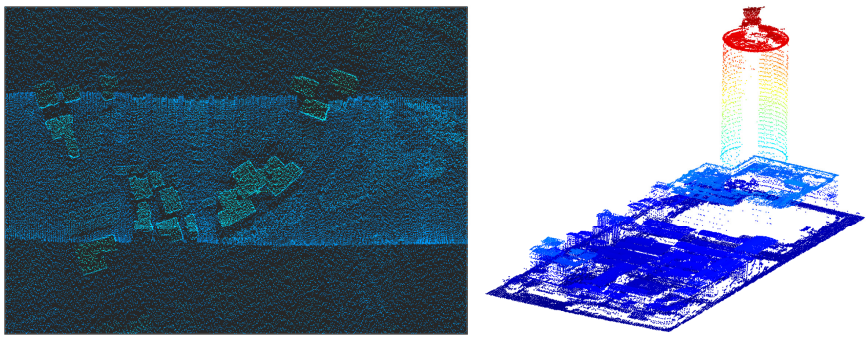

Fig. 3. Examples of airborne LiDAR point clouds with multiple-strip scanning (left) and incomplete surface sampling (right).

The recently proposed generalization of PCA, called weighted PCA, deals with outliers by assigning weights to the points (Kriegel et al., 2008). A distance-based weighting function is generally used in the weighted PCA to decrease the influence of outliers. The problem of data with missing information is solved by reconstructing the missing parts by utilizing an iterative data filling procedure with the aid of the existing parts (Stanimirova et al., 2007). However, these two approaches and strategies for missing information and outliers cannot be successfully applied to LiDAR point clouds. The distance-based weighting in weighted PCA can cope with outliers and noise. However, this weighting strategy cannot handle point clouds with incomplete surfaces well. In addition, this strategy also suffers difficulties coming from different point densities of point clouds caused by multiple-strip LiDAR scanning. As for the strategy of information reconstruction, it is difficult and unsuitable for LiDAR point clouds. The existing point cloud reconstruction methods (Kazhdan et al., 2006; Yeh et al., 2011) rely on accurate point normal for surface interpolation and are only suitable for dense and noiseless point clouds. The missing information is generally present in the airborne LiDAR data because of the self-occlusions of the terrain relief and ground objects, such as trees and buildings, and multiple-strip scanning is common in large-area LiDAR data acquisition (see Fig. 3). Therefore, a novel weighting function with geometric median is proposed to alleviate these problems in the eigen-feature analysis as described in Section 2.3.

2.3 Eigen-feature analysis of weighted covariance matrices

To reduce the surface sampling problems of LiDAR data in eigen-feature analysis, each point in a local point set, i.e., a point set in a neighborhood ball, is assigned a weight to represent its spatial contribution in weighted PCA, which is similar to the weighting strategy in computing the discrete surface gradient and

curvature (Luo et al., 2009; Merigot et al., 2011). We also use the geometric median, which represents the localized center of a point set, to calculate the covariance matrix instead of the sample mean which is sensitive to surface sampling. With the proposed weighting scheme and geometric median, the covariance matrix in (4) is reformulated as

$$\mathbf{C}(\mathbf{P}) = \frac{\sum_{p_i \in \mathbf{P}} w_{p_{gm}}(p_i)(p_i - p_{gm})^T(p_i - p_{gm})}{\sum_{p_i \in \mathbf{P}} w_{p_{gm}}(p_i)}, \tag{5}$$

where $w_{p_{gm}}(p_i)$ is the weight of point p_i with respect to the geometric median p_{gm}. The geometric median determination will be described at the end of this section.

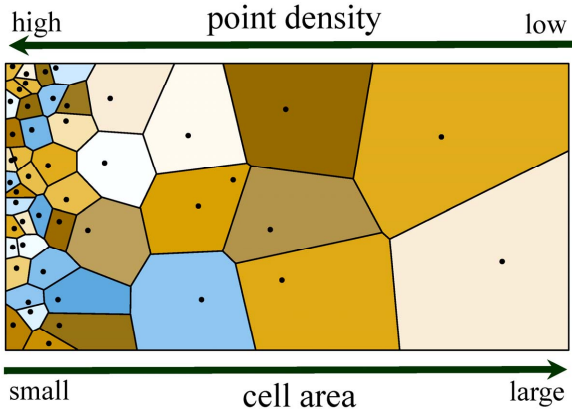

Fig. 4. Illustration of the relation between the areas of Voronoi cells and the point densities.

A possible solution in measuring the spatial contributions of the points is to compute the areas of surfaces occupied by the points with the aid of Voronoi diagram. The point occupied space can be defined as the Voronoi cell of that point. However, the construction of the Voronoi diagram for each point in a point cloud involves finding the best-fitting plane, projecting the points to that plane, and constructing a Voronoi diagram on that plane. These processes are computationally intensive because of the repetitive processes in a large amount of points. In addition, these processes may be non-robust for points in high-curvature regions because the

calculation of Voronoi cells depends on the approximation of point surface, i.e., the best-fitting plane. The Voronoi diagram can be robustly determined for polygonal models or dense and uniformly sampled point clouds, such as the data tested in (Luo et al., 2009; Merigot et al., 2011); however, it is difficult for LiDAR point clouds with inherent noise and incomplete shapes, i.e., missing information. In this paper, a simple and efficient approach is proposed to approximate the areas of Voronoi cells. Fig. 4 shows that the areas of Voronoi cells have a direct link to the point densities. A larger Voronoi cell area leads to a smaller point density. Therefore, the point density is inversely proportional to the area of the point occupied surface, and it is an efficient approach to measure the spatial contributions of points. In addition, the point distance, which is a commonly used weighting in tensor voting and PCA (You and Lin, 2011; Kriegel et al., 2008), is introduced in our weighting scheme. A point close to the geometric median is assigned a large weight and vice versa. The weighting function is defined by combining the factors of surface area and distance as

$$w_{p_{gm}}(p_i) = G_{p_{gm}}(p_i) \big/ \sigma(p_i)^{\beta}, \tag{6}$$

where $\sigma(p_i)$ represents the point density of point p_i, which is defined as the normalized number of points within a neighborhood ball centered at point p_i. The number of points is normalized by *n*, that is, the number of points in the given local point set. β is an unknown real number used to well fit the point densities to areas of occupied surfaces and to search for the optimal geometric median. $G_{p_{gm}}(p_i)$ is the Gaussian distance between point p_i and geometric median p_{gm}, which is defined as

$$G_{p_{gm}}(p_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|p_i - p_{gm}\|^2}{2\sigma^2}}, \tag{7}$$

where σ is the standard deviation of the Gaussian function. In the experiment, σ^2 is set to 0.5 based on our empirical observation. Note that compared with Euclidean and Hausdorff distances, the use of Gaussian distance can better resist outliers

because the points far from the geometric median are assigned smaller weights.

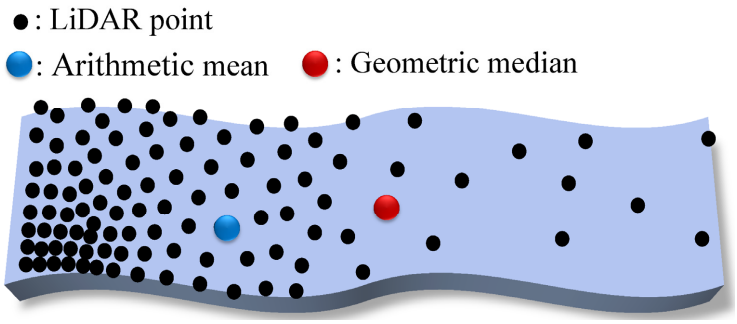

Fig. 5. Illustration of the difference between the mean (blue point) and the geometric median (red point) of a point set.

The geometric median is a center estimator of an arbitrary point set that minimizes the weighted sum of the Euclidean distances between the geometric median and all of the data points. Given that the outliers are removed, the geometric median has the property of sampling insensitivity compared with the sample mean used in (4), as exemplified in Fig. 5. The sample mean (blue point) is the arithmetic average of a set of values; thus, the sample mean is sensitive to LiDAR surface sampling. By contrast, the geometric median (red point) is the center of a local shape, which is potentially sampling insensitive. Therefore, the geometric median can aid in the description of the majority of a surface data. Formally, the geometric median is defined as

$$\arg\min_{p_{gm}} \sum_{i=1}^{n} w_{p_{gm}}(p_i) \times \left\| p_i - p_{gm} \right\|^2 . \tag{8}$$

Calculating the geometric median depends on the point weights, which is measured based on the obtained geometric median and the constant β in the weighting function. Therefore, an iterative process is adopted to obtain the geometric median, the point weights, and the constant β. This process is described in the pseudo-code. Initially, β is set to 1, and the geometric median p_{gm} is set to the mean of the input points, i.e., $p_{gm}^{c} = \sum_{i=1}^{n} p_i / n$. The point weights are then iteratively

computed with the current geometric median, and the geometric median is recalculated with the obtained point weights. In each iteration, the function *MovingDirection*() is performed to probe for the optimal β. This process is achieved by checking whether the increase and decrease of β can better fit the optimization function in (8). The offset *d* of β is returned from this function. The iteration is terminated if the difference between the current geometric median p_{pg}^{c} and the previous geometric median p_{pg}^{o} is less than a user-defined threshold δ. The obtained point weights and geometric median are used in calculating the covariance matrix of **P**, and the eigen-features are then obtained from the eigenvalues of the covariance matrix. Note that in the case of uniformly sampled point sets, the weights of points are the same because of the same point density. In this case, Eq (8) is simplified to

$$\arg\min_{p_{gm}} \sum_{i=1}^{n} k \times \left\| p_i - p_{gm} \right\|^2, \tag{9}$$

where *k* represents the point density. The minimization of (9) occurs when the geometric median is equal to the data mean, meaning that the proposed method is simplified to the standard PCA.

Input: P: input point set $\{p_1, \cdots, p_n\}$.

Output: p_{gm}: geometric median; $w_{p_{gm}}$: point weights.

Procedure *Weighting* () {

$\beta \leftarrow 1$; $p_{gm}^{c} \leftarrow \sum_{i=1}^{n} p_i / n$;

Compute the point weights by using the initial geometric median p_{gm}^{c};

repeat

$p_{gm}^{o} \leftarrow p_{gm}^{c}$;

d=*MovingDirection*(p_{gm}^{c});

//probe for the optimal β and return the offset of β, denoted by *d*.

$\beta \leftarrow \beta + d$;

Re-compute the point weights $w_{p_{gm}}$ using (6) with the new β;

$p_{gm}^{c} \leftarrow \sum_{i=1}^{n} w_{p_{gm}} p_i \Big/ n$;

until $\left\| p_{gm}^{o} - p_{pg}^{c} \right\|^2 < \delta$; // δ is a user-defined threshold.

Output p_{gm}^{c} and $w_{p_{gm}}$;

}

3. Experimental Results

3.1 Evaluation of proposed weighting scheme

Several simulated LiDAR data were generated and tested to evaluate the proposed method. The simulated data are generated by an airborne LiDAR simulator, which has two components: sensor and platform (Chen et al., 2014). The parameters for the sensor include the scanning field of view (FOV) and the laser pulse rate, i.e., scanning frequency. The platform component is used to simulate the aircraft condition, which includes the following parameters: trajectory, height, and velocity. In the experiments, the laser pulse rate was set to 80 kHz (laser pulse rate generally ranges from 25 kHz to 100 kHz); the FOV was set to 40 degrees; the flying height was set to 1000 m (flying height generally ranges from 500 m to 2000 m); and the flying speed was set to 450 km/hr.

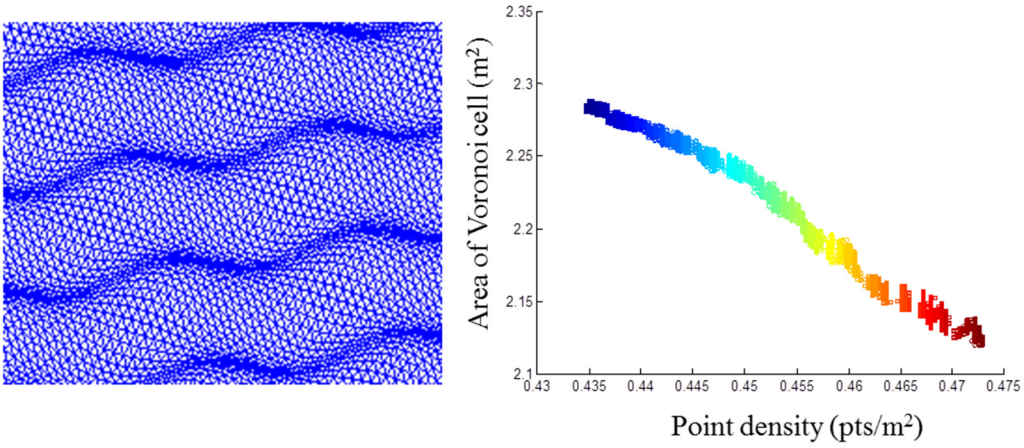

Fig. 6. Illustration of the relation between areas of Voronoi cells and point densities. The tested simulated data is shown in the left. The unit of point density is number of points per square meter, that is, pts/m^2.

Simulated data acquired from a surface were tested to check the relation between the areas of Voronoi cells, i.e., the areas of point occupied surfaces, and the point densities. A Voronoi diagram was generated for this simulated point set, and radius of neighborhood ball was set to 10 m. The areas of Voronoi cells and the point densities are calculated and plotted in Fig. 6. The experiment shows that the areas of Voronoi cells are nearly inversely proportional to the point densities and indicate that the areas of point occupied spaces can be approximated by point densities. Note that the areas of Voronoi cells are theoretically inversely proportional to the point density. However, the point density is defined as the number of points within a neighborhood ball, meaning that it is calculated in a discrete space. The approximated point density is the reason why the fitting line in Fig. 6 is slightly curved.

3.2 Analysis of proposed eigen-features

A LiDAR point cloud from overlapping strips was tested to evaluate the linearity and sphericity eigen-features generated by our method and PCA. The linearity feature can be used to detect line structures, whereas the sphericity feature allows for the exhibition of high-curvature points. These two features are useful geometric descriptors for point cloud classification, partition, and feature extraction. The comparisons of these two eigen-features generated by our method and PCA are shown in Figs. 7 and 8. The eigen-features are visualized by colors ranging from blue (the lowest value) to red (the highest value). Note that the color brightness in the overlapping and un-overlapping regions is different because of the different point densities. Fig. 7 reveals from the visualization of linearity features that the points lying around the boundaries of overlapping strips (see Area A) have larger errors compared with those in the un-overlapping regions. This difference in errors is caused by the eigen-feature analysis being sensitive to the sampling density. Therefore,

generating reliable eigenvalues and eigen-features through the PCA is difficult when the variation of sampling densities in the overlapping and un-overlapping regions is large. The linearity features of the points lying near the boundaries of building walls also exhibit large errors, as shown in the close-up views of Area B, because the eigen-feature analysis suffers from the missing information caused by the self-occlusions of buildings. These problems coming from the surface sampling of LiDAR can be alleviated through the weighted PCA with the proposed geometric median and weighting strategy, and thus, the classification accuracy can be improved by using the generated eigen-features.

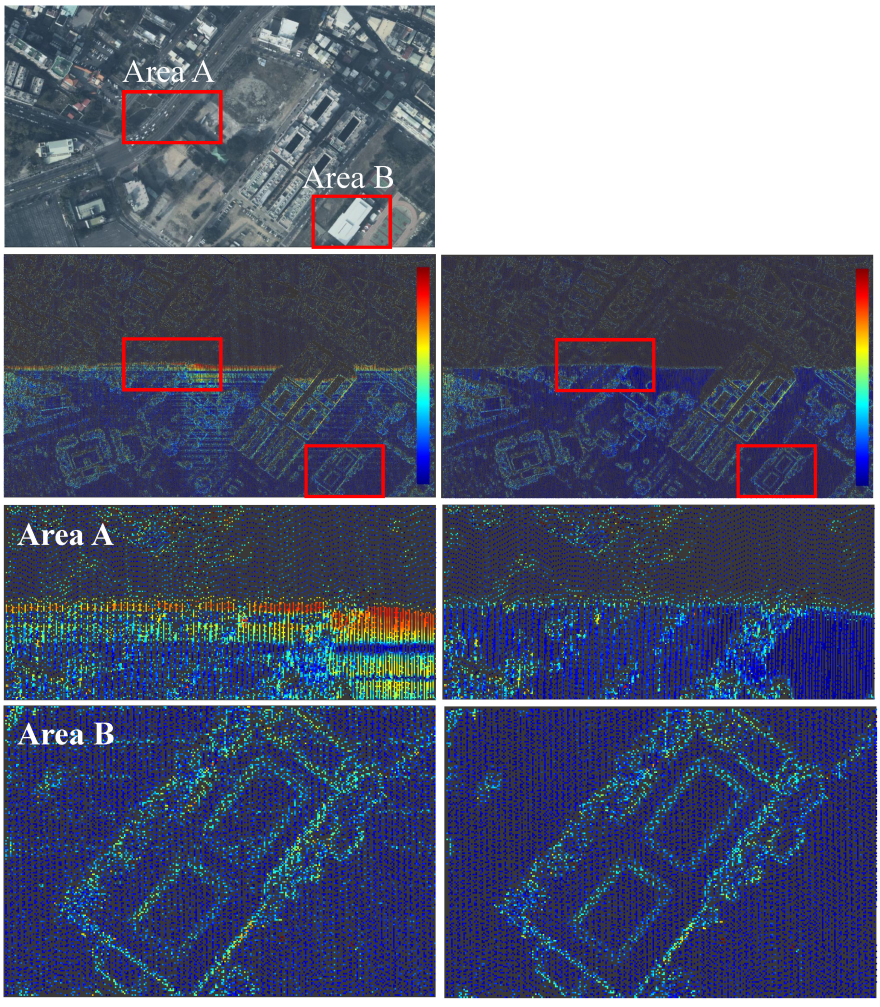

Fig. 7. Comparison of linearity feature. First row: orthoimage of the test data. Second row: linearity features of points visualized by colors ranging from blue (the lowest

value) to red (the highest value). Third and fourth rows: close-up views of Areas A and B. The linearity features generated by PCA and our method are shown in the left and right, respectively.

Similarly, in Fig. 8, the ridges of building roofs are not clear in the visualization of the sphericity features generated by the standard eigen-feature analysis because the PCA is sensitive to point distribution. By contrast, the roof ridges are clearly presented in the visualization of our sphericity features, meaning that better sphericity features are obtained. From the experiments shown in Figs. 7 and 8, we conclude that the surface sampling problems of LiDAR can be alleviated in the eigen-feature analysis by using the proposed weighting strategy and geometric median.

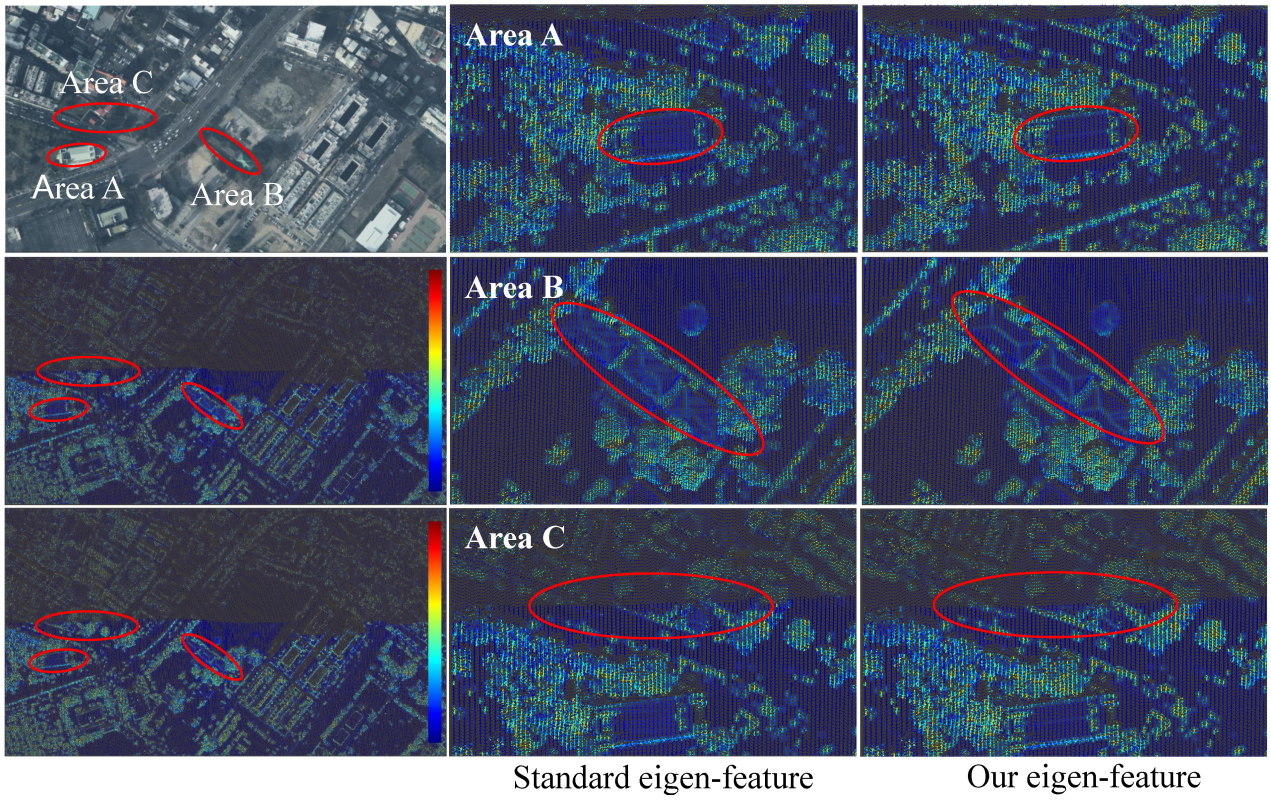

Fig. 8. Comparison of sphericity features. Left: orthoimage of the test data and the sphericity feature generated by PCA (middle) and our method (bottom). Middle: close-up views of the sphericity features generated by PCA. Right: our sphericity features. The point sphericity is visualized by colors.

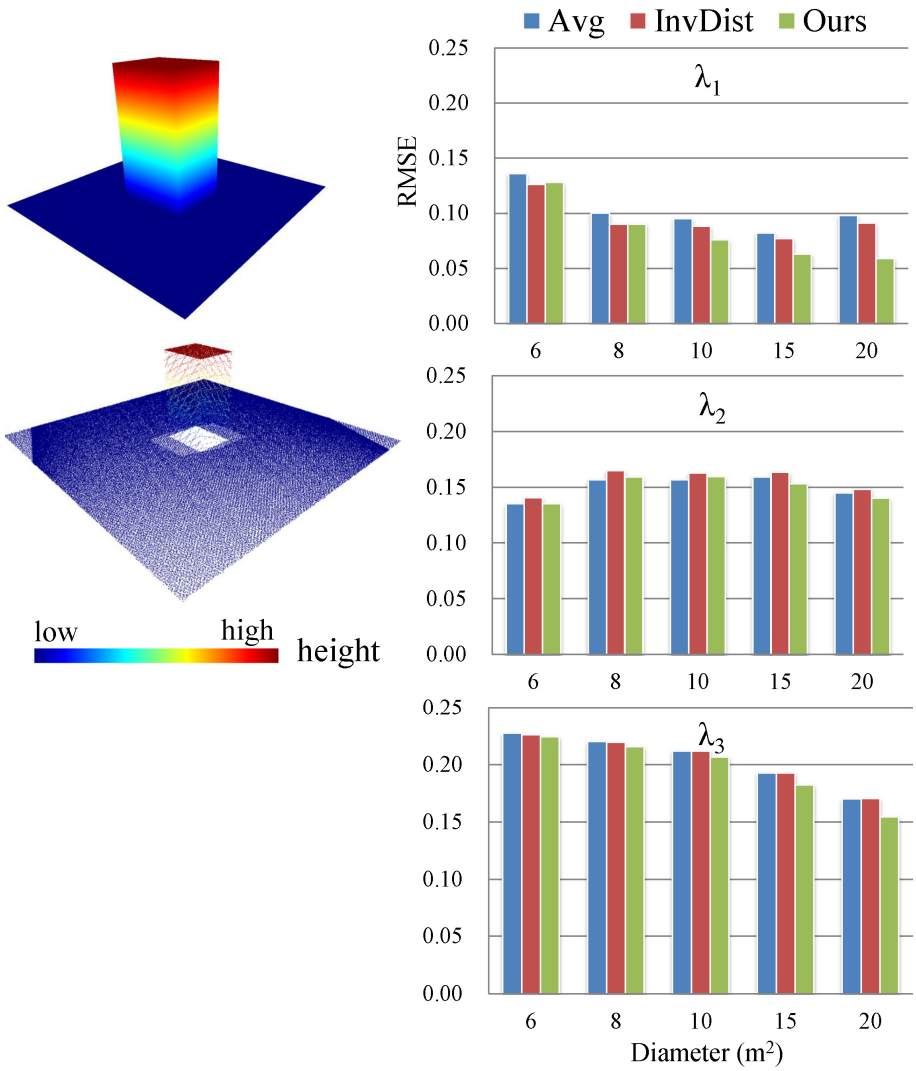

Fig. 9. Qualitative analysis of eigen-features. Left: uniformly sampled point cloud (top) and simulated data generated by the LiDAR simulator (bottom). The point heights are visualized by colors. Bottom: comparisons of the eigenvalues ($\lambda_1, \lambda_2, \lambda_3$) generated by using different weighting strategies, including average weighting, denoted by "Avg", inverse distance weighting, denoted by "InvDist", and the proposed weighting. The *x*-axis is the diameter of the neighborhood ball used in the covariance matrix computation. The *y*-axis is the RMSE of the eigenvalues calculated from the uniformly sampled and simulated point clouds.

A quantitative analysis of the computed eigen-features was also conducted by using different weighting strategies on a simulated LiDAR data. The experimental

procedure is as follows. A polygonal building model is selected, and a uniformly sampled point cloud and a simulated LiDAR point cloud of this model are generated. The eigenvalues of the standard covariance matrix of the uniformly sampled point cloud and the simulated point cloud are computed using different weighting strategies, including average weighting, inverse distance weighting, and the proposed weighting. The eigenvalues calculated from the uniformly and regularly sampled point cloud are regarded as the ground truth in the comparison. The eigenvalues computed from the simulated point cloud are evaluated and compared with the ground truths by using root-mean-square error (RMSE). The comparison is shown in Fig. 9. Compare our weighting function with the average and inverse distance weights, on average, the accuracy of λ_1 is improved by 18.5% and 11.8%, respectively; that of λ_2 is improved by 0.7% and 4.1%; and that of λ_3 is improved by 3.7% and 3.5%. The improvement of λ_1, i.e., the eigenvalue of the first principal component, is greater than those of λ_2 and λ_3. This experiment indicates that the proposed weighting function that considers the point distribution in the point weighting is better than the average and inverse distance weighting functions which suffer difficulties coming from incomplete surface sampling and different point densities of LiDAR point clouds. In addition, this result meets the visual comparisons shown in Figs. 7 and 8, where the improvement of the linearity eigen-feature (mainly dependent on λ_1) is larger than that of the sphericity eigen-feature (mainly dependent on λ_3). We conclude from the qualitative and quantitative analyses that the eigenvalues and eigen-features from the proposed weighted covariance matrix can better describe the geometric characteristics of LiDAR point clouds, compared with those from the standard covariance matrix.

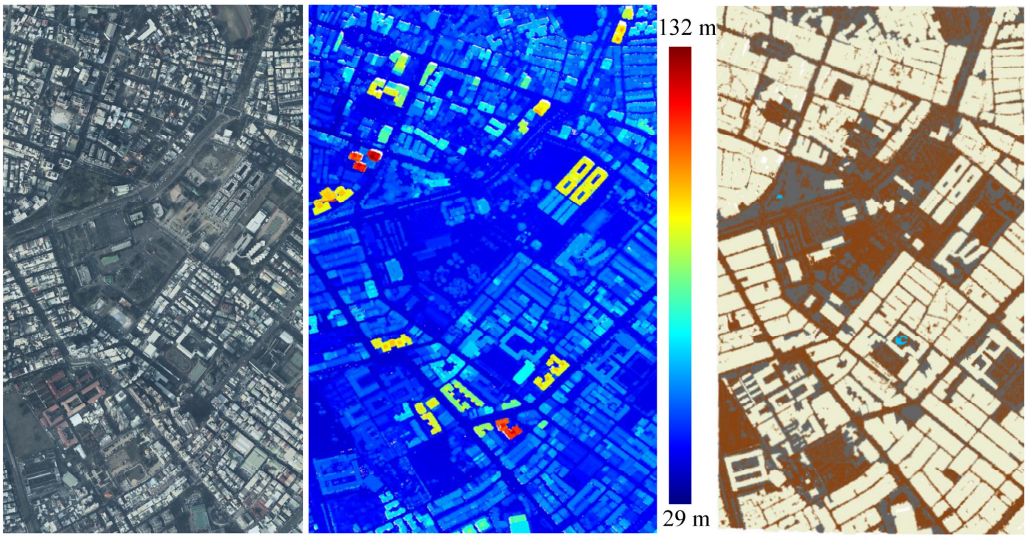

Fig. 10. Test dataset #1. Left: orthoimage. Middle: corresponding LiDAR data visualized by colors on the basis of point heights. Right: ground truth data. The ground and water points are visualized by dark brown and blue. The building and non-building points are displayed by light brown and gray.

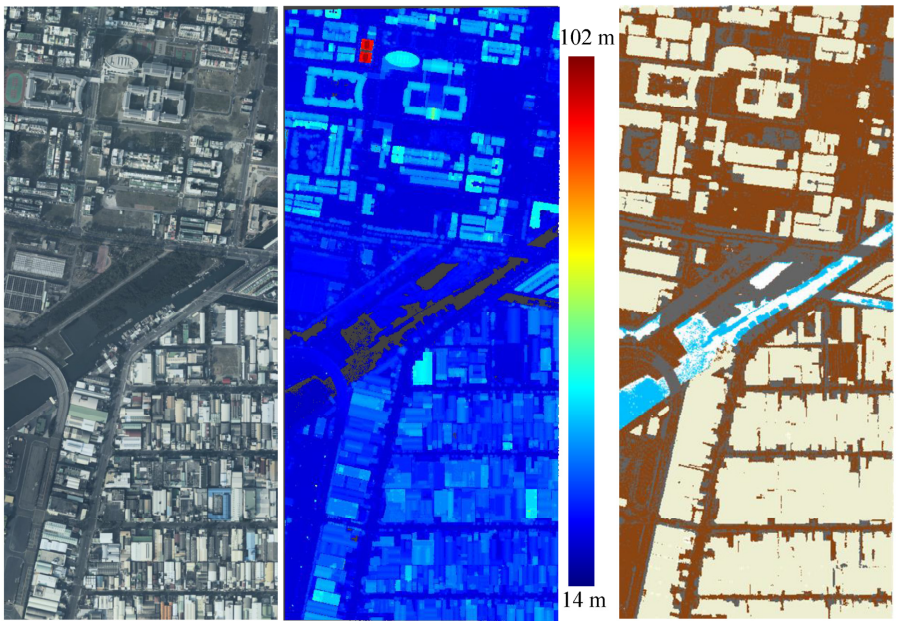

Fig. 11. Test dataset #2. Left: orthoimage. Middle: corresponding LiDAR data visualized by colors. Right: ground truth data.

3.3 Analysis of point cloud classification

The point cloud classification was evaluated on two airborne LiDAR datasets,

which were acquired by Optech ALTM 30/70 at Tainan, Taiwan in 2011. The first dataset shown in Fig. 10 has 2.1 million points with an average point density of 3.5 points/m^2. The second dataset shown in Fig. 11 has 3.3 million points with an average point density of 3.1 points/m^2. The outliers are removed manually, and the ground truths used in evaluating classification accuracy are semi-automatically generated by using the software TerraScan with the aid of orthoimages. The orthoimages are used only in generating ground truths and not in the classification. In the ground truths, point clouds are classified to four classes: ground, water, building, and non-building. The SVM classifier is used to evaluate the generated eigen-features. Only the points belonging to the building and non-building classes are tested. In the experiment, various point cloud features are used, including the eigen-based geometric features (containing linearity, planarity, and sphericity), the height-based features (including height difference, i.e., the height difference between the LiDAR point and the lowest point in the neighborhood cylindrical volume, and the height variance, which is formulated as the height variance of the points in the neighborhood cylindrical volume), and the echo feature, which is defined as the number of returns. For more details of the height-based and echo-based features, please refer to (Mallet et al., 2011). Note that the process of feature selection is not performed in classification; thus, the selected features have the same influence on the classification results.

The classification accuracy is computed based on the ground truths, and the results are shown in Table 1. The accuracy of our classification improves on average by 2.71% and 1.60% in the training and testing steps, respectively, compared with the classification using standard eigen-features. The confusion matrix of our classification shown in Table 2 indicates that the classification accuracy of non-buildings is better

than that of buildings. Note that our classifications have accuracies between 90% and 95%, which is slightly lower than those from the previous studies (Antonarakis et al., 2008; Brodu and Lague, 2012) with accuracies above 95%. This result is caused by the random selection of our training set and the skipping of the feature selection process. However, this study addresses on the geometric eigen-features, and the main purpose of these experiments is to prove that the classification accuracy can be improved using the generated eigen-features. To further check the improvement of the generated features, only the eigen-based features are used in the classification. The classification result shown in Table 2 indicates that the classification accuracies are improved by 4.05% and 2.21% in Datasets 1 and 2, respectively. In addition, the visual comparison in Fig. 12 shows that some incorrect classifications near the building boundaries are improved by using the generated eigen-features.

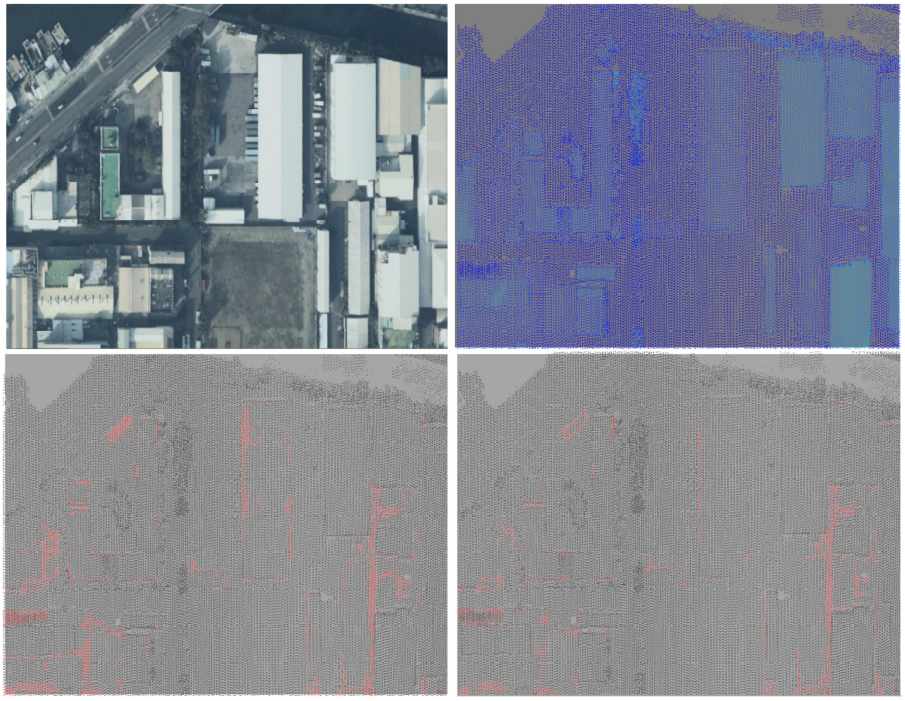

Fig. 12. Visual comparison of point cloud classifications using the standard eige-features (left) and the generated eigh-features (right). Top: orthoimage and cooresponding LiDAR data (from dataset #2); bottom: classification results. The

correct and incorrect classified points are displayed by gray and red, respectively.

Table 1. Comparison of classification accuracy. Eigen-based, height-based, and echo-based features are used in this experiment.

	PCA		Our approach	
	Dataset #1	Dataset #2	Dataset #1	Dataset #2
Training	91.3%	92.6%	93.5%	95.8%
Testing	90.8 %	88.2%	92.0%	90.2%

Table 2. Confusion matrix of the classification using the proposed method.

	Class	Building	Non-building
Dataset #1	Building	88.7%	11.3%
	Non-building	7.7%	92.3%
Dataset #2	Building	86.2%	13.8%
	Non-building	6.6%	93.4%

Table 3. Comparison of classification accuracy. Only the eigen-based features are used in this experiment.

	PCA	Our approach
Dataset #1	69.5%	73.6%
Dataset #2	68.1%	70.3%

4. Conclusions and Future Work

This paper addresses the issue of airborne LiDAR point cloud classification in urban areas. The main goal is to generate reliable eigen-features from a point cloud to improve classification accuracy. Our method is based on the weighted covariance matrix with the geometric median, which is insensitive to the non-uniform and incomplete sampling natures of airborne LiDAR scanning. In the eigen-feature analysis, each point in a local point set is assigned a weight to represent the spatial contribution of the point. Point density, which has the property of computational efficiency, is introduced in the weighting function to approximate the area of surface

occupied by a point. In addition, the geometric median, which is the geometric center of a point set, is used instead of the sample mean in calculating the covariance matrix. An iterative solver is proposed to obtain the geometric median and point weights. The experiment results on the simulated point cloud shows that the accuracy of the eigenvalues calculated from the proposed weighted covariance matrix improves by 0.7% to 18.5%, compared with that of the eigenvalues calculated from the standard covariance matrix. Besides, in the visual comparisons of the linearity and sphericity features, the generated eigen-features provided a better description of the geometric characteristics of the line structures and high-curvature regions of point clouds. The accuracy of the point cloud classifications using our eigen-features improves by 1.6% to 4.5%. Given the qualitative and quantitative analyses on airborne LiDAR point cloud and simulated data, we conclude that the eigen-features calculated from the proposed covariance matrix can improve classification accuracy compared with that calculated from the standard covariance matrix. In the near future, we aim to apply our method to point cloud registration and segmentation, which are also important topics in point cloud processing. We also plan to investigate the application of our method on the point clouds captured from laser terrestrial mobile mapping systems, discuss the effects of outliers in eigen-feature calculation, and determine the value of parameter σ in the weighting scheme according to the point neighborhood distribution.

References

Antonarakis A. S., Richards K. S., Brasington J., 2008. Object-based land cover classification using airborne lidar. Remote Sensing of Environment 112(6), 2988-2998.

Axelsson, P., 1999. Processing of laser scanner data — algorithms and applications.

ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3), 138-147.

Bork, E. W., Su, J. G., 2007. Integrating LIDAR data and multispectral imagery for enhanced classification of rangeland vegetation: a meta analysis. Remote Sensing of Environment 111(1), 11-24.

Brodu, N., Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. ISPRS Journal of Photogrammetry and Remote Sensing 68, pp. 121-134.

Charaniya, A., Manduchi, R., Lodha, S., 2004. Supervised parametric classification of aerial lidar data. IEEE Conference on Computer Vision and Pattern Recognition Workshop, pp. 30–37.

Chen, J.-Y., Lin, C.-H., Hsu, P.-C., Chen, C.-H., 2014. Point Cloud Encoding for 3D Building Model Retrieval, IEEE Transactions on Multimedia, to appear.

Choi, Y.-W., Jang, Y.-W., Lee H.-J., Cho G.-S., 2008. Three-dimensional LiDAR data classifying to extract road point in urban area, IEEE Geoscience and Remote Sensing Letters 5(4), 725-729.

Cobby, D.M., Mason, D. C., Horritt, M. S., Bates, P. D., 2003. Two-dimensional hydraulic flood modelling using a finite element mesh decomposed according to vegetation and topographic features derived from airborne scanning laser altimetry. Hydrological Processes 17(10), 1979-2000.

Cortes, C., Vapnik, V., 1995. Support-vector network. Machine Learning 20, 1–25.

Daszykowski, M., Kaczmarek, K., Heyden, Y. V., Walczak, B., 2007. Robust statistics in data analysis - a review: basic concept. Chemometrics and Intelligent Laboratory System. 85(2), 203–219.

Demantke, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point cloud. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII, pp. 97-102.

Flood, M., 2001. LIDAR activities and research priorities in the commercial sector. International Archives of Photogrammetry and Remote Sensing 34(3), 3-7.

Gressin, A., Mallet, C., Demantké, J., David, N., 2013, Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge, ISPRS Journal of Photogrammetry and Remote Sensing 79, pp. 240-251.

Gross, H., Thoennessen, U., 2006. Extraction of lines from laser point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Calgary, Canada, pp. 86-91.

Gross, H., Jutzi, B., Thoennessen, U., 2007. Segmentation of tree regions using data of a full-waveform laser. In: ISPRS Conference Photogrammetric Image Analysis (PIA), vol. 36, IAPRS, Munich, Germany.

Gumhold, S., Wang, X. and MacLeod, R., 2001. Feature extraction from point clouds. In Proceedings of the 10th International Meshing Roundtable, pp. 293-305, 2001.

Haala, N., Brenner, C., 1999. Extraction of buildings and trees in urban environments. ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3), 130-137.

Hui, L., Di, L., Xianfeng, H., L., Deren, 2008. Laser intensity used in classification of lidar point cloud Data. IEEE International Conference on Geoscience and Remote Sensing, 7-11 July, Boston, pp. 1140-1143.

Jolliffe, I.T., 1986. Principal component analysis. Springer Verlag.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. Proceedings of the fourth Eurographics symposium on Geometry processing, pp.

61-70.

Kriegel, H. P.; Kröger, P.; Schubert, E.; Zimek, A., 2008. A general framework for increasing the robustness of PCA-based correlation clustering algorithms. Proceedings of the 20th international conference on Scientific and Statistical Database Management, pp. 418-425.

Lodha, S., Kreps, E., Helmbold, D., Fitzpatrick, D., 2006. Aerial LiDAR Data Classification Using Support Vector Machines (SVM). Proceedings of the International Symposium on 3D Data Processing Visualization, and Transmission, IEEE, Chapel Hill, NC, pp. 567-574.

Lohmann, P., Koch, A., Schaeffer, M., 2000. Approaches to the filtering of laser scanner data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 540-547.

Luo, C., Safa, I., Wang, Y., 2009. Approximating gradients for meshes and point clouds via diffusion metric. SGP '09 Proceedings of the Symposium on Geometry Processing, Aire-la-Ville, Switzerland, pp. 1497-1508.

Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform lidar data for urban area classification. ISPRS Journal of Photogrammetry and Remote Sensing 66 (6), S71-S84.

Merigot, Q., Ovsjanikov, M., Guibas, L., 2011. Voronoi-based curvature and feature estimation from point clouds. IEEE Transactions on Visualization and Computer Graphics 17 (6), 743-756.

Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 66 (3), 247-259.

Pearson, R. K., 2005. Mining Imperfect Data: Dealing with Contamination and Incomplete Records, SIAM.

Rutzinger, M., Höfle, B., Hollaus, M., Pfeifer, N., 2008. Object-Based Point Cloud Analysis of Full-Waveform Airborne Laser Scanning Data for Urban Vegetation Classification. Sensors 8 (8), 4505-4528.

Samadzadegan, F., Bigdeli, B., Ramzi., P., 2010. Classification of LiDAR data based on multi-class SVM. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 1-6.

Sampath, A., Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Transactions on Geoscience and Remote Sensing 48 (3), pp. 1554-1567.

Secord, J., Zakhor, A., 2007. Tree detection in urban regions using aerial lidar and image data. IEEE Geoscience and Remote Sensing Letters 4(2), 196-200.

Stanimirova, I., Daszykowski, M., Walczak, B., 2007. Dealing with missing values and outliers in principal component analysis. Talanta 72 (1), 172-178.

Stassopoulou A., Caelli T., 2000. Building detection using bayesian networks, International Journal of Pattern Recognition and Artificial Intelligence 83, 715-733.

Vosselman, G., 2000. Slope based filtering of laser altimetry data. International Archives of Photogrammetry and Remote Sensing 33(B3/2), 935-942.

Wagner, W., Hollaus, M., Briese, C., Ducic, V., 2008. 3D vegetation mapping using small-footprint full-waveform airborne laser scanners. International Journal of Remote Sensing 29(5), 1433-1452.

Yeh, I-C., Lin, C.-H., Sorkine, O., Lee, T.-Y., 2011. Template-based 3D model fitting

using dualdomain relaxation. IEEE Trans. on Visualization and Computer Graphics 17, 1178-1190.

You, R.-J., Lin, B.-C., 2011. Building feature extraction from airborne lidar data based on tensor voting algorithm. Photogrammetric Engineering and Remote Sensing 77, 1221-1232.