Using Gradients and Tensor Voting in 3D Local Geometric Descriptors for Feature Detection in Airborne LiDAR Point Clouds in Urban Regions

Jaya Sreevalsan-Nair and Akshay Jindal

Abstract

Structural or geometric classification of three-dimensional (3D) point clouds of urban regions from airborne LiDAR enables feature (object-based) classification, and 3D reconstruction. Here, we consider positive semidefinite symmetric second-order tensors as local geometric descriptors (LGDs), which gives structural classification. We compute LGDs using local neighborhood, and their eigenvalue-based features are conventionally used for object-based classification and 3D reconstruction. We combine derivative based 2D gradient energy tensor, and anisotropically diffused 3D voting tensor, using a multi-scale approach, to compute a LGD. We represent LGDs as second-order tensors, and compare the relevant eigenvalue-based features and saliency maps obtained from them. We visually compare the outcomes of our LGD with conventionally used covariance matrix.

Index Terms

Structure tensor, Tensor voting, Eigenvalue-based features, 3D reconstruction

I. Introduction

Conventionally used methods such as covariance analysis of point clouds capture its geometry efficiently and approximately. Hence, they have been used widely in object-based classification and 3D reconstruction of airborne LiDAR (Light Detection and Ranging) datasets. However, we have found that covariance analysis does not detect certain features in urban regions, such as sharp features in gabled roofs, and unoriented regions in foliage, which are required for reconstruction. We hypothesize that detecting sharp features and unoriented regions as strong line-type and (critical) point-type features, respectively, can reduce the eigen-entropy of the structural classification. Clear separability of geometric classes can improve the 3D reconstruction and alleviate problems of building-tree (class) disambiguations [7]. Local geometric descriptors (LGDs) define the local neighborhood of each point, and the eigen-analysis of LGDs is used for geometric and object-based classifications of LiDAR point clouds and 3D reconstruction [7], [3], [8]. Our goal is to improve point classification using new LGDs.

II. Related Work

Structural classification indicates membership of each point to geometric classes, based on the shape of its local neighborhood. The classes correspond to the point belonging to line-, surface- or point- type features, based on its local neighborhood. The classification is determined by the relativeness of the eigenvalues of the LGD ($\lambda_0 \geq \lambda_1 \geq \lambda_2$) [7], [3], [8]. For a point x in point cloud $\mathcal{P}$ and given radius r, we take the spherical neighborhood as $N(x) = \{x, y \in \mathcal{P} : \|y - x\| < r\}$. $\lambda_0 \gg \lambda_1 \approx \lambda_2$ imply curve-shaped, $\lambda_0 \approx \lambda_1 \gg \lambda_2$ disc-shaped, and $\lambda_0 \approx \lambda_1 \approx \lambda_2$ spherical neighborhoods. These definitions correspond to the point belonging to a line-, surface, and point-type (critical point) feature, respectively. While 3D tensor formulation of LGDs gives probabilistic classification [3], [8] and is a conventionally used method for classification, the 2D tensor formulation enables detection of line- and point-type features. However, in the latter, there are several points which have zero eigenvalues, due to which they need not be classified as a feature point.

Eigenvalue-based features from the LGDs, such as aniso-tropy $\mathcal{A}$, have been used for structural and object-based classification in airborne LiDAR point clouds [7]. In addition to these feature values we consider saliency values [8]. Saliency values for each structural class give us the likelihood of a point belonging to that class. Saliency values for line-, surface-, and point-type features are given as c_l, c_s, and c_p, respectively. Here, we use anisotropy and saliency values (defined in Table I) to evaluate the point classification, specifically in the line-type features.

A. 3D Tensor Formulation

Conventionally, covariance matrix is used as LGD for 3D point clouds in LiDAR community [7]. The covariance matrix is given by $T_{3DCM} = \sum_{y \in N(x)} (y - \bar{x})(y - \bar{x})^T$, where $\bar{x}$ is the centroid of the local neighborhood $N(x)$. However, anisotropically

This document is preprint, as on June 05, 2017. The authors are with Graphics-Visualization-Computing Lab, International Institute of Information Technology Bangalore (IIIT-B), Bangalore 560100, India. email: jnair@iiitb.ac.in ; akshay.jindal@iiitb.org . This work has been partially supported by NRDMS programme of the Department of Science and Technology, Government of India. This work has been built on the thesis work of Beena Kumari at IIIT-B.

TABLE I
DEFINITIONS OF EIGENVALUE-BASED FEATURE VALUES (ANISOTROPY $\mathcal{A}$, AND SALIENCY VALUES $\{c_l, c_s, c_p\}$ FOR LINE-, SURFACE- AND POINT-TYPE FEATURES) FOR 3D AND 2D SECOND-ORDER POSITIVE SEMIDEFINITE SYMMETRIC TENSORS, WHEN USED AS LGDS.

Tensor→	**3D**	**2D**
Eigenvalues	$\lambda_0 \geq \lambda_1 \geq \lambda_2$	$\lambda_0 \geq \lambda_1$
Sum $\mathbf{S}$	$(\lambda_0 + \lambda_1 + \lambda_2)$	$(\lambda_0 + \lambda_1)$
$\mathcal{A}$	$(\lambda_1 - \lambda_2)/\lambda_0$	$(\lambda_0 - \lambda_1)/\lambda_0$
c_l	$(\lambda_0 - \lambda_1)/\mathbf{S}$	$(\lambda_0 - \lambda_1)/\mathbf{S}$
c_s	$2 * (\lambda_1 - \lambda_2)/\mathbf{S}$	0
c_p	$3 * \lambda_2/\mathbf{S}$	$2 * \lambda_1/\mathbf{S}$

diffused 3D voting tensor [8] has been shown to classify points belonging to line-type features more accurately[1]. The voting tensor is computed as an unoriented ball tensor at point x, $V(x) = \sum_{y \in N(x)} \mu_y . \left(I_d - \frac{t(y)t(y)^T}{t(y)^T t(y)} \right)$, where for $y \in N(x)$, $t(y) = (y - x)$ and $\mu_y = exp\big(- \frac{\|t(y)\|_2^2}{\sigma^2} \big)$ and I_d is identity matrix of size of the dimensionality, i.e., 3. We use the anisotropically diffused voting tensor, T_{3DVT}, computed from eigenvalues λ_i, corresponding eigenvector e_i of $V(x)$, for $i = 0, 1, 2$, and diffusion parameter $\delta = 0.16$ [9], as follows:
$T_{3DVT} = \sum_{i=0}^{2} exp(-\lambda_i/\delta) * e_i e_i^T$.

B. 2D Tensor Formulation

Structure tensor, in image processing and computer vision, uses the gradient of function value at a point to detect line-type features, such as edges and ridges [4]. The performance of the structure tensor, given as $T_S = \nabla f . \nabla f^T$, for function f, has been improved by using higher order derivatives, in the gradient energy tensor (GET) [2]. GET uses the two-dimensional gradient of a function f, given by $\nabla f = (\frac{\mathrm{d}f}{\mathrm{d}x}, \frac{\mathrm{d}f}{\mathrm{d}y}) = (f_x, f_y)$, the gradient of the Laplacian of f, given by $\mathcal{T} f = \nabla(\nabla^T \nabla f)$, and the Hessian of the function, given by $\mathcal{H} f = \nabla \nabla^T f$. The GET is defined as $T_{GET}(f) = (\mathcal{H}f)(\mathcal{H}f)^T - 0.5\big((\nabla f)(\mathcal{T}f)^T + (\mathcal{T}f)(\nabla f)^T\big)$. The GET is an improvement of the energy tensor [1] and the structure tensor. It is used for improving detection of points of interest (POI) in an image. We consider the point-type features in 3D point clouds to be analogous to POI in 2D images. Hence, we propose the use of the GET in improving the LGDs of point clouds. Structure tensor is a second-order positive semidefinite symmetric tensor. The GET is a second-order symmetric tensor, but not positive semidefinite at all points. However, it has been found that the GET is positive semidefinite at the POIs (such as corner points), and hence the negative eigenvalues of the tensor can be truncated to zero [5]. Thus, we effectively reduce the GET to be a positive semidefinite second-order symmetric tensor. Here, we propose a method of aggregating the LGD and GET to improve structural classification.

C. Multi-scale Approach

In our previous work [8], we have found that multi-scale approach performs better than single-scale ones, which could be either adaptive or optimal. Hence, we compute eigenvalue-based features using tensors computed at multiple scales. The average of the saliency values computed across multiple scales give the multi-scale values, since probabilities of exclusive events of multiple scales can be added [8].

However, for computing anisotropy, we require a multi-scale aggregated tensor. Hence, we compute a multi-scale aggregated tensor, T_{MS}, as a weighted sum of tensor products of unit vectors of eigenvectors, where the weights are normalized eigenvalues. For dimensionality of the tensor, d, N_r scales, given that the eigenvalues and corresponding eigenvectors of the LGD at scale, r, are λ_i and e_i for $0 \leq i \leq d-1$, we get: $T_{MS} = \sum_{r=1}^{N_r} \frac{1}{\sum_{i=0}^{d-1} \lambda_i(r)} \cdot \sum_{i=0}^{d-1} \lambda_i(r) \hat{e_i(r)} \hat{e_i(r)}^T$

III. OUR APPROACH

Our objective is to enhance line-type and point-type feature detection, so as to improve object-based classification or 3D reconstruction downstream. Our goal is to improve outcomes of the eigenvalue-based feature, namely anisotropy, as well as saliency maps, namely $\{c_l, c_s, c_p\}$ in the LGDs. In our proposed four-step method for improving probabilistic point classification in airborne 3D LiDAR, we compute all the LGDs as aggregate multi-scale tensor.

Use of 2D Tensors: The conventionally used LGD is three-dimensional, which uses covariance information (in T_{3DCM}) or perceptual organization based on differences (in T_{3DVT}) [8]. However, the use of higher order derivatives of the height map in the LGDs can improve the structural classification outcomes, as has been done in image data [1], [2], [5]. We consider the

[1]We disambiguate between "tensor voting" and "voting tensor," where the former is the approach used in computer vision for perceptual grouping based on Gestalt's principles and the latter is the positive semidefinite symmetric second-order tensor computed during tensor voting.

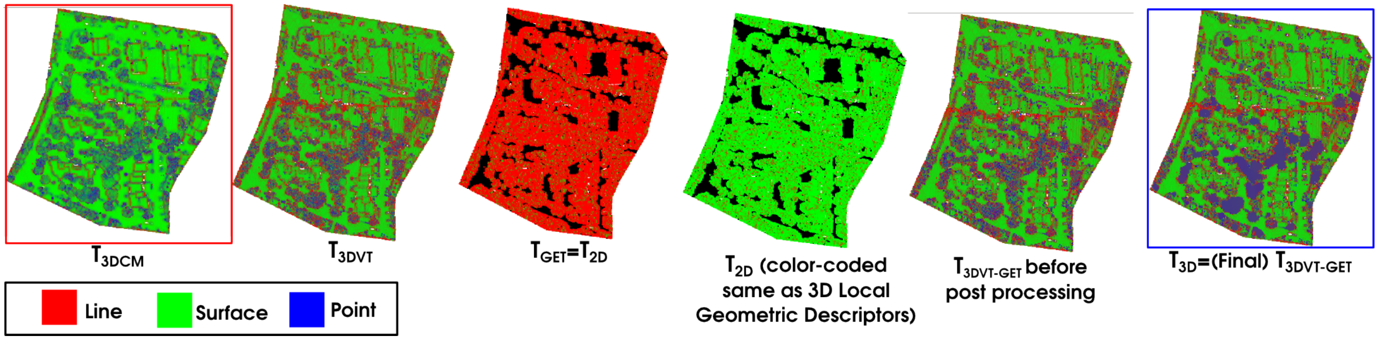

Fig. 1. In Area-2 of Vaihingen dataset (266,675 points), T_{3D} improves line- and point-type feature detection, thus, distinguishing gabled roofs and foliage, respectively. T_{3DVT} detects (77,679, 173,991, 15,005) points as (line, surface, point)-type features; $T_{3DVT-GET}$ detects (73,332, 68,175, 25,168), and T_{3D} detects (64,849, 158,937, 43,889), respectively.

airborne LiDAR point cloud to be a digital elevation model (DEM), in which case the altitude $z = f(x, y)$, is a function of latitude (y) and longitude (x) values of each point. Using this representation, we compute the 2D structure tensor, similar to that of an image, using spatial derivatives. However, different from a structured data structure in an image, the (x, y) domain is unstructured in the case of airborne LiDAR point clouds. Hence, we use the derivative computation used for unstructured grid [6] for finding gradient of z, given by ∇z. Structure tensor is computed as $\nabla z \nabla z^T$. We use a variant of the structure tensor, which is the gradient energy tensor [2], [5] of z, given by $T_{2D} = T_{GET}(z)$. We use T_{2D} to improve the eigenvalue-based features given by T_{3DVT}. In Table I, we tabulate the definitions of the eigenvalue-based features for LGDs as 2D tensors.

Probabilistic Point Classification: We use the point classification using eigenvalues [3], [8]. When using 3D tensors, such as T_{3DCM}, $V(x)$, or T_{3DVT}, the points with high c_l, c_s, c_p values have high likelihood of belonging to line-, surface-, and point-type features, respectively. By using anisotropically diffused voting tensor T_{3DVT} [8], we enhance the line-type features, in comparison to the conventionally used covariance tensor (Figure 1)[2]. For 2D tensors, we detect line- and point-type features using the c_l and c_s values of T_{2D}, as given in Table I. However, we observe that the points identified with high c_l values of T_{2D} (Figure 1) are the POIs [1], and hence belong to point-type features. Hence, we use these points to correct the point-type features in T_{3DVT}. We thus substitute $\{c_l, c_s, c_p\}$ of T_{3DVT} of such points with $\{0, c_p, c_l\}$ of the T_{2D}, and modify the saliency results of these points to correspond to the same as T_{2D}(Figure 1). We refer to the resultant LGD as $T_{3DVT-GET}$. Thus, we improve on the detection of point-type features by using $T_{3DVT-GET}$, specifically targeting the POIs (Figure 1).

Post-processing: We convert the probabilistic point classification to a deterministic one by using the maximum value of the saliency map, so that the structural classification can be used for object-based classification or 3D reconstruction. Thus, points with maximum of the saliency values $\{c_l, c_s, c_p\}$ being c_l becomes a line-type; with that of c_s, a surface-type; and with that of c_p, a point-type feature. We perform a post-processing to uniformly describe foliage as (critical) point features. Hence, for each point, if there is a majority of point-type features in its local neighborhood, we correct all points in the local neighborhood to point-type feature. Our post-processing step is akin to a voting method. However, we have not studied the influence of different traversals through the point cloud in converting the point-type features, such as, initiations using seeds, using ranking order, etc. We use the resultant 3D tensors as LGDs, referring to them as T_{3D}.

A. *Our Proposed Four-Step Method*

Our proposed method entails the following steps in sequence: 1. Multi-scale aggregate of the anisotropically diffused 3D voting tensor, T_{3DVT}, is computed at each point.
2. Multiscale aggregate of the gradient energy tensor T_{2D}, is computed at each point.
3. We use the c_l value of T_{2D} to identify points of interest, and correct the T_{3DVT} using its corresponding T_{2D} value. Thus, we get $T_{3DVT-GET}$ at each point.
4. At each point, we check if its local neighborhood has a majority of point-type features. If yes, then we change all points in the local neighborhood to be point-type features. Thus, we get T_{3D} at each point.

IV. Experiments & Results

We have conducted our experiments on Area-2 and Area-1 of the benchmark Vaihingen dataset[3]. Visualizations of the eigenvalue-based features, A and saliency values of the LGDs, using 5 different scales, are shown in Figure 2.

Area-1 belongs to the "inner-city" of Vaihingen, characterized by buildings with complex shapes and sparse vegetation. Area-2 is the "high-riser", with several buildings surrounded by trees. Our proposed LGD detects line-type features, specifically gabled

[2]Red boxes highlight conventionally used T_{3DCM} and blue boxes highlight our proposed T_{3D}.

[3]http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html

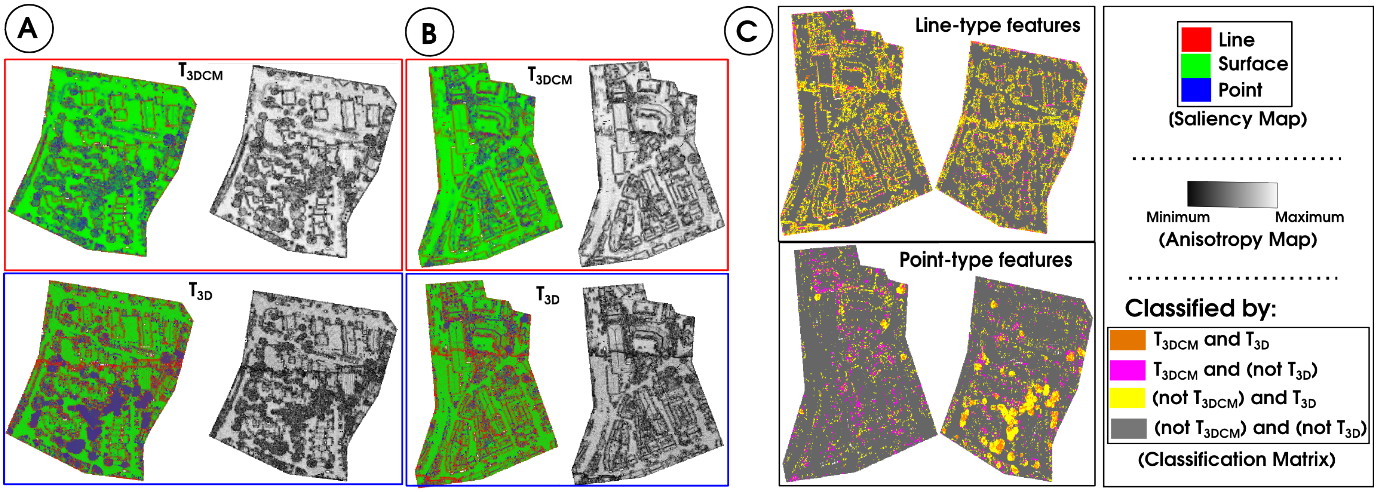

Fig. 2. Experimental results: [A] Comparison of T_{3DCM} and T_{3D} show that foliage is better visible in the case of latter, for Area-2 of Vaihingen dataset. [B] Comparison of T_{3DCM} and T_{3D} show that gabled roofs are better visible in the case of latter, for Area-1 of Vaihingen dataset (179,997 points). [C] Classification matrix shows which local geometric descriptor (T_{3DCM} or T_{3D}) has identified the points in point cloud as (left) line-type and (right) point-type features. Points identified exclusively by T_{3D} (yellow) show building outlines (left) and foliage (right).

roofs, in Area-1 and point-type features, specifically foliage, in Area-2 more accurately and emphatically than conventionally used T_{3DCM} (Figure 2). These results are attributed to the use of higher order derivatives in GET. The classification matrix for a specific class determines between two LGDs, the subset of points which were classified as belonging to that class by both the LGDs, or one of the two exclusively, or not by both. The matrix results (Figure 2 (C)) show that T_{3D} detects more line-type features exclusively than what T_{3DCM} does exclusively. For the point-type features, while T_{3DCM} finds more points in the class exclusively, T_{3D} finds clusters of point-type features to indicate foliage. The latter is as per design.

V. Conclusions

We have shown how combining the use of tensor voting and gradient energy tensor enables in better structural classification of airborne 3D point clouds. We have shown that the use of our proposed descriptor also improves the eigenvalue-based features, such as anisotropy, which is expected to improve 3D reconstruction and object-based classification.

References

[1] Michael Felsberg and Gösta Granlund. Poi detection using channel clustering and the 2d energy tensor. In *Joint Pattern Recognition Symposium*, pages 103–110. Springer, 2004.

[2] Michael Felsberg and Ullrich Köthe. Get: The connection between monogenic scale-space and gaussian derivatives. In *Intl, Conf. on Scale-Space Theories in Comp. Vision*, pages 192–203. Springer, 2005.

[3] Patric Keller, Oliver Kreylos, Marek Vanco, Martin Hering-Bertram, Eric S Cowgill, Louise H Kellogg, Bernd Hamann, and Hans Hagen. Extracting and visualizing structural features in environmental point cloud LiDaR data sets. In *Topological Methods in Data Analysis and Visualization*, pages 179–192. Springer, 2011.

[4] Hans Knutsson, Carl-Fredrik Westin, and Mats Andersson. Representing local structure using tensors ii. In *Image analysis*, pages 545–556. Springer, 2011.

[5] Ullrich Köthe and Michael Felsberg. Riesz-transforms versus derivatives: On the relationship between the boundary tensor and the energy tensor. In *Intl. Conf. on Scale-Space Theories in Comp. Vision*, pages 179–191. Springer, 2005.

[6] Akshay Narayan, Jaya Sreevalsan-Nair, Kelly Gaither, and Bernd Hamann. Isosurface Extraction from Hybrid Unstructured Grids Containing Pentahedral Elements. In *Proceedings of 4th Intl. Conf. on Information Vis. Theory and Applns. (IVAPP) 2012*, pages 660–669. INSTICC, 2012.

[7] Joachim Niemeyer, Franz Rottensteiner, and Uwe Soergel. Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 87:152–165, 2014.

[8] Jaya Sreevalsan-Nair and Beena Kumari. Local Geometric Descriptors for Multi-Scale Probabilistic Point Classification of Airborne LiDAR Point Clouds. *(to appear) in Modeling, Processing, and Visualization of Anisotropy*, 2017. preprint at http://www.iiitb.ac.in/GVCL/pubs.html.

[9] Shengfa Wang, Tingbo Hou, Shuai Li, Zhixun Su, and Hong Qin. Anisotropic Elliptic PDEs for Feature Classification. *Visualization and Computer Graphics, IEEE Transactions on*, 19(10):1606–1618, 2013.