Local Geometric Descriptors for Multi-Scale Probabilistic Point Classification of Airborne LiDAR Point Clouds

Jaya Sreevalsan-Nair and Beena Kumari

Abstract

Point classification is necessary for detection and extraction of geometric feature (folds, creases, junctions, surfaces), and subsequent 3D reconstruction of point-sampled geometry of topographic data captured using airborne LiDAR technology. Geometry-based point classification (line-, surface-, point-type features) is determined using shape of the local neighborhood, given by the local geometric descriptor (LGD) at every point in the point cloud. Covariance matrix of local neighborhoods is the conventionally used LGD in the LiDAR community. However, it is known that covariance analysis has drawbacks in detection of sharp features, which are a subset of the line-type features. Here, we compare the performance of new variants of existing LGDs, such as weighted covariance matrix, and that based on tensor voting concept, in geometric classification with that of covariance matrix. We propose a multi-scale probabilistic saliency map based on eigenvalues of the LGDs for computing the classification. Usually the state-of-the-art performance analyses of LGDs in the classification outcomes are done downstream after feature extraction. We propose that the comparisons may be done upstream at the classification stage itself, which can be achieved by expressing these LGDs as positive semidefinite second-order tensors. We perform qualitative comparisons of the tensor fields based on shape and orientation of the tensors, and the classification outcomes using visualizations. We visualize LGDs using superquadric tensor glyphs and point rendering, using our proposed saliency map as colormap. Our detailed comparative analysis shows that the new variant of LGDs based on tensor voting classify line-type features, especially sharp features, better than covariance-based LGDs. Our proposed LGD based on tensor voting performs better than the covariance matrix, for our goal of detecting sharp features, e.g. gabled roofs in buildings.

I. Introduction

Geometric feature detection is a key operation in the processing of three-dimensional (3D) point clouds, which includes surface reconstruction, surface matching, shape detection, registration of point clouds, and finding deformations in time-varying point sampled geometry. The definition of feature is however application-specific, which makes the process of feature extraction subjective. Features are generally defined as entities which help the user to gain meaningful insight from the data. In some cases, features may be a subset of either raw or derived data, which persist through multiple scales, time-steps, and/or other attributes, which give multiple series of the same dataset.

We focus on *geometry-based point* or *geometric classification*, which is the first step in *detection and classification of features*, such as folds, creases, junctions, and planar surfaces. In point clouds, the features are determined using local geometric descriptors (LGDs). Such a descriptor at a point is a variable which *describes* the shape of local neighborhood of any point [4]. The shape of the local neighborhood as a key criterion for point classification, which gives the *geometric classes*[1], namely line-, surface-, and (critical/degenerate) point-type features. The outcomes of *point classification* imply that the point will belong to a feature, namely, the line-, surface-, or point-type feature, respectively. The choice of LGD plays a crucial role in influencing the point classification outcomes. We refer to geometry-based point classification as “classification,” hereafter.

The 3D point-sampled geometric datasets generally encountered in graphics and modeling communities and LiDAR (Light Detection and Ranging) datasets are essentially 2.5-dimensional data. Specifically, in the case of airborne LiDAR data, the point cloud is a height map of two-dimensional planar geometry (latitude-longitude coordinate system). Airborne LiDAR point clouds include geometry of arbitrary topology from multiple objects in environmental scans, owing to which several methods in point-sampled geometry cannot be directly extended to LiDAR point clouds. The resolution of the scans are different. In airborne LiDAR points, the resolution is lower, compared to other point-sampled geometry obtained from indoor or other controlled settings.

Covariance analysis of local neighborhood of LiDAR point clouds is used conventionally for geometry-based point classification, which gives satisfactory results [2], [10]. Classification using covariance matrix is followed by a processing step, such as minimum spanning tree [6], [10], [23] to find sparse set of points which reveals the features. Multi-scale extension of the covariance-based methods has improved the identification of various visually significant features [10], [23]. However, pruning of an already sparse sets of points leaves these methods ineffective in detecting sharp features. At the same time, the

This document is preprint, as on June 05, 2017. The authors are with Graphics-Visualization-Computing Lab, IIIT Bangalore. email: jnair@iiitb.ac.in , beena.kumari@iiitb.org

[1] Here, we use *geometric classes* and *feature classes* interchangeably.

computer vision community have proposed the use of weighted covariance matrix and tensor voting[2] for unique signatures for local surface description [28] and perceptual organization for feature extraction [22], [30].

To bridge the gap between these communities, we propose improving classification outcomes in LiDAR point clouds, specifically of line-type features which include sharp features, by using variants of weighted covariance matrix and voting tensor as LGDs. Our goal is to compare the performance of the aforementioned LGDs in extracting sharp features, such as gabled roofs in buildings.

Motivating Problem: In the LiDAR community, 3D point clouds are beneficial as they do not have as many occlusion problems or shadow casts found in aerial imagery. In practice, fusion of the two types of datasets gives good results for object-based classification (buildings, road, natural ground, and vegetation). Building detection is a key outcome, which is followed by outline delineation and 3D reconstruction of the detected buildings. The outlines of building, which are derived using aerial imagery, are known to be more accurate and of better quality in comparison to those from point clouds [25]. In practice, building outlines are derived using a fusion of imagery and 3D data. Improving 3D building reconstruction implies improving either the fusion of datasets from hybrid sources, or the geometry extraction of building outlines exclusively from LiDAR point clouds; both of which have been identified as open research problems [25]. For the latter, we propose extraction of line-type features and assembling them to extract outlines. In this paper, we address the improvement of line-type feature extraction by proposing a probabilistic method for identifying points which belong to the line-type features. In the scope of this paper, we perform only the geometry-based point classification, along with preliminary results of line-type feature extraction.

Here, we compare the performance of different variants of LGDs based on the weighted covariance matrix and tensor voting, for geometry-based classification. The latter takes into account continuity and proximity principles. Tensor voting methods are known to behave consistently across a variety of datasets, including de-noised LiDAR point clouds. Additionally, we propose application of anisotropic diffusion after applying tensor voting, for enhancing line-type features, as well as giving classification outcomes similar to that of covariance matrix. Owing to the inherent uncertainty in classification, we propose a probabilistic approach for computing saliency map that gives the classification. We additionally use a multi-scale method to compute our novel probabilistic saliency map, where the radius of the local neighborhood is the scale [10], [22], [23]. Since we did not find any existing literature on methods of comparing classification outcomes of two different LGDs, for performance analysis, we perform comparisons by aligning the reference frames of the LGDs. We perform qualitative comparative analysis using visualization. For visualizing the classification outcomes, we take the following approach of: (a) representing LGDs as second-order tensors and using state-of-the-art superquadric tensor glyph visualization [12], [26], and (b) using our novel multi-scale saliency map to guide the colormap for visualization. We propose the use of juxtaposed views and classification matrix visualization, for composite visualizations for comparative analysis of classification outcomes of different LGDs. Our work is unique in comparing point classification outcomes. Existing methods for surface reconstruction and surface fitting use classification as one of the preliminary steps, without any in-depth analysis, which limits the scope of usage of geometry for classification.

Using our proposed methods and additional quantitative analysis, we have shown how different LGDs perform in line-type feature detection (Figure 1[3]). We observe that more points are detected as line-type features, when using tensor voting in comparison to conventional covariance analysis. The larger number of points allows better extraction of sharp (line-type) features [4]. Our contributions are:

1) multi-scale probabilistic geometric classification of airborne LiDAR point clouds using a novel saliency map;
2) novel approaches for visual comparison of performance of LGDs (Figures 1 and 2) in classification, such as: (a) expressing LGDs as positive semidefinite second-order tensors, and consequently, using superquadric tensor glyphs for visualization; and (b) visualizing the classification matrix;
3) comparative analysis of variants of two existing LGDs for point classification, computed using weighted covariance matrix and tensor voting, respectively.

Notations: We refer to a point in a point cloud, $\mathscr{P}$, as $x \in \mathscr{P}$, and the local neighborhood of x, for a given radius r, is $N(x) = \{x, y \in \mathscr{P} : \|y - x\| < r\}$. Covariance matrix, voting tensor and weighted covariance matrix are annotated as C, V, and C_w. Our proposed modifications to the LGDs are indicated as superscripts, e.g. LRF and AD refer to the alignment of the local reference frame (LRF) and anisotropic diffusion, respectively. Thus, C^{LRF} and V^{LRF} are the LRF-aligned weighted covariance matrix and voting tensor, respectively. V^{LRF-AD} is the tensorial outcome of applying anisotropic diffusion on V^{LRF}. We use shorthand notations for our proposed LGDs, C^{LRF} and V^{LRF-AD}, as T_T and T_N, to highlight the tensor representation of the LGDs and their relationship to the tangent and normal spaces of the point, respectively.

[2] Here, we disambiguate *tensor voting* as the algorithm, and *voting tensor* as the second-order tensor, which is the outcome of the algorithm.

[3] In Figures 1, 4, and 7–10, color coding correspond to a geometric class or the combination of classes a point belongs to, which is determined by using the saliency maps. We use the colorblind safe color palette options from ColorBrewer2.0 http://colorbrewer2.org/.

[4] We have demonstrated results on fan-disk and smooth-feature datasets, apart from airborne LiDAR datasets, purely for more comprehensible comparative analysis of LGDs for classification.

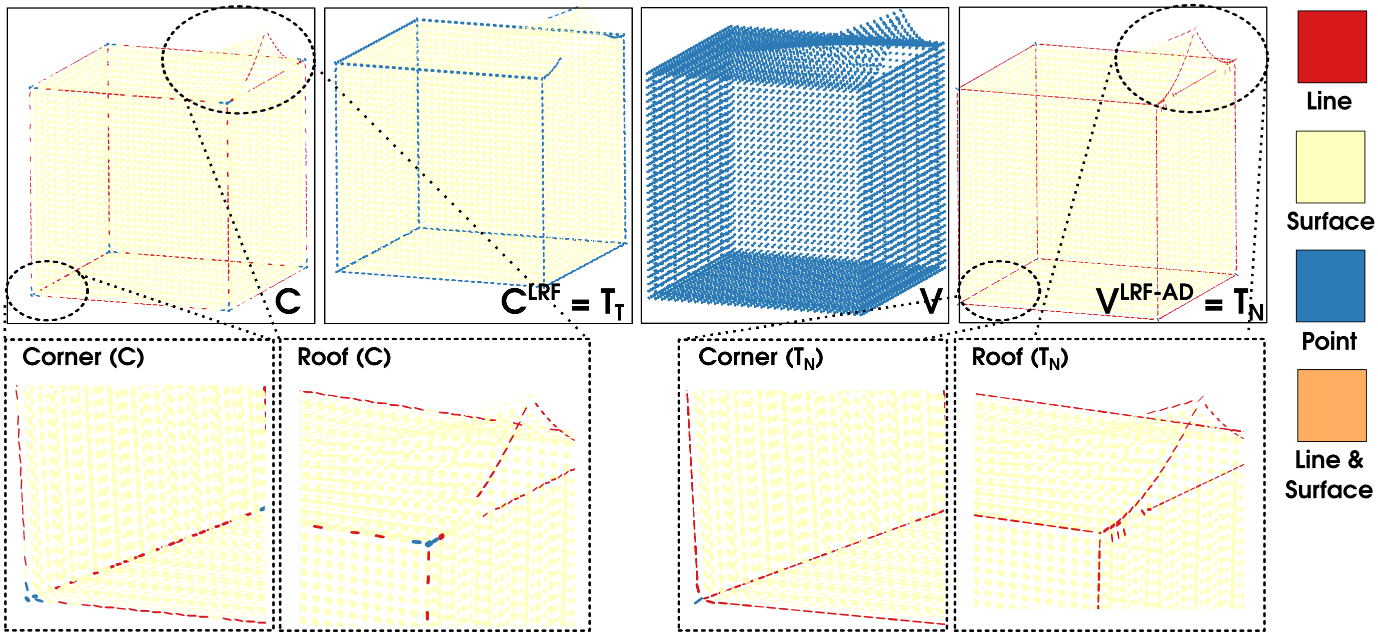


Fig. 1: Superquadric tensor glyphs for visually comparing LGDs: (Top) C, C^{LRF}, V, and V^{LRF-AD}, using a geometric class-based colormap. The LGDs computed at scale, $r = r_{min}$ for smooth-feature (6,177 points). The bottom row shows zoomed out portions of the corner and roof, where glyphs represent (left) C and (right) V^{LRF-AD}. Note that, in comparison to C, V^{LRF-AD} shows the line-type features using more cylindrical glyphs with red color, indicating higher saliency in line-type features. Thus, our novel saliency map determines the geometric classification.

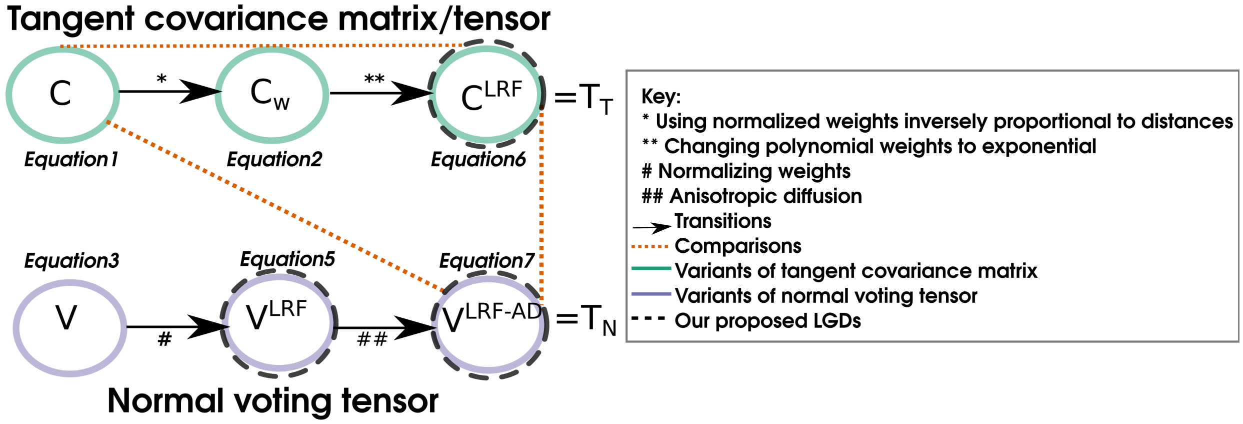

Tangent covariance matrix/tensor
C
*
C_w
**
C^LRF
=T_T
Equation1
Equation2
Equation6
Equation3
Equations5
Equation7
V
#
V^LRF
##
V^LRF-AD
=T_N
Normal voting tensor
Key:
* Using normalized weights inversely proportional to distances
** Changing polynomial weights to exponential
# Normalizing weights
## Anisotropic diffusion
Transitions
Comparisons
Variants of tangent covariance matrix
Variants of normal voting tensor
Our proposed LGDs

Fig. 2: Summary of the different LGDs we compute and how we compute them. The dotted lines indicate the pairwise comparisons of LGDs done in this paper.

II. Related Work

We describe relevant work on LGDs, its tensor representations, and the tensor field visualization techniques. We have briefly looked at relevant work on the use of LGDs for the classification in computational geometry and LiDAR communities.

LGDs for Point Clouds: Gumhold et al. [6] have proposed feature classification and extraction using correlation matrix and neighborhood graph. The correlation matrix is the total least squares problem for surface fitting, which has been proposed by Hoppe et al. [8]. In [6], tensor voting scheme was explicitly avoided for feature detection and extraction, in order to avoid volume discretization. However, more recent research [22] has shown that tensor voting can be applied directly on point clouds without volume discretization. Gelfand et al. [4] have made a distinction between high- and low-dimensional LGDs. High-dimensional or richer LGDs (spin images, shape contexts, their proposed integral volume descriptors) are generally used for object recognition and shape retrieval. The low-dimensional LGDs (curvature-related descriptors) are easier to compute, store and use – hence is used with voting schemes or iterative alignment methods for shape matching and point cloud registration. LGDs considered for the point classification are generally low-dimensional ones. Integration based LGDs are recommended for feature classification in [4], however point classification in LiDAR point clouds is conventionally done using differentiation-based low-dimensional descriptors.

Tensor Representation of LGD: Knutsson [13] has used tensor definition for structure tensor, which is based on differentiation of functions and is used as LGD for 3D space. Knutsson et al. [14] have enumerated LGDs for images, some of which can be extended to 3D point clouds.

Point/Feature Classification: In the LiDAR community, geometric classification is often called "structural classification." Point classification and feature classification are often interchangeably used in literature. Several methods for feature classification of point clouds exist in literature [22], [31]. Attributes from feature/structural classification (e.g. linearity, anisotropy, etc.) are extensively used for semantic or object-based classification in LiDAR point clouds [2], [16], [17], which use covariance analysis. Kim et al. [11] have used 3D tensor voting for wall and roof segmentation and classification in LiDAR point clouds. They have generated surface meshes from the point clouds, where tensor voting is used for fitting a planar surface in local neighborhoods and segmenting the surface mesh [19]. The mesh segmentation is similar to [19], [30]. While our proposed LGD using tensor voting can be used for orientation just as in [11], our goal is to extract line-type features, which can be further used for segmentation.

Covariance analysis of local neighborhoods based on centroid is a robust method for normal estimation [15], but not necessarily for finding the shape of the neighborhoods. Two neighboring points with approximately similar local neighborhoods with the same centroid will have the same LGD, which makes the LGD fail specifically for the classification of sharp features. Moreover, all neighbors in the local neighborhood are considered equal in the covariance matrix, which is not the case. The rationale is that its nearer neighbors have more "influence" in reconstructing the surface at the point than its distant neighbors. Thus, a weighted covariance matrix computed with respect to a point is a more accurate "signature" of a point compared to the conventionally used covariance matrix. Kobbelt and Botsch [15] have discussed that for consistent orientation of normals estimated using covariance analysis, propagation along a minimum spanning tree will be needed. Similarly, Tombari et al. [28] have made the argument of lack of repeatability of sign of LRF when using the covariance matrix, and have proposed a weighted covariance matrix based on the point itself instead of the centroid, for surface matching. They have proposed an LGD for surface matching, based on signatures of histograms, where local reference frame can be made unique, repeatable, and an invariant for a point. For classification of sharp features, moving least squares method [1], [3] or Gauss map clustering [31] have been used effectively, which are not centroid-based methods. Local tensor-based techniques are a tradeoff between computational complexity and accuracy in feature detection; e.g. use of tensor voting [22], [30] for feature classification.

Multi-scale Classification: Pauly et al. [23] have proposed the use of multi-scale surface variation, estimated using covariance matrix of local neighborhood. There, surface variation at a user-defined scale gives feature weights, which on appropriate thresholding gives features. Keller et al. [10] have used a similar multi-scale approach, for LiDAR point clouds, in determining feature weights from covariance matrix of local neighborhoods. However, the difference between the methods in [23] and [10] is that a single adaptive scale and averages across multiple scales have been used, respectively. Algorithms for finding optimal neighborhood size or scale has been of interest to the LiDAR community [2], [5], [17], [33].

Tensor Voting: Guy and Medioni [7] have proposed tensor voting scheme to detect and classify the feature points in structured point cloud. The voting scheme uses the proximity and continuity principles of Gestalt psychology to propagate the votes. Each point is encoded as tensor based on given input data information, and nonlinear tensor voting has been performed using local neighbors to update the tensor information at each point in 3D data. Saliency maps, computed from eigen analysis of the second-order tensor obtained after vote aggregation, give the likelihood of each point belonging to different feature classes, i.e., ball-, stick-, or plate-tensor.

Park et al. [22] have used tensor voting and surface variation to classify and detect line features in point clouds, where th surface variation function is computed using a multi-scale method. Our proposed tensor, obtained after anisotropic diffusion upon tensor voting, V^{LRF-AD}, is different from that from tensor voting, V, in [22] as the classification in the latter does not confirm to the conventional method [10], as point-type features in the former are equivalent to line-type features in the latter, as per the classification given in the latter; while ours matches with the conventional method. Park et al. compute an optimal scale from multiple scales for further analysis, whereas we propose an *aggregated* multi-scale saliency map. In our case, the anisotropic diffusion after tensor voting enhances line-type feature detection as well as modifies the classification outcomes of the new tensor to match with that of the covariance matrix. Our approach based on anisotropic diffusion is inspired from [30], where anisotropic diffusion is performed after tensor voting for feature classification and extraction in polygonal mesh data, and subsequent mesh segmentation.

Tensor Field Visualization: We use superquadric tensor glyphs [12], [26] for comparing different (positive-semidefinite) second-order tensor fields of the LGDs. These glyphs are designed using mirror symmetry, symmetry, continuity, and disambiguity as guiding principles. Thus, by design, these glyphs show the shape and orientation of a positive-semidefinite second-order tensor more effectively than other conventionally used glyphs. Zhang et al. [34] have recently proposed the use of *tender glyphs* for simultaneously comparing two different tensor fields.

III. Local Geometric Descriptors

Conventionally used covariance-based methods have limitations in capturing line-type features [31]. They often detect points on the sharp features contrarily as weak line-type features (Figure 1). However, the use of an LGD which enhances the sharp features can improve the classification. Hence, we propose two different LGDs, which are variants of the weighted covariance matrix, C_w, and tensor voting, V. We further exploit the tensor representation of the LGDs for its analysis.

In our work, C_w is an improvement over C for two reasons. First, C_w at the point gives a unique LGD, thus giving a "signature" of the point itself. Additionally, in the case of the classification using C_w. the normal estimation is not done using the centroid. Second, the contribution of all neighbors in the computation of $C(x)$ are equal, i.e., a distant neighbor will "influence" the shape of the local neighborhood as much as a neighbor in closer proximity. Contrarily, these contributions are inversely proportional to the distance between the neighbor and the point, x. Hence, a weighted covariance matrix, such as C_w, is ideal as an LGD. Here, we study how the changes proposed by Tombari et al. [28] influence the geometric classification. Similarly, tensor voting can also be considered as an improvement over C. Tensor voting approach follows the Gestalt principles of proximity and continuity, which makes the resultant tensor viable as an LGD.

Comparing the performance of LGDs can be done in two ways, namely, comparing the relative performance of each LGD against the ground truth and comparing the outcomes of the LGDs, taking two at a time. For the latter, the classification outcomes for two LGDs, say C_w and V, are comparable if the comparisons can be done using a common invariant, e.g. local reference frame (Figure 3). Additionally, one must be aware that there are cases where the outcomes of the classification of the two LGDs are different and hence, not comparable, e.g. C (or C_w) and V (Figure 1). However, certain modifications to the LGDs can resolve these differences and facilitate the comparisons subsequent to modifications, e.g., anisotropic diffusion after tensor voting makes its classification outcomes comparable to those of C (Figure 1).

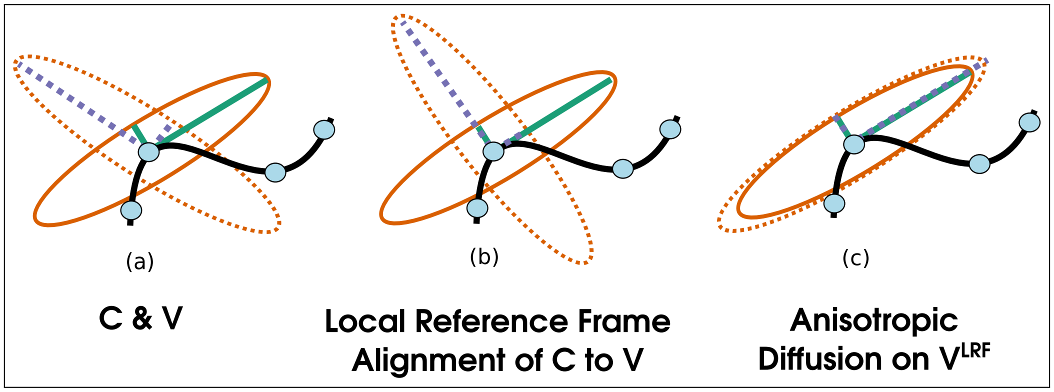

(a)
(b)
(c)
C & V
Local Reference Frame Alignment of C to V
Anisotropic Diffusion on V^LRF

Fig. 3: A schematic showing modifications to LGDs, C and V, to compare their point classification outcomes effectively. *For points/samples (circle) on a curve (in $\mathbb{R}^2$), the LRF and shape of local neighborhood obtained from the spectral decomposition of the LGD are shown (solid lines and ellipse for those of C; and dashed lines and ellipse for those of V). (a) C and V at a point; (b) The LRF of C has been aligned to that of V; (c) Anisotropic diffusion has been applied to V, which gives V^{LRF-AD}. Similar to C, this is applicable to C_w as well.*

Local Geometric Descriptors LGDs at a point, in $\mathscr{P}$, encodes the geometry of the local neighborhood of the point. LGDs are important, as the local geometric analysis leads to the global description for the entire point cloud. Here, we focus on two LGDs, computed using the conventionally used covariance matrix and tensor voting, respectively. Here, we discuss the construction of these variants of the LGDs.

Weighted Covariance Matrix: The covariance matrix [8], also called correlation matrix [6], is computed as $C(x) = \sum_{y \in N(x)} (y - \bar{x})(y - \bar{x})^T$, where $\bar{x}$ is the centroid of the local neighborhood $N(x)$. We refer to it as the "tangent covariance matrix," as it is constructed using tangent vectors. C is conventionally used as an LGD in the LiDAR community [2], [5], [17], [33]. However, in computer vision applications, weighted covariance matrix is used for surface description and local sign disambiguation [28]. Since we are performing surface description as opposed to surface fitting in LiDAR point clouds, we propose the use of weighted covariance matrix, C_w (Equation 1) for the same. The weights have the property of inverse proportionality to distance between the concerned point and its neighbor. *Local reference frames* (LRF) are invariants that define the local neighborhood of a point uniquely. The eigenvectors of the LGD are conventionally used as basis vectors of the LRF. C_w gives repeatable LRFs, as opposed to C. Repeatable LRFs give non-repeatability of sign of principal directions across multiple objects in the scene [28], which are applicable in the case of LiDAR point clouds.

Voting Tensor: In tensor voting [19], there is a notion of a receiver and a voter, where the voter casts a vote to the receiver. A vote, defined in the form of a tensor, contains information of the neighbor in the local neighborhood. A predefined voting field is used to aggregate the votes as a tensor, and to provide a saliency map for each feature type. Saliency values of the resultant tensor, which is the sum total of all information collected at the receiver, are scalar values derived from the spectral decomposition of the tensor. The saliency values, "pointness," "curveness," and "surfaceness", correspond to feature (or structural) classes, namely, ball- (point), stick- (line), and plate- (surface) tensors, respectively. Thus, these classes correspond to the feature classes in point sampled geometry. In tensor voting, we initialize the tensor $V(x)$ as a ball tensor, for unoriented points (i.e. points without predetermined normal information) [22], as is our case (Equation 1). In Equation 1, I_d is a d-dimensional identity matrix, and scale parameter σ. We use the radius of the neighborhood, r as the scale parameter, thus, $\sigma = r$. While σ influences the vote propagation range, the Gaussian function, μ_y, is the attenuation factor for the size of the vote collected from y at x. Here, we can see that $V(x)$ is a weighted covariance matrix just as $C_w(x)$, with the difference that they are constructed from normals and tangents, respectively. For $y \in N(x)$, $t(y) = (y - x)$ and $z(y) = \frac{\|t(y)\|_2}{r}$,

$$C_w(x) = \sum_{y \in N(x)} w_y t(y) t(y)^T; \qquad V(x) = \sum_{y \in N(x)} \mu_y . \left(I_d - \frac{t(y)t(y)^T}{t(y)^T t(y)} \right) \tag{1}$$

$$\text{where } w_y = \frac{1 - z(y)}{\sum_{y \in N(x)} 1 - z(y)}; \qquad \text{and } \mu_y = exp\left(- \frac{\|t(y)\|_2^2}{\sigma^2} \right).$$

Alignment of Local Reference Frame for Local Geometric Descriptors: An LRF is an invariant of the LGD of a point, and a common LRF for two LGDs implies that the orientations of the local neighborhoods defined by the LGDs are the same, at the point. Thus, comparison of LRF-aligned LGDs reduces to qualitative comparison of the shapes of these local neighborhoods at each point. Hence, we propose variants of C_w and V, whose LRFs are aligned, to facilitate their comparisons. Our construction of these proposed LGDs are:

1) Modification to V, by normalizing the weights for I_d in Equation 1. Given $\sigma = r$, we get $\mu_y = exp(-z(y)^2)$. Thus, we replace μ_y (Equation 1) with μ'_y (Equation 2) to give an *LRF-aligned voting tensor*, denoted as $V^{LRF}(x)$.

$$V^{LRF}(x) = \sum_{y \in N(x)} \mu'_y . \left(I_d - \frac{t(y)t(y)^T}{t(y)^T t(y)} \right), \text{ where } \mu'_y = \frac{\mu_y}{\sum\limits_{y \in N(x)} \mu_y} \tag{2}$$

2) Modification to C_w, by normalizing the tangent vector $t(y)$ and modifying the weights in the computation of C_w from w_y in Equation 1 to w'_y. This change in weight (Equation 3) is permissible as $w'y$ has the same property that the weights w_y and μ_y (Equation 1) have, i.e., they are monotonous functions of dimensionless distance quantity $z(y)$, and are inversely proportional to the distance of the neighbor to the point, i.e. $\|t(y)\|$. Thus, we get the *LRF-aligned tangent covariance matrix*, $C^{LRF}(x)$, as follows:

$$C^{LRF}(x) = \sum_{y \in N(x)} w'_y t(y) t(y)^T, \text{ where } w'_y = \frac{\mu_y}{t(y)^T t(y) \cdot \sum\limits_{y \in N(x)} \mu_y}. \tag{3}$$

However, by aligning LRFs for the LGDs this way, the weights for weighted covariance matrix, C^{LRF}, are not convex, unlike the weights proposed in [28]. Since convexity is not recommended as a requirement in [28], we disregard it for the sake of alignment of LRFs. Even though Equations 2 and 3 lead to different computations of the LGDs, they effectively perform rotations of the eigenvectors, as schematically shown in Figure 3. Hence, we refer to these modifications as LRF "alignment".

Anisotropic Diffusion on Normal Voting Tensor: While tensor voting is good for perceptual grouping using its saliency maps, the point classification through its tensor V or its variant (V^{LRF}) is not similar to that of the covariance matrix (C) or its variants (C_w or C^{LRF}), as shown in Figure 1. Hence, we perform anisotropic diffusion after aggregating votes in the tensor, so that the resulting tensor V^{LRF-AD} will give similar classification outcomes as C. We explain the rationale behind differences and the resolution using anisotropic diffusion in Section VI.

Wang et al. have used V^{LRF-AD} as the diffusion tensor in the heat diffusion equation [30], for polygonal mesh segmentation. The rationale is that anisotropically controlling diffusion velocities in the diffusion tensor, would lead to slower diffusion across sharp edges and faster diffusion along sharp edges. Thus, anisotropic diffusion has been proven to enhance line-type features in a surface mesh. Along these lines, we propose applying anisotropic diffusion after tensor voting, so that the resultant tensor can be used as the LGD of the point cloud to classify line-type features more accurately than the tangent covariance matrix. We perform anisotropic diffusion by modifying the eigenvalues of the diffusion tensor as an exponential term and addition of a diffusion parameter, δ, which controls diffusion velocities.

Anisotropic diffusion is applied to V^{LRF} by modifying its eigenvalues $(\lambda_V^{LRF})_i$ (for $i = 0, 1, 2$) using an exponential function, $f((\lambda_V^{LRF})_i) = exp(-(\lambda_V^{LRF})_i / \delta)$. However, the use of an exponential function reverses the order of eigenvalues:

$$(\lambda_V^{LRF})_0 \geq (\lambda_V^{LRF})_1 \geq (\lambda_V^{LRF})_2, \text{ gives}$$
$$f((\lambda_V^{LRF-AD})_0) \leq f((\lambda_V^{LRF-AD})_1) \leq f((\lambda_V^{LRF-AD})_2.$$

Thus, for anisotropically diffused LRF-aligned voting tensor V^{LRF-AD}, with indices $i = \{0, 1, 2\}$ corresponding to the descending order of eigenvalues, we get:

$$(\lambda_V^{LRF-AD})_i \quad = \quad exp(-(\lambda_V^{LRF})_{2-i} / \delta); \text{ and } (e_V^{LRF-AD})_i = (e_V^{LRF})_{2-i} \tag{4}$$

$$\text{Finally, } V^{LRF-AD}(x) \quad = \quad \sum_{i=0}^{2} (\lambda_V^{LRF-AD})_i \cdot (e_V^{LRF-AD})_i (e_V^{LRF-AD})_i^T \tag{5}$$

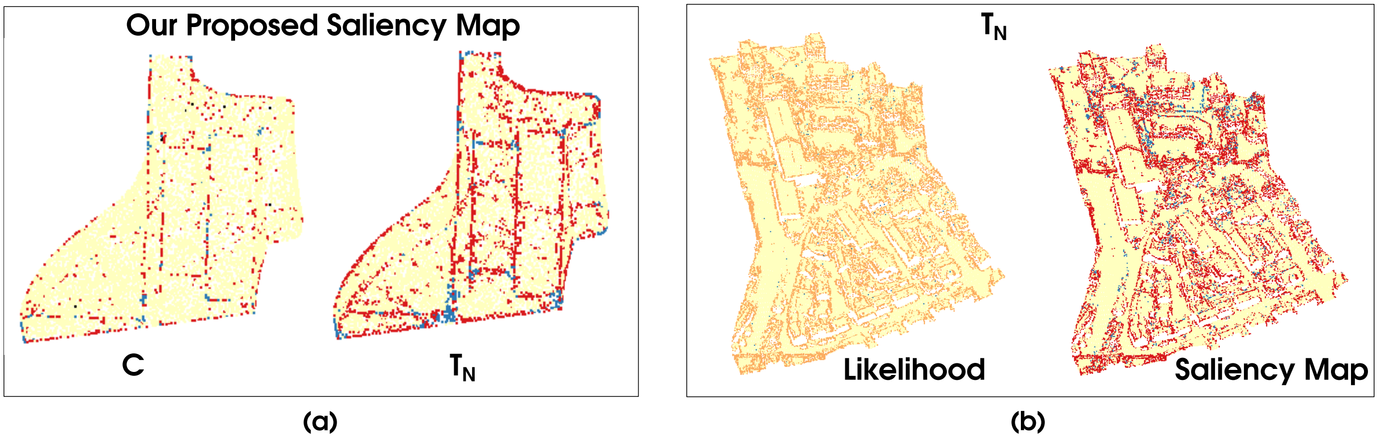


Fig. 4: (a) Comparing LGDs for fan-disk (10,921 points), using our proposed saliency map. (Left) is the tangent covariance matrix C and (right) is our proposed anisotropically diffused voting tensor V^{LRF-AD} (i.e., T_N). (b) Comparison of saliency maps computed for our proposed voting tensor V^{LRF-AD}, using likelihood values in shape classes [line,point,surface], given in (left) Equation 8, and (right) Equation 9, for Region-1 in Vaihingen (179,997 points).

Impact of Proposed Modifications: Here, we mathematically analyze the modifications caused by the alignment of LRFs of C_w and V and anisotropic diffusion of V, shown schematically in Figure 3. Let us first compare the eigenvalues and eigenvectors from $C^{LRF}(x)$, and $V^{LRF}(x)$. Suppose the set of eigenvectors for $C^{LRF}(x)$ and $V^{LRF}(x)$ are given by $(e_C^{LRF})_i$ and $(e_V^{LRF})_i$, respectively, for $i = 0,1,2$. Similarly, let the set of eigenvalues be $(\lambda_C^{LRF})_i$ and $(\lambda_V^{LRF})_i$, respectively.

$$V^{LRF}(x) \quad = \quad I_d - C^{LRF}(x), \text{ from Equations 2 and 3.} \tag{6}$$

$$(\lambda_V^{LRF})_i \quad = \quad 1-(\lambda_C^{LRF})_{(2-i)}, \text{ and hence, } (e_V^{LRF})_i = (e_C^{LRF})_{(2-i)}, \text{ for } i = 0,1,2. \tag{7}$$

This implies that the ordering of the eigenvalues are reversed in $V^{LRF}(x)$ compared to those of $C^{LRF}(x)$, thus reversing the order of their eigenvectors (i.e. major eigenvalue of V^{LRF} corresponds to minor eigenvalue of C^{LRF}, and so on). This explains the difference in the classification outcomes between the descriptors in Figure 1. However, it is desirable that the ordering of the eigenvalues be the same, so that the eigenvectors at a point can be the same, thus giving the same LRF and similar classification outcomes from both descriptors. A similar transformation is used in [21] to change tangential orientations to normal ones for structure estimation in images.

Thus, from Equations 4 and 6, we conclude that $(e_V^{LRF-AD})_i = (e_C^{LRF})_i$. This shows that the LRFs of V^{LRF-AD} and C^{LRF} match (i.e. their major eigenvectors correspond to each other, and similarly, their middle and minor ones). This also implies that the ordering of their eigenvalues also match, thus giving similar classification outcomes for both the LGDs, i.e. variants of the covariance matrix and the anisotropically diffused voting tensor (Figures 1 and 4(a)).

Positive Semidefinite Second-order Tensor Representation We did not find any metrics in current literature for comparing LGDs and their classification outcomes. We propose expressing these LGDs as a specific mathematical quantity, for validating applications of mathematical operations such as, comparisons and replacements. We have found that second-order tensors are the most natural choice of quantity, owing to the inherent construction of the LGDs as tensor products. Local neighborhood can be described using various variables or approaches, such as spin images, covariance analysis, tensor voting, curvature, etc. Here, we revisit the use of tensors in representing LGDs used for structural analysis in image processing and computer vision [14], to apply the same for point-sampled geometry. While some LGDs are explicitly defined as tensors (e.g. in tensor voting [20] and curvature tensor [27], [29]), the others are described using matrix representation (covariance matrix [8], [10]). Nevertheless, in many such cases, the LGD is constructed using an outer product or tensor product of vectors, $T(x) = xx^T$, where $x \in \mathbb{R}^d$ is either the tangent or the normal at the point. From Equations 1, 2, 3, and 5, we see that all LGDs discussed in this paper are indeed *positive semidefinite second-order tensors*. By design, the tensor fields of LGDs correspond to the *tangent* or the *normal spaces to the manifold* defined by the point cloud, respectively, at the concerned point [24]. The tangent and normal spaces form disjoint sets themselves. Here, the covariance matrix and its variants (C, C_w, C^{LRF}) correspond to the tangent space; and the tensor and its variants from tensor voting (V, V^{LRF}, V^{LRF-AD}), to the normal space. Hence, hereafter, we use the shorthand notations for C^{LRF} and V^{LRF-AD} are T_T and T_N, respectively.

IV. Multi-scale Probabilistic Point Classification

For probabilistic point classification, we propose an appropriate saliency map of the feature classes at each point, which is aggregated from LGDs across multiple scales.

Multi-scale Approach: The advantage of using multi-scale approaches in comparison to single scale is that they do not rely on a single scale for procuring the geometric information. The distributed approach improves the richness of the geometric information. Determining the shape of the local neighborhood has high degree of uncertainty, especially in multi-object boundaries in LiDAR/computer vision datasets. When using single scale, obtained optimally or adaptively, the error margin becomes high for such boundary cases. Additionally, multi-scale approach determines persistence from various scales, as opposed to adaptive scale, which depends on a scale where the persistence is high. Thus, multi-scale methods capture persistence of features better. However, the number of scales and bounds of the scales become crucial for the success of the multi-scale method. Currently, these values are user-determined, which is a limitation. Multi-scale approaches also fail for cases where there is a large difference in sampling density in the dataset, as the covariance analysis fails in the case of sparse local neighborhoods. This can be alleviated by using k-nearest neighborhoods instead of spherical neighborhood [15]. One of the the drawbacks of using adaptive scales or optimal scales is that they become computationally intensive for large point clouds. More operations are required, in the former, for computing adaptive scales for each point, and in the latter, for optimizing a single scale for all points.

Despite their drawbacks, multi-scale approaches have worked favorably for point classification in certain applications, as preserving the persistence of features across several scales gives a better measure of the feature strength or saliency map [10], [22], [23]. Keller et al. [10] have proposed probabilistic point classification for LiDAR datasets, where eigenvalues of the covariance matrix is used to determine neighborhood shape and feature strengths at each scale, and finally, the likelihood of the point belonging to the three feature classes are averaged across scales. For each scale, r (i.e., radius of local neighborhood), given that $\lambda_0(x,r) \geq \lambda_1(x,r) \geq \lambda_2(x,r)$, which are eigenvalues of $C(x)$, three subsets corresponding to the classes are defined to bin all the points in the point cloud, using a user-defined threshold, ε: $P_l(r) = \{x \in \mathscr{P} | \lambda_1(x,r)/\lambda_0(x,r) < \varepsilon\}$; $P_s(r) = \{x \in \mathscr{P} | \lambda_2(x,r)/\lambda_0(x,r) < \varepsilon\}$; and $P_p(r) = \{x \in \mathscr{P} | \lambda_2(x,r)/\lambda_0(x,r) \geq \varepsilon\}$. Here, by design, $P_l(r) \subset P_s(r)$. The likelihood of a point belonging to (line, surface, point) feature classes, $\{L_l, L_s, L_p\}$, are computed using a multi-scale approach. While Keller et al. have not used the likelihoods explicitly for classifying the points, they have used these values to compute feature strengths, which is further used for generating feature graphs, and refining the classification. Given N_r scales uniformly sampled in [r_{min}, r_{max}],

$$\begin{aligned} \mathbf{L_{l,s,p}(x)} &= \frac{1}{N_r} \cdot \sum_{\mathbf{i=1}}^{\mathbf{N_r}} \phi_{\mathbf{l,s,p}}(\mathbf{x}, \mathbf{r_i}), \qquad \text{where } \phi_s(x,r) = \begin{cases} 1, & \text{if } \lambda_2(x,r) \geq \varepsilon\lambda_0(x,r) \\ 0, & \text{otherwise} \end{cases}; \\ \phi_l(x,r) &= \begin{cases} 1, & \text{if } \lambda_1(x,r) < \varepsilon\lambda_0(x,r) \\ 0, & \text{otherwise} \end{cases}; \text{ and } \phi_p(x,r) = \begin{cases} 1, & \text{if } \lambda_2(x,r) < \varepsilon\lambda_0(x,r) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

The averaging operation of the likelihoods across scales, can be seen as a union of N_r mutually exclusive events of the point belonging to the given class for N_r scales, where the events have equal probability, i.e. $1/N_r$. However, $L_{l,s,p}(x)$, as computed in Equation 8, causes higher incidence of points having "mixed" likelihoods, specifically across line- and surface-type features (visualized as points belonging to "line & surface" class in Figure 4(b)). This is due to the subset relationship, $P_l(r) \subset P_s(r)$ and binarization of the scale-wise contribution of each class towards multi-scale aggregation (Equation 8). The former leads to the probabilities not adding up to one. The latter causes ambiguities in classification of points that lie in boundaries of two different objects or point clusters e.g. a point lying on a line but in the close proximity of a junction. Additionally, the high dependency of the algorithm on a user-defined parameter ε leads to erroneous classification. To alleviate the ambiguity, we propose a probabilistic multi-scale saliency map.

Multi-scale Saliency Map for Probabilistic Point Classification: Saliency map of an attribute set is a set of values which maps a value for specific attributes, based on the relative "saliency" or "differentiating quality" of the attribute with respect to the others in the set. For instance, in tensor voting, eigenvalues of the resultant tensor have been used conventionally as saliency map of its eigenvectors [7]. In LiDAR point clouds, saliency maps are derived from confidence index of 1-, 2- and 3-dimensional features (i.e. line, surface, and point, respectively) [2]. We propose to use likelihoods of a point belonging to the feature classes as the saliency map of classification of the point. These likelihoods, which explicitly imply the classification outcomes, are derived from the spectral decomposition of LGD, thus, preserving the properties of the LGD. Thus, our classification is a probabilistic or fuzzy one.

We use the likelihood with which a tensor can belong to each of the three shape classes [12], [32] found in 3D point clouds. Given that $\lambda_0(x) \geq \lambda_1(x) \geq \lambda_2(x)$, $c_{l,s,p}(x,r)$, we compute the likelihood of the point x belonging to line, surface, or point class, at a given scale r, $\{c_l, c_s, c_p\}(x,r)$. At each scale, we ensure the sum of likelihoods for a point, $(c_l + c_s + c_p)$, is unity. For multi-scale computation, we average the likelihoods of a point belonging to a given class across N_r scales, i.e.

$$\begin{aligned} \mathbf{L_{l,s,p}(x)} = \frac{1}{N_r} \cdot \sum_{\mathbf{i=1}}^{\mathbf{N_r}} \mathbf{c_{l,s,p}}(\mathbf{x}, \mathbf{r_i}); \qquad \text{where} \qquad & c_l(x,r) = \frac{\lambda_0(x,r) - \lambda_1(x,r)}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}; \\ c_s(x,r) = \frac{2 * (\lambda_1(x,r) - \lambda_2(x,r))}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}; \text{ and } \qquad & c_p(x,r) = \frac{3 * \lambda_2(x,r)}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}. \end{aligned} \tag{9}$$

Since multiple scales can be viewed as a union of mutually exclusive events with equal probability, we can justify averaging of saliency maps across scales for computing the final saliency map. Our proposed saliency map disambiguates classification of points to one of the three classes (Figure 4(b)).

A Note on Scale Parameters: Scale parameters are used for computation of both the tensor V (Equation 1), using tensor voting, as well as saliency maps (Equation 9). However, the parameters are used differently. In the former, during the voting process, as the scale increases, more tokens (or votes) can influence the given point due to increase in size of kernels, and smooth out the noisy data and weak features. Similarly, when the scale is reduced, more details are preserved, which undesirably makes tensor voting more sensitive to noise. In multi-scale classification, scale is used to measure the persistence of the features, thus ensuring a robust feature extraction method. The scale can also be viewed as the regulator for smoothness [23].

V. Comparison of Local Geometry Descriptors

We use tensor field visualization for qualitative comparison of LGDs and their classification outcomes. For comparative visualizations, we use both juxtaposed views and classification matrix visualization. In juxtaposed views, we use point rendering as well as tensor glyph visualization, using a geometric class-based colormap. In both renderings, our proposed saliency map is used to demonstrate the shape and orientation of the LGD and its classification outcomes. Our work is unique in the application of superquadric tensor glyphs for visualizing LGDs in 3D LiDAR point clouds.

A. Juxtaposed Views

The visualization tasks we propose for qualitatively comparing the two second-order tensor fields T_T and T_N are to compare the tensor properties (i.e. shape and orientation) at each point in $\mathscr{P}$, and to compare the classification outcomes at each point. Juxtaposed views [9] of corresponding visualization of the fields for each of the tasks is an effective design pattern as long as we implicitly establish the same orientation and affine transformations for visualizations corresponding to both the fields in a single view. Juxtaposed view is easy to implement and effectively conveys the information on comparing the fields. We use juxtaposed views of LiDAR point clouds using both tensor glyph visualization and point rendering (Figures 1 and 4, respectively).

Tensor Glyph Visualization: We use superquadric tensor glyphs to compare the tensor properties of the LGDs in a single scale. The glyphs are colored using our proposed saliency map. Both saliency map and shape of glyphs encode the information on the likelihood of a point falling into the three feature classes, for instance, red color and cylindrical shape of glyphs indicate line-type features (Figure 1). Thus, the saliency value and the shape of the local neighborhood are both encoded in the glyph visualization of the tensor field.

In addition to single scale analysis, we visualize glyphs of *a multi-scale aggregated tensor representation of the LGD*, or simply multi-scale aggregated LGD. However, we have used this tensor purely for representation, as we have not explored the multi-scale aggregation of LGDs in detail. The aggregated LGD is generated by averaging the unit vectors of eigenvectors and normalized eigenvalues (as the saliency values, c_l, c_s, c_p) values. By construction, the saliency map of the multi-scale aggregated LGDs is the multi-scale saliency map (Equation 9).

Choice of Colormaps: Visualization of saliency maps is done by mapping the data to the RGB color model and using the colormap to render the points. A trivial mapping of the parameterized values $\{c_l(x), c_s(x), c_p(x)\}$ to (red, blue, green) channels, respectively, enables us to visualize the differences in the classification outcomes, computed from different LGDs. This is applicable for both point and glyph rendering. However, this channel-based colormap results in a color palette with red and green hues, which is not favorable for protanopic vision [18]. To resolve this, we use the likelihoods (Equation 8) or saliency values (Equation 9) of a point to determine the class(es) (i.e., line, surface, and point-type features) the point is highly likely to belong to. Both Equations 8 and 9 constrain the classes to be either line-, surface-, or point-type features. In the likelihood based model, the points could belong to two classes simultaneously, which specifically occurs in the case of line- and surface-type feature classes. Hence, overall, we use a geometric class-based color palette, for “line”, “surface”, “point”, and “line & surface” classes (Figure 1).

Geometry-based Point Classification Algorithm:

1) octree construction and outlier removal in $\mathscr{P}$ (as given in [10]);
2) computation of LGD for each point $x \in \mathscr{P}$ ($C(x)$, $T_T(x)$, or $T_N(x)$) (Equation 1, 3 or 5) or its variants;
3) probabilistic point classification by spectral decomposition of LGD for each point, and computation of saliency map for each scale, r, (Equation 9);
4) multi-scale point classification by averaging saliency maps across scales (Equation 8 or 9)
5) visualization of feature classification using (a) juxtaposed views of point or glyph rendering using saliency map, or (b) classification matrix of a specific feature class.

B. Classification Matrix Visualization

We use a superimposed visualization of a *classification matrix* of a feature class of two different LGDs. The classification matrix stores four sets of points which have been labeled as the concerned feature class, by both the LGDs, or exclusively by

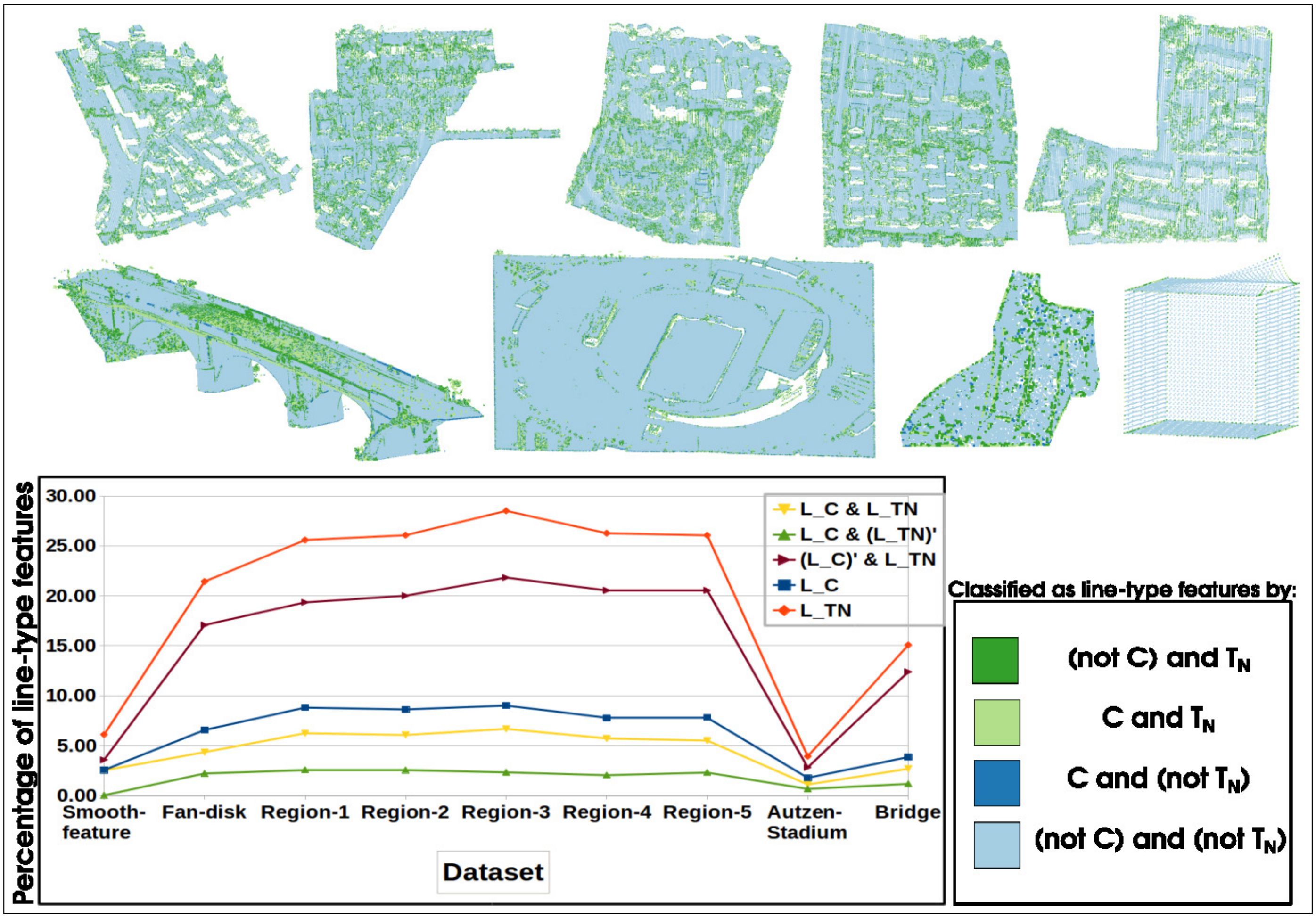


Fig. 5: Classification matrix visualization of line-type features between two LGDs, C and T_N. The color mapping shows which LGD caused a point to be labelled as a line-type feature. L_{TN} is the set of points identified as line-type by T_N and L_C by C. The y-axis in the plot shows the percentage of points identified as line-type, given by the sizes of the sets themselves or set intersections.

each of the LGDs, or by neither. The matrix rows pertain to the sets of points classified by the first descriptor, and the columns are those by the second one. Numerically, the classification matrix gives the size of these sets in its cells, and here, we extend its use to visualizing the points belonging to each cell. We color each of the four sets using different colors. We show the classification matrix visualizations for line class in Figures 5 and 6, to compare between C and T_N, as well as, between T_T and T_N, respectively.

VI. Experiments & Results

We compare LGDs (C, T_T, and T_N) and their classification outcomes to demonstrate that choice of an appropriate LGD can improve line-type point classification (Figure 2). Comparing T_T and T_N shows the performance of two different LGDs in classification, with a fixed LRF. Finally, comparing C and T_N displays the improvement of the outcomes in the classification when using the latter, over the former. C is the state-of-the-art LGD used for airborne LiDAR point clouds.

Experiments: Apart from qualitative comparisons in juxtaposed views of point rendering (Figure 7) and glyph visualizations (Figure 8), we have quantified points classified as line-type features using classification matrices (Figures 5 and 6). We have also studied the influence of scale in LGD computation using tensor glyph visualization (Figure 8(b)), and influence of diffusion velocity in computation of V^{LRF-AD} (Figure 9). The default value of diffusion parameter δ is 0.16, for computing V, V^{LRF}, V^{LRF-AD}, as has been used in [30]. Dataset-specific parameters are given in Table I. For point classification using algorithm by Keller et al. [10], we have used threshold parameter, $\varepsilon = 0.5$. For multi-scale approach, the radii of local neighborhood, used as scales, are measured with respect to a normalized bounding box of the dataset. The normalization of the bounding box gives a canonical view volume, which is a cube of size 2 units and centered at (0,0,0), in $\mathbb{R}^3$.

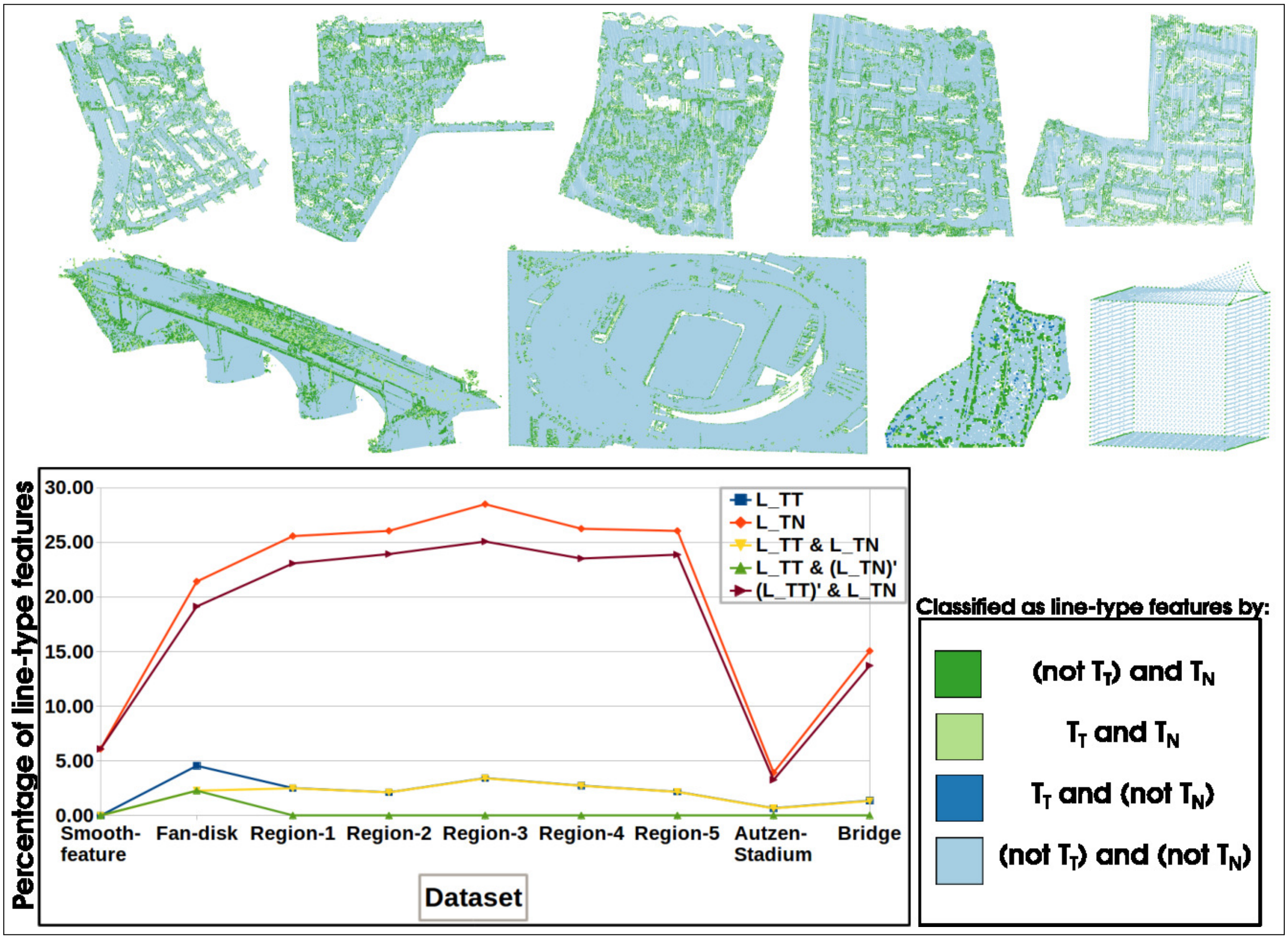


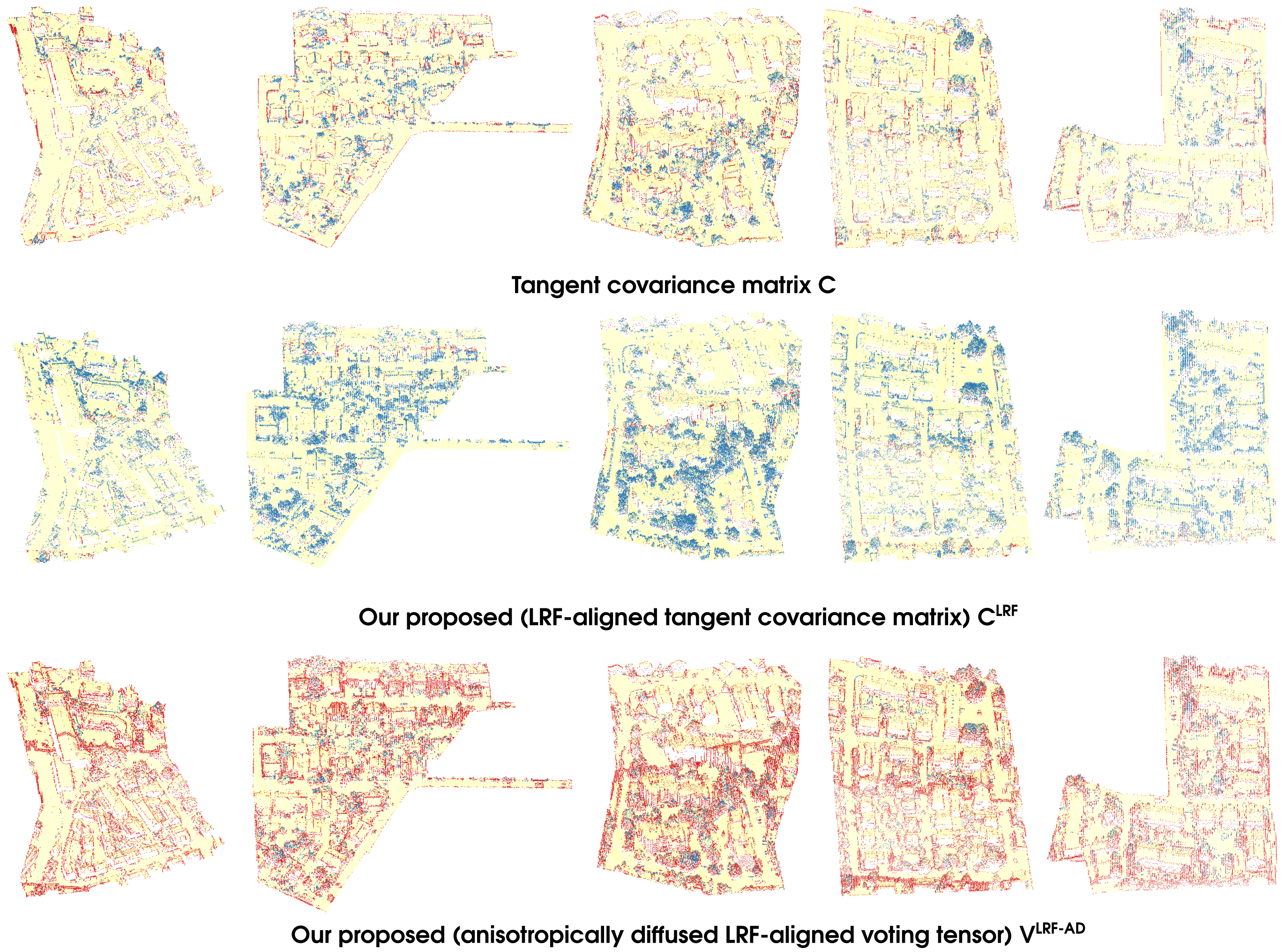
Fig. 6: Classification matrix visualization of line-type features between T_T and T_N. Color mapping is similar to that in Figure 5. L_{TN} is the set of points identified as line-type by T_N and L_{TT} by T_T.

TABLE I: Datasets and the parameters used for computing the local geometric descriptors in our experiments. $r_{min}(T_T)$ and $r_{max}(T_T)$ are user-defined minimum and maximum values for scales, i.e. radius of local neighborhood, used when computing C, C_w, and C^{LRF}. Similarly, $r_{min}(T_N)$ and $r_{max}(T_N)$ are user-defined values for V, V^{LRF}, and V^{LRF-AD}.

Dataset	Region-5*	Region-1*	Region-2*	Region-3*	Region-4*	Autzen Stadium†	Bridge‡	Smooth-feature‡	Fan-disk‡
Size (#points)	163,305	179,997	231,725	266,675	323,896	693,895	5,284,131	6,177	10,921
$\mathbf{r_{min}(T_T)}$	0.009	0.009	0.009	0.009	0.009	0.007	0.004	0.100	0.015
$\mathbf{r_{max}(T_T)}$	0.011	0.011	0.011	0.011	0.011	0.009	0.006	0.200	0.020
$\mathbf{r_{min}(T_N)}$	0.009	0.009	0.009	0.009	0.009	0.008	0.006	0.100	0.040
$\mathbf{r_{max}(T_T)}$	0.011	0.011	0.011	0.011	0.011	0.010	0.008	0.200	0.050

*http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html
†http://www.liblas.org/samples/
‡http://visionair.ge.imati.cnr.it/ontologies/shapes/viewmodels.jsp

We have used datasets (Table I) of airborne LiDAR point clouds*, and point set geometry (for bridge)† to showcase the results of our chosen LGDs and their classification outcomes. There are no datasets existent with ground truth for geometry-based point classification. Hence, we have used simple manifold datasets‡ to showcase for preliminary results, only because the geometry in these datasets is more predictable (i.e. visually verifiable) than the LiDAR datasets. Visual comparative study of behavior and classification outcomes of various LGDs can be better done on these datasets, even in the absence of ground truth for classification.



Tangent covariance matrix C

Our proposed (LRF-aligned tangent covariance matrix) C^{LRF}

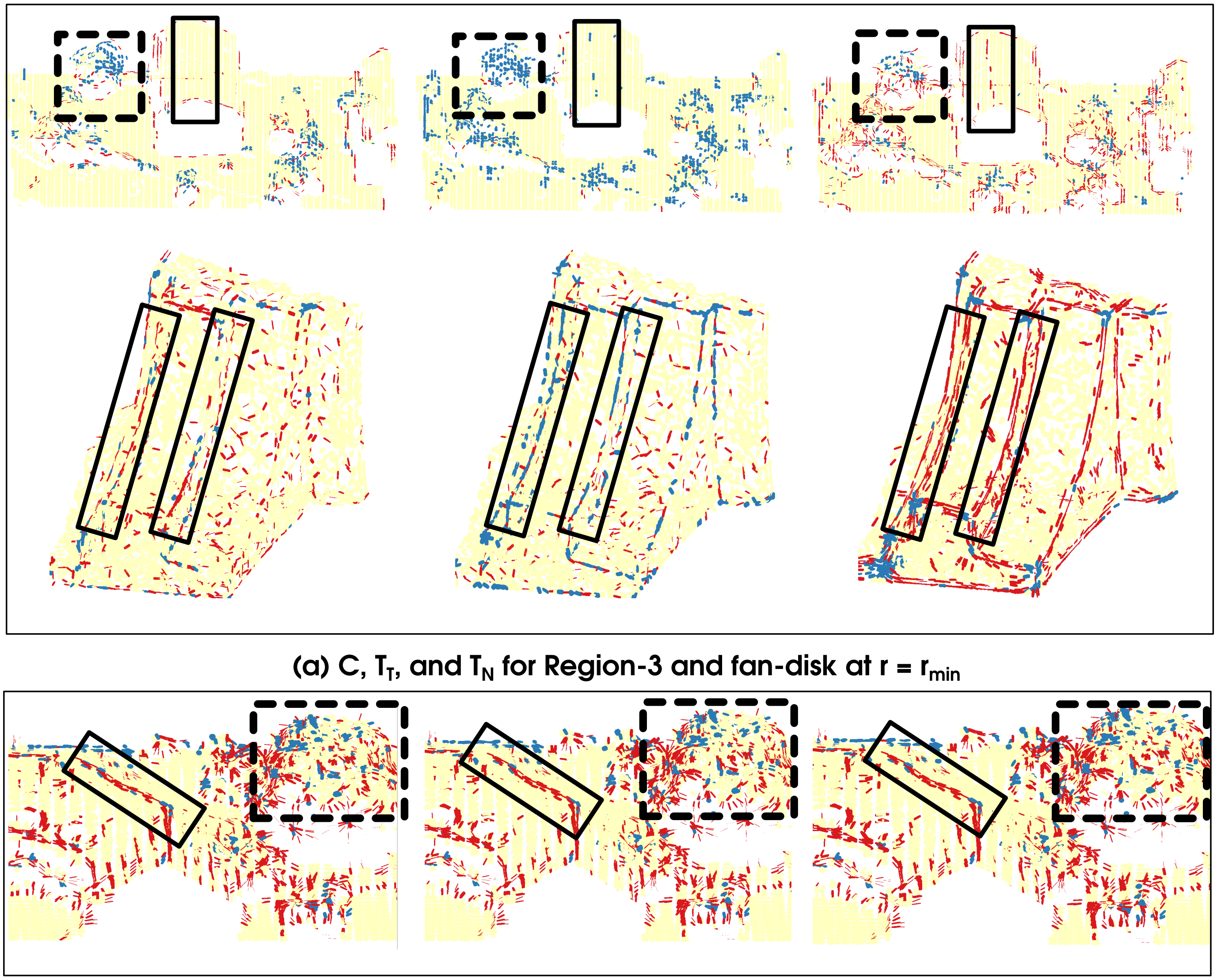
Our proposed (anisotropically diffused LRF-aligned voting tensor) $V^{LRF\text{-}AD}$

Fig. 7: Point rendering of the classification outcomes of LGDs for Regions1-5 (left to right) of Vaihingen benchmark datasets (airborne LiDAR point clouds). Here, images are juxtaposed column-wise, for comparison.

Overall Performance of T_N in Point Classification: We observe that T_N captures sharp features such as edges on smooth-feature, arches and railings in the bridge, stadium field, etc. better than the covariance matrix (Figures 5 and 7). However, T_N does not detect point-type features as well as T_T (and C). The computational complexity of T_N is similar to that of C and T_T. The better performance of T_N in line-feature detection can be explained by the tensor voting approach. V is generated in the normal space which encodes local surface geometry better. V (or V^{LRF}) detects surface-type features well, as V is constructed in the normal space, which captures local surface geometry better. {*Note that all classification outcomes considered in this work use the definition as provided in the case of the covariance matrix, C.*} Upon anisotropic diffusion, these surface-type features are classified as line-type features. Thus, owing to the "persistence" of line-type features in V^{LRF}, the feature detection is more accurate in V^{LRF-AD}, compared to that to C (or C_w or C^{LRF}). However, a noticeable shortcoming of T_N is in detection of degenerate points (junctions/corners), which is not as effective as that by T_T or C_w. Since the point-type features are not preserved during anisotropic diffusion, they tend to be detected as weak line-type features, e.g. the corners (Figure 1) and foliage (Figure 8).

The plot comparing T_N with C (Figure 5) shows that T_N detects more points as line-type features than C. T_N detects a significant percentage of points detected as line-type features exclusively; and a larger percentage of the points detected as line-type by C, is detected by T_N as well. A very small percentage of points are detected as line-type by C exclusively. Except in smooth-feature, T_N detects as many line-type features exclusively, as are detected by both T_N and C. Similar results are observed in the plot comparing T_N with T_T (Figure 6), except that T_T detects fewer line-type features exclusively, compared to C.

Why does V^{LRF-AD} detect line-type features better ? The classification outcomes using V (or V^{LRF}) are different from that of C (or C_w or C^{LRF}) (Figure 1). However, they have certain *patterns* of correspondences, such as surface-type in former are line-type in latter. These patterns are visible from both tensor shapes and saliency maps (Figures 1 and 8) and are better understood using the LRF-aligned descriptors. They are attributed to the reversing of ordering of eigenvalues observed when



(a) C, T_T, and T_N for Region-3 and fan-disk at r = r_{min}

(b) Region-1: C computed at r_{min}, r_{max} and multi-scale aggregate

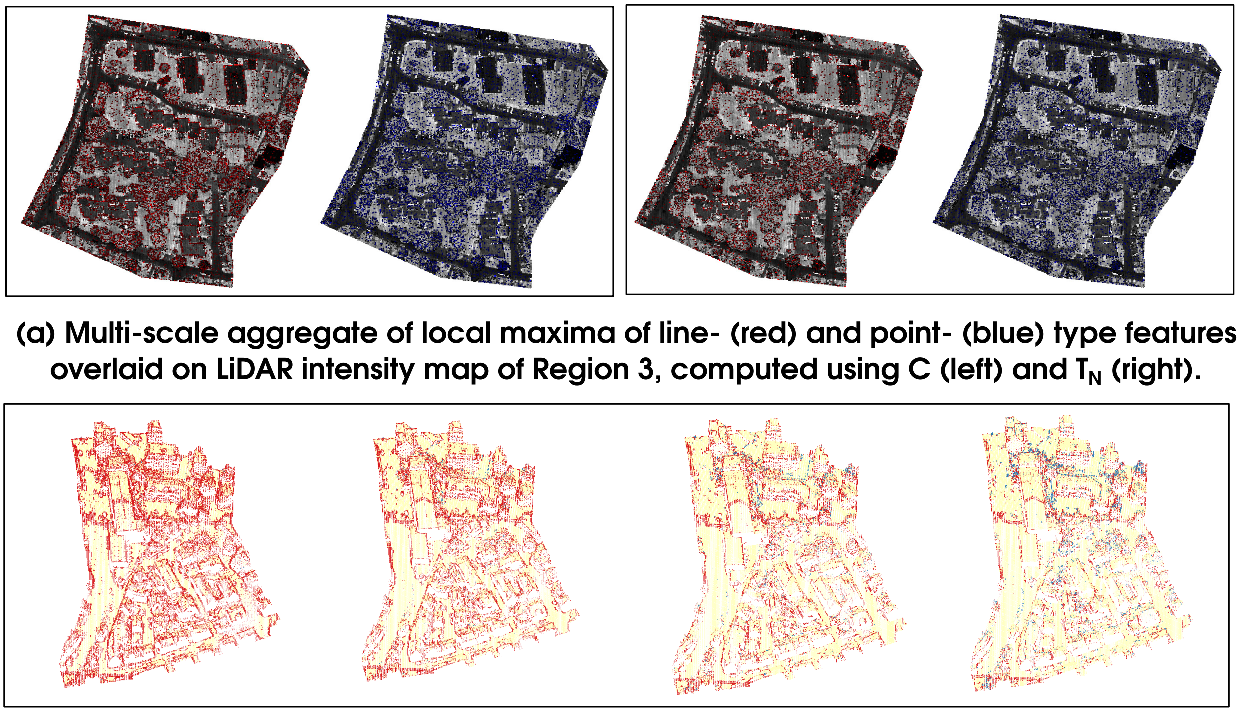
Fig. 8: Superquadric tensor glyph visualization of LGDs of $\mathscr{P}$, for tensor representation at one of the scales. The solid and dashed boxes highlight the sharp features (e.g., gabled roof, edges, etc.), and undefined geometry (e.g. foliage), respectively.

the LRFs of C^{LRF} and V^{LRF} align, i.e. $((\lambda_V^{LRF})_i = 1 - (\lambda_C^{LRF})_{(2-i)}$ for $i = 0,1,2$ given in Equation 6). Thus, a disc-shaped neighborhood (for surface-type features) in V $[(\lambda_V^{LRF})_0 \gg (\lambda_V^{LRF})_1 \approx (\lambda_V^{LRF})_2]$ is equivalent to cylindrical-shaped neighborhood (for line-type features) in C $[(\lambda_C^{LRF})_0 \approx (\lambda_C^{LRF})_1 \gg (\lambda_C^{LRF})_2]$. Similarly, point- and surface-type features in C are captured as point-type features in V. Additionally, the classification outcomes of V "persist" through anisotropic diffusion in our proposed LGD, V^{LRF-AD}.

T_T (and hence, C) and T_N have similar computational complexity as, $T_T + T_N = I_d$ (used in the definition of ball-tensor). The complexity of tensor voting is given as $O(dNl\log N)$ for N points in $\mathscr{P} \in \mathbb{R}^d$ [20]. Space requirements for all second-order LGDs are $O(Nd^2)$. The point-rendering using saliency maps is less computationally expensive than the glyph-based visualization. Hence, we use point-rendering to analyze the dataset globally, and glyphs for local regions-of-interest. The latter is specifically useful in finding misclassifications in a local region.

Juxtaposed Views:

Juxtaposed views of the tensor glyph visualizations show the combined differences in tensor shapes and the saliency maps, across the instances; e.g., the line-type features on sharp edges in gabled roof and fan-disk (Figure 8(a)) clearly are more cylindrical in T_N compared to that of T_T and C. While comparing C and T_N, we need to consider the orientation of the glyphs, in addition to the color and size, since they are not LRF-aligned. However, for comparison of LRF-aligned C^{LRF} and $V^{LRF} - ad$, we can ignore comparisons of glyph orientation. Glyph rendering across different scales as well as using multi-scale aggregate (by averaging unit eigenvectors and normalized eigenvalues across scales) do not show any perceivable



(a) Multi-scale aggregate of local maxima of line- (red) and point- (blue) type features overlaid on LiDAR intensity map of Region 3, computed using C (left) and T_N (right).

(b) Increasing value of diffusion value, at 0.08, 0.12, 0.16, and 0.20 (from left to right)

Fig. 9: (Top) Visualization of multi-scale aggregate of local maxima of the saliency map for line-type (red) and point-type (blue) features. The likelihood of a point to be local maxima of line-type saliency is used as saturation of red, and similarly, that of point-type saliency, as saturation of blue. (Bottom) Effects of change in diffusion parameter δ when computing T_N for Region-1. Default value for all other experiments is $\delta = 0.16$ [30]. Boxes show sharp features, such as gabled roof.

differences in LiDAR datasets (Figure 8(b)). Additionally, glyph rendering and saliency maps help the user to decide some of the user-defined parameters. Our method requires several user-defined parameters, namely, range of scale, number of scales, and diffusion parameter, similar to [10], [30].

Multi-scale Probabilistic Feature Classification: Multi-scale aggregation of sal-iency maps, by averaging, has shown more accurate classification of surface- and line-type features, as the scale-wise differences are implicitly taken into consideration. As can be seen in the plot (Figure 10), the number of points classified in each feature class does not change considerably across scales, for different LGDs (C, T_T, T_N), We chose the fan-disk dataset here as it has considerable distribution of points across the feature classes. We also observe that the surface-type features which are misclassified as line-type features, get correctly classified in higher scales (Figure 8(b)). In the case of LiDAR point clouds, multi-scale aggregation of local maxima of the saliency maps for line- and point-type features from T_N do not give useful information (Figure 9), as is expected in tensor voting.

Anisotropic Diffusion: The change in diffusion parameter influences the outcomes of line-type feature classification in T_N, e.g. gabled roofs in the Vaihingen dataset highlighted as the line-type features (black boxes in Figure 9). At lower values of diffusion velocity parameter, δ, points are over-expressed as line-type features, and increase of δ causes most of those features to increasingly be classified as point-type. Outcomes at higher values of δ appear to be similar to those of T_T.

Weighted Covariance Analysis: T_T (or C_w) captures sharp and point-type features better than C in two different instances (Figures 1, 7, and 8). Firstly, T_T detects sharp features as strong point-type ones; whereas C captures them as weak line-type features, e.g. edges in fan-disk dataset (Figure 8). Capturing sharp features as point-type features by T_T (or C_w) need not be considered inaccurate as these points can be treated as degenerate points owing to being intersection points of multiple planes. Secondly, the points in the foliage of trees in Vaihingen dataset (Figure 8) are captured as point-type features, owing to random orientation of normals, by T_T (or C_w), whereas both C and T_N detect them as a mixture of line- and point-type features. Foliage is considered as a crowd of multiple objects, where T_T (or C_w) resolves the shape of local neighborhood at the point as spherical. However, the use of centroid in C gives inaccurate result in resolving the same, due to the random distribution of the points in the neighborhood. These two observations for T_T (or C_w) are attributed to both modifications; the point-based analysis (as opposed to the centroid-based) of the local neighborhood as well as weighted contributions of neighbors (i.e. using weights inversely proportional to the distances of neighbor to the point).

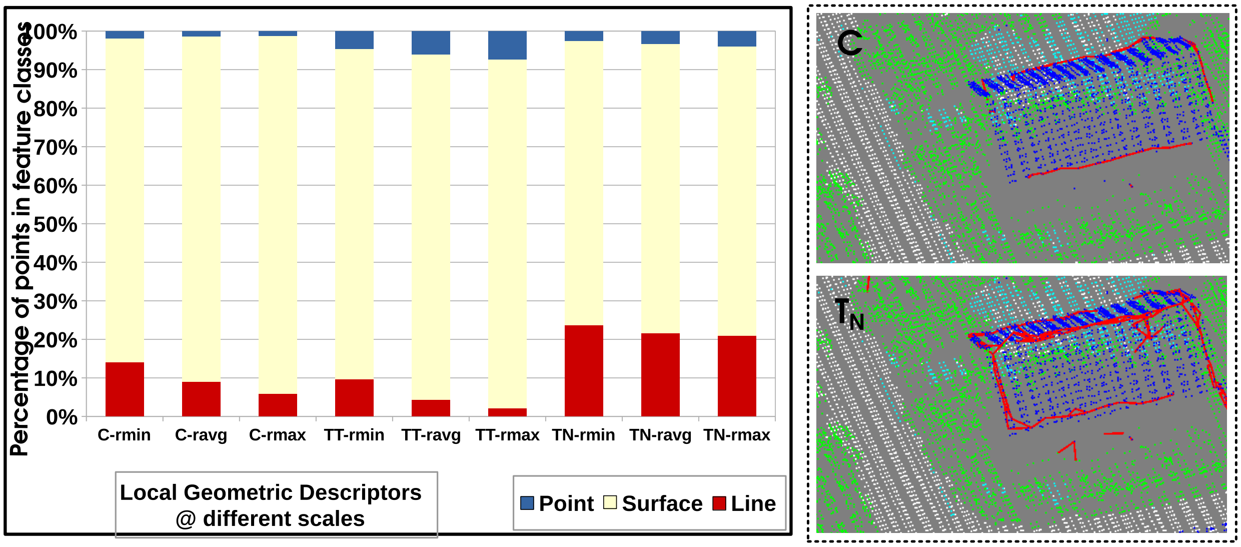


Fig. 10: (Left) The plot shows the variation in percentage of points detected in each feature class using each descriptor, in each scale, for fan-disk. From left-to-right, three columns represent r_{min}, r_{avg}, r_{max}, in that order; for each LGD. Columns 1-3 correspond to C, 4-6 to T_T, and 7-9 to T_N. With minor variations in classification across scales for each LGD, T_N detects more line-type features (red) comparatively. (Right) Point rendering of buildings (blue), foliage (green), road (white) of a region of interest of LiDAR point cloud of Region-4 of Vaihingen dataset, and preliminary results of line features (red) extracted using feature graph in [10]. Note that sharp features in gabled roofs can be observed in LGD T_N, unlike in C.

VII. Conclusions

Here, we have studied LGDs of a point cloud for LiDAR datasets, for geometry-based point classification. We have compared two LGDs with the conventionally used covariance matrix, namely, an anisotropically diffused voting tensor, T_N, and a weighted covariance matrix, T_T. T_N performs better extraction of line-type features in point clouds than the conventionally used C; e.g. the gabled roofs in the Vaihingen dataset are detected (Figure 10). T_T, enhances point-type features, however we reserve its in-depth study as future work. For comparing the classification outcomes accurately, we have introduced variants of these LGDs by aligning their LRFs. Expressing the LGDs as positive semidefinite tensors has enabled us to use tensor field techniques on the LGDs, e.g. visualization for qualitative comparison. We have introduced a probabilistic multi-scale saliency map based on its LGDs for geometric classification. We have used superquadric tensor glyphs for visually representing the LGDs of a point cloud. The visualizations have enabled us to compare and qualitatively analyze the behavior of these LGDs, e.g. shape and orientation.

Our proposed methods have a limitation of user-defined parameters for multi-scale computation and diffusion parameter, as its predecessors [10], [30]. While these parameters give the user the control to bring out the best outcomes upon visual inspection, it would help the user to have a default setting, which can be automated.

Acknowledgements

The authors wish to thank Akshay Jindal for running experiments; Srujana Merugu, Ingrid Hotz, T. K. Srikanth, and Vijay Natarajan, as well as several participants of Dagstuhl seminar 16142 for their helpful discussions; and anonymous reviewers for suggestions on improving the manuscript. This work has been partially funded by NRDMS programme of Dept. of Science and Technology, Government of India. The second co-author has been funded by sponsored projects with EMC^2-RSA India Pvt.; and FRHS, Bangalore, during her graduate study.

References

[1] Joel Daniels, Linh K Ha, Tilo Ochotta, and Claudio T Silva. Robust smooth feature extraction from point clouds. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pages 123–136. IEEE, 2007.

[2] Jérôme Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based Scale Selection in 3D LiDAR Point Clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5):W12, 2011.

[3] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T Silva. Robust moving least-squares fitting with sharp features. In *ACM transactions on graphics (TOG)*, volume 24, pages 544–552. ACM, 2005.

[4] Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics, Eurographics Association.

[5] Adrien Gressin, Clément Mallet, Jérôme Demantké, and Nicolas David. Towards 3D LiDAR Point Cloud Registration Improvement Using Optimal Neighborhood Knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:240–251, 2013.

[6] Stefan Gumhold, Xinlong Wang, and Rob MacLeod. Feature extraction from point clouds. In *Proceedings of 10th international meshing roundtable*, volume 2001. Citeseer, 2001.

[7] Gideon Guy and Gérard Medioni. Inference of Surfaces, 3D Curves, and Junctions from Sparse, Noisy, 3D Data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(11):1265–1277, 1997.

[8] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, 26(2):71–78, July 1992.

[9] Waqas Javed and Niklas Elmqvist. Exploring the design space of composite visualization. In *Visualization Symposium (PacificVis), 2012 IEEE Pacific*, pages 1–8. IEEE, 2012.

[10] 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.

[11] Eunyoung Kim and Gérard Medioni. Urban scene understanding from aerial and ground lidar data. *Machine Vision and Applications*, 22(4):691–703, 2011.

[12] Gordon Kindlmann. Superquadric Tensor Glyphs. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pages 147–154. Eurographics Association, 2004.

[13] Hans Knutsson. Representing local structure using tensors. In *6th Scandinavian Conference on Image Analysis, Oulu, Finland*, pages 244–251. Linköping University Electronic Press, 1989.

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

[15] Leif Kobbelt and Mario Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004.

[16] Beena Kumari and Jaya Sreevalsan-Nair. An interactive visual analytic tool for semantic classification of 3d urban lidar point cloud. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 73. ACM, 2015.

[17] Jean-François Lalonde, Ranjith Unnikrishnan, Nicolas Vandapel, and Martial Hebert. Scale selection for classification of point-sampled 3d surfaces. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 285–292. IEEE, 2005.

[18] Adam Light and Patrick J Bartlein. The end of the rainbow? color schemes for improved data graphics. *Eos*, 85(40):385–391, 2004.

[19] Gérard Medioni, Chi-Keung Tang, and Mi-Suen Lee. Tensor voting: Theory and applications. *Proceedings of RFIA, Paris, France*, 3, 2000.

[20] Philippos Mordohai and Gérard Medioni. Dimensionality Estimation, Manifold Learning and Function Approximation Using Tensor Voting. *The Journal of Machine Learning Research*, 11:411–450, 2010.

[21] Rodrigo Moreno, Luis Pizarro, Bernhard Burgeth, Joachim Weickert, Miguel Angel Garcia, and Domenec Puig. Adaptation of tensor voting to image structure estimation. In *New Developments in the Visualization and Processing of Tensor Fields*, pages 29–50. Springer, 2012.

[22] Min Ki Park, Seung Joo Lee, and Kwan H Lee. Multi-scale Tensor Voting for Feature Extraction from Unstructured Point Clouds. *Graphical Models*, 74(4):197–208, 2012.

[23] Mark Pauly, Richard Keiser, and Markus Gross. Multi-scale Feature Extraction on Point-Sampled Surfaces. *Computer graphics forum*, 22(3):281–289, 2003.

[24] Paul Renteln. *Manifolds, Tensors, and Forms: An Introduction for Mathematicians and Physicists*. Cambridge University Press, 2013.

[25] Franz Rottensteiner. Status and further prospects of object extraction from image and laser data. In *2009 Joint Urban Remote Sensing Event*, pages 1–10. IEEE, 2009.

[26] Thomas Schultz and Gordon L Kindlmann. Superquadric Glyphs for Symmetric Second-order Tensors. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1595–1604, 2010.

[27] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 902–907. IEEE, 1995.

[28] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010*, pages 356–369. Springer, 2010.

[29] Wai-Shun Tong and Chi-Keung Tang. Robust estimation of adaptive tensors of curvature by tensor voting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):434–449, 2005.

[30] 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.

[31] Christopher Weber, Stefanie Hahmann, and Hans Hagen. Methods for feature detection in point clouds. In *OASIcs-OpenAccess Series in Informatics*, volume 19. Schlöss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[32] C-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical Diffusion Measures for MRI from Tensor Basis Analysis. In *5th Scientific Meeting and Exhibition of International Society of Magnetic Resonance in Medicine ISMRM '97*. ISMRM, 1997.

[33] Bisheng Yang and Zhen Dong. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 81:19–30, 2013.

[34] Changgong Zhang, Thomas Schultz, Kai Lawonn, Elmar Eisemann, and Anna Vilanova. Glyph-based comparative visualization for diffusion tensor fields. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):797–806, 2016.