Contour Extraction in Buildings in Airborne LiDAR Point Clouds Using Multi-scale Local Geometric Descriptors and Visual Analytics

Jaya Sreevalsan-Nair, *Senior Member, IEEE,* Akshay Jindal, and Beena Kumari

Abstract

Topographic Light Detection and Ranging (LiDAR) captures geometric information of the topography of a geographical region, often using airborne platforms. The research and practice of analysis of point clouds acquired using LiDAR is more recent in comparison to that of LiDAR imagery. Point clouds are unstructured datasets, where its geometric or structural classification labels the constituent points as belonging to line-, surface- or point-type features. We focus on line-type features in the LiDAR point clouds of urban residential areas, which enables extraction of building outlines. We use a multi-scale local geometric descriptor (LGD), computed using tensor voting and gradient energy tensor to enhance specific line-type features, e.g., gable roofs. Given that LGDs are positive-semidefinite second-order tensors, we propose a tensor-based data analytic workflow for extraction of boundaries in building roofs using the LGD. We use the tensor representation of the LGD to extract "tensorlines," which are then post-processed for extracting feature lines of the building roofs. Our proposed workflow provides the flexibility to the human-in-the-loop for exploration of point clouds for roof boundary tracing for selected buildings. We demonstrate the workflow for a two-plane gable roof.

Index Terms

Structure tensor, Tensor voting, Gradient Energy Tensor, Contour extraction, Tensorlines, LiDAR point clouds.

I. Introduction

TOPOGRAPHIC Light Detection and Ranging (LiDAR) technology has advanced from using imagery to point clouds as its data format for reconstructing geographical regions. Recent literature indicates that there is active interest in investigating what analyses the point clouds can exclusively provide [1]. While the community has access to few interactive visualization tools for LiDAR point clouds, which provides end-to-end process workflows, there still remains a gap in tools which can be used for computational analysis of the data. We attribute the gap to the absence of a formal, but generic, data analytic workflow definition for LiDAR point clouds.

Our motivation is to demonstrate a data analytic workflow, leading to an interactive tool, where scientists and researchers can use computational models to explore the datasets. At the same time, our work focuses on maximizing the use of data structures already being used in LiDAR point cloud processing. For instance, the covariance matrix provides local geometric information from these point clouds acquired from airborne LiDAR, and is predominantly used for object-based or semantic classification [2]. We have demonstrated that multiple local geometric descriptors (LGDs), which can be represented as positive-semidefinite symmetric second-order tensors, can serve the purpose of inferring local geometry [3]. Thus, LGDs can be treated as data elements, which encode the information about the shape of local neighborhood of a point. Here, we propose exploiting the tensor analysis of these data elements in applications of airborne LiDAR point clouds, wherein, we focus on contour (or line segment) extraction of roofs. Even though the LGDs give structural (or geometric) classification, they are conventionally not used beyond providing features for semantic classification of point clouds. Covariance matrices, which are computed point-wise, often provide the feature vectors or parameters in various classification or clustering techniques, e.g. supervised learning techniques [4], [5], [6], [7], unsupervised techniques [8], etc.

Boundaries of roofs are traced as a first step towards building reconstruction, which in itself is a significant step in three-dimensional (3D) reconstruction of LiDAR point clouds [9]. Our hypothesis is that the LGDs can be the *central actors* for providing the roof line boundaries. We test this hypothesis by devising a set of techniques that use LGDs and its visualizations for building roof reconstruction. Our goal is to define a tensor-based data analytic workflow for geometric processing of LiDAR point clouds, in addition to its role in its semantic classification. Here, we aim to use the second-order tensor properties of LGDs for 3D boundary detection of building roofs. The novelty here is that we use eigenvectors for line segment extraction, while only eigenvalues of these tensors are widely used for LiDAR point cloud processing, specifically in semantic classification.

There is variability in the performance of LGDs owing to the influence of the local geometry and its variations. Hence, we use visual analytics for decision making in different processes in the workflow that use LGDs for contour extraction of roofs. Thus, our proposed work is semi-automated, with the requirement of a *human-in-the-loop*. In the area of visual analytics, visualizations *steer* analytics in a data analytic workflow. We have previously used visualizations of features of the

This document is a preprint, as on May 16, 2018. The authors are with Graphics-Visualization-Computing Lab (GVCL), International Institute of Information Technology, Bangalore, 26/C Electronics City, Hosur Road, Bangalore 560100, India. *Corresponding author: Jaya Sreevalsan-Nair.* e-mail: jnair@iiitb.ac.in | {akshay.jindal,beena.kumari}@iiitb.org

This work was supported by the NRDMS programme and Early Career Research Award from Science and Engineering Research Board, both awards from the Department of Science and Technology, Government of India. The work of B. Kumari on this work has been funded partially through sponsored projects with EMC^2-RSA India Pvt., Bangalore; and FRHS, Bangalore.

points, e.g. eigenvalue-based features of the LGDs, such as anisotropy and sphericity, for guiding object-based classification of LiDAR point cloud [8]. Since the methods are semi-automated and may not be scalable for building extraction in large point clouds [10], [11], [12], we see the value of our work in the design of interactive tools. Such tools will be useful for either quick exploration of new datasets, or identifying training datasets for supervised learning techniques for contour extraction [13].

In earlier work, analytic processes with the point cloud, such as point segmentation and classification, were performed together. The current data processing workflow of 3D LiDAR point clouds consists of both feature classification and extraction. In literature, "feature" refers to different levels of granularity of object description. Features imply classification parameters [2], the objects (e.g. buildings, foliage, etc.), or geometric entities (e.g. feature lines, edges, etc.) [14]. Weinmann et al. [2] have run extensive experiments on the logistics of object-based classification of LiDAR point clouds. They have recommended the combined use of random forest classifier, optimal size of local neighborhood, and selected feature set based on Correlation-based Feature Selection method [15] for improving the outcomes of the object-based classification of airborne LiDAR point clouds. This result, which is generalized across benchmarks and without any heuristics, emphasizes the growing popularity of supervised learning based classification of LiDAR point clouds, owing to its inherent uncertainties. These uncertainties, due to the variety in the nature and density of objects present in natural environments, make these point clouds different from the kinds encountered in computer graphics [16]. Thus, we decouple the 3D reconstruction from classification, where the decoupling allows us to pose our problem as a 3D reconstruction of labeled point clouds. In this work, we focus exclusively on points labeled as buildings.

Overall, our goal is to propose techniques which use LGDs and visualization for enabling a user to interactively determine and extract shapes of roofs. The visualizations enable decision making for *steering* computations for LiDAR point cloud processing by the human-in-the-loop. Our proposed work is useful for exploratory data analysis, which is an alternative to completely automated methods. Our first step in this approach is in generating wireframe models, as a starting point [13], which we validate using a preliminary surface/plane fitting in Section V. Here, our contributions support a tensor-based data analytic workflow for visual exploration of labeled airborne LiDAR point clouds and extraction of line segments of the building roofs. Our novel methods are:

- The use of visualizations of eigenvalue features of LGDs and other geometric features, for extracting an intermediate data structure, which in our case is a triangulated irregular network (TIN) of a relevant subset of the points,
- A weighted function of eigenvectors of LGDs and other geometric properties for propagating edges in TIN for line segment extraction,
- A post-processing step based on roof topology graph for improving the outcomes of line extractions to give convex and planar roof panels, which we demonstrate for a simple gable roof with two planar panels.

We demonstrate our proposed method on the benchmark dataset of the Vaihingen site, published by the International Society for Photogrammetry and Remote Sensing (ISPRS)[1].

Frequently used notations: Light Detection and Ranging (LiDAR), Local Geometric Descriptor (LGD), Triangulated Irregular Network (TIN), Gradient Energy Tensor (GET).

II. Related Work

There has been active research in the area of contour extraction for buildings from LiDAR point clouds. It is largely posed as a point segmentation problem, where the buildings are identified and isolated using region growing algorithms. Our method belongs to the class of edge-based segmentation methods [17], even though we do not perform an explicit point segmentation. Edge-based methods are good for faster segmentation of points to coherent regions, however they are sensitive to noise. There have been works similar to ours where eigenvectors are used for propagating the contours [16], [18], [19]. Several papers on reconstruction models for entire buildings capture roof reconstruction as a geometric problem [20], [21], [22] where they compute planar equations of point segments on the roof. Several papers also use Random Sample Consensus (RANSAC)-based methods for model estimation of the buildings and its roofs [23], [24], [25].

Roof Shape Extraction: Haala and Kada [21] have reviewed several works on 3D building reconstruction, where they are broadly classified into roof shape and building facade extraction methods. Our work falls in the former class, and specifically in the category of polyhedral roof reconstruction based on segmentation, as opposed to methods using primitive shapes or Digital Surface Model (DSM) simplification. In our approach, the segmentation is performed implicitly by the user, guided by visualizations. Vosselman et al. [26] have used 3D Hough transform for 3D plane detection and reconstruction. Verma et al. [27] have proposed building detection and modeling from aerial LiDAR data by using a topology graph to estimate simple parametric shapes of the building roofs. We propose the use of topology graph in our post-processing step. Shan et al. [22] have distinguished between planar and non-planar points using eigenvalue analysis of local neighborhood of points using the normalized minor eigenvalue of covariance matrix. They have used k-means clustering in the feature space to separate point clusters between roof segments, and further processing to separate the roof segments for reconstruction. The shortcoming of this method is in identifying the number of clusters a priori for the clustering process. Sohn et al. [28] have used binary space partitioning and height clustering which works well for extracting flat roofs with complex shapes, as well as for multi-plane roofs, with post validation for the latter.

More recently, Lin et al. [11] have used multi-view images in addition to point clouds, to find two-dimensional (2D) line-support regions and then construct their proposed line-segment half plane (LSHP) data structures from the line-support regions.

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

The LSHP structures consists of 3D tangential rectangles. However, this approach requires multi-view images. Lin et al. [12] have proposed line segment extraction from the point clouds using the geometric properties of smaller planar surfaces, called facets, which are extracted using alpha-shapes from computational geometric concepts. In our method, we start with existing LGDs, without having to compute additional geometric data structures. Xia and Wang [29] have proposed a fast algorithm for edge extraction using eigenvalues of the gradient tensor of the height function for edge detection, and minimization of path deviation angles for the extraction. We have encapsulated the gradient tensor of height function, which exploits the 2.5D nature of LiDAR point clouds, in our LGD using gradient energy tensor. Similarly our proposed weighting function for identifying points which belong to edges we use the path deviation as one of the components.

Hackel et al. [13] have performed contour detection in unstructured 3D point clouds using supervised learning. We attest to their statement on the need for computing line-type features before finding surface patches, as being lower-level geometric constructs/primitives, the former can guide point cloud segmentation, which gives the latter. In their two-step contour extraction algorithm, they have first predicted the likelihood of a point being on a contour and in the second step, they have used points with higher likelihood to be seed points from which an over-complete contour graph is derived. Then, finally, an optimal subgraph is determined to be the contour. Our structural classification is similar to their first step in outcome, except their method uses supervised learning. The second step is in connecting these points to extract lines. Similar to their work, we reuse the parameters used for semantic labeling [30] and confine ourselves to purely position information of the points, which is universally present in all LiDAR point cloud datasets. They have used local non-maxima suppression for candidate generation, which is similar to our method, except that the candidates are confined to a local triangulation, in our case.

Computer Graphics and Geometry Processing: In computer graphics and geometry processing, extraction of sharp features is an important problem [31], which is of relevance here, since gable roof lines are considered to be sharp features. Their application on (structurally/geometrically) well-defined computer graphics models (e.g. fan-disk, smooth feature, etc.) differs from unorganized point clouds of natural environment scans, since the former works on basic surface features, which is of interest in graphics-based approaches [13]. Weber et al. [31] have proposed a two-step process, where in the first step, Gauss map clustering is used to determine if a point is on a sharp feature or not, and in the second, moving least squares (MLS) method is used for computing the surface reconstruction, considering the points already "tagged" as belonging to sharp features. In our work, we use our LGDs for finding points that belong to the line-segments of the roof, and it is possible for us to implement the MLS method for reconstructing the building. However, in this work, we explore the use of LGDs for "tagging" the points in point cloud as points on building outline, which is equivalent to finding sharp feature points. With the use of multi-scale difference of normals (DoNs) [32] at each point, we are implicitly using the key concept used in [31]. While they have used global triangulation, we use a localized one.

Keller et al. have implemented a line feature extraction approach using eigenvalue-based features of local geometric descriptors have been applied to LiDAR point clouds [16]. Here, we investigate feature line extraction algorithm using tensor voting based LGDs. Our contribution is in proposing alternative approaches for extracting feature lines using LGDs. Different from [16], we use a TIN of selected points. We then compare two approaches for extracting feature lines using local geometric descriptors. We propagate a tensorline [33] on the TIN, based on specific criteria of 2D and 3D local shape and geometric features used for object-based classification [2].

III. Background

Here, we describe the LGD [34] based on tensor voting [3] and gradient energy tensor (GET) [35]. This descriptor identifies sharp features better than simpler LGDs, such as, the covariance tensor[2] or the one based on tensor voting. We explain the use of this LGD in our methodology further, in Section IV. We use a multi-scale approach for computing these LGDs [3], [16], which gives a probabilistic geometric classification of the point cloud. For our proposed point processing methods, we require eigenvalue-based features and other structural features of the LGD. We choose to use this LGD for our experiments, as opposed to the conventionally used covariance tensor, owing to its comparatively better performance in identification of points on gable-roof line as stronger line-type features [3]. That said, our larger goal is to design interactive tools which can use any LGD, represented as positive-semidefinite second-order tensors, as the *central actors*. Here, we show a proof-of-concept using the LGD proposed in [34].

A. Tensor Voting-based Local Geometric Descriptor

Conventionally, the covariance tensor in a local neighborhood of a point is used as a LGD of the point in the LiDAR community [4]. The LGD has been used for structural classification of the point cloud [16], which labels each point as to belonging to line-, surface-, or (critical) point-type features. This is determined by the shape of the local neighborhood, which is either cylindrical, disc-like, or spherical, respectively. This shape is determined by the eigenvalues of the LGD. In [3], a tensor voting-based LGD, upon application of anisotropic diffusion [36], has shown to enhance the line-type features better. This LGD identifies the points on roof boundary and gable-roof lines as line-type features. Tensor voting is initialized by assigning an unoriented ball tensor at point x, given as:

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

[2]Since *covariance matrix* is a positive-semidefinite second order tensor, we refer it to as *covariance tensor* hereafter.

where $t(y) = (y-x)$, $\mu_y = exp\Big(-\frac{\|t(y)\|_2^2}{\sigma^2}\Big)$; I_d is identity matrix of size d, which is the dimensionality of the dataset, i.e., 3, for point clouds. Anisotropic diffusion is then applied to the tensor, which slows down the diffusion process across a sharp feature/edge and makes the diffusion along the feature faster [36]. Thus, the tensor voting-based LGD is given as:

$$T_{3DVT} = \sum_{i=0}^{2} exp\Big(-\frac{\lambda_i}{\delta}\Big) * e_i e_i^T, \tag{2}$$

where eigenvalues λ_i, correspond to eigenvector e_i of $V(x)$, for $i = 0,1,2$, and diffusion parameter is δ, with default value of 0.16 [3], [36]. We follow the convention: $\lambda_0 \geq \lambda_1 \geq \lambda_2$. We observe that the covariance tensor and the tensor voting-based LGD are tensor products of tangent and normal vectors, respectively. Thus, these LGDs are positive-semidefinite symmetric second-order tensors.

B. Gradient Energy Tensor

We observe that "structure tensor" is an *overloaded* (using the object oriented programming parlance) term across different research communities. The covariance tensor is called structure tensor in LiDAR community [18], [37]. However, in the image processing and computer vision communities, structure tensor refers to the second-moment matrix in images, derived from gradients [38], [39]. While the LGDs discussed in Section III-A capture the first-order differences, higher order gradients are not captured. Sreevalsan-Nair and Jindal [34] have demonstrated that the use of higher order gradients can improve the structural classification. In image processing applications, the higher order gradients have been used, in the form of gradient energy tensor (GET) [35]. The GET is a 2D second-order tensor that has been used as an improvement of the energy tensor in finding points of interest (PoI) in the image [40].

In point clouds, modifying the structure tensor, T_S, using GET has shown improvement in its performance in detecting line-type as well as (critical) point-type features in airborne LiDAR point clouds [34]. GET, $T_{GET}(f)$ is computed using: (a) the 2D gradient of the function, f, defining the image, given by $\nabla f = (\frac{\mathrm{d}f}{\mathrm{d}x}, \frac{\mathrm{d}f}{\mathrm{d}y}) = (f_x, f_y)$, (b) the gradient of the Laplacian of f, given by $\mathscr{T} f = \nabla(\nabla^T \nabla f)$, and (c) the Hessian of the function, given by $\mathscr{H} f = \nabla\nabla^T f$.

$$T_{GET}(f) = (\mathscr{H} f)(\mathscr{H} f)^T - 0.5\big((\nabla f)(\mathscr{T} f)^T + (\mathscr{T} f)(\nabla f)^T\big). \tag{3}$$

The 3D LiDAR point cloud can be considered as the height map of the 2D data, i.e. the (longitude, latitude) data; and hence is 2.5D. Thus, $T_{GET}(z)$, where z is the height function, has been used to find points of interest in the LiDAR point cloud. The GET is a symmetric second-order tensor, similar to the LGDs. However, different from the LGDs, the GET is not positive-semidefinite at all points. Nonetheless, it has been observed that the GET is positive-semidefinite at the PoIs, and hence the cases where the tensor has negative eigenvalues, they can be truncated to zero [41]. Thus, we constraint the GET to be a positive-semidefinite second-order tensor.

The integration of tensor voting and gradient-based methods has shown improvement in structural classification in [34]. It has been observed that T_{GET} identifies the (critical) point-type features better than T_{3DVT}. Hence, T_{GET} is used to re-label the structural classification we obtain using the LGD, T_{3DVT}. In general, in object-based classification, trees (dense foliage) have been found to be misclassified as buildings [4]. T_{GET} detects the foliage points as critical points, which we use for correcting points identified as line-type features by T_{3DVT} [34].

Structural Classification of 2D GET: Just as 3D LGDs, such as covariance tensor and T_{3DVT}, give three structural classes, the 2D GET gives two, namely, line-type features and critical points, which are referred to as points of interest (PoI) in [35], [40] [3]. For T_{GET}, we compute the following multi-scale saliency values, using eigenvalues computed at each scale, $\lambda_0(x,r) \geq \lambda_1(x,r)$:

$$\begin{aligned} \mathbf{L_{l,s,p}(x)} &= \frac{1}{N_r} \cdot \sum_{\mathbf{i=1}}^{\mathbf{N_r}} \mathbf{c_{l,s,p}(x, r_i)}; \qquad (4)\\ \text{where } c_l(x,r) &= \frac{\lambda_0(x,r) - \lambda_1(x,r)}{\lambda_0(x,r) + \lambda_1(x,r)};\\ c_s(x,r) &= 0; \text{ and}\\ c_p(x,r) &= \frac{2 * \lambda_1(x,r)}{\lambda_0(x,r) + \lambda_1(x,r)}. \end{aligned}$$

We observe that T_{3DVT} performs the detection of line-type features better than the covariance tensor, and the detection of (critical) point-type features worse than the covariance tensor [3]. Since T_{GET} uses higher order derivatives, which captures a more realistic nonlinear fitting model of the local neighborhood, we give higher *confidence* to the critical points identified by T_{GET}. Thus, at points identified as critical points by T_{GET}, but not by T_{3DVT}, we replace the LGD at those points using an *aggregated* one, computed as a combination of T_{3DVT} and T_{GET}. This specifically enables the points in the foliage/vegetation (object) class to be re-labeled as (critical) point-type features, which is correct as per their isotropic behavior.

Aggregating 3D Tensor Voting-based LGD with 2D GET: Here, we elaborate on the computation of an aggregated LGD at each point by combining tensor voting and GET, which is not explained in [34]. We mark the points x_m, which are classified

[3]The POI is different from our usage of "point-of-interest" in this work. Hereafter, we refer to the points we identify for local triangulation as "points-of-interest" and the POIs identified by GET as critical points.

as critical points by T_{GET}, but are not classified the same by T_{3DVT}, to be *re-labeled*, and update the LGD using an aggregated LGD. For the aggregation, we use our observation of the correspondences between point classification given by the saliency maps of T_{3DVT} and T_{GET}. We find two visual correspondences: (a) the line-type features identified by the GET correspond to critical point-type features identified by T_{3DVT}, and (b) the critical point-type feature of T_{GET} correspond to line-type features of T_{3DVT} (Fig. 1).

Our proposed aggregation entails re-assigning the saliency map of T_{3DVT} at the point to be re-labeled using the saliency map of T_{GET}. Thus, $(L_l, L_s, L_p)_{3DVT}$ is given as $(0, L_p, L_l)_{GET}$. Solving for eigenvalues of T_{3DVT}, we get $(\lambda_0 = \lambda_1)_{3DVT}$, since $L_l = 0$. We then use the expression for $(L_s/L_p)_{3DVT} = (L_p/L_l)_{GET}$, which gives us the ratio, $\gamma = (\lambda_1/\lambda_2)_{3DVT} = 1 + 1.5 * (L_p/L_l)_{GET}$. We reuse the minor eigenvalue λ_2 and all the eigenvectors of T_{3DVT}, as only the saliency maps of T_{GET} need to be integrated and the minor eigenvalue will not make significant change to the tensor. Thus the updated LGD, $T_{3DVT-GET}$, is given as:

$$T_{3DVT-GET}(x_m) = \lambda_2 * \left(\left(1 + \left(\frac{3L_p}{2L_l} \right)_{GET} \right) * \lambda_2 * (e_0 e_0^T + e_1 e_1^T) + e_2 e_2^T \right) \tag{5}$$

$$T_{3DVT-GET}(x_c) = T_{3DVT}(x_c) \tag{6}$$

for a point to be re-labeled, x_m, and a point that need not be re-labeled, x_c; where $\{e_0, e_1, e_2\}$ are major, middle, and minor eigenvectors of multi-scale aggregated T_{3DVT}, respectively, at the point.

Post-processing: We find that the re-labeling using T_{GET} does not conform to the spatial locality of the point-type features. Hence, after re-labeling using T_{GET}, we implement a *simplified region-growing* step to preserve the spatial locality of class types. In the post-processing step, a point is re-labeled as a point-type feature if a majority of points in its local neighborhood belongs to (critical) point-type features. Thus, using such a voting scheme, where a vote implies a neighboring point belongs to (critical) point-type feature, $T_{3DVT-GET}$ of such a point is updated if the number of votes is higher than half the size of its local neighborhood. If updated, the $T_{3DVT-GET}$ is replaced by the average of the $T_{3DVT-GET}$ of all the critical points in the local neighborhood. The use of GET, re-labeling and region growing steps improve visual identification of foliage by $T_{3DVT-GET}$, as shown in Fig. 1.

C. Multi-scale LGD

In certain cases, multi-scale LGDs are preferred over optimal or adaptive scales to capture shape distributions across scales [3]. We use the multi-scale aggregated tensor, as computed in [3], as the tensor representation has to be preserved downstream in the contour extraction workflow for us. The multi-scale aggregated d-dimensional second-order tensor is given by:

$$T_{MS} = \sum_{r=1}^{N_r} \frac{1}{\sum\limits_{i=0}^{d-1} \lambda_i(r)} \cdot \sum_{i=0}^{d-1} \lambda_i(r) \hat{e}_i(r) \hat{e}_i(r)^T \tag{7}$$

where N_r scales, given that the eigenvalues and corresponding eigenvectors of the LGD at scale, r, are λ_i and e_i for $0 \leq i \leq d-1$. The computation of multi-scale saliency values, (c_l, c_s, c_p), which give the likelihoods of a point belonging to the line-, surface, and (critical) point-type features, respectively, is explained in [3]. For eigenvalues of LGD at point x at scale r, $\lambda_0(x,r) \geq \lambda_1(x,r) \geq \lambda_2(x,r)$, the saliency values are given as:

$$\begin{aligned} \mathbf{L_{l,s,p}(x)} &= \frac{1}{N_r} \cdot \sum_{\mathbf{i=1}}^{\mathbf{N_r}} \mathbf{c_{l,s,p}(x, r_i)}; \\ \text{where } c_l(x,r) &= \frac{\lambda_0(x,r) - \lambda_1(x,r)}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}; \\ c_s(x,r) &= \frac{2 * (\lambda_1(x,r) - \lambda_2(x,r))}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}; \text{ and} \\ c_p(x,r) &= \frac{3 * \lambda_2(x,r)}{\lambda_0(x,r) + \lambda_1(x,r) + \lambda_2(x,r)}. \end{aligned} \tag{8}$$

We use the multi-scale expression in Equation 7 for computing both the multi-scale aggregated LGD as well as multi-scale aggregated GET, using the corresponding tensors. The multi-scale aggregated LGD computed using Equation 7 has saliency values the same as those given in Equation 8, by design [3]. Hereafter, our annotations for the 3D anisotropically diffused tensor voting based LGD, T_{3DVT}, and the 2D GET, T_{GET}, imply corresponding multi-scale versions of the tensor. T_{3DVT} is computed using Equations 2 and 7, and T_{GET} using Equations 3 (for height function z) and 7. Points are labelled as per the maximum saliency value, i.e. if for saliency value (L_l, L_s, L_p), the maximum saliency value is contributed by L_l, then the point is classified as line-type feature. Thus, we use a *probabilistic point classification* [3][4].

[4]Throughout the paper, images of saliency maps are color-coded as (R, G, B) corresponding to (L_l, L_s, L_p). Dominance of a color channel indicates higher probability of the corresponding saliency map.

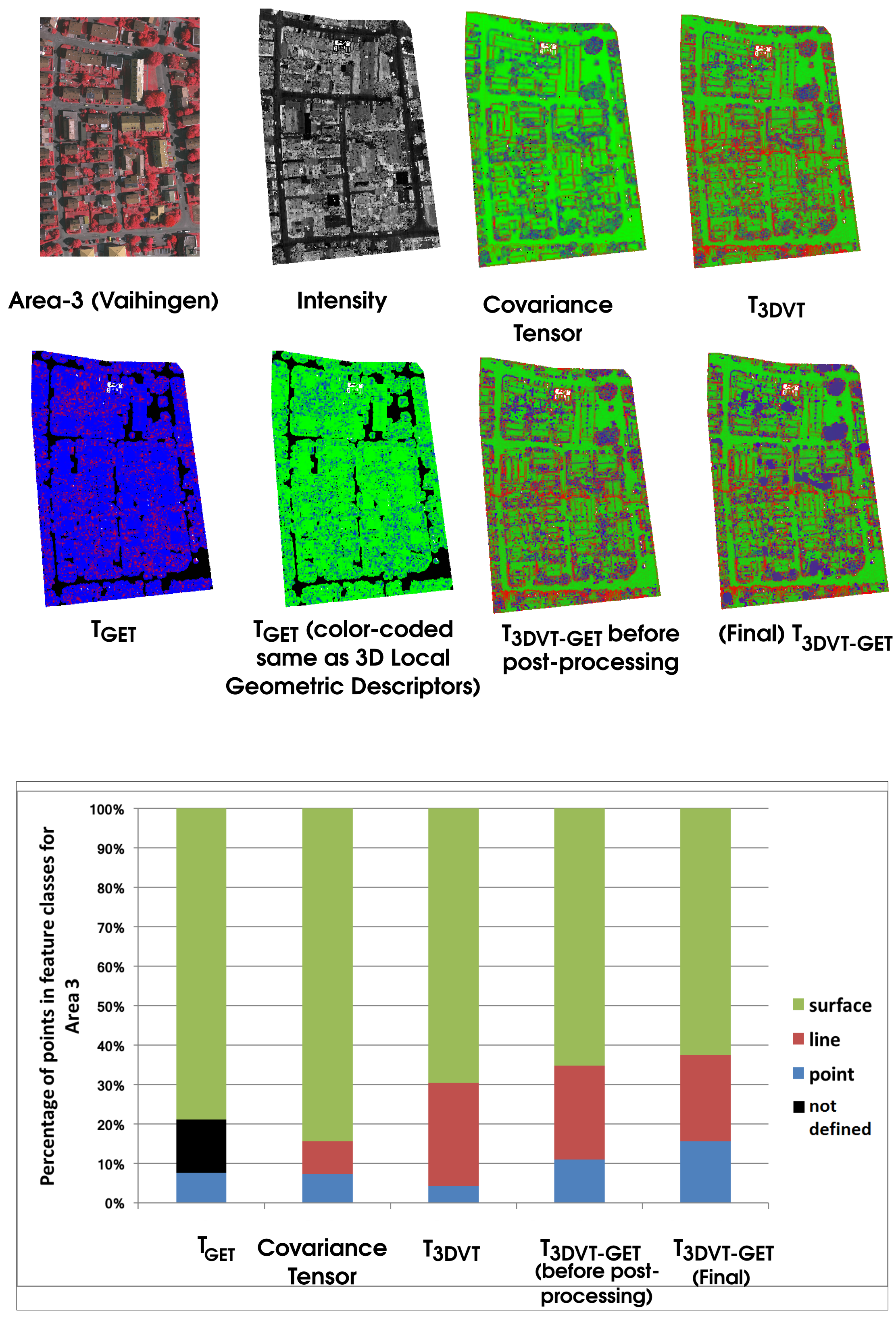


Fig. 1: In the ISPRS benchmark dataset, Area-3 of Vaihingen site (323,896 points), we show the colormaps of intensity and saliency maps computed from different local geometric descriptors. The saliency maps show (red, green, blue) for (line, surface, point) type features, respectively. The plot shows distribution of structural classes obtained from different descriptors. The descriptor, $T_{3DVT-GET}$, is a post-processed aggregation of tensor voting with anisotropic diffusion, T_{3DVT}, and Gradient Energy Tensor, T_{GET} [34]. $T_{3DVT-GET}$ and T_{3DVT} perform better than the conventionally used covariance tensor in detecting the line-type features on gable roofs. $T_{3DVT-GET}$ differentiates foliage from building in the structural classification, in the form of (critical) point-type features. $T_{3DVT-GET}$ and T_{3DVT} identify 13,622 and 44,101 points, respectively, as critical points.

D. Feature Space for Line-type Features

For each point in the point cloud, we define a feature space or feature vector, which is conventionally used for object-based or semantic classification [7], [42]. Generally, the features include height (z), the eigenvalue features from the LGDs, and Difference of Normals (DoN), Δ_n [32]. We use the likelihood of a point belonging to a line-type feature, L_l, as computed in the multi-scale saliency maps [3]. $\Delta_n$at a point gives a strong indication if a point lies on the intersection of two planar segments, e.g. points on gable roofs. Thus, we use these three features, namely, z, L_l, and Δ_n, for our proposed approach for contour extraction. Computation of Δ_n incorporates multi-scale approach, where minimum and maximum scales, r_{min} and r_{max} are used. Δ_n at point x, with unit vector of the surface normal estimate $\hat{n}(x,r)$ at x at scale r, is given as:

$$\Delta_n(x) = 0.5(\hat{n}(x,r_{max}) - \hat{n}(x,r_{min})) \tag{9}$$

We use the normals of locally fitted planes in the neighborhoods to keep the computation independent of LGDs. We can alternatively use the minor eigenvector of the LGD as normal, after orienting it correctly for all points, i.e., by using the appropriate sign for the vector at each point [32].

We compute all features using the multi-scale approach, given in Equation 7. In the case of Δ_n, we use the magnitude of Δ_n, i.e. $\|\Delta_n\|$, as one of the parameters. In our proposed point processing methods, we use the saliency map, L_l, $\|\Delta_n\|$, and height z (Fig. 2).

IV. Our Proposed Method

In this work, we focus on a workflow for extracting contours which are line segments on roofs. Our methods can be potentially extended to other objects such as fences, powerlines, etc. This is a precursor to a complete 3D reconstruction, which is our future work. Here, we use an intermediate spatial data structure, namely a TIN (same as a triangle mesh), on which we localize and apply our proposed tensorline-based contour extraction. Unlike triangulating all points in the building class, we use a local strategy for identifying a subset of points for generating the TIN. Tensorlines are contours extracted using eigenvectors of second-order tensors [33]. The time complexity of Delaunay triangulation is $O(n \log n)$, for n vertices, which are points in the point cloud. Owing to our requirement for an interactive visual application, we reduce the number of points, n, used for the triangulation. We further use visualization of features, namely L_l and Δ_n, to determine the *reduced* subset of points, called *points-of-interest*, used for TIN generation.

Our proposed three-step method includes TIN generation, tensorline extraction, and post-processing of the tensorlines to extract line segments in the roof. For tensorline extraction, we identify the *seed points* of the TIN, from which we propagate the lines. Our proposed tensorline is a sequence of connected edges of the TIN, starting from a seed point. For identifying subsequent edges for propagating the tensorline, we propose a novel weighted function for *scoring* potential candidates. Our post-processing step includes identifying tensorlines belonging to each line segment in the roof, fitting a line to the line segments, correcting the lines to confirm to the roof topology graph, and finally correcting the nodes on the roof topology graph for conforming to planarity of the roof panels. Here, we demonstrate a proof of concept for a simple gabled roof with two planes.

A. Local Triangulation

We observe that compared to the conventional covariance tensor, the LGD $T_{3DVT-GET}$ detects more line-type features, by additionally strengthening the ones for gabled roofs. This is significant, as these line-type features are used for contour extraction. In conventional contour extraction step, several of the line-type feature points get pruned, e.g. in the feature graph generation in [16]. Thus, the higher point density in line-type features using $T_{3DVT-GET}$ ensures that points in line-type features do not lose the sharp (or line-type) feature during pruning, thus ensuring enough number of points for propagating tensorlines.

Tensorlines [33] are streamlines which trace a family of eigenvectors of the LGDs, where *family* implies the i^{th} eigenvector at all points, for $0 \leq i \leq d$, for spatial dimensionality of d. In our case, we use tensorlines computed from major eigenvector (i.e., for $i = 0$). Streamlines in vector fields are the loci of points where the velocity vector is tangential to the streamline at the point. Unlike the unidirectional vectors in vector field, eigenvectors in tensor field are bidirectional. Hence, tensorlines use the principle of minimum deviation when propagating in the direction of the eigenvector. In principle, the feature graph in [16] is a set of connected tensorlines. Different from the feature graph, we propose to find tensorlines from a set of seed points, which maximally covers the desired line-type features on the roof, without the constraint of the tensorlines themselves being *connected*. However, in order to reduce the extraction of irrelevant tensorlines, we impose a constraint of the region in which the tensorlines can be propagated. We propose this region to be a *ribbon* or *band* of line-type features, such that the tensorline construction becomes equivalent to *skeletonization* of the ribbon. Thus, we perform a *local triangulation* so that the localized triangle mesh (or TIN) is used as the *ribbon*. This triangle mesh is effectively used as a spatial data structure for guiding the contour extraction. There are several roof reconstruction methods which use Delaunay triangulation to generate TINs for segmentation and other processes [19], [43]. However, our method is novel in the use of local triangulation and its construction, unlike the commonly used global triangulation[5].

The points pertaining to an isolated building belong to several layers at different heights for the same grid location (i.e. (x,y)). However, we are interested only in the *uppermost* skin for the roof structure. Additionally, it also implies that the 3D

[5]A global triangulation triangulates the entire point cloud, whereas a local one uses a smaller subset.

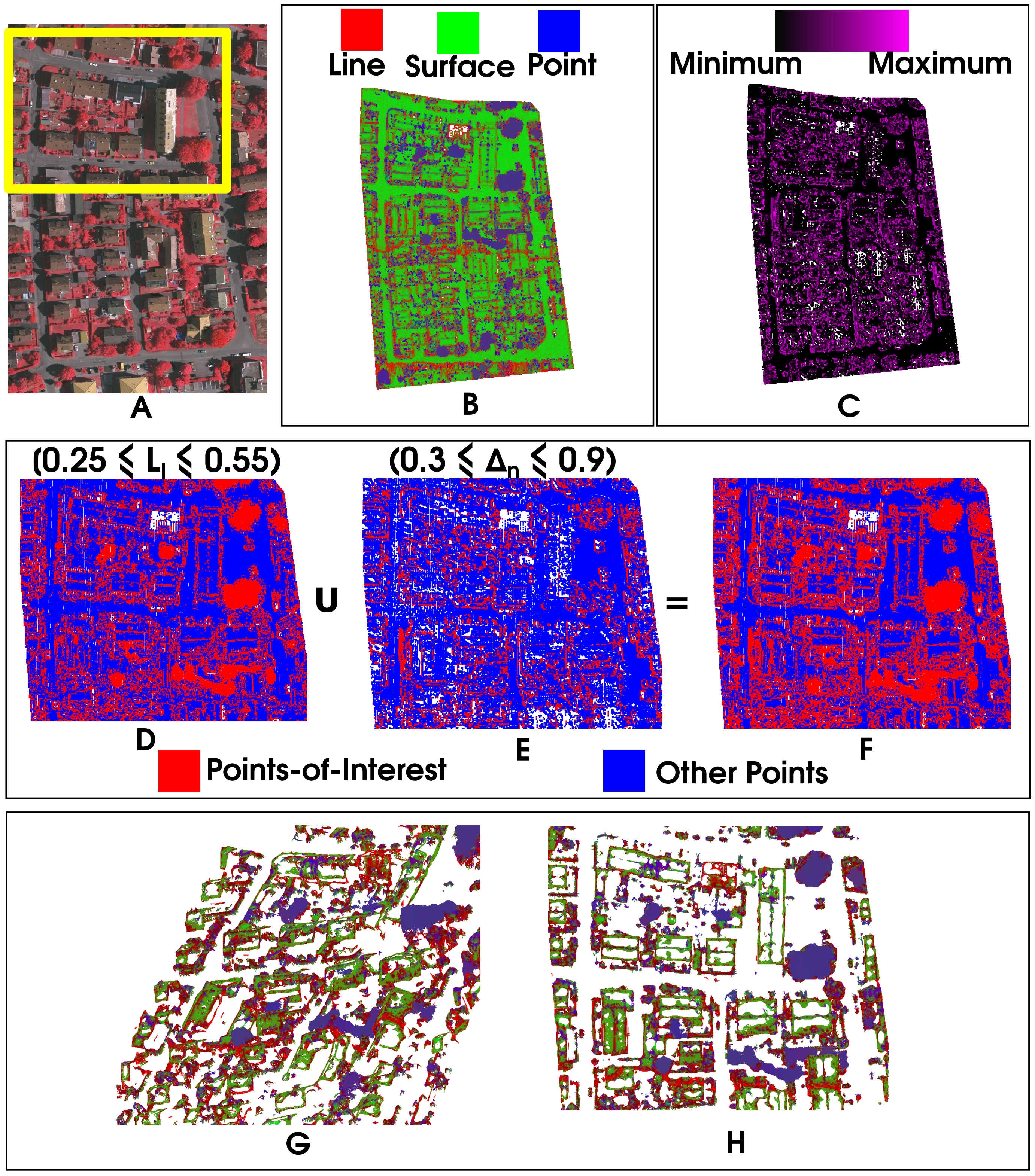


Fig. 2: We generate a local triangular irregular network (TIN) using points-of-interest identified using interval ranges of L_l and Δ_n parameters. For Area-3 of Vaihingen dataset (323,896 points), we show the ortho-map in A, and heatmaps of the saliency map (L_l, L_s, L_p) and difference of normals (Δ_n) in B and C, respectively. We identify subsets of points which have values of L_l and Δ_n in specific interval ranges. The union of these subsets give the set of the points-of-interest. The subsets for the focus area in the region in yellow box in A are shown in D and E, and their union in F. The TIN for the whole of Area-3 and the focus area are shown in G and H, respectively, where TIN is colored using the saliency map. shown in B. G shows 30,467 points-of-interest, yielding 3,891,276 triangles, out of the total 69,563 points in building class in Area-3.

triangulation, global or local, is non-planar and is not guaranteed to be a manifold surface. Hence, the triangulation is unsuitable for popularly used contour extraction methods in scientific visualization, namely the Marching Triangles, whose variant for 3D volumes is the Marching Cubes algorithm [44]. One could argue finding the upper skin by height clustering, i.e. finding the highest plane points [28], which can be then triangulated. However, we have found that our visualizations of the feature space help in identifying predominantly the points on the highest plane, which has been an unintended outcome and improves the planarity of our triangulation.

Points-of-Interest: The local triangulation is implemented on a subset of the point cloud, called *points-of-interest*. These points are identified by finding clusters based on a specific criteria, e.g. same object class [45] or same structural class, which is applicable in our case too. These points-of-interest become vertices of the local TIN. The benefit of using local TIN is to ensure sufficient sampling of points along the line-type features to enable contour extraction, which is indicated by the "width" of the ribbon. Here, we select the *building* points which visually represent the gable line and the roof boundary, and their corresponding local neighborhoods as points-of-interest for local triangulation. We use visualizations of the parameter space, i.e. heatmaps/colormaps of L_l and Δ_n, to interactively find the interval ranges of those parameters which capture the feature lines. While L_l improves the likelihood of points on the gable line to be line-type features, we use Δ_n to ensure the inclusion of the line-type features on the found on the gable-roof line without including other non-boundary points of similar L_l.

In our experiments on Area-3 of the Vaihingen benchmark dataset, we use 30,467 points-of-interest of the total 69,563 points in building class. This shows how we have narrowed down the points needed for further processing, to 43% of building points. The set union operation of interval ranges of more than one feature is a novelty of our approach, which has not yet been considered for LiDAR point processing. Our visualizations using heatmaps (shown in Fig. 2) play a key role in performing these set operations on point cloud for identifying points-of-interest.

Localized Delaunay Triangulation: The motivation for the local triangulation for contour extraction is three-fold – firstly, to reduce the dimensionality through the data analytic workflow, by extracting a piecewise planar data structure in 2.5D data, i.e. the point cloud; secondly, to reduce the computational cost of forming a TIN for the all the points in the building class; and finally, to extract local structures by exploiting spatial locality. Our method progressively propagates the Delaunay triangulation for local neighborhood of each point-of-interest using advancing front techniques [46]. Our local TIN ideally should have complete coverage of all edges, intended to be extracted as line-type features from the point cloud. The points-of-interest are the vertices of these edges.

B. Tensorline-based Contour Extraction Methods

We extract tensorlines from the local TIN, where the tensorlines contain the points that belong to the line segments of the roof. The idea is to effectively *skeletonize* the local TIN to give the sharp features in the buildings in the point cloud. Since the LGDs of line-type features are cylindrical shaped [3], [16], tracing the major eigenvectors directionally gives us the points that ideally lie on the line-type features.

We start with seed points and then *propagate* the tensorline to a neighboring vertex in TIN, at a time. Thus, the problem of tracing tensorlines reduces to identifying the *best edge candidate* in a triangle in the TIN which will allow the tensorline to propagate to the best potential neighboring vertex. This problem is resolved using a two-step approach, namely, identifying seed points and using our novel weighting function to identify the best edge candidate containing the current vertex. Thus, we use the TIN as a guideline for propagating the contour, while imposing a constraint that our computed tensorlines are modified to contain only vertices of the TIN. The original tensorlines have used a fixed step distance to integrate the vector in the propagating direction using Runge-Kutta methods or other integration methods, as is done with streamline computation in vector fields [33]. Our modification of using vertices of TIN implies adaptive step distance for tensorline propagation. Also, we ensure the use of LiDAR points on the edges for a majority part of our proposed workflow, until our post-processing step of line fitting.

Identification of Seed Points: The seed points should ideally be points on the roof boundary and gable lines. While our local TIN contains the seed points as well as its neighboring points, we need to isolate the former. To fulfil the need, we automatically identify them using local maxima and minima in the TIN, in the case of two-plane gable roof. Local maxima in the TIN automatically isolate the points on the gable line, and the local minima give those on the roof boundary.

Weighted Function for Scoring Edges: Starting from a seed point, we use the major eigenvector, e_0, of the LGD of the point as the *guiding* direction to propagate the tensorline/contour to the next point. Unlike the streamline computation in tensorlines, we impose a constraint that our tensorlines are generated exclusively from the vertices of TIN. This constraint requires the use of a *ranking* or a *scoring* function to find the best potential neighboring vertex to the current point on the tensorline. We use four significant criteria in identifying the best neighboring vertex for propagating the tensorline. We have two additive contributions in the scoring function, and its corresponding weights. We use length and angle criteria for contributions and its weights, respectively. One of the two length-angle criteria pairs correspond to eigenvalue decomposition of the LGD, and the other to the edges in the TIN.

We use the major eigenvalue of the LGD of the neighboring vertex and length of the edge connecting the current and the neighboring vertices as the two additive contributions. We use normalized values, where the normalization factor uses the corresponding maximum values in the local neighborhood. The idea here is that an LGD with a higher major eigenvalue and a longer edge are ideal candidates for tensorline propagation.

Given two edges with similar values in either component, we use appropriate weights to decide which edge is to be selected between the two. The first criterion ensures that the tensorline is tangential to the major eigenvector e_0 of the LGD at any point, and the second one, that the deviation from the path at any given point should be in the forward direction. The latter ensures that the tensorline does not reverse direction at any point or introduces cycles. The angle of deviation from the path is not considered for the seed point, since it is the starting point on the tensorline. Given the bidirectional nature of eigenvectors, the minimum angle of deviation enables choice in the direction of tensorline propagation. Thus, the score of an edge between current vertex v_1 and neighboring vertex v_2 is a weighted sum:

$$\begin{aligned} S_e(v_1,v_2) &= \lambda_0(T_{3DVT-GET}(v_2))/ \\ &\quad argmax_{v\in N(v_1)}\lambda_0(T_{3DVT-GET}(v)) \\ S_d(v_1,v_2) &= \|v_2-v_1\|_2/argmax_{v\in N(v_1)}\|v-v_1\|_2 \\ w_e &= 0.5*(\hat{e}_0(T_{3DVT-GET}(v_1))+ \\ &\quad \hat{e}_0(T_{3DVT-GET}(v_2)))\cdot(v_2-v_1)/\|v_2-v_1\|_2 \\ w_d &= (v_1-v_0)\cdot(v_2-v_1)/(\|v_1-v_0\|_2*\|v_2-v_1\|_2) \\ S(v_1,v_2) &= w_e*S_e(v_1,v_2)+w_d*S_d(v_1,v_2) \end{aligned} \tag{10}$$

Here, edge connecting v_0 and v_1 is the previous edge in the tensorline. For finding the potential neighboring vertex of the seed point, w_d is set as 0, in the absence of a *previous* edge. Our algorithm can be flexibly made a multi-pass one by iteratively converting all points in tensorlines in previous passes as seed points. The multi-pass approach allows us to fill in missing links in preceding iterations. Fig. 3 shows the stages in tensorline computation, starting from the local TIN.

Discussion of Our Tensorline-based Approach: Our method is similar to the feature graph extraction proposed in [16]. The motivation behind computing the feature graph and, in a similar vein, tensorlines is for identifying a larger subset of points, in addition to the seed points, which have a higher likelihood of lying on the line features (i.e. boundary lines and gable line) of the roof. Probabilistic methods for structural classification do not guarantee that all points identified as line-type features lie on the line features.

Keller et al. have found points in a local neighborhood of specific size whose major eigenvector has least deviation to that of the seed points, and thus propagates the lines. Our method is similar to this methodology, however, the selection of seed points and the search for neighborhood points are different in the two methods. Their selection of seed points is fixed, i.e. they are the line-type features as well as critical points in the point cloud; whereas ours is from the local minima and maxima of the local TIN.

While Keller et al. have used an octree to search for neighborhood points for propagating the feature graph, we use the local TIN to find neighborhood of intermediate point to propagate the tensorline from triangle to triangle. Similar to our method, in [18], tensorlines are used for edge extraction in building. However, different from our method, Gross et al. have used a set of points as trigger points to propagate the major eigenvector of the covariance tensor, after which points collinear to the trigger points in the direction of the eigenvector are used for constructing edges. Our method alleviates the burden of appropriate selection of seed points and appropriate step-size, which is present in the tensorline computation algorithm [33].

Tensor-based Data Analytic Workflow: In [3], [34] we have shown how the LGDs used for LiDAR point processing can be represented as second-order tensor fields. In this work, we show how the tensor representation can be included in a data analytic workflow. The workflow is consolidated in Fig. 4, where we show that the tensor-based data analytic workflow is supported by intermediate data elements and their analysis.

C. Post-processing Tensorlines

Our tensorlines are noisy and do not define the line segments of the roofs neatly. Hence we perform a four-step post processing on the tensorlines.

1) We first segment the set of tensorlines so that each subset of tensorlines correspond to a line segment in the roof. For this, we interactively perform a *lasso operation* in the graphical user interface (GUI), to identify the tensorlines belonging to the segment.
2) We use a linear least squares line fitting algorithm in order to find a single line segment to find cleaner lines. This is done to minimize the root mean square error of fitting a single line segment for each subset of tensorlines, as shown in Fig. 5-A.
3) However, this does not give us the expected roof topology graph, as shown in Fig. 5. Every roof type has a corresponding roof topology graph, as shown in Fig. 5-B for a simple gable roof with two planes, which guides its geometric reconstruction. Hence, in the third step, we identify the positions for each node in the topology using an optimization method.
4) In the final step, we use coplanarity criteria on roof planes described in the roof topology graph to get the final contour.

In our work, we demonstrate the implementation of the post-processing step on the tensorlines for a simple gable roof with two planes. It must be noted that all steps until the extraction of tensorlines are the same for all roof types.

The lasso operation *requires* the human-in-the-loop, and hence can be a source of Type-II error (or *false negatives*), further discussed in Section V. While we can automate this method using density-based clustering [47] for outlier removal, our current

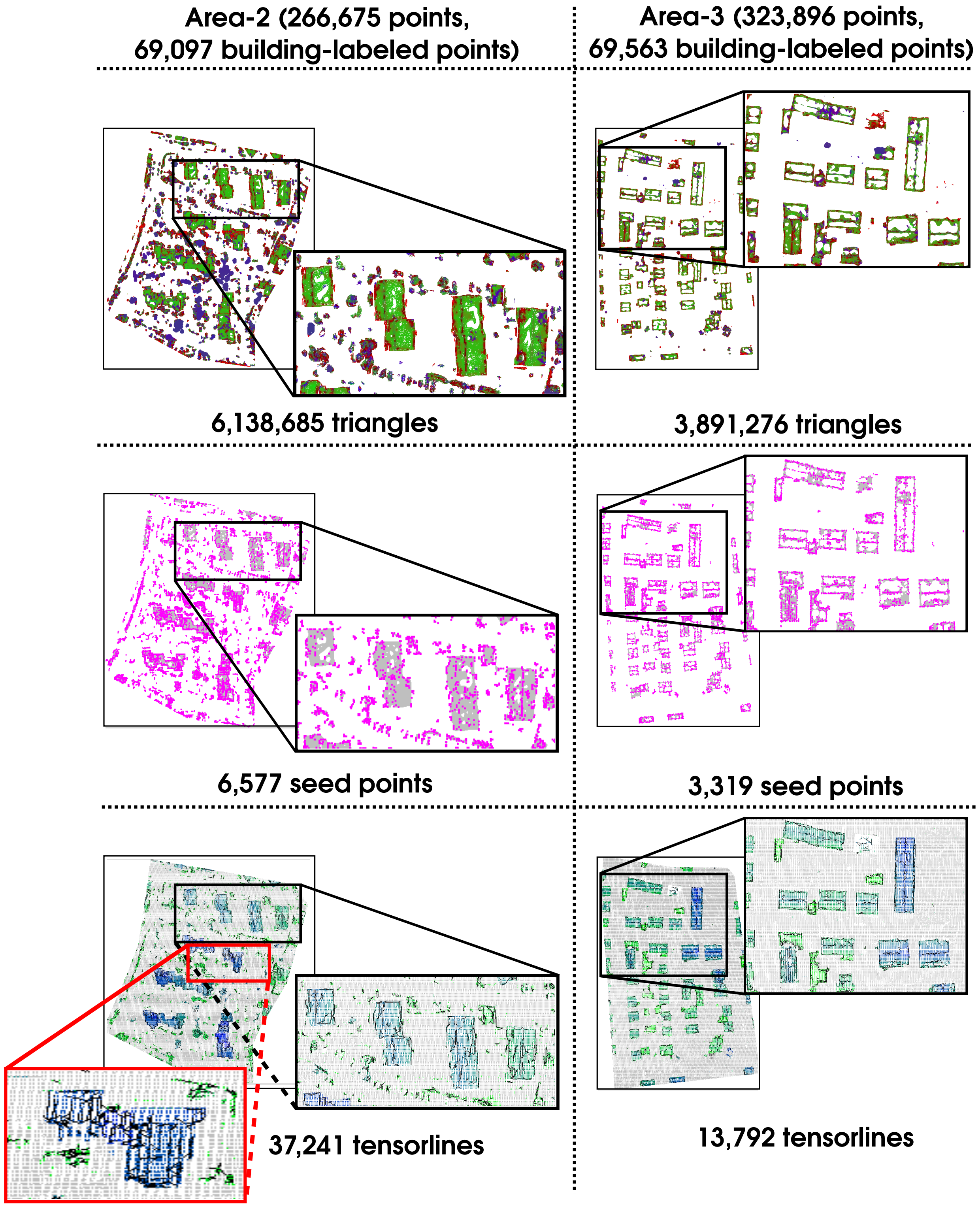


Fig. 3: The sequential outcomes of different stages in our tensorline computation. In Area-2 (left column) and Area-3 (right column) of the Vaihingen dataset, we compute the local triangulation (TIN) based on specific criteria derived from local geometric descriptors, shown in the top row, where the local TIN is colored using the saliency map. This is followed by automated computation of seed points (magenta points) from the TIN (gray surface), shown in the middle row. We then compute tensorlines (black lines) from the local geometric descriptors, for building points (colored green to blue based on height map), as shown in the bottom row). The TIN is generated from a set of points-of-interest which is a union set of points with $L_l \in [0.15, 0.5]$ and $\Delta_n \in [0.3, 0.9]$, in the case of Area-2, and a union set of $L_l \in [0.25, 0.55]$ and $\Delta_n \in [0.3, 0.9]$, for that of Area-3. Thus, we progressively reduce the number of points in the point cloud at each step of our workflow. Here, seed points account for only 9.5% and 4.7% of the building class points, in Area-2 and Area-3, respectively. The red inset for a building in Area-2 shows noisy tensorlines.

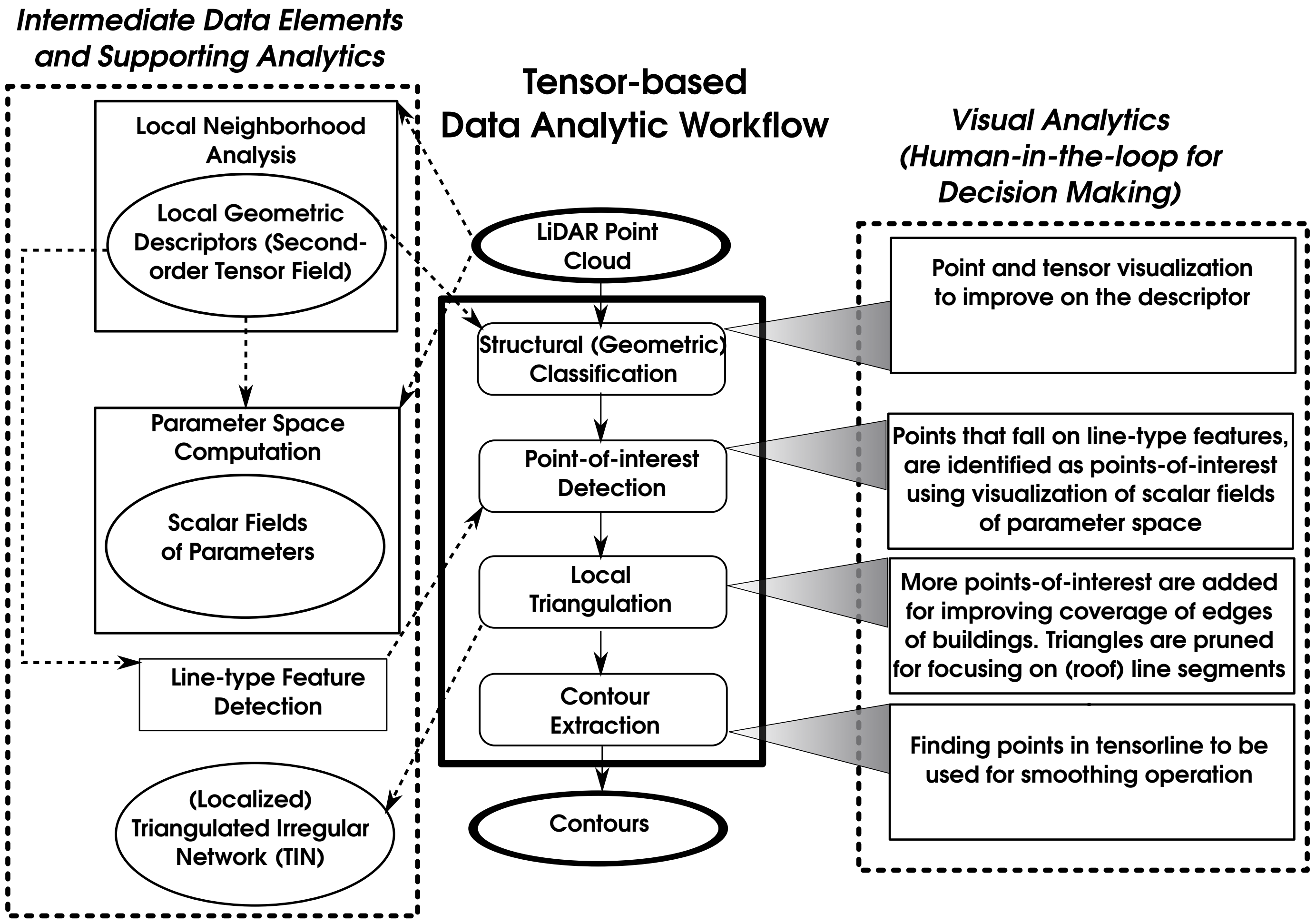

Intermediate Data Elements and Supporting Analytics
Tensor-based Data Analytic Workflow
Visual Analytics (Human-in-the-loop for Decision Making)
Local Neighborhood Analysis
Local Geometric Descriptors (Second-order Tensor Field)
Parameter Space Computation
Scalar Fields of Parameters
Line-type Feature Detection
(Localized) Triangulated Irregular Network (TIN)
LiDAR Point Cloud
Structural (Geometric) Classification
Point-of-interest Detection
Local Triangulation
Contour Extraction
Contours
Point and tensor visualization to improve on the descriptor
Points that fall on line-type features, are identified as points-of-interest using visualization of scalar fields of parameter space
More points-of-interest are added for improving coverage of edges of buildings. Triangles are pruned for focusing on (roof) line segments
Finding points in tensorline to be used for smoothing operation

Fig. 4: Our tensor-based data analytic workflow for LiDAR point processing for contour extraction. The ellipses and rectangles show the data elements and processes, respectively. The workflow, in itself, is shown in the middle, and on the left is the group of intermediate data elements and supporting analytic processes. To the right of the workflow is the set of visual analytic processes we propose, for improving the outcomes of the workflow by providing feedback using visualizations. The visual analytic processes allows the human-in-the-loop to make decisions for improving outcomes of the workflow.

implementation is useful for interactive exclusion of outliers in the tensorlines. Analysis of appropriate density-based clustering algorithm for this application is currently out of the scope of this work.

As shown in Fig. 5-A, we observe that each node in the roof topology graph does not have a unique location, and can have at least two options for its position coordinates based on the end points of the line segments which intersect/meet at the node. These multiple options occur because we have independently computed the line segments without considering their intersections. For instance, for the roof boundary, the two options for each node are from the end points of the line segments obtained using two sets of tensorlines. For the gable roof, the two options are the line fitted from tensorlines as well as intersection points of the line segments in the roof boundary. Additionally, we include a third option of using the average of the points in the first two options of position coordinates at each node of the topology graph. We iteratively check for the three options for the nodes in the topology graph, fixing the roof boundary initially, and subsequently fixing the gable line. We optimize our solution by minimizing the angle distortion of final line segments fitting the topology graph and its corresponding line segments, by using the linear least squares fitting algorithm. The angle distortion is computed as the dot product of the unit vectors of the computed line segment and its corresponding corrected line segment. Now, the line fitting step gives us *closed* boundaries for the roof planes, but, not necessarily planar surfaces.

Once we fix the roof points in correspondence with its roof topology graph, we enforce coplanarity of points describing a roof plane or panel in the topology graph. For instance, for sets of points (1,2,6,5) or (1,2,4,3) in Fig. 5-C, we keep the gable line and an additional node per plane as *fixed*. We then identify nodes which are not coplanar with the fixed points in each roof plane, and project them perpendicularly to the roof plane. For example, in Fig. 5-C, we consider points (1,2,3,5) as fixed points. This is a simplistic approach which we use as a proof-of-concept. Further analysis needs to be done on a rigorous approach to enforce coplanarity. Additionally, this approach is usable for a simple two-plane gable roof, in our case, as we are correcting only two nodes per roof, which in itself gives relatively low deviations, as studied in Section V. It is not guaranteed that this approach will work for complex roof topologies, as identification of fixed points is nontrivial.

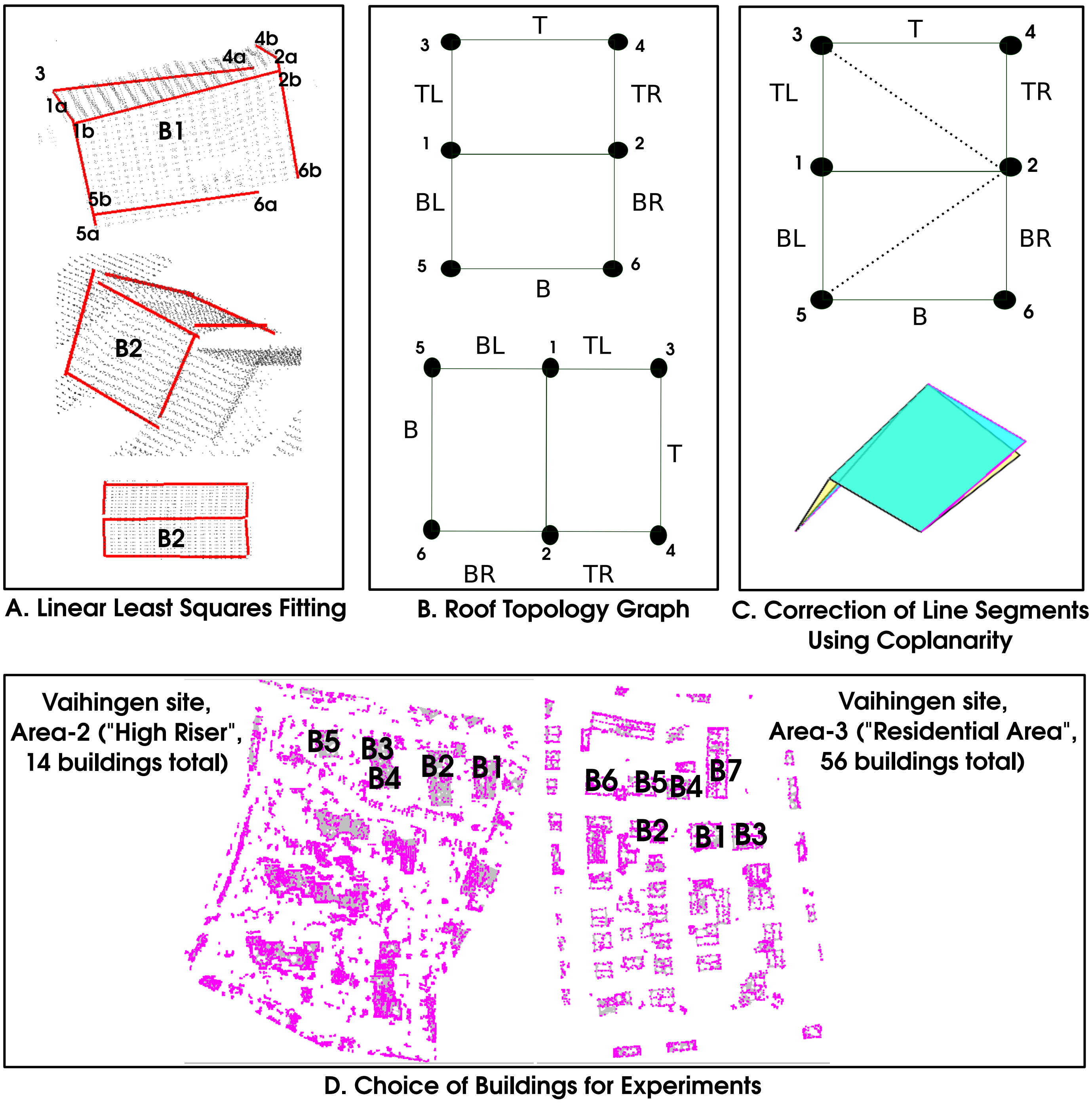

3
4b
4a
2a
2b
1a
1b
B1
6b
5b
6a
5a
B2
B2
A. Linear Least Squares Fitting
T
3
4
TL
TR
1
2
BL
BR
5
6
B
5
BL
1
TL
3
B
T
6
2
4
BR
TR
B. Roof Topology Graph
3
T
4
TL
TR
1
2
BL
BR
5
B
6
C. Correction of Line Segments Using Coplanarity
Vaihingen site, Area-2 ("High Riser", 14 buildings total)
B5
B3
B4
B2
B1
B6
B5
B4
B7
B2
B1
B3
Vaihingen site, Area-3 ("Residential Area", 56 buildings total)
D. Choice of Buildings for Experiments

Fig. 5: Our line-extraction module for building roofs consists of three steps, namely, (i) performing a linear least square on tensorlines clustering along the edges, shown in A, (ii) using the roof topology graph to determine the closed boundary of the roof, as shown in B, and (iii) determining planar roof panels by projecting non-planar points to the approximated roof plane. In B, we show the roof topology graph for a simple gabled roof, for which we maintain the order of line-segments (marked with T,B,R,L for top, bottom, right, and left, respectively). This ordering is maintained for giving consistent clockwise walks in the two roof planes/panels, irrespective of its orientation. For our experiments in Area-2 and Area-3 of the Vaihingen dataset, we have selected five and seven buildings, respectively, for reconstruction, as shown in D.

D. Summary

Our tensorline-based workflow for line segment extraction in building roofs from airborne LiDAR point clouds can be summed up as the following algorithm:

1) Find multi-scale LGD at each point $T_{3DVT-GET}$. Using the LGD, compute the parameter feature vector, namely, saliency maps (L_l, L_s, L_p), and DoN $\|\Delta_n\|$, at each point.
2) Find the points-of-interest $\mathscr{S}_l$, using relevant classes and interval ranges of feature parameters identified by heatmaps.
3) Compute the TIN of $\mathscr{S}_l$ using Delaunay triangulation.
4) Find seed points for tensorline extraction, e.g. local minima and maxima of the TIN.
5) For each edge added to in each tensorline, check potential front propagating edge by ranking the scoring function (Equation 10) for all neighboring vertices of the current vertex in each tensorline.
6) Segment or isolate tensorlines belonging to each line segment of the roof, i.e. edge in the roof topology graph, using a lasso operation.
7) For each segment of tensorlines, perform a least squares line fitting algorithm to find the best oriented segment.
8) Correct the computed line segment by assigning position coordinates to the nodes in the roof topology graph based on the possible options for each node.
9) Identify fixed nodes in the roof topology graph for defining roof planes, and project the non-fixed nodes to ensure planarity in the reconstructed roof planes.

V. Experiments

We have performed our experiments on two datasets, namely Area-2 and Area-3 of Vaihingen dataset. The regions have been described as the "High Riser" and the "Residential Area". The choice of Area-3 is especially apt owing to the large number of buildings with gable roofs. For multi-scale computations using normalized coordinates, we have used r_{min} and r_{max} as 0.008 and 0.012, with five scales, where the radii of local neighborhood are normalized in the bounding box of the dataset. For T_{3DVT} and its variants, including $T_{3DVT-GET}$, we have used diffusion parameter $\delta = 0.16$ [3]. The focus of our current work has been on the geometry extraction for building roofs amongst other objects, as they give the signature for an urban or residential region and are conventionally used for 3D reconstruction. Our choice of buildings with simple gable roofs is shown in Fig. 5-D.

A. Results and Discussions

The results of our contour extraction method using LGDs and our visual analytic approach are as shown in Fig. 6. Our optimization method has given minimum distortion when taking the third option for all the nodes of the roof topology graph, i.e. the average of the two options for position coordinates obtained from least squares line fitting algorithm. We have also compared our baseline algorithm for feature graph extraction proposed in [16] (using covariance tensor) with the best results of Marching Triangles algorithm and tensorline-based methods, as shown in Fig. 7. Overall, our workflow shows better coverage of the feature lines of the roofs, i.e. both boundaries and gable lines. The improvement in the coverage is due to the use of points-of-interest which has larger coverage of both line-type features and other relevant points which satisfy specific criteria. Our method is unlike the feature graph extraction proposed in [16] which limits its selection of nodes to the line-type features.

Quality of the Tensorlines and its Equivalence to the Feature Graph: The extraction of feature graph in [16] is intended to remove extraneous lines. However, our visualizations demonstrate that their coverage of sharp features by the feature graph is limited, for both building boundaries and gable lines. Since the boundaries are not extracted completely in the feature graph, even flat roof boundaries cannot be extracted directly. Our method, however, oversamples the contours for both boundaries as well as gable roof lines. We argue that denser set of contours is preferable to the sparser one, as the line fitting step will give better results with more samples.

Errors Accumulated during Post-processing Step: As expected, the contours extracted in different steps of our post-processing method deviate from the subsets of the original point cloud, which form the original tensorlines. As plotted in Fig. 8, we have computed the deviation caused by the post-processing step using the root mean square errors (RMSE) between: (a) line segments computed using linear least squares line fitting and tensorlines, and (b) the final line segments and tensorlines. The final line segments are the outcome of the corrections for both closed boundaries of line segments, and coplanarity in the roof planes. Our experiments show that the RMSEs are negligible (the maximum error is $\sim$ 5.6%, i.e. $\sim$ 14m, per line segment in the roof topology graph). However, we observe through our visualizations that the shape of the roof has distorted slightly in few buildings. These distortions reflect in our ground truth analyses in Table I. One of our future exercises would be to correct line segments using other related properties such as symmetry of the structure, etc..

Ground Truth Analysis: The Vaihingen dataset has been published for ISPRS benchmark challenge for reconstruction from airborne LiDAR point clouds [6]. We have compared our results with the ground truth images from three different sources: (GT-a) the roof boundaries from the challenge organizers (Institut für Photogrammetrie und GeoInformation, Leibniz Universität Hannover), (GT-b) semantic (object-based) labeling ground truth for one of the datasets, Area-3, as published by the challenge organizers, and (GT-c) Openstreetmap [48] data, downloaded for the region encompassing both Area-2 and Area-3. The ground truth analysis has been done by using overlay of our results on ground truth images, as shown in Fig. 10 – Sets A and B are

[6]ISPRS benchmark challenge for detection and reconstruction: http://www2.isprs.org/commissions/comm3/wg4/detection-and-reconstruction.html

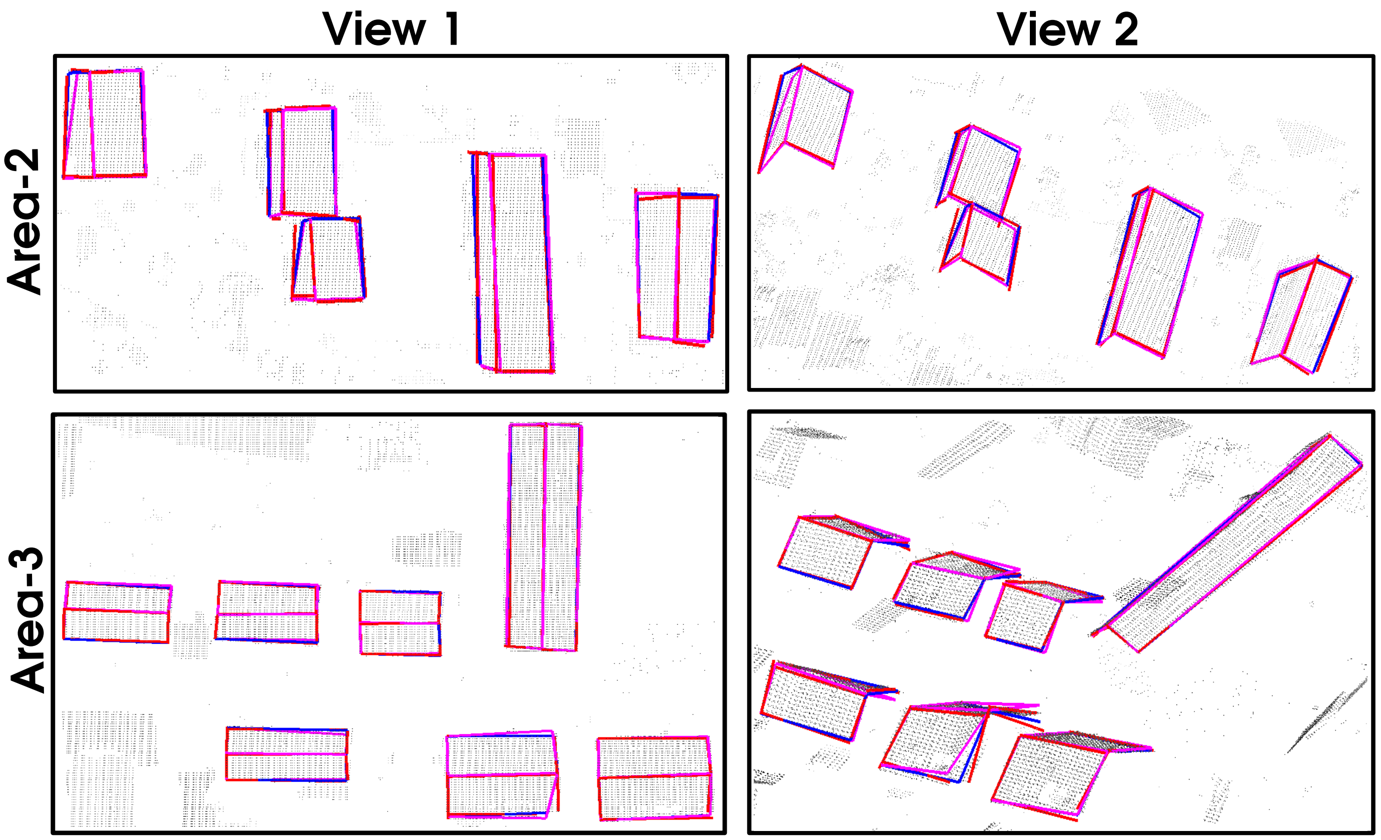


Fig. 6: Extracting line segments from tensorlines of building roofs is done in three sequential steps: (i) linear least square fit, (ii) roof topology graph based correction, and (iii) projection of points for coplanarity. Results from (i), (ii), and (iii) are shown in red, blue, and magenta line segments, respectively, for buildings marked in Fig. 5-D. Correcting the line segments based on roof topology graph using averages of options has given the best results. Visually, our final output, i.e. magenta line segments, aligns the best to roofs observed in the raw point clouds.

outcomes for Area-2 and Area-3, respectively, using (GT-a); Set C is that for Area-3 using (GT-b), and Set D is that for both Area-2 and Area-3 using (GT-c).

Unlike other automated contour extraction techniques [23], [29], the metrics based on number of detected edges are not applicable to our work as our methodology involves user interaction on the GUI to pick edges from the visualization. Hence, we have used the metrics based on pixel coverage of our extracted roof boundaries and the planes fitted in the boundaries. We have recorded the quantitative analysis based on confusion matrices for the ground truth analyses in Table I. We observe the following from Fig. 10 and Table I:

- In general, we find that our results have higher accuracy compared to other roof extraction methods. This observation is as expected, since our method is not automated and is based on the decisions made by the human-in-the-loop guided by appropriate visualizations.
- The ground truth analyses of sets A and B perform weaker than that of C and D. This could be attributed to the thicker roof boundaries without the filling of roof panels, which leads to ambiguity in coverage. In a similar vein, the recall in sets A and B are comparatively lesser than those of sets C and D, owing to higher ratio of false negatives to true positives. We attribute this observation due to the roof boundaries being marked, in the ground truth in sets A and B, to be wider than a single pixel.
- Since we have performed the roof reconstruction for buildings with the topology of two-plane gable roof only, the true negatives in each of the ground truth analyses are relatively high, which skews the results involving true negatives, such as, specificity, and false positive rate. Even though computation of Matthews correlation coefficient (MCC) depends on the value of true negatives, it provides balanced measure for classes with large variations in size. Hence, we comment further on MCC as well as other metrics which do not depend on true negatives, namely, recall, precision and F1-score.
- We observe that our extracted roof boundaries do not sufficiently cover the ground truth data in all the sets. This could be attributed to the user being conservative in interactively choosing points *within* the roof boundaries as opposed to *on* or *outside of* the boundaries. This behavior is characterized by the relatively higher percentage of false negatives. This could also be attributed to the step of enforcing coplanarity in the roof planes. These observations lead us to conclude that while our proposed method shows *high precision*, it has *lower sensitivity* or *high miss rate*.
- The precision is relatively low in set D, which is due to high ratio of false positives to true positives. This could be

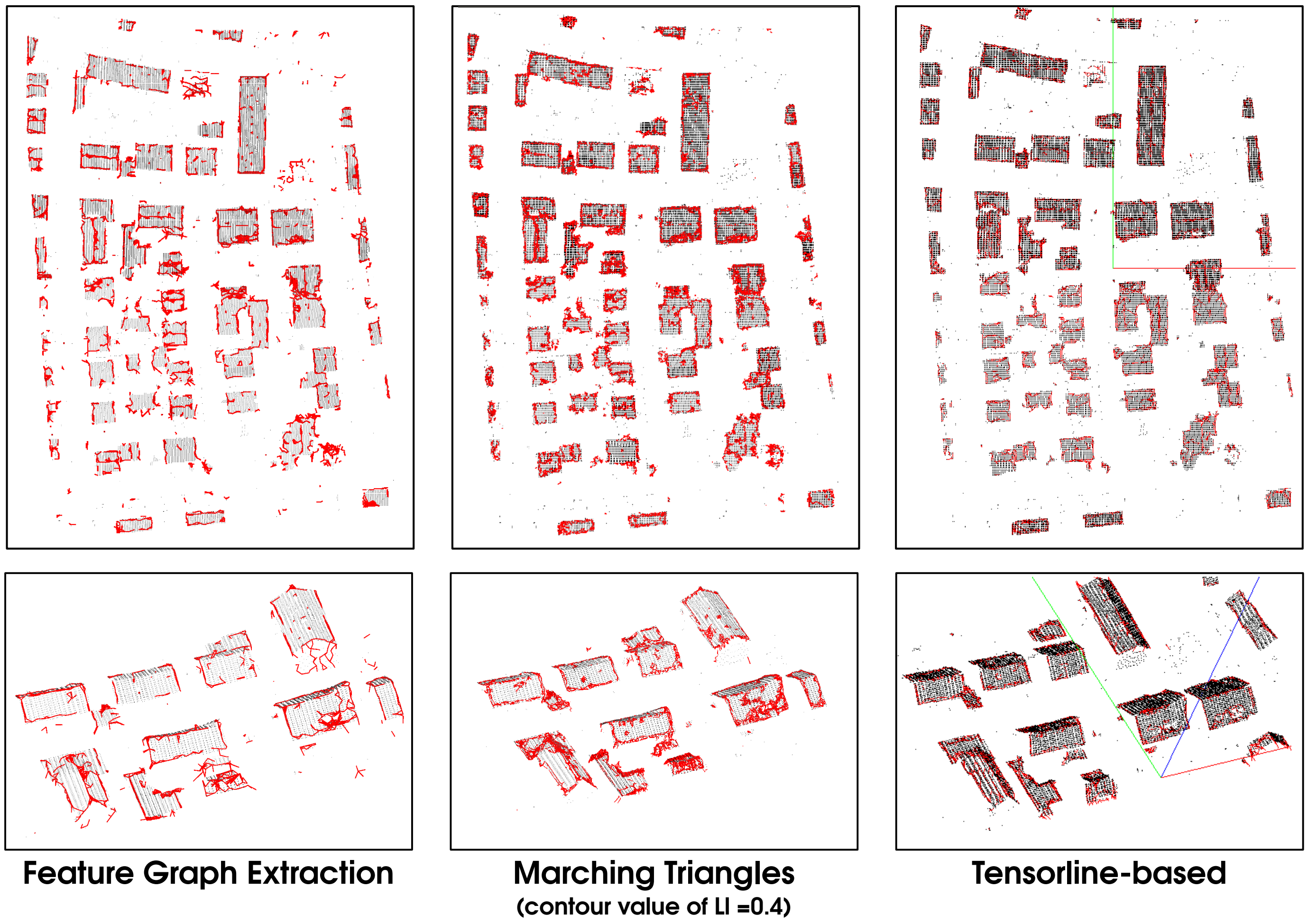


Fig. 7: Comparison of feature graph extraction [16] and similar methods using Marching Triangles for contour function constructed with the help of visualization, and tensorline-based method. Our results for our focus area in Area-3 of Vaihingen (bottom) as well as the entire Area-3 (top) show that our visual analytic method outperforms the feature graph extraction method in extracting building edges and gable roofs using cleaner lines.

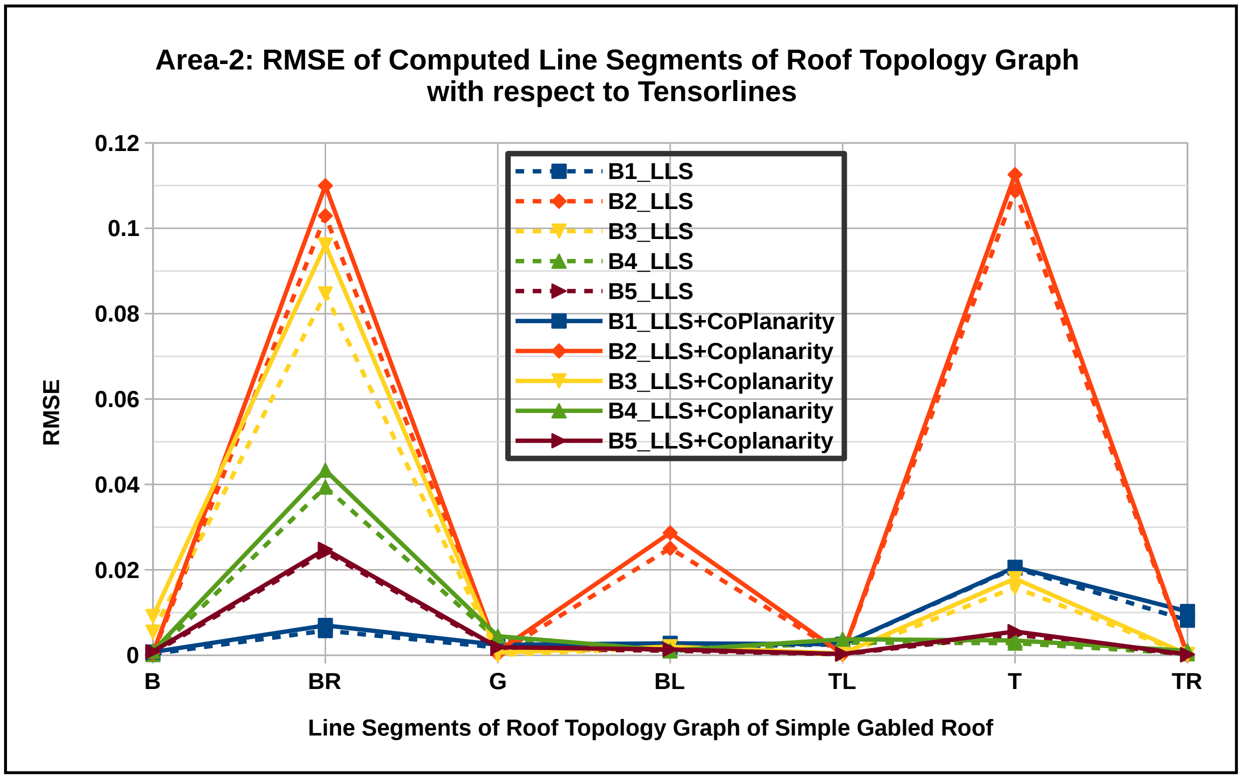
attributed to discrepancies seen between building footprints in Openstreetmap and reference data [49]. The F1-score of Set A is relatively low, owing to the comparatively low value of recall.

- The value of MCC is within $[0.84, 0.90]$ for all sets, except for set A, for which it is 0.79841. The values of MCC being closer to $+1$, than -1, imply that there is high correlation between observed and predicted values. The lower value of MCC in set A can be attributed to higher number of true negatives.

Run-time Performance Analysis: We have implemented our algorithm on workstation with a 1.9 GHz AMD quad core processor, A8-4500M, with 4 MB L2-cache; 6 GB 1600 MHz DDR3 memory, and video graphics with AMD Radeon HD 7640G and 7670M Dual GPU (1 GB DDR3 dedicated). Our process-wise and cumulative run-time performance analysis are shown in Fig. 9. Our algorithm works with the entire point cloud until the generation of the TIN, after which we use the TIN or the localized triangle mesh exclusively for further processing. Hence, our algorithm takes cumulative preprocessing time of 285.093 seconds for Area-2, and 384.332 seconds for Area-3. The use of TIN for further processing makes our algorithm for roof reconstruction real-time, by design, so that we get interactive speeds for GUI operations. Our timing measurements are computed as an average of five runs. We have observed negligible variances in each of the tasks.

We observe that computing T_{GET} is most compute-intensive process, owing to the computation of higher order derivatives (going up to third order derivatives). Comparatively, triangulation using Delaunay algorithm is lesser time consuming in our implementation, owing to the use of optimized implementation of the algorithm in CGAL [50].

As can be seen from our analysis, the preprocessing time is a function of the size of the LiDAR point cloud. Additionally, we attribute the high preprocessing time for Area-2 and Area-3 to our serial implementation. That said, our algorithm is embarrassingly parallel, since the computations in our workflow are either for each point in the point cloud, or for each triangle in the TIN. Thus, we can reduce the time taken for the steps until the TIN generation using parallel implementation. There is existing work on parallel implementation of octree building and computation of covariance matrices using Graphical Processing Units (GPUs) has been demonstrated in [51]. To the best of our knowledge, other similar work on building reconstruction do



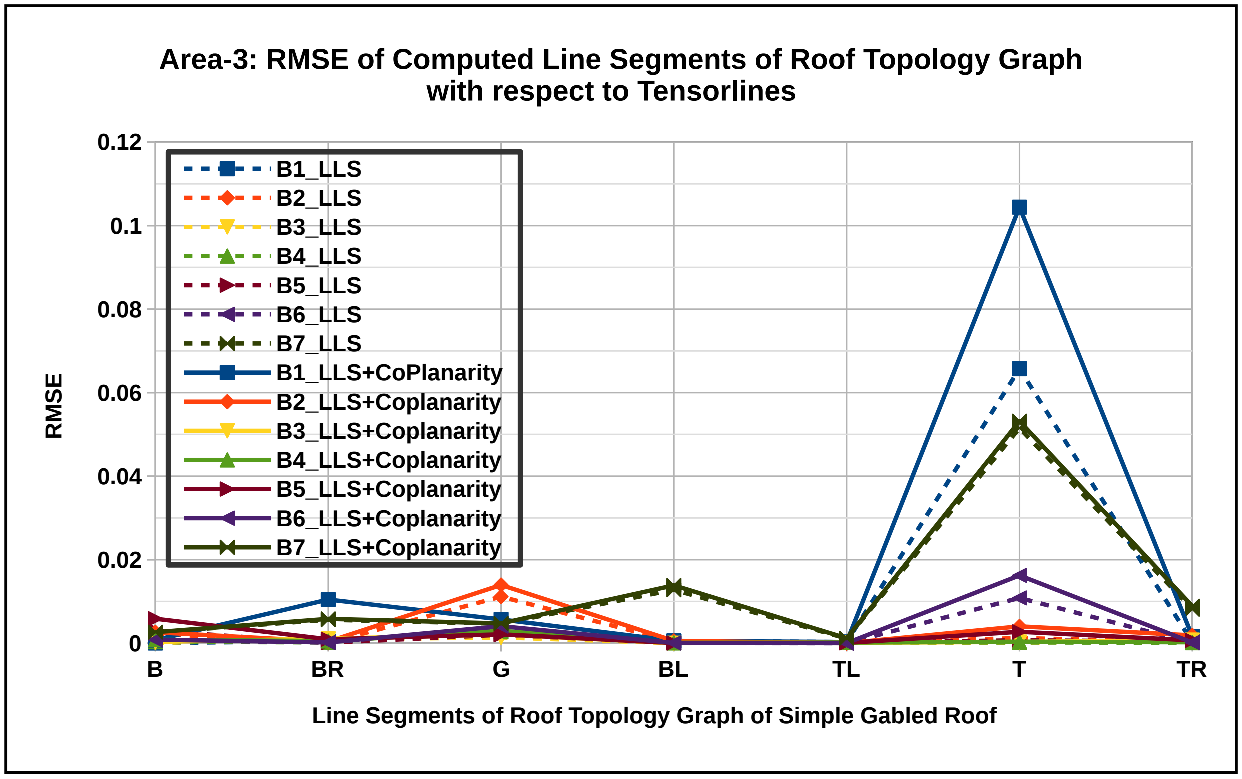


Fig. 8: The dotted lines show the root mean square error (RMSE) between corrected line segments using roof topology graph with respect to the tensorlines, and solid lines show the RMSE between the line segments corrected further for coplanarity with respect to the tensorlines. The RMSE is measured in distances in the bounding box of Area-2 and Area-3 datasets, after being normalized between -1 and 1. We observe that the maximum RMSEs or deviations from the tensorlines are 0.113 and 0.104 for Area-2 and Area-3, respectively. The respective RMSE percentages, which are 5.6% and 5.2%, are low given areal measurements of Area-2 and Area-3 are $(225 \times 256m)$ and $(193m \times 277m)$, respectively.
(The legend keys include the following notations: B1,...,B7 are the buildings identified for our proof-of-concept (given in Fig. 5), LLS is for linear least square fitting of tensorlines belonging to each line segment of the roof topology graph (Fig. 5), and CoPlanarity is for correction based on coplanarity of points in each roof panel.)

not usually provide run-time performance analysis. This can be attributed to the goal in those works being accurate one-time results for building and/or roof extraction, as opposed to real-time interactive exploration of and geometry extraction from the point clouds.

Comparison with Recent Methods: We have compared our results against the challenge results [7] We observe that our method has higher completeness (recall) and lesser correctness (precision) compared to the results given in the challenge. We could attribute this to the quality of selections made by the user in different steps in our proposed algorithm. Since we interactively retrieve the boundaries of the roof after tracing the tensorlines, the quality of our reconstruction depends on the quality of the local geometric descriptor, the choice of the TIN, as well as the tensorlines computed from the TIN.

For qualitative analysis using visualization, we have compared our results with the benchmark challenge results of [52], [53], [54]. In Fig. 11, we demonstrate a qualitative analysis of overlays of roof planes for both Area-2 and Area-3 of the Vaihingen site, with the comparative results of MEL-HE [52], MON2 [53] and MON5 [54]. We observe that for Area-2, the surface coverage of roof planes extracted from our methodology is not sufficient. The insufficiency is observed in our ground truth analysis of Set A in both Fig. 10 and Table I.

Similar to MON2 [53] and MEL-HE [52], our current implementation suffers the limitation of extraction of roofs planes of surface area less than $10m^2$. However, different from their methods, our limitation stems from the performance of the LGD which requires sufficient samples from a local neighborhood for providing accurate parameters, such as saliency maps (L_l, L_s, L_p). We alleviate this issue by using multiple scales. We also plan to explore the use of adaptive local neighborhood sizes in alleviating issues such as these, which are discrepancies owing to a fixed local neighborhood size.

MON5 [54] is an improvement over MON2. The methodology in MON5 uses a LGD, namely the covariance tensor, and improves upon the building reconstruction by detecting sharp features using statistical analysis. We propose to include statistical analysis in either improving the descriptor or in detecting sharp features more accurately. Currently, our methodology depends predominantly on gradient analysis of local neighborhoods. User interactivity guided by statistical analysis can improve the sensitivity of our methodology.

The core difference of our method lies in the facility for graphical exploratory data analysis using visualizations, in comparison to the methods evaluated at the benchmark challenge which are for automated detection of building and geometric extraction of roofs. Thus, overall, we provide the flexibility to the user to perform multiple runs of evaluation on the point clouds before running a single or a combination of automated methods. Our methodology can, thus, be considered to be a precursor to the automated methods or supervised learning techniques.

B. *Discussion on Visual Analytics*

In this paper, we have proposed the use of visual analytics in determining the interval ranges of parameter values to be chosen for extracting the local TIN. The human-in-the-loop, who is guided by the visualizations, is significant in making decisions on interval ranges of values of features, namely L_l and Δ_n, for determining coverage of roof boundaries and gable roof for all the buildings in the dataset (Fig. 2). The use of visualizations involves trial and error in the entire process. At the same time, it poses an opportunity to show the effectiveness of interactive tools for in-depth data explorations.

Our prototype tool has the ability to perform lasso operations which captures points that fall within the user-defined region. Currently, we use the lasso operation on identifying tensorlines belonging to the edges of the roof topology graph. This operation has a similar scope of automation using machine learning techniques, such as density clustering, which can identify elliptical shaped clusters, as are expected in the case of line segments in roofs.

Our visual analytic approach has the potential for further customization of similar data analytic processes on LiDAR point clouds. Our work is a first-cut attempt in the scope of this potential. The difference between the usage and non-usage of visual analytics is observable in our results in Fig. 6 where our methods outperform the feature graph extraction method.

VI. Conclusions

In this paper, we have proposed a contour extraction method driven by the local geometric descriptor (LGD) for airborne LiDAR point clouds in urban and residential regions. Our methodology is semi-automated as it is guided by visualization which limits the usage of our proposed building roof extraction to a smaller scale. However, our work can be used in the form of a graphical user interface (GUI) editor, for building extraction or for generating training data for supervised methods for roof reconstruction. We have used a LGD, $T_{3DVT-GET}$, which uses tensor voting and gradient energy tensor, and is represented as a positive semi-definite second-order tensor. We have used $T_{3DVT-GET}$ as it strengthens the line-type features on the gable roofs as well as on roof boundary. We use both eigenvectors and eigenvalues for generating tensorlines, which are analogous to streamlines in vector fields. We have extracted the tensorlines using a local triangulated irregular network (or triangular mesh) as our underlying data structure. Through LGDs, we have explored the feature space used for semantic classification (object-based classification) for geometric extraction. Conventionally the LGDs are not used beyond semantic classification. Our motivation for this work has been to provide a proof-of-concept for generalized data analytic workflows for LiDAR point clouds, such as the one based on second-order tensors, as well as re-using the computations used in semantic classification. Given that the supervised learning outcomes for semantic classification have been improving, we consider our work to be the next organic step in the data analytic workflow of the LiDAR point clouds, where we perform 3D reconstruction of labeled point

[7]Benchmark challenge results are available at http://www2.isprs.org/commissions/comm3/wg4/results/a2_recon.html for Area-2 and at http://www2.isprs.org/commissions/comm3/wg4/results/a3_recon.html for Area-3.

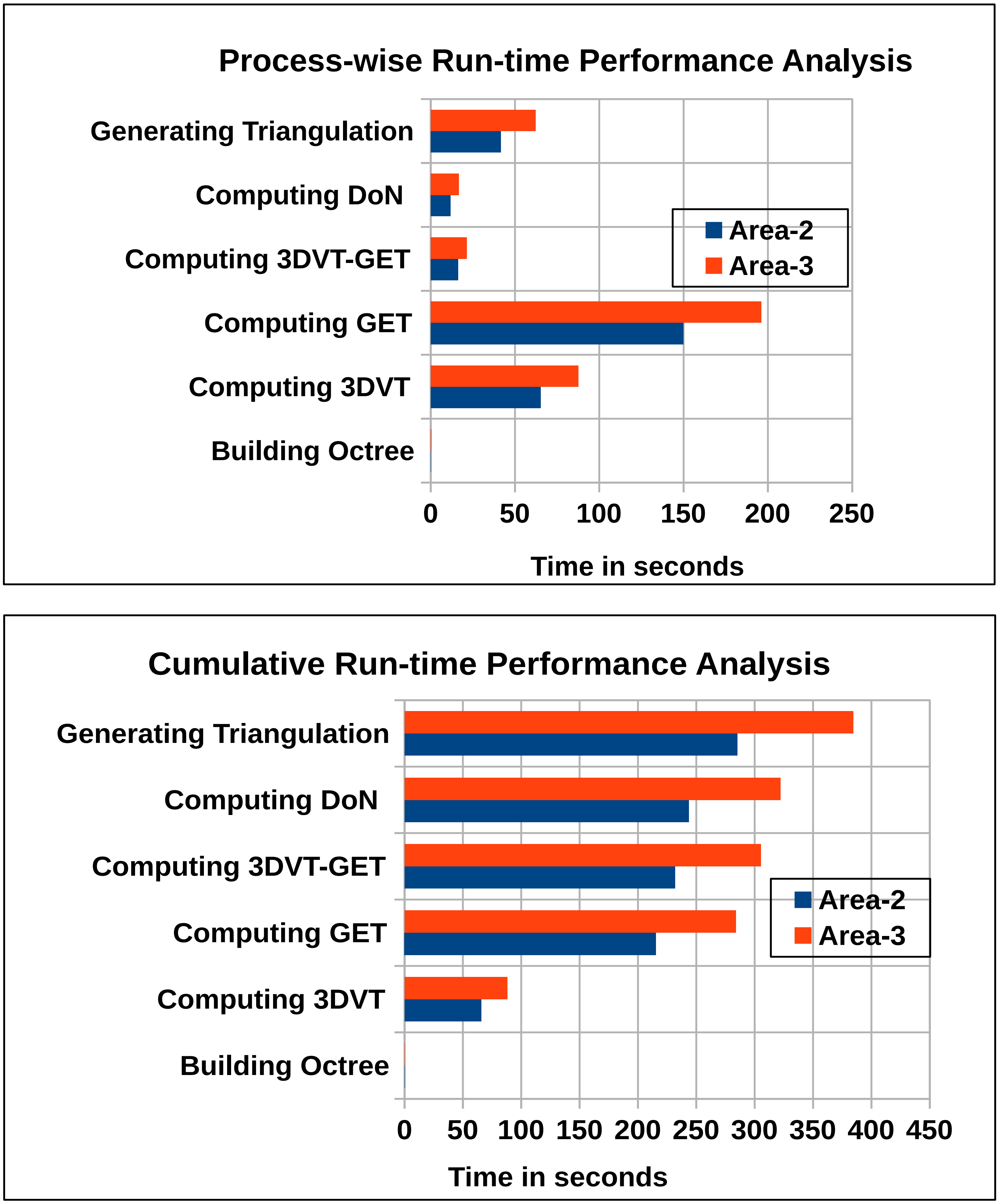


Fig. 9: Run-time performance analysis of serial implementation of our algorithm until the generation of the localized triangular mesh or TIN.

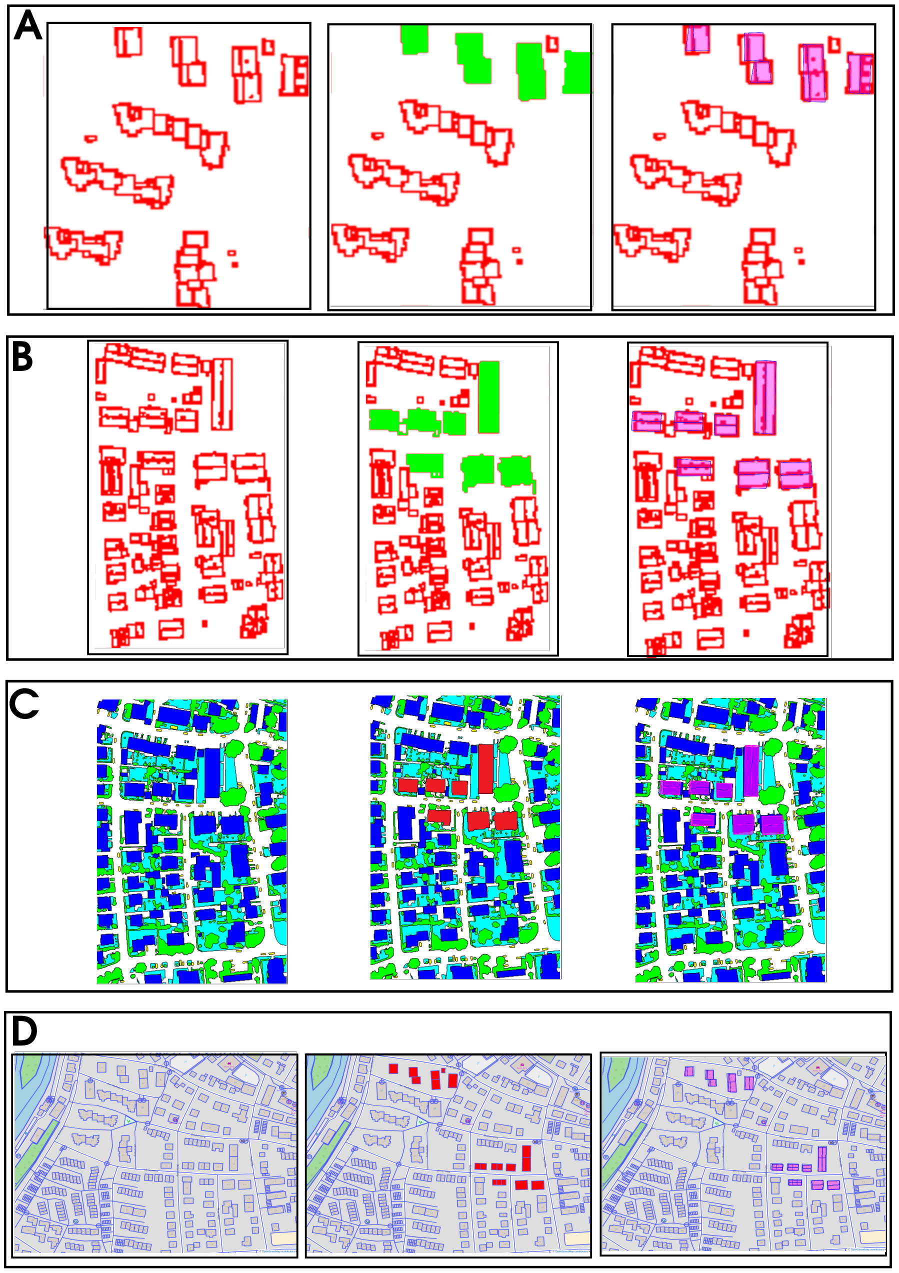


Fig. 10: Overlay of our extracted roof planes, fitted within our feature line extraction with different ground truth data. Sets A and B are ground truth obtained from data source [55] for Area-2 and 3, respectively. Set C is the ground truth for semantic labeling in Area-3, available from ISPRS benchmark challenge website, and Set D is the Openstreetmap [48] data obtained for Vaihingen site containing both Area-2 and 3. The left column shows the ground truth; the middle column shows the ground truth with the buildings marked in green (in Sets A and B) or red (in sets C and D); and the right column shows the ground truth with an overlay of our roof extraction results, shown in transparent magenta polygons and blue feature lines.

Set	A (Area-2)	B (Area-3)	C (Area-3)	D (Area-2 and Area-3)
True Positive	40583	51453	10472	13514
False Positive	5236	4173	1428	3172
False Negative	14217	12980	819	735
True Negative	804550	574317	230085	778751
Error rate	0.02250	0.02668	0.00925	0.00491
Accuracy	0.97750	0.97332	0.99075	0.99509
Recall	**0.74057**	**0.79855**	0.92746	0.94842
Precision	0.88572	0.92498	0.88000	**0.80990**
Specificity	0.99353	0.99279	0.99383	0.99594
FPR	0.00647	0.00721	0.00617	0.00406
MCC	0.79841	0.84527	0.89859	0.87404
F1-score	**0.80667**	0.85713	0.90311	0.87370

TABLE I: Quantitative ground-truth analysis using confusion matrix obtained from ground truth data in Sets A-D (Fig. 10). Confusion matrix gives the observed true positive (TP), false positive (FP), false negative (FN), and true negative (TN); and the computed metrics, namely, error rate (ERR), accuracy (ACC), true positive rate (TPR)/recall (REC), positive predictive rate (PPR)/precision (PREC), true negative rate (TNR)/specificity (SPEC), true negative rate (TNR), false positive rate (FPR), Matthews correlation coefficient (MCC) and F1-score.

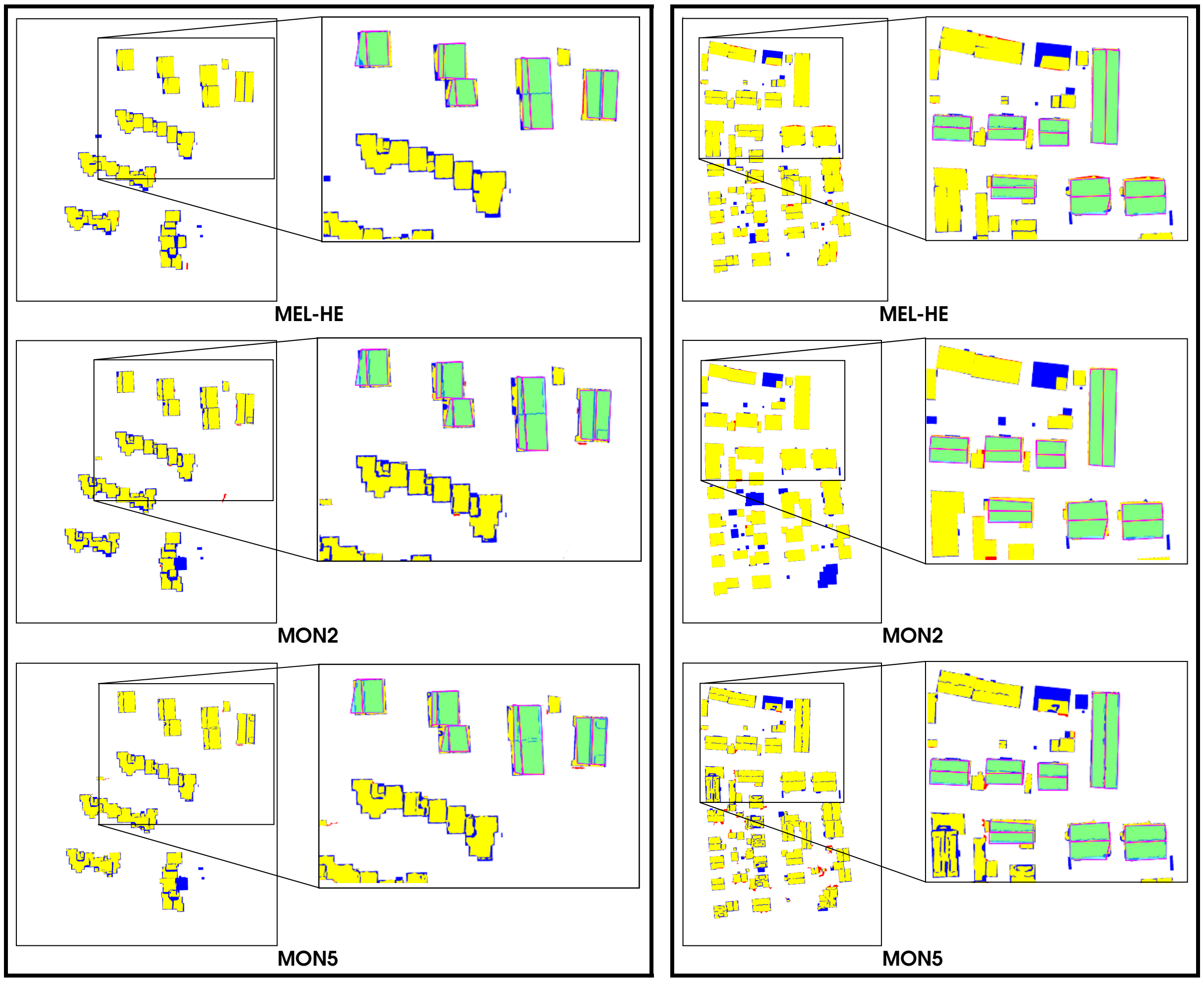


Fig. 11: For Area-2 (left box) and Area-3 (right box) of Vaihingen dataset, we have compared the ISPRS benchmark challenge submissions by MEL-HE [52], MON2 [53], and MON5 [54] by overlaying our results on their respective image results. The benchmark challenge submissions are shown in full in the left column in each of the boxes, with insets showing a zoomed-in version additionally with an overlay of our results, in the right column. The benchmark results, as published, show (true positive, false negative, false positive) pixels, with respect to the ground truth, in (yellow, blue, red). For the overlay of our results, we show our extracted feature lines on the roof in magenta, and planes fitted within the feature lines, as translucent cyan surfaces. Visually our results for the selected simple gable roofs are comparable with those of the benchmark results.

clouds. As the first step towards 3D reconstruction, we extract line segments on gabled roofs. We have devised solutions for problems of identification of seed points, extraction of tensorlines, and correcting line segments, fitting with the starting point of our methodology from a local TIN. Overall, our method is a novel approach to visual analytics-guided contour extraction in buildings in airborne LiDAR point clouds, which can be tested for other types of LiDAR data, which is beyond the scope of our current work.

Acknowledgment

The opportunity to present early versions of this work at the Dagstuhl Seminar 16142 was valuable in motivating the work further. The authors would like to thank S. Sadagopan and B. S. Daya Sagar for encouraging and supporting this work, in different capacities. This work has used third-party software libraries, namely, PCL [56], CGAL [50], and OpenCV [57] for implementations of point cloud processing, geometric processing, and image processing algorithms, respectively. The Vaihingen data set has been published by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [55][8]The authors are grateful to Franz Rottensteiner, Uwe Breitkopf, and Francesca Bovolo for their inputs for the ground truth analysis.

References

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

[8]Vaihingen dataset acquisition project: http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html

[2] Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, 2015.

[3] Jaya Sreevalsan-Nair and Beena Kumari, *Local Geometric Descriptors for Multi-Scale Probabilistic Point Classification of Airborne LiDAR Point Clouds*, pp. 175–200, Springer Cham, Mathematics and Visualization, 2017.

[4] 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*, vol. 87, pp. 152–165, 2014.

[5] AM Ramiya, Rama Rao Nidamanuri, and R Krishnan, "Semantic Labelling of Urban Point Cloud Data," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 907–911, 2014.

[6] J Niemeyer, F Rottensteiner, and U Soergel, "Conditional random fields for lidar point cloud classification in complex urban areas," *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. 1, no. 3, pp. 263–268, 2012.

[7] Nesrine Chehata, Li Guo, and Clément Mallet, "Airborne lidar feature selection for urban classification using random forests," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. Part 3, pp. 207–212, 2009.

[8] 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*. ACM, 2015, p. 73.

[9] Aparajithan Sampath and Jie Shan, "Building boundary tracing and regularization from airborne lidar point clouds," *Photogrammetric Engineering & Remote Sensing*, vol. 73, no. 7, pp. 805–812, 2007.

[10] Yuxiang He, C Zhang, and Clive S Fraser, "An energy minimization approach to automated extraction of regular building footprints from airborne lidar data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, pp. 65, 2014.

[11] Yangbin Lin, Cheng Wang, Jun Cheng, Bili Chen, Fukai Jia, Zhonggui Chen, and Jonathan Li, "Line segment extraction for large scale unorganized point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 102, pp. 172–183, 2015.

[12] Yangbin Lin, Cheng Wang, Bili Chen, Dawei Zai, and Jonathan Li, "Facet segmentation-based line segment extraction for large-scale point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1–16, 2017.

[13] Timo Hackel, Jan D Wegner, and Konrad Schindler, "Contour detection in unstructured 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1610–1618.

[14] Mark Pauly, Richard Keiser, and Markus Gross, "Multi-scale Feature Extraction on Point-Sampled Surfaces," *Computer graphics forum*, vol. 22, no. 3, pp. 281–289, 2003.

[15] M.A. Hall, *Correlation-based feature subset selection for machine learning*, Ph.D. thesis, Department of Computer Science, University of Waikato, New Zealand, 1999.

[16] 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*, pp. 179–192. Springer, 2011.

[17] Anh Nguyen and Bac Le, "3d point cloud segmentation: A survey," in *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*. IEEE, 2013, pp. 225–230.

[18] Hermann Gross and Ulrich Thoennessen, "Extraction of lines from laser point clouds," in *Symposium of ISPRS Commission III: Photogrammetric Computer Vision PCV06. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006, vol. 36, pp. 86–91.

[19] A Novacheva, "Building roof reconstruction from lidar data and aerial images through plane extraction and colour edge detection," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 53–57, 2008.

[20] Qian-Yi Zhou and Ulrich Neumann, "Complete residential urban area reconstruction from dense aerial lidar point clouds," *Graphical Models*, vol. 75, no. 3, pp. 118–125, 2013.

[21] Norbert Haala and Martin Kada, "An update on automatic 3d building reconstruction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 570–580, 2010.

[22] Jie Shan and Aparajithan Sampath, "Building extraction from lidar point clouds based on clustering techniques," *Topographic laser ranging and scanning: principles and processing*, pp. 421–444, 2008.

[23] Huan Ni, Xiangguo Lin, Xiaogang Ning, and Jixian Zhang, "Edge detection and feature line tracing in 3d-point clouds by analyzing geometric properties of neighborhoods," *Remote Sensing*, vol. 8, no. 9, pp. 710, 2016.

[24] André Henn, Gerhard Gröger, Viktor Stroh, and Lutz Plümer, "Model driven reconstruction of roofs from sparse lidar point clouds," *ISPRS Journal of photogrammetry and remote sensing*, vol. 76, pp. 17–29, 2013.

[25] Jixing Yan, Wanshou Jiang, and Jie Shan, "Quality analysis on ransac-based roof facets extraction from airborne lidar data," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 1, pp. 367–372, 2012.

[26] George Vosselman, Ben GH Gorte, George Sithole, and Tahir Rabbani, "Recognising structure in laser scanner point clouds," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 46, no. 8, pp. 33–38, 2004.

[27] Vivek Verma, Rakesh Kumar, and Stephen Hsu, "3d building detection and modeling from aerial lidar data," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2213–2220.

[28] Gunho Sohn, Xianfeng Huang, and Vincent Tao, "Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data," *Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 11, pp. 1425–1438, 2008.

[29] S. Xia and R. Wang, "A fast edge extraction method for mobile lidar point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. PP, no. 99, pp. 1–5, 2017.

[30] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3d point cloud data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, pp. 313–318, 2013.

[31] Christopher Weber, Stefanie Hahmann, and Hans Hagen, "Sharp feature detection in point clouds," in *Shape Modeling International Conference (SMI), 2010*. IEEE, 2010, pp. 175–186.

[32] Yani Ioannou, Babak Taati, Robin Harrap, and Michael Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 2012, pp. 501–508.

[33] David Weinstein, Gordon Kindlmann, and Eric Lundberg, "Tensorlines: Advection-diffusion based propagation through diffusion tensor fields," in *Proceedings of the conference on Visualization'99: celebrating ten years*. IEEE Computer Society Press, 1999, pp. 249–253.

[34] Jaya Sreevalsan-Nair and Akshay Jindal, "Using Gradients and Tensor Voting in 3D Local Geometric Descriptors for Feature Detection in Airborne LiDAR Points in Urban Regions," in *(to appear) in Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 1–4, preprint at http://www.iiitb.ac.in/GVCL/pubs.html.

[35] Michael Felsberg and Ullrich Köthe, "GET: The connection between monogenic scale-space and gaussian derivatives," in *International Conference on Scale-Space Theories in Computer Vision*. Springer, 2005, pp. 192–203.

[36] Shengfa Wang, Tingbo Hou, Shuai Li, Zhixun Su, and Hong Qin, "Anisotropic Elliptic PDEs for Feature Classification," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 10, pp. 1606–1618, 2013.

[37] 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*, vol. 38, no. Part 5, pp. 97–102, 2011.

[38] Hans Knutsson, Carl-Fredrik Westin, and Mats Andersson, "Representing local structure using tensors ii," in *Image analysis*, pp. 545–556. Springer, 2011.

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

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

[41] Ullrich Köthe and Michael Felsberg, "Riesz-transforms versus derivatives: On the relationship between the boundary tensor and the energy tensor," in *International Conference on Scale-Space Theories in Computer Vision*. Springer, 2005, pp. 179–191.

[42] Yangming Li and Edwin B Olson, "A general purpose feature extractor for light detection and ranging data," *Sensors*, vol. 10, no. 11, pp. 10356–10375, 2010.

[43] Ben Gorte, "Segmentation of tin-structured surface models," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 4, pp. 465–469, 2002.

[44] William E Lorensen and Harvey E Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*. ACM, 1987, vol. 21, pp. 163–169.

[45] Florent Lafarge and Clément Mallet, "Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation," *International journal of computer vision*, vol. 99, no. 1, pp. 69–85, 2012.

[46] M Gopi, Shankar Krishnan, and Cláudio T Silva, "Surface reconstruction based on lower dimensional localized delaunay triangulation," in *Computer Graphics Forum*. Wiley Online Library, 2000, vol. 19, pp. 467–478.

[47] Aparajithan Sampath and Jie Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on geoscience and remote sensing*, vol. 48, no. 3, pp. 1554–1567, 2010.

[48] Mordechai Haklay and Patrick Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[49] Robert Hecht, Carola Kunze, and Stefan Hahmann, "Measuring completeness of building footprints in openstreetmap over space and time," *ISPRS International Journal of Geo-Information*, vol. 2, no. 4, pp. 1066–1091, 2013.

[50] Andreas Fabri and Sylvain Pion, "Cgal: The computational geometry algorithms library," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 538–539.

[51] Beena Kumari, Avijit Ashe, and Jaya Sreevalsan-Nair, "Remote Interactive Visualization of Parallel Implementation of Structural Feature Extraction of Three-dimensional Lidar Point Cloud," in *Big Data Analytics*, pp. 129–132. Springer, 2014.

[52] Yuxiang He, Chunsun Zhang, and C Fraser, "A line-based spectral clustering method for efficient planar structure extraction from lidar data," *Proceedings of ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Antalya, Turkey*, vol. 1113, pp. 103108, 2013.

[53] Mohammad Awrangjeb and Clive S Fraser, "Automatic segmentation of raw lidar data for extraction of building roofs," *Remote Sensing*, vol. 6, no. 5, pp. 3716–3751, 2014.

[54] Syed Ali Naqi Gilani, Mohammad Awrangjeb, and Guojun Lu, "Robust building roof segmentation using airborne point cloud data," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 859–863.

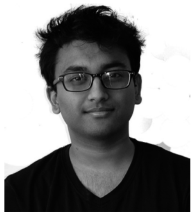
[55] Michael Cramer, "The dgpf-test on digital airborne camera evaluation–overview and test design," *Photogrammetrie-Fernerkundung-Geoinformation*, vol. 2010, no. 2, pp. 73–82, 2010.

[56] Radu Bogdan Rusu and Steve Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[57] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal: Software Tools for the Professional Programmer 25*, pp. 120–123, 2000.

Jaya Sreevalsan-Nair received the B.Tech. degree in aerospace engineering from the Indian Institute of Technology Madras, Chennai, India, in 2000, the M.S. degree in computational engineering from the Mississippi State University, Starkville, MS, USA, in 2002, and the Ph.D. degree in computer science from the University of California, Davis, CA, USA, in 2007.

She is an Associate Professor at the International Institute of Information Technology Bangalore. Her research interests are visual analytics and computational modeling. Her current focus is on the use of matrices and tensors in visual analytics of geospatial datasets and complex networks.



Akshay Jindal received the M.Tech. degree information technology from the International Institute of Information Technology Bangalore, Bangalore, India, in 2017.

He is currently a Software Development Engineer at @WalmartLabs, Bangalore. His research interests include the areas of computer graphics and Artificial Intelligence (AI).

Beena Kumari received the B.Tech. degree in electronics and communication engineering from Shri Mata Vaishno Devi University, Katra, Jammu and Kashmit, in 2010 and the M.S. degree in research in information technology from the International Institute of Information Technology Bangalore, Bangalore, India, in 2016.

She is currently a Computer Vision Engineer at Continental AG. Her research interests include LiDAR data analysis, feature analysis in scientific data, tensor analysis, and object tracking for ADAS.