Surface Denoising based on the Variation of Normals

and

Retinal Shape Analysis

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Sunil Kumar Yadav

Berlin, 2018

This work is dedicated to my mother.

Erstgutachter: Prof. Dr. Konrad Polthier
Zweitgutachter: Prof. Dr. Ligang Liu and Prof. Dr. Friedemann Paul

Tag der Disputation: April 12, 2018

Acknowledgements

I foremost thank my advisor, Prof. Dr. Konrad Polthier, for his continuous support of my research as well as for providing and being part of a uniquely conducive environment that was an important precondition for all my recent years efforts spent on thoroughly exploring the thrilling field of 3D geometry processing. I am particularly grateful to Ulrich Reitebuch for inspiring discussions, constructive and critical remarks. This thesis would not exist without him and I am forever thankful for his invaluable advices and constant support. To all members of the mathematical geometry processing group, I would have not make it this far without your assistance.

I would also like to thank Dr. Alexander U. Brandt, head of the Neurodiagnostic Laboratory for giving me the opportunity to work in this exciting research area and for providing constant support, encouragement and advice throughout this thesis development. I am grateful to Prof. Dr. Friedemann Paul, head of Clinical Neuroimmunology Research Group at Charité University Berlin for sharing his medical knowledge. My sincere thanks also goes to Dr. Ella Maria Kadas and Seyedamirhosein Motamedi, for the very fruitful discussions concerning our work, which helped me significantly in understanding the results, as well as for the support of my scientific activity.

Finally, this thesis would not have been possible without my family and friends who accompanied me during the last few years.

Contents

Table of Contents vii

1 Introduction 1
- 1.1 Main Contributions 2
- 1.2 Related Work in Surface Denoising 4

2 Element-based Normal Voting Tensor (ENVT) 7
- 2.1 The Curvature Tensor 7
- 2.2 Normal Voting Tensor (NVT) 9
- 2.3 Element-based Normal Voting Tensor (ENVT) on a Triangulated Surface 11
- 2.4 Feature Analysis based on the ENVT 13
- 2.5 Curvature Approximation in the ENVT Framework 14
- 2.6 Summary 16

3 ENVT-based Mesh Denoising Algorithm 17
- 3.1 Method 17
  - 3.1.1 Local Binary Neighbor Selection 18
  - 3.1.2 Element-based Normal Voting Tensor (ENVT) 19
  - 3.1.3 Eigenvalues Binary Optimization 20
  - 3.1.4 Denoising using ENVT 20
  - 3.1.5 Effect of Noise on the Proposed Method 23
- 3.2 Experiments, Results and Discussion 25
  - 3.2.1 Parameters Tuning 25
  - 3.2.2 Visual Comparison with State-of-the-art Methods 27
  - 3.2.3 Quantitative Comparison with State-of-the-art Methods 30
  - 3.2.4 Convergence and Running Time Complexity 30
- 3.3 Summary 33

4 Point Set Denoising Algorithm using Vertex-based NVT 35
- 4.1 Method 35
  - 4.1.1 Vertex Normal Filtering 35
  - 4.1.2 Feature Detection 38
  - 4.1.3 Constraint-based Vertex Position Update 39
- 4.2 Experiments, Results and Discussion 41
  - 4.2.1 Parameters Tuning 42
  - 4.2.2 Visual Comparison with State-of-the-art Methods 43
- 4.3 Summary 43

5 RoFi: Robust and High Fidelity Mesh Denoising 45
5.1 Method 45
5.1.1 Robust Bilateral Face Normal Processing 45
5.1.2 High Fidelity Mesh Reconstruction 47
5.2 Experiments, Results and Discussion 49
5.2.1 Visual Comparison 52
5.2.2 Quantitative Comparison 54
5.3 Summary 57

6 Introduction to the Retinal Shape Analysis 59
6.1 Structure of Human Eye 59
6.1.1 The Retina (The Inner Layer) 59
6.1.2 The Fovea 61
6.1.3 The Optic Nerve Head (ONH) 62
6.2 Retinal Imaging using Optical Coherence Tomography 62
6.3 Neurological Disorders and their Effect on Retinal Shape 64
6.3.1 Multiple Sclerosis (MS) 64
6.3.2 Neuromyelitis Optica Spectrum Disorder 65
6.3.3 Ideopathic Intracranial Hypertension (IIH) 65
6.4 Previous Works 65
6.4.1 Fovea 3D Shape Analysis 65
6.4.2 Optical Nerve Head 3D Shape Analysis 67
6.5 Contributions 68

7 CuBe: 3D Shape Analysis of the Fovea 69
7.1 Method Theory 69
7.1.1 Invariant Features of the Fovea 70
7.1.2 Cubic Bézier 70
7.1.3 Least Square Optimization 71
7.2 Materials and Methods 72
7.2.1 Method Pipeline 73
7.3 Parameters 75
7.4 Experiments, Results and Discussion 80
7.4.1 Quantitative Analysis of Volume to Radial Re-sampling 80
7.4.2 Re-test Reliability 81
7.4.3 Model Accuracy 82
7.4.4 Application in HC and Patients with Autoimmune Neuroinflammatory Disorders 84
7.5 Summary 85

8 Optic Nerve Head 3D Shape Analysis using ILM and RPE Manifold Surfaces 87
8.1 Method 88
8.1.1 ILM Surface Computation 88
8.1.2 ILM Surface Smoothing 88
8.1.3 Ellipse Fitting to the BMO Points 89
8.1.4 Correspondence between ILM and RPE Surfaces 90
8.1.5 BMO Region Segmentation 91
8.1.6 3D Shape Parameters 91
8.2 Statistical Shape Analysis of the ILM Surface 95

8.2.1 ONH Surfaces Alignment . . . 95
8.2.2 Mean Shape Reconstruction . . . 96
8.3 Experiments and Results . . . 96
8.3.1 Re-test Reliability . . . 96
8.3.2 Clinical Validation . . . 97
8.4 Summary . . . 97

9 Summary **99**

Bibliography **101**

Chapter 1

Introduction

Recent innovations and developments in 3D acquisition technology, such as 3D laser scanners, magnetic resonance imaging, ultrasound, optical coherence tomography and LiDAR scanning allow fast, robust and highly accurate digitization of complex 3D objects. The analysis and processing of these geometric data are vital in various scientific disciplines and industries, such as neuroscience, mechanical engineering, animation, etc. Discrete differential geometry, a relatively new field in mathematics and computer science that concerns between acquisition and production of a geometrical data. Discrete differential geometry is a rapidly growing research field, equipped with concepts of classical and modern differential geometry of smooth manifolds, mathematical models and algorithms for analyzing and manipulating geometric data. Geometry processing is basically a combination of several mathematical algorithms, which are applied to geometrical models in various real life applications, for example, Figure 1.1 shows a general pipeline for geometry processing.

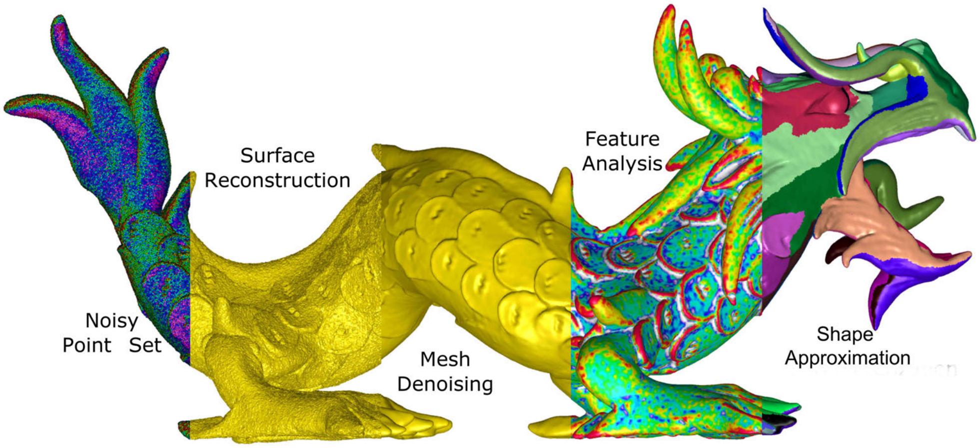


Figure 1.1: A general pipeline for geometry processing.

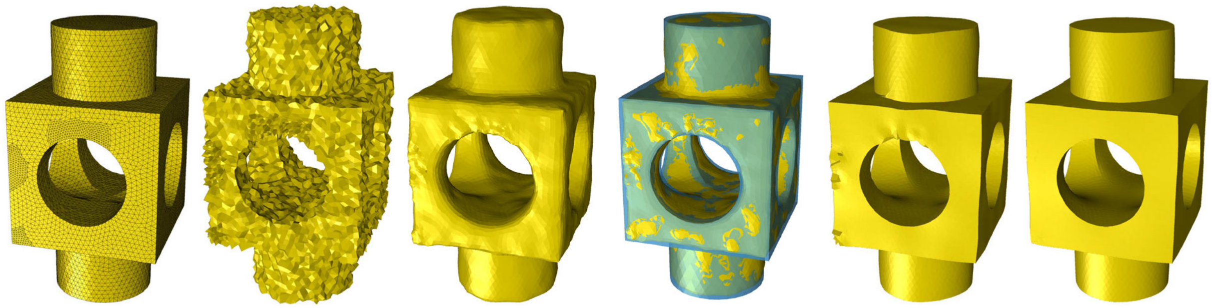
The thesis is divided in two parts. The first part consists of Chapter 2–5. In this part, we introduce a shape analysis operator (Chapter 2), its application to mesh denoising (Chapter 3) and point set denoising (Chapter 4). This thesis also discusses about the robust statistics framework and introduces a robust and high fidelity mesh denoising using a robust estimator (Chapter 5).

In the second part, we apply geometry processing algorithms including the proposed algorithm in the first part of thesis, for the retinal shape analysis. An general introduction about a human eye, the retinal shape and common neurological disorders are given in Chapter 6. Chapter 7 comprises of an algorithm to perform a parametric modeling of the fovea, which is an important region in the retina. The optical nerve head morphometry is performed in Chapter 8 using the proposed geometry processing algorithms.

Here, we give an introduction about surface denoising, general problems in surface denoising and an overview of state-of-the-methods related to these topics.

Surface Denoising: Surface denoising is a central preprocessing tool in discrete geometry processing with many applications in computer graphics such as CAD, reverse engineering, virtual reality and medical imaging. During a surface measurement, noise is inevitable due to various internal and external factors and this degrades surface data quality and its usability. The main goal of a surface denoising algorithm is to remove spurious noise and compute a high quality noise-free surface while preserving sharp features.

In general, noise and sharp features both have high frequency components, so decoupling the sharp features from noise is still a challenging problem in surface denoising algorithms. Traditionally, noise is removed by using a low pass filtering approach, but this operation leads to feature blurring as shown in Figure 1.2. Noise behaviour depends highly on resolution and uniformity of the corresponding mesh. As it can be seen from Figure 1.2(e), feature preservation is a challenging task with non-uniform meshes and also the removal of low frequency noise is not so straight forward (Figure 1.2 (e), top edge of the cylindrical region). Figure 1.2 (f) shows the result obtained by our method [114], which produces a noise-free mesh with sharp features.



(a) Original (b) Noisy (c) Blurring (d) Shrinkage (e) Non-uniform mesh effect (f) Ours [114]

Figure 1.2: General problems in mesh denoising.

Retinal Shape Analysis: In this part of thesis, we introduce couple of geometric processing algorithms for quantification of the retinal shape. As part of the central nervous system, the retina comprises a similar cellular composition as the brain, and many neurological disorders thus affect the retina. Many neuroinflammatory conditions are known to cause modifications to two important regions of the retina: the fovea and the optical nerve head (ONH) and an accurate and robust shape modeling of these regions can diagnose several neurological disorders by detecting the shape changes. A detailed discussion about this part is mentioned in Chapter 6, 7 and 8.

1.1 Main Contributions

The main contributions of this thesis have been divided into two parts: mesh and point set denoising in a proposed normal voting tensor framework and the retinal shape analysis, which includes a parametric modeling of the foveal region and a quantification of the optical nerve head region.

Surface Denoising and Feature Analysis

- We follow the variation of surface normals, which is termed as the normal voting tensor and derive a relation between the shape operator and the normal voting tensor.
- We extend the concept of the directional and the mean curvatures on the dual representation of a triangulated surface using the proposed element-based normal voting tensor (ENVT).
- We propose an ENVT-based mesh denoising algorithm. A binary optimization technique is applied on the spectral components of the ENVT that helps the algorithm to retain sharp features in the concerned geometry and improves the convergence rate of the algorithm.
- We give a stochastic analysis of the effect of noise on a triangulated mesh based on the minimum edge length of elements in the geometry. It gives an upper bound to the noise standard deviation to have minimum probability for flipped element normals.
- We investigate the robust statistics framework for face normal bilateral filtering and propose a robust and high fidelity two-stage mesh denoising method using *Tukey's bi-weight function* as a robust estimator to stop the diffusion at sharp features and to produce smooth umbilical regions.
- We introduce a vertex update scheme, which uses a differential coordinate-based Laplace operator along with an edge-face normal orthogonality constraint to produce a high-quality mesh without face normal flips and this operator also makes the algorithm more robust against high-intensity noise.
- We extend the concept of the ENVT-based mesh denoising for a point set denoising algorithm, where noisy vertex normals are filtered using the vertex-based NVT and the binary optimization.
- For vertex update stage in point set denoising, we add different constraints to the quadratic error metric based on features (edges and corners) or non-feature (planar) points.

Retinal Shape Analysis

- We introduce a parametric modeling of the foveal shape using Cubic Beziér and derive several 3D shape parameters to quantify the foveal shape with high accuracy.
- A 3D shape analysis algorithm is introduced to measure the shape variation in the ONH in different neurological disorders. The proposed method uses manifold surfaces of two different layers of the retina to derive several 3D shape parameters.

Publications Parts of this thesis appear in the following papers:

- ”*Mesh Denoising based on Normal Voting Tensor and Binary Optimization*”, IEEE Transactions on Visualization and Computer Graphics 2017 [114].
  DOI: https://doi.org/10.1109/TVCG.2017.2740384.
- ”*Cube: Parametric Modeling of 3D Foveal Shape using Cubic Bézier*”, Biomedical Optic Express 2017 [112].
  DOI: https://doi.org/10.1364/BOE.8.004181.

- ”*Constraint-based Point Set Denoising using Normal Voting Tensor and Restricted Quadratic Error Metrics*”, Computer and Graphics 2018 (Shape Modeling International 2018) [116].
  DOI = https://doi.org/10.1016/j.cag.2018.05.014 .

- ”*Robust and High Fidelity Mesh Denoising*”, IEEE Transactions on Visualization and Computer Graphics 2018 [115].
  DOI: https://doi.org/10.1109/TVCG.2018.2828818.

- ”*Optic Nerve Head 3D Shape Analysis*”, submitted to IEEE Transactions on Medical Imaging 2018 [113].

1.2 Related Work in Surface Denoising

In last two decades, a wide variety of smoothing algorithms have been introduced to remove undesired noise while preserving sharp features in a geometry. The most common technique for noise reduction is mainly based on the Laplacian on surfaces. We can characterize them as isotropic, anisotropic and two stage smoothing methods. For a comprehensive review on mesh denoising, we refer to couple of survey papers [14–16]. We mention a major part of related works in this section.

Isotropic smoothing methods are the earliest smoothing algorithms. These algorithms have low complexity but suffer from severe shrinkage and further feature blurring [32]. Desbrun et al. [21] introduced an implicit smoothing algorithm that produces stable results for irregular meshes and avoids the shrinkage by volume preservation. Later, the concept of the differential coordinates is introduced by Alexa [4] as a local shape descriptor of a geometry. Differential coordinates preserves the fine details on a triangular mesh, which leads to their further applications in mesh processing, for example, Lipman et al. [63] used the same concept for mesh editing. Mesh quantization [93] and Shape approximation [92] are also done by using the concept of differential coordinates. Sorkine et al. [91] introduced a general differential coordinate-based framework for mesh processing. In continuation, vertex-based Laplacian mesh optimization algorithm was introduced by Nealen et al. for shape optimization and mesh smoothing [72]. Later, Su et. al. exploited the differential coordinates concept for mesh denoising by computing the mean of the differential coordinates at each vertex [94]. This method produces less shrinkage but is unable to preserve shallow features. In general, isotropic smoothing methods are prone to shrink volumes and blur features but effective in noise removal.

Anisotropic diffusion is a PDE based de-noising algorithm introduced by Perona and Malik [82]. The same concept was extended for surface denoising using a diffusion tensor [7, 20]. Similarly, the anisotropic diffusion of surface normals was introduced for surface smoothing [99]. Later, the prescribed mean curvature based surface evolution algorithm was introduced by Klaus et al. [45] that avoids the volume shrinkage and preserves features effectively during the denoising process.

The bilateral filtering was initially proposed by Tomasi et al. [103] for image smoothing. The relation between anisotropic diffusion [45] and the bilateral filtering is explained by Barash [9]. In continuation, Black et al. [13] and Durand et al. [24] expressed anisotropic diffusion and the bilateral filtering in a robust statistics framework [42]. The concept of bilateral smoothing was extended for mesh denoising by Fleishmann et al. [34] and later, Jones et al. [51] explained the same concept in a robust statistics framework. A consistent subneighborhood-based bilateral filtering is introduced by Fan et al. [29]. Solomon et al. proposed a general framework for bilateral and mean shift filtering in any arbitrary domain [90]. These algorithms are simple and effective against noise and feature blurring. In general, anisotropic denoising methods are more robust against volume shrinkage and are better in terms of feature preservation but the algorithm complexity is higher compared to isotropic algorithms.

The two step denoising methods are simple and quite robust against noise. These algorithms consist of two steps: the face normal smoothing and the vertex position update. Face normals are treated as signals on the dual graph of a mesh with values in the unit sphere. The Laplacian smoothing of face normals on a sphere is introduced by Taubin [101] and displacement of the concerned face normal is computed along a geodesic on the unit sphere. The face normal smoothing is done by rotating a face normal on the unit sphere according to the weighted average of neighbor face normals. The bilateral surface normal filtering with the least squares error vertex position update was introduced by Lee et al. [58]. Later, different linear and non-linear weighting functions have been introduced by different algorithms for the face normal smoothing. For example, Yogou et al. [117] computed the mean and the median filtering of face normals to remove noise effect. Later, a modified Gaussian weighting function was applied to the face normal smoothing in an adaptive manner to reduce features blurring [78]. In continuation, the alpha trimming method introduced a non-linear weighting factor which approximates both, the mean and the median filtering [117]. Bilateral normal is one of the most effective and simple algorithms among the two step methods [121], where the weighting function is computed based on the normal differences (similarity measurement) and spatial distances between neighboring faces. Recently, a total variational method has been introduced for the face normal filtering [119]. After the preprocessing of face normals, the vertex position update is done by using the orthogonality between the corresponding edge vector and the face normal [95]. Wang et al. [106] introduced a learning-based mesh denoising algorithm, where the bilateral filtered normal descriptor is used to model the local geometry features. The two step denoising methods are simple in implementation and produce effective results. However, on noisy surfaces, it is difficult to compute the similarity function because of the ambiguity between noise and sharp features, and that leads to unsatisfactory results.

In recent mesh denoising methods, compressed sensing techniques are involved to preserve sharp features precisely and remove noise effectively [111]. For example, L_0 mesh denoising method assumed that features are sparse on a general surfaces and introduced an area based differential operator. This method utilizes L_0 optimization to maximize flat regions on a noisy surfaces to remove noise [43]. The L_0 method is effective against high noise but produces piecewise flat areas on smooth surfaces. Later, the weighted L_1-analysis compressed sensing optimization is applied to recover sharp features from the residual data after global Laplacian smoothing [107]. Recently, the ROF-based (Rudin, Osher and Fatemi) algorithm has been introduced in [110]. This method applies L_1 optimization on both data fidelity and regularization to remove noise without volume shrinkage. In continuation, Lu et al. utilized the L_1-median normal filtering along with vertex preprocessing to produce a noise-free surface [66]. Recently, Yadav et al. [114] proposed a binary optimization-based mesh denoising algorithm, where noise was removed by assigning a binary values to a face normal-based covariance matrix. Centin et al. [16] proposed a normal-diffusion-based mesh denoising algorithm, which removes noise components without tempering the metric quality of a surface. In general, the compressed sensing based denoising algorithms are robust against high intensity noise and recover not only sharp features but also shallow features, but at the same time these algorithms produce false features (piecewise flat areas) on smooth geometries.

A multistage denoising framework was introduced by Bian et al. [12], where feature identification is done by the eigenalysis of the NVT (normal voting tensor) [55] [70] and then geometry is divided into different clusters based on features and smoothing is applied on different clusters independently. Later, the guided mesh normals are computed based on consistence normal orientations and the bilateral filtering is applied [120]. Recently, Wei et al. [108] exploited both vertex and face normal information for feature classification and surface smoothing. In continuation, features detection is done on the noisy surface using quadratic optimization and remove noise using L_1 optimization while preserving features [67]. Multistage denoising algorithms produce effective results against different levels of noise but have higher algorithm complexity because of the different stages.

Similar to mesh denoising, point sets denoising is also studied by several researchers in last two decades [10]. The moving least squares was introduced by Alexa et al. [5] to up-sample a point set in local tangent spaces. Later, a parametrization free, local optimal projector is introduced to remove noise and outliers from point sets using the L_1-median [62]. LOP concept was modified by Huang et al. by adding the weighting term, which makes the algorithm robust against non-homogeneous point density [48]. Algebraic Point Set Surfaces (APSS) algorithm used local moving least squares to reconstruct a smooth surface from a point set [40]. In continuation, RIMLS (robust implicit moving least squares) was introduced by Öztireli et al. and it preserves sharp features by applying bilateral normal filtering before projection [80]. Lange and Polthier [57] introduced the anisotropic smoothing of point sets using the shape operator and principal curvatures. In terms of feature preservation, L_0 point set denoising is introduced by Sun et al. [97] and this algorithm recovers sharp features by computing piecewise flat areas on a point set surface. In continuation, Mattei et al. [68] introduced MRPCA (moving robust PCA) to remove outliers effectively and retain sharp features. Recently, a guided point set denoising algorithm is introduced. This method exploited L_1-median skeleton to extract the feature points and then computed guidance normal by using consistent normals [122].

Chapter 2

Element-based Normal Voting Tensor (ENVT)

On a polygonal mesh surface, estimation of curvature informations including, principal curvatures is quite important for various geometry processing algorithms, for example, shape analysis, mesh segmentation and surface smoothing. In the past, several shape analysis operators have been introduced for feature detection and basic geometry processing operations, for example, the curvature tensor, which was introduced by Taubin [100] and it computes the variation of tangent vectors along a curve on a triangulated surface. The spectral analysis of the curvature tensor approximates the principal curvatures of polygonal surfaces. We follow similar concept and introduce the variation of surface normals termed as the normal voting tensor, and derive a relation between the shape operator and the normal voting tensor. We extend the concept of the directional and the mean curvatures on the dual representation of a triangulated surface. A normal voting tensor is defined on each triangle of a geometry and termed as the element-based normal voting tensor (ENVT). Later, a deformation tensor consisting of the anisotropy of a surface is extracted from the ENVT, and the mean curvature vector is defined based on this ENVT deformation tensor. We also show a similarity of the ENVT with the principal directions and the principal curvatures. In the end of this chapter, we experimented the capability of the ENVT in terms of feature preservation.

2.1 The Curvature Tensor

The curvature notion has a central importance in classical differential geometry and has several concepts and definitions. Here, we explain the concept of directional (normal) curvature using the classical Meusnier's theorem.

Theorem 1. *(Meusnier's theorem) Let $\mathcal{M} \subset \mathbb{R}^3$ be a orientable regular surface with unit normal field $\mathbf{N} : \mathcal{M} \rightarrow \mathbb{R}^3$ and second fundamental form $\mathbf{II} : \mathbf{T}_p\mathcal{M} \times \mathbf{T}_p\mathcal{M} \rightarrow \mathbb{R}$, where $\mathbf{T}_p\mathcal{M}$ is the tangent plane at a point $p \in \mathcal{M}$ spanned by principal curvature directions $\mathbf{T}_1$ and $\mathbf{T}_2$ as the orthonormal basis. Let $c : (-\epsilon, \epsilon) \rightarrow \mathcal{M}$ be a curve parametrized by arc-length with $c(0) = p$, then the directional (normal) curvature of c can be expressed as:*

$$\kappa_p = \mathbf{II}(\dot{c}(0), \dot{c}(0)). \tag{2.1}$$

Let us consider an arbitrary vector $\mathbf{T}_\theta \in \mathbf{T}_p\mathcal{M}$, which can be expressed in terms of the orthonormal basis $\mathbf{T}_1$ and $\mathbf{T}_2$:

$$\mathbf{T}_\theta = \{cos\theta\mathbf{T}_1 + sin\theta\mathbf{T}_2\}, \quad \theta \in \mathbb{R}, \tag{2.2}$$

Let us assume that the curve c is aligned with the tangent vector $\mathbf{T}_\theta$, then $\mathbf{T}_\theta = \dot{c}(0)$. By differential geometry, the shape operator has a diagonal form and because of the orthogonality of the curvature line parametrization, the second fundamental form computes the directional curvature w.r.t. $\mathbf{T}_\theta \in \mathbf{T}_p\mathcal{M}$, and this directional curvature can also be expressed in a quadratic form:

$$\kappa_p(\mathbf{T}_\theta) = \mathbf{II}(\mathbf{T}_\theta, \mathbf{T}_\theta) = \begin{pmatrix} cos\theta & sin\theta \end{pmatrix} \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix}. \tag{2.3}$$

Substituting the value of $\mathbf{T}_\theta$ from Equation (2.2) into second fundamental form gives us the Euler formula for the directional curvature:

$$\kappa_p(\mathbf{T}_\theta) = \mathbf{II}(\mathbf{T}_\theta, \mathbf{T}_\theta) = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta. \tag{2.4}$$

Using the directional curvature from Equation (2.4) and an arbitrary tangent vector $\mathbf{T}_\theta$, Taubin [100] introduced the curvature tensor at point p on a surface $\mathcal{M}$, which is defined as:

$$\mathbf{M}_p = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \kappa_p(\mathbf{T}_\theta) \mathbf{T}_\theta \mathbf{T}_\theta^t d\theta, \tag{2.5}$$

where $\mathbf{T}_\theta$ is a column vector. The curvature tensor $\mathbf{M}_p$ is a covariance matrix of a set of tangent vectors, where each tangent vector is weighted by the corresponding directional curvature. The curvature tensor is rewritten in following form using Equations (2.2) and (2.4):

$$\mathbf{M}_p = \frac{1}{2\pi} \int_{-\pi}^{+\pi} (\kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta)(cos\theta \mathbf{T}_1 + sin\theta \mathbf{T}_2)(cos\theta \mathbf{T}_1 + sin\theta \mathbf{T}_2)^t d\theta. \tag{2.6}$$

The term $\mathbf{M}_p$ is a 3×3 matrix and we can decompose this tensor in terms of the orthonormal basis $\mathbf{T}_1$ and $\mathbf{T}_2$ of the tangent plane $\mathbf{T}_p\mathcal{M}$:

$$\mathbf{M}_p = \begin{pmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{pmatrix} \begin{pmatrix} \sigma_2 & 0 \\ 0 & \sigma_3 \end{pmatrix} \begin{pmatrix} \mathbf{T}_1^t \\ \mathbf{T}_2^t \end{pmatrix}, \tag{2.7}$$

where $\mathbf{T}_1$ and $\mathbf{T}_2$ are the column vectors, thus, $(\mathbf{T}_1, \mathbf{T}_2)$ will be a 3×2 matrix. The non-diagonal components of the eigenvalues matrix (2×2, the central matrix) are zero. Using Equation (2.6), the eigenvalues (σ_2 and σ_3) of the tensor $\mathbf{M}_p$ are computed as:

$$\sigma_2 = \mathbf{T}_1^t \mathbf{M}_p \mathbf{T}_1 = \frac{3}{8}\kappa_1 + \frac{1}{8}\kappa_2, \quad \sigma_3 = \mathbf{T}_2^t \mathbf{M}_p \mathbf{T}_2 = \frac{1}{8}\kappa_1 + \frac{3}{8}\kappa_2. \tag{2.8}$$

The above equation shows a relationship between the principal curvatures and the eigenvalues of the curvature tensor. The mean curvature is defined as the arithmetic mean of the principal curvatures.

$$\kappa_1 = 3\sigma_2 - \sigma_3, \quad \kappa_2 = 3\sigma_3 - \sigma_2, \quad H = \frac{\kappa_1 + \kappa_2}{2} = \sigma_2 + \sigma_3. \tag{2.9}$$

The eigenvectors of $\mathbf{M}_p$ are aligned with the principal directions but its eigenvalues are not approximating the principal curvatures, for example, on a cylinder both of these eigenvalues are non-zero but the minimum principal curvature is zero. The eigenvalues σ_2 and σ_3 can have negative values also and approximate the mean curvature effectively.

Let us consider, for a particular value of θ in the radial direction, point q is lying on the same surface $\mathcal{M}$ in the neighbourhood of p. Let us consider an arc-length parametrized smooth curve c between p and q such that the tangent direction $\mathbf{T}_\theta = \dot{c}(0)$ and the directional curvature $\kappa_p(\mathbf{T}_\theta) = \ddot{c}(0)\mathbf{n}_p$, where $\mathbf{n}_p \in \mathbf{N}$ is the surface normal at point p then, by using Taylor series expansion the directional curvature from p to q can be approximated as:

$$\kappa_p(\mathbf{T}_\theta) = \frac{2\langle \mathbf{n}_p, (q-p) \rangle}{|q-p|}. \tag{2.10}$$

Vertex-based Directional and Mean Curvatures on a Triangulated Surface

In the discrete setting, let us consider a triangulated surface $\mathcal{M} = (\mathbf{V}, \mathbf{F})$, which is a combination of a set of vertices and a set of elements. For a vertex $\mathbf{v}_i \in \mathbf{V}$, the directional curvature w.r.t. $\mathbf{v}_j \in \mathbf{N}_i$, and mean curvature can be defined as:

$$\kappa_{ij} = \frac{2\langle \mathbf{n}_{v_i}, (\mathbf{v}_j - \mathbf{v}_i)\rangle}{|\mathbf{v}_j - \mathbf{v}_i|}, \quad H = \frac{1}{\sum_{j \in \mathbf{N}_i} W_{ij}} \sum_{j \in \Omega_i} W_{ij} \kappa_{ij}, \tag{2.11}$$

where $\mathbf{N}_i$ is the set of neighbour vertices of $\mathbf{v}_i$. The term $\mathbf{H}$ is mean curvature and W_{ij} is a weighting factor associated with each neighbor vertex. In general, this weighting factor is chosen on the basis of an area of elements connected with the vertex $\mathbf{v}_i$. The corresponding tangent plane is computed using the vertex normal $\mathbf{n}_{v_i}$.

2.2 Normal Voting Tensor (NVT)

In general, covariance matrices compute a variation of an entity in a well defined domain, for example, principal component analysis (PCA), where the covariance matrix of an edge vector on a triangular surface computes the variation of vertices in $\mathbb{R}^3$ space. The curvature tensor $\mathbf{M}_p$ is a weighted covariance matrix, which computes the variation of tangent vectors around a point.

In general, surface normals carries anisotropy of a surface hence a covariance matrix of surface normals in a well defined neighborhood computes anisotropic nature of a surface in that region.

To analyze anisotropy of an orientable surface $\mathcal{M} \subset \mathbb{R}^3$, which has well defined normal field $\mathbf{N} : \mathcal{M} \to \mathbb{R}^3$, for any $\Omega \subset \mathcal{M}$, we define a bilinear form $\mathbf{C}_\Omega : \Omega \times \Omega \to \mathbb{R}$. In most general term, we have a map: $\Omega \mapsto \mathbf{C}_\Omega$, where Ω is a geodesic disk of radius r. Now, we define the term $\mathbf{C}_\Omega$ in terms of the tensor product (outer product) of surface normals:

$$\mathbf{C}_\Omega = \frac{1}{|\Omega|} \int_\Omega \mathbf{n} \otimes \mathbf{n} d\Omega, \tag{2.12}$$

where $\mathbf{n} \in \mathbf{N}$ is a column vector and is an order-1 tensor. The symbol $\otimes$ represents the outer product, which can be written as $\mathbf{n} \otimes \mathbf{n} = \mathbf{n}\mathbf{n}^t$. The term $\mathbf{C}_\Omega$ is known as the normal voting tensor. The above equation shows a special case of the Kronecker product of matrices and possesses the following properties:

1. The inner product of normal vectors can be computed using the trace of the outer product.

$$\frac{1}{|\Omega|} \int_\Omega \langle \mathbf{n}, \mathbf{n}^t \rangle d\Omega = \text{Trace}(\mathbf{C}_\Omega). \tag{2.13}$$

2. If the surface normal $\mathbf{n}$ is a non-zero vector then $\mathbf{C}_\Omega$ has matrix rank 1 because for any vector $\mathbf{x} \in \mathbb{R}^3$:

$$\mathbf{C}_\Omega \mathbf{x} = \frac{1}{|\Omega|} \int_\Omega \mathbf{n}(\mathbf{n}^t \mathbf{x}) d\Omega, \tag{2.14}$$

   where $(\mathbf{n}^t \mathbf{x})$ is a scalar quantity.

3. As the surface normal $\mathbf{n}$ is an order-1 tensor so the term $\mathbf{C}_\Omega$ will be an order-2 tensor.

As it is mentioned in Equation (2.12), $\mathbf{C}_\Omega$ is a covariance matrix of surface normals so the normal voting tensor computes the variation of surface normals in Ω geodesic disk. To understand this variation, we follow a theorem by Rodriguez from classical differential geometry.

Theorem 2. *(Rodriguez's theorem) Let* $\mathcal{M} \subset \mathbb{R}^3$ *be a orientable regular surface and* $\mathbf{n}$ *belong to a unit normal field* $\mathbf{N} : \mathcal{M} \rightarrow \mathbb{R}^3$*. Let* $c : I \rightarrow \mathcal{M}$ *be a curve parametrized by arc-length, then* c *is a curvature line on* $\mathcal{M}$ *if and only if there exists a function* $\lambda : I \rightarrow \mathbb{R}$ *with:*

$$\frac{d}{dt}\mathbf{n}(c(t)) = \lambda(t) \cdot \dot{c}(t), \quad t \in I, \tag{2.15}$$

where $-\lambda(t)$ *is the corresponding principal curvature.*

From Rodriguez's theorem, it is clear that the variation of normals depends on mainly two factors: tangent direction $\dot{c}(t)$ and corresponding principal curvature. A similar conclusion, we get from Weingarten map $\mathbf{W}_p : \mathbf{T}_p\mathcal{M} \rightarrow \mathbf{T}_p\mathcal{M}$ as well:

$$\mathbf{W}_p(\mathbf{T}_i) = \kappa_i \cdot \mathbf{T}_i, \quad i = 1, 2. \tag{2.16}$$

From Equations (2.15) and (2.16), it is clear that the variation of surface normals is well approximated in the principal directions using the principal curvatures.

However, it is not so clear in arbitrary directions. For $\theta \in [-\pi, \pi)$, let $c_\theta : (-\epsilon, \epsilon) \rightarrow \Omega$ be a unit speed geodesic along which we study the change of the normal vector. Using Taylor expansion:

$$\mathbf{n}(c_\theta(t)) = \mathbf{n}(c_\theta(0)) + t\mathbf{n}(c_\theta(0))' + \mathcal{O}(t^2), \tag{2.17}$$

where the term $\mathbf{n}(c_\theta(0))'$ is similar to Equation (2.15) and represents the corresponding tangent vector, which is defined in the local neighborhood of point p:

$$\tilde{\mathbf{T}}_\theta = \kappa_1 cos\theta \mathbf{T}_1 + \kappa_2 sin\theta \mathbf{T}_2. \tag{2.18}$$

In the above equation, we added the principal curvatures κ_1, κ_2 as weighting factors in the principal directions $\mathbf{T}_1, \mathbf{T}_2$ to create an anisotropic radial sampling. The tangent vector $\tilde{\mathbf{T}}_\theta$ is spanned by the principal curvature directions and is closer to the maximum principal direction than the minimum principal direction.

The term $\mathbf{n}(c_\theta(0))'$ (Equation (2.17)) can be replaced with $\tilde{\mathbf{T}}_\theta$ (Equation (2.18)), which is the combination of the principal directions and curvatures. The radius term r defines the neighborhood area and as $r \rightarrow 0$, the surface normal $\mathbf{n}(c_\theta(0)) = \mathbf{n}_0$. For a small disk radius the higher order component will be close to zero $\mathcal{O}(t^2) \approx 0$ and the surface normal can be written as a linear combination of $\mathbf{n}_0$ and its corresponding tangent plane $\tilde{\mathbf{T}}_\theta$.

$$\mathbf{n}(c_\theta(t)) = \mathbf{n}_0 + t\tilde{\mathbf{T}}_\theta. \tag{2.19}$$

In other words, when we move in arbitrary direction $\mathbf{T}_\theta$ as mentioned in Equation (2.2), the surface normal changes by $\tilde{\mathbf{T}}_\theta$ in first order approximation. From Equations (2.12) and (2.19), we express the NVT in the following integral equation in a polar coordinate form around a point p:

$$\mathbf{C}_\Omega = \frac{1}{\pi r^2} \int_{-\pi}^{\pi} \int_0^r (\mathbf{n}_0 + t\tilde{\mathbf{T}}_\theta) \otimes (\mathbf{n}_0 + t\tilde{\mathbf{T}}_\theta) t dt d\theta, \tag{2.20}$$

where geodesic disk area $|\Omega| = \pi r^2$. Using Equations (2.18) and (2.20):

$$\mathbf{C}_\Omega = \mathbf{n}_0 \otimes \mathbf{n}_0 + \frac{r^2}{4\pi} \int_{-\pi}^{\pi} (\kappa_1^2 cos^2\theta \mathbf{T}_1 \otimes \mathbf{T}_1 + \kappa_2^2 sin^2\theta \mathbf{T}_2 \otimes \mathbf{T}_2) d\theta. \tag{2.21}$$

So the ENVT can be decomposed into two different covariance matrices:

$$\mathbf{C}_\Omega = \mathbf{C}_0 + \frac{r^2}{2}\mathbf{C}_\kappa, \tag{2.22}$$

where $\mathbf{C}_0$ is the covariance matrix regarding $\mathbf{n}_0$ (when $r \to 0$) and has only one dominant eigenvalue in the surface normal direction. The second term $\mathbf{C}_\kappa$ consists of the variation of surface normals excluding $\mathbf{n}_0$ within disk Ω. The NVT $\mathbf{C}_\Omega$ is a symmetric positive semi-definite matrix and can be written in terms of orthonormal basis:

$$\mathbf{C}_\Omega = \begin{pmatrix} \mathbf{n}_0 & \mathbf{T}_1 & \mathbf{T}_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \mathbf{n}_0 \\ \mathbf{T}_1 \\ \mathbf{T}_2 \end{pmatrix}, \tag{2.23}$$

and the most dominant eigenvalue along the surface normal is computed using the following equation:

$$\lambda_1 = \mathbf{n}_0^t \mathbf{C}_\Omega \mathbf{n}_0 = 1. \tag{2.24}$$

In this case, $\mathbf{C}_\kappa$ will be a zero covariance matrix because it has zero eigenvalue along the surface normal direction and non-zero eigenvalues along the principal directions, which are perpendicular to $\mathbf{n}_0$. The other two eigenvalues along the principal directions can be computed as:

$$\lambda_2 = \mathbf{T}_1^t \mathbf{C}_\Omega \mathbf{T}_1 = \frac{r^2}{2} \mathbf{T}_1^t \mathbf{C}_\kappa \mathbf{T}_1 = \frac{r^2}{2} \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_1^2 cos^2\theta d\theta = \frac{r^2 \kappa_1^2}{4}, \tag{2.25}$$

$$\lambda_3 = \mathbf{T}_2^t \mathbf{C}_\Omega \mathbf{T}_2 = \frac{r^2}{2} \mathbf{T}_2^t \mathbf{C}_\kappa \mathbf{T}_2 = \frac{r^2}{2} \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_2^2 sin^2\theta d\theta = \frac{r^2 \kappa_2^2}{4}, \tag{2.26}$$

From the above equations, it is clear that second and third eigenvalues of the NVT approximate squares of the principal curvatures κ_1 and κ_2. The eigenvalues (σ_2, σ_3) of the curvature tensor are a linear combination of the principle curvatures (κ_1, κ_2). Considering the same example of Cylinder, unlike σ_3, which is non-zero, the least dominant eigenvalue λ_3 of the NVT will be zero. Therefore, the NVT approximates the shape operator better compared to the curvature tensor.

2.3 Element-based Normal Voting Tensor (ENVT) on a Triangulated Surface

The concept of normal voting was introduced by Medioni [70] and was used in various computer vision applications. The basic idea of normal voting tensor is to select the neighborhood of a vertex and these neighborhood vertices vote to estimate feature points on a surface.

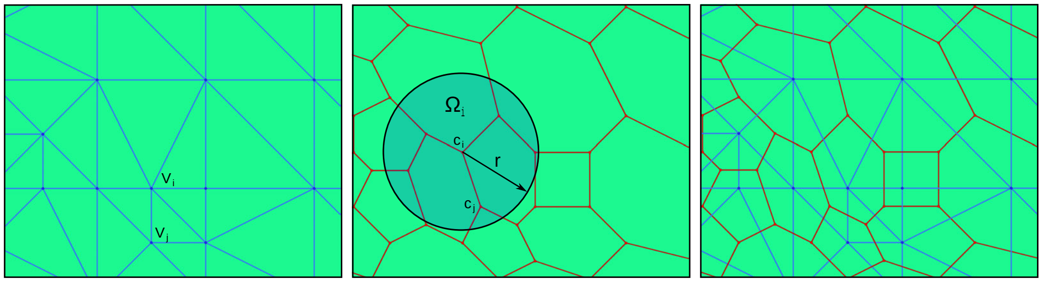
As face normals are a first order information on a surface and give better description about sharp features than vertex normals because the angles between face normals are bigger compared to the angles between vertex normals in those regions. We extended the normal voting concept and define an element-based normal voting tensor on every element of a properly oriented triangulated mesh. Similar to Section 2.2, the element-based normal voting tensor is the weighted average of the outer product of the neighbor element normals. On the geometrical neighborhood Ω_i of face $\mathbf{f}_i$ (Figure 2.1), the ENVT is defined as:

$$\mathbf{C}_{\Omega_i} = \frac{1}{\sum_{j \in \Omega_i} w_{ij} A_j} \sum_{j \in \Omega_i} w_{ij} A_j \mathbf{n}_j \otimes \mathbf{n}_j, \tag{2.27}$$

where A_j is the area of $\mathbf{f}_j$ and w_{ij} is a Gaussian-based weighting function. Weighting by corresponding element area makes the proposed tensor more robust against irregular sampling. The element based normal voting tensor is a symmetric and positive semi definite matrix so, we can represent $\mathbf{C}_{\Omega_i}$ using a rotational and a scaling matrix:

$$\mathbf{C}_{\Omega_i} = R\Lambda R^t = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{pmatrix}, \tag{2.28}$$

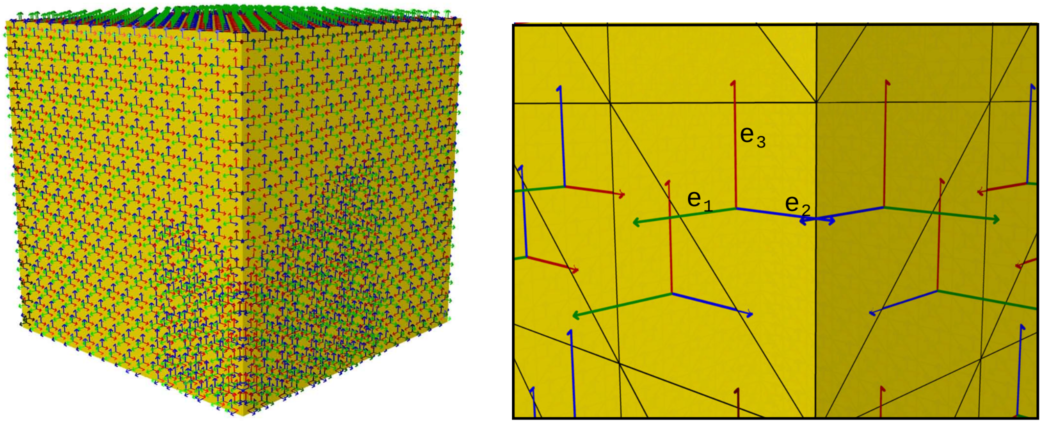
where R and Λ are the rotational and the scalar matrix respectively. The terms $\mathbf{e}_i$ and λ_i are spectral components of the ENVT.



(a) Triangulate mesh (b) Dual Mesh (c) Both

Figure 2.1: A visual representation of primal and dual mesh. (a) Original mesh $\mathcal{M}$, where the vertex $\mathbf{v}_j \in \mathbf{N}_i$ belongs to the 1-ring neighborhood of vertex $\mathbf{v}_i$. (b) Dual polyhedral mesh $\mathcal{M}_d$, which is computed from triangulated mesh $\mathcal{M}$ using Equation (2.31). The geometrical neighborhood is represented by Ω_i, which is a disk of radius r on the dual mesh. (c) Both triangulated and corresponding dual mesh.

As it can be seen from Equation (2.28), the eigenvectors $\mathbf{e}_2$ and $\mathbf{e}_3$ are the basis of a tangent space and align with the principal directions ($\mathbf{T}_1 = \mathbf{e}_2$, $\mathbf{T}_2 = \mathbf{e}_3$). The eigenvector $\mathbf{e}_1$ aligns with the surface normal at that point as shown in Figure 2.2.



(a) Eigenvalues of $\mathbf{C}$ and $\mathbf{B}$ (b) Magnified view

Figure 2.2: The eigenvectors visualization on a Cube model. (a) The eigenvectors are painted on Cube, $\mathbf{e}_1$ in green, $\mathbf{e}_2$ in blue and $\mathbf{e}_3$ in red for both the ENVT and the deformation tensor. (b) The least dominant eigenvector $\mathbf{e}_3$ is aligned with the edge direction, which is the minimum principal direction.

2.4 Feature Analysis based on the ENVT

The ENVT primarily consist of the anisotropy of a triangulated surface and can be written in terms of its spectral components:

$$\mathbf{C}_{\Omega_i} = (\lambda_1 - \lambda_2)\mathbf{e}_1 \otimes \mathbf{e}_1 + (\lambda_2 - \lambda_3)(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2) + \lambda_3(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2 + \mathbf{e}_3 \otimes \mathbf{e}_3), \quad (2.29)$$

where the eigenvalues are sorted in decreasing order: $(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0)$. As shown in Figure 2.3, each term of Equation (2.29) can be defined as the following tensors:

1. The term $(\lambda_1 - \lambda_2)\mathbf{e}_1 \otimes \mathbf{e}_1$ is defined as the stick tensor because this tensor has a stick like shape in the surface normal direction.
2. The term $(\lambda_2 - \lambda_3)(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2)$ is defined as the plate tensor because the two eigenvectors are spanning a plate like shape and the least dominant direction will be normal to this plate like shape.
3. The term $\lambda_3(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2 + \mathbf{e}_3 \otimes \mathbf{e}_3)$ is defined as the ball tensor because this tensor is spanned by all three eigenvectors.

Based on these tensors, we can define the saliency of features on a surface. The stick tensor represents the surface saliency (SMap) where the surface normal is in the direction of the most dominant eigenvector and $(\lambda_1 - \lambda_2)$ shows the strength in that direction. Similarly, the plate tensor is the so-called curve map (CMap) and has two dominant eigenvector. The ball tensor basically represents the intersection of more than two planes and has no well defined direction either for surface normal or for surface tangent and is so-called junction map (JMap) [70].

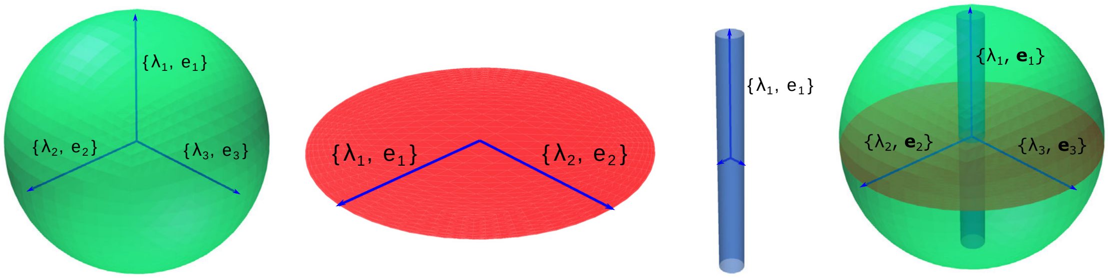
On a noise free triangulated mesh, a planar area has only one dominant eigenvalue in surface normal direction (SMap) and the least dominant eigenvectors represent an orthonormal basis for the tangent plane. Two dominant eigenvalues indicate edge features (CMap), where the weakest eigenvector will be along the edge direction and dominant eigenvalues are in the direction of the surface normal and across the edge respectively. Therefore, the tangent space will be spanned by significant and insignificant directions. At a corner all three eigenvalues are dominant (JMap) and it is not trivial to define a tangent plane. Let us consider the Cube model, and compute the ENVT using Equation (2.28). The eigenvalues vector is sorted and normalized, then for orthogonal features, we can write: $\{\lambda_1 = 1, \lambda_2 = \lambda_3 = 0\}$ (face), $\{\lambda_1 = \lambda_2 = \frac{\sqrt{2}}{2}, \lambda_3 = 0\}$ (edge) and $\{\lambda_1 = \lambda_2 = \lambda_3 = \frac{\sqrt{3}}{3}\}$ (corner). Let us define a new eigenvalues vector $\lambda_p = \{1, \frac{\lambda_2}{\lambda_1}, \frac{\lambda_3}{\lambda_1}\}$ and categorize the following features points:

1. Significant value of $\frac{\lambda_3}{\lambda_1}$ indicates a corner point.
2. Significant value of $\frac{\lambda_2}{\lambda_1}$ indicates a edge.
3. On planar areas, $\frac{\lambda_2}{\lambda_1}$ and $\frac{\lambda_3}{\lambda_1}$, both are insignificant.

Using the above classification, we can define the feature strength value:

$$s_f = \lambda_2 + \lambda_3. \quad (2.30)$$

Figure 2.4(c) shows a visual representation of the feature strength on a smooth Fandisk model. As it can be seen, the ENVT identifies sharp features effectively.



(a) JMap (b) CMap (c) SMap (d) Together

Figure 2.3: A visual representation of different kinds of tensors mentioned in Equation (2.29).

2.5 Curvature Approximation in the ENVT Framework

In the discrete setting, the curvature tensor is well defined for vertices on a triangulated mesh and it approximates the directional and the mean curvatures with high fidelity. Here, we exploit a similar framework to define the directional and the mean curvatures on elements (faces) of a surface. Firstly, we compute a barycentric dual mesh $\mathcal{M}_d$ of the original triangulated surface $\mathcal{M}$. The dual mesh $\mathcal{M}_d$ will be a polyhedral surface and consists of vertices and polygon elements. For the dual mesh $\mathcal{M}_d$, the set of vertices and elements are represented as $\mathbf{V}_d$ and $\mathbf{F}_d$ respectively. Each vertex of the dual mesh has a fixed connectivity of 3 and can be computed as:

$$\mathbf{c}_i = \frac{1}{3}\sum_{\mathbf{v}_j \in \mathbf{f}_i} \mathbf{v}_j, \quad \text{where} \quad \mathbf{f}_i \in \mathbf{F}, \quad \mathbf{v}_j \in \mathbf{V}. \tag{2.31}$$

Note that elements in the dual mesh will be n-sided polygon and number of sides in polygons depends on vertices connectivity in the original triangulated surface as shown in Figure 2.1. Each vertex $\mathbf{c}_i$ of the dual mesh has a well defined vertex normal $\mathbf{n}_i$, which is basically the face normal of the element $\mathbf{f}_i \in \mathbf{F}$ in the original triangulated surface. Patanè et al. [81] discussed a homeomorphism between 2-manifold tringulated mesh and its dual representation and the several combinatorial properties of a dual mesh, which include 1-neighborhood analysis, triangle mesh reconstruction, primal-dual correspondence and dual Laplacian smoothing. The dual transformation is also genus invariant for polygonal mesh. Similar to Equation (2.11), the directional and the mean curvatures on the dual mesh $\mathcal{M}_d$ is defined as:

$$\kappa_{ij}^d = \frac{2\langle \mathbf{n}_i, (\mathbf{c}_j - \mathbf{c}_i)\rangle}{|\mathbf{c}_j - \mathbf{c}_i|}, \quad \mathbf{H}^d = \frac{1}{\sum_{j\in\Omega_i} W_{ij}} \sum_{j\in\Omega_i} W_{ij}\kappa_{ij}^d, \tag{2.32}$$

where $W_{ij} = A_j$ is the area of $\mathbf{f}_i \in \mathbf{F}$. To compute the direction curvature κ_{ij}^d, we need to compute the tangent plane $\mathbf{T}_{ij}$, which is basically the triangle plane of $\mathbf{f}_i \in \mathbf{F}$ and is computed using the face normal $\mathbf{n}_i$. The area weighting will improve the accuracy of curvatures computation as big area elements have trustworthy normals.

The ENVT $\mathbf{C}$ is a positive semi-definite symmetric matrix and can be expressed in a form of the deformation tensor $\mathbf{B}$:

$$\mathbf{C} = \mathbf{B}\mathbf{B}^t \quad \text{s.t.} \quad \det(\mathbf{B}) > 0. \tag{2.33}$$

From Equations (2.27) and (2.33), the deformation tensor is defined in terms of the rotation and the scaling matrix:

$$\mathbf{C} = \mathbf{R}\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\mathbf{R}^{\mathbf{t}} = \mathbf{B}\mathbf{B}^t, \text{and} \quad \mathbf{B} = \mathbf{R}\Lambda^{\frac{1}{2}}. \tag{2.34}$$

Using the above representation of the ENVT, for a given vector $\mathbf{u}$, we can express the ENVT in the following quadratic form :

$$\mathbf{u}^t\mathbf{C}\mathbf{u} = \mathbf{u}^t\mathbf{B}\mathbf{B}^t\mathbf{u} = (\mathbf{B}^t\mathbf{u})^t(\mathbf{B}^t\mathbf{u}) = \|\mathbf{B}^t\mathbf{u}\|^2. \tag{2.35}$$

By using the above representation of the ENVT, similar to Tsuchie et al. [104], a distance between central vertex $\mathbf{c}_i$ and neighbor vertex $\mathbf{c}_j$ on the dual mesh is defined as:

$$d_{ij} = \sqrt{(\mathbf{c}_j - \mathbf{c}_i)\mathbf{C}(\mathbf{c}_j - \mathbf{c}_i)^t} = \|\mathbf{B}^t(\mathbf{c}_j - \mathbf{c}_i)\|, \tag{2.36}$$

where the parameter d_{ij} possesses the following properties:

1. In general, $d_{ij} \neq d_{ji}$.
2. If $\mathbf{c}_i$ and $\mathbf{c}_j$ have the same tangent plane then $d_{ij} = 0$, because d_{ij} compute a distance in the surface normal direction.
3. If the ENVT is an identity matrix then, d_{ij} represent an Euclidean distance in $\mathbb{R}^3$.

The deformation tensor $\mathbf{B}$ carries anisotropy nature of a surface and can be used to approximate the direction curvature along with d_{ij}:

$$\bar{\kappa}^d_{ij} = \frac{\sqrt{2}\mathbf{B}^t(\mathbf{c}_j - \mathbf{c}_i)}{|\mathbf{c}_j - \mathbf{c}_i|}, \quad \bar{\mathbf{H}}^d = \frac{1}{\sum_{j\in\Omega_i} W_{ij}} \sum_{j\in\Omega_i} W_{ij}\kappa^d_{ij}, \tag{2.37}$$

The above equations present the directional ($\bar{\kappa}^d_{ij}$) and the mean ($\bar{\mathbf{H}}^d$) curvatures using the anisotropic deformation tensor and vertex positions of the dual mesh.

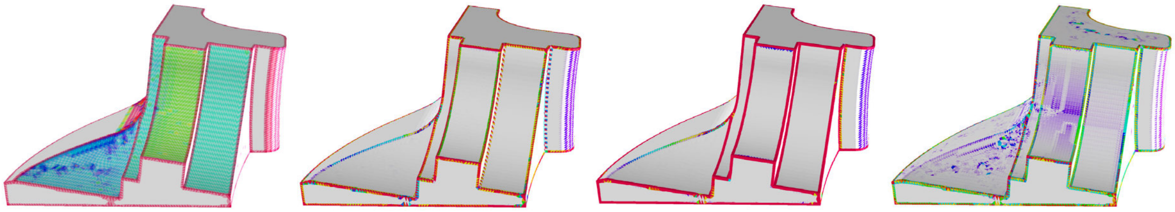

(a) Cotangent MC (b) $\mathbf{H}^d$ (Equation (2.32)) (c) s_f (Equation (2.30)) (d) $\bar{\mathbf{H}}^d$ (Equation (2.37))

Figure 2.4: A visual representation of different kinds of feature analysis operator. (a) traditional mean curvature computed using the shape operator. (b) mean curvature computed using Equation (2.32) on the dual mesh of the Fandisk model. (c) Edges and corners are detected using the eigen-analysis of the ENVT. (d) feature detection using the ENVT deformation tensor as a mean curvature operator.

The deformation tensor $\mathbf{B}$ and the ENVT both have the same eigenbasis. The eigenvalues of the deformation tensor are square roots of the eigenvalues of the ENVT, so it can also be written in its spectral components:

$$\mathbf{B} = \sum_{i=1}^{3} \sqrt{\lambda_i}\mathbf{e}_i\mathbf{e}_i^t. \tag{2.38}$$

Here, $\sqrt{\lambda_2}$ and $\sqrt{\lambda_3}$ approximate the principal curvature, and the corresponding eigenvectors are the principal directions. In Figure 2.2, the least dominant and the most dominant eigendirection $\mathbf{e}_3$

are colored in red and green respectively. The eigenvalue $\mathbf{e}_2$ is painted in blue color. Figure 2.2 (b) shows that the least dominant direction ($\mathbf{e}_3$) is along the edge and the eigenvalue λ_3 corresponding to this direction is equal to zero. The most dominant eigendirection ($\mathbf{e}_1$) approximates the surface normal and across the edge-direction, there will non zero eigenvalue $\sqrt{\lambda_2}$. The tangent space is spanned by the two least dominant directions. Here, we can say that the lower sub-matrix of the deformation tensor approximates the shape operator.

2.6 Summary

In this chapter, a shape analysis operator, the ENVT is introduced. The spectral analysis of the ENVT revealed that it can be decomposed into several feature based tensors, for example, stick, plate and ball tensors. To identify feature points on a triangulated surface, a feature strength parameter s_f is introduced and Figure 2.4 (c) shows that s_f detects sharp features on a geometry effectively. Later, the concept of dual mesh is exploited to derive the mean curvature operator and the directional curvature based on face normals and barycentric points of the corresponding faces. With help of the deformation tensor, which is decomposed from the ENVT, a metric and mean curvature vector is again computed. In the end, similar to the curvature tensor by Taubin [100], correlation between the shape operator and the ENVT is presented and it is shown that two least dominant eigenvalues of the ENVT approximate the squared principal curvatures and corresponding eigenvectors are the principal directions.

Chapter 3

ENVT-based Mesh Denoising Algorithm

In general, noise and sharp features both are high frequency components and decoupling them during a denoising operation, is a challenging task. Several traditional average-based denoising methods have been published to remove noise effectively. However, they are not able to preserve sharp features properly because of the ambiguity between noise and sharp features. Unlike other traditional averaging approaches, our method uses the element-based normal voting tensor to compute smooth surfaces. By introducing a binary optimization on the tensor together with a local binary neighborhood concept, the algorithm better retains sharp features and produces smoother umbilical regions than previous approaches. On top of that, a stochastic analysis on the different kinds of noise is provided based on the average edge length. The quantitative results demonstrate that the performance of our method is better compared to state-of-the-art smoothing approaches. This chapter is based on a published mesh denoising method [114]. Additionally, we added the detailed denoising explanation in 1D using a noisy polygon including a graphical description of the ENVT multiplication to the noisy face normal. The definitions of different kinds of neighborhood are included. We shortened the properties and explanation of the ENVT as it is well explained in Chapter 2.

This algorithm follows a two stage denoising process. In the first stage, noisy face normals are processed to remove undesired noisy component and in the second stage, vertex positions are updated according to processed face normals. The main contributions of this chapter are:

- A tensor-based mesh denoising algorithm is introduced to remove the undesired noise from noisy surfaces with a stable and fast convergence property.
- A binary optimization technique is applied on the eigenvalues of the proposed element-based normal voting tensor (ENVT) that helps us to retain sharp features in the concerned geometry and improves the convergence rate of the algorithm.
- We give a stochastic analysis of the effect of noise on a triangular mesh based on the minimum edge length of elements in that geometry. It gives an upper bound to noise standard deviation to have minimum probability for flipped element normals.

3.1 Method

Figure 3.1 shows the whole pipeline of our algorithm. Face normal smoothing (the yellow blocks in Figure 3.1) consists of four steps: (1) We compute a neighborhood disk for the concerned face using a local binary scheme. (2) We define the ENVT within its neighborhood disk. (3) To remove

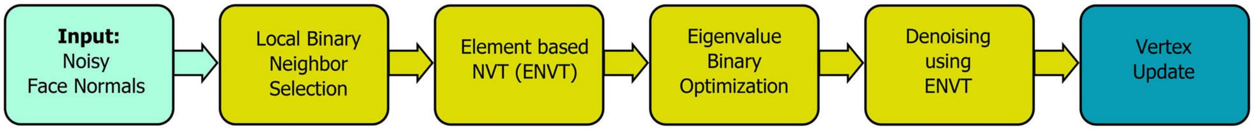


Figure 3.1: The proposed smoothing algorithm pipeline. The yellow blocks show face normal processing and the blue block represents the last stage (vertex update) of the method.

noise effectively, we apply a binary optimization on the eigenvalues of the computed tensor. (4) We multiply the modified ENVT to corresponding face normals to suppress noise. In the last stage (the blue block in Figure 3.1), we update vertex positions using the orthogonality between edge vectors and face normals. In this section, we explain each stage of the proposed algorithm briefly.

3.1.1 Local Binary Neighbor Selection

The first step of our denoising scheme is the preprocessing of face normals using neighboring face normals. To select the neighborhood region $\mathbf{\Omega}$, there are three possibilities: Combinatorial, geodesic and geometrical neighborhood, which are defined as:

1. **Combinatorial neighborhood:** is defined as a set of all elements connected with vertices of the corresponding face:

$$\mathbf{\Omega}_i = \{\mathbf{f}_j | \mathbf{v}_{i1} \in \mathbf{f}_j \vee \mathbf{v}_{i2} \in \mathbf{f}_j \vee \mathbf{v}_{i3} \in \mathbf{f}_j\},$$

   where the neighborhood region is presented by $\mathbf{\Omega}$ and vertices $\mathbf{v}_{i1}$, $\mathbf{v}_{i2}$ and $\mathbf{v}_{i3}$ belong to the face $\mathbf{f}_i$.

2. **Geometrical neighborhood:** is defined as a set of all elements belonging to a disk area of the desired radius and centered at the corresponding element:

$$\mathbf{\Omega}_i = \{\mathbf{f}_j | \quad |\mathbf{c}_j - \mathbf{c}_i| \leq r\},$$

   where c_i and c_j are centroids of the central and neighbor elementa and r is the radius the geometrical neighborhood disk.

3. **Geodesic neighborhood:** is defined as a set of all elements within the shortest distance defined by the radius r:

$$\mathbf{\Omega}_i = \{\mathbf{f}_j | \quad \mathcal{D}(\mathbf{f}_i, \mathbf{f}_j) \leq r\},$$

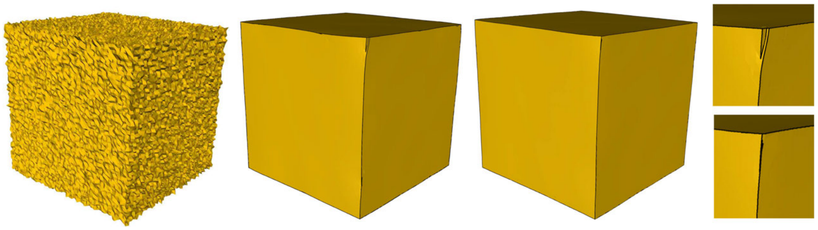
   where $\mathbf{f}_i$ is the source point and $\mathcal{D}(\mathbf{f}_i, \mathbf{f}_j) : \mathcal{M} \to \mathbb{R}$ is a geodesic distance function on a manifold surface $\mathcal{M}$.

The geometrical neighborhood is preferred because it depends only on the disk size irrespective of mesh resolution unlike the topological neighborhood.

The geometric neighborhood elements are weighted based on an angle threshold value ρ [79]. Based on ρ, we assign a binary value to neighborhood elements $\mathbf{f}_j$ w.r.t. the central element $\mathbf{f}_i$ using the following function:

$$w_{ij} = \begin{cases} 1 & \text{if } \angle(\mathbf{n}_i, \mathbf{n}_j) \leq \rho \\ 0.1 & \text{if } \angle(\mathbf{n}_i, \mathbf{n}_j) > \rho, \end{cases} \tag{3.1}$$

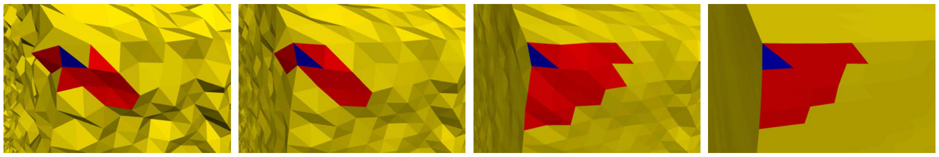
where $\mathbf{n}_i$ and $\mathbf{n}_j$ are face normals of the central element and neighbor elements. By using the value of 0.1, close to a feature, we still allow the area on the other side of that feature to contribute (so



(a) Noisy input (b) $w_{ij} \in \{0,1\}$ (c) $w_{ij} \in \{0.1,1\}$ (d) zoom

Figure 3.2: Effect of two different binary weighting functions in the proposed method. (a) Noisy model. (b) $w_{ij} \in \{0,1\}$. (c) The weight function mentioned in Equation (3.1). (d) A magnified view shows that the weight function mentioned in Equation (3.1) is more effective compared to the exact binary weighting ($w_{ij} \in \{0,1\}$).

the edge direction can be detected from the computed tensor), but the area on ”same” side of the feature will be dominant. Figure 3.2 shows that a contribution of the other side of a feature helps to enhance sharp corners (Figure 3.2(d)). Equation (3.1) shows a discontinuous box filter which takes similar faces into consideration and avoids blurring features within the user defined geometric neighborhood. Figure 3.3 shows that w˙ij depends on the dihedral angle, which can be unstable intially but stabilizes after a few iterations. In further discussion, local binary neighbor definition refers to Equation (3.1).



(a) Noisy input (b) 10 iterations (c) 20 iterations (d) 40 iterations

Figure 3.3: The local binary neighbor selection in the proposed algorithm. Intially, it selects very few neighbor elements $\mathbf{f}_j$ (red color) around the central element $\mathbf{f}_i$(blue) because of the dihedral angle threshold. As iterations increase, it selects more elements with similar element normals.

3.1.2 Element-based Normal Voting Tensor (ENVT)

In chapter 2, a detailed analysis about the ENVT is given. For mesh denoising purpose, we modified the weighing function according to the local binary neighbour concept along with a triangle area factor. The ENVT $\mathbf{C}_i$ is a covariance matrix, defined on the face $\mathbf{f}_i$:

$$\mathbf{C}_{\mathbf{\Omega}_i} = \frac{1}{\sum_{j\in\mathbf{\Omega}_i} w_{ij} A_j} \sum_{j\in\Omega} w_{ij} A_j \mathbf{n}_j \otimes \mathbf{n}_j, \tag{3.2}$$

where A_j is the area of $\mathbf{f}_j$ and w_{ij} is the weighting function as mentioned in Equation (3.1). Weighting by corresponding element areas makes the ENVT more robust against irregular sampling. The eigenanalyis of the given tensor identifies features on triangulated surfaces similar to the methods [55] and [12]. In our algorithm, the ENVT is represented as a mesh denoising operator which is able to suppress noise contents from noisy surfaces while preserving sharp features. The ENVT $\mathbf{C}_{\Omega_i}$ is a symmetric and positive semi definite matrix so we can represent it using an orthonormal basis of the eigenvectors $\mathbf{e}_k$ and real eigenvalues λ_k:

$$\mathbf{C}_{\mathbf{\Omega}_i} = \sum_{k=1}^{3} \lambda_k \mathbf{e}_k \otimes \mathbf{e}_k. \tag{3.3}$$

As mentioned in chapter 2, on a noise-free triangulated mesh, a planar area has only one dominant eigenvalue in surface normal direction. Two dominant eigenvalues indicate edge features where the weakest eigenvector will be along the edge direction. At a corner, all three eigenvalues are dominant.

3.1.3 Eigenvalues Binary Optimization

Let us consider a noisy mesh, corrupted by a random noise with standard deviation σ_n bounded by minimum edge length. On a planar area (face) of a geometry: $\lambda_1 \gg \sigma_n$, and the other two eigenvalues will be proportional to noise intensity, $\lambda_2, \lambda_3 \propto \sigma_n$. Similarly, on an edge of a geometry: $\lambda_1, \lambda_2 \gg \sigma_n$ and $\lambda_3 \propto \sigma_n$. On a corner of a geometry: $\lambda_1, \lambda_2, \lambda_3 \gg \sigma_n$.

The concept of binary optimization is applied to remove noise effectively by setting the less dominant eigenvalues to zero and the dominant eigenvalues to one. Our optimization technique removes noise not only from planar areas but also along edge directions of sharp features during the denoising process. The binary optimization is implemented by introducing a scalar threshold value τ, which is proportional to noise intensity $\tau \propto \sigma_n$ and is smaller than the dominant eigenvalues. The $\tilde{\lambda}$ are modified eigenvalues of the ENVT using the following optimization technique. There are three eigenvalues for feature classification. Therefore, our optimization method checks the following three cases:

- At corners of noisy surfaces (smooth or sharp), the smallest eigenvalues should be bigger than the threshold value i.e. $\lambda_3 \geq \tau$. Hence:

$$\tilde{\lambda}_i = 1, \quad i \in \{1, 2, 3\} \quad \text{if} \quad \lambda_3 \geq \tau.$$

- At edges of a noisy geometry (smooth or sharp), the less dominant eigenvalue should be smaller than the threshold value i.e. $\lambda_3 < \tau$ and $\lambda_2 \geq \tau$. Hence:

$$\tilde{\lambda}_2 = \tilde{\lambda}_1 = 1, \quad \tilde{\lambda}_3 = 0 \quad \text{if} \quad \lambda_2 \geq \tau, \quad \lambda_3 < \tau.$$

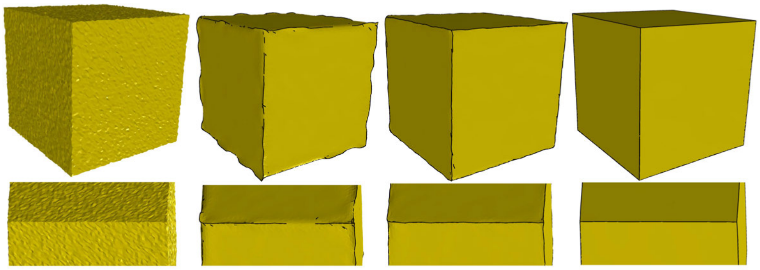
- In the last case, we check for planar area of a geometry. Having $\lambda_2 < \tau$ and $\lambda_3 < \tau$ shows that the only dominant eigenvalue is λ_1. Hence:

$$\tilde{\lambda}_1 = 1, \quad \tilde{\lambda}_2 = \tilde{\lambda}_3 = 0 \quad \text{if} \quad \lambda_1 \geq \tau, \quad \lambda_3,\lambda_2 < \tau.$$

There are three possible combinations during the eigenvalue binary optimization. The threshold τ has to be set by the user according to noise intensity.

3.1.4 Denoising using ENVT

Our denoising method is inspired by feature classification characteristic of the eigenvalues of the ENVT. The smallest eigendirections (one for edges of a geometry, two in planar areas) represent noise. Multiplication of the ENVT with corresponding element normals will suppress noise in the weak eigendirection. This operation also projects element normals in the strongest eigendirection.



(a) Noisy (b) (c) (d)

Figure 3.4: The results with different combinations of steps of the proposed algorithm. (a) Noisy model, (b) Without using the eigenvalues binary optimization and local binary neighborhood selection. (c) Without the local binary neighbor selection. (d) Using both eigenvalues binary optimization and local binary neighborhood selection. The black curve shows the edge information in the smooth geometry and is detected using the dihedral angle.

Traditionally, face normal smoothing is done by rotating the concerned face normal along a geodesics on a unit sphere whereas our method aligns face normals by projection. We project noisy face normals towards smooth normals by multiplication of the ENVT to the corresponding face normal.

A demonstration of the ENVT multiplication to a noisy face normal in $\mathbb{R}^2$ is shown in Figure 3.5. If $\lambda_2 > \lambda_1$, then it will strengthen the face normal in the desired direction $\mathbf{e}_1$ and suppresses noise in the $\mathbf{e}_2$ direction. The whole procedure consists of the following steps:

- A noisy face normal $\mathbf{n}$ is decomposed according to the eigenbasis ($\mathbf{e}_1$ and $\mathbf{e}_2$) of the element based normal voting tensor, Figure 3.5 (left).
- Then, the modified eigenvalues (λ_1 and λ_2) get multiplied to the corresponding eigendirections to suppress the weak eigenvalue in that eigendirection, Figure 3.5 (middle).
- Finally, the new element normal $\mathbf{n}^p$ is obtained by normalizing $\mathbf{C}_{\mathbf{\Omega}_i} \cdot \mathbf{n}$, in Figure 3.5 (right).

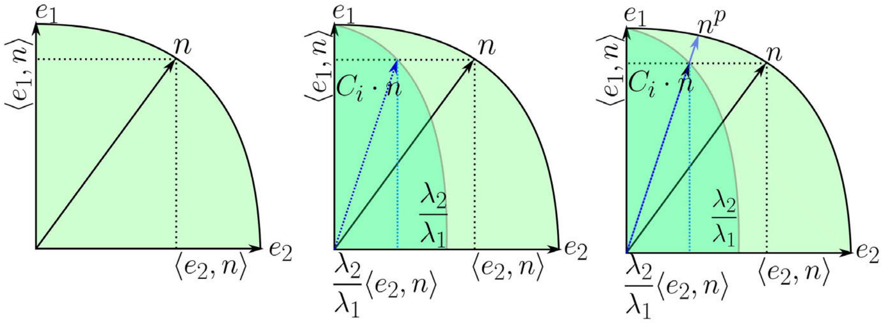


Figure 3.5: Shows the basic idea behind the proposed method to remove noise in $\mathbb{R}^2$ where $\mathbf{e}_i$ and λ_i represent eigenvectors and eigenvalues of the proposed element based voting tensor and $\boldsymbol{n}$ shows the noisy normal. We rotate the noisy normal $\boldsymbol{n}$ towards the dominant eigendirection $\boldsymbol{e_1}$ by a corresponding tensor multiplication.

Anisotropic Face Normal Denoising

We recompute the ENVT by using the same eigenvectors with modified eigenvalues:

$$\tilde{\mathbf{C}}_{\mathbf{\Omega}_i} = \sum_{k=1}^{3} \tilde{\lambda}_k \mathbf{e}_k \otimes \mathbf{e}_k. \tag{3.4}$$

Now, $\tilde{\mathbf{C}}_{\mathbf{\Omega}_i}$ will have the quantized eigenvalues according to different features on a surface. To remove noise, we multiply the corresponding element normal with the newly computed tensor $\tilde{\mathbf{C}}_{\mathbf{\Omega}_i}$. The multiplication will lead to noise removal while retaining sharp features.

$$\tilde{\mathbf{n}}_i = d\mathbf{n}_i + \tilde{\mathbf{C}}_{\mathbf{\Omega}_i}\mathbf{n}_i = d\mathbf{n}_i + \sum_{k=1}^{3} \tilde{\lambda}_k \langle \mathbf{e}_k, \mathbf{n}_i \rangle \mathbf{e}_k, \tag{3.5}$$

where d is the damping factor to control the speed of preprocessing of face normals. We use $d = 3$ for all experiments. The second row of Figure 3.6 shows face normal denoising using the tensor multiplication.

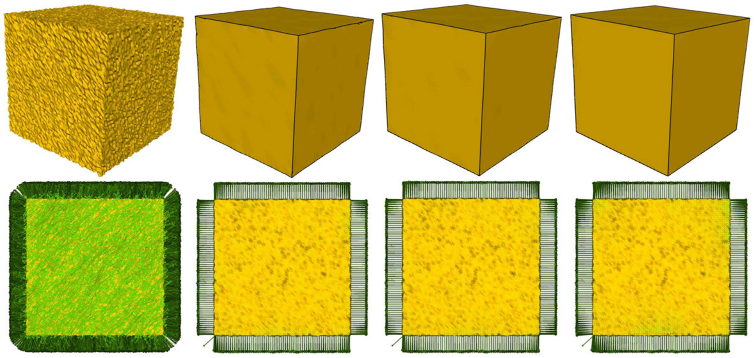

(a) Noisy input (b) 30 iterations (c) 60 iterations (d) 200 iterations

Figure 3.6: Stable convergence of our method with different number of iterations and corresponding processed face normals (XY-plane view). (a) Noisy cube model, (b) The result after 30 iterations, low frequency noise can be seen on planar areas of cube model. (c) After 60 iterations, smoother compared to Figure (b). (d) After 200 iterations. There is no significant difference between Figure (c) and (d).

Vertex Update

In the last stage, we synchronize vertex positions with the corresponding newly computed face normals. To compute a proper vertex position, the orthogonality between edge vectors and face normals is used [95]. The energy function is then defined as follows:

$$\min_{\mathbf{v}_i} \sum_{j=0}^{\mathbf{N}_i-1} \sum_{(i,j)\in\partial F_k} \|\tilde{\mathbf{n}}_k \cdot (\mathbf{v}_i - \mathbf{v}_j)\|^2, \tag{3.6}$$

where $\mathbf{v}_i$ is the vertex position and $\mathbf{N}_i$ represents the number of vertices of the vertex star of $\mathbf{v}_i$, ∂F_k is a set of the boundary edges of the vertex star of $\mathbf{v}_i$ shared with face $\mathbf{f}_k$. The term $\tilde{\mathbf{n}_k}$ is the smooth face normal at $\mathbf{f}_k$. Taubin [101] explained that the face normal vector can be decomposed into a normal and a tangential component and the main problem here is to find vertex positions which minimize the tangential error. The possible solution of Equation (3.6) may be a mesh with degenerate triangular faces. Like Taubin [101], to avoid the degenerate solution, we are using gradient descent that will lead to optimal vertex positions.

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \frac{1}{3F(\mathbf{v}_i)} \sum_{j=0}^{\mathbf{N}_i - 1} \sum_{(i,j)\in\partial F_k} (\tilde{\mathbf{n}}_k \cdot (\mathbf{v}_i - \mathbf{v}_j))\tilde{\mathbf{n}}_k, \tag{3.7}$$

where $F(\mathbf{v}_i)$ is the number of faces connected to the vertex $\mathbf{v}_i$. We iterate the whole procedure several times and the number of iterations depends on the noise intensity. Figure 3.4 shows the effect of each stage of face normal processing in the proposed algorithm.

3.1.5 Effect of Noise on the Proposed Method

Noise is inevitable during digital data acquisition of real life objects. The high intensity of noise flips edges in a geometry and that leads to inconsistent face normals on a geometry. As we mentioned in section 3.2, the ENVT is defined on properly oriented surfaces with consistence face normals because the spectral decomposition of the ENVT is invariant to face normal orientations. In this section, we give a stochastic approximation about the relation between noise and geometry resolution to prevent edge flips in a geometry. Let us consider a smooth triangular mesh $\mathcal{M}_s$ which is corrupted by noise $\mathcal{N}$, $\mathcal{M} = \mathcal{M}_s + \mathcal{N}$. Noise $\mathcal{N}$ can be approximated by a random vector X_n consisting of three independent random variables. We assume that the random vector X_n follows the Gaussian distribution. This is a realistic model for noise from 3D scanning [96]. Let σ_n be the standard deviation of noise in each independent direction, then:

$$\begin{aligned} P\{|X_n| \le \sigma_n\} &= 0.682, \\ P\{|X_n| \le 2\sigma_n\} &= 0.954, \\ P\{|X_n| \le 3\sigma_n\} &= 0.997. \end{aligned}$$

To explain the probability of normal flips, we switch to the 2D case of a polygon in $\mathbb{R}^2$. Let us consider an edge vector l between two vertices $\mathbf{v}_0$ and $\mathbf{v}_1$ in $\mathbb{R}^2$: $\mathbf{l} = (\vec{\mathbf{v}}_0 - \vec{\mathbf{v}}_1)$. We give a probabilistic estimation of the effect of noise on the edge $\mathbf{l}$ w.r.t. the noise intensity (standard deviation) σ_n. Our analysis is mainly focused on the proper orientation of edge normals. Wrong orientation of the edge normal $\mathbf{n}_l$ leads to an edge flip in a smooth geometry. We denote by $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$, the sets of correctly oriented edge normals and wrong oriented edge normals respectively. The probabilistic estimation of the orientation of edge normals based on noise intensity and edge length is given as follows:

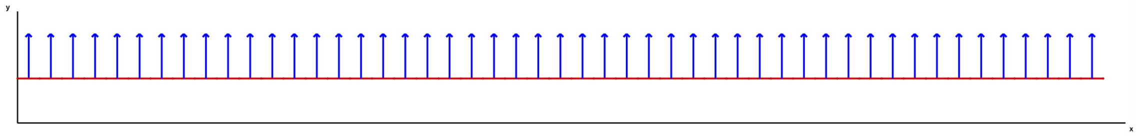
- Probability of an edge to have a correctly oriented edge normal:

$$P\{\vec{\mathbf{n}}_l \in \mathbf{\Omega}_1\} \ge \begin{cases} 0.682 & \text{if } \sigma_n \le \frac{|\mathbf{l}|}{2} \\ 0.954 & \text{if } \sigma_n \le \frac{|\mathbf{l}|}{4}. \end{cases}$$

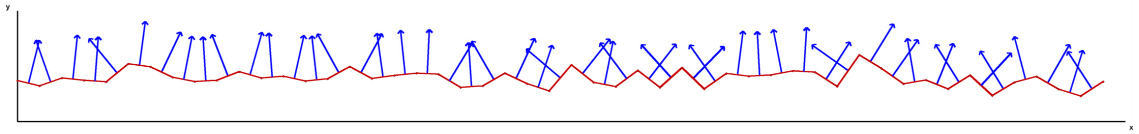
- Similarly, the probability of an edge to have a wrong oriented edge normal:

$$P\{\vec{\mathbf{n}}_l \in \mathbf{\Omega}_2\} \le \begin{cases} 0.318 & \text{if } \sigma_n \le \frac{|\mathbf{l}|}{2} \\ 0.046 & \text{if } \sigma_n \le \frac{|\mathbf{l}|}{4}. \end{cases}$$

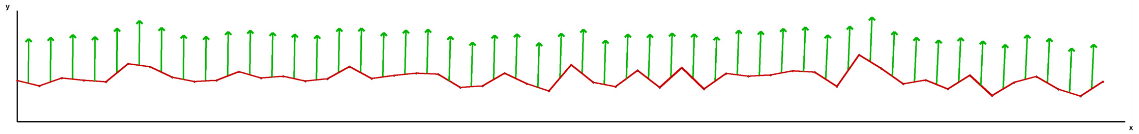
Due to the presence of noise, edge flipping may occur when the vector sum of vertex dislocations at the edge is bigger than the edge length as shown in Figure 3.7. This is similar to sampling theorem,



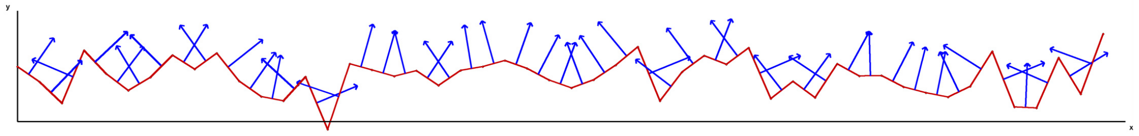
(a) A smooth polygon with edge normals and 50 vertices.



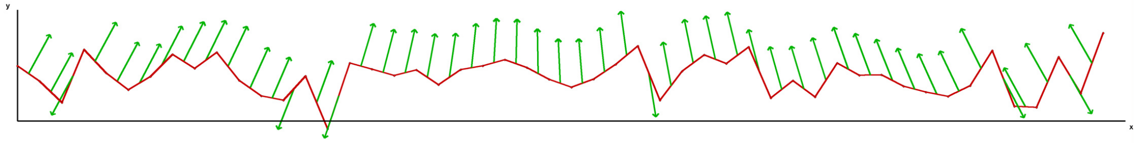
(b) A Gaussian noise ($\sigma_n = 0.4l_e$) is added.



(c) Denoised edge normals of the polygon.



(d) A Gaussian noise ($\sigma_n = l_e$) is added.



(e) Denoised edge normals of the polygon.

Figure 3.7: A visual representation of the effect of noise intensity on the proposed method. (a) A smooth polygon. (b) Corrupted with a Gaussian noise, where σ_n is smaller than the half of the edge length. (c) Smoothed edge normals without any flips. (d) The same polygon corrupted with a Gaussian noise, where noise intensity is equal to the average edge length of the polygon. (e) Denoised edge normals using the proposed method. Due to bigger noise intensity, there are several normal flips in smooth edge normals.

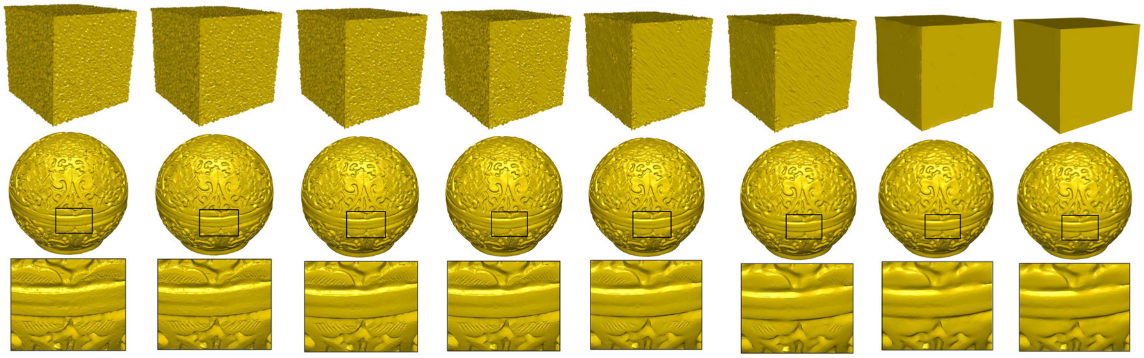

(a) Noisy (b) $\boldsymbol{\tau = 0.01}$ (c) $\boldsymbol{\tau = 0.02}$ (d) $\boldsymbol{\tau = 0.03}$ (e) $\boldsymbol{\tau = 0.04}$ (f) $\boldsymbol{\tau = 0.05}$ (g) $\boldsymbol{\tau = 0.1}$ (h) $\boldsymbol{\tau = 0.5}$

Figure 3.8: The effect of the eigenvalue binary optimization threshold value $\boldsymbol{\tau}$ on results of the proposed algorithm. The first row shows the cube model ($|V| = 24578$, $|F| = 49152$) corrupted by synthetic Gaussian noise ($\sigma_n = 0.4l_e$) where l_e is the average edge length and the second row shows the scanned box model (real data with $|V| = 149992$, $|F| = 299980$) and corresponding results regarding different values of $\boldsymbol{\tau}$. The third row shows the magnified area of the box model. For the box model, the algorithm produced the optimal result with smaller value of $\boldsymbol{\tau = 0.03}$ because of low noise whereas the cube model needed the higher value of $\boldsymbol{\tau = 0.5}$ because of high intensity noise. The bigger value of $\boldsymbol{\tau}$ can lead to the feature blurring as shown in Figure (h) for the box model.

where a signal can be reconstructed properly if and only if the data is sampled with a frequency bigger than twice the highest frequency of a data signal.

Using the given analysis for a given probability density function and an upper bound to the standard deviation, we can estimate the expected number of edge flips in a geometry. If a surface is affected by noise only in normal direction, then there is no edge flip, irrespective of the probability density function of noise.

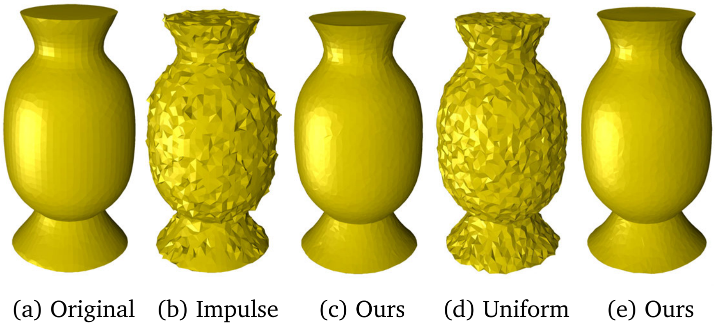
We also experimented with uniformly distributed noise where the random variable X_n follows uniform distribution thus, we can write: $P\{|X_n| \leq \sigma_n\} = 1$. If the noise intensity is less than half of the minimum edge length in a geometry then there will be no edge flip as shown in Figure 3.9.

3.2 Experiments, Results and Discussion

We evaluated the capacity of our algorithm on various kinds of CAD (Figure 3.11 - 3.17), CAGD (Figure 3.8, 3.16) models corrupted with synthetic noise and real scanned data (Figure 3.18, 3.19) models with different types of features. Noisy surfaces with non-uniform mesh corrupted with different kinds of noise (Gaussian, Impulsive, Uniform) in different random directions are also included in our experiments. We compared our method to several state-of-the-art denoising methods in which we implemented [45], [121], [43] and [34] based on their published article and several results of [1], [120], [108] and [67] are provided by their authors.

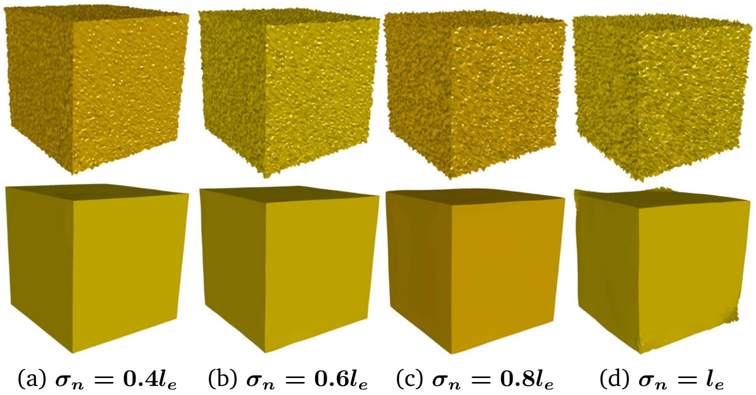
3.2.1 Parameters Tuning

We discussed several parameters (geometric neighbor radius r, dihedral angle threshold ρ, eigenvalue threshold τ, damping factor d and iteration p) and throughout, the whole experimentation, we fixed $\rho = 0.8$, $d = 3$. Effectively, there are only **3 parameters** to tune the results, in which τ



(a) Original (b) Impulse (c) Ours (d) Uniform (e) Ours

Figure 3.9: The results obtained by our method against different kinds of noise. (a) Original vase model. (b) $1/3$ of the vertices of the vase model are corrupted by impulsive random noise. (c) Corresponding result with our method. (d) The vase model is corrupted by uniformly distributed noise and (e) corresponding result.



(a) $\boldsymbol{\sigma_n = 0.4 l_e}$ (b) $\boldsymbol{\sigma_n = 0.6 l_e}$ (c) $\boldsymbol{\sigma_n = 0.8 l_e}$ (d) $\boldsymbol{\sigma_n = l_e}$

Figure 3.10: Robustness against different levels of noise: The first row shows the cube model corrupted with different levels of noise. The second row shows the corresponding results obtained by the proposed method. In Figure (d), noise level is bigger than the feature size and it is impossible to decouple features from noise. As a consequence, we are not able to recover the perfect cube.

is the most important as it depends on noise intensity but at the same time this parameter is not highly sensitive. We use $\boldsymbol{\tau \in \{0.3 - 0.4\}}$ for synthetic data and $\boldsymbol{\tau \in \{0.05 - 0.1\}}$ for real data because real data have smaller noise intensity compared to synthetic data. The neighborhood radius r depends on the number of elements within the geometric neighborhood region. We iterate several times ($\boldsymbol{p \in \{40 - 60\}}$) to obtain better result. In the quantitative comparison Table 3.1, the parameters are given in the following format:$\boldsymbol{(\tau, r, p)}$. For the [120] and [108] methods, we mention *Default* in the parameter column because smooth models are provided by those authors. We are following a similar pattern for other algorithms too: (σ_c, σ_s, p) for [34], (σ_s, p) for [121], (λ, s, p) for [45] and α for [43], where σ_s, σ_c are the standard deviations of Gaussian functions in the bilateral weighting, s and λ represent the step size and the smoothing threshold, α controls the amount of smoothing.

To see the effect of different values of τ, we have experimented with two different models, the Box model (real data, with different level of features and less noise) and the Cube model (with limited features and high noise). With smaller values of $\tau \in [0.01, 0.05]$, there is not much change

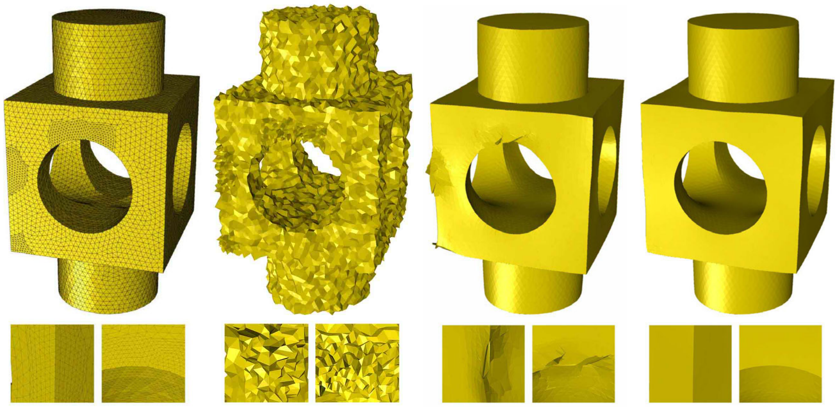

(a) Original (b) Noisy (c) Combinatorial (d) Geometric

Figure 3.11: Comparison between geometric and combinatorial neighborhood on a non-uniform mesh block model. Figure (c) and (d) show the results obtained by our method using the topological and the geometric neighborhood.

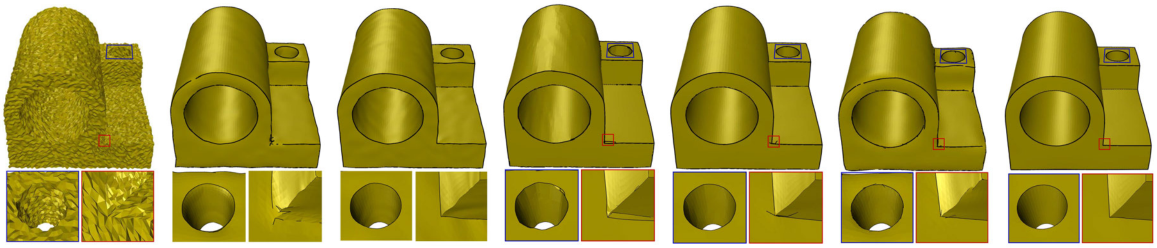

(a) Noisy (b) [45] (c) [121] (d) [43] (e) [120] (f) [108] (g) Ours

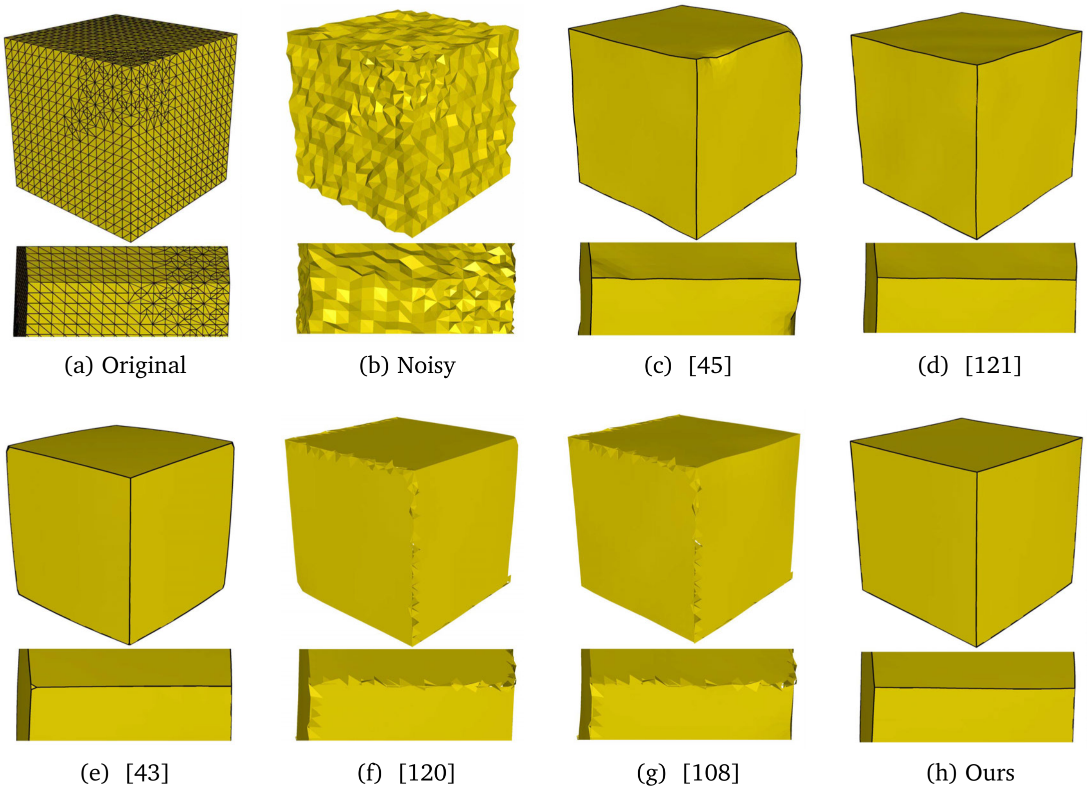
Figure 3.12: Non-uniform triangulated mesh surface corrupted by Gaussian noise ($\sigma_n = 0.35l_e$) in normal direction where l_e is the average edge length. The first row shows the results obtained by state-of-the-art methods and the proposed method. The second row shows the magnified view of the corner and the cylindrical hole of the corresponding geometry.

in the cube model because of higher noise whereas the box model manages to remove noise and as well as shallow features with increasing τ. So τ is responsible for removing noise and also for preserving features. If the feature size is smaller than the noise intensity, feature preservation is an ill-posed problem as shown in Figure 3.8.

Figure 3.11 shows that the geometrical neighborhood is more effective against irregular meshes compared to the topological neighborhood. The geodesic neighborhood is quite similar to the geometrical neighborhood but it is not appropriate when a model is corrupted by high intensity of noise.

3.2.2 Visual Comparison with State-of-the-art Methods

The Block (Figure 3.11), the Joint (Figure 3.12), the Cube (Figure 3.13) and the Devil (Figure 3.15) have non-uniform meshes corrupted with Gaussian noise in random direction. Figure 3.12 shows that the proposed method produces a smooth model with sharp features without creating any false features (piecewise flat areas like [43]) while method [121] does not manage to remove low fre-


(a) Original (b) Noisy (c) [45] (d) [121]

(e) [43] (f) [120] (g) [108] (h) Ours

Figure 3.13: The Cube model consists of non-uniform triangles corrupted by Gaussian noise ($\sigma_n = 0.3l_e$) in normal direction. The first row shows the results produced by state-of-the-art methods and our proposed method. The second row shows magnified view of one of the sharp edges in the cube model. The results show that the proposed method has sharper and straighter edges compared to state-of-the-art methods.

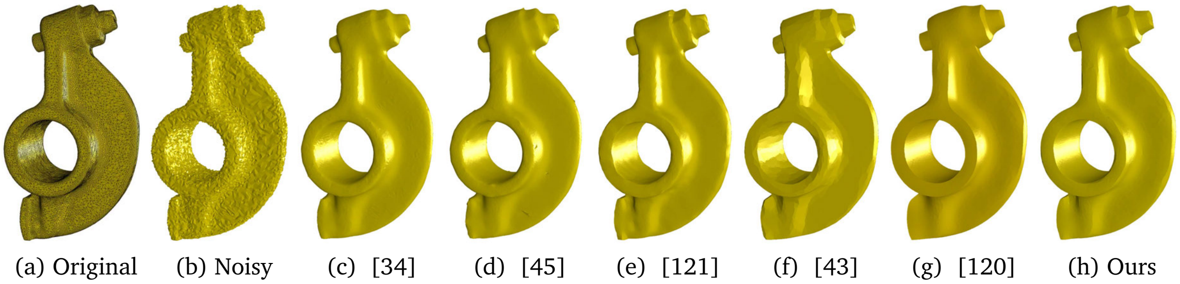

(a) Original (b) Noisy (c) [34] (d) [45] (e) [121] (f) [43] (g) [120] (h) Ours

Figure 3.14: Rockerarm model corrupted by Gaussian noise ($\sigma_n = 0.3l_e$) in normal direction. The results are produced by state-of-the-art methods and our proposed method.

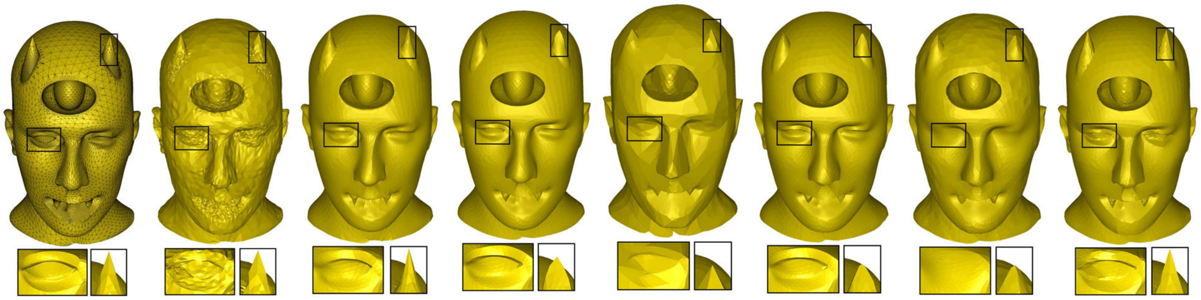

(a) Original (b) Noisy (c) [45] (d) [121] (e) [43] (f) [120] (g) [108] (h) Ours

Figure 3.15: The Devil model consists of non-uniform triangles corrupted by Gaussian noise with standard deviation $\sigma_n = 0.15 l_e$. The first row shows the results produced by state-of-the-art methods and our proposed method. The second row shows the magnified view of the left eye and the right horn of the devil. The results show that the proposed method has minimum shrinkage at the horn area.

quency noise (we can see smooth ripples). Method [120] produces good results but at the narrow cylindrical area, it could not manage to retain the circular area. It also produces some false features at the non-uniform sharp corner. Method [108] could not manage to retain sharp features. Similarly, we can see this behavior for the Cube model in Figure 4.4. The Rockerarm model (Figure 3.14) has a considerably non-uniform mesh and our method better retains sharp features (the screw part) compared to [120], [45], [34] while removing noise better compared to methods [121], [43]. Figure 3.15 shows the robustness of our method against volume shrinkage. The horns of the model have minimum shrinkage compared to state-of-the-art methods. The Fandisk model contains both cylindrical and sharp feature regions and is corrupted by high intensity Gaussian noise in random direction. Figure 3.17 shows that the proposed method delivers both sharp features and umbilical regions without noise component and false features. Figure 3.16 shows that the proposed algorithm does not outperform state-of-the-art method in terms of features preservation (the left eye of the Julius model) but removes noise better compared to methods [120], [67], as it can be seen by mean curvature coloring (flat area of the Julius model).

For real data, we can not see considerable differences between the proposed method and state-of-the-art denoising methods because the noise intensity is quite low and state-of-the-art methods also produce good results. Figure 3.18 shows that our method better retains features in the right eye of the angel model, but for the Rabbit model, our result is quite similar to [120] and better compared to other methods. For the Julius model, our method produces a smoother result, but at the cost of some fine features (around the eyes) compared to method [67]. For the Vase model, method [67] produces quite similar results to ours. However, the shrinkage effect is bigger in method [67] as shown in Table 3.1. Figure 3.19 shows the robustness of the proposed algorithm against irregular meshes and holes in the real data. The Gorgoyle and the Eagle model have several holes and spikes, but our smoothing algorithm manages to produce a smooth surface with proper features.

Our method is invariant against different kinds of noise as shown in Figure 3.9 where the vase model is corrupted by impulsive and uniform noise. The proposed method does not produce appropriate results above a certain level of noise as shown in Figure 3.10. Figure 3.20 shows that our method is robust against random edge flips.

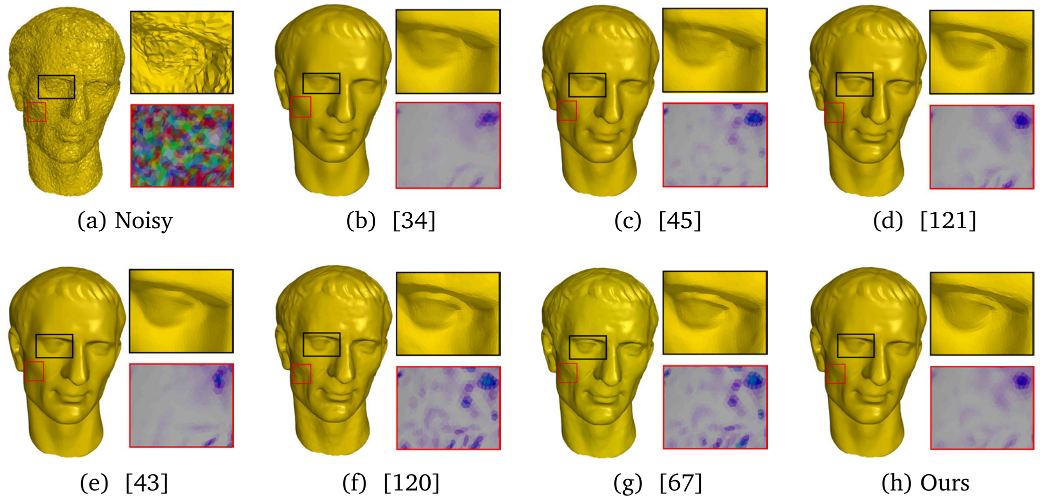


Figure 3.16: The Julius model corrupted by Gaussian noise ($\sigma_n = 0.2l_e$) in random direction. The results are produced by state-of-the-art methods and the proposed method. Our method does not outperform state-of-the-art methods and the output is quite similar to method [121] (eye of the Julius model). The proposed method removes low frequency noise better than methods [120], [67] where we can see small ripples of noise on the Julius model, as shown in the second row of the figure by the mean curvature coloring.

3.2.3 Quantitative Comparison with State-of-the-art Methods

In this section, we give a quantitative comparison of our method with state-of-the-art methods. We are using two different parameters: E_v (L^2 vertex-based error) and MSAE (the mean square angular error). The positional error from the original ground truth model is represented by E_v and defined as [95]:

$$E_v = \sqrt{\frac{1}{3\sum_{k\in F} A_k} \sum_{i\in V} \sum_{j\in F_v(i)} A_j \operatorname{dist}(\tilde{v}_i, T)^2},$$

where F is the triangular element set and V represents the set of vertices. A_k and A_j are the corresponding element areas and $F_v(i)$ is the number of elements in the i^{th} vertex-ring. $\operatorname{dist}(\tilde{v}_i, T)$ is the closest L^2-distance between the newly computed vertex $\tilde{v}_i$ and the triangle T of the reference model. The MSAE computes the orientation error between the original model and the smooth model and is defined as:

$$MSAE = E[\angle(\tilde{\mathbf{n}}, \mathbf{n})],$$

where $\tilde{\mathbf{n}}$ is the newly computed face normal and $\mathbf{n}$ represents the face normal of the reference model. E stands for the expectation value. The quantitative comparison Table 3.1 shows that our method performs better for most of the models e.g. Cube, Devil, Joint etc. For some model like Fandisk, our method produces quite similar numeric errors as state-of-the-art methods.

3.2.4 Convergence and Running Time Complexity

Our smoothing algorithm has a stable and fast convergence (as shown in Figure 3.21) because of the eigenvalue binary optimization. Modification of the eigenvalues of the ENVT will not affect the orientation of the corresponding face normal when noise is removed because the difference between two eigenvalues will be zero and also the less dominant eigenvalue will be zero. There will be no

Models	Methods	MSAE (degrees)	$E_v(\times 10^{-3})$	Parameters
Cube $\lvert F\rvert = 3808$ $\lvert V\rvert = 1906$	[34]	5.267	3.142	(0.3, 0.3, 30)
	[45]	1.935	1.432	(0.05, 0.05, 50)
	[121]	1.159	1.207	(0.3, 60)
	[43]	1.343	4.476	(1.4)
	[120]	40.41	1.166	(Default)
	[108]	41.63	1.446	(Default)
	Ours	**0.700**	**1.034**	(0.3, 0.15, 50)
Devil $\lvert F\rvert = 25906$ $\lvert V\rvert = 12986$	[34]	4.983	58.86	(0.3, 0.3, 20)
	[45]	3.444	32.53	(0.2, 0.05, 30)
	[121]	2.641	8.95	(0.3, 50)
	[43]	6.699	26.28	(4.0)
	[120]	2.870	9.44	(Default)
	[108]	5.396	13.52	(Default)
	Ours	**2.702**	**6.901**	(0.1, 1.0, 30)
Joint $\lvert F\rvert = 52226$ $\lvert V\rvert = 26111$	[34]	3.777	0.530	(0.3, 0.3, 40)
	[45]	2.630	0.766	(0.009, 0.05, 50)
	[121]	1.808	0.263	(0.4, 100)
	[43]	1.768	0.500	(1.4)
	[120]	0.956	0.179	(Default)
	[108]	2.874	0.366	(Default)
	[67]	1.16	1.49	(Default)
	Ours	**0.829**	**0.171**	(0.3, 0.05, 60)
Fandisk $\lvert F\rvert = 12946$ $\lvert V\rvert = 6475$	[34]	8.567	4.422	(0.4, 0.4, 40)
	[45]	5.856	4.910	(0.07, 0.05, 30)
	[121]	2.727	1.877	(0.4,70)
	[43]	4.788	5.415	(1.4)
	[120]	**2.221**	**1.702**	(Default)
	[67]	3.1	4.42	(Default)
	Ours	2.692	1.964	(0.3, 0.2, 50)
Rockerarm $\lvert F\rvert = 48212$ $\lvert V\rvert = 24106$	[34]	5.737	44.22	(0.3, 0.3, 40)
	[45]	5.982	51.35	(0.2, 0.05, 50)
	[121]	5.713	23.47	(1.0, 100)
	[43]	7.468	34.70	(1.4)
	[120]	6.846	30.45	(Default)
	Ours	**5.410**	**20.05**	(0.3, 1.0, 60)
Vase $\lvert V\rvert = 3827$ $\lvert F\rvert = 7650$	[67]	**2.92**	1.72	(Default)
	[43]	4.26	0.83	(1.6)
	Ours	3.34	**0.42**	(0.3, 0.03, 30)

Table 3.1: Quantitative Comparison

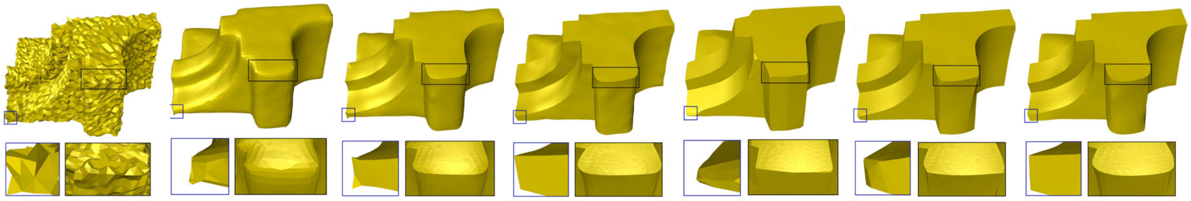

(a) Noisy (b) [34] (c) [45] (d) [121] (e) [43] (f) [120] (g) Ours

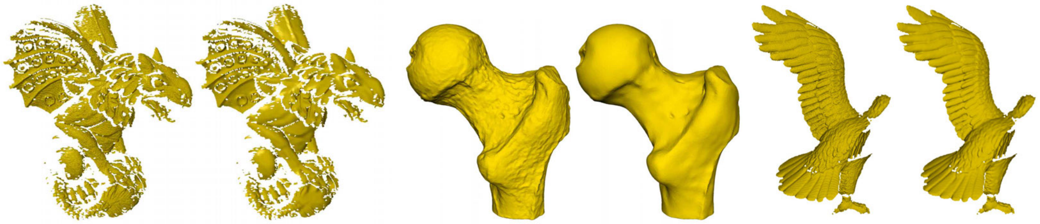
Figure 3.17: Fandisk model corrupted with a Gaussian noise ($\sigma_e = 0.3l_e$) in random direction. The first row shows the results produced by state-of-the-art methods and our proposed method. The second row shows the corner and the cylindrical region of the concerned model. The proposed method does not produce any false features in the umbilical region while retains sharp features and corners.


(a) Noisy (b) [34] (c) [45] (d) [121] (e) [43] (f) [120] (g) Ours

Figure 3.18: Triangulated mesh surface (real data) corrupted by 3D scanner noise. Both rows show the results obtained by state-of-the-art methods and the proposed method for the Angel and the Rabbit models.

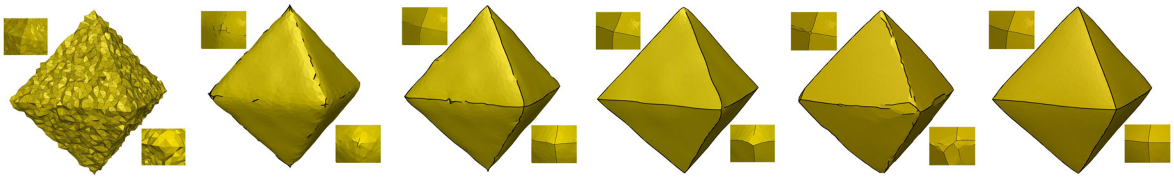
more modification on noisy surfaces after some iteration when we meet the explained scenario as shown in Figure 3.6, where there is no significant change (visually) after 60 iterations. Figure 3.21 shows that the proposed method converges with minimum error compared to the methods [45] and [121]. We can see that after 40 iterations, our method is almost stable and does not produce any significant changes. The eigenvalue binary optimization not only helps in preserving features, but also improves the convergence rate of the algorithm.

Running time complexity of the proposed method is similar to most of the two-step denoising methods [77], [101], [117], [121]. The neighborhood computation is done by the growing disk method to compute the ENVT. The ENVT computation has the complexity of $O(c \cdot n_f \cdot p)$, where c is the number of elements within the neighborhood, n_f and p are the numbers of elements and iterations respectively. The tensor multiplication procedure has the running complexity of $O(n_f)$. Similarly, the vertex update procedure has the complexity of $O(c \cdot n_v \cdot p)$, where n_v is the number of vertices in the geometry. In general, $n_f > n_v$, so the overall complexity of the algorithm will be $O(c \cdot n_f \cdot p)$. The number of elements in the geometric neighborhood c plays an important role in the running time of the algorithm as shown in Table 3.2. For example, the Devil model has smaller number of elements and larger running time compared to the Joint model because of the different geometric neighborhood radius. The Bilateral normal method [121] uses a fix number of neighborhood elements (depending on the valence of vertex) for face normal smoothing and



(a) The gorgoyle model (b) The balljoint model (c) The eagle model

Figure 3.19: Results obtained by our method against different kinds of real data captured by the laser scanner. The figure (a) and (c) show real life scans with a lot of holes and our method manages to produce good results.



(a) Random edge flip (b) [34] (c) [45] (d) [121] (e) [43] (f) Ours

Figure 3.20: Robustness against random edge flips after adding noise. (a) Noisy model corrupted with $\sigma_n = 0.3 l_e$ and random edge flips, (b) to (f) show that the proposed method preserves better features (magnified corners) compared to state-of-the-art methods.

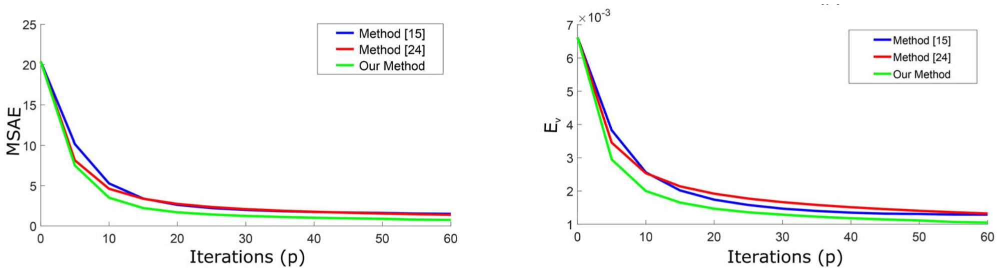
is a bit faster compared to the proposed method. However, the other recent two step denosing methods [120], [108] are slower compared to our method because of their additional denoising steps.

Table 3.2: Running Time (in seconds)

Models	Cube	Devil	Joint	Fandisk	Rockerarm	Vase
Time (s)	1.2	51.9	38.2	5.06	27.5	1.1

3.3 Summary

This chapter presented a simple and effective tensor multiplication algorithm for feature-preserving mesh denoising. The concept of the element-based normal voting tensor (ENVT) has been introduced and eigenanalysis of this tensor leads to decoupling of features from noise. The proposed method does not need any additional Laplacian-based smoothing technique to remove noise, like multistage state-of-the-art methods [29], [12], [120], [108]. This method removes noise by multiplying the ENVT to the corresponding face normal, and this reduces complexity of the algorithm. This chapter also introduced the concept of eigenvalue binary optimization that not only enhances sharp features but also improves the convergence rate of the proposed algorithm. The local binary neighborhood selection helps to select similar elements in neighborhood to compute the element based normal voting tensor which avoids feature blurring during the denoising process. This chapter provided a stochastic analysis of noise effect on the geometry depending on the average edge length of the triangulated mesh. On the basis of this analysis, user can provide an upper bound on the noise



(a) MSAE (b) E_v

Figure 3.21: Convergence plot comparison between the methods [45], [121] and our method. The error metrics MSAE and E_v are computed for the Cube model and p is the number of iterations. Figure shows that the proposed method has better convergence rate compared to the methods [45] and [121].

standard deviation depending on the minimum edge length to reconstruct the smooth surface from a noisy surface. The experimental results (visual and quantitative) show capability of the proposed algorithm. This method produces good results not only in terms of visual but also quantitatively with all kinds of data including CAD, CAGD and real data. This chapter have also shown robustness of the algorithm against different kinds and levels of noise and also discussed wrong orientation of triangles in presence of strong noise.

Chapter 4

Point Set Denoising Algorithm using Vertex-based NVT

In general, point set surfaces are acquired by 3D scanners and during this acquisition process, noise and outliers are inevitable. For a high fidelity surface reconstruction from noisy point sets, a feature preserving point set denoising operation should be performed to remove undesired noise and outliers components from the input point set. To suppress these noise components and outliers with feature preservation, we introduce an anisotropic point set denoising algorithm in the normal voting tensor framework. The proposed method is implemented in three different stages. In the first stage, noisy vertex normals, which are computed using PCA, are filtered using the vertex-based normal voting tensor and the binary optimization. In the second stage, feature points are categorized into corners and edges using an anisotropic covariance matrix, which is computed based on filtered vertex normals. In the last stage, vertex positions are updated according to the processed vertex normals using restricted quadratic error metrics. For the vertex update, we added different constraints to the quadratic error metric based on feature (edges and corners) and non-feature (planar) points.

4.1 Method

The proposed method has three different stages, vertex normal filtering, feature detection and vertex position update. Let us consider that a non-uniform and noisy input point set, which represented as, $\mathbf{V} = \{\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{n-1}\} \subset \mathbb{R}^3$, where n is the number of vertices. These data points are acquired by a 3D laser scanner and intrinsically do not have vertex normals. Principal component analysis (PCA) [46] is applied to compute consistent initial vertex normals, which are represented as, $\mathbf{N} = \{\mathbf{n}_0, \mathbf{n}_1, \cdots, \mathbf{n}_{n-1} \subset \mathbb{R}^3\}$.

4.1.1 Vertex Normal Filtering

This is the first stage of the proposed method, where noisy vertex normals are filtered using the vertex-based NVT and the binary optimization similar to Method [114]. Vertex normal filtering is done in following steps:

Vertex-based Normal Voting Tensor

As we discussed in Chapter 2, covariance matrices compute the variation of an entity in a well defined domain, for example, the covariance matrix of edge vectors in a point set computes the variation of vertices in $\mathbb{R}^3$ space. Similarly, the covariance matrix of vertex normals in a well defined neighbourhood computes the anisotropic nature of a point set in that region. To analyze an anisotropic nature of a shape (point set surface), we define the vertex-based NVT, which is computed using neighbour vertex normals:

$$\mathbf{C}_{\boldsymbol{\Omega}_i} = \frac{1}{\sum_{j\in\boldsymbol{\Omega}_i} w_{ij}} \sum_{j\in\boldsymbol{\Omega}_i} w_{ij}\mathbf{n}_j \otimes \mathbf{n}_j, \tag{4.1}$$

where $\boldsymbol{\Omega}_i$ is the geometric neighbourhood disc centred at the vertex $\mathbf{v}_i$ and it is computed using KNN. The symbol $\otimes$ represents the outer product. The weight function w_{ij} is computed based on the similarity between neighbour normals. To define w_{ij}, we follow local binary neigbourhood concept, where the central vertex normal looks at neighborhood vertex normals and assigns a binary value (1 or 0) according to the normal difference. The weight term w_{ij} is defined as:

$$w_{ij} = \begin{cases} 1 & \text{if } \angle(\mathbf{n}_i, \mathbf{n}_j) \leq \rho \\ 0 & \text{if } \angle(\mathbf{n}_i, \mathbf{n}_j) > \rho, \end{cases} \tag{4.2}$$

where ρ is the local binary neighbourhood threshold, which is used to select similar vertex points within the disc region $\boldsymbol{\Omega}_i$. The weight term w_{ij} is not exactly similar to the weight function mentioned in Chapter 3 because vertex normals are more sensitive to noise compared to face normals, for example at edges, angles between vertex normals is smaller compared to face normals. Unlike face normals filtering, we are not considering the vertices, which are on the other side of edges.

The term $\mathbf{C}_{\boldsymbol{\Omega}_i}$ is a covariance matrix. Therefore, it is a symmetric and positive semi-definite matrix and can be represented in terms of its spectral components:

$$\mathbf{C}_{\boldsymbol{\Omega}_i} = \sum_{k=1}^{3} \lambda_k \mathbf{e}_k \otimes \mathbf{e}_k, \tag{4.3}$$

where λ_k and $\mathbf{e}_k$ are the corresponding eigenvalues and eigenvectors. Let us sort the eigenvalues in decreasing order $\{\lambda_1 \geq \lambda_3 \geq \lambda_3 \geq 0\}$ and the we can rewrite $\mathbf{C}_{\boldsymbol{\Omega}_i}$ as:

$$\mathbf{C}_{\boldsymbol{\Omega}_i} = (\lambda_1 - \lambda_2)\mathbf{e}_1 \otimes \mathbf{e}_1 + (\lambda_2 - \lambda_3)(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2) + \lambda_3(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2 + \mathbf{e}_3 \otimes \mathbf{e}_3), \tag{4.4}$$

where the first term of the right hand side is known as the stick tensor and has only one dominant eigenvalue in the normal direction. The second term is spanned by two dominant eigen-directions and the normal direction is defined in the least dominant direction. This term is known as the plate tensor. The third term is spanned all dominant eigen-directions and does not have a well defined normal direction [70]. From above description, it is clear that the vertex-based NVT consists anisotropic nature of a point set and feature points can be easily detected using the eigenvalues of $\mathbf{C}_{\boldsymbol{\Omega}_i}$. For example, if there is only one dominant eigenvalue then it is a planar area, if two eigenvalues are dominant then it is an edge and if all eigenvalues are dominant then it is a corner.

Eigenvalues Binary Optimization

In our method, the vertex-based NVT is applied as a denoising operator on a noisy point set. As we discussed in last section that the vertex-based NVT is capable of detecting features on point sets as shown in Figure 4.1(d). However, on a noisy point set, the behaviour of the spectral components of the vertex-based NVT will change. Let us assume that a point set is corrupted by a random noise with standard deviation σ_n. Due to presence of noise, the eigenvalues of the vertex-based NVT will alter, for example, on a planar area, the other two eigenvalues will be non-zero and proportional

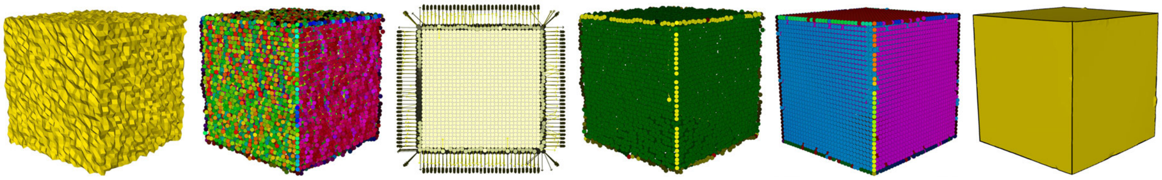

(a) (b) (c) (d) (e) (f)

Figure 4.1: The algorithm pipeline. (a) a noisy point set and the surface is reconstructed using ”ball pivoting” algorithm [11]. (b) noisy ertices are colored according to vertex normals variation. (c) vertex normals filtering using the vertex-based NVT and the eigenvalues binary optimization. (d) feature points (yellow and red vertices) detection using the anisotropic covariance matrix. (e) vertex positions are updated using the proposed constraint-based quadratic error metrics. (f) the smooth surface is reconstructed using [11].

to noise intensity. Similarly, on an edge of a geometry, the least dominant eigenvalue $\lambda_3 \propto \sigma_n$. On a corner of a geometry: $\lambda_1, \lambda_2, \lambda_3 \gg \sigma_n$. To remove noise effect from the vertex-based NVT, the eigenvalues of $\mathbf{C}_{\mathbf{\Omega}_i}$ should be modified according to Equation 4.4, which means , on a planar area and on an edge, the least dominant eigenvalues should be zero and at a corner all eigenvalues are equal and dominant.

The concept of binary optimization is applied to the eigenvalues $\mathbf{C}_{\mathbf{\Omega}_i}$, where each eigenvalues will be assigned a binary number (1 or 0) to remove noise components effectively. Similar to Method [114], a threshold value τ is used for the eigenvalues binary optimization. The term τ should be chosen according to the noise intensity $\tau \propto \sigma_n$ and smaller than the dominant eigenvalues. Let us consider that the $\tilde{\lambda}$ are modified eigenvalues of the vertex-based NVT after the eigenvalues binary optimization, which is applied based on feature classification:

- At corners of a noisy point set (smooth or sharp), the smallest eigenvalues should be bigger than the threshold value i.e. $\lambda_3 \geq \tau$. Hence:

$$\tilde{\lambda}_i = 1, \quad i \in \{1,2,3\} \quad \text{if} \quad \lambda_3 \geq \tau.$$

- At edges of a noisy point set (smooth or sharp), the less dominant eigenvalue should be smaller than the threshold value i.e. $\lambda_3 < \tau$ and $\lambda_2 \geq \tau$. Hence:

$$\tilde{\lambda}_2 = \tilde{\lambda}_1 = 1, \quad \tilde{\lambda}_3 = 0 \quad \text{if} \quad \lambda_2 \geq \tau, \quad \lambda_3 < \tau.$$

- In the last case, we check for planar area of a geometry. Having $\lambda_2 < \tau$ and $\lambda_3 < \tau$ show that the only dominant eigenvalue is λ_1. Hence:

$$\tilde{\lambda}_1 = 1, \quad \tilde{\lambda}_2 = \tilde{\lambda}_3 = 0 \quad \text{if} \quad \lambda_1 \geq \tau, \quad \lambda_3,\lambda_2 < \tau.$$

The above procedure will remove noise components from the eigenvalues of the vertex-based NVT.

Vertex Normal Denoising

To remove noise components, we project noisy vertex normals towards smooth normals by multiplication of the vertex-based NVT to the corresponding vertex normal. This multiplication procedure will suppress noise in weak eigendirections and strengthens vertex normals in strongest eigendirections.

Before multiplication, we recompute the modified vertex-based NVT by using the same eigenvectors with modified eigenvalues:

$$\tilde{\mathbf{C}}_{\mathbf{\Omega}_i} = \sum_{k=1}^{3} \tilde{\lambda}_k \mathbf{e}_k \otimes \mathbf{e}_k. \tag{4.5}$$

To remove noise, we multiply the corresponding vertex normal with the modified tensor $\tilde{\mathbf{C}}_{\mathbf{\Omega}_i}$. The multiplication will lead to noise removal while retaining sharp features.

$$\tilde{\mathbf{n}}_i = d\mathbf{n}_i + \tilde{\mathbf{C}}_{\mathbf{\Omega}_i}\mathbf{n}_i = d\mathbf{n}_i + \sum_{k=1}^{3} \tilde{\lambda}_k \langle \mathbf{e}_k, \mathbf{n}_i \rangle \mathbf{e}_k, \tag{4.6}$$

where d is the damping factor to control the denoising speed of face normals. We use $d = 3$ for all experiments.

4.1.2 Feature Detection

This is the second stage of the proposed denoising, where we classify the three different sets: corner, edge and planar points using the spectral analysis of an anisotropic covariance matrix. To detect feature points on a point set, which has filtered vertex normals, an anisotropic covariance matrix is defined as:

$$\mathbf{A}_i = \frac{1}{\sum_{j \in \mathbf{\Omega}_i} \tilde{w}_{ij}} \sum_{j \in \mathbf{\Omega}_i} \tilde{w}_{ij} (\mathbf{v}_j - \mathbf{v}_c) \otimes (\mathbf{v}_j - \mathbf{v}_c), \tag{4.7}$$

where the weight function $\tilde{w}_{ij}$ is similar to Equation (4.2) and is computed using filtered vertex normals. The terms $\mathbf{v}_c$ and $\tilde{w}_{ij}$ are defined as:

$$\mathbf{v}_c = \frac{1}{\sum_{j \in \mathbf{\Omega}_i} \tilde{w}_{ij}} \sum_{j \in \mathbf{\Omega}_i} \tilde{w}_{ij} \mathbf{v}_j, \quad \tilde{w}_{ij} = \begin{cases} 1 & \text{if } \angle(\tilde{\mathbf{n}}_i, \tilde{\mathbf{n}}_j) \leq \rho \\ 0 & \text{if } \angle(\tilde{\mathbf{n}}_i, \tilde{\mathbf{n}}_j) > \rho \end{cases}, \tag{4.8}$$

where $\mathbf{v}_c$ is the centroid of vertices, which has similar vertex normals. Similar to the vertex-based NVT, $\mathbf{A}_i$ is also a symmetric and positive semi definite matrix and can represented in its spectral components:

$$\mathbf{A}_i = \sum_{k=1}^{3} a_k \mathbf{x}_k \otimes \mathbf{x}_k, \tag{4.9}$$

where a_k and $\mathbf{x}_k$ are the corresponding eigenvalues and eigenvectors. In the proposed method, the eigenvalues $\{a_1, a_2, a_3\}$ are exploited to classify feature points as following:

1. On a planar area, the two eigenvalues will be dominant in the tangent directions and the least dominant direction will be in the normal direction.
2. On an edge, there will be only one dominant eigenvalue along the edge direction.
3. On a corner, either all eigenvalues are dominant or none of them are significant.

Let us sort the eigenvalues in decreasing order $\{a_1 \geq a_2 \geq a_3 \geq 0\}$ and corner, edge and planar points are represented by $\mathbf{V}_c = \{\mathbf{v}_i^c\}$, $\mathbf{V}_e = \{\mathbf{v}_i^e\}$ and $\mathbf{V}_f = \{\mathbf{v}_i^f\}$ respectively, such that $\mathbf{V} = \mathbf{V}_c \cup \mathbf{V}_e \cup \mathbf{V}_f$.

$$\begin{cases} \mathbf{V}_f = \{\mathbf{v}_j \in \mathbf{V} \mid a_1, a_2 \geq \tau, \quad a_3 < \tau\}, \\ \mathbf{V}_e = \{\mathbf{v}_j \in \mathbf{V} \mid a_1 \geq \tau, \quad a_2, a_3 < \tau\}, \\ \mathbf{V}_c = \{\mathbf{v}_j \in \mathbf{V} \mid (a_1, a_2, a_3) \geq \tau \vee (a_1, a_2, a_3) < \tau\}, \end{cases} \tag{4.10}$$

where τ is the same threshold variable, which is used in the binary optimization of the vertex-based NVT at vertex normal filtering stage.

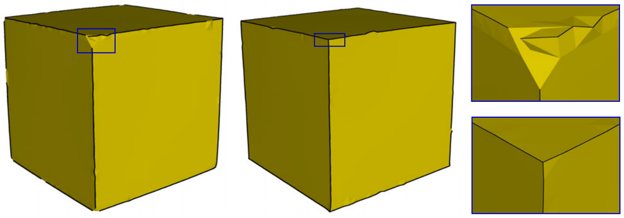

(a) (b) (c)

Figure 4.2: Figure shows the effect of the proposed constraint-based vertex position update scheme. Figure (a) represents the smooth model reconstructed by using Equation (4.16) not only for flat regions but also at feature points. Figure (b) represents the result obtained by the proposed scheme, where flat regions follow Equation (4.16), Edges are reconstructed using Equation (4.15) and corners positions are updated using Equation (4.13). Figure (c) shows the magnified view of the Cube corner. Surfaces are reconstructed using the "ball pivoting" algorithm [11].

4.1.3 Constraint-based Vertex Position Update

In the last stage of the denoising algorithm, we update vertex positions with a distance-based constraint, where the resulting filtered point set remains within a prescribed distance to the input noisy point set. To compute the optimal position of a vertex w.r.t. the filtered vertex normal, restricted quadratic error metrics are used in this algorithm. The restriction to the quadratic error metric is introduce based on different feature points and later vertex position is updated along with a distance-based constraint.

Distance-based Constraint

In the proposed method, we use a distance-based constraint, which limits the deviation between the initial noisy point set and the filtered point set. Therefore, during the denoising process, the maximum deviation of a vertex position is limited by the user. The discrete maximum deviation measure between two point sets $\mathbf{V}$ and $\tilde{\mathbf{V}}$ is defined as:

$$d_m(\mathbf{V}, \tilde{\mathbf{V}}) = max\|\mathbf{v}_i - \tilde{\mathbf{v}}_i\|_{\mathbb{R}^3} \quad \forall i = 0, \cdots, n-1, \tag{4.11}$$

where numbers of vertices in both point sets are equal and the term d_m is a metric and is defined on all set of point sets with equal number of vertices.

Vertex Update at Corners

After a corner point detection using the anisotropic covariance matrix, we minimize the following energy function to compute the optimal vertex position of a corner point:

$$\min_{\tilde{\mathbf{v}}_i^c} \sum_{j \in \mathbf{\Omega}_i} \|\tilde{\mathbf{n}}_j \cdot (\tilde{\mathbf{v}}_i^c - \mathbf{v}_j)\|^2. \tag{4.12}$$

The vertex $\tilde{\mathbf{v}}_i^c \in \mathbf{V}_c$ is a corner point and $\mathbf{v}_j \in \mathbf{V}$ represents the neighbourhood points of the corner point. We can associate a set of planes with each vertex based on the filtered normal, and we define the error of the vertex with respect to this set as the sum of squared distances to its plane. By minimizing the above equation, the new vertex position can be computed as:

$$\tilde{\mathbf{v}}_i^c = \begin{cases} \left(\sum_{j \in \mathbf{\Omega}_i} \mathbf{N}_j \right)^{-1} \sum_{j \in \mathbf{\Omega}_i} \left(\mathbf{N}_j \mathbf{v}_j \right) & \text{if} \quad d_m \leq \epsilon \\ \mathbf{v}_i^c & \text{if} \quad d_m > \epsilon \end{cases}, \tag{4.13}$$

where $\mathbf{N}_j = \tilde{\mathbf{n}}_j \otimes \tilde{\mathbf{n}}_j$ and ϵ is a user input, which keeps the deviation between noisy and filtered corner points. The above equation computes the minimum plane distance for the corner points. The additional distance-based constraint does not allow to move the corner point beyond the user input.

Vertex Update at Edges

On an edge vertex $\mathbf{v}_i^e \in \mathbf{V}_e$ of a noisy point set, which has a filtered vertex normal, we use a restricted version of a quadratic error metric. On an edge vertex, the anisotropic covariance matrix $\mathbf{A}$ has only one dominant eigenvalue and corresponding eigenvector aligns with the edge direction. For each edge vertex, we define a plane $\mathbf{T}_e$, which is spanned by the two least dominant eigenvectors of $\mathbf{A}$. The eigenvectors $\mathbf{x}_2$ and $\mathbf{x}_3$ are the orthonormal basis of tangent plane $\mathbf{T}_e$ and the dominant eigenvector $\mathbf{x}_1$ represents the normal of this plane.

First of all, we project all neighbourhood vertices and the projected vertex normal to the plane $\mathbf{T}_e$:

$$\mathbf{v}_j^p = \mathbf{v}_j - \langle (\mathbf{v}_j - \mathbf{v}_i^e), \mathbf{x}_1 \rangle \mathbf{x}_1,$$
$$\tilde{\mathbf{n}}_j^p = \tilde{\mathbf{n}}_j - \langle \tilde{\mathbf{n}}_j, \mathbf{x}_1 \rangle \mathbf{x}_1,$$

where $\mathbf{v}_j^p$ and $\mathbf{n}_j^p$ are the projected vertex position and its normal on the plane $\mathbf{T}_e$ and $j \in \mathbf{\Omega}_i$. Now, we define a quadratic energy function similar to Equation (4.12):

$$\min_{\tilde{\mathbf{v}}_i^e} \sum_{j \in \mathbf{\Omega}_i} \|\tilde{\mathbf{n}}_j^p \cdot (\tilde{\mathbf{v}}_i^e - \mathbf{v}_j^p)\|^2 + \frac{1}{n_{\mathbf{\Omega}_i}} \|\mathbf{x}_1 \cdot (\tilde{\mathbf{v}}_i^e - \mathbf{v}_j^p)\|^2, \tag{4.14}$$

where $n_{\mathbf{\Omega}_i}$ is the number of vertices within $\mathbf{\Omega}_i$ disc. The above energy function is defined on the plane $\mathbf{T}_e$. We try to minimize the energy function similar to Equation (4.13), but the covariance matrix of the projected normal is not invertible because the least dominant eigenvalue along the the normal of this covariance is zero, which reduces the rank of the corresponding matrix. To fix this problem, additional terms are added to the above energy function.

$$\tilde{\mathbf{v}}_i^e = \begin{cases} \left(\sum_{j \in \mathbf{\Omega}_i} \mathbf{N}_j^e + \mathbf{x}_1 \otimes \mathbf{x}_1 \right)^{-1} \sum_{j \in \mathbf{\Omega}_i} \left(\mathbf{N}_j^e \mathbf{v}_j^p + \mathbf{x}_1 \otimes \mathbf{x}_1 \mathbf{v}_i^e \right) & \text{if} \quad d_m \leq \epsilon \\ \mathbf{v}_i^e & \text{if} \quad d_m > \epsilon \end{cases}, \tag{4.15}$$

where $\mathbf{N}_j^e = \tilde{\mathbf{n}}_j^p \otimes \tilde{\mathbf{n}}_j^p$ and the factor $\mathbf{x}_1 \otimes \mathbf{x}_1$ makes sure that the covariance matrix is invertible. For each edge vertex, the above equation computes the optimal position by minimizing the distance between the lines on a well defined plane. This operation effectively preserves sharp features along edges and removes noise effectively.

Vertex Update on Flat Regions

The flat region is detected using the anisotropic covariance spectral analysis. For the flat region, the matrix $\mathbf{A}$ has two dominant eigenvalue. to remove noise from flat regions, we allow to move the vertex position only in the corresponding vertex normal direction. We use the same energy function mentioned in Equation (4.12) along with a restriction, which allows to move a vertex only in the normal direction. Therefore, the optimal vertex position is computed using the following equation:

$$\tilde{\mathbf{v}}_i^f = \begin{cases} \mathbf{v}_i^f + \frac{\lambda}{\sum\limits_{\mathbf{j}\in\mathbf{\Omega}_i} W_{ij}} \sum\limits_{\mathbf{j}\in\mathbf{\Omega}_i} W_{ij} \langle \tilde{\mathbf{n}}_j \cdot (\mathbf{v}_j - \mathbf{v}_i^f)\rangle \tilde{\mathbf{n}}_i & \text{if} \quad d_m \leq \epsilon \\ \mathbf{v}_i^f & \text{if} \quad d_m > \epsilon \end{cases}, \tag{4.16}$$

where λ controls the amount of smoothing on flat regions and the weight function W_{ij} is a combination of the similarity and the closeness function and defined as:

$$W_{ij} = \exp\Big(-\frac{16|\tilde{\mathbf{n}}_i - \tilde{\mathbf{n}}_j|^2}{r^2}\Big) \cdot \exp\Big(-\frac{4|\mathbf{v}_j - \mathbf{v}_i^f|^2}{r^2}\Big), \tag{4.17}$$

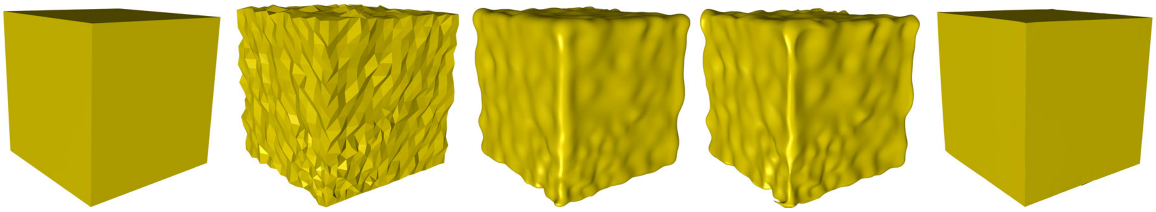
where r is the radius of the disk $\mathbf{\Omega}_i$. By iterating the whole procedure including the vertex normal filtering and the vertex update stages, the proposed algorithm produces a noise-free point set with proper sharp features.

4.2 Experiments, Results and Discussion

(a) Original (b) Noisy (c) APSS (d) RIMLS (e) Ours

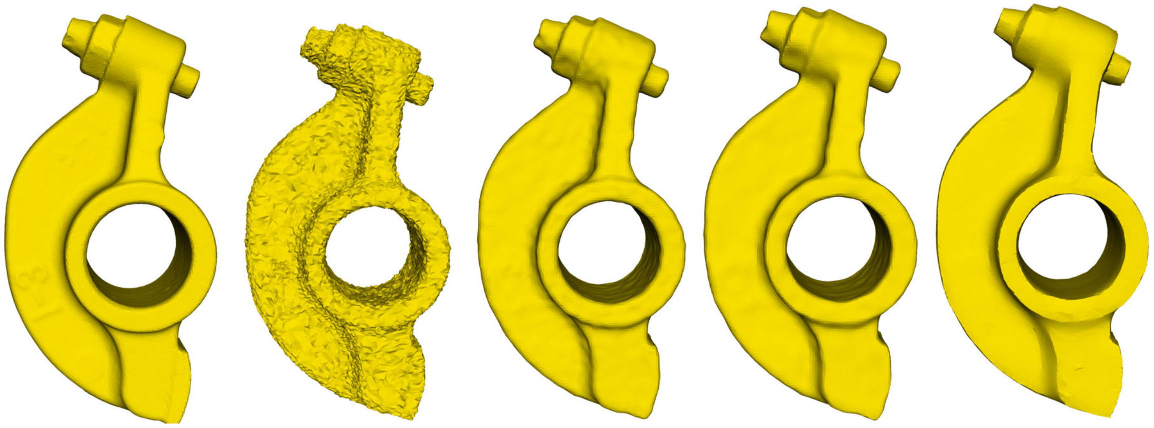
Figure 4.3: The Fandisk model corrupted by Gaussian noise ($\sigma_n = 0.28 l_e$), where l_e is the average distance between the vertices of the model. Figure shows that the proposed method is able to preserve sharp features effectively compared to state-of-the-art methods. Surfaces are reconstructed using the "ball pivoting" algorithm [11].

We evaluated the capacity of our algorithm on various kinds of point set (Figure 4.3, 4.4, 5.9) models corrupted with synthetic noise and real scanned data (Figure 4.6, 4.7). We compared our method with two state-of-the-art denoising methods [40] and [80], which are implemented in MeshLab.



(a) Original (b) Noisy (c) APSS (d) RIMLS (e) Ours

Figure 4.4: The Cube model with non-uniform distribution of vertices, corrupted by Gaussian noise ($\sigma_n = 0.3l_e$), where l_e is the average distance between the vertices of the model. Figure shows that the proposed method is able to preserve sharp features effectively compared to state-of-the-art methods and does not create bumpy structures. Surfaces are reconstructed using the "ball pivoting" algorithm [11].



(a) Original (b) Noisy (c) APSS (d) RIMLS (e) Ours

Figure 4.5: The Rockerarm model corrupted by Gaussian noise (($\sigma_n = 0.3l_e$) in normal direction. The results are produced by state-of-the-art methods and our proposed method. The proposed method removes noise effectively and also enhances sharp features around the cylindrical region. Surfaces are reconstructed using the "ball pivoting" algorithm [11].

4.2.1 Parameters Tuning

We discussed several parameters: geometric neighbor radius r, dihedral angle threshold ρ (Equation (4.2)), eigenvalue threshold τ (Equation (4.10)), damping factor d (Equation (4.6)), iteration p and vertex-diffusion speed λ (Equation (4.16)). Throughout, the whole experimentation, we fixed $\lambda = 0.1$, $d = 3$. The geometric neighbour radius depends on the resolution of point sets and it is fixed to twice of the average distance between vertices. Effectively, there are only **3 parameters** (τ, ρ, p) to tune the results, in which τ is the most important as it depends on noise intensity but at the same time this parameter is not highly sensitive. We use $\boldsymbol{\tau} \in \mathbf{\{0.3 - 0.4\}}$ for synthetic data and $\boldsymbol{\tau} \in \mathbf{\{0.05 - 0.1\}}$ for real data because real data have smaller noise intensity compared to synthetic data. We iterate several times ($\boldsymbol{p} \in \mathbf{\{20 - 100\}}$) to obtain better result.

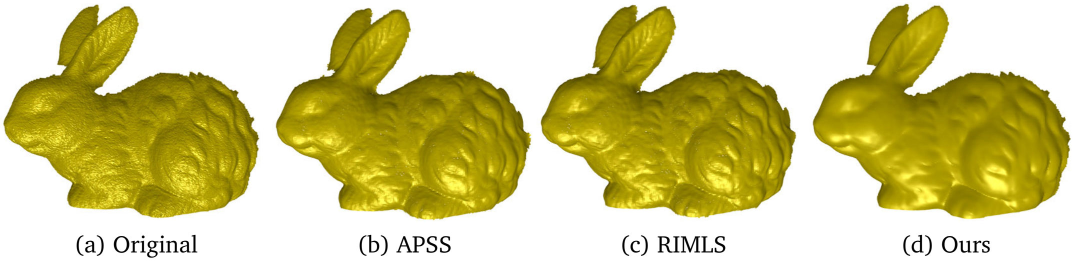

(a) Original (b) APSS (c) RIMLS (d) Ours

Figure 4.6: Real data of the Rabbit model, acquired by a 3D scanner and results produced by the proposed method and state-of-the-art methods. Surfaces are reconstructed using the ”ball pivoting” algorithm [11].

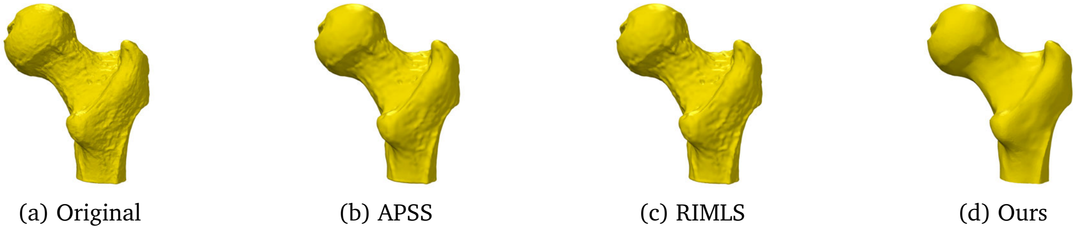

(a) Original (b) APSS (c) RIMLS (d) Ours

Figure 4.7: The balljoint model, which is corrupted by a scanner noise and results produced by the proposed method and state-of-the-art methods. Surfaces are reconstructed using the “ball pivoting” algorithm [11].

4.2.2 Visual Comparison with State-of-the-art Methods

The Cube (Figure 4.4) and the Fandisk (Figure 4.3) have non-uniform sampling corrupted with Gaussian noise. Figure 4.4 shows that the proposed method produces a smooth model with sharp features without creating any false and bumpy features like APSS [40] and RIMLS [80] methods. The Rockerarm model (Figure 4.5) has a considerably non-uniform sampling and our method better enhances sharp features (around cylindrical region) and removes noise effectively compared to [40] and [80].

For real data, Figure 4.6 shows that our method removes noise effectively while retaining features on the surface compared to [40] and [80] methods. Similarly, Figure 4.7 represents the applicability of the proposed method in medical data analysis. As it can be seen from the figure, our method removes noise effectively from the spherical region and retains sharp features in the cylindrical region. Methods [40] and [80] are not able to remove noise components properly.

4.3 Summary

This chapter presented a simple and effective tensor multiplication algorithm for feature-preserving point set denoising. The proposed method is basically an extension of the ENVT-based mesh denoising. Similar to the concept of the ENVT the proposed algorithm used vertex-based NVT and eigenanalysis of this tensor leads to decoupling of features from noise. Noise components are removed by the multiplication of the vertex-based NVT to the corresponding vertex normal. The concept of eigenvalue binary optimization not only enhances sharp features but also improves convergence rate of the method. Similar to the ENVT-based mesh denoising algorithm, the local binary neighborhood

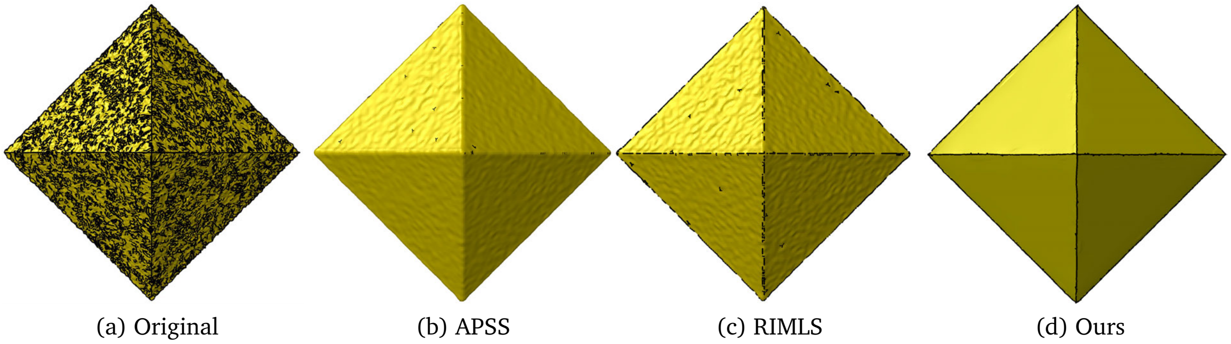

(a) Original (b) APSS (c) RIMLS (d) Ours

Figure 4.8: The Octahedron model, which is corrupted by Gaussian noise and results produced by the proposed method and state-of-the-art methods. Surfaces are reconstructed using the "ball pivoting" algorithm [11].

selection helps to select similar vertices in neighborhood to compute the vertex-based NVT and it leads to minimal feature blurring during the denoising process. After vertex normal filtering, the algorithm classifies feature points into edges, corners and flat regions using an anisotropic covariance matrix. For the vertex update stage, This method used restricted least square error metrics, which are different for different kinds of features. The vertex position reconstruction used restricted quadratic error metrics that helps the algorithms to recreate sharp edges and corners. The experimental results show the capability of the proposed algorithm.

Chapter 5

RoFi: Robust and High Fidelity Mesh Denoising

This chapter presents a simple and effective two-stage mesh denoising algorithm, where in the first stage, face normal filtering is done by using bilateral normal filtering in the robust statistics framework. *Tukey's bi-weight function* is used as the similarity function in bilateral weighting, which is a robust estimator and stops the diffusion at sharp edges to retain features and removes noise from flat regions effectively. In the second stage, an edge-weighted Laplace operator is introduced to compute a differential coordinate. This differential coordinate helps the algorithm to produce a high-quality mesh without any face normal flips and makes the method robust against high-intensity noise.

We propose a robust and high fidelity two-stage mesh denoising method, where face normal processing is mainly motivated by a robust statistics framework in anisotropic diffusion and bilateral filtering. Basically, the robust statistics framework is focused on developing different estimators, which are robust to outliers. In the mesh denoising and robust statistic framework, sharp features are considered as *outliers* and a proper estimator will avoid these *outliers* to preserve sharp features and remove undesired noise components from a noisy surface.

We apply *Tukey's bi-weight function* for the similarity function as a robust estimator, which stops the diffusion at sharp features and produces smooth umbilical regions. In the vertex update step, we use a differential coordinate-based Laplace operator along with an edge-face normal orthogonality constraint. This Laplace operator is computed using the edge length as a weight function to produce a high-quality mesh without face normal flips and it also makes the algorithm more robust against high-intensity noise. This chapter is basically based on the published method [115]. Additionally, the robust estimators are explained in detail.

5.1 Method

The proposed method is implemented in two different stages. In the first stage, noisy face normals are processed using bilateral filtering in the robust statistics framework and in the second stage, a high fidelity mesh is reconstructed using an edge-face normal orthogonality constraint and an edge-weighted differential coordinate. Both stages are iterated until a noise-free surface is produced.

5.1.1 Robust Bilateral Face Normal Processing

In the first step of the proposed algorithm, we smooth noisy face normals using the bilateral filter, which is a non-linear filter and it computes output using the weighted average of the input:

$$\tilde{\mathbf{n}}_i = \frac{1}{K_i} \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s) \mathbf{n}_j, \tag{5.1}$$

where $\mathbf{n}_j$ represents a noisy face normal and belongs to the geometric neighbour disk Ω_i of the central face normal $\mathbf{n}_i$. The geometric neighborhood is more robust against non-uniform meshes compared to the combinatorial neighborhood [114]. The term K_i is defined as:

$$K_i = \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s),$$

where a_j is the area of the corresponding face. The term $f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c)$ and $g(|\mathbf{n}_i - \mathbf{n}_j|, \sigma_s)$ represent the closeness and the similarity functions whose kernel widths are controlled by σ_c and σ_s respectively and $\mathbf{c}_i$ is the centroid of face i.

Robust Estimation

The problem of the estimation of a smooth surface from a noisy surface can be written by using tools of robust statistics. As shown by Black and Sapiro [13], Durand and Dorsey [24] and Jones et al. [51], to compute a smooth surface, Equation (5.1) can be written as the following minimization problem:

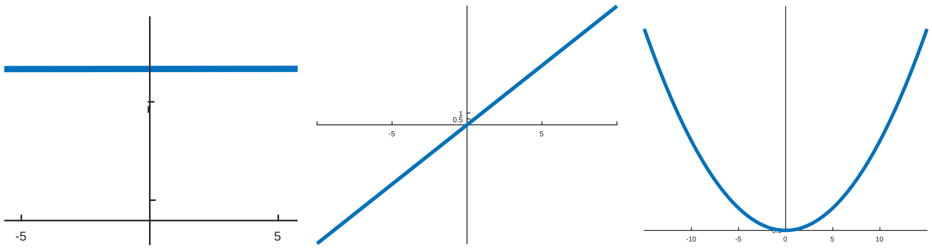
$$\min_{\tilde{\mathbf{n}}_i} \sum_{j \in \Omega_i} a_j f(|\mathbf{c}_i - \mathbf{c}_j|, \sigma_c) \rho(|\tilde{\mathbf{n}}_i - \mathbf{n}_j|, \sigma_s), \tag{5.2}$$

where ρ is a robust error norm. The minimization of the above function will lead to smooth face normals. To solve Equation (5.2), the first derivative of ρ-function should be zero and it is a so-called *influence function* $\Psi(x, \sigma_s) = \rho'(x, \sigma_s)$, where x can be any variable (in our algorithm $x = |\mathbf{n}_i - \mathbf{n}_j|$). The influence function shows the behaviour of a ρ-function.

For anisotropic diffusion and bilateral filtering, the influence function $\Psi(x, \sigma_s)$ and $\rho(x, \sigma_s)$-function can be written in terms of the similarity function $g(x, \sigma_s)$ [24]:

$$\Psi(x, \sigma_s) = x g(x, \sigma_s), \quad \rho(x, \sigma_s) = \int x g(x, \sigma_s) dx.$$

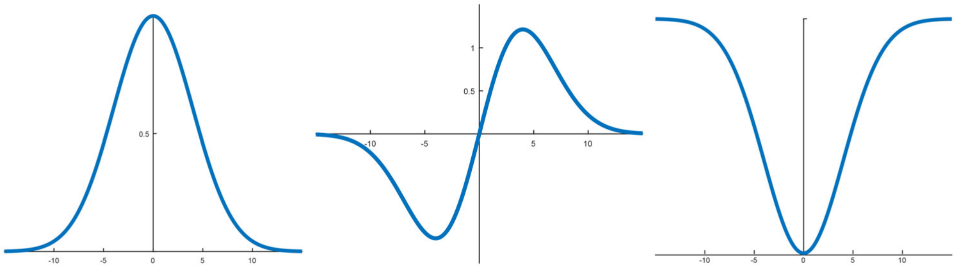
The selection of a ρ-function is crucial to avoid outliers. In our framework, the outliers will be sharp features and an appropriate ρ-function will help the algorithm to retain sharp features and remove noise components.



(a) $g(x, \sigma_s)$ (b) $\Psi(x, \sigma_s)$ (c) $\rho(x, \sigma_s)$

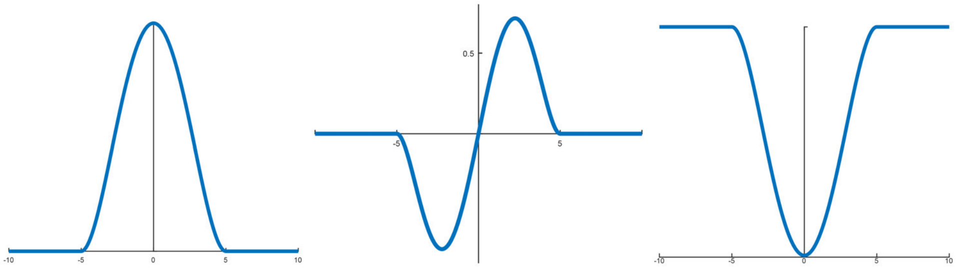
Figure 5.1: An isotropic linear similarity function with corresponding influence function and ρ-function.

For example, the least square estimator (a quadratic ρ-function) is the most naive estimator because it has a linear influence function without any bounds and is very sensitive to outliers as shown in Figure 5.1. Figure 5.2 shows Gaussian-based ρ-function which has a bounded influence function $\Psi(x)$ and gives smaller weight to outliers compared to the least square estimators. Figure



(a) $g(x, \sigma_s)$ (b) $\Psi(x, \sigma_s)$ (c) $\rho(x, \sigma_s)$

Figure 5.2: A Gaussian-based similarity function with corresponding influence function and ρ-function [13].



(a) $g(x, \sigma_s)$ (b) $\Psi(x, \sigma_s)$ (c) $\rho(x, \sigma_s)$

Figure 5.3: Tukey bi-weight similarity function with corresponding influence function and ρ-function [13].

5.3 shows that *Tukey's bi-weight function* with corresponding influence function and ρ-function and it produces more robust results compared to the least square and Gaussian-based estimators because it completely cuts off the diffusion at sharp features (outliers) and removes noise from non-feature areas. *Tukey's bi-weight function* is defined as [13]:

$$g(x, \sigma_s) = \begin{cases} \frac{1}{2}\{1 - (x/\sigma_s)^2\}^2 & \text{if } |x| \leq \sigma_s, \\ 0 & \text{if } |x| > \sigma_s. \end{cases} \tag{5.3}$$

In terms of preserving features, *Tukey's bi-weight function* is more effective compared to Gaussian function because it does not allow the diffusion across sharp features and removes noise components along these sharp features effectively. In our algorithm, we use *Tukey's bi-weight function* as the similarity function to produce feature preserving smooth face normals. For the closeness function $f(\mathbf{c}_i - \mathbf{c}_j, \sigma_c)$, Gaussian function is used.

5.1.2 High Fidelity Mesh Reconstruction

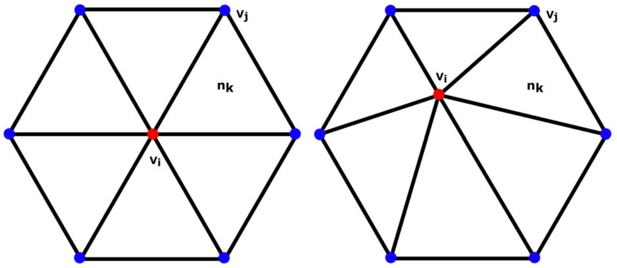
In the second step of the denoising process, we need to impose the effect of processed face normals to corresponding vertices. To reconstruct a high fidelity mesh, we minimize the following quadratic energy function:

$$\min_{\tilde{\mathbf{v}}_i} \{|\mathbf{v}_i - \tilde{\mathbf{v}}_i|^2 + \sum_{j \in \mathbf{N}_i} \sum_{(i,j) \in \partial F_k} |\tilde{\mathbf{n}}_k \cdot (\tilde{\mathbf{v}}_i - \mathbf{v}_j)|^2 + |\mathbf{R}_i|^2\}, \tag{5.4}$$

where $\tilde{\mathbf{v}}_i$ and $\mathbf{v}_i$ are the newly computed and the noisy vertex positions. The terms $\mathbf{N}_i$ and ∂F_k represent the set of vertices and the boundary edge of the vertex star of $\mathbf{v}_i$ respectively. The second term of the energy function synchronizes a vertex position to the corresponding face normal by using the concept of orthogonality between edge vector and face normal [101]. The term $\mathbf{R}_i$ helps algorithm to produce a high fidelity mesh and makes the proposed algorithm robust against high-intensity noise. This term is computed using a differential coordinate and can be written as:

$$\mathbf{R}_i = \mathbf{D}_i + \mathbf{D}_{t_i}, \tag{5.5}$$

where $\mathbf{D}_i$ and $\mathbf{D}_{t_i}$ represent the differential coordinate and its tangential component respectively.



(a) $\mathbf{D}_i = 0$ (b) $\mathbf{D}_i \neq 0$

Figure 5.4: An example to show the effect of differential coordinate $\mathbf{D}_i$ at the second stage of the proposed algorithm.

The differential coordinate at the vertex $\mathbf{v}_i$ is defined as:

$$\mathbf{D}_i = \triangle\mathbf{v}_i - \mathbf{v}_i, \quad \triangle\mathbf{v}_i = \frac{1}{\sum\limits_{j \in N(i)} w_{ij}} \sum_{j \in N(i)} w_{ij}\mathbf{v}_j. \tag{5.6}$$

The weighting term is defined as $w_{ij} = |\mathbf{v}_i - \mathbf{v}_j|^2$. As shown in Figure 5.4, the differential coordinate $\mathbf{D}_i$ computes the deviation vector from the centroid of the 1-ring neighbour. If the central vertex is positioned at the centroid of the 1-ring polygon, then $\mathbf{D}_i = \mathbf{0}$, otherwise it computes a distance vector from the centroid of the 1-ring polygon. The term $\mathbf{D}_i$ helps the proposed algorithm to produce mesh triangles of better quality (high fidelity mesh). An ideal example would be equilateral triangles, if the 1-ring polygon had 6 vertices. The term $\mathbf{D}_{t_i}$ is computed w.r.t the corresponding vertex normal, which is approximated using the processed face normals:

$$\bar{\mathbf{n}}_{v_i} = \frac{1}{\sum\limits_{j \in F_v(i)} a_j} \sum_{j \in F_v(i)} a_j \tilde{\mathbf{n}}_j, \tag{5.7}$$

where $\tilde{\mathbf{n}}_j$ and a_j represents the processed face normal and the area of the corresponding face. The term $F_v(i)$ represents the set of faces, which are connected to the vertex $\mathbf{v}_i$. Now by using $\bar{\mathbf{n}}_{v_i}$, the tangential component of the differential coordinate is computed as:

$$\mathbf{D}_{t_i} = \mathbf{D}_i - \langle \mathbf{D}_i, \bar{\mathbf{n}}_{v_i} \rangle \bar{\mathbf{n}}_{v_i}, \tag{5.8}$$

where the term $\langle \mathbf{D}_i, \bar{\mathbf{n}}_{v_i} \rangle \bar{\mathbf{n}}_{v_i}$ represents the normal component of the differential coordinate.

Effect of a Linear Combination of the components of Differential Coordinates

The term $\mathbf{R}_i$ is effective against high-intensity noise and helps the algorithm to produce high-quality mesh triangles. It is a linear combination of the normal and the tangent components of

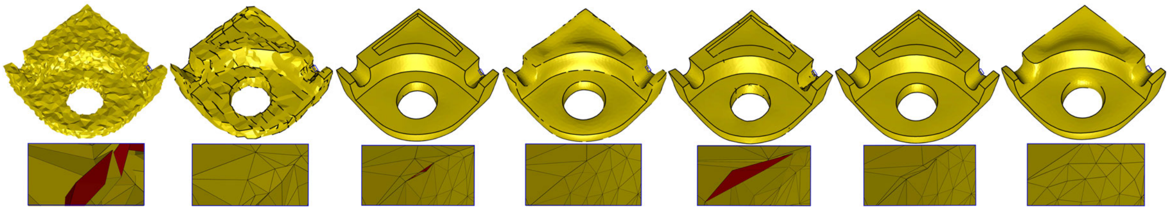

(a) Noisy (b) $\sigma_s = 0.1$ (c) $\sigma_s = 0.5$ (d) $\sigma_s = 1.5$ (e) $\lambda_I = 0.0$ (f) $\lambda_I = 0.2$ (g) $\lambda_I = 0.6$

Figure 5.5: The Bearing model is corrupted with a uniform noise ($\sigma_n = 0.5l_e$) in random direction (a). For Figure (b)-(d), $\lambda_I = 0.2$ and for Figure (e)-(g), $\sigma_s = 0.8$. The black curve shows sharp edge informations in the smooth geometry and is detected using the dihedral angle $\theta = 65°$. The second row shows the magnified view of the mesh structure at a sharp edge. The red color faces are with wrong orientation (flipped normals).

$\mathbf{D}_i$ (Equation (5.5)). In general, noise affects a geometry in the surface normal direction. Therefore, the normal component of $\mathbf{D}_i$ is more effective against high-intensity noise. However, the diffusion of the normal components does not improve the mesh quality and it causes a volume shrinkage during the denoising process. Similarly, diffusion of tangential components leads to a better mesh quality, less shrinkage, and less shape deformation. However, it is less effective against high-intensity noise. As we mentioned earlier that we target to produce a noise-free geometry with sharp features, a better mesh quality, and minimum volume shrinkage. Therefore, we give more weight to the tangential component of $\mathbf{D}_i$ compared to the normal component of $\mathbf{D}_i$. Figure 5.7 shows the effect of different components of $\mathbf{D}_i$. As it can be seen, the addition of $\mathbf{R}_i$ produces a better mesh quality (Q = 0.66, mentioned in Section 5.2.2), and acceptable volume shrinkage compared to other combinations of $\mathbf{D}_i$.

Minimization of the energy function mentioned in Equation (5.4), is done by the gradient descent method and the updated vertex position is:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \frac{1}{3|F_v(i)|} \sum_{j \in \mathbf{N}_i} \sum_{(i,j) \in \partial F_k} (\tilde{\mathbf{n}}_k \cdot (\mathbf{v}_i - \mathbf{v}_j))\tilde{\mathbf{n}}_k, + \lambda_I \mathbf{R}_i, \tag{5.9}$$

where λ_I is the isotropic smoothing or the mesh fidelity factor. We iterate the whole procedure mentioned in Equation (5.9) several times to get the desired result. In each iteration, using the same value of λ_I leads to feature blurring, shape deformation and volume shrinkage in the smoothed surface as noise is getting lower with the iterations. To overcome this problem, the effect of isotropic smoothing should get reduced after each iteration. We reduce λ_I in every iteration by $\lambda_I^{p+1} = 0.6\lambda_I^p$, where p represents the number of the current iterations.

5.2 Experiments, Results and Discussion

We have performed our denoising scheme on various kinds of CAD (Figures 5.5, 5.6, 5.7, 5.9, 5.11, and 5.12) and CAGD (Figures 5.10, and 5.12) models. These models consist of different levels of features and are corrupted with different kinds of synthetic noise (Gaussian and uniform) in different random directions. We also included real scanned data (Figures 5.8, 5.13, and 5.15). We provide visual and quantitative comparison of our method to several state-of-the-art denoising methods. We implemented [45], [34], [121], [43], and [114] based on their published article and several results of [16], [108], [120], [106], and [67] are provided by their authors.

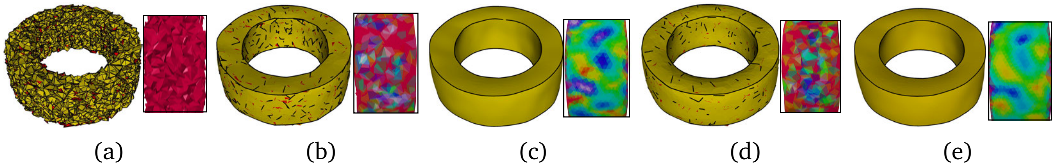


Figure 5.6: Effects of the proposed vertex update method and *Tukey's bi-weight function*. The addition of the edge-based Laplacian operator produces a high fidelity mesh (low value of Q) and *Tukey's bi-weight function* suppressed low frequency components better compared to the Gaussian function. (a) noisy input. (b) output using a Gaussian based bilateral weighting and the least square error constraint for vertex update [95], (E_v, MSAE, Q) = (0.001, 15.9, 3.69). (c) output using a Gaussian based bilateral weighting and the proposed vertex update method, (E_v, MSAE, Q) = (0.00096, 3.77, 0.80). (d) output using the proposed face normal filtering and the least square error constraint for vertex update [95], (E_v, MSAE, Q) = (0.0013, 10.8, 4.68). (e) output using the proposed face normal filtering and the proposed vertex update method, (E_v, MSAE, Q) = (0.00094, 3.63, 0.81). In each figure, the second column shows the cotangent mean curvature, which is coloured between white(0) and red(2.0).

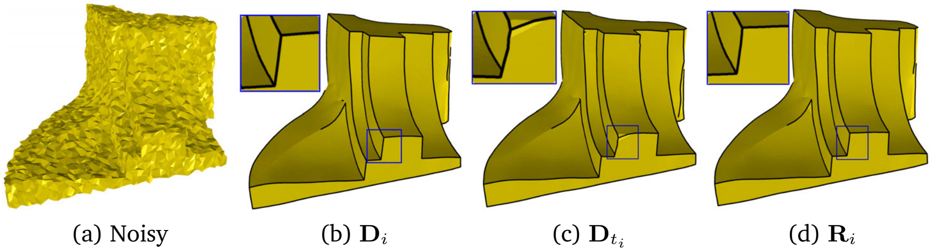

(a) Noisy (b) $\mathbf{D}_i$ (c) $\mathbf{D}_{t_i}$ (d) $\mathbf{R}_i$

Figure 5.7: Visual and quantitative comparison of the different components of $\mathbf{D}_i$. (a) the Fandisk model corrupted by a uniform noise ($\sigma_n = 0.65 l_e$) in a random direction, (E_v, MSAE, Q) = (0.077, 28.09, 1.01). (b) output by using $\mathbf{D}_i$ only, (E_v, MSAE, Q) = (0.0156, 2.23, 0.71). (c) output by using $\mathbf{D}_{t_i}$ only, (E_v, MSAE, Q) = (0.014, 2.08, 0.7). (d) output by using $\mathbf{R}_i$ only, (E_v, MSAE, Q) = (0.0147, 2.13, 0.66). The terms E_v,, MSAE and Q are quantitative parameters and explained in Section 5.2.2.

In the proposed method, we discussed five different parameters: kernel widths of the closeness and the similarity functions (σ_c and σ_s), geometric neighbour disk (Ω), isotropic factor λ_I, and the number of iterations p. Throughout the algorithm, the radius of the neighbour disk is defined as $r = 2c_a$, where c_a represents the average distance between the centroid of faces. Furthermore, the kernel width of the closeness function is half of the radius of the neighbour disk ($\sigma_c = c_a$). The average centroid distance-based radius will make the algorithm robust against irregular sampling as denser regions will have more faces compared to sparse regions during the face normal processing. Effectively, the user has to tune only 3 parameters (σ_s, λ_I, and p) to get the desired result. In Table 5.1, the proposed method parameters are mentioned in the following format (σ_s, λ_I, p). Similarly, (τ, r, p) for [114], where τ is the feature threshold, r is the radius of the geometric neighborhood disk, and p is the number of iterations. The parameters (σ_c, σ_s, p) are for [34], (σ_s, p) are for [121], (λ, s, p) are for [45], and α is for [43], where σ_c and σ_s are the standard deviation of Gaussian function in the bilateral weighting. The variables s, λ represent the step size and the smoothing threshold [45] and α controls smoothing in [43].


(a) Noisy (b) $\sigma_s = 0.2$ (c) $\sigma_s = 0.5$ (d) $\sigma_s = 1.5$

Figure 5.8: Effect of the parameter σ_s on the Box model (magnified view), which has different level of features. The parameter λ_I is fixed to 0.05.

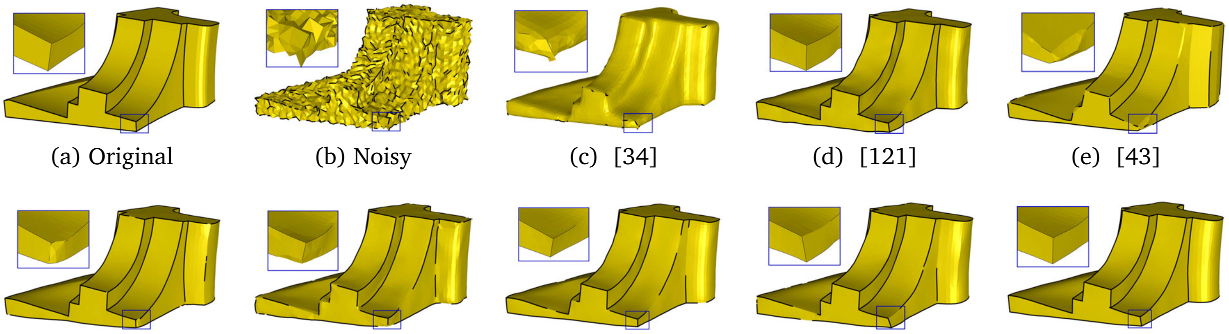

(a) Original (b) Noisy (c) [34] (d) [121] (e) [43]
(f) [120] (g) [106] (h) [67] (i) [114] (j) Ours

Figure 5.9: The Fandisk model corrupted with a Gaussian noise ($\sigma_n = 0.3 l_e$) in a random direction. Both rows show the results produced by state-of-the-art methods and our proposed method. The sharp feature curve (black curve) is computed using the dihedral angle $\theta = 70°$.

Figure 5.5 shows the effect of the parameters σ_s and λ_I. The black curve represents sharp features and is computed using the dihedral angle, $\theta = 65°$, between neighboring faces. In Figures 5.5 (b-d), the isotropic factor is fixed, $\lambda_I = 0.2$. The small value of $\sigma_s = 0.1$ produces piecewise flat regions and does not produce a noise-free surface with sharp features (Figure 5.5 (b)). For $\sigma_s = 1.5$, sharp features are smoothed out and for $\sigma_s = 0.5$, the algorithm produces an optimal result. Similarly, from Figure 5.5 (e-g), the parameter σ_s is fixed, $\sigma_s = 0.8$. For $\lambda_I = 0$, the reconstructed surface has several edge flips (faces with wrongly oriented normals) and the algorithm does not preserve all desired features (Figure 5.5(e)). However, the algorithm produces a high fidelity mesh with sharp features when we use $\lambda_I = 0.2$ (Figure 5.5(f)) and blurs sharp features with a bigger isotropic factor (Figure 5.5 (g)). In general, the value of λ_I depends on noise intensity on surfaces. The higher noise the more normal flips will be produced during the denoising process. Therefore, it is better to use a bigger value of λ_I to avoid these normal flips. At the same time, higher values for λ_I can produce edge blurring and volume shrinkage. Additionally, Figure 5.8 shows the effect of σ_s on the Box model, which has a different levels of features. As it can be seen, small-scale features disappear with a bigger value of σ_s. In the proposed method, λ_I is responsible to produce a high fidelity mesh against high-intensity noise and σ_s helps to preserve sharp features effectively.

Figure 5.6 shows a comparison with different weighting functions (Gaussian and *Tukey's bi-weight*) for face normal filtering and also represents a comparison between the proposed vertex update method and Method [95]. The quantitative parameters indicate that the proposed face normal filtering and the vertex update method better preserve features (better MSAE) and a high fidelity mesh (low value of Q). From the mean curvature colouring, it is clear that *Tukey's bi-weight* removes low frequency components of noise effectively and produces a uniform mean curvature

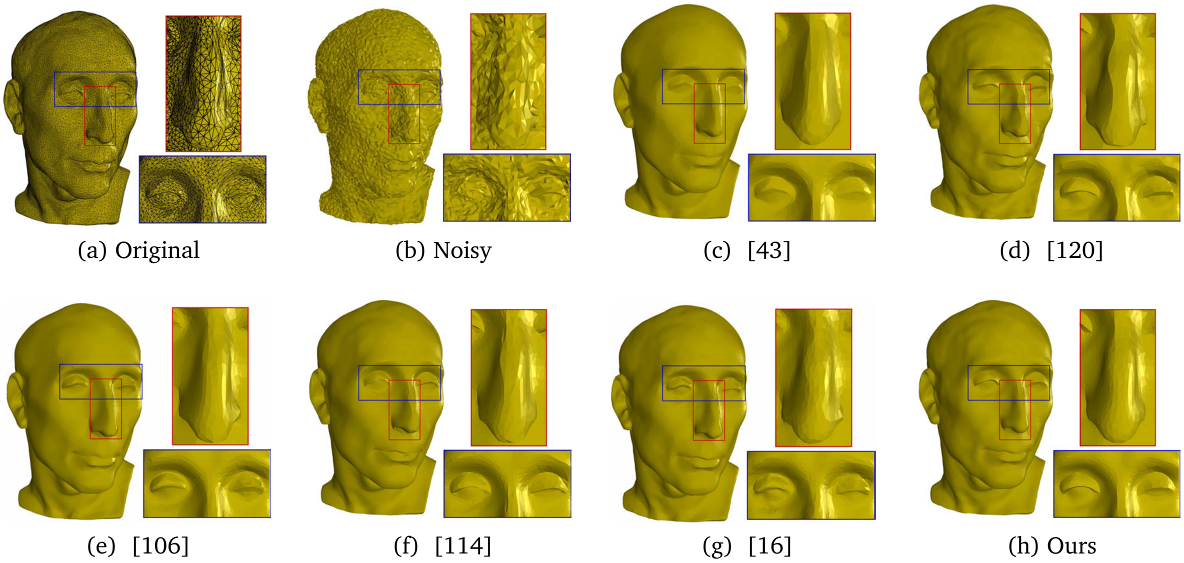


Figure 5.10: The Nicola model, a non-uniform triangulated mesh surface, which is corrupted by a Gaussian noise in a random direction. The magnified view of the eyes and the nose shows that the proposed method better preserves sharp features without creating piecewise flat areas compared to state-of-the-art methods.

on the cylindrical region of Ring model compared to the Gaussian weighting function. Similar comparisons are shown in Figures 5.9-5.12, where Method [121] uses a Gaussian function for the face normal filtering and for the vertex update, it follows Method [95].

5.2.1 Visual Comparison

Figure 5.9-5.14 represent a visual comparison of the proposed method with several state-of-the-art methods. Figure 5.9 shows the Fandisk model, which is corrupted with a moderate intensity Gaussian noise in a random direction. The sharp feature curve (black curve) shows that the proposed method preserves sharp features (especially at corners) effectively and does not produce false features in the cylindrical region compared to state-of-the-art methods. Method [114] and [67] manage to produce features at sharp corners but do not recover the sharp features on the whole surface. Method [43] creates false features in the cylindrical region and is not able to preserve the corner features [16]. The Nicola model (Figure 5.10) has a non-uniform mesh and is corrupted by a Gaussian noise in a random direction. The magnified view shows that our method preserves sharp features around the eye regions better than state-of-the-art methods and does not produce any false features around the nose area (piecewise flat areas). Figure 5.11 shows the Joint sharp model with non-uniform mesh corrupted with a Gaussian noise in the normal direction. As it is shown, the proposed method reconstructs the cylindrical region and corners without creating any false features. Method [114] produces a quite similar result to ours while Method [120] and [43] produce false features around the cylindrical region. The other state-of-the-art methods are not able to preserve sharp features. Figure 5.12 provides a visual comparison between state-of-the-art methods and the proposed method where models are corrupted with high-intensity noise. As shown in Figure 5.12, our method is able to remove noise components properly and retains sharp features without creating any edge flips. Method [121] is able to preserve sharp features but produces edge

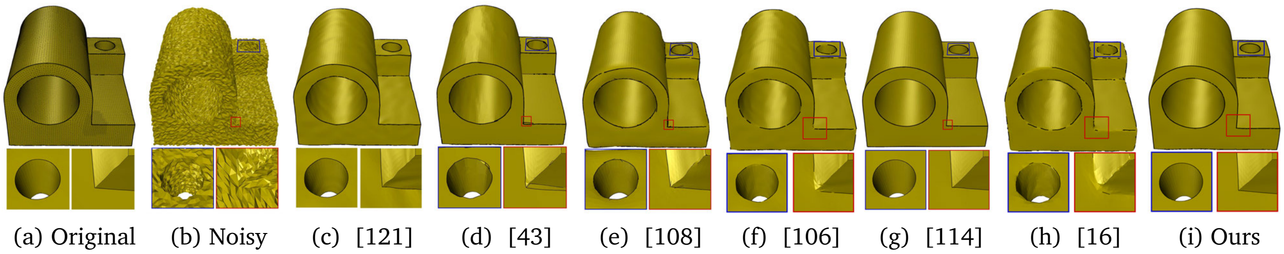

(a) Original (b) Noisy (c) [121] (d) [43] (e) [108] (f) [106] (g) [114] (h) [16] (i) Ours

Figure 5.11: Non-uniform triangulated mesh surface, which is corrupted by a Gaussian noise ($\sigma_n = 0.35 l_e$) in the normal direction. The first row shows the results obtained by state-of-the-art methods and the proposed method. The second row shows a magnified view of the corner and the cylindrical hole of the corresponding geometry.

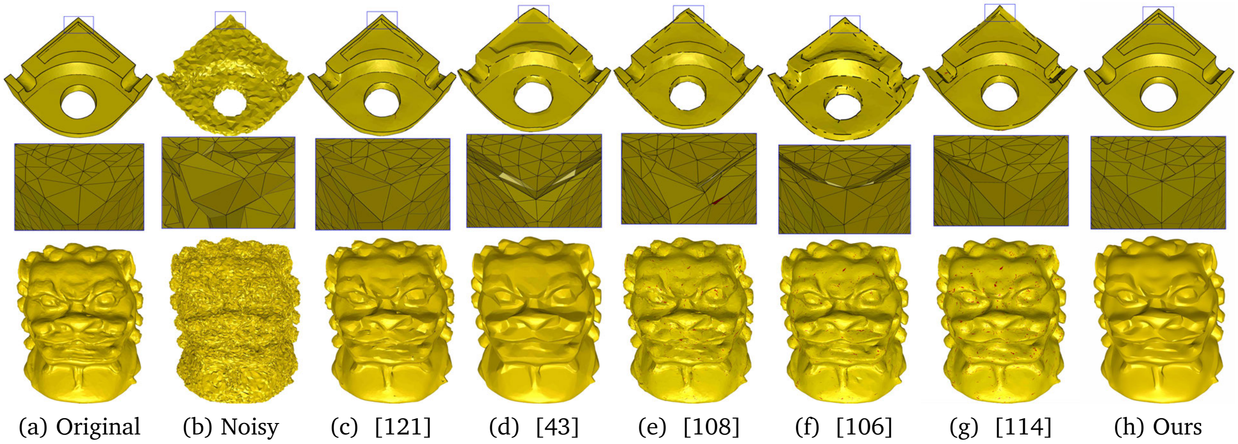
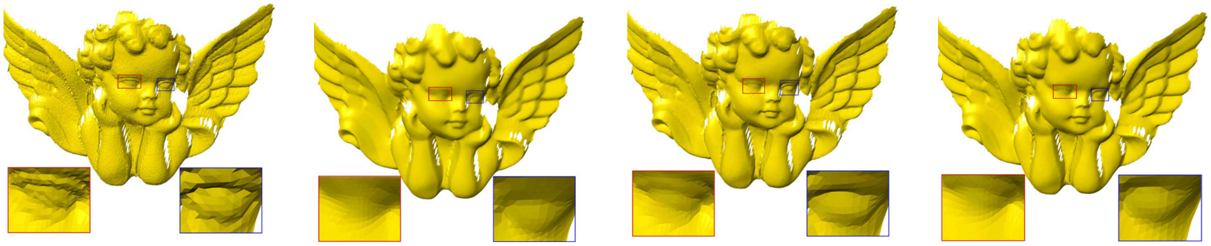

(a) Original (b) Noisy (c) [121] (d) [43] (e) [108] (f) [106] (g) [114] (h) Ours

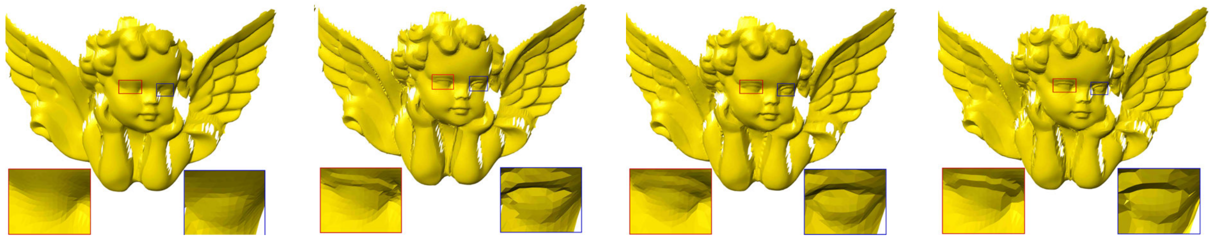
Figure 5.12: Figure (b) shows the noisy models, which are corrupted by a uniform noise ($\sigma_n = 0.5 l_e$) in a random directions. Figure (c)-(i) show the results produced by state-of-the-art methods and the proposed method. The second row shows the magnified view of the mesh quality at the sharp corner. The sharp feature curves are computed at $\theta = 65°$ (for the Bearing model) and the red color faces are with wrong orientation (flipped normals).

flips and the mesh quality is not optimal. Method [43] is using an isotropic regularization factor, which helps the algorithm to produce a good quality mesh on the planar areas but fails to produce similar triangles near the sharp edges (second row Figure 5.12(e)). Method [106] is not producing an optimal result because the algorithm is trained for low and moderate levels of noise.

In general, real data is captured using the 3D scanners and have a quite low intensity of noise. Thus, it is difficult to see a considerable difference between the results of the proposed method and state-of-the-art methods. Figure 5.13 shows that our algorithm is capable of better preserving features around both eyes compared to state-of-the-art methods. Methods [114] and [121] are able to retain the feature in the right eye but fail for the left eye. The other state-of-the-art methods are not able to preserve fine levels of features in both eyes. In case of the Pierrot model (Figure 5.14), our method removes noise effectively around the eyes and preserves features. Methods [67] and [114] retain similar scale of the feature but are not able to remove noise properly. Figure 5.15 shows the applicability of the proposed method to Kinect data. The David model is corrupted with lower noise compared to the Girl model and our method is able to remove these noise components



(a) Original (b) [43] (c) [120] (d) [67]



(e) [106] (f) [114] (g) [16] (h) Ours

Figure 5.13: The Angel model, a triangulated mesh surface (real data) obtained by a 3D scanner. The proposed method better preserves sharp features (both eyes) compared to state-of-the-art methods.

effectively while preserving features properly.

5.2.2 Quantitative Comparison

For the quantitative comparison, we are using three different quantities. Two of them show differences between the ground truth and smooth model, the so-called L_2 vertex-based positional error and mean square angular error (MSAE) respectively. The third parameter computes the mesh quality of a smoothed surface. The L_2 vertex-based positional error E_v from the original ground truth model is defined as [95]:

$$E_v = \sqrt{\frac{1}{3\sum_{k \in F} a_k} \sum_{i \in V} \sum_{j \in F_v(i)} a_j \mathrm{dist}(\tilde{\mathbf{v}}_i, T)^2},$$

where F is the triangular element set and V represents the set of vertices. The terms a_k and a_j are the corresponding face areas. The distance $\mathrm{dist}(\tilde{\mathbf{v}}_i, T)$ is the closest L_2-distance between the newly computed vertex $\tilde{\mathbf{v}}_i$ and the triangle T of the reference model.

The MSAE computes the orientation error between the original model and the smooth model and is defined as:

$$\mathrm{MSAE} = E[\angle(\tilde{\mathbf{n}}, \mathbf{n})],$$

where $\tilde{\mathbf{n}}$ is the newly computed face normal and $\mathbf{n}$ represents the face normal of the reference model. The term E stands for the expectation value.

Mesh quality of the surface mainly refers to the shape of the faces in the mesh. For a better mesh quality, the size and angles of the triangular face should not be too small or too large. To measure the quality of faces, we use the ratio of the circumradius-to-minimum edge length of the corresponding face [89]:

$$Q = \frac{1}{|F|} \sum_{i \in F} \frac{r_i}{e_i}$$

Models	Methods	MSAE	$E_v{\times}10^{-}3$	Q	Parameters
Nicola $\lvert F\rvert = 29437$ $\lvert V\rvert = 14846$	[34]	7.45	172.7	1.18	(0.3, 0.3, 30)
	[45]	6.269	76.87	1.00	(0.05, 0.05, 50)
	[121]	**5.25**	**66.07**	1.02	(0.3, 60)
	[43]	6.50	99.58	0.91	(1.4)
	[120]	5.82	69.63	1.05	(Default)
	[114]	5.900	72.8	1.02	(0.3, 0.15, 50)
	Ours	5.5	66.14	**0.9**	(1.6, 0.06, 25)
Bearing $\lvert F\rvert = 6950$ $\lvert V\rvert = 3475$	[121]	3.17	0.590	1.45	(0.3, 50)
	[43]	4.70	1.24	1.58	(4.0)
	[108]	3.95	0.783	2.62	(Default)
	[114]	4.43	0.698	1.66	(0.3,0.08,80)
	[106]	5.67	0.691	1.73	(Default)
	Ours	**2.32**	**0.588**	**0.74**	(0.8,0.2,80)
Joint $\lvert F\rvert = 52226$ $\lvert V\rvert = 26111$	[34]	3.777	0.530	0.83	(0.3, 0.3, 40)
	[45]	2.630	0.766	0.78	(0.009, 0.05, 50)
	[121]	1.808	0.263	0.77	(0.4, 100)
	[43]	1.768	0.500	0.75	(1.4)
	[120]	0.956	0.179	0.83	(Default)
	[108]	2.874	0.366	1.67	(Default)
	[67]	1.16	1.49	**0.71**	(Default)
	[114]	**0.829**	**0.171**	0.91	(0.3, 0.05, 60)
	Ours	1.032	0.198	0.73	(1.0, 0.2, 150)
Fandisk $\lvert F\rvert = 12946$ $\lvert V\rvert = 6475$	[34]	8.567	4.422	0.87	(0.4, 0.4, 40)
	[45]	5.856	4.910	0.79	(0.07, 0.05, 30)
	[121]	2.727	1.877	0.76	(0.4,70)
	[43]	4.788	5.415	0.766	(1.4)
	[120]	2.221	**1.702**	0.97	(Default)
	[67]	3.1	4.42	0.7	(Default)
	[114]	2.692	1.964	0.89	(0.3, 0.2, 50)
	Ours	**2.201**	1.844	**0.7**	(0.55, 0.2, 100)
Chinese Lion $\lvert F\rvert = 118276$ $\lvert V\rvert = 59140$	[121]	9.98	19.64	3.12	(1.0, 100)
	[43]	8.93	25.34	1.51	(1.4)
	[108]	19.22	21.5	2.38	(Default)
	[114]	21.39	21.89	2.16	(0.3, 1.0,100)
	[106]	15.16	19.48	2.94	(Default)
	Ours	**8.795**	**18.35**	**0.87**	(0.4, 0.3, 100)

Table 5.1: Quantitative Comparison

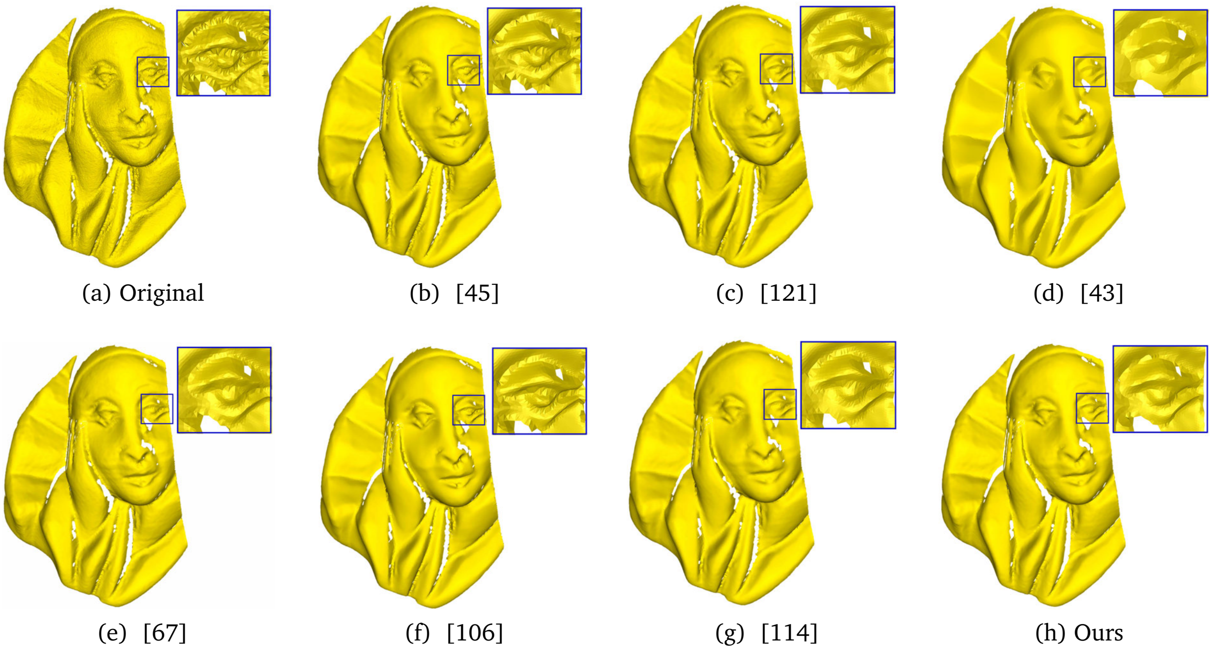

(a) Original (b) [45] (c) [121] (d) [43]
(e) [67] (f) [106] (g) [114] (h) Ours

Figure 5.14: The Pierrot model (real data) is corrupted by a 3D scanner noise. The figures shows the results obtained by state-of-the-art methods and the proposed method.

where r_i and e_i are circumradius and minimum edge length of the corresponding triangle. Ideally, every face belonging to the mesh should be an equilateral triangle with a quality index of $Q = 1/\sqrt{3} \approx 0.58$.

Table 5.1 shows the quantitative comparison of our method with state-of-the-art methods. For the models which are corrupted by high-intensity noise (Bearing, Chinese Lion, and Sharp Sphere), the proposed method has lower values of E_v, MSAE and Q compared to Methods [121] and [43]. Method [121] produces a quite similar value of E_v but bigger values for MSAE and Q because of the several edge flips. He et al. [43] produces a good quality mesh in planar regions but does not produce a similar quality of triangles near sharp features, which leads to a higher value mesh quality index. For the Nicola model, our method produces quite similar values for the position

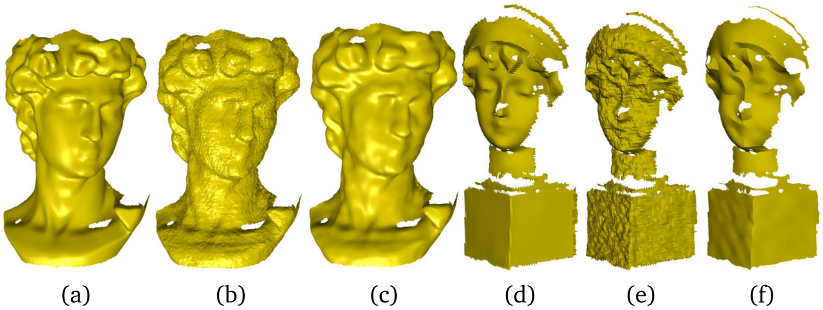
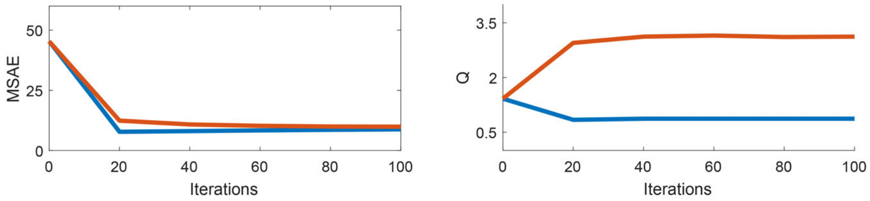

(a) (b) (c) (d) (e) (f)

Figure 5.15: Figure (a) and (d) are the original models. Figure (b) and (e) represent the scans captured by Kinect sensors [106]. The smooth outputs are shown in Figure (c) and (f) using the proposed algorithm.



(a) MSAE (b) Mesh Quality Factor Q

Figure 5.16: Convergence plots for MSAE and the mesh quality factor Q for Lion model. The red and blue curves represent Method [121] and the proposed method respectively.

and orientation error (slightly bigger) to Method [121], because our algorithm also follows the bilateral filtering technique and in addition, we use an isotropic factor which introduces a small amount of volume shrinkage. At the same time, it produces a good quality mesh. Figure 5.16 shows a convergence rate comparison with Method [95]. As it can be seen, the proposed method not only converges faster but also has a lowe and consistent mesh quality factor Q because of the edge-weighted differential coordinate.

Table 5.2: Running Time (in seconds)

Models	Fandisk	Bearing	Nicola	Joint	Lion
Time (s)	5.8	4.2	4.3	45.3	92

As it is explained by [114], the two-stage denoising methods have quite a similar running time complexity. The proposed algorithm also follows the two-stage denoising scheme and has the complexity of $O(n_c \cdot n_f \cdot p)$, where n_c is the number of faces within the neighbourhood, n_f and p are the numbers of faces and iterations respectively. The implementation of the algorithm is quite straightforward and simple. First of all, we compute the smooth face normals using Equation (5.1) then rearrange vertices by following Equation (5.9). Our implementation is done in a single computation thread using Java. Table 5.2 shows the running time of our algorithm for different models according to the parameter values mentioned in Table 5.1.

5.3 Summary

This chapter presented a robust and high fidelity mesh denoising algorithm, which is robust in terms of feature preservation because of the robust statistics framework. It produces a high fidelity mesh, which is measured using a mesh triangle quality (Q) and minimum number of normal flips. In the first step of the proposed method, the face normal filtering is done using *Tukey's bi-weight function* as a robust estimator. *Tukey's bi-weight function* completely stops the diffusion across sharp edges and removes noise along sharp features and from flat areas. In the second step, vertex positions are updated using edge-face normal orthogonality constraints along with differential coordinates. Differential coordinates help the algorithm to produce a high-quality mesh, makes it robust against high-intensity noise and avoids normal flips during the denoising process. Experimental section shows the robustness of the proposed algorithm not only against different kinds and levels of noise but also against face normal flipping. The quantitative comparison table shows that our method produces better mesh quality compared to state-of-the-art methods. Similarly, for the positional and the orientation error, the proposed method is either better or similar to state-of-the-art methods.

Chapter 6

Introduction to the Retinal Shape Analysis

This is the second part of the thesis, which is mainly focused on applications of geometry processing algorithms for the shape analysis of the human retina in different diseases. This part of our work is mainly motivated by the need of automatic shape quantifying algorithms, which would aid in the diagnosis of several neurological disorders by detecting shape changes in the retina. Before going to a detailed explanation of the retinal shape analysis algorithms, here, we provide an overview of the anatomical background of the human eye including the different retinal layers, the optic nerve head and the fovea. Then, the basic working principle of OCT (optical coherence tomography), which is a non-invasive, high-resolution retinal imaging technology to provide the underlying imaging data, is explained. Additionally, neurological disorders, which affect the shape of the retina in different regions, are discussed. In continuation, we also discuss about state-of-the-art methods related to the shape analysis of the two most important regions of the retina: the fovea and the optic nerve head.

6.1 Structure of Human Eye

The eye is one of the most complex organs of the human body. The working principle of the human eye is quite similar to a camera. It is self-focusing, adjusts automatically for different light intensity, has a self-cleaning lens, and feeds signals to the brain with parallel-processing capabilities. The human eye mainly consists of three different layers [109]:

1. The outer layer (the cornea and the sclera)
2. The middle layer (the iris, the ciliary body and the choroid)
3. The inner layer (the retina)

In the following, we describe in more detail the most relevant parts of the retina, which are the retinal layers and this thesis is mainly focused on the morphometry of these particular regions.

6.1.1 The Retina (The Inner Layer)

The retina (see Figure 6.1) is the sensory membrane, which is approximately $0.5mm$ thick and lines the inner surface of the back of the eyeball. The retina is a complex structure, which consists of several layers of cell bodies and cell processes and each layer serves a specific purpose. It consists of different types of neuron, photoreceptors, bipolar cells, horizontal cells, amacrine cells and ganglion cells, which are responsible for transmitting the signals to the brain. It is considered as a part of the central nervous system as it is created during embryogenesis from neural tissue. It has three

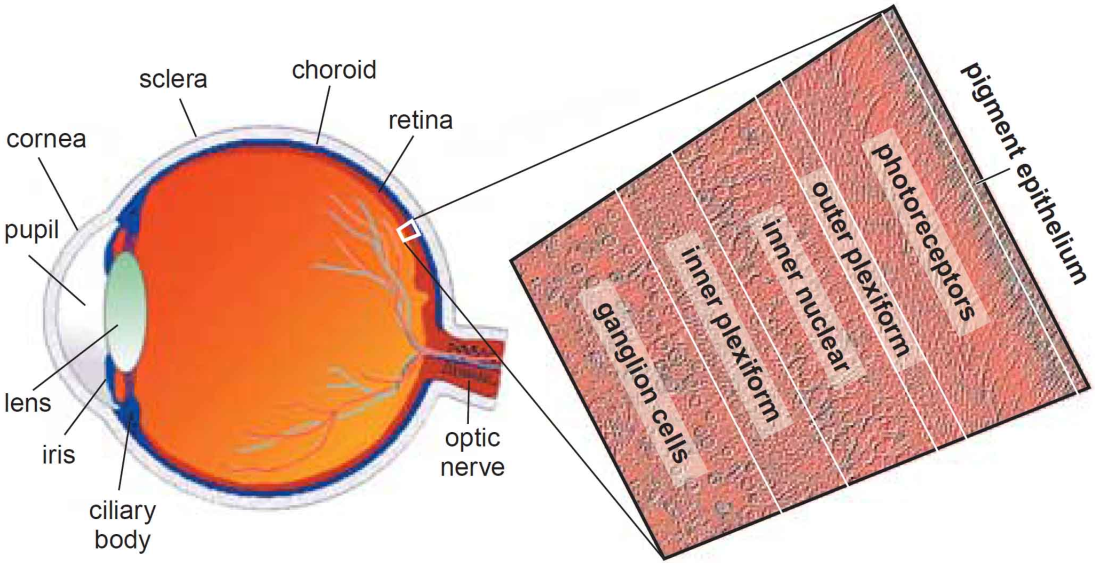


Figure 6.1: The human eye structure [56]

rows of the nerve cell bodies separated by two layers as shown in Figure 6.2. The retina includes photoreceptors that take light focused by the cornea and lens and convert it into chemical and nervous signals which are transported to visual centers in the brain by way of the optic nerve.

Figure 6.3, show a cross sectional of the retina with corresponding layers. A detailed explanation about each layer is given as:

1. The Inner limiting membrane (ILM) is the innermost boundary of the retina and defines a boundary between the vitreous body and the retina. It is formed by astrocytes and end feet of the mueller cells.

2. The retinal nerve fiber layer (RNFL) is formed by the retinal ganglion cell axons.

3. The ganglion cell layer (GCL) consists of nuclei of ganglion cells. Ganglion cells are connected to nerve fibers. In humans, there are two types of ganglion cells, ON center and OFF center, which are responsible for the major output of the retina to visual centers of the brain. ON center ganglion cells are activated when light falls on the receptive filed and OFF center ganglion cells are activated when light falls on the periphery of the receptive field. So, ganglion cells transmit two kinds of signals, dark-on-light (OFF) and light-on-dark (ON), which create a train of spikes (informations) into the brain [56].

4. The inner plexiform layer (IPL) consists of synaptic connections between the axons of bipolar cells and dendrites of ganglion cells.

5. The inner nuclear layer (INL), as it is shown in Figure 6.3, not only consists of the cell bodies of horizontal cells, bipolar cells and amacrine cells but also interplexiform neurons, Müller cells, and sometimes displaced ganglion cells. Bipolar cells receive chemical signals from the photoreceptors and pass it to the ganglion cells and amacrine and horizontal cells send signals using various excitatory and inhibitory chemicals. Horizontal cells have a bigger receptive field compared to bipolar cells because each horizontal cell receives signals from many cones and a single bipolar cell is connected to a small numbers of cones. As a result, horizontal and

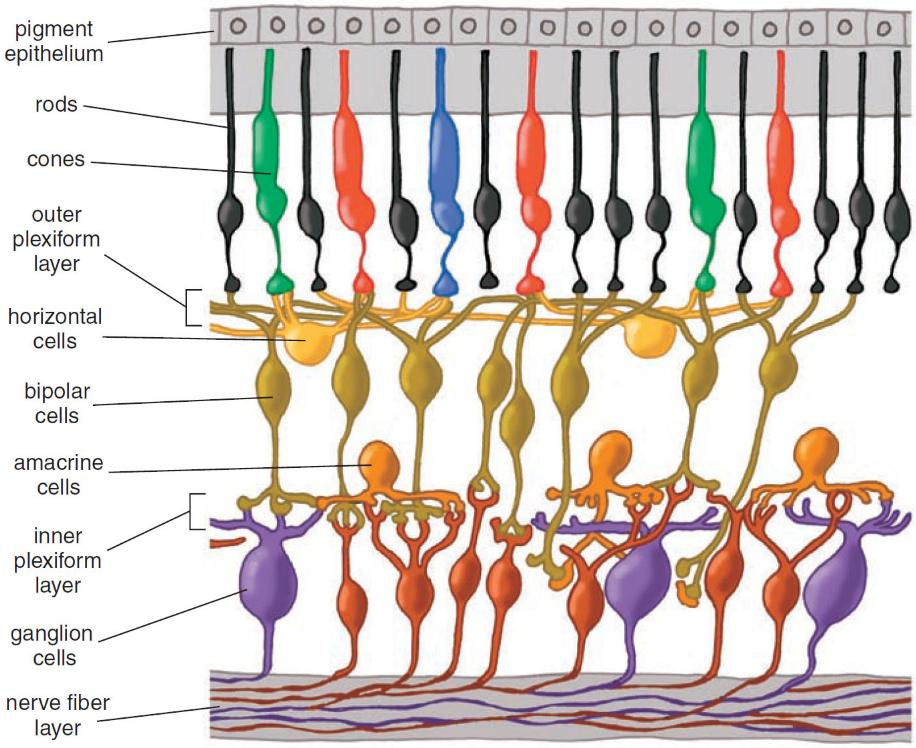


Figure 6.2: Layers [56]

amacrine cells modulate the photoreceptors signal and provide additional information to the ganglion cell.

6. The outer plexiform layr (OPL) is the region, which is connecting the photoreceptors to the bipolar and horizontal cells.
7. The outer nuclear layer (ONL) is the light detecting part of the retina and it contains the cell bodies of the photoreceptors, cones and rods. Both rods and cones respond to light and are responsible for quite different image properties. Rods, which are active with low light levels and approximately 20 times more abundant than cones [109]. Cones are responsible for color vision and about 50 percents of the cones are located around the macula region.
8. The retinal pigment epithelium (RPE) is a single layer of cells between the retina and choroid that provides essential nutrition and waste removal for the photoreceptor cells.

6.1.2 The Fovea

In this section, we discuss about one of the most important regions of the retina for this work, the fovea. The fovea is an avascular zone in the macula of the retina. This area is approximately $1.5 - 2mm$ in diameter within the larger macula where the retina thins out greatly. Although the fovea represents a tiny anatomical region of the retina, the number of cortical neurons per unit of visual field devoted to foveal input is much larger and contains more neurons than the network devoted to perifoveal vision. The density of the photoreceptors per unit area in the fovea provides exquisite visual acuity, and interactions between these and other retinal neurons are responsible for high contrast sensitivity. The inner retinal cellular elements at various distances possibly determines

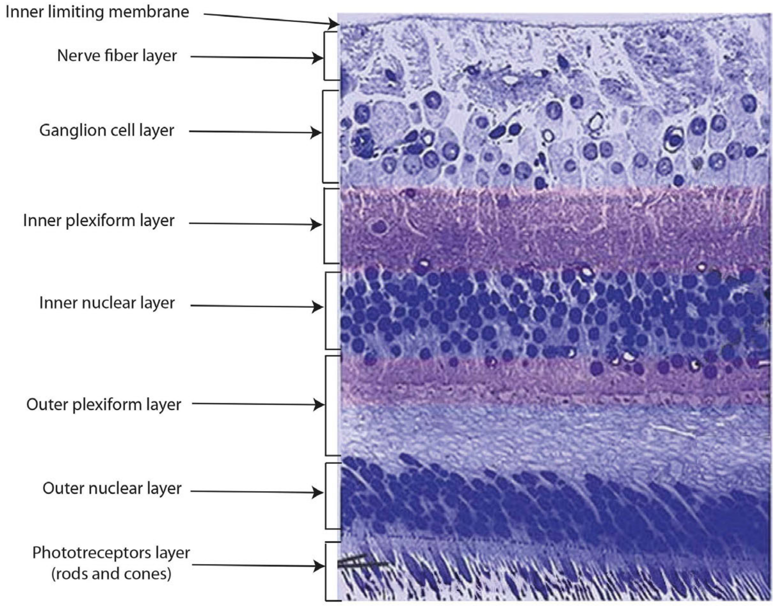


Figure 6.3: Retinal layers [56]

the appearance of the shape of the pit. The ganglion cells and the innermost layers are pushed aside. At the center of the foveal pit, only four retinal layers are present, the ILM, the OPL, the ONL and the photoreceptors as shown in Figure 6.5.

6.1.3 The Optic Nerve Head (ONH)

The optic nerve head is a circular area in the back of the inside of the eye where the optic nerve connects to the retina. At this location, 1.2 to 2.0 million ganglion cell axons converge and turn 90° to start their exit from the eye. This region is also known as the blind spot because no photoreceptors are available. The ONH is also the entry and exit point as well for a number of large blood vessels that supply the retina as shown in Figure 6.4(A). The ONH is also known as the optic disc and it has a slightly oval structure, which consists of a peripheral rim composed of neural tissues and the central depression (not always), the so-called cup. The region, which has neural tissues and excludes the cup is known as the disk margin. The disk margin is an important parameter to quantify the neural tissue in the ONH region and also the size of the ONH. The size of the physiologic cup is developmentally determined and is related to the size of the disc. In general, the size of the optic disc follows a near Gaussian distribution and in the non-highly myopic white population the mean optic disc area is $2.5mm^2$.

The Bruch's membrane opening (BMO) is the another important morphological parameter in the ONH region. The BMO is also known as the neural canal opening and occurs at the level of the RPE and the Bruch's membrane (BM). The outer border of the optic nerve is defined by the termination of BM around the optic nerve, forming a hole called the BMO. This is the physiological opening for axons of the retinal ganglion cells to exit the eyes globe.

6.2 Retinal Imaging using Optical Coherence Tomography

Optical Coherence Tomography (OCT) is a non-invasive diagnostic technique that provides a cross sectional view of the retina with high resolution. This imaging technique was introduced in 1991

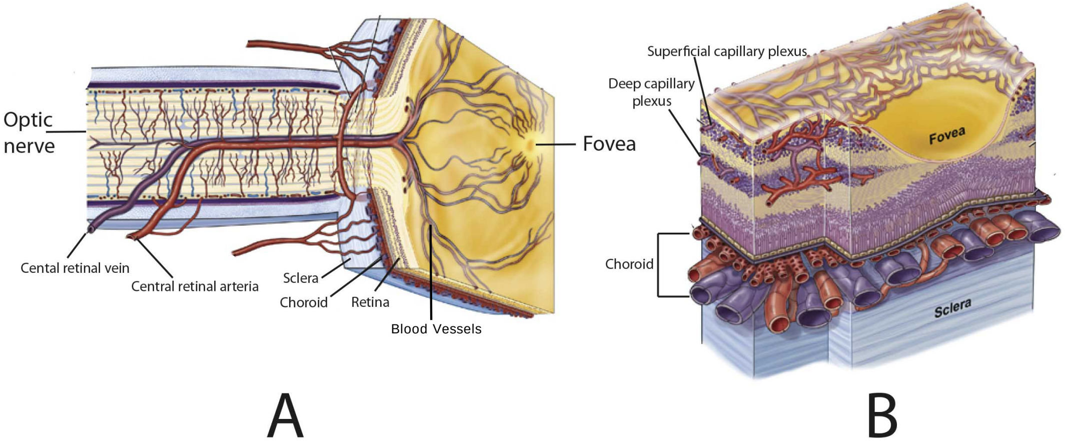


Figure 6.4: The ONH and the fovea shape [6].

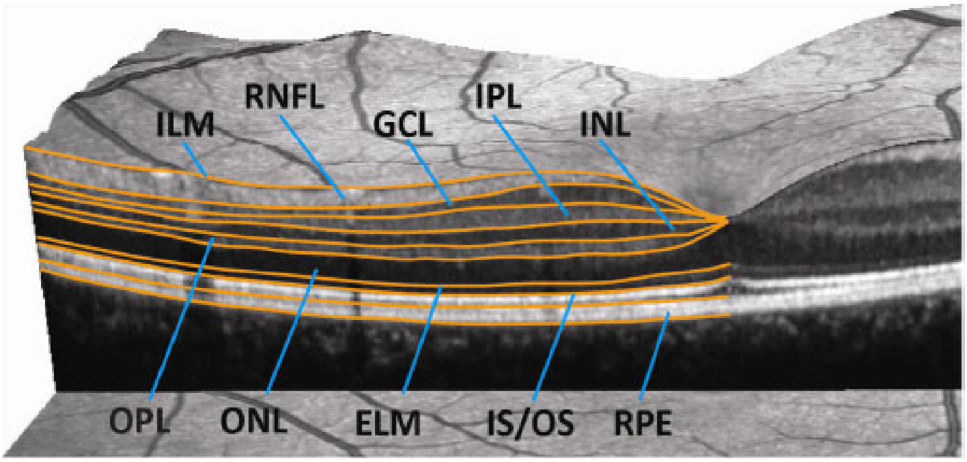


Figure 6.5: Layers obtained by a OCT device [124].

by Huang et al. [47]. OCT creates cross-sectional image by measuring the echo time delay and intensity of the reflected and backscattered light. The first OCT images were produced in a time domain fashion and the corresponding device is known as time-domain OCT (TDOCT). TDOCT was capable to produce 400 A-scans per second using 6 radial scans oriented 30° apart. Later, SDOCT (spectral domain OCT) has been introduced. The advantage of SDOCT over TDOCT is that the interference pattern is split by a grating into its frequency components and all of these components are simultaneously detected by CCD sensors. Compared to TDOCT, SDOCT is capable of aquiring 3D image data in very short time and with high accuracy. Figure 6.5 shows a section of a volume from the macular region with multiple layers highlighted by the yellow lines acquired with an SDOCT device. Most of the SDOCT devices add eye tracker systems to avoid the movement artifacts. Currently, the most common four commercially available SDOCT devices are, Cirrus HD-OCT (Carl Zeiss Meditec, Dublin, CA, USA), Spectralis OCT (Heidelberg Engineering, Heidelberg, Germany), RTVue-100 (Optovue Inc., Fremont, CA, USA), and Topcon 3DOCT 2000 (Topcon Corporation, Tokyo, Japan).

6.3 Neurological Disorders and their Effect on Retinal Shape

Optic neuritis (ON), multiple sclerosis (MS), neuromyelitis optica spectrum disorders (NMOSD), Ideopathic Intracranial Hypertension (IIH) among many neurological disorders affect the macula, the fovea and the ONH. While diagnosis of these diseases is done in a variety of ways and often established by using several diagnosing techniques, such as magnet-resonance imaging (MRI) and antibody tests, it is also known that many of these diseases can be detected evaluating OCT data sets of the retina. Despite a variety of approaches in OCT to study and quantify the changes that the retina undergoes in these disease, currently there is a lack of methods for quantifying these morphologically. In this section we present several characteristics of disorders that are the focus of this thesis and the features that have been captured in previous studies by OCT measurements. Note, that compared to ophthalmological conditions, changes in neurological disorders can be often in the rage of a few μm which poses a particular challenge in developing method that are able to quantify these changes morphologically.

6.3.1 Multiple Sclerosis (MS)

Multiple sclerosis is the most common chronic inflammatory disorder of the central nervous system of young adults. The most common symptoms in MS are visual disturbances including blurred vision, visual field defects and color de-saturation and it is caused by the axonal damage and the the inflammation to the optic nerve as part of the CNS. When these causing effects exceed a threshold, clinically it is termed as the optic neuritis (ON) [124]. There are four courses of MS: relapsing-remitting MS (RRMS), secondary-progressive MS (SPMS), primary-progressive MS (PPMS), progressive-relapsing MS (PRMS). In RRMS, which is the most common MS course, the patient suffers from attacks of new or increasing neurologic symptoms and these attacks are called relapses, which are followed by periods of partial or complete recovery (remissions). During remissions, all symptoms may disappear, or may continue and become permanent. PPMS is characterized by accumulation of disability from the first symptoms, without any relapses or remissions. SPMS is a combination of RRMS and PPMS. Patients, who are diagnosed with RRMS may eventually transition to a secondary progressive course in which there is a progressive worsening of neurologic function over time. Despite the high costs for the treatment per patient and per year there is no drug that can cure MS.

Standard parameters for assessing structural changes of the retina in MS are the peripapillary RNFL thickness from a ring scan centered at the ONH, total macular volume (TMV) of all layers, as well as thickness of individual layers. Retinal damage has been noticed during the course of MS. A 20% loss of RNFL thickness is expected after ON because of the retrograde degeneration of retinal axons as a consequence of an inflammatory attack to the optic nerve [124]. After initial swelling, due to edema in the acute phase, degeneration of the retina especially in the RNFL occurs [3]. A significant thinning is noticed within the GCL and IPL complex (GCIPL) also after ON and it is the more reliable parameter for longitudinal monitoring of ON-related retinal damage [124]. In case of non-ON, RNFL thinning is happening compared to healthy controls (HC) and SPMS patients are more severely affected compared to RRMS and PPMS patients. In case of non-ON, the RNFL and GCL reduction is caused by the ”dying back” of retinal ganglion cells [124]. Also SDOCT studies in clinically isolated syndrome patients, which is the first potential herald of a developing MS, show that degeneration of retinal neurons occurs very early in the course of the disease [74]. Although still a very active research topic, many studies have shown a relationship between RNFL and GCIPL thickness reduction and several MRI-derived measurements of brain atrophy [38, 123]. A meta-analysis that investigated OCT layers association with neurodegeneration in MS in articles form

1991 to 2016, found that the largest and most robust differences between the eyes of people with MS and HC eyes are in the peripapillary RNFL and macular GCIPL, which mkes these layers perfect candidates for diagnosis and monitoring, while INL might capture inflamatory disease activity [28].

6.3.2 Neuromyelitis Optica Spectrum Disorder

Neuromyelitis optica (NMO) is a rare autoimmune inflammatory disorders of the central nervous system but an important differential diagnosis of MS. It is characterized by severe, immune-mediated demyelination and axonal damage predominantly in the optic nerves and in the spinal cord. The discovery of a disease-specific serum NMO-IgG antibody that selectively binds aquaporin-4 (AQP4) has led to an increased understanding of a diverse spectrum of disorders. As treatment options for NMO and MS are quite different, it is vital to distinct NMO from MS. From retinal damage point of view, NMO causes severe neuronal damage and greater thinning of the RNFL compared to MS in ON. Basically, the thinning of the peripapillary and macular RNFL, and of the GCL is more severe in ON eyes in NMO compared to MS [88]. NMOSD frequently presents bilaterally and even simultaneously ON and the risk of recurrence is higher than in MS [98]. In a recent study [75] evidence of microstructural changes for a specific type of NMOSD without ON in the fovea reagion was presented. These changes lead to modifications in the shape of the fovea visible in SDOCT macula volume scans.

6.3.3 Ideopathic Intracranial Hypertension (IIH)

Ideopathic Intracranial Hypertension is also known as pseudotumor cerebri and it is characterized by an increased pressure of the cerebrospinal fluid surrounding the brain, which is known as intracranial pressure (ICP). In general, this condition occurs to young and obese women of childbearing age (20-44 years). The main symptoms of IIH are headache, visual disturbances, pulsating tinnitus, photopia, eye pain, diplopia and nausea. Papilledema with subsequent visual field loss is the most feared clinical consequence, which mainly determines the therapy and outcome of the syndrome. Patients with newly diagnosed IIH present RNFL thickening compared to HC, which decreased after three months under IIH treatment, visible and quantifiable in the ring scan with OCT [44]. ONH presents moderate up to extreme swelling and correlates with cerebrospinal fluid pressure (CSF) in IIH patients compared to HC, which can be quantified by using SDOCT [52, 53]. The increased volume of the ONH is associated with lower visual acuity. Moreover SDOCT, particularly 3D ONH volume measurements, in a clinical trial setting, provide reliable measurement of the effects of papilledema on the ONH [39], as well as in longitudinal studies, ONH volume could be employed for therapy monitoring [2].

6.4 Previous Works

This part of thesis is mainly focused on the shape analysis of two structures of the retina: the fovea and the ONH. In this section, we will give an overview of the related works in the fovea and the ONH morphometry.

6.4.1 Fovea 3D Shape Analysis

Foveal shape analysis is a promising approach for quantitative OCT next to thickness or volume measurements. The few studies in this regard can be divided into two categories: data-driven, which directly employ segmentation lines from the OCT scan itself to compute various metrics, and model-driven, which use mathematical constructs to create a representation of the macula and fovea in order to calculate different features directly from the model.

Data-driven approaches are based on the analysis of 2D images (B-scans) with results computed from one B-scan or averaged over two B-scans. To analyze the variability of the healthy foveal shape and to investigate the relationship between this structure and the foveal avascular zone (FAZ), Tick et al. computed several parameters: pit depth, central foveal thickness, maximal retinal thickness, pit diameter, pit cross-sectional area, and the foveal inner retinal area. Using these parameters, the authors could show that the healthy foveal structure strongly correlates with its neurovascular structure [102]. Similarly, Chiu et al. introduced the parameters foveal photoreceptor thickness and foveal width, and found a large individual variation of FAZ size and shape in healthy retinae and a negative correlation between FAZ diameter and foveal thickness [19]. Other studies investigated the way retinal structures change in the context of prematurity [41] or albinism [69] using fovea shape parameters like pit depth.

One of the first attempts to mathematically describe the foveal shape was published by Barak et al. [8]. Their approach employs an automated symbolic regression software that fits a section of the foveal profile around the center which was able to detect different patterns in the premacular hole foveal configurations and normal foveal configurations. A similar approach was used by Nesmith et al. to characterize changes that occur in the foveal anatomy with aging in a large number of OCT scans [73]. The first more general mathematical model was created by Dubis et al., who used Difference of Gaussians (DoG) function for a 2D fit of several radial scans passing through the lowest point of a macular scan [23]. The method derived three pit metrics (diameter, depth and slope) from a fitted model, which was symmetrical with respect to the fovea center. The model's symmetry was problematic, because of the fovea's asymmetry, which stems mostly from differences between the nasal and temporal retinal nerve fibers [87, 102]. The model was thus not accurate enough for modeling the asymmetric nature of the fovea and had problems covering the wide range of fovea shapes in the healthy population, due to other model fitting constraints, which were e.g. not able to model an extended flat part at the foveal pit in many healthy foveae [87, 102]. Scheibe et al. applied a different approach to overcome these drawbacks by creating a more flexible 2D model, which is fitted to each supporting direction in a circular region around the foveal center [86]. The authors derived five parameters analytically: mean retinal thickness inside a radius of $1mm$, foveal bowl area, retinal radius, and maximum height of the foveal rim. By applying this model to a large cohort of healthy subjects, the authors were able to confirm and further investigate the asymmetric nature of the fovea. Although, capable of modeling a large variety of data, and having a smaller root mean square error (RMSE) than previous approaches, the method has difficulties deriving 3D parameters such as volume due to the complexity of the computation. Liu et al. created a different 2D model, which uses a sloped piecemeal Gaussian function (SPG) to model the asymmetry and the foveal flatness [64]. The authors tested their method on a large number of macular scans and showed good RMSE compared to other 2D methods. The main drawback consists in the use of only two scans for each subject, a vertical and a horizontal one, which limits information on only these sections and prohibits generation of 3D metrics like volume. Ding et al. developed an approach, which intends to model the surface of inner retinal layers using a 3D model-fit based DoG and a second order polynomial [22]. The authors used the model coefficients to discriminate between Parkinson patients and a control group. As a drawback, the model coefficients' relation to foveal shape itself is difficult to interpret. Additionally, although the model takes into account the difference in the slope for horizontal and vertical directions, it does not model the asymmetry in the anatomical ones (temporal-nasal), nor does it capture fovea shapes with very flat pits accurately.

6.4.2 Optical Nerve Head 3D Shape Analysis

Analyzing tissue damage and structural changes in the ONH is one of the key goals to improve the diagnosis and understanding of diseases related to this structure. The main focus of ONH research lies in the field of ophthalmology, the most prominent topic being glaucoma. The most common parameters utilized include length, areas, volumes or ratios to quantify various regions of the ONH. The main anatomical structures needed to compute these parameters are the inner limiting membrane (ILM), Bruch's membrane or the lower boundary of the retinal pigment epithelium (denoted throughout the paper, for simplicity, by RPE) and the Bruch's membrane opening (BMO) points. The first two structures comprise the retina, while the BMO points have recently gained more and more attention since Reis et al. [85] introduced these as being the true anatomical structure defining the optic disc as to the clinically identified margin using fundus photography. Following this work, several other groups [17, 118] proved BMO based parameters superiority in reliability compared to cup-to-disk ratio.

For ONH analysis, Enders et al. [26] and Daniel et al. [71] both used manual segmentation of the BMO points and minimal rim width (BMO-MRW), as well as commercially available ILM surface detection with manual correction to study the correlation between visual fields and structural ONH changes. Chauhan et al. [17] proposed BMO-MRW as a marker for early glaucoma detection also using manual segmented BMO points and BMO-MRW measurements, while Pollet et al. [84] used a similar manual process to prove that structure-function relationship was significantly stronger using BMO-MRW over other ONH parameters derived from spectral domain OCT. Furthermore, several studies focused on the description of normative values that characterize the shape of the ONH. Chauhan et al. [18] analyzed BMO-MRW, orientation of the long axis of BMO in a normal population, while Enders et al. described BMO-MRW in micro- [26] and macrodiscs [27]. BMO minimum rim area (BMO-MRA) correlates with the total number of nerve fibers traversing the optic nerve and was introduced and investigated in [25, 35]. Both studies have shown BMO-MRA's high diagnostic power for glaucoma. All these studies used commercial available OCT software to retrieve the ILM surface and BMO points, which were then manually controlled and corrected by experienced graders.

Another important parameter is the ONH volume, previously analyzed mainly in relation to conditions characterized by ONH swelling. To this end, a 2D segmentation has been previously developed by our group. The method [52] is able to robustly detect the lower boundary of the RPE in healthy as well as in ONHs from patients suffering from various neurological disorders that lead to swelling of the ONH. Several other publications investigating the same condition [39, 105] followed, all using a graph-cut based approach to segment the ILM and BM at the ONH. Lee et al. [61] introduced an end-to-end pipeline to compute several morphometric parameters (volumes in different ONH regions and related area parameters). Furthermore, a surface correspondences approach to create a normalized space was presented. However, this pipeline is semiautomatic and needs manual segmentation of the ILM, BM and BMO points. For the development of the mean surfaces, an accurate registration between the ONH surfaces is needed. Eli et al. [37] introduced an ONH registration algorithm to compute the one-to-one correspondence between two ONH surfaces using the hemispherical surface and volume registration. Later, Lee et al. [60] proposed a more sophisticated registration algorithm based on surface currents and a hemispherical demons. Recently, the shape variability of RNFL-choroidal thickness was investigated using a nonrigid surface registration for longitudinal analysis [59].

6.5 Contributions

In **fovea morphometry**, most of the published studies assume that the foveal pit shape follows a Gaussian function-like shape and use the this function for modeling the fovea. However, this assumption is inaccurate, and one of the goal of this thesis was to develop a robust, model-driven 3D macula shape analysis method, which can be computed from standard macular volume OCT scans and from which foveal and macular shape metrics can be derived. In addition to available macular thickness measurements, the developed approach allows a detailed analysis of foveal shape, including depth, diameter, slope, area and volume of different regions, as well as pit shape analysis.

In case of ONH morphometry, most published methods related to **ONH morphometry** are computing traditional parameters (lengths, widths and volumes) and performing layers segmentation either manually or semiautomatically. To the best of our knowledge, none of the published ONH morphometry methods use proper manifold 3D ILM or BM surfaces, but rather 2.5D surface (i.e. a graph function on an XY-grid).

Specifically, the contributions of this part of thesis are:

1. Chapter 7 introduces a parametric modeling of the foveal shape using Cubic Beziér and derives several 3D shape parameters, which quantify the foveal shape with high accuracy. Later, it is shown that the proposed 3D parameters are capable to differentiate between the foveal shape of HC and patients (neuroinflammatory disorders).

2. A 3D shape analysis algorithm is introduced to measure the shape variation in the ONH in different neurological disorders. The proposed method uses manifold surfaces of two different layers (the ILM and the RPE) of the retina to derive several 3D shape parameters (Chapter 8). In the experimentation section, it is shown that the proposed ONH parameters are capable to differentiate between HC and patient (IIH and ON).

Chapter 7

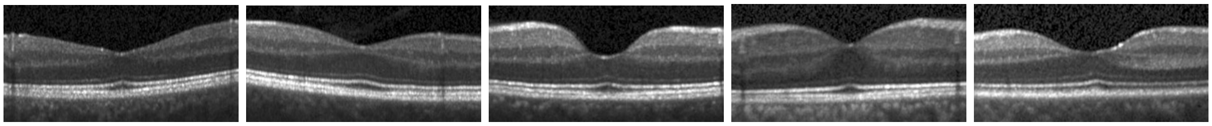
CuBe: 3D Shape Analysis of the Fovea

As part of the central nervous system, the retina comprises a similar cellular composition as the brain, and many neurologic disorders thus affect the retina. Many neuroinflammatory conditions are known to cause modifications to the fovea shape (for example, MS and NMO) and an accurate and robust shape modeling of the fovea can diagnose several neurological disorders by detecting the foveal shape changes. This chapter proposes a method for parametric modeling of the foveal shape, where we exploit invariant features of the macula from OCT data and applies a cubic Bézier polynomial along with a least square optimization to produce a best fit parametric model of the fovea. Additionally, we provide several parameters of the foveal shape based on the proposed 3D parametric modeling. Our quantitative and visual results show that the proposed model is not only able to reconstruct important features from the foveal shape, but also produces less error compared to the state-of-the-art methods. Finally, we apply the model in a comparison of healthy control eyes and eyes from patients with neuroinflammatory central nervous system disorders and optic neuritis, and show that several derived model parameters show significant differences between the two groups. Most of the contents of this chapter are appeared in publication [112]. Additionally, we added a quantitative analysis to check the fidelity of radial re-sampling from a volume scan.

7.1 Method Theory

As it is explained in Chapter 6, most of the model driven state of art methods assume that the foveal pit shape is quite similar to the Gaussian function. Therefore, these methods use the Gaussian basis function for the mathematical modeling of the foveal pit. However, this assumption is not entirely true as we can see in Figure 7.1 that the pit shape varies a lot from the Gaussian shape. Modeling of the flatness of the foveal pit is difficult with the Gaussian basis function. The inner (around the mean value) and outer (away from the mean value) regions of the Gaussian function are depend on each other and the standard deviation of the given Gaussian function. However, the shape of the interior (pit area) and exterior (rim area) regions of the fovea are independent with each other. For example, similar kinds of foveal pits can have significantly different rim heights. In summary, we can say that the Gaussian function is not an optimal basis function to describe macular and foveal shape.

To overcome these problems, we introduce a cubic Bézier based robust and flexible basis function, which is able to encounter all possible variations of the fovea shape. We fit the cubic Bézier polynomial in the interior and exterior regions independently using a least square optimization with invariant features of the fovea which will be discussed in the coming sections.



(a) Big rim diameter (b) Small pit area (c) Flat pit (d) Big rim height (e) Asymmetric pit

Figure 7.1: Various shapes of the foveal pit

7.1.1 Invariant Features of the Fovea

Foveal shape in the healthy population varies [87]. In Figure 7.1, we show a selection of different foveal pit shapes from our cohort to exemplify this. In order to reconstruct all these variations, we need to find some invariant features of the shape. The tangents at the critical points of the macula represent reliable and stable features. The critical points refer to the lowest point of the pit and the highest point around this region as shown in Figure 7.2. The slope at these critical points is always zero and the macula will have a horizontal tangent at these points.

7.1.2 Cubic Bézier

Bézier curve was introduced by Dr. Pierre Bézier in early 1960s. A parametric Bézier curve of polynomial degree n is defined as:

$$Q(t) = \sum_{i=0}^{n} P_{i,n} B_{i,n}(t), \quad 0 \leq t \leq 1. \tag{7.1}$$

Where the $P_{i,n}$ are the control points and $B_{i,n}$ are the Bernstein polynomials.

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = [0, 1.., n]. \tag{7.2}$$

For a Bézier curve of degree n, there would be $n+1$ control points. From Equation (7.1), we can see that the Bézier curve is a weighted average of the control points where weights are defined using the Bernstein polynomials. For $n = 3$, Equation (7.1) becomes the cubic Bézier equation:

$$Q(t) = \sum_{i=0}^{3} P_i B_{i,3}(t), \quad 0 \leq t \leq 1. \tag{7.3}$$

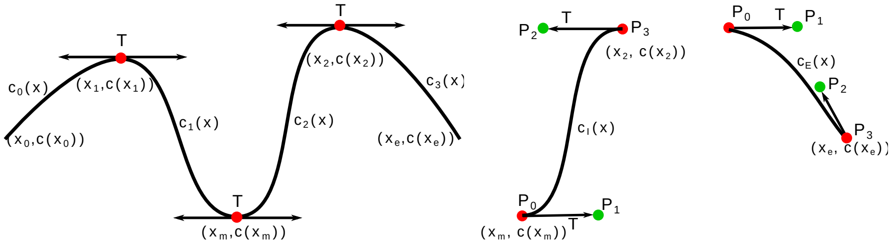
The cubic Bézier will have four control points and is tangent to the first and last control points P_0 and P_3 respectively [30]. The relationship between the control points, in terms of distance α, β and unit tangent directions T_{01} and T_{23} can be written as:

$$\begin{aligned} P_1 - P_0 &= \alpha \cdot T_{01}, \\ P_2 - P_3 &= -\beta \cdot T_{23}. \end{aligned} \tag{7.4}$$

In our curve fitting algorithm, we use the parameters α and β of each Bezier segment as shape parameters for an optimal curve fitting. The direction of a Bézier curve at its endpoints is uniquely determined by the tangent vector. Thus, by choosing the same tangent vector for two adjacent Bezier segments we assure tangent continuity (geometrical continuity G^1) at each junction, and thus throughout the whole composite spline curve.

7.1.3 Least Square Optimization

We use a least square fitting approach to each B-scan/radial scan using a cubic Bézier curve and the tangents at the critical points, derived from the invariant feature of the macular shape. Let us consider $c(x)$ as the central B-scan which contains the minimal retinal thickness point as shown in Figure 7.2(a). The central B-scan can then be decomposed into two interior $(c_1(x), c_2(x))$ and two exterior $(c_0(x), c_3(x))$ segments as shown in Figure 7.2(b).



(a) (b)

Figure 7.2: (a) The central B-scan of a volume with invariant tangent at the critical points. The critical points are shown in red dots. (b) Shows the interior and the exterior segments of the right half of the corresponding B-scan. The inner (P_1, P_2) and end (P_0, P_3) control points are shown in green and red respectively.

Now, we compute the maximum point of the right half of the central B-scan which is represented by $(x_2, c(x_2))$. Let us consider $(x_m, c(x_m))$ and $(x_e, c(x_e))$ the beginning and end points of the right half of the scan $c(x)$ as shown in Figure 7.2(b). Now, we split this half segment into two segments at the critical point $(x_2, c(x_2))$ and each segment can be written as $c_I : [x_m, x_2] \to \mathbb{R}$ for the interior and $c_E : [x_2, x_e] \to \mathbb{R}$ for the exterior segment as shown in Figure 7.2(b).

For interior segment (c_I) the end points P_0 and P_3 have the horizontal tangent lines $T = [1, 0]$ and the inner control points P_1 and P_2 are lying opposite to each other w.r.t. tangent direction, as shown in Figure 7.2(c). So, we can modify Equation (7.4) accordingly:

$$\begin{aligned} P_1 &= P_0 + \alpha \cdot T, \\ P_2 &= P_3 - \beta \cdot T. \end{aligned} \tag{7.5}$$

Similarly, for exterior segment (c_E) of the macula, one of the end points has the horizontal tangent and one of the inner control point (P_1) lies in the same horizontal tangent direction. We can modify Equation (7.4) for the exterior segments of the fovea:

$$P_1 = P_0 + \alpha \cdot T. \tag{7.6}$$

The other inner control point P_2 for the exterior segment does not have the horizontal tangent direction ($T = [1, 0]$) so we optimize the value of P_2 without using any invariant feature of the macula.

In the next step, we compute the optimized values of α and β for a best fitting cubic Bézier for each segment of the corresponding scan. We begin the optimization process from the interior segment. By using Equations (7.3) and (7.5), we can modify the cubic Bézier equation for the interior segment (c_I) as follows:

$$Q_I(t, \alpha, \beta) = \alpha T B_1(t) - \beta T B_2(t) + P_0(B_0(t) + B_1(t)) + P_3(B_2(t) + B_3(t)). \tag{7.7}$$

If the corresponding segment has m data points and is represented by D^I then we express the least square energy function as:

$$E(t,\alpha,\beta) = \sum_{i=1}^{m} \|D_i^I - Q_I(t_i,\alpha,\beta)\|^2. \tag{7.8}$$

Now, we minimize the given energy function for t, α and β in the following two steps:

1. We give an initial value to t using the uniform parametrization and compute the corresponding α and β by computing the first derivative of the given energy function w.r.t. α and β:

$$\nabla_\alpha E = 0, \quad \nabla_\beta E = 0. \tag{7.9}$$

2. In the next step, we compute the optimized value of t using the following equation with the values α and β computed at the previous step:

$$\nabla_t E = 0. \tag{7.10}$$

We iterate these two steps until we get the minima of the convex energy function in Equation (7.8).

For the exterior segment (c_E), our optimization procedure will be different compared to the interior segments because only one end point has the horizontal tangent. Using Equation (7.6), we modify Equation (7.3):

$$Q_E(t,P_2,\alpha) = \alpha T B_1(t) + P_0(B_0(t) + B_1(t)) + P_2 B_2(t) + P_3 B_3(t)). \tag{7.11}$$

Similar to the energy function in Equation (7.8), we define another energy function which depends on t, P_2 and α:

$$E(t,P_2,\alpha) = \sum_{i=1}^{m} \|D_i^E - Q_E(t_i,P_2,\alpha)\|^2. \tag{7.12}$$

Where D^E represents data points corresponding to the external segment. The minimization of this given function will be done similar to the interior segment parameter optimization. In the first step, we compute the derivative of energy function w.r.t. P_2 and α for a given initial uniform parametrized t. Then, in the next step, we compute the optimized t with the new obtained values for P_2 and α.

7.2 Materials and Methods

To evaluate the method, we used macular volume scan ($25° \times 30°$, 61 vertical or horizontal B-scans, 768 A-scans per B-scan, $9 \leq ART \leq 15$) captured with the Spectralis OCT (Heidelberg Engineering, Heidelberg, Germany). In the ART (Automatic real time) mode, the selected number of B-scans (9-15) are averaged. The voxel dimensions in the horizontal, axial directions and distance between B-scans in this data set were approximately 11.69, 3.87, and $125\mu m$ respectively. A total of 187 OCT scans, 95 from healthy controls (HC) and 92 from patients with different autoimmune neuroinflammatory diseases, were selected from the NeuroCure Clinical Research Centers' imaging database. All scans underwent quality control by an experienced rater. Automatic layer segmentation was performed with the device's software (Eye Explorer 1.9.10.0 with viewing module 6.0.9.0). The institute's imaging database only contains images derived from local studies that were approved by the local ethics committee at the Charité - Universitätsmedizin Berlin and were conducted following the Declaration of Helsinki in its currently applicable version.

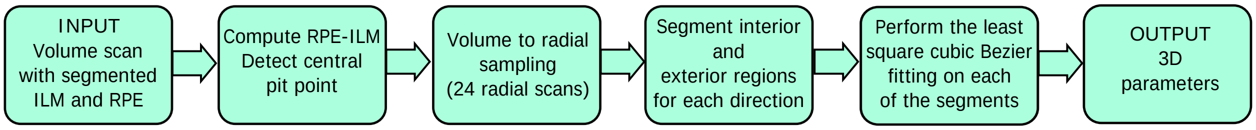


Figure 7.3: The pipeline of the proposed algorithm.

All computations were carried out using MATLAB 2016b (MathWorks, Inc., Natick, MA, USA). The statistical analysis was conducted using R version 3.3.2 [49]. We use ICC (inter class correlation) and GEE (Generalized estimating equation) for the statistical measurement. The ICC measures the repeatability of the proposed parameters and GEE (p-value) shows the significance between the two groups of data for each of the proposed parameters.

7.2.1 Method Pipeline

Figure 7.3 shows the pipeline of our algorithm. In order to import the data into MATLAB, Heidelberg Spectralis OCT raw data format was exported from the device. This data contains additional to the image information, the coordinates of the inner limiting membrane (ILM) and the Bruch's membrane (BM). The whole algorithm is implemented in following steps:

1. **ILM-RPE Computation and Minimal Foveal Point Detection**: In the first step, the height difference between the inner limiting membrane (ILM) and the Bruch's membrane (BM) of each volume scan is extracted. This represents the macular thickness surface. Using this difference has the advantage of removing the slant of the scan created at the measurement and/or by the curved shape of the eye. Let us consider a volume scan and the corresponding thickness profile represented as the graph function $\mathcal{M} : (x, y) \rightarrow \mathbb{R}$, where $(x, y) \in \Omega$ and Ω represents our region of interest. A volume scan is the combination of A-scans and B-scans obtained from the OCT scanner. We assume that x and y represent A-scans and B-scans directions respectively.

   To determine the fovea's center, a region Ω of $1\ mm$ radius is taken from this surface centered at the fovea automatically detected by the OCT device. The information about this center point is contained in the raw data export. In order to detect the lowest point of the foveal surface, we look at the minima of this region:

   $$\mathcal{M}_m = \mathcal{M}(x_m, y_m). \tag{7.13}$$

   Where x_m, y_m are the coordinate of the minimum value $\mathcal{M}_m$ of the volume scan.

   If several minimas are detected, then the median point of them is taken as the center of the foveal pit.

2. **Volume to Radial Sampling**: This is the second step of the proposed method as shown in Figure 7.3. For 3D shape analysis of the fovea, we need the information from the whole volume, and therefore we re-sample the scan into a radial one. The radial scans capture the foveal pit shape accurately as they have more samples near to the center compared to the outer region of the fovea.The sampling from volume $\mathcal{M}$ to radial $\mathcal{M}_p$ can be done in following steps:

   (a) We create a polar grid $\mathcal{M}_p(r, \theta)$, centered at the (x_m, y_m) with zero height value. The radius and the angle between the radial lines will be defined by the user.

(b) We compute the height value of $\mathcal{M}_p(r,\theta)$ using the bilinear interpolation between the nearest four points of the $\mathcal{M}(x,y)$ which are closest to the corresponding (r,θ).

(c) Now, we have $n = (2\pi/\theta)$ radial scans represented as:

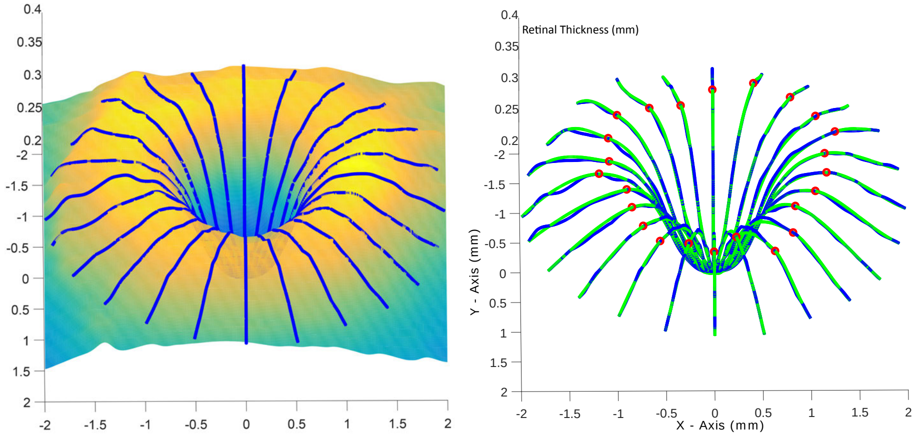
$$c(r,\theta_i) \in \mathcal{M}_p(r,\theta) \quad \text{where} \quad i = 1,\cdots,n.$$

These radial scans approximate the original volume scan as shown in Figure 7.4(a).

In our experimentation, we choose $r = 2mm$ and $\theta = 15^\circ$ for the radial sampling.

3. **Segmentation of the Radial Scans:** Each of the radial scan $c(r,\theta_i)$ is segmented into interior c_I and exterior c_E regions at the corresponding maximum (critical) point, as shown in Figure 7.2(b) and (c).

4. **Cubic Bézier Fitting using Least Square Optimization:** For each of the segments of the radial scan, we fit a cubic Bézier with the least square optimization as explained in section 7.1.3. For the interior segment, we use Equation (7.8), (7.9), and (7.10) to compute optimized α, β and t. Then, we reconstruct the interior segment using the optimized parameters by assigning these to Equation (7.7). Similarly, for exterior segment, we use Equation (7.12) and compute optimized α, P_2 and t then reconstruct the exterior segment by using Equation (7.11). To get a complete 3D parameterized modeling of the fovea, we have to fit a cubic Bézier to each of the re-sampled radial scans. Figure 7.4(b) shows the 3D parametrized model with 24 radial scans.



(a) Resampling

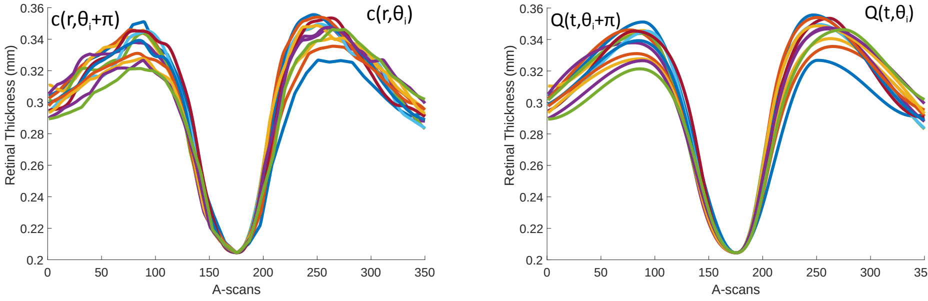
(b) Cubic Bézier parametrization for each radial scans

Figure 7.4: 3D shape reconstruction procedure. (a) Shows a volume scan with 61 B-scans and 768 A-scans and corresponding 24 radial directions (blue) using the bilinear interpolation. (b) Represents the 24 fitted radial scans (green) using the least square optimization. The red points show the critical points for each radial scan.

5. **Parameters Computation:** This is the last step of the algorithm. Now, we have a cubic Bézier parameterized 3D radial scans of a volume scan. By using this parametrization, we compute several parameters for the volume scan. The analytical formulations of these parameters are shown in the next section.

7.3 Parameters

In this section, we present several shape parameters for a volume scan using a cubic Bézier parameterization. Rim point of a radial scan is defined as the maximum height point in the corresponding radial scan. Let us consider that there are n number of radial scans re-sampled from a volume scan and $(p_1, p_2, \cdots, p_n) \in \mathbb{R}^3$ are the corresponding rim points. A visual representation of some of the 3D parameters are shown in Figure 7.7 and 7.8. Figure 7.6 shows two parametrized radial scans ($Q(t, \theta_i)$ and $Q(t, \theta_i + \pi)$) of a volume scan which differ by the angle π. Several basic parameters are defined based on these two radial scans. Q_I and Q_E represent parameterized interior and exterior segments respectively. Most of the 3D parameters are computed by first computing the values for each radial scans and then taking the average over these scans.



(a) Re-sampled radial scans (b) Cubic Bézier parametrization for each radial scans

Figure 7.5: 2D shape reconstruction (a) Shows re-sampled radial scans from a volume scan divided in two parts: $c(r, \theta_i)$ and $c(r, \theta_i + \pi)$ (b) Shows corresponding fitted radial scans using the least square optimization which are represented by $Q(t, \theta_i)$ and $Q(t, \theta_i + \pi)$ and $i = 1, \cdots, n$. A full radial scan can be consider as a combination of $c(r, \theta_i)$ and $c(r, \theta_i + \pi)$ and is shown in different colors.

Central Foveal Thickness (h_{cft}): refers to the central foveal thickness which is defined as the minimum height of the fovea at the center of the pit. From Equation (7.13), CFT can be written as:

$$h_{cft} = \mathcal{M}(x_m, y_m). \tag{7.14}$$

The h_{cft} of each radial scan will be same because (x_m, y_m) is the center of radial and represents the beginning as well as the lowest point for each radial scan.

Average Rim Height (h_r): is defined as the average of the maximum height in each radial scan of a volume (as shown in Figure 7.2). Average rim height is written as:

$$h_r = \frac{1}{n}\sum_{i=1}^{n} max(c(r, \theta_i)). \tag{7.15}$$

Rim Disk Area (A_r): To compute the rim disk area, we need to compute the normal to the disk plane as shown in Figure 7.7. The covariance analysis of all the rim points provides not only the

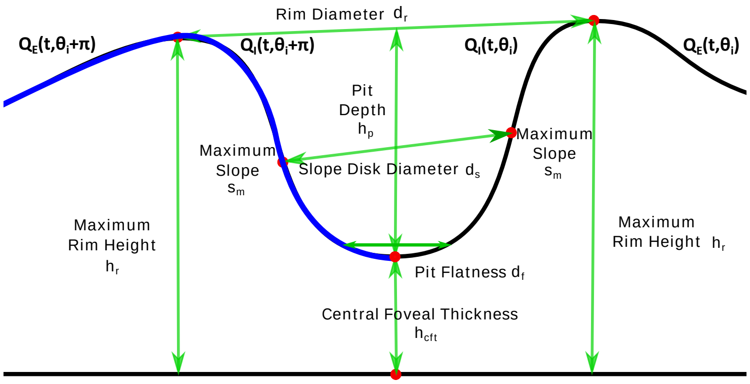
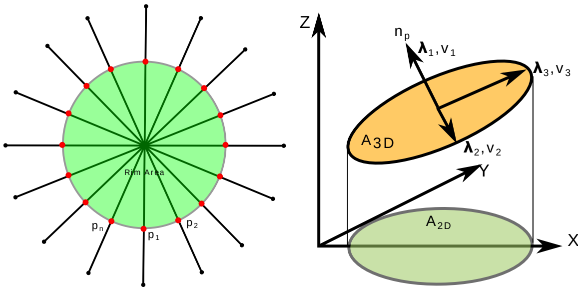


Figure 7.6: Visualization of the 2D parameters on the central B-scan.

disk normal but also shape information as described below. Rim points covariance matrix can be computed as:

$$
\begin{aligned}
C_r &= \frac{1}{n}\sum_{i=1}^{n}(p_c - p_i)^T(p_c - p_i), \\
p_c &= \frac{1}{n}\sum_{i=1}^{n} p_i.
\end{aligned}
\tag{7.16}
$$



(a) Top view (b) 3D area

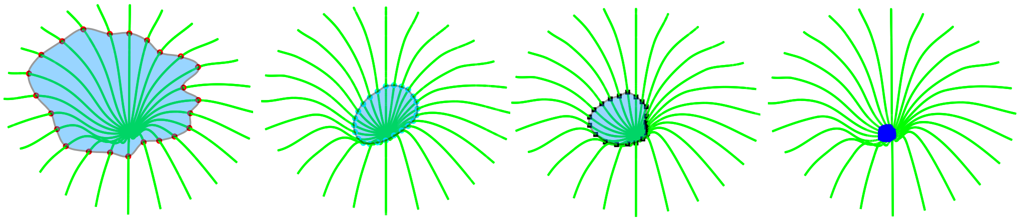
Figure 7.7: a: The top view of the rim disk. b: The normal vector of the rim disk plane and corresponding covariance eigenvalues and eigenvectors.

Eigen analysis of the matrix C_r provides shape information about the rim disk. Let us consider $\lambda = [\lambda_3, \lambda_2, \lambda_1]$ represents the eigenvalues sorted in decreasing order: $\lambda_3 \geq \lambda_2 \geq \lambda_1$ and $[v_3, v_2, v_1]$ are the corresponding eigenvectors. The most dominant eigenvalues λ_3 and λ_2 represent the **major** and the **minor** axis of the rim disk respectively. The least dominant eigen direction will be the disk normal so $n_p = v_1$. Now, the **Rim Disk Area** can be computed as:

$$
A_{3D} = \frac{A_{2D}}{cos\theta}, \quad \theta = \angle(n_p, n_z), \tag{7.17}
$$

where $n_z = [0, 0, 1]$ and $A_{2D} : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the projection of the A_{3D} on XY-plane.

Average Rim Disk Diameter (d_r): Let us consider p_{θ_i} and $p_{\theta_i+\pi}$ are two rim points corresponding



(a) Rim disk (b) Inner rim disk - all points are at 0.5 mm from the foveal center used for the inner rim volume computation (c) Slope disk (d) Pit disk

Figure 7.8: A visual representation of the 3D parameters. The rim and the inner rim volume is defined as the volume covered by the corresponding disk area. The radius for the inner disk area and the volume is defined by user.

to the two parametrized opposite radial scans($Q(t, \theta_i)$ and $Q(t, \theta_i + \pi)$) then average rim diameter is written as:

$$d_r = \frac{2}{n} \sum_{i=1}^{n/2} \|p_{\theta_i} - p_{\theta_i + \pi}\|^2. \tag{7.18}$$

Average Pit Depth (h_p): represents the depth of the foveal pit and can be computed as the difference between the average rim height and central foveal thickness.

$$h_p = h_r - h_{cft}. \tag{7.19}$$

Average Maximum Pit Slope (s_m): Average maximum pit slope measures the steepness of the foveal pit. It is defined as the average of the maximal slope of each radial scan. The slope of a parametrized radial scan $Q(r, \theta_i)$ is defined as:

$$s_i = \frac{dQ_I^y(t)/dt}{dQ_I^x(t)/dt} \quad i = 1, \cdots, n.$$

Where $Q_I^y(t)$ and $Q_I^x(t)$ represent parametrized x and y coordinates for the interior segment. To calculate the maxima of the above equation, we compute the gradient, $\nabla_t s_i = 0$ and we get the value of t_{m_i} for the maximal slope (s_{m_i}):

$$\begin{aligned} s_{m_i} &= -\frac{P_3^y - P_0^y}{\alpha - \beta - \sqrt{\alpha\beta} - (P_3^x - P_0^x)}, \\ t_{m_i} &= \frac{\sqrt{\alpha}}{\sqrt{\alpha} + \sqrt{\beta}}, \end{aligned} \tag{7.20}$$

where P_3^y and P_0^y are the y coordinates of the end control points of the $Q_I(t, \theta_i)$. Similarly, P_3^x and P_0^x are the x coordinates of the end control points of the $Q_I(t, \theta_i)$. Average maximum pit slope of a volume is defined as:

$$s_m = \frac{1}{n} \sum_{i}^{n} s_{m_i}. \tag{7.21}$$

Average Slope Disk Diameter (d_s): Average slope disk diameter is computed similar to the average rim diameter d_r. Slope width is computed between two opposite parametrized radial scans: $Q(t, \theta_i)$ and $Q(t, \theta_i + \pi)$ and the corresponding maximum slope points are $p^s_{\theta_i}$ and $p^s_{\theta_i+\pi}$ such that:

$$\begin{aligned} p^s_{\theta_i}(x) &= Q^x(t_m, \theta_i), \\ p^s_{\theta_i}(y) &= Q^y(t_m, \theta_i), \\ p^s_{\theta_i+\pi}(x) &= Q^x(t_m, \theta_i + \pi), \\ p^s_{\theta_i+\pi}(y) &= Q^y(t_m, \theta_i + \pi). \end{aligned} \tag{7.22}$$

Where t_m represents maximum slope point in parametric domain as shown in Equation (7.20) and $Q^x(t_m, \theta_i)$, $Q^y(t_m, \theta_i)$ are the corresponding x and y coordinate. Average slope width of a volume is defined as:

$$d_s = \frac{2}{n} \sum_{i=1}^{n/2} \| p^s_{\theta_i} - p^s_{\theta_i+\pi} \|^2. \tag{7.23}$$

Slope Disk Area (A_s): The slope disk area can be computed similar to the rim disk area. Let us consider $(p^s_1, p^s_2, \cdots, p^s_n) \in \mathbb{R}^3$ are the maximum slope points corresponding to the each radial scan and are computed using Equations (7.20) and (7.22). Covariance of maximal slope points can be computed using Equation (7.16) to obtain the normal vector, major and minor axes of the slope disk.

Pit Flat Disk Area(A_f): The pit flat disk area measures the flatness of the foveal pit around the center and is computed using a threshold value τ for each of the radial scans. In each radial scan, we compute a point $p^f_{\theta_i}$ where the retinal thickness is smaller than τ. Then the corresponding segment from the center (x_m, y_m) to the computed point $p^f_{\theta_i}$ is treated as flat. This can be done in following two steps:

1. First of all, we compute the parametric value t_f corresponding to the threshold value τ:

$$\tau - Q^y_I(t_f, \theta_i) = 0. \tag{7.24}$$

2. Now, we compute the x value corresponding the parametrized value t_f:

$$d_{f_i} = Q^x_E(t_f, \theta_i). \tag{7.25}$$

The point $p^f_{\theta_i} = (d_{s_i}, \tau, \theta_i)$ corresponds to the pit flat point for a single radial scan. Similarly, let us consider $(p^f_1, p^f_2, \cdots, p^f_n) \in \mathbb{R}^3$ as the pit flat points. The area, major and minor axis of the pit flat disk can be computed similar to the rim disk area. We compute the **average pit flat disk diameter (d_f)** using d_{f_i} from Equation (7.25):

$$d_f = \frac{2}{n} \sum_{i=1}^{n} d_{f_i}. \tag{7.26}$$

Rim Volume (V_r): In general, the volume under a surface in Cartesian and Polar domains and can be computed using the following equation:

$$V_r = \int\int_A f(x,y)dxdy = \int_0^{2\pi}\int_0^R c(r,\theta)rdrd\theta.$$

After the discretization process of the whole volume into n radial directions, the above equation will become :

$$V_r = \frac{2\pi}{n}\sum_{i=1}^{n}\int_0^R c(r,\theta_i)rdr.$$

We modify the above equation using the following substitution. The interior parametrized curve $Q_I(t,\theta_i)$ is defined between the center and the rim points.

Therefore, the parametric value $t \rightarrow \{Q_I^x(t,\theta_i), Q_I^y(t,\theta_i)\}$ will be $t = 0$ and $t = 1$ at the center and the rim points respectively. Similarly, $r = Q_I^x(t,\theta_i)$, $R = Q_I^x(1,\theta_i)$, $c(r,\theta_i) = Q_I^y(t,\theta_i)$ and $dr = \frac{dQ_I^x(t,\theta_i)}{dt}dt$. After the substitution, the rim volume can be written as:

$$V_r = \frac{2\pi}{n}\sum_{i=1}^{n}\int_0^1 Q_I^y(t,\theta_i)Q_I^x(t,\theta_i)\frac{dQ_I^x(t,\theta_i)}{dt}dt. \tag{7.27}$$

In the above equation, $R = P_3^x$ and at the end point of the parametric curve t is equal to one. R might have different values for different radial scans because of the asymmetry of the foveal pit.

Inner Rim Volume (V_{IR}): Is defined as the volume of the fovea within the fixed radius from the center. The radius is given by the user. Let us consider t_u to be the parametric value corresponding to the user input radius r_u for each radial direction. Now, we can modify Equation (7.27):

$$V_{IR} = \frac{2\pi}{n}\sum_{i=1}^{n}\int_0^{t_u} Q_I^y(t,\theta_i)Q_I^x(t,\theta_i)\frac{dQ_I^x(t,\theta_i)}{dt}dt. \tag{7.28}$$

Before using the above equation, we have to make sure that the radius R will be equal to the minimum of the P_3^x from all the radial directions. In case, R is bigger than any of the P_3^x, we need to introduce the exterior segment into the integration as well.

Pit Volume (V_p): We calculate the total volume (V_t) under the rim disk, $V_t = A_{3D}h$, where h is the average height of the rim points. Then the pit volume can be defined as:

$$V_p = V_t - V_r. \tag{7.29}$$

Parameters	ICC	LCI	UCI
Pit depth (h_p) (mm)	0.9813	0.9663	0.9904
Central foveal thickness (h_{cft}) (mm)	0.9894	0.9808	0.9945
Rim height (h_r) (mm)	0.9762	0.9572	0.9877
Rim diameter (d_r) (mm)	0.9250	0.8686	0.9608
Rim disk area (A_r) (mm^2)	0.9192	0.8588	0.9576
Major axis rim disk (λ_3^r) (mm)	0.9099	0.8434	0.9526
Minor axis rim disk (λ_2^r) (mm)	0.9069	0.8384	0.9509
Slope disk diameter (d_s) (mm)	0.9309	0.8785	0.9639
Slope disk area (A_s) (mm^2)	0.9467	0.9056	0.9723
Major axis slope disk (λ_3^s) (mm)	0.9365	0.8881	0.9669
Minor axis slope disk (λ_2^s) (mm)	0.9467	0.9056	0.9723
Pit disk diameter (d_f) (mm)	0.8966	0.8215	0.9454
Pit disk area (A_f) (mm^2)	0.9155	0.8528	0.9556
Major pit disk length (λ_3^f) (mm)	0.8990	0.8255	0.9467
Minor pit disk length (λ_2^f) (mm)	0.9135	0.8494	0.9545
Rim Volume (V_r) (mm^3)	0.8957	0.8201	0.9449
Inner rim volume (V_{IR}) (mm^3)	0.9668	0.9400	0.9831
Pit Volume (V_p) (mm^3)	0.9606	0.9297	0.9796
Pit slope (s_m) ($Degrees$)	0.9697	0.9457	0.9844
α_m (mm)	0.9172	0.8555	0.9565
β_m (mm)	0.8102	0.6864	0.8967

Table 7.1: Repeatability test for the 3D parameters. Abbreviations: ICC - intra-class correlation coefficient, LCI - lower confidence interval and UCI - upper confidence interval.

7.4 Experiments, Results and Discussion

Scans from 187 eyes of 95 healthy control (HC) and 92 patients with autoimmune neuroinflammatory diseases (with and without previous optic neuritis) were processed.

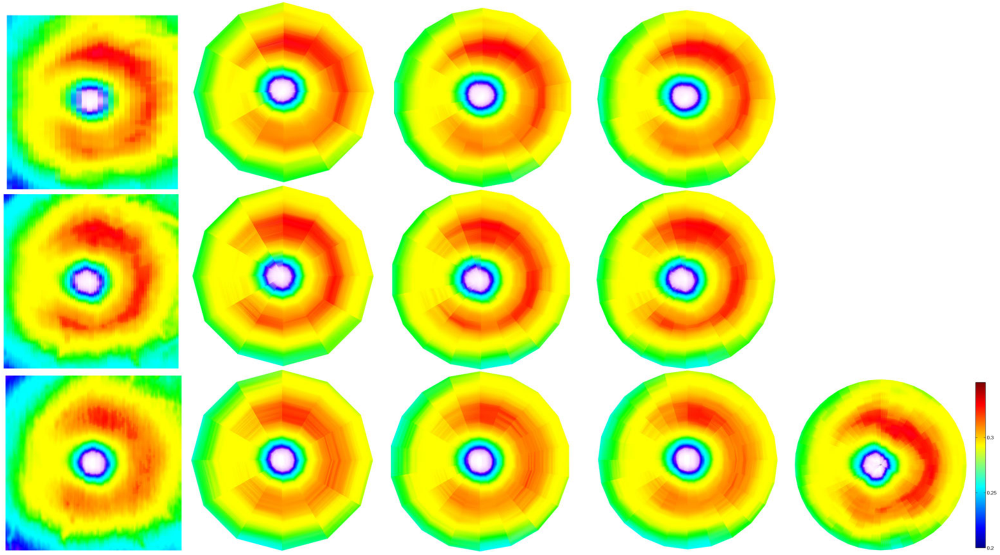
7.4.1 Quantitative Analysis of Volume to Radial Re-sampling

An important aspect in the re-sampling process of the volume are the number of radial scans. Lower number of radial scans will lead to data loss and higher number will increase the running time of the algorithm. Figure 7.10 shows the height map of each volume scan (61, 97,193 B-scans) and the corresponding re-sampled scans with different radial number (12, 18, 24). A star scan ($n = 48$) measured by the OCT device is used as the ground truth.

Table 7.2: The distance between a volume scan and re-sampled radial scan using the L_2-metric [114]

Volume to radial distance comparison (in μm)			
Volume scan	$n = 12$	$n = 18$	$n = 24$
61 B-scans	0.188	0.167	0.183
97 B-Scans	0.151	0.166	0.164
193 B-scans	0.157	0.161	0.166

We also tested the fidelity of the re-sampling, by computing the L2-metric [114] for each of the volumes mentioned above, see Table 7.2. This metric basically calculates the closeness between two geometries. The very low values obtained show that there is no significant difference between these volumes and radial scans derived, therefore the scan with 61 B-scans delivers enough information for the our analysis. Moreover, the radial samples capture the foveal pit shape accurately as there are more samples into the pit region compared to the outer region of the fovea.



(a) Volume Scan (b) $n = 12$ (c) $n = 18$ (d) $n = 24$ (e) Ground Truth, $n = 48$

Figure 7.9: From left to right: The first column consists of volume scans with 61, 97 and 193 B scans respectively. The second column shows 12 re-sampled radial lines from the corresponding volume scan. Similarly, third and fourth column show 18 and 24 re-sampled radial lines from the corresponding volume scan. The last column shows a volume measured with a radial scan pattern consisting of 48 radial B-scans.

7.4.2 Re-test Reliability

The proposed algorithm was able to successfully model all the scans; visual inspection of model results did not suggest modelling failure in any of the included scans. Table 7.1 shows the test-retest reliability applied to a group of data including three repeated measurements of 30 healthy eyes. Intra-class correlation coefficient (ICC) for the 3D derived parameters varied from 0.8102 for β_m to 0.9894 for the Central foveal thickness as shown in Table 7.1. From Table 7.1, the parameter β_m has the lowest repeatability. Basically, α_m and β_m are the average of all α and β belonging to radial scans of the interior region of the fovea. As mentioned in Equation (7.5), α and β represent the distance between the control points P_0, P_1 and P_3, P_2 respectively. The lowest ICC value of the β_m indicates the distance between the P_3 and P_2 is not consistent. Perhaps, the small inaccuracy in rim point detection is leading to the low repeatability of the β_m. On the other hand, α depends on the center or beginning control point(P_0) which is fix for a volume scan. Therefore, α_m has a better repeatability compared to the β_m.

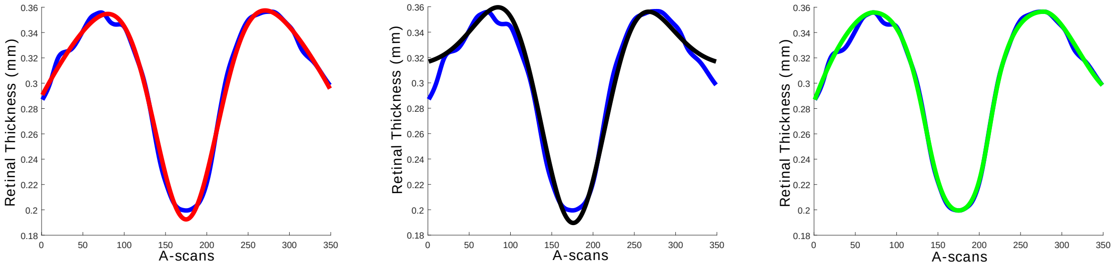
Parameters	HC		Patients		GEE
	Mean (SD)	Min-Max	Mean (SD)	Min-Max	p
Pit depth (h_p) (mm)	0.128 (0.019)	0.079-0.165	0.1109 (0.022)	0.04-0.15	**3.9E-08**
Central Foveal Thickness (h_{cft}) (mm)	0.224 (0.016)	0.186-0.266	0.223 (0.014)	0.199-0.2640	0.611
Rim height (h_r) (mm)	0.352 (0.014)	0.319-0.389	0.335 (0.018)	0.285-0.378	**5.4E-13**
Rim Diameter (d_r) (mm)	2.13 (0.19)	1.68-2.58	2.14 (0.17)	1.76-2.51	0.497
Rim disk area (A_r) (mm^2)	3.58 (0.66)	2.22-5.24	3.51 (0.59)	2.44-4.90	0.450
Major axis rim disk (λ_3^r) (mm)	0.625 (0.11)	0.372-0.867	0.615 (0.1)	0.42-0.86	0.539
Minor axis rim disk (λ_2^r) (mm)	0.535 (0.1)	0.32-0.817	0.524 (0.089)	0.33-0.78	0.451
Slope disk diameter (d_r) (mm)	0.608 (0.015)	0.34-0.955	0.636 (0.16)	0.274-0.101	0.178
Slope disk area (A_s) (mm^2)	0.313 (0.12)	0.1-0.71	0.348 (0.17)	0.059-0.839	0.111
Major axis slope disk (λ_3^s) (mm)	0.053 (0.02)	0.019-0.123	0.06 (0.029)	0.01-0.839	0.08
Minor axis slope disk (λ_2^s) (mm)	0.045(0.018)	0.0001-0.1	0.049(0.026)	0.00002-0.127	0.18
Pit disk diameter (d_f) (mm)	0.182(0.031)	0.11-0.295	0.197(0.044)	0.1-0.433	**0.016**
Pit disk area (A_f) (mm^2)	0.027(0.01)	0.011-0.068	0.033(0.018)	0.0079-0.149	**0.012**
Major axis pit disk (λ_3^f) (mm^2)	0.0048(0.0017)	0.0023-0.011	0.0058(0.003)	0.0013-0.025	**0.009**
Minor axis pit disk (λ_2^f) (mm^2)	0.004(0.0015)	0.0013-0.01	0.0048(0.0028)	0.0012-0.022	**0.004**
Rim volume (V_r) (mm^3)	1.02(0.19)	0.558-1.544	0.944(0.17)	0.597-1.364	**0.002**
Pit volume (V_p) (mm^3)	0.239(0.044)	0.152-0.366	0.236(0.054)	0.134-0.384	0.73
Inner rim volume (V_{IR}) (mm^3)	0.11(0.018)	0.06-0.153	0.1(0.019)	0.062-0.147	**8.9E-4**
Pit slope (s_m) ($Degree$)	12.74(2.68)	6.32-20.65	10.65(2.62)	3.4-16.29	**6.32E-8**
α_m (mm)	0.25(0.063)	0.133-0.470	0.25(0.077)	0.1-0.45	0.46
β_m (mm)	0.61(0.1)	0.36-0.83	0.56(0.089)	0.335-0.845	**1.7E-4**

Table 7.3: Analysis of all the 3D parameters defined for the HC and patient group. The last column shows the GEE analysis between the two groups. Abbreviations: HC - healthy controls. SD - standard deviation, Min - minimum value, Max - maximum value, GEE - generalized estimating equation models analysis accounting for the inter-eye/intra-subject dependencies, p - p value

7.4.3 Model Accuracy

For root-mean-square-error (RMSE) comparison, we have implemented two state-of-the-art methods for foveal shape analysis, the one proposed by Ding et al. [22], and the one described in Dubis et al. [23] and compared them with our method. For a better readability, we renamed [22] as M_1 and [23] as M_2. In case of M_1 we implemented the 3D approach, as presented in the paper [22]. For M_2 we implemented the 3D version by extending the 2D method presented by the authors on our radial re-sampled volume. In order to visualize the behaviour of M_1 vs. M_2 vs. CuBe, we present the fitting results on the central B-scan for the same subject in Figure 7.10. As shown in the figure, CuBe is able to capture the exact foveal shape with the lowest RMSE compared to methods M_1 and M_2.

The method M_2 [23] derives only three parameters (slope, rim diameter and rim height) to characterize the shape of the fovea, which is a too small number to encounter all the variation of the fovea shape. The optimization procedure of the mentioned energy function in this method is complex because of the two different Gaussian basis and the energy function can have unstable minima quite often. The method M_1 [22] needs eight parameters $(A_0, A_{11}, \cdots, A_{22})$ to reconstruct a 3D model of a volume scan. However, these parameters do not represent any morphological information of the fovea, which makes their interpretation in relation to the morphological characteristics rather difficult. Scheibe et al. introduced a stable and accurate modeling of the fovea using the double derivative of an exponential function [86]. This method [86] is applied to re-sampled radial scans from 3D macula cube scans. There are four parameters used to parametrize a radial scan. However, the term γ (see [86]) makes the algorithm more complex and leads to difficulties in optimization and analytical analyses. Additionally, the direct analytical derivation of 3D parameters, e.g volumes presents several computational problems. Recently, Liu et.al [64] has introduced a sloped piecemeal Gaussian model for characterizing the foveal pit shape. This method uses a com-



(a) M_1, RMSE = $4\mu m$ (b) M_2, RMSE = $8\mu m$ (c) CuBe, RMSE = $2.5\mu m$

Figure 7.10: Comparison of the proposed method with two state-of-the-art methods [22] and [23]. Blue curve is the raw input (ILM-RPE difference) at the central B-scan and black, red and green are the reconstructed curves with [23], [22] and our method (CuBe) respectively. The RMSE values show that the proposed method manages to reconstruct the shape with minimum artifacts, specifically the pit shape.

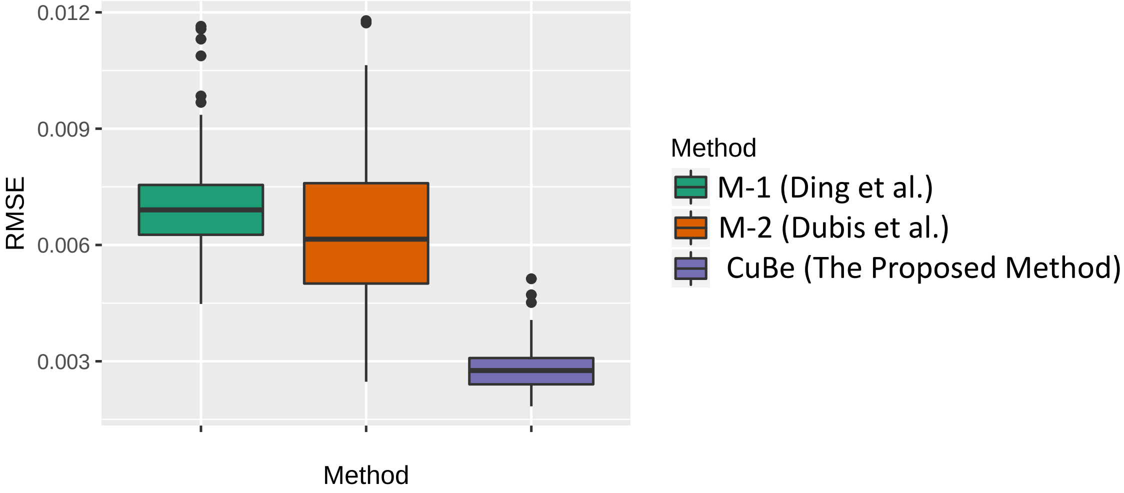


Figure 7.11: RMSE values for 3D M_1 , M_2, and $CuBe$ for the whole data set.

bination of linear and Gaussian basis function along with an additional parameter which encounters the pit bottom flatness. As this method is using piecewise basis function, the optimization process is not straight forward. However, this method only characterizes the foveal pit and not the complete fovea shape.

In Figure 7.11, we show the different RMSE values for 3D fitting on a volume scan. The comparison has been analyzed between the proposed method, CuBe and methods M_1, M_2. The RMSE values for our approach show overall lower values compared to the other methods as our method reconstructs the foveal pit more accurately as shown in Figure 7.10. In the proposed method, the highest RMSE occurs in regions where the segmentation is mostly influenced by blood vessels. These blood vessels produce several ”jumps” spatially close to each other as shown in Figure 7.12 (bottom). These ”jumps” have a strong influence also in the detection of the maximum height points (rim points) which can lead to a higher RMSE because our parametrization scheme depends on these critical points. In such cases, peaks are representing the vessels and not the retinal tissue, which might induce an additional noise when investigating differences between healthy and

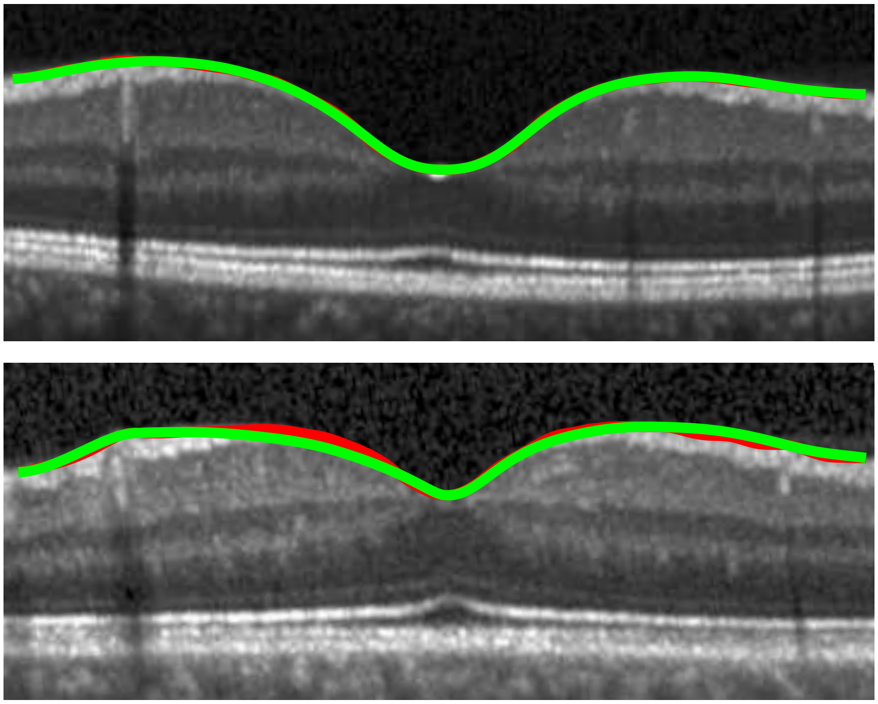


Figure 7.12: Two examples of the 2D curve fitting results with the lowest (top) and the highest (bottom) RMSE values selected from the entire data set analyzed. The original data is shown in red and the fitted curve in green. The RMSE values are from top to bottom, $1.0\ \mu m$, $7.1\ \mu m$.

pathological data [76].

An important aspect of the presented method is the ability to utilize characteristic fovea properties for the whole circular region (re-sampled radial region) because of the flexible and robust cubic Bezier parametrization scheme. This implies that we derive the discussed 3D parameters in a completely new fashion. This gives us the possibility to explore new morphological features of the fovea in terms of the Rim Disk, Slope Disk, and Pit Flat Disk by looking at the eigenvectors of the covariance matrix defined in Equation (7.16). By computing the pit flatness and its area from the 3D reconstruction, we are able to provide not only a better visual insight of this region but also a potentially new diagnostic parameter for further investigations into several diseases e.g. those characterized by nerve fibers and ganglion cells loss [74, 75, 88]. Another beneficial aspect of the presented 3D approach is in the flexibility of defining the foveal radius. This can be computed from the model itself as a specific feature of each radial Bézier curve part, and as such it would take into account the asymmetric nature of this region, or as a variable radius, defined by the user.

7.4.4 Application in HC and Patients with Autoimmune Neuroinflammatory Disorders

Having all the parameters described in the previous section, we can perform a first analysis of the morphological aspects of the data set. Table 7.3 shows all the 3D parameters defined for the HC and patient groups. The measurements obtained with our method have similar values to the ones encountered in the literature. For characteristics like Rim Disk Area, Slope Disk Area, Pit Flat Disk Area a comparison to the existing literature was not possible since this is the first time that these parameters has been introduced.

We also investigated the capability of the derived parameters to differentiate between HC and patients. To this end we performed generalized estimation equation (GEE) analyses. Table 7.3 represents that several parameters show significant differences between HC and patients. These results open new possibilities for further investigation of the fovea shape derived parameters in more specific clinical diseases.

7.5 Summary

This chapter introduced a reliable, accurate and meaningful approach for fovea shape analysis, which is able to correctly model the profile of the foveal region and reconstruct its 3D shape. The proposed method has been shown to robustly encounter possible variations in foveal shape in HC but also in foveas that undergo considerably changes during the course of neuroinflammatory disease.

The mathematical model created is simple and has the advantage of a straight forward derivation of parameters. The computed parameters are in direct relation to the geometry modeled, and therefore provide an intuitive way of interpretation for further medical analysis and clinical interpretation. A major advancement of the developed method is that it is possible to analyze the foveal shape in a clinical context, especially from the 3D perspective.

Several derived foveal shape parameters showed statistically, significant differences between HC and patients with neuroinflammatory diseases of the central nervous system and could potentially reveal more insights into the foveal morphology and the changes it undergoes especially when correlated with other clinical information. Further applications in ophthalmologic diseases like macular degeneration are worth to be investigated as well and it would be interesting to see the foveal shape variation in HC, in terms of proposed 3D parameters.

Chapter 8

Optic Nerve Head 3D Shape Analysis using ILM and RPE Manifold Surfaces

Due to several ophthalmological problems and neurological disorders (for example, IIH and ON), the ONH shape varies and the quantification of these shape variations lead to an accurate diagnosis of different diseases. To provide an ease to clinicians, the automatic shape analysis of the ONH is always preferable. The presented method automatically generates a 3D ONH model and then allows computation of several 3D parameters describing the ONH. The method starts with a high-resolution OCT volume scan as input. From this scan, the model-defining inner limiting membrane as inner surface and the retinal pigment epithelium as outer surface are segmented, and the Bruch membrane' opening as the model origin is detected. Based on the generated ONH model by triangulated 3D surface reconstruction different parameters (areas, volumes, annular surface ring, mininum distances) of different ONH regions can then be computed. Additionally, the bending energy (roughness) in the BMO region on the ILM surface and 3D BMO-MRW surface area are computed. We show that our method is reliable and robust across a large variety of ONH topologies (specific to this structure) and present a first clinical application. This chapter is based on a published method [113]. Additionally, statistical shape analysis of a group of ILM surfaces is included. The statistical shape analysis of a group of ILM surfaces is performed in two different steps: in the first step, we align the ILM surfaces w.r.t each other and then in the next step, we compute a mean shape of a group of ILM surfaces. Later, the shape variation is shown using the standard deviation of the corresponding points.

The objective of our proposed method is to retrieve robust and reliable 3D quantitative measurements that describe different aspects of the various shapes of the ONH. Specifically,:

- A fully automatic pipeline is introduced for 3D shape analysis of the ONH region.
- The segmentation of ILM and the lower boundary of RPE along with BMO points are done automatically.
- Triangulated 3D surfaces of the ILM and the lower boundary of RPE are computed.
- Several 3D shape analysis parameters are introduced along with traditional parameters.
- Statistical shape analysis of a group of ILM surfaces is added.
- We validate and prove the robustness of our algorithm in a dataset with clinical relevance.

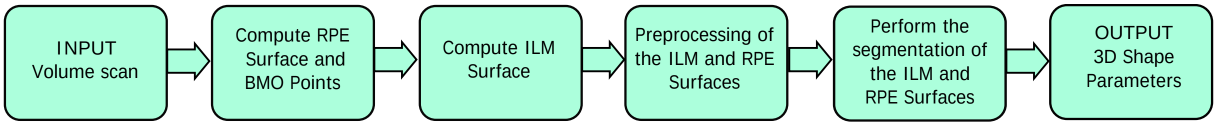


Figure 8.1: Pipeline

8.1 Method

In this section, we explain the procedure to compute several 3D shape parameters of the ONH including the preprocessing of the ILM and the RPE surfaces and correspondence between them. Figure 8.1 shows the algorithm pipeline, where an OCT volume scan is the input of the algorithm. Next, we compute the triangulated ILM and RPE surfaces along with the BMO points. These three structures represent the ONH 3D shape and serve as inputs for further shape analysis of the ONH. In the remainder of the paper, the ILM and the RPE surfaces will be represented as $\mathcal{M}_{ilm}$ and $\mathcal{M}_{rpe}$, respectively and the BMO points are denoted by $\mathcal{P}$.

The ILM and the RPE surfaces are triangulated manifold surfaces and can be written in terms of the set of vertices and faces (triangles):

$$\mathcal{M}_{ilm} = \{\mathbf{V}_{ilm}, \mathbf{F}_{ilm}\}, \quad \text{and} \quad \mathcal{M}_{rpe} = \{\mathbf{V}_{rpe}, \mathbf{F}_{rpe}\}.$$

These two surfaces can have different numbers of vertices and number of triangles. Let us consider that n_{ilm} and m_{ilm} represent the numbers of vertices and faces in the ILM surfaces. Similarly, n_{rpe} and m_{rpe} denote the size of $\mathbf{V}_{rpe}$ and $\mathbf{F}_{rpe}$ respectively. The BMO points are represented as: $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 | i = 1 \cdots n_p\}$, where n_p is the number of the BMO points. The computation of the RPE surface and the BMO points is explained in a published method [113].

8.1.1 ILM Surface Computation

The ILM separates the retina from the vitreous body and defines a critical boundary layer for the ONH. Several ILM segmentation methods have been published to separate the ILM layer around the ONH region, and most of them compute the ILM layer as a graph function and are unable to capture complex and variable topological structures of the ONH. To compute the ILM surface, we previously developed a method introduced by Gawlik et al. [36]. This method is based on an active contour method of Chan-Vese type and produces a true 3D ILM segmentation unlike other state-of-the-art methods. Figure 8.2(a) shows the ILM surface, which is computed using the Marching Cubes algorithm [65], where the input level sets are computed using the method proposed by Gawlik et al. [36]. The lower right corner of Figure 8.2(a) shows that the approach is capable to reconstruct a complex topological structure in the ONH region and the Marching Cubes algorithm produces a manifold ILM surface $\mathcal{M}_{ilm}$ with proper neighborhood and properly oriented face normals.

8.1.2 ILM Surface Smoothing

During data acquisition of the ONH region using the 3D OCT scanner, noise is inevitable due to various internal, and external factors. As it can be seen from Figure 8.2(a), the ILM surface is not smooth and has various noise components. Staircase artifacts are also shown in the left corner of Figure 8.2 as an effect of the Marching Cubes algorithm and a volume scan resolution. For an accurate shape parameter computation, these artifacts, including noise components should be removed at the early stage of the algorithm. To compute a noise free and a high-fidelity ILM surface, we use a robust mesh denoising algorithm, proposed by Yadav et al. [115]. In general,

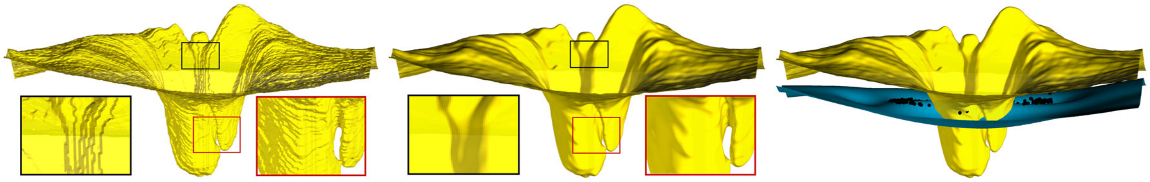

(a) Noisy ILM surface (b) Smooth ILM surface (c) Smooth ILM and RPE surfaces with BMO

Figure 8.2: The ILM surface computed using the marching cube algorithm and level sets produced by Method [36].

mesh denoising algorithms are divided into two categories: isotropic and anisotropic methods. Isotropic methods remove noise effectively but produce a volume shrinkage, which leads to an incorrect shape analysis of the ONH. Anisotropic methods are feature preserving mesh denoising algorithms and induce a small volume shrinkage compared to the isotropic methods. In case of the ILM surface, the anisotropic methods treat the Marching Cube artifacts as features and lead to an incorrect shape analysis. Method [115] is a combination of both isotropic and anisotropic methods and produces a high fidelity smooth ILM surface without staircase artifacts and minimum volume shrinkage as shown in Figure 8.2(b). This method also produces a high quality triangular mesh with proper face normal orientation, which is vital in further shape analysis of the ONH.

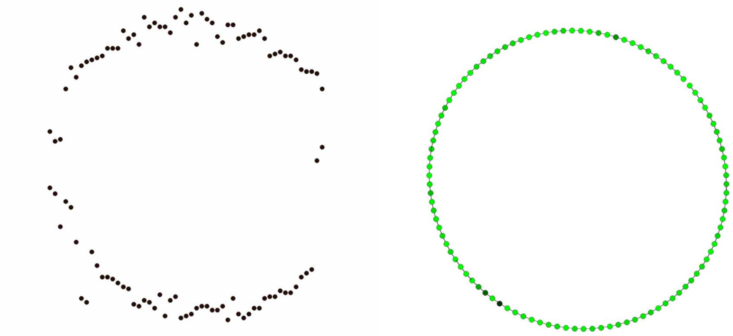

(a) Noisy BMO Points (b) Smooth BMO Points

Figure 8.3: (a) Noisy and irregular BMO points, which are computed automatically. (b) Smooth and uniform density BMO points are computed using an ellipse fit.

8.1.3 Ellipse Fitting to the BMO Points

The proposed algorithm uses the BMO points, which are computed automatically using Method [113]. Due to blood vessels around the ONH, noise components and 3D OCT scan patterns, the BMO points are non-uniform and noisy as shown in Figure 8.3(a). To remove these artifacts, we fit an ellipse to the BMO point onto XY-plane. First of all, the BMO points $\mathcal{P}$ are projected onto the corresponding XY-plane and denoted as a points set: $\mathcal{P}_{2D} = \{\tilde{\mathbf{p}}_i \in \mathbb{R}^2 | i = 1, \cdots, n_p\}$. Then, we apply Method [33] to compute a fitted ellipse to the BMO pints in $\mathbb{R}^2$. Another key parameter in the ONH shape analysis is the center of the BMO points. This is computed as the barycentric of the all $\mathcal{P}_{2D}$ points:

$$\tilde{\mathbf{p}}_c = \frac{1}{n_p} \sum_{\tilde{\mathbf{p}}_i \in \mathcal{P}_{2D}} \tilde{\mathbf{p}}_i, \tag{8.1}$$

where $\tilde{\mathbf{p}}_c \in \mathbb{R}^2$. Figure 8.3 shows that the ellipse fitting is not only removing noise but also increases data points uniformly.

8.1.4 Correspondence between ILM and RPE Surfaces

The total retina at the ONH region is delimited by ILM and RPE. For further analysis, it is necessary to find the corresponding points between these two surfaces. In the proposed method, we compute vertices in the RPE surface corresponding to each face ($f_i \in \mathbf{F}_{ilm} | i = 1, \cdots, m_{ilm}$) of the ILM surface. In general, the RPE surface, represented here as a function graph : $\mathcal{M}_{rpe} : \mathbb{R}^2 \to \mathbb{R}$, has a less complex structure compared to ILM. In the OCT scanner, the number of the A-scans (x-direction) and the number of the B-scans (y-direction) are fixed, which creates a regular XY-grid as a domain for the RPE graph function. Therefore, the index of each vertex of the RPE surface can be computed using the numbers of x-lines (vertical lines) and y-lines (horrizontal lines) and the sampling size in both directions, denoted by ϵ_x and ϵ_y respectively. The numbers of x-lines and y-lines are computed using the following equation:

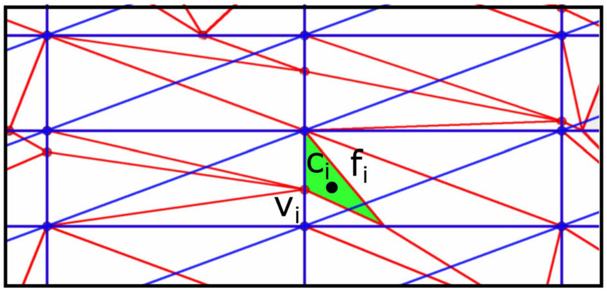


Figure 8.4: Correspondence between ILM and RPE surfaces, the face $f_i \in \mathbf{F}_{ilm}$ (green triangle) has a corresponding vertex $\mathbf{v}_i \in \mathbf{V}_{rpe}$ in the RPE surface.

$$\begin{aligned} \text{xline} &= \frac{x_{max} - x_{min}}{\epsilon_x} + 1, \\ \text{yline} &= \frac{y_{max} - y_{min}}{\epsilon_y} + 1, \end{aligned} \tag{8.2}$$

where x_{max}, x_{min}, y_{max} and y_{min} are the bounding values of the RPE surface in x and y directions. For each face $f_i \in \mathbf{F}_{ilm}$, the vertex of the RPE surface onto the XY-plane is computed, which approximates the position of the corresponding A-scan and B-scan in the volume scan. Let us consider that $\mathbf{c}_i$ represents the centroid of the face f_i. To compute the corresponding vertex in the RPE surface, we project the face f_i onto the corresponding XY-plane, $\tilde{\mathbf{c}}_i$ represents the projected centroid. The terms $\tilde{c}_{x_i}$ and $\tilde{c}_{y_i}$ are the corresponding x and y coordinates. We compute the x-index (i_x), the y-index (i_y), and the vertex (i) index using xline and yline in the RPE surface for the face f_i using the following equation:

$$\begin{aligned} i_x &= \left\lceil \frac{\tilde{c}_{x_i} - x_{min} + \epsilon_x/2}{\epsilon_x} \right\rceil, \\ i_y &= \left\lceil \frac{\tilde{c}_{y_i} - y_{min} + \epsilon_y/2}{\epsilon_y} \right\rceil, \\ i &= i_x + i_y \cdot \text{xline}, \end{aligned} \tag{8.3}$$

where $\lceil \quad \rceil$ represents the ceil function and i denotes the corresponding vertex in the RPE surface. For an accurate computation, we check the neighborhood of vertex i of the RPE surface and the corresponding vertex is computed as:

$$\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_j \in \Omega_i \mid \quad \min|\tilde{\mathbf{c}}_i - \tilde{\mathbf{v}}_j|, \tag{8.4}$$

where Ω_i represents the 3×3 neighborhood (at XY-plane) of vertex i. The term $\tilde{\mathbf{v}}_i$ is the projection of vertex $\mathbf{v}_i \in \mathbf{V}_{rpe}$ onto XY-plane. Finally, we get the set $\mathcal{C} = \{\mathbf{v}_i \in \mathbb{R}^3 \mid i = 1, \cdots, m_{ilm}\}$ represents the set of RPE surface vertices corresponding to each face in $\mathbf{F}_{ilm}$. A visual representation of the correspondence computation is shown in Figure 8.4, where the RPE's regular XY-grid is shown in blue and the projected ILM's vertices and edges are painted in red.

8.1.5 BMO Region Segmentation

For the ONH shape analysis, the region inside the BMO points is of special interest since BMO points represent the optic disc margin. To segment this region, we exploit the elliptic representation of the BMO points in $\mathbb{R}^2$ along with their center as mentioned in Equation (8.1). First, we compute the center of the ILM and the RPE surfaces corresponding to the center $\tilde{\mathbf{p}}_c$ of the BMO points. Let us consider that $\tilde{\mathbf{V}}_{ilm}$ and $\tilde{\mathbf{V}}_{rpe}$ represent projected sets of vertices onto XY-plane. Then, the indices of the centers of both surfaces can be computed as:

$$\begin{aligned} \text{ilmc} &= j \in 1, \cdots, n_{ilm} \mid \quad \min|\tilde{\mathbf{v}}_j^{ilm} - \tilde{\mathbf{p}}_c|, \\ \text{rpec} &= j \in 1, \cdots, n_{rpe} \mid \quad \min|\tilde{\mathbf{v}}_j^{rpe} - \tilde{\mathbf{p}}_c|, \end{aligned} \tag{8.5}$$

where $\tilde{\mathbf{v}}_j^{ilm} \in \tilde{\mathbf{V}}_{ilm}$ and $\tilde{\mathbf{v}}_j^{rpe} \in \tilde{\mathbf{V}}_{rpe}$. The vertices $\mathbf{v}_{ilmc} \in \mathbf{V}_{ilm}$ and $\mathbf{v}_{rpec} \in \mathbf{V}_{rpe}$ represent the center of the ILM and the RPE surfaces receptively.

To compute the BMO regions in the ILM and the RPE surfaces, we transform the ellipse into a circle using the affine transform. Let us consider $\tilde{\mathbf{p}}_i^e$ represents the fitted ellipse point to $\tilde{\mathbf{p}}_i$ as mentioned in section 8.1.3 and $\tilde{\mathbf{p}}_i^c$ denotes the corresponding affine transformed point on a circle of radius r. This transformation reduces the complexity of the BMO region computation and improves the speed of the algorithm. Now, by using the circular representation of the BMO points, the BMO regions in both RPE and ILM surfaces is computed as:

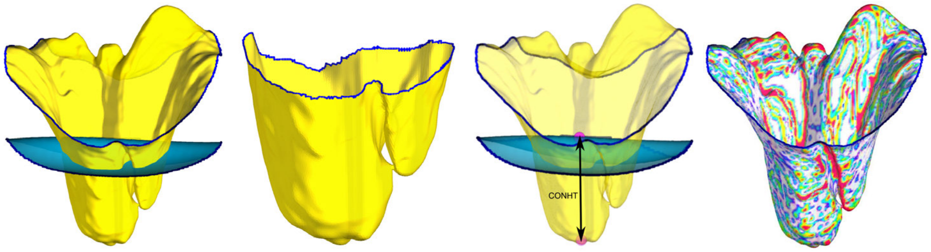
$$\begin{aligned} \Omega_{ilm} &= \{f_j \in \mathbf{F}_{ilm} \mid \quad |\tilde{\mathbf{c}}_j^{ilm} - \tilde{\mathbf{v}}_{ilmc}| \leq r\}, \\ \Omega_{rpe} &= \{f_j \in \mathbf{F}_{rpe} \mid \quad |\tilde{\mathbf{c}}_j^{rpe} - \tilde{\mathbf{v}}_{rpec}| \leq r\}, \end{aligned} \tag{8.6}$$

where r is the radius of the transformed circle and $\tilde{\mathbf{c}}_j^{ilm}$, $\tilde{\mathbf{c}}_j^{rpe}$ represent the centroid of a face in the ILM and the RPE surfaces respectively. The computation shown in Equation (8.6), is done using disk growing method.

8.1.6 3D Shape Parameters

ONH Cup Volume: The ONH cup is defined as a segment of the ILM surface which is below the RPE surface as shown in Figure 8.5(a) and (b). Note that the cup is not present in every ONH volume scan. For example, mostly in the case swollen ONH scans, the ILM surface is always above the RPE. To detect the availability of the ONH cup, we go to the each face $f_i \in \mathbf{F}_{ilm}$ and compute its centroid $\mathbf{c}_i$. As we discussed in Section 8.1.4, each face f_i of the ILM has the corresponding vertex $\mathbf{v}_i \in \mathcal{C}$ in RPE. If $c_i^z - v_i^z \geq 0$ for all faces in ILM, then there is no cup available in the ONH region. Otherwise, there is a cup. The terms c_i^z and v_i^z show the corresponding z-coordinates (height). Similarly, we can compute the cup region:

$$\Omega_{cup} = \{f_i \in \mathbf{F}_{ilm} \mid \quad (c_i^z - v_i^z) \leq 0\}, \tag{8.7}$$



(a) BMO Region (b) V_{cup} (c) CONHT (d) E_b

Figure 8.5: A visual representation of the different morphological parameters.

where Ω_{cup} consists of faces (triangles) of ILM, which are below the RPE surface. As it can be seen from Figure 8.5(b), the cup region is also a manifold surface with proper face normal orientation. To compute the volume of the cup accurately, we exploit the face normal information at each triangle of the region. The cup volume is computed using the following equation:

$$V_{cup} = \sum_{(f_i \in \Omega_{cup})} \mathbf{A}_i h_i, \tag{8.8}$$

where $\mathbf{A}_i$ represents the area of a triangle, which is a projection of the face f_i of the ILM surface onto XY-plane and h_i is the height w.r.t the RPE surface. These variables are defined as:

$$\mathbf{A}_i = \frac{1}{2}(\vec{\mathbf{e}}_0 \times \vec{\mathbf{e}}_1), \quad h_i = (c_i^z - v_i^z), \tag{8.9}$$

where $\vec{\mathbf{e}}_0$ and $\vec{\mathbf{e}}_1$ are the connected edges of the projected triangle. The cross product between the two edges will take care of the orientation of the corresponding face and enables a precise volume computation even in complex topological regions.

Central ONH Thickness (CONHT): The CONHT is defined as the height difference between the center of the ILM and the RPE surfaces as shown in Figure 8.5(c). The CONHT of a ONH volume scan is computed as:

$$\text{CONHT} = (v_{ilmc}^z - v_{rpec}^z), \tag{8.10}$$

where v_{ilmc}^z and v_{rpec}^z show the corresponding height value (z-coordinates) of the center vertices $\mathbf{v}_{ilmc}$ and $\mathbf{v}_{rpec}$ respectively introduced in section 8.1.5.

BMO Region Volume: The BMO region volume is computed using the segmented ILM and RPE surfaces. We separated the cup volume from the BMO region volume such that it does not include the cup volume, if it exists. Then, the BMO region volume can be computed as:

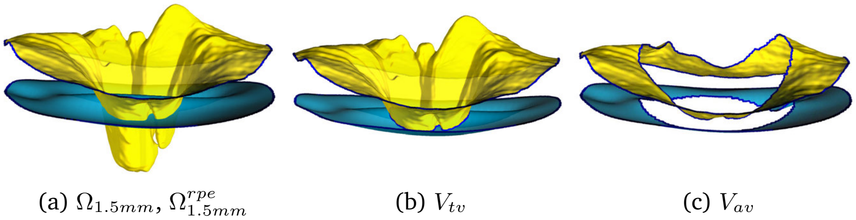
$$\Omega_{bmo} = \Omega_{ilm} \setminus \Omega_{cup}. \tag{8.11}$$

Similar to the ONH cup volume, we compute the BMO region volume using the similar formula:

$$V_{bmo} = \sum_{(f_i \in \Omega_{bmo})} \mathbf{A}_i h_i, \tag{8.12}$$

where $\mathbf{A}_i$ is area of the face f_i which belongs to the set Ω_{bmo} and is computed using Equation (8.9). Similarly, h_i is also computed using the corresponding vertices in the RPE surface as it is mentioned in Equation (8.9).

ONH Total Volume: Similar to the ONH cup and the BMO region volumes, the ONH total volume is



(a) $\Omega_{1.5mm}, \Omega^{rpe}_{1.5mm}$ (b) V_{tv} (c) V_{av}

Figure 8.6: A visual representation of total and annular volume regions.

also computed using ILM and RPE surfaces. The total volume is computed from the circular region, with radius $1.5\,\text{mm}$, centered at $\mathbf{v}_{ilmc}$ and $\mathbf{v}_{rpec}$ for ILM and RPE surfaces respectively, as shown in Figure 8.6(a). Similar to Equation (8.6), we compute the circular regions for ILM and RPE surfaces using the following equation:

$$\begin{aligned} \Omega^{ilm}_{1.5mm} &= \{f_j \in \mathbf{F}_{ilm} \mid \quad |\tilde{\mathbf{c}}^{ilm}_j - \tilde{\mathbf{v}}_{ilmc}| \leq 1.5\,\text{mm}\}, \\ \Omega^{rpe}_{1.5mm} &= \{f_j \in \mathbf{F}_{rpe} \mid \quad |\tilde{\mathbf{c}}^{rpe}_j - \tilde{\mathbf{v}}_{rpec}| \leq 1.5\,\text{mm}\}, \end{aligned} \tag{8.13}$$

where $\Omega^{ilm}_{1.5mm}$ and $\Omega^{rpe}_{1.5mm}$ are the sets consisting of all faces within the 1.5mm region of the ILM and RPE surfaces from their centers. The other parameters were defined in Equation (8.6). Then, the total volume region is calculated using Equation (8.13):

$$\Omega_{tv} = \Omega_{1.5mm} \setminus \Omega_{cup}, \tag{8.14}$$

where Ω_{tv} represents the total volume region on the ILM surface as shown in Figure 8.6(b), which is further employed for the total volume computation:

$$V_{tv} = \sum_{(f_i \in \Omega_{tv})} \mathbf{A}_i h_i, \tag{8.15}$$

where $\mathbf{A}_i$ is the area of the face $f_i \in \Omega_{tv}$ and h_i is the height w.r.t. corresponding vertex in the RPE surface similar to Equation (8.9).

ONH Annular Region Volume: The ONH annular region represents the ONH outer region, see Figure 8.6(c). On ILM surface this region is computed using the following equation:

$$\Omega_{av} = \Omega_{1.5mm} \setminus \Omega_{bmo}, \tag{8.16}$$

where Ω_{av} consists of all the ILM surface faces which belong to the annular region of the ONH. Similar to Equation (8.15), we compute the volume of the annular region using the ILM and RPE surfaces correspondence.

$$V_{av} = \sum_{(f_i \in \Omega_{av})} \mathbf{A}_i h_i. \tag{8.17}$$

The annular region volume helps to see the change in the outer region of the ONH volume in different cohort.

Bending Energy: The roughness on the ILM surface within the BMO region is an important parameter and commonly known as the bending energy on a manifold surface. The bending energy measures the fairness of a surface in terms of the curvature. In general, the outer region of the ILM surface is quite smooth and flat unlike the one inside the BMO, which has very complex topological structure. In this paper, we define the bending energy within the BMO region using the element-based normal voting tensor (ENVT) [114]. The ENVT exploits the orientation information (face normals) to compute a shape analysis operator at each face $f_i \in \Omega_{bmo}$ and is defined as:

$$\mathbf{M}_i = \frac{1}{\sum_{f_j \in \Omega_{bmo}} a_j} \sum_{f_j \in \Omega_{bmo}} a_j \mathbf{n}_j \otimes \mathbf{n}_j, \tag{8.18}$$

where $\mathbf{n}_j$ represents the normal of face f_i and $\mathbf{n}_j^T$ the transpose of $\mathbf{n}_j$. The term a_j is the area of the face f_i. To assure robustness against irregular sampling of the ILM surface, we weight Equation (8.18) by the corresponding face area a_j. The symbol $\otimes$ represents the outer product. The ENVT, $\mathbf{M}_i$ is a symmetric and positive semidefinite matrix, so, it can be decomposed into its spectral components:

$$\mathbf{M}_i = \sum_{k=0}^{2} \lambda_k^i \mathbf{e}_k \otimes \mathbf{e}_k, \tag{8.19}$$

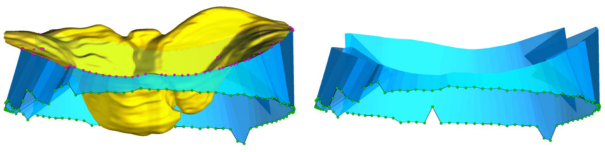
where $\breve{}_k^i = \{\lambda_0^i, \lambda_1^i, \lambda_2^i\}$ are the eigenvalues vector and these eigenvalues are sorted in decreasing order ($\lambda_0^i \geq \lambda_1^i \geq \lambda_2^i \geq 0$). The corresponding eigenvector is denoted by $\mathbf{e}_k$. In general, the dominant eigenvalue λ_0^i has the corresponding eigenvector in the direction of the face normal and the remaining two eigenvectors will be aligned to the principle curvature direction on the ILM surface. On the planer region, only λ_0^i will be significant, on the edge region, λ_0^i and λ_1^i will be significant and at the corners, all of these eigenvalues are significant. So using the anisotropic properties of these eigenvalues, we define the bending energy inside the BMO region using the following equation:

$$E_b = \sum_{f_i \in \Omega_{bmo}} \lambda_2^i + \lambda_3^i, \tag{8.20}$$

where λ_2^i and λ_3^i are the two least dominant eigenvalues of the ENVT of the face f_i. Figure 8.5(d) shows how each face of the BMO region is colored based on Equation (8.23). The color is scaled from white (flat regions) to red (sharp features).

BMO-MRW: BMO-MRW, has been proposed by [85] as a valid alternative structural measure. It computes the minimum distance between the BMO points and the ILM surface. The average BMO-MRW, denoted by avg_{mrw}, is calculated as:

$$avg_{mrw} = \frac{1}{n_p} \sum_{i=1}^{n_p} |\mathbf{p}_i^{mrw} - \mathbf{p}_i^e|. \tag{8.21}$$



(a) BMO-MRW with ILM (b) BMO-MRW

Figure 8.7: BMO-MRW surface with and without ILM.

BMO-MRW surface area: BMO-MRW surface area is computed by taking the whole region defined by all BMO-MRW. We combine the ellipse fitted BMO points $\mathcal{P}_{2D}$ and the z-coordinates from $\mathcal{P}$ and represent these as: $\mathcal{P}_e = \{\mathbf{p}_i^e \in \mathbb{R}^3 | i = 1 \cdots n_p\}$, as it can be seen in Figure 8.7 (green points). For each point $\mathbf{p}_i^e \in \mathcal{P}_e$, we compute a point $\mathbf{p}_i^{mrw}$ on the ILM surface:

$$\mathbf{p}_i^{mrw} = \mathbf{v}_j \in \mathbf{V}_{ilm} | \quad \min |\mathbf{v}_j - \mathbf{p}_i^e|. \tag{8.22}$$

The MRW points $\mathcal{P}_{mrw} = \{\mathbf{p}_i^{mrw} \in \mathbb{R}^3 | i = 1 \cdots n_p\}$ are lying on the ILM surface as shown in Figure 8.7(a) (violet points). We create a quad surface using point sets $\mathcal{P}_e$ and $\mathcal{P}_{mrw}$ by introducing edges between the corresponding vertices in both point sets and connecting the neighbor points as shown in Figure 8.7(b). The number of quad elements in the MRD surface is equal to the number of points in each point sets and is represented as: $\mathcal{Q} = \{q_i | i = 1 \cdots n_p\}$. MRW-MRA is computed as:

$$A_{mrw} = \sum_{q_i \in \mathcal{Q}} a_{q_i}, \tag{8.23}$$

where a_{q_i} represents the area of the quad q_i.
BMO Area: As shown in Figure 8.3, BMO area (BMOA) represents the area under the fitted ellipse to the BMO points and is computed using the conic representation:

$$A_{bmo} = \pi r_1 r_2, \tag{8.24}$$

where r_1 and r_2 are the major and minor axes of the fitted ellipse.

8.2 Statistical Shape Analysis of the ILM Surface

To understand the variation in the shape of the ONH regrading different neurological disorders, we use statistical shape analysis tool that quantifies the shape variability of a population of shapes. This tool identifies statistical shape difference between healthy controls and patients and provides an improved diagnosis. From geometry processing point of view, statistical shape analysis is closely related to super-resolution surface reconstruction [31,54], where several low resolution 3D scan are merged to reconstruct a high resolution surface, which has enhanced feature and surface quality.

For statistical shape analysis of ONH surfaces, we follow a procedure, which is similar to super-resolution surface reconstruction techniques. To perform statistical analysis on ONH shapes, first of all, we require intrinsic correspondence information between the ONH surfaces and then next step , we compute a mean shape of a group of ONH surfaces using the correspondence information. Let us consider that $\mathcal{M}_{ilm}^i$ represents a set of ILM surfaces, where $i = 0, \cdots, n_s - 1$ and n_s is the number of ILM surfaces. During statistical shape analysis, $\mathcal{M}_{ilm}^0 = \{\mathbf{V}_{ilm}^0, \mathbf{F}_{ilm}^0\}$ is considered as the reference geometry and other geometries of $\mathcal{M}_{ilm}^i$ are registered to the $\mathcal{M}_{ilm}^0$.

8.2.1 ONH Surfaces Alignment

This is the first stage of the whole statistical shape analysis process of a set of ONH surfaces. At this stage, we align ILM surfaces with each other. In general, we choose one of the surface as a reference geometry and register other geometries w.r.t the reference geometry rigidly. This shape registration algorithm uses point sets as a geometric data representation. Here, we are following the ICP (iterative closest point) registration algorithm [50] along with the singular value decomposition. we compute the translational and rotational displacement between the geometries based on the correspondence points. It is an iterative registration algorithm and keeps registering the geometries until it meets the convergence criteria. After the registration process, we compute the corresponding vertices in the ILM surfaces $(\mathcal{M}_{ilm}^i \, | i = 1, \cdots, n_s - 1)$ w.r.t the each vertex $(\mathbf{v}_j \, | j = 0, \cdots, n_{ilm}^0)$ of the reference geometry $\mathcal{M}_{ilm}^0$. Let us consider $\mathbf{C}_j$ is a matrix of size $(n_s - 1 \times 3)$ and each row of $\mathbf{C}_j$ consists of the corresponding vertices w.r.t. $\mathbf{v}_j$.

8.2.2 Mean Shape Reconstruction

Mean shape of a set of ILM surfaces is computed similar to Method [54]. Let us consider that $\mathcal{M}^s_{ilm}$ represents the means shape of the ILM surfaces and it has the same number of the vertices of the reference geometry $\mathcal{M}^0_{ilm}$ and $\mathbf{v}^s_j$ is the vertex of the mean shape and it is computed as:

$$\mathbf{v}^s_j = \frac{1}{n_s} \sum_{j=1}^{n_s-1} \mathbf{C}_j. \tag{8.25}$$

The above equation computes the centroid of the corresponding points. Figure 8.8 shows the mean shape of 80 ILM surfaces. The computation is done in two steps, alignment and mean shape reconstruction. Figure 8.8 (c) shows the variation of the 80 shapes and this variation is computed using the standard deviation of the corresponding points of ILM surfaces.


(a) ILM Surface (b) Mean Shape (c) Variations

Figure 8.8: (a) A single noisy ILM surface. (b) The ILM mean surface, which computed using Equation (8.25) with 80 ILM surfaces. (c) Standard deviation of the shapes.

8.3 Experiments and Results

In order to validate our method, we performed repeated-measurement reliability tests, investigated ONH shape in healthy subjects, and tested if our method is able to detect differences in patients with diseases known to affect the ONH in form of swelling and atrophy. Finally, we measured the implementation's performance.

8.3.1 Re-test Reliability

In order to estimate the repeated-measurement reliability we took three repeated scans of both eyes from 10 healthy subjects measured twice at a time interval of one week. Table 8.1 shows the repeatability results. We see that our method scores highly at every parameter presented, with lowest intraclass correlation coefficient (ICC) of 0.905 for CONHT, and highest 0.998 for V_{cup}.

Parameters	ICC	LCI	UCI
ONH Cup Vol.(V_{cup}) (mm^3)	0.998	0.996	0.999
Central ONH Thickness (CONHT) (mm)	0.905	0.813	0.958
BMO Region Vol (V_{bmo}) (mm^3)	0.993	0.986	0.997
ONH Total Vol. (V_{tv}) (mm^3)	0.995	0.989	0.998
ONH Annular Vol. (V_{av}) (mm^3)	0.983	0.965	0.993
Bending Energy (E_b)	0.911	0.824	0.961
BMO-MRW Surface Area (A_{mrw}) (mm^2)	0.910	0.823	0.960
BMO-MRW (d_{mrw}) (mm)	0.993	0.986	0.997
BMO Area (A_{bmo}) (mm^2)	0.989	0.976	0.995

Parameters	HC		IIH		ON		GEE		
	Mean (SD)	Min-Max	Mean (SD)	Min-Max	Mean (SD)	Min-Max	p(HC-IIH)	p(HC-ON)	p(IIH-ON)
V_{cup}	0.044 (0.075)	0.000-0.370	0.004(0.012)	0.000-0.049	0.035(0.052)	0.000-0.256	**0.0012**	0.353	**1.95E-07**
CONHT	0.002 (0.25)	-0.545-0.550	0.403(0.340)	-0.242-0.966	0.012(0.231)	-0.399-0.668	**9.91E-06**	0.747	**4.22E-06**
V_{bmo}	0.572 (0.21)	0.107-1.035	1.521 (1.022)	0.436-4.3098	0.444(0.189)	0.105-1.050	**5.02E-05**	**0.0012**	**2.66E-06**
V_{tv}	2.523 (0.250)	2.114-2.988	3.309 (0.799)	2.351-5.365	2.163(0.295)	1.522-3.087	**2.57E-05**	**5.28E-13**	**6.54E-10**
V_{av}	1.906(0.181)	1.504-2.268	1.78(0.357)	0.572-2.200	1.683(0.210)	1.162-2.118	0.146	**2.19E-10**	0.257
E_b	0.026 (0.009)	0.011-0.065	0.029 (0.013)	0.008-0.066	0.029(0.010)	0.011-0.066	0.477	0.038	0.793
A_{mrw}	1.791 (0.459)	0.763-2.789	2.975(1.016)	1.439-5.351	1.414(0.364)	0.615-2.765	**1.35E-06**	**5.44E-06**	**3.94E-11**
d_{mrw}	0.135(0.054)	0.261-0.035	0.263 (0.105)	0.127-0.447	0.093(0.040)	0.030-0.260	**3.52E-07**	**2.69E-05**	**1.81E-12**
A_{bmo}	1.906 (0.352)	1.158-2.646	2.682 (1.109)	1.057-5.709	1.861(0.377)	1.030-3.326	0.0028	0.407	**0.001**

Table 8.2: Analysis of all the 3D parameters defined for the HC and patient group. The last column shows the GEE analysis between the two groups. Abbreviations: HC - healthy controls. SD - standard deviation, Min - minimum value, Max - maximum value, GEE - generalized estimating equation models analysis accounting for the inter-eye/intra-subject dependencies, p - p value

Table 8.1: Repeatability test for the 3D parameters. Abbreviations: ICC - intra-class correlation coefficient, LCI - lower boundary of 95% confidence interval and UCI - upper boundary of 95% confidence interval.

8.3.2 Clinical Validation

In this section, we present the results of our automatic pipeline approach for 248 OCT scans, from three groups, 71 healthy control eyes (HC), 31 eyes of patients suffering from idiopathic intracranial hypertension (IIH) [52], which causes swelling of the ONH (papilledema), and 146 eyes of patients with autoimmune central nervous system disorders and a history of optic neuritis (ON), an inflammatory optic neuropathy that damages the optic nerve leading to neuroaxonal degeneration [83, 124]. We hypothesized that patients with IIH would show significant changes in comparison to healthy controls indicating ONH swelling. And that patients with a history of ON would show significant changes in comparison to healthy controls indication ONH atrophy. The results presented in Table 8.2 demonstrate that our approach successfully captures the differences. The only parameter showing no difference between groups is bending energy, E_b. Although we expected that in the IIH affected eyes the ONH shape inside the BMO to have a smoother convex shape, the data is still extremely heterogeneous. Thus, the bending energy reflects this extreme variability in the data. Especially inside the BMO the topologies from one ONH to the other can be extremely different.

8.4 Summary

This chapter introduced an automatic and robust pipeline for the computation of several parameters that characterize ONH shape. In particular, this chapter consists of an algorithm that computes true 3D parameters derived from triangulated mesh surfaces of the ILM and RPE. The proposed method uses the RPE lower boundary, which is a smooth 2.5D surface and 3D ILM surface, which is computed using a novel Chan-Vese level set approach and Marching Cube algorithm. The preprocessing of these surfaces are done by employing a robust mesh denoising method. The resulting high fidelity surface preserves features with minimal shrinkage. It is important to remember that we are interested in detecting even minimal shape changes, thus, obtaining a surface that has no staircase artifacts, but preserves features is crucial for parameter computation. Having three components, ILM and RPE and BMO points, the proposed method computes the correspondence between surfaces ILM and RPE, such that each face of the ILM surface has a corresponding vertex in RPE. After this step, we have all the ingredients to derive the parameters. Although the computed parameters, except the bending energy, were already introduced in several previous studies (mostly for the investigation of glaucoma or papilledema), this is the first algorithm to provide quantifiable measurements derived in a true 3D context. Furthermore, this method able to provide a first proof of the applicability in a clinical setting, by showing the power of several parameters introduced in differentiating between inflammation in form of swelling (IIH) and destruction in form of reduction of layer thickness (ON).

In summary, the results showed that the proposed method successfully and robustly utilizes ILM, RPE and BMO points and it enables computing the morphometric parameters, that capture shape characteristics of the ONH and further can be used to diagnose several neurological disorders. In future, this algorithm can provide an aid to diagnose several neurological disorders.

Chapter 9

Summary

Through the development of this thesis, starting from the curvature tensor, we have been able to understand the variation of tangent vectors to define a shape analysis operator and also a relationship between the classical shape operator and the curvature tensor on a triangular surface. In continuation, the first part of thesis analyzed the variation of surface normals and introduced a shape analysis operator, which is further used for mesh and point set denoising. In the second part of thesis, mathematical modeling and shape quantification algorithms are introduced for retinal shape analysis.

In the first half, this thesis followed the concept of the variation of surface normals, which is termed as the normal voting tensor and derived a relation between the shape operator and the normal voting tensor. The concept of the directional and the mean curvatures is extended on the dual representation of a triangulated surface. A normal voting tensor is defined on each triangle of a geometry and termed as the element-based normal voting tensor (ENVT). Later, a deformation tensor is extracted from the ENVT and it consists of anisotropy of a surface and the mean curvature vector is defined based on the ENVT deformation tensor.

A ENVT-based mesh denoising algorithm is introduced, where the ENVT is used as a shape operator. A binary optimization technique is applied on the spectral components of the ENVT that helps the algorithm to retain sharp features in the concerned geometry and improves the convergence rate of the algorithm. Later, a stochastic analysis of the effect of noise on the triangular mesh based on the minimum edge length of the elements in the geometry is explained. It gives an upper bound to the noise standard deviation to have minimum probability for flipped element normals.

The ENVT-based mesh denoising concept is extended for a point set denoising, where noisy vertex normals are filtered using the vertex-based NVT and the binary optimization. For vertex update stage in point set denoising, we added different constraints to the quadratic error metric based on features (edges and corners) or non-feature (planar) points.

This thesis also investigated a robust statistics framework for face normal bilateral filtering and proposed a robust and high fidelity two-stage mesh denoising method using Tukey's bi-weight function as a robust estimator, which stops the diffusion at sharp features and produces smooth umbilical regions. This algorithm introduced a novel vertex update scheme, which uses a differential coordinate-based Laplace operator along with an edge-face normal orthogonality constraint to produce a high-quality mesh without face normal flips and it also makes the algorithm more robust against high-intensity noise.

The second half of thesis focused on application of the proposed geometric processing algorithms on the OCT (optical coherence tomography) scan data for quantification of the human retinal shape. The retina is a part of the central nervous system and comprises a similar cellular composition as the brain. Therefore, many neurological disorders affect the retinal shape and these neuroinflammatory conditions are known to cause modifications to two important regions of the retina: the fovea and the optical nerve head (ONH). This thesis consists of an accurate and robust shape modeling of these regions to diagnose several neurological disorders by detecting the shape changes. For the fovea, a parametric modeling algorithm is introduced using Cubic Beziér and this algorithm derives several 3D shape parameters, which quantify the foveal shape with high accuracy. For the ONH, a 3D shape analysis algorithm is introduced to measure the shape variation regarding different neurological disorders. The proposed algorithm uses triangulated manifold surfaces of two different layers of the retina to derive several 3D shape parameters. The experimental results of the fovea and the ONH morphometry confirmed that these algorithms can provide an aid to diagnose several neurological disorders.

Bibliography

[1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, *Gaussian kd-trees for fast high-dimensional filtering*, ACM Trans. Graph., 28 (2009), pp. 21:1–21:12.

[2] P. Albrecht, C. Blasberg, M. Ringelstein, A.-K. Müller, D. Finis, R. Guthoff, E.-M. Kadas, W. Lagreze, O. Aktas, H.-P. Hartung, F. Paul, A. U. Brandt, and A. Methner, *Optical coherence tomography for the diagnosis and monitoring of idiopathic intracranial hypertension*, J. of Neurol., 264 (2017), pp. 1370–1380.

[3] P. Albrecht, R. Fröhlich, H.-P. Hartung, B. C. Kieseier, and A. Methner, *Optical coherence tomography measures axonal loss in multiple sclerosis independently of optic neuritis*, J. of Neurol., 254 (2007), pp. 1595–1596.

[4] M. Alexa, *Differential coordinates for local mesh morphing and deformation*, The Vis. Comput., 19 (2003), pp. 105–114.

[5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, *Point set surfaces*, in Proceedings of the Conference on Visualization ’01, VIS ’01, Washington, DC, USA, 2001, IEEE Computer Society, pp. 21–28.

[6] B. Anand-Apte and J. Hollyfield, *Developmental anatomy of the retinal and choroidal vasculature*, in Encyclopedia of the Eye, Elsevier, 2010, pp. 9–15.

[7] C. L. Bajaj and G. Xu, *Anisotropic diffusion of surfaces and functions on surfaces*, ACM Trans. Graph., 22 (2003), pp. 4–32.

[8] Y. Barak, M. P. Sherman, and S. Schaal, *Mathematical Analysis of Specific Anatomic Foveal Configurations Predisposing to the Formation of Macular Holes*, Investig. Ophthalmol. and Vis. Sci., 52 (2011), pp. 8266–8270.

[9] D. Barash, *Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation*, IEEE Trans. Pattern Anal. Mach. Intell., 24 (2002), pp. 844–847.

[10] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, *A survey of surface reconstruction from point clouds*, Computer Graphics Forum, 36 (2017), pp. 301–329.

[11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, *The ball-pivoting algorithm for surface reconstruction*, IEEE Trans. Vis. Comput. Graph., 5 (1999), pp. 349–359.

[12] Z. Bian and R. Tong, *Feature-preserving mesh denoising based on vertices classification*, Comput. Aided Geom. Des., 28 (2011), pp. 50–64.

[13] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, *Robust anisotropic diffusion*, IEEE Trans. Image Process., 7 (1998), pp. 421–432.

[14] M. Botsch, P. Alliez, L. Kobbelt, M. Pauly, and B. Lévy, *Polygon mesh processing*, A K Peters, 2010.

[15] M. Botsch, M. Pauly, C. Rossl, S. Bischoff, and L. Kobbelt, *Geometric modeling based on triangle meshes*, in ACM SIGGRAPH 2006 Courses, ACM.

[16] M. Centin and A. Signoroni, *Mesh denoising with (geo)metric fidelity*, IEEE Trans. Vis. Comput. Graph., 24 (2018), pp. 2380–2396.

[17] B. C. Chauhan and C. F. Burgoyne, *From clinical examination of the optic disc to clinical assessment of the optic nerve head: A paradigm change*, American J. of Ophthalmol., 156 (2013), pp. 218 – 227.

[18] B. C. Chauhan, V. M. Danthurebandara, G. P. Sharpe, S. Demirel, C. A. Girkin, C. Y. Mardin, A. F. Scheuerle, and C. F. Burgoyne, *Bruch's membrane opening minimum rim width and retinal nerve fiber layer thickness in a normal white population*, Ophthalmol., 122 (2017), pp. 1786–1794.

[19] T. Y. P. Chui, Z. Zhong, H. Song, and S. A. Burns, *Foveal avascular zone and its relationship to foveal pit shape*, Optometry and Vis. Sci., 89 (2012), pp. 602–610.

[20] U. Clarenz, U. Diewald, and M. Rumpf, *Anisotropic geometric diffusion in surface processing*, in Proc. Vis. 2000., Oct 2000, pp. 397–405.

[21] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, *Implicit fairing of irregular meshes using diffusion and curvature flow*, in Proc. of the 26th Annu. Conf. on Comput. Graph. and Interact. Tech., ACM Press., 1999, pp. 317–324.

[22] Y. Ding, B. Spund, S. Glazman, E. M. Shrier, S. Miri, I. Selesnick, and I. Bodis-Wollner, *Application of an oct data-based mathematical model of the foveal pit in parkinson disease*, J. of Neural Transm, 121 (2014), pp. 1367–1376.

[23] A. M. Dubis, J. T. McAllister, and J. Carroll, *Reconstructing foveal pit morphology from optical coherence tomography imaging*, Br. J. of Ophthalmol., 93 (2009), pp. 1223–1227.

[24] F. Durand and J. Dorsey, *Fast bilateral filtering for the display of high-dynamic-range images*, in Proc. of the 29th Annu. Conf. on Comput. Graph. and Interact. Tech., SIGGRAPH '02, New York, NY, USA, 2002, ACM, pp. 257–266.

[25] P. Enders, W. Adler, F. Schaub, M. M. Hermann, T. Dietlein, C. Cursiefen, and L. M. Heindl, *Novel bruch's membrane opening minimum rim area equalizes disc size dependency and offers high diagnostic power for glaucoma*, Investigat. Ophthalmol. & Vis. Sci., 57 (2016), p. 6596.

[26] P. Enders, F. Schaub, W. Adler, R. Nikoluk, M. M. Hermann, and L. M. Heindl, *The use of bruch's membrane opening-based optical coherence tomography of the optic nerve head for glaucoma detection in microdiscs*, Br. J. of Ophthalmol., 101 (2017), pp. 530–535.

[27] P. Enders, F. Schaub, M. M. Hermann, C. Cursiefen, and L. M. Heindl, *Neuroretinal rim in non-glaucomatous large optic nerve heads: a comparison of confocal scanning laser tomography and spectral domain optical coherence tomography*, Br. J. of Ophthalmol., 101 (2017), pp. 138–142.

[28] A. P. et al., *Retinal layer segmentation in multiple sclerosis: a systematic review and meta-analysis*, The Lancet Neurol., 16 (2017), pp. 797–812.

[29] H. Fan, Q. Peng, and Y. Yu, *Robust feature-preserving mesh denoising based on consistent subneighborhoods*, IEEE Trans. Vis. Comput. Graph., 16 (2009), pp. 312–324.

[30] G. E. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code*, Academic Press, Inc., Orlando, FL, USA, 4th ed., 1996.

[31] A. Fatemeh, K. Y. Joo, S. Andrei, and A. Nina, *Rotating scans for systematic error removal*, vol. 28, pp. 1319–1326.

[32] D. A. Field, *Laplacian smoothing and delaunay triangulations*, Commun. in Appl. Numer. Methods, 4 (1988), pp. 709–712.

[33] A. Fitzgibbon, M. Pilu, and R. B. Fisher, *Direct least square fitting of ellipses*, IEEE Trans. Pattern Anal. Mach. Intell, 21 (1999), pp. 476–480.

[34] S. Fleishman, I. Drori, and D. Cohen-Or, *Bilateral mesh denoising*, in ACM SIGGRAPH 2003, pp. 950–953.

[35] S. K. Gardiner, R. Ren, H. Yang, B. Fortune, C. F. Burgoyne, and S. Demirel, *A method to estimate the amount of neuroretinal rim tissue in glaucoma: Comparison with current methods for measuring rim area*, American J. of Ophthalmol., 157 (2014), pp. 540–549.

[36] K. Gawlik, F. Hauser, F. Paul, A. Brandt, and E. M. Kadas, *An active contour method for ilm segmentation in onh volume scans in retinal oct*, submitted to IEEE Trans on Med. Imag., (2017).

[37] E. Gibson, M. Young, M. V. Sarunic, and M. F. Beg, *Optic nerve head registration via hemispherical surface and volume registration*, IEEE Trans. Biomed. Eng., 57 (2010), pp. 2592–2595.

[38] E. Gordon-Lipkin, B. Chodkowski, D. S. Reich, S. A. Smith, M. Pulicken, L. J. Balcer, E. M. Frohman, G. Cutter, and P. A. Calabresi, *Retinal nerve fiber layer is associated with brain atrophy in multiple sclerosis*, Neurol., 69 (2007), pp. 1603–1609.

[39] O. S.-S. C. f. t. N. I. I. H. S. Group, *Baseline OCT measurements in the idiopathic intracranial hypertension treatment trial, part i: Quality control, comparisons, and variability*, Investig. Ophthalmol. and Vis. Sci., 55 (2014), pp. 8180–8188.

[40] G. Guennebaud and M. Gross, *Algebraic point set surfaces*, ACM Trans. Graph., 26 (2007).

[41] D. X. Hammer, N. V. Iftimia, R. D. Ferguson, C. E. Bigelow, T. E. Ustun, A. M. Barnaby, and A. B. Fulton, *Foveal fine structure in retinopathy of prematurity: An adaptive optics fourier domain optical coherence tomography study*, Investig. Ophthalmol. and Vis. Sci., 49 (2008), pp. 2061–2070.

[42] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics - The Approach Based on Influence Functions*, Wiley, 1986.

[43] L. He and S. Schaefer, *Mesh denoising via l0 minimization*, ACM Trans. Graph., 32 (2013), pp. 64:1–64:8.

[44] G. Heidary and J. F. Rizzo, *Use of optical coherence tomography to evaluate papilledema and pseudopapilledema*, Semin. in Ophthalmol., 25 (2010), pp. 198–205.

[45] K. Hildebrandt and K. Polthier, *Anisotropic filtering of non-linear surface features*, Comput. Graph. Forum, 23 (2004), pp. 391–400.

[46] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Surface reconstruction from unorganized points*, SIGGRAPH Comput. Graphics, 26 (1992), pp. 71–78.

[47] D. Huang, E. Swanson, C. Lin, J. Schuman, W. Stinson, W. Chang, M. Hee, T. Flotte, K. Gregory, C. Puliafito, and a. et, *Optical coherence tomography*, Science, 254 (1991), pp. 1178–1181.

[48] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, *Consolidation of unorganized point clouds for surface reconstruction*, ACM Trans. Graphics, 28 (2009), pp. 176:1–176:7.

[49] R. Ihaka and R. Gentleman, *R: A language for data analysis and graphics*, J. of Computat. and Graph. Stat., 5 (1996), pp. 299–314.

[50] M. Irani and S. Peleg, *Improving resolution by image registration*, CVGIP: Graph. Model. and Image Process., 53 (1991), pp. 231 – 239.

[51] T. R. Jones, F. Durand, and M. Desbrun, *Non-iterative, feature-preserving mesh smoothing*, ACM Trans. Graph., 22 (2003), pp. 943–949.

[52] E. M. Kadas, F. Kaufhold, C. Schulz, F. Paul, K. Polthier, and A. U. Brandt, *3d optic nerve head segmentation in idiopathic intracranial hypertension*, in Bildverarbeitung für die Medizin 2012, Springer Berlin Heidelberg, 2012, pp. 262–267.

[53] F. Kaufhold, E. M. Kadas, C. Schmidt, H. Kunte, J. Hoffmann, H. Zimmermann, T. Oberwahrenbrock, L. Harms, K. Polthier, A. U. Brandt, and F. Paul, *Optic nerve head quantification in idiopathic intracranial hypertension by spectral domain OCT*, PLoS ONE, 7 (2012).

[54] Y. J. Kil, B. Mederos, and N. Amenta, *Laser scanner super-resolution*, in Proc. of the 3rd Eurographics, SPBG'06, 2006, pp. 9–16.

[55] H. S. Kim, H. K. Choi, and K. H. Lee, *Feature detection of triangular meshes based on tensor voting theory*, Computer-Aided Des., 41 (2009), pp. 47 – 58.

[56] H. Kolb, *How the retina works*, American Scientist, 91 (2003), pp. 28–35.

[57] C. Lange and K. Polthier, *Anisotropic smoothing of point sets*, Computer Aided Geometric Design, 22 (2005), pp. 680 – 692.

[58] K.-W. Lee and W.-P. Wang, *Feature-preserving mesh denoising via bilateral normal filtering*, in 9th Int. Conf. on Comput. Aided Des. and Comput. Graph. (CAD-CG'05), Dec 2005.

[59] S. LEE, M. HEISLER, P. J. MACKENZIE, M. V. SARUNIC, AND M. F. BEG, *Quantifying variability in longitudinal peripapillary rnfl and choroidal layer thickness using surface based registration of oct images*, Transl. Vis. Sci. & Tech., 6 (2017).

[60] S. LEE, E. LEBED, M. V. SARUNIC, AND M. F. BEG, *Exact surface registration of retinal surfaces from 3-d optical coherence tomography images*, IEEE Trans. Biomed. Eng., 62 (2015), pp. 609–617.

[61] S. LEE, M. YOUNG, M. V. SARUNIC, AND M. F. BEG, *End-to-end pipeline for spectral domain optical coherence tomography and morphometric analysis of human optic nerve head*, J. of Medical and Biol. Eng., 31 (2011), pp. 111–119.

[62] Y. LIPMAN, D. COHEN-OR, D. LEVIN, AND H. TAL-EZER, *Parameterization-free projection for geometry reconstruction*, ACM Trans. Graphics, 26 (2007).

[63] Y. LIPMAN, O. SORKINE, M. ALEXA, D. COHEN-OR, D. LEVIN, C. RÖSSL, AND H.-P. SEIDEL, *Laplacian framework for interactive mesh editing*, Int. J. of Shape Model. (IJSM), 11 (2005), pp. 43–61.

[64] L. LIU, W. MARSH-TOOTLE, E. N. HARB, W. HOU, Q. ZHANG, H. A. ANDERSON, T. T. NORTON, K. K. WEISE, J. E. GWIAZDA, L. HYMAN, AND THE COMET GROUP, *A sloped piecemeal Gaussian model for characterising foveal pit shape*, Ophthalmic and Physiol. Opt., 36 (2016), pp. 615–631.

[65] W. E. LORENSEN AND H. E. CLINE, *Marching cubes: A high resolution 3d surface construction algorithm*, in Proc. of the 14th Annu. Conf. on Comput. Graph. and Interact. Tech., SIGGRAPH '87, New York, NY, USA, 1987, ACM, pp. 163–169.

[66] X. LU, W. CHEN, AND S. SCHAEFER, *Robust mesh denoising via vertex pre-filtering and l1-median normal filtering*, Comput. Aided Geom. Des., 54 (2017), pp. 49 – 60.

[67] X. LU, Z. DENG, AND W. CHEN, *A robust scheme for feature-preserving mesh denoising*, IEEE Trans. Vis. Comput. Graph., 22 (2016), pp. 1181–1194.

[68] E. MATTEI AND A. CASTRODAD, *Point cloud denoising via moving rpca*, Computer Graphics Forum, 36 (2017), pp. 123–137.

[69] B. K. MCCAFFERTY, M. A. WILK, J. T. MCALLISTER, K. E. STEPIEN, A. M. DUBIS, M. H. BRILLIANT, J. L. ANDERSON, J. CARROLL, AND C. G. SUMMERS, *Clinical insights into foveal morphology in albinism*, J. of pediatr. ophthalmol. & Strabismus, 52 (2015), pp. 167–172.

[70] G. MEDIONI, *Tensor voting: Theory and applications*, 2000.

[71] D. R. MUTH AND C. W. HIRNEISS, *Structure–function relationship between bruch's membrane opening–based optic nerve head parameters and visual field defects in glaucoma*, Investigat. Ophthalmol. & Vis. Sci., 56 (2015), pp. 3320–3328.

[72] A. NEALEN, T. IGARASHI, O. SORKINE, AND M. ALEXA, *Laplacian mesh optimization*, in Proc. of the 4th Int. Conf. on Comput. Graph. and Interact. Tech. in Australasia and Southeast Asia, GRAPHITE '06, 2006, pp. 381–389.

[73] B. NESMITH, A. GUPTA, T. STRANGE, Y. SCHAAL, AND S. SCHAAL, *Mathematical Analysis of the Normal Anatomy of the Aging FoveaAnatomy of the Aging Fovea*, Investig. Ophthalmol. and Vis. Sci., 55 (2014), pp. 5962–5966.

[74] T. Oberwahrenbrock, M. Ringelstein, S. Jentschke, K. Deuschle, K. Klumbies, J. Bellmann-Strobl, J. Harmel, K. Ruprecht, S. Schippling, H.-P. Hartung, O. Aktas, A. U. Brandt, and F. Paul, *Retinal ganglion cell and inner plexiform layer thinning in clinically isolated syndrome*, Multiple Scler. J., 19 (2013), pp. 1887–1895.

[75] F. C. Oertel, J. Kuchling, H. Zimmermann, C. Chien, F. Schmidt, B. Knier, J. Bellmann-Strobl, T. Korn, M. Scheel, A. Klistorner, K. Ruprecht, F. Paul, and A. U. Brandt, *Microstructural visual system changes in AQP4-antibody–seropositive NMOSD*, Neurol. Neuroimmunol. Neuroinflamm., 4 (2017), p. e334.

[76] F. C. Oertel, H. Zimmermann, J. Mikolajczak, M. Weinhold, E. M. Kadas, T. Oberwahrenbrock, F. Pache, J. Bellmann-Strobl, K. Ruprecht, F. Paul, and A. U. Brandt, *Contribution of blood vessels to retinal nerve fiber layer thickness in nmosd*, Neurol Neuroimmunol Neuroinflamm., 4 (2017).

[77] Y. Ohtake, A. Belyaev, and I. Bogaevski, *Mesh regularization and adaptive smoothing*, Computer-Aided Des., 33 (2001), pp. 789 – 800.

[78] Y. Ohtake, A. Belyaev, and H. Seidel, *Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals*, in IN VIS. MODEL. AND VIS., 2002.

[79] T. Ojala, M. Pietikainen, and D. Harwood, *A comparative study of texture measures with classification based on featured distributions*, Pattern Recognition, 29 (1996), pp. 51–59.

[80] C. Öztireli, G. Guennebaud, and M. Gross, *Feature preserving point set surfaces based on non-linear kernel regression*, vol. 28, 2009, pp. 493–501.

[81] G. Patanè and M. Spagnuolo, *Triangle mesh duality: Reconstruction and smoothing*, in Math. of Surf., M. J. Wilson and R. R. Martin, eds., Berlin, Heidelberg, 2003, Springer Berlin Heidelberg, pp. 111–128.

[82] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. on Pattern Anal. and Mach. Intell., 12 (1990), pp. 629–639.

[83] A. Petzold, M. P. Wattjes, F. Costello, J. Flores-Rivera, C. L. Fraser, K. Fujihara, J. Leavitt, R. Marignier, F. Paul, S. Schippling, C. Sindic, P. Villoslada, B. Weinshenker, and G. T. Plant, *The investigation of acute optic neuritis: a review and proposed protocol*, Nature Reviews Neurology, 10 (2014), pp. 447–458.

[84] F. Pollet-Villard, C. Chiquet, J.-P. Romanet, C. Noel, and F. Aptel, *Structure-function relationships with spectral-domain optical coherence tomography retinal nerve fiber layer and optic nerve head measurements*, IInvestigat. Ophthalmol. & Vis. Sci., 55 (2014), pp. 2953–2962.

[85] A. S. C. Reis, N. O'Leary, H. Yang, G. P. Sharpe, M. T. Nicolela, C. F. Burgoyne, and B. C. Chauhan, *Influence of clinically invisible, but optical coherence tomography detected, optic disc margin anatomy on neuroretinal rim evaluation*, Investigat. Ophthalmol. & Vis. Sci., 53 (2012), p. 1852.

[86] P. Scheibe, A. Lazareva, U.-D. Braumann, A. Reichenbach, P. Wiedemann, M. Francke, and F. G. Rauscher, *Parametric model for the 3d reconstruction of individual fovea shape from OCT data*, Exp. Eye Res., 119 (2014), pp. 19–26.

[87] P. SCHEIBE, M. T. ZOCHER, M. FRANCKE, AND F. G. RAUSCHER, *Analysis of foveal characteristics and their asymmetries in the normal population*, Exp. Eye Res., 148 (2016), pp. 1–11.

[88] E. SCHNEIDER, H. ZIMMERMANN, T. OBERWAHRENBROCK, F. KAUFHOLD, E. M. KADAS, A. PETZOLD, F. BILGER, N. BORISOW, S. JARIUS, B. WILDEMANN, K. RUPRECHT, A. U. BRANDT, AND F. PAUL, *Optical coherence tomography reveals distinct patterns of retinal damage in neuromyelitis optica and multiple sclerosis*, PLOS ONE, 8 (2013), pp. 1–10.

[89] J. R. SHEWCHUK, *Delaunay Refinement Mesh Generation*, PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997. Available as Technical Report CMU-CS-97-137.

[90] J. SOLOMON, K. CRANE, A. BUTSCHER, AND C. WOJTAN, *A general framework for bilateral and mean shift filtering*, ArXiv e-prints, 32 (2014).

[91] O. SORKINE, *Differential representations for mesh processing*, Comput. Graph. Forum, 25 (2006), pp. 789–807.

[92] O. SORKINE, D. COHEN-OR, D. IRONY, AND S. TOLEDO, *Geometry-aware bases for shape approximation*, IEEE Trans. Vis. Comput. Graph., 11 (2005), pp. 171–180.

[93] O. SORKINE, D. COHEN-OR, AND S. TOLEDO, *High-pass quantization for mesh encoding*, in Proc. of the 2003 Eurographics, SGP '03, 2003, pp. 42–51.

[94] Z.-X. SU, H. WANG, AND J.-J. CAO, *Mesh denoising based on differential coordinates*, in IEEE Int. Conf. on Shape Model. and Appl., June 2009, pp. 1–6.

[95] X. SUN, P. ROSIN, R. MARTIN, AND F. LANGBEIN, *Fast and effective feature-preserving mesh denoising*, IEEE Trans. on Vis. and Comput. Graph., 13 (2007), pp. 925–938.

[96] X. SUN, P. ROSIN, R. MARTIN, AND F. LANGBEN, *Noise in 3d laser range scanner data*, in IEEE Int. Conf. on Shape Model. and App., June 2008, pp. 37–45.

[97] Y. SUN, S. SCHAEFER, AND W. WANG, *Denoising point sets via l0 minimization*, Computer Aided Geometric Design, 35-36 (2015), pp. 2 – 15.

[98] S. B. SYC, S. SAIDHA, S. D. NEWSOME, J. N. RATCHFORD, M. LEVY, E. FORD, C. M. CRAINICEANU, M. K. DURBIN, J. D. OAKLEY, S. A. MEYER, E. M. FROHMAN, AND P. A. CALABRESI, *Optical coherence tomography segmentation reveals ganglion cell layer pathology after optic neuritis*, Brain, 135 (2011), pp. 521–533.

[99] T. TASDIZEN, R. WHITAKER, P. BURCHARD, AND S. OSHER, *Geometric surface smoothing via anisotropic diffusion of normals*, in Proc. of the Conf. on Vis. '02, VIS '02, 2002, pp. 125–132.

[100] G. TAUBIN, *Estimating the tensor of curvature of a surface from a polyhedral approximation*, in Fifth Int. Conf. on Comput. Vision, Jun 1995, pp. 902–907.

[101] G. TAUBIN, *Linear anisotropic mesh filtering*, in IBM Research Report RC22213(W0110-051), IBM T.J. Watson Research, 2001.

[102] S. TICK, F. ROSSANT, I. GHORBEL, A. GAUDRIC, J.-A. SAHEL, P. CHAUMET-RIFFAUD, AND M. PAQUES, *Foveal shape and structure in a normal population*, Investig. Ophthalmol. and Vis. Sci., 52 (2011), pp. 5105 – 5110.

[103] C. TOMASI AND R. MANDUCHI, *Bilateral filtering for gray and color images*, in Proc. of the Sixth Int. Conf. on Comput. Vision, ICCV '98, Washington, DC, USA, 1998, IEEE Computer Society, pp. 839–846.

[104] S. TSUCHIE AND M. HIGASHI, *Surface mesh denoising with normal tensor framework*, Graph. Model., 74 (2012), pp. 130 – 139.

[105] J.-K. WANG, R. H. KARDON, M. J. KUPERSMITH, AND M. K. GARVIN, *Automated quantification of volumetric optic disc swelling in papilledema using spectral-domain optical coherence tomography*, Investigat. Ophthalmol. & Vis. Sci., 53 (2012), pp. 4069–4075.

[106] P.-S. WANG, Y. LIU, AND X. TONG, *Mesh denoising via cascaded normal regression*, ACM Transactions on Graphics (SIGGRAPH Asia), 35 (2016), pp. 1–232.

[107] R. WANG, Z. YANG, L. LIU, J. DENG, AND F. CHEN, *Decoupling noise and features via weighted l1-analysis compressed sensing*, ACM Trans. Graph., 33 (2014), pp. 18:1–18:12.

[108] M. WEI, J. YU, W. M. PANG, J. WANG, J. QIN, L. LIU, AND P. A. HENG, *Bi-normal filtering for mesh denoising*, IEEE Trans. Vis. Comput. Graph., 21 (2015), pp. 43–55.

[109] C. E. WILLOUGHBY, D. PONZIN, S. FERRARI, A. LOBO, K. LANDAU, AND Y. OMIDI, *Anatomy and physiology of the human eye: effects of mucopolysaccharidoses disease on structure and function – a review*, Clin. & Exp. Ophthalmol., 38 (2010), pp. 2–11.

[110] X. WU, J. ZHENG, Y. CAI, AND C.-W. FU, *Mesh denoising using extended rof model with l1 fidelity*, Comput. Graph. Forum, 34 (2015), pp. 35–45.

[111] L. XU, R. WANG, J. ZHANG, Z. YANG, J. DENG, F. CHEN, AND L. LIU, *Survey on sparsity in geometric modeling and processing*, Graph. Model., 82 (2015), pp. 160 – 180.

[112] S. K. YADAV, E. M. KADAS, S. MOTAMEDI, T. OBERWAHRENBROCK, F. C. OERTEL, K. POLTHIER, F. PAUL, AND A. U. BRANDT, *Cube: parametric modeling of 3d foveal shape using cubic bèzier*, Biomed. Opt. Express, 8 (2017), pp. 4181–4199.

[113] S. K. YADAV, S. MOTAMEDI, K. GAWLIK, K. POLTHIER, F. HAUSSER, F. PAUL, AND A. U. BRANDT, *Optic nerve head 3d shape analysis*, submitted to IEEE Trans. on med. imag., (2018).

[114] S. K. YADAV, U. REITEBUCH, AND K. POLTHIER, *Mesh denoising based on normal voting tensor and binary optimization*, IEEE Trans. Vis. Comput. Graph., 24 (2018), pp. 2366–2379.

[115] ——, *Robust and high fidelity mesh denoising*, IEEE Trans. on Vis. and Comput. Graph., PP (2018), pp. 1–1.

[116] S. K. YADAV, U. REITEBUCH, M. SKRODZKI, E. ZIMMERMANN, AND K. POLTHIER, *Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics*, Computers and Graphics, 74 (2018), pp. 234 – 243.

[117] H. YAGOU, Y. OHTAKE, AND A. BELYAEV, *Mesh smoothing via mean and median filtering applied to face normals*, in Proc. of Geom. Model. and Process., 2002, pp. 124–131.

[118] M. YOUNG, S. LEE, M. RATEB, M. F. BEG, M. V. SARUNIC, AND P. J. MACKENZIE, *Comparison of the clinical disc margin seen in stereo disc photographs with neural canal opening seen in optical coherence tomography images*, J. of Glaucoma, 23 (2014), pp. 360–367.

[119] H. Zhang, C. Wu, J. Zhang, and J. Deng, *Variational mesh denoising using total variation and piecewise constant function space*, IEEE Trans. Vis. Comput. Graph., 21 (2015), pp. 873–886.

[120] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, *Guided mesh normal filtering*, Comput. Graph. Forum, 34 (2015), pp. 23–34.

[121] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai, *Bilateral normal filtering for mesh denoising*, IEEE Trans. on Vis. and Comput. Graph., 17 (2011), pp. 1521–1530.

[122] Y. Zheng, G. Li, S. Wu, Y. Liu, and Y. Gao, *Guided point cloud denoising via sharp feature skeletons*, The Visual Computer, 33 (2017), pp. 857–867.

[123] H. Zimmermann, A. Freing, F. Kaufhold, G. Gaede, E. Bohn, M. Bock, T. Oberwahrenbrock, K.-L. Young, J. Dörr, J. T. Wuerfel, S. Schippling, F. Paul, and A. U. Brandt, *Optic neuritis interferes with optical coherence tomography and magnetic resonance imaging correlations*, Multiple Scler. J., 19 (2012), pp. 443–450.

[124] H. Zimmermann, T. Oberwahrenbrock, A. U. Brandt, F. Paul, and J.-M. Dörr, *Optical coherence tomography for retinal imaging in multiple sclerosis*, Degener. Neurol. and Neuromuscul. Dis., (2014), pp. 153–162.

Zusammenfassung

Im Rahmen dieser Arbeit ist es gelungen, ausgehend vom Krümmungsoperator, die Variation von Tangentenvektoren und die Beziehung zwischen klassischem Formoperator und Krümmungstensor einer triangulierten Fläche zu verstehen. In der ersten Hälfte folgt diese Arbeit dem Konzept der Variation der Flächennormalen, welche als Normal-Voting-Tensor (NVT) bezeichnet wird, und leitet eine Beziehung zwischen dieser und dem Formoperator ab. Damit wird das Konzept von Richtungskrümmungen und mittlerer Krümmung um die duale Representation einer triangulierten Fläche erweitert. Ein Normal-Voting-Tensor wird für jedes Dreieck einer Geometrie definiert und als elementbasierter Normal-Voting-Tensor (element-based NVT, ENVT) bezeichnet. Später wird aus dem ENVT ein Deformationstensor extrahiert, der die Anisotropie einer Fläche enthält. Basierend auf diesem Deformationstensor wird die mittlere Krümmung definiert.

Ein neu eingeführter Entrauschungs-Algorithmus für polyedrische Flächen benutzt den ENVT als Formoperator. Dabei wird eine binäre Optimierungstechnik auf den spektralen Komponenten des ENVT angewendet. Dieser Algorithmus erhält scharfe Kanten in der bearbeiteten Geometrie und hat eine deutlich erhöhte Konvergenzrate. Des Weiteren wird das Konzept des ENVT-basierten Flächen-Entrauschens zu einem Entrauschungsalgorithmus für Punktwolken erweitert.

Diese Arbeit untersucht auch ein „Robust Statistics Framework“ zum Filtern von Flächennormalen und führt eine robuste und hochqualitative Zwei-Schritt-Methode zum Entrauschen von polyedrischen Flächen ein. Diese Methode nutzt die Tukeys-bi-weight-Funktion als robusten Schätzer und stoppt dadurch die Diffusion an scharfen Kanten und erzeugt glatte umbilische Regionen. Ein Schema zum Vertex-Update wird eingeführt, das einen Laplace-Operator auf Basis differentieller Koordinaten zusammen mit einer Zwangsbedingung für Rechtwinkligkeit zwischen Kanten und Flächennormalen benutzt, um hochqualitative Flächen ohne falsch orientierte Normalen zu erzeugen. Dadurch wird der Algorithmus u.a. robuster gegen Rauschen mit starker Intensität.

In der zweiten Hälfte der Arbeit werden die vorgeschlagenen Geometrie-Prozessierungsalgorithmen auf Optische-Kohärenztomographie-Scandaten (Optical Coherence Tomography, OCT) angewendet, um die Form der menschlichen Retina/Netzhaut zu quantifizieren. Diese ist Teil des zentralen Nervensystems und weist eine ähnliche Zellzusammensetzung auf wie das Gehirn. Daher können viele neurologische Erkrankungen die Form der Retina beeinflussen. Neuroimmunologische Störungen verursachen bekanntermaßen unter anderem Veränderungen in zwei wichtigen Regionen der Retina: der Fovea und dem Sehnervenkopf (optic nerve head, ONH). Diese Arbeit enthält eine genaue und robuste Form-Modellierung dieser Regionen, führt eine parametrische Modellierung der Form der Fovea mittels kubischer Beziér-Kurven ein und leitet verschiedene 3D-Formparameter ab, die die Form der Fovea mit hoher Genauigkeit quantifizieren. Für den ONH wird ein 3D-Formanalyse-Algorithmus eingeführt, um die Formveränderung bei verschiedenen neurologischen Erkrankungen zu vermessen. Der vorgeschlagene Algorithmus nutzt triangulierte 2-Mannigfaltigkeiten für Oberflächen zweier retinaler Schichten zur Bestimmung mehrerer 3D-Formparameter. Durch das Erkennen dieser spezifischen Formveränderungen kann zukünftig die Diagnose und Verlaufsbeobachtung neurologischer Erkrankungen vereinfacht werden.