Feature detection on meshes

Geometric Modeling - Saarland University

Christen Millerdurai
Javier Usón Peirón
Marco Schichtel
February 10. 2022

Introduction

- Meshes are piecewise linear surfaces, finding continuous normals, curvatures is difficult.
- Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Meyer et al, a robust framework to approximate the discretized properties of shapes.
- Feature detection and extraction on meshes.
- Extracted features can be used to perform downstream tasks.

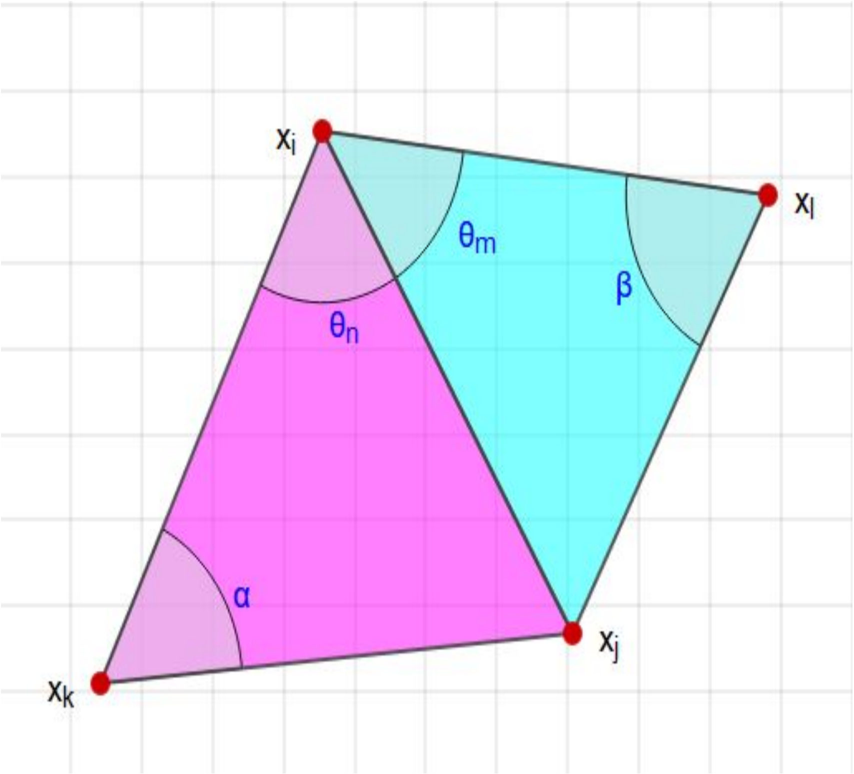
Features

- Gaussian Curvature k_G.
  - Allows to determine if shape is elliptic or hyperbolic, parabolic.

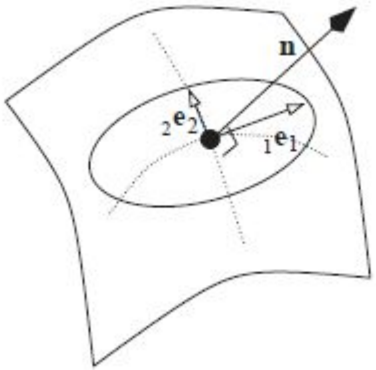
$$\kappa_G(\mathbf{x}_i) = (2\pi - \sum_{j=1}^{\#f} \theta_j) / \mathcal{A}_{\text{Mixed}}$$

- Mean Curvature k_H
  - Average of the normal curvatures

$$\mathbf{K}(\mathbf{x}_i) = \frac{1}{2\mathcal{A}_{\text{Mixed}}} \sum_{j \in N_1(i)} (cot\ \alpha_{ij} + cot\ \beta_{ij})\ (\mathbf{x}_i - \mathbf{x}_j)$$



Features



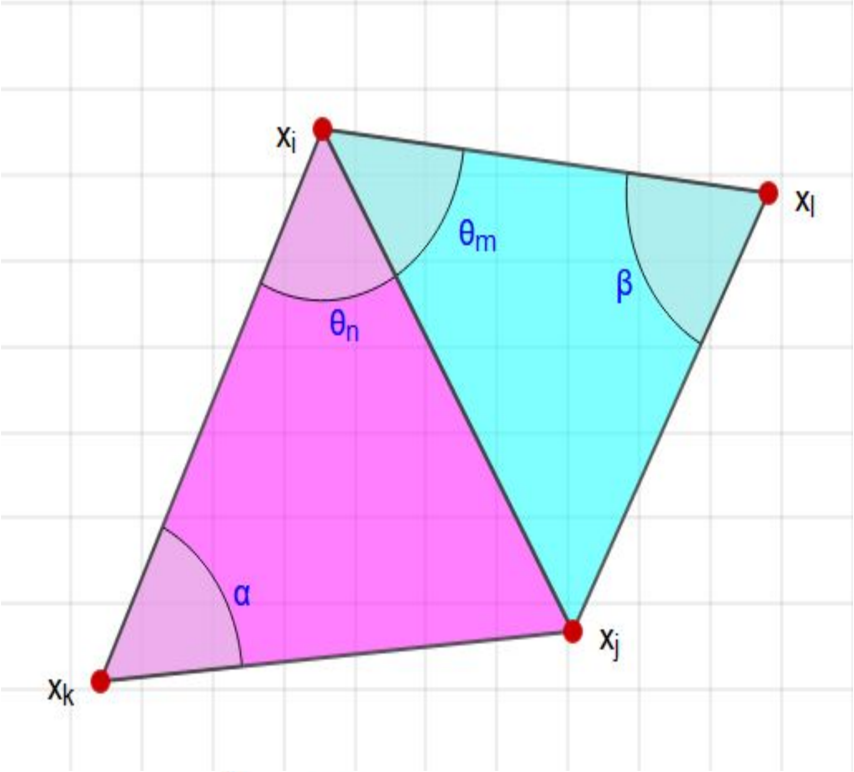
- Principal Curvature k_1 and k_2
  - Extremum values for normal curvature

$$k_1(x_i) = k_H(x_i) + \sqrt{k_H^2(x_i) - k_G} \qquad k_2(x_i) = k_H(x_i) - \sqrt{k_H^2(x_i) - k_G}$$

- Principal Directions e_1 and e_2
  - Directions for principal curvature
  - Least-Square minimization for curvature tensor B: $\sum_j w_j \left(\mathbf{d}_{i,j}^T B \, \mathbf{d}_{i,j} - \kappa_{i,j}^N \right)^2$
  - Eigenvectors for B are principal directions

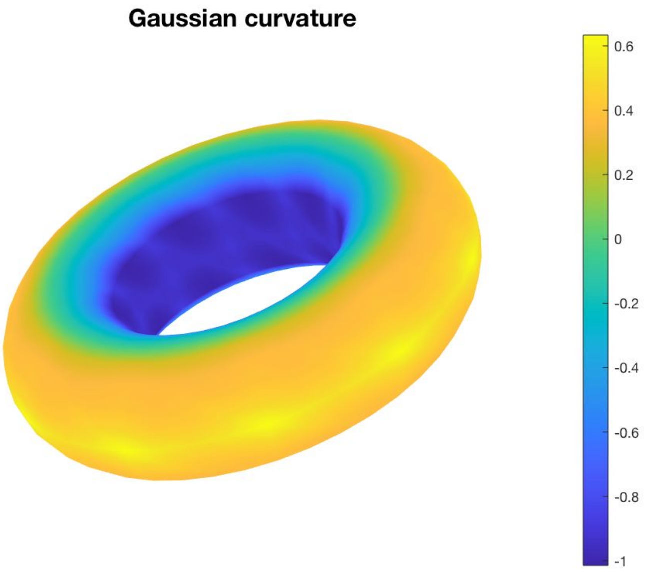
Implementation

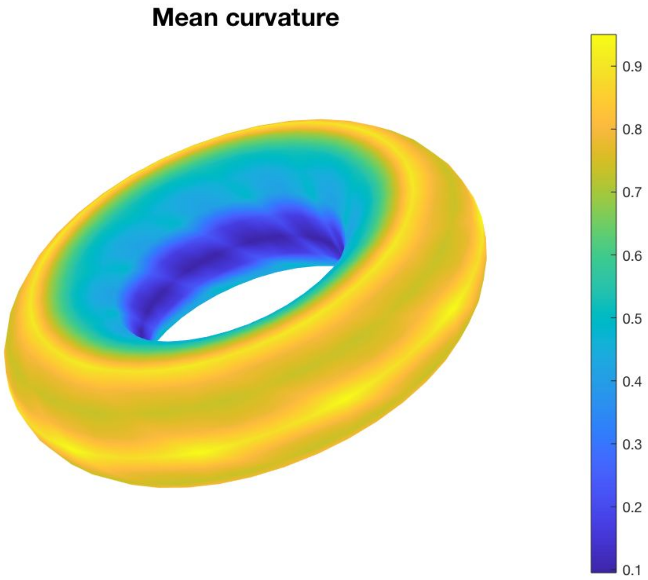
- Straight forward implementation of previous formulas.
- For Mean Curvature: sum by triangle not vertex.



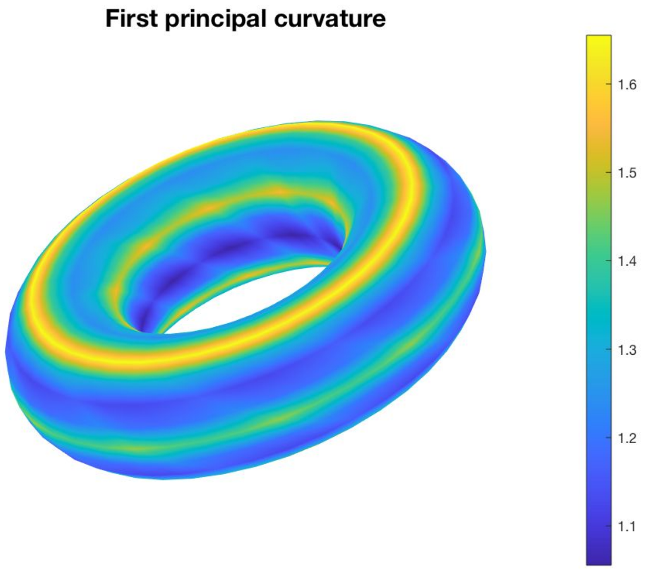
$$\mathbf{K}(\mathbf{x}_i) = \frac{1}{2\mathcal{A}_{\mathrm{Mixed}}} \sum_{j \in N_1(i)} (cot\ \alpha_{ij} + cot\ \beta_{ij})\ (\mathbf{x}_i - \mathbf{x}_j)$$

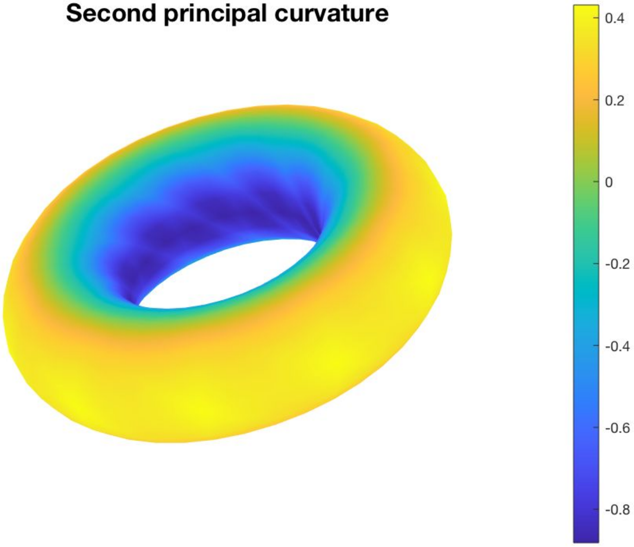
Let us test them!





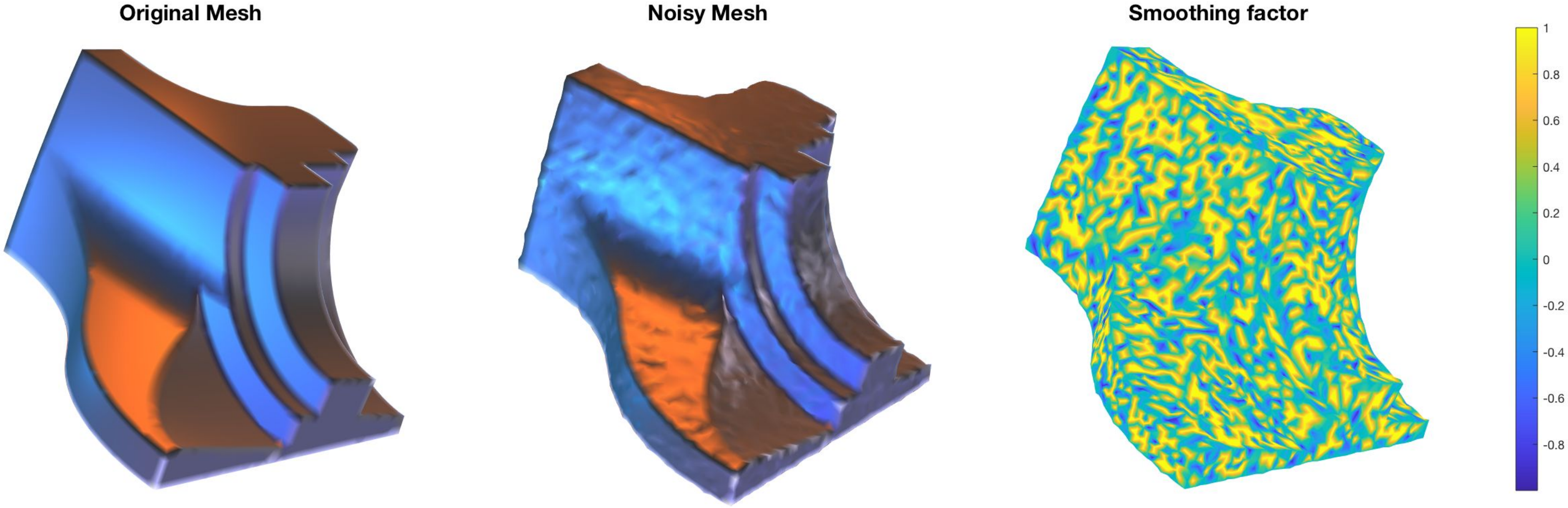
Let us test them!





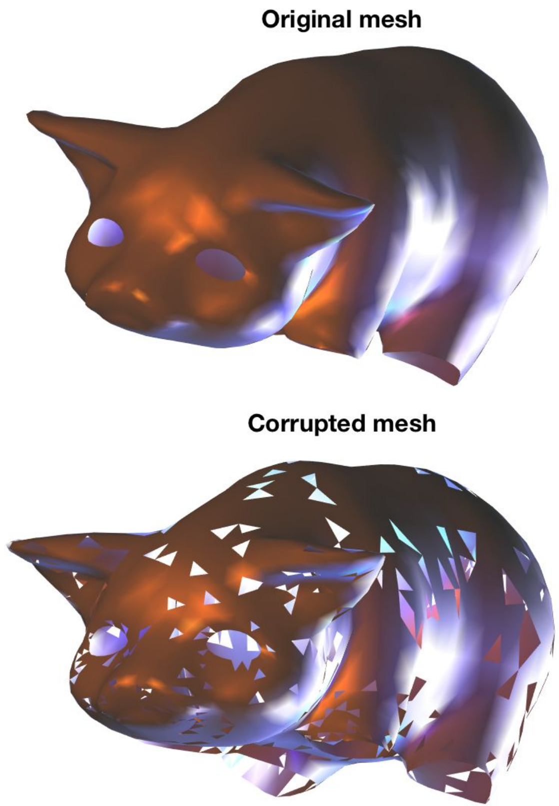
Applications - Noise detection

- Use curvatures for mesh smoothing (noise removal)



Applications - Classifier

- Something corrupted our meshes!
- We develop a classifier to trace them back
- Calculate the normalized histogram of principal curvatures for each mesh (It will act as a probability density function)
- Compare them using Wasserstein distance



Applications - Classifier

Original	Cube	Eightparam	Fandisk	Hand	Head	Mushroom	Pig	Pumpkin
Cube	0.0000	0.3725	0.0931	0.3611	0.0618	0.3418	0.2534	0.2977
Eightparam	0.3725	0.0000	0.2794	0.0114	0.4344	0.0307	0.1191	0.0748
Fandisk	0.0931	0.2794	0.0000	0.2679	0.1549	0.2486	0.1603	0.2045
Hand	0.3611	0.0114	0.2679	0.0000	0.4229	0.0193	0.1076	0.0634
Head	0.0618	0.4344	0.1549	0.4229	0.0000	0.4036	0.3152	0.3595
Mushroom	0.3418	0.0307	0.2486	0.0193	0.4036	0.0000	0.0883	0.0440
Pig	0.2534	0.1191	0.1603	0.1076	0.3152	0.0883	0.0000	0.0442
Pumpkin	0.2977	0.0748	0.2045	0.0634	0.3595	0.0440	0.0442	0.0000

Applications - Classifier

Corrupted	Cube	Eightparam	Fandisk	Hand	Head	Mushroom	Pig	Pumpkin
Cube	0.0625	0.3100	0.031	0.2987	0.1243	0.2793	0.1909	0.2352
Eightparam	0.3726	0.0019	0.2794	0.0114	0.4344	0.0307	0.1191	0.07486
Fandisk	0.1367	0.2359	0.0435	0.2244	0.1985	0.2051	0.1167	0.1610
Hand	0.3624	0.0101	0.2693	0.0013	0.4242	0.0206	0.1089	0.0647
Head	0.0652	0.4378	0.1583	0.4263	0.0033	0.4070	0.3186	0.3629
Mushroom	0.3440	0.0285	0.2508	0.0171	0.4058	0.0022	0.0905	0.0462
Pig	0.2716	0.1009	0.1784	0.0895	0.3334	0.0701	0.0181	0.0261
Pumpkin	0.2932	0.0793	0.200	0.0679	0.3550	0.0485	0.0397	0.0044

Conclusion

- The meshes are just discretizations of the surface (approximations).
- Principle, Gaussian, mean curvatures provide an quantitative way to describe the surface, unlike connectivity of the meshes.
- Hence, we have successfully acquired the features of the underlying surface and demonstrate classification using these features.

Future work

- Extending to higher order mesh structures. (Hexagonal or arbitrary number of vertices per face)
- Applications: Denoising, extension to n-d surface properties.

Thank you

Questions?