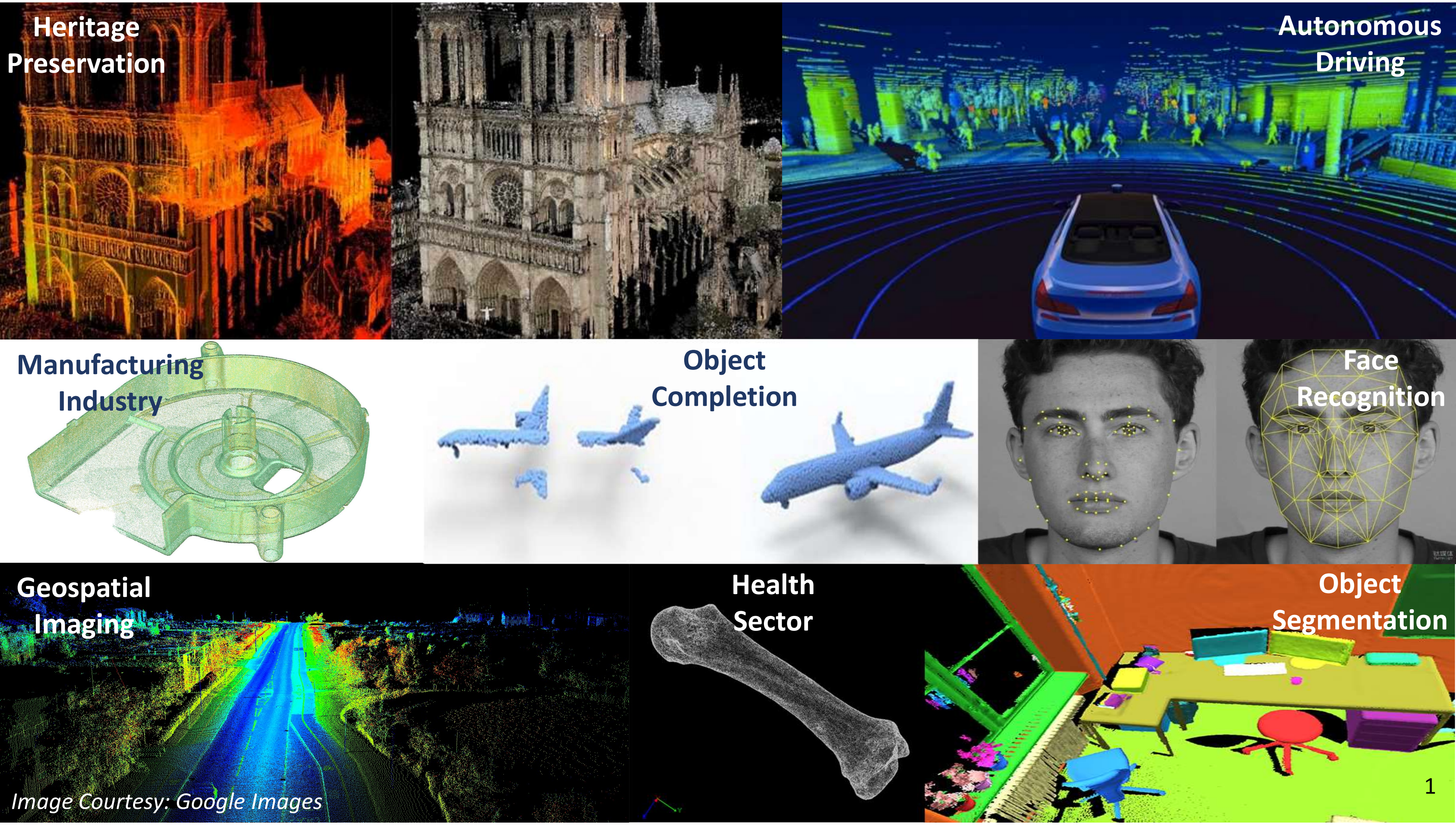
Understanding the Significance, Processing & Analysis of Point Clouds

Presenter:
Stuti Pathak
PhD Student

Universiteit Antwerpen


Heritage Preservation
Autonomous Driving
Manufacturing Industry
Object Completion
Face Recognition
Geospatial Imaging
Health Sector
Object Segmentation
Image Courtesy: Google Images

Structure Tensor-Based Geometric Features of Point Clouds

- For a point cloud P, if N_{p_i} is the set of k-nearest neighbours of a point $p_i \in \mathrm{P}$, then the Covariance Matrix(C) or Structure Tensor(S) is defined as:

$$C = S = \begin{bmatrix} p_{i_1} - p_i \\ p_{i_2} - p_i \\ . \\ . \\ p_{i_k} - p_i \end{bmatrix}^T . \begin{bmatrix} p_{i_1} - p_i \\ p_{i_2} - p_i \\ . \\ . \\ p_{i_k} - p_i \end{bmatrix}$$

where $p_{i_j} \in N_{p_i}$.

- Spherical or cylindrical neighbourhoods of fixed radius centered at point p_i are also used.
- Here, $\lambda_1 > \lambda_2 > \lambda_3$.

Geometric Feature	Expression	Significance
Linearity	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$	Describes how well the points constitute a line.
Planarity	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$	Describes how well the points constitute a plane.
Sphericity or Scattering	$\frac{\lambda_3}{\lambda_1}$	Describes how well the points constitute a sphere.
Omnivariance	$\sqrt[3]{\lambda_1 . \lambda_2 . \lambda_3}$	Describes the average point density in all directions.
Anisotropy	$\frac{\lambda_1 - \lambda_3}{\lambda_1}$	Describes whether the points are distributed in a specific direction, or if they are randomly distributed.
Change of Curvature or Surface Variation	$\frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$	Describes the degree of bending of a curved line or a plane and is the derivative of the curvature.

Table 1. Description of Geometric Features based on the eigenvalues of the Structure Tensor

Pauly et al., 2003.
West et al., 2004.

Point Cloud and Mesh Simplification

- Definition: Given a collection of 3D data points (or an initial triangular mesh), sample a subset of points (or find a new mesh with a smaller number of vertices) such that both the overall shape and the most salient features are preserved in the simplified point cloud (or mesh).

- Significance: Huge amount of redundant data produced by point cloud capturing devices makes the processing, transmission and storage of data extremely expensive.

- Most of the existing techniques either take meshes as input or triangulate the point cloud as a pre-processing step.

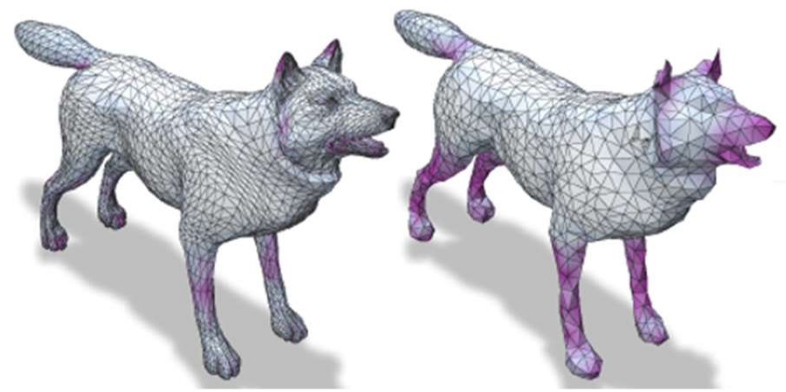


Figure 1. Mesh Simplification

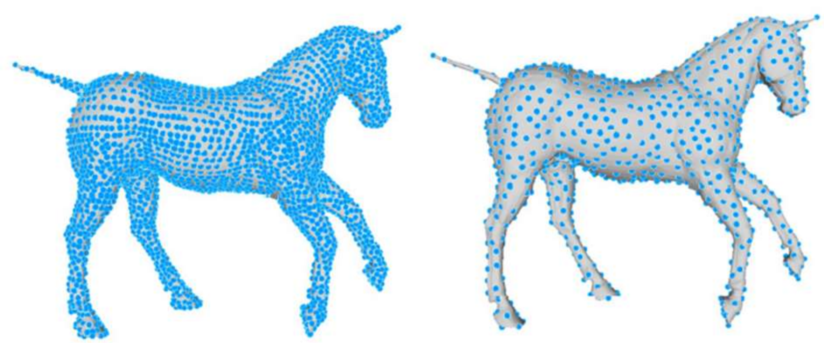


Figure 2. Point Cloud Simplification

Potamias et al., 2022.

Mesh Decimation

- Characterize all vertices.

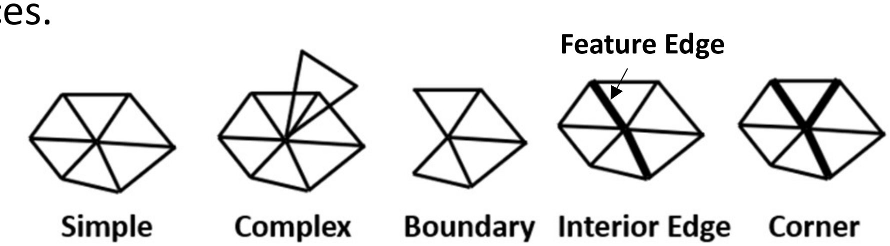


Figure 3. Vertex Classification

- Assign a distance criterion to each vertex type which determines the potential deletion candidacy.

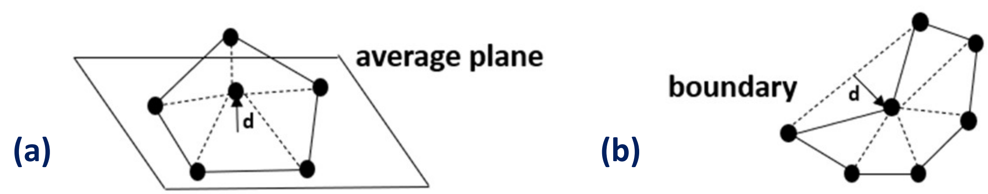


Figure 4. (a) Distance to face, (b) Distance to edge

- Multiple passes are made through the whole mesh. In each pass:
  - a vertex and its edges are removed.
  - the newly formed loop is re-triangulated.

Schroeder et al., 1992.

Energy Function Minimization

- Minimize the energy function E:

$$E = E_{dist} + E_{rep} + E_{spring}$$

where:

  - E_{dist} is the sum of squared distances from the points to the mesh.
  - E_{rep} is kept proportional to number of points (m).
  - E_{spring} places a spring on each edge of the mesh.

- $E_{dist}\uparrow$ and $E_{rep}\downarrow$ when $m\downarrow$. So both work together and penalize each other to minimize E.

- A minimum of $E_{dist} + E_{rep}$ may not exist which is solved by the introduction of E_{spring}.

Hoppe et al., 1993.

Coplanar Merging

- Group all nearly coplanar polygons.
- Create edge list.
- Remove redundant edges and keep only boundary edges.
- Reconstruct polygons.
- Re-triangulate to deal with holes.

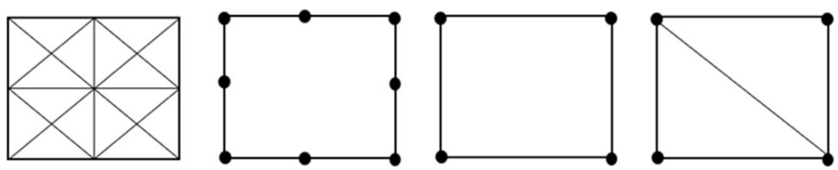


Figure 5. Overview of Simplification Process

Hinker et al., 1993.

Vertex Clustering

- Assign weights to all vertices based on their importance.
- Triangulate all the vertices.
- Divide the object into cells such that all vertices in the same cell form a cluster.
- Synthesis: Assign each cluster a single vertex which is defined by COM of all the weighted cluster vertices.
- Elimination: Triangles with three same representative vertices are replaced by points and ones with only two are replaced by an edge.

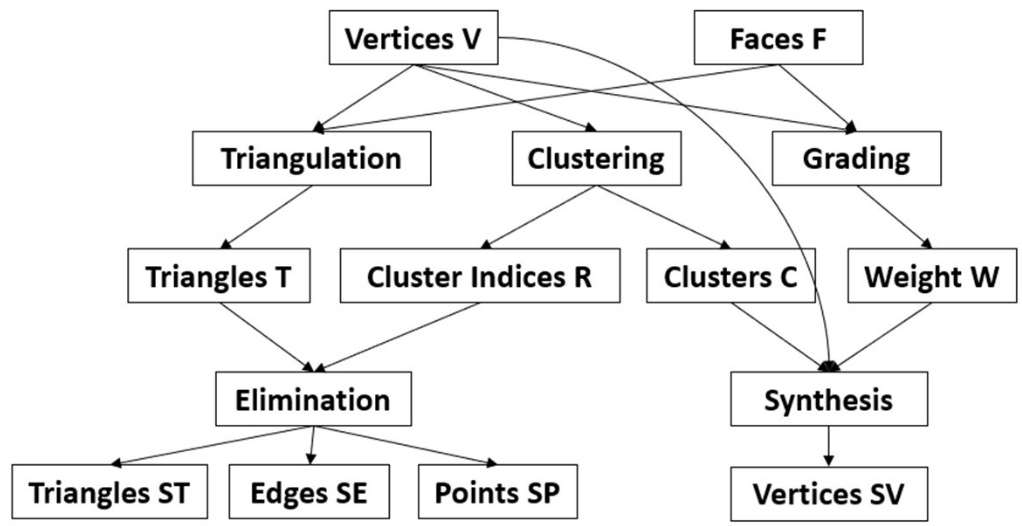


Figure 6. Overview of Simplification Process

Rossignac et al., 1993.

Re-Tiling

- Construct a triangular mesh from the original vertices.
- Choose a set of new candidate vertices that lie in the planes of the original mesh and may coincide with the original vertices.
- Move each point away from all nearby points by a relaxation method.
- Mutual tesselation: Create triangular meshes using both original and new vertices.
- One by one remove original vertices and triangulate to maintain the original topology.

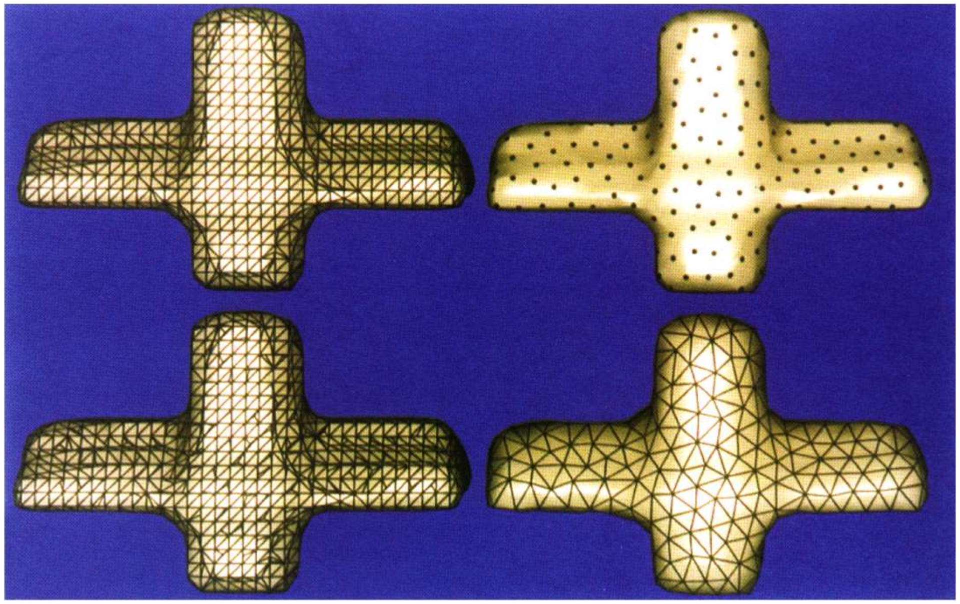


Figure 7. Overview of Simplification Process

Turk et al., 1992.

Graph Neural Network-Based Simplification

- A point-wise MLP projects all the points to a latent space.
- A GNN (optionally with mesh edges) captures the local geometric features.
- Use Farthest Point Sampling (FPS) to find cluster centers in latent space.
- Each cluster center is connected to all of its k-nearest neighbours by edges.
- A final layer updates the cluster center positions based on the neighbours to minimize visual perceptual error and preserve salient features.

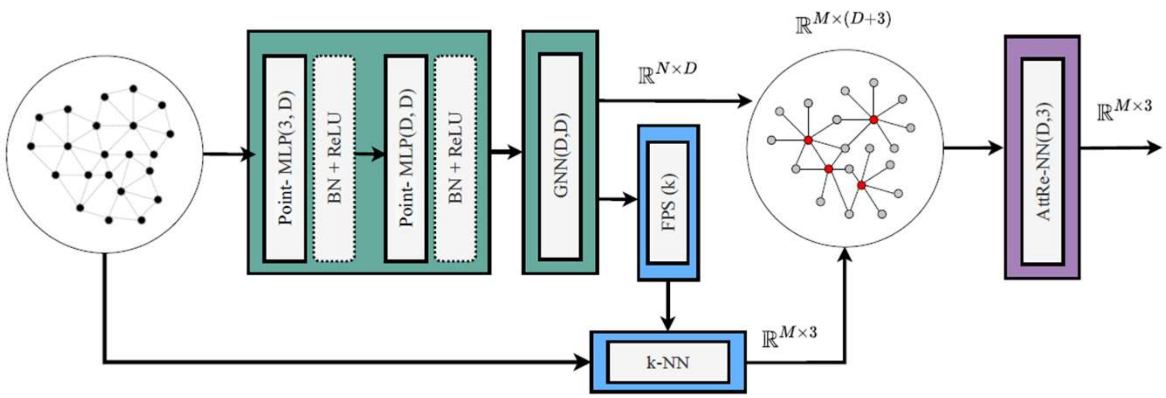


Figure 8. Overview of Simplification Process

Potamias et al., 2022.

Laplace-Beltrami Operator for Point Clouds

- For a point cloud P, if A_i is the Voronoi weight at point $p_i \in \mathrm{P}$, then the Laplace Operator(L) is defined as:

$$L[i][j] = \begin{cases} G_h(i,j)A_j & i \neq j \\ G_h(i,i)A_i - \sum_{j=1}^{n} [G_h(i,j)A_j] & \text{otherwise} \end{cases}$$

where $G_h(i,j) = \frac{1}{4\pi h^2} e^{-\frac{\left\|p_i - p_j\right\|^2}{4h}}$

- In weighted Voronoi diagram, weights are assigned to each point based on their importance. These weights determine the size of a point's Voronoi cell.
- Since L is not symmetric, it is decomposed into $L = GD$ where D is a diagonal matrix with $D[i][i] = A_i$ and $G[i][j] = G_h(i,j)A_j$ for $i \neq j$ and $G[i][j] = -\frac{1}{A_j}\sum_{j \neq i} G_h(i,j)A_j$ otherwise.
- Eigenvalues $\lambda_1, \dots, \lambda_n$ and eigenvectors $\phi_1, \dots, \phi_n$ of L are computed by solving generalized eigenvalue problem $G\rho_i = \beta_i D^{-1}\rho_i$ where $\lambda_i = \beta_i$ and $\phi_i = D^{-1}\rho_i$ as $GD(D^{-1}\rho_i) = G\rho_i = \beta_i(D^{-1}\rho_i)$.

Luo et al., 2009.

Role of Gaussian Processes

- Terrain Modelling: GPs can succesfully model large-scale terrains and preserve the spatial features efficiently.
- Object Segementation: GPs provide ehanced segmentation accuracy by reducing over-segmentation.
- Free Space Detection: GPs are used to fuse multimodal information (2D from camera and 3D data from Lidar) from autonomous vehicles to aid detection of space available to freely transverse.
- Scene Categorization: Multi-class GP classification have been used to categorize outdoor scenes with informative uncertainity estimates as an added value.
- Object tracking and Shape Detection: GPs can efficiently track the dynamic behaviour of objects represented by sparse point clouds along with estimating the position, orientation and shape of the object.
- Point cloud registration: GPs have been used to extract keypoints in multiple sets of point clouds for further aligning and merging into a globally consistent model.

References

[1] Pauly, M., Keiser, R. and Gross, M., 2003. “Multi-scale feature extraction on point-sampled surfaces.” Computer Graphics Forum 22(3), pp. 81–89.

[2] West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M. and Iverson, A. E., 2004. “Context-driven automated target detection in 3-D data.” Proceedings of SPIE 5426, pp. 133–143.

[3] Potamias, R. A., Bouritsas, G., and Zafeiriou, S., 2022. “Revisiting point cloud simplification: A learnable feature preserving approach.” European Conference on Computer Vision.

[4] Schroeder, W. J., Jonathan, A. Z., and William, E. L., 1992. “Decimation of triangle meshes.” Proceedings of the 19th annual conference on Computer graphics and interactive techniques.

[5] Hoppe, H., DeRose, T., Duchampy, T., McDonaldz, J. and Stuetzlez, W., 1993. “Mesh optimization.” Proceedings of the 20th annual conference on Computer graphics and interactive techniques.

[6] Hinker, P. and Hansen, C., 1993. “Geometric optimization.” Proceedings Visualization, pp. 189-195.

[7] Rossignac, J. R. and Borrel, P., 1993. “Multi-resolution 3D approximations for rendering complex scenes.” Modeling in computer graphics. Springer, Berlin, Heidelberg. 455-465.

[8] Turk, G., 1992. “Re-tiling polygonal surfaces.” Proceedings of the 19th annual conference on Computer graphics and interactive techniques.

[9] Luo, C., Safa, I. and Wang, Y., 2009. “Approximating gradients for meshes and point clouds via diffusion metric.” *Computer Graphics Forum*. Vol. 28. No. 5. Oxford, UK: Blackwell Publishing Ltd.

References

[10] Vasudevan, S., Ramos, F., Nettleton, E. and Durrant-Whyte, H., 2009. "Gaussian process modeling of large-scale terrain." Journal of Field Robotics, 26(10), 812-840.

[11] Shin, M.O., Oh, G.M., Kim, S.W. and Seo, S.W., 2017. "Real-time and accurate segmentation of 3-D point clouds based on Gaussian process regression." IEEE Transactions on Intelligent Transportation Systems, 18(12), pp.3363-3377.

[12] Xiao, Z., Dai, B., Li, H., Wu, T., Xu, X., Zeng, Y. and Chen, T., 2017. "Gaussian process regression-based robust free space detection for autonomous vehicle by 3-D point cloud and 2-D appearance information fusion." International Journal of Advanced Robotic Systems, 14(4), p.1729881417717058.

[13] Paul, R., Triebel, R., Rus, D. and Newman, P., 2012. "Semantic categorization of outdoor scenes with uncertainty estimates using multi-class Gaussian process classification." IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2404-2410).

[14] Kumru, M. and Özkan, E., 2018. "3D extended object tracking using recursive Gaussian processes." 21st International Conference on Information Fusion (FUSION) (pp. 1-8). IEEE.

[15] Hitchcox, T. and Forbes, J.R., 2020, October. "A point cloud registration pipeline using gaussian process regression for bathymetric slam." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4615-4622). IEEE.

Thank you!