On-Mesh Bilateral Filter

Bridging the Gap Between Texture and Object Space

Author: Mihnea Bernevig
Supervisors: Prof. Dr. Elmar Eisemann, Mathijs Molenaar
EEMCS, Delft University of Technology, The Netherlands

1. Introduction

The bilateral filter is an edge-preserving, non-linear, noise-reducing filter that considers **both the spatial distance and intensity differences between pixels**, enabling denoising while retaining sharp edges and details[1]. Figure 1 [2] shows an image blurred with a bilateral filter. Features like the design of the scarf are maintained through the blur process.

Traditional 2D image processing methods like bilateral filters often produce artifacts when applied to texture images because they disregard the 3D mesh structure, focusing solely on 2D spatial relationships and intensity differences. They take **discrete** texels (texture pixels) into account rather than the 3D spatial context of the mesh.

This paper introduces **On-Mesh Bilateral Filtering**, a method adapting the bilateral filter for non-contiguous texture mappings by **incorporating 3D spatial distances into a geometry-aware kernel**.

This approach ensures more accurate and context-sensitive smoothing, respecting both the mesh topology and texture properties, thereby enabling a more intuitive and direct workflow for artists and reducing preparation time for 3D models.

To implement the On-Mesh Bilateral Filter, we need to answer the following three questions:

1. **How can we determine the neighbors of a texel?**
2. **How can we calculate the spatial distance between a texel and its neighbor?**
3. **How can we calculate the intensity difference between a texel and its neighbor?**

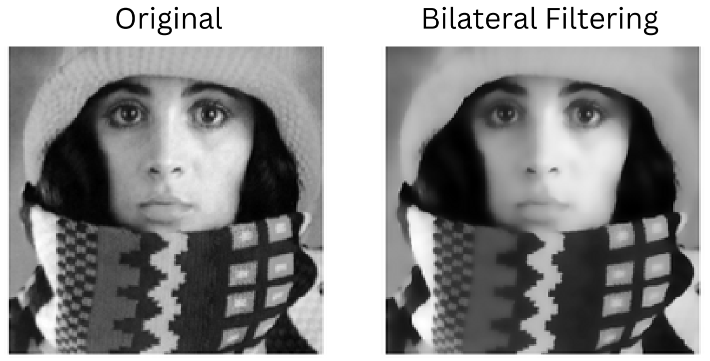

Figure 1: Edges on the scarf are not blurred out by the filter.

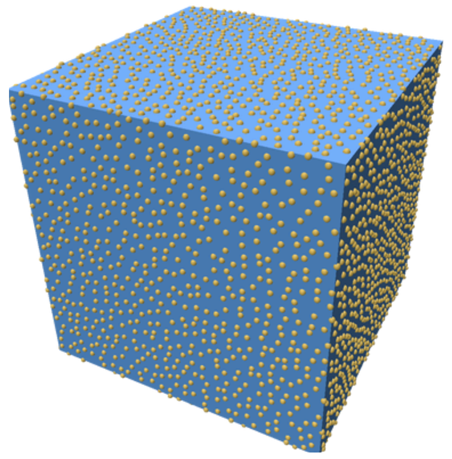

Figure 2: Fast Poisson Disk Sampling on a Cube Mesh.

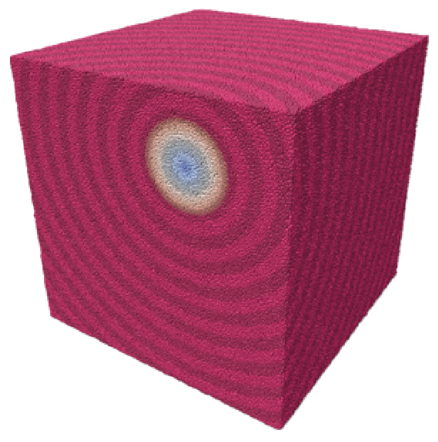

Figure 3: Visualization of geodesic distance on a point cloud. Points past 0.28 units are marked in red.

2. Methodology

Question 1 is answered by generating a large number of uniformly distributed sample points on the surface of the mesh. This is achieved using a Fast Poisson Disk Sampling algorithm [3], implemented in Geometry-Central [4]. The density of sample points should be at least equal to that of texels, such that each texel is sampled at least once. Figure 2 shows an example of a mesh sampled using Fast Poisson Disk sampling.

To answer **question 2**, a point cloud is generated from all sample points. The Point Cloud Heat Geodesic Distance algorithm [5], also implemented in Geometry-Central, can then be used to calculate the distance from a source point to all other points in the cloud. A visualization of the distance from a source point is shown in Figure 3.

As for **question 3**, UV coordinates for each of the sample points can be interpolated based on the mesh face they fall on, and thus the color of the texel they are sampling can be deduced.

The algorithm that was implemented can be summarised as follows:

- Sample the mesh with Fast Poisson Disk Sampling, interpolate the UV coordinates for each sample, and generate a Point Cloud from the sample points.
- For each texel that is mapped to the mesh, find its center in 3D space. From the Point Cloud, find the closest point to this 3D position.
    - From this point, calculate the geodesic distance to all other sample points.
    - Based on the geodesic distance and the intensity differences, apply the Bilateral Filter to obtain the filtered texel intensity.
- Combine all filtered texels to obtain the final texture image.

3. Results

Both the Gaussian and the Bilateral Blur filters were implemented as part of this research project. The implementations show effective noise removal capabilities, on par with those of their traditional 2D counterparts. A comparison between the filtering methods can be seen in Figure 4.

The sampling method also performs well on more complex mesh structures, such as the one in Figure 5. In this case, the chest body and lid are 2 different, unconnected structures. The geodesic distance image shows that distance is still properly calculated.

The filter also exhibits correct behaviour when filtering over meshes where faces don't have the same texel density. This can be attributed to the fact that the sampling method chosen maintains a consistent sample density over the entire surface of the mesh.

Figure 5: Gaussian Distance visualization(left); Normal texture (middle); Texture blurred with On-Mesh Bilateral Filter (right); Mesh and textures created by Onekro [6].

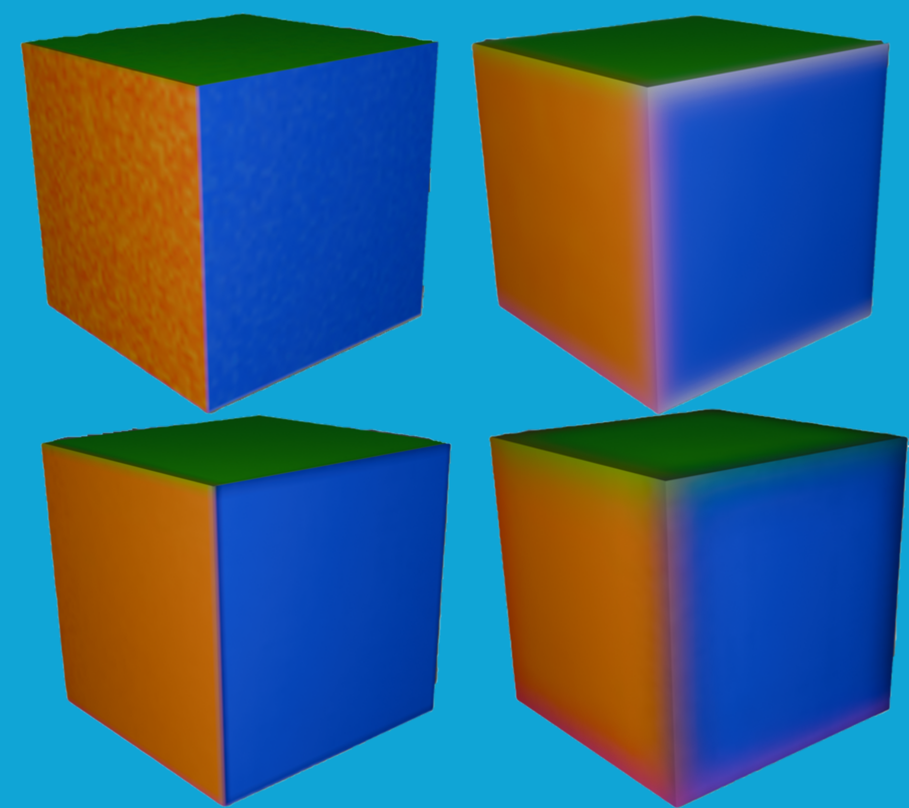

Figure 4: Noised texture on cube mesh (top left); Texture blurred with traditional Gaussian Filter (top right); Texture blurred with On-Mesh BilateralFilter (bottom left); Texture Blurred with On-Mesh Gaussian Filter (bottom right)

4. Performance

To ensure accurate filtering behavior, a large number of mesh samples is required. This leads to an increase in the time needed to calculate geodesic distances. Depending on the number of texels that need to be filtered, the complete filtering operation can take a very long time to finish. This makes the filter impractical to use for high resolution texture images.

5. Conclusion

This research explores the development and implementation of On-Mesh Bilateral Filtering, a novel approach that enhances texture filtering on 3D models. Despite challenges in optimizing performance for high-resolution textures, this method lays a solid foundation for future advancements in digital graphics.

6. Further research

Future research should focus on optimizing the performance of the On-Mesh Bilateral Filter, particularly by reducing the number of samples needed for geodesic distance calculations. Approaches such as calculating geodesic distances for a subset of samples and interpolating distances or creating smaller point clouds for each mesh face could improve efficiency. Additionally, developing more efficient sampling methods and leveraging computational advancements could further enhance performance, especially for high-resolution textures.

[1] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.
[2] R. van den Boomgaard, "Bilateral filter," University of Amsterdam. [Online]. Available: https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/LectureNotes/IP/LocalOperators/bilateralfilter.html.
[3] R. Bridson, “Fast Poisson disk sampling in arbitrary dimensions,” *University of British Columbia*, 8 2007. [Online]. Available: https://doi.org/10.1145/1278780. 1278807
[4] N. Sharp, K. Crane et al., “Geometrycentral: A modern c++ library of data structures and algorithms for geometry processing,” https://geometry-central.net/, 2019.
5] K. Crane, C. Weischedel, and M. Wardetzky, “The heat method for distance computation,” *Commun. ACM*, 7 vol. 60, no. 11, pp. 90–99, Oct. 2017. [Online]. Available: http://doi.acm.org/10.1145/3131280
[6] Onekro, “Chest pixelart,” 2021, licensed under Creative Commons Attribution-NonCommercial (http://creativecommons.org/licenses/by-nc/4.0/). [Online]. Available: https://skfb.ly/onK9F

Author contact: m.bernevig@student.tudelft.nl
You can try the filter discussed in this project at: https://github.com/MBernevig/On-Mesh-Bilateral-Filter