Fast Edge-based geodesic Poisson Disk Remeshing

Tom Uhlmann, Libor Váša and Guido Brunnett

Chemnitz University of Technology, Germany

Abstract

Triangular meshes of high complexity are common when created by a 3D scanner device and must be reduced for further processing. The geodesic Poisson disk remeshing [FZ08] is a method that generates a simplified mesh with highly regular triangles at the cost of exorbitant computation time. In this paper we will outline a new approach to this technique that makes it applicable for highly complex models. Our approach operates directly on the surface of the mesh, therefore works for meshes of arbitrary topology. Meshes consisting of millions of triangles can be reduced to an arbitrary complexity in just a few minutes while the original approach processes meshes with thousands of triangles in the same time. Our easy to implement remeshing technique also provides several options to preserve features.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Today's industry and research departments use 3D models for the development of new products, physical simulations, the creation of virtual worlds and various other applications. Each of these branches has individual requirements towards the models. While software that draws objects requires them to have few elements in order to render many objects in real-time, physical simulations demand meshes that consist of regular triangles to guarantee the stability of the applied algorithms. Remeshing techniques are used to fit certain properties of a mesh to the specific purpose.

The focus of our remeshing technique is on the simplification of meshes with high triangle count. This is especially important for meshes that are generated from 3D scanner data.

2. Previous Work

A lot of research has already been done in the field of remeshing. Due to the length restrictions we refer the reader to comprehensive state-of-the-art papers. Alliez et al. [AUGA08] described, categorized, and summarized techniques developed prior to 2008. By their categorization scheme our technique is a high quality one with elements of feature remeshing. We also work directly on the mesh and therefore do not need a global parametrization. Another survey was done recently by Payan et al. [PRS14] which focused on semi-regular meshing. In contrast to these techniques we do not build a base mesh nor use a parametrization.

One particular algorithm that has mainly inspired our technique is the direct geodesic Poisson disk sampling algorithm [FZ08] by Fu and Zhou. Their technique generates meshes with highly regular triangles, but their algorithm is computationally expensive. While we adopted the general idea from their technique, we use a completely different computation scheme, hence sacrificing a little on the quality side but are able to process meshes with millions of triangles in short time.

3. Fast Edge-based geodesic Poisson Disk Remeshing

The basic idea of the algorithm is to distribute a number of sampling vertices isotropically over the surface of the original mesh. To create this isotropic distribution, a geodesic disk of constant radius is assigned to each sampling vertex. New sampling vertices can be created on the border of the set union of disks only, guaranteeing a minimal spacing between all sampling vertices. Edge splits are used during the sampling phase to merge each new sampling vertex with the original mesh. The final mesh is obtained by removing all original vertices from the mutually tessellated mesh. There-

DOI: 10.2312/egsh.20151000

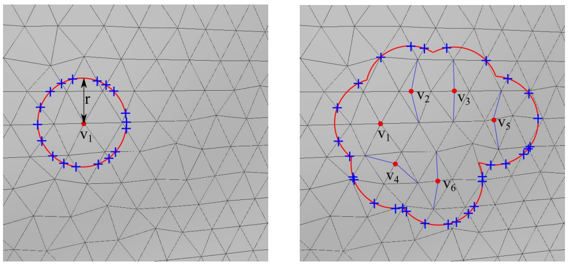

Figure 1: *Two steps of our sampling algorithm. The left side shows the first sampling vertex v_1 and its associated geodesic disk of radius r colored in red. The blue crosses indicate the intersection points of the disk with the edges of the mesh. These are the possible positions for the next sampling vertex. On the right side: The sampling algorithm after the 5 sampling vertices v_2 through v_6 have been added. The red line is the border of the set union of geodesic disks of all 6 sampling vertices. Note that this is only an illustration, our technique does not compute the whole border. The blue crosses are again the possible position for the next sampling vertex. The blue edges have been added by the edge split operations.*

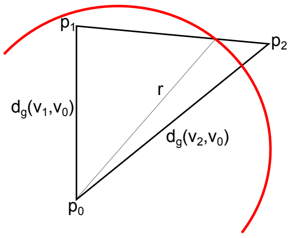

Figure 2: *Construction of the triangle to find the next sampling vertex. p_0 is the origin of the local coordinate system and the red partial circle is the border of the geodesic disk intersecting the border edge. v_0 is the closest sampling vertex to the border edge (v_1, v_2).*

fore we follow the concept in [FZ08]. A vertex removal operation is applied on each original vertex and the resulting hole is patched by a local parametrization and Delaunay triangulation.

The input of our algorithm is the triangle mesh $\mathcal{M}$ and the vertex count ϑ the final mesh should have. The following steps are performed:

1. Determine disk radius r with ϑ and surface area $A_{\mathcal{M}}$ using equation (1)
2. Select a random start vertex v and assign geodesic distance 0
3. Compute all geodesic distances up to distance r and insert border edges into cache C
4. if $C = \emptyset$ Exit
5. Select an edge from C and calculate the new sampling vertex position
6. Create a new sampling vertex and fuse it into the mesh by applying an edge split operation
7. Assign geodesic distance 0 to new sampling point
8. Continue with step 3

Lagae and Dutré [LD06] described the relation of the disk radius and the number of samples on the unit square. Tailored to our algorithm the disk radius for a mesh with surface area $A_{\mathcal{M}}$ can be calculated using the equation:

$$r = 2 \cdot \sqrt{A_M} \cdot \sqrt{\frac{1}{2 \cdot \sqrt{3} \cdot \vartheta}} \cdot 0.844. \tag{1}$$

The constant value of 0.844 is an algorithm specific constant established in [FZ08] and works for our implementation as well.

The algorithm successively adds new sampling vertices and includes them into the mesh by the edge split operation. Each sampling vertex has an associated disk and the possible sampling vertex positions in each step are the points, where the border of the set union of disks intersects the edges of the mesh. To find these border edges (v_1, v_2) efficiently the condition

$$d_g(v_1) \le r \quad \& \quad d_g(v_2) \ge r, \tag{2}$$

where $d_g(v)$ is the geodesic distance of a vertex v, is evaluated during the calculation of the geodesic distances for each processed edge. If this condition is met, a point exists on the edge where the border of the set union of disks intersects this edge. Once a border edge has been found, it is stored in the cache. We do not need to compute the exact border using this technique, hence saving a considerable amount of calculations.

This is the major difference to the technique of Fu and Zhou [FZ08]. They compute the geodesic disk explicitly by computing the equidistant geodesic curve of radius r for each sampling vertex. They have to merge the equidistant curve of the new sampling vertex with the old border in each step to obtain the current border. Our approach is faster to compute and much easier to implement.

To obtain the actual position of the next sampling vertex v_s we construct a triangle (p_0, p_1, p_2) with $p_0, p_1, p_2 \in \mathbb{R}^2$ from the border edge (v_1, v_2) as illustrated in figure 2. Then we solve the equation:

$$\|p_1 + t \cdot (p_2 - p_1)\| = r. \tag{3}$$

Once the quadratic equation is solved, the valid solution t is used as interpolation parameter on the corresponding border edge (v_1, v_2) to obtain v_s.

After the sampling phase is finished the mesh consists of the original vertices, the new sampling vertices, and their

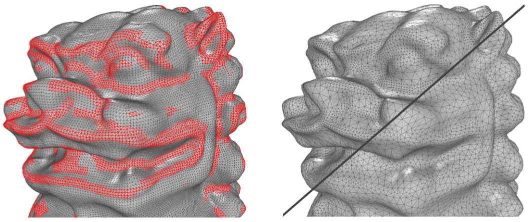
Feature detection
Full & adaptive sampling

Figure 3: *Recognized features (left image) can be preserved entirely (top slice) or up to a specific amount (bottom slice).*

corresponding mesh elements. We remove the original vertices by successively applying vertex removal operations on them. The resulting hole is locally parametrized and closed using constrained Delaunay triangulation [FZ08]. Finally we apply post processing steps to enhance the quality of the triangles further. Namely we apply a user defined amount of edge flips and second order umbrella operator [KCVS98] iterations. This increases the regularity of the triangles while introducing only a small additional approximation error, as long as the iteration count is kept low.

3.1. Preserving Features

The preservation of features, such as sharp or smooth edges, is essential for the appearance of an object. To preserve the features we adapted the basic concepts from [FZ08] but implemented them in a different way, so that they work with our approach. There are two options for feature preservation that we have elaborated. The first one is an additional sampling of the feature regions, that we will refer to as the manual feature preservation, because the feature regions are user inputs. The second option is a density function. This method will be referred to as the automatic feature preservation. To recognize features we use the uniformly supported second-order difference [Wan06] with user determined threshold values.

3.1.1. Manual Feature Preservation

To preserve determined features it is possible to keep all feature vertices in the region as sampling vertices. This can, however, result in a completely different tessellation within this region in comparison to the rest of the mesh. Therefore we do a presampling of the feature region with the described sampling, but we scale the disk according to a specific parameter. This way additional vertices are created in the feature region but the tessellation fits more adequately into the new mesh. Figure 3 shows an example result of the manual feature preservation.

3.1.2. Automatic Feature Preservation

To preserve features automatically during the sampling we use the density function:

$$d_{gs}(v) = d_g(v) \cdot (1 + \lambda \cdot H). \quad (4)$$

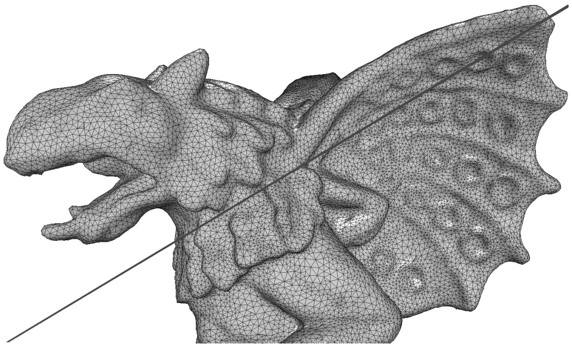

Figure 4: *Additional mesh elements are created in regions with high curvature due to the automatic feature preservation.*

It expands the geodesic distances according to the mean curvature H of the respective vertex and a scaling factor λ. Note that stretching geodesic distances is equivalent to locally contracting the radius of the disk. The larger λ is chosen, the stronger the curvature will affect the density of the created samples. More samples are placed in regions with high curvature this way, resulting in an improved feature preservation within these areas. The impact of the automatic feature preservation is illustrated in figure 4.

The density function also affects the computation of r. If the formula is not adapted, more vertices than specified by ϑ are created. We determine the average mean curvature of the surface and weave it into the formula for computing r to compensate for the stretching of the distances. The updated equation for calculating r is:

$$r = 2 \cdot \sqrt{A_M} \cdot \sqrt{\frac{1}{2 \cdot \sqrt{3} \cdot \vartheta}} \cdot 0.844 \cdot \left(1 + \sum_{i=i}^{|V|} \frac{\lambda \cdot H(v_i)}{|V|}\right). \quad (5)$$

4. Results and Discussion

We tested our technique with a variety of models on a PC with an Intel Xeon X5 at 3.60 GHz CPU and 32GB of RAM. We reduced each model by 90% of its complexity and compared our technique to the simplification algorithm of MeshLab [CR*ay]. Table 1 lists the results. The computation time for our algorithm depends mainly on the size of the input model. The size of the output model has barely an effect. In comparison to MeshLab's Quadric Edge Collapse decimation algorithm we are slightly slower and can not compete in terms of approximation error, but our technique produces triangles with significantly higher quality. Our current implementation is single threaded. We believe that a parallel version would outperform MeshLabs algorithm.

In comparison to the results listed by the authors of the original Poisson disk remeshing, we can process a mesh with 8 million faces in the same time they process one with 8

Model	$\|F_o\|/\|F_r\|$	$Time_s$	E_{RMS}	AR(Ø)
Angel	614k/60k	9.46	0.0890	0.824
		6.15	0.0184	0.656
Face	2.63M/243k	39.17	0.0652	0.835
		25.59	0.0133	0.659
Gargoyle	7.95M/787k	124.58	0.0098	0.828
		83.27	0.0021	0.685

Table 1: *Results of simplified meshes created with our technique and by Quadric Edge Collapse. This table lists, in order from left to right, the models name, the face count of the original mesh and the simplified one, the processing time in seconds, the RMS approximation error measured by the Metro tool [CRS96], and the average aspect ratio [JCLL10] of the triangles. The top row shows our algorithms results and the bottom row MeshLabs results for each model, respectively.*

thousand, yet their aspect ratio is approximately 0.1 higher than ours in average.

We encountered problems with our technique when creating meshes of equal or higher complexity than the input mesh. New samples cannot be created inside the triangles because of the edge based approach. We worked around this problem by applying edge splits on the mesh until a sufficient complexity has been achieved. In general we found that the input mesh should have double the number of vertices of the output mesh to get good results. A similar problem arises in regions with high grading. If a mesh contains stray triangles that are much larger in size then the others, these triangles may not be remeshed appropriately.

5. Conclusion and Future Work

We proposed a new approach to the geodesic Poisson disk sampling method for direct remeshing of triangulated surfaces. Our approach approximates calculations through the use of the existing edges resulting in a very fast algorithm.

The strong side of our algorithm is especially the downsizing of large scale models. While the processing speed is very high, the introduced approximation error is still low. In comparison with specialized simplification algorithms, our approach achieves results of considerably higher regularity. The outlined feature preservation methods support user-specified as well as automated preservation of features.

For future work we like to verify how recent approaches to the computation of geodesic distances fit into our technique and may help to improve the quality and accuracy of the resulting mesh. The geodesics in heat [CWW13] by Crane et al. and the saddle vertex graph technique [YWH13] by Ying et al. seem to be promising approaches. Another possible improvement could be a sophisticated placement spot for the next sampling vertex, because the random selection does not lead to optimal results. Using the intersection point of two disks would improve the isotropy due to the optimal utilization of the available space. It must, however, first be determined how this point can be found efficiently.

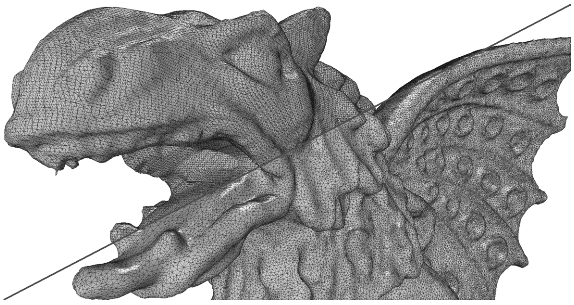

Figure 5: *Gargoyle original and remeshed version created with our technique.*

References

[AUGA08] Alliez P., Ucelli G., Gotsman C., Attene M.: Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring, Mathematics and Visualization* (2008), Springer. 1

[CR*ay] Cignoni P., Ranzuglia G., et al.: Meshlab. http://meshlab.sourceforge.net, 2005–today. Visual Computing Lab - ISTI - CNR. 3

[CRS96] Cignoni P., Rocchini C., Scopigno R.: *Metro: Measuring Error on Simplified Surfaces*. Tech. rep., Istituto per l'Elaborazione dell'Informaizione - Consiglio Nazionale delle Recerche, Paris, France, France, 1996. 4

[CWW13] Crane K., Weischedel C., Wardetzky M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5 (Oct. 2013), 152:1–152:11. 4

[FZ08] Fu Y., Zhou B.: Direct sampling on surfaces for high quality remeshing. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 115–124. 1, 2, 3

[JCLL10] Jong B.-S., Chiang C.-H., Lee P.-F., Lin T.-W.: High quality surface remeshing with equilateral triangle grid. *The Visual Computer 26*, 2 (2010), 121–136. 4

[KCVS98] Kobbelt L., Campagna S., Vorsatz J., Seidel H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 105–114. 3

[LD06] Lagae A., Dutré P.: An alternative for wang tiles: colored edges versus colored corners. *ACM Trans. Graph. 25*, 4 (Oct. 2006), 1442–1459. 2

[PRS14] Payan F., Roudet C., Sauvage B.: Semi-regular triangle remeshing: A comprehensive study. *Computer Graphics Forum* (2014), n/a–n/a. 1

[Wan06] Wang C. C. L.: Incremental reconstruction of sharp edges on mesh surfaces. *Comput. Aided Des. 38*, 6 (2006), 689–702. 3

[YWH13] Ying X., Wang X., He Y.: Saddle vertex graph (svg): A novel solution to the discrete geodesic problem. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 170:1–170:12. 4