

Obtuse Triangle Elimination for Isotropic Remeshing

Yiqun Wang
NLPR, CASIA and Univ. of CAS
#95 East Zhongguancun Road
Beijing 100190
wangyiqun2016@ia.ac.cn

Chengcheng Tang
Stanford University
450 Serra Mall
Stanford, California 94305
tangcc@stanford.edu

Dong-Ming Yan
NLPR, CASIA
#95 East Zhongguancun Road
Beijing 100190
yandongming@gmail.com

Xiaohan Liu
NLPR, CASIA and Univ. of CAS
#95 East Zhongguancun Road
Beijing 100190
liuxiaohan2017@ia.ac.cn

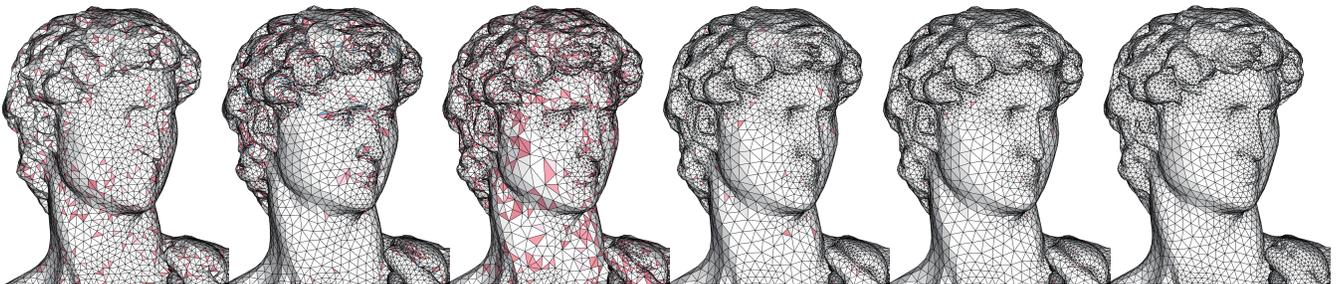


Figure 1: A comparison of different remeshing approaches. The input David’s head model (60k vertices and 120k triangles) is remeshed with approximately 60k faces with each method. From left to right are results of MPS [Yan and Wonka 2013] ($\#_{obt} = 7k$), RAR [Dunyach et al. 2013] ($\#_{obt} = 7.2k$), MAI [Hu et al. 2017] ($\#_{obt} = 15.6k$), CVT [Yan et al. 2009] ($\#_{obt} = 1.4k$), NOB [Yan and Wonka 2016] ($\#_{obt} = 724$), and ours ($\#_{obt} = 0$). Obtuse triangles are colored in pink. Here, $\#_{obt}$ indicates the number of obtuse triangles after remeshing.

ABSTRACT

In this poster, we propose a simple yet efficient algorithm for eliminating obtuse triangles for isotropic remeshing. Our method can either be applied directly on the input mesh, or be used as a post-processing step on the results from other remeshing algorithms. Our approach outperforms the state-of-the-art approaches in terms of the mesh quality.

CCS CONCEPTS

• **Computing methodologies** → **Mesh geometry models**;

KEYWORDS

Remeshing; Obtuse triangle; Local operation

ACM Reference format:

Yiqun Wang, Dong-Ming Yan, Chengcheng Tang, and Xiaohan Liu. 2017. Obtuse Triangle Elimination for Isotropic Remeshing. In *Proceedings of SIGGRAPH ’17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017*, 2 pages. <https://doi.org/10.1145/3102163.3102179>

1 INTRODUCTION

Triangle meshes are ubiquitous representations of 3D shapes in computer graphics, which can be acquired easily nowadays using 3D scanning devices or motion capture techniques. Newly obtained raw meshes always contain a large number of triangles, often with noises, which are difficult to be used directly in the downstream scenarios, such as modeling and simulation. A remeshing process is often necessary to improve the mesh quality for various applications.

Large varieties of algorithms have been proposed to address the remeshing problem. From an algorithmic point of view, there are two main common building blocks in various algorithms, i.e., resampling vertices and rebuilding the connectivity. The most successful remeshing approaches are those based on *Centroidal Voronoi Tesselation* (CVT), which have both an elegant theoretical background and a family of robust implementations [Alliez et al. 2008]. However, computing CVTs heavily relies on global optimization algorithms.

*D.-M. Yan is the corresponding author.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGGRAPH ’17 Posters, July 30 - August 03, 2017, Los Angeles, CA, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5015-0/17/07...\$15.00
<https://doi.org/10.1145/3102163.3102179>

When the number of desired points increases, the performance decreases drastically. Recent works of blue-noise sampling have been introduced for the purpose of remeshing [Yan and Wonka 2013]. But the resulting mesh quality has been lowered down due to the randomness nature of the point distribution. The most related work with ours is the discrete optimization based approaches, where only local mesh operations are applied. For example, the work of [Dunyach et al. 2013] achieves real-time adaptive remeshing, which can be used in animation applications. Later, Hu et al. [2017] further improve the minimal angle bounds under the same framework, at the cost of efficiency. All the above mentioned approaches cannot simultaneously guarantee the bounds of the minimal and the maximal angles in the output mesh.

In our approach, we use a similar framework as that of [Dunyach et al. 2013], where vertex insertion, mesh smoothing, vertex removal, and valence optimization are repeatedly applied. However, our vertex insertion and deletion strategies are completely different from previous works, which result in better remeshing quality than state-of-the-art approaches.

2 OUR APPROACH

We present a simple approach to eliminate obtuse triangles for isotropic remeshing. Given an input mesh surface, \mathcal{M} , our goal is to approximate \mathcal{M} with a new mesh, \mathcal{M}' , whose angles are bounded between a minimal and a maximal value, i.e., each angle, $\theta_j \subset t_i, (j = 1, 2, 3)$, is bounded, $\theta_j \in (\theta_{min}, \theta_{max})$. Here, t_i is a triangle of the new mesh, \mathcal{M}' . As an important special case, when the maximal value, $\theta_{max} = 90^\circ$, we achieve a non-obtuse remeshing of \mathcal{M} . In our experiments, we set $\theta_{min} = 30^\circ$ and $\theta_{max} = 90^\circ$ by default.

The proposed framework consists of four main components, i.e., 1. Initialization. 2. Obtuse triangle removal. 3. Minimal angle improvement. 4. Valence optimization. We briefly explain each step in the following.

- (1) **Initialization.** The output mesh \mathcal{M}' is first initialized by resampling the input mesh with a user specified number of points. Then, we perform 3 to 5 steps of Lloyd iteration [Lloyd 1982] to get a good initial mesh. Note that this step is optional, our algorithm can also start from an input mesh directly.
- (2) **Obtuse triangle removal.** All the obtuse triangles of \mathcal{M}' are collected in a list. We select the first k triangles in the list to process. For each selected obtuse triangle, t , we first split its longest edge, followed by an edge flip to increase the valence of the newly inserted vertex. Once the first k obtuse triangles are processed, we perform a local tangential smoothing to further improve the mesh quality (5 iterations in our implementation).
- (3) **Minimal angle improvement.** Next, we optimize small angles. Similar to the process above, we first collect all the triangles whose minimal angles are smaller than θ_{min} in a list. Then, for the first k triangles in the list, we collapse the opposite edge of the minimal angles.
- (4) **Valence optimization.** Finally, we optimize the mesh valence using the method proposed by Botch and Kobbelt [2004], followed by a mesh smoothing operation.

Steps 2 to 4 are repeated until the desired angle bounds are satisfied.

3 RESULTS AND DISCUSSION

A prototype of the presented algorithm is implemented as a plugin built on top of the open-source software, “graphite”¹. All the results shown in this poster are computed on a PC with a Win7 OS, 3.6GHz CPU, and 16G memory.

We compare our algorithm against the most recent representative remeshing approaches, including three main categories. (1) Methods based on blue-noise sampling, i.e., *Maximal Poisson-disk Sampling* (MPS) [Yan and Wonka 2013]. (2) Discrete optimization, i.e., *Realtime Adaptive Remeshing* (RAR) [Dunyach et al. 2013] and *Minimal Angle Improvement* (MAI) method [Hu et al. 2017]. (3) CVT based approaches, i.e., standard CVT approach [Yan et al. 2009] and augmented CVT with non-obtuse (NOB) remeshing [Yan and Wonka 2016]. We select a challenging model for the experiments, i.e., David’s head. This model has very rich geometric details in different scales, which cannot be handled successfully by previous approaches to produce non-obtuse meshes. Our approach is able to produce a satisfied result efficiently. The comparison of our result against several representative algorithms is shown in Fig. 1.

In the next, we plan to further improve the performance and robustness of our algorithm, and make more in-depth comparisons with state-of-the-art approaches.

ACKNOWLEDGMENTS

This project was partially supported by the National Natural Science Foundation of China (Nos. 61372168, 61620106003).

REFERENCES

- P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. 2008. Recent Advances in Remeshing of Surfaces. In *Shape Analysis and Structuring*. 53–82.
- Mario Botsch and Leif Kobbelt. 2004. A Remeshing Approach to Multiresolution Modeling. In *Proc. of Symp. of Geometry Processing*. 189–196.
- Marion Dunyach, David Vanderhaeghe, Loic Barthe, and Mario Botsch. 2013. Adaptive Remeshing for Real-Time Mesh Deformation. In *Eurographics Short Papers Proceedings*. 29–32.
- Kaimo Hu, Dong-Ming Yan, David Bommes, Pieere Alliez, and Bedrich Benes. 2017. Error-Bounded and Feature Preserving Surface Remeshing with Minimal Angle Improvement. *IEEE Trans. on Vis. and Comp. Graphics* (2017).
- S. A. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.
- Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. 2009. Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram. *Computer Graphics Forum (Proc. SGP)* 28, 5 (2009), 1445–1454.
- Dong-Ming Yan and Peter Wonka. 2013. Gap Processing for Adaptive Maximal Poisson-Disk Sampling. *ACM Trans. on Graphics* 32, 5 (2013), 148:1–148:15.
- Dong-Ming Yan and Peter Wonka. 2016. Non-obtuse Remeshing with Centroidal Voronoi Tessellation. *IEEE Trans. on Vis. and Comp. Graphics* 22, 9 (2016), 2136–2144.

¹<http://alice.loria.fr/software/graphite>