Rounding 3D meshes

October 16, 2018

Contact. Sylvain Lazard (lazard@loria.fr), LORIA Nancy, GAMBLE team, http://gamble.loria.fr/

Context. In most software, 3D polygonal meshes have vertices whose coordinates are represented with fixed-precision floating-point numbers, say doubles. Unfortunately, actions, such as rotations and intersections, create vertices whose coordinates cannot be *exactly* represented with doubles. A natural problem is thus to round the coordinates to doubles, without creating self-intersections between the faces and without changing the geometry too much; the faces can however be subdivided. This problem is of interest in academic and industrial contexts because many 3D digital models contain self intersections (45% of the 10 thousands triangle meshes in the famous Thingiverse data base) and many applications require models without self intersections.

Problem. Given a set of interior-disjoint triangles in 3D whose vertices have arbitrary coordinates, we want to compute a set of interior-disjoint triangles, whose geometry is close to that of the input and such that the output vertices have coordinates of fixed precision, typically integers or fixed-precision floating-point numbers (eg. int, float, double). For simplicity, people aim at rounding the coordinates on a regular grid such as the grid of integers. This problem is well resolved for segments in 2D (see [3,4] and references therein) but, until recently, the only known algorithm for triangles in 3D solved a variant problem in which vertices are rounded on a grid whose granularity, $1/n$, depends on the combinatorial size n of the input [2].

Goal. We recently presented an algorithm that solves the problem in 3D on the integer grid [1]. This solution is not yet known to be practical because its worst-case complexity $O(n^{19})$ is very bad but we conjecture a good complexity of $O(n\sqrt{n})$ in practice on non-pathological data.

The goal of this Master thesis, which could lead to a Ph.D. thesis, is to investigate and eventually develop a practical and efficient solution for this problem. The approach will be to implement a simplified version of our algorithm in [1] and, in parallel, to work at improving its efficiency, both in theory and in practice.

References

[1] Olivier Devillers, Sylvain Lazard, and William J. Lenhart. 3D Snap Rounding. In *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2018/8743, doi:10.4230/LIPIcs.SoCG.2018.30.

[2] Steven Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. *Discrete & Computational Geometry*, 22(4):593–618, 1999. doi:10.1007/PL00009480.

[3] Leonidas J Guibas and David H Marimont. Rounding arrangements dynamically. *International Journal of Computational Geometry & Applications*, 8(02):157–178, 1998. doi:10.1142/S0218195998000096.

[4] John Hershberger. Stable snap rounding. *Computational Geometry*, 46(4):403–416, 2013. doi:10.1016/j.comgeo.2012.02.011.