Computers & Graphics xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Issue on CAD/Graphics 2017

Efficiently computing feature-aligned and high-quality polygonal offset surfaces

Q1 Wenlong Meng[a], Shuangmin Chen[a,*], Zhenyu Shu[b], Shi-Qing Xin[c], Hongbo Fu[d], Changhe Tu[c]

[a] Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China
[b] School of Information Science and Engineering, Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China
[c] School of Computer Science and Technology, Shandong University, Jinan 250101, China
[d] School of Creative Media, The City University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 14 June 2017
Accepted 2 July 2017
Available online xxx

Keywords:
Offsetting
Particle system
Feature alignment

ABSTRACT

3D surface offsetting is a fundamental geometric operation in CAD/CAE/CAM. In this paper, we propose a super-linear convergent algorithm to generate a well-triangulated and feature-aligned offset surface based on particle system. The key idea is to distribute a set of moveable sites as uniformly as possible while keeping these sites at a specified distance away from the base surface throughout the optimization process. In order to make the final triangulation align with geometric feature lines, we use the moveable sites to predict the potential feature regions, which in turn guide the distribution of moveable sites. Our algorithm supports multiple kinds of input surfaces, e.g., triangle meshes, implicit functions, parametric surfaces and even point clouds. Compared with existing algorithms on surface offsetting, our algorithm has significant advantages in terms of meshing quality, computational performance, topological correctness and feature alignment.

1. Introduction

An *offset* surface [1], also called a parallel surface, consists of all the points that are at a constant distance d to an input surface. The computation of surface offsets is a common and fundamental operation in various applications in CAD/CAE/CAM [2–4], e.g., hollowed or shelled solid model generation for rapid prototyping.

There is a large body of literature on computing offset surfaces. Existing methods can be roughly divided into three categories depending on the specific representation form of the input surface. For parametric curves or surfaces, a commonly used approach [5–7] is to generate parametric offsets first, followed by carefully handling tangent discontinuities, cusps and self-intersections. When the input is a polygonal surface or implicit surface [1,8,9], one has to build a volumetric scalar field with a dense resolution and then extract the iso-surface at the specified distance. However, such an approach has at least two disadvantages including (1) it requires a huge time/space cost since the total number of voxels is $O(1/\epsilon^3)$, where ϵ is the accuracy tolerance, and (2) the final offset surface does not have a desirable triangulation quality. Finally, it seems that offset surfaces can be obtained by a series of mesh boolean operations [10] across a sufficiently large number of spheres centered at the base surface, but experimental results show that it cannot work well in practice due to the fact that the meshing quality gets worse and worse after many boolean operations. This motivates us to develop an easy-to-use tool for generating a well-triangulated and feature-aligned offset for an input surface that can be a polygonal surface, a parametric surface, an implicit surface, or even a point cloud.

In this paper, we propose a super-linear convergent algorithm to generate polygonal offsets. The key idea is to distribute a set of moveable sites as uniformly as possible while keeping these sites at a specified distance from the original surface throughout the optimization process. Because of the uniform distribution of these sites, an additional quick step of simply connecting sites is sufficient for producing the final triangle mesh. An example is shown in Fig. 1.

Our main contributions are at least threefold:

1. Taking the uniformity of sites as the objective function whereas the specified distance to the base surface as the hard constraint, we formulate the offsetting problem using particle system, which can be efficiently solved due to the closed-form formula of the gradients of the objective function.

---

* Corresponding author.
E-mail addresses: chenshuangmin@nbu.edu.cn, csmqq@163.com (S. Chen).

http://dx.doi.org/10.1016/j.cag.2017.07.003
0097-8493/© 2017 Elsevier Ltd. All rights reserved.

Please cite this article as: W. Meng et al., Efficiently computing feature-aligned and high-quality polygonal offset surfaces, Computers & Graphics (2017), http://dx.doi.org/10.1016/j.cag.2017.07.003

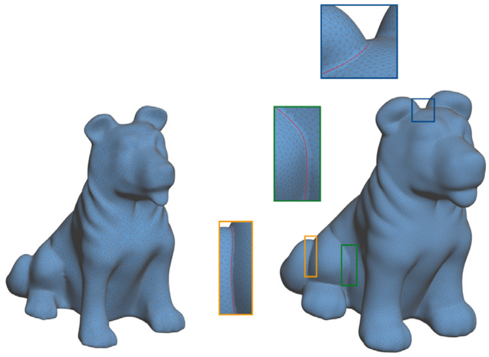
(a) Input (b) Offset surface

Fig. 1. Our algorithm is able to produce a feature-aligned and high-quality offset surface (b) for the input surface (a); See the close-up views.

2. Throughout the optimization process, we use the moveable sites to predict the potential feature regions of the final offset surface, which is in turn enforced on the objective function to guide the distribution of the moveable sites, leading to a feature-aligned triangulation.
3. The algorithm framework is powerful and supports various kinds of input surfaces, including polygonal surfaces, parametric surfaces, implicit surfaces and even point clouds.

2. Related work

At least three kinds of works are related to the theme of this paper, including surface offsetting, particle system, and remeshing.

2.1. Surface offsetting

Existing offset algorithms assume that the input surface has a specific representation form. When the input surface has a parametric form, it is quite often to represent the offset surface as a parametric form as well. Existing algorithms of this kind focus on seeking a polynomial/rational alternative to approximate the exact parametric form, and handling tangent discontinuities, cusps and self-intersections. For example, Filip et al. [11] developed a theorem on approximation accuracy using the bounds of second derivatives of the original curves and surfaces. Piegl and Tiller [12] proposed to approximate the offset surface with the fewest number of control points. Kumar et al. [13] developed a set of trimming techniques to handle invalid local intersections. The above-mentioned methods, whose input and output are both in parametric form, are different from the goal in this paper, i.e., generating a high-quality polygonal offset surface.

When the input is a polygonal or implicit surface, one can build a volumetric scalar field to encode signed distances to the base surface and then extract the offset surface based on the marching cube technique [8,14,15]. However, the resolution of voxelization is hard to set. Coarse voxelization may lead to a topologically incorrect reconstructed offset surface but an over-dense voxelization requires a huge time/space cost. What is important is that it cannot produce a high-quality triangle mesh to represent the offset surface.

Theoretically speaking, mesh boolean operations [10] seem to be able to compute the offsets individually for each face, edge, and vertex and then return the union of the basic offset elements as the final offset surface. However, experimental results show that mesh boolean operations cannot work well in practice. First, these basic offset elements highly overlap, causing a notorious difficulty in unionizing a large number of such objects. Second, performing mesh boolean operations across a large number of objects is inefficient and cannot guarantee a desirable meshing quality. Similarly, point based reconstruction algorithms [9,16], based on point shifting and filtering operations, cannot guarantee the meshing quality either.

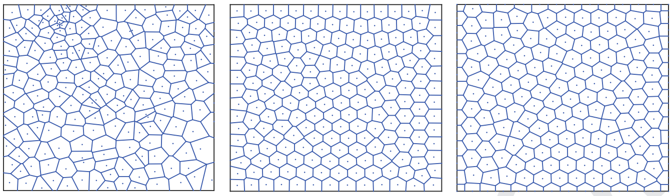
(a) Input (b) CVT (c) Particle system

Fig. 2. For 200 input sites (a), CVT requires about 0.45 s and 91 iterations to get the distribution in (b), while the particle system requires only 0.01 s and 48 iterations to achieve (c). Note that the distribution in (c) is sufficient for the triangulation purpose in practice.

2.2. Particle system vs. CVT

There are many application occasions where we need to distribute a set of sites as uniformly as possible. Both centroidal Voronoi tessellations (CVT) [17,18] and particle systems [19–22] can serve for this purpose. Du and Wang [23] introduced the Lloyd method to compute CVT and apply it into optimal tetrahedral mesh generation, while Liu et al. [24] proposed a quasi-Newton method to compute CVT and demonstrated the extraordinary ability in surface remeshing. Particle system, by contrast, has a sound basis in physics and can serve for the same purpose by minimizing the global inter-particle forces to make the particles (sites or vertices) keep the optimal balanced state, leading to a collection of uniformly distributed particles. Generally speaking, particle system is able to generate a desirable site distribution with less computational cost [25] in contrast to CVT. As Fig. 2 shows, particle system runs about many times faster than CVT in producing a uniform distribution of almost the same quality. Therefore, in this paper, we adopt particle system to iteratively optimize the distribution of sites (serving as vertices of the final offset surface).

2.3. Remeshing

A wide range of applications require meshes with high-quality triangulation to facilitate numerical computation, and thus remeshing is an important research topic in computer graphics. Roughly speaking, there are three kinds of remeshing depending on various purposes. The first kind targets at uniform triangulation, which seeks for an as-uniform-as-possible vertex distribution [24]. The second kind of remeshing algorithms aims at isotropic or anisotropic triangulation assuming that the base surface is equipped with a density function or an anisotropic metric to encode the underlying distance. For example, Chen et al. [26] developed an isotropic remeshing method based on constrained centroidal Delaunay mesh(CCDM), while Zhong et al. [27] introduced a particle-based approach for anisotropic surface meshing. The third kind is to align triangulation with geometric features. For example, Lai et al. [28] presented an algorithm which turns an unstructured triangle mesh into a quad dominant mesh with mesh edges well aligned to the principal directions of the underlying surface.

Please cite this article as: W. Meng et al., Efficiently computing feature-aligned and high-quality polygonal offset surfaces, Computers & Graphics (2017), http://dx.doi.org/10.1016/j.cag.2017.07.003