Computer-Aided Design 63 (2015) 26–38

Contents lists available at ScienceDirect

Computer-Aided Design

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

A robust conforming NURBS tessellation for industrial applications based on a mesh generation approach ☆

R. Aubry [a,∗], S. Dey [b], E.L. Mestreau [a], B.K. Karamete [a], D. Gayman [a]

[a] Sotera Defense Solutions, 1501 Farm Credit Drive, Suite 2300, McLean, VA 22102-5011, USA

[b] US Naval Research Laboratory, Code 7131, 4555 Overlook Ave SW, Washington DC 20375, USA

HIGHLIGHTS

- A tessellation technique based on a mesh generation is described.
- A minimum number of elements is generated to encode the shape of a model as compressed as possible.
- Conformity is guaranteed by construction.
- NURBS singularities are commented and handled properly.
- Drawbacks of parametric methods are highlighted.

ARTICLE INFO

Article history:
Received 24 February 2014
Accepted 30 December 2014

Keywords:
NURBS surfaces
Surface triangulation
CAD tessellation
Surface untangling
Parabolic lines
NURBS singularities

ABSTRACT

A NURBS tessellation technique is presented with the goal to robustly approximate CAD surfaces that define the boundary of complicated three dimensional geometric shapes with a minimum number of triangles. The minimization is achieved by generating anisotropic triangles in the three dimensional space. New procedures are presented to handle numerical stability issues due to the anisotropy. The tessellation is generated using a mesh generation viewpoint, as opposed to the more classical viewpoint of graphical visualization of surfaces in CAD. This ensures topological conformity of the resulting mesh. A tiered approximation approach is used for speed and robustness. Degeneracies associated with NURBS curves and surfaces are given special attention as they occur frequently in naval and aerospace conceptual-to-early design process. Analogies with a classical mesh generation process are discussed and several numerical examples illustrate the method.

1. Introduction

Tessellations are critical to the design process as they provide early visual depiction of geometry under consideration. They may also serve as the basis for generation of analysis suitable meshes by either providing a starting mesh or by facilitating its sizing field and gradation, where the sizing field is the sizing dictated by the user at a given location in space. An optimal tessellation would:

- minimize the number of triangles
- bound the distance between the surface and the discrete facets by a user specified normalized error tolerance
- bound the angle between the surface normal and the discrete facets relying on the provided error tolerance
- robustly handle various types of degeneracies in the surface and curve representation commonly employed in the conceptual and early design of aircraft and ship structures
- be a conforming mesh.

The last requirement may not be necessary for visual purposes. However, it greatly simplifies further remeshing procedures. Summing up, the tessellation generation can therefore be reformulated as an interpolation problem.

In the literature, one of the first attempts to achieve optimal interpolation with triangles is proposed by D'Azevedo et al. [1]. Based on a convex quadratic model, they show that the Delaunay triangulation in a transformed plane minimizes the maximum interpolation error. This work is later extended to minimize the gradient error in [2]. Lane et al. [3] appear to have been the first to give a bound in the distance between an analytic surface and its triangulation. However the bound is global and the mesh is isotropic.

☆ This paper has been recommended for acceptance by Xiuzi Ye.

∗ Corresponding author.
E-mail address: romain.aubry.ctr.fr@nrl.navy.mil (R. Aubry).

http://dx.doi.org/10.1016/j.cad.2014.12.009
0010-4485/

This bound was later improved by Filip et al. [4]. However, only orthogonal triangles are considered. Later, Sheng et al. [5] give bounds for a general triangulation. Bézier surfaces are considered and an approximation of the second derivatives through Chebyshev polynomials is proposed to seek a practical algorithm. Piegl et al. [6] generalize this approach to trimmed NURBS surfaces relying on yet another approximation of the second derivatives. More recently, Anglada et al. [7] generalize the bounds for the anisotropic case. As noted in Elber [8], one of the main drawbacks of relying on the second derivative of the parametrization is the dependence of the error estimate on the parametrization of the surface. In order to remove this dependence, other characteristics of the surface are advocated, such as the principal radii of curvature. As mentioned in [9], the previous metric tries to bound the distance between the surface and its tessellation. However, no guarantee on the approximation of the tangent plane is given. Furthermore, surface anisotropy increases the sensibility to face normal change as a small change in coordinates may produce a large change in the face normal, which does not appear for isotropic three dimensional triangles. The proposition of [9] consists mainly in limiting the anisotropy of the metric. Another class of methods relies on the coupling between the surface tessellation and the display by the graphic card of the resulting triangles. Advantage is taken to tessellate only a subset of surfaces visible on the screen [10,11] at the price of maintaining a dynamic environment. NURBS surfaces are converted into their rational Bézier counterparts, as in Rockwood et al. [12]. These methods reach a high speed of display but are neither adaptive nor anisotropic, relying on a user specified distance on the screen. Some methods split the surface into subpatches until trimming contours form a unique loop [12,13]. The main drawback of this approach is the necessity of handling non conforming patches inside the surface. In all the previous references, the tessellation is performed from the parametric plane without any reference to the three dimensional space. Common NURBS features include degeneracies [14–16], where the G^1 continuity may not even exist [17]. Tangent plane and normal computations may be jeopardized. Therefore, some other methods prefer to rely on an error estimate based on a three dimensional criterion. In Balázs et al. [18], the distance between the control points and the bilinear surface approximation is considered. In Shu et al. [19], a similar "flat test" is used. In the same spirit, Piegl et al. [20] do not assume any differentiability of the surface and rely on an approximating plane of the four corners of the control polygon. In Haimes et al. [21], triangles are split at their centroid if the distance between the centroid and the surface is larger than the defined tolerance. We conclude this literature review by noting that very few references about CAD tessellation generation exist from the main CAD vendors, while CAD tessellation is not an entirely solved problem [22].

Originally, tessellations were primarily created for visualization purposes where every geometric face is tessellated independently. This approach leads to non conforming meshes at face boundaries which may be acceptable for visualization but not for analyses or mesh generation input that relies on conforming tessellations. In this work, a finite element mesh generation process is followed, where edges are meshed first followed by faces that use the existing edge meshes for the boundaries. Conformity is therefore enforced by the design of the meshing process. Conformity is essential to classify mesh entities against their respective geometry entities [23], to be able to remesh a surface from the original tessellation [24], or simply to have a conforming mesh to output the tessellation compared to a stereolithographic model (STL), where many topological pathologies may arise [25–27]. Finally, minimizing the number of triangles allows encoding a geometry with a minimum number of triangles given an accuracy threshold. It also provides the ability to generate sizing fields with a minimum number of elements [28].

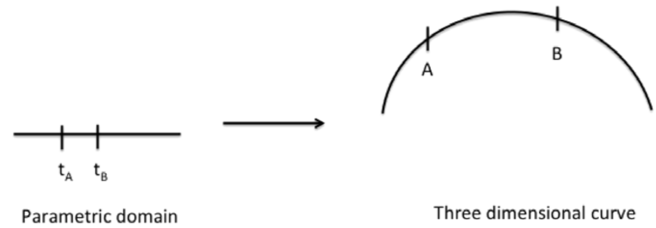

Fig. 2.1. A parametric trimmed curve within parameters t_A and t_B in the parametric one dimensional domain, bounded by vertices **A** and **B** in three dimensions.

The rest of this paper is as follows. Parametric curves and surfaces along with their approximation are reviewed in Section 2; they represent the most widely used class of surface definition in CAD kernels. This allows us to define basic tools to guarantee an accurate tessellation. Some practical pitfalls that are associated with this approach are also discussed. Section 3 presents the core of the method which relies as much as possible on using the three dimensional space instead of relying on G^1 continuity of the parametric form. Finally, Section 4 illustrates the capabilities of the method on some representative examples.

2. Parametric curves and surfaces

In this section, some basic notions of curves and surfaces are recalled to address the tessellation interpolation problem. It gives the basic tools to measure the tessellation accuracy, and to illustrate the potential pitfalls met in practice. As far as notations are concerned, quantities belonging to the parametric space are noted with lower case letters, while quantities belonging to the three dimensional space are noted with capital letters.

2.1. Parametric curves

In this part, parametric curves are considered. Then, interpolation error estimates are commented. Let **C** be a parametric curve from a domain $\omega \subset \mathbb{R}$ to $\Gamma \subset \mathbb{R}^3$ as:

$$\mathbf{C} : \omega \subset \mathbb{R} \rightarrow \Gamma \subset \mathbb{R}^3 \quad (2.1)$$
$$t \rightarrow \mathbf{C}(t).$$

Typically, the parametric curve is assumed to be twice differentiable with respect to t. However, this may not be true in practice.

2.1.1. Length computation

The main aim of a tessellation is to approximate the geometry of the curve as accurately as possible. The length of the curve represents a critical tool to accomplish this task. The length L of a parametric curve from parameter t_0 to t_1 reads:

$$L = \int_{t_0}^{t_1} \|\mathbf{C}'(l)\| \, dl \quad (2.2)$$

where $\|\mathbf{C}'(l)\|$ is the norm of the tangent vector of the curve. For a straight mesh edge **AB** on the parametric line reparametrized with respect to t between 0 and 1, as shown in Fig. 2.1, it reads:

$$L = \int_0^1 \sqrt{(\mathbf{B}-\mathbf{A})^T \|\mathbf{C}'(t)\| (\mathbf{B}-\mathbf{A})} \, dt. \quad (2.3)$$

However, the three dimensional length of the curve does not indicate if more mesh edges should be used to discretize the curve accurately. An interpolation error is needed to drive the refinement process. By modifying the metric used in the length of a curve, it is possible to obtain more insight on where to refine, as shown in the next section.