A CURVATURE-SENSITIVE PARAMETERIZATION-INDEPENDENT TRIANGULATION ALGORITHM

OSCAR RUIZ, JOHN CONGOTE, CARLOS CADAVID, JUAN G. LALINDE

ABSTRACT. Triangulations of a connected subset F of parametric surfaces $S(u,v)$ (with continuity C^2 or higher) are required because a C^0 approximation of such F (called a *FACE*) is widely required for finite element analysis, rendering, manufacturing, design, reverse engineering, etc. The triangulation T is such an approximation, when its piecewise linear subsets are triangles (which, on the other hand, is not a compulsory condition for being C^0). A serious obstacle for algorithms which triangulate in the parametric space $u-v$ is that such a space may be extremely warped, and the distances in parametric space be dramatically different of the distances in R^3. Recent publications have reported parameter -independent triangulations, which triangulate in R^3 space. However, such triangulations are not sensitive to the curvature of the $S(u,v)$. The present article presents an algorithm to obtain parameter-independent, curvature-sensitive triangulations. The invariant of the algorithm is that a vertex v of the triangulation if identified, and a quasi-equilateral triangulation around v is performed on the plane Π tangent to $S(u,v)$ at v. The size of the triangles incident to v is a function of $K(v)$, the curvature of $S(u,v)$ at v. The algorithm was extensively and successfully tested, rendering short running times, with very demanding boundary representations.

INTRODUCTION

The object of this article is to describe the algorithms developed to calculate a Piecewise Continuous approximation for a Boundary Representation (B-Rep), whose FACEs are have C^2-continuous underlying parametric surfaces $S : R^2 \rightarrow R^3$ as carrier geometries. In [12], a previous investigation of the CAD CAM CAE Laboratory calculated patterned triangulations by laying a regular orthogonal grid of Steiner vertices on the parameter space pre-image $S^{-1}(F)$ of the FACE F to triangulate. The triangulation calculated in this manner is limited in these aspects: (i) constant vertex density in the triangulation ignores the local characteristics of the surface S in R^3. Ideally, a larger or smaller triangle density should be calculated, as per the local curvature of S. (ii) a serious flaw of triangulation algorithms in parametric space is that the parameterization (u,v) is in general warped: this means, the parametric distance $|(u_1,v_1)-(u_2,v_2)|$ is significantly different to the euclidean distance $|S(u_1,v_1)-S(u_2,v_2)|$, and it is so in an irregular manner, usually suffering different warping in different local points and in different directions. These characteristics may produce a non-manifold triangulation for strongly warped parameterizations ([10]).

2000 *Mathematics Subject Classification.* Primary 65D17; Secondary 65D18, 68U05.

Key words and phrases. triangulation, parametric surface, parameter-independent, curvature-sensitive.

This work was supported by EAFIT University and the Colombian Council for Science and Technology -Colciencias-.

In the current investigation, a triangulation T is sought, which is sensitive to the surface curvature but not sensitive to parameterizations. In particular, no sensitive to warped parameterizations.

1. LITERATURE SURVEY

Several classifications of the reviewed literature are possible: in the first place, [11], [4] and [6] treat the re-meshing of an already triangulated B-rep. Level of Detail is tangentially treated in [9], [6] and [16]. [2] and [5] deal with the quasi-equilateral triangulation in F by iterative point search on $U \times V$ 2D parametric space. [18] and [1] pay special attention to the approximation of the face edges as NURBS or Bezier curves in R^2.

In [9] an initial mesh is refined according to the disposition of the observer and the scene lights. An emphasis is set on multi-resolution only on the triangles that actually are seen by the observer. An directed acyclic graph (DAG) is formed, which tracks the modification operations performed on the vertices, edges or faces of a initial model. A Hausdorff distance between the reference and the current surfaces at the modified feature (edge, vertex, face) is evaluated, and the modifications are performed starting at sites with small value of such a measure (i.e. simplifications which only *slightly* modify the current surface when compared with the original one). The algorithms are designed to work in *image space* rather than in *object space*: subdivision is only performed if it does not surpass a threshold in the error introduced in the model, and it has an effect on the image. For example, if a triangle affects only one pixel there is no point in it being further subdivided.

In [18] an emphasis is set in producing watertight tessellations (borderless 2-manifolds in R^3) by using connectivity information. The face-face connectivity between the contiguous faces F_1 and F_2 is represented as a planar trimming curve $C_{1,2}(u)$ that is the common limit between the 2D regions (in parametric space $U-V$) that bound F_1 and F_2. A curvature-sensitive algorithm places vertices on the $C_{1,2}(u)$ curve. In the current article, the $C_{1,2}(u)$ curve is not required, as the implemented algorithm directly samples the edge curve in R^3 using the *curve* sampling interval specified by the user. In our algorithm, this sample on R^3 is tracked back to the $U-V$ plane by forming a piecewise linear approximation of the trimming curve $C_{1,2}(u)$.

In [11], the authors start with a watertight 2-manifold M with C^0-continuity (a triangulated tessellation), and build a set of parameterizations for M. Each parameterization covers what is called an *internal node* (representing an M_i 2-manifold with border) in the Reeb Graph describing the topological chances in M along the range of a Morse function $f : M \longrightarrow R$. As per the Morse theory, M_i represents a portion of the M manifold, for which f has no singular points (topological changes of M) and therefore represents the complete log of the topological evolution of M. Four types of M_i are possible: cylinders, cups, caps, and branchings, according to the borders of M_i. For each type, a pre-defined routine is used, which parameterizes M_i. The step of making compatible the parameterizations for $M_i, i = 0, 1, 2, \ldots$ is avoided by remeshing the parameterizations with higher density at the borders of M_i. In this form, still a series of parameterizations is possible, while guaranteeing a watertight remeshed M_r version of M.

In [2] and [3] a parameterization-independent algorithm is proposed to triangulate a surface. The aim of the authors is to produce a nearly uniform triangulation. That is, a triangulation in which the triangles be quasi-equilateral. A vertex $p = S(u_0, v_0)$ is chosen on $S(u, v)$ and the plane tangent to S at p, $T_p(p)$, is calculated. On $T_P(p)$, a circle with radius R and its regular inscribed polygon with n sides (called Normal Umbrella - NU) are constructed along with the n incident triangles covering the 2Π angle around p. Each angle that contributes to 2Π is projected onto S, with vertex $p = S(u_0, v_0)$ and projection rays perpendicular to $T_p(p)$. The radius R is inversely proportional to the local curvature. Our own implementation of [3] was found that when the region already sampled closes onto itself, in the EDGE neighborhoods or near FACE holes, an illegal overlap of triangles is produced and the algorithm to avoid it is difficult to control.

In [1] the display of a trimmed NURBS face is discussed, in which a *compilation* stage is performed. The compilation stage is equivalent to what other authors call *the triangulation*. The face in parametric UV space corresponds to a 2D connected region with holes, bounded by curved Bezier approximations of the NURBS trimming curves. Bezier approximations are used because there exist reasonable algorithms for the finding of a root of a Bezier curve. The region in UV space is cut into sub-regions which have monotonically increasing or decreasing values of the U and V parameters. These subregions are triangulated separately. As an improvement, the algorithm implemented in this paper avoids the splitting of the UV region into subregions. It also requires only linear intersections (not Bezier ones), leading to a very simple implementation.

[4] presents a mesh-improving method that starts with a t opologically valid although geometrically poor triangular mesh. The geometric degeneracies are classified as *needles* (quasi isosceles triangles that have two vertices very close to each other) and *caps* (triangles with one angle very close to 180°). The elimination of needles is relatively simple. Elimination of each cap requires the slicing of the *whole* mesh along a particular plane, producing an over-population of triangles. The distance between the final and initial triangulations is used to accept or reject the cap and needle elimination. [6] starts from reverse engineering or tessellation triangular meshes to execute quality improvement and property control on them. The article applies the subdivision and simplification functions to augment and diminish the degree of freedom of the mesh, respectively. Several heuristics are applied to refine the mesh: geometric error, face size, faces shape quality, edge size and vertex valence. In neither [4] nor [6] the mesh modifications are evaluated against the original solid, but against an existing triangulation of it. A comparison with our article is not possible, since our work seeks an initial triangulation for a given solid.

[5] propose a quasi - isometric local mapping from a parametric surface $S(u, v) : U \times V \rightarrow R^3$ by using the control polyhedron (called there the *surface net*) of the parametric surface. The reasoning is that the surface net closely follows the warping of the parametric surface, while at the same time is very similar to a *locally* developable surface (in turn a planar surface). If we assume that a 1-1 function $f : U \times V \rightarrow S_D \subset R^2$ is known (S_D is the developed surface net), then a quasi equilateral triangulation could be calculated on S_D, and taken to the $U \times V$ domain by using f^{-1}. From $U \times V$ the triangulation is taken to R^3 by using the parametric equations $S(u, v)$. The image in $U \times V$ of the quasi equilateral triangles in S_D is not

quasi-equilateral, but their image in R^3 would be. The paper presents no examples in which S_D does not exist for the original surface, and a subdivision must be done, but mentions this possibility.

[16] discusses the issue of triangulation a trimmed surface F by sub-dividing a rectangular domain in the $U \times V$ space using Quadtrees. Each quadtree is recursively subdivided if its corner points in R^3 deviate from a plane beyond a prescribed limit. The trimming NURBS curves, which limit the face F to triangulate are represented as piecewise linear in R^3 and in the parametric $U \times V$ space also. The quadtrees which are completely inside the piecewise linear boundary are trivially triangulated. The ones cut by a loop segment are triangulated only in its internal extent. The quadtree portions in $U \times V$ external to the boundary loops are not triangulated. The paper mentions but does not discuss a process of conciliation between the triangulations of adjacent faces in order to have a seamless triangulation at the faces boundaries.

[13] and [15] are quite important references, used in this paper, regarding the triangulation of 2D regions. In the present work, a Constrained Delaunay Triangulation was used, which respects prescribed edges defined on a set of planar points.

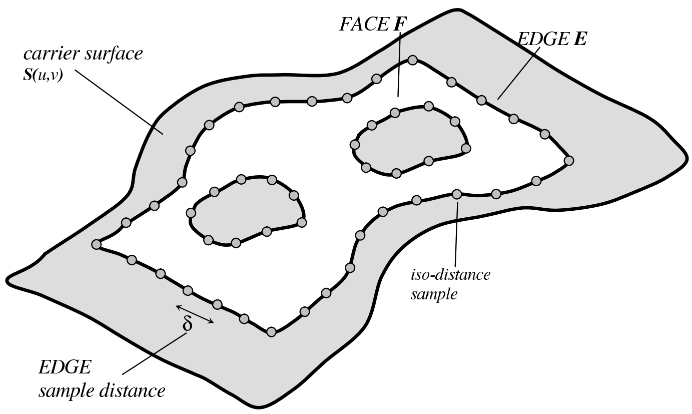
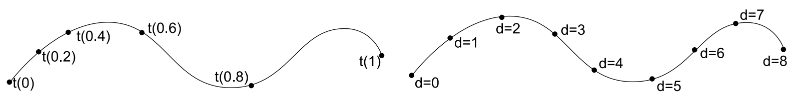


FIGURE 1. The iso-distance sample of edges on F generate the triangulation initial vertex set.

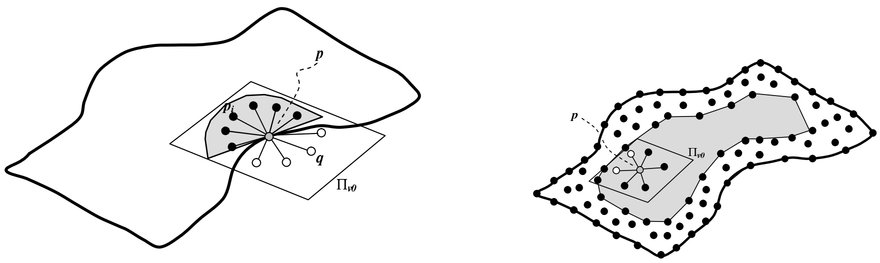


(a) Parametric Sampling Edge (b) Arc Length Edge Sampling

FIGURE 2. Warping of the parameter in a parametric curve

2. METHODOLOGY

Two algorithms for the triangulation of a face F will be discussed here: (a) a random sprinkle of triangulation vertices in the parametric pre-image of the face $S^{-1}(F)$, and (b) a star shape proposal of vertices centered on an 'accepted' triangulation vertex. These algorithms will be called 'sprinkle' and 'star', respectively.



(a) Tangent Plane Π_{v_0} at the point p. (b) Growth of the sampled region

FIGURE 3. Algorithm Initial steps: (i) equi-angled sample star , (ii) growth of the point sampled region

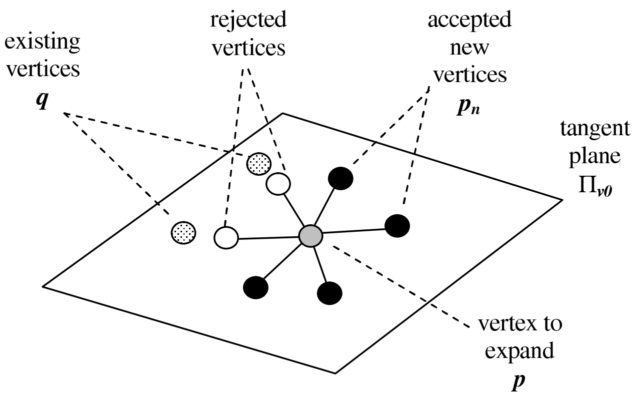
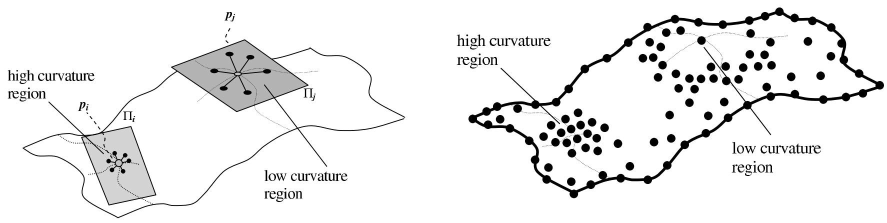


FIGURE 4. Accepted and rejected vertices in star expansion.



(a) Sampling distance as function of the curvature (b) Sampling density as function of the curvature

FIGURE 5. Regions with higher curvature are sampled with smaller sampling interval.

In both cases the algorithms will be sensitive to the local curvatures of F and independent to the parameterization u, v of S.

2.1. **Differential Geometry of S.** The Gaussian (K) and Mean (H) curvatures of a surface are given by ([8],[17]):

$$K = \frac{ln - m^2}{EG - F^2} \quad H = \frac{En - 2Fm + Gl}{2\,(EG - F^2)} \tag{1}$$

with :

$$E = I\langle \vec{u}, \vec{u}\rangle,\ F = I\langle \vec{u}, \vec{v}\rangle,\ G = I\langle \vec{v}, \vec{v}\rangle,\ l = II\langle \vec{u}, \vec{u}\rangle,\ m = II\langle \vec{u}, \vec{v}\rangle,\ n = II\langle \vec{v}, \vec{v}\rangle \tag{2}$$

being I and II the First and Second fundamental forms of the surface $S(u,v)$, respectively.

2.2. **Implemented Algorithms.** The sprinkle and star implemented algorithms have in common: (i) The pre-image of F under $S(u,v)$, named here $S^{-1}(u,v)$ must be calculated. This must be a connected region in the $U-V$ parametric space, which is approximated as the interior of a polygon, simple, with holes, in $U-V$ (R^2) space. (ii) The density of sampled points on the face F, at the point $p = S(u,v) \in F$ is proportional to the local curvature of the supporting surface $S(u,v)$, as per equations 1 and 2. (iii) The different velocities of the parameterization u,v in the directions u and v are taken into consideration in order to isolate the triangulation of such velocities. Each algorithm has a different approach for such a goal. (iv) The algorithms start with the generation of an initial triangulation vertices on the edges of the internal and external loops of the face F boundary (∂F). The iso-metric sample of ∂F appears in figures 1 and 2(b). The Sprinkle and Star algorithms generate the triangulation vertex set. The calculation of the connectivity proceeds independently, and several algorithms are used, regardless of the vertex generation algorithm used.

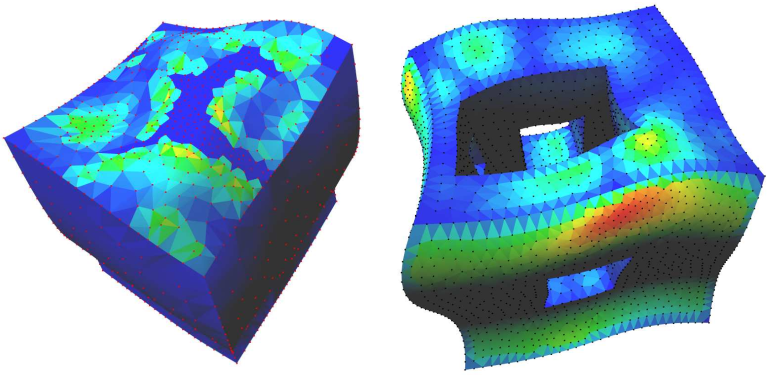
2.3. **Sprinkle Algorithm.** The Sprinkle algorithm for the triangulation of a face $F \subset R^3$ mounted on (i.e being a connected subset of) a parametric surface $S(u,v)$ consists of the following steps: 0- Iso-metric sample of the edges of the loops ∂F which bound F (Figure 1). 1- Calculation of the pre-image of F, $S^{-1}(F)$. 2- Calculation of the bounding box of $S^{-1}(F)$, named $minmax(S^{-1}(F))$ here. 3- Iterative generation of a set V of euclidean points $p = S(u,v) \in R^3$ by a random process $(u,v) = (random, random)$ in $S^{-1}(F)$, by mapping the (u,v) point onto $minmax(S^{-1}(F))$. 4- Validation of the (u,v) points generated in (3) if a ball in euclidean space R^3, $B(r(K), S(u,v))$ of radius $r(K)$, centered in $S(u,v)$, contains no other generated point $S(random, random)$. The generation stops when a given number ($N = 1000$) of attempts to sprinkle the random points on $S^{-1}(F)$ fail. 5- Calculation of a modified Delaunay Triangulation with the points in V. Results are shown in Figure 6(a).

It must be pointed out that the radius of the ball $r(K)$ is sensitive to the curvature K in the surface S at the point $S(u,v)$ (Equations 1, 2). In this manner, the vertex density per area is proportional to the local curvature measure of the surface.

2.4. **Star Algorithm.** The Star algorithm consists of the following steps: 0- Iso-metric sample of the edges of the loops ∂F which bound F and queueing of such vertices in queue Q. 1- Calculation of the pre-image of F, $S^{-1}(F)$ (Figure 1). 2- For each vertex $v = S(u,v) \in Q$: 2.1 calculation of the osculating plane $\Pi_{v,n}$ tangent to S at $S(u,v)$, 2.2 generation of a regular hexagon X with radius $r(K)$ and center at

$S(u,v)$, embedded in $\Pi_{v,n}$. 2.3- rejection of the vertices of X which are too close to the current set of vertices on F. 2.4- rejection of the vertices of X whose projection on S falls outside F. 2.5 projection of the remaining vertices of X back onto F and inclusion in queue Q. 2.6 Elimination of v from Q and inclusion of v into the set of final triangulation vertices V. 3. Calculation of a modified Delaunay Triangulation with the points in V.

In the Star algorithm, the rejection in point 2.3 uses also the curvature-sensitive radius $r(K)$.



(a) Random Sprinkle of points in $U-V$ space.

(b) Star Sampling of points on osculating planes $\Pi_{v,n}$ tangent to S

FIGURE 6. Vertex Layout Methods: Sprinkle vs. Star Sampling.

2.5. **A Pseudo - Delaunay Triangulation.** To generate the connectivity information of the set of discretized points two approaches were used. The first one is to calculate a triangulation in parametric space (using [14]), resulting in Figure 8(a). The second one (called *Pseudo-Delaunay*) is a contribution of our work and it is described next.

An initial PL approximation of ∂F, the loops bounding F, is available (see previous sections) in a queue Q_E. The implemented algorithm consists of the following steps: 1- Extraction of the next edge $e=(v_i,v_j)$ from Q_E. 2- Identification of the vertex $v \in V$ such that the triangle $t=(v_i,v_j,v)$ is a pseudo-Delaunay one (see below). 3- Addition of the edges (v_i,v) and (v_j,v) to the queue Q_E.

A pseudo-Delaunay triangle $t=(v_i,v_j,v)$ with $v_i \in F$ is tested in this manner: (a) Find the circumcenter c_t of the triangle t and the radius r_t of the planar circle containing t. (b) Consider the ball $B_t(c_t,t_t)$ centered in c_t with radius r_t. (c) Test every vertex w of V for inclusion on B_t. If no $w \in V$ is inside B_t, t is a pseudo-Delaunay triangle (Figure 7).

The results a triangulation in parametric space $U-V$ vs. the pseudo-Delaunay one in R^3 are displayed in Figures 8(a) and 8(b), respectively.

3. RESULTS

The algorithm was tested with several solids: Bearing, Helmet, Crank, Support. The Figures 9(a), 9(b), 9(c), 9(d) represent the results of the algorithm with solid of different characteristics. The test solids B in which the star algorithm was applied

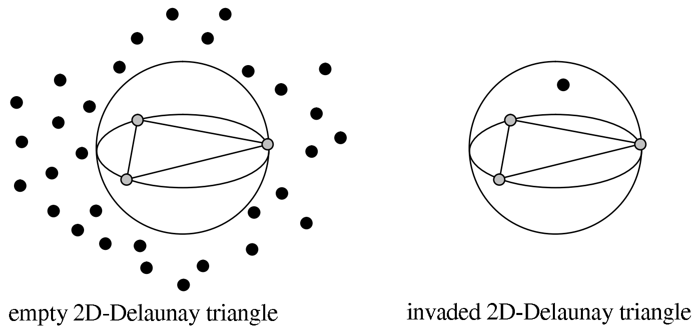
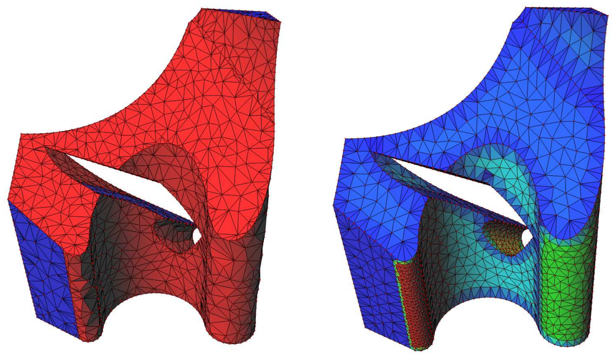


FIGURE 7. Pseudo-Delaunay validation of triangles lying on the face F.



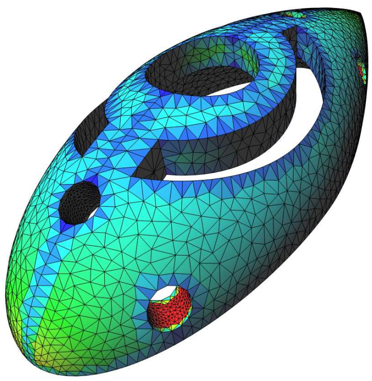
(a) Parametric Triangulation (b) Pseudo-Delaunay Triangulation in Space

FIGURE 8. Triangulations in Parametric Space and Directly in 3D Spaces

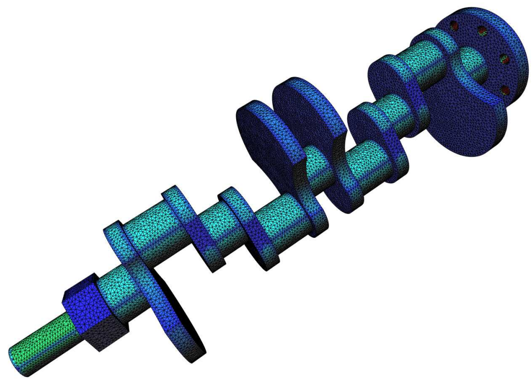
have the following characteristics: (a) the boundary representation of B, ∂B, must have several connected components (i.e. several shells), (b) ∂B must have a number of faces F with high curvature variations, (c) the boundary of a number of faces F must have several connected components (i.e. the face F must have holes), (d) the faces F of B must have a large variation in size (i.e. no sampling rate is applicable to all ∂B, but instead the algorithm must find correct sampling intervals for each face F).

4. CONCLUSIONS AND FUTURE WORK

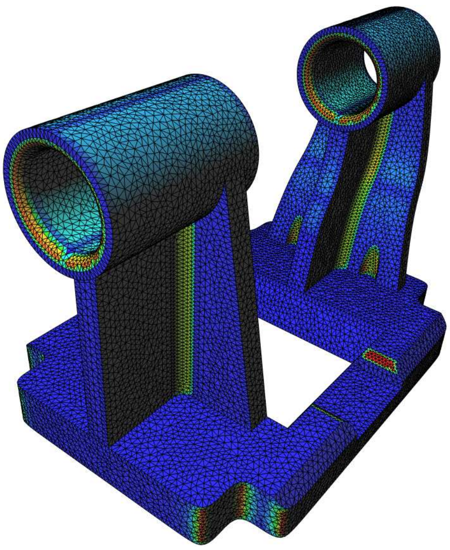
The algorithms implemented present advantages with respect to their closest counterpart, the algorithms by Attene et al ([3]), as follows: (i) in Attene's algorithms (implemented as part of our work), the triangles present poor quality as the mesh closes onto itself, approximately at the medial axis of the face F. (ii) In [3] no reference is made to faces with holes, while our results sufficiently prove that our algorithm addresses such cases. (iii) [3] takes into consideration the high curvature regions by expanding a n-side regular polygon with high n, which leads to more extreme isosceles triangles. In our approach, local high curvature regions



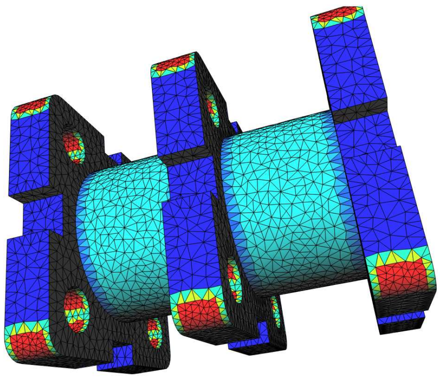
(a) Helmet



(b) Crank



(c) Bearing Support



(d) Central Support

FIGURE 9. Case Studies Run

produce a star expansion with a small-radius, regular hexagon on $Pi_{n,v}$, which leads to approximately equilateral triangles on F.

The following points need to be addressed in a near future: (a) theoretical proofs of the correctness of the implemented algorithms and heuristics, (b) introduction of Finite Element Analysis measures for the goodness of the triangles generated and adequation of the algorithm to observe such measures, (c) reduction of the time and memory complexity of the algorithms, (d) modification of the algorithms in order to triangulate surfaces whose parametric function $S(u,v)$ is not an injection.

REFERENCES

1. Salim S. Abi-Ezzi and Srikanth Subramaniam, *Fast dynamic tessellation of trimmed nurbs surfaced*, Computer Graphics Forum **13** (1994), no. "Issue 3", 107–126.
2. Marco Attene, Bianca Falcidieno, Michela Spagnuolo, and Geoff Wyvill, *A mapping-independent primitive for the triangulation of parametric surfaces*, Graph. Models **65** (2003), no. 5, 260–273.
3. ______, *A mapping-independent primitive for the triangulation of parametric surfaces*, 2003.

4. M. Botsch and L. Kobbelt, *A robust procedure to eliminate degenerate faces from triangle meshes*, Vision, Modeling and Visualization (VMV01), Stuttgart, Germany, November 21 - 23, 2001, 2001.
5. Wonjoon Cho, Nicholas M. Patrikalakis, and Jaime Peraire, *Approximate development of trimmed patches for surface tessellation*, Computer Aided Design **30** (1998), no. 14, 1077–1087.
6. H. Date, S. Kanai, T. Kishinami, I. Nishigaki, and T. Dohi, *High quality and property controlled finite element mesh generation from triangular meshes using multi-resolution technique*, Journal of Computing and Information Science in Engineering **5** (2005), no. 4, 266–276.
7. Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf, *Computational geometry*, Springer-Verlag, 2000, ISBN 3-540-65620-0.
8. M.P. DoCarmo, *Differential geometry of curves and surfaces*, Prentice-Hall, 1976.
9. R. Klein, A. Schilling, and W. Strasser, *Illumination dependent refinement of multiresolution meshes*, Proceedings of Computer Graphics International (CGI '98) (Los Alamitos, CA), IEEE Computer Society Press, 1998, pp. 680–687.
10. Erwin Kreyszig, *Differential geometry*, Dover Publications Inc, 1991, ISBN-10: 0486667219, ISBN-13: 978-0486667218.
11. Giuseppe Patan, Michela Spagnuolo, and Bianca Falcidieno, *Para-graph: Graph-based parameterization of triangle meshes with arbitrary genus*, Computer Graphics Forum **23(4)** (2004), 783–797.
12. O. Ruiz and S. Pena, *Aspect ratio- and size-controlled patterned triangulations of parametric surfaces*, Ninth IASTED Intl. Conf. Computer Graphics and Imaging (Innsbruck, Austria), February 13-15 2007, ISBN Book 978-0-88986-644-7, ISBN CD 978-0-88986-645-4, pp. 1–6.
13. J. Shewchuk, *Delaunay refinement algorithms for triangular mesh generation*, Computational Geometry: Theory and Applications **22** (2002), no. 1–3, 86–95, citeseer.ist.psu.edu / shewchuk01delaunay.html.
14. Jonathan Richard Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*, Applied Computational Geometry: Towards Geometric Engineering (Ming C. Lin and Dinesh Manocha, eds.), Lecture Notes in Computer Science, vol. 1148, Springer-Verlag, May 1996, From the First ACM Workshop on Applied Computational Geometry, pp. 203–222.
15. ______, *General-dimensional constrained delaunay and constrained regular triangulations i: Combinatorial properties*, Tech. report, Department of Electrical Engineering and Computer Sciences University of California at Berkeley, Berkeley, CA 94720, December 2005, To appear in Discrete & Computational Geometry.
16. C. Shu and P. Boulanger, *Triangulating trimmed nurbs surfaces*, International Conference on Curves and Surfaces, Saint-Malo, France, May 2000, pp. 381 – 388.
17. Michael Spivak, *Comprehensive introduction to differential geometry*, 1999, ISBN-10: 0914098705, ISBN-13: 9780914098706.
18. Wolfgang A. G. Stöger and Gerhard Kurka, *Watertight tessellation of b-rep nurbs cad-models using connectivity information*, Proceedings of the International Conference on Imaging Science, Systems and Technology, CISST '03, June 23 - 26, 2003, Las Vegas, Nevada, USA, Volume 2, eds. Hamid R. Arabnia and Youngsong Mun. CSREA Press, 2003.

CAD CAM CAE LABORATORY, EAFIT UNIVERSITY, MEDELLIN, COLOMBIA
E-mail address: {oruiz,jcongote,ccadavid,jlalinde}@eafit.edu.co
URL: http://www1.eafit.edu.co/cadcamcae/