High rate compression of CAD meshes based on subdivision inversion

Compression de maillages CAO basée sur la subdivision inverse

Guillaume Lavoué, Florent Dupont and Atilla Baskurt

LIRIS UMR 5205 CNRS
43, Bd du 11 novembre, 69622 Villeurbanne Cedex, France
phone: + (33) 4 72 44 83 95, fax: + (33) 4 72 43 15 36, email: glavoue@liris.cnrs.fr
web: http://liris.cnrs.fr/guillaume.lavoue

ABSTRACT. *In this paper we present a new framework, based on subdivision surface approximation, for efficient compression and coding of 3D models represented by polygonal meshes. Our algorithm fits the input 3D model with a piecewise smooth subdivision surface represented by a coarse control polyhedron, near optimal in terms of control points number and connectivity. Our algorithm, which remains independent of the connectivity of the input mesh, is particularly suited for meshes issued from mechanical or CAD parts. The found subdivision control polyhedron is much more compact than the original mesh and visually represents the same shape after several subdivision steps, without artifacts or cracks, like traditional lossy compression schemes. This control polyhedron is then encoded specifically to give the final compressed stream. Experiments conducted on several CAD models have proven the coherency and the efficiency of our algorithm, compared with existing methods.*

KEYWORDS: *CAD mesh, Compression, Visualization, Approximation, Subdivision surface.*

RÉSUMÉ. *Nous présentons dans cet article, une approche, basée sur une approximation par surfaces de subdivision, pour la compression et le codage de modèles 3D représentés par des maillages polygonaux. Notre algorithme approxime le modèle 3D par une surface de subdivision lisse par morceaux, représentée par un polyèdre de contrôle grossier optimisé en termes de nombre de points de contrôle et de connectivité. Notre algorithme, qui est indépendant de la connectivité du maillage d'origine, est particulièrement adapté aux maillages issus de pièces mécaniques ou CAO. Le polyèdre de contrôle obtenu est beaucoup plus compact que le maillage d'origine et représente visuellement la même forme après plusieurs itérations de subdivision, sans artefacts ou discontinuités comme celles introduites par la plupart des méthodes de compression avec pertes. Ce polyèdre de contrôle est ensuite codé spécifiquement pour donner le flux compressé final. Des expériences menées sur plusieurs modèles CAO ont prouvé la cohérence et l'efficacité de notre algorithme en comparaison d'autres méthodes existantes.*

MOTS-CLÉS : *Maillage CAO, Compression, Visualisation, Approximation, Surface de subdivision.*

I. Introduction and context

Advances in computer speed, memory capacity, and hardware graphics acceleration have highly increased the amount of three-dimensional models being manipulated, visualized and transmitted over the Internet. In this context, the need for efficient tools to retrieve, protect or reduce the storage of this 3D content, mostly represented by polygonal meshes, becomes even more acute. The context of our work is the SEMANTIC-3D project (http://www.semantic-3d.net), supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications), which is thoroughly within the scope of these problematics. This project focuses on the tools and methods required to implement new operational services for retrieving 3D content through the Web and communicating objects. It aims at developing an industrial application prototype: An information and communication system for remote access and assistance, interconnecting originators (mechanical part designers), nomadic users (e.g. automotive industry technicians) and a central 3D data server. Accordingly, one of the principal issues is the transmission of 3D mechanical models through low bandwidth channels in a visualization objective on various terminals. The 3D model database to handle comes from the car manufacturer Renault, and contains thousands of quite irregular polygonal meshes representing CAD parts. Thus an efficient compression tool is needed to reduce the amount of data carried by this 3D content, knowing that the original NURBS information is not available.

Many efficient techniques have been developed for encoding polygonal meshes [1,2,3] but fundamentally, this representation remains very heavy in terms of amount of data (a large points set, on top of the connectivity has to be encoded). Moreover, lossy compression schemes like wavelet based ones [4,5] produce artifacts, visually damaging for piecewise smooth mechanical objects. Other models exist to represent a 3D shape: NURBS surfaces or subdivision surfaces. These models are much more compact. A subdivision surface is a smooth (or piecewise smooth) surface defined as the limit surface generated by an infinite number of refinement operations using a subdivision rule on an input coarse control mesh. Hence, it can model a smooth surface of arbitrary topology (contrary to a NURBS model which needs a parametric domain) while keeping a compact storage and a simple representation (a polygonal mesh). Moreover it can be easily displayed to any resolution. Subdivision surfaces are now widely used for 3D imaging and have been integrated to the MPEG4 standard [6].

For all these reasons, we have developed a new algorithm, based on subdivision surface fitting for efficiently compressing 3D meshes, for low bandwidth transmission and storage. The 3D models are first approximated by a piecewise smooth subdivision surfaces, associated with control polyhedrons which are then encoded specifically to give the compressed bit streams. Hence the 3D model, once approximated, will be transmitted in the form of an encoded coarse polyhedron and, at the reception, displayed to any resolution, according to the terminal capacity, by iterative subdivisions. Note that this decompression process is very simple and therefore adapted for mobile terminals. Figure 1 shows this application scheme. Section II details the related work about mesh compression and subdivision surface fitting, while subdivision surfaces and the overview of our method are presented in section III. Sections IV, V and VI detail the three steps of our subdivision surface fitting approach: the decomposition of the model into surface patch, the patch boundaries approximation and the local subdivision surface creations. Finally section VII presents the final control mesh construction and encoding, and the results of our experiments.

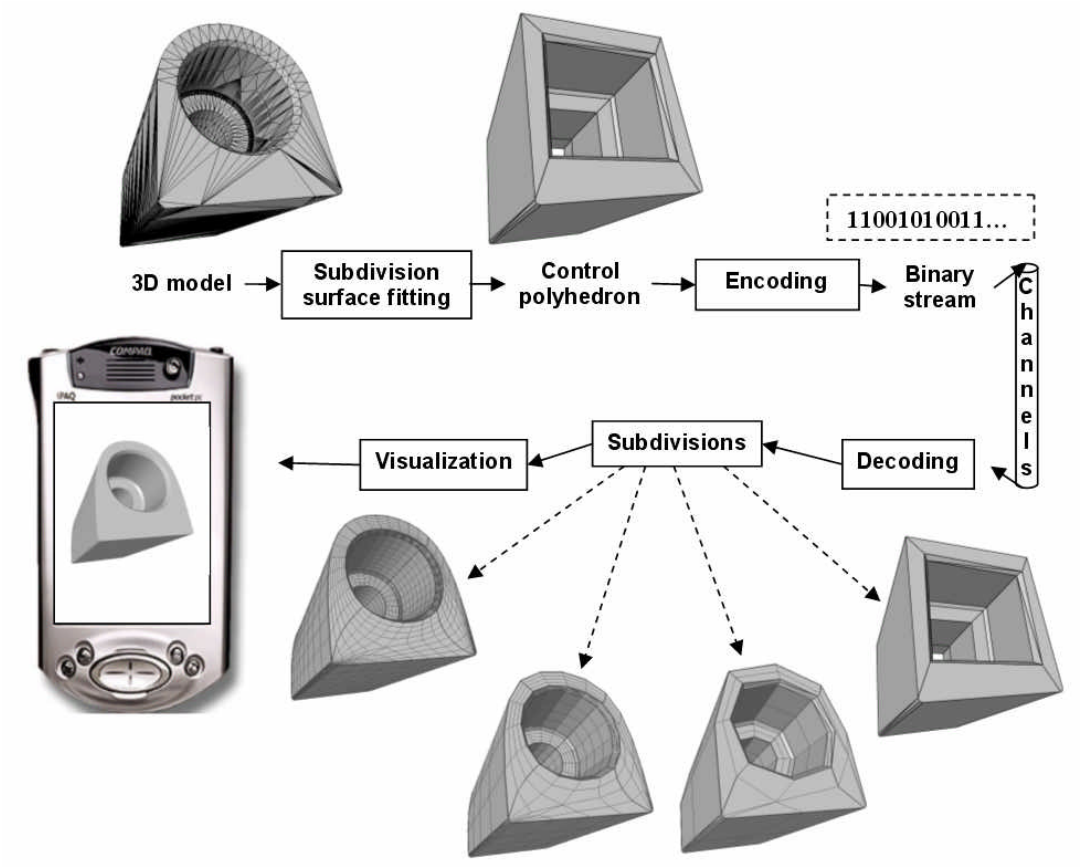

Figure 1: *Application of our subdivision based compression scheme.*

Figure 1: *Application de notre algorithme de compression basé sur la subdivision.*

II. Related work

II.1. Mesh compression

A lot of work has been done about polygonal mesh compression. A good review can be found in [7]. This representation contains two kinds of information: *geometry* and *connectivity*, the first describing coordinates of the vertices in the 3D space, and the later describing how to connect these positions. The connectivity graph is often encoded using a region growing approach based on faces [2], edges [3] or vertices [1]. Others techniques consider progressive approaches which encode a base mesh and then vertex insertion operations [8]. Fewer efforts have been done about geometry compression which is often simply performed by predictive coding and quantization. Other researches have put more efforts on geometry driven mesh coding, using wavelets [4,5] or spectral compression [9]. On the whole, better mesh compression methods give between 1 and 2 bytes per vertex; although this represents an excellent result, the output bit stream remains large for complex objects because of the high number of vertices to encode. Moreover lossy compression schemes [4,5,8,9] often produce artifacts, visually damaging for smooth mechanical objects. That is why we have chosen to approximate input meshes with subdivision surfaces (see Figure 1), of which control polyhedrons should contain much lesser faces to store or transmit,

knowing that after several refinement steps, the subdivision surface will visually represents the shape of the original mesh (of which original connectivity will not be kept).

II.2. Subdivision surface approximation

Several methods already exist for subdivision surface fitting, most of them take as input a dense mesh, simplify it to obtain a base coarse control mesh and then displace the control points (geometry optimization) to fit the target surface. Lee et al. [10], Ma et al. [11] and Mongkolnam et al. [12] use the Quadric Error Metrics from Garland and Heckbert [13] for simplification. Kanai [14] uses a similar decimation algorithm which directly minimizes the error between the original mesh and the subdivided simplified mesh. These simplification based approaches allow to easily extract a control mesh with the same topology than the target object, however, the control mesh connectivity strongly depends on the input mesh and therefore can give quite bad results if the input mesh is very irregular, which is the case for our CAD models. Hence, in our algorithm, in order to remain independent of the original connectivity, we first decompose the object into surface patches, and then we use the boundaries of the patches and the curvature information to transmit the topology of the target object to our control polyhedron. Algorithms from Suzuki[15], or Jeong and Kim [16] also remain independent of the target mesh, by iteratively subdividing and shrinking an initial control mesh toward the target surface. Unfortunately these methods fail to capture local characteristics for complex target surfaces.

Once a coarse control mesh has been constructed, then the geometry has to be optimized by moving control points to match the subdivision surface with the target model. Lee et al. [10] and Hoppe et al. [17] sample a set of points from the original mesh and minimize a quadratic error to the subdivision surface. Suzuki et al. [15] propose a faster approach, also used in [16] and [12]: the position of the control points is optimized, only by reducing the distance between their limit positions and the target surface. Hence only subsets of the surfaces are involved on the fitting procedure, thus results are not so precise and may produce oscillations. Ma et al. [11] consider the minimization of the distances from vertices of the subdivision surface after several refinements, to the target mesh; our algorithm follows this framework while using not a point to point distance minimization, but a point to surface minimization, by using the local quadratic approximants introduced by Pottmann and Leopoldseder [18]. This algorithm allows a more accurate and rapid convergence.

To our knowledge, the optimality in terms of control point number and connectivity represents a minor problematic in the existing algorithms but seems particularly relevant for mechanical or CAD objects. Only Hoppe et al. [17] optimize the connectivity (but not the number of control points) by trying to collapse, split, or swap each edge of the control polyhedron. Their algorithm produces high quality models but need of course an extensive computing time. Our algorithm optimizes the connectivity of the control mesh by analyzing curvature directions of the target surface, which reflect the natural parameterization of the object. The number of control points is also optimized by enriching iteratively the control polyhedron according to the error distribution. Moreover this approach allows to directly control the approximation error, whereas simplification based methods [10,11,12,14] indirectly control the error by modifying the decimation level.

III. Overview of our algorithm and orientations

III.1. Overview

Our framework for compression of 3D models is the following: Firstly the target 3D objects are segmented into surface patches (see Section IV), of which boundaries are extracted. Secondly, the network of boundaries is approximated with piecewise smooth subdivision curves (see Section V), this step provides a network of control polygons. Thirdly, for each patch a local approximating subdivision surface is created, using the subdivision control polygons representing its boundaries (see Section VI). Finally, the control mesh defining the whole surface is created assembling every local control meshes, and encoded specifically to give the output bit stream (see Section VII). These steps are summarized and illustrated on figure 2.

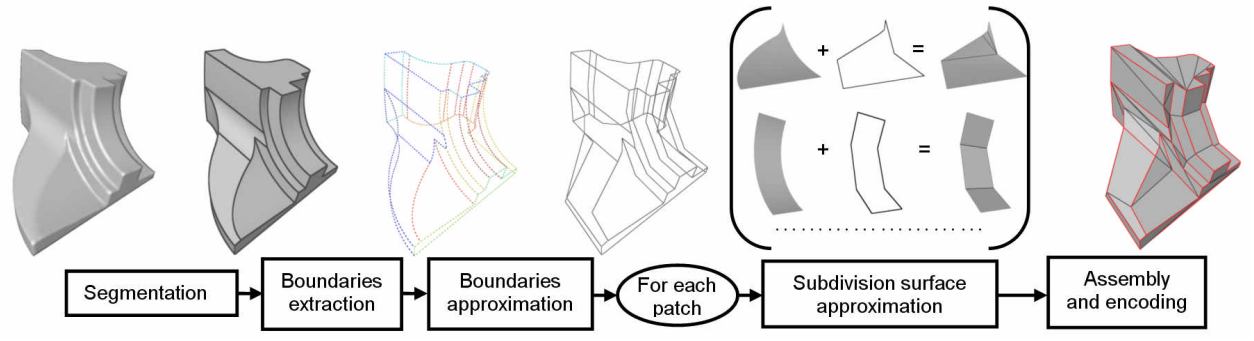

Figure 2: *Overview of our compression framework.*

Figure 2: *Les différentes étapes de notre algorithme de compression.*

III.2. Subdivision surface presentation

The basic idea of subdivision is to define a smooth shape from a coarse polyhedron by repeatedly and infinitely adding new vertices and edges according to certain subdivision rules. Doo and Sabin [19], and Catmull and Clark [20] first introduced subdivision schemes based on quadrilateral control meshes. Their schemes respectively generalized bi-quadratic and bi-cubic tensor product B-splines [21]. Today, many subdivision schemes have been developed, based on quadrilateral [22] or triangular meshes [23,24]. Moreover special rules have been introduced by Hoppe et al. [17] to handle *sharp* edges. Subdivision surfaces offer many benefits: Firstly, they can be generated from arbitrary meshes (arbitrary topology), this implies no need of trimming curves (which are necessary for NURBS). Secondly they can be generated at any level of detail, according to the terminal capacity for instance. And thirdly, subdivision surfaces are at least C1 continuous (except around *sharp* edges of course).

Within our approximation framework, we have to choose a subdivision scheme, among all these existing rules. For a given surface to approximate, the choice of the appropriate subdivision scheme is critical. Indeed, even if in theory any triangle can be cut into quads or any quad can be tessellated into triangles, results are not equivalent. The nature of the control polyhedron (quads or triangles) strongly influences the shape and the parameterization of the resulting subdivision surface. The body of the cylinder, for instance, is much more naturally parameterized by quads than by triangles. These reasons have led us to choose the hybrid quad/triangle scheme developed by Stam and Loop [25]. This scheme reproduces Catmull-Clark on quad regions and Loop on

triangle regions. At each subdivision step, the base mesh is firstly linearly subdivided: Each edge is splitted into two, each triangle into four and each quad into four (see Figure 2). Secondly, each vertex is replaced by a linear combination of itself and its direct neighbors. When a vertex is entirely surrounded by triangles or quads we use smoothing masks of Figure 3.a and Figure 3.b and otherwise we use the mask from Figure 3.c, which depends on the numbers of edges n_e and quads n_q surrounding the vertex.

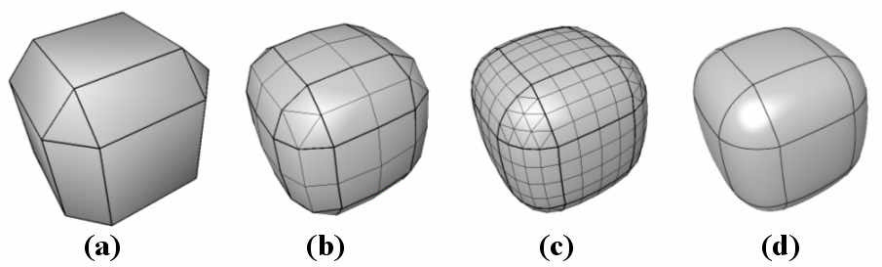

Figure 3: *Example of quad/triangle subdivision. (a) Control mesh, (b,c) One and two subdivision steps, (d) Limit surface.*

Figure 3: *Exemple de subdivision quad/triangle. (a) Maillage de contrôle, (b,c) Une et deux itérations de subdivision, (d) Surface limite.*

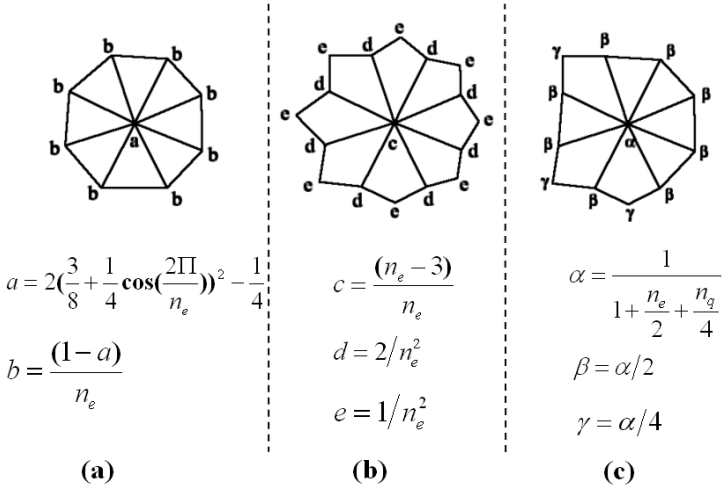

Figure 4: *Smoothing masks for Loop (a), Catmull-Clark (b) and the quad-triangle scheme (c) (extracted from [25]).*

Figure 4: *Masque de lissage pour Loop (a), Catmull-Clark (b) et le schéma quad/triangle (c) (extraits de [25]).*

III.3. The approximate squared distance

The subdivision curve approximation of the boundaries (see Section V), as the subdivision surface optimization (see Section VI), requires a convergence process. The purpose, starting from an initial surface (resp. curve) is to fit this surface to the target data by displacing iteratively the control points by minimizing an energy term. This optimization problem ties up with the smooth parametric curve and surface approximation problematic. Several algorithms exist for this

purpose concerning curves [26,27,28,29] or surfaces [30,31]. They are mostly based on a data parameterization which is very complex to optimize. Other approaches [32,33] construct a regular grid on the data to overcome this parameterization problem, but these techniques are not adapted for subdivision surfaces which do not rely on a parametric formulation. Hence, we have chosen to generalize the Active B-Spline approach from Pottmann and Leopoldseder [18] which is based on the minimization of local approximate squared distances from the target data and thus does not require parameterization. We have extended this method, which has proven to converge much faster [18,34] than traditional ones, for subdivision curves and surfaces. Their principal contribution is the definition of local approximants of the squared distance from a point to a surface (resp. curve). Thus the minimization of this point to surface (resp. curve) distance is much faster than traditional point to point distance. The local approximant of point to surface quadratic distance is defined as follows: Considering a smooth surface F, we can define at each point t_0, a Cartesian system (e_1,e_2,e_3) whose first two vectors e_1,e_2 are the principal curvature directions and e_3 is the normal vector. Considering this frame, the local quadratic approximant $F_d(p)$ of the squared distance of a point p at $(0,0,d)$ to the surface F is given by [18]:

$$F_d(x_1,x_2,x_3) = \frac{d}{d+r_1}x_1^2 + \frac{d}{d+r_2}x_2^2 + x_3^2 \qquad (1)$$

where x_1, x_2 and x_3 are the coordinates of p with respect to the frame (e_1,e_2,e_3) and $?_1$ (resp. $?_2$) is the curvature radius at $F(t_0)$, corresponding to the curvature direction e_1 (resp. e_2).
The local distance approximant from a point to a 3D curve is similar, the reader may refer to [18] for a detailed derivation and proof of these formula.

IV. Segmentation into patches

The problem of subdivision surface fitting is quite complex to resolve, particularly in our case, since we aim at remaining independent of the target mesh connectivity. Hence we have chosen to previously segment the object into near constant curvature surface patches. Benefits are numerous: The inverse subdivision problem is simplified whereas boundaries of the patches can be used to retrieve the topology. Moreover this decomposition may bring adaptivity to the fitting process or for the visualization (we can imagine, once we have the complete control polyhedron, subdivide only a desired part of the object). Our segmentation method, detailed in [35], is based on the curvature tensor field analysis and presents two distinct complementary steps: A region segmentation and a boundary rectification.

IV.1. A curvature based region segmentation

Firstly, a pre-processing step identifies *sharp* edges and vertices. This information is necessary for the continuation of the algorithm. Secondly the curvature tensor is calculated for each vertex; we have implemented the work of Cohen-Steiner et al. [36] based on the Normal Cycle. This curvature estimation procedure has proven to be the most efficient and stable among the others and gives very satisfying results even for bad tessellated objects. Then vertices are classified into clusters in the curvature space, according to their principal curvature values *Kmin* and *Kmax*. A region growing algorithm is then processed, assembling triangles into connected labeled regions according to vertex clusters. Finally a region adjacency graph is processed and reduced in order to merge similar regions according to several criteria (curvature similarity, size and common

perimeter). This algorithm extracts near constant curvature, topologically simple patches from the 3D-objects, and gives good qualitative results in terms of general shape and disposition of the segmented regions. Nevertheless, boundaries of the extracted patches are often jagged and present artifacts (see Figure 5.a). In this context, the objective of the boundary rectification process is to suppress these artifacts, in order to obtain clean and smooth boundaries corresponding to real natural boundaries of the object.

IV.2. A boundary rectification

Firstly, boundary edges are extracted from the previous region segmentation step (see Figure 5.c). Then for each of them, a *boundary score* is calculated which notifies a degree of correctness. To define this score, we consider the principal curvature directions *dmin* and *dmax* (see *dmin* in figure 5.b) which define the lines of curvature of the object. Indeed, they represent pivotal information in the geometry description [37]. The curvature tensors at the natural boundaries of an object tend to be anisotropic with a maximum direction following the curvature transition and therefore orthogonal to the boundaries. Thus the boundaries will tend to be parallel to the minimum curvature directions *dmin*. Therefore the *boundary score* of each edge directly depends on its angles with its vertices minimum curvature directions. According to this score, estimated correct boundary edges are marked (see Figure 5.c) and are used in a contour tracking algorithm to complete the final correct boundaries of the object (see Figure 5.d).

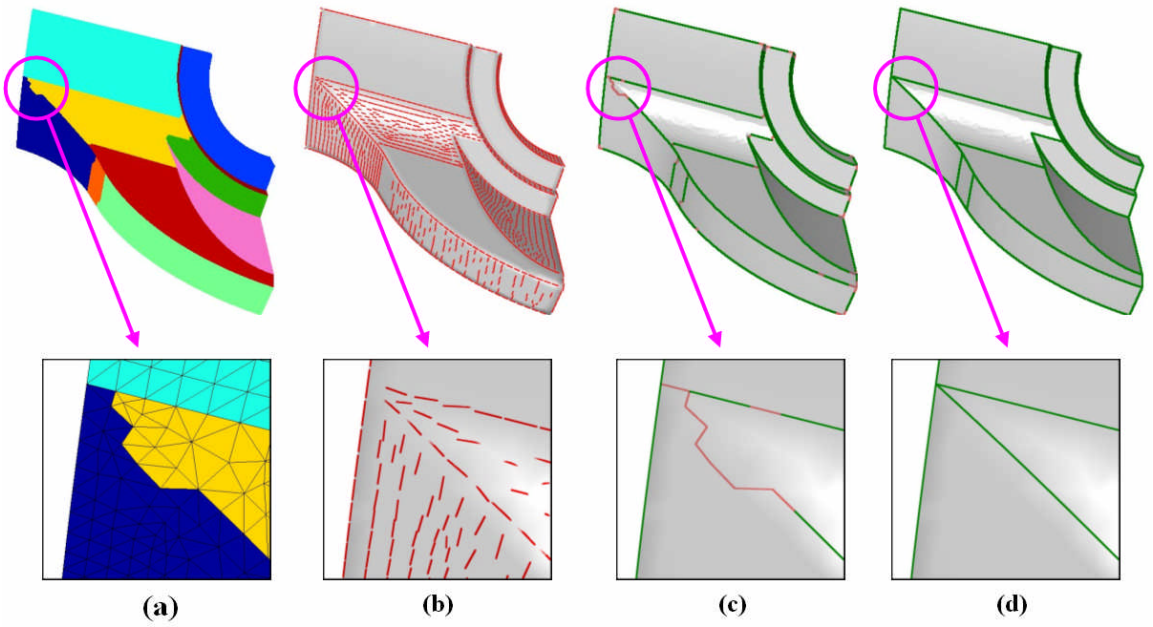

Figure 5: *The different steps of the Boundary Rectification for the "Fandisk" object with a zoom on an artifact correction. (a) Segmented object. (b) Minimum curvature directions. (c) Correct boundary edge extraction and marking. (d) Corrected boundaries after the contour tracking.*

Figure 5: *Les différentes étapes de la rectification de frontières pour l'objet Fandisk avec un zoom sur la correction d'un artefact. (a) Objet segmenté. (b) Directions de courbure minimum. (c) Extraction et marquage des arêtes frontières correctes. (d) Frontières corrigées après le suivi de contours.*

V. Boundaries approximation

Once the 3D object has been segmented, our algorithm approximates the network of patch boundaries with subdivision curves. Each piece of boundary is approximated with a subdivision curve, then every control polygons are assembled (junction points are tagged as *sharp*) to give a control polygon network (see Figure 2). Our purpose is to simplify the subdivision surface fitting algorithm; indeed, considering a surface patch, its boundary control polygons can then be extracted from the network; according to subdivision properties, these control polygons will represent the boundaries of the control polyhedron of the approximating subdivision surface.

V.1. Subdivision curve presentation

A subdivision curve is created using iterative subdivisions of a control polygon. In this paper we use the subdivision rules defined for surfaces by Hoppe et al. [17] for the particular case of *sharp* or boundary edges: New vertices are inserted at the midpoints of the control segments and new positions P_i' for the control points P_i are computed using their old values and those of their two neighbors using the mask:

$$P_i' = \frac{1}{8}(P_{i-1} + 6 \times P_i + P_{i+1}) \qquad (2)$$

With these rules, the subdivision curve corresponds to a uniform cubic B-Spline, except for its end segments. We also consider specific rules (those defined by Hoppe [17] for *corner* vertices) to handle *sharp* parts and extremities:

$$P_i' = P_i \qquad (3)$$

This subdivision curve will coincide with the boundary generated by commonly used subdivision surface rules like Catmull-Clark [20], Loop [23] or the quad-triangle scheme from Stam and Loop [25]. An example of subdivision curve is presented in Figure 6.

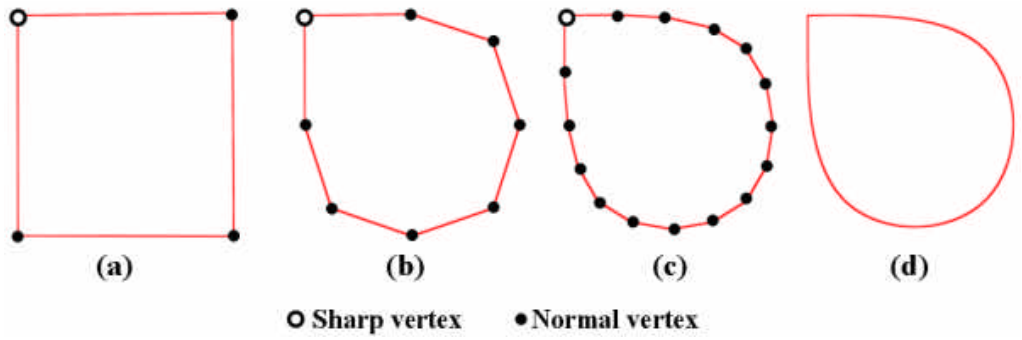

Figure 6: *Example of subdivision curve with one sharp vertex. (a) Control polygon. (b,c) 2 iterations of subdivision. (d) Limit curve.*

Figure 6: *Exemple de courbe de subdivision avec un sommet vif. (a) Polygone de contrôle. (b,c) 2 itérations de subdivision. (d) Courbe limite.*

V.2. Approximation algorithm

Our subdivision curve approximation algorithm, detailed in [38], presents two distinct steps: An initialization of the subdivision curve, analyzing curvature distribution of the target curve, and an optimization which moves and adds control points by minimizing a global distance to the target.

V.2.1. Subdivision curve initialization

Our subdivision curve represents a uniform cubic B-Spline curve except for its end segments, therefore except at its ends, the curve is composed with polynomial curve segments S_i. We have studied the behavior of the curvature on such a segment, in order to make the connection between the optimal number of control points and the curvature of the target curve.

Theorem:
Considering a uniform cubic B-Spline segment, local curvature maxima are necessarily located at the extremities.

***Proof**:*
For a cubic B-Spline segment S_i, the curvature C(u), at each parameter, is defined by $C(u) = \left\| \ddot{S}_i(u) \right\|$. The second derivative of a cubic B-spline is a linear B-spline. Moreover the largest norm value of a line segment occurs at one of its endpoints. Hence, curvature maxima are necessarily located at the extremities.

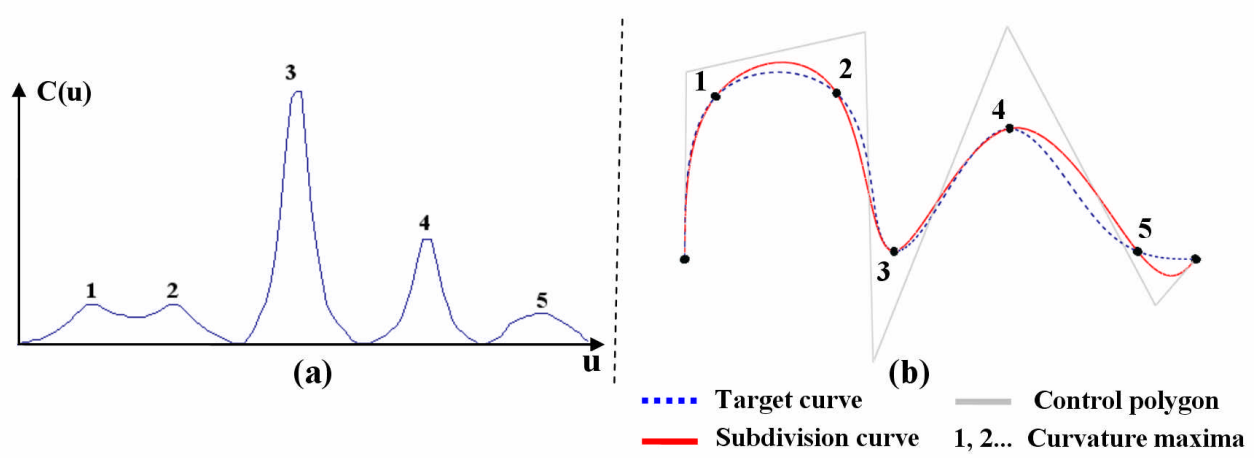

***Figure 7:** Example of initial control point processing. (a) Curvature variation over the target curve. (b) Corresponding maxima and initial subdivision curve with the associated control polygon.*

***Figure 7:** Exemple de détermination des points de contrôle initiaux. (a) Variation de la courbure de la courbe cible. (b) Maxima correspondants et courbe initiale avec son polygone de contrôle.*

According to this theorem, a local maximum of curvature located over the target curve is associated with the extremity of a B-Spline segment and therefore there is necessary at least one associated control point whose limit position is at the extremum. So, for *n* local curvature maxima we can affirm that at least *n* initial control points are needed. The placement of the *n*

control points is determined with a linear system. Indeed, for a subdivision curve, the limit position P_i' of a control point P_i can be processed according to its neighbors:

$$P_i' = \frac{1}{6}(P_{i-1} + 4 \times P_i + P_{i+1}) \quad (4)$$

Since we know that these limit positions must coincide with the local curvature maxima, we obtain the linear system. Figure 7 shows an example of this initialization process.

V.2.2. Subdivision curve optimization

Once the initial subdivision curve has been processed, the optimization algorithm fits this curve to the target data by displacing iteratively the control points P_i and adding new ones. We have extended the method from Pottmann et al [18] for subdivision rules (see Section III.3). The optimization process is the following, at each iteration several sample points S_k are chosen on the subdivision curve, and the associated projections O_k are calculated on the target curve. In our case, sample points are the vertices of the subdivision curve at a finer level, after application of several steps of subdivision. Thus sample points S_k can be computed as linear combinations of the control points Pi (see Section V.1). Then for each S_k the local quadratic approximant F_d^k of the squared distance function of S_k to the target curve, is expressed (see Section III.3). And finally new positions of control points are processed by minimizing the sum of the local quadratic approximants (least square method). The convergence of this algorithm is very fast, figure 8 presents an example; at the second iteration the target curve is perfectly fitted. If after several iterations, the approximation error remains high, then new control points are inserted at the maximum error position. At the end of the algorithm, each piece of boundary from the boundary network is perfectly fitted with a subdivision curve.

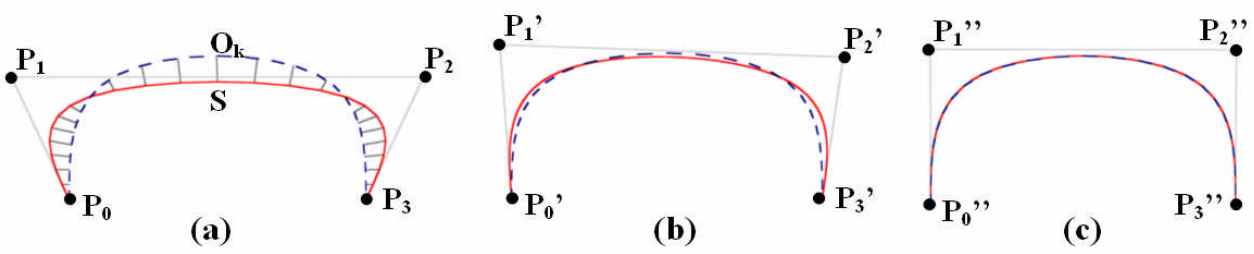

Figure 8: *Example of the optimization procedure, (a) Initial subdivision curve, (b,c) Resulting curve after respectively 1 and 2 optimization iterations.*

Figure 8: *Exemple de la procédure d'optimisation. (a) Courbe de subdivision initiale. (b,c) Courbes obtenues après respectivement 1 et 2 itérations d'optimisation.*

VI. Subdivision surface fitting

Once the boundary network has been approximated by subdivision curves, we construct, for each patch, thanks to this network, a local approximating subdivision surface. The algorithm, detailed in [39], is the following: For each patch, the corresponding control polygons representing its boundaries are extracted from the network. Then, an initial subdivision surface is created by optimally linking the boundary control points with respect to the lines of curvature of the target

surface. Finally, for each patch, the initial subdivision surface is optimized by iteratively moving control points and enriching regions according to the error distribution.

VI.1. Local subdivision surface initialization

VI.1.1. Principle

Considering a surface patch, once the control polygon representing its boundary has been extracted, the purpose is to create edges and facets by connecting the control points in such a way that the corresponding created initial subdivision surface is the better approximation of the target surface for these given control points, regarding to the resulting error. For this purpose, we consider the lines of curvature of the original surface, represented by local directions of minimum and maximum curvature. Control lines of a subdivision surface are strongly linked to the lines of curvature. Indeed the topology of a control polyhedron will strongly influence the geometry information of the associated limit surface, which is also carried by lines of curvature [37]. This coherency between control lines and lines of curvature is shown in the example on Figure 9.

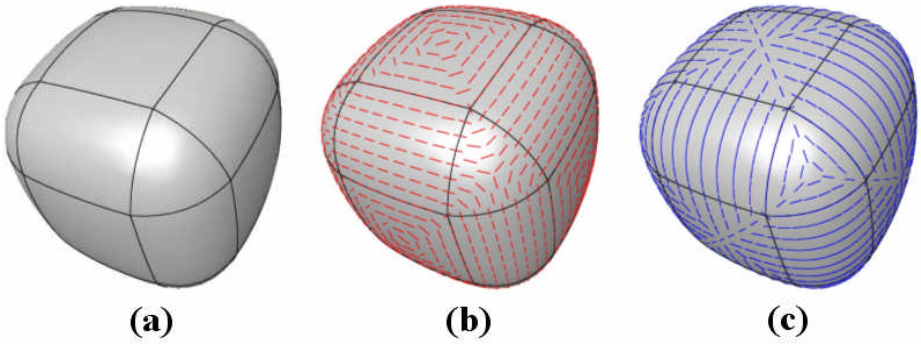

Figure 9: *The coherency between control lines (a), minimum (b) and maximum (c) directions of curvatures.*

Figure 9: *La cohérence entre les lignes de contrôle (a) et les directions de courbure minimum (b) et maximum (c).*

Thus, for each couple of control points from the boundary control polygon, a Coherency Score *SC* is calculated, taking into account the coherency of the corresponding potential control edge with the lines of curvature of the corresponding area on the target surface. The mechanism is illustrated on Figure 10: For each potential edge *E*, we consider its vertices P_0, P_1 and their respective limit positions P_0^∞, P_1^∞. Then we calculate the pseudo geodesic path between these limit positions, to simulate the control line, by applying the Dijkstra algorithm on the vertices of the original surface. Finally we consider the curvature tensors of the *n* vertices V_i of this path, and particularly their curvature directions. The coherency score *SC* for this potential edge *E* is:

$$SC(E) = \frac{min(\sum_{i=1}^{n} q\ min_i, \sum_{i=1}^{n} q\ max_i)}{n} \qquad (5)$$

where $q\,min_i$ (resp. $q\,max_i$) is the angle between the minimum (resp. maximum) curvature direction of the vertex V_i and the segment $P_0^\infty\ P_1^\infty$. This score $SC\ \hat{I}\ [0,90]$ is homogeneous to an angle value in degrees.

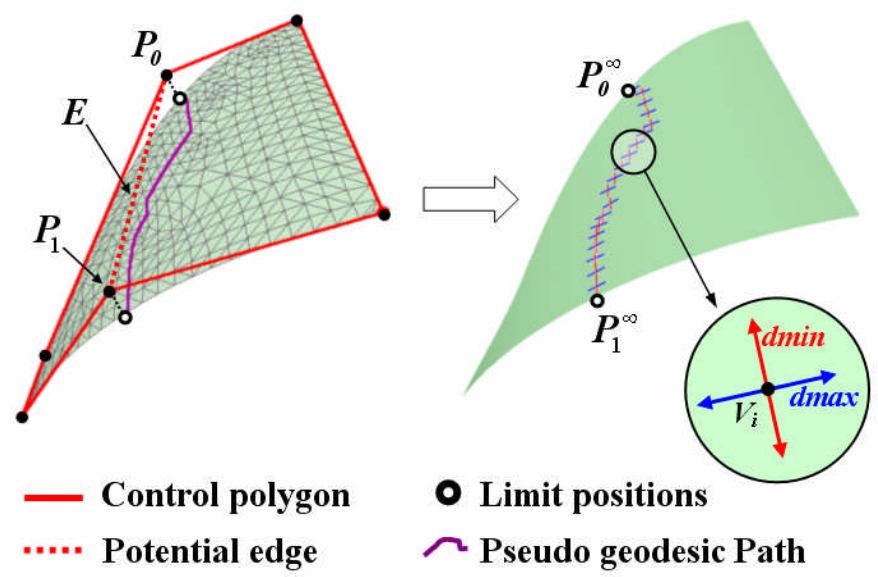

Figure 10: *Mechanism for edge score definition.*

Figure 10: *Mécanisme pour la définition du score des arêtes.*

VI.1.2. Algorithm

Our algorithm is the following: At each iteration, we consider the potential edge associated with the smallest score *SC* (dotted segments in Figure11.b) and we cut the boundary control polygon along this edge. This operation is repeated until it remains only plane polygons. Then for each of them, we check its convexity; if it is convex, we create a facet, and if not, we decompose it into convex parts, using the algorithm from Hertel and Mehlhorn [40]. By assembling created facets we obtain our initial control polyhedron (see Figure 11.c) of which limit surface (see Figure 11.d) represents in most case a quite good approximation of the original surface (see Figure 11.a).

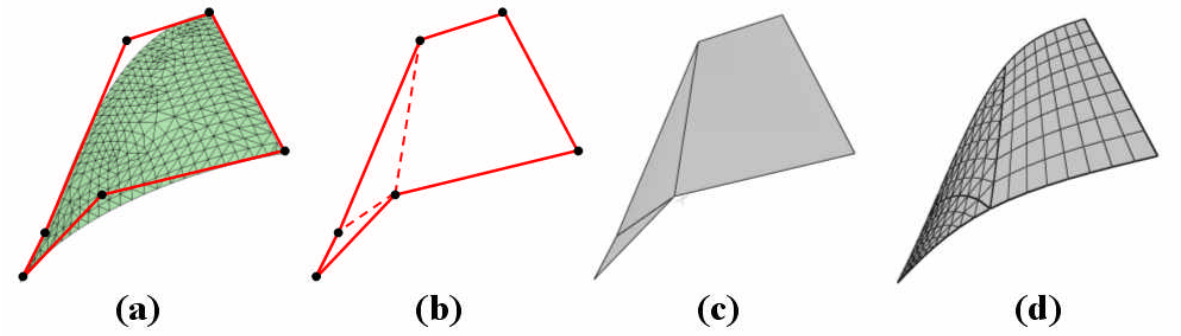

Figure11: *Example of local subdivision surface initialization.*

Figure11: *Exemple d'initialisation d'une surface de subdivision locale.*

VI.2. Local subdivision surface optimization

The initial subdivision surface often represents a sufficient approximation of the target surface patch, even if the initialization process considers first of all the boundary information. Indeed, owing to the curvature based segmentation step, surface patches are quite simple surfaces, of which most of the geometry information is carried by the boundary. However, in some cases some more control points may be needed to correctly approximate the target shape. Considering this purpose, we have defined two complementary mechanisms: An enrichment mechanism which adds points and optimizes the connectivity according to the error position and distribution, and a geometry optimization algorithm, generalizing Pottmann and Leopoldseder method [18] for the complex quad-triangle subdivision rules.

VI.2.1. Enrichment and connectivity optimization

In this section we present how to modify and optimize the connectivity of our control polyhedron. We have two mechanisms to consider: An enrichment of the mesh, consisting in the addition of new control points, and an optimization of the connectivity, insuring that, for a given set of control points, the associated connectivity (set of faces and edges) is the better possible regarding to the resulting error. This mechanism is quite complex to implement, therefore, since the connectivity has been optimized in the initialization step, we will just try to limit its departure. Hence we have integrated these two mechanisms into a single algorithm, which considers the error distribution to enrich precisely the polyhedron, while trying to keep a near optimal connectivity.

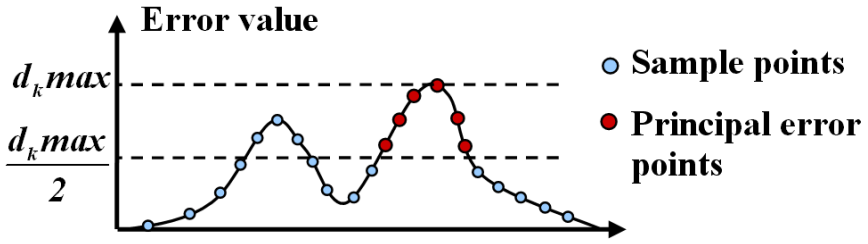

Figure 12: *Principal error field extraction (2D example).*

Figure 12: *Extraction du champ d'erreurs principal (exemple en 2D).*

Considering a target surface and a corresponding initial subdivision surface, the first step of this algorithm is the principal error field extraction. The goal is to extract not only the maximum error point but an area (a set of error points) corresponding to the error field in order to be able to analyze the error distribution. For this purpose we consider sample points S_k, on the subdivision surface and associated distances d_k to the corresponding projections on the target surface. Then, we extract and add to our error set, $S_k max$ corresponding to the maximum error $d_k max$, and every sample points corresponding to a similar error (we have fixed a threshold $T = d_k max/2$) and connected to an other point of the error point set. This extraction is shown for a 2D case in Figure 12. Once we have the principal error field, we study its dispersion to modify the control mesh. If several control faces F_k are concerned by the error field (they contain at least one error point) it means that the topology in this region is not correct, hence, we merge these faces and then add a point in the resulting face and connect it with its neighbors. Figure 13.a shows a target surface and Figure 13.b shows the initial subdivision surface with the corresponding error field (error

points are marked). Corresponding faces (Figure 13.c) have been merged, before adding a new control point (see Figure 13.d and e).

VI.2.2. Geometry optimization

For a given target surface and a given initial subdivision surface, this process aims at displacing control points by minimizing a global error over the whole surface. To achieve this purpose, we use a least square method based on the quadratic distance approximants defined by Pottmann and Leopoldseder [18] (see Section III.3). Our algorithm is the following:

- The curvature is calculated for each vertex of the target surface.
- Several sample points S_k are chosen on the subdivision surface, they correspond to vertices of the subdivided polyhedron at a finer level l_0. The associated footpoints (projections of the sample points on the target surface) are extracted. For each of them, we calculate the curvature tensor, by a linear interpolation of those of the surrounding vertices, using barycentric coordinates. This tensor allows us to construct the Frame (e_1,e_2,e_3) and the curvature radiuses $?_1$ and $?_2$, useful for the point to surface distance computation (see Equation 1). Sample points S_k can be computed as linear combinations of the initial control points P_i^0 (see Section III.2); they correspond to control points $P_i^{l_0}$ at the finer level l_0.

$$S_k = C_k(P_1^0, P_2^0, ..., P_n^0) \quad (5)$$

  The functionals C_k are determined using iterative multiplications of the l_0 subdivision matrices M_l associated with our subdivision rules, which give the new positions of controls points according to the old ones. These subdivision matrixes M_l are such as $P^l = M_l \times P^{l-1}$ with $P^l = [(P_1^l, P_2^l, ...P_n^l)]^T$. Thus the functionals C_k for the level l_0, are the lines of the matrix C such as:

$$C = \prod_{l=1}^{l_0} M_l \times L_{l_0} \quad (6)$$

  L_{l_0} is the limit matrix which gives the limit positions, proposed by Stam and Loop [25], of the considered control points at the level l_0.

- For all S_k, local quadratic approximants F_d^k of the squared distances to the target surface are expressed according to the frames (e_1,e_2,e_3) at the corresponding Footpoints. The minimization of their sum F gives the new positions of the control points P_i^0.

$$F = \sum_k F_d^k(S_k) = \sum_k F_d^k(C_k(P_1^0, P_2^0, ..., P_n^0)) \quad (7)$$

  The minimization of this quadratic function leads to the resolution of a linear squared system.

Concerning the choice of the number of sample points S_k, we have chosen $l_0=2$ refinements for all examples in this article. As for each refinement, the number of vertices will increase by a factor of at least four, the number of equations will be about sixteen times the number of

unknowns. That ensures a stable solution when solving equation (7) in the least squares sense. This algorithm provides a very fast convergence, which is critical since this geometry optimization is a computationally costly procedure.

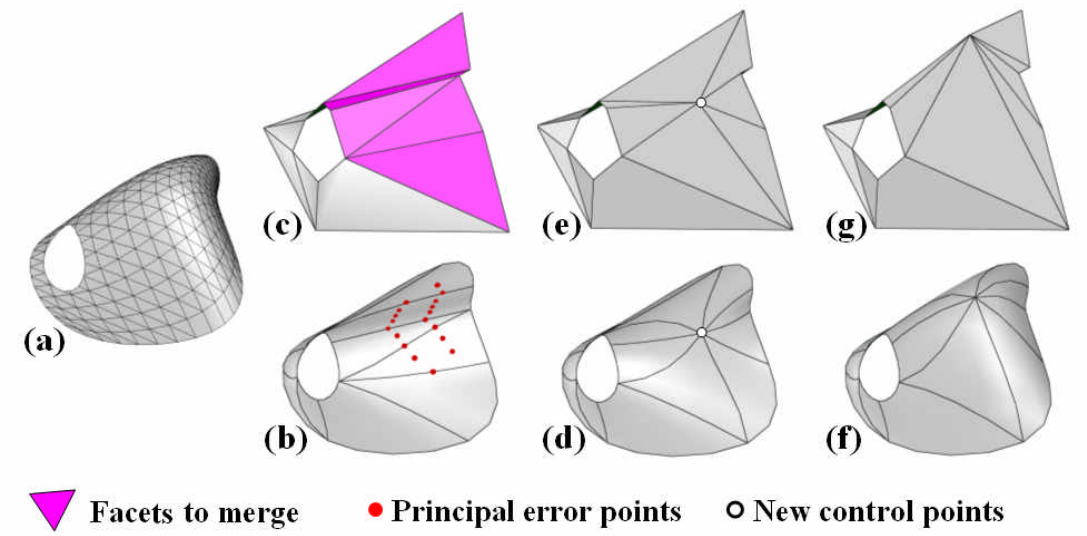

Figure 13: *Connectivity and geometry optimization example. (a) Original surface. Initial (b,c), enriched (d,e) and optimized (geometry) (f,g) subdivision surface.*

Figure 13: *Exemple d'optimisation de la connectivité et de la géométrie. (a) Surface originale. Surface de subdivision initiale (b,c), enrichie (d,e) et optimisée (géométrie) (f,g).*

VI.2.3. Whole optimization algorithm

Our algorithm for the optimization of local subdivision surfaces is the following:

Begin *Subdivision Surface Optimization*

While ($E>E_{Limit}$}

// E is the approximation error and E_{Limit} a threshold value.

While ($E>E_{Limit}$ *and* $m<m_0$}

// m is the geometry optimization iteration number and m_0 a maximum number.

Call the geometry optimization procedure (see Section VI.2.2). The subdivision surface is moved toward the target surface, by minimizing a sum of quadratic distances.

End While

If ($E>E_{Limit}$}

A new control point is inserted onto the subdivision surface according to the error distribution (see Section VI.2.1).

End If

End While

End *Subdivision Surface Optimization*

m_0 was fixed to 5, in order to limit the number of iterations for the geometry optimization, since its convergence is very fast (often 3 or 4 iterations) and seeing that this process remains computationally costly. Note that boundary control points are fixed, to insure that no crack will appear later, during the construction of the final whole control polyhedron containing every control meshes of the different patches.
Figure 13 shows the complete process. Boundaries of the target surface (see Figure 13.a) have been approximated and an initial subdivision surface has been constructed (see Figures 13.b and 13.c). The associated approximation L1 error is $E=30.7\times10^{-3}$. Then the error distribution is analyzed and corresponding faces are merged. A new point in inserted (see Figure 13.d and 13.e) and the surface geometry is optimized (3 iterations) (see Figure 13.f and 13.g). The final approximation error is $E=5.8\times10^{-3}$.

VII. Construction and coding of the final control polyhedron and results

VII.1 Encoding

Once each patch has been fitted with a subdivision surface, the final control polyhedron for the whole object is created by assembling local control polyhedrons while marking local boundary control edges as *sharp* (specific subdivision rules which respect sharpness of the edges). This control polyhedron containing triangles, quadrangles, higher order polygons and marked edges is then encoded. Concerning connectivity information, we have chosen to implement the Face Fixer [3] algorithm (see Section II.1) seeing that this encoding scheme is based on edges and allows to process arbitrary polygonal meshes and not just fully triangulated ones. In addition this scheme, which provides quite good compression rates, is able to encode easily face groupings which can be useful, in a perspective way, to transmit the segmentation results within the object. This algorithm encodes the connectivity graph by a list of n labels among k ($k\sim10$, depending on the maximum face degree), with n the number of edges. The corresponding bit stream is created using an arithmetic coder which achieves quite good results. Concerning geometry encoding, a 10 bit quantization is performed. Then we eliminate some coordinates, indeed, once the positions of three vertices of a planar face are known, we have to encode only 2 coordinates for the remaining vertices. Flags on the edges (*sharp* or not) are represented by a n sized binary vector, encoded by a run length algorithm. Thus the total size of the compressed stream is the sum of the connectivity (C), geometry (G) and flags (F) sizes (see examples in Figure 14, C, G and F are given in bytes).

VII.2 Results and discussions

Our compression scheme was tested on the mechanical database from Renault, these models are issued from CAD, and thus associated with highly irregular connectivity (see mesh examples on Figure 15). Figure 14 presents the results for the Fandisk mesh (Figure 14.a) and for several objects from Renault database. All these experiments were conducted on a PC, with a 2Ghz XEON bi-processor; processing times are between 5 and 10 seconds for the whole compression process (the decompression is instantaneous). The models have been translated and scaled in a bounding box of length equal to 1. Figure 14 shows initial objects (with patch boundaries), control polyhedrons and associated limit surfaces while detailing the number of vertices and faces of the original objects and of the corresponding control polyhedrons. Original and compressed

sizes, in bytes, are also highlighted. Control polyhedrons have widely less faces and vertices compared with initial surfaces and the approximation errors remain very low (limit surfaces are very close from original objects). Mean L1, L2 and maximum errors are shown on Table I, they are calculated between the original object and the subdivision surface after 4 refinement steps. Table I shows original binary sizes (*BS*), sizes of these binary files compressed with the Zip coder (*ZIP*) and associated compression rates (*ZIP CR*). Although the ZIP coder is lossless without any quantization, these values can be compared with compressed file sizes (*CS*) obtained with our compression algorithm, which achieves extremely high compression rates (*CR=BS/CS*).

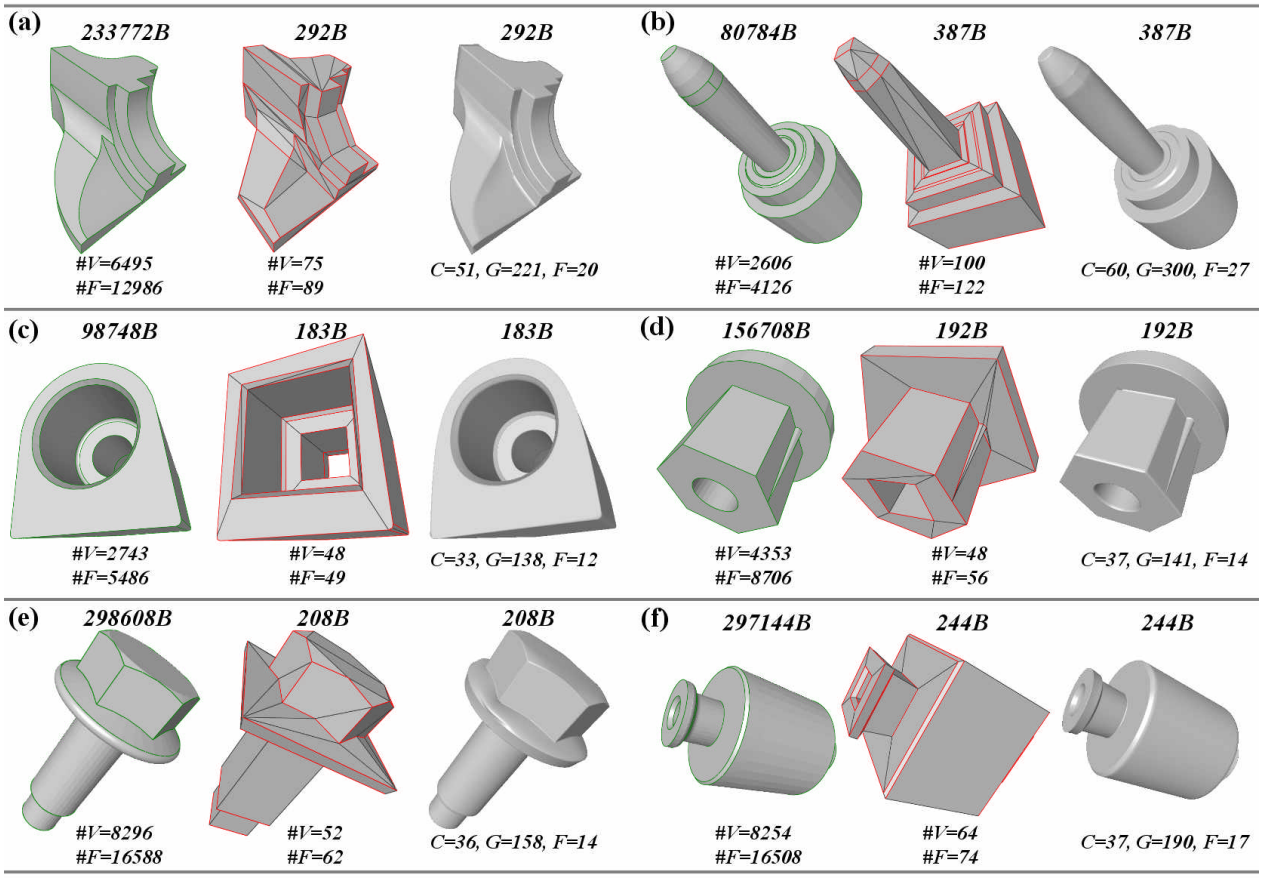

Figure 14: *Results of our fitting scheme for different mechanical parts. Initial objects (patch boundaries are marked), control polyhedrons, and limit surfaces.*

Figure 14: *Résultat de notre algorithme d'approximation pour différentes pièces mécaniques. Objets initiaux (les frontières des patchs sont marquées), polyèdres de contrôle et surfaces limites.*

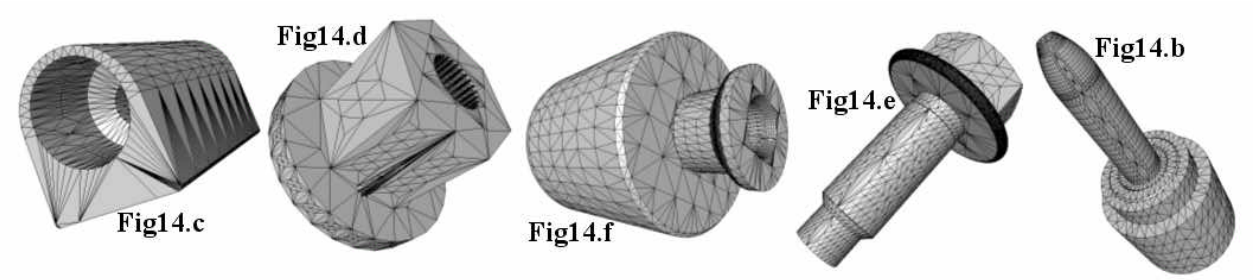

Figure 15: *Examples of mesh connectivity of our 3D model database and corresponding numbers on Figure 14.*

Figure 15: *Exemples de connectivité de notre base de modèles 3D et correspondance avec la figure 14.*

	BS (Bytes)	**ZIP (Bytes)**	**ZIP CR**	**CS (bytes)**	**CR**	**L1 Error (10^{-3})**	**L2 Error (10^{-3})**	**Max Error (10^{-3})**
Fig14.a	233 772	59 438	**3.93**	292	**801**	0.887	0.012	10.18
Fig14.b	80 784	12 014	**6.72**	387	**209**	0.953	0.012	33.09
Fig14.c	98 748	29 680	**3.33**	183	**540**	0.985	0.014	5.94
Fig14.d	156 708	16 849	**9.30**	192	**816**	0.765	0.011	7.31
Fig14.e	298 608	45 539	**6.56**	208	**1436**	2.588	0.043	21.66
Fig14.f	297 144	43 714	**6.80**	244	**1217**	1.766	0.021	12.01

Table I: *Original binary sizes (BS), Zipped binary sizes (ZIP) and associated rate (ZIP CR). Compressed size (CS) with our algorithm, associated compression rates (CR) and associated L1, L2 and maximum errors.*

Table I: *Tailles binaires originales (BS), tailles des fichiers binaires zippés (ZIP) et taux correspondants (ZIP CR). Taille des objets compressés (CS) avec notre algorithme, taux de compression correspondants (CR) et erreurs L1, L2 et maximum associées.*

Table II shows a comparison, for the Fandisk object, with different state of the art algorithms: Alliez and Desbrun progressive encoding [8] and the wavelets based algorithms from Khodakovsky et al. [4] and Valette and Prost [5]. Our algorithm achieves drastically better compression rates (~800), while keeping a low geometric error. Coders from Alliez and Valette are presented in their lossless versions, thus the geometric error is limited to the quantization error *QE* (a 10 bits quantization, like ours).

	Alliez at al. [8]	**Valette and Prost [5]**	**Khodakovsky et al. [4]**	**Our scheme**
Size (bytes)	14 075	10 603	6 063	292
CR	**17**	**22**	**39**	**801**
L2 Error (10^{-3})	QE	QE	0.045	0.012

Table II: *Compressed sizes, associated compression rates and L2 errors for several compression algorithms applied to the Fandisk object.*

Table II: *Tailles des fichiers compressés, taux de compression associés et erreurs L2 pour différents algorithmes de compression appliqués à l'objet Fandisk.*

Results are also particularly suited for our visualization task; indeed, resulting surfaces after subdivision are quite smooth and visually pleasant, without discontinuities or noise like those produced by lossy compression schemes like wavelet based ones for instance. Particularly, our algorithm, thanks to the segmentation step, preserves sharp features.

VIII. Conclusion

We have presented a new framework for compression and coding of 3D models. Our approach, particularly adapted for mechanical objects, is based on subdivision surface fitting. Our approximation algorithm aims at optimizing the connectivity and the control points number of the generated subdivision control polyhedron. After a segmentation step, the 3D object is decomposed into surface patches of which boundaries are approximated with subdivision curves which lead to initial local subdivision control polyhedrons by linking control points of the boundary control polygons. These edges are created with respect to the lines of curvature, to preserve the natural parameterization of the target surfaces. Local subdivision surfaces are then iteratively enriched and optimized until the approximation errors become correct. The final whole control polyhedron containing triangles, quadrangles, higher order polygons and *sharp* edges is then created by assembling local subdivision control polyhedrons, and encoded using an efficient edge based algorithm followed by an entropic coding for the connectivity and a 10 bit quantization for the geometry. Results show quite impressive compression rates compared with state of the art algorithms. Thanks to subdivision properties, at the decompression step, the object can be displayed at any resolution. Moreover limit surfaces are visually pleasant (at least C0 and piecewise C1), without artifacts or cracks, like traditional lossy compression schemes, while sharp features of the original models are preserved. Our method is effective for mechanical models since they present large constant curvature regions which are particularly adapted for subdivision inversion. On the other hand, our method is less suited for natural objects. Concerning perspectives, we plan to improve the connectivity optimization during mesh enrichments, by conducting a deeper analysis of the error dispersion. Finding a way to treat natural noisy objects is also of interest.

Acknowledgements

This work is supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the Semantic-3D national project (http://www.semantic-3d.net).

Bibliography

[1] TOUMA (C.), GOTSMAN (C.), Triangle mesh compression, *Graphic Interface Conference Proceedings*, pp. 26–34, 1998.
[2] GUMHOLD (S.), STRASSER (W.), Real time compression of triangle mesh connectivity, *ACM Siggraph Conference Proc.*, p. 133-140, 1998.
[3] ISENBURG (M.), SNOEYINK (J.), Face Fixer : Compressing Polygon Meshes with Properties, *ACM Siggraph Conference Proc.*, p. 263-270, 2001.
[4] KHODAKOVSKY (A.), SCHRODER (P.), SWELDENS (W.), Progressive Geometry Compression, *ACM Siggraph Conference Proc.*, p. 271-278, 2000.

[5] VALETTE (S.), PROST (R.), A Wavelet-Based Progressive Compression Scheme For Triangle Meshes : Wavemesh, *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no 2, p. 123-129, 2004.
[6] MPEG4, ISO/IEC 14496-16. Coding of Audio-Visual Objects : Animation Framework eXtension (AFX), 2002.
[7] GOTSMAN (C.), GUMHOLD (S.), KOBBELT(L.), Simplification and compression of 3d meshes, *European Summer School on Principles of Multiresolution in Geometric Modelling*, A. Iske, E. Quak, M., Floater (eds.), Springer, 2002.
[8] ALLIEZ (P.), DESBRUN (M.), Progressive Encoding for Lossless Transmission of 3D Meshes, *ACM Siggraph Conference Proc.*, p. 198-205, 2001.
[9] KARNI (Z.), GOTSMAN (C.), Spectral compression of mesh geometry, *ACM Siggraph Conference Proc.*, p. 279-286, 2000.
[10] LEE (A.), MORETON (H.), HOPPE (H.), Displaced subdivision surfaces, *ACM Siggraph Conference Proc.*, p. 85-94, 2002.
[11] MA (W.), MA (X.), TSO (S.), PAN (Z.), A direct approach for subdivision surface fitting from a dense triangle mesh, *Computer Aided Design*, vol. 36, no 6, p. 525-536, , 2004.
[12] MONGKOLNAM (P.), RAZDAN (A.), FARIN (G.), Reverse Engineering Using Loop Subdivision, *Computer-Aided Design & Applications*, vol. 1, pp. 619-626, 2004.
[13] GARLAND (M.), HECKBERT (P.), Surface simplification using quadric error metrics, *ACM Siggraph Conference Proc.*, p. 209-216, 1997.
[14] KANAI (T.), Meshtoss : Converting subdivision surfaces from dense meshes, *Vision, Modeling and Visualization Conf. Proc.*, p. 325-332, 2001.
[15] H. Suzuki, S. *Takeuchi, F. Kimura, T. Kanai* Subdivision surface fitting to a range of points, IEEE *Pacific graphics Proc.*, p. 158-167, 1999.
[16] JEONG (W.), KIM (C.), Direct reconstruction of displaced subdivision surface from unorganized points, *Journal of Graphical Models*, vol. 64, no. 2, pp. 78-93, 2002.
[17] HOPPE (H.), DEROSE (T.), DUCHAMP (T.), HALSTEAD (M.), JIN (H.), MCDONALD (J.), , J. Schweitzer, W. Stuetzle, Piecewise smooth surface reconstruction, *ACM Siggraph Conference Proc.*, p. 295-302, 1994.
[18] POTTMANN (H.), LEOPOLDSEDER (S.), A concept for parametric surface fitting which avoids the parametrization problem, *Computer Aided Geometric Design*, vol. 20, no 6, p. 343-362, 2003.
[19] DOO (D.), SABIN (M.), Behavior of recursive division surfaces near extraordinary points, *Computer Aided Design*, vol. 10, p. 356-360, 1978.
[20] CATMULL (E.), CLARK (J.), Recursively generated b-spline surfaces on arbitrary topological meshes, *Computer-Aided Design*, vol. 10, no 6, p. 350-355, 1978.
[21] FARIN (G.), *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1996.
[22] KOBBELT (L.), Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Computer Graphics Forum*, vol. 15, no 3, p. 409-120, 1996.
[23] LOOP (C.), Smooth subdivision surfaces based on triangles, Master's thesis, Utah University, 1987.
[24] DYN (N.), LEVIN (D.), GREGORY (A.), A butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics*, vol. 9, no 2, p. 160-169, 1990.
[25] STAM (J.), LOOP (C.), Quad/triangle subdivision, *Computer Graphics Forum*, vol. 22, no 1, p. 79-85, 2003.
[26] HOSCHEK(J.), Intrinsic parametrization for approximation, *Computer Aided Geometric Design*, vol. 5, no 1, p. 17-31, 1988.
[27] SPEER (T.), KUPPE (M.), HOSCHEK(J.), Global reparametrization for curve approximation, *Computer Aided Geometric Design*, vol. 15, no 9, p. 869-877, 1998.
[28] SAUX (E.), DANIEL (M.), Data reduction of polygonal curves using B-Splines, *Computer-Aided Design*, vol. 31, no 8, p. 507-515, 1999.
[29] SAUX (E.), DANIEL (M.), An improved hoschek intrinsic parametrization, *Computer Aided Geometric Design*, vol. 20, no 8,9, p. 513-521, 2003.

[30] MA (W.), KRUTH (J.), Parametrization of randomly measured points for the least squares fitting of b-spline curves and surfaces, *Computer Aided Design*, vol. 27, no 9, p. 663-675, 1995.
[31] ROGERS (D.), FOG (N.), Constrained b-spline curve and surface fitting, *Computer Aided Geometric Design*, vol. 21, no 10, p. 641-648, 1989.
[32] KRISHNAMURTHY (V.), LEVOY (M.), Fitting smooth surfaces to dense polygon meshes, *ACM Siggraph Conference Proc.*, p. 313-324, 1996.
[33] FORSEY (D.), BARTELS (R.), Surface fitting with hierarchical splines, *ACM Transactions on Graphics*, vol. 14, no 2, p. 134-161, 1995.
[34] YANG (H.), WANG (W.), SUN (J.), Control point adjustment for B-spline curve approximation, *Computer-Aided Design*, vol. 36, no 7, p. 539-552, 2004.
[35] LAVOUE (G.), DUPONT (F.), BASKURT (A.), A new CAD mesh segmentation method, based on curvature tensor analysis, *Computer-Aided Design*, In press, 2005.
[36] COHEN-STEINER (D.), MORVAN (J.), Restricted delaunay triangulations and normal cycle, *ACM Sympos. Computational Geometry*, p. 237-246, 2003.
[37] ALLIEZ, (P.), COHEN-STEINER (D.), DEVILLERS (O.), LEVY (B.), DESBRUN (M.), Anisotropic Polygonal Remeshing, *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 485-493, 2003.
[38] LAVOUE (G.), DUPONT (F.), BASKURT (A.), A new subdivision based approach for piecewise smooth approximation of 3D polygonal curves, *Pattern Recognition*, In press, 2005.
[39] LAVOUE (G.), DUPONT (F.), BASKURT (A.), Toward a near optimal quad/triangle subdivision surface fitting, *IEEE Int. Conf. on 3-D Digital Imaging and Modeling (3DIM)*, In press, 2005.
[40] HERTEL (S.), MEHLHORN (K.), Fast triangulation of simple polygons, *International FCT-Conference on Fundamentals of Computation Theory Proc.*, vol. 158, pp. 207--218, 1983.