Toward a near optimal quad/triangle subdivision surface fitting

G. Lavoue , F. Dupont and A. Baskurt
LIRIS FRE 2672 CNRS, Villeurbanne, France.

December 9, 2004

Abstract

In this paper we present a new framework for subdivision surface fitting of arbitrary surfaces (not closed objects) represented by polygonal meshes. Our approach is particularly suited for output surfaces from a mechanical or CAD object segmentation for a piecewise subdivision surface approximation. Our algorithm produces a mixed quadrangle-triangle control mesh, near optimal in terms of face and vertex numbers while remaining independent of the connectivity of the input mesh. The first step approximates the boundaries with subdivision curves and creates an initial subdivision surface by optimally linking the boundary control points with respect to the lines of curvature of the target surface. Then, a second step optimizes the initial control polyhedron by iteratively moving control points and enriching regions according to the error distribution. Experiments conducted on several surfaces and on a whole segmented mechanical object, have proven the coherency and the efficiency of our algorithm, compared with existing methods.

1 Introduction

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 [1]. For all these reasons, given an input target mesh, subdivision surface approximation algorithms become quite beneficial in terms of compression (the original mesh can be stored or transmitted in the form of a coarse control polyhedron), remeshing (the subdivided control polyhedron is often much regular than the original mesh), reverse engineering or animation. In this context, we present an algorithm for fitting a piecewise smooth subdivision surface to an input mesh aiming at getting close to the optimality in terms of control points number and connectivity of the subdivision control polyhedron. Our method is particularly suited for mechanical surfaces or CAD parts; indeed in these cases the research of the optimality is quite relevant. Input surfaces must be open, but can have multiple holes (our method cannot handle closed surfaces) but this restriction is not critical since a closed surface can be segmented to give adapted output surfaces. Section 2 details the related work about subdivision surface fitting, while the overview of our method is presented in section 3. Sections 4 and 5 deal with the two distinct steps of our method: the initialization and the optimization of the subdivision surface. Finally, in section 6, results are presented, evaluated and compared with existing methods.

2 Related Work

Several methods already exist for subdivision surface fitting, most of them take as input a dense mesh and obtain the subdivision control mesh connectivity by simplification: Lee et al. [2] and Ma et al. [3] use the Quadric Error Metrics from Garland and Heckbert [4]. Kanai [5] uses a modified version which directly minimizes error between the original mesh and the subdivided simplified mesh. With these simplification based approaches, the control mesh connectivity strongly depends on the input mesh. Figure 1 shows an example of the approximation method from Kanai [5] applied on two different meshes representing the same shape. It appears obvious that results are quite different. Particularly, the control polyhedron in Figure 1.e obtained for the bad tesselated mesh of Figure 1.d is not correct and gives a quite poor limit surface (see Figure 1.f) regarding to the original one. In our algorithm, in order to remain independent of the original connectivity, we use the boundaries and the curvature information of the target surface to transmit the topology to our control polyhedron. Suzuki et al. [6] also remain independent of the target mesh, by iteratively subdividing and shrinking an initial hand defined control mesh toward the target surface. Unfortunately this method fails to capture local characteristics for complex target surfaces, and is only suited for genus 0 closed surfaces. Jeong and Kim [7] use a similar shrink wrapping approach and encounter the same problems with complex topologies.

Concerning the geometry optimization, Lee et al. [2] and Hoppe et al. [8] sample the original mesh with a set of points and minimize a quadratic error to the subdivision surface. Suzuki et al. [6] propose a faster approach, also used in [7]: the position of each control point is optimized, only by reducing the distance between its limit position and the target surface. Hence only subsets of the surfaces are involved in the fitting procedure thus results are not so precise and may produce oscillations. Ma et al. [3] consider the minimization of the distance 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 approximant introduced by Pottmann and Leopoldseder [9]. This algorithm allows more accurate and rapid convergence (see Section 3.2).

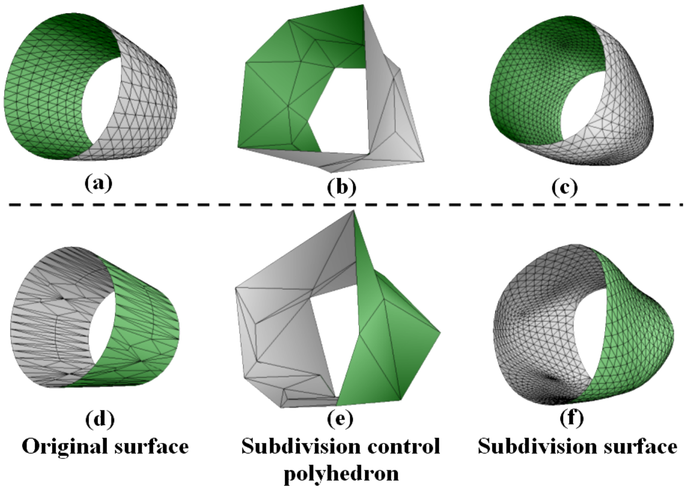

Figure 1: An example of subdivision surface fitting using the algorithm from Kanai [5].

To our knowledge, the optimality in terms of control point number and position represents a minor problematic in the existing algorithms but is particularly relevant for mechanical or CAD objects. Only Hoppe et al. [8] 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.

3 Framework

Our framework for subdivision surface fitting is the following: firstly an initial subdivision surface is constructed, independently of the target surface connectivity (see Section 4), by first approximating boundaries and then using curvature information. Secondly this initial surface is enriched by inserting new control points while optimizing the geometry and the connectivity.

3.1 The choice of the subdivision scheme

Within our framework, we have to choose a subdivision scheme. Many subdivision rules exist, some of them are adapted for triangular control meshes, like Loop [10] and others are adapted for quadrilateral ones, like Catmull-Clark [11]. 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 quadrangles (quads for short) or any quad can be tessellated into triangles, results are not equivalent. The fact is that the nature of the control polyhedron (quads or triangles) strongly influences the shape and the parameterization of the resulting surface. The body of a 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 [12]. This scheme reproduces Catmull-Clark on quad regions and Loop on triangle regions (see Figure 2). Corresponding subdivision masks are detailed in Figure 8.

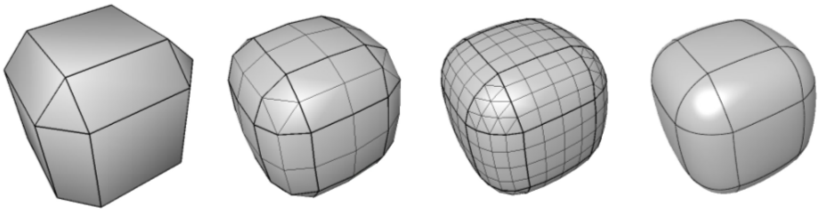

Figure 2: Example of Triangle-quad subdivision.

3.2 The approximate squared distance

The subdivision curve approximation of the boundaries (see Section 4.2), as the subdivision surface optimization (see Section 5.1), 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 [13, 14] or surfaces [15]. They are mostly based on a data parameterization wich is very complex to optimize. Other approaches [16] 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 parameterization. Hence, we have chosen to generalize the Active B-Spline approach from Pottmann and Leopoldseder [9] 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 have proven to converge much faster than traditional ones [9, 17] 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 Ψ, we can define at each point t_0, a Cartesian system (e_1, e_2, e_3) whose first two vectors e_1, e_1 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 Ψ is given by [9]:

$$F_d(x_1, x_2, x_3) = \frac{d}{d+\rho_1} x_1^2 + \frac{d}{d+\rho_2} x_2^2 + x_3^2 \tag{1}$$

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 ρ_1 (resp. ρ_2) is the curvature radius at $\Psi(t_0)$, 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 [9] for a detailed derivation and proof of these formula.

4 Subdivision surface initialisation

4.1 Overview

The purpose of the initialization process is twice: transmitting the topology from the original target surface to our initial control polyhedron and optimizing the connectivity of this control polyhedron. The initialization algorithm is the following: first, the boundaries of the target surface are approximated by piecewise smooth subdivision curves; then our process will attempt to connect control points of the corresponding control polygons, in order to create the optimal set of facets that will represent our initial control polyhedron. These edges are chosen according to the curvature directions of the original surface. According to these edges, the topology of the surface is reconstructed, in a simple and efficient manner even for complex topologies.

4.2 Boundary curve approximation

4.2.1 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 subdivision surface by Hoppe et al. [8] for the particular case of *crease* 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:

$$P'_i = \frac{1}{8}(P_{i-1} + 6P_i + P_{i+1}) \tag{2}$$

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 [8] for *corner* vertices) to handle sharp parts and extremities:

$$P'_i = P_i \tag{3}$$

This subdivision curve will coincide with the boundary of a subdivision surface generated by commonly used subdivision rules like Catmull-Clark [11] or Loop [10].

4.2.2 The approximation algorithm

Our main purpose concerning this part of the process is to get close to the optimality in terms of number and placement of the found control points. Our algorithm is an extension for subdivision rules, including sharp vertex processing, of the Active B-Spline Curve developed by Pottmann and Leopoldseder [9] (see Section 3.2). One shortcoming of their method is the high dependency to the initial active curve. Hence we have introduced a new process, analyzing curvature properties of B-Splines, which computes a near optimal evaluation of the initial number and positions of control points. Describing this curve approximation method is beyond the scope of this paper, thus we invite readers to refer to [18] for complete explanations and details about this algorithm. Globally, our boundary curve approximation works as follows: firstly, sharp vertices are detected and the boundary is cut into smooth parts. Then, for each smooth part, we apply the approximation algorithm described in [18] to find a near optimal approximating subdivision curve. Then smooth parts are connected with associated *sharp* tagged control points. An example is shown on Figure 3.

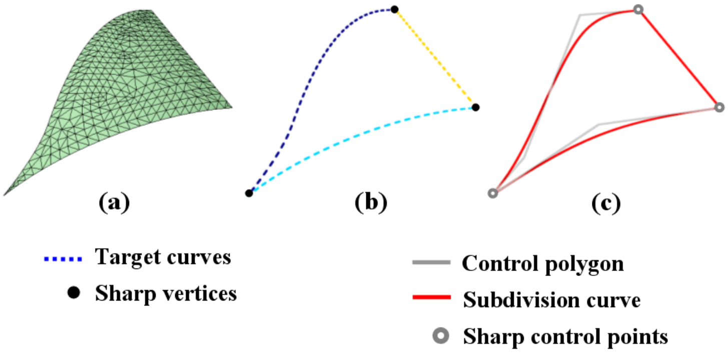

Figure 3: Example of boundary approximation.

4.3 Edge score definition

Once the boundary control polygons have 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. For this purpose, we consider the lines of curvatures 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 [19]. This coherency between control lines and lines of curvature is shown in the example on Figure 4.

Thus, for each couple of control points from the boundary control polygons,

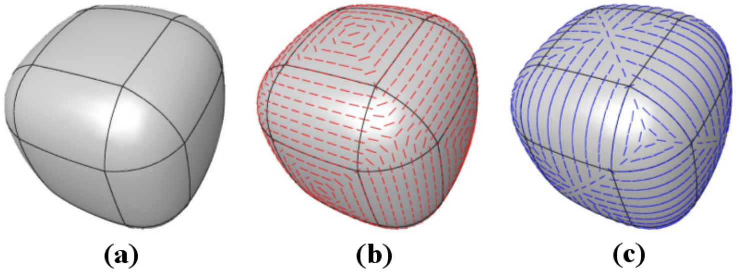

Figure 4: The coherency between control lines (a), minimum (b) and maximum (c) directions of curvatures.

a Coherency Score (SC) is calculated, taking into account the coherency of the corresponding potential control edge with the lines of curvatures of the corresponding area on the target surface.

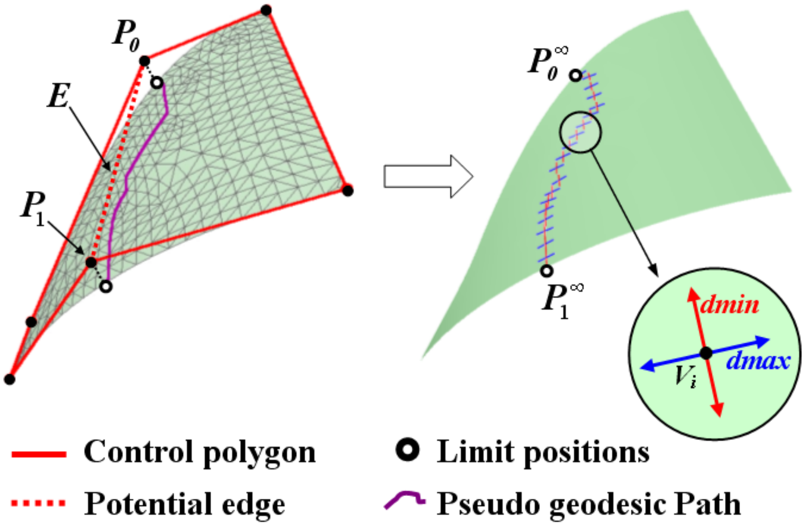

Figure 5: Mechanism for edge score definition.

The mechanism is illustrated on Figure 5: 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:

$$SC(E) = \frac{min(\sum_{i=1}^{n} \theta min_i, \sum_{i=1}^{n} \theta max_i)}{n} \tag{4}$$

with θmin_i (resp. θmax_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 $SC \in [0, 90]$ is homogeneous to an angle value in degrees. Two special cases are taken into account, concerning the nature of vertices V_i belonging to the path:

- If V_i owns an isotropic curvature tensor (plane or spherical region), hence the directions of curvature do not carry information. In these cases θmin_i and θmax_i are set to 45, to not influence the final score.

- If V_i is on a boundary (while not being the beginning or the end of the path), then a penalty is introduced, because if the corresponding potential edge represents a correct control edge, thus it should not cross or touch a boundary. Therefore in these cases θmin_i and θmax_i are set to 90.

4.4 Topology extraction and reconstruction

The reconstruction of a control polyhedron having the same topology that the original surface is not a trivial problem, knowing that the target surface can have multiple holes (and therefore multiple boundary control polygons). Alliez et al. [19] use parameterization and constrained Delaunay algorithms for topology reconstruction; we aim at avoiding such complex processes knowing that moreover, parameterization does not always work on surfaces with multiple holes. Our algorithm is the following: we extract a single contour, that we call the Topologic Contour, representing the boundary of the final control polyhedron. In the case of a single boundary target surface, it is automatic. In the case of a multiple boundaries target surface, we have several control polygons, hence we link them by creating edges and doubling certain control points. For n boundaries, we create $(n-1)$ edges, by choosing those associated with smallest scores SC. This process is illustrated in Figure 6.

For a two holes surface (see Figure 6.a), we have created one single topological oriented contour (see Figure 6.b). The difficulty here is to create a coherent Topological contour. Figure 6.c presents this problematic. We have chosen to start the topologic contour from control point B_0^1 which, therefore, becomes C_0, then B_1^1 becomes C_1 and B_1^2 becomes C_2 and then a question occurs: does the topological contour have to continue on B_0^2 or B_2^2? Even if this question seems trivial for a plane object, it becomes very complex in the case of a topologically complex, multiple holes surface and moreover will be critical for the rest of the process. Our solution is the following (see Figure 6.d): first, considering limit positions of B_1^1 and B_1^2, we mark every edge belonging to their pseudo-geodesic path. Then we extract triangles from the previous path ($TB_0^1B_1^1$) and from each possible path ($TB_0^2B_1^2$ and $TB_1^2B_2^2$). Finally we calculate the shortest path, considering marked edges as impassable, from $TB_0^1B_1^1$ to $TB_0^2B_1^2$ and $TB_1^2B_2^2$. The shortest path (the blue arrow in the example) gives us the correct control point to integrate to the topological contour (see Figure 6.e). Once the topological contour has been extracted, our algorithm is quite simple (see Figure 7). We consider the potential edge associated with the smallest score SC

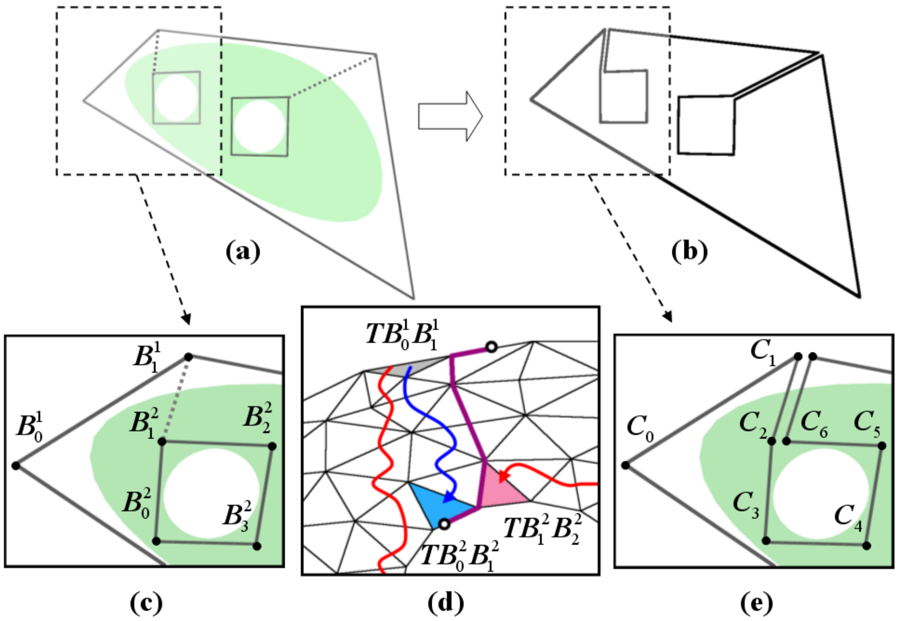

Figure 6: Topology extraction for a surface with three boundaries.

(dotted segments in Figure 7), and we cut the contour along this edge, creating two sub-contours. This algorithm is repeated recursively on sub-contours until it remains only plane contours (see contours 1,2,3 on Figure 7). Then for each plane contour, 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 [20]. By assembling created facets we obtain our initial polyhedron of which limit surface (see Figure 7) represents in most case a quite good approximation of the original surface.

5 Subdivision surface optimization

Even if the initial subdivision surface often represents a good approximation of the target surface, the initialization mechanism considers only the boundary information. Hence we have now to take into account the interior data. Considering this purpose, we have defined two complementary mechanisms: A subdivision inversion algorithm, generalizing Pottmann and Leopoldseder method [9] for the complex quad-triangle subdivision rules, and a enrichment mechanism which adds points and optimizes connectivity according to the error position and distribution.

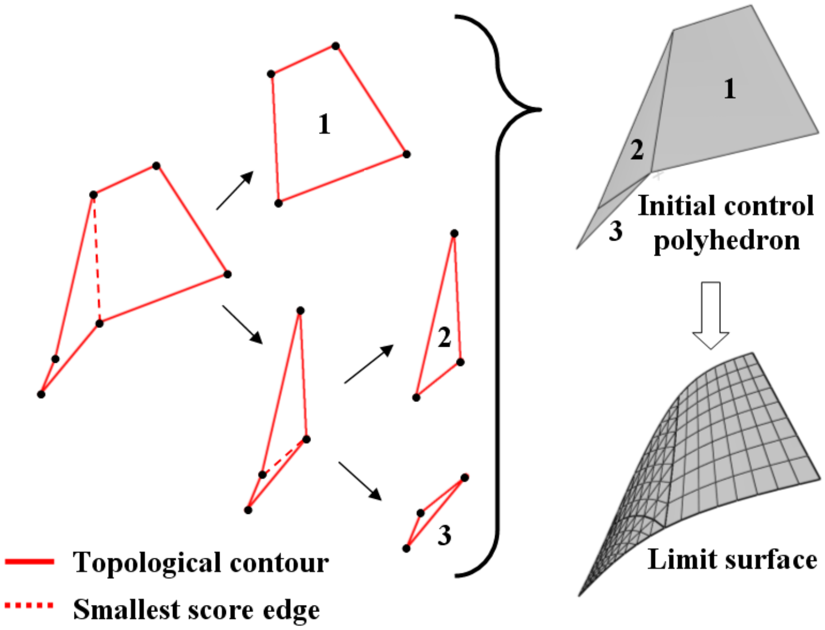

Figure 7: The initial control polyhedron creation mechanism.

5.1 Quad-triangle subdivision inversion

For a given target surface and a given approximating 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 approximant defined by Pottmann and Leopoldseder [9] (see Section 3.2). Our algorithm is the following:

- The curvature is calculated for each vertex of the target surface. We have implemented the work of Cohen-Steiner et al. [21], based on the Normal Cycle. This estimation procedure has proven to be the most efficient and stable among the others and gives very satisfying results even for bad tessellated objects.

- 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 radius ρ_1 and ρ_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 Figure 8); they correspond to control points $P_i^{l_0}$ at the finer level l_0.

$$S_k = C_k(P_1^0, P_2^0, ..., P_n^0) \tag{5}$$

- The functionnals C_k are determined using iterative multiplications of the l_0 subdivision matrices M_l associated with our subdivision rules (see Figure 8). These subdivision matrix M_l are such as:

  $$P^l = M_l.P^{l-1} \tag{6}$$

  with $P^l = [(P_1^l, P_2^l, ..., P_n^l)]^T$. The subdivision masks depends on the nature of the neighboorhood of the considered control points. For a given control point P_i, surrounded by n_e edges and n_q quads, the mask is given in Figure 8.c. If P_i is surrounded only by triangles (resp. quads) the mask is given in Figure 8.a (resp. Figure 8.b). We also take into account border points (see Equation 2) and sharp border points (see Equation 3). The functionnals C_k, for the level l_0, are the lines of the matrix C such as:

  $$C = \prod_{l=1}^{l_0} M_l \times L_{l_0} \tag{7}$$

  L_{l_0} is the limit matrix which gives the limit positions, proposed by Stam and Loop [12], 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.

  $$F = \sum_k F_d^k(S_k) = \sum_k F_d^k(C_k(P_1^0, P_2^0, ..., P_n^0)) \tag{8}$$

Figure 9 shows an example of the algorithm. Figure 9.b and 9.e show respectively a hand-made bad initial control polyhedron and the corresponding limit surface. After only 5 iterations, the limit surface (Figure 9.g) is perfectly fitted with the original one. Resulting errors are respectively 15.9×10^{-3} and 5.0×10^{-3} after 1 and 5 iterations for our algorithm against respectively 21.4×10^{-3} and 16.1×10^{-3} for the traditional point-to-point minimization used by Ma et al. [3] for example. Our algorithm is clearly faster to converge. All surfaces considered in the experiments were normalized in a cubic bounding box of length equal to 1.

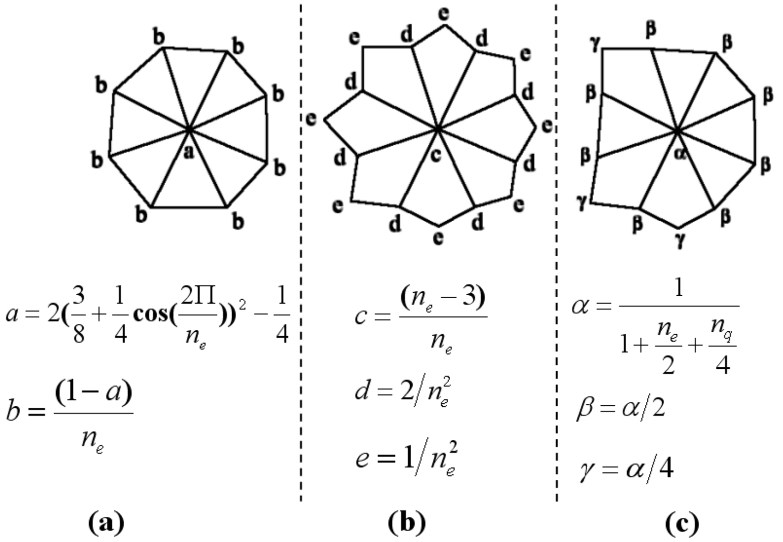
a
b
c
d
e
α
β
γ
$a = 2(\frac{3}{8} + \frac{1}{4}\cos(\frac{2\Pi}{n_e}))^2 - \frac{1}{4}$
$b = \frac{(1-a)}{n_e}$
$c = \frac{(n_e - 3)}{n_e}$
$d = 2/n_e^2$
$e = 1/n_e^2$
$\alpha = \frac{1}{1 + \frac{n_e}{2} + \frac{n_q}{4}}$
$\beta = \alpha/2$
$\gamma = \alpha/4$
(a)
(b)
(c)

Figure 8: Smoothing masks for Loop, Catmull-Clark and the quad-triangle scheme.

5.2 Enrichment and connectivity optimization

In this section we present our method to enrich precisely the polyhedron, while trying to keep a near optimal connectivity. 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 an add to our error set, $S_k max$ corresponding to the maximum error, and every sample points corresponding to a significant error (we have fixed a threshold $T = \frac{d_k max}{2}$) and connected to an other point of the error point set. This extraction is shown for a 2D case in Figure 10.

Then, we distinguish two cases, illustrated in Figure 11 for the target objects from figure 12.b and 12.a:

1. The error field corresponds to a local error. Hence, if several 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 11.b show such an error field (error points are marked in red). Corresponding faces have been merged, before adding a new control point (see Figure 11.c).

2. The error field is diffuse. Hence, there is no precise error center, the error

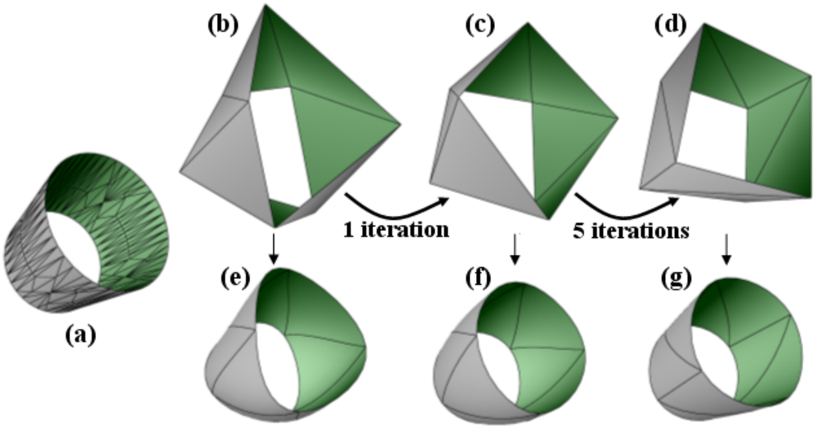

Figure 9: Example of our surface optimization. (a) original surface. (b,c,d) control polyhedrons after 0,1 and 5 iterations. (e,f,g) limit surfaces.

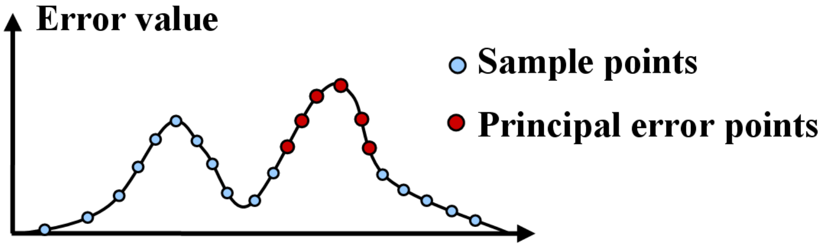

Figure 10: Principal error field extraction.

> field corresponds rather to a lack of degrees of freedom. Thus, every concerned face F_k is enriched. A point is added at the center and connected to its neighbors. If two faces are adjacent we also cut their common edge. An example is shown on Figure 11.e and 11.f. This mechanism concerns also cases were there exist one principal error but the error field already contains a control point. This means that the control point does not bring enough freedom to model the target surface, hence we enrich every face of the field.

We detect these two cases, simply by considering the percentage of the error point set with an error close to $d_k max$ (the theshold $0.80 \times d_k max$ gives satisfaying results). If this percentage is lower than a threshold (usually 50%) thus the error set is considered as a Gaussian like distribution associated with a local error (case 1), otherwise the error set is considered as a plateau like distribution (case 2). This quite simple algorithm has given satisfying results in our experiments.

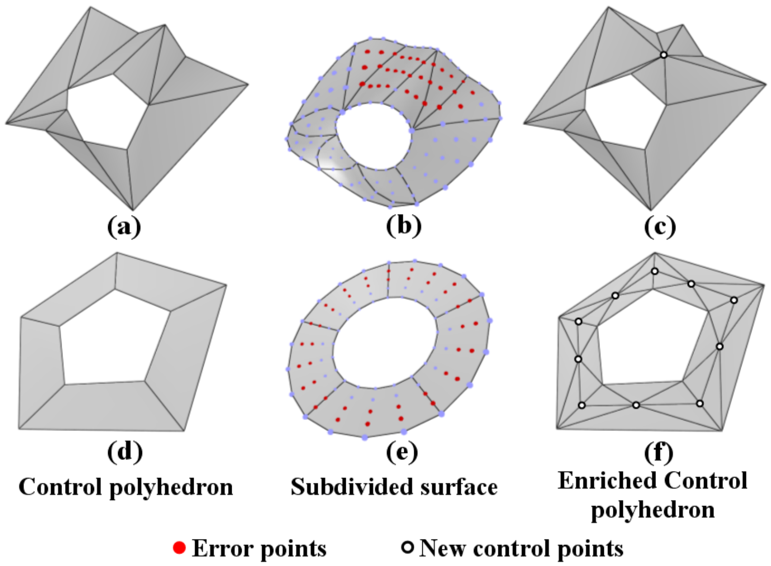

Figure 11: The two cases of error distribution with corresponding enrichment.

6 Complete algorithm and results

Our whole algorithm is the following:

```
Begin $Subdivision\ Surface\ Approximation$
  The initial subdivision surface is created using boundary curves and curvature information (see Section 4).
  while $E > \epsilon$ do
    // $E$ is the approximation error and $\epsilon$ a threshold value.
    while $E > \epsilon$ and $m < m_0$ do
      // $m$ is the iteration number and $m_0$ a maximum number.
      Optimisation procedure (see Section 5.1). The subdivision surface is moved toward the target surface, by minimizing a sum of quadratic distances.
    end while
    if $E > \epsilon$ then
      A new control point is inserted onto the subdivision surface according to the error distribution (see Section 5.2).
    end if
  end while
End $Subdivision\ Surface\ Approximation$
```

Our approximation method was tested on several different objects (with $\epsilon = 5 \times 10^{-3}$ and $m_0 = 5$). Figure 12 presents results for different target surfaces with different topologies. Control polyhedrons have quite small numbers of faces and vertices compared with initial surfaces (convenient for compression tasks) and the approximation errors remain very low (see Table 1). Besides, for

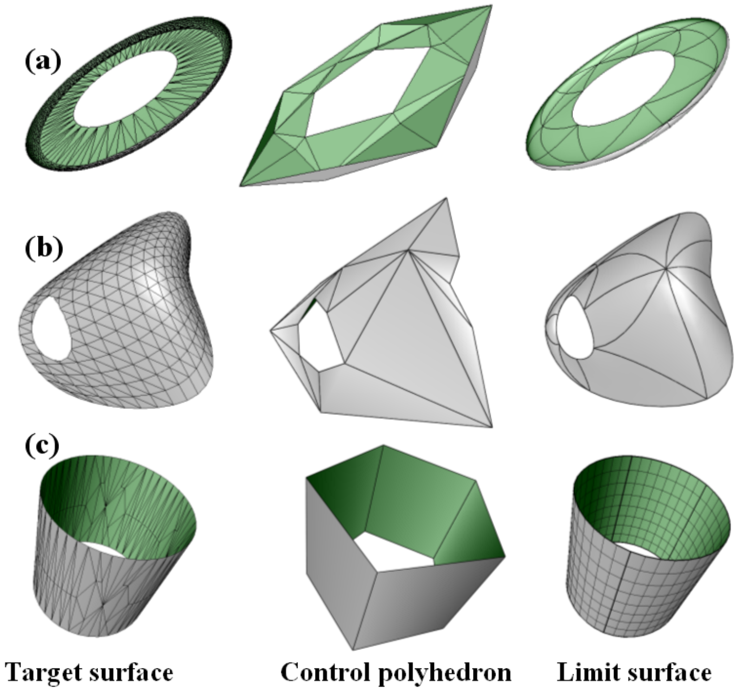

Figure 12: Examples of our subdivision surface approximation scheme.

the example of the cylinder (see Figure 12.c) results are quite better than for the algorithm from Kanai [5] (see Figure 1). Moreover we can distinguish an other advantage, dealing with the remeshing task, on Figure 12.c: the resulting subdivided surface is a quite nicely remeshed model compared with the initial target object. We have applied our method to a whole object (see Figure 13), by previously segmenting it (we have used the algorithm from Lavoué et al. [22]), and then applying our algorithm on each patch (boundaries where fixed to avoid cracks). Then, all patches have been connected by marking boundary control edges as *sharp*. The resulting subdivision surface is quite interesting in terms of vertex and face numbers and approximation errors.

We have compared our results with methods from Ma et al. [3] and Hoppe et al. [8] (see Table 2). We obtain a better approximation error than Ma et al., for a lower number of faces and vertices. Hoppe et al. obtain a better quadratic error than ours but both are quite low and our control polyhedron is lighter than theirs. Moreover their method relies on a very long and complex global optimization while our algorithm is faster (about 5*s* for Fandisk). Ma et al. and Hoppe et al. produce a triangle only control polyhedron, while our method is able to adapt the connectivity to the natural parameterization of the target object by creating triangles and quads.

	$V/F\ Ini$	$V/F\ Ctrl$	$E\ L1\ (\times 10^{-3})$
(Fig12.a)	950/1734	20/30	2.69
(Fig12.b)	324/491	16/17	2.06
(Fig12.c)	168/248	10/5	1.72

Table 1: Vertex and Face numbers of initial objects ($V/F\ Ini$) and control polyhedrons ($V/F\ Ctrl$), and Resulting errors ($E\ L1$).

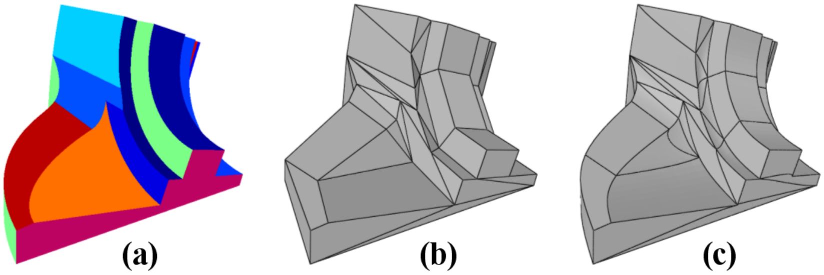

Figure 13: Application of our fitting scheme to a complete object. (a) Segmented object, (b) control polyhedron, (c) limit surface.

7 Conclusion

We have presented a new framework for subdivision surface fitting. Our algorithm, adapted for open surface meshes, is independent of the connectivity of the target mesh and aims at optimizing the generated subdivision surface, in terms of connectivity and control points number. First, boundaries of the target surface are approximated with subdivision curves which lead to a first version of the subdivision surface 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 surface of which topology is reconstructed using pseudo geodesic distances computation. The second step is the following: the initial subdivision surface is iteratively enriched and optimized until the approximation error becomes correct. The optimization step is an extension for subdivision surfaces of the quite fast and efficient method from Pottmann and Leopoldseder and the local enrichment step adds control points while optimizing the connectivity according to the error distribution. Applications of our algorithm are quite large including compression, reverse engineering and remeshing. Concerning perspectives, we plan to improve the connectivity optimization during mesh enrichments, by conducting a deeper analysis of the error dispersion.

	$V/F\ Ctrl$	$E\ L1$	$E\ L2$
Our	75/89	0.82×10^{-3}	$2,63 \times 10^{-6}$
Ma	173/342	5.06×10^{-3}	
$Hoppe$	87/170		$1,00 \times 10^{-7}$

Table 2: Vertex and Face numbers of control polyhedrons ($V/F\ Ctrl$), average errors ($E\ L1$) and average quadratic errors ($E\ L2$) for different approximation methods applied to the Fandisk object.

8 Acknowledgements

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

References

[1] MPEG4, ISO/IEC 14496-16. Coding of Audio-Visual Objects: Animation Framework eXtension (AFX), 2002.

[2] A. Lee, H. Moreton, H. Hoppe, Displaced subdivision surfaces., ACM SIGGRAPH Proceedings, pages 85–94, 2002;

[3] W. Ma, X. Ma, S. Tso, Z. Pan, A direct approach for subdivision surface fitting from a dense triangle mesh., Computer Aided Design 36 (16):525–536, 2004.

[4] M. Garland, P. Heckbert, Surface simplification using quadric error metrics., ACM SIGGRAPH Proceedings, pages 209–216, 1997.

[5] T. Kanai, Meshtoss: Converting subdivision surfaces from dense meshes, 6th International Workshop on Vision, Modeling and Visualization, pages 325–332, 2001.

[6] H. Susuki, Subdivision surface fitting to a range of points., Proceedings of Pacific graphics, pages 158–167, 1999.

[7] W.K. Jeong, H.C. Kim, Direct reconstruction of displaced subdivision surface from unorganized points, Journal of Graphical Models 64 (2):78–93, 2002.

[8] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, W. Stuetzle, Piecewise smooth surface reconstruction., ACM SIGGRAPH Proceedings, pages 295–302, 1994.

[9] H. Pottmann, S. Leopoldseder, A concept for parametric surface fitting which avoids the parametrization problem, Computer Aided Geometric Design, 20 (6) (2003) 343–362.

[10] C. Loop, Smooth subdivision surfaces based on triangles, Master's thesis, Utah University, 1987.

[11] E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, Computer-Aided Design 10 (6):350–355, 1978.

[12] J. Stam, C. Loop, Quad/triangle subdivision, Computer Graphics Forum 22 (1):79–85, 2003.

[13] J. Hoschek, Intrinsic parametrization for approximation., Computer Aided Geometric Design. 5 (1):17–31, 1988.

[14] T. Speer, M. Kuppe, J. Hoschek, Global reparametrization for curve approximation., Computer Aided Geometric Design 15 (9):869–877, 1998.

[15] W. Ma, J. Kruth, Parametrization of randomly measured points for the least squares fitting of b-spline curves and surfaces., Computer-Aided Design 27 (9):663–675, 1995.

[16] V. Krishnamurthy, M. Levoy, Fitting smooth surfaces to dense polygon meshes, ACM SIGGRAPH Proceedings, pages 313–324, 1996.

[17] H. Yang, W. Wang, J. Sun, Control point adjustment for b-spline curve approximation., Computer-Aided Design 36 (7):539–552, 2004.

[18] G. Lavoue, F. Dupont, A. Baskurt, a new data driven approach for 3d curve subdivision inversion, Tech. rep., LIRIS - FRE 2672 CNRS, 2004. (http://liris.cnrs.fr/docs/RR-2004-019.pdf)

[19] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, M. Desbrun, Anisotropic polygonal remeshing, ACM Transactions on Graphics 22 (3):485–493, 2003.

[20] S. Hertel, K. Mehlhorn, Fast triangulation of simple polygons, Lecture Notes In Computer Science, Proceedings of the International FCT-Conference on Fundamentals of Computation Theory. volume 158, pages 207–218, 1983.

[21] D. Cohen-Steiner, J. Morvan, Restricted delaunay triangulations and normal cycle, 19th Annu. ACM Sympos. Comput. Geom., 2003.

[22] G. Lavoue, F. Dupont, A. Baskurt, A new cad mesh segmentation method, based on curvature tensor analysis., Computer-Aided Design, In press.