CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

FRE 2672

Rapport de recherche RR-2004-019

« A new data driven approach for 3D curve subdivision inversion »

Lavoué Guillaume, Dupont Florent, Baskurt Atilla, Mai 2004

Laboratoire d'**I**nfo**R**matique en **I**mage et **S**ystèmes d'information

LIRIS FRE 2672 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon
Secrétariat : Université Claude Bernard Lyon 1, bâtiment Nautibus
43, boulevard du 11 novembre 1918 — F-69622 Villeurbanne cedex
Tél. : (+33) 472 43 26 10 - Fax : (+33) 472 43 15 36 - http://liris.cnrs.fr/

A new data driven approach for 3D curve subdivision inversion

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

May 5, 2004

Abstract

This paper presents an algorithm dealing with the data reduction and the approximation of 3D polygonal curves considering the inverse problem of curve subdivision. Our method, based on the analysis of the data, allows to approximate efficiently a set of straight 3D segments or points with a subdivision curve, in a near optimal way in terms of control point number. It includes an extension for subdivision rules of the Active B-Spline Curve developed by Pottmann et al. Their method, starting with an initial curve and making it converge toward the target one, does not need a data point parameterization, because it relies on local quadratic approximants of the curve distance function. Thus the convergence speed is better than other existing parameterization based methods. A critical parameter of this algorithm is the initial curve position before starting the optimization. To address this, we have developed a theoretically demonstrated approach, analysing curvature properties of B-Splines, that allows to obtain a near optimal evaluation of the initial number and positions of control points. Moreover our original Active Footpoint Parameterization method allows to prevent wrong matching problems occurring particularly for self-intersecting curves. Thus the stability of the algorithm is highly increased. Our method was tested on different sets of curves and gives satisfying results regarding to approximation error, convergence speed and compression rate. This method is in line with a larger 3D CAD object compression scheme by piecewise subdivision surface approximation. The objective is to fit a subdivision surface on a target patch by first fitting its boundary with a subdivision curve whose control polygon will represent the boundary of the surface control polyhedron.

1 Introduction

The context of this 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). The objective is the low bandwidth transmission of CAD objects, represented by 3D meshes, with multi-resolution and adaptivity properties. Meshes are optimized in terms of triangle numbers and original NURBS information are not available. In this context, a 3D compression algorithm is needed but the optimized tessellation and the need of a low bandwidth transmission make this problematic very complex. The chosen approach is to convert the original object into a set of light patches represented by subdivision surfaces. This representation will bring a high compression rate adapted to a low bandwidth and to a multi-resolution displaying because of subdivision properties. Moreover the model will be adaptive because of the prior decomposition into surface patches. This approximation problematic can be linked with the inverse problem for subdivision surfaces. Within this context we present an efficient algorithm dealing with the inverse problem for subdivision curves whose purpose is twice: firstly it represents a sub-problem of the surface case, and secondly, subdivision curves have the property to represent the boundary of a subdivision surface. Thus dealing first with the boundary of a patch and then with its interior should be an efficient solution. Section 2 details the whole 3D-object compression scheme, whereas subdivision curves are presented in section 3. Section 4 presents the related work about smooth curve fitting and sections 5, 6 and 7 deal with the different parts of our method, the initial curve computation, the optimization scheme and the footpoint determination. Finally, results are presented and discussed in section 8 and an example of the surface approximation case in presented in section 9.

2 Presentation of the whole compression process

The whole process can be decomposed into the following parts:

2.1 Decomposition into patches

Firstly the CAD objects are segmented into surface patches. The method used is based on the curvature tensor field analysis and presents two distinct complementary steps: a region based segmentation which decomposes the object into known and near constant curvature patches, and a boundary rectification based on curvature tensor directions, which corrects boundaries by suppressing their artifacts or discontinuities. This method is detailed in [LDB04a] and [LDB04b]. Resulting segmented patches, by virtue of their properties (known curvature, clean boundaries) are particularly adapted to subdivision surface fitting (see Figure 1).

2.2 Patch approximation

One of the most relevant problems in the fact of approximating an object by patches is the apparition of cracks because each patch will be approximated by a different surface whose boundary will not be perfectly matched with the others. A solution is to add constraints during the fitting process but the complexity will highly increase. Indeed if each patch has constraints with its neighbors, the algorithm will become a global optimization problem. Another solution is to treat these cracks after the fitting process but the fitted patches will be modified compared with the first approximation. Our solution is simpler and more effective. For each patch the subdivision surface approximation problem is divided into two sub-problems: a piecewise approximation of the

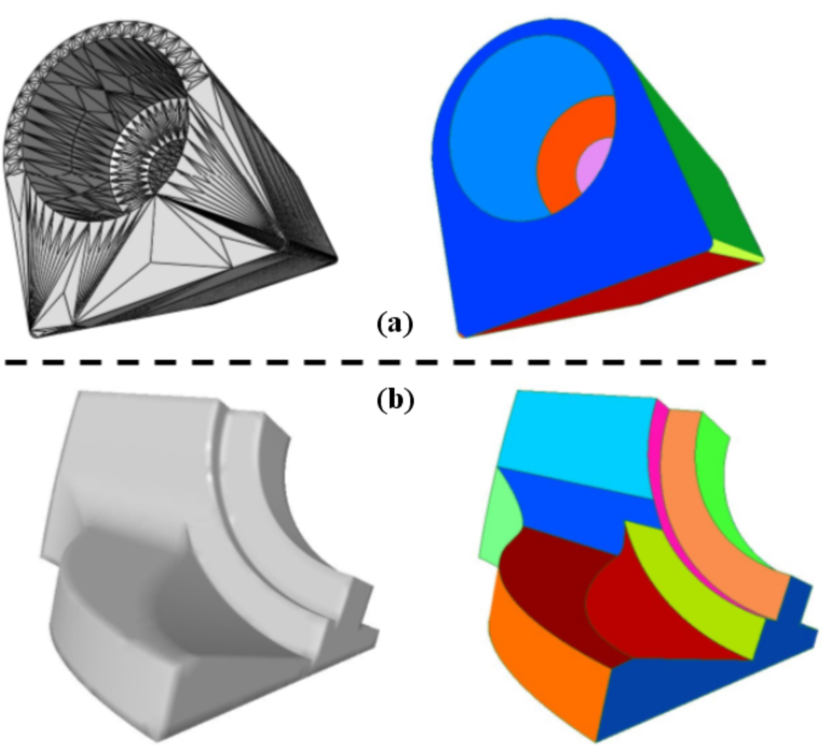

Figure 1: Segmentation of Swivel(a) and Fandisk(b) objects into patches adapted to subdivision surface fitting.

patch boundary and the construction of the final subdivision surface by interpolation of the found boundary and approximation of the interior data, like the approach proposed by Schweitzer [Sch96]. In order to prevent our model from cracks, for each patch the boundary is divided into pieces of boundary corresponding to the different adjacencies with its neighboring regions (see Figure 2). Once each piece of boundary has been approximated by a subdivision curve (this inverse problem is treated in this paper), the corresponding control polygons are put together to form the control polygon of the whole boundary. According to subdivision properties, this control polygon will represent the boundary of the control polyhedron of the subdivision surface representing the corresponding patch. Then, for each patch, a subdivision control mesh is created using its boundary information. The final control mesh defining the whole surface will comprise control meshes of all regions.

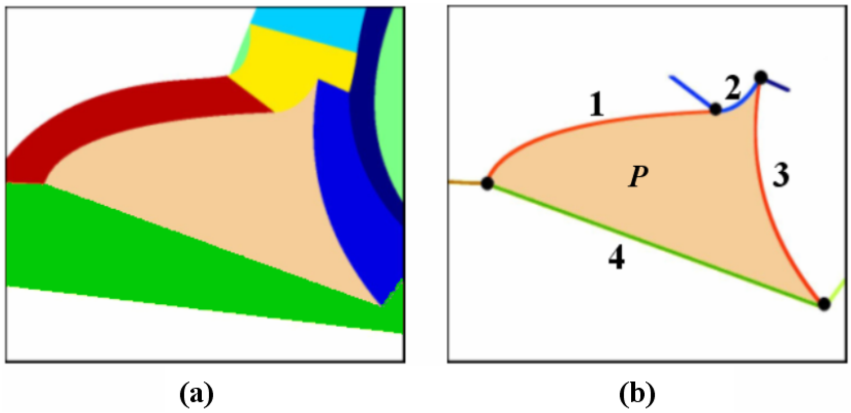

Figure 2: Extraction of pieces of boundary (1,2,3,4) of a segmented patch (P) from the Fandisk object.

3 Subdivision curve presentation

The subject of this paper is the approximation of a polygonal curve with a subdivision curve. 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. [HDD*94] 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 [WW02]:

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

With these rules, the subdivision curve corresponds to a uniform cubic B-spline, except for its end segments. This curve will coincide with the boundary generated by commonly used subdivision surface rules like Catmull-Clark [CC78] or Loop [Loo87]. An example of subdivision curve is presented in Figure 3.

4 Related Work

This inverse problem for subdivision curve ties up with the smooth parametric curve approximation problematic, with the additional difficulty that a subdivision curve does not have a parametric formulation and cannot be evaluated at any point. But this shortcoming can be solved using different techniques like does Schweitzer [Sch96]. The author considers the subdivision curve as an infinite sequence of cubic Bezier segments. Thus any point can be evaluated by founding the Bezier control polygon (given by repeated subdivisions) which includes the considered parameter. Most of the smooth curve approximation methods are based on a data parameterization. Let o_i being the sequence of p points to approximate, and S, our B-spline or subdivision curve, the

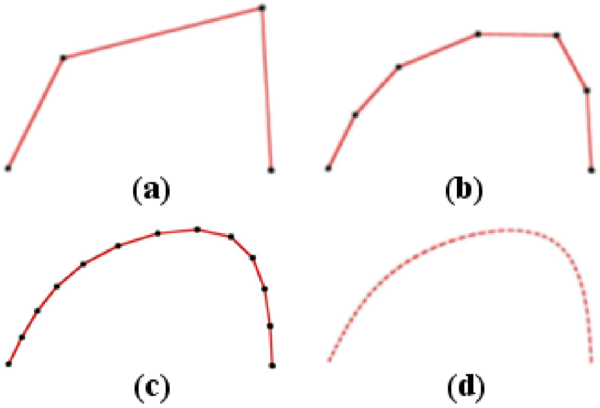

Figure 3: Example of subdivision curve. (a) control polygon, (b,c) 2 iterations of subdivision, (d) limit surface.

usual approaches compute the control points that minimize an error function F:

$$F = \sum_{i=1}^{p} ||S(\xi_i) - o_i||^2 \tag{2}$$

with ξ_i the parameter value assigned to the data point o_i. The minimization of this over determined quadratic system is generally solved by a least square method, which leads to a square linear system. The main problem lies in the choice of the parameters ξ_i which highly influence the result. Three methods are commonly used to assign the approximation parameter locations (see [Lee89]): Uniform, Chord Length and Centripetal parameterizations. But none of these solutions is optimal and really adapted. Schweitzer [Sch96], in its subdivision curve approximation algorithm, starts with a Chord Length parameterization and then corrects the parameters by considering at each iteration the parameters of the data point projections on the approximating subdivision curve. Some authors have proposed more sophisticated and efficient iterative parameter correction procedures, notably the intrinsic parameterization of Hoschek [Hos88], which was improved by Saux et al. [SD03]. The algorithm firstly minimizes the system with respect to S, considering initial ξ_i values, and secondly with respect to ξ_i (separately for each parameter value). Another solution was proposed by Speer [SKH98] which considers a global approach, both the control points and the parameters are considered as unknowns. But fundamentally this parameterization remains a problem because the correction procedures are time consuming and take many iterations to converge(>50).

Pottmann et al. [PLH02] have proposed a new and very efficient approach inspired by the active contour models of Kass et al. [KWT88]. Their approximation scheme does not require parameterization. The idea is to make an active initial B-Spline converge towards a target curve by minimizing local approximate squared distances from the target curve. It is not a point to point distance minimization but a point to curve minimization which allows a very fast convergence (<20 iterations). Unfortunately this method is very dependant of the initial active curve. The original method from Pottmann was extended by Yang et al. [YWS04] to permit a dynamic control point insertion or removal. We have extended this algorithm for subdivision curves (see section 6). One shortcoming of this approach is the instability of the active curve which may require the use of a smoothing term which increases the computational cost. Our algorithm will increase the stability by preventing the active curve from bad crossing or wrong matching using an active pseudo-parameterization of the target data, called Active Footpoint Parameterization (see section 7).

A remaining problem in these approximation algorithms is the choice of the initial number and placement of the control points. Schweitzer [Sch96] starts with 4 control points and then increases iteratively the number according to the resulting error. Saux et al. [SD03] [SD99] consider initially a high number of control points and then determine the minimum number using a dichotomic method. Finally, Yang et al. [YWS04] heuristically determine the number by considering the local direction monotony of the target curve. But their method leads to a generally too high number of control points, so a removal algorithm is generally needed. Concerning the positions of the initial control points, the initial parameterization for the initial curve determination is usually the Chord Length [YWS04] [Sch96] or the Centripetal [SD03] and thus gives rather poor initial results. However these control point initial placement and number are critical because, better is the initial approximation and faster will be the convergence algorithm. A control point insertion or removal, for instance, is a very time consuming task because it will influence a significant part of the curve and thus require an other cycle of iterations. Within this context, we have developed an efficient determination method for the number and positions of initial control points, based on curvature analysis and theoretical foundations (see section 5).

5 Initial control point processing

Correct number and positions of the control points of the initial active curve are critical for our convergence algorithm. Our subdivision curve represents a uniform cubic B-Spline curve except for its end segments (see Section 3), 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 1 *Considering a uniform cubic B-Spline segment, local curvature maxima are necessarily located at the extremities.*

Proof : *Each uniform cubic B-Spline segment S_i is associated with 4 control points $(P_{i-3}, P_{i-2}, P_{i-1}, P_i)$. For any parameter value u such as $0 \leq u \leq 1$, the corresponding position $S_i(u)$ on this segment is defined by:*

$$\begin{aligned} S_i(u) &= \frac{1}{6}((1-u)^3 P_{i-3} + (3u^3 - 6u^2 + 4)P_{i-2} \\ &\quad + (-3u^3 + 3u^2 + 3u + 1)P_{i-1} + u^3 P_i) \end{aligned} \tag{3}$$

Thus the second derivative vector is:

$$\begin{aligned}\ddot{S}_i(u) &= (1-u)P_{i-3}+(3u-2)P_{i-2} \\ &\quad +(-3u+1)P_{i-1}+uP_i \end{aligned} \tag{4}$$

The curvature $C(u)$, at each parameter, is defined by $C = \left\|\ddot{S}_i(u)\right\|$. Our goal is to study the variation of this curvature, thus we have calculated its first derivative which can be expressed as:

$$\dot{C}(u) = \frac{f(u)}{\sqrt{g(u)}} \tag{5}$$

with f linear in u and g quadratic in u. Thus the equation $\dot{C}(u) = 0$ has one or zero solution $u_0 \in]0,1[$, therefore the curvature is either monotonic, or has one extremum over the cubic segment (not located at an extremity).

- *If the curve segment is monotonic there exist one curvature maximum at one extremity ($u = 0$ or $u = 1$), therefore the theorem is verified.*
- *If the curve has one extremum, we will determine if it is a maximum or a minimum. For this purpose, we have studied the sign of the second derivative $\ddot{C}(u)$ of the curvature. We found that:*

  $$\forall u \in [0,1]\ ,\ \ddot{C}(u) \geq 0 \tag{6}$$

  thus $\dot{C}(u)$ is increasing $\forall u \in [0,1]$, and therefore $\dot{C}(u) \leq 0$ for $u \in [0,u_0]$ and $\dot{C}(u) \geq 0$ for $u \in [u_0,1]$. as a result $C(u)$ is decreasing before u_0 and increasing after and therefore $C(u_0)$ represents a local minimum. thus curvature maxima are located at the extremities, therefore the theorem is verified.

Formal and limit calculation have been made using the software Waterloo Mapple©.

According to Theorem 1, 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 we need at least *n* initial control points.

Consequently, our initial curve processing algorithm is the following:

- Smoothing and quantification of the curvature, and detection of the *n* local maxima.
- The number of control points is initialized to *n*, increased by 2 for the extremities is the curve is open.
- The placement of the *n* control points is determined with a linear $n \times n$ system. Indeed, for a subdivision curve, the limit position P_i' of a control point P_i can be processed according to its neighbors:

  $$P_i' = \frac{1}{6}(P_{i-1}+4P_i+P_{i+1}) \tag{7}$$

Since we know that these limit positions must coincide with the local curvature maxima positions, we obtain the linear $n \times n$ system.

Figure 4 shows this initialization process: the variation of the curvature (a) with the determination of the 5 local maxima, and the corresponding initial subdivision curve (b) processed using the positions of these maxima. We can observe that this initialization curve is very satisfying considering the target curve, thus the number of convergence iterations will be considerably reduced. Moreover we have asserted that this number of control points is minimum therefore no control point removal will be needed for the further optimization algorithm.

There exist some special cases, in which our rule for the initialization is modified:

- If the curvature is constant and not null or monotonic thus we consider 4 initial control points (2 and the extremities for open curves), with their limit positions uniformly distributed over the target curve.
- If the curvature is null on the whole curve, we consider a straight line linking up the extremities (this case is not possible for closed curve).

6 Optimization scheme

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. We have extended the method from Pottmann et al. [PLH02]. This method relies on the distance function of the data curve Ψ, which assigns to each point its shortest distance to Ψ. In practical terms, not the distance function itself, but the local quadratic approximants of the squared distance function are considered.

6.1 Local quadratic approximants of the squared distance function to 2D and 3D curves

Considering a 2D space Π and a smooth curve Ψ, the Frenet frame (e_1, e_2) at a curve point $\Psi(t)$ is defined as follows: $e_1 = \dot{\Psi} / \left\|\dot{\Psi}\right\|$ is the unit tangent vector and e_2 the associated unit normal vector. Considering a point p in Π, and its corresponding footpoint $\Psi(t_0)$ (associated with the shortest distance d of the curve), thus the coordinates of p in the Frenet frame defined in $\Psi(t_0)$ are $(0, d)$. Then the local quadratic approximant $F_d(p)$ of the squared distance of p to the curve Ψ is given by:

$$F_d(x_1, x_2) = \frac{d}{d+\rho} x_1^2 + x_2^2 \tag{8}$$

where x_1 and x_2 are the coordinate of p with respect to the Frenet frame and ρ is the curvature radius at $\Psi(t_0)$. The reader may refer to [PH02] for a detailed derivation and proof of this formula. In the case of a 3D space, considering p and the associated footpoint $\Psi(t_0)$, a cartesian coordinate system (e_1, e_2, e_3) is defined such as: $e_1 = \dot{\Psi} / \left\|\dot{\Psi}\right\|$

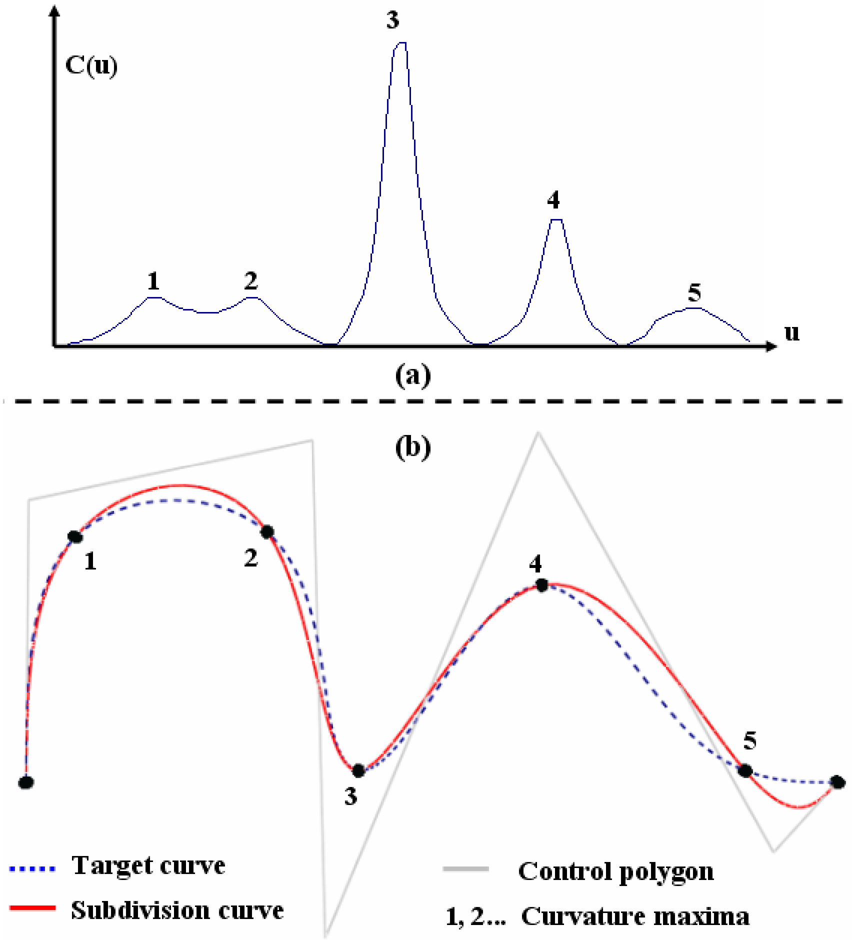

Figure 4: Example of initial control point processing. (a) Curvature variation over the target curve, (b) Corresponding maxima and initial subdivision curve with the associated control polygon.

is the unit tangent vector, e_3 is in the direction of $p - \Psi(t_0)$ and $e_2 = e_3 \wedge e_1$. In this frame, the local quadratic approximant is given by [PLH02]:

$$F_d(x_1, x_2, x_3) = \frac{d}{d+\rho} x_1^2 + x_2^2 + x_3^2 \qquad (9)$$

In our case, the target curve is sampled and therefore not continuous like Ψ. However Pottmann et al. [PLH02] have shown that their estimator is still valid in this case, considering discrete values for ρ as for the Frenet Frame.

6.2 Optimization algorithm

The optimization process is the following, for each iteration:

- Several sample points S_k are chosen on the subdivision curve, and the associated footpoints 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. Sample points S_k can be computed as linear combinations of the control points P_i (see Section 3):

$$S_k = L_k(P_1, P_2, ..., P_n) \tag{10}$$

- For each S_k the local quadratic approximant F_d^k of the squared distance function of S_k to the target curve, is computed according to Frenet frame at O_k.
- New positions of control points are processed by minimizing the sum of the local quadratic approximants:

$$F = \sum_k F_d^k(S_k) = \sum_k F_d^k(L_k(P_1, P_2, ..., P_n)) \tag{11}$$

  The minimization of this quadratic function in the new position of the control points, lead to the resolution of a linear squared system.

These iterations are repeated until the approximation error or the variation of the approximation error is lower than a given threshold. The convergence of the algorithm is very fast. Figure 5 presents three iterations of the algorithm: Figure 5.a shows the initial position of the subdivision curve with a sample point S_k and the corresponding footpoint O_k, whereas Figure 5.b and 5.c present the new positions of the control points after respectively 1 and 2 iterations of the optimization algorithm. At the second iteration the target curve is perfectly fitted.

7 Footpoint determination

7.1 The wrong matching problematic

The footpoint determination algorithm used in [PLH02] and [YWS04], consists, for each sample point, in considering the smallest distance point on the target curve. [YWS04] precomputes the discrete distance field using the Fast Marching Method in order to increase the speed but the result is the same. Considering this method, a problem will occur for a self intersecting target curve or when a part of this target curve is very close to another part: sample points will be associated with incorrect footpoints belonging to wrong parts of the curve. This wrong matching was also observed by [YWS04] who found no general solution. As a consequence, either the convergence of the algorithm will slowed down because a higher number of iterations is required, or the convergence will become impossible. This problem is illustrated in Figure 6.

We have developed an efficient solution for this problem, based on the determination of Generalized Footpoints and their active parameterization.

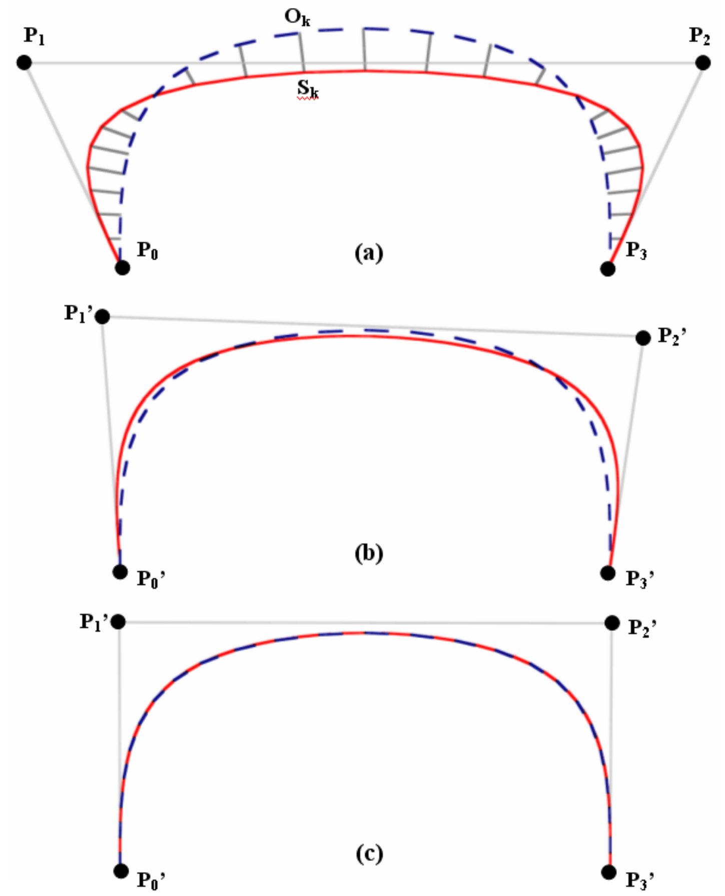

Figure 5: Example of the optimization procedure, (a) Initial subdivision curve, (b,c) Resulting curve after respectively 1 and 2 optimisation iterations.

7.2 Generalized Footpoint computation

The footpoint O_k corresponding to a point S_k is defined as the point belonging to the target curve and associated with the shortest distance from S_k. Since the target curve is usually defined by a highly sampled polygonal curve, the footpoint determination consists generally in computing point to segment distances and considering the projected point associated with the shortest distance. In the continuous case, considering a smooth curve Ψ (at least C^1), a footpoint O_k is necessary issued from an orthogonal projection of S_k onto the curve. We introduce Generalized Footpoints (GF) as the set of points O_k^g issued from an orthogonal projection of S_k. In our polygonal case, Generalized Footpoints are more complex to determine since the normals are not continue but piecewise constant. The determination process is the following: each point T_i of the

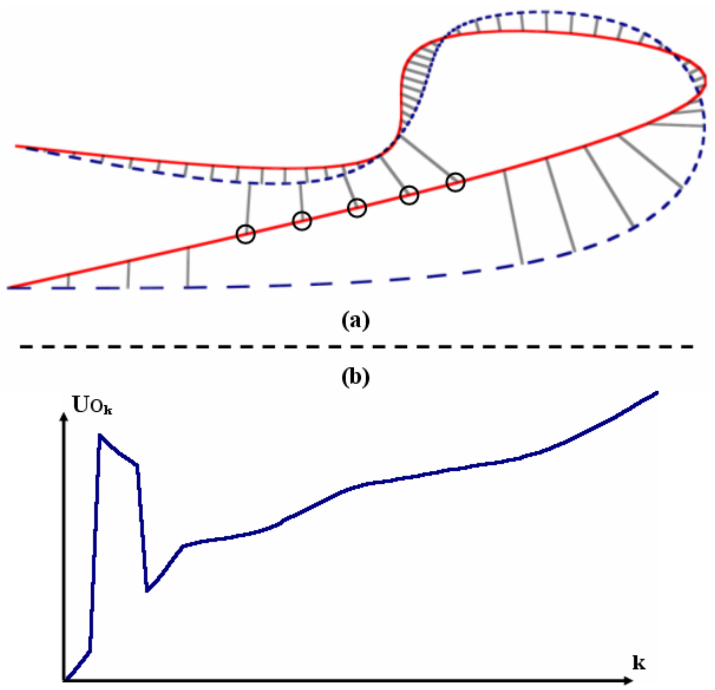

Figure 6: Illustration of the bad footpoint placement problematic, (a) Subdivision curve with misplaced footpoints (corresponding to the surrounded sample points), (b) Footpoint parameter distribution, with an evident discontinuity corresponding to the badly placed ones.

target curve is associated with a parameter t_i (Chord Length Parameterization), such as:

$$\begin{aligned} t_1 &= 0 \; and \; t_n = 1 \\ t_{i+1} &= \frac{\|T_{i+1} - T_i\|}{\sum_{i=n-1}^{i=1} \|T_{i+1} - T_i\|} \end{aligned} \qquad (12)$$

For each point T_i of the target curve, we consider the two incident segments Seg_1 and Seg_2 and the associated projections p_k^1 and p_k^2 of the considered S_k on lines carrying Seg_1 and Seg_2. Different cases are considered:

- $p_k^1 \in Seg_1$ (resp $p_k^2 \in Seg_2$) thus p_k^1 (resp. p_k^2) is considered as a Generalized Footpoint (see Figure 7.a).
- $p_k^1 \notin Seg_1$ and $p_k^2 \notin Seg_2$, and they are not on the same side of their respective segments, thus T is considered as a Generalized Footpoint (see Figure 7.b).
- $p_k^1 \notin Seg_1$ and $p_k^2 \notin Seg_2$, and they are on the same side of their respective segments, there is no associated GF (see Figure 7.c).

Figure 8 shows an example of the determined generalized Footpoints O_k^g for a sample point S_k. Only one of them corresponds to the correct one within our optimization procedure, this choice is detailed in the next subsection.

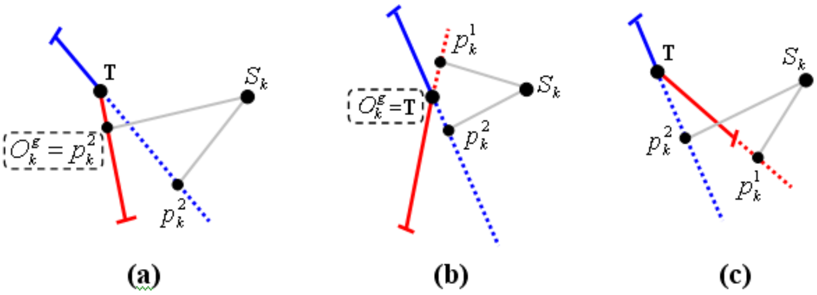

Figure 7: Generalized Footpoints (GF) determination mechanism for a piece of target curve consisting of two segments, (a,b) determination of a GF, (c) no GF.

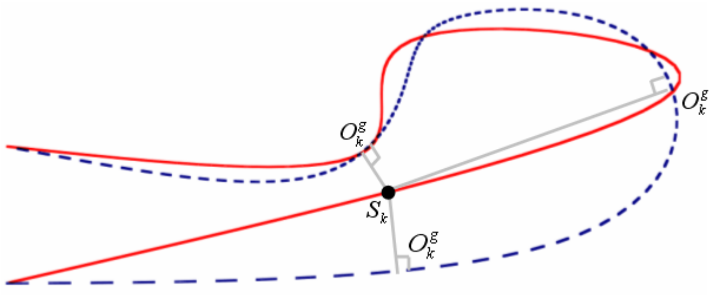

Figure 8: Example of the set of Generalized Footpoints O_k^g for a sample point S_k.

7.3 Active Footpoint Parameterization

Parameter values U_{Ok} are associated with footpoints O_k, they are computed by a linear interpolation between the parameters of the target curve points which surround the considered footpoints. We have studied the variation of these footpoint parameter values U_{Ok} for sample points S_k along the active curve. We consider the set of footpoints O_k as correct if the U_{Ok} distribution is strictly increasing. Figure 6.a shows bad placed footpoints, the corresponding U_{Ok} distribution is presented in Figure 6.b. An evident discontinuity appears in the distribution which notifies the wrong matching problem. Our goal is to find a footpoint distribution which gives a strictly increasing U_{Ok} distribution. Our algorithm, so called Active Footpoint Parameterization is the following:

- We compute Generalized Footpoints (see section 7.2), for each sample point of the subdivision curve.

- For each sample point S_k, we consider among its Generalized Footpoints O_k^g, the smallest increasing one:

$$O_k = argmin_{O_k^g}(\left\|O_k^g - O_{k-1}\right\|),\ (O_k^g - O_{k-1}) > 0 \tag{13}$$

- If the found O_k is too high compared with O_{k-1}, thus it is considered not correct and eliminated, the corresponding S_k is not considered in the optimization process for the current iteration, because not coherent footpoint has been found. It is the same if no O_k can be found.

With this algorithm, the set of determined footpoint parameters is strictly monotonic and increasing, whatever the curve to treat. Thus wrong matching is eliminated and the optimization procedure is prevented from instability or oscillations. As a consequence Figure 9.a shows the curve presented in Figure 6.a, with correct footpoints determined with our Active Footpoint Parameterization method. Results are now correct, resulting footpoints are coherent and adapted to the optimization procedure. The U_{Ok} distribution, presented in Figure 9.b, is increasing and much more correct that the distribution in Figure 6.b.

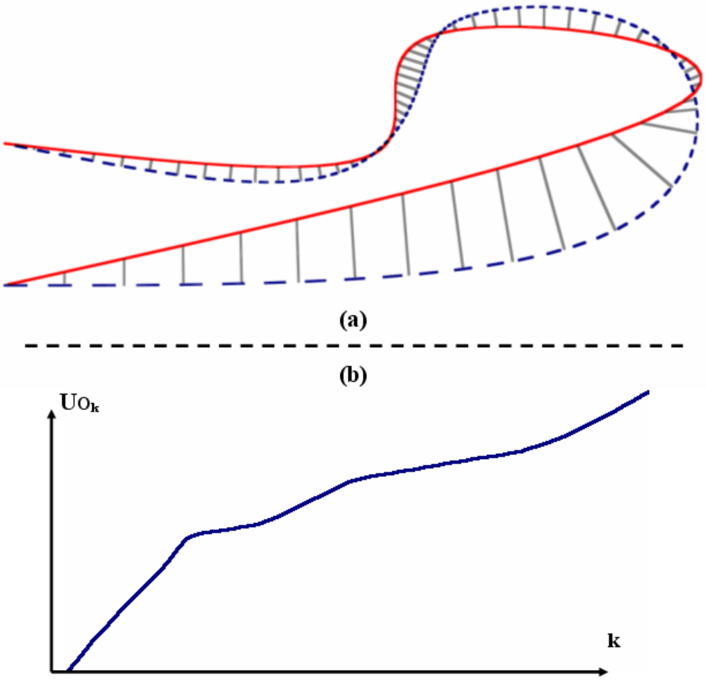

Figure 9: Illustration of Active Footpoint Parameterization results for the footpoints determination, (a) Subdivision curve with corresponding correct footpoints, (b) Footpoint parameter distribution.

8 Complete algorithm and results

The whole subdivision curve approximation algorithm is the following:

1. Initialization of the subdivision curve, according to the curvature of the target curve (see section 5).

2. Computation of the correct footpoints, using the Active Footpoint Parameterization (see section 7). The number of sample points on the subdivision curve is chosen by the user, in our examples we consider vertices of the curve subdivided twice.

3. Optimisation procedure (see section 6). The subdivision curve is moved toward the target curve, by minimizing a sum of quadratic distances. The approximation error E is computed.

4. (2) and (3) are repeated m times until $E < \varepsilon$ or $m < m_0$. ε and m_0 are respectively maximum fixed error and iteration number.

5. If $E < \varepsilon$ the process is terminated, else a new control point is inserted onto the subdivision curve, where the local error is maximum and the process goes to step (2).

We have conducted many tests on different target curves from different natures, in order to demonstrate the efficiency of our method. We present here several examples about the different characteristics of our algorithm. Most of the presented curves are 2D (except for Figure 14) in order to improve visibility but our algorithm work as well on 3D curves. The average error E is defined by the mean of the distances from each sample point to its corresponding footpoint. All curves considered in the experiments were normalized in a bounding box of length equal to 1.

8.1 Initial control points placement examples

First, we have conducted experiments about the efficiency of the subdivision curve initialization presented in section 5. For different target curves, we have considered 2 different initial curves, one processed with our algorithm and the other, with the same number of control points with limit positions evenly sampled on the target curve. Results are presented on Figure 10. On one hand, Figure 10.c presents the initial curve computed with our curvature based method (10 control points) whereas its optimization result which converges after several optimization iterations is in Figure 10.d. On the other hand, the other initial curve, computer by a regular sampling of the same number of control points, and the corresponding convergent result are presented in Figure 10.a and 10.b.

The resulting curve corresponding to our curvature based initialization is very closed to the target curve (the resulting error is $2,302 \times 10^{-3}$) and particularly, is closer than the regular sampling one of which resulting error is $5,168 \times 10^{-3}$. These resulting errors appear more clearly in Figure 11. The evolution of the optimization results is presented for each iteration. The error associated with the curvature based initialization

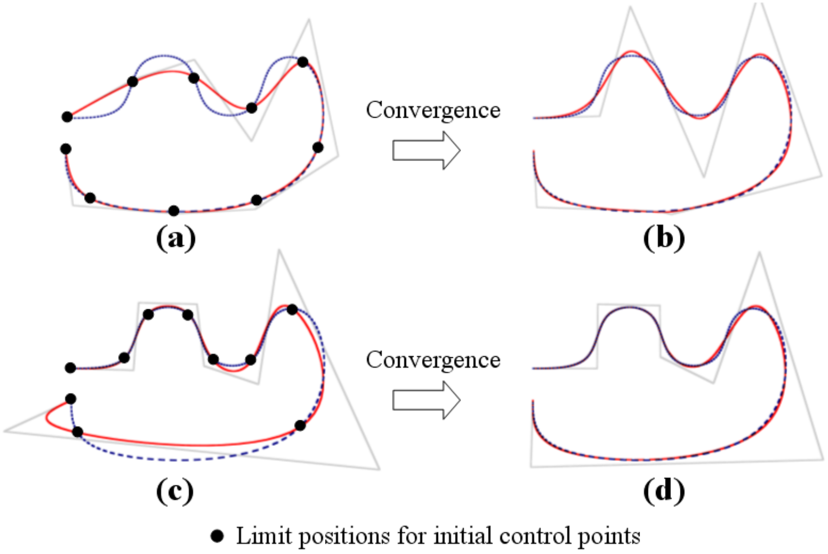

Figure 10: Optimization results starting from two different initial subdivision curves. (a,b) Initial evenly distributed control points and corresponding final convergent result (after 7 iterations). (c,d) Initial control point processing using curvature and corresponding final convergent result (after 7 iterations).

is always lower than the other. Our initialization procedure provides a near optimal set of control points in term of placement, and a number generally sufficient for a good approximation.

Iterations	0	1	3	6
Curve14a, RSI	$17,09$	$8,62$	$5,09$	$5,05$
Curve14a, CBI	$12,35$	$3,15$	$1,90$	$1,85$
Curve14c, RSI	$24,3$	$16,22$	$13,87$	$12,41$
Curve14c, CBI	$26,48$	$8,10$	$4,38$	$4,38$

Table 1: Error ($\times 10^{-3}$) evolution for several iterations, for the target curves presented in Figure 14 and different initialization methods.

Other results for the two target curves of Figure 14 are presented in Table 1. *Curve14a* corresponds to Figure 14.(a,b) and *Curve14c* corresponds to Figure 14.(c,d). For both target curves, the error is much lower at each iteration for the curvature based initial curve (CBI). In the case of *Curve14c* the approximation error for the regular sampling initial curve (RSI) will finally reach the value $4,38 \times 10^{-3}$ ($\approx$ error value for the 3^{rd} iteration of the CBI curve), but at the 25^{th} iteration, thus our curve initialization has considerably increased the convergence speed. In the case of *Curve14a*, the approxi-

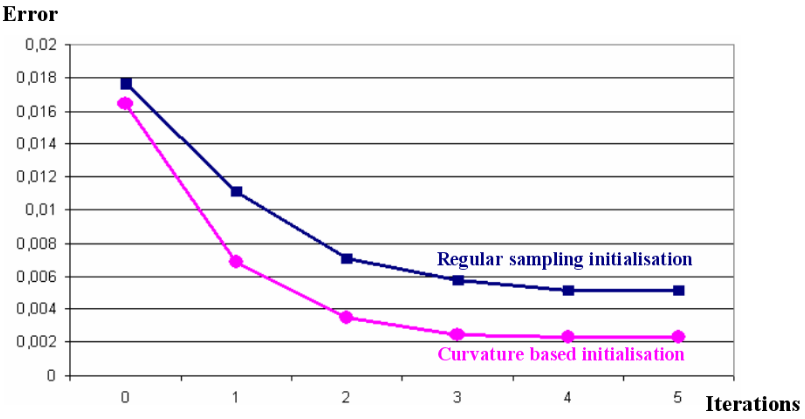

Figure 11: Optimization results evolution for several iterations, starting from two different initial subdivision curves: Curvature bases initialization (CBI) and regular sampling initialization (RSI).

mation errors presented at the 6^{th} iteration for both RSI and CBI curves are approximately the convergence values. Thus in this case our curve initialization has permitted a better approximation, even with an infinite number of iterations.

8.2 Active Footpoint Parameterization examples

We have tested the efficiency of our footpoint determination algorithm (see section 7). We have conducted experiments on curves with close branches (see Figure 12) or self intersections (see Figure 13). Figure 12.a presents a target curve with close branches and the corresponding initial curve with classic footpoint determination, wrong matching problems appear and thus the final curve after several iterations converges toward a bad approximation (see Figure 12.b). On the other hand, our footpoint determination algorithm, presented in Figure 12.c leads to a very satisfying approximation (see Figure 12.d), moreover the convergence was obtained very rapidly after only 4 iterations. Concerning the self-intersected curve presented in Figure 13, results are also very satisfying. Bad convergence results are observed for the classic footpoints determination (see Figure 13.a and 13.b) whereas our method gives a very good approximation (the convergence was obtained after about 5 iterations) by carrying out the footpoint matching successfully.

8.3 Complicated curve examples and compression rate analysis

We have tested our algorithm for complicated target curves in order to test the efficiency of our method whose final purpose is to decrease the amount of data in our final compression objective. Examples are presented for a curve with several concavities and self-intersections (Figures 14.a and 14.b) and for a 3D curve with a complexe shape (Figures 14.c and 14.d). Figures 14.a and 14.c present target curves with the initial

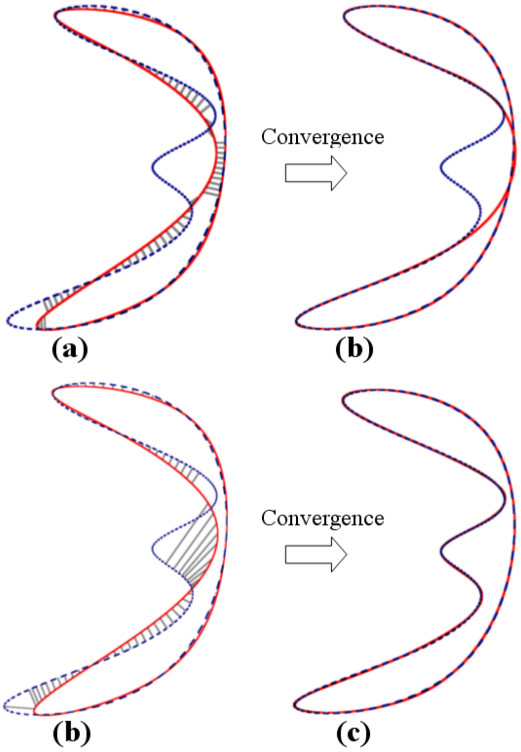

Figure 12: Effect of the Active Footpoints Parameterization on a curve with close parts. Initial curve with associated footpoints computed by the classical method (a) and by our Active Footpoints Parameterization (c) and results of the optimization process (b,d) (4 iterations).

subdivision curves computed according to the curvature analysis presented in section 5. Final subdivision curves processed according to the complete algorithm described at the beginning of this section are presented on Figures 14.b and 14.d. Chosen sample points, for the footpoint determination, are vertices of the curve after two subdivision steps and the error tolerance is $\varepsilon = 1 \times 10^{-3}$.

	(a)	(b)	(c)	(d)
$CtrlNb$	18	21	13	19
$E\ (\times 10^{-3})$	$12,34$	$0,99$	$26,48$	$0,86$
CR	89,5%	87,7%	93,2%	90,2%

Table 2: Resulting errors (E), Final control points numbers ($CtrlNb$) and compression rates (CR) associated to curves approximation of Figure 14.

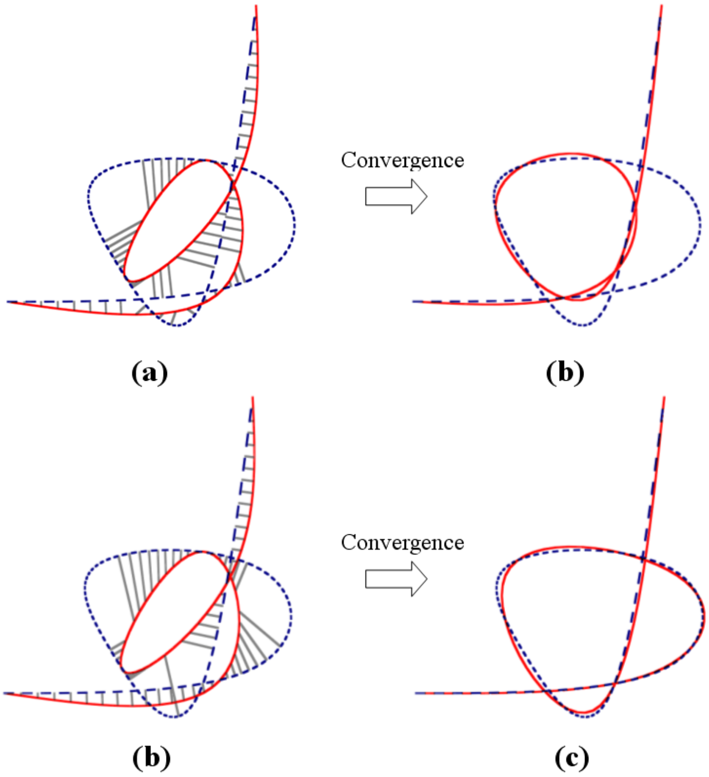

Figure 13: Effect of the Active Footpoints Parameterization on a self-intersecting curve. Initial curve with associated footpoints computed by the classical method (a) and by our Active Footpoints Parameterization (c) and results of the optimization process (b,d) (5 iterations).

Table 2 presents the results. The numbers of points of the target curves are 171 for Figures 14.a and 14.b and 194 for Figures 14.c and 14.d. At the end of the algorithm final numbers of control points (*CtrlNb*) of the approximated subdivision curves are respectively 21 and 19. This is equivalent to compression rates (*CR*) of 87,7% and 90,2%. These are very satisfying results regarding to the small approximation errors E (respectively $0,99 \times 10^{-3}$ and $0,86 \times 10^{-3}$). All experiments were conducted on a PC, with a 2Ghz XEON bi-processor. Processing times were 922 ms for 14.b and 875 ms for 14.d. Initial curve processing times were about 16 ms for each.

9 Future work for surface approximation

The subdivision curve approximation algorithm presented in this paper represents the first step in our surface mesh compression objective, by piecewise subdivision surface

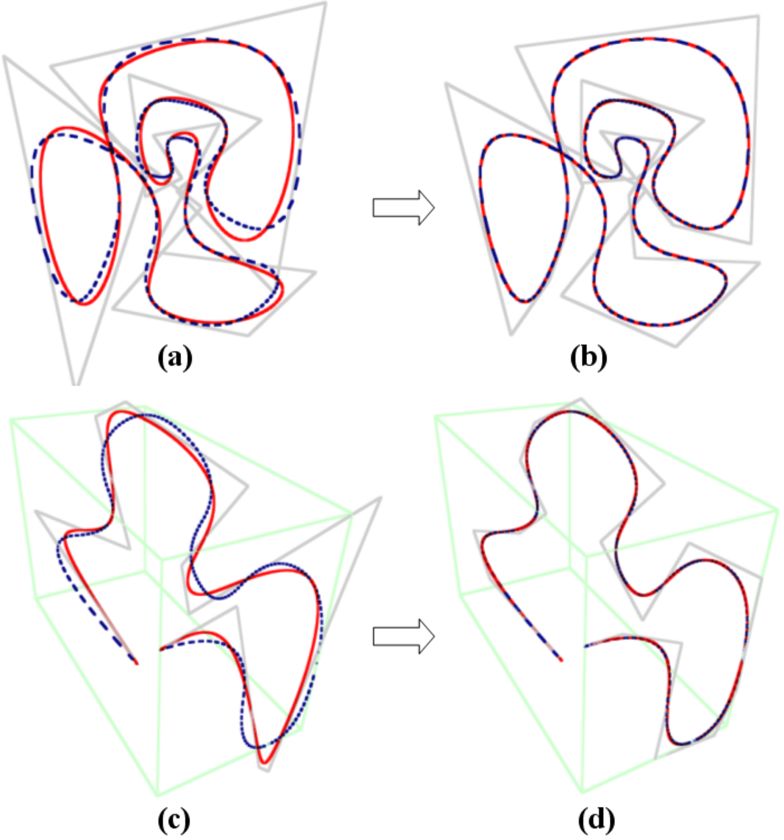

Figure 14: Approximation results for a highly concave, self-intersecting target curve(a,b) and a complex 3D one(c,d) whose bounding box is represented. (a,c) Initial subdivision curves. (b,d) Results after the whole algorithm.

approximation (see section 2). An example of this process is presented in Figure 15.

For a given surface patch, we first extract the boundary (see Figure 15.b), and determine the approximating subdivision curve, containing 6 control points in the example $(P_0, P_1, P_2, P_3, P_4, P_5)$. This control polygon represents the boundary of the searched subdivision control polyhedron, thus we will use it, as a foundation to determine the approximating subdivision surface. In Figure 15.c, the control polyhedron was determined by adding a control point (P_6) to those of the boundary, meshing correctly these control points, and optimizing the placement of P_6 according to the target surface.

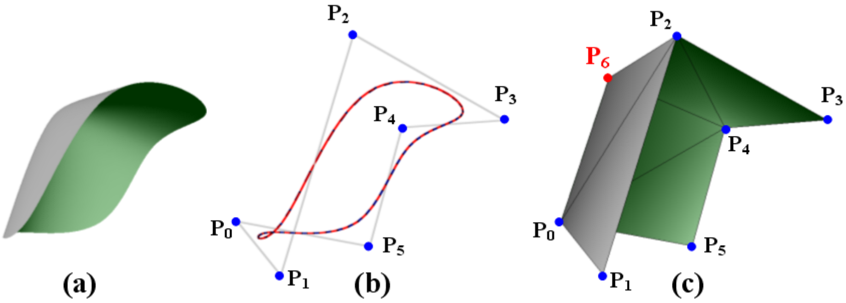

Figure 15: Surface compression mechanism. (a) The mesh to compress. (b) Extraction and approximation of the boundary. (c) Construction of the final subdivision surface control polyhedron.

10 Conclusion

We have presented in this paper an efficient algorithm, based on the analysis of the data, for the inverse problem of curve subdivision. For any polygonal target curve, our algorithm computes the approximating subdivision control polygon optimized in terms of the number of control points. A curvature analysis supported with theoretical foundations permits to compute near optimal numbers and placements of control points for the initial curve construction. The optimization scheme, based on the local quadratic approximants of the squared distance of curves, is an adaptation for subdivision of that presented by Pottmann et al. [PLH02] and dealing with B-Splines. Our original footpoint determination method based on an active parameterization, allows to prevent the wrong matching problem occurring particularly for self-intersecting curves. Thus the stability of the method is highly increased by this good convergence guaranty. Many experiments demonstrate the efficiency of the method for approximation or compression of polygonal curves. This approximation method is involved in a larger surface compression scheme. Target objects are CAD meshes, previously segmented into surface patches. Our purpose is to determine the best approximating subdivision surface for each patch. The method presented in this paper allows to approximate the boundary of a patch with a subdivision curve of which control polygon has the property to represent the boundary of the control polyhedron of the approximating subdivision surface. We plan now to develop the surface approximation algorithm taking as input the target surface patches and their associated subdivision boundary curves.

References

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (1978), 350–355.

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *Computer Graphics (SIGGRAPH 94 Proceedings)* (1994), vol. 28, pp. 295–302.

[Hos88] HOSCHEK J.: Intrinsic parametrization for approximation. *Computer Aided Geometric Design 5*, 1 (1988), 27–31.

[KWT88] KASS M., WITKIN A., TERZOPOULOS D.: Snakes:active contour models. *Intern. J. Computer Vision 1* (1988), 321–332.

[LDB04a] LAVOUE G., DUPONT F., BASKURT A.: Constant curvature region decomposition of 3d-meshes by a mixed approach vertex-triangle. In *Journal of WSCG (WSCG'04 Proceedings)* (2004), vol. 12, pp. 245–252.

[LDB04b] LAVOUE G., DUPONT F., BASKURT A.: Curvature tensor based triangle mesh segmentation with boundary rectification. In *IEEE proc. of Computer Graphics International (CGI'04), In press* (2004).

[Lee89] LEE E.: Choosing nodes in parametric curve interpolation. *Computer-Aided Design 21*, 6 (1989), 363–370.

[Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, Utah University, 1987.

[PH02] POTTMANN H., HOFER M.: *Geometry of the squared distance function to curves and surfaces*. Tech. Rep. 90, Institute of Geometry, Vienna University of Technology, 2002.

[PLH02] POTTMANN H., LEOPOLDSEDER S., HOFER M.: Approximation with active b-spline curves and surfaces. In *Pacific Graphics 02, IEEE Press* (2002), pp. 8–25.

[Sch96] SCHWEITZER J.: *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, 1996.

[SD99] SAUX E., DANIEL M.: Data reduction of polygonql curves using b-splines. *Computer-Aided Design 31*, 8 (1999), 507–515.

[SD03] SAUX E., DANIEL M.: An improved hoschek intrinsic parametrization. *Computer Aided Geometric Design 20*, 8-9 (2003), 513–521.

[SKH98] SPEER T., KUPPE M., HOSCHEK J.: Global reparametrization for curve approximation. *Computer Aided Geometric Design 15*, 9 (1998), 869–877.

[WW02] WARREN J., WEIMER H.: *Subdivision Methods For Geometric Design: A Constructive Approach*. Morgan Kaufmann, 2002.

[YWS04] YANG H., WANG W., SUN J.: Control point adjustment for b-spline curve approximation. *Computer-Aided Design, In press* (2004).