Piecewise surface flattening for non-distorted texture

Conference Paper *in* ACM SIGGRAPH Computer Graphics · July 1991

DOI: 10.1145/122718.122744 · Source: DBLP

Chakib Bennis
IFP Energies nouvelles
35 PUBLICATIONS **383** CITATIONS
SEE PROFILE

Jean-Marc Vézien
Computer Sciences Laboratory for Mechanics a…
33 PUBLICATIONS **426** CITATIONS
SEE PROFILE

Piecewise Surface Flattening for Non-Distorted Texture Mapping

Chakib Bennis [*]

INRIA
BP. 105, 78153 Le Chesnay cedex France

Abstract

This paper introduces new techniques for interactive piecewise flattening of parametric 3-D surfaces, leading to a non-distorted, hence realistic, texture mapping. Cuts are allowed on the mapped texture and we make a compromise between discontinuities and distortions. These techniques are based on results from differential geometry, more precisely on the notion of "**geodesic curvature**": isoparametric curves of the surface are mapped, in a constructive way, onto curves in the texture plane with preservation of geodesic curvature at each point. As an application, we give a concrete example which is a first step towards an efficient and robust CAD tool for shoe modeling.

CR Categories and Subject Descriptors: I.33 [Computer Graphics]: Picture/Image Generation; I.4.3 [Image Processing]: Enhancement-Geometric Correction, Texture.
Additional Keywords and Phrases: Non Distorted Texture Mapping, Piecewise Surface Flattening, Differential Geometry, Geodesic Curvature.

1 Introduction

Texture mapping techniques are widely used to reproduce textural information available in a planar image onto a 3-D surface. This is made possible by making a correspondence between a planar image and a 3-D surface, in order to give each sample point of the output screen reached by the projected 3-D surface an intensity value computed from a point or a set of points of the 2-D image sample. This correspondence is called the "mapping function".

Catmull [8] first introduced a recursive subdivision algorithm to map a 2-D rectangular image onto a 3-D bicubic patch. This method has been refined and enhanced by several authors [9, 6]. These techniques are equivalent to warping the planar rectangle until it takes the shape of the bicubic patch. Unfortunately, these techniques do not preserve distances or angles, resulting in spatial distortions of texture patterns, which can sometimes change the visual appearance of the texture on the surface. Other authors have proposed solutions to reduce these distortions. Bier et al. [5] proposed a 2-part mapping which consists in decomposing the mapping in two steps: the texture pattern is first embedded in a 3-D intermediate surface and then projected onto the target surface in a way that depends only on the geometry of the target object. The distortion is reduced by heuristically choosing the appropriate intermediate surface and the projection method. Unfortunately this is not always easy to do. Fiume et al. [13] have proposed a "polygonal conformal mapping" to map a polygon (e.g., a square) onto an arbitrary convex polygon with preservation of angles. This technique gives good results on polygons for some applications. However, the technique does not preserve distances, thus creating distortions. Moreover, it is not easily extendable to free form surfaces. In [17], Ma et al. used an optimization technique to minimize distortions for general surfaces. The mapping is performed on a grid of sample points of the 3-D surface. Starting from an arbitrary initial mapping, the algorithm converges to the optimal mapping by minimizing a global metric taking into account distances between each point and its direct neighbours on the 3-D grid. A similar technique was proposed at the same time by Schwartz et al. [11] for general surface flattening. Since most surfaces are not developable (i.e. unfolded without deformations or cuts, think of a sphere), distortions still remain. Moreover, optimization techniques offer no control on the distribution of the remaining distortions.

A non-distorted mapping of a planar texture onto a 3-D surface is equivalent to a non-warped flattening of the surface. As it is well known, fully spreading out a non-developable surface would induce distortions. The basic idea of this work is to permit discontinuities on the mapped texture and to make a compromise between cuts and distortions. The cuts here play the role of seam lines (such as on a cloth). The surface is first piecewise flattened (with different maps in the plane), then texture is computed on the surface using the flattened parts. This could have many applications in different fields, varying from graphics concerning non-distorted texture mapping, to cartography and manufacturing (cloth modeling) for piecewise flattening. The example emphasised here is a first step towards an efficient and robust CAD tool for shoe modeling.

This paper introduces new techniques for interactive piecewise flattening of parametric 3-D surfaces, leading to a non-distorted texture mapping. The flattening of a region grows around an isoparametric curve selected by hand. A distortion metric is introduced to control and stop the growth when the accumulated distortion exceeds a previously determined threshold. The flattening methods are based on results from differential geometry ([7] [12]), more precisely on the notion of "**geodesic curvature**": isoparametric curves of the surface are mapped in a constructive way onto curves in the texture plane, with preservation of geodesic curvature at each point.

The next section gives the outlines of the global texture mapping approach. This section also reviews our previous work [3] on revolution surface flattening. In section 3, concepts from differential geometry (geodesic curvature) are introduced and utilized to straightforwardly extend the previous work to more general surfaces. A more robust technique (based on a relaxation procedure) is proposed in section 4. All these techniques are compared on well known examples: a cone and a hemisphere. Section 5 presents an application to shoe modeling. The paper concludes with a discussion of the limitations of the proposed techniques and with some suggestions for future developments.

[*] Current address: Institut Français du Pétrole 1 et 4 avenue de Bois-Préau 92506 Rueil-Malmaison, FRANCE.

2 General considerations and previous work

The surfaces considered here are given by a piecewise parametric representation:

$$\begin{cases} X &= x(u,v) \\ Y &= y(u,v) \\ Z &= z(u,v) \end{cases}$$

We require the surfaces to be C^2 (to have continuous second order derivatives at each point), especially on the joining curves (the reason for this constraint is explained later). Moreover, the two families of isoparametric lines must be nowhere tangent to each other (i.e. a normal and a tangent plane exist at every point).
The surface is first regularly sampled into a grid of 3-D points, along the isoparametrics (in parameters space). Moreover, the sampling must be refined enough to approximate the arc length between two successive sample points along an isoparametric by their euclidian distance. In the following, sample curves along u and v directions are respectively denoted $C_{.j}$ ($v = v_j$) and $C_{i.}$ ($u = u_i$). We note the points of the 3-D grid M_{ij}, and their correspondents in the flattening plane[1] P_{ij}. The euclidian distance between two points M_1 and M_2 is loosely denoted $d(M_1, M_2)$ or $||M_2 - M_1||$.

2.1 Outlines of the general approach

The texture mapping algorithm can be divided into two main steps:

I) An initial chord curve (a portion of an isoparametric) is first selected on the grid and the surface is unfolded around this curve until a preliminarily fixed distortion threshold is reached. The same process is repeated on the unprocessed region of the grid until all the surface, or an interesting part of it, is covered.

II) The processed regions of the grid are then triangulated, and a locally affine interpolation, affine in each triangle, is used to texture them.

Step I) constitutes the piecewise flattening part of the algorithm. More precisely, for each presented flattening technique, an appropriate distortion metric is defined. The initial curve is chosen by hand depending on where the texture is desired to be the less distorted. This curve divides the surface into two sub-regions, e.g. "left" and "right". Let the curve be $C_{i_0.}$, where $v_{j_1} < v < v_{j_2}$}.
Unfolding the surface around this curve is done in three steps:

1. Develop the initial curve: find for each sample point M_{i_0j} of this curve a corresponding point P_{i_0j} in the flattening plane (e.g., a texture plane) .
2. develop the surface on the left side of the initial curve: fix a left side threshold, then develop successively curves $\{C_{i.},\ i < i_0, v_{j_1} < v < v_{j_2}\}$ (parallel to the initial one) until the provided distortion exceeds the left side threshold, or the current curve belongs to an already developed region.
3. Develop the surface on the right side of the initial curve using a right side threshold (the same as the leftside development but developed curves are $\{C_{i.},\ i > i_0, v_{j_1} < v < v_{j_2}\}$)

Step II) constitutes the texturing part: the triangulation of a processed region is obtained by splitting each quad $\{M_{i-1j-1}, M_{i-1j}, M_{ij-1}, M_{ij}\}$, of four neighbouring sample points, into two triangles $\{M_{i-1j-1}, M_{ij-1}, M_{i-1j}\}$ and $\{M_{i-1j}, M_{ij-1}, M_{ij}\}$. This gives rise to two triangles in the texture plane, $\{P_{i-1j-1}, P_{ij-1}, P_{i-1j}\}$ and $\{P_{i-1j}, P_{ij-1}, P_{ij}\}$, where the P_{kl} are the corresponding points of the M_{kl} and have already been obtained with the flattening process. The 3-D triangular faces are projected onto the output screen. A triangle-to-triangle affine interpolation from the texture plane to the output screen is used to compute the texture value at each pixel of the output screen. A Z-buffer is used for hidden surface parts elimination and a prefiltered summed table [10] is used for antialiasing. The locally affine approximation of the mapping function is well explained in [4], and compared with the approximation proposed by Oka. and al. in [18] (see [4] for more details). Local affine approximations to a mapping are also discussed in [14].
From now on we will emphasize the geometrical aspect of the non distorted piecewise flattening.

2.2 Previous work

In [3] we have proposed a piecewise flattening technique for surfaces of revolution. The outline of the algorithm is:

1. Map an initial meridian $C_{i_0.}$ of the surface onto a straight line $D_{i_0.}$ in the plane with distance preservation between sample points. *One has only to fix a correspondence for a starting point and a direction for the straight line. Obtaining the other correspondences is immediate.*
2. Extend the development step by step around $C_{i_0.}$ (at each step a meridian $C_{i.}$ is reached) while mapping parallels onto straight lines orthogonal to $D_{i_0.}$ with distance preservation between sample points, until the distortion threshold is reached. (see Figure 1).

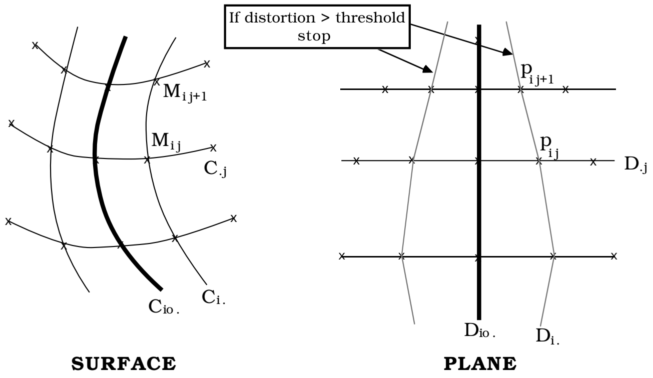

Figure 1: Previous flattening of surfaces of revolution.

This method preserves distances on the initial meridian $C_{i_0.}$ and on parallels $C_{.j}$ but not on the other meridians $C_{i.}$ (unless the surface is a cylinder). Moreover, we preserve the cross angles between $C_{i_0.}$ and $C_{.j}$. Distance distortion on meridians and cross angle distortion (between the two families of curves) increase as one gets far from the first meridian. All distortions are concentrated on meridians $C_{i.}$, $i \neq i_0$, so we choose as a distortion metric for each successive curve $C_{i.}$, the mean of the errors induced on its chord segments:

$$Cr(C_{i.}) = \frac{1}{N-1} \sum_{j=0}^{N-1} \frac{|\, d(M_{ij}, M_{ij+1}) - d(P_{ij}, P_{ij+1}) \,|}{d(M_{ij}, M_{ij+1})} \qquad (1)$$

Here, N is the number of sample points of $C_{i.}$, and P_{kl} are the corresponding texture points of 3-D points M_{kl}. Note that here, due to revolution symmetry, one has only one threshold for both leftside and rightside development.
Figure 2 shows the development of a cone and a hemisphere with this technique. We have developed them entirely, onto only one piece, to emphasize the nature of distortions. Note that there are fewer distortions than with Catmull's technique ([8]) where both surfaces would be mapped onto a rectangle. Moreover, we find for the hemisphere the well known Sanson's projection (pseudocylindric projection) used in cartography.
Although the cone is a developable surface, the algorithm of [3] does not give its proper development. However, this technique gives satisfying results if we develop the surface into sufficiently small pieces, because the meridians of a surface of revolution are geodesics (they have at any point a zero curvature with respect to

[1] For the texture mapping application the flattening plane is considered to be the texture space.

the surface) so they are equivalent to straight lines in the plane. One possible enhancement and extension of the algorithm of [3] to more general surfaces (see Figure 3) consists in computing the geodesic G_1 joining two extremal points of an initial curve and mapping it onto a straight line D_1 in the plane (with distance preservation). Then, for any other point M_{ij}, compute the shortest geodesic G starting from G_1 and joining M_{ij}. Let M be the intersection point of G and G_1. The planar correspondent P of M is obtained by preserving the distance between M and M_2 ($d(M, M_2)$=$d(P, P_2)$). P_{ij} is then given by drawing a straight line D orthogonal to D_1 and by preserving the distance between M and M_{ij} ($d(M, M_{ij})$= $d(P, P_{ij})$).

Figure 2: Cone and hemisphere flattening.

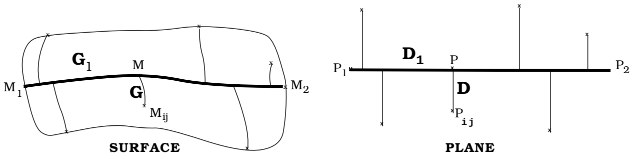

Figure 3: A possible extension using geodesics.

Geodesics on a 3D surface are very attractive because they behave much like straight lines on the plane. Unfortunately, although this class of curves has long been studied in differential geometry [7] and are now well known, their characterization is rather local. Given a curve drawn on a 3-D surface one can determine whether or not it is a geodesic at a given point. The entire curve is then a geodesic if it is geodesic at each point. But what we need here is the inverse problem: find a curve on a surface which is a geodesic between two given surface points. This problem is very difficult because we have to make a global numerical computation using local properties.
Numerical approximations of computation of geodesics proposed up to now [16, 1] are very slow. In addition, they do not always give good results.
In the next section we give an almost equivalent extension avoiding geodesic computation.

3 Geodesic curvature preservation flattening

Instead of mapping onto the plane unknown (and difficult to compute) curves of the surfaces such as geodesics, we use already available isoparametric curves and take into account their topological properties in the mapping process. The main idea of the proposed technique is to map isoparametric curves of the surface onto curves of the plane, with geodesic curvature preservation at sample points and with arc length (i.e. chord length) preservation.
We will first recall some results from the differential geometry of curves in order to define the notion of geodesic curvature, then outline the new general flattening algorithm, and finally give a numerical and constructive algorithm for mapping curves of a surface onto curves of a plane with geodesic curvature and arc length preservation.

3.1 Local properties of curves and surface curves: short review

The results in differential geometry that we use may be found in [7] and [12], both excellent references.
Let C be a curve in R^3 given by arc length parametrization:

$$X(s) = \begin{bmatrix} x(s) \\ y(s) \\ z(s) \end{bmatrix}, s \in [0, a] \subset R,$$

where the Cartesian coordinates $x(s), y(s)$ and $z(s)$ of each point X of C, are differentiable functions of arc length s (the length of C from $X(0)$ to $X(s)$).
To study the local behavior of a curve, a fundamental concept in differential geometry is to use a local frame and to express its local change in its own coordinate system. The Frenet local frame $(X, \mathbf{t}, \mathbf{m}, \mathbf{b})$ is a good candidate for this purpose (see Figure 4). The origin of the frame is point X around which one would like to study the local behavior of the curve. Axes $(\mathbf{t}, \mathbf{m}, \mathbf{b})$ are given by the formulas:

- $\mathbf{t} = X' = \frac{dX}{ds}$
- $\mathbf{m}= \frac{X''}{\|X''\|}$
- $\mathbf{b}=\mathbf{t} \wedge \mathbf{m}$,

where primes denote derivatives with respect to arc length and $\wedge$ denotes the cross product. $\mathbf{t}(s)$ is the *tangent* vector, $\mathbf{m}(s)$ is called *main normal vector* and $\mathbf{b}(s)$ is called *binormal* vector at s. The plane $O(s)$ generated by $\mathbf{t}(s)$ and $\mathbf{m}(s)$ is called the *osculating* plane at s; it is the plane which contains the curve around $X(s)$. One can easily show that variation of the Frenet frame can be expressed through the following formulas:

$$\begin{cases} \mathbf{t}' &= k\mathbf{m} \\ \mathbf{m}' &= -k\mathbf{t} - \tau\mathbf{b} \\ \mathbf{b}' &= \tau\mathbf{m} \end{cases}$$

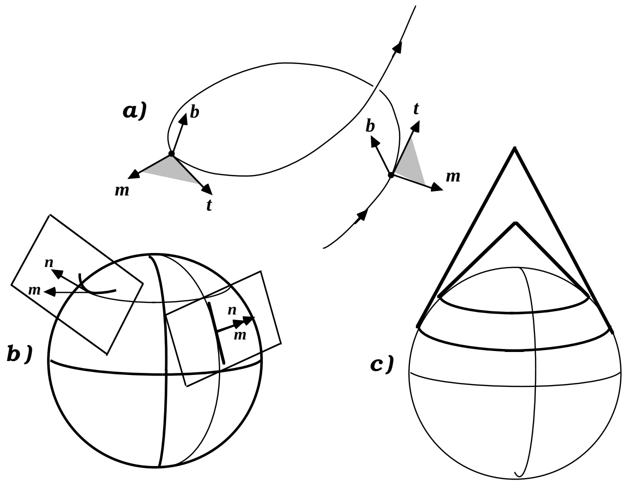

Figure 4: a): Frenet frame variation along a curve, b): Local behavior of a sphere circle, c): cones parallel to a sphere along circles.

The variables $|k(s)|$=$||\mathbf{t}'(s)||$ and $|\tau(s)|$=$||\mathbf{b}'||$ are called respectively *main curvature* and *torsion* at s. Curvature and torsion have an intuitive geometric meaning: let $d\alpha(s)$ and $d\theta(s)$ be respectively the angle between $\mathbf{t}(s)$ and $\mathbf{t}(s + ds)$ and the angle between $\mathbf{b}(s)$ and $\mathbf{b}(s + ds)$, at "consecutive" points $X(s)$ and $X(s + ds)$. Then:

$$|k| = ||\frac{d\alpha}{ds}||, \ |\tau| = ||\frac{d\theta}{ds}||.$$

In other words k and τ are the angular velocities of the tangent and the osculating plane, respectively, as the frame is moved along

the curve with s playing the role of "time".
Moreover, these two variables are independent of parametrization. When k is zero everywhere the curve is a straight line, and when τ is zero everywhere the curve is planar.
Now, consider a surface S given by a parametric function from R^2 to R^3:

$$X(u,v) = \left[\begin{array}{c} x(u,v) \\ y(u,v) \\ z(u,v) \end{array} \right], (u,v) \in [a,b] \times [a,b] \subset R^2,$$

where the Cartesian coordinates x, y, z of a surface point are differentiable functions of u and v. Let us suppose in addition, that the isoparametric lines are nowhere tangent to each other:

$$N(u,v) = X_u \wedge X_v \neq 0 \; \forall u,v \in [a,b]$$

where X_u an X_v are first derivatives vectors according to u parameter and v parameter respectively.
The vector $\mathbf{n}{=}\frac{N}{\|N\|}$ is called the *normal vector* to S at point $X(u,v)$.
The plane $\mathbf{Tp}$ spanned by the set of points Y such that:

$$(X - Y).\mathbf{n} = 0,$$

(where dot denotes the scalar product of vectors) is called the *tangent plane* to surface S at point X.
Let C_S be a curve belonging to surface S and given by arc length parametric function $X(s)$. Recall that the curvature is defined by the acceleration $\mathbf{t}'{=}k\mathbf{m}$ of C_S. This vector can be expressed with two components where the one, $\mathbf{t}'_g$, is tangent to the surface and the other, $\mathbf{t}'_n$, is normal to the surface:

$$\begin{array}{rcl} \mathbf{t}' & = & \mathbf{t}'_g + \mathbf{t}'_n \\ \mathbf{t}'_n & = & (\mathbf{t}'.\mathbf{n})\mathbf{n} \end{array}$$

Seen from a view-point linked to the surface, the acceleration $\mathbf{t}'$ is reduced to tangential component $\mathbf{t}'_g$.
Definition : The *geodesic curvature* k_g of a curve C_S belonging to a surface S, at a point X, is the norm of tangential acceleration[2] of C_S at X according to arc length parameter:

$$|k_g| = ||\mathbf{t}'_g||.$$

k_g corresponds to the curvature of C_S seen from a view-point attached to the surface S. It is different from main curvature k (k_g could be null while k is not null).
C_S is said to be *geodesic at a point* X if and only if k_g is nil at X and C_S is called a geodesic if it is geodesic at every point. One necessary and sufficient condition for C_S to be geodesic at point X is that the main normal vector $\mathbf{m}$ of curve C_S at X is parallel to the normal vector $\mathbf{n}$ to surface S at X.
At any point X the local projection of a curve C_S on the tangent plane along the normal vector to the surface S provides a straight line if C_S is a geodesic, and non-zero curvature (at X) on the planar curve otherwise. Figure 4 shows the behavior of circles of a sphere: a circle is a geodesic if and only if it is a great circle.
Lemma: The *Geodesic curvature k_g of a curve C_S belonging to a surface S at a point X is equal to curvature at X of the planar curve C_{T_p} obtained by locally projecting C_S onto the tangent plane (*$\mathbf{Tp}$*) along the normal vector to surface S at point X.*
As for main curvature, intuitively geodesic curvature $k_g(s)$, at a point $X(s)$, is the angular velocity of tangents to the **resulting projected** curve $C_{T_p}(s)$ according to arc length s.
This Lemma leads us to an efficient and constructive numerical way of mapping a chord line of a surface onto a chord line in a plane with preservation of chord length and geodesic curvature. This is the basis of the new flattening algorithm described below.

3.2 Outline of the new general algorithm

The new algorithm runs as follows (see Figure 5):

1. Map the initial selected curve $C_{i_0.}$ of the surface (for instance $\{u = u_{i_0},\ j_1 \leq j \leq j_2\}$) onto a curve in the plane with geodesic curvature preservation at sample points and with arc length preservation (*distance preservation between any pair of successive sample points*).
2. Extend step by step the development on the left side of $C_{i_0.}$. At each step one reaches a curve $C_{i.}$, $(i < i_0)$ while mapping transversal curves $C_{.j}$ $(\{v = v_j,\ j_1 \leq j \leq j_2\})$ onto curves in the plane with geodesic curvature and arc length preservation. At the same time, one requires preservation of the cross angle between the initial curve $C_{i_0.}$ and each transversal curve $C_{.j}$. The process is stopped when the left side distortion threshold is reached, or curve $C_{i.}$ belongs to an already flattened region.
3. Extend the development on the right side of $C_{i_0.}$ according to the right side threshold (same process as the left side one).

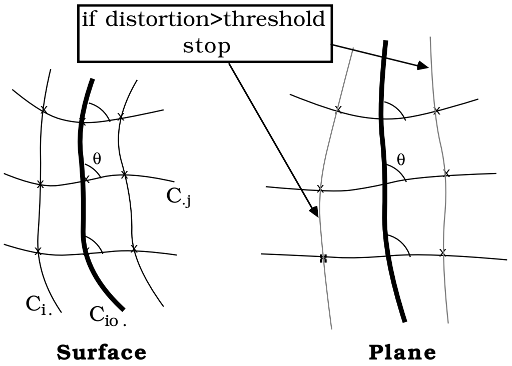

Figure 5: Geodesic curvature preservation flattening.

Notice that here again distances are preserved along the initial curve $C_{i_0.}$ and in the transversal ones $C_{.j}$. All distortions are concentrated on curves $C_{i.}$ parallel to $C_{i_0.}$. So, the distortion metric for a specific curve $C_{i.}$ is the same as for the previous technique (formula (1)).

3.3 Mapping a 3D surface curve onto a planar curve with arc length and geodesic curvature preservation

Let us recall that the surface is sampled and that surface curves are given by chord lines, arc length between two consecutive points being approximated with euclidian distance. Suppose the curve C that we want to map onto the plane contains $n+1$ sample points M_i, $i = 0..n$. Let us denote by $\mathbf{n_i}$ and $\mathbf{Tp}_i$, respectively, the normal vector and the tangent plane to the surface at point M_i. The curve flattening algorithm can then be described as follows:

i) Map the first curve segment M_0M_1 onto a segment P_0P_1 in the plane (let us call this plane Oxy) such that $d(M_0, M_1) = d(P_0, P_1)$. It is sufficient to fix an initial point P_0 and a direction in the plane.

ii) For each j, $2{\leq}j{\leq}n$, P_j is iteratively computed in the plane as follows (see Figure 6):

   1. Project M_j and M_{j-2} onto the tangent plane to the surface at M_{j-1}. This provides two points in $\mathbf{Tp}_{j-1}$, called $\tilde{M}_j$ and $\tilde{M}_{j-2}$ and given by the formulas:

   $$\tilde{M}_j = M_j + ((M_{j-1} - M_j).\mathbf{n}_{j-1})\mathbf{n}_{j-1}.$$

   $$\tilde{M}_{j-2} = M_{j-2} + ((M_{j-1} - M_{j-2}).\mathbf{n}_{j-1})\mathbf{n}_{j-1}.$$

   2. Use a dilation in $\mathbf{Tp}_{j-1}$ to transform $\tilde{M}_j$ into a point M'_j such that $d(M_{j-1}, M_j){=}d(M_{j-1}, M'_j)$.

   $$M'_j = M_{j-1} + \frac{||M_j - M_{j-1}||}{||\tilde{M}_j - M_{j-1}||}(\tilde{M}_j - M_{j-1})$$

[2] That is the reason why we require a surface to be C^2.

3. As P_{j-2} and P_{j-1} are already computed, the desired point P_j is the point of Oxy that preserves simultaneously the angle θ_{j-1} between $\tilde{M}_{j-2}M_{j-1}$ and $M_{j-1}M'_j$, and the distance $d(M_{j-1}, M'_j)$.

$$(P_{j-2}\widehat{P_{j-1}}P_j) = (\tilde{M}_{j-2}\widehat{M_{j-1}}M'_j)$$
$$d(P_{j-1}, P_j) = d(M_{j-1}, M'_j)$$

The way we obtain p_j is to first compute coordinates (x'_1, x'_2) of M'_j according to the local orthogonal frame (M_{j-1}, e'_1, e'_2) in $\mathbf{Tp}_{j-1}$ where axes e'_1 and e'_2 and the coordinates are given by the formulas:

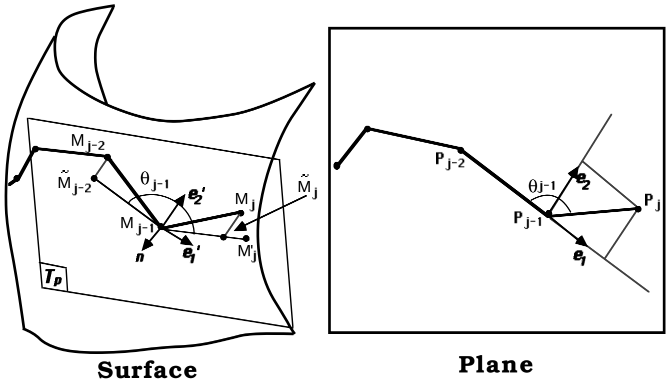

Figure 6: Mapping a curve of a surface onto a planar curve: step *ii*).

$$\begin{cases} e'_1 = \frac{M_{j-1} - \tilde{M}_{j-2}}{\|M_{j-1} - \tilde{M}_{j-2}\|} \\ e'_2 = \mathbf{n}_{j-1} \wedge e'_1 \\ x'_1 = (M'_j - M_{j-1}).e'_1 \\ x'_2 = (M'_j - M_{j-1}).e'_2 \end{cases}$$

P_j is the point of Oxy having the same coordinates according to the orthogonal and positive frame (P_{j-1}, e_1, e_2) given by:

$$\begin{cases} e_1 = \frac{P_{j-1} - P_{j-2}}{\|P_{j-1} - P_{j-2}\|} = a\tilde{\imath} + b\tilde{\jmath} \\ e_2 = -b\tilde{\imath} + a\tilde{\jmath} \\ P_j = P_{j-1} + (x'_1 e_1 + x'_2 e_2) \end{cases}$$

where $(O, \tilde{\imath}, \tilde{\jmath})$ is the canonical coordinate system of Oxy.

Step *ii*) of this algorithm will be used in other circumstances, in what follows . It can be thought of as an operator $\mathcal{P}$. Given three neighbouring surface points (M_1, M_2, M_3) and two corresponding points (P_1, P_2) of (M_1, M_2) in the flattening plane Oxy, $\mathcal{P}$ computes point P_3 in Oxy, that preserves the distance $d(M_2, M_3)$ and the projection of the angle $\theta_2 = (M_1\widehat{M_2}M_3)$ in $\mathbf{Tp}_{M_2}$. We will call this operator the *angle preserver* and we will write:

$$P_3 = \mathcal{P}_{\theta_2}(M_3)$$

Theorem: *The above curve flattening algorithm preserves geodesic curvature and arc lengths within the chord line approximation.*

As we initially have $d(P_0, P_1) = d(M_0, M_1)$ and by construction of P_j, j, $2 \leq j \leq n$, we have $d(P_{j-1}, P_j) = d(M_{j-1}, M_j)$, so arc lengths are preserved.

As the sampling is sufficiently refined, the tangents to a curve can be approximated with chord segments. So, at step *ii*), θ_{j-1} can be considered as the angle variation between two "consecutive" tangent vectors to the locally projected curve, into $\mathbf{Tp}_{j-1}$. We preserve θ_{j-1} and chord lengths. We then preserve locally the angular velocity of tangents to the curve projection into $\mathbf{Tp}_{j-1}$. **(Q.E.D)**.

Notice that the computation of P_j involves geodesic curvature preservation at M_{j-1}.

3.4 Preserving the cross angles between the initial curve and a transversal curve

In fact, the preserved angles are the cross angles between the local projection of the two curves onto the tangent plane at their intersecting point (see Figure 7). Suppose that the initial curve $C_{i_0}.$ and a transversal curve $C_{.j}$ meet at the point M_{i_0j}. Let us denote by M_{i_0j-1}, M_{i_0j+1} and by M_{i_0-1j}, M_{i_0+1j}, respectively, the neighbours of M_{i_0j} along the u and v direction. When we project these four neighbours onto $\mathbf{Tp}_{i_0j}$, we obtain a quadrilateral $(\tilde{M}_{i_0j-1}, \tilde{M}_{i_0-1j}, \tilde{M}_{i_0j+1}, \tilde{M}_{i_0+1j})$. Four angles must then be preserved:

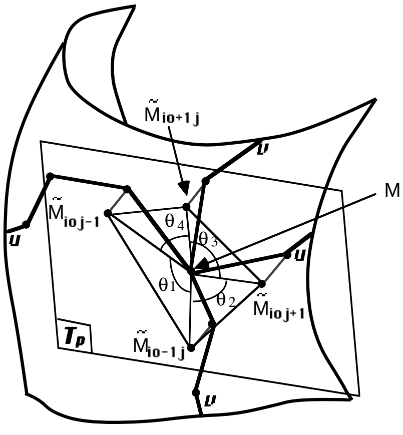

Figure 7: Cross angle preservation.

$$\theta_1 = (\tilde{M}_{i_0j-1}\widehat{M_{i_0j}}\tilde{M}_{i_0-1j}) \;\; \theta_2 = (\tilde{M}_{i_0-1j}\widehat{M_{i_0j}}\tilde{M}_{i_0j+1})$$

$$\theta_3 = (\tilde{M}_{i_0j+1}\widehat{M_{i_0j}}\tilde{M}_{i_0+1j}) \;\; \theta_4 = (\tilde{M}_{i_0+1j}\widehat{M_{i_0j}}\tilde{M}_{i_0j-1})$$

As the initial curve $C_{i_0}.$ is already mapped, the points $P_{i_0j-1}, P_{i_0j}, P_{i_0j+1}$ are available in the flattening plane. In addition, we have $(P_{i_0j-1}\widehat{P_{i_0j}}P_{i_0j+1}) = (\tilde{M}_{i_0j-1}\widehat{M_{i_0j}}\tilde{M}_{i_0j+1})$ (geodesic curvature preservation at M_{i_0j}, on $C_{i_0}.$). We then only have to preserve θ_1 and θ_4, respectively, when we initialize the mapping of $C_{.j}$ on the left side of $C_{i_0}.$ (when we compute P_{i_0-1j}) and on the right side of $C_{i_0}.$ (when we compute P_{i_0+1j}). This implies the preservation of all the other angles. P_{i_0-1j} and P_{i_0+1j} are then given by the angle preserver operators:

$$P_{i_0-1j} = \mathcal{P}_{\theta_1}(M_{i_0-1j}), \;\; P_{i_0+1j} = \mathcal{P}_{\theta_4}(M_{i_0+1j})$$

Moreover, this preserves the angle $(\tilde{M}_{i_0-1j}\widehat{M_{i_0j}}\tilde{M}_{i_0+1j})$, hence, geodesic curvature at M_{i_0j} on $C_{.j}$.

3.5 Flattening of the cone and the hemisphere

Figure 8-a and 8-b show the flattenings of the cone with this technique. In 8-a, the cone is spread out around a generatrix. In 8-b a circle is used as initial curve. Both give the same result, which is the proper development of the cone (an angular sector of a disc).

In Figure 9, we show the mapping of a checkerboard pattern onto the cone according to three different techniques: With Catmull's technique, in 9-a and 9-b, squares are compressed along circles as one gets close to the tip. The advantage of this technique is that one can manage to avoid the line of discontinuities with repeated patterns.

The second mapping (9-c and 9-d) is based on the flattening technique described in section 2.2 (previous work). One can see that distortions are not very noticeable beside the initial generatrix (9-c). Squares are more and more stretched along meridians (the inverse of flattening distortions impression) and angles more distorted, beside the cutting line (9-d).

The mapping based on geodesic curvature flattening (9-e and 9-f) provides no distortions but the texture contains a line of discontinuities (9-f). Thus it is topologically faithful.

Figure 8: a) and b): Development of a cone around a meridian and a parallel respectively. c) and d) Development of a hemisphere around a meridian and the equator respectively.

The cone contains 1281 points. Its flattening takes about 0.05s with the algorithm of section 2.2 and 0.5s with the geodesic curvature preservation algorithm on a GOULD 9000. Mapping texture from the flattened cone has taken about 25s.

The full development of the hemisphere around the central meridian is shown in Figure 8-c. The initial meridian and the equator are mapped onto straight lines, because both are geodesic curves. The parallel circles are mapped onto circles, because along each parallel circle there exists a cone tangent to the sphere: as along the circle the tangent planes are the same according to the sphere or to the cone, at each point geodesic curvature is the same according to the two surfaces. So, flattening of a parallel circle of the sphere gives the same result as if it were considered as belonging to the cone, hence a circle. One can notice that with the latter flattening technique, angles are less distorted than with the technique of section 2.2 (Figure 2). The same flattening is obtained with Bonne's projection (or pseudo-cylindric projection), also used in cartography. In [2] it is shown that, starting from different initial curves, geodesic curvature preservation flattening provides other types of projections used in cartography. For instance, when the initial curve is not a great circle, the development provides a cone (a conic projection). The hemisphere has about 300 points. Its flattening takes about 0.008s with the technique described in section 2.2 and has taken about 0.2s with the geodesic curvature preservation algorithm.

The major disadvantage of this technique is due to preservation of geodesic curvature in only one direction during the development process. This concentrates all the distortions on the curves parallel to the initial one. This also makes the technique strongly dependent on the initial curve. Figure 8-d shows the development of the hemisphere around the equator. One obtains a rectangle: the equator is a geodesic, and is mapped onto a straight line. All the meridians being geodesics, they are also mapped onto straight lines orthogonal to the first one and having the same length. However this technique remains suitable for all developable surfaces (or developable parts of a surface). It also gives good results on almost developable surfaces.

In the next section, we present a technique that takes into account geodesic curvature in both directions and then reduce the drawbacks encountered with non-developable surfaces.

Figure 9: Mapping a checkerboard onto a cone with different techniques.

4 Incorporating geodesic curvature in both directions: Relaxation technique

The technique described in this section consists of distributing distortions in both directions. This is done in two steps:

1. One first develops the surface around an initial curve, taking into account at each point the geodesic curvature in both directions u and v and the projected cross angles. As will be explained later, this development induces a gradient of distortions on the flattened region. This gradient enables one to measure the distortions and stop the development propagation when necessary.
2. A relaxation procedure is then used to reduce and better distribute the distortions in the flattened region.

   For simplicity, in the following, the projected angle (in the tangent plane) between two intersecting curves is loosely called the angle.

4.1 Development technique

The new development algorithm is almost the same as the algorithm of the previous section. One first maps the initial curve C_{i_0}. onto the plane with geodesic curvature and arc length preservation. The development of the region is then propagated step by step to the curves parallel to $C_{i_0 .}$, on the left side of $C_{i_0 .}$, then on the right side of $C_{i_0 .}$. The new feature introduced here is the way in which the points of the parallel curves are mapped onto the flattening plane.

As illustrated in Figure 10, let $M_{ij} = C_{i.} \cap C_{.j}$ be the point being processed. P_{ij} is obtained by preserving at each neighbour $M_{kl} \in \{M_{i-1j}, M_{ij+1}, M_{i+1j}, M_{ij-1}\}$ already processed, the three angles $(\theta^1{}_{kl}, \theta^2{}_{kl}, \theta^3{}_{kl})$. $\theta^1{}_{kl}$ and $\theta^2{}_{kl}$ are cross angles (facing M_{ij}) between the curves that intersect at M_{kl}. Preserving $\theta^3{}_{kl}$ is equivalent to preserving the geodesic curvature at M_{kl} on the curve containing M_{ij} and M_{kl}. One can not always preserve all three angles. For instance, in Figure 10 the point M_{i+1j-1} has not yet been processed, so one cannot preserve $\theta^2{}_{ij-1}$. Each angle preservation provides a different point in the flattening plane. This point is obtained using the angle preserver:

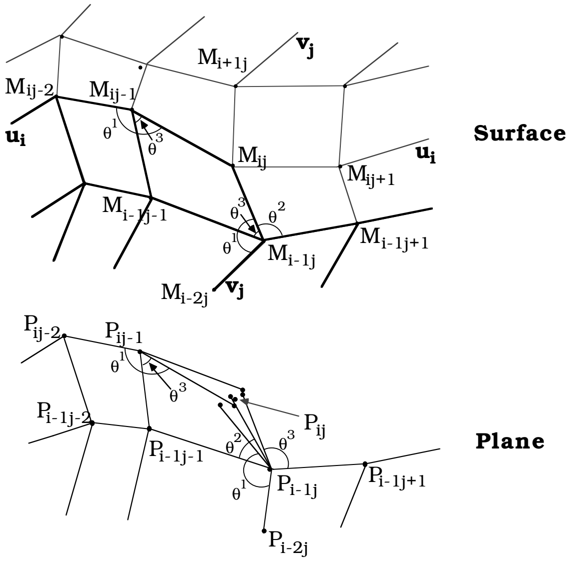

Figure 10: Preserving angles at processed neighbours (development step).

$P_{ij}^{r,kl} = \mathcal{P}_{\theta^r{}_{kl}}(M_{ij})$, $r = 1, 2, 3$; $kl \in \{i-1j, ij+1, i+1j, ij-1\}$

The point P_{ij} corresponding to M_{ij} in the flattening plane is then the centroid of the points[3] $P_{ij}^{r,kl}$.

The choice of the centroid induces slight errors on the angles and distances. The accumulation of these errors provides a gradient of distortions in the scanning direction. So, for a better distribution of the distortions the curve being processed is not scanned from one extremal point to the other extremal point. Instead, the curve is scanned from the central point to the extremal points.

With this technique distortions are present on both $C_{i.}$, $C_{.j}$ curves. The distortions increase in diagonal directions as one gets far from the central point of the initial curve. Let $v_{j_1} \leq v \leq v_{j_2}$ on the initial curve $C_{i_0 .}$. The distortion metric for a specific curve $C_{i.}$ is then:

$$C(C_{i.}) \quad = \quad \frac{1}{2I+J}\Big(\sum_{k=j_1}^{k=j_2-1} \frac{||d(M_{ik}, M_{ik+1}) - d(P_{ik}, P_{ik+1})||}{d(M_{ik}, M_{ik+1})} + \sum_{k=i_0}^{k=i-1} \frac{||d(M_{kj_1}, M_{k+1j_1}) - d(P_{kj_1}, P_{k+1j_1})||}{d(M_{kj_1}, M_{k+1j_1})} + \sum_{k=i_0}^{k=i-1} \frac{||d(M_{kj_2}, M_{k+1j_2}) - d(P_{kj_2}, P_{k+1j_2})||}{d(M_{kj_2}, M_{k+1j_2})}\Big)$$

Here one has $I{=}i - i_0$ and $J{=}j_2 - j_1$.

4.2 Relaxation procedure

In the above development, when mapping a point onto the flattening plane, one does not take into account all the neighbouring points (see figure 11). The reason is that some neighbours have not been processed yet. In addition, for a given neighbour one can not always preserve all the angles. The relaxation procedure consists of recomputing points of the obtained flat piece several times until the change becomes insignificant. At each iteration one uses the results of the previous iteration. The location of a point P_{ij}^n at iteration n is then:

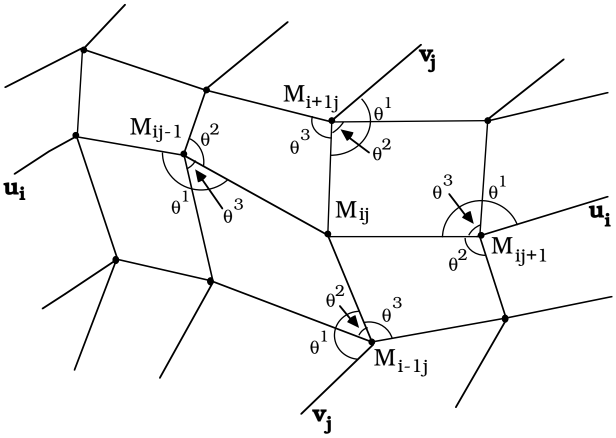

Figure 11: Preserving three angles at four neighbours (relaxation step).

$$P_{ij}^n = \frac{1}{12} \sum_{r=1}^{3} (\mathcal{P}^{n-1}_{\theta^r{}_{i-1j}} + \mathcal{P}^{n-1}_{\theta^r{}_{i+1j}} + \mathcal{P}^{n-1}_{\theta^r{}_{ij-1}} + \mathcal{P}^{n-1}_{\theta^r{}_{ij+1}})\,(M_{ij}).$$

P_{ij}^n is thus the centroid of the twelve points obtained by preserving the three angles of each neighbour. The angle preserver $\mathcal{P}^{n-1}$ uses the flattening points of iteration $n-1$.

Let $P_1, P_2, \cdots, P_n$ be a cluster of n points within a plane and let P be a point in this plane. One calls the "dispersion" of the cluster around P the value:

$$Dsp(P) = \frac{1}{n} \sum_{k=1}^{n} ||P - P_k||^2.$$

$Dsp(P)$ is minimum for the centroid P_m of the cluster of points. $Dsp(P_m)$ is then called the dispersion of the cluster of points. For our purposes, the quality of the flattening at a given point P_{ij} can be measured by the dispersion $Dsp(P_{ij})$ of the twelve points given by the twelve angle preservers. The smaller $Dsp(P_{ij})$ is, the better is the flattening at P_{ij}. Thus taking the centroid of the twelve points at each iteration is better for distributing and reducing the distortions.

The quality of the whole flattening can be measured by the mean dispersion:

$$Dsp = \frac{1}{N} \sum_{ij} Dsp(P_{ij}).$$

Here N is the number of points within the flattened piece. Thus, the relaxation procedure stops when the dispersion Dsp^n at iteration n has not changed significantly, i.e. when $\frac{|Dsp^n - Dsp^{n-1}|}{Dsp^n}$ becomes less then some fixed variation threshold (about 10^{-3}).

The relaxation algorithm can thus be simply described as follows:

```
$Dsp^{n-1} = Greatnumber$
$Dsp^n = Smallnumber$

While $\frac{|Dsp^n - Dsp^{n-1}|}{Dsp^n} > threshold$
```

[3] When the surface is developable the preservation of each angle gives the same point in the flattening plane.

```
        For each i,j indexing the region points
$P^n_{ij} = \frac{1}{12} \sum_{r=1}^{3} (\mathcal{P}^{n-1}_{\theta^r{}_{i-1j}} + \mathcal{P}^{n-1}_{\theta^r{}_{i+1j}} + \mathcal{P}^{n-1}_{\theta^r{}_{ij-1}} + \mathcal{P}^{n-1}_{\theta^r{}_{ij+1}}) (M_{ij})$
            $Dsp^{n-1} \leftarrow Dsp^n$
            $Dsp^n = \frac{1}{N} \sum_{ij} Dsp(P^n_{ij})$
        endFor
endWhile.
```

4.3 Flattening of the sphere

Figures 12-a and 12-b show the flattening of the hemisphere around the central meridian before (12-a) and after (12-b) the relaxation process. With the relaxation technique, it is clear that distortions are better distributed and angles more preserved than previously (i.e., with the geodesic curvature flattening preservation in only one direction).

Figures 12-c and 12-d show the flattening of the hemisphere around the equator before (12-c) and after (12-d) the relaxation process. One can notice that after the relaxation, the flattening is almost the same as when the initial curve is a meridian. Starting from the development around the meridian the relaxation has taken 6 iterations and 3 seconds. Starting from the development around the equator the relaxation has taken about 16 iterations and 15s.

Figure 12: Development of a hemisphere - a: around a meridian before and after relaxation b: around the equator before and after relaxation.

Figure 13 highlights the compromise between discontinuities and distortions. A digitized photograph (in Figures 13-a and 13-b) and an artificial material (in Figures 13-c and 13-d) are mapped onto the sphere. Catmull's technique is used in the pictures on the left: patterns are strongly deformed beside the pole, but there are no discontinuities. In the pictures on the right the mapping is based on the relaxation procedure. The sphere is segmented into four equal pieces, a meridian being chosen for each piece. Distortions are hardly noticeable, but seam lines are evident in the mapped photograph. In the mapped material, seam lines are not quite visible but one can guess where they are because of the sudden changes of orientation.

Figure 13: Comparison on the sphere of Catmull's mapping and the mapping based on the piecewise relaxation.

5 Application: shoe modeling

One of the possible applications of our mapping and texturing techniques is computer-aided shoe design. Shoemakers naturally use geodesics to cut patterns, by drawing lines on a shape of the shoe: these lines become the edges of the flattened 3-D patterns constituting the different pieces of the resulting shoe.

To determine the pattern of a region (see picture of Figure 14-a), the shoemaker sticks a paper strip on a median curve of the region to flatten. He then cuts the sheet from this line in fine "fish bones"; each of them is folded back on the shoe form, determining a geodesic on the surface. Finally, the trace of the region border is marked on each bone, giving the edge of the flattened zone. One notes that the choice of the geodesics is here completely empirical, and therefore difficult to automate. Our method enables one to overcome this difficulty.

Figure 14-b shows a wire frame of a shoe shape. A real wooden shoe pattern has first been sampled by a laser sensor along successive slices. Each slice has been approximated by a *spline* curve. A spline surface (of NURBS type) has then been generated by transverse interpolation between splines. The parametric surface has then been sampled along isoparametrics. The parametric surface modeling algorithm is explained in detail in [15].

In Figure 14-b, the shoe model is being mapped: the pink zone is already flattened, while the yellow side is currently treated: the starting curve for development is drawn in green. The sole (in white) is not treated yet.

As described above, we use isoparametrics as cut lines: starting from a given curve, we develop the shoe surface until a deformation threshold is reached. As the whole surface is not developable, we can play on the width of the pieces by varying the distortion threshold.

The shoe form is cut into three pieces (which flattenings are shown in Figures 14-c, 14-d and 14-e): the sole, the interior and the exterior sides. The sole, as relatively flat, is flattened with the simple geodesic curvature flattening. The relaxation process has been applied to both sides.

Figures 14-f and 14-g show, from different viewpoints, the entirely textured shoe shape, obtained by mapping on it successively a digitized natural leather and an artificial weaving.

Figures 14-h and 14-i show the modeling of a sandal and its flattened pieces. Small pieces have been obtained with the geodesic curvature preservation algorithm. The big piece has needed the re-

laxation algorithm. Finally, one can see the sandal textured with the leather and the artificial material in Figures 14-j and 14-k.
Each of the shoe models has taken less than 3s computation time for flattening the pieces. The affine interpolation for texture mapping has taken between 15 and 20s.
From a practical point of a view, an efficient CAD tool should enable one to draw "manually" the region edge curves on the 3-D surface. Our techniques could then be used for the flattening of each region, by developing the parametric pattern containing the selected region, projecting the edges on the 2-D mapping, and finally cutting the plane along these borders. In this case, the choice of the initial development curve could even be automatic: the user needs only to know the distortion rate induced on each piece of the shoe. This will be pursued in future work. The algorithm of drawing curves on 3-D surfaces is described in [15].

6 Conclusion

We have presented in this paper new and efficient algorithms for non-distorted texture mapping. Unlike more conventional approaches based on global minimization of distortions, our techniques enable a controlled unfolding around an initial curve by choosing a distortion metric on isoparametrics of the surface. Moreover, distortions are lessened by introducing discontinuities on the unfolded surface. Possible applications (among others) could be umbrella and underwear designing, and, more generally, manufacturing.
The new algorithms are easy to implement, although they are based on uncommon concepts (from differential geometry); nevertheless, they present several aspects for further study. First, a human intervention for the choice of the initial curve and the distortion threshold is necessary. Also, our techniques can only be used on surfaces given explicitely by their parametric equations, thus reducing their scope. The generalization to polygonal surfaces would then be desirable. Another disadvantage is that seam lines (cuts) are located on isoparametric curves, and thus depend on the parametrization chosen. On most natural objects covered with planar texture (clothes and walls for example), seam lines are located on lines of main curvature, giving a harmonious look to the mapping. It could then be useful to parametrize the surface again along the main directions ([1]) before flattening it. This is not always true, though, and in certain cases (shoe modeling for example) aesthetics are important. This is a very subjective notion; drawing edge curves by hand on the surface becomes necessary in such cases.
Another interesting problem consists of how to reduce as much as possible the number of cut pieces . A preliminary idea would be to extend a previously mapped piece, by choosing its borders as being the initial curves of the flattening algorithms described in this paper. Another solution consists of finding a strategy to merge different pieces previously obtained. This is still an open problem.
The last point that could be explored is the texture orientation: how can one locate and orient the several flattened pieces in the texture plane, in order to obtain a good appearance at seam lines. A possible solution would consist of minimizing a global metric of positions and orientations on the common borders of the unfolded parts.

Acknowledgements

The authors are very grateful to Dr. D. Geman, Dr. J. Ralston and Dr. P. Sander for revising the paper; to Dr. M. Gangnet and Dr. F. Schmitt for their valuable comments; to L. Doghman, F. Ledru and L. Vinet for their precious help; and to Dr. A. Gagalowicz for his support.

References

[1] J.M. Beck, R.T. Farouki, and J.K. Hinds. Surface analysis methods. *IEE CGA*, pages 18–37, December 1986.

[2] C. Bennis. Synthèse de textures hiérarchiques planes - développement de surfaces 3d pour un placage de textures minimisant les distorsions. *Thèse de Doctorat en Science, Université de Paris XI, centre d'Orsay*, Décembre 1990.

[3] C. Bennis and A. Gagalowicz. Hierarchical texture synthesis on 3-d surfaces. *EUROGRAPHICS' 89*, pages 257–268, September 1989.

[4] C. Bennis and A. Gagalowicz. Mapping de textures sur une approximation triangulaire des surfaces. *PIXIM' 89*, pages 139–152, 1989.

[5] E. Bier and K. Sloan. Two-part texture mapping. *IEEE Computer Graphics and Applications*, pages 40–53, September 1986.

[6] J.F. Blinn and M.E. Newell. Texture and reflection in computer generated images. *Communications of the ACM, 19, 10*, pages 542–547, October 1976.

[7] M.F. Do Carmo. Differential geometry of curves and surfaces. *Prentice-Hall, Englewood Cliffs, Inc.*, 1976.

[8] E. Catmull. A subdivision algorithm for computer display of curved surfaces. *Ph.D. Dissertation. Dept. of Computer Sciences, University of Utah*, December 1974.

[9] E. Catmull and A.R. Smith. 3-d transformation of images in scanline order. *Computer Graphics*, 14(3), July 1980.

[10] F.C. Crow. Summed-area tables for texture mapping. *SIGGRAPH 84, Proc. of Computer Graphics*, pages 207–212, July 1984.

[11] E.L. Schwartz et al. Computational neuroscience: Applications of computer graphics and image processing to 2d and 3d modelling of functional architechture of visual cortex. *CGA, Vol. 8, No. 4*, pages 13–23, July 1988.

[12] G. Farin. Curves and surfaces for aided geometric design. *Academic Press, San Diego, Inc.*, 1988.

[13] E. Fiume, A. Fournier, and V. Canale. Conformal texture mapping. *EUROGRAPHICS' 87*, pages 53–64, 1987.

[14] P. Heckbert. Fundamentals of texture mapping and image warping. *UCB/CSD 89/516, Computer Science Dept, Univ. of California, Berkeley.*

[15] G. Iglesias and S. Coquillart. Curve design on surfaces. *In preparation.*

[16] S.D. Ma and A. Gagalowicz. Determination of local coordinate systems for texture synthesis in 3-d surface. *EUROGRAPHICS'85*, September 1985.

[17] S.D. Ma and H. Lin. Optimal texture mapping. *EUROGRAPHICS'88*, pages 421–428, September 1988.

[18] M. Oka, K. Tsutsui, A. Ohba, Y. Kurauchi, and T. Tago. Real-time manipulation of texture-mapped surfaces. *SIGGRAPH 87, Proc. of Computer Graphics*, 21(4):181–188, 1987.