Chapter 8

Re-Synchronization of Spectral Watermarks by Umbilical Points

8.1 Introduction

In this Chapter, we propose a blind watermarking scheme based on automatic feature points detection. The irregular sampling of 3D shapes is a challenging issue for extending well-known signal processing tools. 3D shape watermarking schemes have to resist to common resampling operations used for example in some compression applications. We propose an automatic selection of intrinsic feature points that are robust against surface remeshing. They are detected as multi-scale robust degeneracies of the shape curvature tensor field. The impact of the sampling on the curvature estimation is studied. These points are then used as seeds in the partition of the shape into fast approximated geodesic triangles. Each of them is then remeshed with a regular connectivity and watermarked in the mesh spectral domain. The watermark embedding is performed by spectral decomposition like in Chapter 7. The watermark perturbations computed on the remeshed triangles are then projected on the original points of the 3D object. We discuss the robustness of the feature points and of the overall scheme under various watermarking attacks.

Here we assume that input 3D objects are *free-form manifold triangle meshes*. Free-form stands for surfaces that are not globally composed of quadrics (spheres, cylinders, planes etc.) which do not present local key features. The blind watermarking scheme operates as follows. Robust feature points are first detected as intrinsic singularities of the curvature tensor. The curvature is estimated on the polyhedral surface using a multi-scale variant of a state of the art method based on normal cycle theory [37]. We show that these feature points can be accurately retrieved after affine transforms, noise addition and mesh sampling resolution changes.

The second step of the scheme is a unique partition of the surface driven by the selected feature points. The partitioning is necessary since the memory size of 3D models becomes unaffordable for most geometry processing tools. In order to guarantee the unicity of this partition we chose a geodesic Delaunay triangulation that in addition produces well shaped triangular patches.

In the last two steps each curved triangle patch is then remeshed with a regular connectivity and finally watermarked in the mesh spectral domain (see Chapter 7). The repetition of the watermark in each patch ensures that if some feature points are lost after a malicious attack, the watermark will still be retrieved in the patches that are not adjacent to these points.

The watermark retrieval is performed in the same way and does not need the availability of the original content. The use of robust feature points for driving the embedding of information in a surface leads to a scheme that is robust against white noise, affine transforms, remeshing and cropping.

This Chapter is organized as follows. The first Section presents the automatic detection of feature points and discusses their robustness. The second Section is devoted to the geodesic partition of the mesh based on the feature points. The next Section presents the embedding and detection procedures. The last Section concludes the Chapter with a study of the limitations of the proposed method as well as a discussion of future research lines.

8.2 Feature Points

Feature lines and feature points are common ways of describing intrinsic characteristics of 3D surfaces [59, 68]. Feature lines may be classified as intensity discontinuity lines or curvature discontinuity lines. Discontinuity lines are usually known as *crease* and *border* lines and are composed of connected sharp edges where the surface normal presents an abrupt change of direction. Curvature discontinuity lines such as *valleys* and *ridges* are identified as the loci of points where the normal curvature assumes an extreme value along a line of curvature. Feature points are generally defined by the bifurcations or end points of feature lines.

We introduce *umbilical points* as feature points for the general case of triangle meshes in this paper. These points have already been used for fingerprinting and surface matching purposes by Ko et al. [83] in the particular case of NURBS surfaces. Umbilical points present the same curvature in all directions of the tangent plane and are topological singularities of the curvature tensor field [123]. They can also be found as the intersections or bifurcations of feature lines of the curvature tensor field also known as *separatrices*. We show that carefully selected umbilical points are good candidates for our watermarking scheme as they are robust against usual watermarking attacks.

8.2.1 Umbilical Points

The detection of umbilical points is based on a robust estimation of the curvature tensor of the mesh (see Chapter 2). Many methods have been developed to estimate gaussian and mean curvature of triangle meshes but they usually do not provide an accurate and robust estimation of the principal curvature directions or of the more general curvature tensor (a good survey can be found in [106]).

We exploit the *normal cycle* theory [37] to estimate the curvature tensor in each point of the surface. This choice is motivated by the provable good convergence properties (for the particular case of Delaunay triangulations) of this estimator. In addition we extend this method by performing a multi-scale analysis of the curvature that leads to a better estimation of the umbilical points as illustrated below.

8.2.2 Curvature Tensor Estimation

As seen in Chapter 2, the curvature tensor $\boldsymbol{T}$ at a point p of a surface S is symmetric and given by:

$$\boldsymbol{T}(p) = \begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = t^T \boldsymbol{T} t, \tag{8.1}$$

where $t = (u, v)$ is a tangent vector of S at v described in parameter coordinates. This tensor is also commonly expressed in 3D by:

$$\boldsymbol{T'}(p) = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix}^T \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} = t'^T \boldsymbol{T'} t, \tag{8.2}$$

where $t' = (u, v, 0)$ the third coordinate is in the normal direction of S at p. The eigenvalues λ_1, λ_2 of $\boldsymbol{T}$ are the minimal and maximal curvatures and the eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2$ of $\boldsymbol{T}$ are the corresponding principal curvature directions $\boldsymbol{\kappa_{min}}, \boldsymbol{\kappa_{max}}$ in the tangential frame.

$$\lambda_1 = \kappa_{max}, \lambda_2 = \kappa_{min}, \boldsymbol{e}_1 = \boldsymbol{\kappa_{max}}, \boldsymbol{e}_2 = \boldsymbol{\kappa_{min}}, \tag{8.3}$$

with $|\lambda_1| \leq |\lambda_2|$. $\boldsymbol{T'}$ is expressed in the 3D frame with coordinates (u, v, w), u, v axes belong to the tangential plane at p and w axis is the normal direction at p. $\boldsymbol{T}'$ has an eigenvalue equal to zero and the two others are the same as the eigenvalues of $\boldsymbol{T}$: $\lambda_1 = \kappa_{max}, \lambda_2 = \kappa_{min}, \lambda_3 = 0$. Two eigenvectors point towards the principal directions like for $\boldsymbol{T}$ and the third corresponding to eigenvalue 0 points in the normal direction $\boldsymbol{n} : \boldsymbol{e}_1 = \boldsymbol{\kappa_{max}}, \boldsymbol{e}_2 = \boldsymbol{\kappa_{min}}, \boldsymbol{e}_3 = \boldsymbol{n}$.

Estimating this curvature tensor field on a 3D mesh is performed by using normal cycles theory. In a few words, the curvature tensor can naturally be defined for each point along an edge of the input mesh because there is an obvious maximum (resp. minimum) curvature across (resp. along) an edge. The averaging of the contributions of each density line of tensors along the edges of an arbitrary region B leads to a simple expression to estimate the 3D curvature tensor for a vertex v:

$$\boldsymbol{T'}(v) = \frac{1}{|B|} \sum_{edges\, e} \beta(e) |e \cap B| \bar{e} \bar{e}^T \tag{8.4}$$

where $|B|$ is the neighborhood area of v, $\beta(e)$ is the angle between normals of the triangles adjacent to edge e. $|B \cap e|$ is the length of e in B, and $\bar{e}$ is the unit vector aligned with e. Since the normal cycle theory states that the neighborhood B may be chosen arbitrarily, we define it as the intersection between the mesh and a sphere of radius r. This radius is the parameter that specifies the scale of the curvature estimation (see figure 8.1) and is expressed in percentage of the bounding box diagonal. Once we have

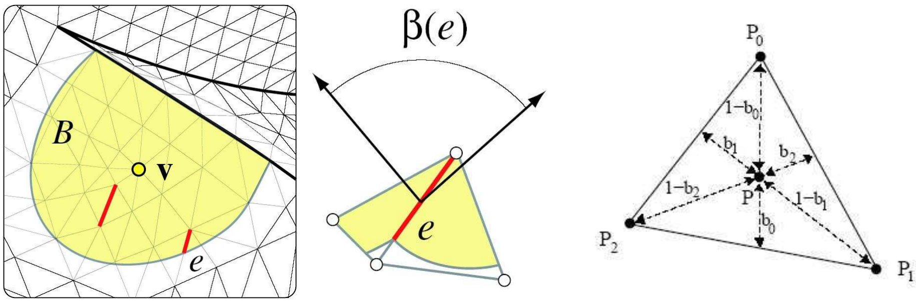

Figure 8.1: *On the left, neighborhood B used to compute the curvature tensor and angle β between the normals of the faces adjacent to the edge e. On the right, barycentric coordinates enable the interpolation of the tensor estimation on the faces of the mesh.*

computed the curvature tensor for each vertex, a continuous curvature tensor field is obtained by linearly interpolating the curvature tensors on each triangle of the mesh. This can easily be done using barycentric coordinates (figure 8.1). The normal directions, associated with the minimal amplitude eigenvalue and the principal directions of curvature are also interpolated on the triangles of the mesh. Points for which $\kappa_{min} = \kappa_{max}$ are called umbilical points. They can be seen as degeneracies of the curvature tensor field. Their detection and the necessary processing of the continuous tensor field are presented in the next subsection.

8.2.3 Curvature Tensor Field Filtering and Topological Analysis

Tensor Topology

The topology of the curvature tensor field estimated on a 3D mesh is a piecewise-linear symmetric tensor field that can be described by its degenerate points: the umbilical points. We show in this Section, following the work of Tricoche in [123] and Delmarcelle et al. in [41], that there are only two types of umbilical points: *wedges* and *trisectors*. Fig. 8.4 illustrates this

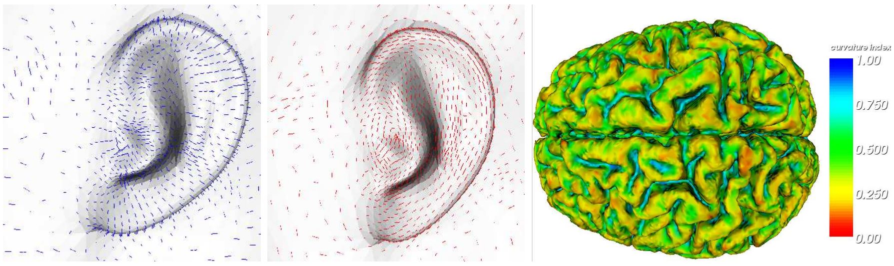

Figure 8.2: *Curvature estimation using multi-scale analysis and normal cycle theory. From the left to the right, maximum curvature directions, minimum curvature directions and curvature index which is given by* $\frac{1}{2} - \frac{1}{\pi}\arctan(\frac{\kappa_{min}+\kappa_{max}}{\kappa_{max}-\kappa_{min}})$.

classification.

The topology of the curvature tensor field is the topology of its eigenvector fields. The topology of vector fields is often described by their topological skeletons which are determined by locating degenerate points and their connecting hyperstream trajectories. We show later that since eigenvectors have sign indeterminacy, the structure of eigenvector fields are different from regular, signed vector fields.

In the sequel, we analyze the curvature tensor field as a 2D tensor $\boldsymbol{T}$ which is obtained by projecting the tensor $\boldsymbol{T'}$ into the tangential plane (i.e. $\boldsymbol{T}$ is described in parameter space with 2D coordinates u, v), given by the eigenvectors corresponding to the principal directions.

An umbilical point is defined as follows: p_d is an umbilical point of surface S if the minimal and maximal curvatures are equal: $\kappa_{min} = \kappa_{max}$. Here, we point out the link between umbilical point and tensor field degenerate points. Indeed, a point p_d is a degenerate point of a tensor field $\boldsymbol{T}$ iff the two eigenvalues of $\boldsymbol{T}$ are equal to each other at p, i.e. iff $\lambda_1 = \lambda_2$. It is obvious that umbilical points are the degenerate points of the curvature tensor field.

Let us denote by κ the common curvature or eigenvalue of an umbilical point p_d. At p_d, the tensor field is proportional to the identity matrix:

$$\boldsymbol{T}(p_d) = \begin{pmatrix} \kappa & 0 \\ 0 & \kappa \end{pmatrix}, \tag{8.5}$$

and there is an infinity of eigenvectors that verify $\boldsymbol{T}(p_d)\boldsymbol{e} = \kappa\boldsymbol{e}$. It follows that hyperstreamlines of principal curvature cross each other in degenerate points. The umbilical points verify the conditions:

$$\begin{cases} T_{11}(p_d) - T_{22}(p_d) = 0 \\ T_{12}(p_d) = 0 \end{cases} \tag{8.6}$$

which we can use to locate them. As explained before, we use bilinear interpolation of the tensor components between vertices. In order to compute the tensor at any point D in a triangle A, B, C, we express the 3D tensors $\boldsymbol{T'}(A)$, $\boldsymbol{T'}(B)$ and $\boldsymbol{T'}(C)$ on the frame built with the normal of the triangle and the edge A, B. We then use barycentric coordinates (see Fig. 8.1) to interpolate the tensor eigenvectors and reconstruct the tensor at D.

Index, sectors and separatrices

In vector fields, there are various types of critical points, such as nodes, foci, centers and saddle points, that correspond to different local patterns of the neighboring streamlines. These patterns are characterized by the vector gradients at the position of the critical points.

Likewise in tensor fields, different types of degenerate points occur that correspond to different local patterns of the neighboring hyperstreamlines. These patterns are determined by the tensor gradients at the position of the degenerate points. Consider the following partial derivatives:

$$[ll]a = \frac{1}{2}\frac{\partial(T_{11} - T_{22})}{\partial u} \qquad b = \tfrac{1}{2}\tfrac{\partial(T_{11} - T_{22})}{\partial v} \tag{8.7}$$

$$c = \frac{\partial(T_{12})}{\partial u} \qquad d = \tfrac{\partial(T_{12})}{\partial v} \tag{8.8}$$

evaluated at the degenerate point p_d, with (u, v) parameter coordinates. In the vicinity of p_d we can expand tensor components to first-order as

$$\begin{cases} \frac{T_{11} - T_{22}}{2} \approx a\Delta u + b\Delta v \\ T_{12} \approx c\Delta u + d\Delta v \end{cases} \tag{8.9}$$

where (Δu,Δv) are small displacements from p_d in parameter space (i.e. in the tangent plane of S at p_d). An important quantity for characterization of degenerate points is

$$\delta = ad - bc \tag{8.10}$$

which is invariant under rotation. We now introduce the *tensor index* at a degenerate point.

The *index* at the degenerate point p_d of a tensor field is the number of counter-clockwise revolutions made by the eigenvectors when travelling once in a counter-clockwise direction along a closed path encompassing p_d. The path is chosen close enough to p_d so that it does not encompass any other degenerate points.

Indices of tensor fields are half-integers while indices of vector fields are integer (1 for a node, -1 for a saddle etc.). This arises from the sign ambiguity of the eigenvectors. It can be shown [123] that if $\delta \neq 0$, the index I at p_d is given by

$$I = \frac{1}{2} sign(\delta) = \pm\frac{1}{2} \tag{8.11}$$

The index at the degenerate point p_d as well as the δ quantity characterize the pattern of neighboring hyperstreamlines and enable to identify the type of this degenerate point. When traveling on a closed path encompassing p_d, we encounter two types of angular sectors (see Fig. 8.3):

1. hyperbolic sectors α_i, where trajectories sweep past the degenerate point and
2. parabolic sectiors β_i, where trajectories lead away or towards the degenerate point.

For example, the singularity in Fig. 8.3 has three hyperbolic and three parabolic sectors. By analogy with vector fields, Tricoche et al. introduce the term of *separatrices* for refering to the dividing hyperstreamlines that separate one sector from the next [41]. Let be θ_k the angle between separatrix s_k and the (parameter) u-axis. It can be shown that $u_k = tan\theta_k$ must be a root of the cubic equation

$$du^3 + (c + 2b)u^2 + (2a - d)u - c = 0 \tag{8.12}$$

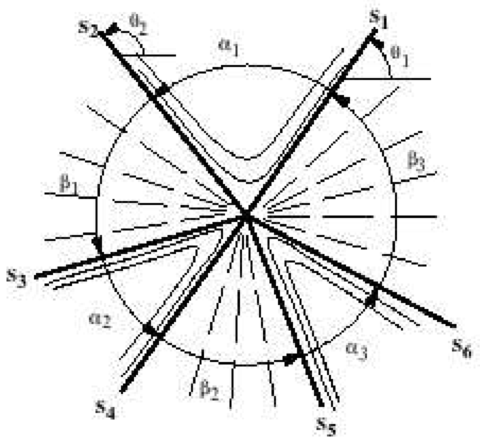

Figure 8.3: *Description of local hyperstreamlines patterns around an umbilical point. Hyperbolic sectors span β_j angles and parabolic sectors span α_i angles. The hyperstreamlines separating different sectors are called* separatrices *(image courtesy of Delmarcelle et al. [41]).*

Therefore, there are at maximum three separatrices (real roots u_k) and degenerate points (i.e. umbilical points) have no more than three sectors.

Let us imagine an hypothetical singularity with n_p parabolic and n_h hyperbolic sectors ($n_p + n_h \leq 3$). The parabolic sectors span angles β_j ($j = 1, ..., n_p$) and the hypernolic sectors span angles α_i ($i = 1, ..., n_h$). The eigenvectors rotate an angle β_j within a parabolic sector and $\alpha_i - \pi$ within an hyperbolic sector. During a counter-clockwise revolution around the singularity, the eigenvectors rotate an angle $2\pi I = \sum_{j=1}^{n_p} \beta_j + \sum_{i=1}^{n_h} (\alpha_i - \pi)$. Since[1] $\sum_{j=1}^{n_p} \beta_j + \sum_{i=1}^{n_h} \alpha_i = 2\pi$, the index of an umbilical point is given by

$$I = 1 - \frac{n_h}{2} \tag{8.13}$$

It follows that the number of hyperbolic sectors at an umbilical point is

$$n_h = 2 - sign(\delta) \tag{8.14}$$

[1]The sum of angles spanned by the different sectors is equal to 2π in parameter space. In 3D, without considering the tangent plane, this equation would be verified only for planar points. Indeed, saddle points have a sum of angles superior to 2π and parabolic points have a sum of angles inferior to 2π.

Trisector and wedge points

When $\delta > 0$, $n_h = 3$; the degenerate point has three hyperbolic sectors and no parabolic sector. It is a trisector point. When $\delta > 0$, $n_h = 1$, the umbilical point has one hyperbolic sector. It is then a wedge point (see Fig. 8.4).

Wedges and trisectors are stable structures in tensor fields: they cannot be broken into more elementary singularies with smaller index. They can however merge (when observed at different scales) with each other, creating unstable combined singularities of higher index. The merging of umbilical points corresponds to $\delta = 0$ (see Fig. 8.5).

Finally, we also introduce the *tensor topological rule* which gives a condition on the number of singularities for a tensor field defined on a surface S of Euler characteristic $\chi(S)$ (see Chapter 2). The Poincaré-Hopf theorem [123] stipulates that the sum of indices at critical points of a vector field defined across a surface S is equal to $\chi(S)$. This also holds for tensor fields. If the tensor field has only isolated degenerate points consisting exclusively of w wedges, t trisectors, n nodes, c centers, f foci and s saddles:

$$\frac{1}{2}(w - t) + n + c + f - s = \chi(S). \tag{8.15}$$

This rule, however, does not give an estimation of how many critical points can be found on a surface S but gives a condition of the sum of each type of umbilical point.

For tensor fields defined on a triangular mesh, Tricoche has shown that a triangle can have at most an umbilical point. To detect an umbilical point, we simply compute the curvature tensor at each point of the mesh and solve the equation 8.12 in parameter space for each triangle. We then can classify the type of umbilical point by computing δ and the index I.

Several observations must be made at this stage. First, the number of umbilical points is constrained by the topological rule as well as by the number of triangles. Second, the presence of noise in the mesh causes instabilities when computing the eigenvectors of the curvature tensor [121]. The consecutive instabilities of the directions of the tensor eigenvectors cause the detection of unstable pairs of umbilical points. To deal with this

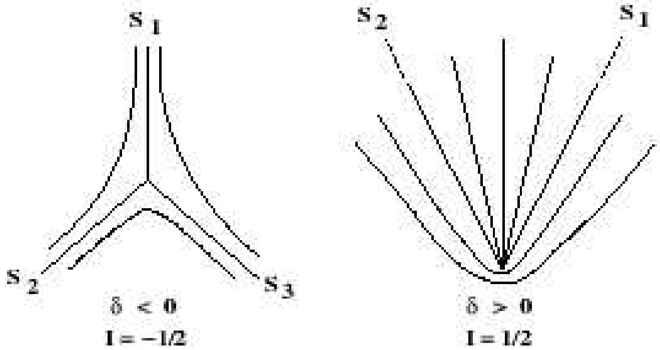

Figure 8.4: *There are only two types of umbilical points :* wedge *and* trisector *umbilical points.*

issue, we propose a multi-scale evaluation of the tensor as well as a filtering of the curvature tensor.

Tensor Filtering

We have already pointed out that evaluating the tensor at higher scales smoothes the curvature hyperstreamlines directions. Indeed, it is equivalent to estimate the tensor at larger neighborhoods and as these neighborhoods become larger, the tensors evaluated in two neighboring points are very similar, yielding a locally smooth tensor field. However, this strategy can lose some important small details of the surface. That is why we propose to filter the curvature tensor field.

Tensor field smoothing can be achieved in many ways. The most straightforward manner is to simply smooth the tensor components by taking a mean of curvature tensor components over the 1-ring neighborhood. As we deal with a surface, this should be done in parameter space which is much more difficult to estimate over a 1-ring neighborhood than on a triangle (this requires a local parameterization and as seen in Chapter 2, there is no natural parameterization for a general surface.). Indeed, a tensor defined on a plane (or other Euclidian space) can be filtered by simply averaging each of its components with the corresponding component of the neighboring tensors because each tensor is described in a unique frame. For a surface with non-zero curvature, there is no unique frame

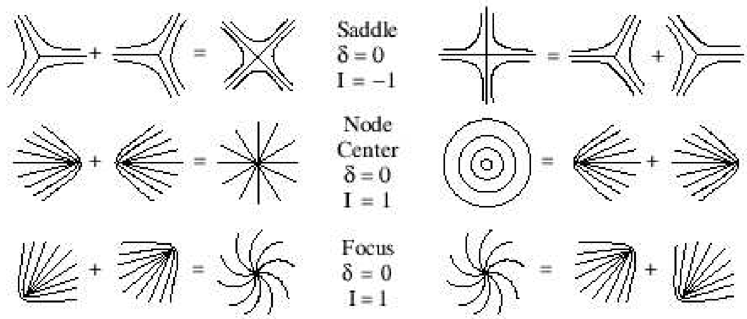

Figure 8.5: *Unstable merging of umbilical points in combined degenerate points (image courtesy of Delmarcelle et al. [41]).*

that enables to describe each curvature tensor. The components of these tensors are expressed in a local frame defined by the local tangent plane and surface normal. A parameterization enables to define (u, v) coordinates that are locally orthogonal and lay on the tangent bundle. It follows that comparing the corresponding components of two neighboring tensors must be done by re-projecting one tensor on the local frame of the other one. It is no more certain that the reprojected tensor components have the same meaning in another frame. This is why we prefer to deal with eigenvectors and eigenvalues which are intrinsic to the curvature tensor.

So, we propose to filter the eigenvectors directions of the curvature tensor field over each 1-ring. In order to filter the curvature tensor at p, we first smooth normal directions of each point q_i of the 1-ring of p, noted $N(p)$. The filtering of normal directions $\boldsymbol{n}(q_i)$ of each point q_i is simply made by averaging its direction by the directions of the normal directions of its 1-ring $N(q_i)$:

$$\boldsymbol{n}(q_i) := \boldsymbol{n}(q_i) + \nu \sum_{q_j \in N(q_i)} w_{ij} \boldsymbol{n}(q_j) \tag{8.16}$$

where ν is a smoothing parameter, w_{ij} is given by the ratio of the distance between the point j and i and of the sum of these lenghts over $N(q_i)$. When the normal directions of each point have been smoothed, we compute the principal directions at all points q_i of $N(p)$ and apply a similar filter as equation 8.16. We then reconstruct the tensor at p with the smoothed normal directions and principal directions. This operation leads to fairer tensor fields as illustrated on Fig. 8.6. This filtering step enables to get rid of non-valid umbilical points due to computational instabilities caused by noise.

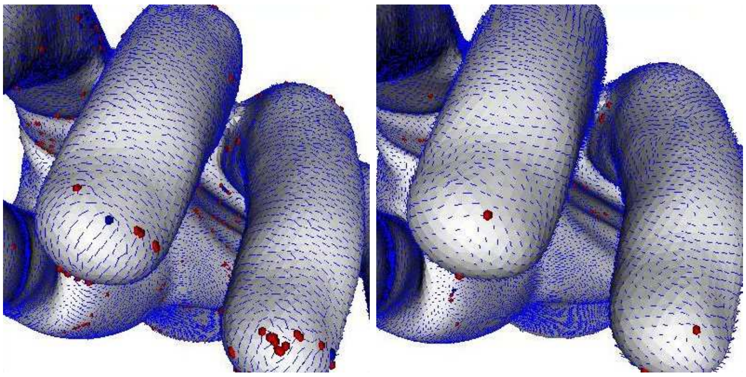

Figure 8.6: *Principal direction fields and umbilical points estimated on the* Hand *model. On the left, estimation of the curvature tensor at* $.01$ *percent of the bounding box diagonal. On the right, same estimation after two steps of tensor filtering.*

Tensor Topology Simplification

We have seen that the combination of a pair of umbilical points of the same type at a same location p_d results in more complex degenerate point (focus, node, center or saddle) with integer index. If a pair of close umbilical points is composed of a wedge and a trisector, at some distance, it can be said that their combination vanishes. Indeed, there is no more degeneracy when a wedge is combined with a trisector [123] and the topological rule still stands. Inspired by works for representing tensor fields for Computational Fluid Dynamics, we simply merge close pairs of different umbilical points. We define a pair of umbilical points as close if their respective triangles share a point or an edge. This simplification enables to get rid of unsignificant umbilical points due to too small details or computational instabilities.

8.2.4 Multi-Scale Analysis and Robust Umbilical Points

Since we want to use umbilical points as feature points for our watermarking scheme, we have to select the most robust ones. Indeed, even after filtering, some umbilical points are due to noise in the geometry or to the coarseness of the sampling of the input mesh. Multi-scale analysis enables to discriminate intrinsic umbilical points from those due to noise. Using larger neighborhoods to estimate the curvature tensor smoothes the principal directions field and cancels umbilical points by pairs as illustrated on figure 8.7. It is quite obvious that the scale is of crucial impor-

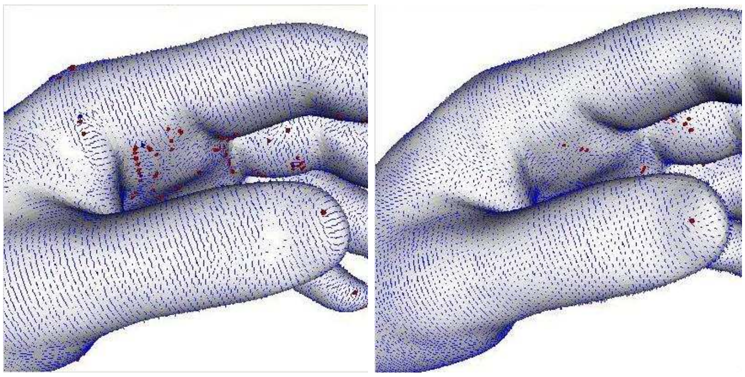

Figure 8.7: *Principal direction fields and umbilical points estimated on the* Hand *model. The radii of the neighborhoods used for the estimation are respectively* .01 *and* 1 *percent of the bounding box diagonal length. Larger scales smooth principal direction fields and cancel umbilical points due to noise. However, if the radius is taken too large, we may lose local properties of the surface.*

tance for curvature estimation. A surface can be locally noisy and though seem to be smooth at a larger scale. If the radius of the neighborhoods increases, the neighborhoods of connected points tend to be nearly the same, resulting in nearly identical curvature estimates. In consequence, varying the neighborhood size also filters the curvature tensor. Some points can be umbilical points at some scale interval and not at other scales.

In our scheme, we estimate the curvature tensors at several scales in order to smooth the tensor field and estimate the robustness of the umbilical points at these different scales. Scales may not be too large for two obvious reasons. The first is that we may lose local properties by excess averaging.

The second is the computational cost that tends to $O(n^2)$ if the neighborhood extends to the whole mesh. We start from a scale $s(t=1)_0 = .01$ (percent of the bounding box diagonal), and we detect umbilical points at scales $s(t)_i = t2^i s_0$ with $i = 1, ..., 5$ and $t = 1, 10, 100$.

To assess their robustness, we assign a score to the umbilical points depending on their existence, type and position for all the selected scales. This score is simply incremented if the same type of umbilical point at the same position is detected at several scales. Among the most robust umbilical points, we then select those presenting the larger *mean curvature* (i.e. the *trace* of the curvature tensor) within a neighborhood N. The size of N is defined as a function of the bounding box diagonal and is a parameter that directly influences the geodesic partition of the mesh and the watermark embedding capacity. The choice of scales and neighborhoods based on the bounding box diagonal implies that the scheme cannot resist against cropping. However, the good robustness properties of so selected umbilical points in the difficult cases of noise addition, affine transforms and sampling changes such as remeshing or decimation is presented in Section 8.5. This is the second step of our watermarking scheme and the subject of the next section.

8.2.5 Analogy with the Scale-Invariant Feature Transform

In the case of 2D images, David Lowe has developed the Scale-Invariant Feature Transform (SIFT [92]) which also uses a multi-scale analysis of feature points. These image feature points are detected by using a variable-scale Gaussian as scale-space kernel. The feature points are actually scale-space extrema in the difference-of-Gaussian function convolved with the image, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor. The difference of Gaussian can also be seen as a good approximation of the scale-normalized Laplacian of Gaussian which produces the most stable minima and maxima points when compared with other image feature point detectors [92]. Our surface curvature multi-scale analysis is very close to SIFT even if we cannot extend the convolution of Gaussians to arbitrary surfaces. The sampling frequency in scale-space for surfaces has been empirically proposed and may be improved by following some features of Lowe's work.

A particularity of our multi-scale analysis on 3D triangulated surfaces is the third-order equation 8.12 which, in the parameter space of each triangle, rules the existence and type of a feature point. This implies that the maximum number of features is bounded by the number of triangles of the surface and could be exploited by a multiresolution analysis (see Chapter 2).

8.3 Partition And Remeshing

We extend to 3D meshes an idea of Bas et al. [16] that is dedicated to 2D images. The key idea is to exploit the selected feature points as centroids for a Voronoi diagram (and the dual Delaunay triangulation) to build a unique partition of the image in several segments. Then the repetition of a watermark in each segment provides good robustness properties: if some feature points are lost after an attack, the remaining segments will still allow watermark detection.

In the case of 3D surfaces, Voronoi diagrams and Delaunay triangulations do not use the Euclidian distance but the *geodesic distance* introduced in Chapter 2. Some proposals have been published to provide robust and fast estimates of this distance [81, 107] that improve the classical Dijkstra estimation also performing in $O(n \log n)$. Since we are interested in a whole partition of the surface using the robust umbilical points that must resist against sampling changes such as decimation or remeshing, we can afford a slight loss in mesh local distance precision to speed up this estimation.

We propose a $O(n)$ approximated estimate of the whole Voronoi diagram and its dual Delaunay triangulation using a sampling independent wavefront strategy that enables a fast and unique partition of the surface.

8.3.1 Sampling Independent Wavefronts

Wavefronts are computed as the intersection of the surface and spheres of increasing radius and centered on the seed. This type of wavefront is independent of the sampling of the surface and leads to far better results than simple connectivity based wavefronts. The latter often does not provide circular patches after some iterations on the general case of surfaces that present regions of different point density. Actually, in order to ensure that the intersection between the spheres and the surface is topologically homeomorphic to a disk, a combinatorial wavefront is also used to discard the non-connected parts of the intersection. Our method is of course not as accurate as those of Peyre et al. [107] and Kirsanov et al. [81] but provides a good trade-off between speed and accuracy.

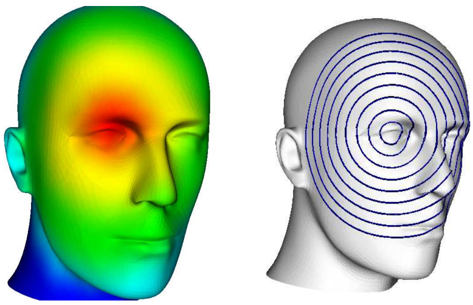

Figure 8.8: *On the left, a Dijkstra-based wavefront propagation. On the right, a sampling independent wavefront propagation.*

8.3.2 Building The Geodesic Delaunay Triangulation

We denote the set of the k detected umbilical points U and its elements by an index u_i with $i = 1, ..., k$. The set of the m centroids of the Voronoi diagram is denoted by C and its elements by c_{i_1,i_2,i_3} where i_1,i_2 and i_3 are the indices of the three umbilical points corresponding to this centroid.

The strategy to build the geodesic Delaunay triangulation of the surface mesh is the following. We first use the wavefront propagation to find the centroids of the Voronoi diagram which are the circumcircle centers of the Delaunay triangulation. Each umbilical point $u_i \in U$ is taken as a seed and k simultaneous wavefronts are propagated from them. When three different wavefronts starting from $u_{i_1}, u_{i_2}\, and\, u_{i_3}$ meet in a triangle, a new centroid noted c_{i_1,i_2,i_3} is added in C with the 3D coordinates of the barycenter of this triangle. This process ends when each point of the surface has been visited by at least one wavefront. Fronts are then propagated from each of the m centroids in C to create m geodesic circular patches [2].

[2]The number m of centroids is linked to the number k of umbilical points by the Euler equation which takes the genus of the surface into account. In general, $m \approx 2k$.

The expansion of patch c_{i_1,i_2,i_3} ends when the three seeds $u_{i_1}, u_{i_2}\, and\, u_{i_3}$ that originated this patch have been reached by the wavefront.

It is possible that, if the number of umbilical point is too small, that the geodesic circles do not entirely recover the surface. This issue is usually encountered when a large part of the surface is made of free-form regions (spherical, quadric regions) which are discarded by our strategy to detect umbilical points. It is possible to detect such problem and we solve this issue by adding virtual geodesic circles centered at the farthest point from the set of umbilical points. We iteratively perform this step until the shape is totally recovered by geodesic circles.

A mapping of each of the m geodesic circles to a 2D plane is performed using an harmonic parameterization which minimizes distortions between angles and lengths of the initial mesh and of the projected one [11, 56] (see Chapter 2). We triangulate the three feature points belonging to this patch in the harmonic space and project this triangle back to the surface as illustrated on Fig. 8.9. Fig. 8.10 shows small distortions when using harmonic parameterization when point density is not regular in the geodesic circle. Since the mean distortion is minimized, the region of the circle encompassing more points is less distorted than the region with larger triangles. Another parameterization based on minimizing the worst case could maybe solve this interesting problem. We tackle this problem by simply merging the common edges of neighborhing curved triangles obtained in adjacent parameterizations, giving the priority to the less distorted parameterization. If the distortion is minimal (which is ensured by the harmonic mapping) the projection of this triangle on the 3D surface is a good approximation of the geodesic triangle linking the three umbilical points. When each patch has been processed, we have built the partition of the mesh as illustrated in Fig. 8.11.

8.3.3 Remeshing

Furthermore, a regular remeshing can accurately be achieved in the harmonic space of each patch by subdividing the triangle defined above with the desired regular connectivity (figure 8.12). This remeshing step is important for our watermarking scheme. The secret information is first embedded in the remeshed curved triangles. The variations introduced by

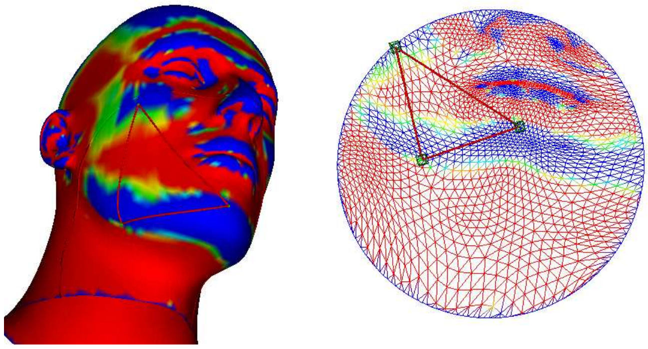

Figure 8.9: *Illustration of the geodesic triangular patch construction in the harmonic plane. On the left, the 3D region of the mesh that has been parameterized on the circle on the right. A triangle has been drawn in the harmonic space and back projected in the 3D space. Gaussian curvature has been represented in color to show the correspondence between the mesh and its harmonic embedding.*

this embedding are then applied to the original points of the mesh by projecting them on the watermarked curved triangle.

8.4 Watermarking Encoding And Decoding

We exploit the mesh spectral domain to embed the watermark in each remeshed curved triangular patch. The spectrum of a 3D mesh is given by the projection of its geometry on basis functions which are the eigenvectors of the Laplacian operator applied to this 3D mesh. As the spectral coefficients depend on a reference axis for x, y, z coordinates, a Principal Component Analysis is performed to retrieve three principal reference axes. This step allows to resist against rotation transforms. The embedding is close to the one presented in our former work (see Chapter 7). In a few words, the watermark is inserted by scrambling the triplets of spectral components in the following way. We sort the three components and permute the median value of this triplet with respect to the mean value of the two other components. The embedding force is given by the eigenvalue of the first triplet that is watermarked.

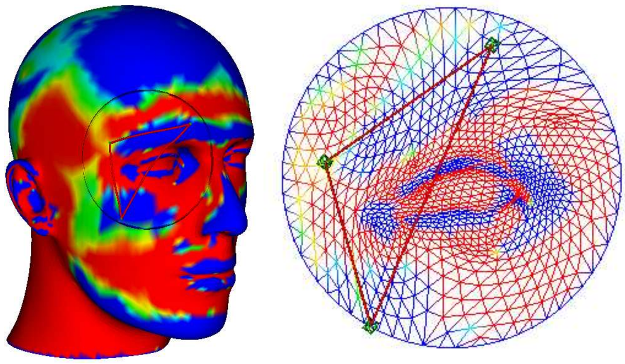

Figure 8.10: *Illustration of the geodesic triangular patch construction in the harmonic plane. On the left, the 3D region of the mesh that has been parameterized on the circle on the right. A triangle has been drawn in the harmonic space and back projected in the 3D space. Gaussian curvature has been represented in color to show the correspondence between the mesh and its harmonic embedding. It can be seen that the higher density of points in the eye of the model induces a small distortion in the curved triangle.*

When a triangle patch is watermarked, the perturbations caused by the watermark are embedded on the mesh by projecting the original points on the remeshed triangle (see Fig. 8.13). This step enables to retrieve the watermark if the number of points of the original mesh is at least equal to the number of points of the set of remeshed triangles points.

The detection is quite forward. The feature points are estimated and tessellated in exactly the same way as explained above. The detection is performed by reading whether the median of the triplet is inferior or superior to the mean of the two other spectral components.

8.5 Results

We show the robustness of the feature points detection against watermarking attacks such as remeshing, affine transforms, noise addition, smoothing in Fig. 8.14. To prove the efficiency of our method, typical attacks are simulated on the watermarked object and corresponding detection results are illustrated in Fig. 8.15. The watermark is considered as

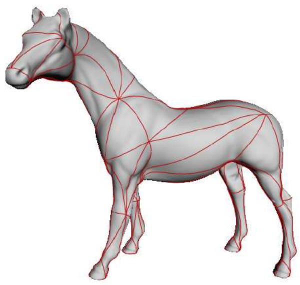

Figure 8.11: *Partition results using a Delaunay geodesic triangulation of the umbilical points.*

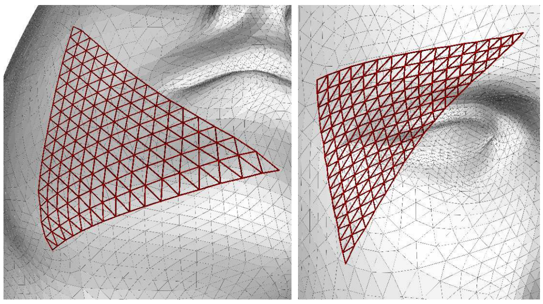

Figure 8.12: *Regular connectivity remeshing for the* head *model.*

detected by majority voting of each triangle patch. Umbilical points resist to affine transforms and to some extent to noise addition and smoothing iterations. Sampling changes do not perturb robust umbilical points as they resist against decimation and subdivision attacks that are the basic operations to remesh meshes.

The watermarking scheme is more sensitive than umbilical points to noise and smoothing attacks. However our blind watermarking scheme resists against sampling changes (subdivision and decimation about 15 percents). The most difficult attack is the decimation, because the curvature of some regions is distorted by this operation, leading to different curvature tensor fields which tend to harmonize themselves on the whole surface. This means that perturbing the hyperstreamlines in some location also affects other locations and the multi-scale analysis can fail to retrieve

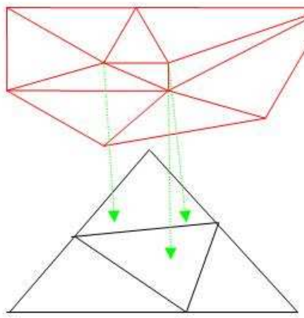

Figure 8.13: *The points of the original mesh are projected on the remeshed surface patch by computing intersections between their normal directions and the surface. The distance between both surfaces has been amplified for clarity purposes.*

some points. Of course, the watermark is not robust against cropping because of the parameters of our scheme depending on the bounding box diagonal.

Because of the higher sensitiveness of remeshing, spectral embedding and re-projection to watermarking attacks, analyzing the dependance of the whole sheme to the loss or displacement of an umbilical point is not easy. In our experiments, for increasing amplitudes of attack, an umbilical point is lost after the whole scheme starts to fail. The improvement of the remeshing (by using a more complex geodesic remeshing [108]) could lead to better robustness results which would then be more dependent to the chosen feature points detection. Umbilical points resist to affine trans-

Models vs. Attacks	noise	smooth.	decim.	RST	subdiv.
Bunny	0.25%	130	15%	OK	1
Horse	0.1%	100	13%	OK	1
Head	0.15%	20	20%	OK	1

Figure 8.14: *Robustness of umbilical points against usual watermarking attacks: noise, smoothing, decimation, RST and subdivision.*

forms and to some extent to noise addition and smoothing iterations. Umbilical points resist particularly well to decimation and subdivision attacks that are the basic operations to remesh meshes. In conclusion, sampling changes also do not perturb robust umbilical points. Fig. 8.16 illustrates the local distortion introduced by our watermarking scheme for several insertion forces. Finally Fig. 8.17 shows the watermarked version of the *Head* model with just noticeable distortions.

Models vs. Attacks	noise	smooth.	decim.	subdiv.
Bunny	0.15%	20	14%	1
Horse	0.05%	20	10%	1
Head	0.05%	10	15%	1

Figure 8.15: *Watermarking results for various models and noise, smoothing, decimation as well as subdivision attacks. Naturally the watermarking scheme is more sensitive than the umbilical points to noise and smoothing attacks. Our blind watermarking scheme resists to sampling changes (subdivision and decimation up to 15%).*

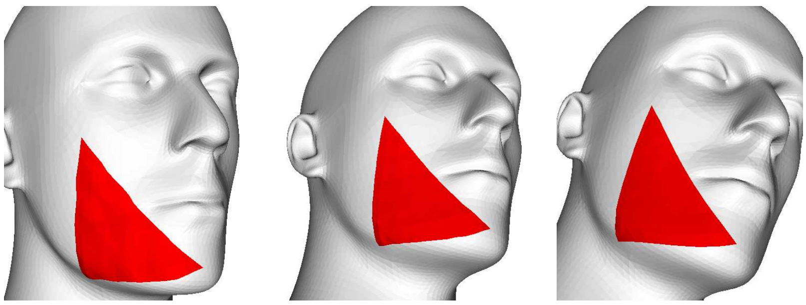

Figure 8.16: *From left to right the submesh in red has been watermarked with watermarking forces of 20, 50 and 90 on 200 spectral components. The latter respects the invisibility criterion.*

8.6 Conclusion

This Chapter has presented a new approach for 3D models watermarking. The good properties of the automatically selected feature points have been shown for self-resynchronization of a 3D object in particular when the sampling of the object changes. This technique is not limited to

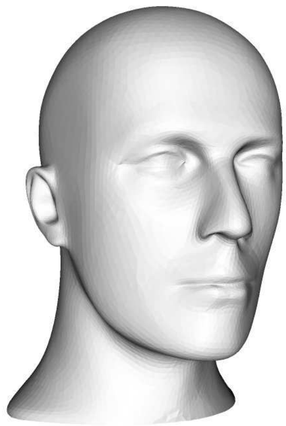

Figure 8.17: *Watermarked* Head *model with just noticeable distortions. The watermark strength has been selected so that the distortions due to the watermark are visible.*

the watermarking field but could be of interest in several applications, for example in the reverse engineering and medical imaging fields, requiring a pre-processing self-resynchronization step. However, we have shown that these feature points have limitations. One of them is related to the problem of the cropping attack. Indeed, an umbilical point is intrinsically linked to the scale at which it is detected. Cropping does not enable to robustly chose scales. We have also pointed out some problems due to the (small) distortions caused by parameterization.

This work has led to the following publications [5, 6].

In the next Chapter, we present another scheme based on other feature points that are better candidates to resist to cropping.