Chapter 2

Digital Shape Processing

2.1 Introduction

This Chapter first introduces basic definitions related to 3D digital shapes represented by 3D meshes. We then present an overview of digital surface processing tools that have been recently developed as extensions of signal processing and based on Differential Geometry. We focus on 3D mesh parameterization, curvature properties, geodesics, multiresolution analysis as well as spectral decomposition which are the main tools we use in the other Chapters of this thesis.

2.2 Shapes and their digital representations

3D virtual objects approximate a continuous surface (often referred to as a *shape*) in the three-dimensional space. The varied digital representations of a shape can be classified in two categories:

- acquired digitalizations of existing solid shapes (3D mesh, point sample geometry, voxel-based representations).
- creations making use of the assembly of parametric surfaces, generally by Computer Aided Design (CAD) softwares (parametric surfaces, implicit surfaces, Constructive Solid Geometry (CSG)).

The first category relates to raw data acquired by range laser technology, stereovision, medical imaging technologies (CT scan, MRI, etc), which are then modeled in a digitized surface after image processing. The second category is composed of surfacic data obtained by assembly of logical or

geometric elementary volumes or by connecting implicit or parametric patches (i.e. pieces of surface which are homeomorphic to a disk).

Both representation categories must face the issue of being fluently displayed on 2D screens (or stereo devices such as glasses). Current technology relies on standard graphics rendering hardware which use triangulated data. This fact partially explains why much of the work in the area of Computer Graphics and, particularly, of 3D watermarking deals with 3D triangle meshes. We give here the details of the most used representations of each category of digital surfaces: 3D meshes and NURBS.

2.2.1 Non-Uniform Rational B-Splines (NURBS)

Mechanics industry produces two main kinds of surfaces: functional surfaces, which enable the manufactured product to operate its function; and free surfaces, for aesthetical purposes. Free surfaces are created with CAD softwares by making use of parametric surfaces. The fidelity of these surfaces to the constraints imposed by functional surfaces is impossible to guarantee. However, they are characterized by many degrees of freedom which enable to model very smooth (fair) and complicated surfaces. These surfaces are parameterized by a set of *control points* and *weights*.

The construction of surfaces by NURBS uses monodimensional B-Splines which can be recursively computed. Let $U = t_0, t_1, ..., t_m$ be a set of increasing scalar values (i.e : $t_i \leq t_{i+1}$) and $N_{i,k}$ a set of normalized B-Splines parameterized by u. The recurrence is initialized by:

$$N_{i,0}(u) = \begin{cases} 1 & if\, t_i \leq u < t_{i+1} \\ 0 & otherwise \end{cases} \tag{2.1}$$

Then the following recurrence enables to compute B-Splines of any degree:

$$N_{i,k}(u) = \frac{u - t_i}{t_{i+k} - t_i} N_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(u) \tag{2.2}$$

Finally, a vector w of weights and a set P of control points enable to parameterize the combination of mono-dimensional B-Splines of *degree* k and l to obtain a surface. The NURBS surface is then given by:

$$S_{NURBS} = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{i,j} R_{i,j,k,l}(u, v) \tag{2.3}$$

where

$$R_{i,j,k,l}(u,v) = \frac{w_{i,j}N_{i,k}(u)N_{j,l}(v)}{\sum_{r=0}^{n}\sum_{s=0}^{m} w_{r,s}N_{r,k}(u)N_{s,l}(v)} \tag{2.4}$$

Creating a NURBS surface respecting the constraints of a functional surface is solved by selecting a correct set of points and corresponding weights. The NURBS surfaces have good differential properties directly controled by the degree of the B-spline. Furthermore, this parametric representation of the surface enables a fast sampling for rendering.

Usually, the control points are not located on the NURBS surface. Modifying a control point and its weights results in a morphing of the whole surface. This also means that there is no direct access to the points of the surface. Editing such surfaces is therefore much more complex than editing 3D meshes whose points lay on the surface and can be directly modified by the user.

2.2.2 3D triangle meshes

A 3D triangle mesh M can be seen as the embedding in the 3D space of geometric primitives: *vertices*, *edges* and *faces* (respectively sets V, E and F). The set of these primitives is often referred to as the *connectivity* of the mesh. The embedding of vertices (a.k.a. *points* or *nodes*) in the 3D space is named *geometry* and is represented by the set of the point coordinates (x, y, z). Their number is noted n and ranges from 10^3 to more than 10^9 (see Fig. 2.1).

It is useful to define some properties of points, edges or faces of a mesh. Two points p_i and p_j of a mesh M are *adjacent* (a.k.a. *neighbors*, *1-connected*) if there exists an edge $e(i,j) \in M$ connecting these points. The set of points connected to a point p_i, is noted $N(p_i)$ and is often called $1-ring$ neighborhood (or simply «neighborhood») of p_i. The *valence* or *degree* of a point p_i is the cardinality of its neighborhood, i.e. the number of points connected to p_i. The *adjacency matrix* A represents the connectivity of the mesh M:

$$A_{ij} = \begin{cases} 1 & if\ \exists\, e(i,j) \in M \\ 0 & otherwise \end{cases} \tag{2.5}$$

Normal direction vectors can be defined on points and faces and are respectively called *point normals* and *face normals*. For faces, the normal vector is given by the normal direction of the plane of the triangle. For a

point, the normal vector is given by the average of normal vectors of the incident faces to this point.

In a *manifold* mesh, each edge is incident to at most two faces. It follows that the neighborhood of each point is locally homeomorphic to a disk which is the definition of a *manifold surface*. If an edge is incident to only one face, it is called a *boundary edge*. A *feature edge* is a non-boundary edge such that the angle between the normal vectors of its adjacent faces is superior to a given threshold. These edges are important since human eyes are sensitive to lines in the mesh composed by aligned feature edges. Usually the surface is not differentiable across these edges.

A 3D *polygonal mesh* is composed of planar polygons which carry the information of transparency, color or texture of the surface. The faces of a 3D *triangular mesh* are all triangles.

Among triangular meshes, there is an important particular case which is related to 2D triangulations: *Delaunay triangulations*. Let P be a set of points (p_0,...,p_n) in the plane. The Delaunay triangulations of P are such that for each triangle (p_i, p_k, p_l), its circumcenter circle does not contain any other point of P than those of this triangle. The dual of a Delaunay triangulation is called Voronoi diagram of P. This diagram is composed of regions corresponding to each point of P. The Voronoi region (a.k.a. Voronoi cell) corresponding to p_i is the set of points of the plane that are closer to p_i than to any other point p_j of P.

The interesting property of these triangulations is related to the fact that most triangles are *well shaped*, i.e. they are not obtuse (see Fig. 2.2). This is useful for Finite Element (FE) analysis which solves linear systems based on the adjacency matrix with weights that are function of the cotangent of triangle angles. For obtuse triangles, the cotangent is negative and the linear system can lose its semi-definite positiveness and consequently, its stability.

It is possible to define Delaunay triangulations for 3D surfaces, provided that distances are computed by the *geodesic distance* instead of the Euclidian distance. The geodesic distance is presented later in this Chapter.

The remainder of the thesis is now entirely dedicated to shapes represented by 3D triangular meshes (a.k.a. *triangulated surfaces*).

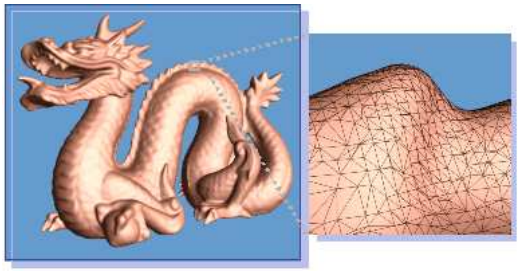

Figure 2.1: *The Stanford* Dragon *model. The close-up shows the details of a region of the triangulated surface.*

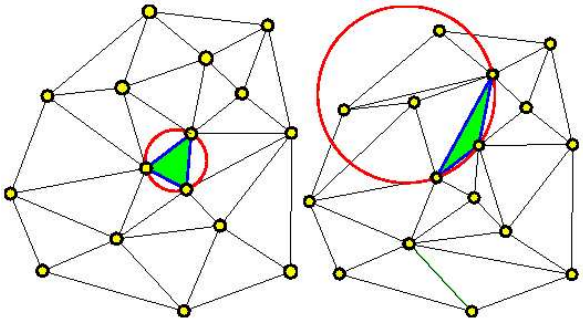

Figure 2.2: *On the left, Delaunay triangulation of a set of 2D points. On the right, arbitrary triangulation of the same set of points. The triangles of a Delaunay triangulation are well-shaped, the circumcenter circle of triangle only contains the points of this triangle. This is not true for the other triangulation composed of many obtuse triangles. The operations enabling to iteratively modify a triangulation to another are called* edge flips *(image courtesy of François Cayre).*

2.3 Digital Topology

Another important notion is the *topology* of a 3D mesh. This word is frequently used with different meanings in papers related to Computer

Graphics and specifically to 3D watermarking. Indeed, topology can be referred to as:

- a synonym of connectivity [30],
- a description of the local geometric configuration of a face or vertex neighborhood [99],
- the mathematical study of the properties (genus, Euler characteristic, ...) of geometric solids or figures that are not changed by homeomorphisms [65, 134].

These definitions are very different even though connectivity properties depend on the shape topology. In order to avoid confusions, we distinguish these different meanings by respectively naming them connectivity, local geometric configuration and shape topology [134].

Now we define some notions of shape topology related to 3D triangular meshes. A very important topological feature of a triangulated surface S is the Euler-Poincaré characteristic, noted $\chi(S)$:

$$\chi(S) = n - n_e + n_f \tag{2.6}$$

where n is the number of points, n_e is the number of edges and n_f is the number of triangles of S. The Euler-Poincaré characteristic for a sphere is equal to 2 and for a torus is equal to 0.

The Euler equation links the Euler-Poincaré characteristic to the topological properties of the surface:

$$\chi(S) = n - n_e + n_f = 2(c - g) - b \tag{2.7}$$

where g is the *genus* of the surface, c is the number of connected parts of the surface and b is the number of boundaries of the surface. The genus refers to the number of holes of a surface: the genus of a sphere is 0 and the genus of a torus is 1.

2.4 Parameterization

The parameterization of a surface is a useful mathematical tool that enables to express the surface in a local 2D frame (see Fig. 2.3). Parameterization of triangulated surfaces (a.k.a. *digital parameterization*) has been deeply explored by Computer Graphics researchers to improve the quality of texture mapping (see Fig. 2.4), remeshing etc.

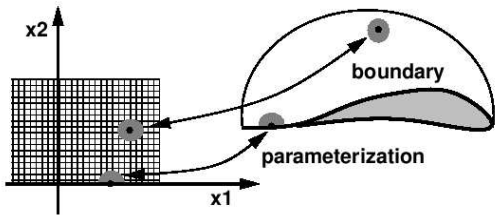

Figure 2.3: *Parameterization of a surface. For each point of the surface a local 2D orthogonal frame (u, v) is defined in the tangent plane. The 2D (u, v) space is named* parameter space.

The parameterization of a manifold surface S embedded in $\mathbb{R}^3$ is defined by a map $\phi : U \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi(u, v) = S(x, y, z)$. The parameterization must respect the following conditions. Let $\boldsymbol{s}_u$ and $\boldsymbol{s}_v$ be the tangent vectors in u and v directions at a point p of S. Then, for any point p of S, $\|\boldsymbol{s}_u \wedge \boldsymbol{s}_v\|_p \neq 0$. Furthermore, $\boldsymbol{s}_u \wedge \boldsymbol{s}_v|_p = \boldsymbol{n}_p$, where $\boldsymbol{n}_p$ is the point normal in p.

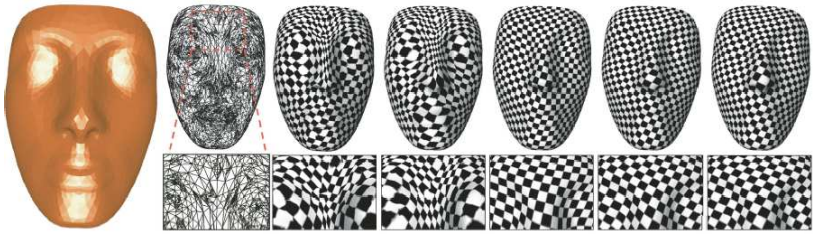

Figure 2.4: *Digital parameterization of a human face model. For each point of the triangulated surface a local 2D orthogonal frame (u, v) is defined in the tangent plane. The quality of this parameterization is evaluated by mapping on the 3D surface, a dashed texture defined in the 2D parameter space. From left to right, the original model, the flat-shaded model and texture mappings respectively using Tutte, Floater, authalic, conformal and intrinsic parameterizations (image courtesy of Pierre Alliez [43]).*

Sometimes it is not possible to define such a parameterization for any kind of surface, but it is always possible to decompose the surface in *patches* (a.k.a. *charts*) for which a parameterization exists [44].

2.4.1 Digital parameterization

Several parameterizations have been proposed for different topologies. The most common are planar parameterization and spherical parameterization. Focusing on planar parameterizations, the goal of digital parameterization is to minimize the distortion between the triangulated surface S embedded in $\mathbb{R}^3$ and the 2D parameter space U. In this context, «minimizing distortions» is understood as angle, edge length and triangle area preservation maximization. Here, we give the foundations of shape distortion energy estimation as well as a detailed description of most common digital parameterizations.

2.4.2 Distortion energy estimation

There is an infinity of possible mappings between a triangulated surface S embedded in $\mathbb{R}^3$ and a region of $U \subseteq \mathbb{R}^2$. Here, we consider the mapping ϕ which satisfies the following conditions:

- **Linearity:** On each triangle of S, ϕ must be continuous and linear.
- **Bijectivity:** This enables to compute the inverse of the mapping ϕ^{-1}.
- **Local minimization of the distortion:** Locally the mapping must perform minimal energy transforms (e.g. rigid transforms rather than bendings).

The mapping ϕ between S and U can be considered as an application $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Since this function is at least C^1, it can be locally approximated by a linear approximation. This approximation is given by a 3×2 matrix A which we can consider as the application from a region of $\mathbb{R}^2$ to the local tangent plane on S embedded in $\mathbb{R}^3$. If we introduce coordinates in the tangent plane of S, then the coordinates transformation from $\mathbb{R}^3$ to $\mathbb{R}^2$ can be specified by a 2×3 matrix P. The composition of these transformations applies the tangent plane of S to the parameter plane. This is specified by a 2×2 matrix B:

$$B = PA. \tag{2.8}$$

If B is a rigid transform (i.e. rotation or reflection), then there is no local distortion. In other cases, we can compute the invariants of B to measure the varied cases of local distortion.

These distortions can be classified as:

- uniform scaling: size distortion.
- non-uniform scaling and bending: shape distortion.

These distortions can be written in function of the derivatives of f. The matrix A of the linear approximation of f is composed of its partial derivatives:

$$A = (f_u, f_v) \tag{2.9}$$

where f_u and f_v are two vectors of $\mathbb{R}^3$. If e_1 and e_2 are normalized orthogonal basis vectors of R^2, we can write $f_u = Ae_1$ and $f_v = Ae_2$. Considering the unit square composed of e_1 and e_2 with the origin specified in the down left vertex of this square, we can also write that $|f_u \wedge f_v|$ is the area of the transformed paralelogram defined by e_1 and e_2. This gives us a measure of size distortion. Now if f_u and f_v are orthogonal and $|f_u| = |f_v|$, then there is no shape distortion.

We may thus consider $f_u \cdot f_v$ as a measure of torsion and ${f^2}_u - {f^2}_v$ as a measure of non-uniform scaling along parameter directions u and v.

We may observe the following important inequalities:

$$|f_u \wedge f_v| \leqslant |f_u|\,|f_v| \leqslant \frac{1}{2}({f^2}_u + {f^2}_v). \tag{2.10}$$

which is an equality in case of absence of distortion. If we integrate this local distortion estimation over the parameter space, we have:

$$\int |f_u \wedge f_v|\,dudv \leqslant \int |f_u|\,|f_v|\,dudv \leqslant \int \frac{1}{2}({f^2}_u + {f^2}_v)dudv. \tag{2.11}$$

The last term of the inequality is called *Dirichlet's energy* which is easy to minimize. The parameterizations minimizing this energy are referred to as *harmonic parameterizations*. The first term is simply the area of the surface which is constant if the surface does not change. If there is no distortion, then the Dirichlet's energy is equal to the surface area and the parameterization is called *conformal*. In practice, the application of the surface boundary is fixed to the boundary of U in parameter space. If the distortion between these two boundaries is null then it is possible to find a conformal parameterization. If the boundary distortion is not null then the best

minimization is given by the harmonic parameterization. It is also possible to minimize the shape distortion energy $\int |f_u|\,|f_v|\,dudv$ which is a highly non-linear process. This is why it is easier to deal with the Dirichlet's energy.

We now present the most common parameterizations: Tutte, harmonic, conformal, authalic and intrinsic parameterizations.

2.4.3 Tutte parameterization

The most simple parameterization (see Chapter 7) is given by Tutte parameterization (a.k.a *Tutte projection*). This is very close to Isomap embedding of graphs. For each point $p_i \in \mathbb{R}^3$ of a triangulated surface, we must find parameter coordinates $q_i = (u_i, v_i)$ such that the distortion energy is minimized.

In this case, the parameterization is simply given by minimizing:

$$\sum_i \sum_{j \in N(i)} w_{i,j} \left| q_i - q_j \right|^2, \tag{2.12}$$

where $w_{i,j} = \frac{1}{|N(i)|}$ is the inverse of the valence of point i. The minimization results in a linear system:

$$\sum_i \sum_{j \in N(i)} w_{i,j}(q_i - q_j) = 0. \tag{2.13}$$

with boundary conditions for points belonging to boundary edges. This parameterization does not minimize distortions but produces a regular triangulation which can be used as an initialization of non-linear distortion energy minimization techniques.

2.4.4 Conformal and harmonic parameterizations

We focus on the minimization of Dirichlet's energy which is here extended to triangulated surfaces by adding its piecewise-linear contributions in each triangle. Considering two triangles, the first embedded in $\mathbb{R}^2$ (parameter space) and the second in $\mathbb{R}^3$ then it is possible to find [109]:

$$E_{Di} = \sum_{j \in N(i)} cot\alpha_i \left| q_i - q_j \right|^2 \tag{2.14}$$

where α_i is the angle i of the triangle embedded in $\mathbb{R}^3$ and $\left| q_i - q_j \right|^2$ is the length of the edge in parameter space which is seen by this angle. The total energy is given by:

$$E_D = \sum_i \sum_{j \in N(i)} \left(cot\alpha_{ij} + cot\beta_{ij} \right) \left| q_i - q_j \right|^2, \tag{2.15}$$

where $N(i)$ is the 1-ring of i. This function is quadratic in terms of q_i. Here again we must solve a linear system obtained by differentiating this equation and imposing boundary conditions. If the triangulated surface is not a Delaunay triangulation, the system may become instable since the cotangents become negative. In this case, other weights are possible to estimate an harmonic parameterization (see Chapter 8). When there is no distortion, the Dirichlet's energy is the conformal energy and the parameterization is also conformal. If there is still distortion after minimization, the Dirichlet's energy is called *harmonic energy* and the parameterization is also referred to as harmonic.

2.4.5 Authalic parameterization

It is possible to minimize area distortions and to produce an *authalic* parameterization. Desbrun et al. [43] propose to minimize the following energy:

$$E_{authalic} = \sum_i \sum_{j \in N(i)} \frac{\left(cot\gamma_{ij} + cot\delta_{ij} \right)}{\left| p_i - p_j \right|^2} (q_i - q_j)^2. \tag{2.16}$$

where γ_{ij} and δ_{ij} are angles i adjacent to the edge $e(i,j)$ and p_i, p_j are the 3D coordinates of points i and j. This energy is also quadratic in terms of q_i and minimizing it can be done by solving a linear system.

Intrinsic parameterizations [43] tend to minimize a linear combination of the Dirichlet's energy and the authalic energy. These parameterizations are illustrated on Fig. 2.4.

2.5 Differential Geometry

In 2D, the *curvature* κ at a given point p on a curve is defined as the inverse of the radius r of the *osculating circle* at p:

$$\kappa = \frac{1}{r}. \tag{2.17}$$

The osculating circle (see Fig. 2.5) can be found as follows: for any two points p_1 and p_2 near p compute the unique circle passing through p_1, p_2, p (if these points are collinear, than the circle has infinite radius). The osculating circle is the circle that we obtain in the limit if p_1 and p_2 are moved towards p. In 3D, we have the following definitions.

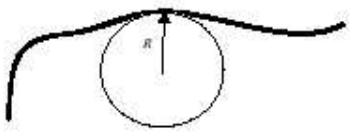

Figure 2.5: *Curvature of a 2D curve at a point p. The osculating circle is represented. The curvature at p is given by the inverse of the radius of the osculating circle.*

A *normal curve* is the intersection of a surface with a plane containing the normal $\boldsymbol{n}$ at p (see Fig. 2.6). For a given direction $\boldsymbol{d}$ in the tangent plane, there is a unique normal curve, obtained by intersecting the plane spanned by $\boldsymbol{n}$ and $\boldsymbol{d}$ with the plane.

The curvature of a normal curve is called *sectional curvature*. The sign of the curvature is given by the sign of the scalar product of the normal direction at p and the normal associated to the normal curve.

The *principal curvatures* $(\kappa_{min}, \kappa_{max})$ are the minimal and maximal sectional curvatures (see Fig. 2.7).

The *principal curvature directions* $(\boldsymbol{\kappa_{min}}, \boldsymbol{\kappa_{max}})$ are the directions in the tangent plane for which the maximum and minimum are attained. These directions are perpendicular to each other [44] (see Fig. 2.8).

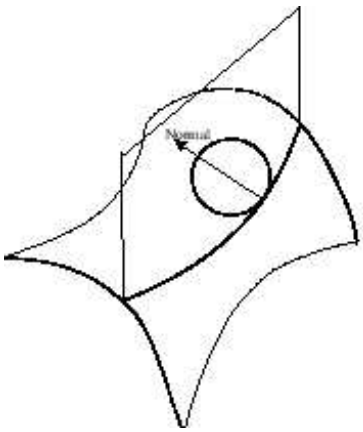

Figure 2.6: *Curvature of a 3D curve at a point p. The osculating circle is represented for the normal curve. The curvature at p is given by the inverse of the radius of this osculating circle and is called sectional curvature.*

The sectional curvature corresponding to the direction in the tangent plane forming an angle α with the first principal curvature direction is given by:

$$\kappa = \kappa_{max} \cos^2(\alpha) + \kappa_{min} \sin^2(\alpha) \tag{2.18}$$

Gaussian curvature (a.k.a. *intrinsic curvature*) and is defined as:

$$\kappa_G = \kappa_{max} \kappa_{min}, \tag{2.19}$$

and measured in $[m^{-2}]$. If the surface is isometrically deformed (i.e. the distances between points on the surface along the surface do not change) the Gaussian curvature is preserved. The sign of the gaussian curvature is negative for saddle points for which principal curvatures have different signs.

Mean curvature is defined by:

$$\kappa_H = \frac{\kappa_{max} + \kappa_{min}}{2}, \tag{2.20}$$

and is measured in $[m^{-1}]$. The sign of the mean curvature can help to segment concave and convex regions of the surface which have positive gaussian curvature (see Fig. 2.9).

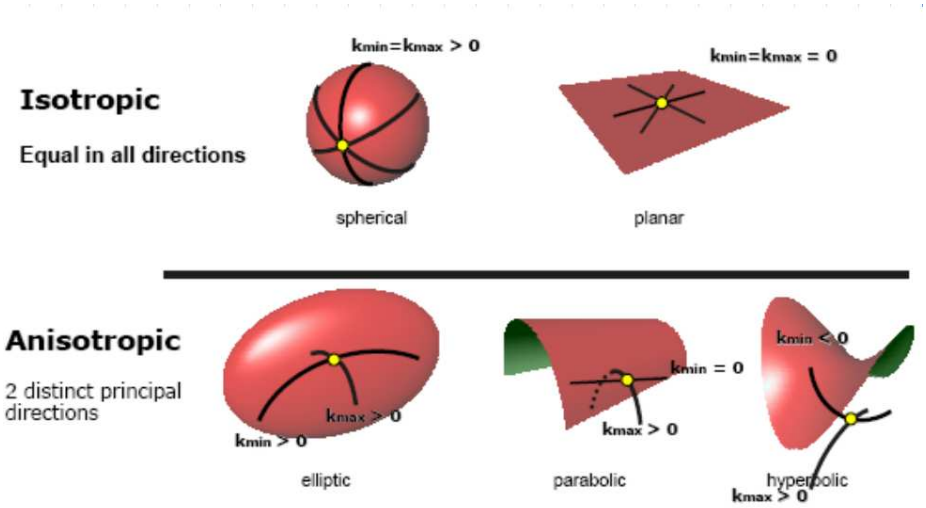

Figure 2.7: *Principal curvatures for basic 3D shapes.*

Given a parameterization (u, v) of a manifold differentiable surface $S(u, v)$, we already know that the point normal $\boldsymbol{n_p}$ of $p \in S$ is given by the cross product of basis vectors along parameter u and v directions. These vectors are given by $\boldsymbol{s}_u = \frac{\partial S}{\partial u}$ and $\boldsymbol{s}_v = \frac{\partial S}{\partial v}$ and $\boldsymbol{n}_p = \frac{\boldsymbol{s}_u \wedge \boldsymbol{s}_v}{\|\boldsymbol{s}_u \wedge \boldsymbol{s}_v\|}$. We can also define the following features:

$$E = (\boldsymbol{s}_u \cdot \boldsymbol{s}_u), F = (\boldsymbol{s}_u \cdot \boldsymbol{s}_v), G = (\boldsymbol{s}_v \cdot \boldsymbol{s}_v), \tag{2.21}$$

$$L = (\boldsymbol{n}_p \cdot \boldsymbol{s}_{uu}), M = (\boldsymbol{n}_p \cdot \boldsymbol{s}_{uv}), N = (\boldsymbol{n}_p \cdot \boldsymbol{s}_{vv}), \tag{2.22}$$

where $\boldsymbol{s}_{uu} = \frac{\partial^2 S}{\partial u \partial u}$, $\boldsymbol{s}_{uv} = \frac{\partial^2 S}{\partial u \partial v}$ and $\boldsymbol{s}_{vv} = \frac{\partial^2 S}{\partial u \partial v}$. Then we have formulas to compute the gaussian curvature:

$$\kappa_G = \frac{LN - M^2}{EG - F^2}, \tag{2.23}$$

as well as the mean curvature:

$$\kappa_H = \frac{LG - 2FM + EN}{2(EG - F^2)}. \tag{2.24}$$

We refer to *integral curvature* as the integral of a curvature feature on a given neighborhood. The integral gaussian curvature on a point p of a

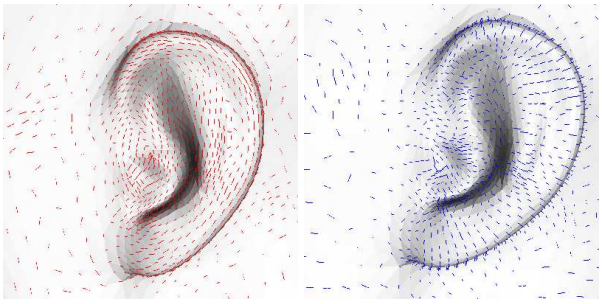

Figure 2.8: *Principal curvatures directions for a triangulated human ear. On the left, minimal curvature directions and on the right, maximal curvature directions.*

triangulated surface S is given by:

$$\sum_{j \in N_f(i)} \theta_j - 2\pi \tag{2.25}$$

where θ_i are the angles incident to p and $N_f(i)$ is the set of faces adjacent to p. This quantity is also known as *angle excess*. It must be divided by the area of the 1-ring to obtain the gaussian curvature. For a 2D planar mesh, the angle excess as well as the gaussian curvature are equal to zero.

Estimating the mean curvature on a triangulated surface can be done by using the following formula:

$$\frac{1}{2} \sum_{i \in N(i)} (\cot \alpha_i + \cot \beta_i) \|p - p_i\| \tag{2.26}$$

where angles α_i and β_i are the two angles opposite to the edge $e(i, j)$.

It is also possible to compute the curvature tensor at a point p of a triangulated surface S. The tangent plane is a first order approximation of the surface at p and is determined by the normal direction $\boldsymbol{n}_p$. The curvature tensor is a second order approximation of the surface S defined by directed derivatives of the normal. The directed derivative (in direction $\boldsymbol{t}$ of the tangent plane) of the normal direction $\boldsymbol{n}$ is given by:

$$D_{\boldsymbol{t}}(\boldsymbol{n}) = \boldsymbol{n}_u \boldsymbol{s}_u + \boldsymbol{n}_v \boldsymbol{s}_v \tag{2.27}$$

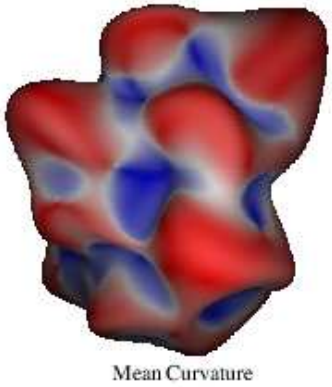

Figure 2.9: *Mean and gaussian curvatures of a molecula surface model. The red color is associated to convex regions (positive mean curvature and positive gaussian curvature), the blue color to concave regions (negative mean curvature and positive gaussian curvature). Grey regions are associated to negative gaussian curvature regions.*

and

$$\begin{pmatrix} \boldsymbol{n}_u \\ \boldsymbol{n}_v \end{pmatrix} = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \begin{pmatrix} u \\ v. \end{pmatrix} \tag{2.28}$$

The curvature tensor (a.k.a. *Second Fundamental Form*) in directions $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ of the tangent plane is given by:

$$\begin{aligned} T(\boldsymbol{t}_1, \boldsymbol{t}_2) &= (D_{\boldsymbol{t}1}\boldsymbol{n}) \cdot \boldsymbol{t_2} && (2.29) \\ &= (D_{\boldsymbol{t}2}\boldsymbol{n}) \cdot \boldsymbol{t_1} && (2.30) \end{aligned}$$

and

$$T(\boldsymbol{t}_1, \boldsymbol{t}_2) = \begin{pmatrix} u_1 & v_1 \end{pmatrix} \begin{pmatrix} L & M \\ M & N \end{pmatrix} \begin{pmatrix} u_2 & v_2 \end{pmatrix} \tag{2.31}$$

The eigenvalues of the curvature tensor are the principal curvatures and the corresponding eigenvectors are the principal curvature directions. Furthermore, the determinant of the curvature tensor is equal to the gaussian curvature and the trace of the tensor is equal to the mean curvature. Chapter 8 gives a deeper analysis of the curvature tensor.

2.6 Geodesic distances

Another important tool for digital surface processing is the geodesic distance that we use in Chapters 8 and 9. The *geodesic distance* between two points p_1 and p_2 of a surface S is given by the length of the shortest path on S that connects these two points. This distance is superior or equal to the Euclidian distance. The geodesic path and the geodesic distance can be approximated by the well-known Dijkstra algorithm. However, this approximation gives paths that follow the edges of the triangulated surface and the estimated geodesic distance is an upper-bound estimation of the true geodesic distance (see Fig. 2.10).

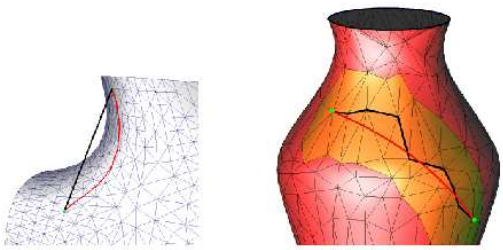

Figure 2.10: *Geodesic distance estimations on triangulated surfaces. On the right, comparison of the Euclidian distance and geodesic distance. On the right, comparison of the Dijkstra shortest path in black and the Fast Marching shortest path in red. The latter path is interpolated on triangles while Dijkstra path is restricted to mesh edges.*

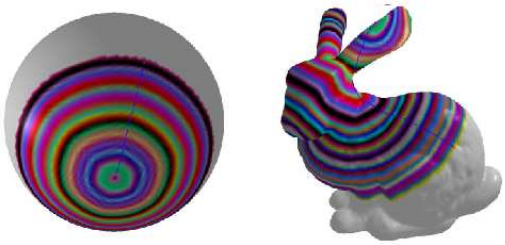

Figure 2.11: *Fast Marching wavefronts on triangulated surfaces. On the left, a sphere and on the right the* Bunny *model.*

A better approximation is based on the *Eikonal equation* which is non-linear:

$$|\nabla U(x)| = P(x), \tag{2.32}$$

where U represents the weighted distance function to a set of points and $F = \frac{1}{P} \geq 0$ is the speed of the propagating front. In order to solve this equation we must find the value of $U(x)$ in each triangle (x, x_1, x_2) of S. Let us consider a 2×2 matrix M where rows are composed of vectors $\overrightarrow{xx_1}$ and $\overrightarrow{xx_2}$ and using an first order approximation:

$$\nu := \begin{pmatrix} U(x) - U(x_1) \\ U(x) - U(x_2) \end{pmatrix} = U(x)a + b, \tag{2.33}$$

where $a = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $b = -\begin{pmatrix} U(x_1) \\ U(x_2) \end{pmatrix}$. The derivative along the direction $\overrightarrow{xx_k}$ is by definition equal to $< \nabla U, \overrightarrow{xx_k} >$. Thus, $\nabla U \cong M^{-1}\nu$. If we introduce these results in the Eikonal equation and denoting $Q := (MM^T)^{-1}$ we have:

$$(a^T Q a)U(x)^2 - (2a^T Q b)U(x) + b^T Q b = P(x)^2. \tag{2.34}$$

which is a second order equation with a positive discriminant ensured by *Cauchy-Schwartz* theorem [107]. Once we have solved the Eikonal equation, a shortest path C can be computed by solving the following differential equation:

$$\frac{dC}{dt} = -\overrightarrow{\nabla U} \tag{2.35}$$

with $C(0) = x_1$. This estimation method is often referred to as the *Fast Marching* algorithm. The paths estimated by using this algorithm cross triangles of the mesh and are not restricted to the edges as the Dijkstra estimation. It follows that the estimated geodesic distance is also better than the Dijkstra estimation. Results are compared on figure 2.10. Propagating wavefronts are illustrated on figure 2.11

2.7 Spectral Analysis

Among signal processing tools that have been recently extended to 3D meshes, the *spectral decomposition* has been first used for filtering purposes. Then, based on several results of graph theory and spectral analysis of matrices, the spectral decomposition has been used in compression, segmentation, registration and watermarking applications. Chapter 7 gives a detailed presentation of this 3D mesh transform.

2.8 Multiresolution Analysis

Multiresolution analysis and *wavelets* have received considerable attention in recent years. The basic idea behind multiresolution analysis (see Fig. 2.12) is to decompose a complicated function into a «simpler» low resolution part, together with a collection of perturbations, called *wavelet coefficients*, necessary to recover the original function. For many of the functions encountered in practice, a large percentage of the wavelet coefficients are small, meaning that good approximations can be obtained by using only a few of the largest coefficients. Impressive «lossy» compression rates for 3D meshes have been achieved using this type of approximation [9]. Based on *subdivision surfaces*, the multiresolution analysis also enables continuous Level-Of-Detail control (see Chapter 3), compression of functions defined on surfaces, surface optimization, numerical solution of integral and differential equations on surfaces as well as multiresolution editing of a surface.

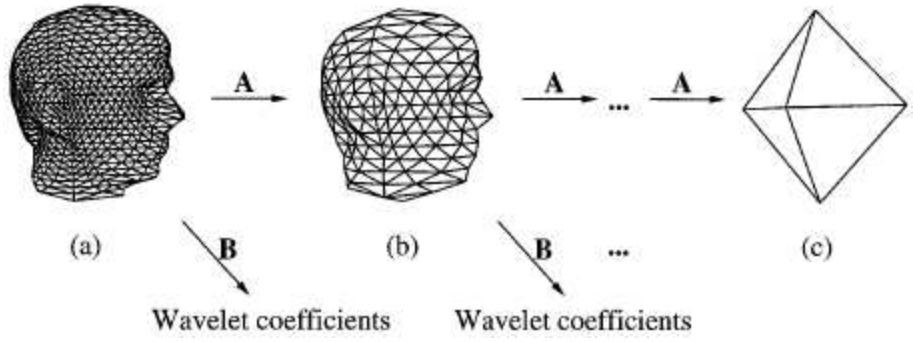

Figure 2.12: *Decomposition of a triangulated surface in the multiresolution framework. (a) is the original model, (b) a coarser version of this model and (c) the coarsest level of the multiresolution analysis. These cascades are referred to as a* filter bank *algorithm which is composed of low-pass filters* $\boldsymbol{A}$ *and high-pass filters* $\boldsymbol{B}$*. These filters are also called* analysis filters*.*

Let us first denote K^0 as the coarsest mesh obtained by the multiresolution analysis. K^0 is called the *base mesh*. Meshes K^j are the j-level resolution of the original mesh in the multiresolution filter-bank. The two

basic ingredients of multiresolution analysis are a sequence of nested linear function spaces and an inner product. Lounsbery et al. [91] use a sequence of spaces $V^0 \subset V^1 \subset ...$ associated with the base complex. To describe meshes, the approximation spaces V^j consist of piecewise linear functions; specifically, V^j is the space of continuous piecewise linear functions over a partition K^j of K^0 created by performing j recursive steps of 4-to-1 splitting to the faces of K^0, as shown in figure 2.13. We can thus say that the number of points of the j-level resolution mesh is equal to $2^j n_0$ where n_0 is the number of points of the base mesh. As j increases, the

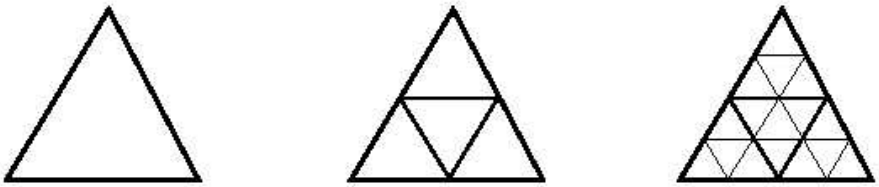

Figure 2.13: *4-to-1 splitting recursive operations enable to refine a coarse mesh. (a) the initial face, (b) after one 4-to-1 split, (c) after two 4-to-1 splits.*

triangulated surface K^j obviously becomes more dense, and so the functions in V^j are better able to model arbitrary continuous functions on K^0. The inner product used by Lounsbery et al. is the standard inner product defined as [45]:

$$< f, g >:= \int_{\boldsymbol{x} \in K^0} f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} \tag{2.36}$$

where $d\boldsymbol{x}$ is the differential area of K^0 embedded in $\mathbb{R}^m$, so that all faces have unit area.

The inner product is used to define the following orthogonal complement spaces, also called *wavelet spaces*.

$$W^j := \{f \in V^{j+1} | < f, g >= 0 \, \forall g \in V^j\} \tag{2.37}$$

Intuitively, W^j captures the detail that is missed when a function in V^{j+1} is approximated by a function on V^j.

Basis functions for V^j are called *scaling functions*. In the piecewise linear case, particularly simple scaling functions for V^j are the *hat functions* on K^j: the i-th hat function $\phi_i^j \in V^j$ is the unique function in V^j that is one at $\boldsymbol{x}_i^j$ and zero for all other points of K^j.

A *wavelet* $\psi_i^k(\boldsymbol{x})$ (see Fig. 2.14) is a basis function for one of the wavelet spaces W^k. A *wavelet basis* for V^j consists of a basis for V^0 together with bases for the wavelet spaces $W^0, ..., W^{j-1}$.

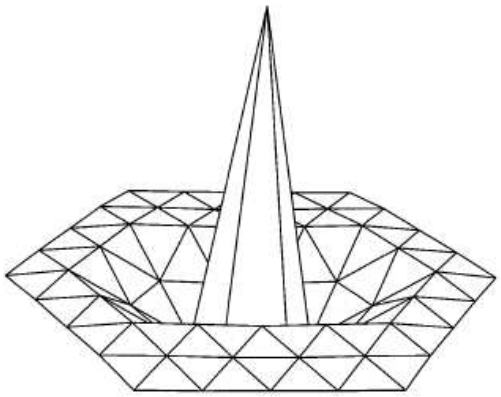

Figure 2.14: *Wavelet function example for a triangulated surface.*

The parameterization $\rho^j \in V^j$ for V^j can be expanded in the hat function basis as:

$$\rho^j(\boldsymbol{x}) = \sum_i v_i^j \phi_i^j, \quad \boldsymbol{x} \in K^0 \tag{2.38}$$

where v^j denotes the point positions of the geometry associated to K^j. A multiresolution representation of ρ^j refers to its expansion in a wavelet basis:

$$\rho^j(\boldsymbol{x}) = \sum_i v_i^0 \phi_i^0 + \sum_{k=0}^{j-1} \sum_i w_i^k \psi_i^k(\boldsymbol{x}), \quad \boldsymbol{x} \in K^0, \tag{2.39}$$

where w_i^k are the wavelet coefficients.

The filterbank analysis algorithm can be used to convert between the hat function expansion and the multiresolution representation. The geometric interpretation of filterbank analysis shown in Fig. 2.12. The full detail model, described by $\rho^j(\boldsymbol{x})$ is successively decomposed into a lower resolution approximation together with a collection of coefficients that multiply the wavelets. The result is a simple base mesh together with wavelet coefficients at various levels of detail. The operators $\boldsymbol{A}$ and $\boldsymbol{B}$ in Fig. 2.12 refer to sparse matrices whose entries are given by Lounsbery et al. The filterbank analysis has an inverse process called *filterbank synthesis* that recovers the full resolution model from its multiresolution representation.

2.9 Conclusion

This Chapter has given a detailed overview of the varied digital shape processing tools that have been recently developed by the Computer Graphics community for 3D triangulated surfaces. The mathematical background of the digital parameterization, geodesic distance, curvature tensor estimations as well as of the multiresolution analysis have been presented with references to their applications and/or the Chapters of the thesis where these tools are used.