INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Simplex Meshes: a General Representation for 3D Shape Reconstruction

Hervé Delingette

N° 2214

Mars 1994

PROGRAMME 4

Robotique,

image

et vision

Rapport de recherche

1994

ISSN 0249-6399

Simplex Meshes: a General Representation for 3D Shape Reconstruction

Hervé Delingette

Programme 4 — Robotique, image et vision
Projet Epidaure

Rapport de recherche n° 2214 — Mars 1994 — 55 pages

Abstract: In this report, we develop the concept of simplex mesh as a representation of deformable models. Simplex meshes are simply connected meshes that are topologically dual of triangulations. In a previous work, we have introduced the simplex mesh representation for performing recognition of partially occluded smooth objects. In this paper, we present a physically-based approach for recovering three-dimensional objects, based on the geometry of simplex meshes. Elastic behavior is modeled by local stabilizing functionals, controlling the mean curvature through the simplex angle extracted at each vertex. Those functionals are viewpoint-invariant, intrinsic and scale-sensitive. They control either the normal orientation or the curvature continuity of the mesh or its closeness to a given reference shape.

Unlike deformable surfaces defined on regular grids, simplex meshes can have their topology locally altered. We have developed an adaptation and refinement process that respectively concentrates vertices and increases the mesh resolution at highly curved or inaccurate parts. The adaptivity of a mesh is governed by the minimization of a local energy. Furthermore, we have defined some general mesh transformations that enables the recovery of complex models from parts of simpler shapes. We present several modeling results extracted from structured range data or three dimensional images, including objects of complex topology and a complete model of a human body.

(Résumé : tsvp)

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Représentation et reconstruction de surfaces à l'aide de maillages simplexes

Résumé : Dans ce rapport, nous développons le concept de maillage simplexe en tant que représentation de modèles déformables. Les maillages simplexes sont caractérisés par un nombre d'arètes constant à chaque sommet. Nous avons précédemment introduit les maillages simplexes dans le but d'effectuer la reconnaissance de surfaces lisses en milieu complexe. Nous présentons ici un système de reconstruction d'objets tridimensionnels fondé sur la déformation physique de maillages simplexes. La nature élastique des déformations est modélisée à l'aide de forces stabilisatrices qui contrôlent la courbure moyenne discrète à chaque sommet. Ces stabilisateurs ont une formulation invariante par isométrie, sont exprimés à l'aide de paramètres intrinsèques et sont définis en fonction de l'échelle à laquelle on veut régulariser le maillage. De plus, on peut contrôler la régularité du maillage en établissant ou bien une continuité d'orientation des normales, ou bien une continuité de la courbure moyenne ou enfin, une mémoire de forme a priori.

Contrairement aux modèles déformables utilisant des maillages rectangulaires réguliers, les maillages simplexes sont adaptatifs. En particulier, nous avons défini un algorithme permettant de concentrer et d'ajouter des sommets aux endroits de forte courbure. Nous avons également défini des opérations de transformations de maillages simplexes permettant de créer des modèles géométriques complexes en assemblant plusieurs modèles de formes plus simples. Nous présentons plusieurs modèles reconstruits à partir de données tridimensionnelles pouvant avoir une topologie complexe.

1 Introduction

The emergence of high resolution three-dimensional images either in the form of range data or voxel images, enforces the need for general shape reconstruction techniques. Despite recent advances in imaging technology, segmentation and filtering must be performed in a first stage on raw data in order to recover models suited for recognition or motion tracking. The difficulty stems from the necessary flexibility of object reconstruction systems to include a wide variety of man-made or natural shapes. Moreover, the increasing resolution of three dimensional data images, require the ability to generate optimal objects representations with a large amount of data compression. Hence, flexibility should be achieved both in terms of geometry and topology. Geometry relates to the local control of shape whereas topology relates to the global model structure.

Dynamically deformable models were first proposed by Terzopoulos *et al.*[TWK87] and have attracted significant interest for their intuitive and clay-like behavior. A common approach consists in formalizing the deformation as a variational problem involving an internal energy that enforces some continuity constraints, and an external energy controlling the closeness of fit. Several researchers have applied the dynamic model fitting scheme to range data or medical images[CCA92][DHI91b][VM93][Gue93][McI93][NA93][Lei93].

Elastic models successfully address the problem of shape control. However, few researchers have proposed general adaptive reconstruction techniques for solving both geometric and topological aspects. In particular, surface models based on regular grids are not well suited for representing surfaces with a spherical topology[Gue93] and moreover cannot be easily locally refined or decimated. Local nonuniform refinement over rectangular grids can be achieved by overlaying hierarchical splines[WW92] but the problem of changing the genus or connectivity of tensor product splines has not yet been addressed.

Triangulations on the other hand, are well-suited for local topology control. Delaunay triangulations[Boi84] for instance, generate an optimal tessellation given a set of three dimensional data. However, splines defined over triangulations are difficult to handle, because a lot of nodal shape functions are needed in order to enforce C^1 or C^2 continuity across triangles (up to 18 nodal shape functions for C^1 continuity across triangles). Chen *et al.*[CS93] fitted a progressively refined triangulation on range data and used Gregory-Bézier patches for constructing a G^1-continuous representation. In [McI93], McInerney used a semi-regular triangulated sphere for a finite-element based deformable balloon, but provided only uniformed refinement capabilities. Vasilescu *et al.*[VT92] developed an adaptive and discontinuity-preserving algorithm for meshes of interconnected springs. Hoppe *et al.*[HDDM93] addresses both surface reconstruction and mesh simplification problems by minimizing a single objective function over an initial triangulation. However, they do not provide any mechanism for smoothing the triangulated mesh in order to interpolate missing part of the data or removing spurious noise.

This paper presents a shape reconstruction algorithm that offers both geometric and topological flexibility. As in Hoppe *et al.*[HDDM93], we chose to represent a surface with a discrete mesh without considering a continuous surface representation. We use a simply connected mesh or *simplex mesh* as a surface representation in a deformable model fitting approach. The simplex mesh representation has several desirable properties that makes them well suited for the recovery of geometric models from range data. The surface reconstruction system that we are presenting in this paper, has the following characteristics:

Generality The simplex mesh representation is general because it can represent all types of orientable surfaces regardless of their genus and end numbers. Moreover, simplex meshes can model the deformation of k dimensional surfaces by a straightforward generalization of internal forces expressions to $k+1$-simplex meshes.

Simplicity and Efficiency of Implementation The meshes and contours are considered as physically based models. The displacement between two iterations is derived from the computation of an internal and external force. The internal force has a non linear expression but is derived from a finite differences method. Finite elements methods, unlike finite differences methods, usually require the inversion of large and sparse matrices but requires fewer vertices because the surface is interpolated between each elements. However, our approach is to use an adaptive scheme to minimize the number of vertices needed to describe a given shape. Consequently, most models shown in this report can be deformed in real-time.

Local Shape Functionals We have defined at set of shape functionals that are derived from the minimization of a local energy and have the properties of being viewpoint invariant and scale sensitive. Normal orientation or mean curvature continuity constraints may be applied to any part of a simplex mesh. Finally, we can constrain a mesh to be as close as possible to a given reference shape.

Adaptability Three different levels of adaptability was defined on a simplex mesh. A first algorithm consists in adapting the spacing of vertices in order to obtain a concentration of vertices at parts of high mean curvature. It therefore releases an optimal shape description for a given mesh. The second algorithm consists in refining the mesh by adding vertices at parts that do not correctly fit the data. This procedure controls the overall accuracy of a mesh. Finally, we adapt the mesh topology by creating end contours where the data is incomplete.

Contour-Surface Interaction Contours or closed curves may be simply defined on a simplex mesh. Those contours are deformable models and the contact between a surface mesh and a contour can be controlled either in terms of normal orientation constraint or in terms of mean curvature. The combination of surface and contour models is an important feature for recovering complex objects.

The principal contributions of this paper are:

- It introduces new geometric quantities such as the *simplex angle* or the *discrete mean curvature*. Furthermore, we characterize the mean curvature on a three dimensional surface, in terms of minimal contact between the minimal sphere and the surface.

- It introduces a set of internal forces that enforce smoothness or shape constraints by controlling the simplex angle at each vertex. This method sharply contrasts with traditional expressions of internal forces by its intrinsic, viewpoint invariant and scale sensitive nature. Furthermore, they do not entail any shrinking effect on the mesh.
- It presents an algorithm for concentrating vertices at parts of high curvature by locally minimizing an energy. The mesh deformation is therefore governed by two minimizations; the first one, acting on the simplex angle, controls the mesh geometry and the second one, acting on the metric parameters controls its topology.

The paper is organized as follows. In section 2, we describe the topological and geometric properties of simplex meshes especially in comparison with triangulations. In section 3, the internal forces enforcing smoothness or shape constraint on a simplex mesh are introduced . The automatic refinement process is explained in section 4 and section 5 presents various results on range data.

2 Simplex Meshes

A simplex mesh has constant vertex-connectivity. We will refer to a k-simplex mesh for a $(k+1)$-connected mesh. Formally, a k-Simplex Mesh (kSM) $\mathcal{M}$ of $\mathbb{R}^d$ is defined as a pair $(V(\mathcal{M}), N(\mathcal{M}))$ where:

- $$V(\mathcal{M}) = \{P_i\}, \{i = 1, \ldots, n\}, P_i \in \mathbb{R}^d \tag{1}$$
- $$N(\mathcal{M}) : \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}^{k+1}$$
  $$i \longmapsto (N_1(i), N_2(i), \ldots, N_{k+1}(i)) \tag{2}$$
- $\forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, k+1\}, \forall l \in \{1, \ldots, k+1\}, l \neq j$
  $$N_j(i) \neq i \tag{3}$$
  $$N_l(i) \neq N_j(i) \tag{4}$$

$V(\mathcal{M})$ is the set of vertices of $\mathcal{M}$ and $N(\mathcal{M})$ is the associated connectivity function. Equations (3) and (4) prevent a mesh from exhibiting loops or biarcs. We further restrict a k-simplex mesh to be connex such that for any two vertices there exists a path joining them. In this paper, we will only consider 1 and 2-simplex meshes. 1-simplex meshes are polygons and are well-suited for representing planar curves. 2-simplex meshes on the other hand, can represent any type of three dimensional surfaces and will be simply referred, in the remaining part of this paper, as "simplex mesh".

It is important to make a distinction between the topological nature of a mesh represented by its connectivity function $N(\mathcal{M})$ and its geometric nature corresponding to the position of its vertices $V(\mathcal{M})$. The structure of a simplex mesh is the one of a simply connected graph and does not in itself constitute a new surface representation. In particular, dual cell complexes have been studied extensively in the computer-aided design field. However, the main contribution of this work, is to exhibit the topological and geometric properties inherent to those meshes and demonstrate their relevance for object reconstruction as well as for object recognition.

2.1 Topology

We have coined the word *k-simplex mesh* in order to stress the existence of a k-simplex at each vertex. We recall that a k-simplex of $\mathbb{R}^d$, $k < d$ is a set of $k+1$ independent points of $\mathbb{R}^d$. For instance, a segment of non-zero length is a 1-simplex, a triangle of non-zero area is a 2-simplex and a tetrahedron of non-zero volume is a 3-simplex.

2.1.1 Duality triangulation-simplex mesh

The most interesting way of considering simplex meshes is through duality of triangulations. The structure of a k-simplex mesh is indeed closely related to the structure of a k-simplicial complex or k-triangulation. A k-simplicial complex of $\mathbb{R}^d$ is a set of p-simplices ($1 \leq p \leq k < d$) whose intersections are either empty or a simplex of lower dimension. In practice, we will only consider simplicial complexes where two p-simplices intersection is either empty or a $(p-1)$-simplex.

Property 1 (Duality between k-Simplex Mesh and k-Triangulation) *The dual graph of a k-simplex mesh is a graph of a k-triangulation. Therefore, a k-simplex mesh (resp. k-triangulation) can be obtained from a k-triangulation (resp. k-simplex mesh).*

Figure 1 shows how simplex meshes may be derived from polygons and triangulations. When a triangulation has boundaries, a simplex mesh may be derived by considering the dual of the bounding contours (see Figure1.c). We define a p-face of a k-simplex mesh as being the dual of a $(k-p)$ simplex of a k-triangulation. For instance, a 1-face of a 2-simplex mesh is an edge and a 2-face of a 2-simplex mesh, or simply a *face*, is a polygon. In general, a p-face of a k-simplex mesh is a $(p-1)$-simplex mesh and is, therefore, made of q-faces ($q < p$). A simplex mesh is said to be *regular* if all p-faces have the same number of vertices.

Table 1 describes the relation between faces of a k simplex mesh ($k \leq 2$) and the simplices of a k-triangulation.

	1−triangulation $\Longleftrightarrow$ 1−SM	2−triangulation $\Longleftrightarrow$ 2−SM
$p=1$	Vertex $\Longleftrightarrow$ Edge	Vertex $\Longleftrightarrow$ 2-face
$p=2$		Triangle $\Longleftrightarrow$ Vertex

Table 1: Duality between triangulations and simplex meshes

1-triangulation and 1-simplex mesh are 2-connected chains or *contours*. They have remarkable topological properties. First, they are dual of themselves and their edges may be considered as either a simplex of a triangulation or a face of a simplex mesh. Second, all

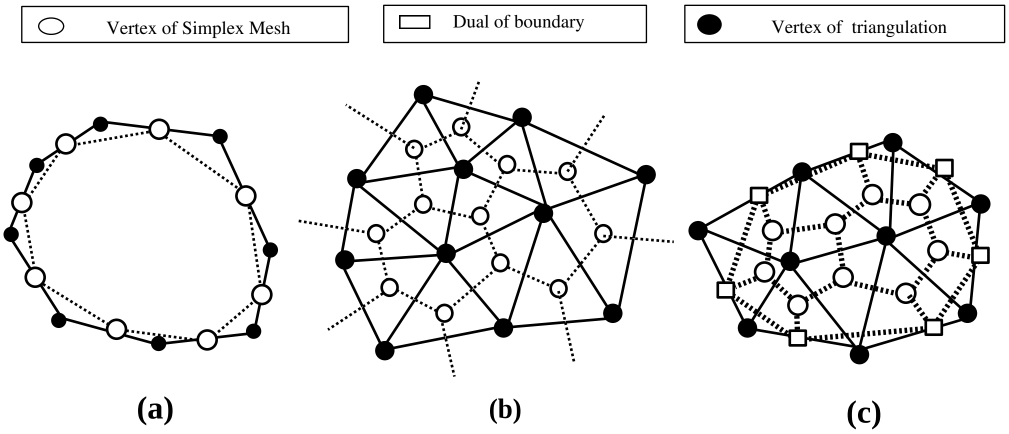


Figure 1: a) A 1-simplex mesh and its dual; b) A 2-simplex mesh and its dual triangulation; c) Same as (b). The dual of the triangulation boundary is considered to extract the simplex mesh

edges have two vertices and therefore *every contour is regular*. We add the notion of *end* or "invisible edge" in order to model "opened" contours. The number of ends of a contour uniquely characterizes its topology.

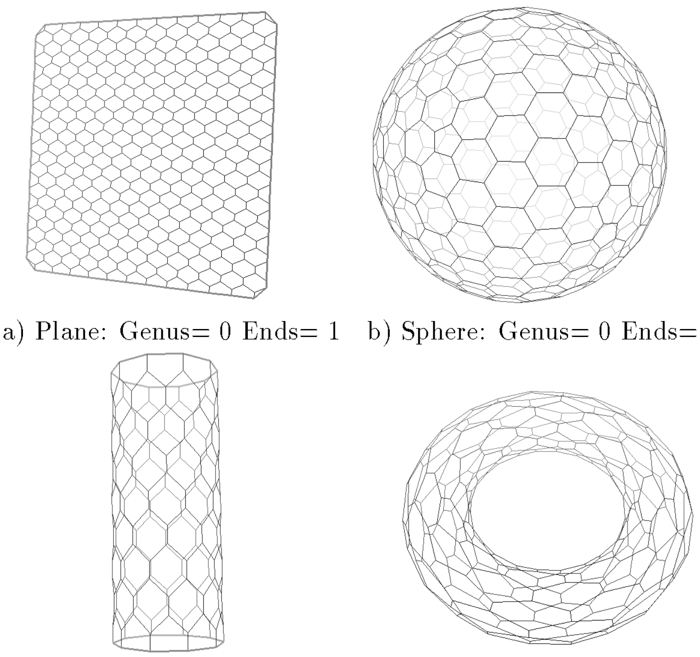
2-simplex meshes are made of vertices, edges and faces which are themselves contours. The Euler relation links the number of vertices V, of edges E and of faces F with the genus g of the mesh:

$$\begin{array}{c} F - \frac{V}{2} = 2 * (1 - g) \\ E = \frac{3V}{2} \end{array} \tag{5}$$

The genus characterizes the topology of the surface and corresponds to the number of *handles* of the surface [Car76]. In order to represent all kinds of two-manifolds we add, as for contours, the notion of "*end*" or "invisible face". The genus and the number of ends are used to classify two-manifolds since two surfaces with the same genus and end number can be mapped with a homeomorphism and therefore belong to the same topological class. Figure 2 shows some simplex meshes of different genuses and end numbers. Simplex meshes can represent the whole range of surfaces, even non-orientable surfaces.

Simplex Meshes are dual of triangulations. Hence, their connectivity functions are mapped by an homeomorphism. They are *topologically equivalent*, but they are *not geometrically equivalent*.

Property 2 (Geometric equivalence between simplex meshes and triangulations) *Simplex meshes are not geometrically equivalent to triangulations. Triangulations are equivalent to simplex meshes with planar faces.*



a) Plane: Genus= 0 Ends= 1 b) Sphere: Genus= 0 Ends= 0

c) Cylinder: Genus= 0 Ends= 2 d) Torus: Genus= 1 Ends= 0

Figure 2: Four 2-simplex meshes.

Proof For a three-dimensional simplex mesh, the number of degrees of freedom is $3V$ while for a three-dimensional triangulation of equivalent topology the number of degrees of freedom is $3F$. The Euler equation 5 states that there are almost twice as many vertices in a simplex mesh as faces. Hence, there cannot be an homeomorphism mapping a $3F$ dimensional space into a $3V$ dimensional space. However, if a simplex mesh has planar faces, then a simplex mesh is solely described by the $3F$ equations of each face, and there may exist an homomorphism transforming a simplex mesh into a triangulation.

An example of geometric duality is the Voronoï diagram associated with a Delaunay triangulation. Given a set of points of $\mathbb{R}^d$, the Voronoi diagram is a set of cells centered around each vertex such that a point in that cell is closer to that vertex than to any other vertices[Boi84]. A Voronoï cell is a polyhedron whose faces are bisectors of some pair of vertices. In general, a Voronoï diagram in $\mathbb{R}^d$ is $(d+1)$-connected, but when more than $(d+1)$ vertices lie on a d-sphere, the diagram may become degenerate.

In this paper, we make no assumptions on the planarity of simplex mesh faces. Hence, it is important to note that a simplex mesh representation is clearly distinct from a triangulation representation since we cannot associate an underlying triangulation with a given simplex mesh.

In practise, however, we need to triangulate a simplex mesh in order to compute the area of a face, or to properly visualize a mesh. Depending on the task required we can either build a triangulation with a dual structure such as in figure 1, or we can triangulate each face by joining each vertex to the centroid of the face vertices. For rendering, the former method has the advantage of creating a triangulation with few vertices but it results in a systematic underestimation of the mesh curvature. The latter method is used to estimate the area of a face.

Conversely, we can generate a simplex mesh from any given triangulation by a generating mesh with a dual structure (see figure 1) and by setting the vertex position at the centroid of each triangle. In particular, we can apply this method for generating a simplex mesh from any set of points, by considering the associated Delaunay triangulation.

2.1.2 Regularity

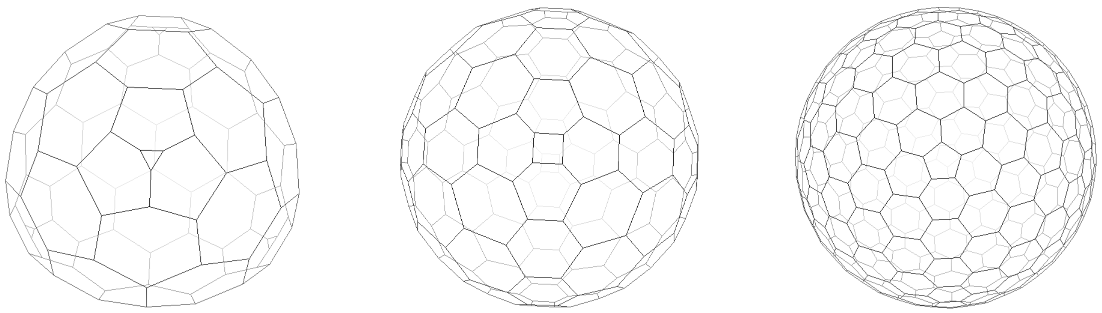
Rectangular grids as well as triangulations can regularly tessellate the plane. The third regular tessellation of the plane is made of hexagons and can be seen as a 2-simplex mesh of infinite extent. Table 2 compares the three types of representation in terms of connectivity.

All rectangular grids are regular since all faces have the same number of vertices. However, they cannot correctly represent surfaces of genus different than one, and especially surfaces topologically equivalent to a sphere. There are only three regular 2-simplex meshes of genus 0: the tetrahedron, the cube and the dodecahedron. With their duals, respectively the tetrahedron, the octahedron and the dodecahedron, they comprise the five Platonic solids. In practice, we define a set of quasi-regular primitive shapes but with a controlled number of vertices. We are currently using four primitives displayed in figure 2. The torus is the only regular simplex mesh only composed of hexagons. For spherical representation, we use regularly tessellated tetrahedron, octahedron or dodecahedron, made of a constant

	Vertex to Vertex Connectivity	Face to Face Connectivity	Regular Tesselation
Triangulation	$n \geq 3$	3	Equilateral Triangles
Rectangular Grid	4	4	Squares
Simplex Mesh	3	$n \geq 3$	Regular Hexagons

Table 2: Comparison between triangulation, rectangular grid and 2-simplex mesh

number of respectively, triangles, squares or pentagons and a variable number of hexagons controlled by a frequency parameter (see Figure 3).



a) b) c)

Figure 3: Duals of the tesselated tetrahedron (a), octahedron (b) and icosahedron (c) with frequency= 5

2.1.3 Contours on a simplex mesh

A contour $\mathcal{C}$ on a simplex mesh $\mathcal{M}$ is a 1-simplex mesh whose vertices and edges are subsets of $\mathcal{M}$. Precisely, given a 2SM $\mathcal{M} = (V(\mathcal{M}) = \{P_i\}(i = 1, \ldots, n), N(\mathcal{M}))$, a contour $\mathcal{C}$ is defined as $\mathcal{C} = (J(\mathcal{C}))$ with:

$$\begin{aligned} J(\mathcal{C}) : \{0, \ldots, l-1\} &\longrightarrow \{1, \ldots, n\} \\ i &\longmapsto J(i) \end{aligned} \tag{6}$$

with the following three properties:

- $\forall i \in \{0, \ldots, l-1\}, \exists j \in \{1, 2, 3\}, J((i+1) \bmod l) = N_j(J(i))$
- $\forall i \in \{0, \ldots, l-1\}, \forall j \in \{0, \ldots, l-1\}, P_{J(i)} \neq P_{J(j)}$

- $\forall i \in \{0, \ldots, l-1\}, \exists j \in \{1,2,3\}$ such that $\forall k \in \{0, \ldots, l-1\}, N_j(J(i)) \neq J(k)$.

The first property states that a contour is closed and that neighboring vertices on $\mathcal{C}$ are connected on $\mathcal{M}$. The second property prevents a contour from self-intersecting. The third property states that a vertex on a contour cannot have its three neighbors on the same contour. Definition 6 are consistent with the definition of a 1-simplex mesh, therefore properties of section 2.1.1 prevail.

In terms of surface topology, contours on a 2SM can be classified in two categories depending whether they are "dividing" or not. A contour is said to be "dividing" if there are two vertices such that all paths between the two vertices cross the contour. Intuitively, a "dividing" contour creates two disjoint regions on a mesh (see Figure4). Contours on meshes of genus zero other than p-faces are necessary "dividing" whereas contours on meshes of higher genuses may be of either types.

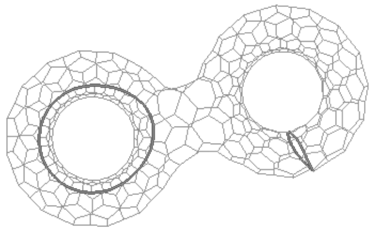
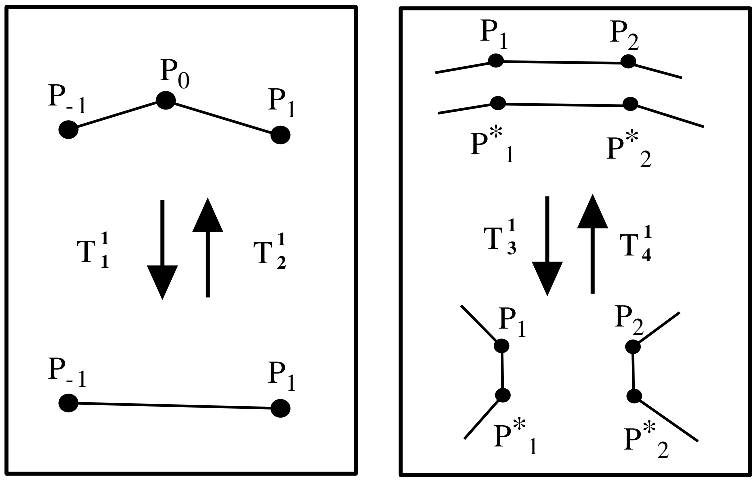


Figure 4: The right contour is dividing while the left one is not

2.1.4 Mesh Transformation Operations

Simplex Meshes as well as triangulations provide the abilitity to have their topology locally altered. Many possible transformations can be defined to alter a triangulation or a simplex mesh. Among the most widely used on triangulations are the "edge split", "edge collapse" and "edge swap"[HDDM93]. We define a limited set of transformations $\{T_i^k\}, (i = 1, \ldots, t^k)$ to refine or decimate a k-simplex mesh. We further constrain those basic transformations to be both *complete* and *independent.* The completeness property relates to the ability to transform any simplex mesh onto another simplex mesh, possibly of different topology, with a finite number of basic operations. A set of transformations are independent if a basic transformation cannot be obtained by a combination of other basic transformations. Among those operations, we distinguish the "Eulerian" transformations from the "Global" transformations. The former ones leave the topology of a mesh unchanged and hence are used for refining or decimating a mesh. The latter ones change either the topology or the connectivity of a mesh and correspond to high level operations.

We are able to define four complete and independent basic transformations $\{T_1^k, T_2^k, T_3^k, T_4^k\}$ on a k-simplex mesh ($k = 1, 2$). The first (resp. last) two, are Eulerian (resp. global) transformations and are inverse of each other. Figure 5 shows the four transformations operating on a contour. $\{T_1^k, T_2^k\}$ consist in inserting or removing a vertex, whereas $\{T_3^k, T_4^k\}$ consist in connecting two non adjacent edges. Depending whether the edges belong to the same contour, $\{T_3^k, T_4^k\}$ result in either splitting or merging contours.



a) b)

Figure 5: (a) The two Eulerian transformation T_1^1, T_2^1 ; (b) The two transformations T_3^1 and T_4^1 split or merge contours. We have $T_3^1 = T_4^1$

On a 2-simplex mesh, the two Eulerian transformations consist in inserting or deleting edges whereas the global operations correspond to either connecting two faces or cutting along a contour (see Figure 6). When the contour is dividing, the cutting operation results in splitting the mesh into two parts. Otherwise, it results in decreasing the genus of the mesh.

2.2 Geometry

The previous section described the properties of a simplex mesh considered as a k-connected graph independently of the space in which it is embedded. The geometry of a k-manifold as well as the geometry of a k-simplex mesh is highly dependent, however, on the dimension of the embedding space. In this section, we will only consider the representation of surfaces with k-simplex meshes of $\mathbb{R}^{k+1}$ and more precisely of planar contours and three-dimensional surfaces represented with 1-simplex meshes and 2-simplex meshes. Furthermore, we will

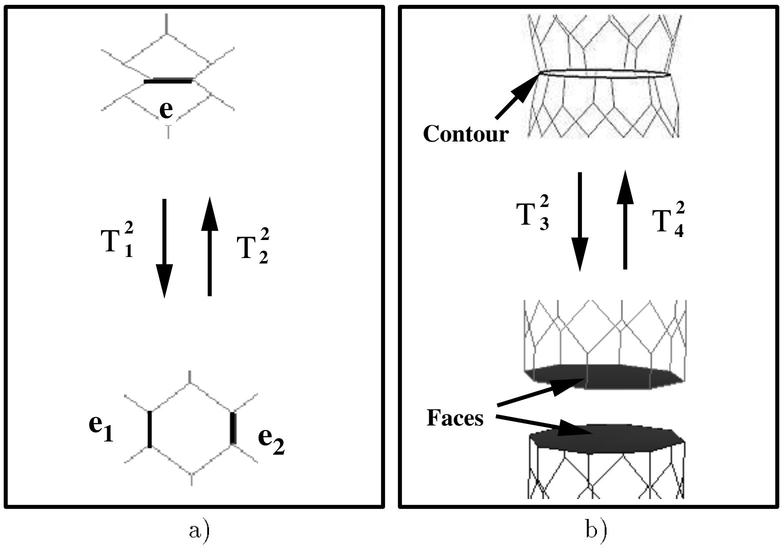


Figure 6: (a) The two Eulerian transformations T_1^2, T_2^2 ; (b) The two global transformations T_3^2 and T_4^2 that change the mesh topology or connectivity.

only consider *non-degenerate* meshes for which the k-neighbors of each vertex are linearly independent, i.e. their matrix coordinates is of rank k. This does not restrict in practice the generality of representation.

We introduce the notion of *Simplex Angle* on a simplex mesh, that generalizes in many ways, the angle used in planar geometry. In the following sections, we will describe how this angle is linked to the mean curvature on a simplex mesh and how the shape of a mesh may be uniquely determined. Finally, we will compare the geometry and representations of triangulations with those of simplex meshes.

2.2.1 Simplex Angle

Let $\mathcal{M}$ be an oriented simplex mesh of $\mathbb{R}^3$. Let $P_i \in \mathbb{R}^3$ be a vertex of a 2-simplex mesh, and $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ its three neighbors. The three neighboring points define a plane and its normal vector $\overrightarrow{N_i}$:

$$\overrightarrow{N_i} = \frac{P_{N_1(i)} \wedge P_{N_2(i)} + P_{N_2(i)} \wedge P_{N_3(i)} + P_{N_3(i)} \wedge P_{N_1(i)}}{\|P_{N_1(i)} \wedge P_{N_2(i)} + P_{N_2(i)} \wedge P_{N_3(i)} + P_{N_3(i)} \wedge P_{N_1(i)}\|}$$

Definition 1 (Simplex Angle on a simplex Mesh) *Let S_1 be the circumscribed circle at the three neighboring vertices $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$. This circle is of radius r_i and of* *center C_i. Let S_2 be the circumscribed sphere at the four vertices $(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ and let R_i and center O_i be ithe radius and center of S_2 (see Figure 7.a). The simplex angle at vertex P_i, $\varphi_i = \angle(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ is defined as:*

$$\begin{array}{c} \varphi_i \in [-\pi, \pi] : \\ \sin(\varphi_i) = \frac{r_i}{R_i} * sign(\overrightarrow{P_i P_{N_1(i)}} \cdot \overrightarrow{N_i}) \\ \cos(\varphi_i) = \frac{\|O_i C_i\|}{R_i} * sign(\overrightarrow{O_i C_i} \cdot \overrightarrow{N_i}) \end{array} \tag{7}$$

$\|O_i C_i\|$ represents the distance of the plane $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ from the center of the sphere O_i. Since we supposed the three neighbors of each vertex to be linearly independent, the circle is not degenerate ($r_i \neq 0$).

The simplex angle φ_i is independent of the position of $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$ on the circle S_1 and of P_i on a hemisphere of S_2. It is null when P_i is on the plane defined by its three neighbors. The relations in equation 7 are made explicit by drawing the projection of the sphere in the plane defined by (O_i, C_i, P_i) (see Figure 7). In this figure, the simplex angle φ_i appears as the angle, in the sense of planar geometry, between the segments that join P_i to the projection of the circle.

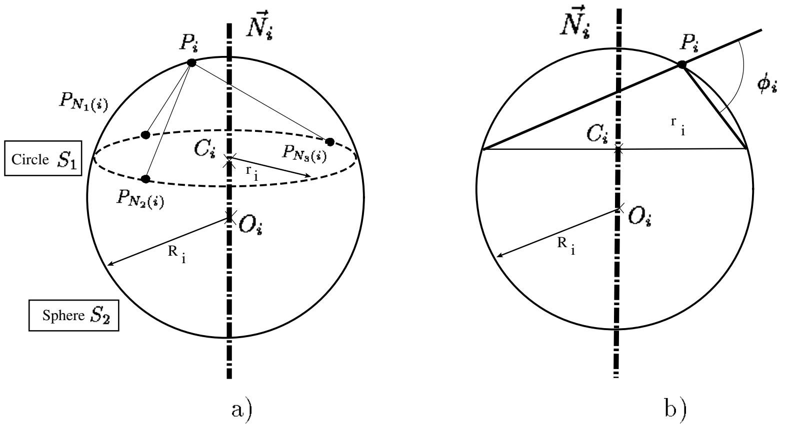


Figure 7: a) The local tetrahedron and its circumsphere S_2 of radius R_i. S_1 is the circle circumscribed to the three neighbors $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$of radius r_i; b) Cross section of a) through a plane defined by O_i, C_i and P_i. The simplex angle may be interpreted as a planar angle.

The definition of φ_i from equation 7 is not well suited for computation since the radius of the circumscribed sphere R_i may take very large values as P_i is close to the plane defined by its three neighbors. Instead, it can be easily computed when defined through an inversion of center P_i. Since an inversion through P_i transforms any sphere into a plane, the circumscribed sphere is transformed into a plane passing through the three transformed neighbors (see Figure 8). If we write $D_i^\star$ as the distance of P_i to the transformed plane and $\overrightarrow{u_i}$ the unit normal of that plane then we have:

$$D_i^\star = \frac{1}{2R_i} \qquad O_i = P_i + \frac{\overrightarrow{u_i}}{2D_i^\star} \tag{8}$$

Equation 7 can now be written as:

$$\begin{aligned} \sin(\varphi_i) &= 2r_i D_i^\star sign(P_i \overrightarrow{P_{N_1(i)}} \cdot \overrightarrow{N_i}) \\ \cos(\varphi_i) &= \|2D_i^\star \overrightarrow{P_iC_i} + \overrightarrow{u_i}\| sign(\overrightarrow{O_iC_i} \cdot \overrightarrow{N_i}) \end{aligned} \tag{9}$$

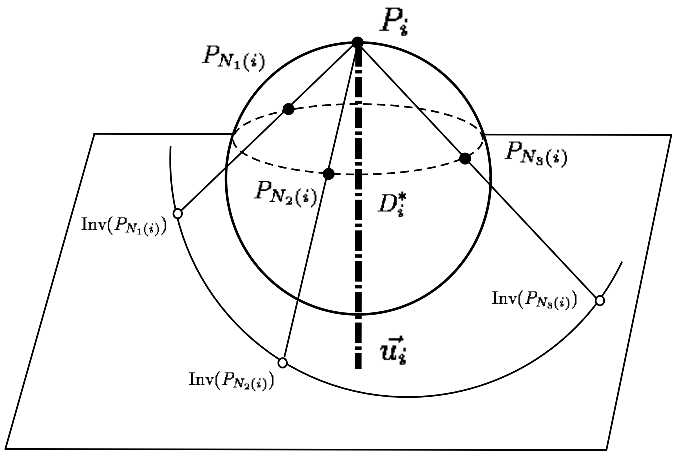


Figure 8: The inversion of center P_i transforms a sphere onto a plane. The distance of P_i from the transformed plane is $D_i^\star$ and $\vec{u_i}$ is the normal vector of the transformed plane.

The simplex angle generalizes several properties of the angle in planar geometry:

- **Stability though inversion** The simplex angle at a vertex P_i is preserved by any inversion of center P_i.
- **Criterion for circumsphere** Given four points (A, B, C, D) of $\mathbb{R}^3$, the condition for P to lie on the same circumsphere is:

$$\begin{gathered} \angle(P, B, C, D) = \angle(A, B, C, D) \\ or \\ \angle(P, B, C, D) = \pi - \angle(A, B, C, D) \end{gathered}$$

- **Half Angle Relation** The simplex angle at a vertex is half the angle at the center of the circumscribed sphere:

$$\angle(A, B, C, D) = \frac{\angle(O, B, C, D)}{2}$$

where O is the center of the circumsphere at (A, B, C, D)

All geometric notions defined previously for 2-simplex meshes can be easily extended to any k-simplex of $\mathbb{R}^{k+1}$.

Property 3 *The simplex angle at a vertex of a* 1*-simplex mesh is the angle in terms of planar geometry, between adjacent edges.*

Proof In a 1-simplex mesh, R_i is the radius of circumscribed circle at the three vertices (P_i, P_{i+1}, P_{i-1}) and r_i is half the distance $||P_{i-1}P_{i+1}||$ (see Figure 7.b). The planar angle φ_i at P_i trivially verifies the two relations of equation 7 $\square$

2.2.2 Simplex Angle and Mean Curvature

A simplex mesh is a discrete structure which does not define any underlying continuous surface. The purpose of this section is to introduce the notion of "discrete" mean curvature on a simplex mesh and to compare it with the same notion on a surface.

Mean curvature on a surface is an extrinsic measure of the surface bending and can be best pictured as the "nose dive averaged over all directions" [Koe90, pp 147]. Mean curvature describes only one aspect the surface shape. A complementary shape information is the Gaussian curvature, measured through the surface geodesy. Both are related to the principal curvatures [Car76]:

Definition 2 (Mean curvature) *Let $\mathcal{S} : (u, v) \longmapsto P(u, v)$ be a three dimensional surface defined over an open set $\Omega \subset \mathbb{R}^2$ and $P(u_0, v_0)$ a point on $\mathcal{S}$. If we write k_1 and k_2 as the two principal curvatures at $P(u_0, v_0)$, the mean curvature H is defined as:*

$$H = \frac{k_1 + k_2}{2} \tag{10}$$

The mean curvature at a point of a surface may be derived by approximating locally the surface with a sphere. We enunciate a property valid for surfaces of all dimensions, which characterizes the mean curvature through the best fit sphere or "minimal sphere". This is an extrinsic characteristic since it refers to the embedding space and not the geodesy of the surface.

Definition 3 (Minimal Sphere) *Let P be a point on a C^2 regular surface of $\mathbb{R}^{k+1}$, $\overrightarrow{N}$ its normal vector and H its mean curvature. The minimal sphere at P is the sphere of radius $1/H$ that passes through P with the a normal vector equal to $\overrightarrow{N}$.*

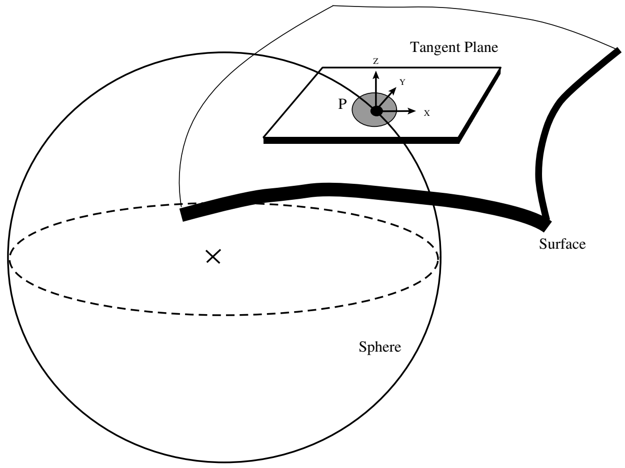


Figure 9: We evaluate the distance between a sphere and a surface through their distance in the normal direction. The minimal sphere is among all spheres passing through P, the one that minimizes its distance to the tangent plane

Property 4 (Minimal Sphere) *Among all spheres passing through P, the minimal sphere locally minimizes its square distance to the surface along the normal direction.*

We measure the distance between a sphere and a surface as the sum over a circular neighborhood, of the distance between pair of points that have the same orthogonal projection in the tangent plane (see figure 9). In Appendix B, a proof of this property in the three dimensional case is presented.

In general the minimal sphere has a contact of order two and is therefore *not osculatory*. A three dimensional surface is best approximated at the second order by the osculating quadric $z = \frac{1}{2}(k_1 x^2 + k_2 y^2)$ [Car76]. The minimal sphere, however, is the *sphere* that realizes the best fit of the surface at P. Only for planar curves is the minimal circle also osculary.

We now introduce the notion of mean curvature on a simplex mesh from the previous geometric definitions.

Definition 4 (Mean Curvature on a simplex mesh) *Let P_i be a vertex of a k-simplex mesh $\mathcal{M}$. If φ_i and r_i are respectively its simplex angle and the radius of the circumscribed circle, then we define the mean curvature H_i at P_i, as:*

$$H_i = \frac{\sin(\varphi_i)}{r_i} \tag{11}$$

Property 5 *The absolute value of the mean curvature is the inverse of the radius of the circumscribed sphere at P_i.*

$$|H_i| = \frac{1}{R_i} \tag{12}$$

However, we justify the choice of the word "mean curvature" with the fact that it shares several properties with the mean curvature defined on a surface:

- The mean curvature is *extrinsic* to the mesh because its sign depends on the surface orientation.
- A mesh on a sphere has constant mean curvature equal to the inverse of the sphere radius.
- On a simplex mesh, the sphere of radius $1/H_i$ is the circumscribed sphere which best fits the mesh vertices around P_i. Therefore, our definition of mean curvature on a simplex mesh also verify the minimal sphere property.

2.2.3 Metric Parameters

Mean curvature does not necessary characterize the intuitive notion of bending. For instance, a point lying on a minimal surface has the same mean curvature as a point lying on a plane. It is difficult to represent a surface shape with intrinsic parameters, without parameterization. For instance, two surfaces with similar mean and Gaussian curvatures are not necessarily isometric[Koe90].

Instead, we introduce k additional parameters at each vertex called *metric parameters* $\epsilon_i = \{\epsilon_{1i}, \ldots, \epsilon_{ki}\}$ that describe how a vertex is located with respect to its $k+1$ neighbors. If we consider the orthogonal projection F_i of P_i on the plane defined by its $k+1$ neighbors, then F_i may be written as a weighted average of the $k+1$ neighboring points:

$$F_i = \epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + \ldots + \epsilon_{ki} P_{N_k(i)} + \epsilon_{(k+1)i} P_{N_{k+1}(i)} \tag{13}$$

$$\epsilon_{1i} + \epsilon_{2i} + \ldots + \epsilon_{ki} + \epsilon_{(k+1)i} = 1 \tag{14}$$

Figure 10 shows how metric parameters are defined on a 1 and 2-simplex mesh.

The position of a vertex P_i is completely determined from the position of its neighbors $(P_{N_1(i)}, \ldots, P_{N_{k+1}(i)})$ and the knowledge of $(\epsilon_{1i}, \ldots, \epsilon_{ki}, \varphi_i)$. For a 2-simplex mesh, the relation is:

$$P_i = \underbrace{\epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + \epsilon_{3i} P_{N_3(i)}}_{F_i} + L(r_i, d_i, \varphi_i)\vec{N}_i \tag{15}$$

where

- r_i is the radius of the circle circumscribing $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$.

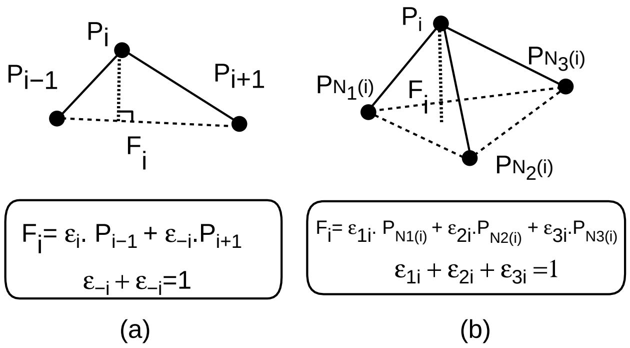


Figure 10: Definition of metric parameters for a contour or simplex mesh

- d_i is the distance between F_i and the center of the circle C_i.
- $L(r_i, d_i, \varphi_i)$ is a function described in Appendix A.
- $\vec{N_i}$ is the normal vector at plane $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$.

The metric parameters control how F_i is located in the plane $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$, whereas the simplex angle controls the height of P_i above those plane. $(\epsilon_{1i}, \epsilon_{2i})$ have an influence on the overall shape of the mesh, however there is no simple relation between the value of the metric parameters and, for instance, local Gaussian curvature.

Given a simplex model $\mathcal{M}$ consisting of n vertices, the shape of $\mathcal{M}$ is uniquely described up to a rigid-motion and scale factor, by the set of $3n$ values $\{(\epsilon_{1i}, \epsilon_{2i}, \varphi_i)\}$. This amounts to stating that the set of $3n$ equations derived by considering Equation (15) at each vertex, are independent. We use those $3n$ values to control the shape of a simplex mesh.

Unlike the discrete mean curvature parameter, the simplex angle at a vertex is not an intrinsic measure. However, it is important to note that we can describe the shape of simplex meshes in terms simplex angle and not in terms of discrete mean curvature. Therefore, the shape functionals that we will describe in section 3.2 will be posed in terms of simplex angles but will attempt to control the mean curvature.

For performing efficient object recognition, it is important to restrict the amount of shape encoding parameters. In [DHI93], we have developed an efficient recognition system based on "semi-regular" simplex meshes. Those models were characterized by equal metric parameters $\epsilon_{1i} = \epsilon_{2i} = \epsilon_{3i} = 1/3$ at each vertex in order to encode the object shape with only the n simplex angle values. In this paper, our goal is to demonstrate the generality and efficiency of the simplex mesh representation and we will not constraint the metric parameters to be equal.

2.2.4 Triangulations and Simplex Meshes

Triangulation and simplex meshes are dual topologically, but are not equivalent geometrically. However, both representations are complementary in terms of curvature estimation. Triangulations unambiguously define a normal vector at each triangle whereas simplex meshes defines a normal vector at each vertex. Similarly, a discrete Gaussian curvature on a triangulation is easily computed through the *spherical excess* or *solid angle*, whereas the mean curvature is estimated through the simplex angle (see figure 11). It is indeed, a well known geometric result[Hil44] that we can estimate the Gaussian curvature around a vertex of a triangulation as the ratio of the solid angle with the sum of the triangles area.

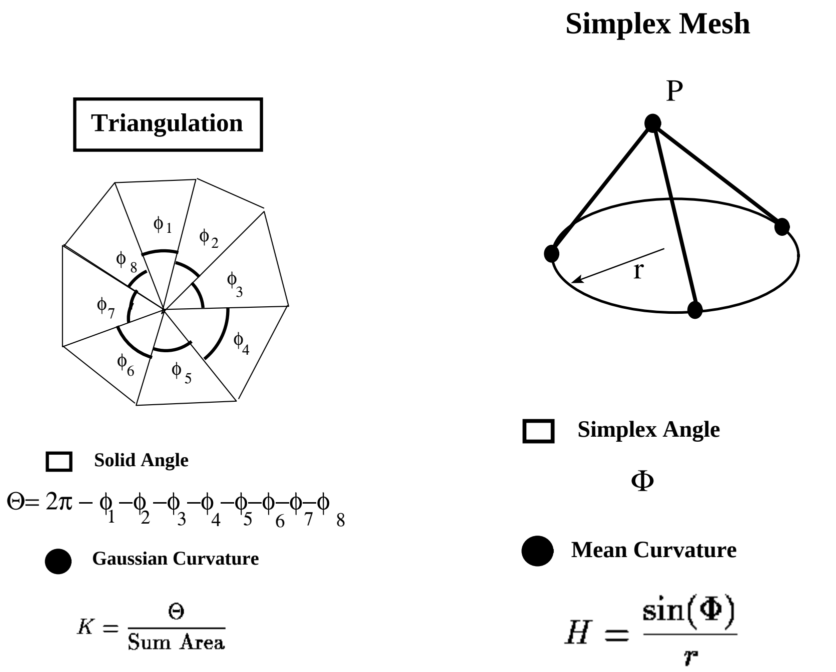


Figure 11: Comparison of the triangulation and simplex meshes representation in terms of Gaussian and mean curvature.

On the other hand, in a triangulation, we can only estimate a measure of the mean curvature over the triangles surrounding a vertex, as the sum of the length of each edges multiplied by their associated dihedral angles[Bor93]. Likewise, in a simplex mesh, we can evaluate a measure of the Gaussian curvature over a face as the signed area of the spherical polygon described by the extremities the normal vectors computed for each vertex of the face (see figure12).

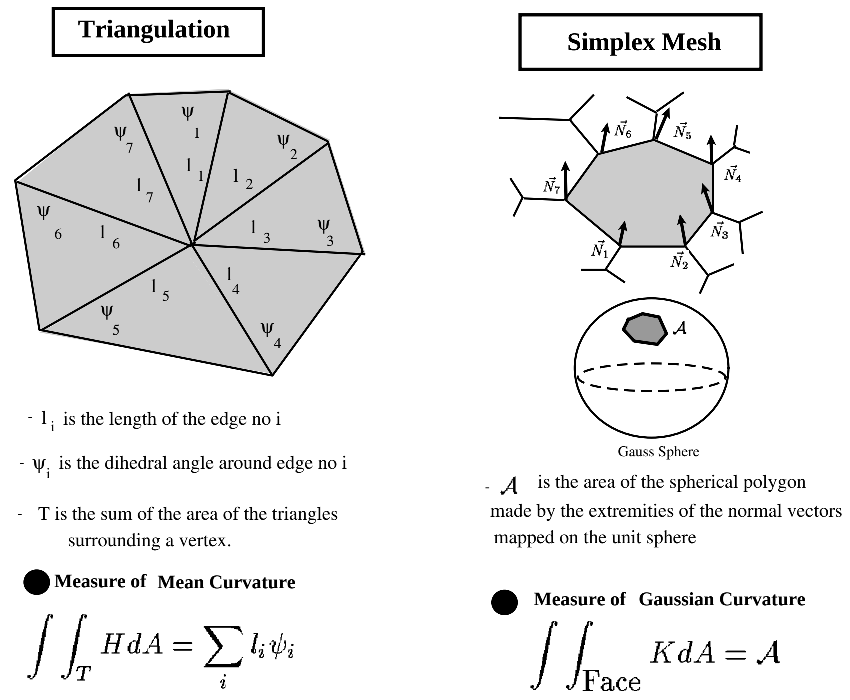


Figure 12: Estimation of respectively mean curvature and Gaussian curvature measures (resp.) on a triangulation and simplex mesh.

Hence, the natural curvature index in a triangulation is the Gaussian curvature. A natural representation of convex polyhedra is the Extended Gaussian Image[Hor84][Ike81] which is based on the discrete Gaussian curvature at each vertex. This representation is especially well-suited for efficiently performing object recognition and determining the attitude of an object in space. However, the main limitation of EGI lies in its ambiguous representation of non convex polyhedra, and in particular of triangulations.

On the other hand, the natural curvature index in a simplex mesh is the mean curvature. In section 2.2.3, we have seen that the metric parameters and the simplex angles at each vertex establish a unique representation of a mesh, up to an isometry and scale. In a SM, we can express the position of a vertex in a simplex mesh as a function of the position of its neighboring vertices and of the shape parameters (see equation 15). It is not possible in a triangulation to establish a similar relation based on discrete Gaussian curvature and the position of the neighboring vertices. This fundamental property explains why simplex meshes are better suited than triangulations for performing deformations and smoothing. Furthermore, we can unambiguously encode the shape of an object of complex topology with the the Simplex Angle Image[DHI93].

3 Smoothness and Shape Control

3.1 Equations of Motion

We propose a modeling scheme based on deformable and adaptive meshes. The dynamics of each vertex is given by a Newtonian law of motion:

$$m\frac{d^2 P_i}{dt^2} = -\gamma \frac{dP_i}{dt} + \vec{F_{int}} + \vec{F_{ext}} \tag{16}$$

where m is the mass unit of a vertex and γ is the damping factor. $\vec{F_{int}}$ constrains either shape or smoothness whereas $\vec{F_{ext}}$ corresponds to some external constraints expressed in terms of distance from some three-dimensional data.

Time is discretized as $t_i = t_0 + i\Delta t$ and Equation (16) is integrated over time using finite differences. We currently use Euler's method with centered differences. If P_i^t is the position of vertex i at time t then the discretized law of motion is :

$$P_i^{t+1} = (1-\gamma).(P_i^t - P_i^{t-1}) + \vec{F_{int}} + \vec{F_{ext}} \tag{17}$$

$\vec{F_{int}}$ and $\vec{F_{ext}}$ are computed at time t.

3.2 Internal Forces

Internal forces determine the response of a physically-based model to external constraints. The response of elastic bodies may be derived from a minimization of an elastic energy, proportional to their deflection from a rest shape. Linear elastic energies corresponding to Tikhonov stabilizing functionals[TA87] have been extensively used in computer vision[WW92] [DHI91b][McI93][CCA92]. In its generalized formulation, the energy is composed of a stretching term and a bending term. Some weights can be adjusted to account for surface discontinuities or normal discontinuities. Linear elasticity leads to numerically efficient algorithms but it does not guaranty the fairness of resulting models since they are not expressed in terms of intrinsic parameters. Moreover, they consistently flatten high curvature parts[Del94]. On the other hand, elastic energies expressed in terms of intrinsic parameters such as square curvature[Bla87] or square derivative of curvature[MS92] are physically meaningful but are numerically difficult to implement and finally they do not necessary accept circles[Del94] or spheres as optimum shapes.

Our framework does not derive $\vec{F_{int}}$ from the minimization of some global elastic energy. Instead we associate with each vertex P_i a local energy $\mathcal{S}_i$ that characterizes the shape of the local tetrahedron $(P_i, P_{N1(i)}, P_{N2(i)}, P_{N3(i)})$. $\vec{F_{int}}$ is taken as the gradient of $\mathcal{S}_i$ and brings the tetrahedron into its state of lowest energy. Therefore each vertex can be seen as an independent particle that interacts with its three neighbors and their surrounding vertices. Szeliski[ST92] used oriented particles to interpolate 3D data without a priori knowledge of connectivity or topology. The important difference between simplex mesh and particles system is that simplex meshes constrain vertex connectivity to three which restricts the generality of representation but leads to smoother shapes and more efficient computation. In our framework, both mesh structure and topology can change dynamically, but the connectivity between vertices remains equal to three.

3.3 Elastic Force $\vec{F_{int}}$

The expression of $\vec{F_{int}}$ is a generalization of intrinsic functionals defined for planar curves defined in[DHI91a][Del94]. We have defined some shape and smoothness functionals on simplex meshes having the following properties:

Intrinsicness They are expressed in terms of mean curvature, and therefore fairness of resulting shape is guaranteed. Furthermore, spheres, cylinders, planes, are among the optimum shapes: $H = cste$.

Invariance with translation, rotation and scale

Dependence with inner-scale . The size of the neighborhood around a vertex controls the scale at which smoothness is measured.

Local support Different types of constraint, different size of neighborhood may be applied on different parts of a mesh.

Let $\mathcal{M}$ be a simplex mesh model with n vertices $\{P_i(i = 1, \ldots, n)\}$. We suppose that two metric parameters ϵ_{1i} and ϵ_{2i} have been assigned at each vertex such that:

$$\begin{array}{ccc} & \epsilon_{1i} + \epsilon_{2i} + \epsilon_{3i} = 1 & \\ 0 < \epsilon_{1i} < 1 & 0 < \epsilon_{2i} < 1 & 0 < \epsilon_{3i} < 1 \end{array} \tag{18}$$

We only consider strictly positive metric parameters to ensure a stable behavior during the deformation. This amounts to constraining the foot F of P_i to lie inside the triangle $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$. It does not, in practice, greatly restrict the variety of shapes.

With each vertex P_i we associate a simplex angle $\varphi_i^\star$ for which the local tetrahedron $(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ is in its rest position. $\varphi_i^\star$ corresponds to the natural state of the local tetrahedron or its state of minimum energy. If we write $P_i^\star$ as the position of vertex P_i if the simplex angle was $\varphi_i^\star$ then the energy $\mathcal{S}_i$ of the local tetrahedron is:

$$\mathcal{S}_i = \frac{\alpha_i}{2} P_i P_i^{\star 2} \qquad (0 \le \alpha_i \le 0.5) \tag{19}$$

$\vec{F_{int}}$ is the gradient of this energy and since $P_i^\star$ is independent of P_i:

$$\vec{F_{int}} = \alpha_i \ \overrightarrow{P_i P_i^\star} \qquad (0 \leq \alpha_i \leq 0.5) \tag{20}$$

This expression corresponds intuitively to the notion of elastic force since it states that $\vec{F_{int}}$ is proportional to the deflection from the rest shape. α_i has to be smaller than 0.5 to guarantee stability. Using equation(15) to express $P_i^\star$, we get:

$$\vec{F_{int}} = \alpha_i.(\epsilon_{1i} \ \overrightarrow{P_i P_{N_1(i)}} + \epsilon_{2i} \ \overrightarrow{P_i P_{N_2(i)}} + \epsilon_{3i} \ \overrightarrow{P_i P_{N_3(i)}} + L(r_i, d_i, \varphi^\star)\vec{N_i}) \tag{21}$$

The choice of $\varphi_i^\star$ determines the constraint to be enforced:

Normal Discontinuity We set $\varphi_i^\star = \varphi_i$. The surface can freely bend around vertex P_i.

Surface Orientation Continuity constraint We have simply $\varphi_i^\star = 0$. Hence, the internal force writes as: $\vec{F_{int}} = \alpha_i(\epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + \epsilon_{3i} P_{N_3(i)} - P_i)$. When metric parameters are equal, we have $\vec{F_{int}} = \frac{\alpha_i}{3}(P_{N_1(i)} + P_{N_2(i)} + P_{N_3(i)} - 3P_i)$ which can be written as $\vec{F_{int}} = \frac{\alpha_i}{3}\Delta_{\mathcal{M}} P$ where $\Delta_{\mathcal{M}}$ is the discrete Laplacian operator defined on the mesh. The mesh structure can be seen as a surface parameterization though we cannot define properly any notion of derivatives.

Mean Curvature Continuity Constraint $\varphi_i^\star$ is defined by :

$$\varphi_i^\star = \arcsin\left(\sum_{j \in N^r(P_i)} \frac{e_{ij} * r_i * \sin(\varphi_j)}{r_j}\right) \tag{22}$$

$$\sum_{j \in N^r(P_i)} e_{ij} = 1 \qquad 0 < e_{ij} < 1$$

where $N^r(P_i)$ is the neighborhood of size r around P_i. Equation (22) states that the natural curvature at P_i is the weighted average of the neighborhood curvatures:

$$k_i^\star = \sum_{j \in N^r(P_i)} e_{ij} * k_j$$

r corresponds intuitively to the notion of rigidity. If a highly rigid model is pinched at vertex P_i, the whole surface would bend slightly since the curvature is averaged over a large number of vertices and would quickly recover its rest shape. For a more flexible surface with a small value of r, a bump would only appear around P_i and the convergence toward its rest shape would be slower. Rigidity is a dynamic characteristic of a vertex and is not related to the notion of fairness. If we write equation (3.3) with $r = 1$ and $e_{i1} = e_{i2} = e_{i3} = 1/3$ we get: $k_{N_1(i)} + k_{N_2(i)} + k_{N_3(i)} - 3k_i^\star = 0$ or $\Delta_{\mathcal{M}} H = 0$.

Shape Constraint Given the constant φ_i^0 by setting $\varphi_i^\star = \varphi_i^0$ we constrain the simplex angle at P_i to φ_i^0. In order to constrain mean curvature k_i^0, the relation is $\varphi_i^\star = \arcsin(r_i * k_i^0)$. In both cases, the surface normal continuity at P_i may not be guaranteed depending on the constraints of neighboring vertices. The resulting action of shape constraint functionals may be seen in figure 13.

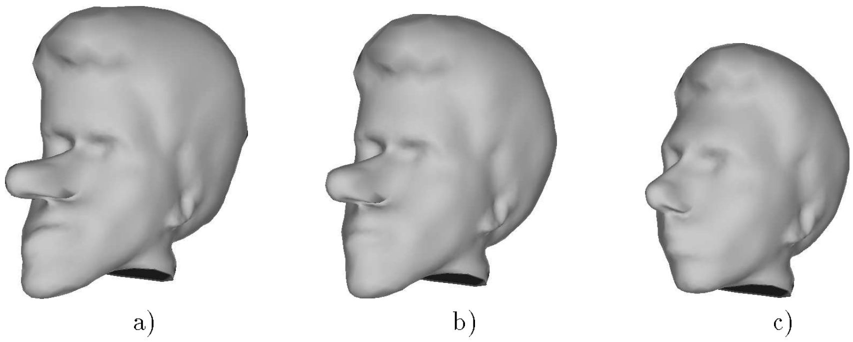


Figure 13: Shape Constraint applied on a face model; a) The initial deformed model; b) Intermediate shape; (c) Recovering its initial shape

Those functionals achieve smoothness constraints by controlling the mean curvature along the mesh. The metric parametric parameters on the contrary are left unchanged in order to guaranty a stable behavior during the deformation.

3.4 Internal Constraint on a Contour

A contour defined on a simplex mesh, is deformable structure that is handled independently of the mesh on which it is embedded. A vertex belonging to a contour is submitted to an internal force $\vec{F_i}nt$ that enforces shape or smoothness constraints of the contour rather than the surface model.

A contour defined on a 2-simplex mesh is a three dimensional curve, and therefore neither expressions defined in the previous section nor those defined in [Del94] do not hold (a three-dimensional contour is not a manifold of codimension one). Several expressions of fairing energies has been proposed ranging from minimal curvature curves[Meh74] or minimum variation curves[MS92] to splines under tension[KWT88][Gue93]. The former expressions are intrinsically formulated but lead to computationally difficult schemes. The spline under tension smoothness measure, on the contrary, have the disadvantage of systematically shrinking the curves toward their center of curvature. Instead, we propose a set of shape and smoothness functionals that do not exhibit any shrinking effect and that are expressed in terms of intrinsic parameters. Those functionals are greatly resemblant to those of equation21 or those found in [Del94].

At each vertex of a contour $\mathcal{C} = (J(\mathcal{C}))$, we define a tangent vector $\vec{T}_{J(i)}$, a normal vector $\vec{Q}_{J(i)}$ and a binormal vector $\vec{B}_{J(i)}$:

$$\vec{T}_{J(i)} = \frac{\overrightarrow{P_{J(i-1)}P_{J(i+1)}}}{\| \overrightarrow{P_{J(i-1)}P_{J(i+1)}} \|}$$

$$\vec{B}_{J(i)} = \frac{\overrightarrow{P_{J(i-1)}P_{J(i)}} \times \overrightarrow{P_{J(i)}P_{J(i+1)}}}{\| \overrightarrow{P_{J(i-1)}P_{J(i)}} \times \overrightarrow{P_{J(i)}P_{J(i+1)}} \|}$$

$$\vec{Q}_{J(i)} = \vec{B}_{J(i)} \times \vec{T}_{J(i)}$$

Furthermore, similarly to equation 15, we express the position of $P_{J(i)}$ as a function of its four neighbors and as function of the triplet $\{\epsilon_{J(i)}, \delta_{J(i)}, \gamma_{J(i)}\}$:

$$P_{J(i)} = \epsilon_{J(i)} P_{J(i-1)} + (1-\epsilon_{J(i)}) P_{J(i+1)} + \lambda_{J(i)} \left(\cos(\gamma_{J(i)}) \vec{R}_{J(i)} + \sin(\gamma_{J(i)}) \vec{T}_{J(i)} \times \vec{R}_{J(i)} \right) \tag{23}$$

where :

- $\epsilon_{J(i)}$ is a metric parameter that characterizes the barycentric coordinate of the orthogonal projection of $P_{J(i)}$ onto the segment $[P_{J(i-1)}P_{J(i+1)}]$: $0 \leq \epsilon_{J(i)} \leq 1$.
- $\delta_{J(i)}$ is an angle representing the discrete curvature at $P_{J(i)}$.
- $\gamma_{J(i)}$ is the angle that the normal vector $\vec{Q}_{J(i)}$ makes with $\vec{R}_{J(i)}$. This angle is somewhat related to the discrete torsion at $P_{J(i)}$.
- $\vec{R}_{J(i)} = \frac{\vec{T}_{L(i)} \times (\overrightarrow{P_{L(i-2)}P_{L(i-1)}} \times \overrightarrow{P_{L(i+1)}P_{L(i+2)}})}{\|\vec{T}_{L(i)} \times (\overrightarrow{P_{L(i-2)}P_{L(i-1)}} \times \overrightarrow{P_{L(i+1)}P_{L(i+2)}})\|}$. $\vec{R}_{J(i)}$ and $\vec{T}_{J(i)} \times \vec{R}_{J(i)}$ constitute an orthogonal frame basis of the normal plane, orthogonal to the tangent vector $\vec{T}_{J(i)}$.
- $\lambda_{J(i)} = L(\frac{\|\overrightarrow{P_{J(i-1)}P_{J(i+1)}}\|}{2}, (\epsilon_{J(i)} - \frac{1}{2})\| \overrightarrow{P_{J(i-1)}P_{J(i+1)}} \|, \delta_{J(i)})$. $L()$ is the function defined at section 2.2.3 and described in Appendix A.

When $P_{J(i)}$ is equidistant from its two neighbors, then $\epsilon_{J(i)} = 1/2$ and equation 23 takes the simpler form:

$$P_{J(i)} = \frac{P_{J(i-1)} + P_{J(i+1)}}{2} + \tan(\delta_{J(i)}) \frac{\| \overrightarrow{P_{J(i-1)}P_{J(i+1)}} \|}{2} \left(\cos(\gamma_{J(i)}) \vec{R}_{J(i)} + \sin(\gamma_{J(i)}) \vec{T}_{J(i)} \times \vec{R}_{J(i)} \right)$$

The internal force applied at $P_{J(i)}$ is the gradient of the energy $\mathcal{S}_i = \frac{\alpha_{J(i)}}{2} \|P_{J(i)}P^\star_{J(i)}\|^2$:

$$\vec{F}_{int} = \alpha_{J(i)} \overrightarrow{P_{J(i)}P^\star_{J(i)}}$$

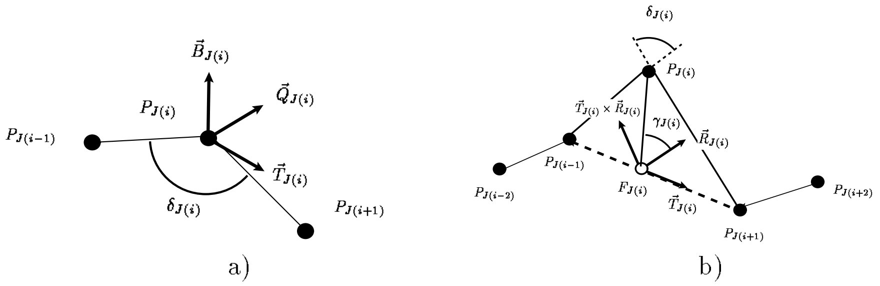


Figure 14: a) Definition of the tangent, normal, binormal vector at a vertex of a contour defined on a 2-simplex mesh; b) The position of a vertex with respect to its fours neighbors is determined with the three parameters $\{\epsilon_{J(i)}, \delta_{J(i)}, \gamma_{J(i)}\}$

$$\begin{aligned}
\vec{F}_{int} &= \alpha_{J(i)}(\epsilon_{J(i)}\ \overrightarrow{P_{J(i)}P_{J(i-1)}} + (1-\epsilon_{J(i)})\ \overrightarrow{P_{J(i)}P_{J(i+1)}} + \\
&\quad \lambda^{\star}_{J(i)}(\cos(\gamma^{\star}_{J(i)})\vec{R}_{J(i)} + \sin(\gamma^{\star}_{J(i)})\vec{T}_{J(i)} \times \vec{R}_{J(i)})) \\
\text{with} &\quad \lambda^{\star}_{J(i)} = L(\frac{\| \overrightarrow{P_{J(i-1)}P_{J(i+1)}} \|}{2}, (\epsilon_{J(i)} - \frac{1}{2})\| \overrightarrow{P_{J(i-1)}P_{J(i+1)}} \|, \delta^{\star}_{J(i)})
\end{aligned}$$

where $\delta^{\star}_{J(i)}$ and $\gamma^{\star}_{J(i)}$ are respectively related to the discrete curvature and torsion and characterize the set of constraints applied on a contour. $\epsilon_{J(i)}$ is the metric parameter and this parameter is fixed during the deformation. We define four different types of constraints on a contour vertex :

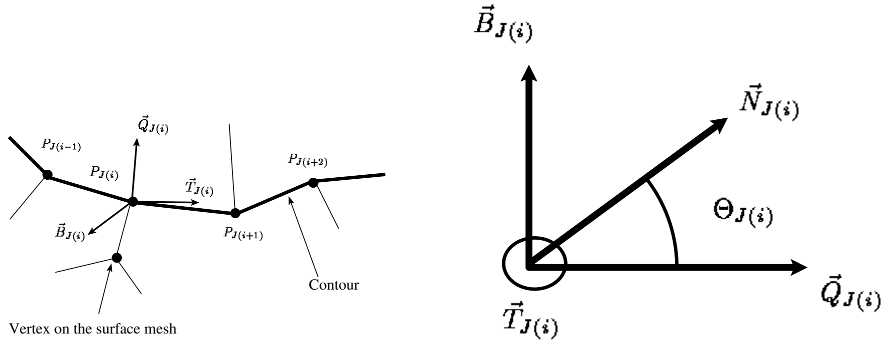
Normal Discontinuity . We set $\delta^{\star}_{J(i)} = \delta_{J(i)}$ and $\gamma^{\star}_{J(i)} = \gamma_{J(i)}$ where $\delta_{J(i)}$ and $\gamma_{J(i)}$ are the current values of curvature and torsion. This expression only constrains the distance between vertices but not their shape.

Normal Continuity Constraint . We set $\delta^{\star}_{J(i)} = 0$ and $\gamma^{\star}_{J(i)} = 0$. The force then simply writes as $\vec{F}_{int} = \alpha_{J(i)} \left(\epsilon_{J(i)}\ \overrightarrow{P_{J(i)}P_{J(i-1)}} + (1-\epsilon_{J(i)})\ \overrightarrow{P_{J(i)}P_{J(i+1)}} \right)$. This force is equivalent to the one obtained when minimizing the square sum of the first derivative as in [KWT88]. The curves that nullify this internal force are straight lines.

Curvature Continuity Constraint . We set $\gamma^{\star}_{J(i)} = 0$ and $\delta^{\star}_{J(i)} = \frac{\delta_{J(i-1)} + \delta_{J(i)} + \delta_{J(i+1)}}{3}$. We can generalize this expression by averaging over a larger extent r:

$$\delta^{\star}_{J(i)} = \frac{\sum_{-r \leq j \leq r} \delta_{J(j)}}{2r+1}$$

r is the rigidity coefficient that is similar to the simplex mesh case. Circled are the only closed curve for which the curvature continuity constraint is null everywhere.



a) b)

Figure 15: a) A contour vertex has one neighbor that does not belong to the contour. We can define at $P_{J(i)}$ a surface normal vector; b) The tangent at the vertex is orthogonal to the surface normal vector $N_{J(i)}$. $\Theta_{J(i)}$ is the angle between the surface normal vector and the contour normal vector.

Shape constraint We set $\delta^\star_{J(i)} = \delta^0_{J(i)}$ and $\gamma^\star_{J(i)} = \gamma^0_{J(i)}$ where $\delta^0_{J(i)}$ and $\gamma^0_{J(i)}$ are two constants.

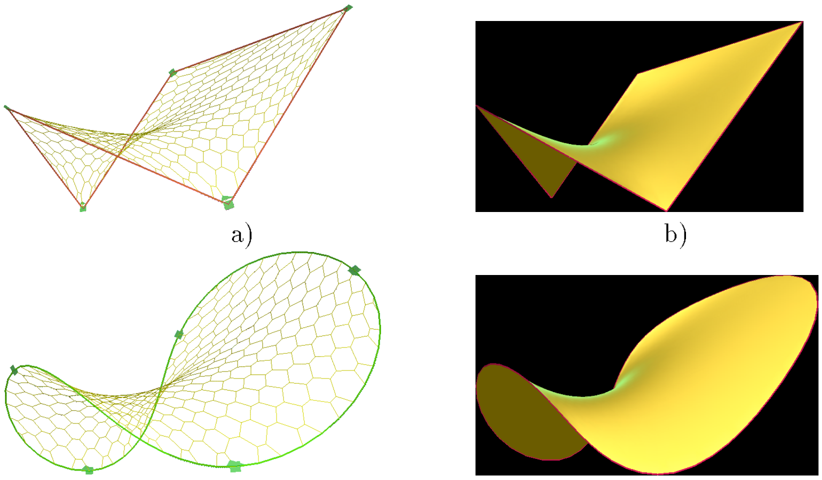
Boundary conditions describe how contours are embedded inside a mesh. The simplex mesh provides a simple way of setting boundary conditions by controlling simplex angles $\varphi_{J(i)}$ at each vertex of contour $\{P_{J(i)}\}$. The underlying assumption is that smoothness forces defined through Equations (22) apply to surface mesh located around each contour. We propose two types of surface-contour constraints:

Curvature Condition Mean Curvature or simplex angle is set by:

$$\varphi_{J(i)} = \arcsin(r_{J(i)} k^0_{J(i)}) \quad or \quad \varphi_{J(i)} = \varphi^0_{J(i)}$$

Tangent Condition The angle between the tangent plane and the contour normal as measured around the contour tangent vector, intuitively corresponds to the angle between the surface and contour. Each contour vertex $P_{J(i)}$ has two neighbors on the same contour and a third on the mesh. Since the normal vector of the surface model at $P_{J(i)}$,$\vec{N}_{J(i)}$, is the normal at the triangle made by its neighbors, $\vec{N}_{J(i)}$ is perpendicular to $\vec{T}_{J(i)}$ and is therefore inside the plane $(\vec{Q}_{J(i)}, \vec{B}_{J(i)})$ (see Figure 15). We will refer as $\Theta_{J(i)}$, the angle between $\vec{N}_{J(i)}$ and $\vec{Q}_{J(i)}$. In order to constrain the surface to make an angle $\Theta^0_{J(i)}$ with the contour at $P_{J(i)}$, we set $\varphi_{J(i)} = \Theta^0_{J(i)} - \Theta_{J(i)}$.

Figure 16 shows the difference between the normal continuity constraint (figure a) and b)) and the curvature continuity constraint (figure c) and d)). The former constraint is equivalent to performing a convolution with a smoothing kernel of order one. The latter



a) b)

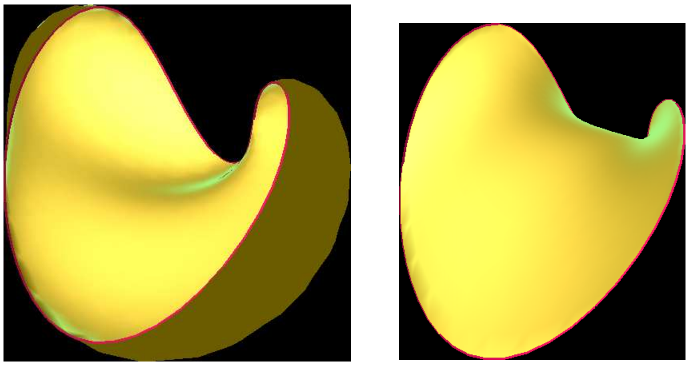
c) d)

Figure 16: a) A contour defined on a simplex mesh is submitted to normal continuity constraints. The contour interpolated the position of five contour vertices displayed with the black squares. The surface mesh has everywhere null mean curvature; b) Rendered display of a); c) Same as a) except that the contour is submitted to curvature continuity constraints. The continuity of normal is then verified at each five fixed vertices; d) Rendered display of c)

constraint, on the other hand, cannot be expressed in terms of linear smoothing but may be seen as smoothing the curvature value along the contour. The contour of figure16 c) and d) has constant (discrete) curvature. In both cases, we have applied a curvature constraint between the contour and the mesh characterized by zero mean curvature $\varphi^0_{J(i)} = 0$. The mesh has everywhere null mean curvature (minimal mesh).

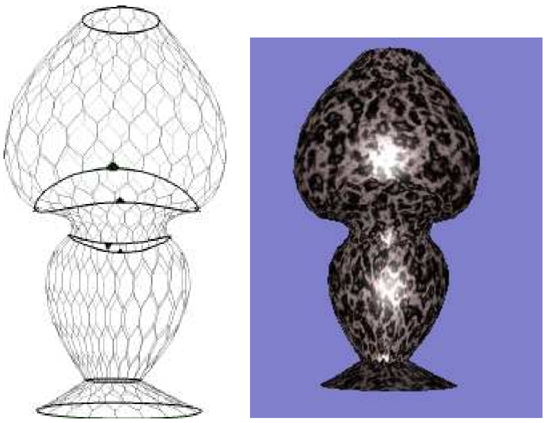
Figure 17 displays a simplex mesh with the same contour constraint as in figure 16 c) and d) but applies tangent constraint between the surface and the contour. Figure 17 a) corresponds to an angle of $\Theta^0_{J(i)} = -95°$ while figure 17 a) corresponds to an angle of $\Theta^0_{J(i)} = +145°$ between the contour normal and the surface normal. In both cases, we apply a curvature continuity constraint on the mesh vertices.

Finally, figure 18 shows a vase that was created from a cylinder by defining three intermediate contours with associated end conditions. Two contours are defined by the interpolation of four non-coplanar points. The mesh is constrained to make an angle of $\pi/4$ and $-\pi/4$ with those two contours. The remaining three contours have a zero mean curvature constraint with the surface mesh.



a) b)

Figure 17: a) We constraint the surface normal vector to make an angle of $\Theta^0_{\mathcal{J}(i)} = -95°$ with the contour normal vector; b) Same as a) with an angle of $\Theta^0_{\mathcal{J}(i)} = +145°$



a) b)

Figure 18: a)A vase created from a cylinder and five contours with appropriate end conditions; b) Texture display of a)

3.5 External force $\vec{F}_{ext}$

3.5.1 User Defined External Constraints

The external forces applied on a simplex mesh originate either from the user interface or from some three dimensional data. In the former case, we have defined two different methods for a user to interactively deform a simplex mesh. The first method consists in using an electronic mouse and clicking at place where the user want to drag a mesh. The pixel position defines a three dimensional line corresponding to the line of sight. The intersection of this line of sight with a plane parallel to the image plane and located at the middle of the scene is computed and a spring force between this intersection point and the closest vertex on a simplex mesh is exerted (see figure 19.a). The neighboring vertices are attracted too with an amplitude decreasing with their distance from the closest vertex. Both stiffness constant and the neighborhood extent can be adjusted to release natural deformations.

The second method makes use of virtual reality techniques, namely a virtual hand driven by two three dimensional trackers (see figure 20). Several modalities has been attached to the virtual hand. One of those is the possibility to generate a repulsive potential field centered at the position of the virtual hand and whose amplitude sharply decreases with distance. Thus, a model is deformed as the user moves the virtual hand closer to the mesh. This is not a very accurate method for shaping a geometric model. It is nonetheless useful for dragging a simplex mesh out of an undesirable position. For more details on the use of virtual reality techniques in conjunction with simplex meshes, see[DSCP94].

3.5.2 Data Constraints

In a surface reconstruction scheme based on deformable models, external forces constrain the displacement of a model toward some three dimensional data. Several expressions of the external force has been proposed but all of them rely either on the notion of distance between the geometric model and a data set or on the notion of potential field. For instance, Nastar[NA93] and Guéziec[Gue93] built three dimensional distance maps in order to find the closest point. Feldmar [FA94] defined a generalized distance from a surface by including the notion of position, normal orientation and principal curvatures in a single distance measure. On the other hand, McInerney *et al.*[McI93] builds a potential field from the gradient image and the output of the Canny-Deriche three-dimensional edge detector. Potential fields techniques has the advantage of only requiring a weak segmentation of the scene. However, oscillations of the model around the local minimum may arise if the potential field is not designed properly.

We use a distance method based on the iterative closest point[BM92]. At each iteration and for every vertex P_i, we search for the closest data point $M_{Cl(i)}$. The expression of the external force is then:

$$\vec{F}_{ext} = \beta_i G \left(\frac{\| \overrightarrow{P_i M}_{Cl(i)} \|}{D} \right) (\overrightarrow{P_i M}_{Cl(i)} \cdot \vec{N}_i) \vec{N}_i \tag{24}$$

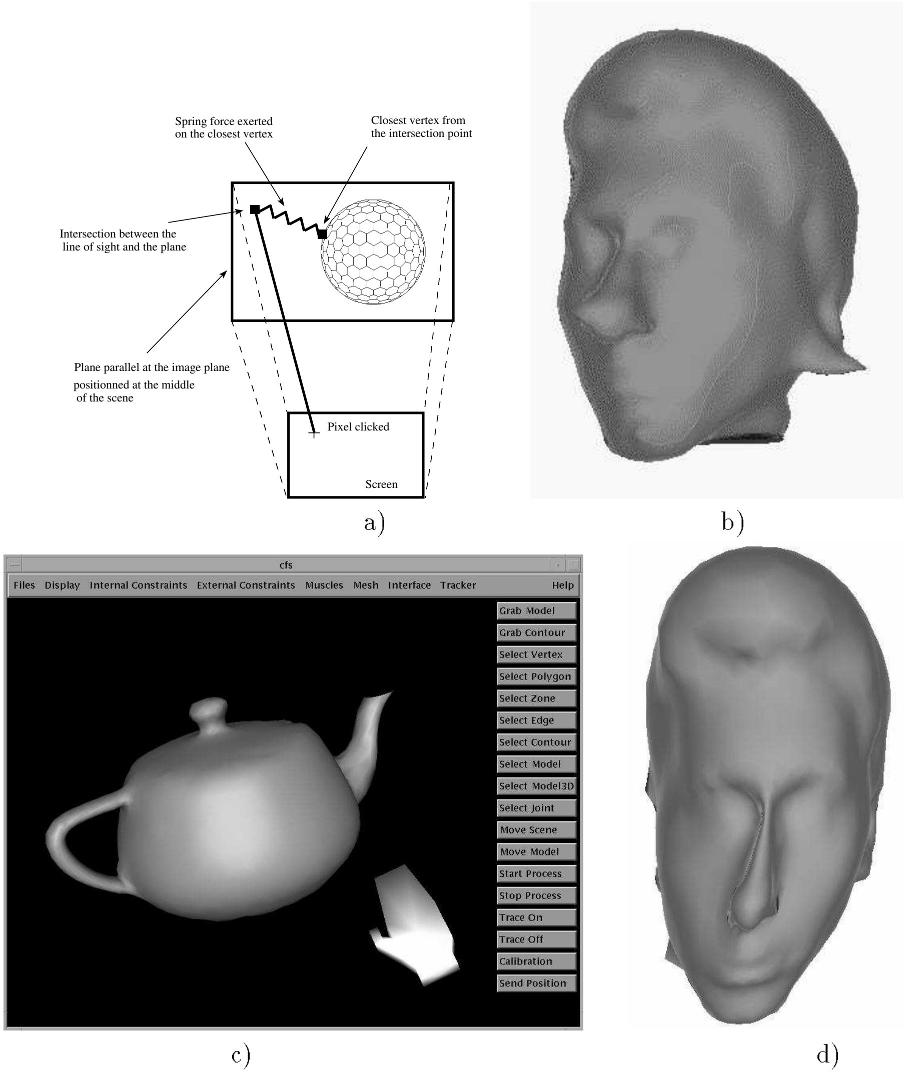


Figure 19: a) Description of the interactive system for deforming a simplex mesh from a mouse action; b) Example of the deformation of a face model around the ear resulting from a mouse action; c) The user-interface includes a virtual hand driven by two trackers located on the user glove; d) Example of a face deformed by a virtual hand on the right side.

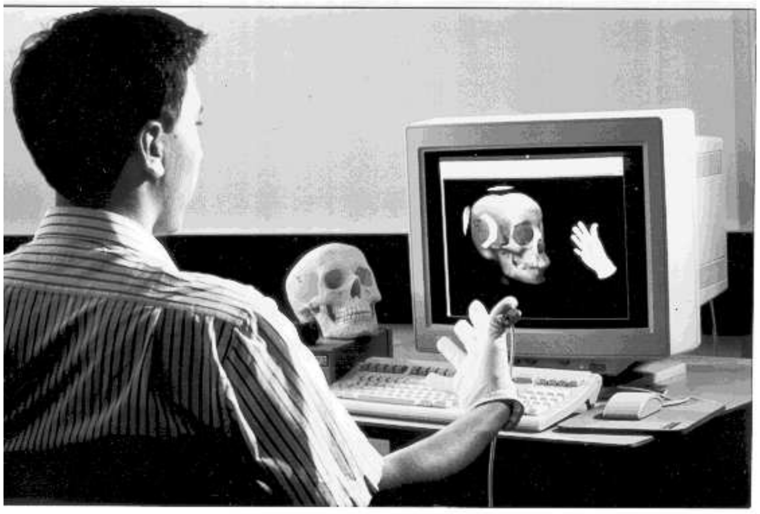


Figure 20: User manipulating simplex meshes with a virtual hand. Two Pohlemus trackers are attached on the glove in order to drive the virtual hand.

where $\vec{N_i}$ is the normal vector at P_i and $G(x)$ is the stiffness function (see figure 21). This approach is similar to the one used in[DHI91b]. Because we are using a discrete surface representation, moving vertices toward their closest data point would result in a very disorganized mesh structure especially in presence of sparse data. In order to solve the data sampling problem, we project the vector $\overrightarrow{P_i M_{Cl(i)}}$ onto the normal direction. Furthermore, having an external force in the normal direction entails smoother deformation of the mesh. The scalar D corresponds to the maximum distance at which a data point strongly attracts a vertex. When the closest computed data point is further than D, then the stiffness decreases sharply and therefore the point has little effect on the vertex position. A limited range of attraction is necessary in order to limit the influence of outliers or to model objects with narrow shapes. In practise, we compute D as a percentage of the radius of the sphere enclosing the whole data set. At the beginning of the deformation, we pick a large value of D (usually 20 percent of the radius) because the mesh may be initialized far from the data. We then decrease the value of D (down to 8 percent) in order to speed-up the search for the closest point.

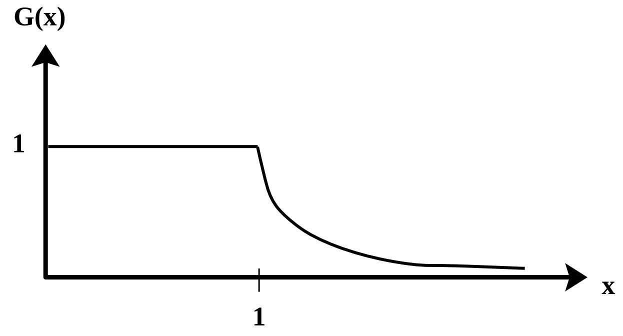


Figure 21: Stiffness Function $G(x)$.

The search of the closest point $M_{Cl(i)}$ is theoretically in $O(m^2)$ for an $m \times m$ range image. However, we were able to decrease the complexity to $O(1)$ for calibrated range data. When the image projection is known, we restrict the search of the closest point in the range image to a line image of the three dimensional line passing through P_i in the direction of $\vec{N_i}$. This line passes through the projection of P_i in the range image, and we only search a few pixels around the projection pixel. When the closest point is found, when search again in a 5x5 window around that pixel. A data point is considered as the closest data point only if it lies inside a cone of apex P_i, of direction $\vec{N_i}$ and of top angle equal to thirty degrees. This algorithm works well with most structured range data such as the NRCC[RC88], Cyberware[SW91] or the Ogis[SI87] sensors.

For unstructured data, we use a kd-tree structure for finding the closest data point inside a sphere of radius D. When D is small enough, the computation time is of the same order as for structured data. When the data set provides both position and normal information, we use a notion of distance between a vertex and a data point that combines their positional and orientation distance (same as Feldmar [FA94]).

Finally, when fitting isosurfaces defined on medical images, we find the closest point $M_{Cl(i)}$ as the intersection of the normal line defined at P_i with the isosurface. This is very efficiently done by keeping at each voxel of the tridimensional image the information of the isosurface. Therefore, the intersection is found by checking whether the voxels intersecting the normal line, contains a portion of isosurface. In practise, we only check the few voxels around P_i. The closest point position $M_{Cl(i)}$ is computed with a sub-pixel precision. The isosurface is extracted with a modified version of the marching cubes algorithm[LC87][TG92]

4 Topology Control

In the previous section, we have described how to control the curvature of a simplex mesh in order to fit some three dimensional data. In this section, we present three algorithms for controlling the simplex mesh topology. Those procedures are presented in a hierarchical manner.

The first algorithm aims at concentrating vertices at parts of high mean curvature. The second algorithm on the other hand, updates the mesh connectivity functions by adding vertices at faces are elongated. Finally, we have designed a highlevel functions that create end contours at inaccurate parts.

4.1 Mesh Adaptivity

The metric parameters $\epsilon_i = \{\epsilon_{1i}, \epsilon_{2i}, \epsilon_{3i}\}$ are the barycentric coordinates of the foot of a vertex P_i on the triangle made by its three neighbors (see section 2.2.3). Those parameters control the relative distance of P_i from its three neighboring vertices. We can draw a parallel between the metric parameters of a simplex mesh with the first fundamental of a continuous surface since both relates to an intrinsic metric. On the hand, the simplex angle may be compared to the second fundamental form of a surface and relates to an extrinsic bending measure. While we have before used the simplex angle to control the variation of mean curvature, we will define in this section a procedure to adapt the spacing of vertices to the mean curvature of the mesh.

The notion of adaptive mesh has been studied by several researchers[McI93][VT92] [HDDM93][TK93][CM93]. In all cases, their aim is to provide an optimal shape description from a mesh with a fixed number of vertices, by concentrating vertices at highly curved parts. Our algorithm uses that approach but is characterized by the following properties:

- The concentration of vertices is governed by the local minimization of an energy $\mathcal{E}_i$. This energy expresses the link between the metric parameters and the variation of mean curvature.

- Vertices of low mean curvature migrate toward neighboring vertices of relatively larger mean curvature. Therefore, the concentration of vertices is governed by the relative variation of curvature.
- Vertices of high mean curvature have metric parameters close to $\frac{1}{3}$ in order to obtain a uniform concentration at highly curved parts.

We proceed by periodically adapting the metric parameters to the mean curvature of a mesh. if ϵ_i^t is the value of the metric parameters at iteration t, then we compute the metric parameter at iteration $t+p$ as:

$$\epsilon_i^{t+p} = \epsilon_i^t + \frac{1}{2}\vec{\nabla}\mathcal{E}_i$$

The energy $\mathcal{E}_i$ is defined similarly to the energy $\mathcal{E}_i$ described in section 3.3:

$$\mathcal{E}_i = \frac{1}{2}(\epsilon_i^\star - \epsilon_i)^2 \tag{25}$$

where $\epsilon_i^\star$ is computed as a function of the variation of the absolute value of the mean curvature. We derive the expression of $\epsilon_i^\star$ by first considering the mean value of the absolute mean curvature $|\bar{H}_i| = (|H_{N_1(i)}|+|H_{N_2(i)}|+|H_{N_3(i)}|)/3$. We then compute the relative mean curvature deviation vector $\delta|H|_i$:

$$\delta|H|_i = \begin{pmatrix} \frac{|H_{N_1(i)}|-|\bar{H}_i|}{|H_i|} \\ \frac{|H_{N_2(i)}|-|\bar{H}_i|}{|H_i|} \\ \frac{|H_{N_3(i)}|-|\bar{H}_i|}{|H_i|} \end{pmatrix}$$

We link the value of the reference metric parameter $\epsilon_i^\star$ with the relative mean curvature deviation vector:

$$\epsilon_i^\star = \frac{1}{3} + \gamma_i \delta|H|_i \tag{26}$$

γ_i is a constant that controls the extent of the adaptivity of the simplex mesh and is usually chosen between 0.03 and 0.25. However, since we guaranty all metric parameter ϵ_{ki} to be greater than 0.05 and lower than 0.833, we may have a γ_i substantially lower than 0.05. Finally we have:

$$\epsilon_i^{t+p} = \epsilon_i^t + \frac{1}{2}(\epsilon_i^\star - \epsilon_i^{t+p}) \tag{27}$$

The coefficient $\frac{1}{2}$ enables a smoother variation of the metric parameters over time. In practice, we choose to update the metric parameters every $p = 10$ iteration in order to stabilize the mesh before re-evaluating the mean curvature over the mesh.

Figure 22 and 23 shows the effect of the mesh adaptation on two isosurfaces shaped as a cube and a cross. The initial meshes correspond to the deformation of a semi-regular spherical mesh. The metric parameters are then equal to $\frac{1}{3}$. After the mesh adaptation, vertices nicely concentrates at highly curved parts, the level of concentration being controlled by the value of γ_i.

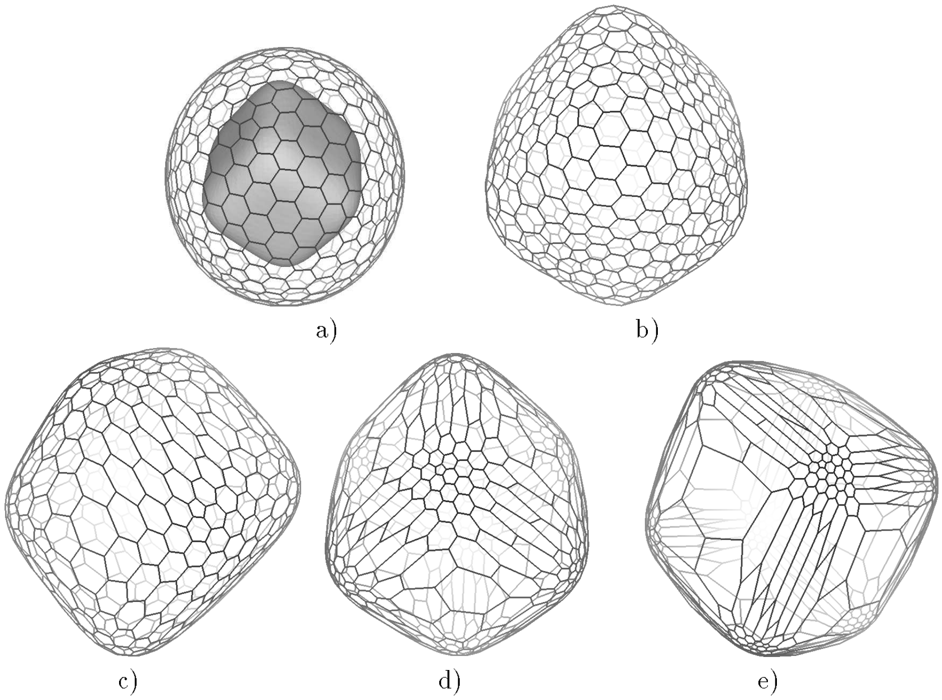
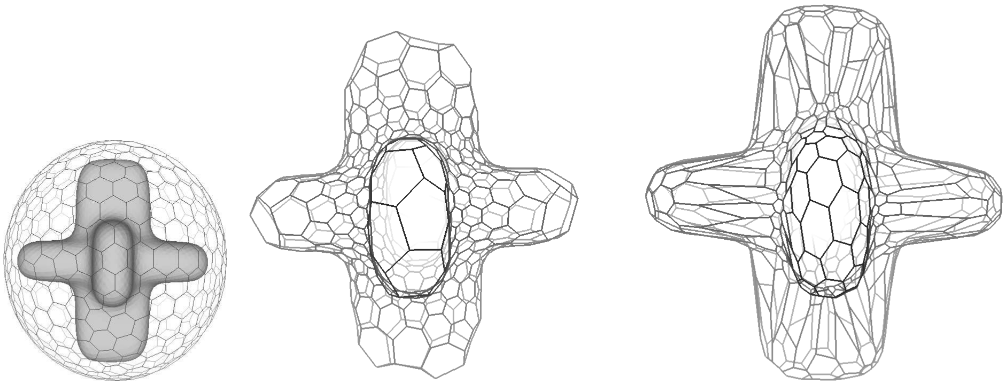


Figure 22: a) The initial mesh with the isosurface. b) the mesh fit on the isosurface. The metric parameters are everywhere equal to $\frac{1}{3}$. c) The adaptive mesh with a value of $\gamma_i = 0.10$. d) The adaptive mesh with a value of $\gamma_i = 0.15$. e) The adaptive mesh with a value of $\gamma_i = 0.20$.



a) b) c)

Figure 23: a) The initial mesh enclosing the isosurface; b) First fit with constant metric parameters; c) Mesh after adapting the metric parameters with $\gamma_i = 0.15$

4.2 Automatic Refinement Procedure

Mesh adaptivity optimally moves vertices toward parts of high curvature. However, the shape description is limited by the number of vertices of the initial mesh. We therefore introduce a method for refining a simplex mesh in order to control the absolute precision of the recovered model.

We refine faces of a mesh according to its curvature, to its distance from some 3D data, to its area and finally to its elongation. The refinement procedure is completed in an iterative way but operates at a higher level than the mesh adaptation. By incorporating the notion of area and elongation, we want to refine the elongated faces corresponding to flat part of the model (see figure 22).

The algorithm consists in first evaluating, for every face of a model, a criterion measuring the need for refinement. Then, faces whose criterion exceeds a given threshold, are refined and the mesh is deformed during a constant number of iterations. The refinement process is repeated until all faces criteria are below the threshold. This approach has the advantage of recovering models satisfying both geometric constraints (regularity and closeness of fit) and topological constraints (optimal vertex concentration).

The criterion is the product of four dimensionless parameters:

The relative area of a face It is computed as the ratio of a face area by the total area of a mesh. The area of a face is defined by triangulating the face as described in section 2.1.1.

The elongation of a face This coefficient is high when the face is elongated. It is computed as 1 plus the difference of length between the longest edge and the smallest edge of a face, divided by the median value of the length of the face edges .

The measure of Gaussian curvature . It is evaluated through the computation of area of the spherical polygon described by the normal vector on the Gaussian sphere (see section 2.2.4 .

The relative distance to 3D data . It is the ratio of the distance to the closest data point by the radius D of the corresponding range data.

The threshold is dimensionless as well and it is therefore evaluated independently of the size of the mesh. We have defined three threshold for coarse, medium or fine resolution. We have verified that those thresholds were practically meaningful.

The refinement may not perform well in the presence of strong noise or outliers where the distance between the mesh vertices and the data is relatively high. At those places, the refinement would result in actually fitting the noise. Therefore, we may only apply the refinement process at specified part of a mesh.

In order to keep the number of vertices per faces as close as possible to six, we either use a T_2^2 operation or a T_5^2 operation (see Figure 24) depending whether the face has more or less than five vertices.

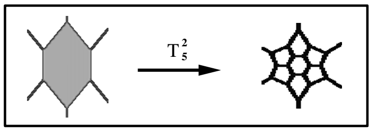
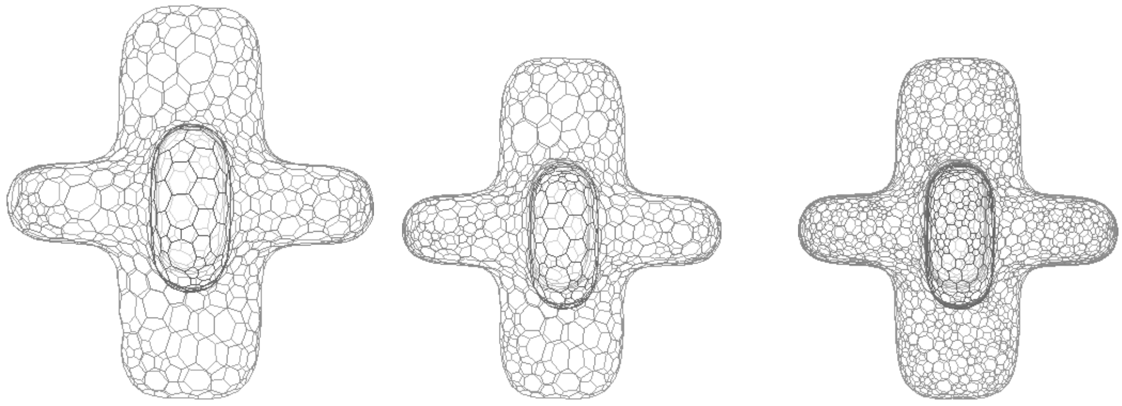


Figure 24: Operation T_5^2 for refining a face

In figure 25, we show an example of mesh refinement with an increasing level of precision.

4.3 End Contour Creation

We have implemented an algorithm that adapts the simplex model topology to the three dimensional data, by creating holes or ends at parts where the mesh is distant from the data. We proceed by first computing at each vertex the distance to the closest data point. We then create zones, i.e. set of faces whose vertices are located at a distance greater than a given threshold from the data. A set of contours surrounding those zones are created and operation T_3^2 is performed to remove the artificial part. Contours are systematically created around the newly created faces. Those deformable contours will deform to fit the shape of the hole existing in the data set.



a) b) c)

Figure 25: a) Starting from the same model as in figure 23(b) we refine the mesh with a low threshold value (without adaptation of the mesh); b) same as a) with a medium threshold value; c) same as a) with a high threshold value.

This procedure has been used in the building of the face model (see figure 27) and the skull model (see figure 32). The building of the face model shows clearly that it would be difficult to make this algorithm fully automatic. In this case, there are two zones labeled as distant. The upper zone corresponds to the hair and the lower zone is located at the neck. However, we want to interpolate the missing data corresponding to the hair rather then removing it. On the other hand, in order to recover the right topology, we need to remove the bottom zone. Therefore, we have chosen to create end contours in a semi-automatic manner by actually choosing which distant zone to remove.

5 Results

The surface reconstruction system we are presenting, has been implemented on a DEC Alpha AXP 3000/500 with a graphics accelerator. We have designed a complete user interface that enables to position the simplex mesh primitive close to the range data and to set the type of constraints on different parts on the mesh. Furthermore, cutting or merging are performed manually by selecting the appropriate contours or faces.

The recovery of three-dimensional objects from range data proceeds in two stages. In a first stage, the model is initialized as one of the four primitives of figure 2 with a limited number of vertices. The initial mesh does not need to be close to the range data, but it behaves more robustly when the mesh roughly encloses the data. We set a curvature continuity constraint on the whole mesh with a high rigidity value. A highly rigid model ensures smoother deformations and help to quickly stabilize the model. Around 50 iterations are usually needed to obtain the first fit. Typically, a mesh with 720 vertices takes less than 30 seconds to stabilize in the first place.

At the end of the first stage, the model tends to oversmooth the original data and furthermore, have few vertices at highly curved locations. In a second stage, we first trim the interpolated part of the mesh as seen in a previous section. We then set a low value of the rigidity parameter and finally apply the iterative adaptation and refinement process. This process can be run over the whole mesh or over selected parts while decreasing the rigidity parameter to its minimum. A new equilibrium is reached when high curved parts are sufficiently refined and when vertices closely approximate the raw data. This stage usually takes from 2 to 6 minutes depending on the shape complexity, the level of adaptivity and refinement required.

Minimal Mesh Figure 26 shows a minimal mesh, having the symmetry of a dodecahedron. The model was created from a tessellated dodecahedron similar to Figure 2.b. We have shrunk the twelve pentagons to a point and have constrained them to stay fixed during the deformation. The mesh resulting from the surface normal continuity constraint is such that each vertex is at the centroid of its three neighbors and therefore has zero mean curvature everywhere.

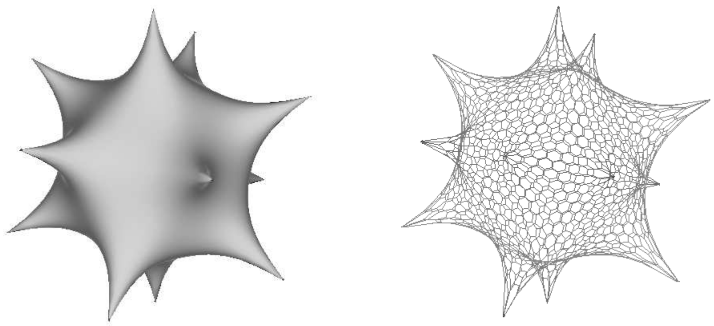


Figure 26: Minimal simplex mesh having the symmetry of a dodecahedron

Human Face Figure 27 shows the different stages of the fitting process on a human face digitized with a Cyberware Inc. range finder. The first fit does not correctly fit the nose and the chin, but nicely interpolates the missing data corresponding to the hair (figure b). We have first applied the adaptation process to show that vertices effectively concentrates at parts of high curvature (figure c). The mesh has been cut at the level of the neck and the bounding contour is deformed to fit the data.

Figures d,e and f display the result of the combined adaptation and refinement algorithms, for different levels of refinement. The chin, the nose, the ears , the eyebrows, the lips are successively recovered by increasing the mesh precision. The final mesh has 1700 vertices compared to the 260 000 data points provided by the digitalizer.

Hand Model A human hand is a rather difficult object to model due to its non-convexity. Performances of dynamic model fitting techniques are indeed limited by the local minimum problem that arises especially for concave shapes. Recovering a hand model from a single surface without a priori knowledge would be certainly difficult since the mesh would have to migrate between the fingers. A more natural approach for recovering complex objects, consists in connecting separately built models, corresponding to approximately convex subsets. Simplex meshes offer the necessary local representation (as opposed to the global representation of rectangular grids) needed to perform a "recovery by parts".

We have built a complete hand model from two Cyberware range images, one for each side of the hand(see figure 29). The palm and fingers are first modeled separately and then connected with several T_3^2 operations. We then compensate the finger displacement between two images by rigidly moving eighteen contours to their correct position while constraining the surface to keep the same curvature. The final model has about 8000 vertices and uses the texture of both range data.

Human Body We have modeled eleven distinct body parts from Cyberware range images. Figure 30 shows the complete body model, created out of the eleven parts and their mirrored image. In order to properly connect two parts, vertices around the junction are smoothed while keeping remaining other vertices fixed.

Mechanical Part We used an NRCC range data "Meca7" to fit a surface with two handles (see Figure31). The initial estimate was an ellipsoid enclosing the dataset. The mesh was refined first and then trimmed above the two handles. Those handles have been manually created by connecting pairwise opposite faces. Finally a contour placed inside each handle helps to overcome the lack of data on the side of the mechanical part.

Human Skull We use a $60 \times 60 \times 60$ CT Scan image of a human skull. The isosurface is a closed surface with a very complex topology. Fitting a mesh on a isosurface greatly differs from previous cases because the data is already segmented and complete. The aim here is to recover a simplified model of the skull in order to perform interactive deformation on a surgery simulator [DSCP94] . In order to obtain a more faithful model, it would be more appropriate to utilize the a priori knowledge of the shape and topology of the isosurface. In this example, we first deform a 2000 vertices spherical simplex mesh toward the skull data. The model then grossly oversmooth the skull especially around the jaw. The adaptation of the mesh greatly decreases its overall distance to the data. No refinement was performed in this example.

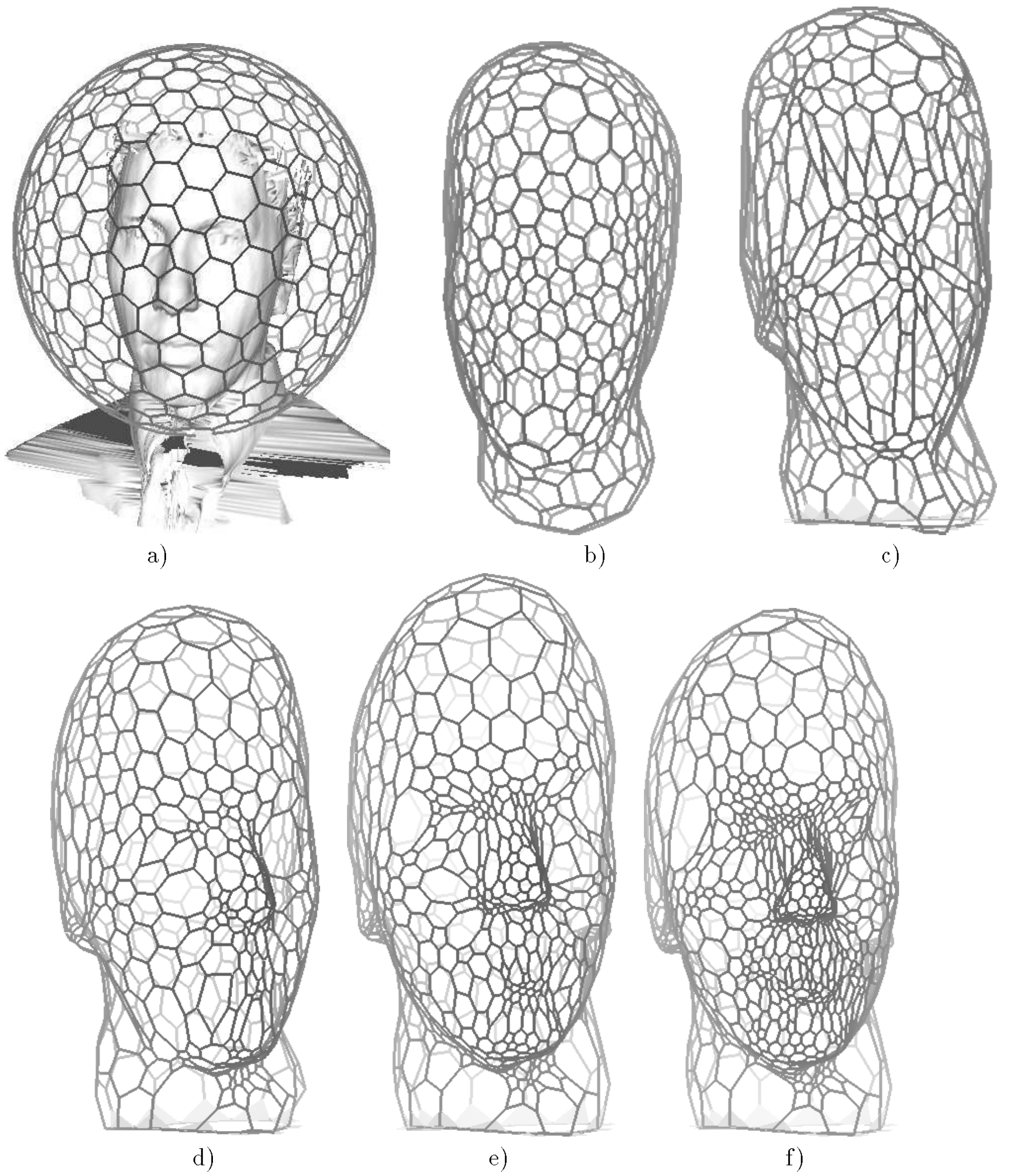
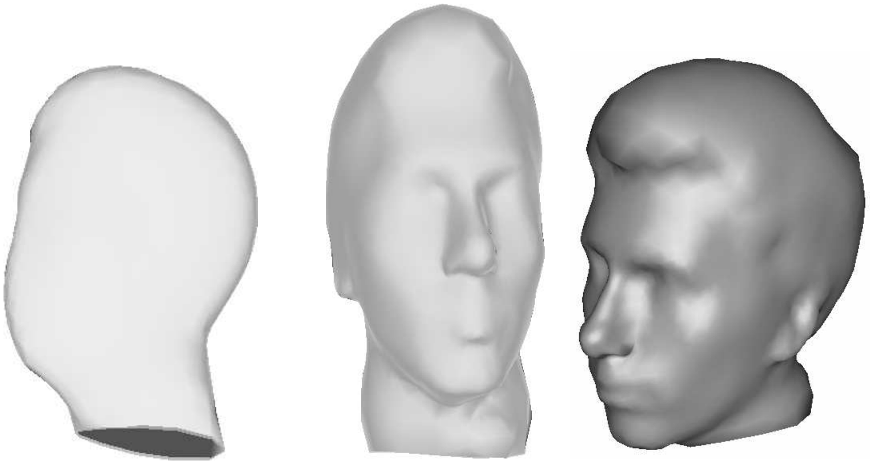


Figure 27: a) The initial mesh is a sphere with 720 nodes; b) the mesh is fit on the surface. The mesh grossly approximate the surface because it is smoothed over a large extent and because the mesh is not adapted; c) Mesh after the adaptation process. The vertices are concentrated around the nose, the chin and the ears. We chose a value of $\gamma_i = 0.15$; d) The mesh of figure b) is iteratively adapted and refined. We chose a small value of $\gamma_i = 0.06$ and a low threshold for controlling the refinement; e) Same as d) but with a medium amount of refinement; f) Same as d) but with a large amount of refinement;



a) b) c)

Figure 28: a) The rendered mesh resulting from the deformation of the sphere. An end contour were created at the location of the neck; b) The rendered mesh corresponding to figure 27(f); The rendered mesh corresponding to figure 27(h);

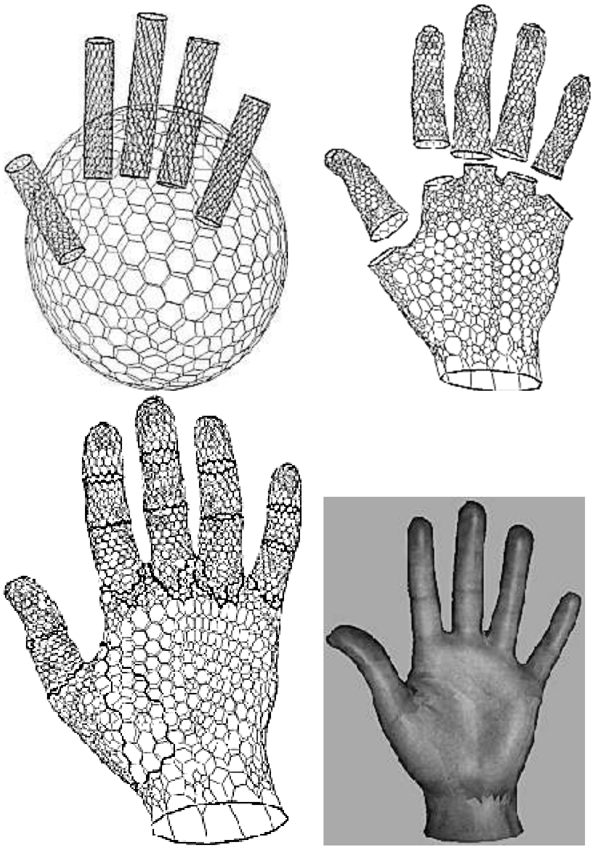


Figure 29: **upper left** Initial estimate of the palm and the five fingers. **upper right** The six meshes after convergence on two Cyberware range images. **lower left** Complete hand model. **lower right** Textured hand.

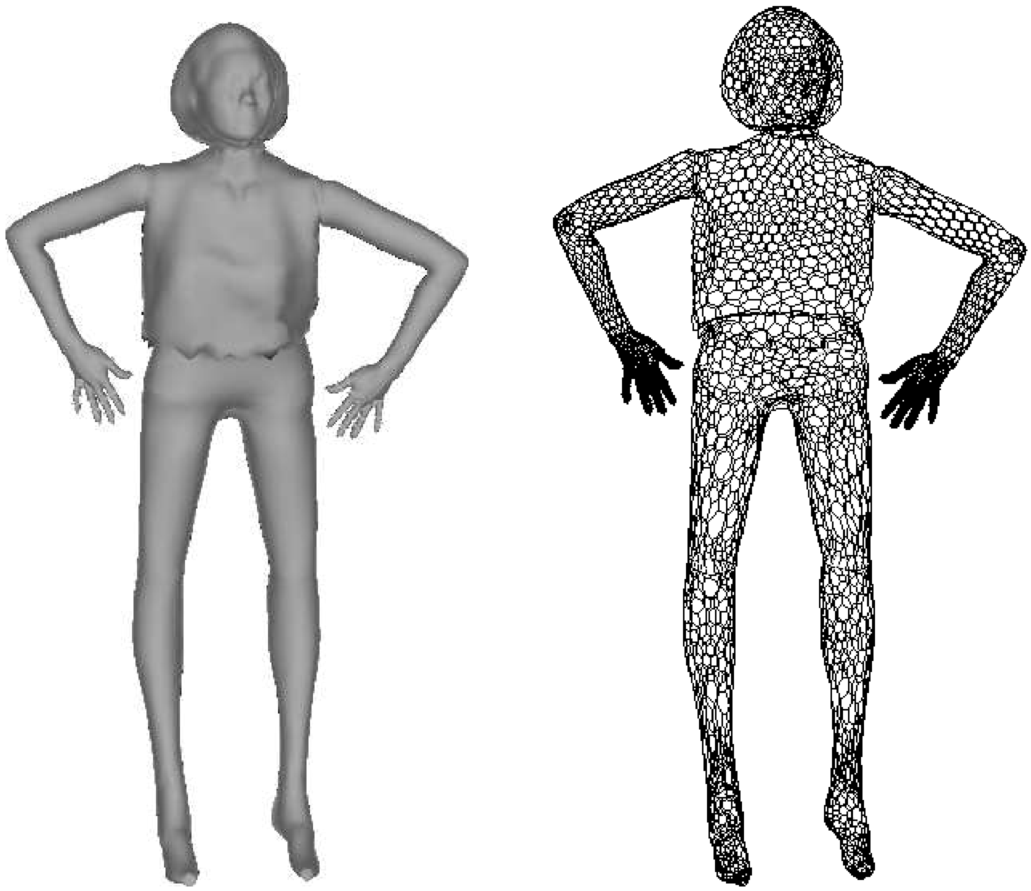


Figure 30: A complete body model recovered from 11 distincts range images. Three of those images (the head and both sides of the hand) were digitalized from a human body, the remaining were acquired from a manikin.

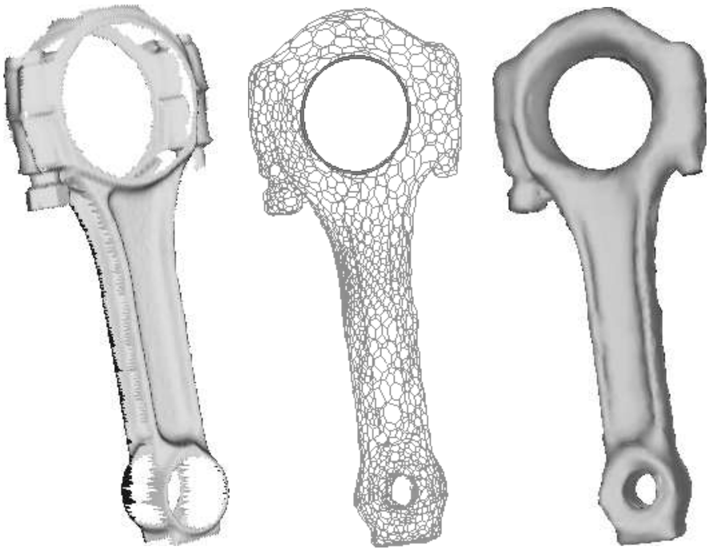


Figure 31: **Left** Range data and its mirrored image corresponding to a mechanical part.**Center and right** The recovered model has two handles . There are two contour inside each handle

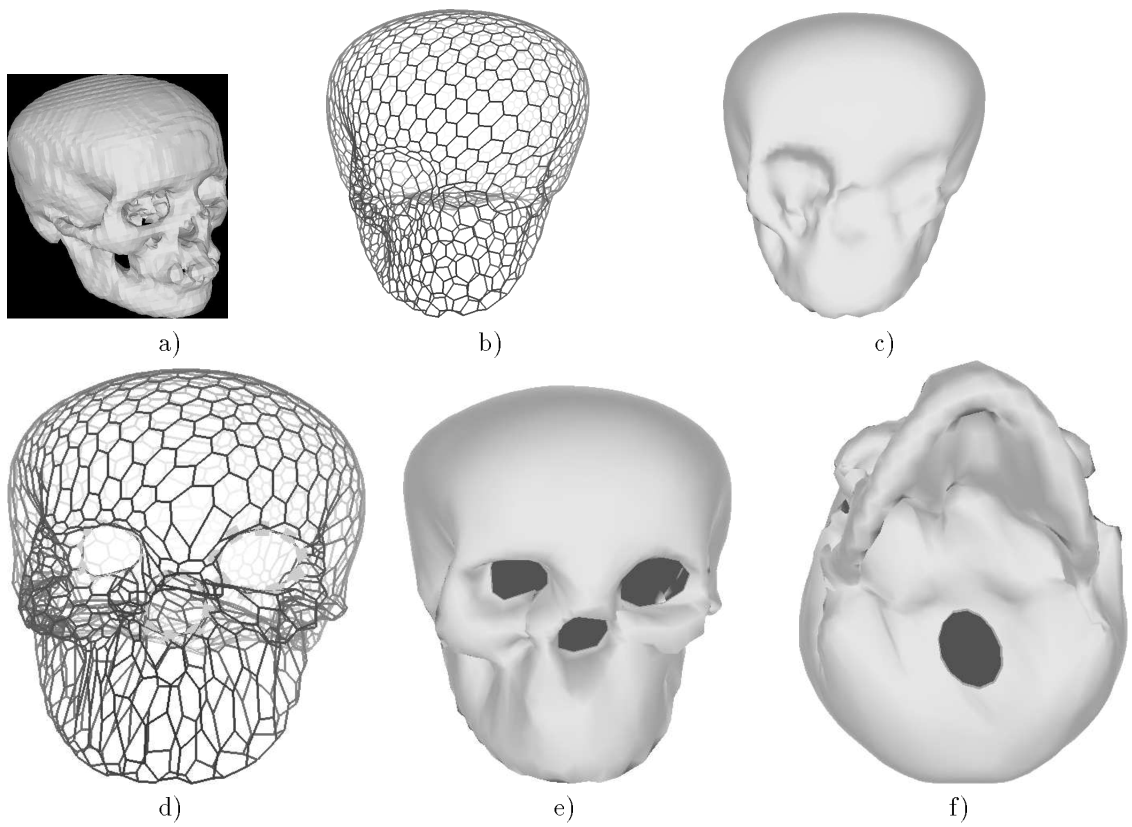


Figure 32: a) The isosurface extracted from a CT Scan Image; b) A sphere with 2000 vertices is deformed toward the skull data. We use a high value of the rigidity parameter; c) Rendered view of b); d) The mesh is now adapted to concentrate vertices at parts of high curvature. End contours located at the eyeballs, the nose and the foramen are automatically created; e) Bottom view of the skull

6 Conclusion

We have presented a general algorithm for recovering three dimensional objects from range data. Simplex meshes modeling differs from previous reconstruction techniques by their highly malleable and their essentially local nature. Intrinsic formulation of the internal constraints leads to natural and stable behavior. A control over the scale of deformation is necessary to prevent the mesh from being trapped in a local minima. We believe that deformable contours lying on deformable surfaces is a very powerful technique for handling surfaces of complex topology.

In the future, we would like to extend the framework of our system by recovery anatomical models from the gradient information extracted on a medical image. We are investigating as well, the recovery of realistic models by decimating a simplex mesh initialized as the dual mesh of a triangulated isosurface. We would like to elastically deform those generic models to efficiently extract anatomical models from different patient. Another research topic is the computation of a continuous surface representation associated with a simplex mesh.

Acknowledgments

We would like to thank Yasuhiko Watanabe and Yasuhito Suenaga of the NTT Human Interface Laboratory for providing the Cyberware and NRCC range images and their encouragement. We thank Bruce Latimer, Court Cutting and David Dean for providing the CT-Scan image of the skull. We would like to thank also Nicholas Ayache and Jacques Feldmar for stimulating discussions. This work was supported in part by a grant from Digital Equipment Corporation.

Appendix A

The relation between L, r_i, d_i, φ_i is shown in Figure 33. If we write θ_U as $\angle UP_iF$ and θ_V as $\angle VP_iF$ then :

$$\tan(\theta_U) = \frac{(r_i - d_i)}{L} \qquad \tan(\theta_V) = \frac{(r_i + d_i)}{L}$$

Since $\varphi_i = \pi - \theta_U - \theta_V$,

$$-\tan(\varphi_i) = \tan(\theta_U + \theta_V) = \frac{\tan(\theta_U) + \tan(\theta_V)}{1 - \theta_U \theta_V}$$

Finally,

$$L(r_i, d_i, \varphi_i) = \frac{(r_i^2 - d_i^2).\tan(\varphi_i)}{\epsilon.\sqrt{r_i^2 + (r_i^2 - d_i^2).\tan^2(\varphi_i)} + r_i} \tag{28}$$

$$\begin{array}{lll} \epsilon = 1 & if & |\varphi_i| < \pi/2 \\ \epsilon = -1 & if & |\varphi_i| > \pi/2 \end{array}$$

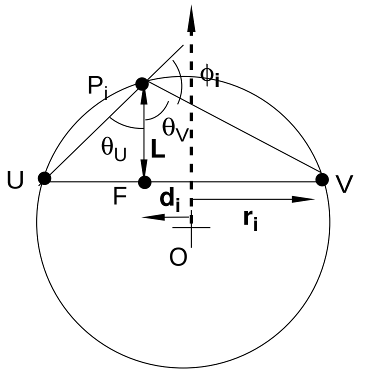


Figure 33: Computation of $L(r_i, d_i, \varphi_i)$

Appendix B

The purpose of this appendix is to prove property 4 which states that among all spheres passing through P, the minimal sphere minimizes its distance to the surface. We will restrict the proof to the case of three dimensional surfaces. The proof proceeds in two steps. In a first step, we will show that a sphere minimizing its distance to a surface should be tangent at the tangent plane of the surface. In a second step, we will prove that among all tangent spheres, the sphere of curvature H is the optimum.

Let Σ be a three dimensional surface defined over an open set $\Omega \subset \mathbb{R}^2$ and $P(u_0, v_0)$ a point on Σ. If we write k_1 and k_2 as the two principal curvature at $P(u_0, v_0)$ and ($\vec{e_1}$, $\vec{e_2}$, $\vec{N}$) as the principal frame at $P(u_0, v_0)$, we can write the Taylor expansion of the surface at the second order as :

$$P(u, v) = P(u_0, v_0) + (u - u_0)\sqrt{E}\vec{e_1} + (v - v_0)\sqrt{G}\vec{e_2} +$$
$$\frac{1}{2}(Ek_1(u - u_0)^2 + Gk_2(v - v_0)^2)\vec{N} + O(u - u_0)^3 + O(v - v_0)^3$$

where $E = \|P_u(u_0, v_0)\|$ and $E = \|P_v(u_0, v_0)\|$. We can therefore locally write the surface in an explicit manner:

$$z = \frac{1}{2}(k_1 x^2 + k_2 y^2) + O(x^3) + O(y^3)$$

Let S be a sphere passing through $P(.u_0, v_0)$. If the center of S does not belong to the tangent plane, we can write a Taylor expansion of the sphere up to order one, in the principal frame as $z = ax + by + O(x^2) + O(y^2)$ where $z = ax + by$ is the equation the tangent plane at the sphere. If the center of S does not belong to the tangent plane at the surface, we cannot define properly a distance between the surface and the sphere. We will discard this case in the remaining part of the proof.

Then we can always suppose that there exist a ball of radius r,$\mathcal{B}_r$ on the tangent plane centered on $P(u_0, v_0)$, where we can define a mapping $\mathcal{P}_1$ from $\mathcal{B}_r$ to Σ and a mapping $\mathcal{P}_2$ from $\mathcal{B}_r$ to S such that $\overrightarrow{M\mathcal{P}_1(M)} \times \vec{N} = \vec{0}$ and $\overrightarrow{M\mathcal{P}_2(M)} \times \vec{N} = \vec{0}$ (see figure 34. Finally, we define the distance D_r between the surface Σ and the sphere S at radius r as:

$$D_r = \int\int_{\mathcal{B}_r} ||\mathcal{P}_1(M)\mathcal{P}_2(M)||^2$$

Since $\overrightarrow{\mathcal{P}_1(M)\mathcal{P}_2(M)}$ is collinear to $\vec{N}$, we can express D_r up to the first order as:

$$D_r = \int\int_{\mathcal{B}_r} (ax + by)^2 dxdy + O(r^5)$$

If we change variables $(x, y) = (l\cos(\theta), l\sin(\theta))$, $(0 \leq l \leq r, 0 \leq \theta \leq 2\pi)$, then D_r evaluates as $D_r = \frac{(a^2+b^2)\pi l^3}{3}$. Therefore, in order to minimize D_r, a necessary condition is $a = b = 0$, which entails that the sphere is tangent to the surface at $P(u_0, v_0)$. Spheres tangent to the surface have a Taylor expansion of order two in the principal frame:

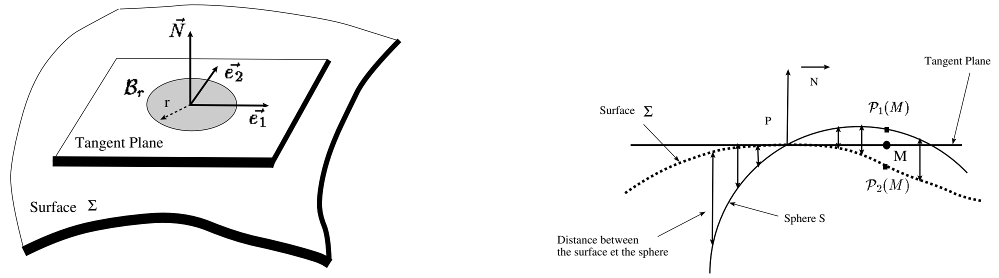
$$z = \frac{c}{2}(x^2 + y^2) + O(x^3) + O(y^3)$$

where c is the curvature of the sphere. For those spheres, the distance D_r is then computed as:

$$D_r = \int\int_{\mathcal{B}_r} ((c - k_1)x^2 + (c - k_2)y^2)^2 dxdy + O(r^7)$$

$$D_r = \frac{\left(8c^2 - 8ck_1 + 3k_1^2 - 8ck_2 + 2k_1k_2 + 3k_2^2\right)\pi r^5}{80} + O(r^7)$$

D_r reaches its minimum when $c = (k_1 + k_2)/2 = H$ $\square$



a) b)

Figure 34: a) The definition of the principal frame and the ball $\mathcal{B}_r$ around P; b) The distance is measured on a circular neighborhood in the tangent plane and along the normal direction

References

[Bla87] A. Blake, A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[BM92] P. J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[Boi84] JD. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions in Graphics*, 1984.

[Bor93] V. Borrelli. *Courbures Discretes.* Master's thesis, Universite Claude Bernard-Lyon 1, 1993.

[Car76] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice Hall, 1976.

[CCA92] I. Cohen, L.D. Cohen, and N. Ayache. Using deformable surfaces to segment 3-d images and infer differential structures. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 242–263, 1992.

[CM93] Y. Chen and G. Medioni. Integrating multiple range images using triangulation. In *Image Understanding Workshop*, pages 951–958, ARPA, 1993.

[CS93] X. Chen and F. Schmitt. Vision-based construction of cad models from range images. In *Proc. of the Fourth Int. Conf. on Computer Vision (ICCV'93)*, pages 129–136, 1993.

[Del94] H. Delingette. Intrinsic stabilizers of planar curves. In *3rd European Conference on Computer Vision (ECCV'94)*, Stockholm, Sweden, June 1994.

[DHI91a] H. Delingette, M. Hebert, and K. Ikeuchi. Energy functions for regularization algorithm. In *Proc. SPIE., Geometric Methods in Computer Vision, Vol. 1570*, pages 104–115, 1991.

[DHI91b] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. In *IEEE Computer Vision and Pattern Recognition (CVPR91)*, pages 467–472, June 1991.

[DHI93] H. Delingette, M. Hebert, and K. Ikeuchi. A spherical representation for the recognition of curved objects. In *Proc. of the International Conference on Computer Vision (ICCV'93)*, pages 103–112, 1993.

[DSCP94] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. *A Craniofacial Surgery Testbed.* Technical Report 2199, I.N.R.I.A., Sophia-Antipolis, France, February 1994.

[FA94] J. Feldmar and N. Ayache. Rigid and affine registration of smooth surfaces using differential properties. In *3rd European Conference on Computer Vision (ECCV'94)*, Stockholm, May 1994.

[Gue93] A. Gueziec. Large deformable splines, crest lines and matching. In *Proc. of the Fourth Int. Conf. on Computer Vision (ICCV'93)*, pages 650–657, 1993.

[HDDM93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Computer Graphics (SIGGRAPH'93)*, pages 19–26, 1993.

[Hil44] D. Hilbert. *Geometry and the Imagination.* Dover, 1944.

[Hor84] B.K.P. Horn. Extended gaussian images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 72(12):1671–1686, December 1984.

[Ike81] K. Ikeuchi. Recognition of 3d objects using the extended gaussian image. In *Proc. of the Seventh IJCAI*, pages 595–600, 1981.

[Koe90] Jan J. Koenderink. *Solid Shape.* MIT Press, 1990.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[LC87] W. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *ACM Computer Graphics (SIGGRAPH'87)*, 21:163–169, 1987.

[Lei93] Leitner. *Segmentation Dynamique d'images tridimensionnelles.* Ph.D. dissertation, Institut National Polytechnique de Grenoble, Grenoble, France, September 1993.

[McI93] D. McInerney, T. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proc. of the Fourth Int. Conf. on Computer Vision (ICCV'93)*, pages 518–523, 1993.

[Meh74] E. Mehlum. Nonlinear splines. In *Computer Geometric Aided Design*, 1974.

[MS92] H. P. Moreton and C. H. Sequin. Functional optimization for fair surface design. In *Computer Graphics (SIGGRAPH'88)*, pages 167–176, July 1992.

[NA93] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *Proc. of the Fourth Int. Conf. on Computer Vision (ICCV'93)*, pages 275–279, May 1993.

[RC88] M. Rioux and L Cournoyer. *The NRCC three-dimensional Image data files.* Technical Report NRCC 29077, National Research Council of Canada, Ottawa, Canada, 1988.

[SI87] K. Sato and S. Inokuchi. Range-imaging system utilizing nematic liquid crystal mask. In *International Conference on Computer Vision (ICCV'87)*, pages 657–661, 1987.

[ST92] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH'92)*, pages 185–194, July 1992.

[SW91] Y. Suenaga and Y. Watanabe. A method for the synchronized acquisition of cylindrical range and color data. *IEICE Transactions*, E 74(10):3407–3416, October 1991.

[TA87] A.N. Tikhonov and V.A. Arsenin. *Solutions of Ill-Posed Problems.* Winston, 1987.

[TG92] J.P. Thirion and A. Gourdon. *The 3D Marching Lines Algorithm and its application to crest lines extraction.* Technical Report 1672, I.N.R.I.A., May 1992.

[TK93] H. Tanaka and F. Kishino. Adaptive mesh generation for surface reconstruction. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR'93)*, pages 88–94, New-York, June 1993.

[TWK87] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3d object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1987.

[VM93] B. Vemuri and R. Malladi. Constructing intrinsic poarameters with active models for invariant surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 668–681, 1993.

[VT92] M. Vasilescu and D. Terzopoulos. Adaptative meshes and shells. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR'92)*, pages 829–832, 1992.

[WW92] W. Welch and A. Witkin. Variational surface modeling. In *Computer Graphics (SIGGRAPH'92)*, pages 157–166, July 1992.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399