Simplex Based Animation

H. Delingette, H. Watanabe, Y. Suenaga

ABSTRACT

We propose a methodology for animating complex models based on simplex mesh representation and deformation. A simplex mesh is characterized by a connectivity between vertices of three and can be obtained by duality from a triangulation. In addition to their generality of representation, simplex meshes have a compact and unambiguous shape description, related to the notion of mean curvature.

We devised a physically-based metamorphism algorithm that rests on the intrinsic shape representation of these meshes. This algorithm handles transformations between objects of different topology or boundary conditions. Various simplex mesh shapes were extracted from range images or some triangulated data and then metamorphosed by applying generalized mesh transformations.

Keywords: object representation, duality, triangulation, modeling, metamorphism.

1 INTRODUCTION

Physically based techniques use a large set of nodes to represent objects which makes them well suited for generating complex shapes. Each node is moved independently according to Newton's law of motion. A "cohesion" force is used to correlate the displacements of neighboring nodes thereby constraining the freedom of the surface. Physically based modeling gives the model a dynamic behavior resulting in a more intuitive interaction.

First introduced by Terzopoulos, Kass and Witkin[10][20] to extract contour or axial-symmetric surfaces from video images, elastically deformable models have been extensively used both in computer vision and computer graphics[4][6]. The equations of motion are derived by minimizing quadratic elastic energies such as the bivariate generalized spline functionals[18], through some variational principals. Solutions are computed over time by using finite differences with explicit [6] or semi-implicit[4] schemes, or finite-element analysis[2]. External constraints have been designed to fit range-data[6][20], to enhance the user interface or to simulate physical phenomena such as object contact, viscoelasticity or animated characters.

Though appealing for their clay-like behavior, elastic models are difficult to manipulate because they do not respond to global constraints. Shaping an object with local constraints such as assigning node positions or normal orientations requires too many operations and is practically ineffective. Welch and Witkin[21] proposed a framework where curves can be attached to a tensor-product spline surface while minimizing some objective function. Curves provide a natural boundary condition of surface models and handling curves instead of surfaces leads to improved interfaces since three-dimensional curves can be directly input.

Our modeling system is physically-based as well, but a different surface representation is introduced. Simplex meshes are general enough to represent surfaces of all topology and boundary conditions. In particular they overcome the pole problem that arise when representing a closed object with tensor-product splines. Moreover, we can add or remove nodes from the mesh structure in a natural manner without perturbing the mesh continuity and connectivity. Internal constraints may be defined in terms of intrinsic parameters without great computational cost. Unlike most elastic deformable models algorithms, internal constraints are achieved through the minimization of some local energy. Therefore, we can consider a simplex mesh as a network of independant particles with fixed connectivity. Global interaction is handled though contours similarly to Welch and Witkin[21].

Section 2 introduce the concept of simplex mesh while Sections 3 and 4 describe surface and contour internal constraints. Range data interaction and mesh transformation operations are discussed in Sections 5 and 6.

2 SIMPLEX MESH

2.1 Definition

Among all the possible surface representations, polyhedra are of wide use in computer graphics. Most polyhedral representations employ either triangular or rectangular patches. Triangulation handles planar patches, therefore, surface normals may be computed along the surface without ambiguity. On the other hand, rectangular grids can be interpreted as a tensor product of two splines which decreases the complexity of the representation. Both representations share the same geometric property as to yield a regular tessellation of the plane.

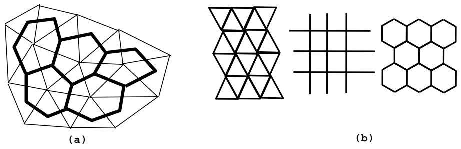

Figure 1: (a) Dual of triangulation; (b) the three regular tessellation of plane

However, there is a third regular tessellation pattern (see Figure 1(b)) based on hexagons which can be developed from the duality of regular triangulation. The duality of polyhedra associates faces with vertices and edges with edges and plays an important role in the field of regular polytopes.

We introduce another powerful polyhedral representation of three-dimensional surfaces which is characterized by a connectivity between vertices that is equal to three and that are dual triangulations (see Figure 1 (a)). We coined the word “Simplex Mesh” to describe a mesh representing a surface in dimension n for which each node is linked to n neighbors. Therefore in a simplex mesh, we can define around each vertex a n-simplex made of $n + 1$

nodes. In particular, for a three-dimensional surface, a tetrahedron, the 3-simplex, is defined at each node. Table 1 summarizes the connectivity of the three representations and their dual nature.

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

Table 1: Properties of triangulation, rectangular grid and simplex mesh

We will consider only closed meshes for which the three-connectivity is valid at each node. The Euler relation links the number of vertices V, the number edges E, the number of faces F and the genus g of the surface :

$$F - \frac{V}{2} = 2 * (1 - g) \qquad E = \frac{3V}{2} \tag{1}$$

A simplex mesh $\mathcal{M}$ is fully described by its n vertices $\{P_i\}$ and by the $3n/2$ vertex to vertex relations $\{(P_i, P_j)\}$. From this, vertex to face and edge to face relations can be derived and each face can be consistently oriented. We will write $N_j(i)$ as the index of the $j^{th}, (j = 1, 2, 3)$ neighbor of node number $i, (i = 0, .., n-1)$ so that $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$ are the three nodes connected to P_i.

The advantage of the simplex-mesh is that a normal vector can be defined at each node by considering the normal at the plane defined by its three-neighbors. Furthermore we can associate each tetrahedron with a circumscribed sphere which provides a measure of mean curvature of the surface. This is in contrast to triangulated mesh that provide a measure of gaussian curvature through the spherical-excess of dihedral angles[11].

Simplex meshes like triangulations can represent all orientable surfaces. This is in contrast with rectangular grids that exhibits poles for closed surfaces of genus 0.

Simplex-meshes have a vertex to vertex connectivity of three but each face consists of a variable number of vertices. We define a *p-face* as a face consisting of p vertices. Meshes whose faces have the same number of vertices are called regular. If the surface has one handle (genus strictly positive), it is possible to build a regular hexagonal simplex-mesh with a variable number of nodes. On the other hand, for closed surface without handles, there are only three regular models : tetrahedron, cube and dodecahedron.

Another advantage of simplex meshes is the possibility of introducing the notion of *End* or "empty face". A face is labeled as an end when we want to create a hole in the surface. The numbers of ends and handles are powerful characteristics for classifying surfaces. For example, a cylinder has two ends while a sphere has none; both have no handles. On the other hand, a torus has one handle but no ends.

Precise rendering of simplex meshes is difficult since each face is actually wedged and normals are impossible to define at each face. The current solution is to build a triangulated model by associating the center of a p-face with each vertex. The face center is computed as the centroid of the p vertices which tends to flatten the overall shape.

2.2 Equation of Motion

The dynamics of each vertex is given by a newtonian law of motion:

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

where m is the mass unit of a node and γ is the damping factor. $\vec{F_{int}}$ is the force created to make the surface continuous while $\vec{F_{ext}}$ corresponds to external constraints defined by either the user or some three-dimensional data.

Time is discretized as $t_i = t_0 + i * \Delta t$ and Equation (2) is integrated over time using finite differences with explicit scheme. A more stable though more complex implementation would use semi-implicit schemes[4]. If P_i^t is the position of vertex i at time t then the discretized law of motion is :

$$P_i^{t+1} = (1-\gamma)(P_i^t - P_i^{t-1}) + \vec{F_{int}} + \vec{F_{ext}} \tag{3}$$

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

3 SHAPE FUNCTIONALS

The internal forces of a physically-based model determines the model's response to external constraints. Such response should be both intuitive and invariant by isometries.

Researchers have proposed a wide variety of elastic energy terms. Energies proportional to the squared curvature[12][9][1] have the advantage of being physically meaningful but they do not accept circles[5] or spheres as optimum. Moreton et al.[14] introduced a fairness measure proportional to the squared derivative of curvature which yields spheres, tori and cylinders as optimum shape.

Curvature-based energies lead to complex non-linear expressions of $\vec{F_{int}}$. Linear elastic energies where the curvature is approximated by second order derivatives have been extensively used in the computer-aided-geometric design field[7] as well as for deformable modeling [21][2][6][20]. In the generalized formulation, the energy is composed of a stretching term and a bending term. Some weights can be adjusted to take into account surface discontinuities and normal discontinuities. Linear elasticity has proved to be efficient but there is no guarantee that fair shapes will be delivered since they are not expressed in terms of intrinsic parameters. They have a tendency to consistently flatten high curvature parts and do not accept circles or spheres as optimum[5].

Our framework does not derive $\vec{F_{int}}$ from the minimization of some global elastic energy. Instead we associate with each node P_i a local energy $\mathcal{S}_i$ that characterizes the state of local tetrahedron $(P_i, P_{N1(i)}, P_{N2(i)}, P_{N3(i)})$. More precisely the position of P_i can be determined from the position of its three neighbors by the relation:

$$P_i = \underbrace{\epsilon_{1i}P_{N_1(i)} + \epsilon_{2i}P_{N_2(i)} + \epsilon_{3i}P_{N_3(i)}}_{F} + L(r_i, d_i, \phi_i)\vec{N_i} \tag{4}$$

where

- F is the foot of P_i.

- r_i is the radius of the circle circumscribing $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$.
- d_i is the distance between F and the circle's center.
- ϕ_i is the simplex angle at P_i.
- $L(r_i, d, \phi_i)$ is the function described in appendix A.
- $\vec{N_i}$ is the normal vector at plane $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$.

Figure 2 shows the relation in the local tetrahedron:

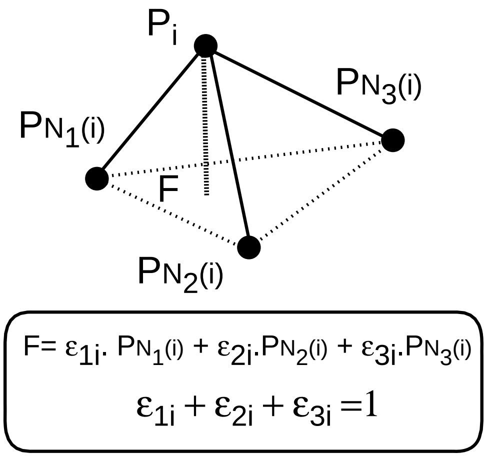

Figure 2: Curvature definition for a simplex angle ϕ_i

The simplex angle ϕ_i characterizes the mean curvature at P_i. With each node P_i we associate a simplex angle $\phi_i^\star$ for which the local tetrahedron $(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ is in its rest position. $\phi_i^\star$ 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 node P_i if the simplex angle was $\phi_i^\star$ then the energy $\mathcal{S}_i$ of the local tetrahedron is:

$$\mathcal{S}_i = \frac{\alpha_i}{2} P_i {P_i^\star}^2 \qquad (0 \leq \alpha_i \leq 0.5) \tag{5}$$

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

$$\vec{F_{int}} = \alpha_i\ \overrightarrow{P_i P_i^\star} \qquad (0 \leq \alpha_i \leq 0.5) \tag{6}$$

In this framework each node can be seen as an independant particle that interacts with its three neighbors and their surrounding nodes. Interacting sets of particles have been used in computer graphics to model viscous fluids [13] or thermoplasticity[19]. Szeliski[17] used particles that tended to align their orientations to interpolate 3D data without a priori knowledge of connectivity or topology. There is one important difference between simplex mesh and particles system in that simplex meshes constrain nodes connectivity to three which restricts the generality of representation but leads to smoother shapes and more efficient computation.

Different constraints are set depending on the computation of $\phi_i^\star$. We define three types of constraints applicable to any part of a simplex mesh.

The first constraint enforces smoothness along the surface and corresponds to a default option when no underlying shape is known. Resulting surfaces are piecewise C^2 continuous and their fairness is guaranteed since the laplacian of mean curvature over the mesh is null: $\Delta H = 0$. Spheres, cylinders, cones satisfy this equation as well as minimal surfaces for which $H = 0$. Minimal surfaces are of great interests in computer-aided design since they minimize the surface area spanned between two curves.

The second constraint is related to the notion of rest shape to which the model converges when no external constraints are applied. A representation of simplex meshes invariant by rotation, translation and scale is used to describe the underlying shape.

The third constraint concerns the adding of surface normal discontinuities in order to model sharp edges or conic points.

4 CONTOURS

4.1 Definition

A contour $\mathcal{C}$ attached to a mesh $\mathcal{M}$ is defined as a set of nodes $\{P_{L(i)}\}, (i = 0, l)$ such that:

- $L(-1) = L(l), L(l+1) = L(0)$
- $\forall i, \exists j, L(i+1) = N_j(L(i))$
- $\forall i, \forall j, P_{L(i)} \neq P_{L(j)}$

Therefore, $\mathcal{C}$ is a closed curve with consecutive nodes connected in mesh $\mathcal{M}$. The third condition states that $\mathcal{C}$ does not self-intersect (see Figure 3). When a contour encloses a p-face, the face is labelled as an end and the surface is trimmed along the contour. A contour is managed as an entity independantly from the mesh it is attached to. A node is attached to no more than one curve.

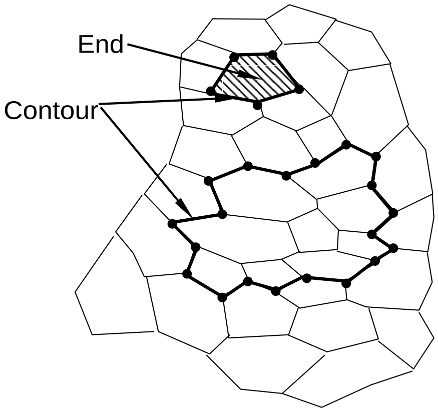

Figure 3: Two contours attached to a simplex mesh.

A contour is handled independently of the surface mesh and is also subjected to internal and external constraints. The only difference in terms of implementation is that internal force $\vec{F}_{int}$ at each node $P_{L(i)}$ is computed in order to ensure the elastic behavior of the contour instead of the surface model.

We propose an elastic functional that does not entail any shrinking effect and that exhibits stable behavior. Under this internal constraint contours are smoothed and converge toward their stable shape, circles. The expression of $\vec{F}_{int}$ is a generalization of functionals described in [5] and previous section.

4.2 Boundary Conditions

Boundary conditions describe how contours are embedded inside a mesh and are therefore important control parameters for shaping a model. The simplex mesh provide a simple way of set boundary conditions by controlling simplex angles $\phi_{L(i)}$ at each node of curve $\{P_{L(i)}\}$. The underlying assumption is that smoothness forces apply to surface nodes located around each contour. We defined two types of surface-contour constraints:

Curvature Constraint : Mean curvature is constrained at nodes surrounding a contour.

Tangent Constraint : 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. This angle can be controlled through the value of $\phi_{L(i)}$

Figure 9 shows a vase created from a cylinder and five contours. Two contours are defined by interpolation of four non-coplanar points and have tangent constraints of $\pi/4$ and $-\pi/4$ respectively while the three others are circles with null curvature constraints. Tangent constraints guarantees a C^1 continuity across a contour while curvature constraints leads to C^0 continuity only.

5 DATA CONSTRAINT

Three dimensional ranging device have gained popularity in computer graphics for building realistic models of existing objects. Physically based modeling systems are well suited for this task since data is often noisy or incomplete. The fitting process is performed by a potential field that drags the surface model close to the three dimensional data while the interpolating capability results from the internal smoothness constraints. Local minima may arise especially when models are initialized far from the data. We avoid this problem by interactively positionning and scaling the initial mesh as to get a reasonable estimate of the object shape. The mouse may be used also to drag a mesh out of a local minimum.

We follow a similar formulation as in [6] where the potential field has a limited range. For each mode P_i we determine the closest point in the data structure $P_{Cl(i)}$ and compute the attracting force as:

$$\vec{F}_{ext} = \beta_i G\left(\frac{\| \overrightarrow{P_iP_{Cl(i)}} \|}{D}\right)(\overrightarrow{P_iP_{Cl(i)}} \cdot \vec{N}_i)\vec{N}_i \tag{7}$$

where $\vec{N}_i$ is the surface normal at P_i and $G(x)$ is the function of Figure 4. D is the maximum distance at which some data points attracts a node point and is computed as a function of the overall mesh size. The function $G(x)$ is designed such that the force is linear when data is within a distance D of the surface model but decreases sharply otherwise. A bounded range potential field is important in order to limit the influence of outliers or to models objects with narrow shapes. In order to get a smooth deformation in presence of sparse data, the force is projected on the normal surface vector.

The search for the closest point $P_{Cl(i)}$ is theoretically in $O(m^2)$ for a $m \times m$ range image but we were able to decrease the complexity to $O(m)$ by restricting the search along one image line.

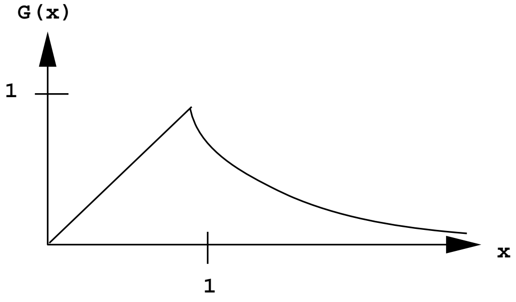

Figure 4: Function $G(x)$.

6 MESH TRANSFORMATION

6.1 Definition

A mechanism for mesh refinement is important to provide the maximum of flexibility to a modeling system. Simplex mesh structures can be locally altered without exhibiting any irregularity in the mesh connectivity. Therefore, nodes may be added or deleted locally without perturbing geometric continuity and a mesh is handled as an indefinitely expendable surface.

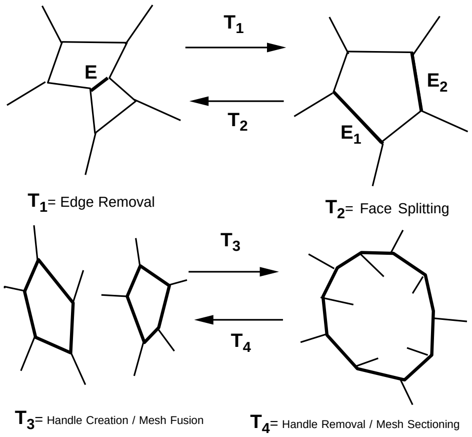
T1
E
T2
E2
E1
T1= Edge Removal
T2= Face Splitting
T3
T4
T3= Handle Creation / Mesh Fusion
T4= Handle Removal / Mesh Sectioning

Figure 5: The four basic mesh transformations

We define four basic mesh transformation operations (see Figure 5). The first two, T_1 and T_2, are the inverse of each other and can be interpreted as, respectively, edge removal and face splitting operations. These transformations do not change the mesh topology but instead decrease or increase the density of nodes. T_3 is interpreted as a handle creation operation if both faces belong to the same mesh or as a mesh fusion operation otherwise. In the latter case the number of handles of the resulting mesh is the sum of the two handle numbers. T_4 amounts to cutting a mesh along a contour and results in either removing a handle or sectioning a mesh into two parts.

The general mesh transformation T that transforms a mesh $\mathcal{M}_1$ into $\mathcal{M}_2$ is an ordered set of the four basic operations :

$$T = \{T_{f(0)}, T_{f(1)}, .., T_{f(q)}\} \quad with \quad f(i) \in \{0, 1, 2, 3, 4\}$$

A transformation from $\mathcal{M}_2$ to $\mathcal{M}_1$ is:

$$T = \{T_{Inv(f(q))}, T_{Inv(f(q-1))}, .., T_{Inv(f(0))}\}$$

where $Inv(x)$ a function defined as:

$$Inv(1) = 2 \quad Inv(2) = 1 \quad Inv(3) = 4 \quad Inv(4) = 3$$

In particular, there are two important macro-transformations, T_5 and T_6 that respectively increase and decrease locally the mesh resolution by duplicating and removing a face (see Figure 6). Both can be decomposed into combinations of T_1 and T_2.

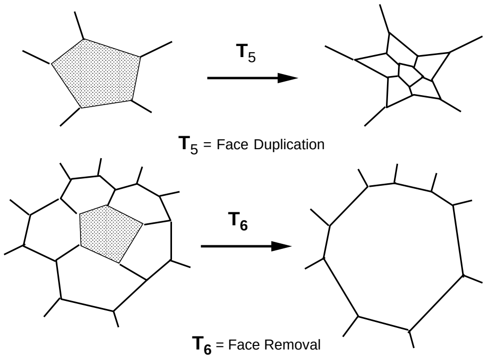

Figure 6: Mesh resolution transformations

7 EXAMPLES

We present two examples of models build from different range images. The three dimensional data with associated texture were provided by a Cyberware digitizer[16]. The modeling process can be decomposed into two stages. In the first stage, the model is initialized as a generic shape, either an ellipsoid, a cylinder or a plane, and then fit to the range data. In the second stage the mesh is refined as parts of high curvature and is eventually attached with another simplex mesh.

The first example shows how to build a hand model given two range images, one for each side of the hand (see Figure 10). The palm and fingers are first modeled separately and then connected to each other with several T_3 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 nodes and uses the texture of the two range data.

The second examples combines six range images of some body parts taken either from a mannequin or from a real person (see Figure 11). Junctions between the several meshes are smoothed until they reaches C^2 continuity.

8 INTERFACE

We designed an interface that provides the ability to interactively select a node, a face or a part of a mesh and then assign it some property. In addition to setting internal and external constraints, user may "nail" some nodes or use the mouse to drag both surface and contour nodes. Viewing point as well as contours and surfaces may be rotated, translated or scaled. Range images can be displayed as well providing a quick way to check the goodness of fit.

Constraint visualization is possible by attaching color patches to each node. Computational time is proportional to the number of mesh nodes but is, in most cases, small enough to provide real time feedback.

9 ANIMATION

9.1 Metamorphism

We combine two important properties of simplex mesh, their intrinsic shape representation and their mesh transformation operations in order to transform an object $\mathcal{A}$ into an object $\mathcal{B}$ (see Figure 13). A simplex mesh shape is fully described by the $3n$ values $\{\epsilon_{1i}, \epsilon_{2i}, \phi_i\}(i = 0, .., n)$ and its mesh structure. If model $\mathcal{A}$ has shape $\{\epsilon_{1i}^A, \epsilon_{2i}^A, \phi_i^A\}(i = 0, .., n)$ and model $\mathcal{B}$ has shape $\{\epsilon_{1j}^B, \epsilon_{2j}^B, \phi_j^B\}(j = 0, .., m)$, we can transform $\mathcal{A}$ into $\mathcal{B}$ by first applying a generalized mesh transformation $T_{\mathcal{A} \rightarrow \mathcal{B}}$ and then assigning the $3m$ shape parameters to each node. The internal constraint force F_{int} brings the mesh to its rest shape that corresponds to the shape of object $\mathcal{B}$. Since shape is described independantly of scale, we can render object deformations with a constant volume or constant area.

In the current status, all objects are either extracted from a range image or from some triangulated data. Given a simplex mesh representing an object $\mathcal{A}$, we fit the mesh on object $\mathcal{B}$ and modify its structure in order have a precise rendering of the object. We record and store the list of basic transformations as the generalized mesh transformation $T_{\mathcal{A} \rightarrow \mathcal{B}}$. Future extension would directly compute $T_{\mathcal{A} \rightarrow \mathcal{B}}$ from the two simplex meshes without having the fitting stage.

9.2 Articulated simplex mesh

Rotational joints are widely used especially to animate human parts models. A common approach is to consider hierarchical layered models with the lowest layer representing the bones and the upper layer the human skin[8][3]. Bones are rigidly rotated around some predefined axis while the skin is of elastic nature. Interaction between layers constrains the skin to closely surrounds the bones.

Layered models are closely related to the human physiology, but some magic parameters tune-up is necessary to render realistic animation. We propose an algorithm that simply and efficiently animates articulated simplex meshes. A joint is uniquely defined by two contours, one face and two vertices (see Figure7).

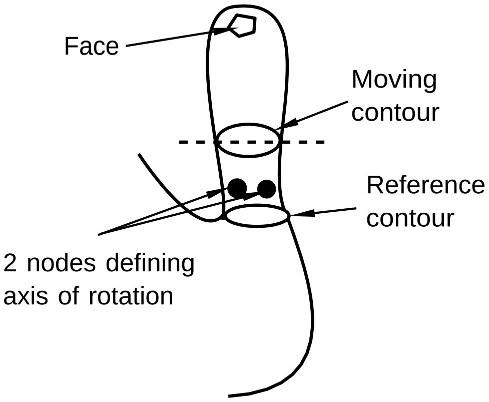

Figure 7: Rotational joint definition

The first contour is the moving contour where the surface is articulated. A contour splits a mesh into two parts and therefore a face indicates which part of a mesh is supposed to rotate. The two vertices defined the axis of rotation and the rotation angle is measured between the two segments that join the center of the moving contour to respectively the center of the face and the center of the second contour called refence contour. When the user changes the articulation angle of $\delta\theta$, all nodes on the moving side of the articulation contour are rotated of the same amount $\delta\theta$ while the contour is rotated around its center of the amount $\delta\theta/2$. This relation indicates that the contour bisects the angle between the moving part and the reference part which in general gives a natural effect. Surfaces nodes are constrained to keep constant curvature which results in a natural smoothing of the joint. More realistic effects such as the formation of wrinkles could be rendered by setting tangent end conditions at the moving contour instead of curvature constraints.

We have build a twenty degrees of freedom articulated hand from the model shown in Figure 10. We created four joints per finger following the same taxonomy as [15]. Joints are created fully interactively by selecting the five items. Figure 12 shows the model with the twenty articulations as seen from the interface. User may select each joint and set its rotation angle.

10 CONCLUSION

In this paper we have presented a physically based modeling system that represents three-dimensional surfaces as simplex meshes. By combining the notion of ends, mesh transformations, contours, smoothness and curvature constraints, our system's framework gives the user a large freedom of action at a local as well as a global level. Range images are handled as additional shape constraints and resulting models may be merged to render realistic images. An interactive interface that provides the ability to select a node, a face or a part of a mesh and to assign it some property was designed. We have developed in this perspective an animation scheme directly related to simplex mesh structures that simulates a rotational joint at a given contour. A twenty degrees of freedom hand model was thus created from the mesh as Figure 12. Current limitations include the lack of a continuous surface representation associated with a simplex mesh and a computational cost too high for interactively manipulating large models.

Appendix A

The relation between L, r_i, d_i, ϕ_i is shown in Figure (8). If we write θ_U as $\angle UP_iF$ and θ_V as $\angle VP_iF$ then :

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

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

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

Finally,

$$L(r_i, d_i, \phi_i) = \frac{(r_i^2 - d_i^2).\tan(\phi_i)}{\epsilon.\sqrt{r_i^2 + (r_i^2 - d_i^2).\tan^2(\phi_i)} + r_i} \tag{8}$$

$$\begin{array}{ll} \epsilon = 1 & if \quad |\phi_i| < \pi/2 \\ \epsilon = -1 & if \quad |\phi_i| > \pi/2 \end{array}$$

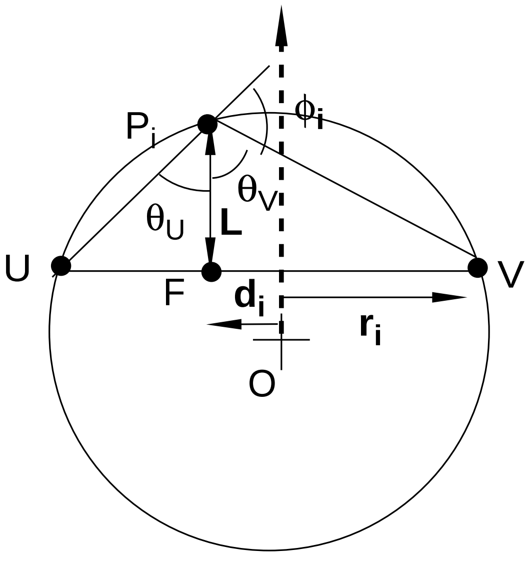

Figure 8: Computation of $L(r_i, d_i, \phi_i)$

References

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

[2] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Computer Graphics (SIGGRAPH'91)*, pages 257–266, July 1991.

[3] J. Chadwick, D. Haumann, and R. Parent. Layered construction for deformable animated characters. In *Computer Graphics (SIGGRAPH'89)*, pages 243–252, July 1989.

[4] K. F. D. Terzopoulos. Deformable models. *The Visual Computer, 4,6*, pages 306–331, 1988.

[5] H. Delingette, M. Hebert, and K. Ikeuchi. Energy functions for regularization algorithms. In *Geometric Methods in Computer Vision*, San Diego, August 1991. SPIE.

[6] 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.

[7] G. Farin. *Curves and Surfaces for Computed Aided Geometric Design.* Academic Press, 1989.

[8] J.-P. Gourret, N. Magnenat Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. In *Computer Graphics (SIGGRAPH'89)*, pages 21–30, July 1989.

[9] H. Hagen and G. Schulze. Automatic smoothing with geometric surfaces patches. *Computer Aided Geometric Design*, 4:213–236, 1987.

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

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

[12] E. Mehlum. Nonlinear splines. *Computer Aided Geometric Design*, pages 173–207, 1974.

[13] G. Miller and A. Pearce. Globular dynamics a connected particle system for animating viscous fluids. In *Computer Graphics (SIGGRAPH'89), Course 30 notes: Topics in Physically-based Modeling*, pages R1–R23, Boston, MA, August 1990. ACM.

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

[15] H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. In *Computer Graphics (SIGGRAPH'91)*, pages 339–347, July 1991.

[16] 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.

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

[18] D. Terzopoulos. Computing visible-surface representation. Technical Report Artificial Intelligence Memo 800, M.I.T., 1985.

[19] D. Terzopoulos, J. Platt, and K. Fleisher. From goop to glop: Heating and melting deformable models. In *Graphics Interface*, pages 219–226, June 1989.

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

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

Figure 9: Vase created from a cylinder with five contours

Figure 10: (a) Initialization of palm and fingers models; (b) All meshes are fit to the first range image; (c) Final model; (d) Texture display

Figure 11: Depth Cued, Solid and textured display of bust model.

Figure 12: Articulated hand

Figure 13: Metamorphism