In Journal of the Brazilian Computer Society 3(3): 52-64, April 1997

Homeomorphisms and Metamorphosis of Polyhedral Models Using Fields of Directions Defined on Triangulations

Marcelo E. Kallmann
Computer Graphics Lab.
Lausane - Switzerland - CH-1015
kallmann @lig.di.epfl.ch

Antonio A. F. Oliveira
Programa de Sistemas e Computação - COPPE/UFRJ.
21945-970 - CP:68511 - Rio de Janeiro - Brazil.
oliveira@cos.ufrj.br

Abstract *Many approaches have been proposed to generate the shape interpolation or morphing of two polyhedral objects given in a facet based representation. Most of them focus only the correspondence problem, leaving the interpolation process to just an interpolation of corresponding vertices. In this article we present a new approach which uses fields of directions defined on triangulations(FDTs) to treat both the problem of getting an homeomorphism between the models and that of morphing them. Consider that an scaled version(P_1) of one of the objects, has already been adequately placed in the interior of the other(P_2). The objective of the first part of the approach, is to obtain a field of unit vectors defined on a triangulation of the space between P_1 and P_2. This field must have no singularities and the trajectories determined by it will be later used to get warping and morphing transformations between P_1 and P_2. The morphing transformations obtained have the good property of being topology preserving ones but it can be hard to get an FDT defined on a triangulation of P_1 - P_2 and the intermediate models can have a very large number of faces. To illustrate those aspects, transformations between simple models are presented.*

Keywords: *Morphing, Shape Interpolation, Boundary Representation, Triangulation.*

1 Introduction

Morphing two objects A and B, consists in obtaining a sequence of intermediate objects which describes a continuous transformation between A and B. Morphing is also known as *shape blending, shape interpolation* and *metamorphosis.*

Morphing animations have achieved widespread use in the entertainment industry. In this context, the main goal is to generate good-looking transformations. Morphing techniques are also used in simulations of biological evolution processes and in the development of new products in industrial designing, as shown in [10]. In these cases, it is particularly desirable that all the intermediate objects generated are topologically consistent. This can really simplify some tasks, like the manufacture of prototypes from the computer generated models.

In a common approach to get a morphing of an object A into another(B), the problem is divided into two parts. The first one consists in obtaining an homeomorphism $H:A \rightarrow B$. The first step in the process of obtaining H is to solve the so called *correspondence problem*, what consists in matching the most prominent features of one object with those of the other.

The objective of the second part is to get a continuous transformation $T: A\times[0,1] \rightarrow R^{2,3}$ which takes every point $a \in$ A into H(a) . This part is sometimes referred as the *interpolation problem*.

Several approaches to the morphing problem when the objects are polyhedral models have already been published. Some of them are based on the idea of refining the two models until they have a common topology and then interpolating the positions of corresponding vertices, either linearly or in other simple way. These approaches differ however, in how they try to obtain common topology refinements of the original models:

*i)*Hong et al. [12] tries to match the faces of one model with those of the other taking into account the distance between the centroids of the faces. To make a complete matching possible, degenerated faces are included to equalize the number of them in the two models. In each pair of faces matched, the number of vertices is also equalized to simplify a posterior interpolation process.

*ii)*Bethel & Uselton [5] also add vertices and faces to two polyhedral models until a common topology is achieved. In this case, however, this is done by using adjacency graphs of the models.

*iii)*Kent et al. [7] describe a method for the case where the models represent polyhedra of genus zero. Both models are initially projected onto a sphere. Then, the arrangement of the edges of the two models projections , is determined. By projecting that arrangement back onto the polyhedra, one can obtain two new models for them with the same topology.

The strategy employed by other approaches is of a different kind

*i)*Kaul & Rossignac [1] constructs the Minkowski sum of gradually scaled versions of two given polyhedra to obtain a continuous transformation between them. This is done by scaling one model gradually from 100% to 0% while simultaneously scaling the other from 0% to 100%. Since the Minkowski sum of two polyhedra is always another polyhedra, it is possible to obtain consistent intermediate models. However, this is not an easy task in a surface based representation. Moreover, it is not possible to guarantee the preservation of the topology during the transformation. Intermediate polyhedra can have more genus than the originally given ones.

*ii).*Others approaches use models based in a volumetric representation, avoiding a lot of geometric difficulties the precedent ones have. However, in most cases, a heavier computation is required. Also, additional difficulties arise when one wants to retrieve topological information from those models. Some works that make use of volumetric objects can be found in [2], [6] and [11].

None of the proposals listed above can guarantee that only intermediate models without self-intersecting faces will be generated . This confirms the intricacy of the morphing problem, and shows that finding a general solution for the problem is a very difficult task.

In this paper, it is proposed a new approach to obtain homeomorphisms and morphings of a class named *continuous dilatation*s, between two polyhedral models of genus 0. That approach guarantees topology preserving transformations and consistent intermediate models for a considerably greater set of cases when compared with precedent works.

Related theoretical results for the 2-dimension case can be found in [3] and [13].

The article is organized as follows. Some fundamental definitions and notations are given in section 2. Formal definitions and other considerations about field of directions defined on triangulations(FDTs) are presented in section 3. Section 4 presents an algorithm to get an homeomorphism between two models and outlines two morphing processes using FDTs. Section 5 focuses on the problem of shelling a triangulation while section 6 analyzes specifically that of preserving the topology during the transformations. Section 7 discusses the results obtained and suggests some topics for future research.

2 Definitions and Notation

A set $P \subseteq R^n$ is a *kD-manifold*, $k \leq n$, iff any $p \in P$ has a neighborhood V_p such that $V_p \cap P$ is homeomorphic to the k dimensional open sphere.

Let S be a set of points in $\Re^d$ and H the convex hull of S. A *triangulation* T of S is a set of non-degenerated simplices of dimension d which additionally satisfies the following properties:

- A simplex in T has all vertices in S.
- Two different simplices in T have disjoint interiors.
- A facet of a simplex in T is either on the boundary of H, or is shared by exactly two simplices.
- A simplex in T contains no points of S other than its vertices.

Our interest is limited to 2 or 3D triangulations., the last ones also called tetrahedralizations. Moreover, although the concepts of model and polyhedron given below, can be defined in an arbitrary dimension, we will do that only in R^3. 2D versions of that concepts can be straightforwardly obtained from the 3D ones.

We call a *model* a finite set of non-degenerated triangles in R^3 such that every edge of a triangle is shared by exactly another triangle and no subset of triangles has the same property. We are only interested in models such that the union of its triangles is a 2D-manifold. In this case, that union will be called a *simple closed polyhedral surface* and the set it delimits a *simple polyhedron*. Since all polyhedral surfaces and polyhedra referred hereafter are simple, we will not mention that fact explicitly anymore.

The term "*mesh*" is sometimes used to refer to a model. According to the definitions above, a model is a boundary representation*(B-Rep)* scheme for a polyhedron.

The *topology* of a model is described by its simplicial complex vertex/edge/face. The *geometry* of a model is the instance of its topology determined by the coordinates of its vertices. Vertices, edges and faces are called *topological elements*.

Two models are said *homeomorphic* if there exists an homeo*morphism*. between the closed polyhedral surfaces defined by them.

Two models will be called *equivalent* if they represent the same object, even though their vertex/edge/face simplicial complexes are different. For example, if some faces of a given model M are sub-divided into smaller co-planar faces, another model, equivalent to M, is generated. That model is said a *refinement of M.*

Let M_T be the set of all models with the topology induced by the *Hausdorff* metric. The problem of determining a morphing between two given models M_i and $M_f \in M_T$, consists in obtaining a continuous *function* m:[0,1]$\rightarrow M_T$, such that, $m(0)$ and $m(1)$ are equivalent to M_i and M_f., respectively. $m(0)$ will be called here the *initial model* and $m(1)$ the *final model*. If $t \in (0,1)$, $m(t)$ will be said an *intermediate model*.

Given a model M and a triangulation T, T is said to be *constrained by* M if every face of M is also a face of a simplex in T. Given a model M, it may not be possible to obtain a triangulation constrained by it(See [8])..However, there is always a refinement of M for which this is possible.

.

If in the definition of a triangulation of a finite set of points, we replace the convex hull H by the boundary of a polyhedron *P* containing all the points, we will define a *triangulation of P*.

A non-oriented polygonal line whose sequence of vertices is $v_1, v_2, \ldots, v_k$ will be noted $[v_1, v_2, \ldots, v_k]$. To refer to the oriented version of it we will write $[v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k]$. int(S) will refer to the interior of a set S.

3 Fields of Directions Defined on a Triangulation

A *field of directions defined on a triangulation (FDT)* is simply a function that associates each simplex of a triangulation $T \subseteq \Re^d$ with a direction in $\Re^d$.

. Let D be a FDT defined on a triangulation T of a set $U \subseteq \Re^d$, *t* a simplex of T and *p* some point in *t*. Also let $[p_b, p_e]$ be the intersection of *t* with the line parallel to D(*t*) passing by *p*. If $p_b \neq p_e$ assume that $p_b - p_e$ has the direction of D(t). In this case, the *trajectory in t through p* is the oriented segment $p_b \rightarrow p_e$. Otherwise it is only the point p and is called degenerated. At most one of the trajectories in t through its vertices is non-degenerated.

Consider the case where d=3. If a tetrahedron *t* has a vertex *v* such that the trajectory in t through v is non-degenerated, then t can be classified as:

i) A vertex-face tetrahedron when the trajectory in t through *v* begins at *v* and ends at some point on the face opposite to *v*.

ii) A face-vertex tetrahedron if f the *t-trajectory* through *v* starts at some point on the face opposite to *v* and ends at *v*.

If the trajectory in t through all its vertices is a single point, then there is a point p on an edge *e* of *t* such that the t-trajectory starting at p ends on the edge opposite to e. For thar reason, in this case, t is said an *edge-edge* tetrahedron.

If d=2, there are only two classes of triangles: *vertex-edge* and *edge-vertex* ones. The definition of those classes is similar to those of vertex-face tetrahedra and face-vertex tetrahedra, respectively.

Figure 1 illustrates the three types of tetrahedra in relation to the trajectories through their vertices.

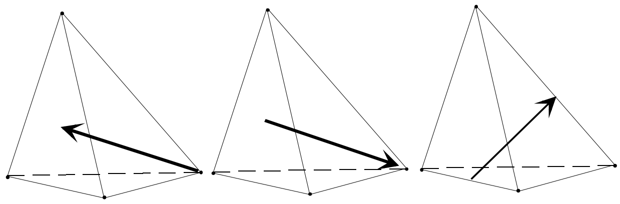


Figure 1 - From left to right: a vertex-face tetrahedron, a face-vertex one and an edge-edge one.

Given, two subsets of T, V and V', we will note $V <_D V'$ if there exists a trajectory induced by D that meets V prior to V'. If $V <_D V'$ or all trajectories intersecting both V and V' meet them at the same time we note $V \leq_D V'$.

A topological element *e* of a simplex *t* is said an *input element* if $e <_D int(t)$ and an *output* one if $int(t) <_D e$. If *e* is neither an input element of *t* nor an output one, it is said to be *tangent* to t. We observe that, in this case, the trajectories in *t* through the points on *e* are entirely contained in a facet of *t*.

For example, a vertex-face tetrahedron has three input faces and one output face. A face-vertex tetrahedron has one input face and three output faces while an edge-edge tetrahedron has two input faces and two output faces. We will represent the union of all input facets of a simplex *t* by IN(t) and that of all output faces of *t* by OUT(*t*).

A *trajectory induced* by D or D-trajectory is a maximal element of the set of oriented polygonal lines $\gamma \subseteq U$ having the property that if γ intercepts a simplex *t* of T then $\gamma \cap t$ is a trajectory in t .

In the remainder of this section, our interest will be limited to FDTs defined on three-dimensional triangulations. Let S_0 and S_1 be two closed polyhedral surfaces delimiting polyhedra P_0 and P_1 respectively. Assume that $int(P_0) \supseteq P_1$. Hereafter U will refer to $P_0 - int(P_1)$ and will be called the *region between* S_0 and S_1.

A FDT D defined on a tetrahedralization T of U is said to be *admissible*, iff it satisfies the following properties:

- All D-trajectories start on S_0 and end on S_1.
- Two D trajectories have no common points.

The following lemma gives another way of characterizing an admissible FDT..

Lemma 1 :

If a FDT D satisfies conditions *i-iii* below, D is admissible. Conversely, if D is admissible and $\forall$ t $\in$ T, D(t) is not parallel to a face of t, then *i-ii* are implied.

i) Each vertex and edge in the interior of U is an input element for a unique tetrahedron and an output element for only another one.. They are tangent elements to all other tetrahedra containing them.

ii) Each vertex and edge on S_0 (alternatively: on S_1) is an input (output, respectively) element of a single one tetrahedron and tangent to all the others containiing it.

iii)The relation $\leq_D$ is a partial order on the set of tetrahedra in T.

Proof:

($\Rightarrow$)

Two D-trajectories can only intersect on a topological element of T. If they meet at a point on the relative interior of a face in int(U), let t_1 and t_2 be the tetrahedra adjacent to that face. One of the trajectories reaches the face from t_1 and and the other from t_2 . Hence, we have simultaneously that $t_1 \leq_D t_2$ and $t_2 \leq_D t_1$ contradicting *iii*. Clearly, two D-trajectories cannot meet in the relative interior of a face on U border .

Also, conditions *i and ii* avoid that a trajectory meets at a vertex or on an edge of T. In view of all that, any two D-trajectories must be disjoint.

Condition *iii* determines that a D-trajectory cannot intersect a tetrahedron twice. In consequence, all D-trajectories cannot be closed and must have a finite number of segments. Thus, any trajectory must have an initial and a final point. Then, condition *i* determines that these extreme points cannot be vertices or edge points in int(U). Also, condition *iii* percludes that these points are in the relative interior of a face in int(U). Finally condition *ii* determines that all D-trajectories go from points on U_{out} to points on U_{in}. and not the opposite

(Ü)

Let D be an admissible FDT, *v* a vertex of T in the interior of U and Γ_v the D-trajectory through *v*. Let t_1 and t_2 be respectively, the last tetrahedron traversed by Γ_v before it reaches *v* and the first tetrahedron crossed by it after *v*. As $\forall$ t $\in$ T, D(t) is not parallel to a face of t, t_1 and t_2 are well defined. *v* will be an output element of t_1 and an input element of t_2. If it is a non-tangent element of any other tetrahedron there will be two D-trajectories meeting at *v* what contradicts the fact that D is admissible.

If e is an edge in int(U), a similar argument can be applied considering that all D-trajectories intersecting the relative interior of e come from(or leave to) the same adjacent tetrahedron.•

An admissible FDT which satisfies condition *iii*_above will be called *acyclic*. Hereafter we will assume that an admissible FDT is also acyclic. In fact, all the results that will be presented in the following sections require that condition. Also, for sake of simplicity we will suppose that the direction assigned to a tetrahedron is not parallel to any of its faces.

Let Π_t: IN(*t*)→OUT(*t*) be the function which takes every starting point of a trajectory in t into the end point of that trajectory. Π_t will be referred as the D-projection through t and Π_t^{-1}, the inverse D-projection through t.

Let *s*=[*abc*], be a triangle contained in an output face of *t* and $T_t^{-1}(s)$ a 2d-triangulation of $\Pi_t^{-1}(s)$.

Figures 2 and 3 illustrate some possibilities of the triangulation $T_t^{-1}(s)$. In those figures, the direction assigned to the tetrahedron is always that of the vector *y*-*x*.

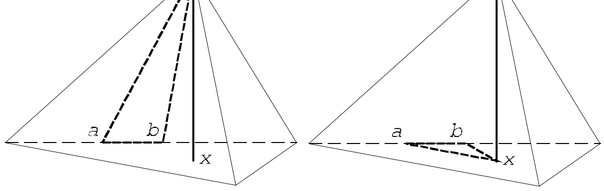


Figure 2 - The inverse projection of triangle [*abc*] in a face-vertex tetrahedron *t* where D(t) is vertical. This projection generates a single triangle [*abx*].

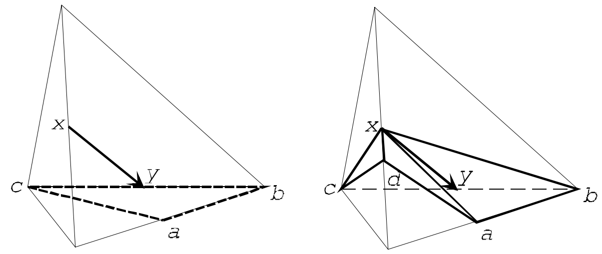


Figure 3 - The inverse projection of triangle [*abc*] in an edge-edge type tetrahedron *t* where D(*t*) = *y* - *x*. This projection consists of three triangles [*abx*], [*adx*] e [*cdx*].

Figure 3 shows the inverse D-projection of a triangle in an edge-edge type tetrahedron. The part of this projection on one of the faces is the quadrilateral [*abxd*] which must be triangulated in the construction of T_t^{-1}(*[abc])*

Let S_0 and S_1 be as defined above and M_0 and M_1 be models representing S_0 and S_1 respectively. Let s_1 be a triangular face of M_1 and t_1 the tetrahedron incident to s_1. Make $M^*(s_1) = \varnothing$, apply Π_{t1}^{-1} to s_1 and obtain $T_{t1}^{-1}(s_1)$. Then, for each triangle s' in $T_{t1}^{-1}(s_1)$ and not on S_0 , let *t*' be the other triangle containing it, apply $\Pi_{t'}^{-1}$ to s' and obtain $T_{t'}^{-1}(s')$. If $T_{t1}^{-1}(s_1)$ has triangles on M_0 add them to $M^*(s_1)$. Repeat the process above for every triangle in $T_{t'}^{-1}(s')$ and so on. At the end of that process $M^*(s_1)$ will be a 2D triangular mesh covering a connected region of S_0 . The union of all the meshes obtained by applying the process above to all faces of M_1, is a mesh M* covering S_0. That is, M* and M_0 are equivalent.

Let *v* be a vertex of M* and H(*v*) the end of the D trajectory that starts in *v*. Determining the set {H(*v*) : *v* is a vertex of M*} can be made by incorporating to the process described above a projection/interpolation procedure. Having the correspondence function H, one can get an homeomorphism between S_0 and S_1 by interpolation.

That process gives a natural way to obtain a metamorphosis between S_0 and S_1. of a kind called a continuous dilatation . A continuous dilatation has the good property of preserving the topology during the transformation. However it has also some major inconveniences. One of them is the fact that it requires that an object is contained in the other. Of course, it is always possible to scale one model so that it fits inside the other. However, this must be done in a way that transformations of satisfactory blending quality can be obtained what is not trivial at all. Another major drawback is the fact that it can be hard to obtain a transformation which takes a predefined set of features of the first object into one of the second. The more sophisticated sketch described in section 4 has to be employed to deal with this problem.

Hereafter, For the sake of commodity, we will identify a model with the polyhedral surface it represents So, a model will be both a set of triangles and the union of them.

4 Obtaining Homeomorphisms and Metamorphosis.

Let M_i and M_f be two models representing polyhedra of genus zero. The following sequence of steps obtains a homeomorphism between them:

Algorithm 1 - "Obtaining an Homeomorphism between two Models"

STEP 1. Apply a rigid motion T_R followed by a scaling transformation T_S to M_f , making the resulting model M_f' be properly enclosed by M_i.

STEP 2. Obtain a triangulation T of the region *U* between M_i and M_f', where an admissible FDT D can be defined. This may require the insertion of additional vertices. In this case, refine M_i and M_f' to include these new vertices. Call the resulting models M_0 and M_1.

STEP 3. Let T_0 be a copy of T and for each face *f* of M_1, make M(*f*) = *f*. Also, for each vertex *v* of M_1, make H(*v*)=*v*. Then execute :

While $T_0 \neq \emptyset$:
{Let *t* be a tetrahedron which is $\leq_D$ -maximal in T_0.
For each input face f_i of *t* do:
{For each output face f_o of *t* do:
{$\forall$ vertex *v* of M(f_o) such that $\Pi_t^{-1}(v) \in f_i$,
make H($\Pi_t^{-1}(v)$) = H(*v*) ;
Make M'(f_o) = Π_t^{-1}(M(f_o)) $\cap f_i$. ;
$\forall$ vertex *v*' of M'(f_o) such that H(*v*') has not yet been determined do:
{Find *v*'' = Π_t^{-1}(*v*').
Determine the triangular face of M(f_o) where *v*'' lies;
Let {u_i, i=1,2,3} be the vertices of Ψ and {α_i, i=1,2,3} the baricentric coordinates of *v*'' relative to Ψ.
Make H(*v*')=$\Sigma\alpha_i$H(u_i), i=1,2,3;
} *end-for*
}*end-for*
Obtain M(f_i) by aggregating the models M'(f_o), obtained for every output face f_o of *t*.
} *end-for*
Remove *t* from T_0.
} *end-while*

STEP 4. Obtain M by aggregating all M(*f*) where *f* is a face of M_0. M will be called the unified model. It is equivalent to M_0 but replacing every vertex *v* of it by H(*v*), it becomes equivalent to M_1.

STEP 5. By linear interpolation extend H to the whole M_0. That extension defines a homeomorphism between M_0 and M_1.

The following paragraphs analyze each task performed by the algorithm above.

1. A lot of articles have been published about optimally positioning a 2 or 3d-object in relation to another (See, for instance, [14]). However, in the specific case of the algorithm above, we consider that it is not worthwhile to automate this task. It is clear that better results can be obtained if similar features of the objects are placed close to each other. But measuring that similarity and weighting the importance of each feature in the quality of the final result is not so simple,specially considering that there are some subjective aspects involved. For that reason we have assumed that adequately choose the transformations T_R and T_S is the user's responsibility.

2. Constructing a tetrahedralization of a polyhedral region, is a complex problem which has several alternative approaches. The one of least worst case complexity is Palios and Chazelle's [15] which is O((n + r^2) log *n*), where *n* and *r* are respectively the total number of vertices and that of reflex ones. For practical reasons however we have implemented another approach. In that approach a Delaunay triangulation of the two models vertices is first obtained. After, this triangulation is refined until it becomes constrained both by M_0 and M_1. A detailed description of that approach is given in [9].

3. Defining an FDT on a given triangulation T is in general not difficult, but can also be laborious and even impossible. This problem is the object of section 5 presented next.

4. Let e and e' be any two edges of T. Define $\mu(e,e')$ as the number of connected components of the intersection of *e*' with the set of points $\geq_D e$. Let μ be the sum of $\mu(e,e')$ over all pairs of edges of T and *m* be #(T). The number of triangles generated in step 3 is O(mμ) what can be very high even in examples of mild dimension. Although other tasks are also time consuming, the performance of the algorithm is considerably related to the time spent in step 3. To reduce this time the mesh can be simplified after each projection to avoid that every vertex that is encountered or created gets mapped to a vertex of M*.

5. The fact that we require that the models have only triangular faces do not increase the algorithm time complexity. Observe that the refinement made in Step 2 consists in introducing in the models, triangles which are already faces of tetrahedra in T. Hence, all we need to get that refinement is to incorporate to the algorithm used to construct T, a procedure for identifying its triangular faces which are in $M_i \cup M_f$. Of course that can be done in constant time per face.

Hereafter tpm will stand for topology preserving morphing.

Having the FDT D, continuous dilatation processes that morphs M_1 into M_0 can be obtained. Let *p* be a point on M_0 and t_p the D-trajectory starting at *p* and ending at H(*p*). Refer by L(*p*) to the total length of t_p and define $H_w(p)$ as the point on t_p whose distance to *p* along t_p is *w*L(*p*). Finally, define M(*w*), $w \in [0,1]$, as U{$H_w(p)$, $p \in S_0$}. M($\cdot$) is a tpm between M_0 and M_1. A computational procedure to obtain M(w) is givien in section 6. Define M'(*w*) as the model obtained from M_0 by replacing each one of its vertices *v* by $H_w(v)$. We must observe that $w \in [0,1] \rightarrow$ M'(w) may not be a tpm.

Now, let $w \in [0,1] \rightarrow$ N(w) be any tpm between M_0 and M_1. Taking into account that M_f has been possibly moved, rotated and scaled in step 1, to get a tpm between the original given models define $N^*(w) = [(1-w)Id + wT_R^{-1}T_S^{-1}]\ N(w)$. If T_S^{-1} is replaced in that expression by an adequate scaling transformation, whose amplitude depends on *w*, volume preserving morphings can be obtained.

Now consider the problem of obtaining a tpm between two models which takes features in one of them into corresponding ones in the other. When these features are topological elements of the models, the approach described below can be tried.

Without lack of generality we can suppose that M_0 and M_1 have the same topology. It will be also assumed that a complete correspondence between the topological elements of them have already been established.

First of all, the two models are positioned so that one intersect the other properly. Then, let U be the region delimited by two convex approximately spherical polyhedral surfaces, one (U_{out}) enclosing both models and the other (U_{in}) contained in the intersection of them. Get two triangulations of U (T_0 and T_1) such that T_i is constrained by M_i, i=0,1. To do that, possibly new vertices have to be inserted on the faces of M_i, i=0,1. So, introduce in M_i, i=0,1, the topological elements that have to be created in function of those insertions. Additional elements can be necessary to make the resulting models M'_0 and M'_1 have the same topology. Obtain H', a face-to-face homeomorphism between M'_0 and M'_1 which also takes each face of M_0 into the one of M_1 corresponding to it.

Now, define on T_i, an admissible FDT D_i which is transversal to M_i, i=0,1. Continuously transform D_0 into D_1, in such way that any intermediate FDT obtained (D_w, $w\in[0,1]$) is also admissible . Each D_w, $w\in[0,1]$ induces a system of reference for the points in U where the coordinates of a point *p* are the following:

1) The coordinates in a parametrization of U_{out} of the point $s_w(p)$, where the D_w-trajectory through *p* starts.
2) $d_w(p)$, the distance along that trajectory between $s_w(p)$ and p divided by the total length of the trajectory.

Define $H_{out} : U_{out} \to U_{out}$ as the function which takes $s_0(p)$ into $s_1(H'(p)), \forall\ p \in U$. A morphing between M_0' and M_1' can, thus, be obtained by defining M'_w as the set of points whose coordinates in the system induced by D_w are :

$$(I_{out}(w,s_0(p)), (1-w)d_0(p) + w.d_1(H'(p))),$$

where:

i) *p* is a point on M'_0.

ii) I_{out}: $[0,1]\times U_{out} \to U_{out}$ is a continuous function such that:

ii.1) $I_{out}(w, .)$ is a homeomorphism $\forall\ w\in[0,1]$.

ii.2) $I_{out}(0,p)=p$ and $I_{out}(1,p)= H_{out}(p)$, $\forall\ p \in U_{out}$.

This approach is considerably more general than the previous one, but its implementation in 3D has difficulties of different sorts. Before commenting these difficulties we must observe that the 2D version of that approach is plausible. The main result supporting that assertion is the following:

Lemma 2:

Let U be a polygonal bordered annular region in the plane and D_0 and D_1 two admissible FDTs defined, respectively, on Triangulations T_0 and T_1 of U. Assume that T_0 and T_1 have both *n* vertices and that those on the border of U are the same for the two triangulations. Then, there is a continuous function D:[0,1]→{admissible FDTs defined on a triangulation of U} such that:

i) $D(0) = D_0$ and $D(1) = D_1$.

ii) There is a sequence ($I_k=[a_k,a_{k+1})$, k=1,...,K) of $O(n^3)$ disjoint semi-open intervals covering [0,1], which satisfy the following conditions:

ii.a) Let T_w be the triangulation on which D(w) is defined. For all w in I_k, T_w has the same topology.

ii.b) Let $w \in I_k$. Knowing $D(a_k)$ and $D(a_{k+1})$, we can determine D(w), by means of an interpolation process which is O(1) for triangle.

The proof of Lemma 2 is quite long and can be found in [17]. It has not been possible up to now, to extend it to 3D. In fact, obtaining a morphing of two admissible FDTs even if they are defined on the same 3D-triangulation of U, is a hard problem whose complexity is unknown. Another specific difficulty of that approach is the fact that obtaining I_{out} means to find a 2D tpm between the topological elements of two models equivalent to M_0. Notice that this can be considerably more difficult than morphing two simple polygons. Moreover, in this case we have a time-varying systems of trajectories instead of a fixed one as in the continuous dilation case. Even in 2D, this determines that more complex intermediate models are eventually generated.

Due to the high complexity of the 3D version of that approach we have not implemented it yet. Some experiments in 2D can be found in [17].

5 Obtaining Admissible FDTs.

Let, once more, M_0 and M_1 be two models representing the boundary of polyhedra P_0 and P_1, where P_0 encloses P_1. Also, let U be the region between them. The process of obtaining an admissible TFD defined on a triangulation T, if there is one, will be called here a *topological excavation* or a *shelling* of T. It can be implemented in the following way:

Algorithm 2 - “Shelling T”

1. First all tetrahedra in T are labeled as *full*. The label full indicates that the field has not yet been defined in the tetrahedra . Those where the field has already been defined will be labeled as *empty*.

2. Make $F_f = M_0$. F_f represents the boundary of the full tetrahedra region.

3. While there are full tetrahedra, select those whose intersection with F_f is a non-empty 2d-manifold. Call those tetrahedra *removable*.

3.1. If there is no removable tetrahedron among the ones labeled as full then either apply a backtracking procedure or change the triangulation locally. That backtracking procedure can be a recursive enumeration one and the strategies for changing or refining the triangulation in order to create removable tetrahedra are predominantly heuristic and will not be presented here. A situation where a backtracking is necessary is shown in figure 4.

3.2. Otherwise, choose a removable tetrahedron *t*, label it as empty and associate a direction D(*t*) to it. *t* will be a face-vertex, edge-edge or vertex-face tetrahedron, according to t has one, two or three faces on F_f. In any case D(*t*) will be the direction of an oriented segment (p_i, p_o) ,where:

- p_i is a point in the relative interior of the topological element of *t* which is common to all its faces in F_f.
- p_o is a point in the relative interior of the opposite element.

3.3. Update F_f,. When this is done, it is possible that some removable tetrahedra become non-removable and vice-versa. Thus, it is necessary to update the classification of all tetrahedra labeled as full which have an element in common with *t*.

Suppose we have a shellable triangulation T with n tetrahedra. To get an O(n) algorithm to obtain a shelling of T, assuming that no backtracking is necessary, we have to slightly modify the algorithm above as follows.

Consider that we have a list (L) which contains all removable full tetrahedra and possibly, some non-removable. To pick a removable tetrahedron, we take one (t) out of L and check if it is removable. This can be done in O(1) time if we have marked all topological elements of T which are contained in F_f. If t is not removable, take another tetrahedron from L, repeat the test and so on.

Now, observe that if in the process to get a shelling of U there no backtracking is necessary, the classification of a full tetrahedron can change from removable to non-removable at most twice. Hence, the number of times tetrahedra are taken out of L is bounded by 2n.

For an empty or non-removable tetrahedron become removable when another *(t)* is made empty, it must be adjacent to that other. Thus, to make L maintain the property of containing all removable tetrahedra after *t* is made empty, we only need to determine which of the tetrahedra adjacent to t must be added to L. Hence, updating L after a triangle is made empty takes O(1) time.

In view of the above, if no backtracking or re-triangulating is necessary, the shelling algorithm is O(n).

Figure 5 pictures a 3D FDT obtained by employing the shelling algorithm given above.

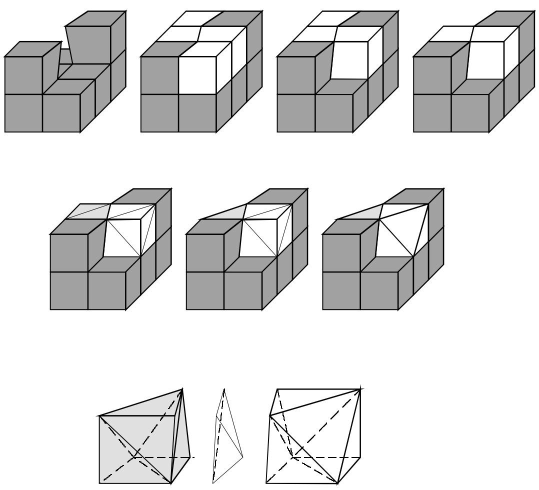


Figure 4 - A situation where a backtracking is necessary. From left to right in the first row, we have P_1 and three instances of a shelling process. The second row presents three other posterior instances of that process. The tetrahedra labeled as full in the situation depicted in the rightmost figure of that row are represented in the third row. They are all non-removable.

Consider that P_0 is a simple convex polyhedron. Even in this case we can choose P_1 in such a way that no polynomial triangulation of U has a shelling. An example of that can be obtained from the trivial knot whose spanning disk has exponential size given by Snoeyink in [18]. This fact implies that the time complexity of the shelling algorithm above, if backtracking or re-triangling is necessary, can be also exponential. Our experience, however, indicates that the non-existence of a removable tetrahedra is not usual, although it occurs.

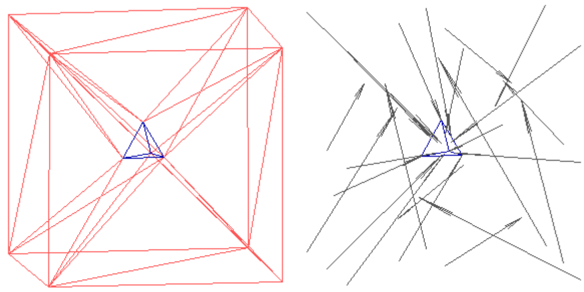


Figure 5 - A tetrahedralization between a cube and a tetrahedron and the directions obtained by an implementation of algorithm 2

In contrast with the 3D case, any triangulation of a 2D annular region with polygonal borders admits a topological excavation. Of course to prove that, it is sufficient to consider the case where the vertices are on the border of the polygons. Lemma 3 bellow, treats that case.

Lemma 3 - Let P_0 and P_1 be two planar polygons such that $P_1 \subseteq \mathrm{int}(P_0)$ and U the region between P_0 and P_1. whose vertices are on $P_0 \cup P_1$ Then it is always possible to obtain an admissible FDT defined on T.

Proof:

Let S_3 be the set of triangles which have all three vertices on the same polygon and $S_{21,}$ the set of those with vertices on both polygons. The triangles with vertices on S_{21} form a cycle (t_k, k=0,...,m-1).

Let t be a triangle with all vertices on P_j, $j \in \{0,1\}$. Define *v* as the only vertex of *t* which is not adjacent to any other triangle of T. Assign to *t* the direction of $(-1)^j(v\text{-}p)$, where *p* is a point in the relative interior of the edge opposite to *v*.

Define v_k as the only vertex of t_k which also belongs to $t_{(k-1)\mathrm{mod}\ m}$ but not to $t_{(k-2)\mathrm{mod}\ m}$. Let $P_{i(k)}$, $i(k) \in \{0,1\}$ be the polygon containing v_k and p_k be any point on the edge of t_k opposite to v_k. Assign to t_k the direction of $(-1)^{i(k)}(p_k - v_k)$.

If *v* is a vertex of a single triangle t of T then $t \in S_3$ and *v* is assigned to t. Now, suppose that *v* belongs to more than one triangle. Then:

i)v is adjacent to a sequence $S_v = (t_i, t_{i+1}, \ldots, t_{(i+j)\bmod(m)})$ of three or more consecutive elements of$(t_k$, k=0,...,m-1). According to the rule given above to define D in the set S_{21}, among the triangles in S_v only $,t_{(i+1)\bmod(m)}$ can be assigned to *v*.

ii) v cannot be assigned to one in S_3.

Thus, by Lemma 1, D is admissible.•

Figure 6 presents a sequence of partial results obtained when a 2D version of the topological excavation algorithm is applied. Observe that a triangle is associated to an edge-vertex direction if it has a single edge on the set of full triangles border at the moment it is made empty. Otherwise it will have two edges on that border and will be associated to a vertex-edge direction.

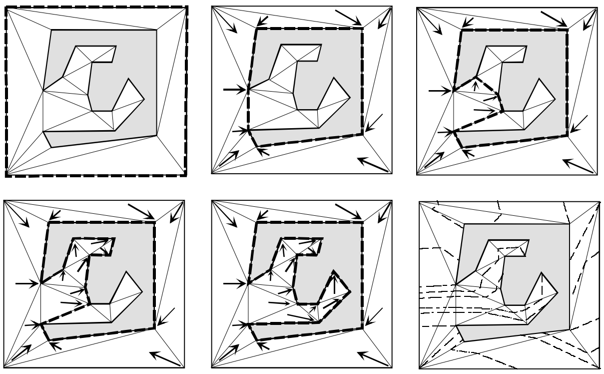


Figure 6 - The 2-dimension version of the topological excavation algorithm.

6 Obtaining M(w).

Having the admissible FDT obtained in step 2 of algorithm 1, the simplest way to get a tpm between M_0 and M_1 is to include in the step 3 of that algorithm, a simple procedure to morph IN(t) into OUT(t), where t is the tetrahedron picked from T_0 to be processed . At any moment of the morphing process so obtained , the transformation happens in a single tetrahedron. Specially when the tetrahedra of T have very different sizes, this can make the transformation have a very poor blending quality. In view of that, more "parallel" morphings, like M(·) defined in section 4, where the transformation can occur simultaneously in several tetrahedra, must be tried. The goal of this section is to present a computational process to construct the intermediate models (M(w)) of that morphing. For sake of commodity we repeat here the definition of M(w) given in that section.

"Let *p* be a point on M_0 and τ_p be the trajectory induced by D which starts at *p* and ends at H(*p*). Refer by L(*p*) to the total length of τ_p and define $H_w(p)$ as the point on τ_p whose distance to *p* along τ_p is *w*L(*p*). We Define M(*w*) as $\cup\{H_w(p), p\in M_0\}$."

Now, let *f* be a face of the unified model M. Define *tunnel of f* as the set $\{p\in U, p \geq_D f\}$

To get a more precise idea of how complicate an intermediate model must be to avoid containing topological artifacts let us consider the case where intermediate models M'(*w*) are obtained from M by replacing each one of its vertices *v* by $H_w(v)$. Figure 7 illustrates why the morphing obtained in that simpler way is not a topology preserving one.

Figure 7-left shows an FDT D defined on a triangulation T of the region U between the borders of a triangle and a square. The unified model M in this case is the polygon [a,b,c,d,e,f,g] whose realization coincides with the border of the square. The D-trajectories starting at vertices of M are also represented in that figure. Dashed lines are used to represent these trajectories. The polygon $M'_{0.5}$=[a',b',c',d',e',f',g'] is also pictured. Observe that the edge [g',a'] intersects properly the trajectory through a'. That is, [g',a'] is not constrained to the tunnel of [g,a]. It is just that situation that must be avoided because it can determine the occurrence of self-intersections. The figure 7-right pictures a situation where this effectively happens. Observe that this occurs because [c,d] is not contained in the tunnel delimited by the trajectories through c and d. Differently, between c and d, M_w is the polygonal line [c,x,y,d] which is completely contained in that tunnel.

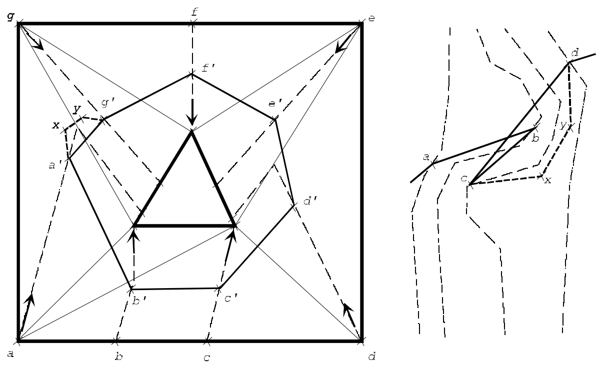


Figure 7 - A situation where connecting intermediate vertices without respecting the tunnels causes a self-intersecting intermediate polygon.

Back to the problem of constructing M(w), the following result will be needed:

Lemma 4 - Let *f* be a face of M and *t* a tetrahedron of T . Then, $\Lambda(f,t,w)$=(tunnel of $f \cap t \cap M(w)$) is planar.

Proof:

As no trajectory can have more than one point in $\Lambda(f,t,w)$ it cannot have interior. Then, it is sufficient to prove that for any pair of points, p_1 and $p_2 \in \Lambda(f,t,w)$ and any $\lambda\in[0,1]$, $\lambda p_1 + (1\text{-}\lambda)p_2$ is also in $\Lambda(f,t,w)$. Given a point *p* in U, let S(*p*) refer to the distance along the trajectory through *p* from its starting point to *p*. Then, as both S(.) and L(.) are linear in $\Gamma(f, t)$ = tunnel of $f\cap t$, we have that:

$S(\lambda p_1 + (1\text{-}\lambda)p_2) = \lambda S(p_1) + (1\text{-}\lambda)S(p_2) = \lambda w L(p_1) + (1\text{-}\lambda)wL(p_2) = w(\lambda L(p_1)) + (1\text{-}\lambda)L(p_2) = wL(\lambda p_1 + (1\text{-}\lambda)p_2)$, what means that $\lambda p_1 + (1\text{-}\lambda)p_2 \in \Lambda(f,t,w)$.•

From that lemma it is possible to derive M(*w*) by means of the non-optimized procedure given below:

Algorithm 3 - “Constructing M(w)”
For each face f of the unified model M do:
{ Choose a vertex v of f and determines $H_w(v)$.
Let t_0 be a tetrahedron containing $H_w(v)$.
Label t_0 as unsolved.
While there are unsolved tetrahedra choose one of them (t) and do:
{ Determine $\Gamma(f, t)$ = tunnel of $f \cap t$.
For each vertex u_i, i=1,...,k of $\Gamma(f, t)$,
find $S(u_i)$ and $L(u_i)$.
Determine $\Lambda(f,t,w)$ by solving:
$\Sigma(\lambda_i S(u_i)) = w\, \Sigma(\lambda_i L(u_i))$.
Label t solved.
For every unlabeled tetrahedra t' adjacent to t and intersecting $\Lambda(f,t,w)$ label t' unsolved.
} *end-while*
Erase the labels of all tetrahedra .
} *end-for*

A more detailed version of this algorithm for the 2D case is presented in [13].

7 Implementation Results and Future Research

Figure 8 shows the results obtained with the field and the tetrahedralization displayed in figure 5. Note the unified model obtained lying on the boundary of the cube.

This unified model has its vertices coordinates changed, by linear interpolation, until it gets the shape of the inner tetrahedron.

Figure 9 shows an example obtained with the 2D version of the algorithm, where a polygon with the shape of a car is gradually transformed into a boat.

Figure 10 compares between two morphings. The one above is obtained by using a simple linear interpolation between corresponding vertices. In the one below we have interpolated along the trajectories through vertices.

Finally, figure 11 shows the concatenation of many morphings between some simple models.

In several cases, the intermediate models become very complicated, with many small faces. This occurs when many projections of faces have to be made, in the step 3 of algorithm 1. When intermediate models are too wiggly, smoothing processes can be applied.

More than simply introducing an approach to get homeomorphisms or tpms between polyhedral models, this article intends to give a more precise idea about the complexity of these problems. They still require a lot of work to be fully understood.. Topics for future research are:

1. Determine the complexity of morphing two TFDs defined on 3D triangulations.
2. Establish conditions for a polyhedral region have a shellable polynomial triangulation .
3. Reduce the complexity of the projection process presented in the step 3 of algorithm 1.
4. Get efficient methods to obtain a tpm between two models representing the same convex polyhedral surface, which takes each face of one into a face of the other.
5. Answer the question: Is a triangulation whose vertices are on two parallel planes, always shellable ?

Acknowledgments

The authors would like to thank to Jorge Stolfi and to the anonymous referees for their valuable suggestions. We also acknowledge the CNPq for the support received through the GEOTEC/PROTEM project.

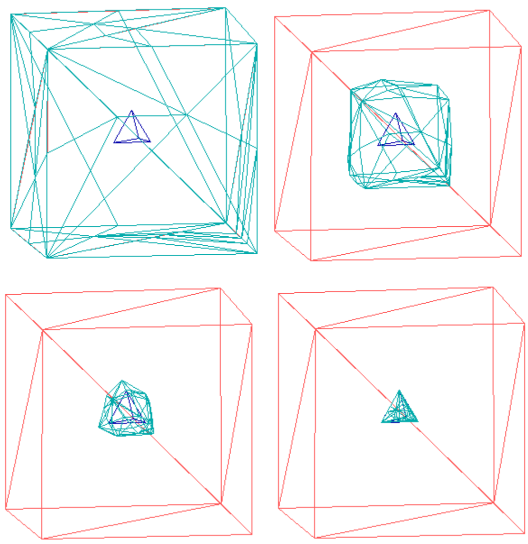


Figure 8 - A morphing obtained by interpolating between the vertices of the unified model and the end positions of their trajectories.

8 References

[1] A. Kaul and J. Rossignac, Solid-Interpolating Deformations: Construction and Animation of PIPs. In *Proceedings of Eurographics’91*, pages 493-505, September 1991.

[2] A. Lerios, C. D. Garfinkle and M. Levoy, Feature-Based Volume Metamorphosis. In *Computer Graphics (SIGGRAPH ’95 proceedings)*, volume 29, pages 449-456, 1995.

[3] A. Oliveira, S. Nascimento and S. Meerbaum, Morphing Fields of Directions Defined on Triangulations to Morph Simple Polygons. In *Abstracts of the 12th European Workshop on Computational Geometry (CG ’96 proceedings)*, pages 117-119, Münster, 1996.

[4] D. DeCarlo and J. Gallier, Topological Evolution of Surfaces. In *Proceedings of Graphics Interface'96*. pages 194-203, 1996.

[5] E. Bethel and S. P. Uselton, Shape Distortion in Computer-Assisted Keyframe Animation. In *State-of-The-Art in Computer Animation: Proceedings of Computer Animation'89*, (N. Magnenat-Thalmann and D. Thalmann, Eds.), pages 215-224, Springer-Verlag, 1989.

[6] J. F. Hughes, Scheduled Fourier Volume Morphing. In *Computer Graphics (SIGGRAPH'92 proceedings)*, volume 26, pages 43-46, July 1992.

[7] J. R. Kent, W. E. Carlson and R. E. Parent, Shape Transformation for Polyhedral Objects. In *Computer Graphics (SIGGRAPH'92 proceedings)*, volume 26, pages 47-54, July 1992.

[8] J. Ruppert and R. Seidel, On the Difficulty of Tetrahedralizing 3-Dimensional Non-Convex Polyhedra. In *Proceedings of the ACM Symposium on Computational*, pages 380-392, 1989.

[9] N. P. Weatherill, and O. Hassan, Efficient Three-Dimensional Delaunay Triangulation With Automatic Point Creation and Imposed Boundary Constraints. In *International Journal for Numerical Methods in Engineering*, volume 37, pages 2005-2039, 1994.

[10] S. E. Chen and R. E. Parent, Shape Averaging and its Applications to Industrial Design. In *IEEE Computer Graphics and Applications* , pages 47-54, 1989.

[11] T. He, S. Wang and A. Kaufman, Wavelet-based Volume Morphing. In *Proceedings of Visualization '94*, pages 85-91, 1994.

[12] T. Hong, N. Magnenat-Thalmann and D. Thalmann, A General Algorithm for 3-D Shape Interpolation in a Facet-Based Representation. In *Proceedings of Graphics Interface'88*. pages 229-235, 1988.

[13] A. Oliveira, S. Nascimento, and S. Meerbaum, Using Fields of Directions Defined on a Triangulation to Obtain a Topology Preserving Continuous Transformation of a Polygon into Another. In *Proceedings of VIII SIBGRAPI*, pages 95-102, 1995.

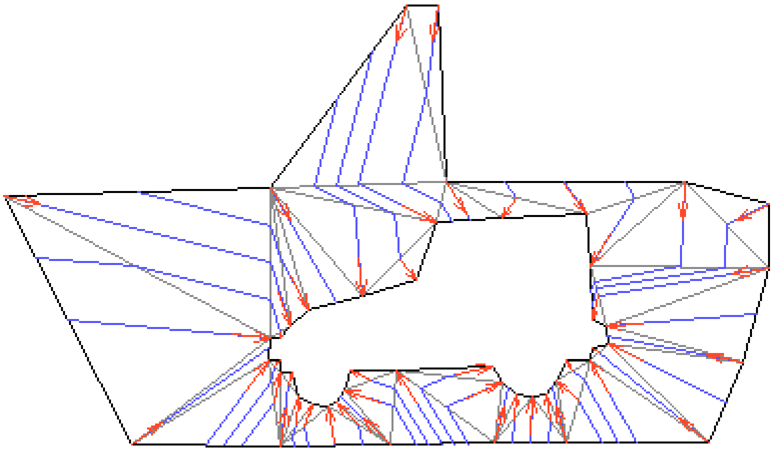
[14] D. P. Huttenlocher and K. Kedem, Computing the Minimal Hausdorff Distance from Point Sets Under Translation. In *Proceedings of the 6th ACM Symposium on Computational Geometry*, pages 340-349, 1990.

[15] L. Palios and B. Chazelle, Triangulating a Non-convex Polytope. In *Discrete and Computational Geometry*, pages 505-526, 1990.

[16] B. Chazelle, Triangulating a Simple Polygon in Linear Time. In *Discrete and Computational Geometry*, pages 485-524, 1991.

[17]. A Oliveira and S. Nascimento , Transformando Campos de Direções para transformar Polígonos simples, *Tech. Report. of the Systems Engineering and Computer Science Dept., COPPE/Federal University of Rio de Janeiro.*, 1997.

[18] J. Snoeyink, A trivial Knot whose Spanning Disks have exponential size. In *Proceedings of the 6th ACM Symposium on Computational Geometry*, pages 139-147, 1990



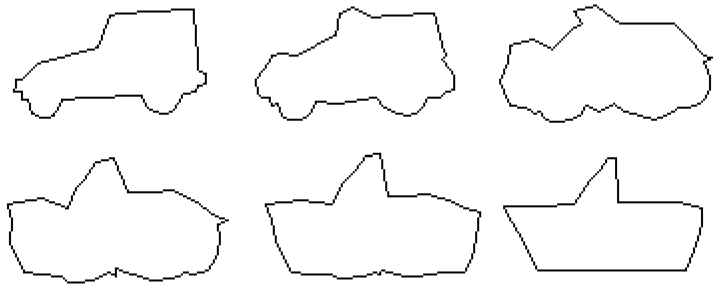


Figure 9 - An example of a morphing obtained by the 2D version of the algorithm. The first picture shows the directions generated and the trajectories that passes through vertices. The second picture shows six instances of the transformation obtained.

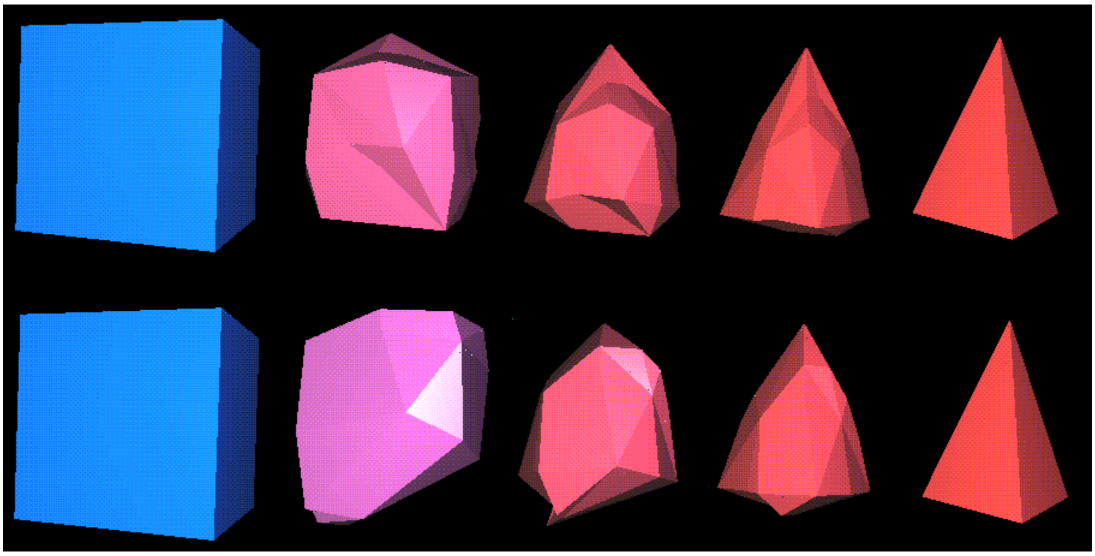


Figure 10 - Comparison between the use of two different interpolation process. The first sequence was obtained using a simple linear interpolation, while the second one used a linear interpolation along the vertices trajectories.

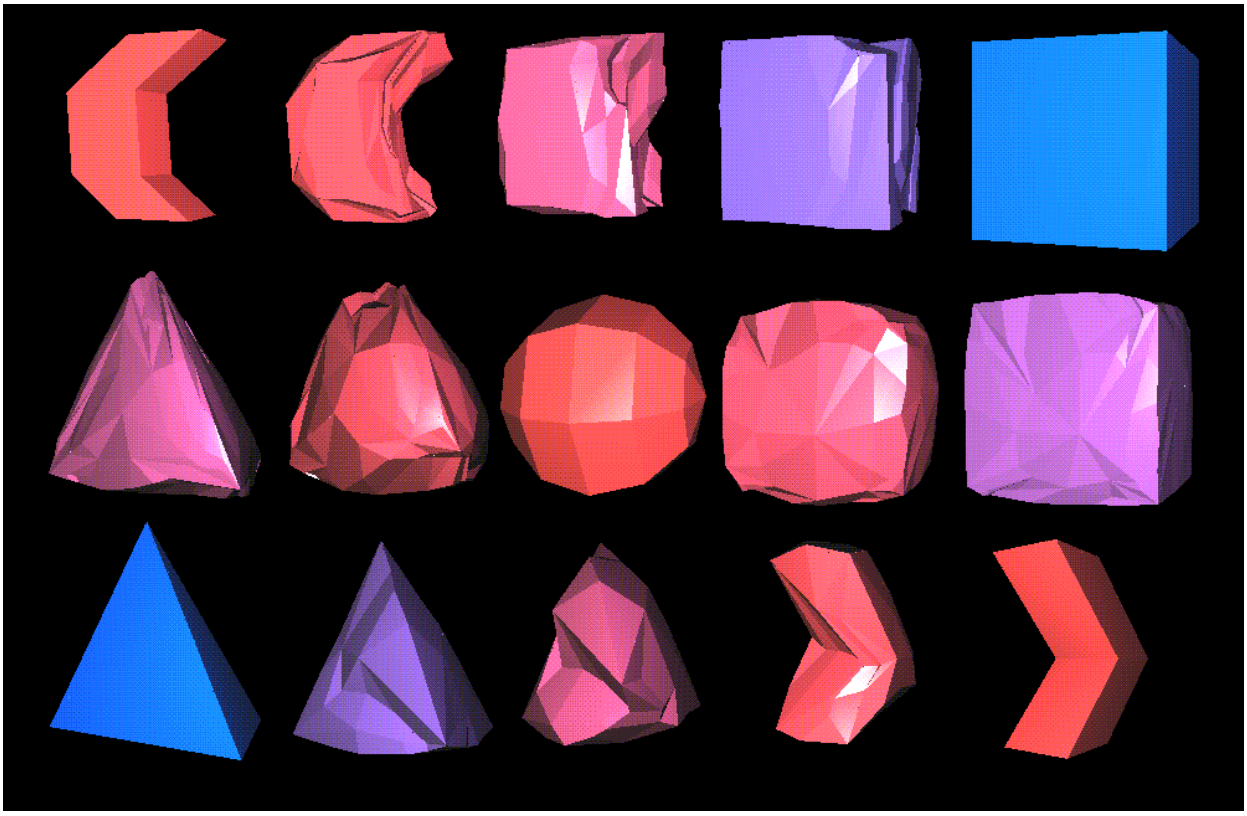


Figure 11 - A composition of many metamorphoses of simple polyhedral models. In the first line, from left to right, a non-convex polyhedron is transformed into a cube. In the second line, from right to left, the cube is transformed until it gets the shape of a "sphere with planar faces" and then of a tetrahedron. In the last line, the tetrahedron is transformed into another non-convex polyhedron.