Volume 15 (1996), Number 3

Blob Metamorphosis based on Minkowski Sums

E. Galin, S. Akkouche

Laboratoire d'Informatique Graphique Image et Modélisation
Ecole Centrale de Lyon
B.P. 163, 69131 Ecully Cedex

e-mail : [galin|samir]@cc.ec-lyon.fr

Abstract

This paper addresses the metamorphosis of soft objects built from skeletons. We propose a new approach that may be split into three steps. The first step consists in an original splitting of the initial and the final shapes with a view to creating a bijective graph of correspondence. In the second step, we assume that the skeletons are convex polygonal shapes, and thus take advantage of the properties of Minkowski sums to characterize the skeletons of intermediate shapes. Eventually, we characterize the intermediate distance and field functions; we describe a set of interpolation methods and propose to use a restricted class of parametrized distance and field functions so as to preserve coherence and speed-up computations.
We show that we can easily extend those results to achieve a Bézier like metamorphosis where control points are replaced by control soft objects; in this scope, we have adapted existing accelerated techniques that build a Bézier transformation from a set of convex polyhedra to any kind of convex polygonal shapes.
Eventually, we point out that matching all components of the initial and the final shapes generates amorphous intermediate shapes based on an overwhelming number of intermediate sub-components. Thus, we propose heuristics with a view to preserving coherence during the transformation and accelerating computations. We have implemented and tested our techniques in an experimental ray-tracer.

Keywords: animation; blobs; implicit surfaces; Minkowski sums; metamorphosis; skeletons; soft objects.

1. Introduction

Implicit surfaces have proved to be particularly efficient for modeling smooth objects of any topology. They have been successfully used in physically based simulation [7, 4], as well as in descriptive animation systems [18, 13].

Although several metamorphosis techniques have been proposed for polygonal models, very few attempts have been made concerning soft objects.

Wyvill [20] has proposed several heuristics for matching and interpolating soft objects based on skeletal elements with a view to avoiding over distorted or amorphous intermediate shapes. They may be split into two categories: techniques that match elements according to their position in space (cellular matching), and techniques that require extra information about the elements (hierarchial matching). A pre-processing step ensures that both initial and final shapes have the same number of components by creating null components whenever necessary. The correspondence process matches those components bijectively, and the transformation is defined by evaluating the surface generated by interpolating the skeletons position and field intensity, which are the main varying parameters.

However, this method suffers from two major limitations: it may not match components of different skeleton type, and components must be bijectively paired.

Pasko [14] has proposed to avoid the correspondence process by directly interpolating the implicit functions of the initial and the final shapes. This method has been tested in the scope of R-functions theory [17], and is somewhat similar to the *surface in-betweening* method proposed in [20]. However, this technique lacks control over the transformation.

In this paper, we propose a new metamorphosis technique for soft objects built from skeletons. Skeletons may be convex polygonal shapes of whatever dimension, i.e points, line segments, convex polygons or convex polyhedra, and field functions may be of whatever type, which allows a vast variety of shapes.

We propose to characterize the whole transformation by a *generic soft object model* whose instanciations through-out time interpolate the initial and the final shapes.

We assume that the animator provides us with a first rough graph of correspondence that matches parts of the initial and the final shapes, i.e. sets of components. Since this graph is in general non-bijective, we split each component into sub-components with a view to creating a new graph bijectively matching those sub-components. An intermediate sub-component is associated to each graph link, and intermediate shapes are defined by summing their influences.

The transformation of initial and final sub-components is achieved by transforming their skeletons, their distance functions and their potential functions.

A very interesting feature of Minkowski sums is that they can implicitly and directly combine any kind of convex polygonal skeleton, whatever their dimension may be, and yield a *generic representation* of intermediate convex polygonal shapes. We take advantage of those properties to characterize the skeletons of intermediate sub-components, and re-use existing accelerated techniques [5, 6].

Since directly interpolating distance and potential functions generates complex formulations, we tackle the transformation problem by defining a class of parametrized functions and characterizing intermediate functions as specific members with interpolated parameters. Thus, we achieve a concise *generic representation* which both preserves their mathematical properties and shape coherence.

The overall *generic representation* of the transformation enables us to iteratively and efficiently combine several interpolations so as to define a Bézier-like metamorphosis where control knots are replaced by control soft objects.

The remainder of this paper is organized as follows. After some fundamental concepts discussed in section 2, we address the splitting of components and the creation of a bijective graph of correspondence in section 3. Section 4 deals with the creation of intermediate skeletons and field functions. We characterize Bézier soft objects in section 5.

2. Fundamental concepts

In this section, we recall the basic definitions of soft objects built around skeletons, as well as the main properties of Minkowski sums bearing our algorithms.

2.1. Soft objects

Through out this paper, soft objects will refer to implicit surfaces built around skeletons [18, 2]. A soft object A is generated by summing the influences of scalar field components (also referred to as elements) $F_{A_i}(x,y,z)$ associated to their skeletons $\mathcal{S}_{A_i}$. The global field function $F_A(x,y,z)$ of an object A may be defined as:

$$F_A(x,y,z) = \sum_{i=1}^{i=N_A} F_{A_i}(x,y,z)$$

The surface of the object may be characterized from this global potential field $F_A(x,y,z)$ as the points of space whose potential equals a threshold value denoted T_A.

$$\Sigma_A = \{M(x,y,z) \in \mathbb{R}^3, F_A(x,y,z) = T_A\}$$

Although skeletons may be of any type [2], we restricted them to convex polygonal shapes of any dimension for some reasons that will be discussed in the next section. Through out the remainder of this paper, skeletal elements will refer to polygonal convex shapes, i.e. points, line segments, convex polygons and convex polyhedra.

Each component of the global field function $F_A(x,y,z)$ may be split into a distance function $d_{A_i}(x,y,z)$ and a potential function $f_{A_i}(r)$, where r stands for the distance to the skeleton [1]. We will refer to the following notation:

$$F_{A_i}(x,y,z) = f_{A_i} \circ d_{A_i}(x,y,z)$$

Throughout this paper, a component A_i of a soft object A will be fully characterized by its skeleton $\mathcal{S}_{A_i}$, its distance function $d_{A_i}(x,y,z)$ and its potential function $f_{A_i}(r)$, and denoted as $\{\mathcal{S}_{A_i}, d_{A_i}(x,y,z), f_{A_i}(r)\}$.

2.2. Minkowski sums

As mentioned in the introduction, our soft object metamorphosis technique strongly relies on the transformation of polygonal skeletons bearing the implicit surfaces.

Several metamorphosis techniques have been proposed in the literature [9, 10, 11], however, they are dedicated to the transformation of polyhedra, and fail to combine polygonal shapes of any dimension (i.e. points, line segments, polygons and polyhedra).

The metamorphosis technique based on Minkowski sums [8] do not suffer from the same limitations, and may be successfully used to transform any polygonal shape into another one.

Definition: The Minkowski sum of two sets A and B, denoted $A \oplus B$, is defined as $A \oplus B = \{a+b, a \in A, b \in B\}$.

The linear interpolation based on Minkowski sums generates a family of intermediate polygonal shapes $C(t)$ that smoothly interpolates the initial shape A into the final shape B:

$$C(t) = (1-t)A \oplus tB$$

Several methods have been proposed to characterize intermediate shapes. A general technique has been proposed in [8]: the metamorphosis is defined by the evolution of a list of faces that may intersect or overlap; those faces create the hull of intermediate shapes. However, this technique does not take advantage the a priori known topology of the initial and the final shapes.

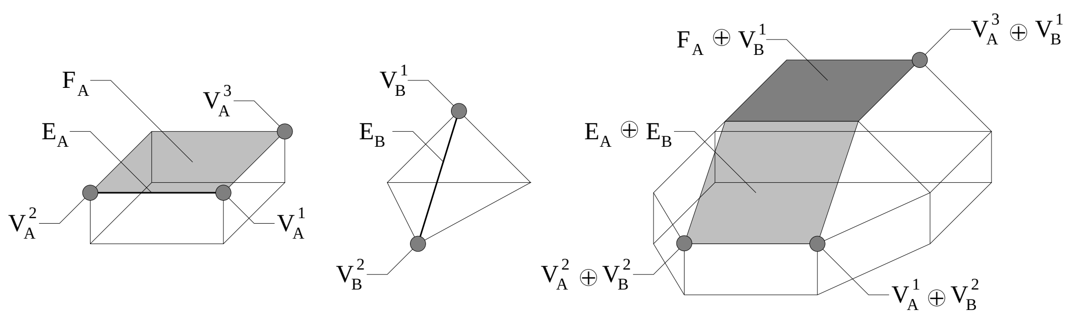

Figure 1: *Characterization of the vertices of the Minkowski sum of two convex polyhedra*

Whenever the argument shapes are convex, the topology of the intermediate convex shapes remains unchanged through out time (there is neither creation nor destruction of vertices, edges or faces). The following theorems [16, 8, 5] characterize the Minkowski sum of convex polyhedra:

Theorem: The Minkowski sum of two convex polyhedra is a convex polyhedron.

Theorem: Given two convex polyhedra A and B, any vertex V_C of C may be written as $V_C = V_A \oplus V_B$, V_A and V_B are said to be the parent vertices of V_C, and the relationships between V_A, V_B and V_C is unique.

Those theorems lead to the following fundamental results [6] that characterize our soft objects skeleton metamorphosis.

$C = A \oplus B$ may be defined as the convex hull of the vertices $V_C = V_A \oplus V_B$ provided argument polyhedra are convex (*Figure 1*). The transformation may be characterized by a generic topology, whose geometry is defined in terms of a reference table between the vertices of C and the parent vertices of A and B. In the remainder of this paper, the topology will be denoted $\mathcal{T}(A \oplus B)$, and the reference table $\mathcal{R}(C,A,B)$.

The topology $\mathcal{T}(A \oplus B)$ may be created once and for all. Straight forward techniques consist in building the convex hull of candidates vertices $V_C = V_A \oplus V_B$, and require $O\left((N_A N_B)^2\right)$ computational time, where N_A and N_B stand for the number of vertices of A and B. Accelerated techniques that take advantage of the topology of argument polyhedra have been proposed [6] and lower the overall complexity to $O(N_C \log(N_C))$, where N_C stands for the number of vertices of C.

We have successfully adapted those techniques to any kind of convex polygonal shape.

3. The correspondence process

In this section, we address the metamorphosis of an initial shape A into a final shape B. Both shapes are characterized by N_A and N_B components A_i and B_j, whose skeletons are denoted $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$ respectively. Their corresponding field functions are denoted F_{A_i} and F_{B_j}, according to the previous formulation.

We propose to define intermediate shapes as soft objects whose components are derived from the components of the initial and the final shapes. The correspondence process creates a graph matching elements of the initial and the final shapes. Since this graph may not be bijective, we split components A_i and B_j into sub-components whenever necessary with a view to creating a new graph bijectively matching those sub-components. A new intermediate component is associated to each graph link, and intermediate shapes are defined by summing their influences.

3.1. Splitting of components

In this section, we consider a soft object A built over N_A elements A_i. We state that a component A_i may be split into *sub-components* $s_k(A_i)$ so that summing their influences equals the contribution of the component A_i itself.

Since the splitting of skeletons into convex *sub-skeletons* leads to unwanted bulges [3], we propose to split the potential function $f_{A_i}(r)$ into N *sub-functions* $f_{s_k(A_i)}(r)$. The characterization of $s_k(A_i)$ may be achieved as follows:

$$\begin{cases} s_k(A_i) = \{\mathcal{S}_{A_i}, d_{A_i}(x,y,z), f_{s_k(A_i)}(r)\} \\ \sum_{k=1}^{k=N} f_{s_k(A_i)}(r) = f_{A_i}(r) \end{cases}$$

Although any kind of *sub-function* $f_{s_k(A_i)}$ might be used, we propose to define them as $f_{s_k(A_i)} = \alpha_k f_{A_i}$, where α_k weights the contribution of sub-component $s_k(A_i)$ to creation of the component A_i. As we will see later, this technique reduces computations, since $f_{s_k(A_i)}$ share the same mathematical properties as f_{A_i}.

3.2. Transformation of the graph of correspondence

In this section, we assume that a graph of correspondence between the elements of the initial and the final shapes has been created. This graph specifies which parts of A and B should undergo metamorphosis.

In practice, the correspondence process may be partially or fully controlled by the animator, or achieved through heuristics, such as *hierarchial matching* as proposed in [20].

We assume that the graph matches N_{A_i} components B_j (referred to as *target components*) to each initial component A_i, and N_{B_j} components A_i (referred to as *source components*) to each final component B_j. Note that $N_A \cdot N_B$ links are generated at most when all components of A and B are matched together, which generally leads to unwanted amorphous blobby intermediate shapes. The exact number of links will be denoted as N_C (*Figure 2*).

Note: the set of indexes j of target components B_j associated to an initial component A_i will be denoted $J(i)$; and $I(j)$ will refer to the set of indexes i of source components A_i linked with a final element B_j respectively. In *Figure 2*, $I(2) = \{0,2\}$, $J(1) = \{0\}$ and $I(1) = .$

Initial and final components A_i and B_j may not be bijectively matched, and may have none, one or multiple links in the graph of correspondence. The splitting process creates a new decomposition of initial shapes A and B whose new components are bijectively matched.

Whenever an initial component A_i is multiply linked with N_{A_i} target components B_j, we split it into N_{A_i} *sub-components* $s_k(A_i)$ whose skeletons $\mathcal{S}_{s_k(A_i)}$ are the same as $\mathcal{S}_{A_i}$, and whose potential field functions are weighted by a coefficient α_{ik}. The same decomposition may be applied to the final components B_j: each of them may be split into N_{B_j} sub-components $s_k(B_j)$,

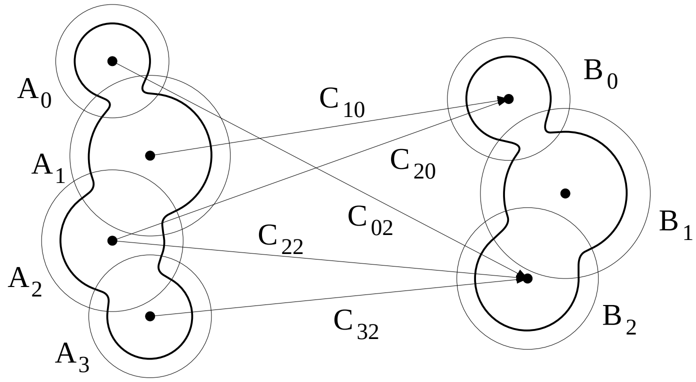

Figure 2: *Correspondence graph between soft objects based on convex polygonal skeletons*

whose skeletons are the same as $\mathcal{S}_{B_j}$, and whose field functions are weighted by a coefficient β_{kj}. Note that elements A_i or B_j that bear a single link in the graph need not be split.

Whenever components A_i or B_j do not bear any link in the graph, we create a corresponding null component $\mathcal{N}(A_i)$ or $\mathcal{N}(B_j)$, and treat this case as a single link, which is straight-forward. The characterization of those null components will be addressed later.

Note: in practice, any kind of values may be affected to those weights, which both provides a tight control over the transformation and allows the creation of special effects, provided that the weights α_{ij} and β_{ij} should form a partition of the unit, as mentioned in the previous section.

3.3. Null components

Initial or final components may not be matched with any other components during the correspondence process (*Figure 2*), which involves the creation of corresponding null components.

We propose to associate a specific null component $\mathcal{N}(A_i)$ to each link-free component A_i. Its characterization may be achieved as follows:

$$s_k(A_i) = \left\{ \mathcal{S}_{A_i}, d_{A_i}(x,y,z), f_{\mathcal{N}(A_i)}(r) \equiv 0 \right\}$$

Since the skeletons and of the initial and the final components are the same, the intermediate skeleton remains unchanged and no Minkowski sum is required, which saves computations. Moreover, the distance function is preserved, which also prevents from transforming it. As we will see later, several characterizations of $f_{\mathcal{N}(A_i)}(r)$ are possible.

3.4. Weight computation

In this section, we propose heuristics for the computation of the weights α_{ij} and βij involved in the splitting of the components of the initial and the final shapes based on an original energy term characterizing each sub-component $s_k(A_i)$ and $s_k(B_j)$.

3.4.1. Energy

Given a component A_i of a soft object A, we consider its field function $F_{A_i}(x,y,z)$ as an energy density spread around the skeleton $\mathcal{S}_{A_i}$. Thus, the energy $\mathcal{E}_{A_i}$ of the component A_i may be defined as:

$$\mathcal{E}_{A_i} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F_{A_i}(x,y,z)\, dx\, dy\, dz$$

Note: since the implicit function $F_A(x,y,z)$ is defined as the sum of its elements functions $F_{A_i}(x,y,z)$, the overall energy of the soft object A is defined as $\mathcal{E}_A = \sum_{i=1}^{N_A} \mathcal{E}_{A_i}$. The normalized energy of a component A_i will refer to $\dfrac{\mathcal{E}_{A_i}}{\mathcal{E}_A}$.

Since we opted to work with a restricted class of distance and field functions $\mathcal{D}$ and $\mathcal{F}$ respectively, the computation of the energy may be coded easily.

3.4.2. Heuristics

In this section, we assume that the correspondence graph between elements has been created by whatever method, and we address the computation of the weights α_{ij} and β_{ij}.

Through out the transformation, α_{ij} weights the contribution of the component A_i to the creation of the component B_j; whereas β_{ij} characterizes the participation of the component A_i to the creation of the component B_j. Although any kind of value might be affected to those weights, which permits special effets, we propose two kind of heuristics for automating their computation.

The following heuristic is rather straight-forward for it does not take into account the characteristics of initial and final components A_i and B_j; it equally spreads their energy $\mathcal{E}_{A_i}$ and $\mathcal{E}_{B_j}$ between their corresponding sub-components $s_k(A_i)$ and $s_k(B_j)$.

Heuristic: set weights as $\alpha_{ij} = \dfrac{1}{N_{A_i}}$ and $\beta_{ij} = \dfrac{1}{N_{B_j}}$.

The second heuristic computes weights α_{ij} and β_{ij} so that the energy of the time varying component $C_{ij}(t)$ should be proportionally linked to the energy of its arguments A_i and B_j.

Heuristic: set weights as $\alpha_{ij} = \dfrac{\mathcal{E}_{B_j}}{\sum_{k=1}^{N_{A_i}} \mathcal{E}_{B_k}}$ and $\beta_{ij} = \dfrac{\mathcal{E}_{A_i}}{\sum_{k=1}^{N_{B_j}} \mathcal{E}_{A_k}}$.

We stress that the influence of those weights are the more subtle as the number of components involved in the transformation is high. In practice, the animator may rely on those heuristics that provide him with a first solution, and play with parameters to fit the metamorphosis.

4. Characterization of intermediate shapes

The transformation of components A_i and B_j relies on the transformation of the bijectively matched sub-components $s_j(A_i)$ and $s_i(B_j)$.

The transformation may be characterized by the evolution of intermediate components $C_{ij}(t)$ associated to each pair of initial-final sub-components $\left(s_j(A_i), s_i(B_j)\right)$. Thus, intermediate shapes may be defined by their global field function as follows:

$$F_{C(t)} = \sum_{i=1}^{N_A} \sum_{j \in J(i)} F_{C_{ij}(t)} = \sum_{j=1}^{N_B} \sum_{i \in I(j)} F_{C_{ij}(t)}$$

The surface of intermediate shapes is defined as the points of space whose potential equals an intermediate threshold value $T_C(t)$ that interpolates the initial and the final thresholds T_A and T_B:

$$\begin{cases} \Sigma_{C(t)} = \{M(x,y,z), F_{C(t)}(x,y,z) = T_C\} \\ T_C = (1-t)T_A + tT_B \end{cases}$$

Each intermediate component $C_{ij}(t)$ should be characterized by its time varying skeleton $\mathcal{S}_{C_{ij}(t)}$, its distance function $d_{C_{ij}(t)}(x,y,z)$ and its field function $f_{C_{ij}(t)}(r)$. We address the computation and point out the interesting properties of $C_{ij}(t) = \{\mathcal{S}_{C_{ij}(t)}, d_{C_{ij}(t)}(x,y,z), f_{C_{ij}(t)}(r)\}$ in the following sections.

4.1. Transformation of skeletons

As pointed out in the introduction, the very interesting feature of Minkowski sums is that they can implicitly and directly combine any kind of convex polygonal skeleton, whatever their dimension may be. We recall that the skeletons $\mathcal{S}_{s_k(A_i)}$ and $\mathcal{S}_{s_k(B_j)}$ of sub-components $s_k(A_i)$ and $s_k(B_j)$ are the same as $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$. Since argument skeletons are convex polygonal shapes, the linear interpolation based on Minkowski sums creates a generic skeleton $\mathcal{S}_{C_{ij}}$ that smoothly interpolates the initial and the final skeletons $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$.

$$\mathcal{S}_{C_{ij}(t)} = (1-t)\mathcal{S}_{A_i} \oplus t\mathcal{S}_{B_j}$$

This skeleton is completely characterized by its generic topology $\mathcal{T}(\mathcal{S}_{C_{ij}})$ and the coordinates of its vertices that are easily computed thanks to the reference table $\mathcal{R}(\mathcal{S}_{C_{ij}}, \mathcal{S}_{A_i}, \mathcal{S}_{B_j})$ [6]. Both topology and reference table may be created once and for all, and any intermediate skeleton at any time is generated as a specific instanciation of the generic model.

4.2. Transformation of potential fields

Since the creation of $\mathcal{S}_{C_{ij}(t)}$ has been addressed in the previous section, we are to characterize both $d_{C_{ij}(t)}(x,y,z)$ and $f_{C_{ij}(t)}(r)$ that should satisfy the following constraints:

$$\begin{cases} \sum\limits_{j\in J(i)} f_{C_{ij}(0)}(r) = f_{A_i}(r) \\ d_{C_{ij}(0)}(x,y,z) = d_{A_i}(x,y,z) \end{cases} \qquad \begin{cases} \sum\limits_{i\in I(j)} f_{C_{ij}(1)}(r) = f_{B_j}(r) \\ d_{C_{ij}(1)}(x,y,z) = d_{B_j}(x,y,z) \end{cases}$$

A general technique consists in interpolating the initial and the final distance and potential functions; we propose several approaches in the following sections.

4.2.1. Transformation of potential functions

The transformation between the potential fields $\alpha_{ij} f_{A_i}(r)$ and $\beta_{ij} f_{B_j}(r)$ of sub-components $s_j(A_i)$ and $s_i(B_j)$ may be achieved according to the following formula:

$$f_{C_{ij}(t)}(r) = (1-t)\alpha_{ij} f_{A_i}(r) + t\beta_{ij} f_{B_j}(r)$$

Although this method may cope with any kind of field function, the computation of $f_{C_{ij}(t)}(x,y,z)$ may not be straight forward, the more so as the argument functions may be defined over several intervals:

- whenever $f_{A_i}(r)$ and $f_{B_j}(r)$ are defined over $N_{I_{A_i}}$ and $N_{I_{B_j}}$ intervals respectively, the field function $f_{C_{ij}(t)}(r)$ is defined over $N_{I_{C_{ij}(t)}}$ intervals, which is at most $N_{I_{A_i}} + N_{I_{B_j}}$. Thus, the number of intervals supporting $f_{C_{ij}(t)}(r)$ may increase, which slowers the computations (*Figure 3*).
- whenever field functions are polynomials, the degree of their weighted sum rizes to the maximum degree of argument polynomials. However, summing rational polynomials of different denominators increases the degree of the resulting numerator polynomial when root solving, which also dramatically increases computations.
- the direct interpolation results into an unwanted loss of coherence of intermediate function shapes (*Figure 3*).

We propose an alternative accelerated technique. Functions of the initial and the final shapes $f_{A_i}(r)$ and $f_{B_j}(r)$ should belong to the same class $\mathcal{F}$ characterized by a set of parameters denoted $\mathcal{P}_A$ and $\mathcal{P}_B$ respectively. The intermediate field function $f_{C_{ij}(t)}(r)$ may be defined as a specific element of that class $\mathcal{F}$, whose parameters $\mathcal{P}_{C_{ij}(t)}$ are interpolating the parameters of the initial and the final field function $f_{A_i}(r)$ and $f_{B_j}(r)$.

Several potential field functions have been proposed in the literature [18, 12, 7, 1]. We have implemented the following potential fields in an experimental ray-tracer for testing our algorithms.

$$f(r) = I\left(-\frac{4}{9}\left(\frac{r}{R}\right)^6 + \frac{17}{9}\left(\frac{r}{R}\right)^4 - \frac{22}{9}\left(\frac{r}{R}\right)^2 + 1\right)$$

$$f(r) = I\left(1 - \left(\frac{r}{R}\right)^2\right)^2$$

$$\begin{cases} f(r) = I\left(1 - \dfrac{9(r/R)^4}{P + (9/2 - 4P)(r/R)^2}\right) & 0 \le \left(\dfrac{r}{R}\right)^2 \le \dfrac{1}{4} \\ f(r) = I\left(\dfrac{\left(1-r^2\right)^2}{3/4 - P + (3/2 + 4P)\,(r/R)^2}\right) & \dfrac{1}{4} \le \left(\dfrac{r}{R}\right)^2 \le 1 \end{cases}$$

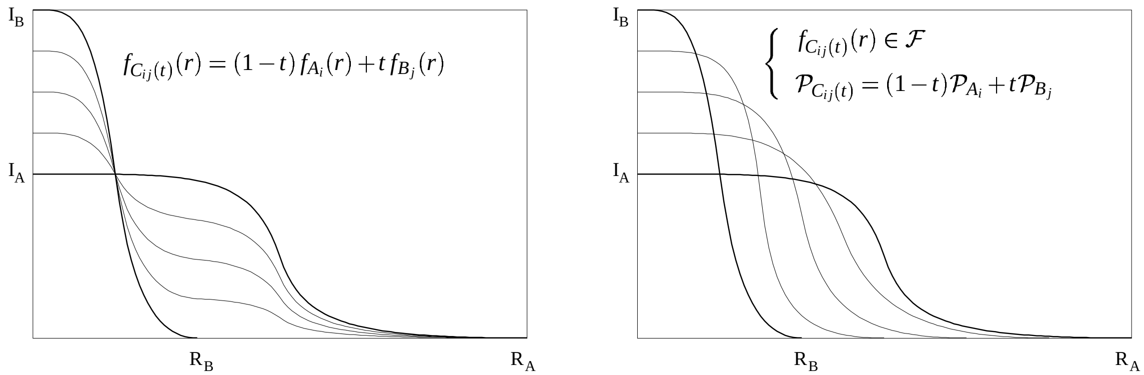

Figure 3: *Characterization of intermediate potential field functions through direct interpolation and parameter transformation*

The shape of those field functions are controlled by a set of parameters $\mathcal{P}\{I,R,P\}$, which are the *intensity* I, the *radius of influence* R and the *stiffness* P. We stress that field functions sharing the same parametrized definition form a class whose members satisfy a set of constraints and share the same mathematical properties.

We recall that we are interpolating the parameters of *sub-components* $s_k(A_i)$ and $s_k(B_j)$, and we assume that $\mathcal{P}_{A_i}$ and $\mathcal{P}_{B_j}$ refer to the column vector of parameters, therefore we may characterize the intermediate field function as:

$$\left\{ \begin{array}{l} f_{C_{ij}(t)}(r) \in \mathcal{F} \\ \mathcal{P}_{C_{ij}(t)} = (1-t)\mathcal{P}_{A_i} + t\mathcal{P}_{B_j} \end{array} \right. \qquad \mathcal{P}_{A_i} = \begin{pmatrix} \alpha_{ij} I_{A_i} \\ R_{A_i} \\ P_{A_i} \end{pmatrix} \qquad \mathcal{P}_{B_j} = \begin{pmatrix} \beta_{ij} I_{B_j} \\ R_{B_j} \\ P_{B_j} \end{pmatrix}$$

Note: null potential functions $f_{\mathcal{N}(A_i)}(r)$ are characterized by a null intensity $I_{\mathcal{N}(A_i)} = 0$, whereas other parameters, such as the radius of influence, may be set to whatever value. In practice, we propose that $\mathcal{P}_{\mathcal{N}(A_i)}$ should be set to $\{0, R_{A_i}, P_{A_i}\}$ with a view to preserving shape coherence.

The common parametrized potential field function will said to be *generic*, and time specific potential field function will be referred to as a precise *instanciation* of the model. The code needed for whatever computations (renderings, polygonizations) is always the same, which increases speed.

4.2.2. Transformation of distance functions

We stress that the transformation of distance functions issue is almost identical to the field function transformation problem. The transformation of the distance functions $d_{A_i}(x,y,z)$ and $d_{B_j}(x,y,z)$ of sub-components $s_j(A_i)$ and $s_i(B_j)$ could be achieved by directly interpolating them:

$$d_{C_{ij}(t)}(x,y,z) = (1-t)d_{A_i}(x,y,z) + td_{B_j}(x,y,z)$$

However, this technique is computationnaly expensive since it requires to compute distances twice as many times as usual for each sub-component.

Although any kind of distance function may qualify, we limited our study to the *ellipsoidal* distance [19], which is an extension of the usual Euclidian distance, also known as *spherical* distance. We propose that both initial and final distance functions should belong to the same *class* $\mathcal{D}$, and characterize intermediate distance function by interpolating their parameters; thus:

$$\left\{ \begin{array}{l} d_{C_{ij}(t)}(x,y,z) \in \mathcal{D} \\ \mathcal{P}_{C_{ij}(t)} = (1-t)\mathcal{P}_{A_i} + t\mathcal{P}_{B_j} \end{array} \right.$$

Here, $\mathcal{P}$ refers to the specific parameters of the distance function ,*e.g.* the column vector of the radii of the ellipsoid when coping with *ellipsoidal distance*. At any time, $d_{Cij(t)}(x,y,z)$ is a specific *instanciation* of the *generic* distance function model $\mathcal{D}$.

5. Bézier blobby metamorphosis

In this section, we present Bézier metamorphosis as an extension of the linear transformation between two soft objects. Control points of the Bézier curve are replaced by control soft objects.

We address the metamorphosis controlled by n soft objects denoted A^k, whose skeletons, distance and field functions are denoted $\mathcal{S}_{A^k}$, $d_{A^k}(x,y,z)$ and $f_{A^k}(r)$ respectively (*Figure 4*).

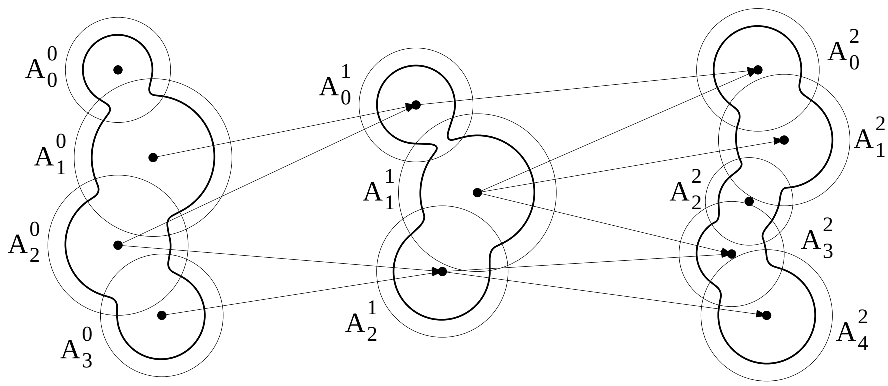

Figure 4: *Bézier metamorphosis*

We take advantage of both the *generic topology* of intermediate skeletons and the *class based* characterization of intermediate distance and field functions to iteratively combine several shapes.

5.1. Characterization of skeletons

The intermediate skeletons are defined as Bézier polygonal shapes [5, 15], which are an extension of the linear interpolation based on Minkowski sums involving several polygonal shapes.

Definition: Given $n+1$ control polygonal shapes denoted $\mathcal{S}_{A^k}$, $k \in [0,n]$, the family of Bézier polygonal shapes is defined as $\mathcal{S}_C = \bigoplus_{k=0}^{k=n} \lambda_k^n(t)\mathcal{S}_{A^k}$, where $\lambda_k^n(t)$ stand for the Bernstein polynomials of degree n.

Since skeletons are convex polygonal shapes, we use the genericity of intermediate skeletons to add or remove control skeletons without creating the whole transformation [6]. The metamorphosis of the skeletons is fully characterized by a generic topology $\mathcal{T}(\mathcal{S}_{A^0} \oplus \cdots \oplus \mathcal{S}_{A^n})$ and the reference table of vertices $\mathcal{R}(\mathcal{S}_C, \mathcal{S}_{A^0}, \cdots, \mathcal{S}_{A^n})$.

5.2. Characterization of distance and potential field functions

Several techniques may be used for characterizing intermediate distance and field functions. Despite its generality, the direct interpolation of those functions proves to be computationally expensive, the more so as the number of soft objects involved in the transformation increases, as pointed out in the previous section.

We propose to re-use our accelerated scheme; intermediate functions of sub-components are characterized by interpolating the shape parameters of their argument components, provided control soft objects distance and potential field functions belong to the same classes $\mathcal{D}$ and $\mathcal{F}$ respectively.

The characterization of those functions is a straight-forward extension of the linear interpolation: their parameters $\mathcal{P}_{C(t)}$ are defined as the Bézier interpolation of arguments $\mathcal{P}_{A^k}$.

6. Conclusion

In this paper, we have proposed a new soft object metamorphosis technique. We have focused our attention on the creation of intermediate shapes that are fully characterized by a *generic representation*. We have pointed out that the genericity of skeletons as well as distance and field functions provides us with a concise representation of intermediate shapes, which reduces computations involved at whatever steps of surface characterization (such as ray-tracing renderings or polygonizations). This generic representation enables us to iteratively combine several control soft objects so as to create a Bézier-like metamorphosis.

We have developed an object-oriented library implementing different classes of soft objects built from convex polygonal skeletons, and linked it to an experimental ray-tracer (see plates *1*, *2*).

We are currently working on accelerated rendering techniques. Since the characterization of intermediate shapes may be achieved at interactive rates, we are currently trying to take advantage of the coherence of both the skeletons and the field functions to speed-up polygonizations and ray-tracing an animation sequence of soft objects.

References

1. C. Blanc and C. Schlick. Extended field functions for soft objects. In *Implicit Surfaces'95*, **1**: 21-32, Grenoble, April 1995. 144, 149
2. J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. In *Computer Graphics*, **24**(2):109-116, January 1990. 144
3. J. Bloomenthal. Bulge Elimination in Implicit Surface Blends. In *Implicit Surfaces'95*, **1**: 7-20, Grenoble, April 1995. 146
4. M. Desbrun and M. P. Gascuel. Highly deformable material for animation and collision processing. In *5th Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994. 143
5. E. Galin and S. Akkouche. Spécification d'un logiciel interactif de déformations d'objets volumiques tridimentionnels basé sur les sommes de minkowski. In *Congrès Interface des Mondes Réels et Virtuels*, 73-81, Février 1994. 144, 145, 151
6. E. Galin and S. Akkouche. Métamorphose d'objets polyédriques, quelques méthodes d'accélération. *Technique et Science Informatiques*, **15**(3): 329-351, April 1996. 144, 145, 146, 149, 151
7. M. P. Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics (Siggraph'93)*, 313-320, August 1993. 143, 149
8. A. Kaul and J. Rossignac. Solid interpolating deformations, construction and animation of pips. *Computer Graphics*, **16**(1): 107-115, January 1992. 145
9. J.R. Kent, R. E. Parent and W. E. Carlson. Establishing Correspondences by Topological Merging, a New Approach to 3-D Shape Transformation. *Graphic Interface'91*, Calgary, June 1991, 271-278. 145
10. J.R. Kent, R. E. Parent and W. E. Carlson. Shape Transformation for Polyhedral Objects. *Computer Graphics (Siggraph'92)*, **26**(2): 47-54, July 1992. 145
11. F. Lazarus and A. Verroust. Feature-based Shape Transformation of Polyhedral Objects. *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994. 145
12. S. Murajami and H. Ichihara. On a 3-D Display Method by Metaball Technique. Journal of papers at *Electronics Communication Conference '87*, **J70-D(8)**: 1607-1615, 1987 (in Japanese). 149
13. A. Opalach and S. Maddock. High level control of implicit surfaces for character animation. In *Implicit Surfaces'95*, **1**: 223-232, Grenoble, France, 1995. 143
14. A. Pasko and V. Savchenko. Constructing functionally defined surfaces. *Implicit Surfaces'95*, **1**: 97-106, Grenoble, France, 1995. 143
15. J. Rossignac and A. Kaul. AGRELs and BIPs, Metamorphosis as a Bézier Curve in the Space of Polyhedra. *Eurographics'94*, **13**(3): 179-184, January 1992. 151
16. J. Serra. *Image Analysis and Mathematical morphology*, Academic Press Inc., 1982. 145
17. V. Shapiro. Real functions for representation of rigid solids. *Computer Aided Geometric Design*, **12**(2): 153-175, 1994. 144
18. B. Wyvill, C. Mc Pheeters and G.Wyvill. Animating soft objects. *The Visual Computer*, 235-242, August 1986. 143, 144, 149

19. B. Wyvill and G.Wyvill. Field functions for implicit surfaces. *The Visual Computer*, **5**: 75-82, 1989. 150

20. B. Wyvill. Metamorphosis of implicit surfaces. In *Modeling, Visualizing and Animating with Implicit Surfaces* (*Siggraph'93* course notes Number 25), Anaheim, CA, USA, August 1993. 143, 144, 146

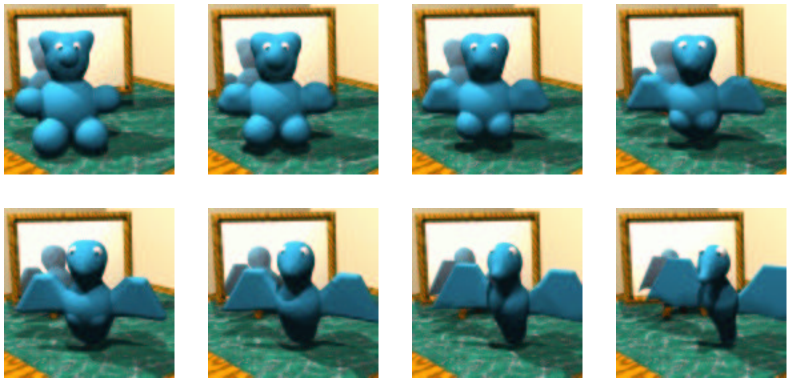

Figure 5: *Metamorphosis between a bear and a bird ; the correspondence graph has been defined as follows: heads and bodies have been matched, and the bear's arms have been transformed into the bird's wings*

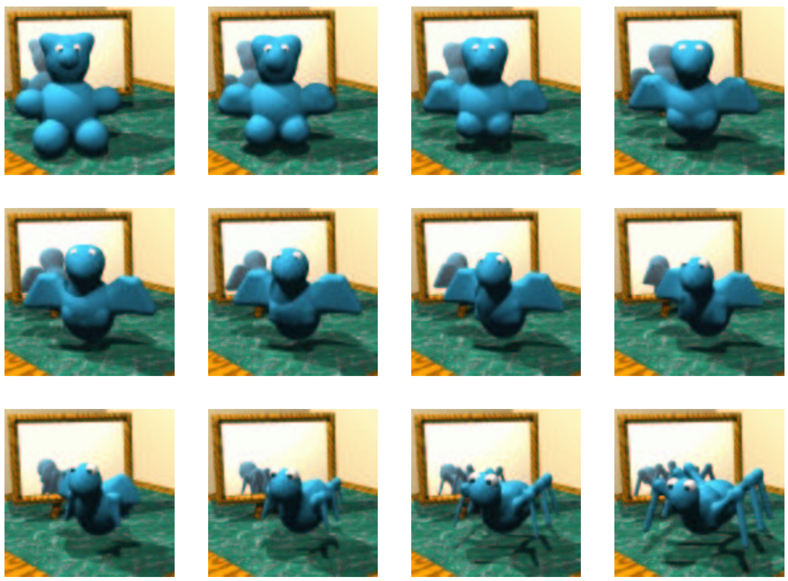

Figure 6: *The metamorphosis is defined as a Bézier curve controlled by the three soft objects: the initial shape is a bear and the final shape is a bug, the intermediate control soft object is a bird whose wings partially appear during the transformation*