Shape Constrained Blob Metamorphosis

E. Galin, S. Akkouche
Ecole Centrale de Lyon
B.P. 163, 69131 Ecully Cedex, France
[galin-samir]@cc.ec-lyon.fr

Abstract

This paper presents several improvement to the metamorphosis of soft objects built from convex polygonal skeletons method proposed in [7]. First, we analyze the creation of amorphous intermediate shapes process, and propose to use local coordinate systems associated to each component so as to control both the dimension of the intermediate skeletons and tune the trajectory path of elements during the transformation with a view to avoiding an over concentration of components.

The control of blending is also a difficult issue : two non-blending components of an initial soft object may be transformed into two blending components of a final object as a result of the matching process. Thus, we propose an original blending graph metamorphosis technique as well as blending functions interpolation schemes that generate smooth transitions between different blending states. Eventually, we address the computation of the global field function of a soft object whose graph holds heterogeneous blending relationships, which may arise during the transformation, and propose a generic formula fully compatible with our metamorphosis model.

Keywords : animation, blending graph, 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 as well as in descriptive animation systems. Although a vast variety of metamorphosis techniques have been proposed for polygonal models, only a few attempts have been made concerning soft objects.

Several heuristics for matching and interpolating soft objects based on skeletal elements have been proposed in [19], those techniques may be split into two categories : techniques matching elements according to their position in space (cellular matching), and techniques that require extra information about the elements (hierarchical matching). A pre-processing step ensures that both initial and final shapes share 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 types, and components must be bijectively paired.

As described in [13], the correspondence process may be avoided by directly interpolating the field functions of the initial and the final shapes, however, this method is somewhat similar to the surface in-betweening technique addressed in [19] and lack control over the transformation.

We have proposed another metamorphosis technique for soft objects built from skeletons in [7]. The key feature is that the whole transformation is characterized by a generic soft object model whose instanciations throughout time interpolate the initial and the final shapes. Skeletons may be convex polygonal shapes of whatever dimension, *i.e.* points, line segments, convex polygons or convex polyhedra, and we rely on linear interpolation based on Minkowski sums [11] so as to create time varying skeletons. We restrict our distance and potential functions to a parametrized class model of functions so as to characterize intermediate functions as specific members with interpolated parameters and avoid complex formulations, this technique also enables us to preserve the shape coherence of intermediate shapes.

Although the Minkowski sum implicitly combines convex polygonal shapes, it also implicitly generates a trajectory path for each component, which may result in crossing components and yields weird intermediate shapes, moreover, the Minkowski sum may also increase the dimension of the resulting intermediate skeletons, *e.g.* the transition between two non colinear line segments generates a deforming polygon. Both those effects may appear as undesirable, and we propose an alternative technique that controls both the *dimension* of the intermediate skeletons and their *trajectory path* during the transformation.

Although soft objects provide implicit blending properties, achieving control so that blending occurs only when it is desirable still remains a difficult issue. A general solution consists in defining a blending graph between the different components of a soft object, and stating that an element's field only blends with the contributions of linked elements.

Although articulated structures preserve their blending graph throughout the animation, a metamorphosis step may involve the transformation of this blending graph : two *non-blending* components of an initial soft object A may be respectively matched with two blending components of a final object B. A first straight forward technique avoids the computation of an intermediate time varying graph : since both initial and final split soft objects share the same components, the time varying potential field is defined as the interpolation of two other fields generated by the set of time varying components linked by the initial and the final blending graphs.

A second technique addresses the characterization of a generic intermediate blending graph whose arcs hold *mixed blending* relationships between components : each arc of the graph holds the interpolation of an initial and a final blending function, which leads to highly *heterogeneous blending graphs*. Therefore, we propose an original technique for computing the potential field at a point of space that copes with heterogeneous graphs.

The remainder of this paper is organized as follows : we recall the basic features of our soft object metamorphosis technique in section 2, and present the control of intermediate shapes in section 3. We eventually address the transformation of blending graphs in section 4.

2 Fundamental concepts

In this section, we define several notations and recall the principles of our soft object metamorphosis technique previously developed in [8].

2.1 Soft objects

Throughout this paper, soft objects will refer to implicit surfaces built around skeletons [17, 3]. A soft object A is generated by summing the influences of N_A scalar field components (also referred to as elements) $g_{A_i}(x,y,z)$ associated to their skeletons $\mathcal{S}_{A_i}$. The global potential field $g_A(x,y,z)$ of an object A may be defined as :

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

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

$$\Sigma = \left\{ M(x,y,z) \in \mathbb{R}^3, g_A(x,y,z) = T_A \right\}$$

Although skeletons may be of any type [3], we restricted them to convex polygonal shapes of any dimension for some reasons that will be discussed in the next section, throughout this paper, skeletal elements will refer to points, line segments, convex polygons and convex polyhedra. Each component contributing to the global field function $g_A(x,y,z)$ may be split into a distance function $d_{A_i}(x,y,z)$ and a potential function $g_{A_i}(r)$, where r stands for the distance to the skeleton [2]. We will refer to the following notation :

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

Thus 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 $g_{A_i}(r)$, and denoted as $\{\mathcal{S}_{A_i}, d_{A_i}(x,y,z), g_{A_i}(r)\}$.

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. We have proposed to use the metamorphosis technique based on Minkowski sums that copes with polygonal shapes of any dimension.

Whenever the argument shapes A and B are convex, the topology of the intermediate convex shapes $(1-t)A \oplus tB$ remains unchanged throughout time (there is neither creation nor destruction of vertices, edges or faces). 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 [14, 11, 6].

We recall the following fundamental results [7] that characterize our soft objects skeleton metamorphosis. The polytope $C = A \oplus B$ may be defined as the convex hull of the vertices V_C 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.

Throughout this paper, the topology will be denoted $\mathcal{T}(A \oplus B)$, and the reference table $\mathcal{R}(A \oplus B)$, both 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 $O({N_A}^2 {N_B}^2)$ computational time, where N_A and N_B stand for the number of vertices of A and B. Accelerated techniques that take advantage of

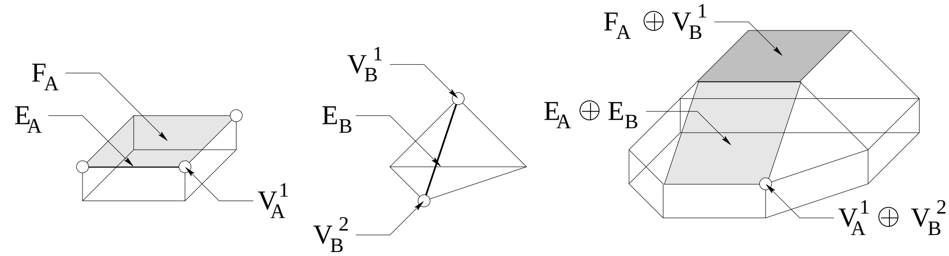

Figure 1. Parent vertices of the Minkowski sum $A \oplus B$

the topology of argument polyhedra have been proposed [7] and lower the overall complexity to $N_C\, ln(N_C)$, where N_C stands for the number of vertices of C.

2.3 Soft object metamorphosis

In [8], we have proposed to characterize the whole transformation by a *generic soft object model* whose instanciations interpolate the initial and the final shapes throughout time. We recall here the main steps of the method.

Given a rough graph of correspondence matching parts of the initial and the final shapes A and B, *i.e.* sets of components A_i and B_j, we split the components of those shapes into sub-components $s_k(A_i)$ and $s_k(B_j)$ with a view to creating a new graph bijectively matching those sub-components [1]. Initial and final components that are not matched involve the creation of specific null components.

Since the splitting of skeletons into convex sub-skeletons leads to unwanted bulges [4], we have proposed to keep the skeletons unchanged and to split the potential functions into sub-functions. Therefore, the skeletons $\mathcal{S}_{s_k(A_i)}$ and $\mathcal{S}_{s_k(B_j)}$ are the same as $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$. The distance functions $d_{s_k(A_i)}(x,y,z)$ and $d_{s_k(B_j)}(x,y,z)$ are also the same as $d_{A_i}(x,y,z)$ and $d_{B_j}(x,y,z)$, whereas the potential functions $g_{s_k(A_i)}(r)$ and $g_{s_k(B_j)}(r)$ are weighted replications of $g_{A_i}(r)$ and $g_{B_j}(r)$, which preserves shape :

$$\begin{cases} g_{s_k(A_i)}(r) = \alpha_{ik}\, g_{A_i}(r) \\ g_{s_k(B_j)}(r) = \beta_{kj}\, g_{B_j}(r) \end{cases}$$

Several heuristics for computing those weights α_{ik} and β_{kj} have been proposed, a straight forward technique may consist in setting weights as follows : $\alpha_{ik} = 1/N_{A_i}$ and $\beta_{kj} = 1/N_{B_j}$ where N_{A_i} and N_{B_j} refer to the number of components linked with the initial and final components A_i and B_j respectively.

[1] The set of indexes j of target components B_j associated to an initial component A_i will be denoted as $J(i)$, and $I(j)$ will refer to the set of indexes i of source elements A_i linked with final components B_j respectively.

An intermediate sub-component $C_{ij}(t)$ is associated to each graph link, and intermediate shapes are defined by summing their influences. Each intermediate component $C_{ij}(t)$ is characterized by :

- a time varying skeleton $\mathcal{S}_{C_{ij}(t)}$, defined as the linear interpolation based on Minkowski sum of the skeletons of the initial and final associated sub-components $\mathcal{S}_{C_{ij}(t)} = (1-t)\mathcal{S}_{A_i} \oplus t\mathcal{S}_{B_j}$,
- a time dependent distance and potential functions $d_{C_{ij}(t)}(x,y,z)$ and $g_{C_{ij}(t)}(r)$ that interpolate their initial and final counterparts.

Since argument skeletons are convex polygonal shapes, the linear interpolation based on Minkowski sums implicitly creates a generic skeleton that smoothly interpolates the initial and the final skeletons, whatever their dimension [6, 7].

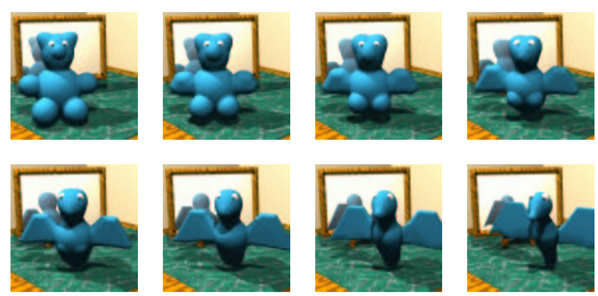

Figure 2. Metamorphosis between a bear and a bird

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* that preserves their shape coherence.

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

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

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.

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.

3 Shape and path control

In the method proposed in [8], the only way to control the transformation was to modify the graph of correspondence matching initial and final shapes, or to curb the splitting of potential functions by altering the weights. However, intermediate amorphous blobby shapes (figure 3) may arise during the transformation whenever the following conditions are met:

- the interpolation based Minkowski sum implicitly generates a path for each component, therefore components may cross during the transformation which may generate weird intermediate shapes,

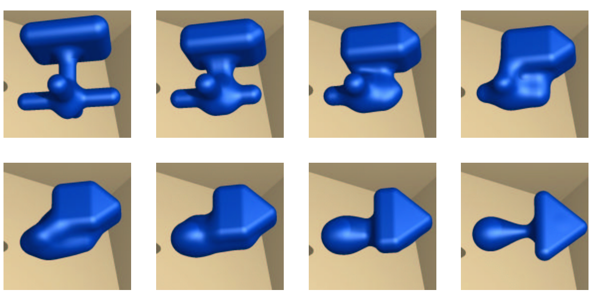

Figure 3. Direct metamorphosis of two shapes resulting into amorphous intermediate stages

- although the Minkowski sum implicitly combines convex polygonal shapes, it may also increase the dimension of the resulting intermediate shapes, *e.g.* the transition between two non colinear line segments generatses a deforming polygon, and the Minkowski sum two non-colinear line segment generates a deforming polygon.

Both those effects may appear as undesirable, and we propose an alternative technique that controls both the *dimension* of the intermediate skeletons and their *trajectory path* during the transformation.

3.1 Transformation of skeletons

In [8], we used to create the Minkowski sum of the skeletons $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$ of the components A_i and B_j by creating the convex hull of the vertices $V_{\mathcal{S}_{A_i}} \oplus V_{\mathcal{S}_{B_j}}$ whose coordinates where taken in the world coordinate system, denoted as $\mathbf{R}_0$, which provided no control over the transformation, the deforming skeleton $\mathcal{S}_{C_{ij}(t)}$ used to be defined as:

$$\mathcal{S}_{C_{ij}(t)}|_{\mathbf{R}_0} = (1-t)\mathcal{S}_{A_i}|_{\mathbf{R}_0} \oplus t\mathcal{S}_{B_j}|_{\mathbf{R}_0}$$

We propose that a local coordinate system should be linked to the skeleton of each component, thus a component A_i of a soft object A will be fully characterized by its coordinate system $\mathbf{R}_{A_i}$, its skeleton $\mathcal{S}_{A_i}$, its distance function $d_{A_i}(x, y, z)$ and its potential function $f_{A_i}(r)$, and referred to as $\{\mathbf{R}_{A_i}, \mathcal{S}_{A_i}, d_{A_i}(x, y, z), f_{A_i}(r)\}$.

Given two soft objects A and B, we define the transformation between two skeletons $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$ as follows:

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

The Minkowski sum is created in a time varying ghost coordinate system, denoted as $\mathcal{R}_{C_{ij}(t)}$, with the only very constraint that it should interpolate the initial and the final coordinate systems $\mathbf{R}_{A_i}$ and $\mathbf{R}_{B_j}$. We address the transformation of those coordinate systems in the following section. The choice of a local coordinate system associated to the skeletons $\mathcal{S}_{A_i}$ and $\mathcal{S}_{B_j}$ provides the animator with a wide span of visual effects.

The process may be automated with a view to systematically reducing the dimension of the varying skeletons, *e.g.* a straight forward heuristic may consist in assuming that all line segments skeletons $\mathcal{S}_{A_i}$ are along the (O, i) axis of $\mathbf{R}_{A_i}$, and that all polygons are in the (O, i, j) plane of $\mathbf{R}_{A_i}$.

3.2 Trajectory control

Given an initial and an final coordinate systems $\mathbf{R}_{A_i}$ and $\mathbf{R}_{B_j}$, characterizing intermediate steps may be achieved by several interpolation techniques. A complete overview is beyond the scope of this paper, however we briefly present the technique we rely on.

Interpolating $\mathbf{R}_{A_i}$ and $\mathbf{R}_{B_j}$ may be achieved by keyframing techniques where the animator defines control intermediate coordinate systems (figure 4). The path followed by the origin of the coordinate system $\mathcal{R}_{C_{ij}(t)}$ may be created thanks to a vast variety of splines, whereas the smooth

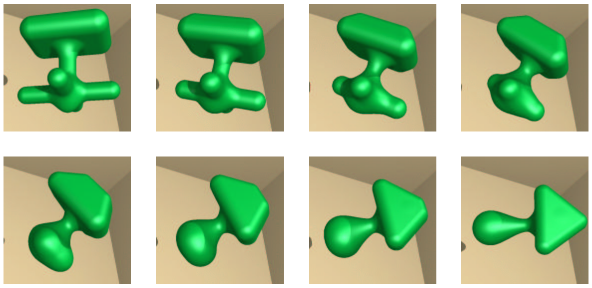

Figure 4. Metamorphosis of two shapes with local coordinate systems

interpolation of the orientations involves quaternion slerping [16, 1].

4 Metamorphosing a blob with a graph structure

A very interesting feature of soft objects is their blending property, however, achieving control so that the blending occurs only when it is desirable or specifying specific blending types requires the use of a tree or a graph characterizing the blending relationships between elements.

In this section, we address two kinds of graphs : *non-blending graphs* that characterize a boolean relationships between argument components by specifying which parts of a soft object should or should not blend, and *heterogeneous blending graphs* that characterize the different blending types involved in the computation of the potential field, *e.g.* union-like blends or super-elliptic blends.

Before any further explanations on the metamorphosis of soft objects bearing a graph structure, we address the outcomings of the splitting process over the graphs' structure defining the blending relationships in the next section.

4.1 Graph splitting

In this section, we consider two initial and final soft objects A and B with blending graphs denoted as $\mathcal{G}_A$ and $\mathcal{G}_B$ respectively whose arcs, denoted as pairs of elements A_i and $A_{i'}$, bear the blending relationships $\mathcal{B}_{ii'}(A_i, A_{i'})$ and $\mathcal{B}_{jj'}(B_j, B_{j'})$ respectively. [2]

We recall that the first step of our metamorphosis technique consists in splitting an original graph of correspondence matching elements of A and B with a view to creating a generic representation of the intermediate soft object. In this step, we create a new representation of A and B with split components $C_{ij}(0)$ and $C_{ij}(1)$ respectively. Thus, we also split the graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ so as to achieve a new representation with split components $C_{ij}(0)$ and $C_{ij}(1)$, the links are created according to the following rules :

- create *blending arcs* whose blending function is the sum of argument components' field functions between elements $C_{ki}(0)$ and $C_{kj}(0)$, *i.e.* that have been created from the same initial component A_k, and create same blending links between components $C_{ik}(1)$ and $C_{jk}(1)$, *i.e.* that have been created from the same final component B_k respectively,
- replicate the link type between two different elements $C_{ij}(0)$ and $C_{kl}(0)$ so as to preserve the arc between elements A_i and A_k, and replicate the link type between elements $C_{ij}(1)$ and $C_{kl}(1)$ according to the existing arc between elements B_j and B_l, this preserves the *blending type*.

Thus, the graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ are split into new graphs denoted $\mathcal{G}_{C(0)}$ and $\mathcal{G}_{C(1)}$ respectively that characterize the blending type of components $C_{ij}(0)$ and $C_{ij}(1)$ so that $C(0)$ and $C(1)$ should be the same as A and B respectively.

4.2 Non-blending graph transformation

A very interesting feature of soft objects is their blending property, however achieving control so that the blending occurs only when it is desirable still remains a difficult issue widely addressed [12, 5, 10]. A general solution consists in defining a blending graph between the different components of a soft object, and stating that an element's field only blends with linked elements.

Although articulated structures preserve their blending graph throughout the animation, a metamorphosis step may involve the transformation of this blending graph : two non-blending (respectively blending) components of an initial soft object A may be respectively matched with two blending (respectively non-blending) components of a final object B. Thus, we address the characterization of smooth transitions between blending and non-blending states.

During the splitting step, primitives of the initial and the final shapes are split and the new links are created, however, splitting the original blending graph only creates new links bearing a boolean flag specifying whether two primitives should or should not blend, therefore, the new split graph still remains valid in the sense that the potential field may be computed at whatever point of space.

We recall that the interpolating soft object $C(t)$ shares the same components as the initial and the final split soft objects $C(0)$ and $C(1)$, therefore, the initial and the final split graphs patterns $\mathcal{G}_{C(0)}$ and $\mathcal{G}_{C(1)}$ may be associated to intermediate components, which results in the creation of

[2] The relationships between components have been set to integer values out of simplicity, relationships with a 1 mark stand for the standard sum of argument primitives.

two soft objects' fields that may be defined as $f_{C_{ij}(t),\mathcal{G}_{C(0)}}$ and $f_{C_{ij}(t),\mathcal{G}_{C(1)}}$. Interpolating those fields as well as the initial and the final threshold values smoothly interpolates soft objects A and B. It is worth noticing that the intermediate soft object $C(t)$ has no specific graph $\mathcal{G}_{C(t)}$, and we will refer to the following notation : $\mathcal{G}_{C(t)} = 0$.

$$f_{C_{ij}(t),0} = (1-t) f_{C_{ij}(t),\mathcal{G}_{C(0)}} + f_{C_{ij}(t),\mathcal{G}_{C(1)}}$$

The algorithm for computing this field value is straight forward and may be split into three steps :

- compute each component's contribution $f_{C_{ij}(t)}$,
- compute the intensities $f_{C_{ij}(t),\mathcal{G}_{C(0)}}$ and $f_{C_{ij}(t),\mathcal{G}_{C(1)}}$, taking both graph $\mathcal{G}_{C(0)}$ and $\mathcal{G}_{C(1)}$ into account successively,
- interpolate the resulting intensities.

This technique creates a generic dependency between the intermediate soft objects and the initial and final shapes.

4.3 Heterogeneous blending graph transformation

So far, blending graphs have been used to define unwanted blending relationships between elements, we propose to use them with a view to defining specific blending properties between pair of components A_i and B_j through *blending types* $\mathcal{B}_{ij}$ and blending functions referred to as $\mathcal{B}_{ij}(f_{A_i}, f_{B_j})$.

Although sum blending, which may be defined as the sum of intensities generated by two elements, have been widely used, several other techniques have been proposed, either in the scope of bulge elimination [4] or in the scope of real function modeling [13]. The traditional blending function is the simple sum of argument potential fields, whereas the union of two components may be achieved by taking the maximum value of argument primitives [13]. Other blending functions may be addressed, for example super-elliptic blending functions [18, 3, 4] may be defined as follows :

$$\mathcal{B}_{ij}(f_{A_i}, f_{A_j}) = \left({f_{A_i}}^n + {f_{A_j}}^n\right)^{1/n}$$

Note : super-elliptic blends generate intermediate blending properties between the sum and the union, *i.e.* increasing the exponent smoothly reduces the blending bulge.

In this section, heterogeneous blending graphs will refer to the graph of a soft object whose blending relationships, *i.e.* blending functions, between elements are different, whereas homogeneous blending graphs will refer to traditional blending that generally only involves the sum of contributing elements' intensities.

In the following sub-sections, the potential field of a soft object A will be fully characterized by the set of its components A_i, its graph $\mathcal{G}_A$, and denoted as $f_{A_i,\mathcal{G}_A}(x,y,z)$, the soft object itself includes a threshold value T_A, and will be referred to as $\{f_{A_i,\mathcal{G}_A}(x,y,z), T_A\}$.

Different blending operators may be coded as nodes in a tree so as to combine primitives much like CSG operators operate on solid primitives : this is a function representation of solids [15, 13] that defines solid shapes as a equipotential representation of a complex potential field generated by combining several primitives.

As mentioned earlier, the metamorphosis of the so defined initial and final shapes may be achieved by avoiding the correspondence process and by directly interpolating the field functions of the initial and the final shapes [19, 13]. In certain cases, the tree representation of both initial and final shapes and the graph of correspondence may yield a time varying interpolating tree However, interpolating two soft objects based on a tree representation of primitives with any kind of graph of correspondence proves to be in general impossible.

Therefore, we propose to transform initial and final shapes' trees into graphs whose arcs will bear the heterogeneous blending relationships, and apply our graph splitting technique. The overall algorithm may be split into four steps :

- create the graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ from the trees $\mathcal{T}_A$ and $\mathcal{T}_B$ of A and B by distributing the relationships at the nodes of the trees to the arcs of the graph,
- split the graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ so as to achieve the new graph representation $\mathcal{G}_{C(0)}$ and $\mathcal{G}_{C(1)}$,
- create both the generic graph and the soft object models characterizing the transformation between $C(0)$ and $C(1)$,
- define the whole metamorphosis as a three step transformation : first transform A into $C(0)$ by interpolating the potential fields, then transform it into $C(1)$, and eventually transform this shape into B through direct interpolation of the potential fields.

We call this technique weak graph interpolation for the very reason that we transform the difficult tree interpolation problem into a more straight-forward graph interpolation problem (figure **??**).

4.3.1 Heterogeneous blending functions

Heterogeneous blending relationships between components makes it difficult to characterize the potential field as a C^0 continuous function. Thus, we propose an adapted formula that defines the potential field as a weighted sum of the blending functions involved in the heterogeneous graph and creates a C^0 continuous function.

Several techniques for computing the field function of a soft object involving a blending graph have been proposed [12, 5, 10]. However, those techniques focus on non-blending relationships, and do not cope with heterogeneous blending graphs. In this section, we wish to compute the potential field at a point of space $M(x,y,z)$ that belongs to the area of influence of a set of skeletons which blend one another with heterogeneous blending schemes. The very issue is to characterize heterogeneous blending relationships between components through a generic formula with a view to preserving the overall genericity of the metamorphosis.

In this section, we consider a soft object A with its blending graph $\mathcal{G}_A$ whose arcs (A_i, A_j) involve a specific blending function $\mathcal{B}_{ij}(f_{A_i}, f_{A_j})$, and address the computation of the potential field $f_A(x,y,z)$ at a point $M(x,y,z)$ of space that is within the range of influence of N components A_i of A. R_{A_i} refers to the radius of influence of the element A_i, and r_i refers to the distance between M and the skeleton $\mathcal{S}_{A_i}$.

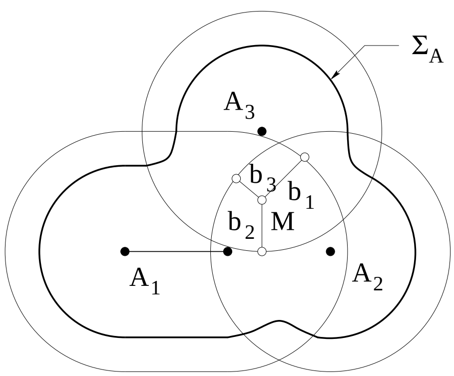

Figure 5. Computation of the resulting potential field with three elements involved in a heterogeneous weak blending graph

Since we wish to create a C^0 continuous function, we want the blending relationships level $\mathcal{B}_{ij}(f_{A_i}, f_{A_j})$ to lose influence whenever $M(x,y,z)$ comes closer to the border of elements A_i and A_j (figure 5). Thus, we propose to use the following border distance decreasing function as a weighting function :

$$b_i = \left(1 - \left(\frac{r_{A_i}}{R_{A_i}}\right)^2\right)^2$$

With those notations, we define $f_A(x,y,z)$ as :

$$f_A = \frac{\displaystyle\sum_{i=1}^{N_A-1}\sum_{j=i+1}^{j=N_A} b_i b_j \mathcal{M}\left(f_{A_i}, f_{A_j}\right)}{\displaystyle\sum_{i=1}^{N_A-1}\sum_{j=i+1}^{j=N_A} b_i b_j}$$

Note : $f_A(x,y,z)$ is a C^0 continuous function if the blending functions $\mathcal{B}_{ij}(f_{A_i}, f_{A_j})$ as well as the potential functions $f_{A_i}(x,y,z)$ are also C^0 continuous. Moreover, we think that this formulation could provide C^1 continuity provided the blending functions $\mathcal{B}_{ij}(f_{A_i}, f_{A_j})$ should prove to be C^1 continuous, however this assumption has not been rigorously demonstrated yet.

We point out that whenever the blending functions $\mathcal{B}_{ij}(f_{A_i}, f_{A_j})$ are the standard blending sum operation, *i.e.* $\mathcal{B}_{ij}(f_{A_i}, f_{A_j}) = f_{A_i} + f_{A_j}$, the resulting potential field $f_A(x,y,z)$ given by the above formula slightly differs from the expected formula :

$$f_A(x,y,z) = \sum_{k=1}^{k=N_A} f_{A_k}(x,y,z)$$

if the number of components influencing a point of space is at least three. Therefore, the shape of a soft object A with a tree is different from the shape of A with its graph $\mathcal{G}_A$, however, changes are rather small.

4.3.2 Blending graph interpolation

In the next section, we address the transformation of the blending graph during the splitting process. The blending graphs $\mathcal{G}_{C(0)}$ and $\mathcal{G}_{C(1)}$ share the same nodes, which are C_{ij} components, whereas their arcs are different.

In this section, we address the characterization interpolated blending types so as to define time varying arcs and interpolate the different blending states, therefore, this technique fully characterizes the intermediate graph $\mathcal{G}_{C(t)}$, and the time varying potential field of intermediate soft objects may be defined as $f_{C_{ij}(t), \mathcal{G}_{C(t)}}$.

Blending function interpolation

We propose to define mixed blending as an original solution for characterizing intermediate steps between two different blending states. Given the potential fields $f_{A_i}(x,y,z)$ and $f_{A_j}(x,y,z)$ of elements A_i and A_j of a soft object A, we assume that those fields may blend according to different *blending functions* denoted $\mathcal{B}^0_{ij}(f_{A_i}, f_{A_j})$ and $\mathcal{B}^1_{ij}(f_{A_i}, f_{A_j})$, intermediate blending states may be defined as :

$$\mathcal{B}^\lambda_{ij}(f_{A_i}, f_{A_j}) = (1-\lambda)\mathcal{B}^0_{ij}(f_{A_i}, f_{A_j}) + \lambda\mathcal{B}^1_{ij}(f_{A_i}, f_{A_j})$$

In the general case, λ characterizes the blending type ratio between $\mathcal{B}^0_{ij}$ and $\mathcal{B}^1_{ij}$.

Characterization of intermediate blending graph links

Given the initial and the final split blending graphs, we propose to create the intermediate generic blending graph $\mathcal{G}_{C(t)}$ by keeping all the nodes and characterizing all the time varying arcs between two components $C_{ij}(t)$ and $C_{i'j'}(t)$ as follows:

- if the arc $(C_{ij}(0), C_{i'j'}(0))$ is the same as the arc $(C_{ij}(1), C_{i'j'}(1))$, then keep it throughout the transformation,
- if it is not the case, define $(C_{ij}(t), C_{i'j'}(t))$ as a time varying blending arc whose blending type interpolates the blending types of $(C_{ij}(0), C_{i'j'}(0))$ and $(C_{ij}(1), C_{i'j'}(1))$.

This technique creates a generic dependency between the intermediate soft objects and the initial and final shapes.

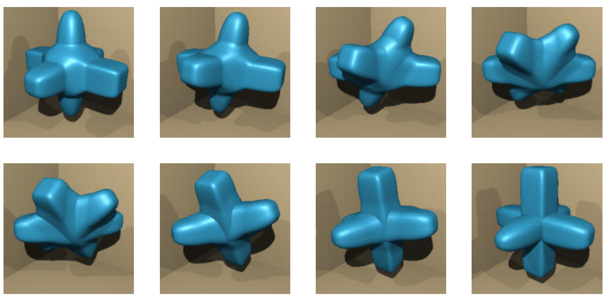

Figure 6. Soft object metamorphosis with straightforward blending tree transformation

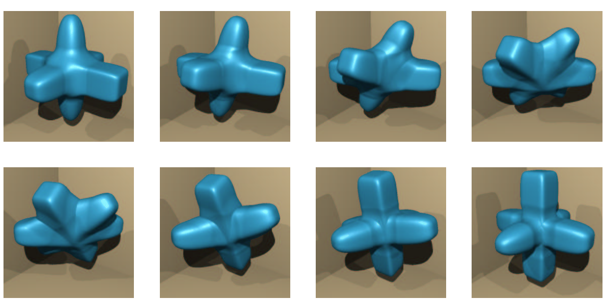

Figure 7. Soft object metamorphosis with weak blending graph transformation

5 Results

Both direct tree interpolation technique and weak graph metamorphosis have been tested an compared using simple tree and graph structures. The following two color plates illustrate the transformation between the sum of three superellipsoidal primitives and their union. In figure 6, the blending bulge smoothly stiffens and disappears only to turn into a sharp edged global union of primitives, whereas in figure 7, the final object gets slightly hollowed because of the specific formula involved in the computation of the potential field with heterogeneous blending graph.

6 Conclusion

In this paper, we have proposed several improvements to our soft object metamorphosis technique that provides the animator with a tighter control over the intermediate shapes : those are the control of the dimension and the design of trajectory paths for intermediate elements. We have also defined an original mixed blending algorithm for interpolating both non-blending graphs and heterogeneous blending graphs and proposed a new potential field characterization that is fully compatible with our metamorphosis technique and correctly handles loops in the topology. We recall that the whole transformation is still characterized by a generic representation that 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.

References

[1] A.H. Barr, B. Currin, S. Gabriel, J.F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions *Computer Graphics (Siggraph'92 Proceedings)* **26** : 313-320, July 1992.

[2] B. Crespin, C. Blanc, C. Schlick. Extended field functions for soft objects. *Proceedings of First Eurographics Workshop on Implicit Surfaces'95*, 21-32, Grenoble, France, April 1995.

[3] J. Bloomenthal, B. Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, **24**(2) : 109-116, January 1990.

[4] J. Bloomenthal. Bulge Elimination in Implicit Surface Blends. *Proceedings of First Eurographics Workshop on Implicit Surfaces'95*, 7-20, Grenoble, April 1995.

[5] M. Desbrun, N. Tsingos, M.P. Gascuel. Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation. *First Eurographics Workshop on Implicit Surfaces*, 171-185, Grenoble, April 1995.

[6] E. Galin, S. Akkouche. Spécification d'un logiciel interactif de déformations d'objets volumiques tridimentionnels basé sur les sommes de minkowski. *Congrès Interface des Mondes Réels et Virtuels*, 73-81, Montpellier, Février 1994.

[7] E. Galin, S. Akkouche. Métamorphose d'objets polyédriques, quelques méthodes d'accélération. *Technique et Science Informatiques*, **15**(3) : 329-351, April 1996.

[8] E. Galin, S. Akkouche. Soft Object Metamorphosis based on Minkowski sums. *Proceedings of Eurographics'96*, **15**(3) : 143-153, August 1996.

Gascuel 93

[9] M.P. Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics (Siggraph'93 Proceedings)*, 313-320, August 1993.

[10] A. Guy, B. Wyvill. Controlled Blending For Implicit Surfaces. *Proceedings of First Eurographics Workshop on Implicit Surfaces'95*, 107-112, Grenoble, France, April 1995.

[11] A. Kaul, J. Rossignac. Solid interpolating deformations, construction and animation of pips. *Computer and Graphics*, **16**(1) : 107-115, January 1992.

[12] A. Opalach, S. Maddock. Implicit Surfaces : Appearance, Blending and Consistency. *Proceedings of Fourth Eurographics Workshop on Animation and Simulation*, , Barcelona, Spain, 1993.

[13] A. Pasko, V. Savchenko. Constructing functionally defined surfaces. *Proceedings of First Eurographics Workshop on Implicit Surfaces'95*, 97-106, Grenoble, France, 1995.

[14] J. Serra. *Image Analysis and Mathematical morphology*, Academic Press Inc., 1982.

[15] V. Shapiro. Real functions for representation of rigid solids. *Computer Aided Geometric Design*, **12**(2) : 153-175, 1994.

[16] K. Shoemake. Animating Rotation with Quaternion Curves. *Computer Graphics (Siggraph'85 Proceedings)*, **19** : 245-254, July 1985.

[17] B. Wyvill, C. Mc.Pheeters, G. Wyvill. Data Structure for Soft Objects. *The Visual Computer*, **2**(4) : 227-234, 1986.

[18] B. Wyvill, G.Wyvill. Field functions for implicit surfaces. *The Visual Computer*, **5** : 75-82, 1989.

[19] B. Wyvill. Metamorphosis of implicit surfaces. *Modeling, Visualizing and Animating with Implicit Surfaces* (*Siggraph'93* course notes Number 25), Anaheim, CA, USA, August 1993.