THE MORPHING SPACE

Marc Alexa and Wolfgang Müller

Department of Computer Science, Interactive Graphics Systems Group
Darmstadt University of Technology, Rundeturmstr. 6, 64283 Darmstadt, Germany
{alexa|mueller}@gris.informatik.tu-darmstadt.de

ABSTRACT

Morphing is a well known technique to generate smooth transitions between two objects. We propose a more general understanding of morphing: First, we use morphing to describe objects as a composite of other objects. Second, we allow this description to incorporate more than two base objects. Given this concept a set of objects produced by morphing among multiple objects forms a mathematical space, which we call morphing space. We present a mathematical framework to discuss the properties of such spaces and their relation to the applied morphing algorithms for synthesizing the elements. We show that not only synthesis of objects but also analysis of objects is possible in a morphing space. Analysis results in the definition of an object in terms of the base objects of which it is comprised. Algorithms are proposed for both the synthesis and the analysis process.

Keywords: morphing, warping, vector space, synthesis, analysis

INTRODUCTION

It is a property of nature that objects change their shape. Morphing describes such an object metamorphosis in terms of the interpolation between two base objects.

In the past, morphing was successfully applied in various application domains and a number of morphing techniques for different problem areas were developed. Surprisingly, until today morphing has been applied mainly to generate in-betweens of two objects. However, there is no reason to limit the approach in this way. We can easily extend the idea to describe mixtures of three, four, or more objects.

Moreover, morphing has only been used to synthesize objects to generate object transitions. The strength of morphing as a formal, domain specific description mechanism for objects has not been fully exploited.

Obviously, we can use morphing to describe objects in terms of a composite between a number of base objects. All these 'composites' form a mathematical space. Similar to traditional morphing, we can use this space to derive smooth transitions between objects or to generate composites between the base objects. In the case of the latter we are speaking of a synthesis of objects. Since we have a space of objects, we can also analyze an object in order to define it in terms of the base objects of which it is comprised. That is, given an object we find the shares of the base object necessary to produce such an object with morphing.

Related Work

Feature-based image morphing was first introduced in the areas of film by Smythe [Smthe90a] for the feature films *Willow*, *Indiana Jones and the last crusade* and *The Abyss*. These and other traditional applications are discussed in detail in Wolberg's monograph on image warping [Wolbe90a]. In order to make the feature specification easier Beier and Neely presented a technique using line transformations, which was used in the production of Michael Jackson's *Black or White* video clip [Beier92a]. Recent image morphing algorithms elaborate on several parts of the morphing process ([AradR95a], [LeeWo96a], [LeeCh96a], [Litwi94a], [Rupre95a]).

Techniques for morphing between volume models make use of representations in the frequency domain [Hughe92a], the wavelet domain [HeWan94a] or the spatial domain. The latter can be seen as extensions of image morphing.

The idea of morphing between shapes was first discussed for 2d-models, more specifically polygons. Sederberg and Greenwood [Seder92a] propose a physically based approach to establish the correspondence between edges of two polygons. The transformation of the polygons was refined to use an implicit descripition in [Seder93a]. Shapira and Rappoprt make use of star-skeleton represesentation of polygons [Shapi95a].

The usefullness of morphing between polyhedral objects is known for a long time. First attempts can be found in [ChenP89a],[HongM88a] and [KaulR91a]. An almost general solution for genus 0 objects was introduced by Kent et al. in [KentC92a]. Recently, the solution was refined in [Carme98a]. Edelsbrunner has developed an implicit shape description that allows to transform one shape into another by means of linear interpolation [Edels96a]. This shape representation is used to construct a space of shapes in [Cheng97a], thus allowing to morph among multiple shapes.

Morphing among more then two images has been proposed currently by Lee, Wolberg, and Shin [LeeWo98a]. They concentrate on combining more than two images and the resulting application-specific problems (transition control, feature specification and efficiency).

The goal of our work, however, is to give a general framework for morphing among more than two objects, independent of the objects' nature. Moreover, we extend the general understanding of morphing in two ways: First, we use morphing to *describe* objects. Second, we allow this description to incorporate more than two base objects. This view of morphing allows for not only the synthesis but also the analysis of objects.

Organization of the Paper

The remainder of this paper is structured as follows. First, the concepts of morphing functions and morphing spaces are introduced formally. We discuss a number of properties of morphing functions and morphing spaces. We present general algorithms for the synthesis and analysis of objects in morphing spaces. The different strategies for the analysis of objects in linear and non-linear morphing spaces as well as quality aspects of the analysis step are discussed in detail. We discuss general application areas to show the usefulness of this concept. Finally, we present a sample application, which shows the usefulness of our concepts.

PROPERTIES OF MORPHING SPACES

In this section we formalize the concept of the "morphing space". Certain properties of the morphing function, and, thus, the morphing space will result in efficient and elegant algorithms for the synthesis and analysis of objects.

Definition of Morphing

Morphing requires two objects as input and produces one object as output. All of these objects belong to the same set of objects. We will denote this as follows:

Axiom: Morphing operates on an **object set**.

In the following we will use the identifier Ω whenever we refer to the object set. The elements of Ω will be identified by upper-case latin letters.

We need at least an equivalence relation on Ω, which we denote with "=". Note that Ω is not necessarily ordered nor has a metric.

There are two more core properties of morphing, and a third property that is typically demanded for most morphs. These properties concern the objects produced by morphing, which are commonly referred to as in-between objects. The first two properties are covered by the following axiom.

Axiom: The set of in-between objects produced by morphing between two objects is ordered and dense.

The term dense reflects the fact, that for every two in-between objects $A < B$ there exists an in-between object C with $A < C < B$. The third property is continuity of the in-between objects. For two reasons we do not include continuity in our axioms of morphing: First, there exist usefull applications where the morph cannot be continous (morphing between non homeomorphic objects), and second, without other properties we cannot formalize continuity (for instance, we have no metric on Ω).

The easiest way to obey the axioms is to represent the in-between objects by an ordered and dense set. The canonical choice is the interval of real numbers [0,1]. We define morphing formally as follows:

Definition: Let A, B elements of Ω and $x \in [0, 1]$. We will refer to m: $\Omega \times \Omega \times R \to \Omega$ as the **morphing function**, if m is well-defined for all $x \in [0, 1]$ and the following equations hold:

$$m(A, B, 0) = A$$
$$m(A, B, 1) = B$$

Properties of Morphing Functions

We will define some common mathematical terms for morphing functions. We limit the discussion to those necessary for the properties of morphing spaces.

Definition: A morphing function is **injective**, if for all $A, B \in \Omega$ the following implication holds:

$$x \neq x' \Rightarrow m(A, B, x) \neq m(A, B, x')$$

We need the property of injectivity only for the transition parameter. Note that we can invert the morphing function on its domain exactly when it is injective. We will denote the inverse of the morphing function with w: $\Omega \times \Omega \times \Omega \to R$. The following is a characteristic property of w:

$$w(A, B, m(A, B, x)) = x$$

For the next property we need to introduce a formal description of the set of in-beween objects:

$$\psi_{AB}(a, b) = \{X \in \Omega | X = m(A, B, x) \wedge x \in [a, b]\}$$

Definition: A morphing function is **compositionable**, if the following equations hold for every

$$C = m(A, B, \zeta):$$
$$\psi_{AC}(0, 1) = \Psi_{AB}(0, \zeta)$$
$$\psi_{CB}(0, 1) = \Psi_{AB}(\zeta, 1)$$

While compositionable morphing functions are crucial for the construction of morphing spaces they also allow the discussion of continuity. In many applications a continous morph seems desirable. The axioms and the resulting definition of the morphing function did not suffice to define continiuty in the common way, since we have no metric on Ω. We have found a way to describe continous morphing functions by an analogy to theorems from calculus.

> Theorem: A compositionable morphing function m is **continous**, if every $C = m(A, B, x)$ can be represented by a composition of functions $m(A, B, \alpha)$ with every fixed $\alpha, 0 < \alpha < 1$.

The theorem is essentially the negated contraposition of the (Bolzano's) interval theorem of real calculus.

Definition of Morphing Space

Suppose we are given n base objects that are the basis of our morphing space. Intuitively, the morphing space consists of all objects that can be produced by applying the morphing function to the base objects and to objects that have been produced by applying the morphing function to the base objects, and so forth. Formally, we use the following definition:

> Definition: Given an object set Ω, objects $B_0, B_1, \ldots, B_{n-1} \in \Omega, n \in N$, and a morphing function m: $\Omega \times \Omega \times R \to \Omega$. Let the sets $\Psi_i \subseteq \Omega, i \in N$ be defined by:
> $$\Psi_0 = \{B_0, B_1, \ldots, B_{n-1}\}$$
> $$\Psi_i = \{m(C, D, x) | C, D \in \Psi_{i-1}, x \in R\}$$
> The set $(\Psi = \Psi_\infty) \subseteq \Omega$ is called the **general morphing space**.

This definition expresses the idea of the space of all objects generated by morphing between the base objects. However, it is not really helpful in practice: Since we have n base objects we are interested in an description (or representation) of elements relative to these objects but not to the actual calculation. We want to describe objects in terms of weights representing the shares of each of the base objects. In order to analyze this case we consider the **finite dimensional morphing space** $\Psi_n \subseteq \Omega$, where n is the number of base objects.
Obviously, we can represent the elements of Ψ_n uniquely by vectors of R^{n-1}. Naturally, we strive to answer the following question: What conditions must m satisfy such that $\Psi_n = \Psi_\infty$ for every set of base objects? Note that the combination of the following two conditions is sufficient:

1. The morphing function must be order independent when applied to more than two objects.
2. The morphing function must be compositionable.

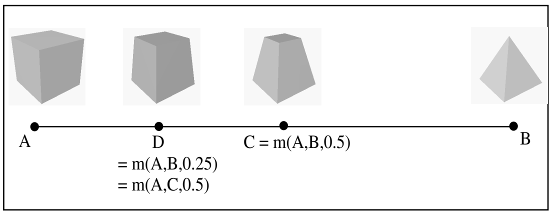

Figure 1: An illustration of a linear morphing function. Object D might be constructed directly from A and B or via C, since the transition parameter of the morphing function is linear.

We did not succeed in deducing a condition that is necessitative and commensureative, but we have found a single equation that ensures order indepence of the morphing function:

$$m(m(A, B, x), m(A, C, y), z) =$$
$$m\left(A, m\left(B, C, \frac{yz}{x + z(y - x)}\right), x + z(y - x)\right)$$

We call morphing functions that satisfy the above equation **neat**. A neat morphing function is linear. To see what we mean by linearity set $B = C$ which reduces the equation to

$$m(m(A, B, x), m(A, B, y), z) =$$
$$m(A, B, x + z(y - x))$$

This is the combination of linearity regarding the source (see also figure 1)

$$m(A, m(A, B, y), z) = m(A, B, zy)$$

and the target parameter

$$m(m(A, B, x), B, z) = m(A, B, x + z - zx)$$

The following theorem is a first step in showing that neat morphing functions imply $\Psi_n = \Psi_\infty$:

> Theorem: Linear morphing functions are compositionable.

We only sketch the proof by mentioning that the bijections $m(A, m(A, B, \zeta), z) = m(A, B, \zeta z)$ and $m(m(A, B, \zeta), B, z) = m(A, B, \zeta + z - \zeta z)$ connect the elements of the two intervals.
The linearity of neat morphing functions gives rise to the assumption that neat morphing functions make Ψ_n an affine space, or, if we pick a zero element a vector space. We proceed with proving this assumption, since the linearity of the space also implies the order indepence of the base objects.

Morphing Space as a Affine/Vector Space

We want to show that Ψ_n is an affine space, if the morphing function is neat. We prove this constructivley by picking a zero element and showing that Ψ_n is a vector space.
We pick the base element B_0 as zero element of the space and denote it with $\varnothing$. We define scalar multi-

plication as

$$\lambda A = m(\varnothing, A, \lambda)$$

for $\lambda \in R$.

The definition of the addition is inspired by the graphical addition of vectors:

$$A + B = m(\varnothing, m(A, B, \tfrac{1}{2}), 2)$$

With these definitions the proof consists of equivalence transformations using that m is neat, and is omitted.

Representation of Elements and Dimension of Ψ_n

We can write an element $A \in \Psi_n$ as

$$A = \sum_{i=0}^{n-1} x_i B_i$$

Consequently, A is completely represented by the x_i. As Ψ_n is an affine space we know that $\sum x_i = 1$. The x_i are called barycentric coordinates of A. Because we can choose only n-1 of the x_i independently the dimension of Ψ_n is bounded by n-1. If we use Ψ_n as a vector space with B_0 as zero element, we will actually use the representation $\vec{x} = (x_1, x_2, \ldots, x_{n-1})$, i.e. dismissing the weight x_0. This has the notational advantage that the null-vector represents B_0. In the remainder of the discussion we will look at Ψ_n as a vector space, because it is of greater relevance to the applications than an affine space. Nevertheless, in most cases the statements for affine spaces can be found by simply exchanging "linear" for "affine".

With the definition of scalar multiplication and addition we have readily given an algorithm that constructs an element out of its vector representation. This algorithm is derived from the representation of the elements as a weighted sum. We can also think of this construction process as a map from R^{n-1} to Ψ_n. Note that this map is at least surjective, because every element of Ψ_n has a representation in R^{n-1}. Naturally the question arises whether the map is also injective, which would make Ψ_n and R^{n-1} isomorphous.

Isomorphism between Ψ_n and R^{n-1}

As in all vector spaces we can speak of linear dependence and independence of objects according to the following definition:

> Definition: Given elements $B_0, \ldots, B_{n-1} \in \Omega$. The elements are **linear independent**, if the equation $\lambda_1 B_1 + \ldots + \lambda_{n-1} B_{n-1} = B_0$ has only the trivial solution $\lambda_0 = \ldots = \lambda_{n-1} = 0$. Otherwise the elements are **linear dependent**.

With this definition we can prove a possible isomorphism between morphing spaces and R^{n-1}. In addition to be neat the morphing function has to be injective. We will show that the linear independence of the base elements and an injective morphing function are both neccesitative and together commensureative for a morphing space to be isomorphous to R^{n-1}.

Clearly, the morphing function has to be injective, otherwise there would be no bijection between Ψ_n and R^{n-1}

If the base elements are not linear independent, there exists a base element that can be represented as a weighted sum of the remaining base elements:

$$B_0 = \lambda_1 B_1 + \ldots + \lambda_{n-1} B_{n-1}$$

But the representation of the left- and right-hand side of this equation are different: The left-hand side is $(1, 0, \ldots, 0)$ and the right-hand side $(0, \lambda_1, \ldots, \lambda_{n-1})$. Again the map from R^{n-1} to Ψ_n would not be injective, and therefore the base elements have to be linear independent.

To show that both conditions are together commensureative we look at an element

$$A = x_1 B_1 + x_2 B_2 + \ldots + x_{n-1} B_{n-1}$$

and assume in contradiction to the statement, there would exist another representation

$$A = x'_1 B_1 + x'_2 B_2 + \ldots + x'_{n-1} B_{n-1}$$

But then we have

$$x_1 B_1 + \ldots + x_{n-1} B_{n-1} = x'_1 B_1 + \ldots + x'_{n-1} B_{n-1}$$

and because of the injectivity of m all $x_i B_i, x'_i B_i$ are uniquely determined. The uniqueness permits term-transformations, and we get

$$\varnothing = (x'_1 - x_1) B_1 + \ldots + (x'_{n-1} - x_{n-1}) B_{n-1}$$

But because the basis $B_0, B_1, \ldots, B_{n-1}$ is linear independent the above equation has only the trivial solution

$$x'_1 - x_1 = \ldots = x'_{n-1} - x_{n-1} = 0$$

which means $x_i = x'_i$. Thus all representations of A are identical, or in other words the map from R^{n-1} to Ψ_n is injective.

ALGORITHMS FOR OBJECT SYNTHESIS AND ANALYSIS

The two basic operations in morphing spaces are synthesis and analysis of elements. Synthesis denotes the process of constructing an element in a given morphing space from a given vector representation. Analysis is the calculation of a vector representation for a given element in a given morphing space.

As will be seen shortly, both problems can be solved without any knowledge of the objects and the morphing technique.

Synthesis of Objects

In order to find a synthesis algorithm we will use the vector space properties of morphing spaces. With this presumption we can present an algorithm that constructs an element in a proveably optimal way, assuming all we have is a classical morphing technique. We will discuss the case of synthesis in non-linear morphing spaces later.

Assume we want to synthesize an object A with the given vector representation $\vec{x} = (x_1, x_2, \ldots, x_{n-1})$. The algorithm can be described as follows:

1. Derive A by morphing between the null object B_0 and the projection of A into the n-2-dimensional subspace spanned up by the base objects without B_0.
2. To calculate the projection identify a null object in the subspace and repeat the procedure of steps one and two.
3. If the subspace is one-dimensional calculate the projection by applying the morphing function directly.
4. Calculate the projection in the next higher-dimensional subspace, and so on, until the object is synthesized.

Object A has the representation $(x_1, x_2, \ldots, x_{n-1})$ in the original morphing space. We want to find its representation in the space with basis $B_1, \ldots, B_{n-1}$. We have to find the intersection of this subspace and a line through B_0 and A. Thus, we have to solve

$$\lambda A = B_1 + \mu_1(B_2 - B_1) + \ldots + \mu_{n-2}(B_{n-1} - B_1)$$

The resulting system of linear equations

$$\begin{bmatrix} x_1 & 1 & \ldots & 1 \\ x_2 & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ x_{n-1} & 0 & \ldots & -1 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu_1 \\ \ldots \\ \mu_{n-2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \ldots \\ 0 \end{bmatrix}$$

can be solved easily. By adding all rows we obtain

$$\lambda \sum x_i = 1 \Rightarrow \lambda = \frac{1}{\sum x_i}$$

and by applying this to the j-th row of our linear equations we find

$$\frac{1}{\sum x_i} \cdot x_j - \mu_{j-1} = 0$$

and thus

$$\mu_j = \frac{x_{j+1}}{\sum x_i}$$

The weights μ_j give the representation for A in the subspace with null element $(1, 0, \ldots, 0)$.

With repeated insertion one eventually obtains the following closed form of the procedure:

$$C_k = m\left(B_k, C_{k+1}, \frac{\sum_{i=k+1}^{n-1} x_i}{\sum_{i=k}^{n-1} x_i}\right)$$

The calculation begins with C_{n-1}, because in this case the nominator of the transition parameter is zero and we immediately obtain $C_{n-1} = B_{n-1}$. Then n-1 applications of the morphing procedure are necessary to find $C_0 = A$.

In the derivation of the algorithm we actually used the vector space properties of the morphing space. The resulting algorithm is applicable also to non-linear morphing spaces. The only problem in the unique construction of an object is the ordering of the base objects. If the morphing space is a vector space the input order of the base objects is not relevant in the construction process. It might be the case that this order is relevant for some morphing techniques. Then, the ordering of the base objects has to remain constant in all applications in order to have reproducible results. In other words, in this case the ordering of the base objects is one parameter of the morphing space.

Analysis of Objects

The analysis of objects, however, cannot be performed without the definition of a distance function on the objects. This need is due to the fact that - in general - analysis has to be done with numerical methods and optimization techniques which require some “progress indicator”. In the following, we will denote the distance function with d.

For the analysis of objects it is useful to distinguish linear and non-linear morphing spaces. If the morphing space is linear much more sophisticated algorithm exist. In non-linear morphing spaces we will further distinguish some situations.

In all cases we need to minimize the distance along (one-dimensional) lines. We employ a numerical method known as Brent’s method [Brent73a].

Note that this one-dimensional minimization is a practical solution for finding the inverse element w.

Analysis in Linear Morphing Spaces

To find the representation of a given object in the morphing space we propose the following technique. First, we project the element on the axes of the space. This is done by minimizing the distance to the elements of one axis. All these projections result in a n-1-dimensional vector. We want to explain, why this is a linear mapping:

Minimizing the distance to the elements of one axis is linear: For linear morphing spaces there exists an object representation, such that the morphing function is a simple linear interpolation. This is due to the fact that the morphing space is a vector space, and in vector spaces all points between to given points can be obtained by linear interpolation.

Assume our distance function is a norm (which is no restriction, since all mathematical distance functions are norms). Thus, the distance function itself is linear, and the iso-distance surfaces are all similar. Minimizing the distance results in a projection with a fixed angle to the axis. These projections are linear.

As we have seen, the projection onto one axis is linear. And because the composition of linear operations is linear, the mapping from the element to the vector of projections is also linear.

Thus, the actual representation is a linear mapping of the computed vector. Linear mappings can be represented by matrix multiplications. All we have to do is to find the representation matrix.
Assume the projection of an element A results in the vector $(a_1, a_2, \ldots, a_{n-1})$ while the representation we want to find is $(x_1, x_2, \ldots, x_{n-1})$. We are searching for the representation matrix M that connects those two representations:

$$M\vec{x} = \vec{a}$$

As is known from linear algebra, the rows of M consist of the images of the base vectors. Therefore we obtain M by projecting the base elements $B_1, \ldots, B_{n-1}$ onto the axes.
Note that we cannot decide, whether the above linear system has a solution. This is due to the fact that the linear independence of $B_1, \ldots, B_{n-1}$ is not known, and not calculable so far. We suggest using the Singular Value Decomposition (SVD) to analyze M.
The SVD is a decomposition of a matrix into a product of an orthogonal matrix, a diagonal matrix and again an orthogonal matrix. This decomposition is always possible [Golub83a]. The SVD sheds some light on the structure of a matrix. For most applications the values of the diagonal matrix (singular values) are of particular interest. If any of the singular values is zero, the determinant of the decomposed matrix is zero and the matrix not invertible. In our context, the above linear system would have no solution. If we replace any zero singular value by infinity, we can invert the diagonal matrix (the orthogonal matrices are invertible anyway). This way we find a best-approximation to the above linear system, no matter what the condition of M is [Press92a].
Furthermore we get a quality measure of our basis. The singular values represent the different scaling factors of the linear mapping. If these scaling factors differ substantially, the quality of the basis is obviously bad. If the scaling factors are all the same, then the basis is orthogonal and to some degree optimal. As a quality measure we use the ratio between the maximum and minimum element of the singular values. A ratio of zero shows that the base elements are linear dependent.

Analysis in Non-Linear Morphing Spaces

In a non-linear morphing space we cannot apply the above method, because the projection on the axes is not necessarily a linear mapping of the vector representation. However, we can treat the morphing space as any n-dimensional space and search for the minimum-distance element using a general minimization algorithm for n-dimensional spaces. The classical techniques assume that the minimum is unique, i.e. there is only one local minimum. In this case one can find the minimum by repeated minimizing along orthogonal lines in the space.

We propose the following scheme to minimize the distance in morphing spaces with a unique local minimum:
Given a morphing space Ψ with the basis $B_0, B_1, \ldots, B_{n-1} \in \Omega, n \in N$ constructed with a morphing function m and an element C, which has to be analyzed. Calculate $C_0 = m(B_0, B_1, n(B_0, B_1, C))$ and then iteratively

$$C_i = m(C_{i-1}, B_{i+1 \bmod n}, n(C_{i-1}, B_{i+1 \bmod n}, C))$$

With this, $C_{i \to \infty}$ converges against C.
The idea is to minimize along lines given by the actual best approximation and a base object. The index i+1 mod n results in cyclic usage of the base objects. This scheme converges due to the fact that the respective direction vectors are linear independent. The advantage is of course that in each step no object has to be predetermined to define the line along which one has to minimize.
In spaces with more than one local minimum we use a simulated annealing scheme adapted to our needs: First, calculate $C_0 = m(B_0, B_1, n(B_0, B_1, C))$. In every step, choose $x \in [0, 1]$ randomly and calculate

$$D = m(C_{i-1}, B_{i+1 \bmod n}, x)$$

The probability to set $C_i = D$ is given by

$$p = e^{\frac{d(C, D) - d(C, C_{i-1})}{kT}}$$

Thus, if D is closer to the minimum than C_{i-1} according to the distance function d, we always use D as the actual approximation (in this case p is greater than one). If D is not closer than the result depends on a randomly chosen value in the interval [0,1].

RESULTS AND APPLICATIONS

Applications can be roughly grouped in synthesis and analysis applications. A very interesting application arises if analysis and synthesis are connected, i.e one analyzes an element in one space and uses the representation in another space to synthesize an object. We will illustrate this with a sample application.

Object Synthesis Applications

The morphing space allows the description of elements as a mixture of the base objects. For complex objects, such a description is very natural and in many cases closely correlated to the way humans describe objects. We also think in categories and attributes.
Another useful property of morphing is the flexibility in defining transitions. By defining and spanning a morphing space, the number of possible transitions not only rises but can also be specified more exactly. The concrete choice of base objects allows to enforce certain properties during the transition.

Figure 2: The base objects of the morphing space used to analyze the object-sequence. The upper left object represents a neutral facial expression, the remaining objects the speaking of the vowels ‘a’, ‘e’ and ‘o’.

Object Analysis Applications

The morphing space is an object space in which all objects share the properties of the base objects. Therefore it is an ideal instrument for object analysis. In other spaces only a small fraction of the objects is directly related the object to be analyzed. In morphing spaces probably all objects have the right properties, thus making the search space smaller and easier to handle.

The analysis of objects consists of finding a vector representation of the object in such a way that the vector could be used to synthesize a best-approximation of the object. This representation can be used in at least two ways in applications:

1. We look only for a good representation of an object in the morphing space. The dimension of the morphing space will usually be much smaller than the dimension of the complete representations due to the object-specific nature of the space. This allows a very good compression of classes of objects with a common property.
2. The base objects have a certain meaning in the application’s context. With the vector representation we find information about how the analyzed object is related to those base objects.

Sample Application

In our sample application we show the mapping from one morphing space into another. The first one is used to analyze the facial expression of a human in sample images. The second one is used to synthesize the same expression on the face of an avatar. For this, both morphing spaces were constructed using a basis of “extreme” facial expressions, one with a set of images of the human speaker, one with geometric representations of the 3d avatar. Other facial expression can be represented as a mixture of these initial expressions. To map the facial expression of the speaker to the avatar we apply the vector representation of the speaker’s expression directly to synthesize an object in the second morphing space, which describes the avatar’s expressions.

In the analysis space we used the scattered data interpolation with radial basis function as a morphing function. For the avatar (see figure 3) we used linear interpolation between the polygons.

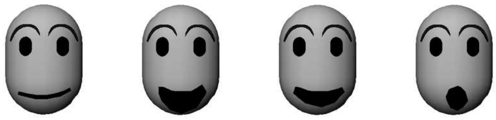

Figure 3: The base objects of the morphing space used for the synthesis of an avatar. The avatar roughly reflects the facial expression of the base images in the analysis space.

The basis consists of four images and is depicted in figure 2. The four faces represent a neutral expression, a spoken “a”, “e” and “o”.

To test this scenario, we used a video sequence of the spoken word “Hello”. Every frame was analyzed in the first morphing space in order to get a vector representation and then an avatar could be synthesized in the second space.

The results obtained with this approach are shown in figure 4. One can see, that the expression of the avatar correlates closely with the expression of the speaker. Some errors of course occur since image distance is sensitive to the unavoidable head movements.

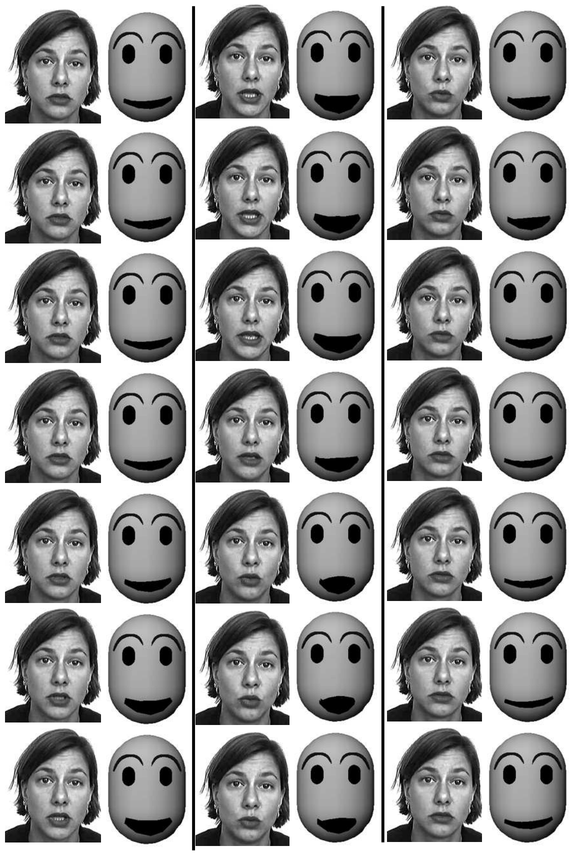

Figure 4: The sequence of synthesized avatars. Every input frame (image of the actor) was analyzed to derive a vector representation in the morphing space. The vector was used to synthesize an avatar in the corresponding morphing space of avatars.

CONCLUSION

The morphing space extends the general understanding of morphing in several ways. First, it extends standard morphing to more than two base objects. Second and even more important, it represents a general mechanism for describing an object in terms of a composite of other objects. Consequently, it can not only be used to synthesize transitions between objects but also to analyze the properties of objects, which is to find its best representation in terms of a morphing between the base objects.

Future work has to focus on efficient and robust search techniques to analyze objects. Especially the analysis of objects that are not part of the morphing space is a tackling task. The algorithms we proposed have proven to solve the general problem but cannot considered to be fast. Genetic algorithms are a promising candidate for this task. Note the close relation between morphing between two objects and the reproduction process of genetic algorithms.

We have to further investigate the conditions (e.g. number of base elements) that make morphing spcaes sufficient to solve real world analysis tasks.

REFERENCES

[AradR95a] N. Arad and D. Reisfeld: Image Warping Using few Anchor Points and Radial Basis Functions. *Computer Graphics Forum,* 14, 1, 23-29, 1995

[Beier92a] T. Beier and S. Neely: Feature-Based Image Metamorphosis. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26, 2, 35-42, 1992

[Brent73a] R. P. Brent: *Algorithms for Minimization without Derivatives.* Prentice Hall, Englewood Cliffs, 1973

[Carme98a] E. Carmel and D. Cohen'Or: Warp-guided Object-space Morphing. *The Visual Computer,* 13, 9/10, 1998

[ChenP89a] S.E. Chen and R.E. Parent: Shape Averaging and Its Applications to Industrial Design. *IEEE Computer Graphics and Applications*, 9, 1, 47-54, 1989

[Cheng97a] H. Cheng, H. Edelsbrunner, and P. Fu: *Shape Space from Deformation*. Manuscript, 1997

[DeCar96a] D. DeCarlo and J. Gallier: Topological Evolution of Surfaces. *Proceedings of Graphics Interface '96*, 194-203, 1996

[Edels96a] H. Edelsbrunner: *Deformable Smooth Surface Design*. Report rgi-96-002, Raindrop Geomagic, Inc., 1996

[HeWan94a] T. He, S. Wang, and A. Kaufman: Wavelet-based Volume Morphing. *Proceedings of Visualization 94*, 85-91, IEEE Computer Society Press and ACM SIGGRAPH, Los Alamitos, 1994

[HongM88a] T. Hong, M. Magnenat-Thalmann, and D. Thalmann: A General Algorithm for 3-D Shape Interpolation in a Facet-Based Representation. *Proceedings of Graphics Intrface '88*, 229-235, 1988

[Hughe92a] J.F. Hughes: Scheduled Fourier Volume Morphing. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26, 2, 43-46, 1992

[KaulR91a] A. Kaul and J. Rossignac: Solid-Interpolating Deformations: Construction and Animation of PIPs. *Eurographics '91*, 493-505, Eirographics Association, Amsterdam, 1991

[KentC92a] J.R. Kent, W.E. Carlson, and R.E. Parent: Shape Transformation for Polyhedral Objects. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26, 2, 47-54, 1992

[LeeCh96a] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin: Image Morphing Using Deformation Techniques. *The Journal of Visualization and Computer Animation,* 7, 1, 3-23, 1996

[LeeWo96a] S.-Y. Lee, G. Wolberg, K.-Y. Chwa, and S.Y. Shin: Image Metamorphosis with Scattered Feature Constraints. *IEEE Transactions on Visualization and Computer Graphics,* 2, 4, 337-354, 1996

[LeeWo98a] S.-Y. Lee, G. Wolberg, and S.Y. Shin: Polymorph: Morphing Among Multiple Images. *IEEE Computer Graphics & Applications*, 1, 60-73, 1998

[Lerio95a] A. Lerios, C.D. Garfinkle, and M. Levoy: Feature-Based Volume Metamorphosis. *Computer Graphics (SIGGRAPH '95 Proceedings)*, 29, 2, 449-456, 1995

[Litwi94a] P. Litwinowicz and L.Williams: Animating Images with Drawings. *Computer Graphics (SIGGRAPH '94 Proceedings)*, 28, 2, 409-412, 1994

[Press92a] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, Cambridge, 1992

[Rupre95a] D. Ruprecht and H. Müller. Image Warping with Scattered Data Interpolation. *IEEE Computer Graphics and Applications,* 15, 2, 37-43, 1995

[Seder92a] T.W. Sederberg and E. Greenwood: A Physically Based Approach to 2-D Shape Blending. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26, 2, 25-34, 1992

[Seder93a] T.W. Sederberg, P. Gao, G. Wang, and H. Mu: 2-D Shape Blending: An intrinsic Solution to the Vertex Path Problem. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27, 4, 15-18, 1993

[Shapi95a] M. Shapira and A. Rappoport: Shape Blending Using the Star-Skeleton Representation. *IEEE Computer Graphics and Applications*, 15, 44-50, Mar. 1995

[Smyth90a] D.B. Smythe: A Two-Pass Mesh Warping Algorithm for Object Transformation and Image Interpolation. ILM Technical Memo #1030, Computer Graphics Department, Lucasfilm Ltd, 1990

[Wolbe90a] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Los Alamitos, 1990