Chapter 5

Spaces of shapes from morphing

In this chapter we formalize the concept of spaces from morphing [Alexa & Müller 1998a; Alexa & Müller 1999a]. Therefore we take a closer look at existing morphing techniques in order to identify the core properties of morphing. Based on this view of morphing we will give a definition of the general spaces from morphing.

By defining morphing in an abstract way we will get a scheme to classify existing morphing techniques. Since the properties of morphing spaces will depend on the properties of the morphing techniques, the classification of morphing techniques allows to predict the properties of morphing spaces. Moreover, certain properties of the morphing space will result in more efficient and elegant algorithms for the analysis of objects. This will be shown later.

5.1 Definition of morphing

In order to find a formal definition of morphing we pose ourselves the question: What do we intuitively think of, when we speak about morphing? The answer could read as follows: Given a source and a target object, morphing will produce a transition between these two objects. The answer has a certain strength, i.e. it does not restrict morphing to a certain application. For instance, the use of morphing as an animation tool might be very common, but it does not seem to be a property of morphing itself. On the other hand, morphing cannot be reduced to the production of single objects.

Notice that we already made a first observation: Morphing operates on objects - these can be real or virtual. Morphing needs 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 5.1 *Morphing operates on an object set.*

In the following we will use the identifier Ω whenever we refer to the object set. The objects of Ω will be identified by upper-case Latin letters. Since Ω is the domain of morphing we assume $\Omega \neq \emptyset$. We want to use Ω to define properties of morphing, thus we need at least an equivalence relation on Ω , which we denote with $=$. Note that we cannot imply that Ω is ordered and we (in general) have no metric on Ω .

We believe there are only 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 5.2 *The set of in-between objects produced by morphing between two objects is ordered and dense.*

We say that morphing induces an ordering on the objects resulting from morphing. 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 useful applications where the morph cannot be continuous (morphing between non homeomorphic objects), and second, without other properties we cannot formalize continuity (for instance, we have no metric on Ω). But nevertheless continuity might be of interest in the application, and after the definition of morphing based on the above axioms we introduce a criterion to assure continuity.

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 can look at the reals of this interval as one parameter that - together with a source and target object - identifies an in-between object. We call this parameter the *transition parameter*. Thus morphing can be described as a function of three variables. Choosing the transition parameter 0 will reproduce the source object and conversely the transition parameter 1 will reproduce the target object. Formally:

Definition 5.1 *Let* $A, B \in \Omega$ *and* $t \in [0, 1]$*. We will refer to* $m : \Omega \times \Omega \times \mathbb{R} \mapsto \Omega$ *as the morphing function, if* m *is well-defined for all* $t \in [0, 1]$ *and the following equations hold:*

$$m(A, B, 0) = A \tag{5.1}$$

$$m(A, B, 1) = B \tag{5.2}$$

In most of the following discussion it will be more convenient to think of m as a generalized morphing function. That means the transition parameter of m is not limited to the interval $[0, 1]$ but can be assigned any real value. The above definition remains the same, except that [0,1] is replaced by the real numbers $\mathbb{R}$ (the interval $[-\infty, \infty]$). The transition parameters 0 and 1 will still reproduce the source and the target object, respectively. Transition parameters outside of $[0, 1]$ can be understood as an extrapolation of the objects.

5.2 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 5.2 *A morphing function is* injective*, if for all* $A, B \in \Omega$ *the following implication holds:*

$$t \neq t' \Rightarrow m(A, B, t) \neq m(A, B, t') \tag{5.3}$$

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.r.t. to the transition parameter as $w : \Omega \times \Omega \times \Omega \mapsto t$. The following is the characteristic property of w:

$$w(A, B, m(A, B, t)) = t \tag{5.4}$$

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

$$\Psi_{AB}(a, b) = \{X \in \Omega | X = m(A, B, t) \vee t \in [a, b]\} \tag{5.5}$$

Now we formalize the following idea: Suppose we produce an object C by morphing between objects A and B. A morph between A and C or C and B will result in objects that could have been produced by morphing between A and B, as well.

Definition 5.3 *A morphing function is* compositionable*, if the following equations hold for every* $C = m(A, B, \tau)$*:*

$$\Psi_{AC}(0, 1) = \Psi_{AB}(0, \tau) \tag{5.6}$$

$$\Psi_{CB}(0, 1) = \Psi_{AB}(\tau, 1) \tag{5.7}$$

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

Theorem 5.1 *A compositionable morphing function m is* continuous*, if every $C = m(A, B, \tau)$ can be represented by a composition of functions $m(A, B, \chi$ with every fixed $\chi, 0 < \chi < 1$.*

In order to prove the above theorem we show the analogy to the Bolzano theorem of real calculus. Calculate $D_1 = m(A, B, \chi)$. If $D_1 = C$ we are done. If not, identify the interval $\Psi_{AD_1}(0, 1) = \Psi_{AC}(0, \chi)$ or $\Psi_{D_1B}(0, 1) = \Psi_{AC}(\chi, 1)$ to which C belongs (remember that m is compositionable). According to the chosen interval calculate $D_2 = m(A, D_1, \chi)$ or $D_2 = m(D_1, B, \chi)$. Again if $D_2 = C$ we are done and if not the interval will be divided by the calculation of D_3. Eventually C is either represented by a D_i or by an infinite process of interval divisions. The latter representation is analogous to the Bolzano's theorem, which requires continuity of the underlying set. More specifically it is the representation of a value by interval division that needs the property of continuity.

5.3 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 5.4 *Given an object set Ω ,* base *objects $B_0, B_1, \ldots, B_{n-1} \in \Omega$, and a morphing function m: Let the sets $\Phi \subseteq \Omega$ be defined by:*

$$\Phi_0 = \{B_0, B_1, \ldots, B_{n-1}\} \tag{5.8}$$

$$\Phi_i = \{m(C, D, t) | C, D \in \Phi_{i-1}, i \geq 1\} \tag{5.9}$$

The set $(\Phi = \Phi_\infty) \subseteq \Omega$ is called the general morphing space.

This definition expresses the idea of the space of all objects generated by morphing between the base objects but 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: Every element $A \in \Phi$ has a representation $a \in \mathbb{R}^n$ with vector elements a_i representing the share of base object B_i. In order to analyze this case we restrict the defintion of general morphing spaces as follows:

Definition 5.5 *Given an object set* Ω *,* base *objects* $B_0, B_1, \ldots, B_{n-1} \in \Omega$*, and a morphing function* m*: Let the sets* $\Phi \subseteq \Omega$ *be defined by:*

$$\Phi_0 = B_0 \tag{5.10}$$

$$\Phi_i = \{m(A, B_i, t) | A \in \Phi_{i-1}, i \geq 1\} \tag{5.11}$$

The set $\Phi_n \subseteq \Omega$ is called the finite dimensional morphing space.

Obviously, we can represent the elements of Φ_n uniquely by vectors in $\mathbb{R}^n$. This might be useful in some special applications but in general this will result in two major drawbacks:

1. The space is limited by the fixed order of base objects, i.e. the space does not contain all objects that might be produced by morphing among the base objects.

2. The space is not closed against the morphing function, i.e. morphing between two elements of the space can result in a non-member of the space (this is due to the limited dimension).

Therefore, we strive to answer the following question: What conditions must m satisfy such that $\Phi_n = \Phi_\infty$ for every set of base objects? Note that the combination of the following two conditions seems to be sufficient:

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

2. The morphing function must be compositionable.

Since we have a condition for the second requirement, we only have to find a condition for the first. In order to present a sufficient criterion we first introduce linear morphing functions.

Definition 5.6 *A morphing function is* linear *if for all* $A, B \in \Omega$ *the following equation holds:*

$$m(m(A, B, x), m(A, B, y), z) = m(A, B, x + z(y - x)). \tag{5.12}$$

Interestingly, linearity encompasses composition:

Theorem 5.2 *Linear morphing functions are compositionable.*

In contradiction to the statement we assume the linear morphing function m is not compositionable. We investigate an element $C = m(A, B, \tau)$ and the sets $\Psi_{AC}(0, 1)$ and $\Psi_{CB}(0, 1)$. Since m is not compositionable at least one of the sets is not equal the corresponding sets $\Psi_{AB}(0, \tau)$ and $\Psi_{AB}(\tau, 1)$. But the bijection

$$m(A, m(A, B, \tau), z) = m(A, B, \tau z)$$

connects the elements of the first intervals and the bijection

$$m(m(A,B,\tau),B,z) = m(A,B,\tau+z-\tau z)$$

connects the elements of the second intervals.

However, linearity is still not enough to ensure order independence. For that we also need distributivity.

Definition 5.7 *A morphing function is* distributive *if for all $A, B \in \Omega$ the following equation holds:*

$$m(m(A,B,x),m(A,C,x),z) = m(A,m(B,C,z),x) \tag{5.13}$$

We have found a single equation that enforces both properties linearity and distributivity:

$$m(m(A,B,x),m(A,B,y),z) = m\left(A, n\left(B, C, \frac{yz}{x+z(y-x)}\right), x+z(y-x)\right) \tag{5.14}$$

We call morphing functions that satisfy the above equation *neat*. These morphing functions fulfill all requirements for $\Phi_n = \Phi_\infty$. Furthermore, the linearity of neat morphing functions gives rise to the assumption that neat morphing functions make Φ_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 independence of the base objects. We will discuss later on what existing morphing techniques could be described in terms of a neat morphing function.

5.4 Morphing space as a affine/vector space

We believe that Φ_n is an affine space, if the morphing function is neat. We found it advantageous to prove this by picking a zero element in the space and to prove that Φ_n together with this zero element is a vector space.

A vector space (or linear space) is characterized by its domain with an addition and a scalar multiplication. In order to show that the morphing space can be a linear space we need a definition for these two operations. We pick the base element B_0 as zero element of the space and denote it with 0. Now we can define the scalar multiplication.

Definition 5.8 *A scalar multiple $\mathbb{R} \times \Phi \mapsto \Phi$ of an element $A \in \Phi$ is given by*

$$\lambda A = m(0, A, \lambda) \tag{5.15}$$

for $\lambda \in \mathbb{R}$.

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

Definition 5.9 *The addition $\Phi \times \Phi \mapsto \Phi$ of two elements $A, B \in \Phi$ is given by:*

$$A + B = m\left(0, m\left(A, B, \frac{1}{2}\right), 2\right) \tag{5.16}$$

Note here that we are making use of a more general understanding of a morphing function with a transition parameter $t \in \mathbb{R}$. Now one can show, that the morphing space with the above operations is a vector space, if the morphing function is neat. The proof consists of simple equivalence transformations and is therefore omitted.

5.4.1 Representation of elements and dimension of Φ_n

We look at an element of the morphing space as a kind of compound of the base elements. The mathematical way to produce such a mixture is a weighted sum. In our case the weights are the representation of an element. Such, we can write an element $A \in \Phi_n$ as

$$A = \sum_{i=0}^{n-1} x_i B_i \tag{5.17}$$

Consequently, A is completely represented by the x_i. As Φ_n is an affine space we know that

$$\sum_{i=0}^{n-1} x_i = 1 \tag{5.18}$$

The x_i are called barycentric coordinates of A. Because we can choose only $n-1$ of the x_i independently, the dimension of Φ_n is bounded by $n-1$. If we use Φ_n as a vector space with B_0 as zero element, we will actually use the representation $\mathbf{x} = (x_1, \ldots, x_n - 1)$, 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 Φ_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 $\mathbb{R}^{n-1}$ to Φ_n. Note that this map is at least surjective, because every element of Φ_n has a representation in $\mathbb{R}^{n-1}$. Naturally the question arises whether the map is also injective, which would make Φ_n and $\mathbb{R}^{n-1}$ isomorphic.

5.4.2 Isomorphism between Φ_n and $\mathbb{R}^{n-1}$

To answer this question, we need to further investigate our base elements. As in all vector spaces we can speak of linear dependence and independence of objects according to the following definition:

Definition 5.10 *Given elements $B_0, B_1, \ldots, B_{n-1} \in \Omega$. 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 $\lambda_1 = \ldots = \lambda_{n-1} = 0$. Otherwise the elements are* linear dependent.

With this definition we can prove a possible isomorphism between morphing spaces and $\mathbb{R}^{n-1}$. 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 necessary and together sufficient for a morphing space to be isomorphic to $\mathbb{R}^{n-1}$ (given that the morphing space is a vector space according to the above definitions).

To show that the morphing function has to be injective we take a look at one of the axes of Φ_n. Its domain is given by the values of $m(0, B_i, x_i)$. These values are represented by vectors $(\ldots, x_i, \ldots)$. If m is not injective, there exists an element $A = m(0, B_i, \tau) = m(0, B_i, \tau')$ with $\tau \neq \tau'$ having the different representations $(\ldots, \tau, \ldots)$ and $(\ldots, \tau', \ldots)$ and the map from $\mathbb{R}^{n-1}$ to Φ_n would not be injective. Thus, m has to be injective.

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, e.g.

$$B_0 = \lambda_1 B_1 + \ldots + \lambda_{n-1} B_{n-1} \tag{5.19}$$

so that not $\lambda_i = 0$ and thus, the left-hand side $(1, 0, 0, \ldots)$ and the right-hand side $(1 - \sum_i \lambda_i, \lambda_1, \ldots)$ are not equal. Again the map from $\mathbb{R}^{n-1}$ to Φ_n would not be injective and, therefore, the base elements have to be linear independent.

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

$$A = x_1 B_1 + \ldots + x_{n-1} B_{n-1} \tag{5.20}$$

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

$$A = x'_1 B_1 + \ldots + x'_{n-1} B_{n-1} \tag{5.21}$$

But then we have

$$x_1 B_1 + \ldots + x_{n-1} B_{n-1} = x'_1 B_1 + \ldots + x'_{n-1} B_{n-1} \tag{5.22}$$

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

$$0 = (x'_1 - x_1) B_1 + \ldots + (x'_{n-1} - x_{n-1}) B_{n-1} \tag{5.23}$$

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

$$x'_1 - x_1 = \ldots = x'_{n-1} - x_{n-1} = 0. \tag{5.24}$$

Thus all representations of A are identical, or in other words the map from $\mathbb{R}^{n-1}$ to Φ_n is injective.

5.5 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 out of 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. The algorithms assume only that a classical morphing technique (i.e. one operating on two objects) is available. In the procedure it is used as a black box: A source and target object and a transition parameter are given as input and an in-between object is returned as output.

5.5.1 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 $\mathbf{x}$. 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 by the base objects without B_0.

2. To calculate the projection, identify a null object in the subspace (i.e. the base object with the least index) and repeat the procedure of steps one and two.

3. If the subspace is one-dimensional (this must happen eventually) 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.

The main problem is to find the vector representation of the projection in the subspace. But as it will turn out, all linear equations have ad hoc solutions and their is no need to solve any matrix equations.

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

$$\lambda A = B_1 + \mu_1(B_2 - B_1) + \ldots + \mu_{n-2}(B_{n-1} - B_1) \qquad (5.25)$$

or in vector formulation

$$\lambda \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \mu_1 \begin{pmatrix} -1 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \ldots + \mu_{n-1} \begin{pmatrix} -1 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \tag{5.26}$$

The resulting system of linear equations

$$\begin{pmatrix} x_1 & 1 & \ldots & 1 \\ x_2 & -1 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ x_{n-1} & 0 & \ldots & -1 \end{pmatrix} \begin{pmatrix} \lambda \\ \mu_1 \\ \vdots \\ \mu_{n-2} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.27}$$

can be solved easily. By adding all rows we obtain

$$\lambda \sum_i x_i = 1 \quad \Rightarrow \quad \lambda = \frac{1}{\sum_i x_i} \tag{5.28}$$

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

$$\frac{1}{\sum_i x_i} x_j - \mu_{j-1} = 0 \tag{5.29}$$

and, thus,

$$\mu_j = \frac{x_{j+1}}{\sum_i x_i} \tag{5.30}$$

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:

$$C_k = m\left(B_k, C_{k+1}, \frac{\sum_{i=k+1}^{n-1} x_i}{\sum_{i=k}^{n-1} x_i}\right), k < n-1 \tag{5.31}$$

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

Now it is already clear, why this algorithm is optimal: We have to combine n objects with a given operation. Clearly, we have to apply this operation at least $n-1$ times. The algorithm has another advantage. If A is really an “in-between” element of the base elements, i.e. we have $x_i \in [0, 1]$ for all i, then all intermediate elements C_k are also “in-between” objects. If we look for example at the definition of addition in morphing spaces, this property is not evident.

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 (since this is the only degree of freedom in the algorithm). 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.

5.5.2 Analysis of objects

The other important task in a morphing space is to find a description of the object in the space, which represents the object's properties best. Until now, we did not use any additional information to describe the morphing space properties and the synthesis algorithm. The analysis of objects, however, cannot be performed without the definition of a distance function on the objects. Note that the notion of morphing functions actually defines a metric. Yet, we cannot compute it.

For this reason we have to resort to practical solutions for a metric. For example, on images one could use the squared distances of pixel colors in a color space.

For the analysis of objects we cannot use one scheme for linear and non-linear morphing spaces. If the morphing space has the characteristic of a vector space we can use a much more sophisticated algorithm to find the representation. In non-linear morphing spaces we will further distinguish some situations.

In all cases we need to minimize along (one-dimensional) lines. In all cases we have to do this, we know that the minimum is a unique local minimum. We therefore employ a numerical method known as Brent's method [Brent 1973]. This method switches between a golden section search and the approximation of the minimum by a parabola, according to the appropriateness.

Note that this one-dimensional minimization could be seen as a practical solution for finding the inverse element of m. Therefore we will use the symbol w as in the above mathematical discussion.

5.5.3 Analysis in linear morphing spaces

We have found a considerably fast method to analyze an object in a morphing space, if the morphing space is linear and the distance function has a certain property. We will explain this property later.

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 claim that this mapping is linear. In this case, the actual representation is a linear mapping of the found vector. Linear mappings can be represented by matrix multiplications. Thus, all we have to do is to find the representation matrix.

First we want to explain why 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 is itself 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.

Note, that we have assumed the distance function to be calculated on a representation of the elements, such that morphing between elements is represented by straight lines. This is the property we spoke about in the beginning of this section. The function d has to be a norm on the vector representation of the elements. It is not enough, that the morphing space is linear and the function d is a norm on another representation.

Assume the projection of an element A results in the vector $(a_1, \dots, a_{n-1})$ while the actual representation of A is $(x_1, \dots, x_{n-1})$. We are searching for the representation matrix M that connects those two representations:

$$M\mathbf{x} = \mathbf{a} \tag{5.32}$$

As we know from linear algebra, the rows of M consist of the images of the base vectors. Therefore, we obtain M by projecting the base elements $B_1, \dots, B_{n-1}$ 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 $B_1, \dots, B_{n-1}$ 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 [Golub & Van Loan 1989]. 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 [Press et al. 1992].

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.

5.5.4 Analysis in non-linear morphing spaces

In a non-linear morphing space we cannot use 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.

Powell's algorithm (see [Press et al. 1992] for instance) would be traditionally the method of choice for such cases. But note that morphing spaces differ somewhat from other vector spaces: In vector spaces one can specify a line by one base point and a direction vector. Conversely, in morphing spaces we need two base points (objects). If a sophisticated algorithm (such as Powell's) specifies some lines, along which one has to minimize, this causes the calculation of additional objects in the morphing space. But constructing objects is expensive and should be avoided as far as possible. Therefore we propose the following scheme to minimize in morphing spaces with a unique local minimum:

Given a morphing space Φ with the basis $B_0, \ldots, B_{n-1} \in \Omega$ constructed with a morphing function m and an element C, which has to be analyzed. Calculate $C_0 = m(B_0, B_1, w(B_0, B_1, C))$ and then interactively

$$C_i = m(C_{i-1}, B_{i+1 \text{ mod } n}, w(C_i - 1, B_{i+1 \text{ mod } n})) \tag{5.33}$$

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 each step no object has to be predetermined to define the line along which one has to minimize.

Note that this trick works only when the distance-minimum is indeed unique in the morphing space. The above algorithm finds the first local minimum it "comes across". It will not find the global minimum if other local minima exist. For such cases more robust methods are needed.

We successfully adopted a simulated annealing scheme [Kirkpatrick et al. 1983] adapted to our needs: First, calculate $C_0 = m(B_0, B_1, w(B_0, B_1, C))$. In every step, choose $x \in [0, 1]$ randomly and calculate

$$D = m(C_{i-1}, B_{i+1 \text{ mod } n}, x) \tag{5.34}$$

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

$$p = e^{\frac{d(C,D) - d(C,C_{i-1}}{kT}} \tag{5.35}$$

Thus, if D is closer to the minimum than C_i 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].

Obviously, this scheme represents a simulated annealing technique in morphing spaces.

5.6 Spaces of meshes from morphing

The conceptual extension of the framework to more meshes is rather straightforward as compared to possibly non-linear morphing functions. Given meshes $\mathcal{M}_i = (V_i, K_i)$ a common connectivity K together with vertex sets $V(\mathbf{e_i})$ is established. The vertex sets form a base of a space, which is reflected by using canonical base vectors $\mathbf{e_i}$ as indices. A morphed shape $(V(\mathbf{s}), K)$ is represented by a vector $\mathbf{s} = (s_0, s_1, \ldots)$ reflecting the shares of the meshes $\mathcal{M}_0, M_1, \ldots$.

Not all techniques presented in this work are equally suited to be extended to more meshes. The correspondence problem discussed in Chapter 2 seems to be relatively easy to extend. All meshes are embedded in the given parameter domain, which leads to barycentric representation of the original vertices. If each set of original vertices V_i needs to be mapped to all other meshes $\mathcal{M}_j, i \neq j$ the complexity would grow quadratically with the number of meshes. However, this is not necessary if a remeshing strategy is used to generate a consistent mesh connectivity (see Section 3.5). This procedure generates the same set of vertices over all shapes, thus, the complexity is linear in the number of meshes times the number of vertices used in the remesh, which is the best we can expect. Concluding, the best way to generate the set $\{(V(\mathbf{e_i}), K\}$ is to embed all meshes in a common parameter domain (spherical or piecewise linear) and then remesh to the desired accuracy. This has been demonstrated by Michikawa et al. [2001] (see Figure 5.1).

The vertex path problem now extends to compute combinations of several vertex vectors. Linear vertex combination is easily extended:

$$V(\mathbf{s}) = \sum_i s_i V(\mathbf{e_i}) \tag{5.36}$$

Surprisingly, any technique involving rotations such as the ones explained in Sections 4.2 and 4.4 seem to be difficult to extend. Instead of interpolating the orientation one could compute the principal components (moments) of the shapes and align them with the canonical axes of the coordinate system. To extend the local morph approach explained in Section 4.5 the linear combination has to be applied to the Laplacian coordinates.

Applications of such spaces of meshes range from modeling and analysis of shapes to animation. Praun et al. have termed the synthesis-analysis part digital geometry processing (DGP) [Praun et al. 2001]. Modeling could be achieved be combining several shape (features) to yield the desired result. This has applications

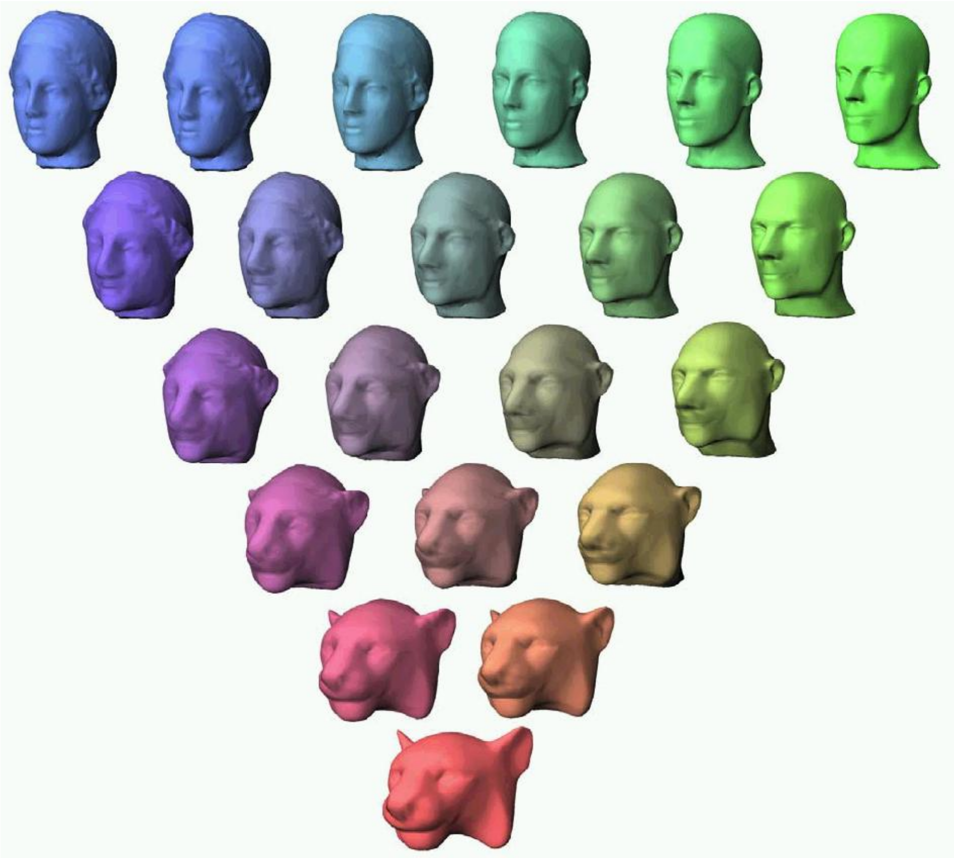

Figure 5.1: A space of shapes generate from three input shapes and linear interpolating their geometry vectors. Correspondence has been established using a coarse base domain and a multiresolution mesh. Reprinted from Michikawa et al. [2001].

in information visualization (see Chapter 6). Using techniques such as the principal component analysis, spectral properties of the mesh family can be explored.

The space of meshes $(V(\mathbf{e_i}), K)$ allows to represent animations as a curve $\mathbf{s}(t)$. This idea will be detailed in Chapter 7.