Chapter 1

Introduction

Art always serves beauty,
and beauty is the joy of possessing form,
and form is the key to organic life
since no living thing can exist without it.
Boris Pasternak, Doctor Zhivago

Every thing has a shape. We can see, touch, even hear shape. It is the fundamental concept for interaction with the world we live in. We compare one shape with another to assess it, to put it into context: Every shape can be seen as as a variation of another shape or the combination of a few shapes.

A formalism is needed to communicate shape. Many such formalisms exist, however, most of them describe shape in an absolute way, without relating to other objects. This cannot be avoided if only one single object has to be described. If many (relating) objects are concerned, an absolute description not only misses information about the relation to other objects it might also be inefficient since the differences of shapes could be described more compactly than the whole shape.

This work mainly deals with the description of shapes as *compounds* of a set of *base shapes*. Each base shape is described absolutely as set of planar patches, being an approximation of the "real" shape. These base shapes can be combined and, altogether, form a *space of shapes*. Such representation of shape is particularly suited to deal with sets of shapes with some common ground. For example, think of a space of shapes representing faces: This space is more appropriate for modeling a face, generating or storing animated sequences of faces. Limiting the possible choices means, in this context, the need for less information or storage space to generate or represent the intended object. An illustration is given in Figure 1.1.

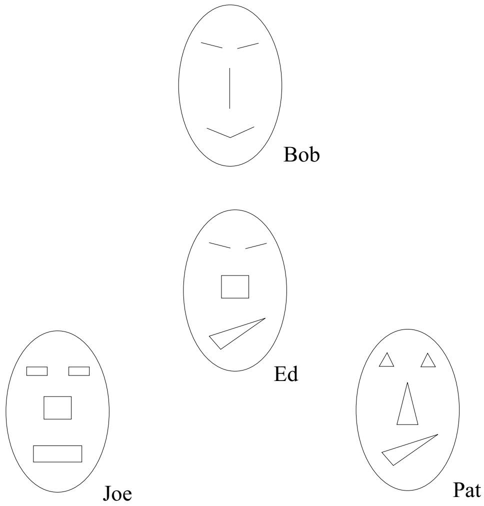

Figure 1.1: Describing shapes as a compound of base shapes. It seems natural to describe Ed as having the eyes of Bob, the nose of Joe, and the mouth of Pat rather than giving details of the shape of each feature. If we wanted to communicate information about Ed's face, the description in terms of Bob, Joe, and Pat is also much smaller and, yet, absolutely precise. Note that the faces of Bob, Joe, and Pat allow to generate a variety of other faces. This could be useful to model a face that can be described in terms of their features. This space of faces also contains sequences smoothly transforming from one face to another. Thus, it allows to generate and store animated sequences.

1.1 Technical motivation

In computer graphics, models of three-dimensional shapes are nowadays mainly represented as *meshes*. A mesh contains a set of vertices describing geometric positions (and other attributes such as color, etc.) and topological information describing edges containing vertices and forming faces. Meshes are universal in the sense that they can represent every shape with arbitrary precision (assuming infinite space to store the description).

In many applications one deals not only with one single mesh but with many meshes. The most prominent example are geometric animations, which is typically stored a set of meshes describing the shape over time. We like to exploit the idea of a shape space, where shapes are described as the combination of a few base shapes. Here, base shapes are meshes, and all combinations are meshes.

We start exploring this idea by looking at the simple case of only two base meshes. The main idea of this work is to use *morphing* techniques to generate the family of shapes described as the combination of two base shapes. Morphing techniques are used to generate smooth transitions from one object to another. They have become popular and widespread in the special effects industry but have applications in many areas such as medical imaging and scientific visualization. We can say that a morph represents the family of shapes generated by two base shapes, i.e. the space is one dimensional. By adding a third base shape and morphing between an element of the family resulting from the first two base shapes we add another dimension. This process can be repeated to add any number of dimensions.

Such spaces of shapes allow to represent each shape in the space with a vector of scalars not longer than the number of base objects spanning the space. Assuming the number of base shapes is relatively small with respect to the amount of information needed to describe a single shape, this is an extremely compact and meaningful way of describing a shape.

Why is the representation meaningful? Imagine a set of faces (smiling, frowning, blinking, staring, etc.) comprising the base of a space. If we want to generate a particular expression we simply describe the face in terms of the features we want. The modeling process is intuitive and simple. In addition, if such a face has to be stored or communicated only the small vector is needed.

The major aim of this work is to build spaces of polyhedral objects and demonstrate their usefulness in practical applications. However, at the current state of science even morphing between two polyhedral objects is a difficult process. For that reason, a large part of this work is dedicated to generating morph sequences between two meshes. This is a remunerative subject in itself, as morphing object representations is superior to morphing representations of space (such as images).

1.2 Overview & Framework

The first part of this work is dedicated to mesh morphing techniques. Mesh morphing techniques involve computations on the geometry as well as the connectivity of meshes. For simplicity this report concentrates on triangle meshes. In the context of morphing it seems to be acceptable to triangulate polygonal meshes prior to the application of a morphing technique. To classify and understand mesh morphing techniques it is helpful to use the now widespread terminology from Spanier [1966]. A mesh $\mathcal{M}$ is described by a pair (K, V), where K is a simplicial complex representing the connectivity of vertices, edges, and faces and $V = (\mathbf{v_1}, \ldots, \mathbf{v_n})$ describes the geometric positions of the vertices in $\mathbb{R}^d$, where typically $d = 3$.

The abstract complex K describes vertices, edges, and faces as $\{0, 1, 2\}$-simplices, that is, edges are pairs $\{i, j\}$, and faces are triples $\{i, j, k\}$ of vertices. The *topological realization* maps K to a simplicial complex $|K|$ in $\mathbb{R}^n$: The vertices are identified with the canonical basis of $\mathbb{R}^n$ and each simplex $\mathbf{s} \in K$ is represented as the convex hull of the points $\{\mathbf{e_i}\} \in \mathbb{R}^n, i \in \mathbf{s}$ Thus, each 0-simplex is a point, each 1-simplex is a line segment, and each 2-simplex is a triangle in $\mathbb{R}^n$.

The *geometric realization* $\phi_V(|K|)$ is a linear map of the simplicial complex $|K|$ to $\mathbb{R}^d$, which is defined by associating the basis vectors $\mathbf{e_i} \in \mathbb{R}^n$ with the vertex positions $\mathbf{v_i} \in V$. The map ϕ_V is an *embedding* if ϕ_V is bijective. The importance of an embedding is that every point $\mathbf{p}$ on the mesh can be uniquely represented with a barycentric coordinate $\mathbf{b}$, i.e. $\mathbf{p} = \phi_V(\mathbf{b})$. Such barycentric coordinates have at most three non-zero components and specify the position of a point relative to a simplex. If the point is coincident with a vertex it is a canonical basis vector, if the point lies on an edge it has two non-zero components, otherwise it has three and lies on a face.

The neighborhood ring of a vertex $\{i\}$ is the set of adjacent vertices $\mathcal{N}(i) = \{j|i, j \in K\}$ and its star is the set of incident simplices $\mathcal{S}(i) = \bigcup_{i\in s, s\in K} s$.

In the classical setting of mesh morphing two meshes $\mathcal{M}_0 = (K_0, V_0)$ and $\mathcal{M}_1 = (K_1, V_1)$ are given. The goal is to generate a family of meshes $\mathcal{M}(t) = (K, V(t)), t \in [0, 1]$ so that the shape represented by the new connectivity together with the geometries $V(0)$ and $V(1)$ is identical with the original shapes, i.e. $\phi_{V(0)}(|K|) = \phi_{V_0}(|K_0|)$ and $\phi_{V(1)}(|K|) = \phi_{V_1}(|K_1|)$. Most of the time the paths $V(t)$ are required to be smooth. The generation of this family of shapes is typically done in three subsequent steps:

1. Finding a correspondence between the meshes. More specifically, computing coordinates W_0, W_1 that lie on the other mesh, i.e. $W_0 \in \phi_{V_1}(|K_1|)$ and $W_1 \in \phi_{V_0}(|K_0|)$. Each coordinate in W_0, W_1 is represented as a barycentric coordinate with respect to a simplex in the other mesh. Note that ϕ_{W_0} will not map $|K_0|$ to $\phi_{V_1}(|K_1|)$ (and vice versa), as only the vertices are mapped to the other mesh but not the edges and faces. Particularly important is the alignment of automatically detected or user specified features of the meshes. The process of finding correspondence between meshes is discussed in Chapter 2.

2. Generating a new, consistent mesh connectivity K together with two geometric positions $V(0), V(1)$ for each vertex so that the shapes of the original meshes are reproduced. The traditional morphing approach to this problem is to create a superset of the simplicial complexes K_0 and K_1. However, remeshing techniques as used in multi-resolution techniques are also attractive. Methods to generate such representation are discussed in Chapter 3.

3. Creating paths $V(t), t \in]0, 1[$ for the vertices. While in general this is an aesthetic problem, several constraints seem reasonable to help in the design process. For example, in most applications the shape is not expected to collapse or self intersect and, generally, the paths are expected to be smooth. Techniques tackling the path problem are discussed in Chapter 4.

An illustration of this process is shown in Figure 1.2.

Using such mesh morphing techniques, a shape space can be build. However, it is interesting to know the mathematical properties of the resulting space from the mathematical properties of the morphing technique. This calls for a theoretical model of shape spaces. In addition, some mesh morphing techniques are better suited to be extended to more than one dimension. These issues are analyzed in Chapter 5.

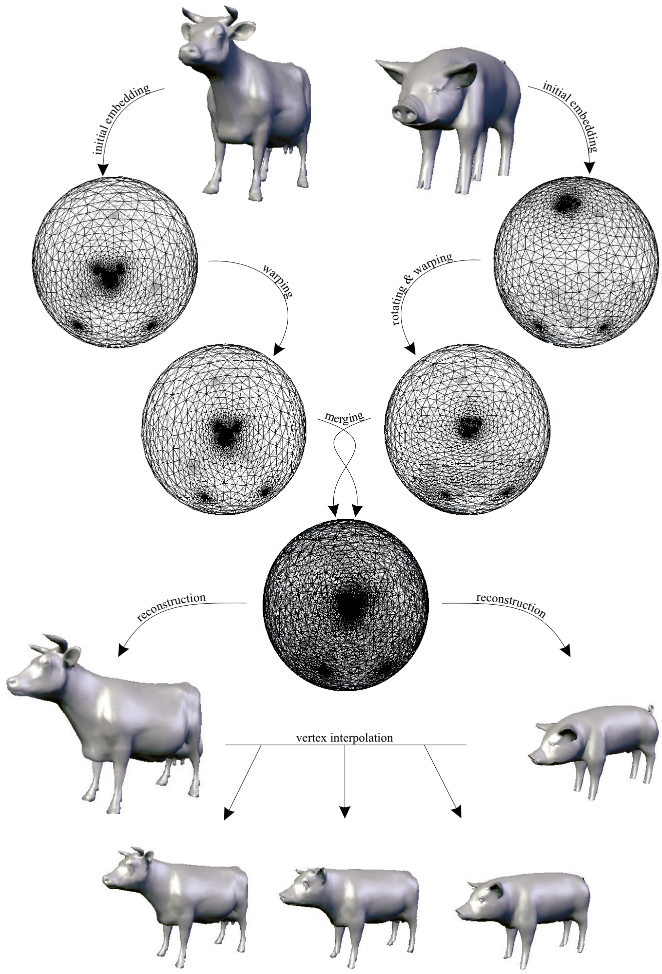

Figure 1.2: The process of mesh morphing illustrated at the example of meshes that allow a mapping to the sphere. First, the meshes are embedded on the sphere, thus, establishing correspondence between the set of vertex positions. The correspondence is refined using information about feature correspondence. The embeddings are used to generate one mesh containing, which can represent both input meshes. A vertex interpolation scheme yields the morph.

The last two chapters of this work show applications of this approach. On one hand, the idea of mapping from an abstract vector to a shape is used for information visualization. On the other hand, spaces of meshes are used to generate and compress geometric animations. This seems to be the most fruitful application of this work so far.

While in general shapes will be represented as meshes throughout this work, other types of representations (polygons,images) are used occasionally. In some cases an algorithm was developed for such other representations and we believe it is best explained for that representation and later extended to meshes. However, sometimes the extension has not been found, yet, we present the algorithm for the sake of completeness of this work.

1.3 Context of prior work

The idea of a *shape space* has been used occasionally for a long time. The term seems to be first used by Herbert Edelsbrunner (in a private communication) and has been published later [Cheng et al. 1998; Edelsbrunner 1999]. Spaces of shapes have been used for a long time in visual computing, particularly face recognition applications (Eigenfaces [Kirby & Sirovich 1990; Turk & Pentland 1991]). Recently, a linear space has been used to model faces [Blanz & Vetter 1999], however, the space is particularly designed for faces. In animation, a configuration space has been limited by forming cross-products of small linear spaces [Ngo et al. 2000]. It seems that most approaches to date use a space to solve a particular problem rather than viewing the space as a fundamental concept for shape representation (as in [Cheng et al. 1998; Edelsbrunner 1999] and this work).

Mesh morphing techniques have drawn much attention recently, following the success of image morphing techniques. Image morphing has been developed for special effects in feature films and commercials (see [Wolberg 1998] for a recent survey on image morphing). The extensions of some of the concepts in image morphing to explicit representations of shapes (such as meshes) are difficult because of topological restrictions. Images do not carry the topology of the objects they depict and, therefore, no problems from different topology can arise. The topology problem as well as feature alignment and local control over the morph are recent topics of research in mesh morphing. Lazarus & Verroust [1998] surveys 3D morphing algorithms. This work has also resulted in survey on mesh morphing [Alexa 2001b; Alexa 2002b].

1.4 Contributions

The primary contributions of this work to the field of computer graphics are:

Mesh morphing Several contributions to the field of morphing meshes have been made.

Feature alignment A mesh morphing technique for topological spheres is introduced, which allows for particular easy feature specification [Alexa 1999; Alexa 2000].

Mesh merging A new algorithm for generating one mesh from two input meshes (mesh overlay) is presented, generalizing previous results. In contrast to other approaches, the algorithm is asymptotically optimal [Alexa 2002b].

Local control In image morphing one can easily specify which region transforms when in the morph, e.g., first the nose, then the eyes and then the rest of a face. We present similar techniques for mesh morphing [Alexa 2001a; Alexa 2002a].

Vertex path A method to generate intuitive and well behaving vertex paths in morphing is introduced. The paradigm employed is that the object is transformed rigidly as much as possible to avoid unnecessary deformations [Alexa et al. 2000].

Visualization of multiparameter data Morph spaces are shown to be a useful way to generate glyphs and icons in the visualization of multiparamter data. Additionally, the paradigm allows for a flexible and ituitive generation of the visualization [Alexa & Müller 1998b; Alexa & Müller 1999b].

Geometric animations Spaces of meshes are shown to be a particularly elegant and effective way to generate, store, and communicate geometric animations. The resulting compression is progressive and the achieved compression ratios progress over previous work [Alexa et al. 2000; Alexa & Müller 2000].