The Visual Computer

Transformation of functionally defined shapes by extended space mappings

V. Savchenko, A. Pasko

Laboratory of Shape Modeling,
Department of Computer Software,
The University of Aizu, Aizu-Wakamatsu City,
Fukushima Prefecture 965-80, Japan
E-mail: {savchen, pasko}@u-aizu.ac.jp

We present a general mathematical framework for transforming functionally defined shapes. The proposed model of extended space mappings considers transformations of a hypersurface in coordinate-function space with its projection onto geometric space. This model covers coordinate space mappings, metamorphosis, and algebraic operations on defining functions, and introduces several new types of transformations, such as function-dependent space mappings and combined mappings. The approach is illustrated by new local deformations created by means of function mappings, feature-based space mapping, offsetting along the normal, thin shell generation, 2D shape blending, and collision-free metamorphosis.

Key words: Transformation – Deformation – Metamorphosis – Space mapping – Implicit surface – Isosurface – Extended space – Real function – F-rep – Shape modeling – Geometric modeling – Solid modeling

Correspondence to: V. Savchenko

1 Introduction

In recent years, much effort has been put into exploring shape transformation as a basic operation in computer graphics. The problem concerns tranforming a given geometric shape into another in a continuous manner. We consider shape transformation as a general type of operation that includes space mappings, metamorphoses, and others. Examples of applications can be found in animation and computer-aided design.

Geometric objects defined by real functions of several variables (so-called implicit models) have proven useful in computer graphics, geometric modeling, and animation (Hoffmann 1993; Pasko et al. 1995; Wyvill et al. 1993). Objects of this kind can be created with skeleton-based functions (Blinn 1982; Wyvill et al. 1986), so-called R-functions for set-theoretic operations (Shapiro 1994), and by processing range or volume data. Although existing deformation techniques are claimed to be model independent, they are practically applied to polyhedral and parametric objects. We note that the functional description provides an additional degree of freedom in manipulating objects, namely, by directly changing defining function values. To combine this advantage with existing deformation techniques, we introduce an extended space with an additional coordinate of a point as the defining function. This allows us to transform geometric coordinates and the function coordinate simultaneously as coordinates of a point of a higher dimension. Thus we can provide a high diversity of transformations.

The motivation of this paper is to present a mathematical formulation of shape transformations as extended space mappings. We show that, in general, this is a difficult numerical problem with several special cases, for which analytical solutions exist. The model of extended space mappings incorporates well-known geometric space mappings and metamorphoses. Moreover, this model introduces several new transformations such as *function-dependent space mappings* and *combined mappings*. We also consider several practical applications of our approach.

The rest of the paper is organized as follows. Section 2 gives an overview of existing shape transformation techniques. We discuss the notion of an extended space and describe our approach to extended space mappings in Sect. 3. We classify space mappings and give some examples in Sect. 4. All illustrations for the examples are generated by ray casting. Section 5 concludes the paper. The appendix describes a volume spline based on Green's function.

2 Previous work

Most existing shape transformation techniques fall into one of the following three categories:

1. Mapping the space onto itself
2. Metamorphosis
3. Modification of defining functions.

2.1 Space mappings

A space mapping establishes a one-to-one correspondence between points of the given space. If applied to some point set in the space, it changes this set to a different one. A mapping can be defined by the functional dependence between the new and old coordinates of a point. Mappings can be controlled by numerical parameters of predefined functions, by control points, and by differential equations. Fournier and Wesley (1983) and Barr (1984) propose specific functions for bending, tapering, and twisting operations. Interactive control of these deformations by a single curve in 3D space is implemented by Lazarus et al. (1993). Sclaroff and Pentland (1991) use natural vibration modes to express a wide range of deformations. If the transformation is not degenerate, an inverse matrix can be calculated and used for the inverse mapping, which is necessary to transform implicit surfaces.

Free-form deformations (FFDs), proposed by Sederberg and Parry (1986) and extended by Coquillart (1990), Coquillart and Jancene (1991), and Hsu et al. (1992) are controlled by user-defined point lattices. Forward space mapping is described by trivariate Bézier volumes and can be effectively applied to polygonal and parametric surfaces. Alternatively, inverse mapping for implicit surfaces requires time-consuming iterative searches or subdivisions (Nishita et al. 1993). Chang and Rockwood (1994) present an intuitive approach to control FFDs with a single Bézier curve. Their algorithm works faster than the standard FFD algorithm because the problem dimension has been reduced from three to one. General deformation techniques providing forward and inverse mapping (Borrel and Bechmann 1991; Borrel and Rappaport 1994) are better suited to implicit surfaces. Most of the deformation methods mentioned are too global to provide a series of small bumps defined by arbitrary points. Although Borrel and Rappoport's (1994) method was designed for localized space mappings, it can lead to nonintuitive results when bounding spheres of several control points intersect. The Bechmann (1994) survey discusses common mathematical foundations of the space deformation techniques mentioned.

Different approaches apply scattered data interpolation techniques: thin-plate spline (Bookstein 1991), distance-weighted interpolants (van Overveld 1992), volume spline based on Green's function (Savchenko and Pasko 1994), multiquadrics (Ruprecht et al. 1995), and multilevel B-spline interpolation (Lee et al. 1996). Ruprecht et al. (1995) attempt to combine space mappings with metamorphosis. Their method does not provide a solution in 3D space, but relies on 2D image morphing.

2.2 Metamorphosis

Metamorphosis (or shape averaging, shape blending, inbetweening, morphing) is an operation on two geometric objects resulting in a new object with an intermediate shape. It can also be thought of as a transformation of the initial object controlled by another object. Morphing was first introduced in image processing and animation to generate intermediate images by transformation of one image into another (Wolberg 1990).

Shape metamorphosis for 3D polyhedral objects is quite well elaborated. An interesting combination of representations is shown by Payne and Toga (1992). Initial polyhedra are converted into distance field volumetric data, then these data are interpolated, and the resulting isosurface is polygonized. A more sophisticated metamorphosis of volumetric objects is proposed by Hughes (1992), who uses the Fourier transforms of two volumetric objects to interpolate between low frequencies with the high frequencies of the first object gradually removed, and the high frequencies of the second object added in. Lerios et al. (1995) generalize image metamorphosis to the case of 3D volume data and combine warping and cross-dissolving. Shape inbetweening for soft objects (Wyvill 1990) ap-

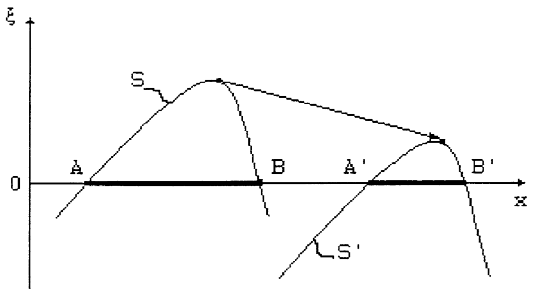


Fig. 1. Transforming a 1D segment *AB* into *A'B'* by applying extended space mapping to the curve *S* defined as $\xi=f(x)$

plies gradually changed weighting of the force property of each source key. The skeleton of the intermediate shape can be constructed by applying the Minkowski sum to initial skeletons (Galin and Akkouche 1996).

2.3 Modification of defining functions

Our examples of metamorphosis illustrate the general idea of transformation of geometric objects by applying algebraic operations to their defining functions. Descriptions of implicit surfaces by blobby model (Blinn 1982), metaballs (Nishimura et al. 1985), soft objects (Wyvill et al. 1986) and distance functions (Bloomenthal and Wyvill 1990; Larcombe 1994) are essentially based on algebraic sums of defining functions, which allow deformation of an object by adding new primitives to its skeleton. Gascuel (1993) proposes deformations of distance surfaces in collisions by algebraic difference of defining field functions. Singh and Parent (1995) apply this technique to deform a polygonal surface, using its elegant immersion in implicit primitives.

A blend surface can be described in terms of algebraic operations on defining functions (Woodwark 1987). These operations are applied in practice to primitives, but not to constructive solids [see, for example, Bowyer (1994)]. The theory of R-functions (Rvachev 1963, 1987) provides a means of function representation for solids constructed by the standard (nonregularized) set operations [see Shapiro (1994) for a survey]. Shapiro (1994) applies algebraic difference to construct a real function defining a regular solid required in constructive solid geometry (CSG). Pasko et al. (1995) define blending, offsetting, and other operations by algebraic sums applied to R-function-based exact descriptions of constructive solids.

Shape reconstruction from given points can be thought of as a special case of transformation. Muraki (1991) proposes applying the blobby model to fit scattered points. To fit an algebraic surface to given points, Bajaj et al. (1992) solve a constrained minimization problem for distance criteria. Savchenko et al. (1995b) use the algebraic sums of an initial defining function and a volume spline to reconstruct a solid from scattered surface points. The reconstruction from given contours is considered a metamorphosis of 2D cross-sections along a spine.

As this overview shows, space mappings, metamorphosis and algebraic operations on defining functions represent various tools for the complex transformation of geometric objects. The general idea of our approach is to give a framework that can combine these techniques and provide the basis for developing new ones.

3 Extended space mappings

3.1 Extended space

Consider an n+1-dimensional euclidean space E^{n+1} and a hypersurface S defined in it by a real continuous explicit function of n variables:

$$\xi=f(x_1, x_2,\ldots, x_n), \tag{1}$$

where $P_n=(x_1, x_2,\ldots, x_n)$ is a point in the subspace E^n and $P_{n+1}=(x_1, x_2,\ldots, x_n, \xi)$ is a point in E^{n+1}. The major requirement to the function f is to have at least C^0 continuity. Intersection of the given hypersurface S with the halfspace $\xi\geq0$ and projection of the result onto the E^n space along the ξ axis gives the point set

$$G=\{P_n{:}P_{n+1}\in S, \xi\geq0\}. \tag{2}$$

Equations 1 and 2 together are equivalent to the more common definition

$$G=\{P_n{:}f(x_1, x_2,\ldots, x_n)\geq0\}, \tag{3}$$

but Eqs. 1 and 2 fit better the formulation of extended space mapping, as we will show in a simple example (Sect. 3.2) and in a general formulation (Sect. 3.3).

We consider G to be a geometric object defined in n-dimensional euclidean space E^n. The space E^{n+1} can be called an *extended space* and can be constructed from E^n by adding one more dimension. We call f a *defining function*. Equations 1 and 2 are called a *function representation* (F-rep) of a geometric object. This representation is closed under set-theoretic and other operations (Pasko et al. 1995). A subset of G with ξ=0 is usually called an *implicit surface* or an *isosurface*.

The following subsections discuss our approach to transformations of constructed geometric objects in E^n with the use of the extended space mappings Φ: $E^{n+1} \rightarrow E^{n+1}$ applied to the hypersurfaces of Eq. 1 in E^{n+1}.

3.2 Simple example

Let us discuss an example illustrated by Fig. 1. The intention is to model a 1D geometric object (a line segment) in E^1 and then to transform it by mappings of the extended space E^2. Following Eqs. 1 and 2, we introduce a 2D curve S (hypersurface in E^2):

$$\xi = f(x),$$

where $P_1=(x)$ is a point in E^1 and $P_2=(x, \xi)$ is a point in E^2. The geometric object defined as

$$G = \{P_1 : P_2 \in S, \xi \geq 0\}$$

is a segment of a line ξ=0 (segment AB in Fig. 1). Consider the following extended space mapping:

$$P'_2 = \Phi(P_2) = P_2 + C,$$

where $C=(c_1, c_2)$ defines a translation in E^2. This mapping applied to the curve S results in S'. The object G is transformed by this mapping from the segment AB to the segment A′ B″ with the description

$$\xi' = f(x' - c_1) + c_2. \qquad (4)$$

Note that, to obtain this transformation with mappings of E^1, one has to think of some composition of 1D translation and scaling.

3.3 General formulation

Let an extended space mapping be formulated as follows:

$$P'_{n+1} = \Phi(P_{n+1}), \qquad (5)$$

where $P_{n+1}=(x, \xi)$ is an original point in E^{n+1} with $x=(x_1, x_2, \ldots, x_n)$ and $P'_{n+1} = (x', \xi')$ is a point resulting from mapping $\Phi(x, \xi)=(\phi_1, \phi_2)$. The transformed geometric object from Eq. 2 is described by the system of equations:

$$\begin{aligned} x' &= \phi_1(x, \xi) \\ \xi' &= \phi_2(x, \xi) \\ \xi &= f(x). \end{aligned} \qquad (6)$$

The problem is to find the explicit description of the transformed geometric object in the form

$$\xi' = f^{\sim}(x'), \qquad (7)$$

which does not follow directly from the system of Eqs. 6. For a given x', Eqs. 6 turn into a system of n+2 equations with n+2 variables x, ξ, ξ'. Although one can find the value of ξ' by solving this system numerically for any given x', it is not applicable in practice. In the following section we discuss the types of extended space mappings for which the function $\xi'=f^{\sim}(x')$ can be restored analytically.

4 Types of the extended space mappings

The general formulation of the extended space mapping covers several specific types of mappings. The rest of the paper describes these mappings.

4.1 Function mappings

Let us assume the identity mapping of the subspace E^n. From Eqs. 6 we have:

$$\begin{aligned} x' &= x \\ \xi' &= \phi_2(x, \xi) \\ \xi &= f(x). \end{aligned} \tag{8}$$

The defining function of the transformed geometric object (Eq. 7) is now given by

$$\xi' = \phi_2(x', f(x')). \tag{9}$$

We examine here a simple form of this dependence:

$$\xi' = f(x') + d(x'), \tag{10}$$

where $d(x)$ is a continuous real displacement function. If $d(x)$ is a constant, the resulting object is a contracted or expanded version of the initial object. In this case, Eq. 10 describes the iso-valued offsetting operation (Pasko et al. 1995). More complex cases are discussed later.

4.1.1 Control by points and curves

The mapping of Eq. 10 gives an attractive possibility for controlling local deformations by placing arbitrary points inside or outside of the initial object surface. Let ξ=$f(x)$ be a defining function of an initial object G, and let $\left\{p_i = \left(x_1^i, x_2^i, \ldots, x_n^i\right): i = 1, \ldots, N\right\}$be a set of N arbitrary points of euclidean space E^n. These control points are placed inside or outside G and are assumed to belong to the surface of a modified object $G^{\sim}$. Thus, the control points define the deformation of G resulting in $G^{\sim}$. The problem is to construct a defining function ξ'=$f^{\sim}(x')$ for $G^{\sim}$ on the base of f and $\{p_i\}$. We use the mapping of Eq. 10 with $d(p_i)$=–$f(p_i)$. We have found the following approximate form of the displacement function to be suitable in some practical applications:

$$d(x) = -\sum_{i=1}^{N} d_i(x), \tag{11}$$

where

$$d_i(x) = \frac{f(p_i) \cdot \left(1 - r_i^2\right)^3}{1 + r_i^2} \quad \text{if } r_i^2 < 1,$$

$$d_i(x) = 0 \quad \text{if } r_i^2 \geq 1,$$

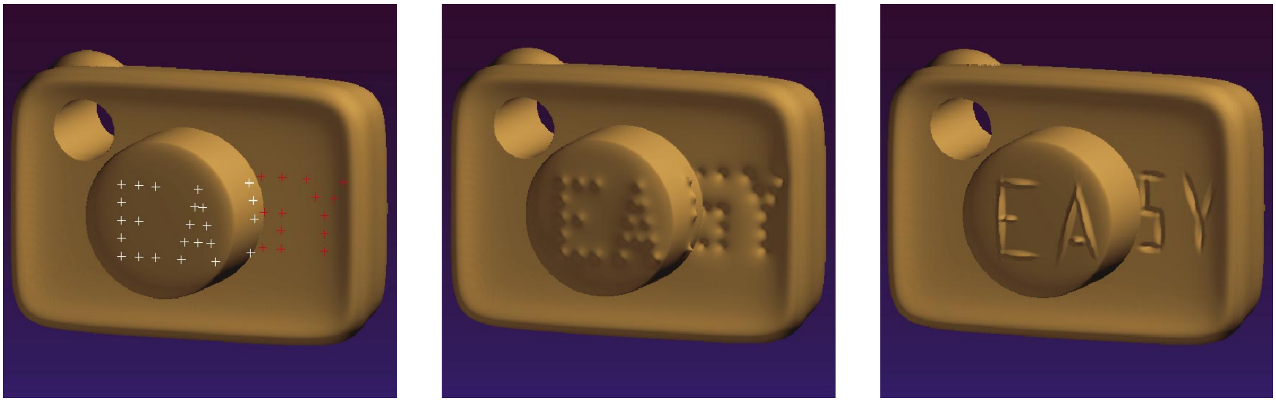
$r_i = \|x - p_i\| / r_{i0}$ and r_{i0} is a radius of influence of ith control point. This type of gaussian-like function is proposed by Wyvill et al. (1994) for parametric surface design. The function of Eq. 11 has a local extremum in each control point. That is why it produces blobbylike deformations of 3D solids. This form of the displacement function allows the introduction of an arbitrarily oriented elliptic area of influence for each control point, but is not the only one possible. Other gaussian-like functions can be designed for specific applications. Figure 2 illustrates modeling of pinching, pricking, and scratching effects with control points placed inside and outside a 3D solid defined by the set-theoretic and blending operations with R-functions. One can notice the local nature of deformations that is practically impossible to achieve by other existing deformation models.

If the displacement function $d(x)$ interpolates defining function values in control points more smoothly (without local extreme points), more global deformation can be obtained. This can be achieved with the volume spline based on Green's function (see the Appendix). We used an algebraic sum of a sphere and a volume spline to reconstruct 3D solids from given scattered surface points (Savchenko et al. 1995b).

It might be interesting to control function mappings by a continuous curve rather than by a finite set of control points. Then the displacement function can be defined as

$$d(x) = -\int_0^1 f(x(t))\, e^{-(r/a)} dt, \tag{12}$$

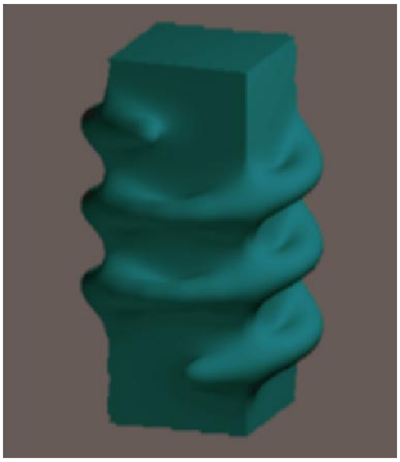
where $r = \|x - x(t)\|$ and the function $x(t)$ defines a control parametric curve. Note that this displacement function resembles that used to define convolution surfaces (Bloomenthal and Shoemake 1991). The deformation of Eq. 10 with the displacement function of Eq. 12 can be referred to as *convolution deformation*. Figure 3 illustrates this type of the function mapping. An initial block is deformed under control of a parametric spiral curve. The deformed shape in Fig. 3 is generated by Simpson's method of integral calculation (Eq. 12). One can notice a sweep-like shape produced by this mapping on the surface of the original object.



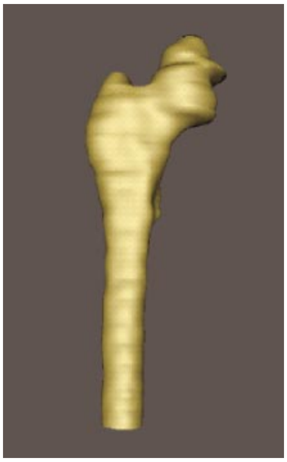
2a

2b

2c

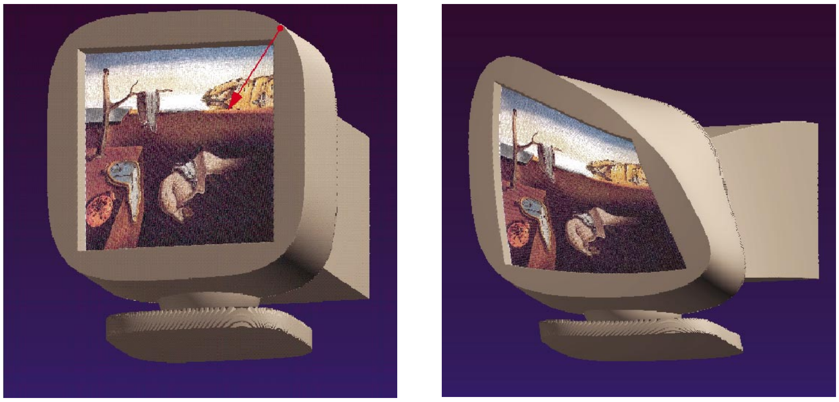


3



4

Fig. 2a–c. Local deformations with function mappings: **a** initial object and control points (*white markers* for point outside the object, *red markers* for inside points); **b** pinching and pricking; **c** scratching (elliptic area of influence)



5a

5b

Fig. 3. Function mapping controlled by a parametric curve

Fig. 4. Reconstruction of a femur from cross-sections. The *z*-coordinate (vertical axis) serves a parameter of the metamorphosis between neighboring cross-sections

Fig. 5. 3D coordinate space mapping with volume splines: **a** an initial set-theoretic object with a *red arrow* indicating the control point displacement; **b** a deformed object. After Salvador Dali's *The persistence of memory* (1931)

4.1.2 Metamorphosis

Assume that $d(x)$ in Eq. 10 is a defining function of some geometric object. The object resulting from this mapping will take some intermediate shape between two objects defined by $f(x)$. Introducing the parametric weighting functions $0 \leq h_1(t), h_2(t) \leq 1$, $h_1(0)=1$, $h_1(1)=0$, $h_2(0)=0$ and $h_2(1)=1$, we have

$$\xi' = h_1(t)\, f(x') + h_2(t)\, d(x'). \tag{13}$$

This expression defines a homotopy map between two hypersurfaces in E^{n+1}. Equivalently, it defines a parametrized family of geometric objects in E^n. If t is interpreted as a time variable, this mapping describes a time-dependent metamorphosis of one shape to another. If t is a geometric space coordinate, the mapping defines a metamorphosis of cross-sections along this coordinate. Figure 4 presents a femur reconstructed by metamorphosis from 2D cross-sections. To obtain a defining function for each cross-section from the contour points, we used the technique described in Sect. 4.1.1; we used $d(x)$ as a spline based on Green's function (see the Appendix). Then, the space coordiante z is interpreted as the metamorphosis parameter t, while $f(x, y)$ and $d(x, y)$ define two neighboring cross-sections for $z=0$ and $z=1$, $h_1(z)=1-z$ and $h_2(z)=z$ for $0 \leq z \leq 1$. A higher degree of interpolation between several cross-sections provides a more smooth surface.

4.2 Geometric coordinate space mappings

Let us consider the case of a one-to-one invertible mapping ϕ_1 of the subspace E^n and the identity mapping for the ξ coordiante. The system of Eqs. 6 takes the form

$$\begin{aligned} x' &= \phi_1(x) \\ \xi' &= \xi \\ \xi &= f(x). \end{aligned} \tag{14}$$

Note that this is the traditional type of E^n space mapping discussed by Barr (1984), Sederberg and Perry (1986), Borrel and Bechmann (1991), and others.

Taking into account the existence of the inverse mapping $x = \phi_1^{-1}(x')$, the transformed geometric object is now defined as:

$$\xi' = f\left(\phi_1^{-1}(x')\right). \tag{15}$$

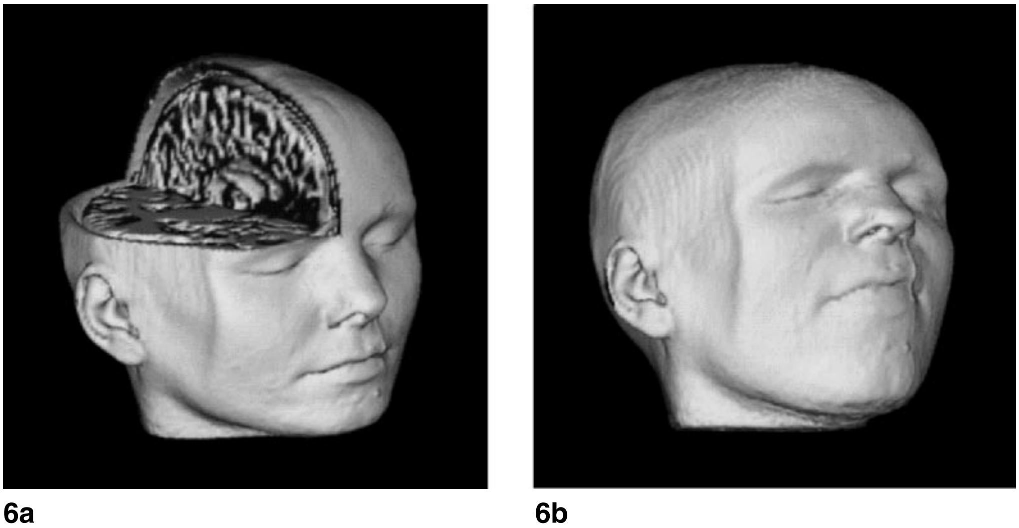
Here we discuss coordinate space mappings driven by control points linked to the features of an object. Space mapping in E^n-defines the relationship between each point in the original object and the deformed object. Let an n-dimensional region of arbitrary shape be given. Let it contain a set of arbitrary control points $\left\{Q_i = \left(q_1^i, q_2^i, \ldots, q_n^i\right) : i = 1, 2, \ldots, N\right\}$ for an undeformed object and corresponding points $\left\{D_i = \left(d_1^i, d_2^i, \ldots, d_n^i\right) : i = 1, 2, \ldots, N\right\}$ for a deformed object. By assumption, the points Q_i and D_i are distinct and given on or near a surface of each of the two objects. These points establish correspondence between features of the two objects. The goal of the construction of the deformed object is to find a smooth mapping function that approximately describes the spatial transformation. The inverse mapping function $\phi_1^{-1}(x')$ can be given in the form

$$x = x' + d(x'), \tag{16}$$

where the components of vector $d(x')$ are the functions interpolating displacements of the initial points Q_i to the points D_i. Since the algorithm does not use any regular grid, control points can be chosen freely by the user.

This approach is applied for 2D mappings by Bookstein (1991) and for 3D mappings by van Overveld (1992). To interpolate the displacements, we applied the volume spline function derived for multidimensional scattered data (Savchenko and Pasko 1994) on the base of the Green's function (Vasilenko 1983). The main advantage of this spline is the minimum bending energy of all functions that interpolate given scattered data. In the 2D case, it is a so-called ‘thin plate’ function (Alfeld 1989; Duchon 1977). The description of the volume spline is given in the Appendix. To interpolate control point displacements, we set $p_i = D_i$ and $\delta_i = D_i - Q_i$.

The example of 3D coordinate space mapping (with volume splines interpolating x, y, and z displacements) is shown in Fig. 5. A constructive object is mapped with the displacement of only one



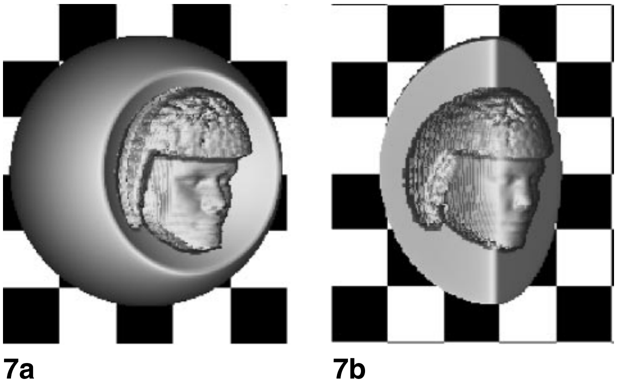


Fig. 6a, b. Simulation of facial expressions with coordinate space mapping defined by 13 control points: **a** initial volumetric head; **b** simulated facial expression

Fig. 7a, b. Offsetting "along the normal": offsets of 3D solids with the distance modulated by the depth data of the human head

control point. Note that the object is defined by a single real function constructed with R-functions. The facial expression simulation with the described mapping is illustrated by Fig. 6. The mapping is applied to a volumetric head defined by the trilinear interpolation of voxel data.

4.3 Function-dependent space mappings

Let the inverse mapping function of Eq. 15 take the specific form

$$\phi_1^{-1}(x') = \psi(x', f(x')),$$

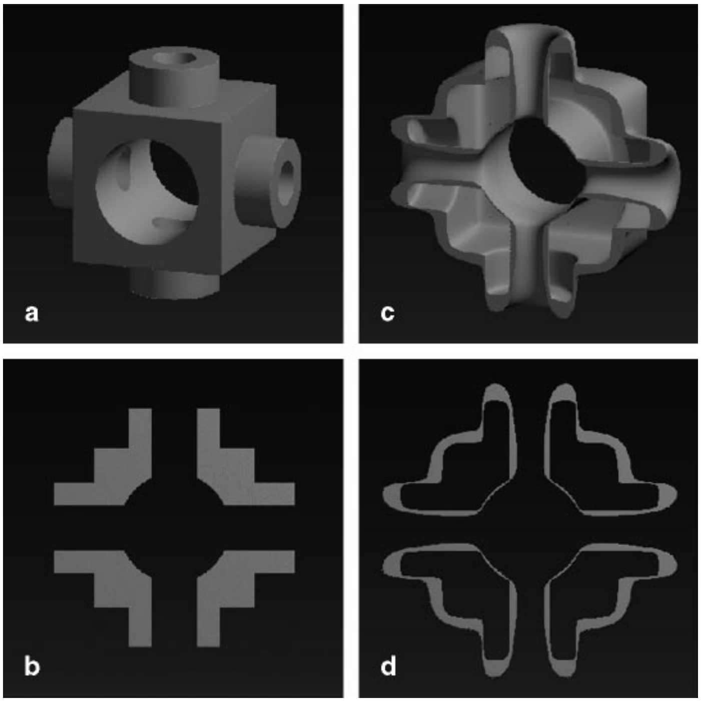
where $\psi=(x_1, x_2, \ldots, x_n)$ is a function generating point coordinates in E^n. For the transformed object, we have

$$\xi'=f(\psi(x', f(x'))). \qquad (17)$$

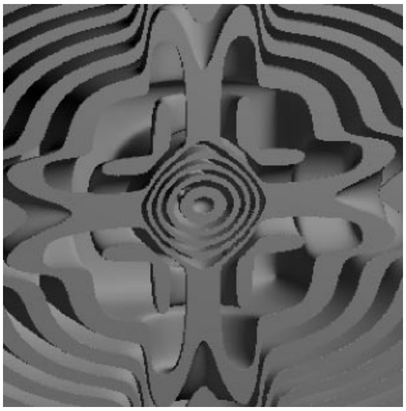
In this case, coordinate space mapping depends on the defining function and therefore on the shape of the initial object. Consider, for example, the following formulation of the offsetting "along the normal" operation:

$$\xi'=f(x'+d\mathbf{N}),$$

where d is a given offset distance and **N** is a gradient vector of the function f in a given point x'. The vector **N** is a normal vector for a point on the surface. This operation is illustrated by Fig. 7. Note that we have reformulated the notion of the offsetting "along the normal" by substituting the function gradient instead of the surface normal. This is provided by the C^1 continuity of the R-functions used to construct the objects. With the traditional approach, the shape of the offset in Fig. 7b would have a crack. To illustrate the power of the mapping of Eq. 17, we have applied the simple transformation



8



9

Fig. 8a–d. Function-dependent coordinate space mapping: **a** initial object; **b** cross-section of the initial object; **c** shell resulting from the transformation with the front half removed; **d** a cross-section of the transformed object

Fig. 9. Mazelike structure generated by function-dependent coordinate space mapping

$x_1=x_2=x_3=\mathrm{a}\,f(x')+b$ to a 3D constructive solid (see Fig. 8). The resulting object is a thin shell of the initial one with blended edges. A mazelike structure (Fig. 9) has been generated by the transformation $x_1=x_2=x_3=a\,\sin(b\,f(x')+c)$ applied to the same initial object.

4.4 Combined mappings

Let us now consider the general type mapping from Eqs. 6 and assume that the inverse mapping $x=\phi_1^{-1}(x')$ is known. We have the following description of the transformed object:

$$\xi'=\phi_2\left(\phi_1^{-1}(x'),f\left(\phi_1^{-1}(x')\right)\right). \tag{18}$$

Note that for our 1D example (see Sect. 3.2), Eq. 4 defines the combined mapping with $\phi_1^{-1}(x')=x'-c_1$ and $\phi_2(\xi)=\xi+c_2$. Now we illustrate Eq. 18 with an example of transformation from one 2D shape to another (Fig. 10) made by a combination of metamorphosis (see Sect. 4.1.2) and coordinate space mappings (see Sect. 4.2). The problem is to obtain a visually smooth transformation between 2D shapes described by the functions $f(x, y)$ (frame a1) and $d_1(x, y)$ (frame a5). The defining functions have been obtained with set-theoretic and blending operations on 2D primitives. The series of frames a1–a5 shows the process of metamorphosis

$$\xi'=(1-h_1(t))\,f(x')+h_1(t)\,d_1(x'),$$

where $h_1(t)=t$ and $0\leq t\leq 1$. This transformation is not satisfactory because of the appearance of two connected contours (frames a2, a3). The series of frames b1–b5 illustrates the coordinate space mapping controlled by the positions of three points. The initial and final positions of the control points are shown in the frames a1 and a5, respectively. The parametric description of this deformation is

$$\xi'=f(x'+h_2(t)\,d_2(x')),$$

where $h_2(t)=t$ and $d_2(x')$ represent two functions interpolating displacements of control points by two coordinates. The accuracy of this transformation depends on the number of control points, so the exact mapping for complicated shapes can hardly be achieved. We tried to use the advantages of both methods by applying combined mapping:

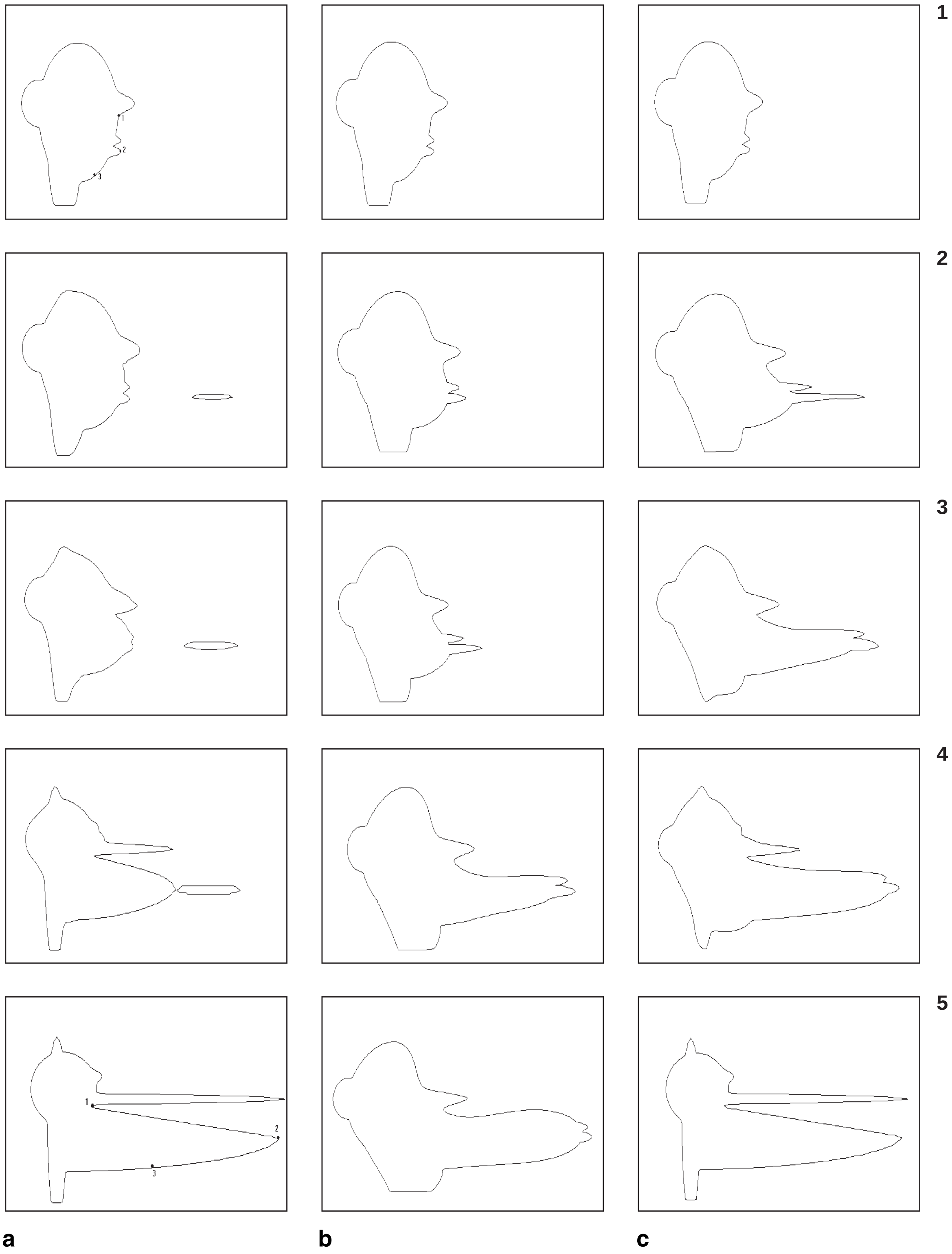
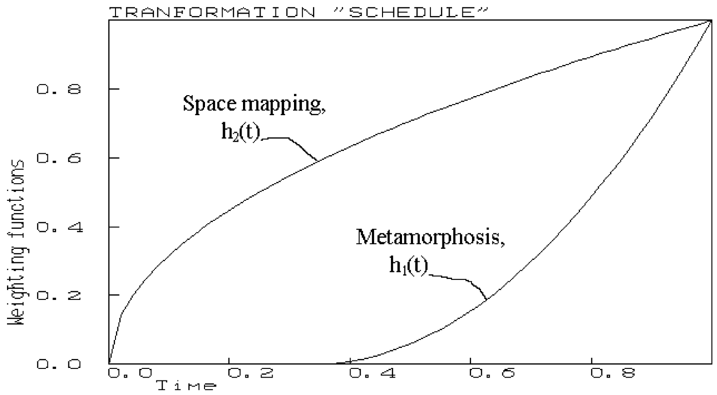
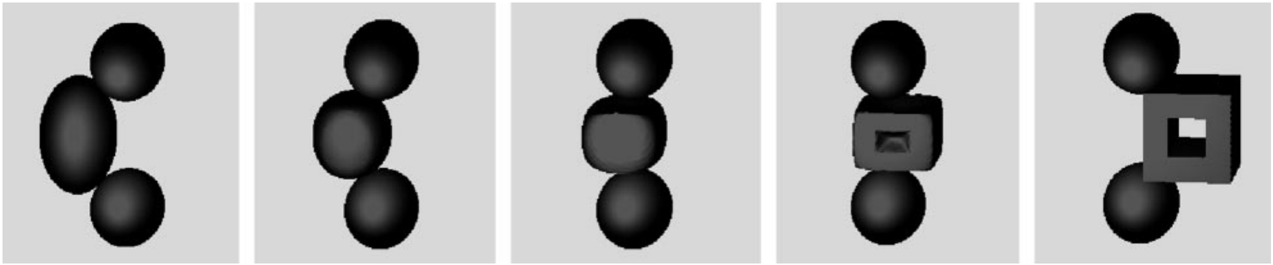


Fig. 10a–c. Transformation between two planar shapes: **a** metamorphosis; **b** coordinate space mapping; **c** combined mapping. Frames 1–5 show the time steps for each transformation method



11



12

Fig. 11. Time ”schedule” for the combined mapping shown in Fig. 10c

Fig. 12. Combined mapping in 3D space: metamorphosis and collision-free motion

$$\xi'=(1-h_1(t))\, f(x'+h_2(t)\, d_2(x'))+h_1(t)\, d_1(x').$$

The main problem here was to establish a “schedule” for the transformation, i.e., to find the parametric weighting functions. Frames c1–c5 illustrate the combined mapping with the schedule shown in Fig. 11, where

$$h_1(t)=0 \quad \text{for } t\le 0.3 \text{ and}$$
$$h_1(t)=(t-0.3)^2/0.49 \quad \text{for } t>0.3$$
$$h_2(t)=\sqrt{t}.$$

The weighting functions were chosen in an interactive design process with the criterion of the visual smoothness of the final transformation without disjoint components. We have started from the observation that the coordinate space mapping (frames b1–b5) does not generate any disjoint components. This means that it has to have a time priority with later corrections made by the metamorphosis. Note that, in frame sequence c1–c5, the second contour does not appear (compare frames a1–a5) and the final shape is the same as that required (compare frame a5). Now, it is unclear how to choose the schedules automatically because the process involves the formalization of human perception of visual smoothness. For example, for 2D and 3D output polygonal objects, we can use the Euler-Poincare formula to detect disjoint components.

We illustrate the 3D application of combined mappings in Fig. 12. The problem is to obtain the visually smooth transformation between the initial ellipsoidal shape and the final shape during motion in presence of obstacles. We obtained a defining function for a final shape of a block with a hole by using set-theoretic operations on 3D primitives. To solve the problem, we use a combination of metamorphosis and coordinate space mapping. We define the space mapping by displacements of control points in the local coordinate system of the moving object. The problem of calculating control point displacements to avoid collisions with obstacles or to minimize interpenetrating areas is an optimization problem. It can be solved by applying, for example, the genetic algorithm used by Savchenko et al. (1995a) to minimize interpenetration areas between cells in the cell colony growth modeling.

Table 1. Classification of the extended space mappings

Dependent on … Transformation of …	Geometric coordinates	Function coordinate	Geometric and function coordinates
Geometric coordinates	Geometric space mappings (Sect. 4.2)	Function-dependent space mappings (Sect. 4.3)	⊗
Function coordinate	New object	Function mappings 4.1	⊗
Geometric and function coordinates	⊗	⊗	Combined mappings (Sect. 4.4)

⊗ Denotes a group of restricted combined mappings not covered in this paper. These types provide a good opportunity for the further development of the extended space mappings

5 Conclusion

In this paper we presented a framework for transforming functionally defined shapes. The proposed general formulation of the extended space mapping covers several specific types of mappings summarized in Table 1. We discussed and illustrated special cases for which analytical solutions exist. The proposed model generalizes well-known coordinate space mappings, metamorphosis, other algebraic operations on defining functions, and less familiar function-dependent space mappings. It also introduces combined mappings. For example, the combination of geometric space mapping with metamorphosis provides wide opportunities to support desirable features in animation, design, and computer vision. The examples show that our approach makes it possible to find a proper solution when collision-preventing or other constrained deformations may be desirable.
Mainly, we have dealt with the functionally defined shapes, but application of this approach to polygonal shapes is an interesting future extension. Such topics as function mappings, function-dependent space mappings, and combined mappings obviously require further research. These mappings can be applied in aesthetic design and computer animation, but other fields of application have to be investigated.

Acknowledgements. We thank John Dill and Thomas Sederberg for their useful comments, and Michael Cohen for his help with stylistic issues of the manuscript.

Appendix. Volume spline based on Green's function

Data given:

- A set of scattered points $\left\{P_i = (x_1^i, x_2^i, \ldots, x_n^i) : i = 1, \ldots, N\right\}$ in E^n;
- Values of a scalar function δ_i=$\delta(P_i)$.

The problem is to find a smooth function $U(x)$ so that $U(P_i)$=δ_i. Green's function is used as a basic function:

$$G_{m,n}(x, P_i) = \begin{cases} \|x - P_i\|^{2m-n} \ln \|x - P_i\| & \text{if } n \text{ is even} \\ \|x - P_i\|^{2m-n} & \text{if } n \text{ is odd,} \end{cases}$$

where

$$\|x - P_i\| = \left(\sum_{j=1}^{n} \left(x_j - x_j^i \right)^2 \right)^{1/2}, x = (x_1, x_2, \ldots, x_n)$$

is an arbitrary point of E^n.
The coefficient $m \geq 2$ defines a norm, and m=2 can be used in practice.
For m=2 and n=2,3, the spline has the following form:

$$U(x) = \sum_{i=1}^{N+k} \lambda_i g_i(x, P_i),$$

where

$$g_i(x, P_i) = G_{m,\,n}(x, P_i), \qquad i = 1, \ldots, N,$$

$$g_{N+1}(x, P_i) = 1,$$

$$g_{N+1+j}(x, P_i) = x_j, \qquad j = 1, \ldots, k-1,$$

$$k = (n+m-1)!/(n!(m-1)!).$$

The spline coefficients λ_i are calculated with the system of $(N+k)$ linear equations:

$$A\begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \cdots \\ \lambda_N \\ \lambda_{N+1} \\ \lambda_{N+2} \\ \cdots \\ \lambda_{N+k} \end{vmatrix} = \begin{vmatrix} \delta_1 \\ \delta_2 \\ \cdots \\ \delta_N \\ 0 \\ 0 \\ \cdots \\ 0 \end{vmatrix}$$

$$\begin{array}{ll} A_{ij} = g_i(P_i, P_j), & i \leq N+k, j \leq N, i \neq j; \\ A_{ii} = 0; & i < N; \\ A_{ij} = g_j(P_i, P_j), & i \leq N, N < j \leq N+k; \\ A_{ij} = 0, & i > N, j > N. \end{array}$$

The system is solved by the Householder method. After defining the coefficients, the spline $U(x)$ can be restored. The choice of this spline was motivated by the following:

- It was derived especially for the case of scattered points.
- Minimal energy property.
- C^k continuity with $k<2\,m–n$.

References

Alfeld P (1989) Scattered data interpolation in three or more variables. In: Lyche T, Schumaker L (eds) Mathematical methods in computer aided geometric design. Academic Press, Boston, pp 1–33

Bajaj C, Ihm I, Warren J (1992) High order interpolation and least-squares approximation using implicit algebraic surfaces. ACM Trans Graph 11:327–347

Barr AH (1984) Global and local deformations of solid primitives. Comput Graph 18:21–30

Bechmann D (1994) Space deformation models survey. Comput Graph 18:571–586

Blinn JF (1982) A generalization of algebraic surface drawing. ACM Trans Graph 1:235:256

Bloomenthal J, Shoemake K (1991) Convolution surfaces. Comput Graph 25:251–256

Bloomenthal J, Wyvill B (1990) Interactive techniques for implicit modeling. Comput Graph 24:109–116

Bookstein FL (1991) Morphometric tools for landmark data. Cambridge University Press, New York

Borrel P, Bechmann D (1991) Deformation of N-dimensional objects. Int J Comput Geom Appl 1:427–453

Borrel P, Rappaport A (1994) Simple constrained deformations for geometric modeling and interactive design. ACM Transact Graph 13:137–155

Bowyer A (1994) SVLIS set-theoretic kernel modeller. Introduction and User Manual, Informaiton Geometers, Winchester, UK

Chang Y-K, Rockwood AP (1994) A generalized de Casteljau approach to 3D free-form deformation. Comput Graph Proceedings, Annual Conference Series, pp 257–260

Coquillart S (1990) Extended free-form deformation: a sculpting tool for 3D geometric modeling. Comput Graph 24: 187–196

Coquillart S, Jancene P (1991) Animated free-form deformation: an interactive animation technique. Comput Graph 25: 23–26

Duchon J (1977) Splines minimizing rotation-invariant seminorms in Sobolev spaces. In: Dodd A, Eckmann B (eds) Constructive theory of functions of several variables. Springer, Berlin Heidelberg New York, pp 85–100

Fournier A, Wesley MA (1983) Bending polyhedral objects. Comput Aided Des 15:79–87

Galin E, Akkouche S (1996) Blob metamorphosis based on Minkowski sums. EUROGRAPHICS'96, Comput Graph Forum 15:143–153

Gascuel MP (1993) An implicit formulation for precise contact modeling between flexible solids. Computer Graphics Proceedings, Annual Conference Series, pp 313–320

Hoffmann CM (1993) Implicit curves and surfaces in CAGD. IEEE Comput Graph Appl 13:79–88

Hsu WM, Hughes GF, Kaufman H (1992) Direct manipulation of free-form deformations. Comput Graph 26:177–184

Hughes JF (1992) Scheduled Fourier volume morphing. Comput Graph 26:43–46

Larcombe MHE (1994) Generalised distance functions and plastic modelling over an armature. In: Bowyer A (ed) Computer aided surface geometry and design, the mathematics of surfaces IV, Clarendon Press, Oxford, pp 281–298

Lazarus F, Coquillart S, Jancene P (1993) Interactive axial deformations. In: Falcidieno B, Kunii TL (eds) Modeling in Computer Graphics, Springer, Berlin Heidelberg New York, pp 241–254

Lee S, Wolberg G, Chwa K-Y, Shin SY (1996) Image metamorphosis with scattered feature constraints. IEEE Trans Visualization Comput Graph 2:337–354

Lerios A, Garfinkle CD, Levoy M (1995) Feature-based volume metamorphosis. Computer Graphics Proceedings, Annual Conference Series, pp 449–456

Muraki S (1991) Volumetric shape description of range data using 'blobby model'. Comput Graph 25:227–235

Nishimura H, Hirai M, Kawai T, Kawata T, Shirakawa I, Omura K (1985) Object modeling by distribution function and a method of image generation (in Japanese). Transactions of IECE, vol. J68-D, No. 4, pp 718–725

Nishita T, Fujii T, Nakamae E (1993) Metamorphosis using Bézier clipping. Comput Graph Appl Proceedings of Pacific Graphics'93, World Scientific, pp 162–175

Overveld CWAM van (1992) Beyond bump maps: nonlinear mappings for the modeling of geometric details in computer graphics. Comput Aided Des 24:201–209

Pasko AA, Adzhiev VD, Sourin AI, Savchenko VV (1995) Function representation in geometric modeling: concepts, implementation and applications. Visual Comput 11:429–446

Payne B, Toga A (1992) Distance field manipulation of surface models. IEEE Comput Graph Appl 12:65–71

Ruprecht D, Nagel R, Muller H (1995) Spatial free-form deformation with scattered data interpolation methods. Comput Graph 19:63–71

Rvachev VL (1963) On the analytical description of some geometric objects. Reports of the Ukrainian Academy of Sciences (in Russian). 153:765–767

Rvachev VL (1987) Theory of R-functions and some applications (in Russian). Naukova Dumka, Kiev, Ukraine

Savchenko VV, Pasko AA (1994) Shape transformations of 3D geometric solids. Proceedings of the International Workshop "Shape Modeling: Parallelism, Interactivity and Applications", University of Aizu, Aizu, Japan, pp 47–53

Savchenko VV, Basnakian AG, Pasko AA, Ten SV, Huang R (1995a) Simulation of a growing mammalian cell colony: collision-based packing algorithm for deformable particles. In: Earnshaw R, Vince J (eds) Comput graphics: developments in virtual environments, Academic Press London, UK, pp 437–447

Savchenko VV, Pasko AA, Okunev OG, Kunii TL (1995b) Function representation of solids reconstructed from scattered surface points and contours. Comput Graph Forum 14:181–188

Sclaroff S, Pentland A (1991) Generalized implicit functions for computer graphics. Comput Graph 25:247–250

Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. Comput Graph 20:151–160

Shapiro V (1994) Real functions for representation of rigid solids. Comput Aided Geom Des 11:153–175

Singh K, Parent R (1995) Implicit function based deformations of polyhedral objects. Wyvill B, Gascuel MP, (eds). Implicit Surfaces'95, Eurographics Workshop, INRIA, Grenoble, France, pp 113–128

Vasilenko VA (1983) Spline functions: theory, algorithms and programs (in Russian). Nauka, Novosibirsk

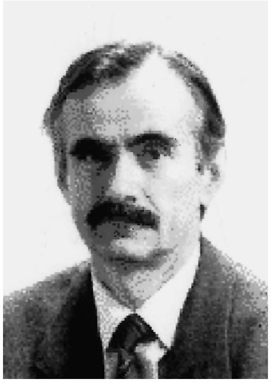
Woodwark JR (1987) Blends in geometric modeling. In: Martin RR (ed) The mathematics of surfaces II, Oxford University Press, Oxford, UK, pp 255–297

Wyvill B (1990) A computer animation tutorial. In: Rogers DF, Earnshaw RA (eds) Computer graphics techniques: theory and practice, Springer, Berlin Heidelberg New York, pp 235–282

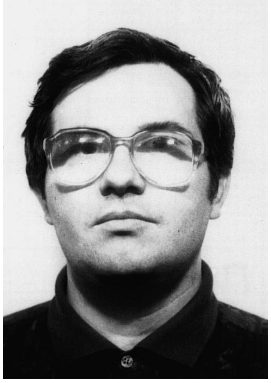
Wyvill G, McPheeters C, Wyvill B (1986) Data structure for soft objects. Visual Comput 2:227–234

Wyvill B, Bloomenthal J, Bajaj C, Beier T, Blinn J, Hart J, Wyvill J (1993) Modeling, visualizing and animating implicit surfaces. SIGGRAPH'93 conference, Anaheim, CA, Course Notes 25. ACM, New York

Wyvill G, McRobie D, Haig C (1994) Free-form modelling with history. Proceedings of the International Workshop 'Shape Modeling: Parallelism, Interactivity and Applications', University of Aizu, Aizu-Wakamatsu, Japan, University of Aizu, Aizu-Wakamatsu, Japan, pp 69–75



VLADIMIR V. SAVCHENKO is a Professor at the Shape Modeling Laboratory, Computer Software Department, University of Aizu, Japan. He received his MSc in Mechanical Engineering from the Moscow Aviation Institute in 1971 and his PhD in Theoretical Mechanics from the Institute of Applied Mathematics in Moscow in 1984. He was Head of the Computational Mechanics Department at the Scientific Research Institute of System Analysis, part of the Russian Academy of Sciences, from 1989 to 1992. His research interests include parallel processing, geometric modeling, computer-aided design, and artificial life.



ALEXANDER A. PASKO is an Assistant Professor at the Shape Modeling Laboratory, Computer Software Department, University of Aizu, Japan. He received his MSc and PhD degrees in Computer Science from the Moscow Engineering Physics Institute (MEPI) in 1983 and 1988, respectively. He was a researcher at MEPI from 1983 to 1992. His research interests include solid modeling, animation, multidimensional visualization, experimental data processing, and computer art.