Shape Modeling with Some Applications to the Cultural Preservation

形状モデリングと文化的保存への応用

Doctoral Thesis

Author: Pasko Galina Ivanovna
Supervisor: Professor Kunii Toshiyasu Laurence

IT Institute of Kanazawa Institute of Technology
Tokyo, Japan
2005
To my Teachers
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>7</td>
</tr>
<tr>
<td>Chapter I \hspace{1em} Shape modeling and applications \hspace{1em}</td>
<td>11</td>
</tr>
<tr>
<td>1.1 Abstraction hierarchy of models</td>
<td>11</td>
</tr>
<tr>
<td>1.2 Shape models of different hierarchy levels</td>
<td>14</td>
</tr>
<tr>
<td>1.2.1 Boundary representation</td>
<td>14</td>
</tr>
<tr>
<td>1.2.2 Constructive Solid Geometry</td>
<td>16</td>
</tr>
<tr>
<td>1.2.3 Function representation</td>
<td>19</td>
</tr>
<tr>
<td>1.3 Blending operations in shape modeling</td>
<td>23</td>
</tr>
<tr>
<td>1.3.1 Requirements to blending operations</td>
<td>24</td>
</tr>
<tr>
<td>1.3.2 Blending based on R-functions</td>
<td>25</td>
</tr>
<tr>
<td>1.3.3 Localization of blending</td>
<td>31</td>
</tr>
<tr>
<td>1.4 Shape metamorphosis</td>
<td>31</td>
</tr>
<tr>
<td>1.5 Modeling heterogeneous objects</td>
<td>34</td>
</tr>
<tr>
<td>1.5.1 Dimensional heterogeneity</td>
<td>35</td>
</tr>
<tr>
<td>1.5.2 Cellular-functional model</td>
<td>36</td>
</tr>
<tr>
<td>1.5.3 Polygonization and trimming implicit surfaces</td>
<td>38</td>
</tr>
</tbody>
</table>
1.6 Cultural preservation ........................................................................................................ 44

1.6.1 Cultural heritage terms and notions .............................................................................. 44

1.6.2 Digital preservation purposes ...................................................................................... 45

1.6.3 Approaches to shape preservation ............................................................................... 47

1.6.4 Problems of cultural preservation ................................................................................ 57

1.6.5 Shape model selection for cultural preservation .......................................................... 58

Chapter II Shape models and operations .............................................................................. 61

2.1 Bounded blending ............................................................................................................ 61

2.1.1 Displacement function ................................................................................................. 61

2.1.2 Bounding by control points ......................................................................................... 63

2.1.3 Bounding solid ............................................................................................................ 66

2.1.4 Properties of bounded blending .................................................................................. 68

2.1.4.1 Control of 3D blend shape ...................................................................................... 74

2.1.4.2 Multiple blending .................................................................................................. 76

2.1.4.3 Partial edge blending ............................................................................................ 77

2.1.4.4 Blend on blend ...................................................................................................... 77

2.1.4.5 Blends in constructive solids ................................................................................ 78

2.2 Space-time bounded blending ........................................................................................ 79

2.2.1 Shape transformation using higher-dimensional blending ........................................ 81

2.2.2 Shape transformation using bounded blending .......................................................... 85
2.2.3 Rapid transition problem .................................................. 89
2.2.4 3D shape transformation using space-time blending .................. 95
2.3 Model of two-dimensional manifold with boundary ...................... 99

Chapter III Algorithms and implementation .................................. 103
3.1 Modeling and rendering software ............................................. 103
3.2 Bounded blending models .................................................... 108
  3.2.1 Model of bounded blending with control points ..................... 108
  3.2.2 Model with a bounding solid ......................................... 110
3.3 Space-time bounded blending models ...................................... 113
  3.3.1 Model of 2D metamorphosis ......................................... 113
  3.3.2 Model of 3D metamorphosis ......................................... 116
3.4 Implicit surface trimming algorithm ....................................... 121

Chapter IV Case studies .......................................................... 127
4.1 Modeling Japanese craft ...................................................... 127
  4.1.1 Virtual Shikki project .................................................. 127
  4.1.2 Implementation issues ................................................ 129
4.2 Metamorphosis using space-time bounded blending .................... 134
  4.2.1 2D metamorphosis in the Dancing Buddhas project ............... 134
  4.2.2 3D metamorphosis using 4D space-time blending .................. 139
4.3. Conceptual abstraction hierarchy analysis of flowers.............................143

4.3.1 Adjunction space level.................................................................143

4.3.2 Cellular structured space level...................................................146

4.3.3 Geometry level.................................................................147

4.3.4 Implementation of geometric level........................................148

4.4. Modeling M.C. Escher’s rinds.....................................................151

Conclusion........................................................................................157

Acknowledgements............................................................................161

References........................................................................................162
Introduction

Modeling of geometric aspects of real life and abstract objects and phenomena has become one of major applications of computer technologies. This area has been traditionally known as “geometric modeling” for more than 20 years. However, geometric modeling is usually connected to polygonal meshes, parametric curves and surfaces. Many other different types of geometric models have emerged such as implicit surfaces, solid and volume models, topological models, grammar-based models (L-systems, fractals), and others. The term "shape modeling" has been introduced to provide a common framework for all these new models. In the general case, a shape can be considered a point set in a multi-dimensional space. Therefore, shape modeling can be defined as an area of computer science studying methods and tools for modeling point sets in geometric spaces (e.g., an n-dimensional Euclidean space).

One of relatively new shape models of higher abstraction level is the function representation (FRep). It is a generalization of traditional implicit surfaces, constrictive solid geometry, voxel, sweeping, and other shape models. This representation supports a wide class of primitive objects and operations on them. The generality of this model comes from the main principle postulating that the modeled object at every step of the modeling process is described in uniform manner by a single procedurally defined real continuous function of point coordinates. This dimension independent description allows for uniform treatment of space-time and higher dimensional objects. In its original form, the function representation supports modeling of homogeneous objects in the dimensionality and material properties.
There are practical challenging applications such as computer-aided design and rapid prototyping of mixed-material objects, geological and medical modeling, digital preservation of cultural heritage, computer animation, and others. These applications formulate the following new requirements to be satisfied by the function representation and its extensions:

- intuitive and interactive user’s control over the modeled shape;
- precise predictability of the result of modeling operations;
- generality of modeling methods independent of specific restrictions and preprocessing steps;
- support of dimensionally heterogeneous and multi-material models;
- models formulation allowing for further conversion into other auxiliary models suitable for different applications;

In this thesis, we review and classify existing shape models in the form of abstraction hierarchy, where the geometric level and the cellular structured space level are the major topics of interest in our work. On this basis, we discuss cellular modeling of dimensionally heterogeneous objects using the extension of the function representation in the form of implicit complexes. We select several open shape modeling problems and propose original solutions to the formulated problems using the function representation for bounded blending and controlled metamorphosis. From the cellular modeling point of view, we concentrate on modeling of two-dimensional cells in the form of trimmed implicit surfaces.
We implement the proposed solutions and illustrate them by several digital preservation related case studies such as the Virtual Shikki project on modeling and Web presentation of Japanese lacquer ware craft, the Dancing Buddhas project, and 3D modeling of the Escher’s drawings. The term *digital preservation* means the capture and archiving of the form and contents of the existing cultural objects through the use of computer modeling techniques and the reproduction of those objects that have already been lost. Traditionally for digital preservation, external surfaces and textures of objects are considered. From our point of view, internal structures (revealing the logic of construction) of objects, as well as their time, and other parametric dependencies also can be added to the consideration.

In Chapter I, the abstraction hierarchy of shape models is briefly introduced, and several geometric models are described in more detail (boundary representation (BRep), constructive solid geometry (CSG) and functional representation). Surveys of related works on blending and metamorphosis operations are given and the problems of localization and control are formulated. Implicit complexes are characterized as an appropriate model for the functionally based heterogeneous objects representation, and a specific question of modeling of two-dimensional cells in the form of trimmed implicit surfaces is discussed.

From the application point of view, we define some cultural preservation terms, classify and compare the different approaches to computer-aided preservation of culturally valuable objects. We justify the selection of the function representation and implicit complexes as the basic mathematical models for the geometric and cellular levels for cultural preservation.
In Chapter II, a solution to the localized and controlled blending problem, an original bounded blending operation is proposed. Two types of this operation are described using the theory of R-functions: bounding by control points and bounding by an additional solid. Then, we illustrate the original properties of the proposed bounded blending operation. The bounding blending operation is also applied to formulate an original approach to shape metamorphosis through the dimension increase and objects blending in space-time, which allows us to eliminate most of existing constrains in shape transformations. To support modeling of implicit complexes, we introduce a model of two-dimensional manifold with boundary as a trimmed implicit surface.

In Chapter III, we briefly describe existing software tools supporting implicit surfaces and FRep modeling, and provide details of the HyperFun language chosen for the implementation of the proposed models. For each of the models (bounded blending and space-time blending), we give and comment line-by-line several HyperFun programs. For trimmed implicit surfaces, we propose an adaptive polygonization algorithm.

In Chapter IV, we present several case studies on cultural preservation and utilization of shapes using the proposed methods. The Virtual Shikki project, where the bounded blending operation was actively used for modeling traditional Japanese lacquer ware items, is described. The proposed space-time blending is used in the case studies of 2D- and 3D metamorphosis, and in the Dancing Buddhas project particularly. Conceptual abstraction hierarchy analysis of flowers is done on the adjunction space and the cellular levels with the geometric level implemented and illustrated using HyperFun tools. Finally, the application of trimmed implicit surfaces for modeling spiral objects by M. C. Escher is presented.
Chapter I  Shape modeling and applications

1.1 Abstraction hierarchy of models

At first, we present a general abstraction hierarchy of shape models, which we use in our work for classification of existing models and explanation of newly introduced models and algorithms. From this hierarchy point of view, our work is concentrated on geometric level and cellular structured space level. Then, in the following section we discuss the existing models of the geometric level. A related model of the cellular structured space level is presented in section 1.5.

On the basis of the works [Kunii 2000, Kunii 2002, Kunii et al. 2003], we provide here a brief characteristic of the abstraction hierarchy of shape models consisting of the following levels:

**Extension theory level**

The highest abstract level is extension theory level with the homotopy theoretical level as its typical case. At the homotopy theoretical level, homotopy can be preserved by making any changes reversible. The homotopy equivalence of object is more general than, for example, topological equivalence (see below) such that the homotopy equivalence can identify a changing object that topologically is not anymore equivalent after the change.
Set theoretical level

For any objects to be processed by computers that are set theoretical automatic machines, these objects have to be defined in set theoretical spaces. It means the objects are considered at the set theoretical space level as sets of elements. For given sets \( X \) and \( Y \), we can introduce new sets by performing so-called set-theoretic operations such as union \( X \cup Y \), intersection \( X \cap Y \), and difference \( X \setminus Y \). By defining reversible operations to add elements to a given set or delete them from it, we can create one level higher than the set theoretical level, namely the homotopy theoretical level mentioned above.

Topology theoretical level

Two topological spaces are topologically equivalent, if there is a function providing homeomorphic mapping, and its inverse exists and is continuous. Any object we deal with usually has subsets as its elements, and hence, by definition, it is a discrete topological space. Any set theoretical space with its subsets as elements is basically a discrete topological space. At this level, the topology we adopt for modeling is discrete topology. A graph theoretical level is considered a special case.

Adjunction space level

At the adjunction space level, a dynamic relation between objects is expressed by an equivalence relation \( f \) between objects such that an object \( X \) has become related via \( f \) with another object \( Y \) by sharing a part \( Y_0 \). There are numerous cases of the adjunction spaces, for example, those when industrial products are assembled.
**Cellular structured space level**

At the cellular structured space level, the inductive dimension $n$ of each object defined at the adjunction space level is added as the degrees of freedom of the object. We abbreviate an inductive dimension to a dimension for simplicity. For example, usually the shapes of industrial products are three-dimensional models. However, models of heterogeneous or mixed-dimensional objects are required in many applications as will be discussed in section 1.5. Such models can be well presented on the cellular structured space level.

**Geometry level**

This level and lower is application domain dependent. At the geometry level, geometrical properties such as coordinate systems and metric systems are added to the objects defined at the cellular structured space level. It results in introducing point coordinates and coefficients of equations for the existing cellular model data. We further discuss the geometry level models in the following sections.

**View level**

Models of the view level are typically defined in two-dimensional space (e.g., picture plane and device coordinates), and projection operations have to be involved for mapping three- and higher dimensional models of the geometry level to models of this level.
1.2 Shape models of different hierarchy levels

Let us briefly characterize the following shape models from the geometric level point of view: boundary representation (BRep) and constructive representations such as Constructive Solid Geometry (CSG), and Function Representation (FRep). In fact, BRep has elements of the cellular structured space level, and constructive representations include information of the set theoretical level along with geometric information. Formal definitions and more details on solids and solid representations can be found in [Requicha 1980, Hoffmann 1989, Savchenko and Pasko 2001].

1.2.1 Boundary representation

A solid can be represented by its boundary surface. To define a boundary surface one can introduce points (vertices), curves (edges), and surface patches (faces), and stitch them together (Fig. 1.1 left). This boundary representation (or BRep) has two parts (Fig. 1.1 right): topological information on the connectivity of vertices, edges, and faces, and geometric information embedding these boundary elements in three-dimensional space. Topological information specifies incidences and adjacencies of boundary elements. Geometric information specifies coordinates of vertices or the equations of the surfaces containing the faces. The boundary of the solid is a two-dimensional manifold. Each point of the boundary has a neighborhood with one-to-one correspondence to a disk in the plane.
Local modifications of the boundary are performed using such operations as moving vertex, edge, or face. Topological modifications are performed using Euler operators, which include adding and removing vertices, edges, and faces. These operators satisfy Euler's formula and thus ensure topological validity of the resulting solids.

**Figure 1.1:** Boundary representation of a cube is based on surface faces (triangles and/or quadrangles) $F_i$, edges $E_k$, and vertices $V_j$.

From the practical modeling point of view, BRep is used for visualization of CSG or FRep defined objects. Currently, most commercial modeling programs use BRep not only for visualization but also for mathematical definition of objects. Systems based on this approach are exceedingly complex and prone to error.

Main disadvantages of BRep are:

- verbose models (especially in the case of free form surface approximation by
triangles).
- 2D-manifold boundaries are not closed under set-theoretic operations on corresponding solids.
- BRep is not directly applicable for representing topological complexes with elements of lower dimensionality.

The typical problems of BRep models in practice are cracks (or gaps) between adjacent faces, inappropriate intersections, incorrect normals, and internal walls (non-manifold topology), and others. Moreover, BRep models do not represent logical structure of the object or history of its creation. Due to above problems, BRep models should not be considered archival quality models. BRep is helpful during the creation and editing of geometric primitives and necessary for rendering using modern graphics hardware. Hybrid systems using BRep based interaction and visualization together with mathematically rigorous representations are needed for digital preservation of cultural heritage objects.

1.2.2 Constructive Solid Geometry

Using CSG modeling, one can begin by selecting simple solids (primitives), specifying their parameters and positions in space, and then using them to construct more complex solids by applying set-theoretic operations such as union, intersection, or subtraction (Fig.
Traditional CSG primitives are ball, block, cylinder, and cone. Linear transformations (translation and rotation) can be used together with regularized set-theoretic operations. A regularized set-theoretic operation includes removing lower dimensional parts from the result of the standard set operation such as dangling surfaces, curves or points.

**Figure 1.2:** Set-theoretic operations between two 2D disks: union ($\cap$), intersection ($\cup$), and subtraction ($\setminus$). The result of each operation is shown as a hatched area.

A CSG object is represented as a binary tree (or a CSG tree) with operations placed at the internal nodes and primitives at the leaves (Fig. 1.3). The point membership classification algorithm defines whether a given point is inside, outside, or on the boundary of the solid. This algorithm recursively traverses the CSG tree starting from the root. In the nodes with linear transformations, the inverse transformation is applied to the current point coordinates. When the recursion reaches the leaves, the point is tested against the corresponding primitives. Then, the classification results are combined in the internal nodes with set-theoretic operations.
From the practical modeling point of view, CSG inherently provides a constructive history, which allows interactive editing of sub-elements. If a complex object is created with CSG, its constructive primitives and the order in which they were processed can be accessed by application algorithms. Furthermore, CSG allows for surface calculations of area and mass calculations of weight, volume, and centricity of solids. The disadvantage of CSG is its limitation in geometrical representation; though it performs well for most mechanical and architectural objects, it is not suitable for producing organically looking free-form shapes.

Figure 1.3: Example of a CSG tree. Operations: R (rotation), - (subtraction), ∪ (union).
1.2.3 Function representation

The function representation (FRep) was introduced in [Pasko et al. 1993b, Pasko et al. 1995] as a generalization of traditional implicit surfaces [Bloomenthal et al. 1997], CSG, sweeping, and other shape models. In FRep, a 3D object is represented by a continuous function of point coordinates as \( F(x,y,z) \geq 0 \). A point belongs to the object if the function is non-negative at the point. The function is zero on the entire surface (called usually an implicit surface) of the object and is negative at any point outside the object. The function can be easily parameterized to support modeling of a parametric family of objects.

In a FRep system, an object is represented by a tree data structure (similar to one used in CSG, see Fig. 1.3) reflecting the logical structure of the object construction, where leaves are arbitrary "black box" primitives and nodes are arbitrary operations [Pasko and Adzhiev, 2004]. Function evaluation procedures traverse the tree and evaluate the function value at any given point. For the node with space mapping an inverse transformation is applied to the current point coordinates. When the recursion reaches a leave, the defining function of the primitive is evaluated. Then, traversing back from the leaves to the root, binary and k-ary operations are applied to the defining function values of the corresponding sub-trees.

The following types of geometric objects can be used as primitives (leaves of the construction tree):

- algebraic surfaces and skeleton-based implicit surfaces [Bloomenthal et al.1997]
- convolution surfaces [J. McCormack and A. Sherstyuk, 1998]
- objects reconstructed from surface points and contours [Savchenko et al. 1995]
- polygonal shapes converted to real functions [Pasko et al. 1996a, Pasko et al. 1996b]
- procedural objects (such as solid noise) [Pasko and Savchenko 1993a]
- volumetric (voxel) objects [Savchenko et al. 1998, Adzhiev et al. 2000].

Many modeling operations have been formulated, which are closed on the representation, i.e., generate another continuous function defining the transformed object as a result. These modeling operations include:

- blending set-theoretic operations [Pasko and Savchenko 1994a]
- offsetting [Pasko and Savchenko 1994b]
- sweeping by a contour [Pasko et al. 1996b] and by a moving solid [Sourin and Pasko 1996]
- projection to a lower dimensional space [Pasko and Savchenko 1997]
- non-linear deformations and metamorphosis [Savchenko and Pasko 1998]

- Minkowski sums [Pasko et al. 2003].

A new operation can be included in the modeling system without changing its integrity by providing a corresponding function evaluation or space mapping procedure. In FRep, there is no principal difference in processing skeleton-based objects, CSG solids, or volumetric objects (with an appropriate samples interpolation). This allowed researchers to solve such long standing problems as metamorphosis between objects of different topology, sweeping by a moving solid, controlled blending for all types of set-theoretic operations, collision detection [Savchenko and Pasko 1995] and hypertexturing for arbitrary solids [Sourin et al. 1996], and direct modeling of space-time and multidimensional objects [Fausett et al. 2000].

The major problems of FRep are time-consuming function evaluation at the point, capturing sharp and thin objects features during rendering, interactive manipulation and editing. Parallel and distributed processing and special hardware can help resolve most of these problems.

The HyperFun language [HyperFun, Adzhiev et al. 1999] was introduced for teaching and practical use of FRep modeling. It is a minimalist programming language supporting all notions of FRep. The following tools are available for processing HyperFun models: a polygonizer that generates a polygonal mesh on the surface of the object and exports it in the VRML format; and a plug-in for the POV-Ray ray-tracer that helps to generate high-
quality photorealistic images. Application software deals with HyperFun models through an interpreter, which evaluates the defining function at any given point.

FReP also naturally supports 4D (space-time) and multidimensional modeling using functions of several variables. The main idea of visualization is to provide a mapping of such objects to a multimedia space with such coordinates as 2D/3D world space coordinates, time, color, textures and other photometric coordinates, and sounds. Deeper connections between multimedia space and geometric multidimensional spaces should be investigated in the context of computer animation, computer art, and cultural heritage preservation applications.

HyperFun was also designed to serve as a lightweight protocol for exchanging FReP models among people, software systems, and networked computers. The size of HyperFun files is usually relatively small (5-10K). This allows for efficient implementation of a client-server modeling system in which a client can run simple interface tasks and generate HyperFun protocols to be sent to the server. The server site can be a powerful parallel computer or a computer cluster that performs time-consuming tasks such as ray-tracing, polygonization, or voxelization. The open and simple textual format of HyperFun, its clearly defined mathematical basis, its support of constructive, parameterized and multidimensional models, its support by free modeling and visualization software, and its ease of use make it a good candidate as a tool for the digital preservation of cultural heritage objects.
1.3 Blending operations in shape modeling

In shape modeling, it is often necessary to generate smooth transition between several surfaces. Such an operation is called blending. Blending in solid modeling means operation of generation of a new surface, which smoothly connects two given surfaces [Rossignac and Requicha, 1984]. Natural blending is considered one of the advantages of implicit surface modeling [Bloomenthal et al. 1997]. The blending effect appears when applying an algebraic sum (or difference) to two or several skeleton-based scalar fields. In the case of blending of two disjoint surfaces, a single blend surface can be generated. Controlled blending has been tackled in implicit surface modeling research in the context of skeletal primitives [Guy and Wyvill 1995, Wyvill and Wyvill 2000], convolution surfaces [Angelidis et al. 2002], or multiple intersecting implicit surfaces [Hartmann 2001]. The addressed problems are control of the overall shape of blend [Wyvill and Wyvill 2000, Barthe et al. 2003], blend on blend [Hartmann 2001], and preventing unwanted blending in cases of contact between different surfaces or a folding skeleton curve [Guy and Wyvill 1995, Angelidis et al. 2002].

Note that the major operations in implicit surface modeling (algebraic sum and difference) do not describe set-theoretic union or subtraction, which are key operations in solid modeling. Let us consider the works particularly oriented to blending versions of set-theoretic operations on solids with implicit surfaces. Blending operations are usually used in solid modeling to create fillets and chamfers. Blending versions of set-theoretic operations (intersection, union, subtraction) on solids approximate the exact results of
these operations by rounding sharp edges and vertices. In the case of the blending union of two disjoint solids with added material, one resulting solid with a smooth surface can be generated.

1.3.1 Requirements to blending operations

Detailed discussions of blending techniques can be found in [Rossignac and Requicha 1984, Woodwark 1987, Pasko and Savchenko 1994a, Bloomenthal et al. 1997]. The major requirements to blending operations are:

- Tangency of the blend surface with the initial surfaces: blending operation should not create additional sharp edges.
- Automatic clipping of unwanted parts of the blending surface: only behavior of the blending surface inside of the blending area is the matter of interest.
- Exact analytical or procedural definitions of blends instead of approximations.
- Blending definition of the basic set-theoretic operations: standard set-theoretic operations result in sharp edges, which need to be smoothed by blending.
- Support of added and subtracted material blends;
- Intuitive control of the blend shape and position: parameters of blending should have clear geometric interpretation.

There are additional and sometimes controversial requirements:
• blended objects should be considered legal arguments of all operations applicable to unblended objects;
• blend on blend operation should be supported including complex vertices where three or more surfaces are blended;
• \( C^1 \) continuity of the blending function everywhere in the domain;
• blending of a single selected edge;
• support of constant-radius blending and variable-radius blending.

Let us consider the works particularly oriented to blending versions of set-theoretic operations on solids with implicit surfaces [Ricci 1973, Middleditch and Sears 1985, Pasko and Savchenko 1994a]. The blending method of [Ricci 1973] is based on the approximations of min/max functions used for set-theoretic operations. The proposed blending has global character in the sense that the entire initial surfaces are replaced by the blending surface. This formulation provides only added material blend for union and subtracted material blend for intersection. The work [Middleditch and Sears 1985] provides analytical definitions for blending surfaces. However, added or subtracted material blends require additional set-theoretic operations for the appropriate truncation of the unwanted parts of the blend solid.

1.3.2 Blending based on R-functions

An analytical definition of blending set-theoretic operations on solids with implicit surfaces was proposed in [Pasko and Savchenko 1994a]. This work was motivated by observing properties of R-functions [Rvachev 1987, Shapiro 1994, Pasko et al. 1995],
which provide exact analytical definitions of set-theoretic operations with $C^1$ and higher-level continuity of the resulting function.

An object resulting from set-theoretic operations has the defining function as follows:

\[ f_3 = f_1 \lor_a f_2 \] for the union;
\[ f_3 = f_1 \land_a f_2 \] for the intersection;
\[ f_3 = f_1 \setminus_a f_2 \] for the subtraction,

where $f_1$ and $f_2$ are defining functions of initial objects and $\lor_a, \land_a, \setminus_a$ are signs of R-functions. One of the possible analytical descriptions of the R-functions is as follows:

\[
\begin{align*}
    f_1 \lor_a f_2 &= \frac{1}{1 + \alpha} \left( f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2} \right) \\
    f_1 \land_a f_2 &= \frac{1}{1 + \alpha} \left( f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2} \right)
\end{align*}
\]

where $\alpha = \alpha(f_1, f_2)$ is an arbitrary continuous function satisfying the following conditions:

\[-1 < \alpha(f_1, f_2) < 1,\]
\[\alpha(f_1, f_2) = \alpha(f_2, f_1) = \alpha(-f_1, f_2) = \alpha(f_1, -f_2)\]

The expression for the subtraction operation is

\[ f_1 \setminus_a f_2 = f_1 \land_a (-f_2) \]

Note that with this definition of the subtraction, the resulting object includes its boundary.
If $\alpha=1$, the above functions become

$$
\begin{align*}
f_1 \land_1 f_2 &= \min(f_1, f_2) \\
f_1 \lor_1 f_2 &= \max(f_1, f_2)
\end{align*}
$$

These R-functions are very convenient for calculations but have $C^1$ discontinuity at any point, where $f_1 = f_2$. If $\alpha=0$, the above general formulation takes the most useful in practice form:

$$
\begin{align*}
f_1 \lor_0 f_2 &= f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \\
f_1 \land_0 f_2 &= f_1 + f_2 - \sqrt{f_1^2 + f_2^2}
\end{align*}
$$ (1.1)

These functions have $C^1$ discontinuity only in points where both arguments are equal to zero. If $C^n$ continuity is to be provided, one may use another set of R-functions:

$$
\begin{align*}
f_1 \lor_m f_2 &= (f_1 + f_2 + \sqrt{f_1^2 + f_2^2})(f_1^2 + f_2^2)^{\frac{m}{2}} \\
f_1 \land_m f_2 &= (f_1 + f_2 - \sqrt{f_1^2 + f_2^2})(f_1^2 + f_2^2)^{\frac{m}{3}}
\end{align*}
$$ (1.2)

In Fig. 1.4, contour lines of the function $F(x,y)$ defining the intersection of halfplanes $x \geq 0$ and $y \geq 0$ are shown. Using the R-functions (1.1), the result of the intersection can be described as:

$$
F(x, y) = x + y - \sqrt{x^2 + y^2}
$$

This function has a curve with a sharp vertex (bold line in Fig. 1.4) as a zero value iso-contour $F(x,y)=0$. Other contour lines are smooth in the entire domain. This property
brings the idea that some displacement of the exact R-function can result in the blending effect between the lines $x = 0$ and $y = 0$.

**Figure 1.4:** Contour lines of the R-function (1.1) defining the intersection of halfplanes $x \geq 0$ and $y \geq 0$.

The following definition of the blending set-theoretic operation was proposed in [Pasko and Savchenko 1994a]:

$$F_b(f_1, f_2) = R(f_1, f_2) + disp_b(f_1, f_2), \quad (1.3)$$

where $R(f_1, f_2)$ is an R-function corresponding to the type of the operation, the arguments of the operation $f_1(X)$ and $f_2(X)$ are defining functions of two initial solids, and $disp(f_1, f_2)$ is a Gaussian-type displacement function. The following expression for the displacement function was used:
\[ \text{disp}_b (f_1, f_2) = \frac{a_0}{1 + \left( \frac{f_1}{a_1} \right)^2 + \left( \frac{f_2}{a_2} \right)^2}, \quad (1.4) \]

where \( a_0, a_1, \) an \( a_2 \) are parameters controlling the shape of the blend. The proposed definition is suitable for blending union, intersection, and difference, allows for generating added and subtracted material, symmetric and asymmetric blends. For example, the intersection operation can be described by an R-function as

\[ R_{\text{int}} (f_1, f_2) = f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \]

with the corresponding blending intersection described as

\[ F_{\text{int}} = f_1 + f_2 - \sqrt{f_1^2 + f_2^2} + \frac{a_0}{1 + \left( \frac{f_1}{a_1} \right)^2 + \left( \frac{f_2}{a_2} \right)^2} \quad (1.5) \]

The proposed displacement function does not get zero value anywhere in the space. This is the reason of the main disadvantage of this definition - the blend has global character and cannot be localized using its parameters.

Blending to the edge is one of the challenging operations. Fig. 1.5 shows blending union of a sphere (top object) to the edge produced by intersection of two other spheres (bottom
object). In the case when the bottom object is constructed using min function (Fig. 1.5b) the edge is present on the blend surface because of $C^1$ discontinuity of the min function on a plane passing through the initial edge. In the case when the bottom object is constructed using the R-function (1.1), the blend surface is smooth (Fig. 1.5c). This example clearly illustrates the necessity of using $C^1$ continuous R-functions, especially in the case of the subsequent applications of blending operations.

![Figure 1.5](image)

**Figure 1.5:** Blending to the edge: a) sphere (top object) to be blended to the intersection of two other spheres (bottom object); b) $\text{min}$ function is used for the bottom object construction; c) R-function is used for the bottom object construction.

An analytical definition of blending set-theoretic operations proposed in [Pasko and Savchenko 1994a] covers added and subtracted material blends as well as symmetric and asymmetric blending. The main problem with this approach is that the blends are global in nature and cannot be localized using the blend parameters.
1.3.3 Localization of blending

Localization of blending means that some area in space can be defined such that the blending surface exists only inside this area or parameters of blending can be chosen such that an additional displacement function for blending takes zero value outside the specified area. The local blends that control how far along a surface the blend adheres were proposed in [Hoffman and Hopcroft 1985, Rockwood 1989]. Controlled blending problem has been also addressed in implicit surface modeling research [Guy and Wyvill 1995, Wyvill and Wyvill 2000, Angelidis et al. 2002].

In the papers [Pasko G. et al. 2002a, Pasko G. et al. 2002b, Pasko G. et al. 2005a] we extended the approach of [Pasko and Savchenko 1994a] by introducing bounded blending operations defined using R-functions and displacement functions with the localized area of influence. The shape and location of the blend is defined by control points on the surfaces of two solids or by an additional bounding solid (see details in Chapter II).

1.4 Shape metamorphosis

Shape transformations in animation and free-form modeling include simple linear transformations (translation, scaling, and rotation), non-linear transformations such as free-form deformations and other non-linear space mappings, and metamorphosis or morphing (transformation of one given shape into another).
The approaches to 2D metamorphosis include physically-based methods [Sederberg et al. 1992, Sederberg et al. 1993], star-skeleton representation [Shapira et al. 1995], warping and distance field interpolation [Cohen-Or et al. 1996], wavelet-based [Yuefeng Zhang and Yu Huang 2000], and surface reconstruction methods [Surazhsky et al. 2001]. A detailed survey on 3D metamorphosis can be found in [Lazarus and Verroust 1998], 3D distance field metamorphosis in [Cohen-Or et al. 1998]. The existing approaches to metamorphosis are based on one or several of the following assumptions: equivalent topology of two given shapes (mainly topological disks or balls are considered), polygonal shape representation, shape alignment (shapes have common coordinate origin and significantly overlap in most of the case studies), possibility of shape matching (establishing of shape vertex-vertex, control points or other features correspondence), the resulting transformation should be close to the motion of an articulated figure.

The aspects of the more general shape transformation problem considered in this work are the following. The initial shapes can have arbitrary topology not corresponding to each other. No restrictions should be imposed on the input shape model, the shapes can be defined as 2D polygons, implicit surfaces, or constructive solids in 2D or 3D. It is generally should not be required that the shapes are aligned or overlap; in fact, they can occupy different positions in space. The one-to-one correspondence established between the boundary points or other shape features is also not required. A combined transformation including metamorphosis and nonlinear motion should be considered.
Implicit surfaces [Bloomenthal et al. 1997] and FRep solids [Pasko et al. 1995] seem to be most suitable for the given task. The described above general type of behavior can be obtained using skeletal implicit surfaces. Several algorithms for matching and interpolating soft objects were proposed [Wyvill et. al. 1986, Wyvill 1993]. Some of the algorithms match elements according to their position in space (cellular matching), but others require extra information about the elements (hierarchical matching). Creating empty components whenever it is necessary at the pre-processing step guarantees that both initial and final shapes have the same number of components bijectively matched by the correspondence process. The intermediate implicit surface is generated in the process of interpolation of the main varying parameters: skeletons position and field intensity. The method has such major limitations as the following: it may not match components of different skeleton type, and components must be bijectively paired.

A more general metamorphosis technique for soft objects based on different skeletons was proposed in [Galin et al. 2000]. Skeletons may be convex polygonal shapes of arbitrary dimension (points, line segments, convex polygons or convex polyhedrons). It is assumed that the animator provides a first rough graph of correspondence that matches parts of the initial and the final shapes. Since this correspondence is in general non-bijective, each component is subdivided into sub-components for creating a new graph bijectively matching those sub-components. An intermediate sub-component is associated with each graph link. The transformation of initial and final sub-components is achieved by transforming their skeletons, distance functions, and potential functions.
Metamorphosis of arbitrary FRep objects can be described using the linear (for two initial shapes) [Pasko et al. 1995] or bilinear (for four initial solids) function interpolation [Fausett et al. 2000], but it can produce poor results for not aligned objects with different topology. Turk and O’Brien [1999] proposed a more sophisticated approach based on interpolation of surface points (with assigned time coordinates) using radial bases functions (RBF) in 4D space. This method is applicable to not aligned surfaces with different topology. However, for the initially given implicit surfaces this requires time consuming surface sampling and interpolation steps.

In [Pasko G. et al. 2003a, Pasko G. et al. 2004a, Pasko G. et al. 2004b], a new method of shape metamorphosis combined with non-linear motion was developed. It is based on increasing the object dimension, function-based bounded blending, and consecutive cross-sectioning for animation (see details in Chapter II).

1.5 Modeling heterogeneous objects

We describe below the problem of modeling objects of mixed dimensions and the approach based on combination of cellular and functional representations. A specific problem of modeling two-dimensional implicitly defined cells is considered.
1.5.1 Dimensional heterogeneity

In section 1.1, we discussed the abstraction hierarchy of models and typical geometric level models of homogeneous solids (n-dimensional point sets in n-dimensional space). In many practical cases, it is quite useful to construct a heterogeneous object model. Heterogeneous objects have internal structures with non-uniform distribution of material and other attributes of an arbitrary nature (photometric, physical, statistical, etc.) along with elements of different dimensions (k-dimensional point sets in n-dimensional space with \( k \leq n \)). It means these objects are heterogeneous from the internal structure and dimensionality points of view. A model of objects with fixed dimensionality and heterogeneous internal structure (multidimensional point sets with multiple attributes or so-called constructive hypervolumes) was presented in [Pasko et al. 2001] and is out of scope of our work. We will concentrate on dimensionally heterogeneous objects consisting of elements of different dimensions. For example, such a model in 3D space can include solids, surfaces, curves, and points appropriately attached to each other. Models of dimensionally heterogeneous objects belong to the cellular structured space level of the abstraction hierarchy (section 1.1).

Research works on modeling dimensionally heterogeneous objects usually exploit one of the cellular object representations and their construction methods. Let us give a brief survey of these models on the base of [Kunii 1999, Adzhiev et al. 2002]. Dimension-independent geometric modeling for various applications is based in [Paoluzzi et al. 1993] on multidimensional simplicial complexes. A representation of \( s \)-sets, i.e. finite
aggregates of disjoint, open regularized cells, was proposed in [Arbab 1990] especially s-sets are proved to be suitable for modeling assemblies. Selective geometric complexes (SGC) introduced in [Rossignac and O’Connor 1990] are non-regularized inhomogeneous point set represented through enumeration as the union of mutually disjoint connected open subsets of real algebraic varieties. The SGC model provides a basis for representing objects of heterogeneous dimensionality possibly incomplete boundaries. The Djinn solid modeling Applications Program Interface (API) [Armstrong et al. 2000] is based on cellulary partitioned objects containing mutually disjoint cells which are manifold point-sets of differing dimensionality in 3D space. An extension of B-splines to surfaces of arbitrary topology is proposed in [Grimm and Hughes 1995], where polyhedral complexes are used to describe the surface topology and to locate control points which are necessary for definition of the basis functions of B-splines. CW-complexes are used in [Hart 1999] to represent the topological structure of an implicit surface. The CW-complexes form a topological skeleton of the objects describing the configuration of critical points. A design procedure for cellular models based on CW-complexes with the emphasis on topology validity of resulting shapes is presented along with the object-oriented implementation in [Kunii 1999, Ohmori and Kunii 2001].

1.5.2 Cellular-functional model

In section 1.2, we described the function representation (FRep), which suits not only for representing solids, but also low dimensional entities required for heterogeneous object
modeling in 3D space. The main idea of using FRep to represent $k$-dimensional objects in $nD$ space with $k < n$, is that the function $f$ takes zero value only at the points of this object and negative everywhere else in space. For example, if one needs to describe a straight line segment in 2D space, an equation of a straight line can be used: $f(x,y) = ax+by+c$. The inequality $f \geq 0$ defines a halfplane. Then, $-f^2 \geq 0$ defines the line itself, where in fact the function $-f^2$ is never positive and only becomes zero at the points on the line. The line can be trimmed using some 2D solid to produce one or several segments. The simplest way to define a segment is $-f^2 \land_0 g \geq 0$, where $g(x,y) \geq 0$ is a definition of a 2D disk, and the symbol `$\land_0$’ stands for set-theoretic intersection defined by an R-function (see section 1.3.2).

The geometric domain of FRep in 3D space includes 3D solids with so-called “non-manifold” boundaries and lower dimensional entities (surfaces, curves, points) defined by zero value of the function. The lower dimensional entities in 3D space can be defined as follows:

- definition of a surface patch requires a 3D implicit surface and a trimming 3D solid or a 2D patch and mapping to 3D space;
- a curve can be defined as the intersection of two surfaces;
- a point can be defined as the intersection of three surfaces, a curve and a surface, or directly by a non-positive distance function to the given point.
It is shown in [Rvachev et al. 2001] that such a function representation of dimensionally heterogeneous objects is very useful in solving interpolation and boundary value problems.

As one can see from above, entities of different dimensions can be represented within FRep. However, a mechanism of combining them in a single object (besides the set-theoretic union) is missing in this representation. Moreover, different applications such as CAD or finite-element analysis require an explicit representation of mixed-dimensional objects along with the functional one. These were the main motivations for the introduction of a new hybrid cellular-functional model in [Adzhiev et al. 2002]. The hybrid model combines a cellular representation and the functional representation. This model allows for independent but unifying representation of geometry and attributes (for representing internal material and other properties), and makes it possible to represent dimensionally non-homogeneous entities and their cellular decompositions. The hybrid model is based on the notion of an implicit complex. Such a complex is defined as a union of properly joined cellular domains with subspaces shared by two or more domains represented by explicitly defined cellular sub-complexes or by low dimensional domains functionally defined using FRep as described above.

**1.5.3 Trimming and polygonization implicit surfaces**

Dimensionally heterogeneous objects include elements of different dimension (points, curves, surfaces, solids) combined into a single entity from the geometric point of view
(point set) and the topological point of view (cellular complex) [Kunii 1999]. The recently introduced cellular-functional model [Adzhiev et al. 2002] allows for representing a heterogeneous object as a cellular complex with explicit cells (BRep, parametric curves, wireframes, point lists) and implicit cells (implicit surface patches, intersections of implicit surfaces). An implicit surface patch can be defined in different ways, for example, as an implicit surface trimmed by an intersecting solid [Pasko and Pasko, 2004]. Some applications such as physical simulation based on finite-element meshes also require conversion of an implicit complex to the pure cellular representation. Such a conversion of a trimmed implicit surface to a polygonal mesh is a subject of our research. Several related works are discussed below as well as the basic polygonization algorithm used in our approach to trimming.

It was proposed in [Rossignac 1996] to use CSG (Constructive Solid Geometry) solids for trimming of boundary faces of CSG models to isolate, in a primitive’s face, its contribution to the boundary of the solid. Such a formulation was shown to present considerable computational advantages over trimming curves commonly used in boundary representations and to have applications in rendering and point-membership classification. In our work, we pursue this idea to model implicit surface sheets and stripes.

To polygonize a boundary of a CSG solid, a uniform subdivision algorithm is applied to the object surface in [Wyvill and Overveld 1997], then the mesh enhanced with an iterative numerical procedure taking into account intersection curves of primitive surfaces.
In general, the problem we deal with is related to the intersection of two implicit surfaces. There are two main numerical approaches to such implicit surface-surface intersection:

- Start with some intersection point found analytically or numerically. Trace the intersection curve by solving a differential equation (see, for example, [Hosaka 1992]).
- Approximate both surfaces by polygons and intersect two obtained polyhedrons.

A surface sheet or a stripe can be mathematically defined as a two-dimensional manifold with boundary (or simply a 2-manifold). The problem solved by a polygonization algorithm is to approximate a given surface by a set of polygons. In [Bloomenthal and Ferguson 1995], a general polygonization algorithm for non-manifold surfaces was proposed. The algorithm can polygonize both 2-manifolds with boundaries and non-manifold surfaces. Because of the algorithm complexity, it is difficult to implement its adaptive version. A simpler algorithm processing only 2-manifolds with boundaries can be obtained by extending conventional polygonization algorithms.

Let us briefly describe the basic implicit surface polygonization algorithm proposed in [Pasko et al. 1986, Pasko et al. 1988] and used in our work on trimming implicit surfaces. The algorithm is based on the uniform bounding box subdivision, trilinear interpolation inside an elementary cell, using hyperbolic arcs for resolving topological ambiguities, and edge connectivity graph tracing for generation of polygons (see Fig. 1.6). A bounding box (parallelepiped) is given in three-dimensional Euclidean space and
supposed to contain the implicit surface. By introducing a regular point grid we represent
the box as a set of elementary cells. The polygonal approximation of the surface can be
obtained by visiting each cell and approximation of surface patches belonging to a cell.
In the general case a cell can contain more than one surface patch (Fig. 1.6a). Using
conventional enumeration of cell edges, the edges connectivity graph can be constructed.
Surface patches correspond to simple cycles of this graph (Fig. 1.6b).

The algorithm of a cell processing follows:

1) evaluation of function values at eight cell vertices;
2) search for intersection points with the surface for edges with different function signs
   in edge vertices;
3) connectivity graph construction;
4) graph tracing for revealing cycles and patches construction.

The intersection point of an edge and the surface can be found by the linear interpolation
of function values in vertices or by a more sophisticated root search algorithm on the
edge. The procedure of the graph construction involves independent inspection of six cell
faces and listing of graph branches between corresponding nodes. In the most difficult
case all four edges of a face intersect the surface (Fig. 1.6c).

The bilinear interpolation of the function on the cell face produces hyperbolic arcs on the
face plane. Calculation of coordinates of the hyperbola centre provides unambiguous
choice of the intersection points connection by two hyperbolic arcs. To obtain the polygonized surface, line segments replace the hyperbolic arcs.

The procedure of graph tracing involves:

1) search for a graph node \( i \) connected with another node \( j \) by the branch \( E_{ij} \);
2) transition to the node \( j \) with the branch \( E_{ij} \) removal;
3) transition to the node \( k \) by the branch \( E_{jk} \) with removal of latter.

Steps 2-3 are repeated until a cycle under consideration is closed and then step 1 is executed again. Additional branches can be introduced in the graph to produce quadrilateral or triangular grid. This procedure stops when the graph is empty. The algorithm produces a set of polygons approximating the initial surface. This algorithm is free of heuristics and ambiguities essential to other algorithms of this kind.

We propose a model of a 2-manifold with boundary in section 2.3 and an algorithm of trimming implicit surfaces in section 3.4, and apply it to cultural objects modeling in section 4.4.
Figure 1.6: Basic polygonization algorithm:

a) example of surface patches inside a cell with hyperbolic arc boundaries on the cell faces;

b) cycles in the connectivity graph corresponding to surface patches;

c) case of four intersection points of cell face edges with the isosurface.
1.6 Cultural preservation

In this section we discuss the potential application of proposed models and algorithms in the digital preservation of culturally valuable shapes. Purposes, different approaches and problems of cultural preservation are discussed in detail. On the base of this discussion we select an appropriate shape model for cultural preservation.

1.6.1 Cultural heritage terms and notions

*Culture* is defined in the Encyclopedia Britannica as: "the integrated pattern of human knowledge, belief, and behavior."

*Cultural heritage* is considered as material objects through which cultural knowledge is transmitted or inherited.

The *Roerich Pact* was the first international agreement focusing exclusively on the protection of cultural heritage, agreed to by all the members of the Pan-American Union in 1935. The *Roerich Pact* and the *Banner of Peace* movement grew to the *Convention for the Protection of Cultural Property in the Event of Armed Conflict*, which was adopted by UNESCO at Hague (Netherlands) in 1954. It covers immovables and movables, including monuments of architecture, art or history, archaeological sites, works of art, manuscripts, books and other objects of artistic, historical or archaeological interest, as well as scientific collections of all kinds.
UNESCO World Heritage Convention of 1972 classifies cultural heritage into the following categories: monuments (architectural works, works of monumental sculpture and painting, elements or structures of an archaeological nature and others); groups of buildings; sites (works of man or the combined works of nature and man, and areas including archaeological sites) which are of outstanding universal value from the historical, aesthetic, ethnological or anthropological point of view.

1.6.2 Digital preservation purposes

The term digital preservation means the capture and archiving of the form and contents of the existing cultural objects through the use of computer modeling techniques and the reproduction of those objects that have already been lost. The digital preservation is a particularly important issue, since cultural objects may be easily demolished, as the recent destruction of the Bamiyan Buddha-images in Afghanistan, and large number of cultural monuments in Iraq has powerfully demonstrated. “List of World Heritage in Danger” by UNESCO includes currently about 35 objects of cultural heritage threatened by serious and specific dangers from total 788 properties with 611 cultural, 154 natural and 23 mixed properties in 134 States [UWHLD 2005].

The cultural preservation has attracted considerable attention in computer graphics, geometric modeling, and virtual reality communities. One can mention such conferences as EUROGRAPHICS’99 with the topic "Bringing to new life our Cultural Heritage", International Cultural Heritage Informatics Meeting (ichim), Museums and the Web,
special sessions on Virtual Heritage during the Virtual Systems and Multimedia Conference (VSMM 2000-2004); Virtual Reality, Archaeology and Cultural Heritage Conference (VAST 2001-2004); special session on heritage applications at the International Conference on 3D Digital Imaging and Modeling (2003, 2005); International Digital Silk Road Symposium UNESCO (2003); special issue on virtual heritage of IEEE Multimedia journal (April-June 2000), special issue on computer graphics in art history and archaeology of IEEE Computer Graphics and Applications (2002), and several large projects discussed below.

Digital tools and techniques may be used for documentation, representation, and dissemination of cultural heritage in the following ways [Addison 2000, Vilbrandt et al. 2001]:

1) Digitizing text and images from existing documents;

2) Digitizing 3D objects using cameras, stereo imagery, panoramic video, 3D laser and other types of scanners [Addison and Gaiani, 2000];

3) Construction of digital models of lost cultural artifacts such as paintings, statues or temples in digital form using existing documents (photographs, drafts, written evidence) or archaeological findings;

4) Reverse engineering and digital representation of the shape and texture of existing three-dimensional physical objects (sculptures, buildings, natural environments, etc.) based on measurements and 3D scanning;

5) Archiving digital representations of reconstructed and reverse engineered objects;

6) Using digitized and reconstructed images and models for presenting cultural heritage
as virtual objects, animations, multimedia documents, Internet sites and virtual tours [VHN, Maya 2003], gaming environments, for example, as in Virtual Notre Dame project [DeLeon and Berry 2000, VND], augmented reality systems [TimeScope], or live site shows as 80-meter projections of heritage objects on the mountains in Salzburg in Austria [Arch].

1.6.3 Approaches to shape preservation

As mentioned above, a shape is considered a point set in a multi-dimensional space. Thus, not only external boundaries, but also internal structures of objects as well as their time and other parametric dependencies can be subjects of digital preservation. In this section, we classify and compare the following different approaches to computer-aided preservation of culturally valuable objects [Vilbrandt et al. 2001, Vilbrandt et al. 2004]:

1) Measurements and drafting
2) Measurements and modeling
3) Scanning and archiving
4) Scanning and meshing
5) Scanning and modeling
Measurements and drafting

Existing shapes are traditionally documented by measuring them and drafting 2D representations of them. Tradition measurement methods range from the steel tape or ruler to the optical theodolite. These methods are cheap, simple and reliable, but they are very slow, labor intensive, do not provide high accuracy, and usually require physically carrying a target or probe to each point to be measured [Addison and Gaiani, 2000]. The main disadvantage is that these methods require touching of the measured object. Hand drafting has been usually applied for documenting the object shape. Computer-aided 2D drafting methods may be currently used, and logically extended to the "measurements and modeling" paradigm described below.

Measurements and modeling

In this approach traditional methods of measurement are used similarly to the previous one. 3D computer models are created rather than 2D drafts, and designer operated interactive 3D modeling rather than automatic reconstruction from input data is used here. The benefits of using 3D graphics techniques in constructing models are: models can be manipulated to provide multiple viewpoints; rotating a model can serve for a better understanding of the physical relationships of the components of the actual structure, as well as the construction techniques involved; 3D models can replicate the actual construction of the original object itself, including features normally hidden to the eye, such as interior bracketing, and the model can be deconstructed to reveal such hidden
features. This approach is especially valuable if the real object has been lost, destroyed or damaged, and is documented only by previous measurements, drawings, photographs and drafts. The goals are to create a 3D model of the object that is as complete as possible, and to represent its internal structure, design logic (showing how components are interconnected or layered), and history of the shape construction, as well as time-dependent aspects and other parametric dependencies.

*De facto* standard industrial modeling tools are usually based on the boundary representation (BRep) of 3D objects (see section 1.2.1). In particular, BRep can be based on a polygonal mesh approximation of the object surface. This modeling scheme is only partly appropriate for achieving the modeling goals described above. BRep data structures do not reflect the object's internal structures (e.g., material distribution) or design logic. Parameterization of BRep models is quite limited. Only simple time-dependent parameterization of BRep is allowed, which does not change the object topology. A constructive modeling approach can be an alternative. It is based on the construction of complex objects using simple primitive elements, combining and transformation operations. The Constructive Solid Geometry (CSG) and the Function Representation (FRep) methods discussed in the section 1.2 support this approach.

Let us consider several examples of cultural preservation modeling based on BRep and CSG using traditional measurements and old documents. In the project on constructing the historic villages of Shirakawa-go [Hirayu et al. 2000] incoming information was presented as drawings, architectural blueprints, photographs and aerial photographs. The
entire village, exterior and interior of individual buildings were modelled using BRep (polygonal meshes), supported by traditional software tools such as Open Inventor and AutoCAD. The authors note the large size of data files and necessity of applying special methods of mesh-reduction for virtual reality purposes.

In the Great Buddha Project [Miyazaki at el. 2000, Ikeuchi and Sato 2001], the Main Hall of the Great Buddha of Kamakura was modelled using BRep-based CAD software on the base of drawings of a similar hall in the Todai-ji temple. The authors also mention the large model size (300,000 polygons).

Another approach to modeling destroyed temples was taken in the Aizu History Project [Vilbrandt et al. 1999, Vilbrandt et al. 2001]. The method, which was used to model buildings from both archaeological data and on-site measurements, reveals how the actual objects were constructed, rather than yielding only the visible surface models produced by the BRep-based approaches. The authors specified CSG as the most likely shape representation for modeling historical architecture with any possibility of archival quality. The Golden Hall at Enichiji temple and the Sazaedô temple were created whenever possible with only CSG based entities. Enichiji was the religious center of the Heian period (794-1185); no buildings or images from that period are extant today. In order to produce a model of the Heian Golden Hall, the authors relied on data introduced in archaeological site reports. In addition to archaeological data, the model was based on standard temple-building practices of the 8-th and 9-th centuries. The currently existing Sazaedô pagoda was modeled using engineering blueprints and supplementing them with
measurements taken on site. The CSG models were created using AutoCAD (Release 12), the last version of this system supporting CSG. Later the Enichiji model was used to create a virtual environment that users can experience in real-time [Calef et al. 2002]. In order to provide an immersive environment, the authors were forced to convert the original CSG model to BRep one supported by the Quake gaming engine.

An example of FRep constructive modeling of existing modern sculptures with further use of them in a virtual reality type application can be found in [Adzhiev et al. 2003, Kazakov at el. 2003]. Starting from measurement of physical sculptures (by Russian artist Igor Seleznev), they produce FRep models of them in the special language HyperFun. Then, the created models were used for making several animations and interactive application with immersive interface [Kazakov at el. 2003]. The user can navigate in the scene containing several sculpture models, perform interactive metamorphosis between these models to obtain a new intermediate shape, and apply sculpting operations to it by removing or adding material. The function representation presents the interactive system user with the opportunity to envision, discover and create novel sculptural forms without imposing many restrictions.

**Scanning and archiving**

There exist several well-developed technologies for automatic non-contact acquisition of 3D point coordinates on the visible surfaces of objects [Addison and Gaiani 2000, Bernardini and Rushmeier 2002]. These technologies are based on lasers, structured light,
sound, and stereo imagery. Archiving of the raw data (the measured point locations) is preferable in any case to archiving shapes inferred from this data. Moreover, the raw data itself can be the best way of actually representing the surface, as was shown in the Digital Michelangelo project [Levoy et al. 2000]. The authors used the combination of laser range scanning and high-resolution digital color imaging to acquire models of many of the major sculptures of Michelangelo. Instead of generation of polygonal meshes, they proposed to archive original range images produced by the scanning equipment, which made the data usable in practice. The datasets of range images obtained with laser range finders provided 18:1 storage savings with no loss in information, if compared with the equivalent polygonal mesh. A special viewer based on range images was developed. The project authors claim, “If one only wants to view a 3D model, and not perform geometric operations on it, then it need not be represented polygonally.”

**Scanning and meshing**

The most widely accepted approach is to produce a polygonal mesh based on the 3D scans, photographs, and other input data. The surface mesh generation can be necessary especially if the measurement equipment does not provide point coordinates directly. For example, in the Pietà Project [Abouaf 1999] the scanner consisted of six black-and-white cameras capturing images of a striped pattern projected on an object. Accompanying software computed a triangle mesh from the captured images using principles of stereo computer vision.
If 3D point coordinates are available from the scanning process in the form of one or several data sets, the step of scan integration is performed [Bernardini and Rushmeier 2002] for the reconstruction of the scanned objects geometry and topology. There are factors which make the task quite complicate: point coordinates are not exact because of measurement errors, some part of the scanned surface can be missed or density of point can be not enough to reconstruct small details, some of the points can have large random deviation.

The methods of the surface mesh reconstruction are usually classified as follows: Delaunay-based methods, surface-based methods, volumetric methods, deformable surfaces, and methods of interpolation using radial-basis functions.

Delaunay-based methods such as the power crust algorithm [Amenta et al. 2001] use only a point cloud as input and a Delaunay complex, which decomposes the convex hull of the set of points and introduces connectivity structure between the points. Then, the surface is reconstructed extracting a sub-complex from the Delaunay complex. These algorithms interpolate the data points so they are very sensitive to noise and points with large deviation.

The zippering approach of Turk and Levoy [Turk and Levoy 1994] is an example of the surface-based methods. Connecting all neighboring points by local operations follows the local parameterization of the surface. Partial connectivity essential to the range images can be used in some of these methods. It is possible to process large datasets by surface-based methods, but it is difficult to represent correct topology in the case of measurement
errors compared with distances between points and size of topological futures (holes, handles and others).

Volumetric methods employ a voxel array, which is a 3D matrix of some scalar field values sampled in the nodes of a 3D grid enclosing the data. In the simplest case a binary voxel array can be used as in [Neugebauer 1997], where the scanned range images are used to carve out an intermediate volumetric model of the object. This step is based on the fact that no part of the object can lie between the measured range surface and the camera of the scanner. Carving is performed by applying the visibility test to each node of the volume. A node does not belong to the object if it is visible from one of the cameras. The intermediate volume describes the topology of the object and approximates its shape within the desired level of detail. Then, an isosurface with given threshold is extracted using some polygonization method such as marching cubes [Lorensen and Cline 1987]. A signed distance or any other real-valued scalar field can be sampled in the nodes of a regular grid. For example, in the algorithm of [Curless and Levoy 1996], to compute a distance value in the node a ray is cast from the sensor through this node to the range surface. The length of the ray segment from the node to the ray-surface intersection point is calculated and combined with distance values for other range surfaces with weights dependent on surface normal and direction to the sensor. Similar approach was applied in the Great Buddha Project [Miyazaki at el. 2000, Ikeuchi and Sato 2001], but the authors utilized octrees to represent volumetric data obtained from range images with reducing memory requirements and without loosing accuracy of the resulting isosurface. Volumetric methods can work with very large datasets. Range surfaces are processed one
by one independently. The required memory and processing speed depend on the resolution of the rectangular 3D grid.

Algorithms of deformable surfaces are based on the approximation of the initial data set by some simple surface and its dynamic modification under influence of external forces and internal reactions and constraints. In [Pentland and Sclaroff 1991], virtual “springs” are attached to each data point and a point on the initial surface. Then, the surface is deformed using the method of finite elements to finally approximate the point cloud.

Methods of the point cloud interpolation using radial-basis functions combine generality of volumetric methods and flexibility of deformable surfaces. A continuous spline function of three variables is constructed to interpolate given scalar values in the scanned points. In [Savchenko at el. 1995], a special carrier function is introduced and evaluated in the data points. A zero-valued isosurface of the difference between the volume spline and carrier function passes through all scanned points. A similar approach was taken in [Carr at el. 2001], but instead of using a carrier function, two additional point clouds were generated inside and outside the initial cloud. These two clouds are assigned positive and negative scalar values, while zero values are assigned to the data points to ensure stable behavior of the volume spline in the proximity of the initial point cloud. These methods can process very large datasets, maintaining the property of the resulting implicit surface to pass through all data points and to fill holes missed during scanning. The final implicit surface is then polygonized to produce a surface mesh.
**Scanning and modelling**

Scanning can provide a set of reference control points for manual modeling or the full point cloud can be used for (semi-) automatic model generation. Voxel models generated from a set of range images [Neugebauer 1997] and implicit surfaces defined using radial-basis functions [Savchenko at el. 1995, Carr at el. 2001] or multi-level partition of unity [Ohtake et al. 2003] can be considered as models providing point membership classification in contrast to meshes. The potential of an automatic search of a simple model structure and parameters fitting of an implicit surface model on the base of range data was illustrated in the work [Muraki 1991]. The blobby model, originally proposed in [Blinn 1982] by analogy with Gaussian approximations of atomic electron density functions for computing images of molecular surfaces, represents the object as an isosurface of a scalar field produced by a set of field generating primitives. The algebraic summation of the individual primitive fields results in a complicated scalar field and its isosurfaces. To construct the blobby model, the number and positions of the primitives should be found to minimize the distance function from the surface to the range data points. The algorithm starts with a single primitive and introduces more primitives by splitting each primitive into two new ones. As the number of primitives is increasing, the model gradually comes to approximate the input range data.

In the case of unknown initial estimation of the model structure, evolution of shapes using techniques such as genetic algorithms can be applied, in the manner of the reported experiments with CSG [Todd and Latham 1992] and analytically defined implicit
surfaces [Bedwell and Ebert 1999]. Here, the overall distance from the shape surface to
the scanned points can serve as an optimization criterion.

1.6.4 Problems of cultural preservation

As we mentioned in [Vilbrandt et al. 2001, Vilbrandt et al. 2004], loss of information is
one of the biggest problems because of using non-standardized (often proprietary) and
frequently changing formats for the archiving of the data and models. Other problems
caused by the mentioned above reasons are difficulties of data exchange between
different systems and across platforms, lack of understanding of modeling processes and
data structures by users, and therefore limited possibilities to verify the application's
operations independently. Methods and procedures must be open to inspection and
inquiry in order to assure that cultural artifacts have been accurately modeled, and it
should be possible to perform independent verifiable evaluation of the results of the given
study. A visible solution of these problems is the use of open, standard, and well-defined
data formats, development and utilization of open-source software. Such an approach can
contribute to the production of secure and long-lasting digital archives for cultural
heritage preservation.

Current shape modeling systems and cultural heritage preservation systems are
traditionally based on polygonal meshes and other boundary representation models.
These models have such disadvantages as lack of construction history and of the
constructive object structure, accumulation of numerical errors resulting in surface cracks,
high complexity of processing algorithms (see section 1.2.1).

Geometric modeling procedures and the fundamental mathematical base for 3D shape modeling, volume rendering, and multidimensional modeling are not well known in the digital archiving community. Basic geometric modeling procedures, the preservation of original data attached to these procedures and of the order of constructive events, and the modeling of dynamic attribute of 3D models are core issues in the development of digital archives.

1.6.5 Shape model selection for cultural preservation

This work is based on the general abstraction hierarchy of shape models discussed in section 1.1. From this hierarchy point of view, the work is concentrated on geometric level and cellular structured space level. In this section, we make selection of models for the geometry and cellular structured levels from the cultural preservation point of view.

The basic mathematical representation of geometry in digital preservation has to serve several purposes [Vilbrandt et al. 2001, Vilbrandt et al. 2004]:

1) Representation of the logic of the object's construction. Information about the logical structure can be used for analysis, classification, and for reuse of the model in the educational and entertainment purposes.
2) Support modeling of parametric families of shapes. A parameterized model can be used for the parameters estimation from the given data of the specific instance of the object (for example, from the scanned surface points) and hence for the practical analysis of the existing cultural items.

3) Support specific and extensible sets of modeling operations and primitives. Shapes of the cultural items are usually very complicated and quite often cannot be modeled with existing tools. Thus, extensibility of the modeling system is an essential requirement.

4) Allow for generation of polygonal and other surface models, as well as voxelization for visualization, animation and virtual objects presentation on the Web.

5) Serve for direct control of rapid prototyping machines with the precision needed to reproduce the modeled objects.

We propose to use the function representation FRep (described in section 1.2.3) as our basic mathematical model [Pasko et al. 1995]. Let us summarize advantages of using FRep in cultural preservation. It satisfies all above requirements, first of all, by providing precise mathematical definition of modeled objects. The clearly defined open text format of FRep models in HyperFun language obviously increases the survival period of the archived models, especially in comparing with not openly specified proprietary formats. The combination of geometry and attributes in the constructive hypervolume model [Pasko et al. 2001] allows for modeling not only the shapes, but material and other object properties. Support of multidimensional and particularly time-dependent models in FRep
is useful for the modeling aging objects with the help of finite element analysis [Kartasheva et al. 2003]. The recovered structure and constructive elements of the object can be employed for rebuilding lost objects and for making "live heritage" applications such as animation or interactive multimedia.

From the heterogeneous objects modeling point of view, FRep is extended by the implicit complex concept [Adzhiev et al. 2002] described in the section 1.5. In this work, we concentrate on modeling 2D cells for implicit complexes, namely using trimming of implicit surfaces.

The main disadvantage of FRep is time-consuming function evaluation, which currently retards interactive applications. However, the compact HyperFun format suits well parallelization, distributed and mobile computing, which can be one of solutions to this problem. Another difficulty is that constructive modeling usually requires high levels of 3D modeling skill and is labor-intensive. One of the possibilities of automation is fitting of a parameterized FRep model to a cloud of surface points [Vilbrandt et al. 2004].
Chapter II  Shape models and operations

2.1 Bounded blending

Blending operations based on R-functions and the problems of blend localization were presented in the section 1.3. New formulations for blending operations with the blend bounded by control points and by an additional bounding solid are described in this section [Pasko G. et al. 2002a, Pasko G. et al. 2002b, Pasko G. et al. 2005a]. The definition (1.3) of a blending operation in section 1.3.2 is used here as the basis with the modifications introduced only for the displacement function. First, we select a displacement function with the local area of influence (section 2.1.1), then, we construct a generalized distance function for bounding blends using control points (2.1.2) and bounding solids (2.1.3), and discuss the properties of bounded blending (2.1.4).

2.1.1 Displacement function

As the first step of defining bounded blending, we propose to select for the definition (1.3) a displacement function of one variable $disp_{bb}(r)$, which should satisfy the following conditions:

1) $disp_{bb}(r) \geq 0$ and $disp_{bb}(r)$ takes the maximal value for $r=0$;

2) $disp_{bb}(r) = 0, r \geq 1$

3) $\frac{\partial disp_{bb}}{\partial r} = 0, r = 1$ (the curve tangentially approaches axis $r$ at $r=1$).
A plot of the desired displacement function is shown in Fig. 2.1. One can derive or find several analytical expressions for such function [Pasko G. et al. 2005a, Pasko G. et al. 2005b]. In fact, this work has been done by many researchers for localizing individual components of skeletal implicit surfaces [Bloomenthal et al. 1997]. One of the possible expressions is

\[ disp_{bb}(r) = \begin{cases} 
\frac{(1-r^2)^3}{1+r^2}, & r < 1 \\
0, & r \geq 1
\end{cases} \quad (2.1) \]

where \( r \) is a generalized distance, which is constructed using defining functions of the initial solids, as it will be shown in the next section. The polynomial function of lower degree also can be taken into consideration as an appropriate expression for the generalized distance [Pasko G. et al. 2005b]:

\[ disp_{bb}(r) = 2r^3 - 3r^2 + 1. \]
2.1.2 Bounding by control points

We discuss here the first and possibly the simplest method of the blend localization. The idea is to place one control point on the surface of each initial solid and to require that the blend exists only “between” these control points. The control points are supposed to be relatively close to the edge resulting from the pure set-theoretic operation and it should be intuitively obvious what area is designated for blending. To satisfy such requirement, the displacement function (2.1) should become zero at the control points with the maximal value on or near the edge. Let us suppose that the initial defining functions \( f_1(X) \) and \( f_2(X) \) have distance properties. Then, the following generalized distance function can be proposed:

\[
\begin{align*}
\alpha f^2 + \beta f_2^2 &= \left( \frac{f_1}{a_1} \right)^2 + \left( \frac{f_2}{a_2} \right)^2, \quad (2.2)
\end{align*}
\]

where \( f_1(X) \) and \( f_2(X) \) are defining functions of two solids as arguments of the blending operation; \( a_1 = f_1(P_2) \), where \( P_2 \) is a control point placed on the surface \( f_2(X) = 0 \); \( a_2 = f_2(P_1) \), where \( P_1 \) is a control point placed on the surface \( f_1(X) = 0 \). Note the following properties of this function:

1) \( r^2(0,0) = 0 \);

2) at the point \( P_1: f_1(P_1) = 0, f_2(P_1) = a_2, \) and \( r^2(f_1, f_2) = 1 \);

3) at the point \( P_2: f_1(P_2) = a_1, f_2(P_2) = 0, \) and \( r^2(f_1, f_2) = 1 \).
These properties mean that the displacement function $\text{disp}_{bb}(r)$ (2.1) takes maximal value at the intersection points of two surfaces, where $f_1(X) = 0$ and $f_2(X) = 0$, and becomes zero at control points $P_1$ and $P_2$. Using Eq. 1.3 and applying the displacement function $\text{disp}_{bb}(r)$ defined by Eq. 2.1 with $r^2$ defined by Eq. 2.2, we get for the bounded blending operations with control points:

$$F_{bb}(f_1, f_2) = R(f_1, f_2) + a_0 \text{disp}_{bb}(r) \quad (2.3)$$

Examples of such bounded blending operations are given in Figs. 2.2-2.3. In Fig. 2.2, we illustrate the bounded blending intersection of two halfplanes $f_1(x, y) \geq 0$ and $f_2(x, y) \geq 0$, where $f_1(x, y) = x$ and $f_2(x, y) = y$. The control point $P_1$ is placed on the line $x = 0$, the boundary of the first halfplane, and the control point $P_2$ is placed on the line $y = 0$, the boundary of the second halfplane. One can observe that the blend is located between the control points due to the displacement function definition. In Fig. 2.2a, the positive values of the parameter $a_0$ correspond to the added material blends, $a_0 = 0$ keeps the result of the pure intersection, and the negative values of $a_0$ correspond to subtracted material blends. Moving control points along the initial boundaries allows for generating symmetric and asymmetric blends as it is illustrated by Fig. 2.2b.

The bounded blending union operation of two rectangles with control points is illustrated by Fig. 2.3.
Figure 2.2: Bounded blending intersection of halfplanes $x \geq 0$ and $y \geq 0$ with control points $P_1$ and $P_2$: 

a) added and subtracted material blends controlled by the parameter $a$; 

b) symmetric and asymmetric blends with different positions of control points.

Figure 2.3: Union of two rectangles and two control points (a); expected bounded blend between control points at the upper left part of the shape (b) and unwanted blend at the lower right part of the shape (b).
Two control points are placed on the corresponding boundaries of the rectangles (Fig. 2.3a). The expected bounded blend appears between the control points at the upper left part of the shape in Fig. 2.3b. However, the rectangles are blended also in the lower right part of the shape, which can be unexpected or unwanted. This example unveils the global character of the blend with control points, which does not allow a designer to select a single vertex or an edge for blending. Some different means of defining bounds are needed to provide truly local character of blending.

2.1.3 Bounding solid

Figure 2.4: Components of the definition of the bounded blending union. Upper part: two solids to be blended \( (f_1 \text{ and } f_2) \) and a bounding solid \( f_3 \). Lower part: behavior of the functions \( r^2 \) and \( \text{disp} \) in the cross-section A of the bounding solid.
The introduction of an additional bounding solid can bring real local character to the blend. It is required that the blending surface exists only inside the bounding solid, and only original surfaces exist outside the bounding solid.

The upper part of Fig. 2.4 shows two solids to be blended (defined by the functions $f_1$ and $f_2$) and a bounding solid (defined by the function $f_3$). To apply the bounded blending definition

$$F_{bb}(f_1, f_2, f_3) = R(f_1, f_2) + a_0 \text{disp}_{bb}(r)$$  

(2.4)

with the displacement function (2.1), we have to provide the following interconnected properties (see lower part of Fig. 2.4):

1) The zero value of the displacement function $\text{disp}_{bb}(r)$ outside the bounding solid;

2) The maximal value of the displacement function $\text{disp}_{bb}(r)$ near the intersection points of two initial surfaces;

3) The zero value of the generalized distance function $r$ at the intersection points of two initial surfaces;

4) The value 1 of the function $r$ outside the bounding solid.

The following definitions satisfy the above requirements:
\[
\text{disp}_{bb}(r) = \begin{cases} 
(1-r^2)^3, & r < 1 \\
1/r^2, & r \geq 1
\end{cases}
\]

with \( r^2 = \frac{r_1^2}{r_1^2 + r_2^2}, r_2 > 0 \), \( 1, r_2 = 0 \), (2.5)

where

\[
r_1^2(f_1, f_2) = \left( \frac{f_1}{a_1} \right)^2 + \left( \frac{f_2}{a_2} \right)^2,
\]

and

\[
r_2^2(f_3) = \begin{cases} 
\left( \frac{f_3}{a_3} \right)^2, & f_3 > 0 \\
0, & f_3 \leq 0
\end{cases}
\]

with numerical parameters \( a_1 \) and \( a_2 \) controlling the blend symmetry, and \( a_3 \) allowing the user to interactively control the influence of the function \( f_3 \) on the overall shape of the blend. This definition of the function \( r \) and the definition (2.1) of the displacement function \( \text{disp}_{bb}(r) \) are not unique and can be changed, if it is necessary in particular applications. We explore the basic properties of the proposed bounded blending in the next section.

### 2.1.4 Properties of bounded blending

In this section, we illustrate such properties of the proposed blending operations with bounding solids as their local character and intuitive control of blend shape and position [Pasko G. et al. 2002a, Pasko G. et al. 2005b]. The analytical definition of blends allows for support of such unusual operations as multiple blending and partial edge blending.
Feature selection for blending

The use of a bounding solid allows the user to select a single feature (vertex or edge) of the constructive solid for blending. The corresponding pure set-theoretic operation should be replaced by the bounded blending operation. In Fig. 2.5, we present an example of union of two rectangular solids in 2D. A 2D disk is used as a solid bounding the blend in the area near a single vertex (Fig. 2.5a). The pure union operation is replaced by the bounded blending union (2.4) and the result is shown in Fig. 2.5b. The expected bounded blend appears inside the bounding disk in the upper left part of the shape in Fig. 2.5b. However, in comparison with Fig. 2.3b, no blending appears in the lower right part of the shape. This example illustrates the local character of the blending with a bounding solid.

Figure 2.5: Blending union at a single vertex selected by the bounding disk.
Figure 2.6: Shapes of 2D blends are controlled by changing parameters of the bounding ellipsoid.
Control of blend shape and position

The shape and position of the blend is controlled by its parameters and by the position and shape of the bounding solid. A family of blends inside the bounding ellipse is shown in Fig. 2.6. Subtracted material blends are described by Eq. 2.4 with negative $a_0$ values, added material blends correspond to positive $a_0$. Fig. 2.6 also shows that the blend changes its shape following the changes of the bounding solid. Fig. 2.7 and Fig. 2.8 illustrate influence of the blend parameters (Eqs. 2.4, 2.5) on the blend shape for the intersection of two halfplanes $f_1 = x$ and $f_2 = y$ with the bounding disk $f_3 = 1 - (x/4)^2 - (y/4)^2$. The following properties of the blend can be mentioned:

- The absolute value of $a_0$ defines the total displacement of the blending surface from the two initial surfaces.
- $a_0 = 0$ means pure set-theoretic operation.
- A positive $a_0$ value yields added material blend (Fig. 2.7), and a negative $a_0$ value gives subtracted material blend (Fig. 2.8).
- Values of $a_1 > 0$ and $a_2 > 0$ are proportional to the distance between the blending surface and the original surfaces defined by $f_2$ and $f_1$, respectively. Symmetric blends in a case when $a_1 = a_2$ (column B in Figs. 2.7 and 2.8), and asymmetric blends (column A, where $a_1 < a_2$ and column C, where $a_1 > a_2$) can be generated by changing these parameters (compare with the central B3 image).
- Parameter $a_3$ controls the influence of the bounding solid on the overall blend shape. It is proportional to the distance of the blend surface to the surface of the bounding solid.
Figure 2.7: Influence of the blending parameters on the shape of added material blend with $a_0 > 0$. 

$\begin{align*}
\text{A} & : a_1 < a_2 \\
\text{B} & : a_1 = a_2 \\
\text{C} & : a_1 > a_2 \\
\text{D} & : a_3 << 1 \\
\text{E} & : a_3 < 1 \\
\text{F} & : a_3 = 1 \\
\text{G} & : a_3 > 1 \\
\text{H} & : a_3 >> 1
\end{align*}$
Figure 2.8: Influence of the blending parameters on the shape of subtracted material blend with $a_0 < 0$.
The implementation of the bounded blending operation is described in Chapter III. Case studies and applications of bounded blending are presented in Chapter IV.

### 2.1.4.1 Control of 3D blend shape

Control of blend shape and position in 3D is illustrated by Fig. 2.9.

**Figure 2.9**: Shape and position of 3D blend is controlled by the bounding ellipsoid
The pure union of two ellipsoids (Fig 2.9a-1) is changed to the bounded blending union using the third ellipsoid (transparent shape). The resulting blend is located strictly inside the bounding ellipsoid (Fig. 2.9a-2), which produces an unusual blending shape localized at the top part of the initial union of ellipsoids (Fig. 2.9a-3). At the next step, we increase the size of the bounding ellipsoid (Fig. 2.9b) and correspondingly change the shape of the blend, which stretches out to the lower part of the initial shape, but keeps its symmetry. Then, we move the bounding ellipsoid to the left (Fig. 2.9c) and make the blending shape asymmetric.

### 2.1.4.2 Multiple blending

![Figure 2.10](image)

**Figure 2.10:** Multiple blending controlled by a bounding solid consisting of six disconnected components

The definition of the bounding solid by a single function allows for unusual operations such as multiple blending. As it is shown in Fig. 2.10, the bounding solid can be constructed using arbitrary primitives and operations. In this example, the disconnected bounding solid controls the blending union of two tori. The bounding solid is described...
using R-functions by a single function $f_3$ in Eq. 2.5 as union of six primitives: four equally sized solid balls and two thin ellipsoids. The result of this operation is a single connected solid with multiple blending components located inside the disconnected bounding solid.

2.1.4.3 Partial edge blending

Figure 2.11: Partial edge blending: blending intersection of two planar halfspaces bounded by an ellipsoid overlapping with a part of the edge.

As the result of the proposed blending operation is located completely inside the bounding solid, it becomes possible to apply both blending and a pure set-theoretic operation to different parts of the initial solids. This would result in partial edge blending as it is illustrated in Fig. 2.11 Here, blending intersection of two planar halfspaces is bounded by an elliptical solid, which overlaps only with a part of the intersection edge. As the result, we can observe the transition of the subtracted material blend to the sharp edge at the point of intersection between the edge and the surface of the bounding ellipsoid.
2.1.4.3 Blend on blend

![Image of blend on blend: (a) initial partial edge blend with added material; (b) new shape for blending and a transparent bounding solid; (c) blend on blend with the bounding solid; (d) blend on blend with the removed bounding solid.]

**Figure 2.12:** Blend on blend: (a) initial partial edge blend with added material; (b) new shape for blending and a transparent bounding solid; (c) blend on blend with the bounding solid; (d) blend on blend with the removed bounding solid.

As the proposed formulation does not restrict the initial shapes, it is possible to apply blending to the result of another blending operation. Fig. 2.12a shows partial edge blending of Fig. 2.11 with the change of subtracted material blend to added material.
A new elliptical shape overlapping with the blend and a new bounding solid are introduced (Fig. 2.12b). The resulting blending is shown in Figs. 2.12c,d. Note that the result has smooth surface transition between the initial edge as well as with the first blending surface.

### 2.1.4.5 Blends in constructive solids

![Figure 2.13: Two bounded blending operations are applied in the construction of a Japanese sake pot: union of the spout elements and intersection for the top part of the pot body.](image)

The proposed bounded blending operations can replace pure set-theoretic operations in the construction of a solid without rebuilding the entire construction tree data structure. We provide in Fig. 2.13 an example of the sake pot construction from the Virtual Shikki project (see section 4.1.1). Initially, the model is constructed using set-theoretic operations (see Fig. 2.13a) with two unwanted sharp edges in the area of the spout and at the top of the pot body. For the spout, a cylindrical bounding solid is used for the
blending union operation. At the top of the body, a cylinder with a hole is used for bounding the blending intersection operation. The entire object is modeled using FRep and is defined by a single procedurally defined function of three variables. The function evaluation procedure traverses the construction tree data structure with primitives as leaves and operations as nodes of the tree. In this example, the bounded blending operations replaced pure set-theoretic operations in the nodes of the tree with two additional sub-trees added for the bounding solids. Note that the bounded blending operations have three solids as their arguments and hence require 3-ary nodes in the construction tree in comparison with binary nodes for the pure set-theoretic operations.

2.2 Space-time bounded blending

Figure 2.14: Several steps of the biological amoeba motion under the microscope view at a magnification of 400x (courtesy of Michael Davidson and Florida State University).
As it was shown in section 1.4 of Chapter I, the existing approaches to metamorphosis are based on one or several of the following assumptions: equivalent topology (mainly topological disks or balls are considered), polygonal shape representation, shape alignment (shapes have common coordinate origin and significantly overlap in most of the case studies), possibility of shape matching (establishing of shape vertex-vertex, control points or other features correspondence), the resulting transformation should be close to the motion of an articulated figure.

The specific aspects of the shape transformation problem considered in this work are the following:

- initial shapes can have arbitrary topology not corresponding to each other;
- the shapes can be defined as 2D polygons, implicit surfaces, or constructive solids in 2D or 3D;
- the shapes are not aligned or overlap, and can occupy different positions in space;
- there is no correspondence established between the boundary points or other shape features;
- a combined transformation is considered including metamorphosis and nonlinear motion similar to the behavior of a biological amoeba illustrated in Fig. 2.14.

A new method of shape transformation including non-linear motion and metamorphosis is proposed in this work. We use bounded blending of FRep objects (Section 2.1) extended to the case of higher dimensional objects in space-time [Pasko G. et al. 2002b, Pasko G. et al. 2003, Pasko G. et al. 2004a, Pasko G. et al. 2004b].
2.2.1 Shape transformation using higher-dimensional blending

A blending operation in shape modeling generates smooth transition between two curves or surfaces.

Note that sometimes the term “blending” is used to designate metamorphosis of 2D shapes, as originally proposed in [Sederberg and Greenwood 1992], but we use it here in the way traditional to geometric and solid modeling, following, for example [Rossignac and Requicha 1984].

Blending versions of set-theoretic operations (intersection, union, and difference) on solids approximate exact results of these operations by rounding sharp edges and vertices (see details in Section 1.3). In the case of blending union of two disjoint solids with added material, the resulting solid is described according to Eq.1.3 as

\[
F_{\text{int}} = f_1 + f_2 + \sqrt{f_1^2 + f_2^2} + \frac{a_0}{1 + \left( \frac{f_1}{a_1} \right)^2 + \left( \frac{f_2}{a_2} \right)^2} \tag{2.6}
\]

where \(f_1(X)\) and \(f_2(X)\) are defining functions of two initial solids, \(a_0, a_1, a_2\) are parameters controlling the shape of the blend. A single resulting solid with a smooth surface can be obtained as it is shown in Fig. 2.15. This property of the blending union operation is the basis of our approach to the shape transformation.
Figure 2.15: Blending union of two disjoint solids (a) with added material results in a single solid (b-d) with the different shapes of the blend depending on parameters.

Let us formulate step by step our approach and illustrate it with 2D initial shapes (Figs. 2.16, 2.17):

1) two initial shapes are given on the $xy$-plane (a ring and a cross in Fig. 2.16a);

2) each shape is considered as a cross-sections of a half-cylinder in 3D space (a cylinder bounded by a plane from one side) as it is shown in Fig. 2.16b and Fig. 2.16c;

3) the axes of both cylinders are parallel to some common straight line in 3D space, for example, to the coordinate $z$-axis, and the bounding planes of two half-cylinders are placed at some distance to give space for making the blend in Fig. 2.16d;
Figure 2.16: Steps of 2D shape transformation using 3D unbounded blending
Figure 2.17: Frames of animation of the ring to cross transformation using unbounded blending (initial shapes cannot be obtained, see $z=-10$ and $z=10$ cross-sections)
4) apply the blending union operation with added material (2.6) to the half-cylinders (Fig. 2.16 d,e), similar to one illustrated in Fig. 2.15;

5) adjust parameters of the blend such that a satisfactory intermediate 2D shape is obtained in one or several 2D cross sections by planes orthogonal to z-axis (Fig. 2.17);

6) considering additional z-coordinate as time, make consecutive orthogonal cross-sections along z-axis (Fig. 2.17) and combine them into 2D animation.

As it is shown in Fig. 2.17, the initial shapes cannot be obtained even in cross-sections with big absolute values of z-coordinate, if global (unbounded) blending is applied. The application of the bounded blending for solving the above problem is given in the following section.

### 2.2.2 Shape transformation using bounded blending

Let us apply the bounded blending described in section 2.1 to get smooth transitions between two half-cylinders for the shape transformation (Figs. 2.18, 2.19). The half-cylinders are bounded by the planes z=0 and z=1 to make the gap [0, 1] along z-axis between them (by half-cylinders we mean semi-infinite cylinders). The bounded blending union operation is defined as

\[
F_{bb}(f_1, f_2, f_3) = R(f_1, f_2) + a_{bb} \text{disp}_{bb}(r)
\]
where \( f_1(X) \) and \( f_2(X) \) are defining functions of two initial half-cylinders, 

\[
R(f_1, f_2) = f_1 + f_2 + \sqrt{f_1^2 + f_2^2}
\]

is a standard R-function for union, and \( \text{disp}_{\text{bb}}(r) \) is a displacement function (2.5) for bounded blending.

The bounding solid for the blend in this case is an infinite slab orthogonal to \( z \)-axis and defined by the function \( f_3 \) as an intersection of two half spaces with the definitions \( z \geq -10 \) and \( z \leq 10 \). As it can be seen from Fig. 2.18, the blending displacement from the exact union of two half-cylinders takes zero value at the boundaries of the bounding solid (planes \( z=-10 \) and \( z=10 \)).

It results in the exact initial 2D shapes obtained at the cross-sections outside the bounding solid: the ring for \( z \leq -10 \) and the cross for \( z \geq 10 \) (see Fig. 2.19 and compare with Fig. 2.17).

The parameters \( a_0, a_1, a_2, a_3 \) of the bounded blend influence the blend shape and respectively the shape of the intermediate cross-sections. The selection of the boundary values for the blend mainly depends on the overall size of the initial shapes and their positions. The key point here is that whatever interval is selected for the bounded blend along \( z \)-axis in 3D space, it can always be scaled to match the required time interval for the shape transformation on a 2D plane.
Figure 2.18: Bounded blending of two half-cylinders with a ring and a cross as initial 2D profiles.
Figure 2.19: Frames of animation of the ring to the cross transformation using bounded blending (exact initial cross-sections at the boundaries of the blend: cross-sections with $z=-10$ and $z=10$).
2.2.3 Rapid transition problem

The main problem with the bounded blending and resulting animation is that the most significant part of the shape transformation happens in the [0,1] interval with the 2D shape changing rapidly from the initial to the final cross-section, which results in the visible “jump” in animation during this time interval. Let us illustrate this problem by the example of animation shown in Figs.2.20, 2.21.

Fig.2.20 shows several steps of the algorithm from initial 2D shapes to bounded blending union of two half-cylinders. Note the sharp upper and lower edge in the areas of the boundaries of the half-cylinders at z=0 and z=1 in the Fig 2.20c. Fig. 2.21 shows frames of animation (z-axis cross-sections of bounded blending object). Note that most of shape transformation happens in the [0,1] interval.

The main reason of the noticed rapid transition problem is that the bounded blending is applied to a half-cylinder bounded by a plane orthogonal to the axis. This set-theoretic subtraction of the half-space from the cylinder results in the sharp edge of the half-cylinder boundary (as seen in Fig. 2.20b) with this edge remaining a significant feature of the blended half-cylinders (see edges at the top and bottom parts of the shape in Fig. 2.20c). To avoid the described problem of the “jump” in the animation or of the rapid transition between shapes in the given interval, we propose to use “smoothed” versions of half-cylinders which undergo bounded blending [Pasko G. et al. 2004a].
Figure 2.20: 2D shape transformation with rapid transition:

a) initial 2D shape (union of two disks) and final 2D shape (cross) for metamorphosis;

b) two half-cylinders with the given 2D shapes as cross-sections;

c) bounded blending union of two half-cylinders; note the sharp upper and lower edge in the areas of the boundaries of the half-cylinders at \( z=0 \) and \( z=1 \).
Figure 2.21: Cross-sections of the bounded blending union – steps of metamorphosis.

Note that most of shape transformation happens in the [0,1] interval.
To avoid the sharp edges in the initial half-cylinders and in the resulting bounded blending, we propose to apply more “smooth” operation between the cylinder and the bounding planar half-space. The pure set-theoretic subtraction resulting in the sharp edge can be replaced by the bounded blending subtraction. For example, the half-cylinder with the cross shape (Fig. 2.20b) was generated by the pure subtraction of the halfspace $z \leq 1$, which instead can be replaced by the bounded blending subtraction based on Eq. 2.4 and Eq. 2.5, where $f_1(x, y, z)$ defines an infinite cylinder, $f_2(x, y, z) = 1 - z$ is subtracted from the cylinder, and $f_3(x, y, z) = 5 - z$ is the bounding solid for the blended subtraction, which defines the area of “smoothing”.

**Figure 2.22:** Two “smoothed” half-cylinders with the given 2D shapes as cross-sections (compare with Fig. 2.20b).
The resulting shape of the “smoothed” half-cylinder is shown in Fig. 2.22 (top). Fig. 2.22 also shows possible shapes of the “smoothed” half-cylinders depending on different parameters of the bounded blending subtraction.

Figure 2.23: Bonded blending between “smoothed” half-cylinders (compare with Fig. 2.20c).

The resulting bounded blending union of two “smoothed” half-cylinders is shown in Fig. 2.23, which, if compared with Fig. 2.20c, does not have sharp edges and has wide controlled area of the cross-sectional shapes transition. This new property is illustrated by the frames of animation in Fig. 2.24 obtained from cross-sections of the object (Fig. 2.23) along the time axis. The animation does not have the interval of the “jump” or the rapid change and the transition between the given 2D shapes can be easily controlled using parameters of blending subtraction (“smoothing” operations) and blending union between “smoothed” half-cylinders.
Figure 2.24: Frames of animation based on the bounded blending between “smoothed” half-cylinders: no “jump” in animation is observed.
2.2.4 3D shape transformation using space-time blending

As no assumptions were made in the proposed approach about the dimensionality of the initial shapes, we can apply it to 3D objects. The bounded space-time blending procedure for initial 3D shapes consists of the following steps analogous to those applied for 2D shapes and illustrated in Fig. 2.25:

1) two initial 3D shapes are given in xyz-space (see the initial cube and the union of two tori in Fig. 2.25);

2) each shape is considered as a 3D cross-section of a half-cylinder defined in 4D space-time (a cylinder bounded by a plane from one side along the time axis);

3) the bounding planes of two half-cylinders are placed at some distance along time axis to provide a time interval for making the blend;

4) the 4D half-cylinders are smoothed using the bounded blending subtraction of the planar half-spaces;

5) the added material bounded blending union operation is applied to the “smoothed” 4D half-cylinders;

6) parameters of the blending union are adjusted such that satisfactory intermediate 3D shapes are obtained in one or several cross sections along the time axis (see Fig. 2.25 showing four intermediate shapes);

7) consecutive orthogonal cross-sections along the time axis are made and combined into a 3D animation.
Figure 2.25: Metamorphosis of a cube into the union of two tori
Note that similar to the 2D case, topological changes of 3D objects are handled automatically. The initial cube in Fig. 2.25 has genus 0, while the final object has genus 4.

Some unwanted disconnected components can appear during the metamorphosis (as in Fig. 2.25 middle-left). Let us now try to fine tune the process by adding user-controlled deformations. The appearance of the disconnected component in Fig. 2.25 can be explained by quite big distance between the initial cube and the final union of two tori. We can improve the shape transformation (metamorphosis) process by adding time-dependent deformation of the cube in the direction of the tori center. This can be done with help of a non-linear space mapping (“warping”) controlled by a single point attached to the front face of the cube and moved towards the tori center [Pasko G. et al. 2004a].

The balance between the deformation and the metamorphosis can be found by selecting appropriate time schedules for both processes. In Fig. 2.26, we can observe that the cube is only deformed at the beginning phase of the transformation process (upper-right frame) and the actual metamorphosis starts later to avoid the possibility for disconnected components to appear.

The implementation of the space-time bounded blending operation and its enhancements are described in Chapter III. Case studies and applications of space-time bounded blending for shape transformations are presented in Chapter IV.
Figure 2.26: Metamorphosis of a cube into the union of two tori improved by applying additional deformation.


2.3 Model of 2-manifold with boundary

In section 1.5, we discussed modeling dimensionally heterogeneous objects and particularly the cellular functional model and implicit complexes, which can include FRep cells of different dimensions. Here, we concentrate on modeling 2D cells, namely surface sheets or stripes that can be mathematically defined as two-dimensional manifolds with boundary (2-manifolds). This work was inspired by art works of M.C. Escher namely “Sphere Spirals” (1958), “Bond of Union” (1956), and “Rind” (1955), showing spiral shaped surface sheets cut of a sphere and human head surfaces. These art works raise questions about a suitable geometric model for a surface sheet of this type and visualization of such a model. Modeling and visualization of the above-mentioned spiral type 2-manifolds using parametric surfaces seems to be a difficult task.

The alternative is to use isosurfaces of functions of three variables (implicit surfaces) [Pasko and Pasko 2004, Schmitt at el. 2004]. A surface sheet can be represented as a set-theoretic difference between some initial carrier implicit surface \( f_A (x,y,z) = 0 \) and a trimming solid \( f_B (x,y,z) \geq 0 \) (see Fig. 2.27, lower left, for modeling Escher’s “Sphere Spirals”). In the case of the intersection with another implicit surface \( f_B (x,y,z) = 0 \), the intersection curves can be presented by thin surface stripes (Fig. 2.27, lower right). We discuss below the problems of defining initial surfaces and trimming solids/ surfaces, and then the polygonization algorithm of the trimmed surface with the mesh adaptation to the surface-surface intersection curves composing the manifold boundary is presented in section 3.4.
If we define the initial surface \( A \) as \( f_A(x,y,z) = 0 \) and the trimming solid \( B \) as \( f_B(x,y,z) \geq 0 \), the idea proposed by Rvachev [Rvachev 1987] is to represent the surface as a solid by the inequality \( -f_A^2 \geq 0 \) and further to apply set-theoretic operations to it. Therefore, the trimmed surface \( T = A \setminus B \) can be defined as \( f_T(x,y,z) = 0 \) with \( f_T = -f_A^2 \& (-f_B) \), where symbol \( \& \) denotes \( \min(f_1, f_2) \) or some other R-function corresponding to the set-theoretic intersection [Rvachev 1987, Pasko et al. 1995]. A similar approach is used in the solid modeler Svlis [Bowyer 1994]. The serious disadvantage of this approach is that it becomes not possible to distinguish two sides of the initial surface represented as \( -f_A^2 \geq 0 \). Therefore, it is not possible to apply to such a model conventional polygonization algorithms based on inside-outside point classification [Bloomenthal et al. 1997, Pasko et al. 1986, Pasko et al. 1988].

An alternative approach is to represent the trimmed surface by a kind of constructive tree, where the initial surface (\( A \) in Fig. 2.27) and the trimming solid (\( B \) in Fig. 2.27) are represented separately as two primitives defined by equations or as two FRep subtrees – arguments of the set-theoretic operation. Then, the initial surface can be polygonized using one of conventional algorithms and the resulting mesh can be trimmed using vertices classification against the trimming solid. The mesh adaptation to the surface-surface intersection curves can be also performed to compose the 2-manifold boundary. The polygon subdivision has to be applied to remove trimmed parts of the initial implicit surface and more precisely approximate the trimmed surface boundary as a side effect.
Note also that surface-surface intersection methods mentioned in section 1.5 treat both intersecting surfaces equally. On the other hand, one could observe in Fig. 2.27 that for a simple initial surface the function defining a trimming solid can be quite complex. It means that evaluation times for these two functions can differ drastically. Therefore, algorithms aiming to decrease the number of evaluations of the more complex function can substantially reduce the overall computation time.

We describe our implicit surface trimming algorithm in section 3.4 and give examples related to cultural preservation in section 4.4.
**Figure 2.27:** “Sphere Spirals”. Modeling 2-manifolds with boundary:

1) surface sheet (lower left) as a set-theoretic difference $A \setminus B$ between an initial surface $A$ and a trimming solid $B$;

2) surface stripes (lower right) as an offset of the intersection curves $A \cap b(B)$ between the initial surface $A$ and the surface of the trimming solid $b(B)$.
Chapter III Algorithms and implementation

3.1. Modeling and rendering software.

Existing software tools supporting implicit surfaces and FRep modeling (described in section 1.2) include Svlis [Bowyer 1994], BlobTree [Wyvill et al. 1998], and HyperFun [Adzhiev et al. 1999]. Svlis [Bowyer 1994] is a CSG modeling system implemented as a library of classes in C++. It supports an extendable set of implicit surface primitives and allows for algebraic operations on the level of primitives. However, algebraic operations are not allowed for constructive solids, which is a serious limitation from the FRep implementation point of view. The separate level of pure set-theoretic operations makes Svlis a CSG modeler on FRep-type primitives. BlobTree [Wyvill et al. 1998] can be considered an extended CSG modeler with primitives restricted to skeletal implicit surfaces. Set-theoretic operations, linear transformations, and deformations are classes of operations supported in BlobTree.

HyperFun [Adzhiev et al. 1999, HyperFun] is a modeling language designed to be a high-level tool supporting all basic concepts of FRep modeling. In particular, HyperFun supports "geometric objects" and "geometric operations" as defined in FRep. A model in HyperFun can contain the specification of several geometric objects. Each object is defined by a function parameterized by input arrays of point coordinates and free numerical parameters. The number of coordinate variables can vary from one to greater than three to allow definition of objects of arbitrary dimension. The function can be quite complex: it is represented with the help of assignment statements; conditional selection
Functional expressions are built using conventional arithmetic and relational operators. It is possible to use standard mathematical functions ('exp', 'log', 'sqrt', 'sin', etc.). Fundamental set-theoretic operations are supported by special built-in operators with reserved symbols ('\|' - union, "\&" - intersection, "\" - subtraction, "~" - negation, "@@" - Cartesian product). The principle feature of the language is the ability to use the FRrep library that contains functions representing geometric primitives and transformations. The library is extendable, and its composition can be changed depending on a particular application area. The library version in general use contains the most common primitives (sphere, torus, blobby object, convolution surface and others) and transformations (blending union and intersection, rotation, twisting and others). Functional expressions can also include references to previously defined geometric objects. The user can create his/her own library of objects for later reuse.

A typical model in HyperFun is shown below:

```plaintext
1  my_model(x[3], a[1])
2  {
-- union of superellipsoid and torus
3    array center[3];
4    superEll = 1-(x1/a[1])^4-(x2/10)^4-(x3/0.8)^4;
5    center = [0, -9, 0];
6    torus = hfTorusY(x,center,3.5,1);
7    my_model = superEll | torus;
8  }
```
Let us comment the above model line by line to informally introduce the syntax of HyperFun, as we will use this language in the following examples of the implemented models and in the case studies of Chapter IV.

The above model is a procedural definition of a 3D solid by an equality \( F(x, y, z) \geq 0 \), where \( F \) is a function of point coordinates \( x, y, z \). This particular model defines a complex constructive solid object as a union of superellipsoid and a torus.

Lines 1, 2 and 8 are the header and the delimitations of the HyperFun object model. The string `my_model` is the object name. The symbols `x[3]` define the input array of point coordinates \( x, y, z \) in 3D space. The symbols `a[1]` define the input array of parameters which can be specified by the user.

Lines 2 and 3 are separated by the comment line.

Line 3 defines the array `center` with three elements.

Line 4 defines a superellipsoid `superEll` by an analytical expression of point coordinates.

Note that parameter `a[1]` is used as the length of one half-axis of the superellipsoid.

Line 5 defines the values of the center coordinates for the torus.

Line 6 defines the torus `torus` using the FRep library function `hfTorusY`.

Line 7 defines the object `my_model` as a union (symbol “|”) of the superellipsoid and the torus.

Application software deals with HyperFun models through using either a built-in interpreter or compilers from HyperFun to Java or C. The HyperFun interpreter has been
implemented as a small set of functions in ANSI C. It is quite easy to integrate them into
the application software since the developer needs to deal with only two C-functions: one
function for parsing of HyperFun program and another function for the evaluation the
function at the given point.

Let us briefly describe the following software tools that are available free of charge and
can be downloaded from the HyperFun Project Web site [HyperFun].

- The *HyperFun Polygonizer* (commonly abbreviated to “HFP”). This program
  polygonizes and displays an object input from a HyperFun file. It has a command line
  interface allowing the user to define a number of modeling and rendering options.
  Support for higher dimensional models is also available. The program also makes it
  possible to output the results in VRML format.

- The *HyperFun for POVRay* ("HFpov") is a plug-in to a popular ray-tracer
  POVRay ([www.povray.org](http://www.povray.org)) which makes it possible to generate high quality
  photorealistic images on an ordinary PC. HyperFun objects can be manipulated as
  POVRay objects. All ray-tracing options are set using POVRay scene descriptions.
  Animation capabilities are also available.

- The *HyperFun for Windows* ("HFW") is an interactive system allowing the user
  to easy master the FRep modeling concepts using the HyperFun language while working
  in a conventional MS Windows environment. This program has an interactive windows
graphical user interface with pop-up menus and toolbars. This program allows the user to specify a FRep model (using a built-in text editor) in the HyperFun language, to compose complex scenes with multiple objects, to specify visual parameters for subsequent rendering and to generate animation sequences. The HFW system supports a concept of "multimedia coordinates" allowing for a richer interpretation of multidimensional models through generation of variety of visual representations (such as animated spreadsheets) along with audio and other multimedia features. Especially useful for the debugging of HyperFun models is the possibility of making 2D cross-sections of 3D models and also direct modeling, rendering, and animation of 2D shapes.
3.2. Bounded blending models

In this section, we describe our HyperFun implementation of bounded blending models with control points and with a bounding solid.

3.2.1 Model of bounded blending with control points

```plaintext
1  obj1(x[2], a[1])
2  {
3     f1=x[2]-3.5;
4     f2=5.5-x[2];
5     f3=x[1]-3.5;
6     f4=9.5-x[1];
7     obj1=f1 & f2 & f3 & f4;
8  }
9  obj2(x[2], a[1])
10  {
11     f5=x[2]-1;
12     f6=4-x[2];
13     f7=x[1]-1.5;
14     f8=5-x[1];
15     obj2=f5 & f6 & f7 & f8;
16  }
17  blenduni(x[2], a[1])
18  {
19     array P1[2], P2[2];
20     f1=obj1(x, a);
21     f2=obj2(x, a);
22     P1=[3.5, 5];
23     P2=[2.5, 4];
24     a1=obj1(P2, a);
25     a2=obj2(P1, a);
26     rr=(f1/a1)^2+(f2/a2)^2;
27     d=0;
28     if (rr<1) then d=a[1]*(1-rr)^3/(1+rr);
29     endif;
30     blenduni=f1+f2+sqrt(f1^2+f2^2)+d;
31  }
```

**Figure 3.1:** HyperFun model of bounded blending of two rectangles with two control points.
The mathematical definition of the bounded blending union with control points was given in section 2.1.2. Here we present and comment the HyperFun model of the bounded blending union of two rectangles with control points (see Fig. 2.3 in section 2.1.2). The HyperFun model is given in Fig. 3.1.

Lines 1-8 define a 2D rectangle \texttt{obj1} as intersection of four halfplanes: \(x \geq 3.5, x \leq 9.5, y \geq 3.5,\) and \(y \leq 5.5.\)

Lines 9-16 define a 2D rectangle \texttt{obj2} as intersection of four halfplanes: \(x \geq 1.5, x \leq 5, y \geq 1,\) and \(y \leq 4.\)

Lines 17-18 define an object \texttt{blenduni} with the parameter \(a[1],\) which is responsible for material adding-subtracting during blending:

\(a[1] > 0\) - added material blending,

\(a[1] < 0\) - subtracted material blending.

Line 19 defines two arrays \texttt{P1} and \texttt{P2} for coordinates of the control points.

Line 20: variable \(f1\) is assigned the \texttt{obj1} function value in the point with coordinates \(x.\)

Line 21: variable \(f2\) is assigned the \texttt{obj2} function value in the point with coordinates \(x.\)

Lines 22-23 define coordinates of the control points \texttt{P1} and \texttt{P2}.

Line 24: variable \(a1\) is assigned the \texttt{obj1} function value in the point with coordinates \texttt{P2} (see the comments to Eq 2.2 in section 2.1.2).

Line 25: variable \(a2\) is assigned the \texttt{obj2} function value in the point with coordinates \texttt{P1} (see the comments to Eq 2.2 in section 2.1.2).

Line 26: variable \(rr\) is assigned the value of the generalized distance (Eq 2.2 in section 2.1.2).
Lines 27-29 calculate the displacement function $d$ value (Eq. 2.1 in section 2.1.2). Line 30 defines blending union of objects $\text{obj1}$ and $\text{obj2}$ with control points $\text{P1}$ and $\text{P2}$ (see Fig. 2.3 in section 2.1.2).

The described model was used to generate images in Fig. 2.3 with HyperFun for Windows version 1.1.

### 3.2.2 Model with a bounding solid

The mathematical definition of the bounded blending with a bounding solid was given in section 2.1.3. Here we give and comment the HyperFun model of the bounded blending union of two rectangles with a bounding disk (see Fig. 2.5 in section 2.1.4). The HyperFun model is given in Fig. 3.2.

Lines 1-8 define a 2D rectangle $\text{obj1}$ as intersection of four halfplanes: $x \geq 3.5$, $x \leq 9.5$, $y \geq 3.5$, and $y \leq 5.5$.

Lines 9-16 define a 2D rectangle $\text{obj2}$ as intersection of four halfplanes: $x \geq 1.5$, $x \leq 5$, $y \geq 1$, and $y \leq 4$.

Lines 17-18 define an object $\text{blended}$ with the four parameters corresponding to the parameter $a_0$ in Eq. 2.4 (section 2.1.3) and parameters $a_1, a_2, a_3$ in Eq.2.5. As HyperFun does not allow array indices to be 0, we had to assign the following elements of the array $a[i]$, $i = 1, 2, 3, 4$, to these parameters: $a[1]$ corresponds the parameter $a_0$ and defines material adding-subtracting during blending, $a[1] > 0$ means added material blending,
\( a[1] < 0 \) means subtracted material blending; \( a[2], a[3], a[4] \) correspond to the parameters \( a_1, a_2, a_3 \), which are the scaling factors for the functions \( f_1, f_2, f_3 \) as it is explained in section 2.1.4.

Line 19: variable \( f_1 \) is assigned the \( \text{obj1} \) function value in the point with coordinates \( x \).
Line 20: variable \( f_2 \) is assigned the \( \text{obj2} \) function value in the point with coordinates \( x \).
Line 21: variable \( f_3 \) is assigned the defining function value of the bounding disk.
Lines 22-26 calculate the generalized distance \( r_\text{r} \) according to Eq 2.5 in section 2.1.3.
Lines 27-29 calculate the displacement function \( d \) value (Eq. 2.1 in section 2.1.1).
Line 30 defines blending union of objects \( \text{obj1} \) and \( \text{obj2} \) with the bounding solid defined by \( f_3 \) (see Fig. 2.5 in section 2.1.4).

The described model was used to generate images in Fig. 2.5 with HyperFun for Windows version 1.1. The values of the model parameters in this example are \( a[1]=0.3, a[2]=1, a[3]=1, a[4]=1 \) with the model space \( x: [0,10], y: [0,6] \). Similar models of bounded blending are used in the examples of section 2.1.4 and in the case study described in section 4.1.
1 obj1(x[2],a[4])
2 {
3 f1=x[2]-3.5;
4 f2=5.5-x[2];
5 f3=x[1]-3.5;
6 f4=9.5-x[1];
7 obj1=f1 & f2 & f3 & f4;
8 }

9 obj2(x[2],a[4])
10 {
11 f5=x[2]-1;
12 f6=4-x[2];
13 f7=x[1]-1.5;
14 f8=5-x[1];
15 obj2=f5 & f6 & f7 & f8;
16 }

17 blenduni(x[2],a[4])
18 {
19 f1=obj1(x,a);
20 f2=obj2(x,a);
21 f3=1-(x[1]-3.5)^2-(x[2]-4)^2;
22 r1=(f1/a[2])^2+(f2/a[3])^2;
23 r2=0;
24 if (f3>0) then r2=(f3/a[4])^2; endif;
25 rr=0;
26 if (r1>0) then rr=r1/(r1+r2); endif;
27 d=0;
28 if (rr<1) then d=a[1]*(1-rr)^3/(1+rr); endif;
30 blenduni=f1+f2+sqrt(f1^2+f2^2)+d;
31 }

Figure 3.2: HyperFun model of the bounded blending union of two rectangles with a bounding disk.
3.3 Space-time bounded blending models

In this section, we describe HyperFun implementation of space-time blending models. Section 3.3.1 presents 2D metamorphosis as bounded blending in 3D space, and section 3.3.2 presents 3D metamorphosis as bounded blending in 4D space-time.

3.3.1 Model of 2D metamorphosis

\begin{verbatim}
1 bb2diskcross(x[3],a[4]){
   -- disks
   2 disk1=1-(x[1]-1)^2-x[2]*x[2];
   3 disk2=1-(x[1]+1.5)^2-x[2]*x[2];
   4 ddisk = disk1 | disk2;

   -- cross
   5 bl1=(x[1]+1)&(0-x[1])&(x[2]-2)&(5-x[2]);
   6 bl2=(x[2]-3)&(4-x[2])&(x[1]+2)&(1-x[1]);
   7 cross = bl1 | bl2;

   -- 1 section exists for x3 <= 0
   8 cyl1=ddisk&(-x[3]);

   -- 2 section exists fo x3 >= 1
   9 cyl2=cross&(x[3]-1);

   -- bounded blending union
   10 f1=cyl1;
   11 f2=cyl2;
   12 f3=(x[3]+10)&(10-x[3]);
   13 r1=(f1/a[2])^2+(f2/a[3])^2;
   14 r2=0;
   15 if (f3>0) then r2=(f3/a[4])^2; endif;
   16 rr=0;
   17 if (r1>0) then rr=r1/(r1+r2); endif;
   18 d=0;
   19 if (rr<1) then d=a[1]*(1-rr)^3/(1+rr); endif;
   20 bb2diskcross=f1+f2+sqrt(f1^2+f2^2)+d;
22 }
\end{verbatim}

Figure 3.3: HyperFun model of the metamorphosis using space-time blending between two disks and a cross.
The mathematical definition of the space-time blending was given in section 2.2.2. Here we give and comment the HyperFun model of the metamorphosis using space-time blending between two disks and a cross (see Fig. 2.20, Fig. 2.21 in section 2.2.3). The HyperFun model is given in Fig. 3.3.

Line 1 defines the object \texttt{bb2diskcross} in 3D space with four parameters corresponding to the parameter $a_0$ in Eq. 2.4 (section 2.1.3) and parameters $a_1, a_2, a_3$ in Eq.2.5 (section 2.1.3). Coordinates $x[1]$ and $x[2]$ correspond to geometric coordinates on the 2D $xy$-plane, coordinate $x[3]$ corresponds to time in the process of the metamorphosis. We assigned the following elements of the array $a[i]$, $i=1, 2, 3, 4$, to the parameters in the mathematical formulation (Eq.2.5) : $a[1]$ corresponds the parameter $a_0$ and defines material adding-subtracting during blending, $a[1] > 0$ means added material blending, $a[1] < 0$ means subtracted material blending; $a[2], a[3], a[4]$ correspond to the parameters $a_1, a_2, a_3$, which are the scaling factors for the functions $f_1, f_2, f_3$ as it is explained in section 2.1.4.

Lines 2-4 define the union of two 2D disks on the $xy$-plane.

Lines 5-7 define a 2D cross on the $xy$-plane as the union of two rectangles, each of which is modeled as the intersection of four half-planes.

Lines 8-9 define two half-cylinders bounded by the planes $x[3]=0$ and $x[3]=1$ and with the gap $[0, 1]$ along $x[3]$-axis between them (by half-cylinders we mean semi-infinite cylinders). The half-cylinder \texttt{cyl1} has the union of two disks \texttt{ddisk} as a cross-section. The half-cylinder \texttt{cyl2} has the cross shape \texttt{cross} as a cross-section.

Lines 10: variable \texttt{f1} is assigned the \texttt{cyl1} function value in the point with coordinates $x$. 
Lines 11: variable $f_2$ is assigned the $cyl_2$ function value in the point with coordinates $x$.

Lines 12: variable $f_3$ is assigned the defining function value of the bounding solid, which is an infinite slab orthogonal to $x[3]$-axis and defined as an intersection of two half-spaces with the definitions $x[3] \geq -10$ and $x[3] \leq 10$.

Lines 13-17 calculate the generalized distance $rr$ according to Eq. 2.5 in section 2.1.3.

Lines 18-20 calculate the displacement function $d$ value (Eq. 2.1 in section 2.1.1).

Line 21 defines blending union of objects $f_1$ and $f_2$ with the bounding solid defined by $f_3$ (see Fig. 2.20 in section 2.2.3). The shape of the blend is determined by values of four parameters $a[i]$, $i = 1, 2, 3, 4$. After the definition of the blend in the 3D space is completed, we consider the additional $x[3]$-coordinate as time, make consequent orthogonal cross-sections along time-axis in the $[-10,10]$ interval, and combine them into 2D animation.

The described model was used to generate animation frames in Fig. 2.21 in section 2.2.3 with HyperFun for Windows version 1.1. The values of the model parameters in this example are $a[1]=5$, $a[2]=2$, $a[3]=2$, $a[4]=1$ with the model space $x[1]: [-2, 6]$, $x[2]: [-4, 4]$, and $x[3]$ defined as a dynamic variable changing in the interval $[-11,11]$.

A similar model of space-time bounded blending is used in the case study described in section 4.2.
3.3.2 Model of 3D metamorphosis

The mathematical definition of the "smooth" bounded blending with additional deformation was discussed in section 2.2. Here we give and comment the HyperFun model of 3D metamorphosis of a cube into the union of two tori improved by smoothing of 4D half-cylinders and by applying additional deformation (see Fig. 2.26 in section 2.2.4). The HyperFun model is given in Fig. 3.4.

Line 1 defines the object \texttt{sbb2tocuMap} in 4D space-time with coordinates \(x[4]\) and with four parameters \(a[4]\) corresponding to the parameter \(a_0\) in Eq. 2.4 (section 2.1.3) and parameters \(a_1, a_2, a_3\) in Eq. 2.5 (section 2.1.3). Coordinates \(x[1], x[2], x[3]\) correspond to geometric coordinates on the 3D space, coordinate \(x[4]\) corresponds to time in the process of the 3D metamorphosis. We assigned the following elements of the array \(a[i]\), \(i=1, 2, 3, 4\), to the parameters in the mathematical formulation (Eq. 2.5): \(a[1]\) corresponds the parameter \(a_0\) and defines material adding-subtracting during blending, \(a[1] > 0\) means added material blending, \(a[1] < 0\) means subtracted material blending; \(a[2], a[3], a[4]\) correspond to the parameters \(a_1, a_2, a_3\), which are the scaling factors for the functions \(f_1, f_2, f_3\) as it is explained in section 2.1.4.

Lines 2-3 define three arrays by 3 elements to be used to describe torus, cube and space mapping.

Lines 4-9 define the union \texttt{dtor} of two tori in the 3D space.

Lines 10-35 define a smooth half-cylinder \texttt{cyl1} in 4D space-time with a block \texttt{block} as its 3D cross-section, which is deformed by non-linear space mapping, controlled by
moving one point attached to its surface. The components of this model are the block with deformation (lines 11-23) and model of the smoothed half-cylinder (lines 24-35) defined using a bounded blending intersection.

Line 10 defines parameters $b_1, b_2, b_3, b_4$ of the bounded blending used in lines 27-33.

Lines 11-20 define non-linear space mapping, controlled by moving one point attached to its surface. Lines 11-12 define the coordinates of the initial $x_0, y_0, z_0$ and final $x_1, y_1, z_1$ positions of the control point for the deformation.

Lines 13-17 define shift $dz$ for the space mapping as a result of motion of the control point in the $z$-direction from $z_0$ to $z_1$. The shift function $dz$ depends on coordinates $x[1], x[2], x[3]$ such that the initial point $x_0, y_0, z_0$ has maximal translation in that direction, and translations of neighboring points are controlled by a bell-shaped function $e$ (lines 13-16) with the center in the point $x_1, y_1, z_1$ with the radius of influence $b$ (line 13).

Lines 18-19 define time schedule $h_1$ of the space mapping. As the time coordinate $x[4]$ changes in the interval [-10, 10] in this model we map it to the interval [0,1] (variable $t$ in line 18) and then define the time-scaling factor $h_1$ by applying square root function to variable $t$ which makes the space mapping (line 20) faster than linear function of time on the interval [0,1].

Lines 21-22 define the block block being deformed by the space mapping (note that coordinate vector $xt$ is the result of space mapping of line 20).

Lines 23-35 define smooth half-cylinder $cyl1$ in 4D space modeled using a bounded blending intersection. The half-cylinder is an intersection of an infinite cylinder $f1$ (block
in time, line 23) and a half-space \( f_2 \) described as \( \text{time} \leq 0 \) (line 25). The bounding solid \( f_2 \) for the blending intersection (lines 27-35) is defined in line 26 as \( \text{time} \geq -5 \). Line 24 defines the time-schedule for the metamorphosis: the time \( t \) on the interval \([0,1]\) is delayed by the value 0.4, squared to make metamorphosis slower than linear, and mapped back to the interval \([-10,10]\).

Lines 36-51 define smooth half-cylinder \( c_{y2} \) in 4D space modeled using a bounded blending intersection with parameters \( b_1, b_2, b_3, b_4 \) (lines 36-39). The half-cylinder is an intersection of an infinite 4D cylinder \( f_1 \) (union of two tori in time, line 40) and a half-space \( f_2 \) described as \( \text{time} \geq 1 \) (line 41). The bounding solid \( f_2 \) for the blending intersection (lines 43-51) is defined in line 42 as \( \text{time} \leq 5 \).

Lines 52-63 define bounded blending union (metamorphosis) between two half-cylinders in 4D space: \( c_{y1} \) and \( c_{y2} \) with the bounding solid defined in line 54 as intersection of two half-spaces \( \text{time} \geq -10 \) and \( \text{time} \leq 10 \), and parameters from the input array \( a[4] \).

The described model was used to generate animation frames in Fig. 2.26 in section 2.2.4 with HyperFun for Windows version 1.1. The values of the model parameters in this example are \( a[1]=7, a[2]=5, a[3]=1, a[4]=1 \) with the model space \( x[1]: [-4, 4] \), \( x[2]: [-4, 4] \), \( x[3]: [-4,11] \) and \( x[4] \) defined as a dynamic variable changing in the interval \([-10,10]\). A similar model of space-time bounded blending is used in the 3D case studies described in section 4.2.
1 sbb2tocuMap(x[4], a[4])
{  
2 array vertex[3], center[3], xt[3];  

-- union of 2 tori  
4 center=[0,0,3];  
5 R=3;  
6 r0=0.5;  
7 tr1=hfTorusX(xt,center,R,r0);  
8 tr2=hfTorusY(xt,center,R,r0);  
9 dtor = tr1 | tr2;  

--- 1 section exists for x4 <= 0  
-- sharp cut: cyl1=block&(-x[4]);  
-- smoothing with bounded blending  
10 b1=-0.3; b2=1; b3=1; b4=1;  

-- block with space mapping  
11 x0 = 0; y0 = 0; z0 = -0.5;  
12 x1 = 0; y1 = 0; z1 = 5;  
13 b = 8.5;  
14 r2 = ((x[1]-x1)^2 + (x[2]-y1)^2 + (x[3]-z1)^2)/(b*b);  
15 e=0;  
16 if (r2<1) then e = (1-r2)^3/(1+r2); endif;  
17 dz = (z1-z0)*e;  

-- h1 - schedule for mapping  
18 t = (x[4]+10)/20;  
19 h1 = sqrt(t);  
20 xt[3] = x[3]-dz*h1;  
21 vertex=[-1.5,-1.5,-3.5];  
22 block = hfBlock(xt,vertex,3,3,3);  
23 f1=block;  

-- half-cylinder in time  
-- postponed metamorphosis  
-- parabola is slower in [0,1] + delay 0.4  
24 time = -10; if t>= 0.4 then time = (20/0.36)*(t-0.4)^2-10; endif;  
25 f2=-time;  
26 f3=time+5;  
27 r1=(f1/b2)^2+(f2/b3)^2;  
28 r2=0;  
29 if (f3>0) then r2=(f3/b4)^2; endif;  
30 rr=0;  
31 if (r1>0) then rr=r1/(r1+r2); endif;  
32 d=0;  
33 if (rr<1) then d=b1*(1-rr)^3/(1+rr);  
34 endif;  
35 cyl1=f1+f2-sqrt(f1^2+f2^2)+d;  

}
-- 2 section exists for x4 >= 1
-- sharp cut: cyl2=dtor&(x[4]-1);
-- smoothing with bounded blending

36 b1=-0.5;
37 b2=1;
38 b3=4;
39 b4=1;
40 f1=dtor;
41 f2=time-1;
42 f3=5-time;
43 r1=(f1/b2)^2+(f2/b3)^2;
44 r2=0;
45 if (f3>0) then r2=(f3/b4)^2; endif;
46 rr=0;
47 if (r1>0) then rr=r1/(r1+r2); endif;
48 d=0;
49 if (rr<1) then d=b1*(1-rr)^3/(1+rr);
50 endif;
51 cyl2=f1+f2-sqrt(f1^2+f2^2)+d;

-------------------------------------------------------------------
-- metamorphosis: bounded blending union

52 f1=cyl1;
53 f2=cyl2;
54 f3=(time+10)&(10-time);
55 r1=(f1/a[2])^2+(f2/a[3])^2;
56 r2=0;
57 if (f3>0) then r2=(f3/a[4])^2; endif;
58 rr=0;
59 if (r1>0) then rr=r1/(r1+r2); endif;
60 d=0;
61 if (rr<1) then d=a[1]*(1-rr)^3/(1+rr);
62 endif;
63 sbb2tocuMap=f1+f2+sqrt(f1^2+f2^2)+d;
64 }

Figure 3.4: HyperFun model of the metamorphosis using space-time blending between a cube and two tori with smooth half-cylinders in 4D and additional non-linear deformation.
3.4 Implicit surface trimming algorithm

In section 2.3 we discussed models of 2-manifolds with boundaries. It was proposed to represent such 2-manifolds as trimmed implicit surfaces modeled using constructive operations (intersection, subtraction) between implicit surfaces and trimming constructive solids. For rendering and other application purposes, it can be necessary to convert such a trimmed surface into a polygonal mesh.

Here, we propose an algorithm for implicit surface trimming based on the extension of existing implicit surface polygonization algorithms [Pasko and Pasko 2004]. The idea is to adaptively polygonize the initial surface near its intersection curves with the trimming surface and to test the obtained polygons against the trimming solid/surface. The proposed algorithm is based on the implicit surface polygonization method described in the previous section. In fact, any conventional polygonization algorithm can be extended in similar way.

The algorithm includes the following steps:

1. Define the initial surface as \( f_A(x,y,z) = 0 \) and the trimming solid as \( f_B(x,y,z) \geq 0 \). In the case of intersection with another implicit surface, it is defined as \( f_B(x,y,z) = 0 \).

2. Introduce a sparse spatial grid in \((x,y,z)\) space.

3. Calculate \( f_A(x,y,z) \) values at the grid points.

4. Obtain the initial triangulation for the surface \( f_A = 0 \) following a conventional polygonization algorithm providing correct topology with the given sparse grid.
5. For each triangle, calculate $f_B(x,y,z)$ values in its vertices and check the following adaptation criteria (see Fig. 3.6):

- Different signs of the function $f_B$ values in the vertices: the triangle intersects the trimming surface (Fig. 3.6a);

- Evaluate $f_B$ in the barycenter of the triangle. If the sign is different from the signs in the vertices, the trimming surface penetrates the triangle but all vertices are outside or inside the surface (Fig. 3.6b). Fig. 3.6c shows the worst case when the trimming surface intersects the triangle but the sign in the barycenter is the same as signs in the vertices. Possible improvements of the algorithm aimed to avoidance of such cases are discussed below.

- If the absolute value of $f_B$ in a vertex is less than some given $\varepsilon$, the triangle is close to the trimming surface with the possible surface-surface intersection (Fig. 3.6d,e).

6. If one of the adaptation criteria is satisfied, start recursive subdivision of the triangle in four triangles by introducing new vertices in the middle of its edges. Place newly introduced vertices on the initial surface using a search in the normal direction. If a triangle adjacent to an edge is not subdivided, an additional triangle has to be inserted to prevent cracks (Fig. 3.5). Repeat the previous step for the four new triangles.

7. If no one of the adaptation criteria is satisfied for the current triangle or the given level of subdivision is achieved, classify the triangle against the trimming solid. Check the values of the function $f_B$ in the vertices.
Three cases are possible:

1) Three positive values. The triangle is completely inside the trimming solid and is not included in the resulting mesh.

2) Three negative values. The triangle is completely outside the trimming solid and is included in the resulting mesh.

3) Different signs. The endpoints of the intersection line segment can be found by the linear interpolation along the corresponding edges. However, more sophisticated root search along the edge can produce more precise intersection point and therefore decrease the overall number of triangles. The triangle is split along this segment and the part lying outside the trimming solid is included in the resulting mesh. In the case of the intersection with another implicit surface, the intersection line segment is used to generate a patch of a thin stripe by stepping from each endpoint in both directions along the corresponding edge of the triangle.

Note that the proposed algorithm starts with the sparse mesh and then invokes its adaptation in the neighborhood of the intersection curves. Moreover, the function $f_B$ is evaluated only in the vertices of the initial surface mesh but not in all 3D grid points. This helps to decrease the number of time-consuming function evaluations while providing required accuracy of the 2-manifold boundary extraction.

The polygonization algorithm based on the uniform space subdivision can result in missing features of the initial surface such as sharp edges and vertices, spikes and other small elements. As it was mentioned before, any polygonization algorithm can be applied
The recently introduced polygonization algorithm with the mesh optimization [Ohtake et al. 2001] can be used to improve the quality of the initial surface mesh. The algorithm modification for adaptation to the trimming surface features can help avoid the unwanted cases such as shown in Fig. 3.6c.

**Figure 3.5:** Avoidance of cracks: subdivision (left triangle) and splitting (right triangle).

An example of trimming an implicit surface and a curve is presented in Fig. 3.7 rendered using the algorithm implementation [Schmitt at el. 2004]. In Fig. 3.7a, a carrier surface is modeled as a superellipsoid. In Fig. 3.7b, a trimming solid is shown which is defined as a sweep of 2D FRep pattern object projected to a sphere. Fig 3.7 (c-d) and Fig. 3.7e show the polygonized trimmed surface and the polygonized intersection curves respectively.

The proposed algorithm is also used to implement examples of trimmed implicit surfaces related to cultural heritage presented in section 4.4.
a) One or two vertices inside the trimming solid

b) Center inside the trimming solid           c) Intersection is not detected

d) Triangle inside and near the trimming surface   e) Triangle outside and near the trimming surface

**Figure 3.6:** Adaptation criteria for a polygon of the initial surface mesh.
Figure 3.7: Trimming an implicit surface (a) carrier surface; (b) trimming solid;
(c-d) polygonized trimmed surface; (e) polygonized intersection curves (top view).
Chapter IV  Case studies

4.1 Modeling Japanese craft

Historical heritage of traditional crafts such as pottery, embroidery or lacquer ware has specific features from the digital preservation point of view. First of all, any craft is a living tradition, not a fixed set of inherited items. It includes masters with their knowledge of the essential craft technology, which is often not presented in written form. This gives opportunity to preserve the technology or even enhance it using computers. On the other hand, it brings up psychological and economical problems, when computer-based technology is considered as not support, but a rival to traditional crafts. The necessity of computer-based preservation is validated by decreasing number of masters, fading technologies, and crafts loosing economical grounds. In this section, we describe practical experience in using computers to model and present traditional Japanese lacquer ware called “shikki”. The descriptions of the Virtual Shikki project, development of specific modeling operations, and Web presentation of virtual lacquer ware are given [Pasko G. et al. 2001, Vilbrandt et al. 2004].

4.1.1 Virtual Shikki project

Parts of a shikki item are produced manually using thin pieces of wood, then they are assembled, painted in different colors, and covered by natural lacquer called “urushi”. There is great variety of shikki items: boxes, small drawers, stands, cups, bowls, sake
pots, chopsticks, notebooks, and even ball pens and pencils. These items are quite different in their topology, geometry, and texture. All mentioned above problems of traditional crafts stand for shikki. Moreover, cheap plastic production makes additional economical pressure on this craft industry, thus making the necessity of the craft preservation even more actual.

The purposes of the Virtual Shikki project are reflected in the following directions of research and development activity:

- Modeling shapes and making parametric families of models of representative shikki items. A parametric family of FRep models will allow us to generate samples of a specific model with different size, width/height ratio, and so on, without repetition of the entire modeling process.

- Digitizing textures. Usually black/white texture patterns are provided on paper for a master to paint shikki. There are technical problems of scanning colored textures from the surface of existing models.

- Producing 3D virtual objects and presenting them on the Internet. Selection of the Virtual Reality Modeling Language (VRML) for the Web presentation of 3D virtual objects seemed to be quite natural recently. However, VRML has well-known drawbacks such as huge data files and long downloading time. Other data formats and browsing tools should be considered. The purpose of the virtual shikki presentation on the Web is to allow people to remotely appreciate the beauty of shapes and textures. This is important from cultural and commercial points of view. However, this brings requirements of high quality shape design and rendering with
representation of all tiny details of the objects. This virtual presentation can be also extended by tools for the user participation in the selection of textures or even in the shape design.

- Producing animations and other multimedia presentations of traditional and virtual lacquer ware. The basic mathematical representation of 3D models should allow easy transformations and metamorphosis of shapes, which is very useful in getting impressive animation.

- Documenting traditional materials and technology. This documentation can also be presented in multimedia format including video, graphics, and virtual models.

- Development of interactive design tools for modeling new items. This is a radical step of developing special computer-aided design (CAD) tools for modeling shapes and material properties of lacquer ware.

- Design or applying existing rapid prototyping machines to produce 3D physical objects from computer models. The ideal result here would be a rapid prototyping machine, which can produce wooden parts with quality equal to hand made ones.

- Internet-based e-commerce activity using interactive computer-aided design, virtual objects presentation and rapid prototyping.

### 4.1.2 Implementation issues

Here, we briefly describe the part of the project dealing with modeling shapes and presenting virtual models of existing items: specific modeling operations, design of typical shikki objects and the devoted Web site.
Specific modeling operations

Shapes of Japanese lacquer ware items have a lot of smooth transitions between different surfaces. Such transitions can be modelled using blending operation. Blending operations for FRep formulated in [Pasko and Savchenko 1994a] suffers from the resulting surfaces being offset (expanded or contracted) everywhere in the space. This is not acceptable in modeling lacquer ware shapes, because blending should not affect original surfaces outside the specified area of influence. To satisfy this requirement, we proposed in [Pasko G. et al. 2002a, Pasko G. et al. 2002b] and implemented a bounded blending operation presented in Chapter II. A sake pot is shown in Fig. 4.1a with the circled area of the resulting bounded blending. Fig. 4.1b shows union of the initial pot spout and ellipsoidal shape (left bottom part of the pot body) to be blended. The blending shape resulting from the bounded blending operation should completely reside inside this solid. The resulting blend satisfying this requirement is shown in Fig. 4.1c. Details of bounded blending operations are given in section 2.1. The proposed bounded blending was extensively used in modeling shapes of lacquer ware in this project.

![Figure 4.1: Bounded blending operation: a) sake pot with the region of blending circled; b) initial pot shape without blending; c) resulting pot shape with bounded blending.](image)

130
Virtual objects presentation

The implementation of the three first stages of the project, namely modeling shapes, digitizing textures, and presentation of virtual objects, includes the following:

1) Creation of several 3D computer models of traditional Japanese lacquer ware items. The basic modeling tool was HyperFun language [Adzhiev et al. 1999, HyperFun].

2) Generation of polygonal models using HyperFun Polygonizer and export to VRML (Virtual Reality Modeling Language) format.

3) Decimation of polygonal shapes using different software tools to achieve as low as possible size of VRML models.

4) Scanning color textures directly from lacquer ware objects with planar surfaces and from photographs.

5) Texturing polygonal models using traditional tools like 3D Studio Max.

6) Generation of images and creation of the Web site [VS]. Several snapshots of the Web site are shown in Figs.4.3.

7) A HyperFun model is available for each object. Each image at the Web site is hyperlinked to the corresponding VRML model, which can be downloaded and visualized using any VRML viewer such as CosmoPlayer. See an example of the sake set VRML model in Fig. 4.3.

The average size of VRML file is 100-500 Kb. However, the size of the sake set file (Fig. 4.2) is 4.5 Mb. On the other hand, HyperFun models for all lacquer ware items did not exceed 5 Kb. In this sense, we can conclude that HyperFun provides a high level of
compression and should be considered as a lightweight network protocol in future. We found that VRML files are too memory expensive, especially in the case of complex shapes and sets. Other and more compact Web3D formats should be considered in future. More radical solution would be to transfer small HyperFun models to the user’s computer and provide a browser able to unfold a polygonal or other representation suitable for interactive visualization.

Figure 4.2: VRML model of the sake set examined using the CosmoPlayer software.
Figure 4.3: Snapshots of the “Virtual Shikki” Web site with images hyperlinked to the HyperFun and VRML models of corresponding lacquer ware items.
4.2. Metamorphosis using space-time bounded blending

Modeling specific shapes required a large amount of routine labour on measuring control points and manual fitting model parameters. Semi-automatic methods should be introduced on the base of 3D scanning of real objects for getting control points and non-linear optimization for automatic parameters fitting.

In this section we describe the case studies of shape metamorphosis in 2D and 3D spaces with use of the space-time bounded blending presented in section 2.2 [Pasko G. et al. 2002b, Pasko G. et al. 2003, Pasko G. et al. 2004b].

4.2.1 2D metamorphosis in the Dancing Buddha project

![Figure 4.4: Initial Buddha shape (left) and a Chinese character (right).](image)

Figure 4.4: Initial Buddha shape (left) and a Chinese character (right).
Let us present a short experimental animation showing a transformation between two arbitrary polygonal topologically different shapes with holes and disjoint components (Fig. 4.4). The idea of this case study came from the “Dancing Buddhas” project [Goodwin et al. 2001] devoted to multimedia augmenting of famous Buddhist texts. We use 2D modeling and metamorphosis techniques to illustrate a concept introduced to the authors of [Goodwin et al. 2001] by a monk at Ryuukouji temple in the Aizu region of Japan, that each character (kanji) of the text of the Lotus sutra is in fact a Buddha.

The Lotus sutra plays a central role in many schools of Japanese Buddhism, most importantly the Tendai school to which Ryuukouji belongs. One central principle of the sutra is the existence of the Buddha-nature in all phenomena and the consequent equivalence of the phenomenal world and the world of enlightenment (Buddhahood). This principle (hongaku) rejects the dualism implicit in symbolization [LaFleur 1983] and thus justifies the equivalence of the signifier (kanji) with the signified (Buddha).

The Ryuukouji text, copied in the twelfth century by a number of individuals, has been designated a national treasure. Each kanji in the text is shown atop a lotus-petal dais similar to the platforms on which Buddha-images are traditionally seated. Thus the text itself illustrates the idea that each written word of the sutra is in fact a Buddha, a concept that we intend to illustrate electronically by metamorphosis of the kanji into a three-dimensional Buddha-image. With the addition of sound, the image can voice the character, creating an aural as well as a visual text. Using such techniques, whole passages or even the entire sutra may be reproduced.
The ultimate goal of the project is to treat each two-dimensional kanji with its lotus-flower platform as a separate object that will make a smooth metamorphosis into a rotating virtual sculpture, also seated on a lotus platform. Differences between one kanji and another can be used to produce variations in the Buddha-images.

The problem in the current case study is to get a new type of 2D transformation of a Buddha shape into a Chinese character. We apply the proposed approach based on space-time bounded blending to define the transformation of two polygonal shapes of arbitrary topology. An arbitrary 2D polygon (convex or concave) can be represented by a real function \( f(x,y) \) taking zero value at polygon edges [Rvachev 1974, Pasko et al. 1996]. The polygon-to-function conversion problem is stated as follows. A two-dimensional simple polygon is bounded by a finite set of segments. The segments are the edges and their extremes are the vertices of the polygon. A polygon is simple if there is no pair of nonadjacent edges sharing a point, and convex if its interior is a convex set. The polygon-to-function conversion algorithm using a monotone set-theoretic expression with R-functions was proposed in [Rvachev 1974], implemented and tested in [Pasko et al. 1996]. The conversion satisfies the following requirements:

- It provides an exact polygon boundary description as the zero set of a real function;
- No points with zero function value exist inside or outside of the polygon;
- It allows for the processing of any arbitrary simple polygon without any additional information.

In the case of a polygon with holes, each hole is processed as a separate polygon, and
then the set-theoretic subtraction is applied between the initial polygon and the hole model. In the case of several disjoint components, each component is processed as a separate polygon, and then the set-theoretic union is applied to all components.

First, two polygonal shapes were obtained from the images: the Buddha shape (see the image in Fig. 4.4 left) consists of the main concave polygon (49 segments) and two simple polygonal holes, the Chinese character (see the image in Fig. 4.4 right) consists of two disjoint components, one of which is a simple concave polygon (left part of the character) and another (right part of the character) is a concave polygonal shape (12 segments) with five holes. As one can see, the initial shapes have quite different topology; they also were placed on the plane without overlapping. We applied the proposed algorithm using bounded blending in 3D space and obtained an animation (see some of the frames in Fig. 4.5). The transformation is quite smooth except some acceleration within the interval $[0,1]$ along $z$-axis, where the most drastic transformations occur. The 3D shape of the blend and respectively the behavior of the 2D shape during the transformation can be controlled by the blend parameters, the length of the gap between the half-cylinders along $z$-axis, and the positions of the planes bounding the blend.

The 2D objects in this case study were modeled in HyperFun language and 2D images were generated as zero contours by the HyperFun for Windows (HFW) software.
Figure 4.5: Frames of the animation: transformation of a Buddha shape into a Chinese character.
4.2.2 3D metamorphosis using 4D space-time blending

Note that similar to the 2D case, topological changes of 3D objects are handled automatically when a metamorphosis process is modeled as 4D space-time bounded blending. In Fig.4.6 we present metamorphosis changing both genus and number of components in the initial shape. The initial torus in Fig. 4.6 has genus 1, while the final object has two disjoint components of genus 0.

A more complex example is shown in Figs. 4.7 and 4.8, where two given FRep shapes have quite different topology: the initial object (letters “robot”, Fig. 4.7a) has five disjoint components, some having genus 1 (letters “o”, “b”, Fig. 4.7a), some - genus 0 (letters “r”, “t”, Fig. 4.7a), and the final object (robot model, Fig. 4.7b) is a topological ball with genus 0. Note the amoebic behavior combining metamorphosis and non-linear deformations during the transformation shown in Fig. 4.8.

The 3D objects in the above examples were modeled in HyperFun language and 3D surfaces were generated as zero isosurfaces by HyperFun for Windows (HFW) software.
Figure 4.6: Transformation of a torus into two solid balls.

Figure 4.7. Initial and final complex 3D shapes with different topology.
Figure 4.8: Intermediate phases of space-time blending for 3D shapes with different topology.
Figure 4.9: Flower structure [Koning 1994].

Figure 4.10: Flower structure illustrated as the reproduction part of the plant [MW 1997].
4.3. Conceptual abstraction hierarchy analysis of flowers

In this section, let us present a case study of the conceptual abstraction hierarchy levels (see section 1.1) in botany, particularly, we look at flowers as a popular theme [Kunii et al. 2003]. We describe the adjunction space level first, and then present the cellular structured space and geometry levels. Finally, the geometric level is implemented using HyperFun modeling and rendering tools.

4.3.1 Adjunction space level

Fig. 4.9 and Fig. 4.10 are clear illustrations of flower structures [Koning 1994, MW 1997]. Through these examples, we can learn the stages of life of flowers. Flowers come to full bloom and after a while they are gone, leaving the petals scattered on the ground and serving as a favorite theme of poetry. In modeling conceptual abstraction hierarchy analysis, this means given any flower, for the petals to fall, they cannot be joined to the receptacle of the flower at the set-theoretical level. This further means the petals and the receptacle have to be disjointed. Then, how are the petals related to the receptacle before the petals are fallen? This relation reflects the situation “the petals are attached to the receptacle such that they fall after a while”.

The traditional structural analysis has been also centered on the static hierarchical relations of the objects. For example, the relations between the receptacle of a flower and its petals present a static hierarchical expression of the flower. However, dynamic
relations such as ‘petals scatter after full bloom’ are hardly represented. Usually physical behaviors of dynamics systems are formulated by a set of time dependent partial differential equations. It is unrealistic in our cases because of the complexity of the objects we are considering. The adjunction space represents the situation as it is.

At the adjunction space level, a dynamic relation of the type we explained above is expressed by an equivalence relation $f$ (symbolically denoted by $\sim$) between objects such that an object $X$ has become related via $f$ with another object $Y$ by sharing a part $Y_0 \subset Y$ such that $y \sim f(y) \mid \forall y \in Y_0, \exists x \in X: x = f(y)$. The **adjunction space** $Y_f$ (also called the **attaching space**, the **adjoining space**, and the **pasted space**):

$$Y_f = Y \sqcup_f X = Y \sqcup X / \sim = Y \sqcup X / (y \sim f(y) \mid \forall y \in Y_0, \exists x \in X: x = f(y))$$

where ‘$\sqcup$’ stands for disjoint union (also called exclusive disjunction), is obtained by identifying each point $y \in Y_0 \mid Y_0 \subset Y$ with its image $f(y) \in X$ so that $y \sim f(y) \mid \forall y \in Y_0$.

Let us denote the receptacle of a flower as a topological space $X$ and the $i$-th petal ($i = 0, 1, 2, 3, \ldots$) as a topological spaces $Y_i$. The adjunction space is a topological space obtained by starting from the receptacle $X$ and by attaching the petals $\sqcup_i Y_i$ to the receptacle via a continuous attaching map $f$ by identifying each point $x \in X \mid Y_{0i} \subset Y_i$ with its image $f(y_i) \in X$ for all $y_i \in Y_{0i}$ and for $i = 0, 1, 2, 3, \ldots$. Here $Y_{0i}$ is the edge of the $i$-th petal that touches the receptacle, and the part of the receptacle $X$ touched by the $i$-th petal is $f(Y_{0i}) \subset X$. Before the petals are fallen, thus, the following equivalence relation holds:
\[ y \sim f(y_i) \mid \forall y_i \in Y_{0i}, Y_{0i} \subseteq Y_i, \exists x \in X: x = f(y_i), \text{ for } i = 0, 1, 2, 3, \ldots \]

Hence, the adjunction space \( Y_f \) consisting of the petals and the receptacle is:

\[
Y_f = \bigcup Y_i \sqcup_f X = \bigcup Y_i \sqcup X / \sim = \bigcup Y_i \sqcup X / (y \sim f(y_i) \mid \forall y_i \in Y_{0i}, \exists x \in X: x = f(y_i),
\]
\[i = 0, 1, 2, 3, \ldots)\]

The identification map \( g \) relates two situations:

- the situation A) such that \( \bigcup Y_i \sqcup X \), that is, there are a set of petals and the receptacle, independent of each other with
- the situation B) such that \( \bigcup Y_i \sqcup_f X \), that is, a set of petals form a part of the flower such that the set of petals are attached to the receptacle at its part where the petals touch the receptacle.

To be more precise, the bottom parts of the petals are attached to the top part of the receptacle. The part of the petals \( \bigcup Y_{0i} \subseteq \bigcup Y_i \) is touched to the part \( f(\bigcup Y_{0i}) \subseteq X \) of the receptacle \( X \), and both parts become shared to form equivalence classes.

The identification map \( g \) is:

\[
g: \bigcup Y_i \sqcup X \rightarrow Y_f = \bigcup Y_i \sqcup_f X
\]
The situation B is, for example, the flower in bloom, and A is, for example, the blossom is gone.

4.3.2 Cellular structured space level

The cellular structured space of flower is obtained by endowing dimensions to the topological spaces at the adjunction space level. This is one level lower than the adjunction space level in conceptual abstraction hierarchy analysis. This process to generate the cellular structured space proceeds as follows.

The petals are closed 3-dimensional cell $\bigcup Y_i = \bigcup B^3_{petal,i}$

The receptacle $X = B^3_{receptacle}$ of the flower is a closed 3-dimensional cell.

The surfaces $\bigcup Y_{0i} = \bigcup B^2_{petal,i} \subset \bigcup Y_i = \bigcup B^3_{petal,i}$ are the parts of petals attached to the top part surface of the receptacle:

$$f(\bigcup Y_{0i}) = f(\bigcup B^2_{petal,i}) \subset X = B^3_{receptacle}$$

The adjoining space at the cellular space level is, then:

$$\bigcup Y_i \cup f X = (\bigcup B^3_{petal,i}) \cup f B^3_{receptacle} = (\bigcup B^3_{petal,i}) \cup B^3_{receptacle} / \sim =$$

$$= (\bigcup B^3_{petal,i}) \cup B^3_{receptacle} / (y \sim f(y) \mid \forall y \in \bigcup B^2_{petal,i}, \exists x \in B^3_{receptacle}: x = f(y),$$

$$i=0,1,2,3, \ldots)$$
Here, the attaching map $f$ and the identification map $g$ are at the cellular space level of conceptual abstraction hierarchy analysis of the flower:

$$f: \sqcup_i B^2_{\text{petal},i} \rightarrow B^3_{\text{receptacle}} \mid B^2_{\text{petal},i} \subseteq B^3_{\text{petal},i}$$

$$g: (\sqcup_i B^3_{\text{petal},i}) \sqcup B^3_{\text{receptacle}} \rightarrow (\sqcup_i B^3_{\text{petal},i}) \sqcup_f B^3_{\text{receptacle}}$$

The surfaces of the petals attached to the receptacle are:

$$\sqcup_i \partial B^3_{\text{petal},i} = \sqcup_i B^2_{\text{petal},i}$$

The top surface of the receptacle where the petals are attached is:

$$\partial B^3_{\text{receptacle}} = B^2_{\text{receptacle}}$$

### 4.3.3 Geometry level

**Figure 4.11:** Anemone and cosmos flowers.

Usually flower structures are illustrated as 2-dimensional annotated drawings as typically seen in Fig. 4.9 and 4.10 [Ross Koning, MW 1997]. They can be classified geometrically
as 2D shapes of the layered structure. By reformulating them in our framework, particularly at the adjunction space level and the cellular space level, conceptually we can more clearly represent relationships of the named elements of flowers, such as whether some elements are touching each other at some surfaces or they are parts of the same larger element, by attaching functions and identification functions. If they are touching each other, the disjoint unions $\sqcup$ hold, and if they are parts of a larger element, set-theoretical unions $\cup$ hold. Then, we can reformulate usual botanical explanations that go, for example, as follows: ‘The flower consists of the part of the androecium, the pistil, and the plinth, the androecium consists of the part of the anther and the filament; the pistil consists of the part of stigma, the style, and the ovary, and the plinth consists of the part of petals, the receptacle, and the calyx’. To carry out the reformulation, we need to go back to the real flowers that are 3D for reanalysis: anemone and cosmos (Fig. 4.11). The left part of Fig. 4.11 shows the flower of anemone. The reanalysis requires extensive work on the nature of the attachments of the elements at various levels of details including the biological cell levels and molecular biology levels.

4.3.4 Implementation of geometric level

The implementation of the geometrical level is conducted to further demonstrate how we carry out conceptual abstraction hierarchy analysis for representing the type of the dynamism ‘the petals are attached to the receptacle such that they fall after a while’ as mentioned above so that we can display it as images. The implementations at the adjunction space level and the cellular structure space level are already given above in
detail. Not all geometrical representations are incrementally modular. The nature of the types of the geometrical levels that are incrementally modular is a quite interesting and fundamental research theme.

As one of such geometrical levels, we use the function representation (FRep) that can directly inherit cellular spatial structures such as dimensions, cell boundaries, and cell attachments. The images created based on the geometrical implementation are illustrated in Figs. 4.12 - 4.14. In Fig. 4.12, the flower pedicel with the receptacle and five attached petals are shown. The function representation and its supporting language HyperFun are used to model and render 3D solid objects using lower dimensional cells. The pedicel and the receptacle are modeled as convolution surfaces with a curve and a point as corresponding skeletons. Each petal is a convolution surface with a 2D cellular complex as a skeleton. An attaching surface of a petal is shown in Fig. 4.12. As mentioned above, the petals $\bigcup_i B_{\text{petal},i}^3$ (i=0, 1,…, 4) are detached and have fallen after a while that is illustrated in Fig. 4.13. The images in this case study have been generated without preliminary object conversion to polygonal meshes by direct ray-tracing of the FRep objects using a HyperFun plug-in to the POVRay ray-tracer (see section 3.1).

Note that in the presented model petals are 3-dimensional cells modeled as FRep solids. Another way is to model them as 2D trimmed implicit surfaces (see sections 1.5.3, 2.3 and 3.4) and represent the entire flower as an implicit complex (section 1.5). In the next section, we provide some examples of trimmed implicit surfaces.
**Figure 4.12:** A geometrical model of flower with five petals $B^3_{\text{petal}, i}$ ($i= 0, 1, \ldots, 4$) attached to the receptacle $B^3_{\text{receptacle}}$.

**Figure 4.13:** The attaching surface $B^2_{\text{petal}, i}$ of the $i$-th petal $B^3_{\text{petal}, i}$.

**Figure 4.14:** The petals $B^3_{\text{petal}, i}$ ($i= 0, 1, \ldots, 4$) have fallen on the ground, detached from the receptacle.
4.4 Modeling Escher’s rinds

The proposed algorithm for trimming and polygonization of implicit surfaces is described in section 3.4. In this section we present several experiments with the implemented algorithm related to modeling some art works of M.C. Escher (“Sphere Spirals”, “Bond of Union”, and “Rind”) showing spiral shaped surface sheets cut of a sphere and human head surfaces [Pasko and Pasko 2004].

Initial spherical algebraic surface (Fig. 2.27), procedural blobby model, and surface of a constructive solid were trimmed using the proposed algorithm. The trimming solid for these examples was modeled using sweeping, offsetting and union operations (see Fig. 4.15). For the trimmed sphere, top and bottom critical points of the trimming solid are placed exactly on the initial sphere with the trimming surface tangent to the sphere in the critical points. This is necessary to check the proposed adaptive polygonization criteria. Fig. 4.16 illustrates the difference between non-adaptive and adaptive polygonization near the top critical point on the sparse grid with 13×13×9 nodes and four recursion levels of adaptation. The proposed polygonization algorithm can be applied to any initial surface and a trimming solid defined by continuous real functions. Fig. 4.17 is another example of a sphere trimmed by a union of the spiral (Fig.4.15b) and two cylinders. Fig. 4.18 shows an adaptive polygonal mesh and a trimmed surface of a blobby object. Its adaptive polygonization on the 20×20×20 initial grid took 40 seconds on a SG Indigo² workstation. Its non-adaptive polygonization on the corresponding 128×128×128 grid takes about 25 minutes.
Figure 4.15: Model of “Sphere Spirals” by Escher:

(a) trimming solid as a union of three spirals
(b) single spiral of the trimming solid
(c) HyperFun model of the spiral (b)
(d) trimmed sphere with the high frequency coefficient $freq$. 

\begin{align*}
\text{f}(x[3], a[1]) & \\
\{ & \\
\text{xc} = x[1]; & \\
\text{yc} = x[2]; & \\
\text{zc} = x[3]; & \\
\text{phi} = a[1]; & \\
\text{z2} = zc*zc; & \\
\text{R} = \sqrt{100.-z2}; & \\
\text{freq} = 0.5; & \\
\text{x0} = \text{R}\cos(freq*zc+phi); & \\
\text{y0} = \text{R}\sin(freq*zc+phi); & \\
\text{xt} = \text{xc}-x0; & \\
\text{yt} = \text{yc}-y0; & \\
r = 2.- z2*0.02; & \\
\text{sweep} & \\
ftrim = r*r - xt*xt -yt*yt; & \\
\text{offset} & \\
\text{offset} = 10 - z2*0.1; & \\
f = ftrim + offset; & \\
\}
\end{align*}
Figure 4.16: Adaptive trimming of a sphere (top part) by a complex solid:

(a) result of non-adaptive trimming with polygonization on a sparse grid $13 \times 13 \times 9$;

(b) mesh obtained by the adaptive polygonization (4 levels of subdivision are used for illustration);

(c) result of adaptive trimming.
Figure 4.17: “Zen spiral”: trimming solid as a union of a spiral and two cylinders (top) and the trimmed sphere (bottom).
Figure 4.18: "Blobby Spiral": a) initial blobby object

b) adaptive polygonal mesh

c) surface of the blobby object trimmed by a spiral solid
The “Constructive Rind” example in Fig. 4.19 was produced after “Rind” (1955) by M. C. Escher. The initial solid was constructed using set operations with R-functions on algebraic primitives, soft objects, and convolution solids.

**Figure 4.19**: “Constructive Rind”: (a) an initial constructive solid
(b) the solid surface trimmed by the spiral solid

Note that Escher’s works are 2-dimensional drawings of 3D objects. Modeling and rendering them directly in 3D as we did in the above examples can be considered as a step to live heritage applications such as multimedia, virtual reality, or gaming interface to artistic and historical cultural environments.
Conclusion

This thesis is devoted to some actual shape modeling issues with some applications to the cultural preservation. On the basis of the survey of existing problems and requirements to the mathematical models we have chosen the function representation for modeling homogeneous solids and cellular implicit complexes for modeling dimensionally heterogeneous objects.

From the analysis of the potential application areas we have selected several modeling problems: localization and control of blending set-theoretic operations, general constraint free shape metamorphosis, modeling and rendering two-dimensional manifolds with boundaries. We tackled the blending localization problem starting from the theory of R-functions and proposed original solution allowing for blend localization completely inside an additional bounding solid. This results in support of such unusual operations as multiple blending and partial edges blending, which hardly can be supported by other blending techniques. The proposed bounded blending operations can replace pure set-theoretic operations in the solid model without rebuilding the entire construction tree data structure. It is worth noting that the operations using bounding solids have three arguments. This brings a requirement for a functionally based modeling system to support n-ary nodes in the construction tree.

We introduced a new approach and analytical formulations for shape metamorphosis on the basis of the dimension increase, bounded blending between higher-dimensional
space-time objects, and cross sectioning the blend area for getting frames of the animation. The proposed original approach can handle non-overlapping 2D and 3D shapes with arbitrary topology defined by input polygons, set-theoretic or analytical expressions. The obtained behavior during the transformation process does not imitate the motion of an articulated figure, but rather has amorphous or amoeba-like character including non-linear motion and metamorphosis. Although the topological changes are generated automatically, analytical or numerical analysis of the transformation is needed to extract the critical points along the time axis, which can be useful for generation of more representative animation and in other applications.

A new model of two-dimensional manifolds with boundaries for implicit complexes was proposed: an implicit surface trimmed by a set-theoretic operation with a trimming solid. The algorithm of trimmed implicit surface polygonization extends conventional polygonization algorithms by including a trimming solid or surface and adapting the polygonal mesh to the 2-manifold boundary. The proposed adaptive solution significantly accelerates polygonization. Depending on the initial surface and the trimming solid/surface complexity, we obtained 20 to 70 times speed-up if compared with the non-adaptive algorithm. Applications of the algorithm include cultural preservation, computer-aided design, and heterogeneous object modeling.

The proposed methods and algorithms have been applied in several case studies. The bounded blending operation was used in modeling traditional Japanese lacquer ware in the Virtual Shikki project, which has resulted in a Web site with 3D models in HyperFun
and VRML formats. The proposed space-time blending was used in the presented case studies of 2D- and 3D-metamorphosis, and in the Dancing Buddhas project particularly, where an animation of Buddha shape transformation to a Chinese character (kanji) was generated. Abstraction hierarchy analysis of flowers was done on the adjunction space and cellular levels with the geometric level implemented and illustrated using HyperFun tools. The application of trimmed implicit surfaces for modeling spiral objects by M. C. Escher was presented. In fact, Escher’s works are 2-dimensional drawings of 3D objects. Modeling and rendering them directly in 3D as we did in the presented examples can be considered as a step to live heritage applications such as multimedia, virtual reality, or gaming interface to artistic and historical cultural environments.

Let us summarize advantages of using FRep in cultural preservation. It provides precise mathematical definition of modeled objects. The clearly defined open text format of FRep models in HyperFun language obviously increases the survival period of the archived models, especially in comparing with not openly specified proprietary formats. The combination of geometry and attributes in the constructive hypervolume model [Pasko et al. 2001] allows for modeling not only the shapes, but also material and other object properties. Support of multidimensional and particularly time-dependent models in FRep is useful for the modeling aging and decaying objects with the help of finite element analysis [Kartasheva et al. 2003]. The recovered structure and constructive elements of the object can be employed for rebuilding lost objects and in ”live heritage” applications such as animation or interactive multimedia.
The main disadvantage of FRep is time-consuming function evaluation, which currently retards interactive applications. However, the compact HyperFun format suits well parallelization, distributed and mobile computing, which can be one of solutions to this problem. Another difficulty is that constructive modeling usually requires high levels of 3D modeling skill and is labor-intensive. Therefore, we examine in [Vilbrandt et al. 2004] the application of fitting of a parameterized FRep model to a cloud of data points as a step towards automation of the modeling process. Automation of the logical structure extraction using genetic programming will be investigated in the future work.

The proposed general operations allow for modeling complex 3D, time-dependent and mixed-dimensional shapes with high precision and flexible control, which is hard to achieve with other shape modeling methods. The proposed solutions are important for the entire shape modeling area and have been already applied in modeling cultural objects, human body, objects of nature (trees, flowers), and in computer animation.
Acknowledgements.

I am very grateful to Professor Tosiyasu Laurence Kunii for the inspiration and the greatest support at every step of this work and to my husband Professor Alexandr Pasko for his every breath and each dot of the work. My sincere thanks are to Mr. Iwashita Nobumasa, Professor Ikeda Makoto, Goto Yuichiro, Abiko Maki and Kato Emiko, Miura Takeshi for creating the comfortable and stimulating research atmosphere at the ITI of Kanazawa Institute of Technology.

I wish also to thank the HyperFun team for providing wonderful modeling and rendering tools. The work on the Virtual Shikki project was supported by the grant of the Fukushima Prefecture government and led by Professor Ikedo Tsuneo. I would like to thank Takase family from Aizu-Wakamatsu for sharing information on the shikki production process and needs, Eric Fausett for texturing VRML models of lacquer ware models.

The author would like to acknowledge the help of the University of Aizu students Hiroshi Igarashi, Yasutaka Iwami, and Tomoko Matsuoka, who participated in the early stages of the research on bounded blending; Yukinobu Katou and Kensuke Masuda, who participated in the early stages of the research on space-time blending.

The amoeba motion images are the courtesy of Michael W. Davidson and Florida State University.
References


169


