Construction of Implicit Complexes: A Case Study

E. Kartasheva[α], V. Adzhiev[β], P. Comninos[β], A. Pasko[γ], B. Schmitt[δ]

[α] Institute for Mathematical Modeling, Russian Academy of Science, Moscow, Russia, ekart@imamod.ru

[β] The National Centre for Computer Animation, Bournemouth University, Poole, UK, {vadzhiev&peterc}@bournemouth.ac.uk

[γ] Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan, pasko@k.hosei.ac.jp

[δ] Computer Graphics Research Institute, Hosei University (Digital Media Professionals), Tokyo, Japan, schmitt_benjamin@yahoo.fr

ABSTRACT

This paper presents a detailed description of a case-study demonstrating a novel method for modelling and rendering of heterogeneous objects containing entities of various dimensionalities within a cellular-functional framework based on the implicit complex notion. Implicit complexes make it possible to combine a cellular representation and a constructive function representation. We briefly describe a formal framework for such a hybrid representation as well as a general structure for implicit complexes. Then, using a representative example, we show how an implicit complex can be constructed geometrically and topologically. We also consider the main rendering issues specific to implicit complexes and describe some implementation problems.

Keywords

Function representation, cellular representation, implicit complex, polygonization, ray-tracing.

1. INTRODUCTION

Heterogeneous object modelling is becoming an important research topic in different application areas, such as volume modelling and rendering, modelling of objects with multiple and varying materials in CAD and rapid prototyping, representing results of physical simulations, geological and medical modelling. Such objects are heterogeneous from two points of view: their internal structure and dimensionality. Varying materials and other attributes of an arbitrary nature constitute a heterogeneous internal structure. A dimensionally heterogeneous object in 3D space can include elements of different dimensions (points, curves, surfaces and solids) combined into a single entity from the geometric point of view (i.e., a point set) and the topological point of view (i.e., a cellular complex).

A model of objects with fixed dimensionality and heterogeneous internal structure (multidimensional point sets with multiple attributes or so-called *constructive hypervolumes*) was proposed in [Pas01]. This model uses real functions of point coordinates (scalar fields) to represent both the object geometry and its attributes. The hybrid cellular-functional model [Adz02] allows for representing a heterogeneous object as a cellular complex with both explicit and implicit cells (cellular domains) of different dimension. Such an object is called an *implicit complex (IC)*, which is defined as the union of properly joined cellular domains. Explicit cells can be represented as point lists, parametric curves and surfaces. Implicit cells can be implicit surfaces and their patches, intersection curves of implicit surfaces, or functionally represented (FRep) solids [Pas95].

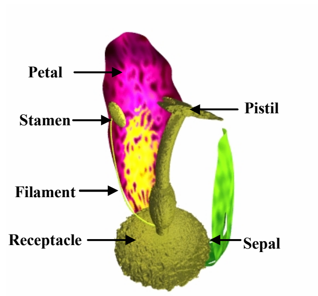


Figure 1. Components of a flower

In this paper, we present a case-study which allows us to demonstrate a novel technology for modelling and rendering of heterogeneous objects using implicit

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG 2005 conference proceedings, ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

complexes. This case-study inspired by [Kun03] is based on a model of a flower (Figure 1) which has the following components of different dimensionalities: a 3D receptacle, 3D stamens, 3D pistil, 2D petals and sepals, and 1D filaments. All the components have a colour that can be expressed as an attribute. Of course, such an object can be modelled and rendered using more traditional means than those based on the cellular-functional framework; however we believe this case-study allows us to show all the conceptual phases of modelling and rendering within the cellular-functional framework as well as outline some implementation issues. We pay particular attention to the theoretical and practical issues of the implicit complex construction.

2. RELATED WORKS

Here, we briefly discuss approaches for modelling dimensionally heterogeneous objects using various cellular representations and previously published work on modelling objects with varying distribution of material and other attributes.

A typical technique for describing heterogeneous objects is to represent them as collections of homogeneous components. To describe complex topology, different spatial subdivisions, topological stratifications [Mid00], and complexes [Ohm01, Pao93] are used.

Topological complexes and their construction methods are discussed in a number of publications on shape modelling and solid modelling. Multidimensional simplicial complexes are used in [Pao93] for dimension-independent geometric modelling for various applications. A Selective Geometric Complex (SGC) [Ros90] is a non-regularised non-homogeneous point set, represented through enumeration as the union of mutually disjoint connected open subsets of the real algebraic variety. A procedure for designing cellular models based on CW-complexes with the emphasis on the topological validity of the resulting shapes is considered in [Kun99, Ohm01].

To specify non-geometric properties of objects, spatial subdivisions are used in computer graphics and in finite element analysis (FEA) as the underlying structures for piecewise analytical descriptions of attribute functions. Usually a basic topological subdivision is selected, which can be described by a topological stratification [Ros90], a cell complex [Cut02], or a voxel model. Different types of functions can be used to describe attributes [Jac99, Par01].

Another approach to modelling heterogeneous objects is based on using real functions of point coordinates. For example, the constructive hypervolume model [Pas01] supports uniform constructive modelling of point set geometry and attributes using such functions. Then, a theoretical framework combining a cellular representation and a constructive function representation was proposed in [Adz02]. An independent cellular and functional representation of the same object is useful, but not sufficient in certain applications. For example, in the above mentioned flower model, each dimensionally homogeneous component (3D receptacle, 2D petals, 1D filaments) can be functionally represented. However, without additional information one cannot separate individual functions for the components from the single function describing the entire object. This additional information about objects components, their dimensions, and attachments to each other are used in applications such as finite element mesh generation, animation, and rendering. The above was the motivation for introducing in [Adz02] a hybrid cellular functional model based on the notion of an implicit complex, which allows for the flexible combination of cellular and functional object geometry models and attribute models. In the current paper, we examine in more detail the construction of implicit complexes.

3. THEORETICAL FRAMEWORK

Here we provide a brief description of a theoretical framework for the representation of implicit complexes. A more formal and detailed consideration can be found elsewhere [Adz02]. In this paper we present novel material concerned with theoretical and practical matters of the description and construction of ICs.

A Hybrid Representation of Geometry

The hybrid geometrical model of heterogeneous objects presented in [Adz02] combines a cellular representation and a constructive representation using real-valued functions. Formally, the hybrid representation for a geometric object $D \subseteq \Omega$ is defined as follows:

$$M_H : D = \{X \mid X \in \Omega, \Omega \subseteq E^n, X \in |K^p|\}$$

where $\Omega \subseteq E^n$ is a modelling space and K^p is an implicit p-dimensional complex. The definition of an implicit complex is based on the concepts of cellular spaces and CW-complexes [Fom97, Mas67]. A CW-complex provides a general representation for different topological complexes including polyhedral and cellular complexes.

D is defined as a closed cellular space (*domain*) and can be represented as a carrier of a CW-complex K, such that D=|K|. The hybrid representation scheme can be defined in the form of the pair $<D,K>$ represented through a union operation as $<D,K>=\bigcup_{i=1}^{N}<D_i,K_i>$. We can use different representation schemes for various $<D_i,K_i>$ pairs. Suppose that for some subdomains we use the cellular representation. Such subdomains and their subdivisions are called *explicit* and are denoted by DE_i and KE_i. Then for the other domains, denoted by DI_j, we use an FRep; so their cellular representation denoted by KI_j is not necessarily known but can, in principle, be built using some known method. We call such domains as well as their subcomplexes *implicit* ones. Then the point set D is represented as the union $D=(\bigcup_{i=1}^{N}DE_i)\cup(\bigcup_{j=1}^{M}DI_j)$. The complex K is also represented as the union of the corresponding explicit subcomplexes KE_i and the implicit subcomplexes KI_j. Thus, $K=(\bigcup_{i=1}^{N}KE_i)\cup(\bigcup_{j=1}^{M}KI_j)$. Note that if the intersection of two cells of an IC is not empty, then it consists of a collection of cells of this complex. The same is true concerning the boundaries of cells. It is necessary to impose constraints on the domain boundaries similar to those for subdomains.

Definition of Implicit Complexes

In the general case, a p-dimensional implicit complex K^p is expressed as $K^p=\{\{e_i^q\}_{i=1,...,n_q}^{q=0,...,p},\{t_j^s\}_{j=1,...,m_j}^{s=0,...,p}\}$, where e_i^q are cells of all the explicit subcomplexes of K^p and t_j^s are implicit cells (such that each t_j^s is the point set coinciding with the carrier of an implicit subcomplex of K^p). Thus, for any DI_j there exist $t_j^s\in K^p$ such that $t_j^s=DI_j$.

Explicit cells e_i^q are defined with respect to the definition of the CW-complex. The shape of each explicit cell e_i^q is defined by a characteristic mapping, and its boundary is mapped onto a subcomplex $M^r\subset K^p$ with dimensionality $r<q$.

Each implicit cell t_j^s is a closed point set defined by an FRep. The boundary of the implicit cell t_j^s is not necessarily mapped onto any subcomplex of K^p and can contain both explicit and implicit cells. Explicit cells are indivisible elements of a subdomain subdivision containing no other cells. Some cells of an implicit complex can lie inside implicit cells of the complex. Note that an *r*-dimensional K^r implicit complex can be reduced to a cellular one. We assume that each implicit cell $t_i^s\in K^r$ (where $0<s\le r$) can be discretized. The corresponding methods were described in detail in [Kar03].

Implicit Complex Description

IC topology is described by relations between its elements. The general structure of a 3D IC is illustrated by Fig. 2. By definition, a 3D IC consists of 0D, 1D, 2D and 3D cells. Let G^p be a set of p-dimensional cells g_i^p. Such a set contains both explicit and implicit cells. There are two main types of relations that establish connections between cells of different dimensionalities: the boundary relation and the "to contain" relation.

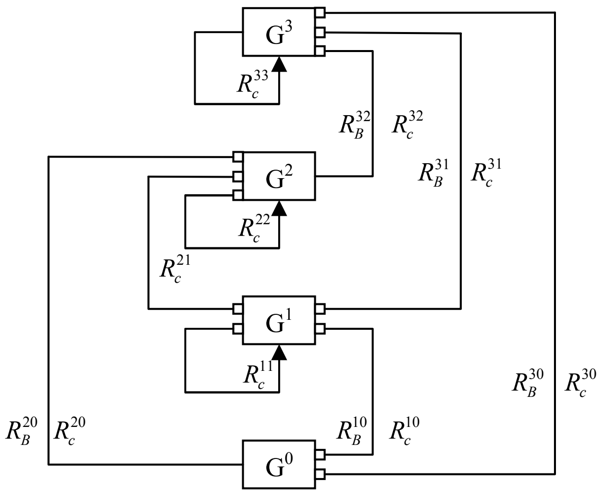


Figure 2. The general structure of a 3D IC

We denote by Rb^{ps} the *boundary relation* between *p*-dimensional and *s*-dimensional cells, $Rb^{ps}\subset G^p\times G^s$, $s<p$. The pair (g_i^p,g_j^s) belongs to Rb^{ps} if g_j^s belongs to the boundary of g_i^p. The relation "*to contain*" is denoted by Rc^{ps}, $Rc^{ps}\subset G^p\times G^s$, $s\le p$. The pair (g_i^p,g_j^s) belongs to Rc^{ps} if $g_j^s\in g_i^p$ and $g_j^s\cap\partial g_i^p\ne g_j^s$. The entire structure of 3D IC is defined by six different boundary relations and nine different "to contain" relations. Other relations are the *co-boundary*, the "*to be contained*", the *incidence* and the *adjacency* relations. These can be derived from the boundary and the "to contain" ones using various operations on relations.

The geometry of ICs is described as follows. An explicit cell e_i^0 is described by its coordinates. An implicit cell t_j^o is defined in FRep with an inequality of the form $F(X)\ge 0$. In E^3 space, the function $F(X)$ for 0D cells (points) can be described using functions representing the intersection of three surfaces, the intersection of a curve and a surface, or directly as the formulation $F(X) = -d(X - X_0)$, where d is the distance from the given point X_0. An implicit cell t_j^o can also be described as an image of a point functionally defined in 2D space.

An explicit 1D cell e_j^1 is defined by a characteristic mapping $\chi_j^1 : X = \varphi_j^1(u)$, which maps the segment $[u_0, u_1]$ in the space of the real parameter into a curvilinear and perhaps a closed segment in the E^3 space. An implicit cell t_j^1 can be described in two ways. It can be defined as an FRep object in E^3 by an inequality of the form $F(X) \geq 0$. In such a case, the 1D cell takes the form of an arbitrary curve defined as the intersection of two surfaces in E^3. Alternatively, the cell t_j^1 can be represented as an image of an FRep curve described in 2D space by an inequality of the form $f_j^1(x) \geq 0$, where $x \in E^2$. This mapping is given by a function of the form $h_j^1 : E^2 -> E^3$.

Explicit 2D cells are represented as images of triangles and quadrilaterals resulting from characteristic mapping. For each cell e_i^2, we define the characteristic mapping $\chi_i^2 : X = \varphi_i^2(u,v)$, which takes the rectangle $u_0 \leq u \leq u_1, v_0 \leq v \leq v_1$ in the parameter space and maps it onto a surface patch in E^3. An implicit 2D cell t_j^2 can also be described in two ways. It can be defined with FRep in E^3 by an inequality of the form $F(X) \geq 0$. Alternatively, one can use a functional description of the form $f_j^2(x) \geq 0$ in E^2 with the subsequent mapping of the form $h_j^2 : E^2 -> E^3$.

To represent an explicit 3D cell, a variety of maps of the form $\chi_i^3 : X = \varphi_i^3(u,v,w)$ can be used to describe the shape of curvilinear polyhedrons. Maps of this kind have been extensively used in describing finite element sets. Such maps can be used to describe the cells and attach them to the boundary cells of the complex by obeying certain boundary conditions. Once again, an FRep of the form $F(X) \geq 0$ is used to describe the implicit cell t_j^3.

And finally, to describe the non-geometric properties of a heterogeneous object, represented by an IC, we use the cellular-function model of the attributes introduced in [Adz02]. Each attribute A_i is defined by a function of the form $S_i : E^3 \rightarrow N_i^{mi}$, where N_i^{mi} is a set of attribute values (which can be a vector or tensor space). N_i^{mi} is embedded into a real-valued space of a proper dimension *mi*. Thus, $N_i^{mi} \subseteq \Re^{mi}$. Attributes assigned to an implicit complex K are described by functions S_i in a piece-wise manner, i.e. $S_i = \{(S_i)_j^s \mid (S_i)_j^s : g_j^s \rightarrow N_i^{mi}, g_j^s \in K\}$.

Implicit Complex Construction

To create an IC, it is necessary to describe the shapes of all its elements and, to specify the entire boundary and the "to contain" relations between its elements. The *attachment operation* is introduced allowing the creation of the IC in a component-wise manner, in order of increasing dimensionality of its components. This process is constructive and iterative.

We start with an empty complex $K_o = \varnothing$, and then at each construction step i we attach a new component L_i to the already formed subcomplex K_{i-1}, thus creating a new complex $K_i = K_{i-1} \cup L_i$ which is a subcomplex of the target complex K. We introduce two types of the attachment operation based on the one defined for CW complexes [Kun99, Ohm01].

The *cell attachment* operation assumes that at each step i of the process another cell g_i^r is attached to the complex K_{i-1}. First we define the shape of this cell using one of the methods described above. Then, we have to modify the relations. So for all implicit and explicit cells $g_j^s \in K_{i-1}$ lying on the boundary of g_i^r we add the pairs (g_i^r, g_j^s) to the boundary relations Rb^{rs} (where $s < r$). Then, for each implicit cell $g_l^p \in K_{i-1}$ (where $p \geq r$) that contains g_i^r, we have to add the pair (g_l^p, g_i^r) to the "to contain" relation Rc^{pr} (where $p \geq r$). Finally, and only if g_i^r is an implicit cell, for all implicit and explicit cells $g_m^q \in K_{i-1}$ (where $0 \leq q \leq r$) lying inside g_i^r we add the pairs (g_i^r, g_m^q) to the "to contain" relation Rc^{rq} (where $q \leq r$).

The *complex attachment* operation deals with the *procedure* of attaching the complex L to the complex K_{i-1}. Assume that L is *properly joined* to the complex K_{i-1} that is $L \cap K_{i-1} = C$ (where C is a subcomplex of both L and K_{i-1}). Thus we have to create a complex $K_i = K_{i-1} \cup L$, $K_i = \bigcup_{p=0}^{p \leq r} G_i^p$. First, we define an attachment map ψ that relates the equivalent cells of the initial complexes. Then, we obtain the sets G_i^p of the complex K_i as quotient sets of the union of the corresponding sets of the initial complexes by the quotient map ψ as follows: $G^p(K_i) = G^p(K_{i-1}) \cup G^p(L)/\sim_\psi$, ($0 \le p \le r$). Finally, we define the boundary and the "to contain" relations of the complex K_i .

4. THE FLOWER CASE STUDY

Here we present a systematic description of how the cellular-functional model of a flower that is considered as an example of an heterogeneous object can be constructed based on the theory presented above.

The Components

To create a model of such a composite object as a flower, we start from modelling its separate components (see Fig. 1).

- The 3D receptacle is modelled using an FRep, and is defined as a half-ellipsoid combined with a solid noise function (algebraic sum with Gardner's noise function). The corresponding FRep function is denoted by F_r.

- The 3D pistil is also defined by an FRep, and the corresponding constructive tree is composed of ellipsoids in the leaves and blending union in the nodes. The corresponding function is F_p .

- The 3D stamens are defined as an algebraic sum of an ellipsoid and Gardner's solid noise function, corresponding to the function F_s .

- The 2D petals and sepals are specified in two steps. First, an object is described by an FRep in 2D space (Fig. 4a). The function F_{pt} describing this object on the $U \times V$ plane is defined as the difference between a large 2D solid ellipse and two smaller ones (representing the holes). Then, tapering and general space mapping deformations are applied to the object. The corresponding mapping function h_{pt} is defined using a technique similar to FFDs and the resulting object is a surface patch in 3D space (Figure 4b).

- The 1D filaments are explicitly defined by spline curve segments defined in 3D space. These curve segments are defined as $\bar{r}(u) = \varphi_f(u), u \in [u_0, u_1]$.

We consider the flower as an heterogeneous object which has a colour property. This is represented by an RGB colour attribute which is described by the function $S = E^3 \to \Re^3$ in a piece-wise manner, so that $S = \{S_i\}$, where each S_i maps the corresponding subset of E^3 into the RGB space.

The listed components differ in dimensionality and representation. Let us show how an accurate definition of a composite object can be made.

The Implicit Complex Structure

Altogether, the complex K describing the entire heterogeneous model of the flower fragment consisting of the receptacle, the pistil, a petal and a stamen, and contains cells corresponding to different flower components (Fig. 3). The receptacle is described by the 3D cell t_1^3 , the pistil by the 3D cell t_2^3 , the stamen by the 3D cell t_0^3 , the filament by the 1D cell e_0^1 , and the petal by the 2D cell t_0^2 . To assemble the listed components together we have to add auxiliary cells describing their interconnections.

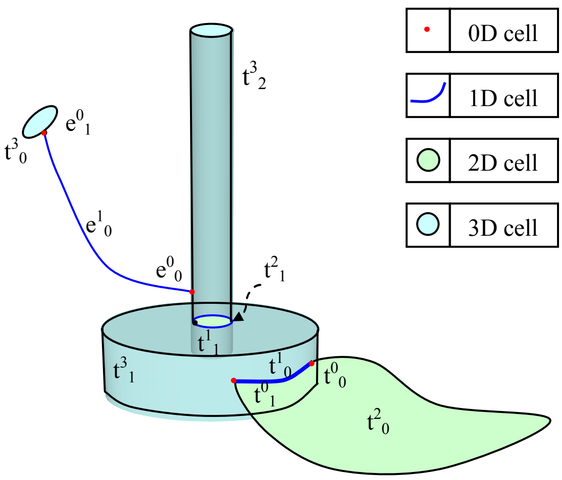


Figure 3. IC structure for 'Flower' model.

Let G^i be a set of cells of dimension i. As each cell is associated with the assigned attribute functions being specified in brackets, we have:

$$G^0 = \{e_0^0(S_2), e_1^0(S_1), \ t_0^0(S_3), t_1^0(S_3)\}$$

$$G^1 = \{e_0^1(S_1), \ t_0^1(S_3), t_1^1(S_2)\}$$

$$G^2 = \{t_0^2(S_3), t_1^2(S_2)\}$$

$$G^3 = \{t_0^3(S_1), t_1^3(S_2), t_2^3(S_1)\}$$

The boundary relations Rb establishing connections between cells include the following pairs:

$$Rb^{10} = \{(e_0^1, e_0^0), (e_0^1, e_1^0)(t_0^1, t_0^0), (t_0^1, t_1^0)\}$$

$$Rb^{21} = \{(t_0^2, t_0^1), (t_1^2, t_1^1)\} ; \ Rb^{20} = \{(t_0^2, t_0^0), (t_1^2, t_1^0)\}$$

$$Rb^{32} = \{(t_2^3, t_1^2), (t_1^3, t_1^2)\} ; Rb^{31} = \{(t_1^3, t_0^1), (t_1^3, t_1^1), (t_2^3, t_1^1)\} ;$$

$$Rb^{30} = \{(t_0^3, e_1^0), (t_1^3, t_0^0), (t_1^3, t_1^0), (t_2^3, e_0^0)\} .$$

All the "to contain" relations are empty for this complex.

A Detailed Mathematical Description

Let us introduce the following mathematical entities:

- *FrepCell(xdim,F)* represents an implicit cell defined by a real-valued function *F(x)* and depends on the space dimension *xdim*.

- *MappedFCell(xdim,F,h)* represents a mapped implicit cell defined by a real value function *F(x)* and a mapping $h{:}E^2\text{->}E^3$.

- *ExplicitCell(bnd,χ)* represents explicit cells defined by their boundaries *bnd* and characteristic mapping χ,. Note that explicit 0D cells are described by just their coordinates.

Now, we can define all the cells in step-by-step manner.

1. The stamen is described by the 3D implicit cell t_0^3 = *FrepCell(xdim=3,F=F$_s$)*.

2. The receptacle is described by the 3D implicit cell t_1^3 =*FrepCell(xdim=3,F=F$_r$)* and the pistil – by the 3D implicit cell t_2^3 = *FrepCell(xdim=3,F* $= F_p \setminus_0 F_r$) where $\setminus_0$ denotes an R-function defining set-theoretic subtraction. Note that such a definition insures that the cells t_1^3, t_2^3 have no common internal points. Initially the pistil and the receptacle were described independently so they may overlap in 3D space. To specify the intersection of the cells t_1^3, t_2^3 we introduce a 2D implicit cell t_1^2 = *FrepCell(xdim=3,F* $= -(F_r)^2 \Lambda_0 F_p$) which represents a surface segment, here Λ_0 denotes an R-function defining a set-theoretic intersection. The boundary of the cell t_1^2 intersects the boundaries of the cells t_1^3, t_2^3, so we have to add a 1D cell t_1^1 = *FrepCell(xdim=3,F* $= -(-(F_r)^2 \Lambda_0 F_p)^2$) describing this intersection.

3. The stamen and the pistil are connected to each other by the filament described by the 1D explicit cell e_0^1. We set e_0^1 = *ExplicitCell(bnd={* $\varphi_f(u_0), \varphi_f(u_1)$ *},* $\chi = \varphi_f(u), u \in [u_0, u_1]$ *}*. We assume that the end points of the curve segment lie exactly on the boundaries of the cells t_2^3 and t_0^3, and the segment has no other common points with t_2^3 and t_0^3. So we can define the intersection of the filament e_0^1 with the pistil t_2^3 and the stamen t_0^3 explicitly by 0D cells e_0^0 and e_1^0. These cells are specified by their Cartesian coordinates: $e_0^0 = \varphi_f(u_0)$, $e_1^0 = \varphi_f(u_1)$.

4. The petal is described by the mapped implicit cell t_0^2. Initially the petal was defined in the same manner by the pair (F_{pt}, h_{pt}). But this definition does not take into account the adjacent component, namely the receptacle. Then, one can formulate the following constraints for the trimmed petal which intersects the receptacle only along the boundary:

$$\begin{cases} F_r(h_{ptx}(u,v), h_{pty}(u,v), h_{ptz}(u,v)) \le 0 \\ F_{pt}(u,v) \ge 0 \end{cases}$$

Here we assume that the mapping function $h_{pt}: E^2 \to E^3$ is defined as $h_{pt}(u,v) = (h_{ptx}(u,v), h_{pty}(u,v), h_{ptz}(u,v))$, where *(u,v)* is a point in E^2. Thi is equivalent to the description of the cell t_0^2 = *MappedFCell(xdim=2,* $F = -F_r(h_{pt}(u,v)) \Lambda_0 F_{pt}(u,v)$, $h = h_{pt}(u,v)$).

5. The petal t_0^2 is a 2D cell in 3D space, and the receptacle t_1^3 is a 3D cell. Their intersection is defined by a curve segment represented by the implicit cell t_0^1. The constraints for this cell (which has to belong to both the surface of the receptacle and the boundary of the petal) can be expressed as the following system:

$$\begin{cases} F_r(h_{ptx}(u,v), h_{pty}(u,v), h_{ptz}(u,v)) = 0 \\ F_{pt}(u,v) \ge 0 \end{cases}$$

Therefore, t_0^1 =*MappedFCell(xdim=2,* $F = -(F_r(h_{pt}(u,v)))^2 \Lambda_0 F_{pt}(u,v)$, $h_0^1 = h_{pt}(u,v)$).

Finally, the start and end points t_0^0 and t_1^0 of the 1D cell t_0^2 are defined in a similar manner taking into account some relevant constraints (omitted here because of shortage of space).

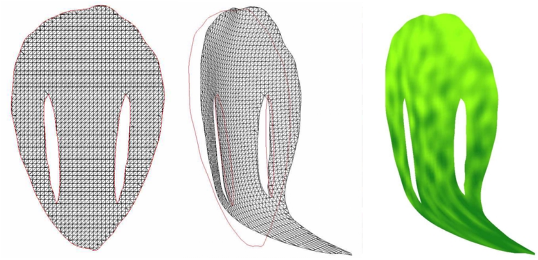
The Implementation Model

We have implemented cellular-functional modelling of heterogeneous objects within an object-orientated framework. Let us outline the principal classes which are directly derived from the presented theoretical description. The basic *IComplex* class represents an implicit complex data structure (Fig.2 is an illustration). Its attributes represent six boundary relations and nine "to contain" relations as well as cells of various dimensionalities. It is assumed that all the cells are enumerated. Each relation is described by the object of the *Relation* class which contains all the pairs of numbers of related cells. The operations of the *Relation* class allow us to get the indices of all the related cells as well as to add and delete pairs of cells. Accordingly, the *IComplex* class includes operations for adding, deleting and modifying relations, as well as inquiry operations on relations.

The implicit complex geometry within the *IComplex* class is specified using objects of classes inherited from the abstract *G<dim>* class parameterized by the cell dimensionality *dim*. Objects of the *ExplicitCell* classes represent explicitly specified parametric curves, surfaces and solid objects. As to implicit cells, they are represented by objects of either *FRepCell* or *MappedFCell* classes. They are all built on the basis of an abstract *FRep* class that is also parameterized by the dimensionality of the coordinate space *xdim*. This basic class contains virtual operations for defining the point membership with respect to some FRep object as well for rendering and discretization. All the classes describing FRep primitives and operations are inherited from the basic abstract *FRep< xdim>* class. *ParamCurve*, *ParamSurf* and *ParamSolid* classes allow the definition of corresponding parametric entities. They also contain virtual operations for rendering and discretization. Our software tools used for implementation of the flower case-study are built using this object-oriented model.

5. RENDERING

In this section we describe how an implicit complex can be rendered.



(a) *(b)* *(c)*

Figure 4. (a) planar 2D surface; (b) polygonized 2D petal in 3D space; (c) textured sepal

An implicit complex includes implicit and explicit cells of different dimensions. Let us first consider the application of existing rendering techniques. Explicit cells, such as points, lines and triangles, can be rendered using traditional techniques such as a standard library (OpenGL or DirectX) and graphics hardware. Implicit cells require more advanced techniques, such as raytracing [Blo97] or polygonization algorithms for 3D implicit cells [Pas88]. These techniques have been extended in [Sch04] by rendering implicit cells of lower dimensions, defined as trimmed implicit surfaces and curves. Here, we assume that each cell of the implicit complex can be rendered individually using one of the above-mentioned techniques.

To render the flower model, one can choose different polygonal based and ray-trace based rendering techniques. If one wishes to visualize the implicit complex using a polygonal representation, one can easily convert each implicit cell to an explicit representation using an ad-hoc polygonization algorithm ([Pas88, Sch04]), and then render the implicit complex with traditional graphics hardware. Figure 4 shows the polygonized sepal of a flower. To generate Fig. 4c a colour attribute implemented through procedural texturing was used.

To directly ray-trace an implicit complex, one needs to combine the existing ray-tracing techniques for explicit and implicit cells using a common Z-buffer.

Figure 5. Rendering a flower modelled as an IC. The filaments are defined as 1D explicit cells, the petals (yellow and magenta) and sepals (green) are defined as 2D implicit cells, and the remaining components (stamens, receptacle and pistil) are defined as implicit 3D cells.

The main problem is to render 1D cells as they can not be ray-traced directly since they are defined as curves and line segments. Therefore, 1D explicit and implicit cells have to be first rendered by techniques other than ray-tracing. For instance, one can retrieve the data stored in a frame buffer and a Z-buffer after using a polygonization routine and graphics hardware for rendering, or directly use an existing line drawing algorithm. Then, the remaining cells of higher dimensions can be ray-traced all together. For each ray, the intersection points with the explicit cells are directly computed, and the intersection points for the implicit cells are procedurally evaluated.

Fig. 5 shows rendered images of the modelled flower. The 2D and 3D cells have attributes representing the colour based on the constructive hypervolume model [Pas01]. Note that for visualization purposes, we replace 1D cells with thin cylinders to be able to shade them. The petals and sepals have first been polygonized, as they are defined as 2D shapes, and then deformed by a forward mapping. First, 1D and 2D cells were rendered by ray-tracing resulting in a frame buffer and a z-buffer. Next, 3D implicit cells were ray traced and combined with the image already stored in the frame buffer, depending on the comparison of the depth of the current ray intersection with the implicit cell and the depth stored in the z-buffer.

6. CONCLUSION

Implicit complexes make it possible to combine a cellular representation and a constructive function representation into a uniform model. In this paper, we have described the theoretical framework and the implementation techniques for the construction and rendering of such models using a simple but representative case-study. We have paid particular attention to the practical problems of construction of a cellular-function model.

This relatively simple case-study has allowed us to show the benefits of the approach which are invaluable for complex real assemblies. Such benefits include: preserving the initial representation of all the components (however different they may be) and guaranteeing topologically correct definitions for all parts and relationships (in particular for boundaries). This approach also allows us to handle conformity between the object's geometry and its attributes which represent its non-geometric properties.

Future work directions include the development of specific operations applicable to entire implicit complexes, an extension of the model to time-dependent implicit complexes; further development of the multidimensional version of the model and its applications, and the implementation of a specialized modelling and animation language which uses this novel modelling technique.

7. REFERENCES

[Adz02] Adzhiev V., Kartasheva E., Kunii T., Pasko A., Schmitt B.: Hybrid cellular-functional modelling of heterogeneous objects, *Journal of Computing and Information Science in Engineering, Transactions of the ASME 2*, 4 (2002), 312-322.

[Blo97] Bloomenthal J. et al.: *Introduction to Implicit Surfaces.* Morgan Kaufmann, 1997.

[Cut02] Cutler B., Dorsey J., McMillan L., Mueller M., Jagnow R.: A procedural approach to authoring solid models. *In Proc. SIGGRAPH'02, ACM TOG 21*, 3 (2002), 302-311.

[Fom97] Fomenko A.T., Kunii T.L.: *Topological modeling for visualization*, Springer-Verlag, Tokyo, 1997.

[Jac99] Jackson T., Liu H., Patrikalakis N., Sachs E., Cima, M.: Modeling and designing functionally graded material components for fabrication with local composition. *Control, Materials and Design 20*, 2/3 (1999), 63-75.

[Kar03] Kartasheva E., Adzhiev V., Pasko, A., Fryazinov O., Gasilov V.: Surface and volume discretization of functionally-based heterogeneous objects. *Journal of Computing and Information Science in Engineering, Transactions of the ASME 3*, 4 (2003), 285-294.

[Kun99] Kunii T.: Valid computational shape modeling: design and implementation. *International Journal of Shape Modeling 5,* 2, (1999), 123-133.

[Kun03] Kunii T.L., Ibusuki M., Pasko G., Pasko A., Terasaki D., Hanaizumi H.: Modeling of conceptual multiresolution analysis by an incrementally modular abstraction hierarchy. *IEICE Transactions on Information and Systems*, vol. E86-D, No. 7, 2003, pp. 1181-1190.

[Mas67] Massey W.S.: *Algebraic topology: An introduction*. Harcourt, Brace&World, Inc, 1967.

[Mid00] Middleditch A., Reade C., Gomes A.: Point-sets and cell structures relevant to computer aided design. *Int. Journal of Shape Modeling 6*, 2, (2000), 175-205.

[Ohm01] Ohmori K., Kunii T.: Shape modeling using homotopy. In Proc. *Int. Conf. on Shape Modeling and Applications*, IEEE Computer Society, 2001, 126-133.

[Par01] Park S. M., Crawford R., Beaman J.: Volumetric multi-texturing for functionally gradient material representation. In Proc. *Sixth ACM Symposium on Solid Modeling and Applications*, ACM Press, 2001, 216-224.

[Pas88] Pasko A., Pilyugin V., Pokrovskiy V.: Geometric modelling in the analysis of trivariate functions. *Computers and Graphics 12*, 3/4 (1988), 457-465.

[Pas95] Pasko A., Adzhiev V., Sourin A., Savchenko V.: Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer 11, 8 (1995), 429-446.

[Pas01] Pasko A., Adzhiev V., Schmitt B., Schlick C.: Constructive hypervolume modeling. *Graphical Models 63,* 6 (2001), 413-442.

[Pao93] Paoluzzi A., Bernardini F., Cattani C., Ferrucci V.: Dimension-independent modeling with simplicial complexes. *ACM TOG 12*, 1 (1993), 56-102.

[Ros90] Rossignac J., O'Connor M. SGC: A dimension independent model for pointsets with internal structures and incomplete boundaries. In *Geometric modeling for product engineering,* ed. by M. Wozny, J. Turner, K. Preiss, 1990.

[Sch04] B. Schmitt, A. Pasko, G. Pasko, T. Kunii: Rendering trimmed implicit surfaces and curves. *In Proc. Afrigraph 2004*, South Africa, ACM SIGGRAPH publication, 2004, pp. 7-13.