Methods for Feature-Based Design of Heterogeneous Solids

H. Liu[1]
T. Maekawa[1,2]
N. M. Patrikalakis[1]
E. M. Sachs[1]
W. Cho[1]
[1]Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA
[2]Yokohama National University, Yokohama, 240-8501, Japan

MIT Design Laboratory Memorandum 03-1

March 4, 2003
Revised: October 20, 2003

Abstract

This paper presents a parametric and feature-based methodology for the design of solids with local composition control (LCC). A suite of composition design features are conceptualized and implemented. The designer can use them singly or in combination, to specify the composition of complex components. Each material composition design feature relates directly to the geometry of the design, often relying on user interaction to specify critical aspects of the geometry. This approach allows the designer to simultaneously edit geometry and composition by varying parameters until a satisfactory result is attained. The identified LCC features are those based on volume, transition, pattern, and (user-defined) surface features. The material composition functions include functions parametrized with respect to distance or distances to user-defined geometric features; and functions that use Laplace's equation to blend smoothly various boundary conditions including values and gradients of the material composition on the boundaries. The Euclidean digital distance transform and the Boundary Element Method are adapted to the efficient computation of composition functions. Theoretical and experimental complexity, accuracy and convergence analyses are presented. The representations underlying the composition design features are analytic in nature and therefore concise. Evaluation for visualization and fabrication is performed only at the resolutions required for these purposes, thereby reducing the computational burden.

Keywords: Solid free-form fabrication, 3D Printing, local composition control, functionally graded materials

1 Introduction

1.1 Background and motivation

One of the great potential benefits offered by *Solid Freeform Fabrication* (SFF) technology is the ability to create parts that have composition variation within them. Such *Local Composition Control* (LCC) has the potential to create new classes of components. Material composition can be tailored within a component to achieve local control of properties (e.g., index of refraction, electrical conductivity, formability, magnetic properties, corrosion resistance, hardness vs. toughness, etc.). By such local control, monolithic components can be created which integrate the function of multiple discrete components, saving part count, space and weight and enabling concepts that would be otherwise impractical. Controlling the spatial distribution of properties via composition will allow for control of the state of the entire component (e.g., the state of residual stress in a component). Integrated sensors and actuators can be envisioned which are enabled by LCC (e.g., bimetallic structures, in-situ thermocouples, etc.). Devices which have as their function the control of chemical reactions are possible. The utility of "mesoscopic" parts made by SFF will depend strongly on the ability to locally control composition.

Among the SFF processes, *Three-Dimensional Printing* (3D

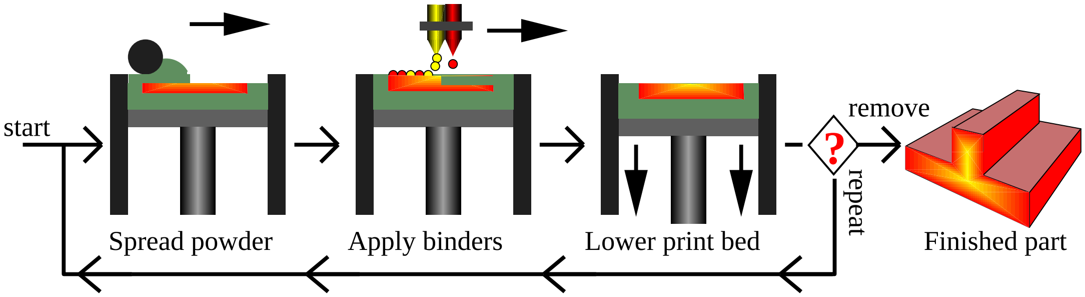


Figure 1: 3D Printing illustrating Local Composition Control (LCC)

Printing) is particularly well suited to the fabrication of parts with LCC. 3D Printing creates parts in layers by spreading powder, and then ink-jet printing materials into the powderbed [40, 41, 42]. In some cases, these materials are temporary or fugitive "glues", but in many cases, these materials remain in the final component. Examples of the latter include: ceramic particles in colloidal or slurry form, metallic particles in slurry form, dissolved salts which are reduced to metal in the powderbed, polymers in colloidal or dissolved form, and drugs in colloidal or dissolved form. 3D Printing has been extended to the fabrication of LCC components by printing different materials in different locations, each through its own ink-jet nozzle(s). Figure 1 illustrates this conceptually with three different colors, each representing the printing of a different material into the powder bed with local control of position. 3D Printing is thus capable of fully three-dimensional control of composition.

Several promising applications are under active development. Drug delivery devices are being created by printing different drugs at prescribed locations within the interior of a pill or implantable device. These drugs are then released into the body according to designed release profiles [25]. Gradient Index Lenses (GRIN) are another class of LCC applications which refract light by gradients in the index of refraction, rather than by external geometry. Such lenses can provide the functionality normally associated with multi-component ground optics at lower cost and in a smaller space. The drug delivery and GRIN applications are for high value added devices which are small in size and thus can reasonably be manufactured by 3D Printing. LCC is also being applied to the fabrication of tooling by 3D Printing. Hard phases such as TiC are being printed local to the surface of a tool for increased wear resistance. Tools with local control of porosity (for venting of gases) are being fabricated by printing a material which acts to block the infiltrant during furnace densification. Although large in size, tooling applications can be economical because small quantities are required.

Realizing the potential utility of LCC in SFF is a challenge because of the absence of knowledge, methods and tools in the area of computer representation and design of parts with LCC. Generic computer representations are necessary to allow for electronic specification of composition within a component and it is desirable to devise a suite of tools which allows a designer to communicate with this representation using high level features that are sensible to a designer. The designer must be able to visualize and interrogate the evolving model. This paper is focused on the design methods for LCC solids. The resulting tools will be generic and applicable to a broad range of SFF technologies.

This paper is organized as follows: in the rest of Section 1, past work on design of LCC objects is reviewed and summarized. A review of feature-based design is given in Section 2. In Section 3, we present the detailed approach of our methods. Complexity, accuracy and convergence analyses of our methods are given in Section 4. In Section 5, the system implementation and some examples are presented. Finally, Section 6 concludes the paper.

1.2 Past work on design of LCC object

Jackson *et al.* [24] presented a review of representation methods for heterogeneous solids such as voxel- or mesh-based structure, volumetric texture-based structure, and generalized modeling methods. Pratt *et al.* [35] classified the existing approaches into exact boundary-based parameterizations of object interiors, volume discretization approaches and non-boundary conformant parameterization methods. Chandru *et al.* [10] suggested using a voxel-based representation to build composite structures by associating material information with each voxel. Pegna and Safi [34] suggested using a finite-element mesh to represent the model and assign material values to each node of the mesh. Liu *et al.* [29] developed a finite-element based representation system and a distance function-based design method and a related efficient evaluation method. The algorithm developed allows specification of the locally controlled composition as a piecewise polynomial or rational function of the minimum distance d from the entire boundary surface.

Kumar and Dutta [28] presented r_m set-based representation method and a Boolean operation-based material composition function design method. Shin and Dutta [47] extended the work in [28] to a constructive representation scheme. Under such a scheme, r_m sets are not disjoint interiors of the heterogeneous object anymore. Where different sets join each other, the material function in the joint is the distance based weighted sum of the material function of each set. They also proposed material design function related to the coordinates, and geometric specific blendings of materials and its sweeping [47]. Park *et al.* [33] developed a volumetric multi-texturing method based on a procedural algorithm for evaluating material variation within a model. Their method attaches material blending functions to entities in an existing solid modeling system. Martin and Cohen [30] presented a framework for representing attribute data independently of geometric data within a trivariate NURBS volume. They extended an existing modeling and data fitting techniques and developed efficient algorithms for volume evaluation and visualization. Siu and Tan [48] presented a scheme of including a grading source in the design of material composition variation and constructed a design method with the extended CSG type Boolean operations. Biswas *et al.* [3] presented a heterogeneous material modeling method for solids based on distance fields.

1.3 Summary of limitations of existing approaches

Current approaches either based on volume meshing or cellular decompositions are awkward in editing geometric and material composition information simultaneously, because they lack the concept of editable LCC features; in effect, they permit sequential editing (first of geometry and then composition), which is not flexible and limits the designer's options. Current LCC models are limited to low level data and operators and do not allow for the symbolic representation of the designer's intent with respect to composition. Also as such, design changes cannot be efficiently propagated. Tessellation of the volume of a model (e.g., via tetrahedral meshing) early in the design and fabrication pathway, although expedient for testing of ideas, does not provide a long term solution for the following reasons: (1) Tessellation implies both approximation of surface geometry and material composition, which is undesirable in general, and for realistic accuracies of approximation leads to verbose evaluated representations, that are unattractive for general LCC modelers. (2) Tessellation approximation accuracy for surface geometry and material composition can be improved via adaptive meshing procedures, however these are difficult to implement robustly and efficiently. (3) Methods for tessellation of a volume into tetrahedral meshes suffer from the general robustness problem in computational geometry relating to inexact computation.

2 Feature-based design

Feature technology has emerged in response to vital industry needs in design and manufacturing. A recent definition of a *feature* is: *a representation of shape aspects of a product that are mappable to a generic shape and functionally significant for some product life-cycle phase* [2]. Compared to the approach of feature recognition, feature-based design is rapid and easier by making use of the information in the process. In addition, from the design point of view, feature-based design has the potential of supporting the design process better, such as improving the quality of design and improving the link between design and applications [44, 45]. With rapid development of feature technology, feature-based design is becoming one of the fundamental design paradigms of CAD systems.

In feature-based design, parts are constructed from a sequence of feature adding operations. This paradigm lends itself to separating the design into two layers, one comprising an unevaluated, generic representation, the other comprising an evaluated, instance representation. Pratt [36] first suggested an explicit volumetric representation of features via extension of the radial-edge data structure, a non-manifold boundary representation data structure. Rossignac [38] proposed a cellular scheme that permits mixed-dimension representation with the Selective Geometric Complex (SGC) structure. He presented methods based on space decomposition and the concept of intentional feature to correct validity errors caused by feature interactions. He also addressed the issue of editing form features. Bidarra *et al.* [1] presented a cellular model as an alternative to SGC model to avoid excessive generality. Cellular model is a connected set of volumetric quasi-disjoint cells. This model integrated shapes of a feature with cells in the cellular model. Each feature has an explicit volumetric representation, a set of associated cells. Feature interactions are maintained in attributes of cells, cell faces and cell edges. With this cellular model, Bidarra *et al.* have classified feature interactions and developed an algorithm for detecting these interactions. A summary of their work on feature modeling is presented in [2].

In terms of feature definition, the procedural method is very general and convenient specifically for object-oriented programming. In this manner, CAD/CAM integration can be carried out more easily. An example of neutral procedural definition language is *Erep* by Hoffmann and Joan-Arinyo [20]. Dedhia *et al.* [17] presented the ASU testbed for rapid prototyping of feature-based applications. Each feature has a feature type identifier, a name, a list of generic, compatible features and a CSG tree representation. Hoffmann and Joan-Arinyo [21] presented a procedural mechanism for generating and deploying user-defined-features (UDF) in a feature-based design paradigm. The proposed paradigm is to address customization needs in a simple, effective way. The usefulness of that mechanism relies on three basic capabilities: use of standard tools, parametrization of UDFs, and graphical interaction. The advantage of UDF also lies in that design changes can be prestructured and then compound features can be made to have greater independence from each other so that validity error induced by feature interaction is less likely to happen. In the area of designing and editing features within the constraint-based frame, Chen and Hoffmann [11] presented semantics for feature attachment; Vasilis *et al.* [51] developed a schema for generic naming of geometric entities; based on the schema mentioned, Chen and Hoffmann [12] presented algorithms for naming matching, which is the mapping from abstract features to the corresponding geometric entities in the boundary representation for the purpose of design reevaluation.

By feature-based design, functionality can be captured using constraints as mathematical equations of the variables that the design depends on. An approach for capturing engineering meaning has been described by Nielsen *et al.* [31]. Ullman [50] explored the evolution of information during the design process and gave some insights in the design of function.

The feature-based approach has also been introduced into assembly design. Shah and Rogers [46] presented an assembly model as an extension of feature-based design. It showed several basic structures that can be used to define relationships between assemblies, parts, features, feature volume primitives and evaluated boundaries. Generic relations which facilitate constraint specification between target and reference entities were also presented. Cugini [16] studied the concept of assembly feature and developed a prototype system for application in aeronautics field. Recently, Brunetti and Golob [8] presented an approach for conceptual design via extension of a recent feature-based parametric part and assembly modeling system. The functions considered are such that they are mappable to a working principle of the assembly. Holland and Bronsvoort [22] introduced an integrated object-oriented product model for both single-part and assembly and showed the usefulness of such modeling in planning.

3 Approaches

In order to overcome limitations summarized in section 1.3, we propose an approach which builds on the concept of feature-based design (FBD) [45, 21, 2], which involves the following key concepts: (1) by introducing the concept of editable LCC features, the simultaneous editing of geometric and material information is formalized and simplified; (2) maintenance of an unevaluated exact representation for the geometry and composition for as long as possible along the information pathway, provides a high level codification of the design useful in data exchange and in a general setting not associated with a specific SFF process; (3) evaluation of the above exact representation is performed as needed at later stages of the pathway, e.g., for visualization and design verification at an appropriate resolution corresponding to the visualization parameters or for fabrication only at the resolution printable by a particular process. Our feature-based approach for modeling parts with LCC can be characterized as a procedural, unevaluated representation. Although the current FBD systems carry rich information in terms of features, they only allow users to create multi-material solids with piecewise constant composition using composite structures and assemblies. Due to the nature of FBD, such systems usually cover a limited number of features. In order to address these problems, we propose to extend the definition of features in geometric feature models so as to define the semantics of an LCC feature and extend an existing FBD system to facilitate model creation through LCC features.

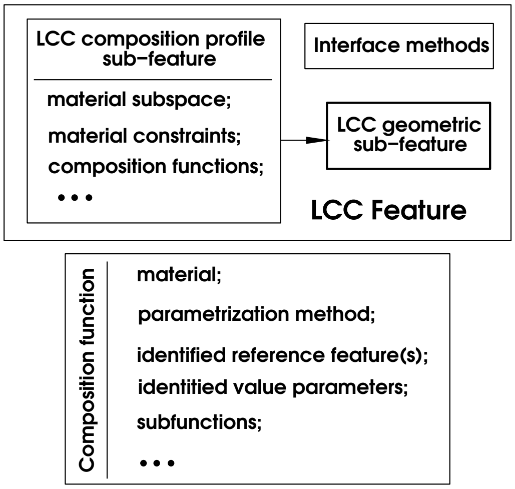


Figure 2: LCC feature class definition

3.1 Definition and classes of LCC features

The basic approach is to identify potential classes of LCC applications and for each class, identify features, which would be useful in design. For the purpose of allowing users to specify composition variation in the interior of a solid, we define an LCC feature as follows: *An LCC feature is a construct composed of two major attributes: (a) a generic parametrizable shape; (b) a composition function defined over that shape.* LCC features as conceived here do not involve direct specification of higher level functional properties of a part (e.g., strength, wear), which are beyond the scope of this work. As shown in Figure 2, in terms of data structure, an LCC feature is composed of two substructures/sub-features, one providing the representation of generic shape, the other providing the representation of composition profile. The arrow in Figure 2 illustrates that the composition sub-feature is applied to the domain of the geometric sub-feature whether it is in the form of volume or surface. The block named “Interface methods” refers to the methods within LCC feature for user interactions. Therefore, an LCC feature can be viewed as primarily comprising two sub-features, for geometric shape and composition profile. The geometric sub-feature can be any standard geometric feature or its extension to a general user-defined feature (UDF) [21], e.g. user-defined surface feature, volume feature, transition feature, pattern feature. Composition profile sub-feature has parameters such as material subspace and constraints on material composition. It also possesses attributes defined through composition functions. Material space is a catalog of materials available to the designer for LCC object design. Material subspace is a subset of the material space. Composition is the vector of volume fractions of each material defined over the material subspace and the generic shape of the feature. Composition function is the mapping function from the geometric sub-feature to the material subspace. Therefore, composition profile sub-feature is a dependent feature of the LCC geometric sub-feature. Composition constraints (design rules) are typically inequalities that specify e.g., what material composition or what gradient of material composition can be fabricated. The procedural or declarative definitions in a manner analogous to conventional feature-based design are needed for the design by LCC features. LCC feature design should be tailored to both geometry and composition design intent.

LCC Feature Examples: A bimetalic sensor or actuator can be designed by defining the composition in a plane and extruding this composition along a line or sweeping it along a curve. A cylindrical Gradient Index Lens with composition gradient as a function of the distance from axis and distance from the bottom face can be created by revolution of a 2D closed sketch, while the composition function is parametrized with respect to the axis and the 2D sketch entities. The impeller of Figure 3-(a), has fin features designed as bosses. Each fin has a composition which varies with height from the base so that it can have high wear resistance at the tip and ductility at the root. This feature is then replicated by patterning. In Figure 3-(b), a drug primitive has a composition as a function of distance to its axis, and a pattern of the drug is inserted to a pill.

Using the above examples as conceptual prototypes, creation methods for the two attributes of LCC features (generic shape and composition function) are discussed in the next two sub-sections 3.1.1 and 3.1.2.

3.1.1 Construction of generic shapes for LCC feature

1. LCC features based on volume features — LCC volumes: The generic shape of a volume feature can be constructed with a series of form features. According to [21], three primary form features are *generated features*, *modifying features* and *datum features*. In such a design paradigm, the generic shape is both an abstraction in terms of features and an evaluated geometry. Under the user-defined feature design scheme, the feature structure is expressed with an acyclic directed graph where the nodes are form features and the edges are dependencies between them. Examples of volume feature creation methods in current FBD systems include Extrusion, Boss, Sweep, Thicken, Draft, Revolution, Sweep Cut, Revolution Cut, Surface Cut, Thicken Cut, Holes, and Shell [49]. A generic shape may frequently be created by alternative methods. For example, a cylinder can be created via either a revolution or an extrusion.

2. LCC features based on user-defined surface features — LCC surfaces: The generic shape of an LCC surface feature is the surface which is user-defined with a series of form features. For some applications graded composition may be desired for the volumetric domain bounded by a set of boundary surface features and some kind of composition variations need to be assigned on these surface features. An example is a smooth blending of composition between the composition values on different sets of boundary. The example of Figure 4 illustrates such a case. Here the top surface of the cylinder has a composition variation which is a function of distance to the axis of the cylinder, the rest of the surfaces belong to the base-extrude feature and have compositions at constant value. These are the two LCC surface features, i.e. the top surface and the rest of the surfaces. The domain of the volume is assigned a composition which is a blend of the LCC surfaces, and it is therefore an LCC volume feature. Another example are applications that require users to define LCC feature shapes with half-space divisions. The half-spaces can be defined with either a plane or a curved surface and used as generic shapes in LCC Surface features.

3. LCC features based on pattern features — LCC pattern: A *pattern feature* is a set of features arranged in circular or multi-dimensional array. The generic shape of an LCC pattern is a group of volumes patterned from a seed volume. The material compositions are patterned from the seed LCC volume in the same way. Figure 3-(b) demonstrates the use of pattern feature to place drug cell primitives in a pill in a cylindrical lattice pattern. Figure 3-(a) also shows another example of circular pattern feature.

4. LCC features based on transition features — LCC fillet: In conventional feature-based geometric modeling, transition features, such as fillets and chamfers, blend two or more surfaces. By extension, an LCC fillet feature blends the composition in a volumetric domain defined by transition surfaces and other user-defined surfaces which do not have to be conventional edge or corner blends. For example, for the root of the fins in Figure 3-(a), the specification of the transition volume can be easily constructed by surface cutting operations. The surfaces can be planes or curved surfaces. In order to edit the transition volume, the surfaces are defined with variable parameters. LCC fillet is a special type of the LCC volume feature in that the material compositions are smooth blending of the adjacent volumes. In Figure 23, a blending method using Laplace's equation is used, as described in later sections.

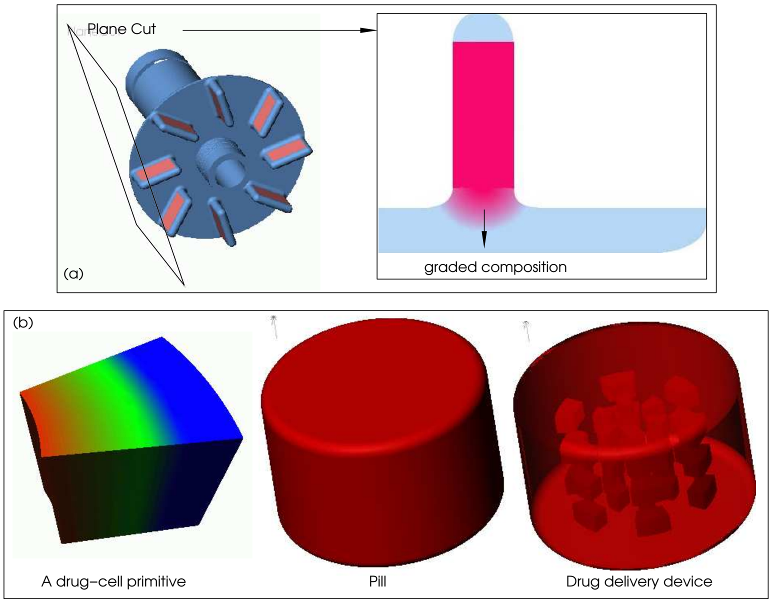


Figure 3: Conceptual illustration of parts created through LCC features

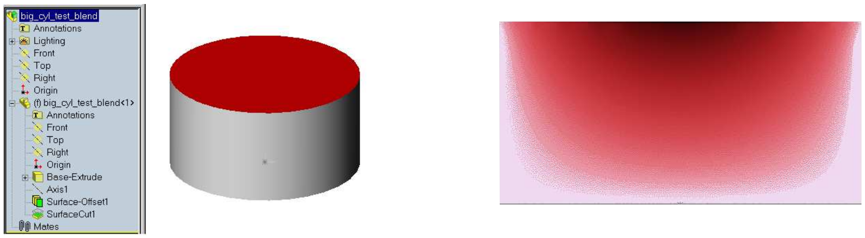


Figure 4: Top surface of the cylinder is an LCC surface with composition as function of distance to its axis

3.1.2 Material composition functions

A *feature-based design* system provides a convenient environment for parametric design of composition because it carries rich information at different levels of abstraction. The scheme is to specify the composition as a function of one variable or several variables. These variables are mapped from any interior point to user specified form features in the model, e.g., to specify composition normal to a surface feature of a part. With such a scheme, when the reference form features are edited, the composition variables are either automatically reevaluated or the user is prompted to change a design for the composition function. Composition function of dependent LCC features can be designed from that of the parent LCC features. For example, general blending of two disjoint LCC features can be designed through a smoothing operation based on Laplace's equation. Such a design is based on user-defined features, therefore it is also editable.

As shown in Figure 2, the composition function contains parameters such as the material it is related to, the parametrization method chosen by the user from the methods library, the identified reference features for parametrization, the identified value parameters and the subfunctions. The subfunctions include the functions that identify reference features from LCC geometric sub-feature or from user-defined feature through GUI, functions that map feature into value parameters, analytic functions that map value parameters to composition ratio of the specified material, etc. For example, in case the feature uses design according to distance from a reference feature, the subfunctions should contain the functions that identify and register the user-specified feature. The subfunctions should also contain the distance function from an input point to a reference feature. Finally, subfunctions should contain the analytic function that maps the distance into a composition ratio of the specified material.

3.2 Distance function based design of composition function

Distance is chosen as the parameter for the design of composition because the distance function is a continuous function, distance as a concept is very intuitive, and distance to different form features captures varieties of parameters. For example, Cartesian coordinates are distances to three orthogonal planes.

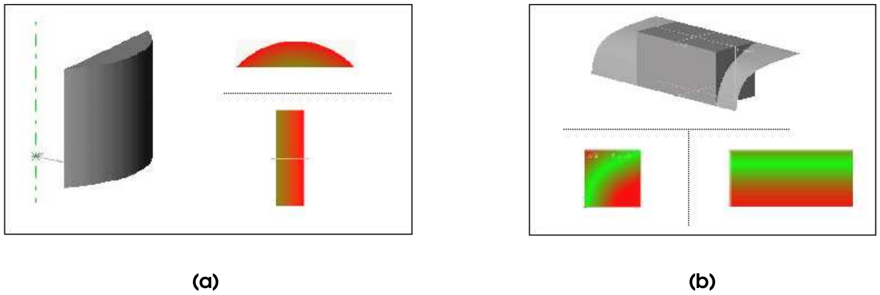


Figure 5: (a) Composition as function of distance to axis; (b) Composition as function of distance to different user defined surface

3.2.1 Single distance based composition function

This method computes composition in a single LCC feature as a single analytic function that uses distance as its independent variable. The distance is the minimum distance from an interior point of the LCC feature to a user selected form feature in the model. The analytic function is defined by the user and applied within a distance value limit supplied by the user. Beyond the distance limit, the value defaults to a constant which allows the function to be continuous at the limit distance. The form feature can be any generated features, modifying features or datum features [21]. Examples in Figures 5 demonstrate such a design method using different form features. For example, in Figure 5-(a) an axis feature is used, while in Figure 5-(b) user-defined surface and sweep feature are used. Distance to simple features such as axis and plane can be easily evaluated, but in the case that the minimum distances are evaluated to free-form surfaces, the computation of distance can be burdensome. Efficient distance computation developed by Liu *et al.* [29] is used. The algorithm is based on preprocessing the geometry of the form features and the digital distance transform algorithm [5]. The method of preprocessing the geometry can be found in [29]. When the evaluation is done at sufficiently fine resolution for visualization or printing, the exact distance can be approximated with the Euclidean digital distance. The square of the Euclidean digital distance is defined as:

$$\begin{aligned} S_{i,j,k} &= min\{d_t((i,j,k),(p,q,r))^2; f_{p,q,r}=0, 1\le p\le L,\\ &\quad 1\le q\le M, 1\le r\le N\}\ ,\\ &= min\{(i-p)^2+(j-q)^2+(k-r)^2;\\ &\quad f_{p,q,r}=0, 1\le p\le L,\\ &\quad 1\le q\le M, 1\le r\le N\}\ , \qquad (1)\end{aligned}$$

where L, M, N are the number of voxels in i, j, k respectively, and $f_{p,q,r}$ is any voxel that intersects the geometry of the reference features, whose Euclidean digital distance is therefore 0. The algorithm developed in [43] is one of the fastest in the literature. The following formulas summarize the basic algorithm:

1. $g_{ijk} = min\{(i-x)^2; f_{xjk}=0, 1\le x\le L\}$,
2. $h_{ijk} = min\{g_{iyk}+(i-y)^2; 1\le y\le M\}$,
3. $s_{ijk} = min\{h_{ijz}+(i-z)^2; 1\le z\le N\}$.

The efficiency is improved by reducing the search areas in step 2 and step 3 calculating h_{ijk} and s_{ijk}. For example, during step 2, the search is limited in $n_j = (g_{ijk} - g_{i(j-1)k} - 1)/2$ for each index j. This number is the intersection of the curves $f_1(n) = g_{ijk} + n^2$ and $f_2(n) = g_{i(j-1)k} + (n+1)^2$. The analysis for Euclidean digital distance algorithm is given in a later section.

3.2.2 Multiple distance based composition function applied simultaneously

This method involves use of multiple distance function-based profiles to a single LCC feature simultaneously. Distance profiles are defined as shown in Figure 6-(a). With the upper and lower distance limits, each distance profile is only applied within a subdomain of the LCC feature. The composition feature of LCC feature is represented by a queue of single distance based features. The last applied profile is at the tail of the queue. The profile at the front of the queue is the default constant value applied to the whole domain represented with Ω. Thus the domain of the LCC feature can be represented as a binary subdivision tree with leaves which are quasi-disjoint subdomains as demonstrated in Figure 7. The root node of the tree has the domain of the whole volume of the LCC feature. Here in Figure 7 it is the whole bounded region of the rectangle. Each left node has the domain defined by the intersection of the parent node and the effective volume of the composition profile at that level. The right node has the domain defined by the difference. The effective volume of the composition profile is the point set such that from every point in the set the distance to the referenced feature is within the specific limit $d_l[d_u,\ d_p]$. For the example in Figure 6-(b), d_u is 0 for both profiles A and B. The composition at each left node (excluding the root) is defined as the weighted sum of that of the parent and the evaluated value of the composition profile at that level, which is expressed mathematically as

$$Comp_{left} = Comp_{parent}\cdot\frac{d_2}{d_1+d_2} +$$

$$Comp_{current\_profile\_in\_queue} \cdot \frac{d_1}{d_1 + d_2} \cdot \qquad (2)$$

where d_1 represents the distance from a point located in the volume of the left node to the right node, d_2 represents the distance from a point located in the volume of the left node to the domain defined by the difference between the current profile and the parent. One special case is that when the domain of a left node completely enclose the domain of the profile, the distance d_2 is equal to zero. The composition at each right node inherits the composition of its parent node. Figure 6 demonstrates how this design method works. The rectangular shape represents an LCC feature, which has two distance-based composition profiles applied to two surface features A and B (A is the one with bold line) on the LCC feature. The whole region Ω of the LCC feature is then subdivided into 4 subdomains which are the leaf nodes in the tree.

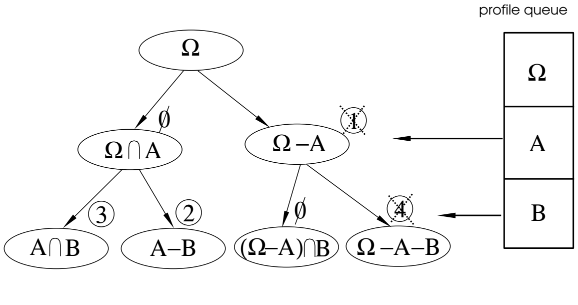


Figure 7: Subdivision tree underlying the multiple profile design method

As in the design method of single distance-based profile, we use the digital distance transform for the evaluation of distance functions. For each distance profile we assign a buffer for the distance transform. For efficient evaluation for visualization and printing, we keep a buffer $\mathcal{B}$ for the whole LCC feature. All these buffers are 3D arrays. With the use of the information in these buffers and the definition parameters such as the distance limits for the distance profiles, we can build up the above described binary tree. On each right node in the tree, we store a list of voxels that define the interface boundaries between the domain of this right node and the domain of its sibling left node in the tree. And on each left node, we store a list of voxels that define the interface boundaries between the effective volume of the current distance profile and the domain of this left node. The algorithm for searching for these voxels is based on the Euclidean distance transform [43] buffers for each composition profile and the information stored for each voxel in buffer $\mathcal{B}$ that maps each voxel to a node in the binary tree. These two lists are used for the evaluation of d_1 and d_2. For the example in Figure 6, the voxel lists are stored as illustrated with numbers in circles in Figure 7. When the neighboring left node has an empty list, it is not necessary to store the list of the right node.

3.3 Laplace's equation based blending design

We use Laplace's equation to compute the blending of the material composition from the boundary conditions motivated by its use in surface design. Laplace's equation has been used extensively in smooth surface design with few parameters, as in Bloor and Wilson [4]. Although we present only the constant coefficient Laplace equation based blending, the same method can be applied to more general elliptic partial differential equation problems. Qian and Dutta [37] presented a related diffusion-based design method for heterogeneous material turbine blades.

3.3.1 Setting of the boundary conditions

Users can intuitively assign boundary conditions for blendings by using the LCC surface features. The procedure is to select one or several surface features in the model and assign some composition design profile to apply on them, then select the domain to blend the LCC surface features, and the system will set the boundary conditions automatically. In the case that the blending is between several LCC volumes, the boundary conditions (boundary composition value or its normal derivative) are derived from those LCC volumes automatically. If all the involved geometric features are parametrizable, the users can edit the parameters affecting the Laplace's equation based blending.

3.3.2 Solving Laplace's equation with the Boundary Element Method

We employed the Boundary Element Method (BEM) [32, 6], which utilizes the second form of Green's theorem to express the potential function in the domain by an integral representation involving the potential function and its normal directional derivative on the boundary and the fundamental solution of Laplace's equation.

If ϕ and ψ are scalar functions of position defined on a region D bounded by a surface Γ (see Figure 8-(a)), the second form of Green's theorem [32] is given by

$$\int_D (\phi(\mathbf{z})\nabla^2\psi(\mathbf{x},\mathbf{z}) - \psi(\mathbf{x},\mathbf{y})\nabla^2\phi(\mathbf{z}))dv(\mathbf{z})$$
$$= \int_\Gamma \left(\phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})\right) ds(\mathbf{y}) \ , \qquad (3)$$

where $\mathbf{x}, \mathbf{z} \in D$ and $\mathbf{y} \in \Gamma$.

If we assume that the function ϕ corresponds to the variable in question and ψ corresponds to the fundamental solution of the Laplace's equation, they satisfy

$$\nabla^2\phi(\mathbf{z}) = 0 \qquad \mathbf{z} \in D \ , \quad (4)$$
$$\nabla^2\psi(\mathbf{x},\mathbf{z}) + \Delta(\mathbf{x},\mathbf{z}) = 0, \quad \Delta(\mathbf{x},\mathbf{z}) = 0, \mathbf{x} \neq \mathbf{z}$$
$$\mathbf{x}, \mathbf{z} \in D \ , \qquad (5)$$

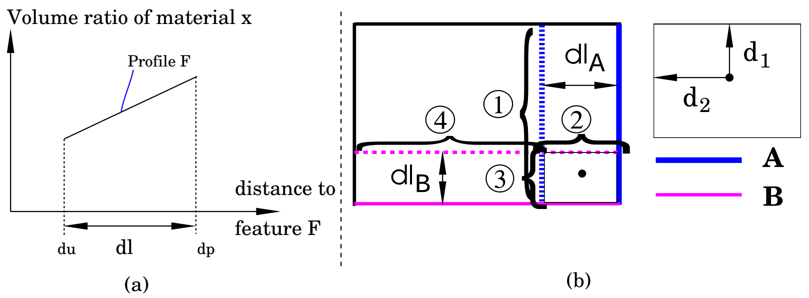


Figure 6: Design method for applying multiple distance based profiles

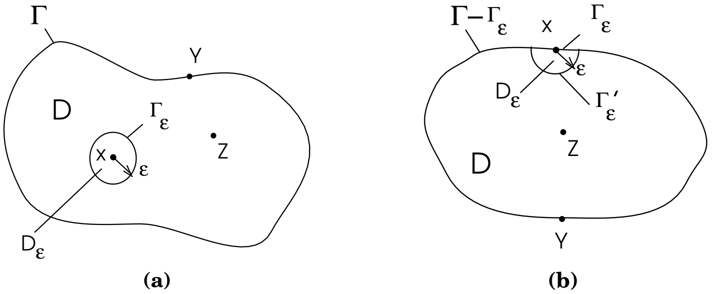


Figure 8: Domain type: (a) Point x is in the interior of the original domain D; (b) Point x is on the original boundary Γ.

where $\Delta(\mathbf{x},\mathbf{z})$ is Dirac's delta function, which plays a role of a unit source applied at point $\mathbf{x}$. The solution to (5) is given by

$$\psi = \frac{1}{4\pi r} \ , \tag{6}$$

where r is a radial distance between the unit source at $\mathbf{x}$ and the point $\mathbf{z}$ that are of interest. Since ψ has a singularity at $\mathbf{x}$, we need to isolate this point to avoid integration through a singularity. As in Figure 8-(a), if we enclose the domain D_ϵ, which includes point $\mathbf{x}$ by a sphere Γ_ϵ of radius ϵ, the new integration domain will become $D - D_\epsilon$ with boundary $\Gamma + \Gamma_\epsilon$. The result is obtained by considering the limit as ϵ tends to zero. Consequently the second form of Green's theorem in equation (3) can be rewritten as

$$\begin{aligned}
&\lim_{\epsilon\to 0}\int_{D-D_\epsilon} \phi(\mathbf{z})\nabla^2\psi(\mathbf{x},\mathbf{z})dv(\mathbf{z}) - \\
&\lim_{\epsilon\to 0}\int_{D-D_\epsilon} \psi(\mathbf{x},\mathbf{z})\nabla^2\phi(\mathbf{z})dv(\mathbf{z}) \\
&= \int_\Gamma \left(\phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})\right) ds(\mathbf{y}) + \\
&\lim_{\epsilon\to 0}\int_{\Gamma_\epsilon} \left(\phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})\right) ds(\mathbf{y}) \ ,
\end{aligned} \tag{7}$$

where $\mathbf{z} \in D - D_\epsilon$, $\mathbf{x} \in D_\epsilon$, $\mathbf{y} \in \Gamma$ or Γ_ϵ. Using equation (5), the left hand side of equation (7) is zero. The limit of the first part of the singularity integral of the right hand side converges to $\phi(\mathbf{x})$, while the second tends to zero. The limits of these singular integrals exist independent of how ϵ goes to zero, and the singularity is said to be *weak*. Rewriting of (7) leads us to the base equation of the BEM:

$$\phi(\mathbf{x}) = \int_\Gamma \left(\psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y}) - \phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y})\right) ds(\mathbf{y}) \ , \tag{8}$$

where $\mathbf{x} \in D$ and $\mathbf{y} \in \Gamma$. This base equation tells us that at any point $\mathbf{x}$ inside the domain D, the potential function $\phi(\mathbf{x})$ can be obtained from the the potential $\phi(\mathbf{y})$ and its derivative $\frac{\partial\phi}{\partial n}(\mathbf{y})$ on the boundary. However $\phi(\mathbf{y})$, $\frac{\partial\phi}{\partial n}(\mathbf{y})$ are initially not known and need to be computed, which is the main task of BEM.

To obtain the unknowns $\phi(\mathbf{y})$, $\frac{\partial\phi}{\partial n}(\mathbf{y})$ we start from (3) again, but this time $\mathbf{x}$ is on the boundary instead of being inside the domain. Similar to the case when $\mathbf{x}$ is inside the domain, the left hand side is zero and we have

$$\begin{aligned}
0 = &\lim_{\epsilon\to 0}\int_{\Gamma-\Gamma_\epsilon} \left(\phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})\right) ds(\mathbf{y}) + \\
&\lim_{\epsilon\to 0}\int_{\Gamma'_\epsilon} \left(\phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})\right) ds(\mathbf{y}) \ ,
\end{aligned} \tag{9}$$

where $\mathbf{x} \in \Gamma_\epsilon$ and $\mathbf{y} \in \Gamma - \Gamma_\epsilon$ or Γ'_ϵ (see Figure 8-(b)). The limit of the first part of the second singularity integral of the right hand side converges to $\frac{\alpha(\mathbf{x})}{2\pi}\phi(\mathbf{x})$, where $\alpha(\mathbf{x})$ is the internal angle at point $\mathbf{x}$, while the second part tends to zero. For a smooth boundary $\alpha(\mathbf{x})$ is π. The first part of the first singularity integral in (9) can be evaluated in the sense of Cauchy Principal Value, as the limit is undefined unless ϵ satisfies certain conditions as it approaches zero, while the second part is a weak singularity and the limit exists independently of how ϵ approaches zero. Equation

(9) therefore reduces to

$$\frac{\alpha(\mathbf{x})}{2\pi}\phi(\mathbf{x}) + \mathrm{P}\!\int_\Gamma \phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) = \int_\Gamma \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})ds(\mathbf{y}) \ , \tag{10}$$

where $\mathbf{x},\mathbf{y} \in \Gamma$ and the symbol P denotes a principal value.

The boundary of the region Γ is discretized by triangular elements $\Gamma_T = \sum_{k=1}^{n} \Gamma_{T_k}$. The collocation points of the boundary integral equation are located at the centroid of the elements. We need to supply either the composition value ϕ or the normal derivative of ϕ, i.e.$\frac{\partial\phi}{\partial n}$, at each collocation point. Since the collocation points are taken at the centroid of the elements, the boundary is always smooth there having $\alpha(\mathbf{x})$ equals to π, and hence (11) at the centroid of the elements takes the form:

$$\frac{1}{2}\phi(\mathbf{x}) + \mathrm{P}\!\int_{\Gamma_T} \phi(\mathbf{y})\frac{\partial\psi}{\partial n}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) = \int_{\Gamma_T} \psi(\mathbf{x},\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})ds(\mathbf{y}) \ , \tag{11}$$

where $\mathbf{x},\mathbf{y} \in \Gamma_T$. At the collocation point $\mathbf{x} = \mathbf{x}_j$, Equation (11) becomes:

$$\frac{1}{2}\phi(\mathbf{x}_j) + \sum_{k=1}^{n}\phi(\mathbf{x}_k)\mathrm{P}\!\int_{\Gamma_{T_k}} \frac{\partial\psi}{\partial n}(\mathbf{x}_j,\mathbf{y})ds(\mathbf{y}) = \sum_{k=1}^{n}\frac{\partial\phi}{\partial n}(\mathbf{x}_k)\int_{\Gamma_{T_k}} \psi(\mathbf{x}_j,\mathbf{y})ds(\mathbf{y}) \ , \tag{12}$$

where $\mathbf{y} \in \Gamma_{T_k}$ and $\mathbf{x}_k$ represents the *k-th* nodal point. Equation (12) can now be written as the linear system

$$Ax = b \ , \tag{13}$$

where A is a nonsingular $n \times n$ matrix, and b is a vector of length n.

3.3.3 Solution methods for the linear system

The drawback of the BEM compared with the Finite Element Method (FEM) is that the matrix A is non-symmetric and fully occupied [6, 32]. Traditionally these linear systems were solved by the direct method, such as the Gauss elimination method due to its robustness. However its computational cost is proportional to n^3, where n is the size of the matrix, and becomes prohibitive when n is large. Recently, a number of efficient iterative methods were developed and are gaining in popularity. Among these newly developed iterative methods, we employed the Generalized Minimum Residual (GMRES) method which is useful for general non-symmetric matrices [27, 39]. If we denote the initial approximation by x_o, the corresponding residual of (13) can be written as $r_o = b - Ax_o$. Also let us denote the Krylov subspace of dimension m by

$$K_m(A, r_o) = span\{r_o, Ar_o, A^2r_o, \cdots, A^{m-1}r_o\} \ . \tag{14}$$

In GMRES the solution of (13) is approximated by

$$x_m = x_o + V_m y \tag{15}$$

where V_m is an orthonormal basis for the Krylov space of dimension m, and y is a vector of length m (typically m is small compared with n). The orthonormal basis V_m is constructed through Arnoldi's procedure which uses the Modified Gram-Schmidt orthogonalization algorithm [39]. And the vector y is determined so that the norm of the residual $r_m = b - Ax_m$ is minimized. It is guaranteed that the GMRES algorithm converges in at most n steps in exact arithmetic, however the algorithm becomes impractical when m is large because of the memory growth of the orthogonal basis V_m. The restarted GMRES overcomes this storage limitation by restarting the iteration after a chosen number of iterations.

When the condition number of the matrix A is large, the GMRES may suffer from slow rate of convergence and low accuracy. To overcome the efficiency and robustness of the iterative methods, *preconditioning* is introduced, where the linear system (13) is transformed into one that has the same solution, but has a smaller condition number. If a preconditioning matrix M approximates the matrix A in some way, then the matrix $M^{-1}A$ may be close to the identity matrix, and hence may have a smaller condition number. If we pre-multiply M^{-1}, the transformed linear system

$$M^{-1}Ax = M^{-1}b \ , \tag{16}$$

has the same solution as that of (13), and may perform better in efficiency and robustness. The preconditioning done in this way is called *left conditioning.* Of course, we can pre-multiply M to have

$$AM^{-1}y = b \ , \tag{17}$$

which is first solved for y, and then for the solution x

$$x = M^{-1}y \ . \tag{18}$$

In this case matrix M is called the right preconditioner. In this work we employed a symmetric successive overrelaxation (SSOR) left preconditioner [39, 27]

$$M = (D + \omega L)D^{-1}(D + \omega U) \ , \tag{19}$$

where D, L, U are the diagonal, strict lower triangular, strict upper triangular matrices of A, respectively, and ω is a parameter which takes the value between 0 and 2. We can also express the SSOR preconditioner as

$$\begin{aligned} M &= (I + \omega LD^{-1})(D + \omega U) \ , \\ &= (D + \omega L)(I + \omega D^{-1}U) \ . \end{aligned} \tag{20}$$

In the actual implementation, M is never inverted and the system $Mr = b - Ax$ is solved for the residual r by taking the advantage of triangular decomposition. If $M = (D +$

$\omega L)(I + \omega D^{-1}U)$, then we first solve the lower triangular system

$$(D + \omega L)y = b - Ax \ , \tag{21}$$

for y and then solve the upper triangular system

$$(I + \omega D^{-1}U)r = y \ , \tag{22}$$

for r. In the implementation, we used the value $\omega = 0.5$. The pseudocode for the GMRES algorithm with left preconditioner is given in the Appendix [39].

4 Complexity, accuracy and convergence analyses

4.1 Euclidean distance transform (EDT)

Complexity An analysis of the complexity of the EDT algorithm developed in [43] is summarized here. Given a 3D binary image with 0-voxels as the reference for distance calculation, the computation includes three parts, the time on computing g_{ijk} (T_1), the time on computing h_{ijk} (T_2) and the time on computing s_{ijk} (T_3). The time cost on the first step (T_1) is obviously $O(V)$, where $V = LMN$ is total number of voxels. The time cost on the second step is sensitive to the input, and therefore we give the upper bound here. The algorithm for the second step will scan for each voxel, therefore, it will cost at least $O(V)$, plus for each j index, the algorithm will loop for at most $(g_{ijk} - g_{i(j-1)k} - 1)/2$ steps. Therefore, for each fixed pair of i and k, the worst estimate for the extra steps is $\sum_j^M (g_{ijk} - g_{i(j-1)k} - 1)/2$ which is equal to $(g_M - g_1 - M + 1)/2$. The value g_M is in the worst case the square of the half dimension of the image along i direction, and it is $L^2/4$. Given there are $L\ N$ number of pair of i and k, the time cost for step 2 in worst case is $O(V) + O(L^2\ L\ N) = O(V + L^3\ N)$. Similarly, the time cost for step 3 in worst case is $O(V + L^3 M)$. If the dimension in each direction of the 3D image is the same, and denoted as N, the worst time cost for steps 2 and 3 is in the order of $O(N^4)$. The algorithm is sensitive to the input shape. In the following, we give two examples, one is a regular cone and the other is a cube. Using these two examples, we can generate the input images and after the first step. The images demonstrated in Figure 9 are the 2D images in $i - j$ planes that are in the middle of the two examples with respect to k axis. For the example of the cube, one can see that in the step 2 computation, mostly the extra iteration of each j is zero, because mostly $g_j = g_{j-1}$. The extra iterations only happen to the voxel next to 0-voxels. The number of such voxels is in the order of $O(N^2)$, and the extra iteration number for each such voxel is at most N. Therefore the extra iteration as a whole is of $O(N^3)$, then the time cost for step 2 is of $O(N^3)$, and similarly the time cost for step 3 is also of $O(N^3)$. Therefore, the time complexity of EDT on a cube is $O(N^3)$. For the example of cone, consider the array of voxels in j direction with both i and k at the center of the planar face of the cone, the extra iteration of all the voxels in this array is in the order of $O(N^2)$. For each array immediately surrounding the center array, the iteration is of $O[(N-1)^2]$. The number of such arrays that have the same number of iterations is linear to the radius relative to the axis. Therefore the total number of extra iterations can be estimated with the following sum:

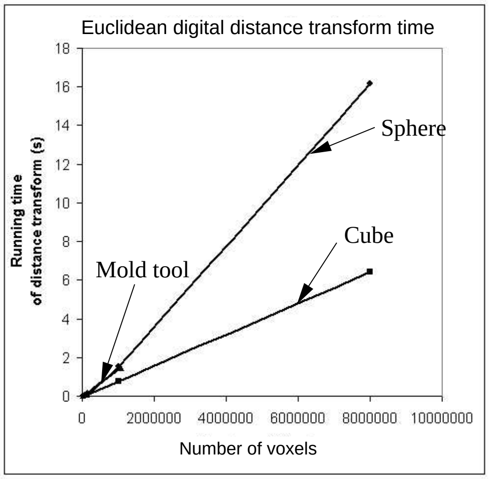


Figure 10: Time performance of Euclidean distance transform

$$N^2 + c\,(N-1)^2 + \cdots + c\,N\,1^2,$$

which is equal to $N^2 + c\ \sum_{i=0} N - 2(N-i)(i+1)^2$. This sum can be simplified to $O(N^4)$ which is equal to $O(V^{4/3})$. One can see that a sphere is a shape that is somewhat in between these two shapes. We tested experimentally for three examples: a cube, a sphere and a tool part which is the example in Figure 20. The time performance curves on these models are shown in Figure 10, one can see the time cost for cube is linear to the volume, the time cost for the tool part is also almost linear to the volume, but slower in terms of slope and the time cost for the sphere is about $O(V^{1.2})$.

Accuracy Given the definition of Euclidean digital distance, we can derive the maximum error of Euclidean digital distance as approximation to exact Euclidean distance between two points. For any two points located in voxels centered at X and Y respectively (Figure 11), with the assumption that the length of each voxel is δ, the square distance measured with Euclidean digital distance is $\overline{XY}^2 = [(|X_x - Y_x|)^2 + (|X_y - Y_y|)^2 + (|X_z - Y_z|)^2]\ \delta^2$. The maximum square distance between the two points is

$$\begin{aligned} \overline{ag}^2 &= (\delta + \delta|X_x - Y_x|)^2 + (\delta + \delta|X_y - Y_y|)^2 + \\ &\quad (\delta + \delta|X_z - Y_z|)^2 \ , \\ &= 3\delta^2 + 2\delta^2(|X_x - Y_x| + |X_y - Y_y| + \\ &\quad |X_z - Y_z|) + \overline{XY}^2 \ . \end{aligned} \tag{23}$$

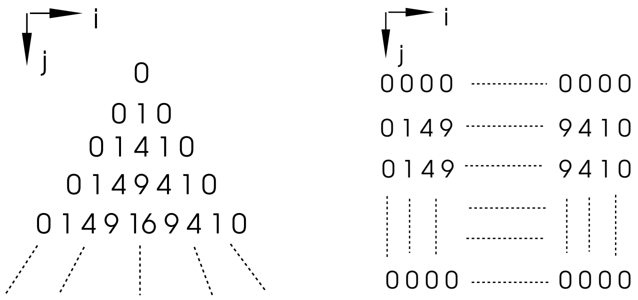


Figure 9: Cone and cube examples for analysis of EDT complexity

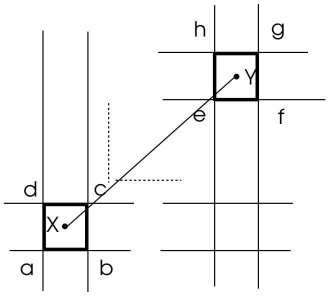


Figure 11: Accuracy of Euclidean digital distance

Therefore the maximum error in terms of square distance is:

$$\overline{ag}^2 - \overline{XY}^2 = 3\delta^2 + 2\delta^2(|X_x - Y_x| + |X_y - Y_y| + |X_z - Y_z|)$$

And

$$\begin{aligned} &|X_x - Y_x| + |X_y - Y_y| + |X_z - Y_z| \leq \\ &\sqrt{3} \cdot \sqrt{(|X_x - Y_x|)^2 + (|X_y - Y_y|)^2 + (|X_z - Y_z|)^2}\ , \\ &= \sqrt{3}\overline{XY}/\delta \qquad . \quad (24) \end{aligned}$$

Therefore, the maximum error in terms of square distance is $3\delta^2 + 2\sqrt{3}\delta\overline{XY}$

4.2 Laplace's equation based blending

Complexity From the description above on the BEM method, we can see that the computation time of the method is $T = T_1 + T_2$, where T_1 and T_2 are the times spent on setting up and solving the linear equation system. Here we omit the time on meshing the boundary into planar panels. Set the number of panels as n, which is equal to the dimension of the matrix for the equation system, then we have n^2 coefficients in the matrix for evaluation, and the time for each coefficient is constant, therefore time T_1 is asymptotically $O(n^2)$. The algorithm of the left-preconditioned GMRES is given in the Appendix. We have used the GMRES algorithm developed by Frayssé *et al.* [19]. The performance of the GMRES algorithm depends on both the number of boundary elements (n) and the number of iterations m of GMRES. Line 1 of the Algorithm consists of matrix-vector multiplication, forward substitution (21) and back substitution (22) of a triangular system. Since matrix-vector multiplication takes n^2 multiplications and additions, and each substitution takes $\frac{n^2}{2}$ arithmetic operations, in total Line 1 costs $2n^2$. Line 4 also involves matrix-vector multiplication, forward substitution and back substitution of a triangular system for each j step, resulting in $2mn^2$ arithmetic operations for m iterations. Line 6 is an inner vector product and Line 7 is constant-vector multiplication and a vector subtraction where both lines are inside the nested for-loops. Therefore the two lines take $m(m+1)n$ arithmetic operations. Line 14 involves transformation of the Hessenberg matrix into upper triangular form by using Givens rotation matrix [27] which costs $\frac{m(m+1)(m-1)}{3}$, and solving the resulting triangular system which can be solved in $\frac{m(m+1)}{2}$. Consequently the GMRES algorithm performance time T_2 is of $O(mn^2 + m^2n + m^3)$. In conclusion, the time cost on solving the blending problem with Laplace's equation is $O(n^2) + O(mn^2 + m^2n + m^3)$. Because in most cases m is small compared with n, the time complexity is almost $O(n^2)$. Figure 12 gives the experimental result on a model called "Sample_split", which confirms our analysis. The computation time with the LU decomposition (LU) method, which is one of the direct methods for solving linear equation system, is of $O(n^3)$. While the computation time with GMRES is of $O(n^2)$.

Convergence Studies in BEM methods have shown that the convergence rate of the constant panel method is of $O(n^{-2})$ [52]. We tested the example "Cube" with boundary conditions such as with one face out of the 6 faces at constant value 1 and the parallel one to it set at value 0.5 and the rest of the 4 faces at constant 0. Comparing with theoretical solution we have the result as in Figure 13 which confirms the quadratic convergence. Figure 14 shows the convergence test on the example in Figure 23. With the number of panels equal to 1000, the relative error of the solution is already less than 2% when the linear system solution is precisely computed. We also tested the convergence of the GMRES method with and without left pre-conditioning. Figure 16 shows the result on a "Mug" example that is shown in Figure 15. The inner surfaces of the "Mug" is designed as an LCC surface with material composition of constant value 0. The outer surfaces of the "Mug" is designed as an LCC

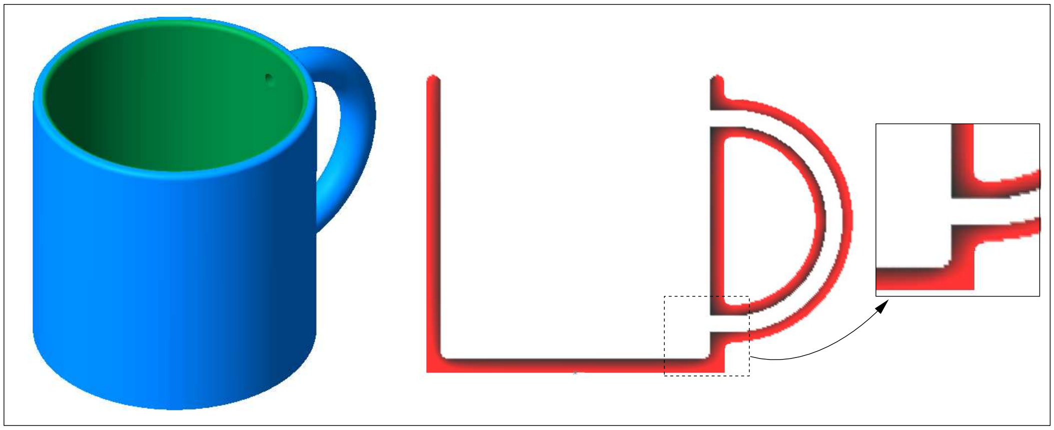


Figure 15: The Mug example with composition of Laplace's equation based blending between its outer surfaces and inner surfaces

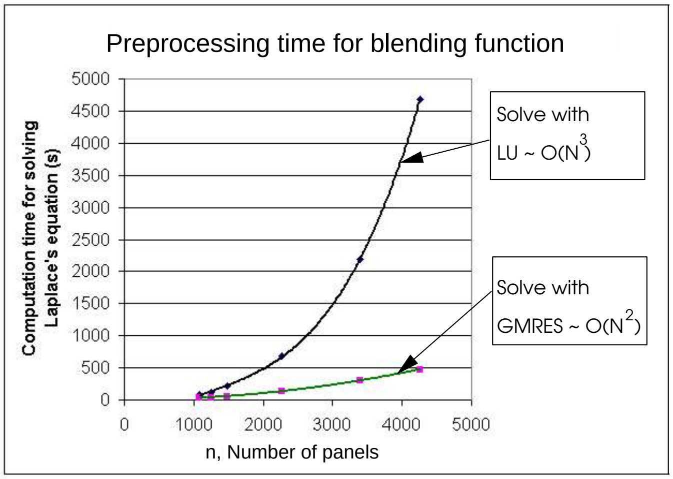


Figure 12: Computation time of the blending function on example "Sample_split"

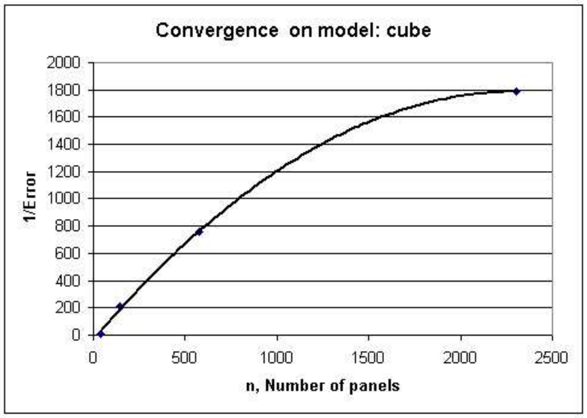


Figure 13: Convergence on example "Cube"

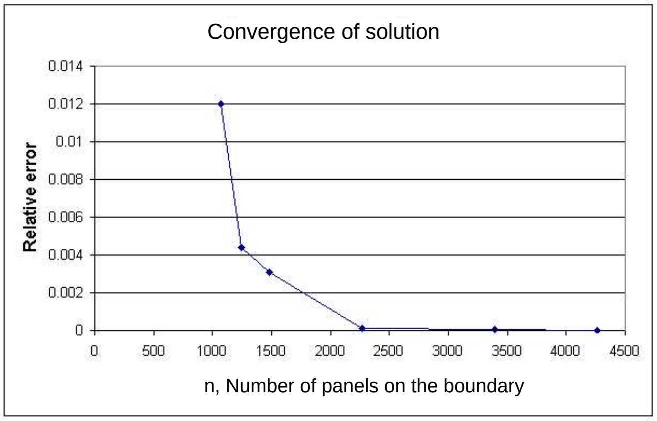


Figure 14: Convergence on example "Sample_split"

surface with constant composition value 1. And the composition for the solid volume is a Laplace's equation based blending. The matrix generated from this example has a large condition number which is 4.1×10^7 compared with other examples, for example the condition number for the example of Figure 24-(b) is only 7.6×10^4. Without preconditioning, the solution will not converge if a solution better than the resolution of 10^{-3} is desired. With preconditioning, the convergence rate of GMRES is significantly improved. For the example of Figure 24-(b), we tested for different densities of discretizations, and Figure 17 shows a significant reduction in the number of iterations when solving the system with GMRES method using left pre-conditioning.

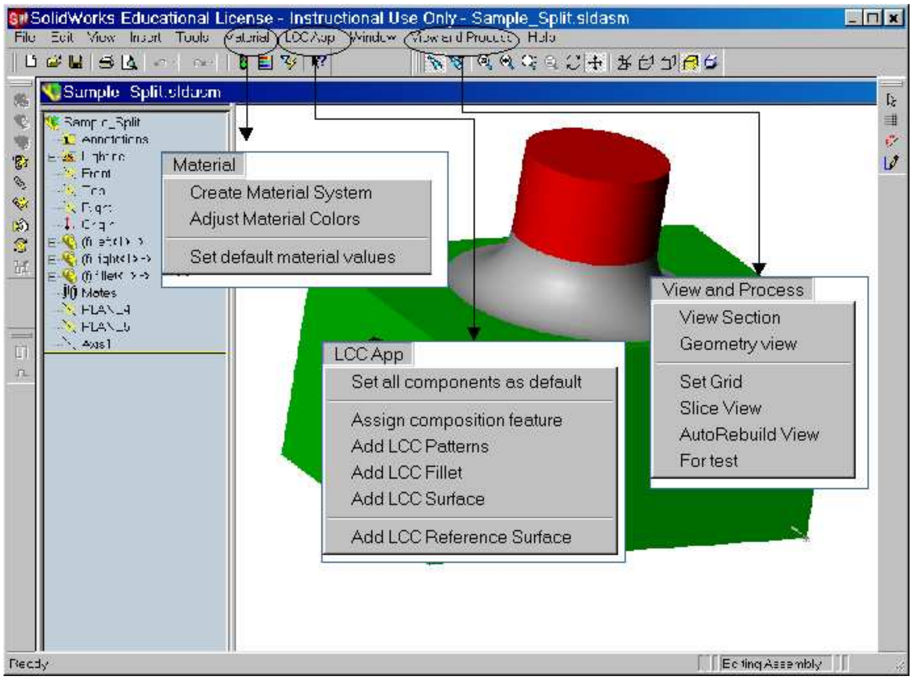


Figure 18: Graphical User Interface

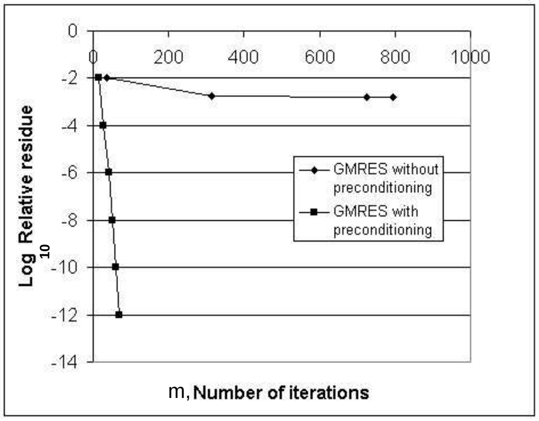


Figure 16: Convergence of GMRES solver with and without preconditioning

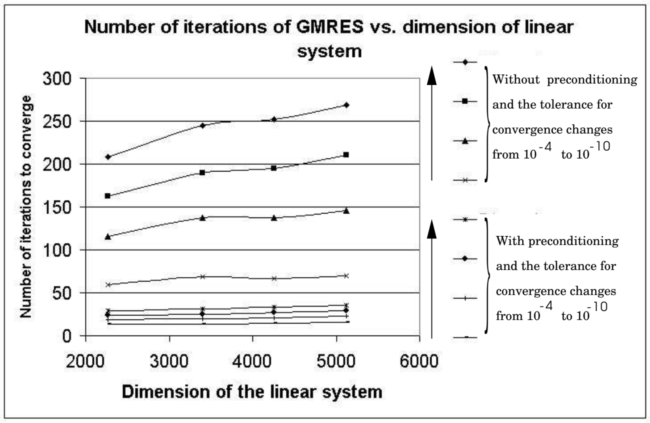


Figure 17: Number of iterations of GMRES solver with and without preconditioning

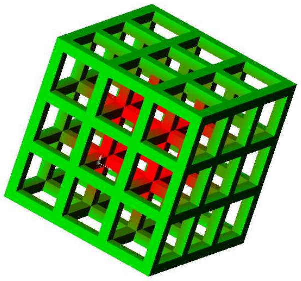


Figure 19: Scaffold for tissue engineering

5 System implementation and examples

A prototype system that includes all the design methods described above is implemented on an Intel Pentium III PC rated at 1GHz. Our system is written in C++ and integrated with SolidWorks [49] system via its API modules, forming a unified solid modeler for heterogeneous objects. Figure 18 shows the user interface of this prototype system. In terms of modules, there exists a material system module, a design module and a visualization and processing module. The material module allows the user to set up an array of materials and assign for each material some property, i.e. the color code. The Design module includes setting default material to all components in the assembly, designing an LCC composition feature for a component or surface in the feature tree of the SolidWorks system, designing an LCC pattern to a component pattern in the feature tree and designing an LCC fillet for a component in the assembly. The composition feature can be designed with any of the methods described previously. The visualization module includes visualization of the cross-section of the LCC object on any user defined plane, visualization of the outer boundary of the object with the material composition color-coded, and processing of the LCC object at user input grid into a data file for lower level postprocessing. The user interface for editing of LCC features is located in the right mouse button pop-up menu for each component or surface feature in the assembly. In the following, several examples are presented.

Tissue scaffold with distance based composition function: Figure 19 shows a tissue scaffold that is assigned a composition profile which is a function of the distance to side planar faces which bound the component (which are defined as the base feature). Such a design can promote the growth of tissue into the scaffold.

Tooling part design: using distance function from surface features: The example in Figure 20 demonstrates how the composition as a function of distance to different surface features is applied to the tooling part. In Figure 20-(b), a material composition profile is assigned as a function of distance to the cooling channel (a sweep feature geometrically) in the tool. Such a design may achieve the local control of porosity in the part and potentially improve the efficiency in tool cooling. In Figure 20-(c), the part is assigned a composition profile that is a function of distance to the outer boundary of the part (exclude the cooling channel). The boundary is composed of a series of features that the user used when designing the geometry. This design can be used to control the wear resistance of the tool where near the surface hard phases such as TiC can be printed. In Figure 20-(d), the tool is assigned two distance based composition profiles, one to the well, dome and the fillets between them, another to the cooling channel. Such a design can facilitate multiple design purposes, i.e. wear resistance control with the first profile and local porosity control with the second profile.

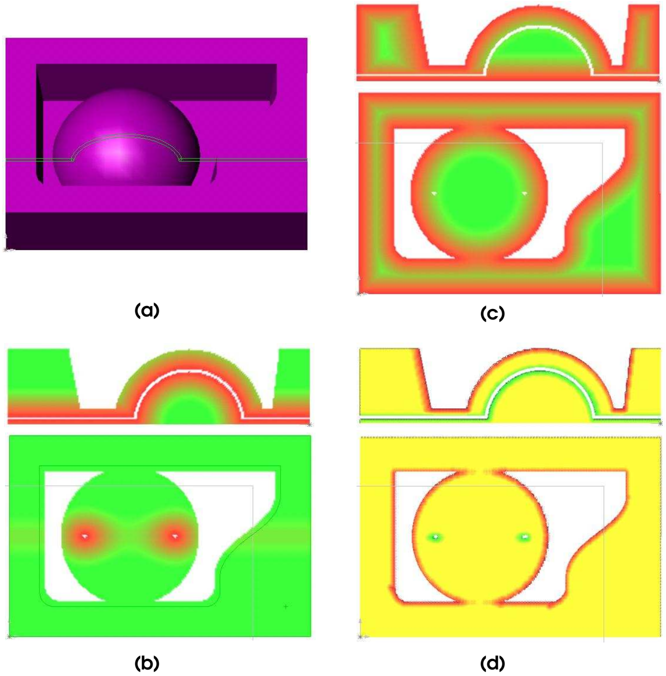


Figure 20: Example of tool part design using distance to features method

Design of multiple overlapping distance function based composition profiles: The example in Figure 21 demonstrates the design result of overlapping composition profiles. Here there are two geometric surface features in the part, the rectangular base and the cylindrical extrusion. Suppose the user wants to modify the compositions near the two surface features with two different distance based profiles, then there is an overlap similar to the example in Section 3.2.2. Here the default composition of the part is 100% yellow, the profile applied to the base grades from 100% red to 100% yellow into the part in the normal direction from the base feature, and the profile applied to the cylindrical extrusion grades from 100% green to 100% yellow into the part in the normal direction from the cylindrical extrusion feature. A design like this allows smooth change of material volume ratio between different profiles.

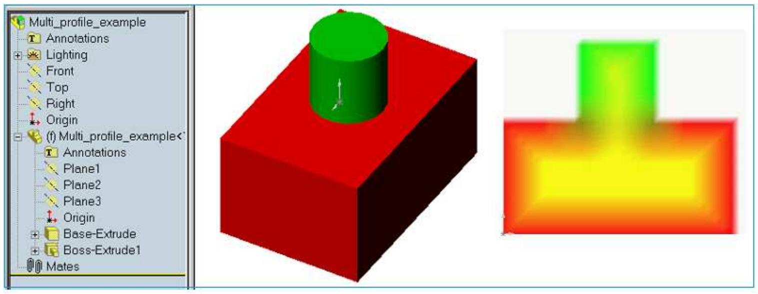


Figure 21: Example of multiple overlapping distance function based composition profiles

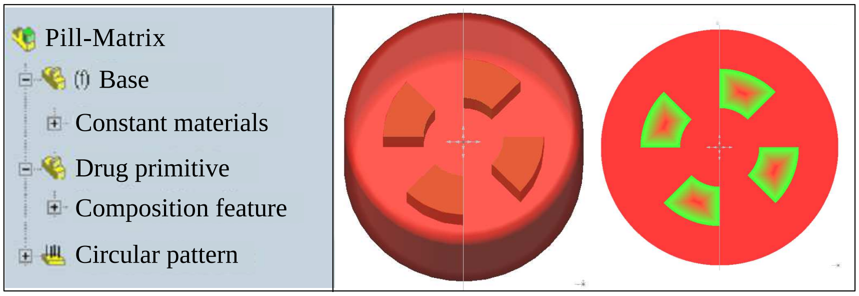


Figure 22: Example of pill matrix with LCC pattern

Pill matrix with LCC pattern The example in Figure 22 shows the design of LCC pattern for structured repeat of a particular LCC feature. Here the original pill is inserted an LCC component which was assigned a distance based composition profile with respect to the boundary of that component. The boundary again is the series of features the designer used in design of the geometry of the component. Then the component is circularly patterned based on user input parameters. Then the original pill was cut with these inserted LCC components. This design may be used to achieve certain desired functional delivery of drugs.

Design of material fillet using Laplace's equation based blending: The example in Figure 23 shows the design of material fillet for adjacent components. The smooth blending is based on solving Laplace's equation with boundary conditions. Here the example is an assembly of three components A, B and C, where the components are derived by cutting a single part with parametrized surfaces (two spherical surfaces). The user can assign to component A and C as separate LCC features and then assign material fillet on component B. Then the automatic blending solving as described in Section 3.3 is done by our system. Using such a design method, smooth transition between LCC volumes is achieved which may provide better material property than is otherwise achievable.

Editing of geometric feature and composition feature simultaneously: Figure 24 shows the simultaneous editing of geometric feature and composition feature with this system. The example in Figure 24-(a) shows editing on an LCC feature that is assigned two different distance based composition profiles. The example in Figure 24-(b) shows editing of the volume fillet of adjacent LCC components.

6 Conclusions

The major barrier to the wide-spread exploration of the potential of LCC in SFF is due to the lack of electronic representations and design tools for objects with LCC. Most CAD research has focused on the representation of 3D geometry of homogeneous objects, on methods and tools for designers to interact with these representations at a high level, and on derivation of machine specific instructions for machining. Current approaches proposed for modeling LCC objects are awkward in editing geometric and material composition information simultaneously. In effect, they permit sequential editing (i.e., first of geometry and then composition), which is not flexible and limits the designer's options. Current LCC models are also limited to low level data and operators and do not allow for the symbolic representation of the designer's intent with respect to composition. Also as such, design changes cannot be efficiently propagated. In order to address these limitations, our proposed approach builds on the concept of feature-based design (FBD) and extends it from a geometric domain to simultaneous material and geometric editing of features. We identify and formalize the concept of LCC features. The classes of LCC features include LCC features that are based on volume, transition, pattern, and (user-defined) surface features. Methods for LCC feature creation and editing are developed. Specifically, material composition functions such as functions parametrized

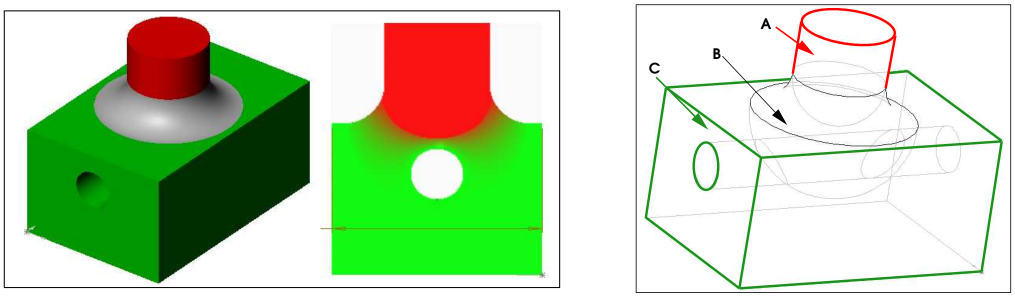
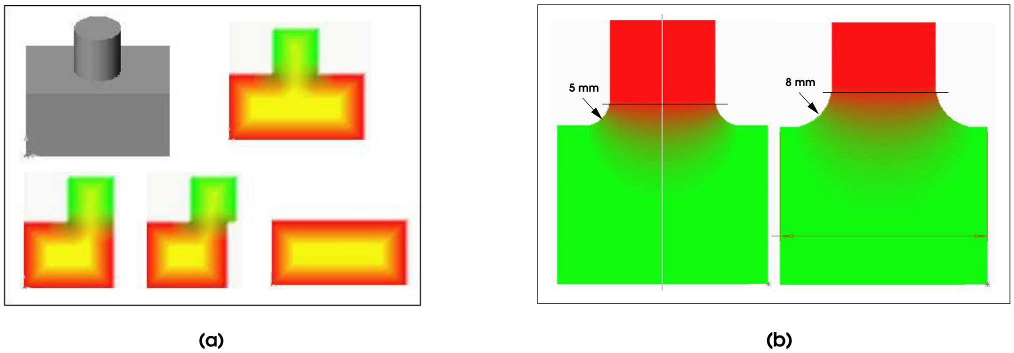


Figure 23: Example of material fillet using Laplace's equation based blending



(a)

(b)

Figure 24: (a) Edit a multi-profile design; (b) Edit a volume fillet design

with respect to distance or distances to user-defined geometric features; and functions that use Laplace's equation to blend smoothly various boundary conditions including values and gradients of the material composition on the boundaries are developed. The Euclidean digital distance transform and the Boundary Element Method are utilized and developed for the efficient computation of composition functions. In addition, the General Minimization of Residual Method is employed as an appropriate iterative method for solving the resulting linear equation system. Theoretical and experimental complexity, accuracy and convergence analyses are presented as well.

With such a feature-based scheme the efficient and robust evaluation of an LCC object is done at different levels of resolution for both visualization and fabrication purposes. An unevaluated exact representation for the geometry and composition is maintained for as long as possible along the information pathway. Therefore, a high level codification of the design useful for data exchange in general setting not associated with a specific SFF process is provided.

In this paper, we gave examples that demonstrate the usefulness of such a system in exploring the potential applications in Solid Freeform Fabrication with local composition control. Examples include tissue scaffold, tool part with local control, drug delivery device, GRIN lense, material fillet,etc.

Appendix

Algorithm: Left-preconditioned GMRES method [39] with x_o an initial guess.

```
1.  Solve $r_o$ from $Mr_o = b - Ax_o$
2.  $v_1 = r_o/\beta$ where $\beta = ||r_o||_2$
3.  for $j = 1, \cdots, m$
4.      Solve $w$ from $Mw = Av_j$
5.      for $i = 1, \cdots, j$
6.          $h_{i,j} = (w, v_j)$
7.          $w = w - h_{i,j}v_i$
8.      end
9.      $h_{j+1,j} = ||w||_2$
10.     if $h_{j+1,j} = 0$ set $m = j$ and go to 13
11.     $v_{j+1} = w/h_{j+1,j}$
12. end
13. Define $V_m = [v_1, v_2, \cdots, v_m]$,
    $\bar{H}_m = [h_{i,j}]_{1 \leq i \leq j+1; 1 \leq j \leq m}$
14. Compute $y_m$ which minimizes $||\beta e_1 - \bar{H}_m y||_2$
```

15. $x_m = x_o + V_m y_m$

In other words the solution in the *i-th* iterate of GMRES is constructed as

$$x^{(i)} = x_o + y_1 v^{(1)} + \cdots + y_i v^{(i)} \tag{25}$$

where y_i is determined to minimize the residual norm $||b - Ax^{(i)}||$.

Acknowledgements

This work was funded in part by NSF DMI-0100194 and ONR N00014-01-1-1065. We are also grateful to Dr. Yuming Liu for providing a computer code for the BEM method and Dr. V. Frayssé et al. for the GMRES codes.

References

[1] R. Bidarra, K. J. Kraker, and W. F. Bronsvoort. Representation and management of feature information in a cellular model, *Computer-Aided Design*, 30(4):301–313, 1998.

[2] R. Bidarra and W. F. Bronsvoort. Semantic feature modeling, *Computer-Aided Design*, 32(3):201–225, March 2000.

[3] A. Biswas, V. Shapiro and I. Tsukanov. Heterogeneous Material Modeling with Distance Fields, *Computer-Aided Geometric Design*, 2003. To appear.

[4] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free-form surfaces: 91787, *Computer-Aided Design*, 22(4):202–212, 1990.

[5] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27: 321-345, 1984.

[6] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. Boundary Element Techniques. *Springer-Verlag*, 1984.

[7] G. Brunetti and A. Stork. Product-centered intuitive 3D interaction for feature-based parametric assembly modelling. *Proceedings of the ProSTEP Science Days'98,* 17-18, 1998.

[8] G. Brunetti and B. Golob. A feature-based approach towards an integrated product model including conceptual design information. *Computer-Aided Design*, 32:877–887, 2000.

[9] V. Capoyleas, X. Chen, and C. M. Hoffmann. Generic naming in generative, constraint-based design, *Computer-Aided Design*, 28(1):17–26, 1996.

[10] V. Chandru, S. Manohar, and C. E. Prakash. Voxel-based modeling for layered manufacturing. *IEEE Computer Graphics and Applications*, 15(6):42–47, November 1995.

[11] X. Chen and C. M. Hoffmann. Toward feature attachment, *Computer-Aided Design*, 27(11):675–702, 1995.

[12] X. Chen and C. M. Hoffmann. On editability of feature-based design, *Computer-Aided Design*, 27(12):905–914, 1995.

[13] W. Cho, E. M. Sachs, N. M. Patrikalakis, H. Liu, H. Wu, T. R. Jackson, C. C. Stratton, J. Serdy, M. J. Cima, and R. Resnick. Methods for distributed design and fabrication of parts with local composition control, to appear in *Proceedings of the 2001 NSF Design and Manufacturing Grantees Conference*, Tampa, FL, USA, January 2001.

[14] W. Cho, E. M. Sachs, N. M. Patrikalakis and D. E. Troxel. A Dithering Algorithm for Local Composition Control with Three-Dimensional Printing. *Computer-Aided Design*, 35(9):851-867. August 2003.

[15] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms,* MIT Press, Cambridge, MA, 1990.

[16] U. Cugini. Feature-based assembly for aeronautics. *CAD Tools and Algorithms for Product Design, P. Brunet, C. Hoffmann and D. Roller (Eds.)*, 31–46, Springer, 2000.

[17] H. Dedhia, V. Pherwani, J. Shah. Dynamic interfacing of applications to geometric modelers via neutral protocol. *Computer-Aided Design*, 29(12): 811-824, 1997.

[18] D. Deneux. Introduction to assembly features: an illustrated synthesis methodology. *Journal of Intelligent Manufacturing*, 10: 29-39, 1999.

[19] V. Frayssé, L. Giraud and S. Gratton. A Set of GMRES Routines for Real and Complex Arithmetics. *Technical report*, CERTACS, TR/PA/97/49, Toulousse, France. http://www.cerfacs.fr/algor/Softs

[20] C. M. Hoffmann and R. Joan-Arinyo. Erep-an editable high-level representation for geometric design and analysis, *Geometric Modeling for Product Realization, P. Wilson, M. Wozny and M. Pratt (Eds.)*, North-Holland, Amsterdam, pp. 129-164, 1993.

[21] C. M. Hoffmann and R. Joan-Arinyo. On user-defined features, *Computer-Aided Design*, 30(5):321-332, April 1998.

[22] W. van Holland and W. F. Bronsvoort. Assembly features in modeling and planning. *Robotics and Computer Integrated Manufacturing*, No. 16, pages 277-294, 2000.

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

[24] T. R. Jackson, W. Cho, N. M. Patrikalakis and E. M. Sachs. Memory analysis of solid model representations for heterogeneous objects. *ASME Transactions, Journal of Computing and Information Science in Engineering*, 2(1): 1-10, March 2002.

[25] W. E. Katstra, R. D. Palazzolo, C. W. Rowe, B. Giritlioglu, P. Teung, and M. J. Cima. Oral dosage forms fabricated by Three-Dimensional Printing^{TM}. *Journal of Controlled Release*, 66(1):1–9, May 2000.

[26] A. Kaufman, D. Cohen, and R. Yagel. Volume graphics. *Computer*, 26(7):51–64, July 1998.

[27] C. T. Kelley. Iterative Methods for Linear and Nonlinear Equations. *SIAM*, 1995.

[28] V. Kumar and D. Dutta. An approach to modeling and representation of heterogeneous objects. *ASME Transactions, Journal of Mechanical Design*, 120:659–667, December 1998.

[29] H. Liu, W. Cho, T. R. Jackson, N. M. Patrikalakis, and E. M. Sachs. Algorithms for design and interrogation of functionally gradient material objects, *Proceedings of 2000 ASME DETC/CIE, 26-th ASME Design Automation Conference,* September, 2000, Baltimore, Maryland, USA. p.141 and CDROM, NY:ASME, 2000.

[30] W. Martin and E. Cohen. Representation and extraction of volumetric attributes using trivariate splines: a mathematical framework. *Proceedings of the 6th ACM Symposium on Solid Modeling and Applications*, In D. C. Anderson and K. Lee (Eds.), pages 234-240, 2001.

[31] E. H. Nielsen, J. R. Nixon, and G. E. Zinsmeister. Capturing and using designer intent in a design with features system. *Proceedings Design Theory and Methodology*, D-E Vol. 31, ASME, pp. 95–102, 1991.

[32] R. Paris and J. Canas. Boundary Element Method. *Oxford University Press*, 1997.

[33] S. Park, R. H. Crawford and J. J. Beaman. Volumetric multi-texturing for functionally gradient material representation. *Proceedings of the 6th ACM Symposium on Solid Modeling and Applications*, In D. C. Anderson and K. Lee (Eds.), pages 216-224, 2001.

[34] J. Pegna and A. Safi. CAD modeling of multi-modal structures for free-form fabrication, 1998. In D. L. Bourell et al, editor, *Solid Freeform Fabrication Symposium*, Austin, Texas, August, 1998. The University of Texas.

[35] M. J. Pratt, A. D. Bhatt, D. Dutta, K. W. Lyons, L. Patil and R. D. Sriram. Progress towards an international standard for data transfer in rapid prototyping and layered manufacturing. *Computer-Aided Design*, 34(14): 1111-1121, December, 2002.

[36] M. J. Pratt. Synthesis of an optimal approach to form feature modeling, *Proceedings of the ASME 1988 Computers in Engineering Conference,* vol. 1, pp. 263–274, New York, 1988.

[37] X. Qian and D. Dutta. Design of heterogeneous turbine blade, *Computer-Aided Design*, 35: 319-329, 2003.

[38] J. R. Rossignac. Issues on feature-based editing and interrogation of solid models, *Computers and Graphics*, 14(2):149–172, 1990.

[39] Y. Saad. Iterative methods for sparse linear systems. *Thomson Publishing:PWS/ITP*, 1996.

[40] E. M. Sachs, E. Wylonis, S. Allen, M. J. Cima, and H. Guo. Production of injection molding tooling with conformal cooling channels using the three dimensional printing process, *Polymer Engineering Science*, 40(5):1232–1247, May 2000.

[41] E. M. Sachs, J. Haggerty, M. J. Cima, and P. Williams. Three-Dimensional Printing Techniques, U.S. Patent No. 5204055, April 20, 1993.

[42] E. M. Sachs, M. J. Cima, P. Williams, D. Brancazio, and J. Cornie. Three dimensional printing: rapid tooling and prototypes directly from a CAD model, *Journal of Engineering for Industry,* 114(4):481-488, November 1992.

[43] T. Saito and J. Toriwaki. New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with applications, *Pattern Recognition*, 27(11):1551–1565, 1994.

[44] O. W. Salomons, F. J. A. M. van Houten and H. J. J. Kals. Review of research in feature-based design, *Journal of Manufacturing Systems*, 12(2):113–132, 1993.

[45] J. Shah and M. Mäntylä. *Parametric and Feature-Based CAD/CAM*, John Wiley, Inc., 1995.

[46] J. Shah and M. T. Rogers. Assembly modeling as an extension of feature-based design, *Research in Engineering Design*, 5:218-237, December 1993.

[47] K. Shin and D. Dutta. Constructive representations for heterogeneous objects. *ASME Transactions, Journal of Computing and Information Science in Engineering*, 1(3): 205-217, September, 2001.

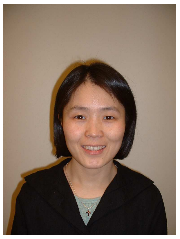
[48] Y. K. Siu and S. T. Tan. Source-based heterogeneous solid modeling. *Computer-Aided Design*, 34:41-55, 2002.

[49] SolidWorks^{TM} (http://www.solidworks.com/), Concord, MA.

[50] D. G. Ullman. The evolution of function and behaviour during mechanical design. *Proceedings Design Theory and Methodology*, D-E Vol. 53, ASME, pp. 91–103, 1993.

[51] C. Vasilis, X. Chen and C.M. Hoffmann. Generic naming in generative, constraint-based design. *Computer-Aided Design*, 28(1):17-26, 1996.

[52] M. Xue, H. Xu, Y. Liu and D.K.P. Yue. Computation of fully-nonlinear three-dimensional wave-wave and wave-body interactions. Part I: Dynamics of steep three-dimensional waves. *Journal of Fluid Mechanics*, 438: 11-39, 2001.



Ms. Hongye Liu is a Doctoral student at MIT. She received a B.E. in Precision Machinery and Instrumentation from the University of Science and Technology of China in 1993, and dual M.S. in Naval Architecture & Marine Engineering and Mechanical Engineering from MIT in 2000. From 1993 to 1997 she worked for the National Synchrotron Radiation Laboratory of China as an engineer.

Dr. Takashi Maekawa is a Professor of Mechanical Engineering at Yokohama National University in Japan since early 2003. Before joining YNU, he was a Principal Research Scientist at MIT. His research interests include computational and differential geometry, and Computer Aided Design and Manufacturing. He received a B.S. and an M.S. in Mechanical Engineering from Waseda University, Tokyo in 1976 and 1978, respectively, an Ocean Engineer's Degree from MIT in 1987, and Ph.D. in Computer Aided Design and Manufacturing from MIT in 1993. From 1978 to 1989 he worked for Bridgestone Corporation as a design and manufacturing engineer.

Dr. Nicholas M. Patrikalakis is the Kawasaki Professor of Engineering at MIT and co-director of the Design Laboratory (http://deslab.mit.edu). His research interests include applications of geometric modeling and software engineering in design, analysis and fabrication of complex systems.

Dr. Emanuel M. Sachs is the Fred Fort Flowers and Daniel Fort Flowers Professor of Mechanical Engineering and specializes in the design of manufacturing processes. He is a co-inventor of Three Dimensional Printing, a manufacturing process for the creation of 3D parts directly from a computer model in layers. 3D Printing is being commercialized in fields-of-use including ceramic molds for castings, direct metal tooling, end-use metal parts, medical devices, ceramic filters, appearance models, and electronic ceramic applications. Dr. Sachs is also the inventor of the "String Ribbon" crystal growth process used by Evergreen Solar, Inc. of Marlboro, MA to manufacture photovoltaic cells. Dr. Sachs is the author or co-author of more than 100 technical papers and is the inventor or co-inventor of more than 25 patents.

Dr. Wonjoon Cho is currently a research affiliate at MIT. His research interests include CAD/CAM, geometric modeling and interrogation, computational and differential geometry and solid freeform fabrication. He received a BS and an MS in naval architecture from Seoul National University in 1986 and 1988, an MS in ocean engineering from MIT in 1992, and a Ph.D. in computer-aided design and manufacturing from MIT in 1997.