Presented at the workshop on *Reasoning with Shapes in Design*, *AID'94*, Lausanne, Switzerland, August 15-18 1994

An algebraic approach to shape computation

Rudi Stouffs and Ramesh Krishnamurti

1 Motivation

There can be no argument that the use of computers in design is ratiocinative. Researchers and design practitioners have long been fascinated with question such as (i) what is design? (ii) how do we design? (iii) how can we use computers to support design without constraining either the process or the artifact ? and, more importantly, (iv) how can we use computers to 'liberate' the process of design ? To some extent, we attempt to address latter two questions.

We consider the computer as a tool for design and presentation, as a *means of communication* between the design professional and client, among professionals and during the design process. Proverbially, a drawing is worth a thousand words. In design, especially architecture, a drawing is worth much more. In architecture, we use sketches, plans, elevations, sections, isometries and perspectives as the means of communication. Drawings are essential to the design process. The wealth of information contained in drawings justifies the time consumed for their creation.

It is in the area of drafting and presentation that the computer chiefly shows its strengths: its capabilities for visualization are well known. In other areas, say in the early design phase, the use of computer lags far behind. It is imperative that the role and use of the computer in architectural design be clearly recognized.

Computers have long been hailed to offer speed of action and improve efficiency. As a result, most computer-aided design system is driven by the cut/copy/paste "economies of digital data" (Dagit, 1993). Many architects or architecture firms are tricked into CAD by unfounded claims of increase in productivity. Six months later they wonder what went wrong. Setting aside discussions on training and adaptation, there is a more powerful consideration. Efficiency does not necessarily improve effectiveness. Speed and accuracy are not synonymous with quality.

One of the main reasons for the use of computers in design should be in the potential to process and evaluate large amounts of information, providing the designer with a better basis for decision making, as well as the ability to explore new horizons in design.

We will not attempt to further add to the lengthy list of conceptions and misconceptions of 'computer-aided design.' We have used and will use the term CAD to refer to the software applications that currently go under that guise. Instead, we consider two aspects of design: (i) the conception of a design, and (ii) its representation (or visualization). While these two go hand in hand – it is hard to conceptualize a design without visualizing it in some form – some design activities can clearly be distinguished as belonging to one or the other. Drafting deals only with representation. On the other hand, the conception of a design is generally viewed as a cyclical – or spiral – process of creation, generation and evaluation. Currently, computers play a minor role in this.

The computer has had much greater impact on the representation of designs, primarily in drafting, but increasingly in other representational forms, such as renderings. In this respect the term *visualization* has been defined to encompass modeling, rendering and animation. However, this definition is most commonly employed in connection with the current 'state of the art' in CAD, namely, the production of "pretty pictures." As such, we prefer to refer to this as *presentation*, as it relates to 'finished' designs. We argue that visualization encompasses every stage in the design process. It has even greater potential in the early design stages, where it enables the explicit making of three-dimensional models in the thought process.

The visualization process can be extended to conceptual sketching and image editing (Charles and Brown, 1992). Visualization is not about pretty pictures in the same way that architecture is not about pretty buildings. The danger of fancy visualization is that it may hide a bad design. At the same time, visualization has the potential of being a powerful tool in the evaluation of designs. Large amounts of information can be successfully displayed in computer-generated images in a way that a human being – designer – can readily process. Indicators of building performance such as lighting and acoustics are obvious examples; urban analysis and design is another in which visualization can assist the designer.

The process of design is all too often procedure oriented. As a result, the benefits from CAD are mostly instrumental, not conceptual and CAD simply optimizes traditional, paper-based design techniques. The generation of a library of components for repetitive, industrial-like assemblies only perpetuates a methodology of architectural design that predates the introduction of computers into the profession. CAD has a definite potential for more, to enable an architecture of the information age, not limited by the constraints of translating the quality of space onto paper.

Plan-based design, two-dimensional representational systems were merely the result of the then available technology. Unless we redefine the purpose of CAD in the professional realm, as a consequence, more mediocre architects will be able to design more bad buildings, while more and more competent architects will turn to CAD for all the wrong reasons. Even in three-dimensional modeling software, boolean operations – union, intersection and difference – on primitive solids may be particularly suited to industrial-age standardized manufacturing, but does not reflect the information-age (Smulevich, 1993).

Instead, we should use the computer to model and visualize three-dimensional space in all its aspects and to create buildings that are not mere three-dimensional compositions of two-dimensional components. As Hoffer (1993) remarks:

> "Our current methods can be characterized as standardized fabrication methods where architects select products from catalogues. In the coming decade the degree of customization will increase. … As architects become more familiar with CAD and with its potential, they will develop a more sophisticated awareness of its capabilities and limitations. This sophistication will translate into the ability to design custom building elements with construction awareness and accuracy, employing cost-effective techniques and alternatives."

Frank Gehry's office makes use of CATIA, a 3D modeler designed for the aerospace industry, to model 'free-form' designs, while considering manufacturing and construction. With this new technology, Gehry can engage in formal – sculptural – explorations without being constrained by uncertainties about the methods and costs of building his designs. In contrast to other CAD software, CATIA represents surfaces by their mathematical formulation. This information can be readily made available to contractors and manufacturers. It enables Gehry to work directly with the craftsman, with the computer as the interpreter, to formalize and construct his designs.

We have explored an alternative approach, based on an algebraic model for shapes, that allows for shapes to be dealt with and manipulated in indeterminate ways. The algebraic approach to shape computation allows for an intuitive and powerful method of reasoning with shapes, in an interactive and generative design environment. Algebraic models for shapes have been adopted elsewhere in shape grammars (Stiny, 1980, 1986, 1991). In these models, a shape is specified as an element of an algebra that is ordered by a *part relation* and closed under the operations of sum, product and difference and the affine transformations. Fundamental to an algebraic model is that, under the part relation, any part of a shape is a shape and can be manipulated as such: thus, users can deal with shapes in indeterminate ways. This is quite distinct from the selection process in current CAD approaches where the only objects that can be selected correspond to those (prescribed minimal entities) that have been predefined by the data-structures.

2 Emergence

The part relation for shapes leads naturally to the concept of emergent shapes: A shape defines an infinite set of shapes, all part of the original shape, that emerge under the part relation. Emergent shapes are not originally envisioned as such, they only become explicit when manipulated as such. Computationally recognizing emergent shapes requires determining a transformation under which a specified similar shape is a part of the original shape. A shape rule constitutes a formal specification of shape recognition and subsequent manipulation. A shape rule has the form $a \rightarrow b$; a (the left hand side) specifies the similar shape to be recognized, b (the right hand side) specifies the manipulation leading to the resulting shape. Thus, shape rule application consists of replacing the emergent shape corresponding to a, under some allowable transformation, by b, under the same transformation. A shape grammar combines a set of (semantically related) shape rules into a formal rewriting system for producing a language of shapes.

The concept of emergent shapes is highly enticing to design search (Mitchell, 1993; Stiny, 1993); the specification of shape rules leads naturally to the generation and exploration of possible designs. However, the concept of search in this context is more fundamental to design than its generational form alone might imply. Any mutation of an object into another one, or parts thereof, whether as the result of a transformation or operation, constitutes an action of search.

A rule constitutes a particular compound operation or mutation, that is, a composition of a number of operations and/or transformations that is recognized as a new, single, operation and can be applied as such. Under this algebraic model, any such composition defines a valid mutation. Similarly, a grammar is merely a collection of rules or operations that yields a certain set of designs given an initial design. As such, the creation of a grammar is only a tool that allows a structuring of a set of operations that has proven its applicability to the creation of a certain set of objects, rather than a framework for generation.

Particular to a rule is that the transformation is not specified but is selected from a body of transformations, e.g., the similarity transformations, according to a (specified) constraint. For a shape rule, the constraint specifies that the object of mutation is a part of the given shape. The process of determining one (or all) transformation for which this constraint holds is termed subshape detection; it relies fundamentally on the algebraic model for shapes and the corresponding maximal element representation.

The maximal element representation for shapes (Krishnamurti, 1992) formalizes the part relation that underlies the algebraic model. This representation is particularly suited to answer the following two questions: Are two shapes identical? and Is one shape a part of another shape? Together, the algebraic model and maximal element representation define a representation scheme for geometric modeling that is both unique and ambiguous. The algebraic model is mathematically uniform for shapes of all kinds, including curved shapes, and applies to non-geometric elements or attributes as well. The proofs of these assertions can be found in Stouffs (1994).

3 Mixed dimensionality

The model provides a natural and intuitive framework for mixed-dimensional shapes. The creative process of design is not so concerned with the dimensionality of each and every individual, even if the final product is a composition of (purely) three-dimensional solid elements. Even in the evaluation of a design, an abstraction is often more valuable. Structurally, walls can be represented as plane segments with simple aliquot attributes; a true three-dimensional model would, unless appropriately approximated, needlessly complicate structural evaluation. In general, even a single component may be represented as different elements of mixed dimensionality, each element projecting information for a specific application.

Mixed-dimensional models have found recent support in solid modeling (Rossignac and Requicha, 1991; Gursoz et al., 1991). By adhering to Boolean set operations in a Euclidean space, these mixed-dimensional approaches come at the expense of intuition in operations and ease of conception. For example, the notion of solids touching is completely absent.

The algebraic model is based on a part relationship that can be freely defined as long as it constitutes a partial order relation. It is advantageous to define the part relation between elements of the same type, e.g., between elements of the same dimensionality. Elements of the same dimensionality belong to the same algebra. A shape may consist of more than one type of element, in which case it belongs to the algebra given by the Cartesian product of the algebras of its element types.

The algebraic model for shapes is independent of an underlying geometric framework. We distinguish the geometry of shapes from Euclidean geometry, instead, consider the Euclidean space to embed the elements in their final representation (Stouffs, 1994). An element of an algebra given by the Cartesian product of the algebras of its spatial element types is compositely embedded in a Cartesian product of Euclidean spaces. These may ultimately be visualized as either a single space or a multiple of spaces. When embedding elements of different types in the same Euclidean space, these elements cohabit without interference. By virtue of the Cartesian product, algebras can be used and applied in combination without mutual interference. This parallel construction allows for objects of quite disparity not only to coexist peacefully in a single (spatial modeling) world but also to be conceived as one at the same time as being handled and operated on quite differently, yet using a conceptually unified approach.

The Cartesian product of algebras is not restricted to a composition of different dimensional algebras. Generalized as such, compound shapes can be regarded either as complex shapes that are composed of segments from different algebras or as consisting of shapes that are coordinated or related. For example, consider a set of drawings from amongst plans, elevations and sections of a same building. Each drawing can be considered a shape, as an element of a two-dimensional algebra, while the set of drawings can be regarded as a single shape that is an element of a Cartesian product of algebras (here, all of the same dimensionality).

The algebraic model is more than a model for the representation of shapes. It is a concept for spatial modeling that applies as well as a model of interaction for computational design. As such, it has the advantages of being clear, straightforward, yet inclusive: many, if not the most common computational design activities can be expressed using the algebraic model. Upon defining the appropriate algebras (e.g., for design and selection), the operations of creation and deletion, but also of selection and deselection can be expressed readily in terms of the shape operations of sum and difference in these algebras.

4 GRAIL

These explorations into shape computation and modeling form a part of the GRAIL project. In it, we explore the challenges and the implications that the algebraic model for shapes poses to computational design in an interactive and generative design environment. GRAIL currently includes the following objectives:

- exploring different ways of interacting with shapes, as detailed above. We are developing shape selection methods that derive from cross-algebra operations, where a selector shape may be in a different dimensional algebra than the shape that is ultimately selected.

- exploring different ways of creating and organizing shapes. The algebraic model is extensible to non-geometric elements and attributes. Elements may be layered and grouped according to both spatial and non-spatial features. In addition, shapes may have derivational dependencies. A uniform method for interactively and programmatically organizing and manipulating shapes through a forest-like organizational structure is being explored. We eventually hope to handle more general graph relationships.

- exploring generative approaches to spatial modeling. We are exploring ways of interactively defining spatial rules and rule application. Part of the GRAIL project subsumes spatial grammar interpreters within a 3-D spatial representational environment for modeling, analysis and visualization.

In addition, a programming language interface is planned for GRAIL so that a number of applications can be built upon it. The design is modular and data integrity is ensured by having each module exercise sole control over the creation, update and maintenance of all data under its purview. Robustness is achieved by ensuring that all operations are algebraically sound. GRAIL is designed to serve as an interface to a graphical system as well as a functional toolkit for application developers.

5 Where next ?

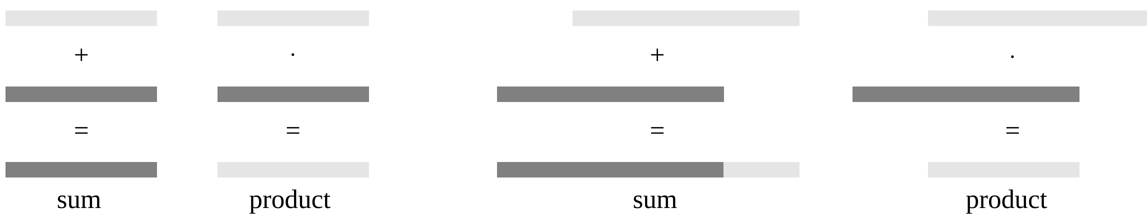
The algebraic model is general and complete; however, representationally it lacks in expressive power and is unlikely to gain recognition as a fundamental approach without a proper extension. Part of the attractiveness of the algebraic model is its ability to include non-geometric attributes within the model. We can easily add constant, symbolic or numerical information to shapes. In each of these examples, the augmented shapes have been derived from shapes of spatial elements by associating symbols, labels or properties, to the elements. Labeled points play an important role in shape grammars. They serve to guide the rule matching process through identification and classification of the rules. They could also be viewed as a semantic extension to what is fundamentally a syntactical expression, i.e., grammars[1].

It follows that the algebraic operations, and the underlying part relation, need to be redefined in order to deal correctly with the associated information. When considering labels, this can be achieved by an ordinary set approach: the sum of two identical points, each with a set of labels, is the single point with the union of both sets associated to it. It may seem less intuitive for segments of a different dimensionality. Instead, consider shapes augmented with weights or colors. For weights (e.g., line thicknesses), the part relation is obvious and the algebraic operations follow naturally (the sum of two identical segments with different weights is the single segment with the maximum of both weights). For colors, a ranking may be specified that maps the colors to real values similar to weights or, otherwise, a three-dimensional color coding (e.g., RGB or intensity, saturation and hue) may be considered with an appropriate part

1. Stiny (1990).

relation.

It would be attractive to consider each of these examples, formally, as a Cartesian product of a shape algebra with a non-spatial algebra. However, the algebraic operations do not distribute over the algebras that make up the Cartesian product as is the case in a Cartesian product of spatial algebras only. In the latter case, given two shapes each consisting of a line segment and a plane segment, the sum of both shapes is the Cartesian product of the sum of both line segments with the sum of both plane segments. In the case of colored shapes, the sum of two line segments with different colors that spatially overlap cannot be considered to be the sum of both line segments with a single color that is the sum of both individual colors. This only applies to the common segment, any other segment that belongs to only one of both shapes has to retain its original color under a proper algebraic model.



We need to consider a different mathematical formalism for augmented shapes. For this purpose we can introduce a characteristic function to a shape. In constructive solid geometry, a solid can be described as the combination of a set of half-spaces under the Boolean set operations of union, intersection and difference. Each half-space is defined by a characteristic function g with the values 0 and 1, that is, a point p is inside the half-space if $g(p) = 1$ and is outside, otherwise. If the surface bounding the half-space can be expressed as an analytic function $f(p) = 0$, it suffices to define $g(p) = 1$ if $f(p) \geq 0$ and $g(p) = 0$ otherwise.

Similarly we define a characteristic mapping f_a for a shape a. However, since a shape is not considered a point set, unless its spatial elements all have dimensionality 0, its definition is slightly different. Consider the set U of all finite arrangements of n-dimensional hyperplane segments of limited but non-zero measure in a k-dimensional space, for a given $n \leq k$. The function f_a is defined from a subset of U to the set {0, 1} such that for any element s of U:

$f_a(s) = 1$ if s is a part of a,
$f_a(s) = 0$ if s and a are disjoint (their product equals 0).

Note that f_a is not a function from U as it is not defined for all shapes of U; for a general shape s, it only holds that $f_a(s \cdot a) = 1$ and $f_a(s - a) = 0$, with $(s \cdot a) + (s - a) = s$.

In order to allow for a characteristic function, consider Δ the set of infinitesimally small elements of U. (Actually, it suffices to choose the elements of Δ small enough such that $f_a(s)$ exists, i.e., has a unique value, for every element s of Δ.) Then, U can be considered a power set over Δ, with any two elements in Δ disjoint. Define g_a a mapping from Δ to {0, 1} such that $g_a(s) = f_a(s)$. Given the choice of elements in Δ, g_a constitutes a function. Let g_a denote the characteristic function for a with respect to Δ. Consider the set G of all characteristic functions g_u with u an element of U and define the function f from U to G with $f(a) = g_a$ that maps each shape onto its characteristic function for Δ.

A range {0,1} distinguishes conceptually between shapes that are a part of (1) and shapes that are disjoint (0). When considering weights, any part is assigned a weight and we must distinguish shapes with different weights. If we consider a weight to be represented as a positive real value, the range of weights constitutes $\Re^+$, the set of positive real numbers. Any infinitesimally small element s of Δ is assigned a single weight and we can consider the characteristic function g_a to be a (step-wise constant or otherwise) function from Δ to $\Re^+$. Upon embedding weighted shapes in the Euclidean space, the result is similar to a spectral (discontinuous) function that specifies a "height" for every point. The characteristic function of two weighted shapes equals the "sum" of the characteristic functions of both shapes, where the operation of sum is defined for the algebra of weights, that is, the sum of two weights is the maximum value of both weights.

In general, when augmenting the shapes with non-spatial information, we only need to redefine the range for the characteristic function, e.g., we create one or more extra dimensions (e.g., three in the case of colors) that define the range space for the characteristic function. In the case of labeled shapes, for a given set of labels L, the range of the characteristic functions is the power set of L and the algebraic operations correspond to the set operations.

A similar treatment can be applied to procedural or functional weights attached to shapes. Here, a function, such as a daylight or structural analysis, is invoked only when certain conditions hold, which are governed by an appropriate characteristic function associated with the shapes in question. However, the details of this scheme have yet to be fully worked out.

References

Gursoz, E.L., Y. Choi and F.B. Prinz, 1991. “Boolean Set Operations on Non-Manifold Boundary Representation Objects,” *Computer Aided Design* 23, 33-39.

Krishnamurti. R., 1992. “The Maximal Representation of a Shape,” *Environment and Planning B: Planning and Design* 19, 267-288.

Mitchell, W.J., 1993. “A Computational View of Design Creativity,” in *Modeling Creativity and Knowledge-Based Creative Design* (eds., J.S. Gero and M.L. Maher), pp. 25-42, Lawrence Erlbaum Associates, Hillsdale, N.J.

Rossignac, J.R. and A.A.G. Requicha, 1991. “Constructive Non-Regularized Geometry,” *Computer Aided Design* 23, 21-32.

Stiny, G., 1980. “Introduction to Shape and Shape Grammars,” *Environment and Planning B: Planning and Design* 7, 409-462.

Stiny, G., 1986. “A New Line on Drafting Systems,” *Design Computing* 1, 5-19.

Stiny, G., 1991. “The Algebras of Design,” *Research in Engineering Design* 2, 171-181.

Stiny, G., 1993. “Emergence and Continuity in Shape Grammars.” in *CAAD Futures '93* (eds., U. Flemming and S. Van Wyk), pp. 37-54, North-Holland, Amsterdam.

Stouffs, R., 1994. *The Algebra of Shapes*, Ph.D. dissertation, Dept. of Architecture, Carnegie Mellon University, Pittsburgh, Pa.