SYNTHESIS OF AESTHETIC PRODUCT FORM

by

STEPHEN N. SMYTH

B.A.I. Mechanical Engineering
B.A. Mathematics
Trinity College, University of Dublin, Ireland, 1998

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2000

Signature of Author.......................
Department of Mechanical Engineering
May 5, 2000

Certified by.......................
David Wallace
Esther and Harold E. Edgerton Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by.......................
Ain A. Sonin
Chairman, Department Committee on Graduate Students

SYNTHESIS OF AESTHETIC PRODUCT FORM

by

STEPHEN N. SMYTH

Submitted to the Department of Mechanical Engineering
On May 5, 2000 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

ABSTRACT

A method for the synthesis of aesthetic product form is described. The approach is based on simulated evolution and stems from the idea of applying computation to the notion of brand DNA, a term which is used in industry to refer to the aesthetic form elements that contribute to brand identity. The designer defines the origin of the evolutionary process by loading an existing surface, interactively defining an archetype, or skeleton, of the desired form or automatically extracting features from physical models. The system uses the parameter set specified by the origin form to stochastically generate a variety of forms, or surface skins. The designer selects appealing surfaces for further evolution in form space. A proof-of-concept system, *3-DNA*, has been implemented that mutates and mates an origin surface. It is envisioned that this synthesis process will be applied within an integrated industrial design cycle that supports the rapid alternation between physical and digital representations of product geometry and maintains consistency with engineering and business models.

Thesis Supervisor: David Wallace
Title: Esther and Harold E. Edgerton Associate Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

The author would like to thank Professor David Wallace for his guidance and support. The members of the MIT Computer-Aided Design Laboratory (CADLab) provided a stimulating and entertaining work environment. Shaun Meredith, Nick Borland and Johnny Chang deserve special mention for their steadfast systems support. Edward Ferara implemented the Maya-DOME interface described in Chapter 5. Bill Liteplo, Priscilla Wang and Juan Carlos Deniz livened up the many thesis night shifts.

Thanks are due to Professor David Gossard, Professor Nicholas Patrikalakis and Professor Seth Teller of the MIT Departments of Mechanical Engineering, Ocean Engineering and Electrical Engineering & Computer Science respectively for their expertise on matters geometric. Thanks are also due to Professor Peter Testa, Professor George Stiny, Jose Duarte and Evan Smyth of the MIT Department of Architecture for their insights on computation and design. Susan Butler of the Department of Molecular Biology and Microbiology, Tufts University, had the daunting task of explaining genetics to an engineer.

This research work was supported by the Ford Motor Company Advanced Design Studio through the MIT-Ford Alliance and the National Science Foundation Award 9414793.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Motivation

The importance of aesthetic considerations in consumer product design is clear. As mature products such as automobiles converge on functionality, efforts are being made to differentiate products by appealing to consumers' emotional response. Coupled with this trend is the ever-present drive to reduce time-to-market as companies strive to lower development costs and respond aggressively to new or changing market segments.

Traditionally, aesthetic industrial design and engineering design are seen as polarized activities in consumer product development. This tension is often evident in the products that result (Wallace, 1991). Where harmony of form and function has been achieved, the manufacturer has often invested in many design iterations between these engineering and industrial design viewpoints. In an age when information technology is enabling professionals with varying backgrounds to represent and communicate complex ideas, it seems that this gap between engineers and industrial designers might be bridged to efficiently create concept designs that marry form and function.

Conceptually, brand DNA is used to indicate form elements that convey a product's brand identity. For example, BMW systematically maintains an inflection curve in the C-pillar of the manufacturer's sedan automobiles called the 'Hofmeister Knick,' (see Figure 1-1). A consequence of this research is the development of a geometric understanding of the makeup of brand DNA. Increasingly, design communication is taking place with external suppliers contracted to design, manufacture and deliver entire product modules. Large automotive suppliers have begun developing in-house styling competence as their customers seek full design services. This fragmentation of aesthetic design execution has potentially negative implications for the cohesiveness of the product image. Once defined, the product form DNA can be distributed to all suppliers of design services so that a consistent product image can be maintained.

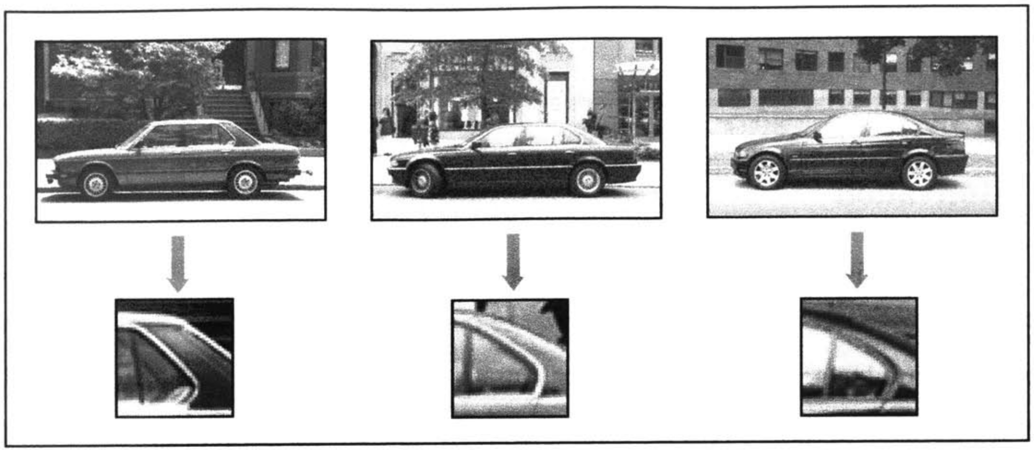


Figure 1-1 BMW Hofmeister Knick maintained in generations of the manufacturer's sedan automobiles.

The philosophy of form-giving tools should evolve to meet these new challenges. Conceptual design of product form has received relatively little attention in design computation. Computer-Aided Drafting (CAD) has provided the ability to efficiently produce precise, detailed representations of product geometry for use in visualization, simulation, manufacturing and assembly. Advanced digital conceptual design tools still rely on a sophisticated understanding of the mathematical description of surfaces and are poorly integrated with engineering modeling tools. Few attempts have been made to tie the fluid and approximate world of conceptual form development with the necessarily precise world of geometry tied to performance and manufacture. High-level design operators that manipulate the mathematical description of form are required to both aid the designer and facilitate communication with engineering and business models. These operators can be viewed as a set of handles that are accessible to designers, synthesis tools and engineering design models. There is a need for the development of conceptual design tools that allow the intuitive manipulation of product designs and form DNA elements. The proposed synthesis framework is an effort to address these challenges in a way that is designer-oriented, compatible with existing conceptual design tools, and intended for integration with an existing architecture of engineering and business models (Abrahamson *et al.*, 2000).

1.2 Concept

This research focuses on geometric modeling and synthesis for aesthetic product design. It is based on the premise that minimizing the input of non-intuitive mathematical information is fundamental to a computer tool for conceptual form development. A key contribution of this work is application of the evolutionary paradigm to searching product form space, as represented by parametric surfaces. Simulating the evolution of form has the benefits of automating the actual process of geometry creation and allowing designers to quickly and thoroughly explore form space. In addition, such multi-start search simulations are now feasible as computational power increases (Parmee in Bentley, 1999). The drawback is that explicit control of geometry may be lost, although this need not be the case if the underlying surface representation is amenable to traditional surface manipulation tools.

The process is as follows. The designer defines the origin of an evolutionary process by specifying the archetype of the desired form based on existing product geometry. The system uses the handles of the archetype to generate a variety of forms, or surface skins based on form DNA elements. The designer selects appealing designs and performs further form searching by a combination of mutation and mating operations. It is important to note that the underlying algorithm differs from a conventional genetic algorithm in that it acts as a suggestive tool and does not attempt to evaluate the aesthetics of the surfaces generated by some fitness function. It is therefore hoped that designers will see this system as a tool for searching form space and not an attempt to quantify the aesthetics of form.

In much the same way that CAD has adopted the parameter specification of features as high-level operators, there is potential in applying this approach to conceptual design (Fontana *et al.*, 1999). In freeform surface design, parameter space has little to do with the product specifications or design intent, and more to do with the mathematical description of form in three dimensions. Automating parts of the form synthesis process using higher-level operators mitigates this limitation of surface modeling. Such an environment might allow the designer to specify the high-level parameter space by using a product archetype or skeleton. Interesting sets of parameter values, or parents, can then be combined or otherwise varied to produce a new set of parameter values that correspond to a new form. The designer selects forms for further evolution at each generation. In this way, many design alternatives can be generated by navigating 3-D form space, all of which contain a blend of the brand identity and the designer input.

1.3 Document Structure

Chapter 1 provides context and motivation for this research. Chapter 2 delves into the background of the topic by exploring relevant literature from industrial design, architecture, mathematics, computer-aided design, computer graphics and artificial intelligence. Related work is classified and the advantages and disadvantages of these approaches with respect to the problem domain are specified. Chapter 3 is the heart of the document and details the synthesis model. Chapter 4 documents a proof-of-concept implementation of the synthesis model, *3-DNA*, and describes several examples. Chapter 5 describes the integration of the form synthesis engine with conventional digital and physical form development tools and, more broadly, with engineering and business models. Chapter 6 summarizes the research contribution, highlights limitations of the synthesis model and suggests future avenues of research.

2 BACKGROUND

This chapter describes work to date on the synthesis of form. The synthesis of form has interested designers, engineers, and artists for some time. The majority of the computational approaches to this problem have been pioneered in architecture (Stiny, 1980), entertainment (Watt and Watt, 1992) and biology (Thompson, 1961), although product design is an emerging application domain (Wallace, 1991; Wallace and Jakiela, 1993; Hsiao and Cheng, 1996; Knoop *et al.*, 1998; Van Breemen *et al.*, 1998; Cagan and Argawal, 1998).

2.1 Shape Grammars

Shape grammars (Stiny, 1980) have been used to generate designs of buildings (Heisserman and Woodbury, 1994) and coffeemakers (Cagan and Argawal, 1998) in a particular style. Figure 2-1 illustrates the definition and use of a simple shape grammar.

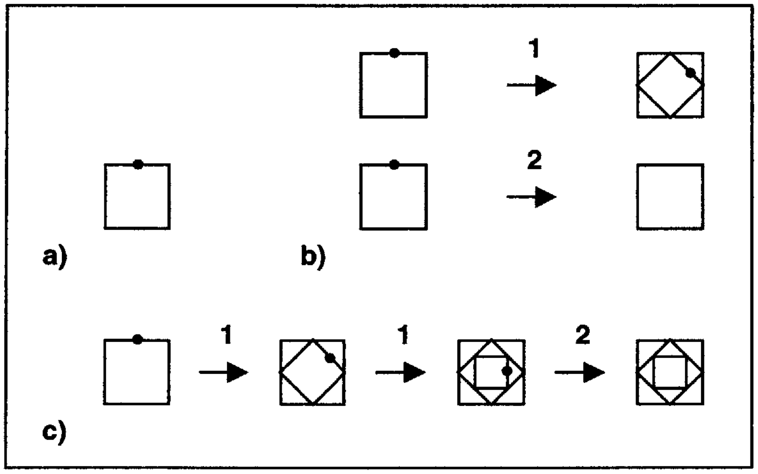


Figure 2-1 Initial shape and shape rules define shape grammar. a) Initial shape, b) shape rules, c) generation of shape using grammar. (Re-drawn from Stiny, 1980)

Although promising, the use of a shape grammar first requires the non-trivial task of defining the rules of the grammar and then shapes are combined to produce novel designs. Nonetheless, Cagan and Argawal (1998) extended the parametric shape grammar to include function labels that drive the configuration of form. The grammar is applied in two dimensions in each of three orthographic views that are in turn used to create a three-dimensional form. This approach is limited to relatively regular forms and does not readily support freeform surface design.

2.2 Modulor

The architect Le Corbusier created a system of proportion called the Modulor (Jeanneret-Gris, 1954). It was based on the observation that the repeated division of the height of an average male by the Golden Section coincided with key articulation points of the body. Interestingly, the set of these divisions forms a Fibonacci series. Figure 2-2 illustrates the concept.

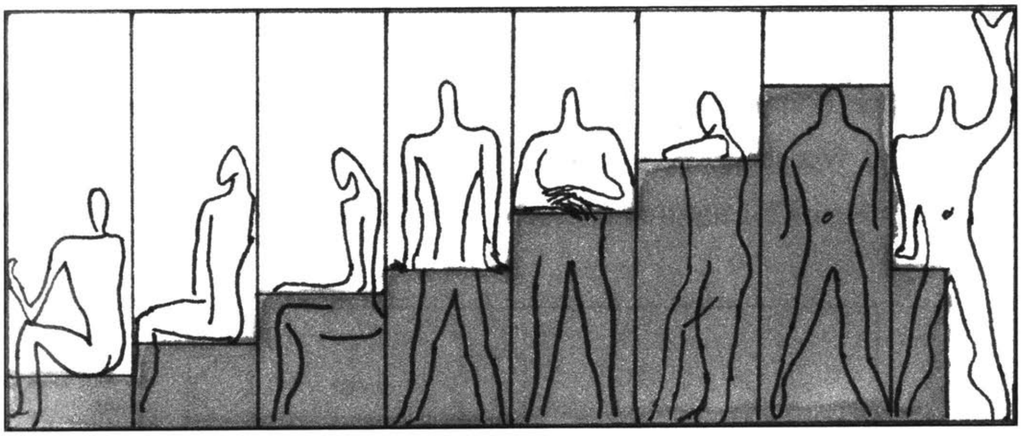


Figure 2-2 Modulor system of proportion. Gray regions indicate division by Golden Section and correspond to key articulation points of male human body. (Re-drawn from Jeanneret-Gris, 1954)

Le Corbusier believed that the Modulor was a system of measurement based on the dimensions of the human form that unified 'arbitrary' measuring systems such as the metric and imperial standards. He also felt that in an age of mass production this system would allow varying forms to fit together more easily. To summarize in the words of Rowland (1965), "as well as enabling us to reap the full benefit of mass-production it will also help us to produce things which satisfy the requirements of our bodies and minds."

Le Corbusier devised a measuring tape for this system of proportions. The tape was used to evaluate the proportions of dimensions such as the height and width of doorways, windows and hallways. The architect claimed to have used the system in the design of several buildings. Although strictly speaking not a form synthesis tool, the Modulor is nonetheless significant as a set of rules which can be used to assess the human factors and visual balance of architectural form.

2.3 Expert Systems

An expert system has been used to automatically generate box-like consumer electronic product layouts and enclosures subject to aesthetic, manufacturing, ergonomic and corporate identity requirements (Wallace, 1991). Figure 2-3 shows the configuration, surfacing, detailing and graphics stages in the conceptual design of a hand-held calculator in the 'Braun style.'

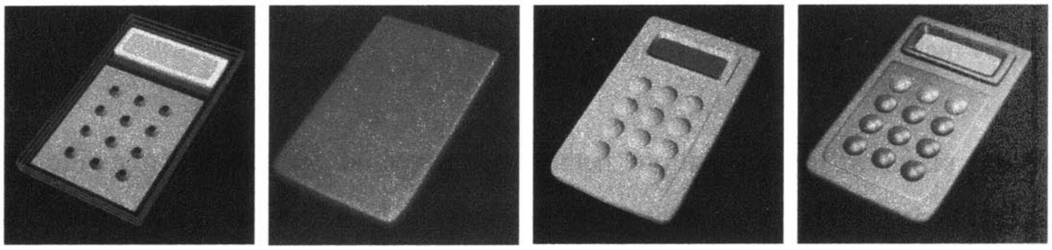


Figure 2-3 Generation of 'Braun-style' calculator. (Images courtesy D.R. Wallace)

This system has the advantage that the 'rules of thumb' for the given product need only be defined once and then applied repeatedly to generate new configurations of a given product. However this strength is also a limitation in that this rule-based synthesis model is not readily adaptable to more complex product topologies and does not support 'organic' forms well.

2.4 Semantic Transformation

Form can be generated by analyzing human responses to shapes and thereby defining the transformations from descriptive words to shape. In a sense, words are the ultimate high-level form operators and constitute a common language for all of the participants of the conceptual design phase: designers, engineers, marketers and test consumers. Unfortunately, the compactness of verbal expression also leads to ambiguity. An attempt was made to overcome this problem by combining fuzzy set theory and consumer-oriented Kansei engineering techniques to analyze the results of consumer surveys and determine the relationships between image and shape-regulating words and car styles (Hsiao and Wang, 1998). The same author also used fuzzy set theory in conjunction with weighted mean and weighted generalized mean methods to merge multiple beta-spline models of automobiles (Hsiao and Cheng, 1996). A weakness of these techniques is that they can only produce forms that are a combination of pre-defined forms.

Information communication theory has also been proposed as a model for the mapping from consumer response to product aesthetics (Van Breemen *et al.*, 1999). Although the results show a statistical clustering in the mapping, this mapping is intrinsically limited to the set of shapes used in deriving the mapping. For this reason it can be viewed as a technique for suggesting only the skin of the product form. Archetypes still need to be defined by means of some direct form manipulator.

While the semantic transformation method has potential for quickly mapping image words into approximate forms, their use would seem to be limited to first order marketing discussions and not aesthetic form design. However, the output of the transformation could act as an input or evolutionary origin for the proposed form synthesis model.

2.5 Procedural Modeling

Typeface design provides a simple example of procedural modeling. A discussion of Donald Knuth's Metafont concept (Hofstadter, 1985) points towards the essence of procedural approaches. Procedural modeling relies on the concept that a single meta-abstraction of a class of objects can be identified, and thus every instance of related objects is a product of a specific tuning of the parameter set associated with the meta-model.

This approach to modeling is used extensively in the entertainment industry to model character animation and generate textures, mountain ranges and plants. Its power lies in the ability to abstract a complex model and control it with relatively few parameters. However, procedural models suffer from a lack of generality and are ideally suited to an application such as animation where the parameters are repeatedly driven by kinematics models. The user is required to understand the implications of changing the input values of the procedure (Watt and Watt, 1992). In short, while a powerful tool for certain types of models, it lacks the flexibility required for general-purpose geometric modeling.

2.6 Evolutionary Design

Evolution is a powerful metaphor for form development. Although the simulated evolution of form does not exactly mimic the natural world the use of the evolutionary paradigm holds many attractions. The user can rapidly create a large variety of complex artifacts without understanding the creation process. Even though an evolutionary model may manipulate procedural models, the user does not explicitly need to understand how the procedural model was constructed and how parameter changes affect the model (Watt and Watt, 1992).

The geometry of 3-D form may be described by a *gene vector* or set, called a *chromosome*, of pieces of information, called *genotypes*, that code for the creation of a form instance. In the simulated evolution of form the genes are the labels of the chosen geometry handles and genotypes are the real number values of the handles. Once the structure of the chromosome has been defined it remains for genetic operations such as *mutation* and *mating* to operate on the gene vectors and generate offspring or *progeny*. Developmental rules are combined with genotypes to generate form instances called *phenotypes* in a process known as *expression*. These phenotypes are evaluated according to some fitness criterion at each generation and selected for further evolution. In optimization applications this criterion is a mathematical function, while in the proposed synthesis model the chosen criterion is the aesthetic judgment of the designer.

Morphological transformations have been used to analyze the evolution of biological forms (Thompson, 1961). A significant limitation of morphological transformations is that the complete structure of the form must vary in a uniform manner. If there are too many independent variables analysis is impractical. Evolutionary models have been applied as genetic algorithms to optimization problems. These algorithms have been applied to design in the generation of textures in two-dimensional function space (Sims, 1991), the generation of virtual creatures (Sims, 1994), computer-generated art (Todd and Latham, 1992), mechanical features (Taura *et al.*, 1998) and implicit surfaces (Bedwell and Ebert, 1998; Bedwell and Ebert, 1999). The Biomorphs program (Dawkins, 1986) experimented with form in two dimensions and numbers among the earliest efforts to create computer art. It is argued that, to date, most of these systems have had a tendency to reflect the 'signature' of the program and not of the artist (Rowbottom in Bentley, 1999).

Recent related research work has focused on a generic evolutionary design system (Bentley, 1999) that generates design forms that meet certain performance criteria. The system is generic in the sense that there is a common core genetic algorithm while application-specific evaluation modules are connected for the design of everything from coffee tables, hospitals floor plans to cars. The system uses a spatial partitioning representation called 'clipped stretched cuboids,' which requires only nine parameters to characterize the geometry. While appropriate for rudimentary visualization purposes, this representation cannot be used to create CNC milling paths required for manufactured freeform surfaces. Little research has been done in the generation of aesthetic forms that are manufacturable.

Although interesting work has been done in this area, several modifications are necessary to make this a practical tool for product form design. The existing work does not address the modeling of parametric surfaces – a convenient means of geometric representation in aesthetic product design (Faux and Pratt, 1979; Hoschek and Lasser, 1993). Therefore it is important that evolutionary techniques for digital concept design support parametric surfaces. This requirement allows the form synthesis model to be compatible with existing conceptual design tools.

2.7 L-systems

L-systems, named for their inventor Lindenmayer (1968), are parallel graph grammars that describe plant topology by exploiting the property of self-similarity evident in the branch structure of plants. These devices have been successfully applied to the modeling of plants and trees in computer graphics (Smith, 1984). In the modeling context, a form is specified by a sentence that is then parsed as a set of rules that control the generation of the model. The fractal nature of these grammars makes them of limited use in product design, as products do not lend themselves to description by self-replicating hierarchies of features. However, the work of Prusinkiewicz *et al.* (1988) on the simulation of development of plants may have an interesting analogue in the use of design intent or history to describe form. Section 3.1.2 refers to research work done on the use of interaction history in the definition of surface features.

3 SYNTHESIS MODEL

The proposed model for the synthesis of aesthetic product form is outlined in Figure 3-1. In keeping with the biological metaphor, the parameter set of the form is called the chromosome or gene vector and is defined manually by the designer, seeded from a surface or generated by machine from physical form element. In the following step genetic operations such as mating and mutation are performed on the gene vector(s). The designer selects interesting variants from the generated forms and performs further operations and continues the evolutionary loop until a satisfactory form concept has been created.

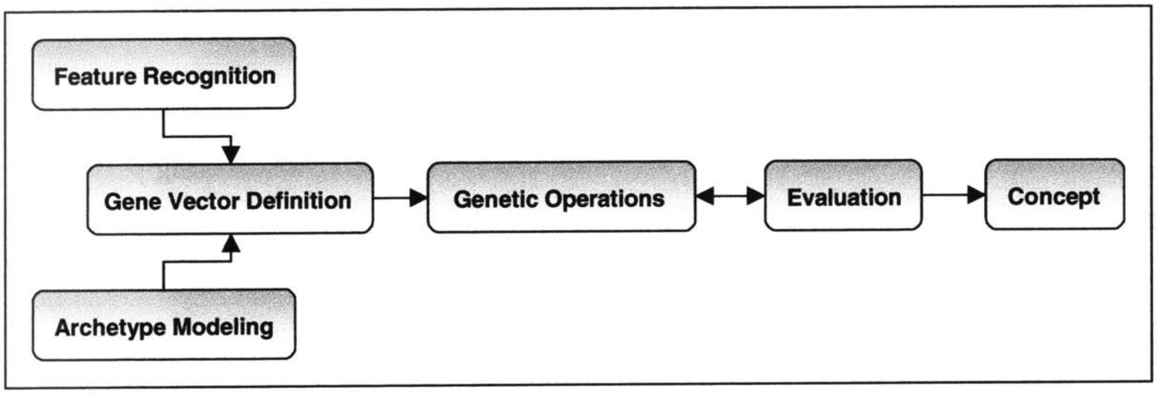


Figure 3-1 Form synthesis model.

An ideal model for the synthesis of aesthetic product form should have high-level form operators, generalized archetype topology for a variety of product forms, support for constraints from engineering and business models. The underlying geometrical representation should be compatible with other conceptual modeling tools.

3.1 Geometric Representation

The choice of geometric representation is critical. It affects the complexity of forms that can be modeled, the parameterization of the forms, the interface with other applications and the performance of the system.

3.1.1 General considerations

A number of geometric representations were considered for this tool: parametric surfaces, solid models and voxels. Each has advantages and disadvantages as outlined in Table 3-1.

Table 3-1 Advantages and disadvantages of geometric representations.

Representation	Advantages	Disadvantages
Parametric Surface	• Freeform modeling • Smooth surfaces • Compatible with other tools	• Difficult to implement • Models not 'watertight' • Difficult to store design intent
Solid Model	• Easy to store design intent • Models 'watertight'	• Difficult to implement • Restricted freeform modeling
Voxel	• Easy to implement • Models 'watertight'	• Memory intensive • Surfaces not smooth

Solid models lend themselves well to parameterization. However they are generally limited to describing the prismatic forms common in mechanical engineering parts. This is not suitable for the organic shapes commonly used by designers of aesthetic form.

Voxels, or volume elements, are useful for interaction and visualization, but do not provide surface quality suitable for manufactured products.

Implicit surfaces were considered because of their convenient fit with procedural models, but suffer from a lack of interactive control handles (Bajaj, 1993). A method of defining surfaces using the boundary value problem was also investigated as a means of limiting the number of control handles to the boundary inputs (Ugail *et al.*, 1999). Although interesting for engineering applications involving surfaces driven by functions, the author felt that this means of surface definition was non-intuitive for aesthetic design.

Parametric surfaces are largely defined with the freeform modeling of aesthetic surfaces in mind and can be specified to maintain the continuity required for manufacturing operations such as CNC milling. In addition parametric surfaces are compatible with other conceptual modeling tools.

3.1.2 Parametric surface considerations

While parametric surfaces are desirable for direct freeform modeling they raise some concerns for this application. In theory there are an infinite number of possible surfaces in form space. Since the evolutionary navigation of form space is an iterative, stochastic process and the number of phenotypes that can be displayed per generation is limited to the display size, it is vital to constrain the search by limiting the number of degrees of freedom. This can be achieved by using only transformations that have some design significance in order reduce the populations of useless forms. For this reason, direct manipulation of the many control points of the pervasive Non-Uniform Rational B-Spline (NURBS) surface representation is not suitable for the evolutionary metaphor. This problem is compounded by the fact that as NURBS surfaces become complex there is a data explosion associated with the control point net required to model detailed features (Wyvill *et al.*, 1997). Therefore there is a compelling motivation for a parameter set that is based on an abstraction of the control point parameter set, on a parametric surface feature representation or simply on transformations of entire NURBS surfaces.

The most trivial representation to implement is the application of transformations to parametric surfaces. Unfortunately this approach also yields the least interesting results as the set of transformations is limited to globally applied scaling operations.

The definition of a mathematical representation of a complex surface feature is a challenging task. Whereas mechanical engineering features are regular, the kinds of features beloved of aesthetic designers are anything but regular. It is argued that the parametric nature of a feature definition lends itself well to an abstract styling primitive because features contain design intent and are not just static or 'dead' geometry (Fontana *et al.*, 1999). Although Fontana focuses on detail features and does not expand the approach to overall features, the analogy with mechanical engineering design features is supported by the observation that designers tend to reuse learned strategies, or styling features, to generate new concepts. Other interesting approaches to freeform feature modeling have included: definition of the surface feature by its construction history (Wyvill *et al.*, 1996); and definition of the surface as a blend of features (Wyvill *et al.*, 1997). These approaches have not been extended beyond the topological limitation of mapping from the unit sphere. A displacement method for modeling depression and protrusion detail features on freeform surfaces (Van Elsas and Vergeest, 1998) also shows promise.

Surface deformers are a means of abstraction of the surface control point set and come in many flavors including lattice (see Figure 3-2), wire, skeleton, wrap and cluster. All of these deformers are described in Alias|Wavefront (1999). Generally, deformation techniques relate a set of high-level handles to the underlying parametric surface geometry. The positions of these handles constitute the genotypes. Although lattice deformers are popular in the entertainment industry for soft body animation (Coquillart, 1990), they are limited to certain topologies and can be difficult to use for fine resolution modeling. A variant of axial deformation, called wire deformation (Singh and Fiume, 1998), uses a wire network metaphor to deform parametric surfaces and, although non-trivial to implement, appears to be suitable for an industrial design application. The wire network of the object may be equivalent to a form archetype. Further research work is needed to ensure that the underlying surface maintains C^2 continuity – important for user-facing surfaces. A skeleton model can be created to underpin the skinned model of the form. Although skeleton structures are principally used for inverse kinematics solvers in animation, they also have potential for modeling. The deformation of a surface can be driven by an associated skeleton. The transformations of the skeleton can constitute the genotypes used in the evolutionary process. Means of defining the gene vector using different types of handles are detailed in the next section.

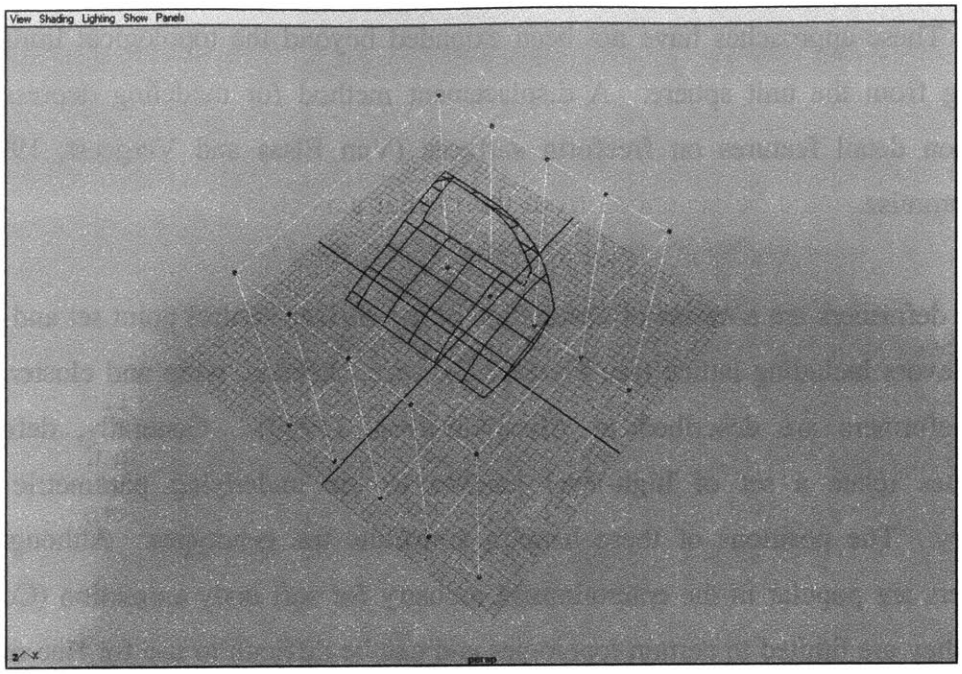


Figure 3-2 Lattice deformer network surrounding surface model.

3.2 Gene Vector Definition

It is envisioned that the archetype used in evolution will be defined manually by the designer, implicitly expressed by an origin surface or generated by machine from physical form element models.

3.2.1 Manual

There are a number of possible manual approaches. The designer blocks out the approximate shape of the product from the 'ground-up' using bounding volumes. These bounding volumes in turn define an archetype and the parameters associated with the extents of the volumes are input parameters into the procedural modeler. These extents are then held constant for the duration of the evolutionary process. Another manual approach consists of constructing a skeleton and using the skeleton to define a parameter set. The evolutionary modeler combines these parameters with variable information.

Skeletons may also be defined manually and associated with an origin surface or used as a profile for extruding an origin surface. An origin surface model could be derived from a previous product model or the reverse-engineered model of a competitor's product.

This becomes the archetype for the evolutionary process with the set of parameters associated with the transformations of the skeleton constituting the gene vector. This approach could be combined with the previously described bounding volume approach. Skeletal spines could be generated from the medial axis (Blum, 1967) of the bounding volumes.

3.2.2 Seeding

Origin surfaces can be used directly as form archetypes. The surface can be associated with a deformer. The deformer transformations constitute genotypes in this case. This approach applies at least to the lattice deformers, although other deformers are being investigated.

3.2.3 Automatic

The automated alternative combines reverse engineering and feature detection techniques to digitize existing styling features and extract parameters. This approach has the interesting possibility of allowing a library of meta-models of styling features to be rapidly created from physical artifacts and subsequently evolved.

3.3 Evolution

3.3.1 Overview

The synthesis model stems from the pioneering work of Todd and Latham (1992), but differs significantly in that parametric surfaces, not constructive solid geometry, are used as the geometric representation. Surfaces are synthesized via sets of form handles specified by deformers such as skeletons or lattice structures. Once gene vectors have been defined with the appropriate form handles, it remains for genetic operations such as mutation or mating to operate on the gene vectors and generate progeny. Procedural models or deformer systems constitute the developmental rules required for the genotypes to find expression as phenotypes or form instances. Interesting phenotypes are selected for further evolution.

3.3.2 Pros and cons

The combination of evolutionary modeling and procedural modeling is powerful (Watt and Watt, 1992). The use of a procedural model means that the user is not required to understand the mathematical representation of the model, while the use of an evolutionary algorithm means that the designer does not explicitly need to understand how the procedural model was constructed or how parameter changes affect the model. The user merely uses some fitness criterion at each generation and directs the evolution of form. In optimization applications this criterion is a mathematical function, while in the proposed synthesis model the chosen criterion is the aesthetic judgment of the designer.

An evolutionary process has the significant advantage that traits of interest can be maintained throughout the evolution of the model. The evolutionary model also allows features to be evolved from predecessor and competitor products. The parameter vectors of these features can be grouped together and mating operations can be applied to these groups.

The evolutionary approach may have the potential drawback of stipulating that the dimensions of the genetic space of the form are known before the evolutionary process begins. This means that parameters must be associated with the product form from the outset, conceivably limiting creative freedom. However, one could view this definition of the parameters as a rigorous definition of form groups and therefore beneficial to understanding the design archetype. Researchers have experimented with evolving the structure of images and the possibility of applying this to the synthesis model is discussed in Section 6.3.4.

3.3.3 Algorithm

The evolutionary algorithm is based on the work of Todd and Latham (1992). It is essentially the search component of a genetic algorithm (Goldberg, 1989). No mathematical function is used to evaluate the fitness of the generated phenotypes. The judgment of the designer is used to evaluate this fitness.

The algorithm is relatively straightforward and is succinctly described in Watt and Watt (1992). Given a gene vector V_{state}, mutated versions V_{mut} are generated according to

$$V_{mut} = V_{state} + mutrate * rand * V_{scale} \text{,} \qquad \textbf{Equation 3-1}$$

where *rand* is a random number in range [-1,1] and V_{scale} is a vector that controls the scaling of the perturbation applied to the individual components of the vector. The *mutrate* is a scalar that controls the degree of perturbation from generation to generation. Typically this evolutionary parameter is set high initially to generate variants that are significantly different. As the user directs the search process, this control should be decreased so that the next generation searches the space more immediately surrounding the desired form.

The process can be directed by ranking all of the variants at each generation. This extra information generates a traversal vector V_{dir}, which may be defined as follows,

$$V_{dir} = B * V_{dir} + A * \left(V_{mut} - V_{state}\right), \qquad \textbf{Equation 3-2}$$

where A and B are weights assigned to each of the mutations and $B > 0$ and $A \leq 0$ according to whether the mutation is desirable or not. This vector allows the user to effectively steer the evolution in a direction of interest. In effect, this is a form of steepest ascent optimization.

Forms are mated by combining gene vectors from multiple parents. There are two principal means of combining the gene vectors explored in this work: weighted averaging and random selection. In random selection the offspring gene vector is composed by randomly selecting the contributing parent for each gene. In practice it has been found that random selection produces the more interesting results and weighted averaging tends to produce bland results (Todd and Latham, 1992). It is also theoretically possible to group sub-forms together and apply the genetic operations uniformly to the whole group, thereby preserving recognizable features in the progeny.

3.3.4 Interaction

Interaction is relatively straightforward. The designer chooses a handle type, i.e. skeleton, lattice, bounding volumes, etc. Depending on the choice of handle a skeleton, origin surface, set of bounding boxes, etc. is then specified. The designer enters genetic operation settings such as the mutation rate and starts the generation. Once the algorithm has generated variants the designer evaluates the form and selects a form for further evolution. This form is used as the parent for a further generation in exactly the same way as before, except that the genetic operation settings are refined to narrow the search space. Although this is a stochastic process, the search space is localized as the evolution is steered and the iterative loop stops when the designer is satisfied with the synthesized form.

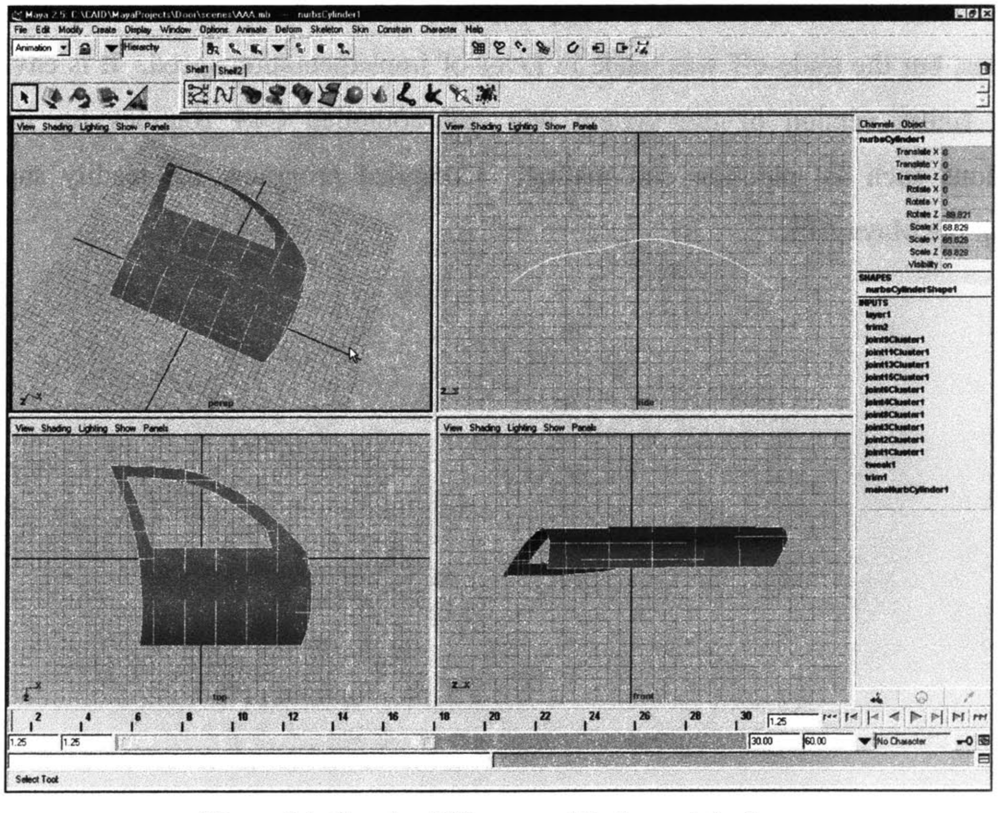
A certain amount of experience is required to select the mutation and mating parameters. For example, when mutating form the mutation rate should initially be set high, to generate widely differing phenotypes. As evolution continues the mutation rate should be decreased to increasingly localize the form search. Careful setting of the tolerance and scale factor parameters also has effect on the search. When mating, crossover techniques such as random selection and weighted averaging of parent gene vectors can aid in the rapid exploration of form space. All of these parameters should be experimented with for best results.

4 IMPLEMENTATION

This chapter describes the system design, implementation and an example application of the form synthesis engine. In the examples described, the gene vector drives either a procedural model or lattice deformer. Work is underway on the next step of the proof-of-concept demonstration—evolution of the form using a skeletal structure.

4.1 Implementation

The model is being implemented as a proof-of-concept software system called *3-DNA*. The system is being implemented as a plug-in of Maya®, a modeling, rendering and animation software package produced by Alias|Wavefront, Ltd. This plug-in implementation strategy was chosen over development of a stand-alone application for a number of reasons. Foremost was the ease-of-implementation associated with leveraging Maya's graphics pipeline, modeling and rendering functionality and graphical user interface (GUI) (see Figure 4-1).



nurbsCylinder1
Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z -89.821
Scale X 68.829
Scale Y 68.829
Scale Z 68.829
Visibility on

SHAPES
nurbsCylinderShape1

INPUTS
layer1
trim2
joint9Cluster1
joint11Cluster1
joint13Cluster1
joint15Cluster1
joint6Cluster1
joint4Cluster1
joint8Cluster1
joint3Cluster1
joint2Cluster1
joint1Cluster1
tweak1
trim1
makeNurbCylinder1

Figure 4-1 Standard Maya graphical user interface.

Development efforts could be concentrated on *3-DNA* functionality. Maya also has an open architecture that allows ready extension of the application using the Maya Embedded Language (MEL) interpreter for scripts and the Maya Application Programming Interface (API) for compiled plug-ins. Much of the Maya modeling functionality is well-suited to conceptual design and is less constraining than the modeling tools found in product surfacing tools such as Alias|Wavefront's Studio®. In addition, Maya has also become a standard in the entertainment industry and is familiar to many geometric modelers. This installed base is an important factor in user-testing *3-DNA* because many professional users tend to be sensitive to the 'look and feel' of an application and are often distracted by the rudimentary interfaces typical of research software systems.

The *3-DNA* system architecture is depicted in Figure 4-2. The system runs on the Microsoft® Windows NT operating system. The system is currently written as a set of MEL scripts. The programs are used to drive the interface and perform the computation of the simulated evolution. The choice of interpreted language over compiled language for the intensive computation of the population of the form space has performance penalties, but the trade-off was made in favor of implementation speed. It is envisioned that a future version of the system will call compiled C++ routines for intensive operations such as mutation and mating. Compiled programs are readily supported through the Maya API.

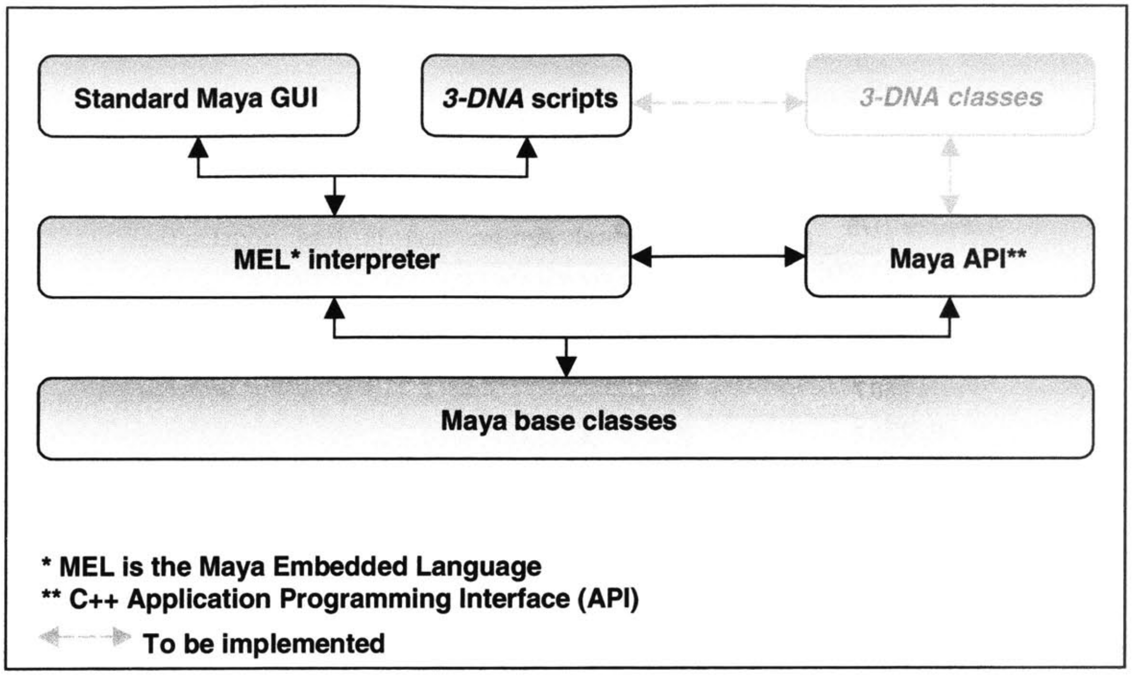
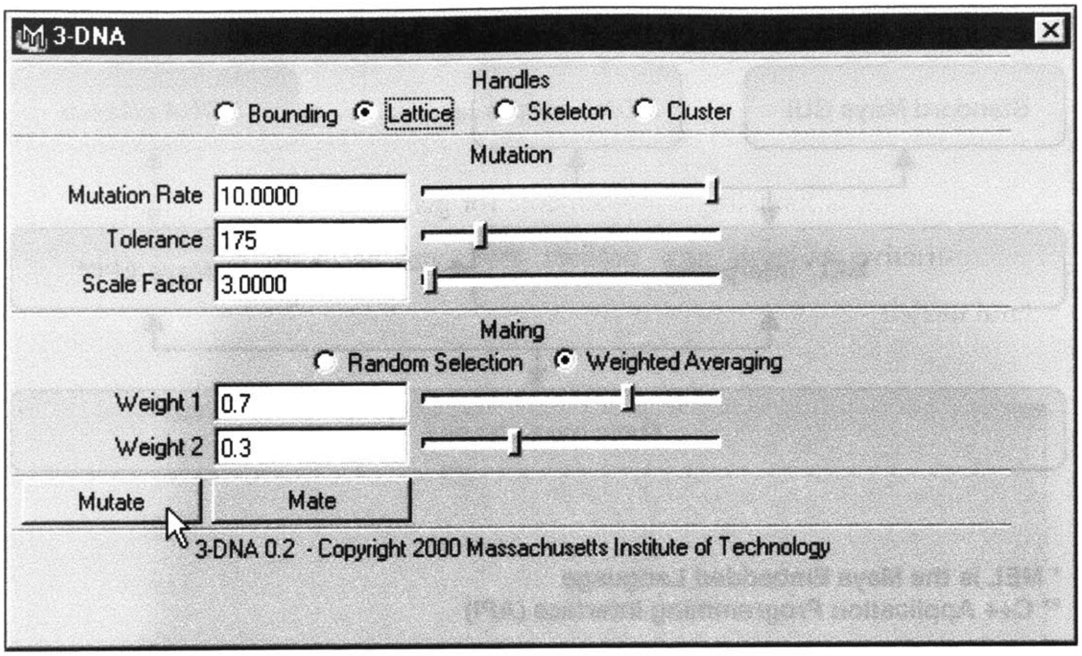


Figure 4-2 ***3-DNA*** **system architecture.**

4.2 Interface

The *3-DNA* interface consists of the simple menu shown in Figure 4-3. The user selects the desired handle type such as bounding volumes, lattices, skeleton and cluster. Although the cluster and skeleton handles are depicted, at present only the lattice deformer and bounding volume handles are implemented. Different inputs are required for the mutation and mating settings. Considering first the mutation options, the *Mutation Rate* determines the degree of randomness of the mutation; the *Tolerance* determines upper and lower bounds on the parameter value mutation; the *Scale Factor* is a scalar that determines the magnification of a change in a parameter value. The mating settings depend on the type of crossover selected: *Random Selection* or *Weighted Averaging*. *Weighted Averaging* requires the *Weight 1* and *Weight 2* inputs to determine the degree of influence of the parent genotypes. Unlike mating in the natural world, computer simulation can allow more than two parents. However, for the purposes of simplicity, *3-DNA* currently only supports two parents.



○ Bounding ● Lattice ○ Skeleton ○ Cluster

Mutation Rate 10.0000

Tolerance 175

Scale Factor 3.0000

○ Random Selection ● Weighted Averaging

Weight 1 0.7

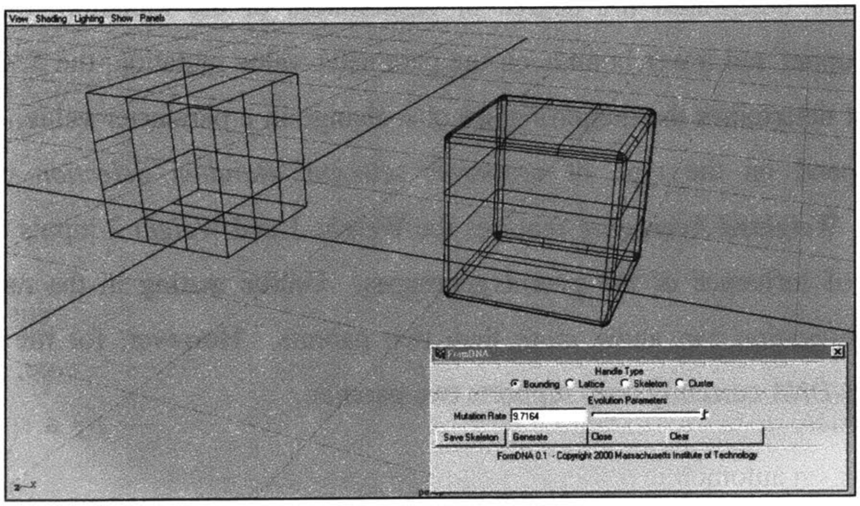
Weight 2 0.3

3-DNA 0.2 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-3 ***3-DNA*** **menu.**

4.3 Procedural Model

A procedural model was used to generate mutations of boxes with varying corner treatments. The user first defines the bounding volume of the box (see Figure 4-4). The gene vector, in this case, contains only one element: a bevel radius that regulates box corner treatments. A procedural model was programmed which has the bounding volume and bevel radius as inputs and outputs a box with the appropriate corner treatments.



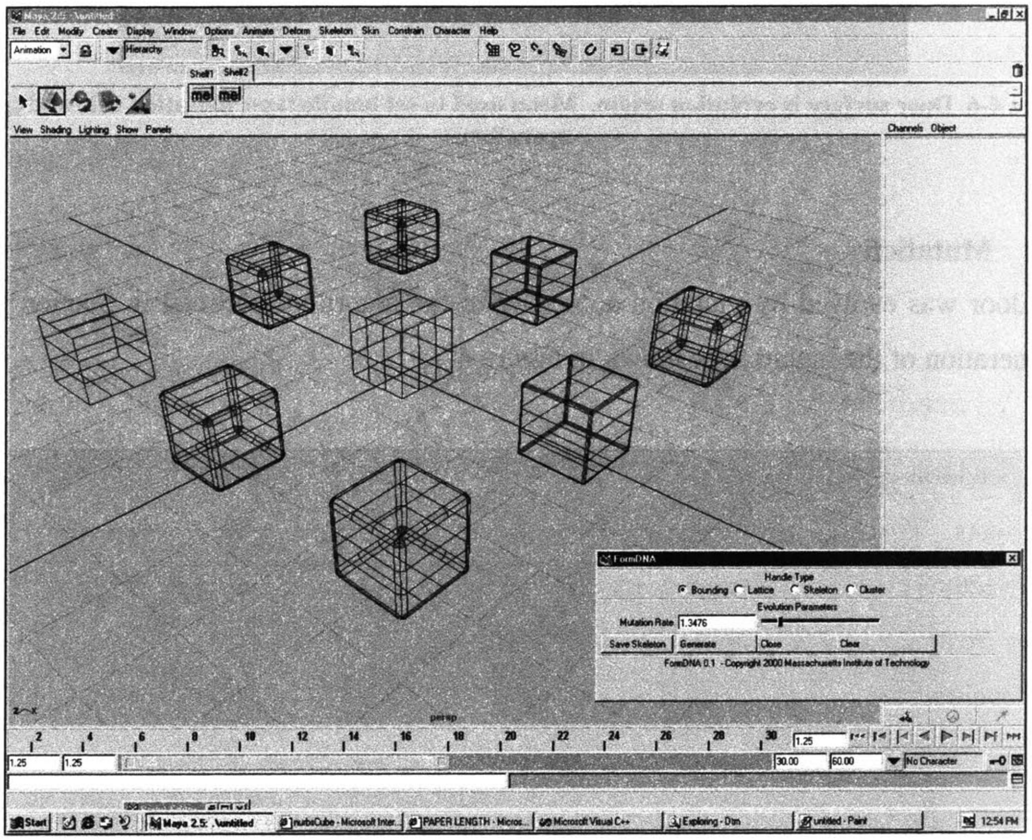
● Bounding ○ Lattice ○ Skeleton ○ Cluster

Mutation Rate 9.7164

FormDNA 0.1 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-4 Definition of bounding volume and resulting procedurally generated box.

A generation in the evolution of the procedurally generated box corner treatments is shown in Figure 4-5. The purpose of this example was to illustrate the construction of the gene vector and the following evolution for simple geometry. As discussed previously, procedural modeling is not suitable for general-purpose modeling and quickly becomes restrictive as each new product archetype needs to be programmed as a procedural model.



Handle Type: ● Bounding ○ Lattice ○ Skeleton ○ Cluster

Evolution Parameters

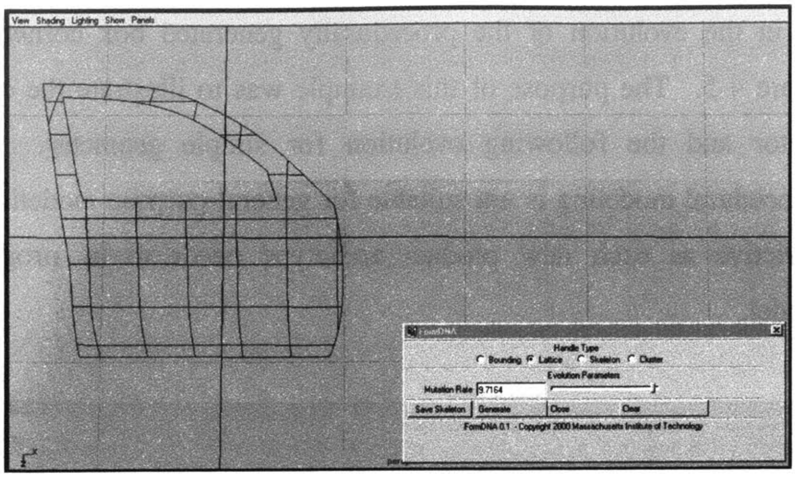
Mutation Rate 1.3476

FormDNA 0.1 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-5 Variants (phenotypes) of box corner treatments generated by procedural model. Eight phenotypes are synthesized per generation. Origin bounding volume is at center.

4.4 Lattice Deformer

Lattice deformers were used in the evolution of an automobile door. In this case the set of form handles of the deformer constitute the gene vector. The archetype, or origin, of the evolutionary process, is defined not by bounding volumes, but by a surface which approximates an automobile door (see Figure 4-6).



Handle Type

○ Bounding ● Lattice ○ Skeleton ○ Cluster

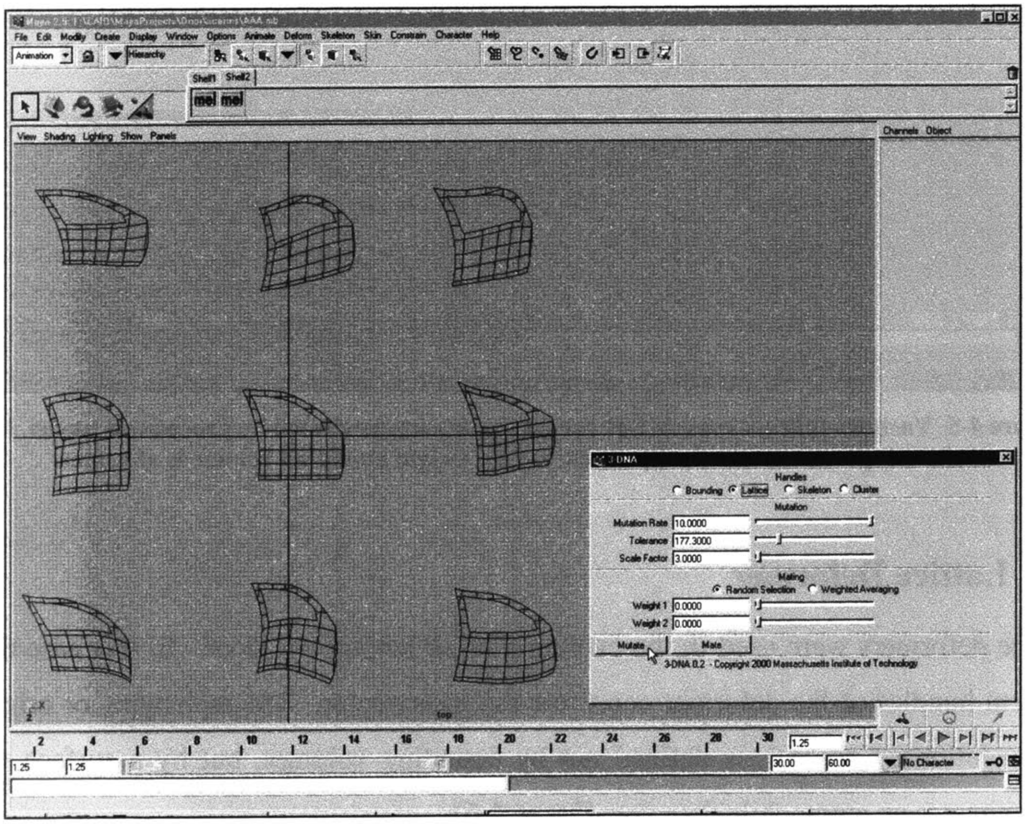
Evolution Parameters

Mutation Rate 9.7164

Figure 4-6 Door surface is evolution origin. Menu used to set handle type, mutation rate and genetic operation.

4.4.1 Mutation

The door was evolved by mutation according to the algorithm outlined in Section 3.3.3. A generation of the mutation is shown in Figure 4-7.



Handles

○ Bounding ● Lattice ○ Skeleton ○ Cluster

Mutation

Mutation Rate 10.0000

Tolerance 177.3000

Scale Factor 3.0000

Mating

● Random Selection ○ Weighted Averaging

Weight 1 0.0000

Weight 2 0.0000

Figure 4-7 Phenotypes of door surface generated using mutated lattice deformer handles. Eight phenotypes are synthesized per generation. Origin form is at center.

The evolution of the door is shown in Figure 4-8. Note that the mutation is initially very pronounced, but converges rapidly to the desired form.

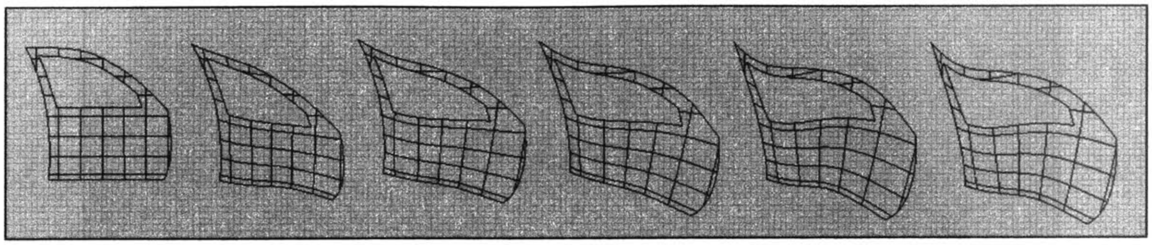


Figure 4-8 Stages in mutation of automobile door at mutation rates of 10.0, 7.5, 5.0, 2.5 and 0.0 respectively from left to right.

4.4.2 Mating

Both weighted averaging and random selection mating as described in Section 3.3.3 have been implemented. These operations were tested on the automobile door. The door was first mutated as in Section 4.4.1 and two phenotypes were selected for mating (see Figure 4-9). This pair was first mated using random selection and the progeny are shown in Figure 4-10. The same pair was then mated using weighted averaging and the single offspring is shown in Figure 4-11.

group...
Translate X 200
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on

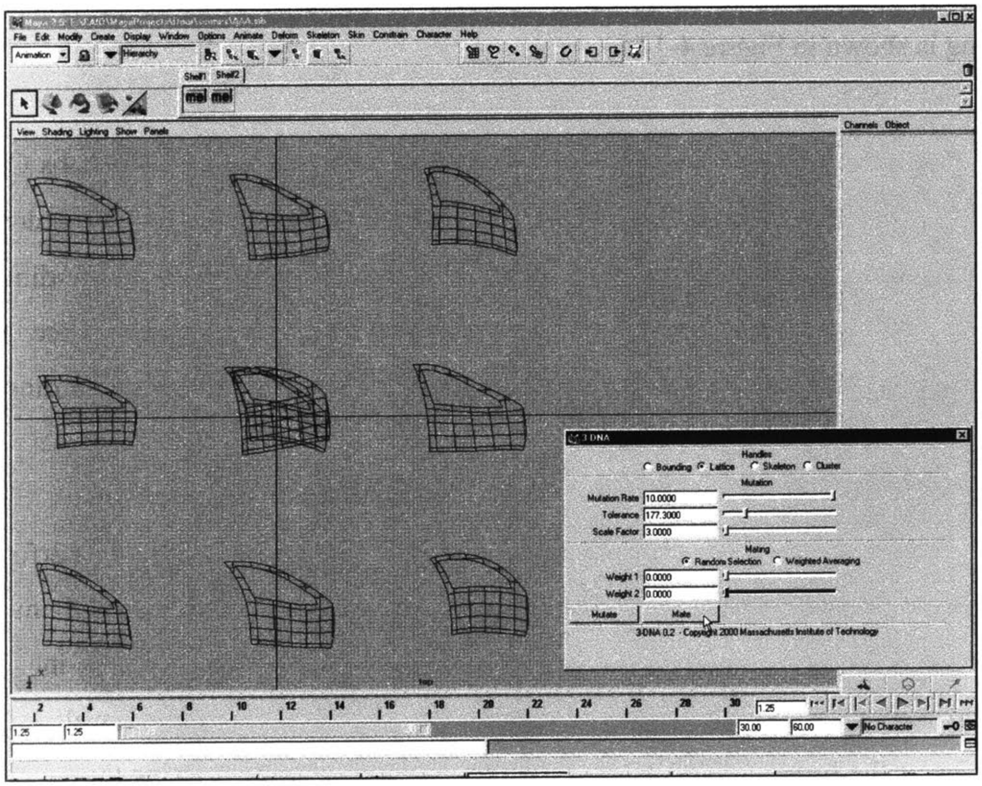
Handles
○ Bounding ● Lattice ○ Skeleton ○ Cluster

Mutation
Mutation Rate 10.0000
Tolerance 177.3000
Scale Factor 3.0000

Mating
● Random Selection ○ Weighted Averaging
Weight 1 0.0000
Weight 2 0.0000

3-DNA 0.2 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-9 Two phenotypes selected for mating.



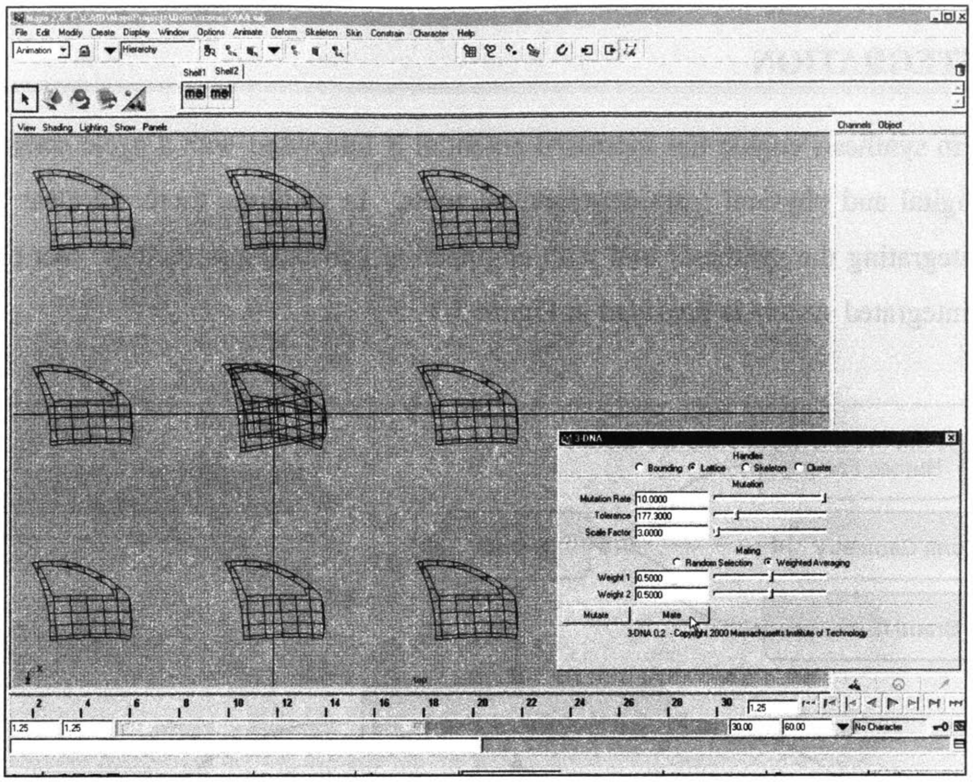
Handles
○ Bounding ● Lattice ○ Skeleton ○ Cluster

Mutation
Mutation Rate 10.0000
Tolerance 177.3000
Scale Factor 3.0000

Mating
● Random Selection ○ Weighted Averaging
Weight 1 0.0000
Weight 2 0.0000

3-DNA 0.2 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-10 Progeny resulting from random selection mating. Parents overlapping at center as reference.



Handles
○ Bounding ● Lattice ○ Skeleton ○ Cluster

Mutation
Mutation Rate 10.0000
Tolerance 177.3000
Scale Factor 3.0000

Mating
○ Random Selection ● Weighted Averaging
Weight 1 0.5000
Weight 2 0.5000

3-DNA 0.2 - Copyright 2000 Massachusetts Institute of Technology

Figure 4-11 Single offspring from weighted average mating. Offspring individual is repeated with parents overlapping at center as reference.

4.5 Performance

3-DNA runs on an Intel Pentium® II 450 MHz XEON processor with 3D Labs Oxygen™ GMX2000 graphics card. Each generation of eight variants takes on the order of six seconds to compute and render. This execution time is directly related to the dimensions of the lattice deformer network. Currently the system defaults to a lattice network containing twenty nodes. As each of these nodes is described by three coordinates, the lattice has sixty degrees of freedom.

As a proof-of-concept implementation this is considered to be an acceptable performance. It should be noted that *3-DNA* is currently implemented as a collection of interpreted scripts and the author expects considerable performance increases with the inclusion of compiled routines. In addition, this type of algorithm also lends itself well to parallel processing as the form variants can be computed with separate processing threads.

5 INTEGRATION

The form synthesis engine has increased potential if integrated with a cycle consisting of other digital and physical form development tools. In addition, there are clear benefits from integrating the synthesis tool with engineering and business models. An overview of this integrated system is provided in Figure 5-1.

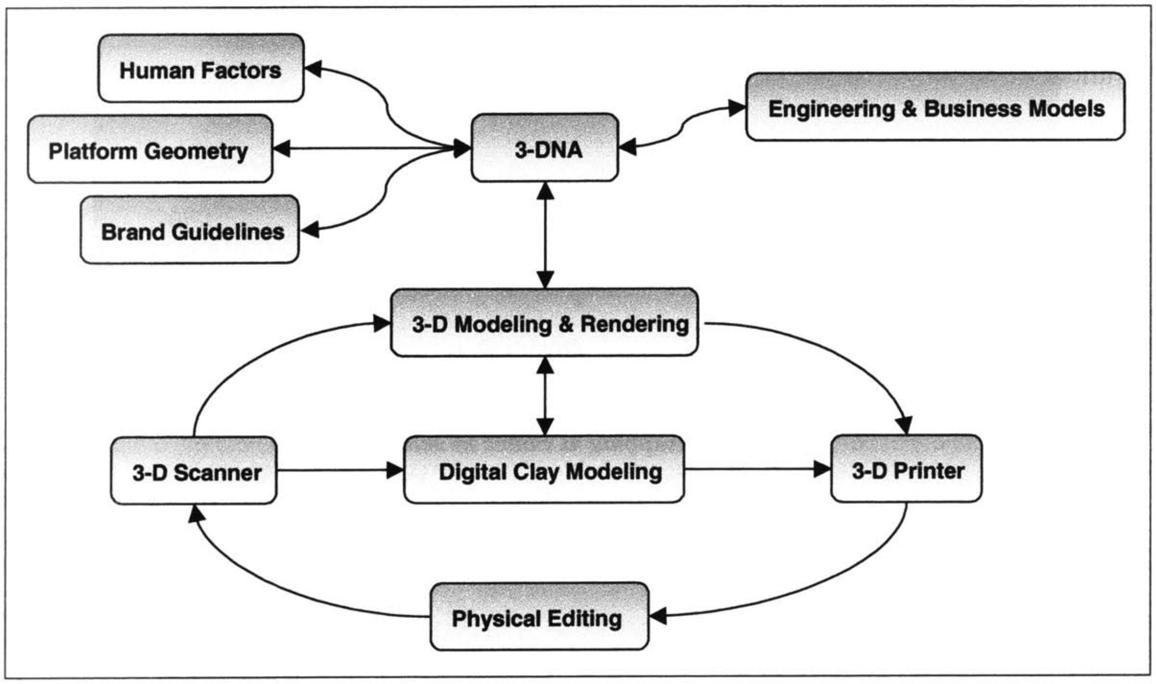


Figure 5-1 Industrial Design Cycle integrated with *3-DNA* and engineering and business models.

5.1 Industrial Design Cycle

The Industrial Design Cycle (Robinson, 1996; Page, 2000) can be generally described as a system that supports the rapid alternation between digital and physical representations of product geometry, as shown in Figure 5-1.

Designers can electronically edit the forms that were generated by the form synthesis system. Rapid prototyping techniques can be used to produce the form, and designers may modify the 3-D artifact. The electronic models are then altered to reflect these changes, thus completing the design cycle. The cycle may be joined at any point. This flexibility is important in supporting the variety of work styles of designers. This is an important benefit when one considers that, although the industrial design cycle exists in a fragmented form in many automotive companies, it is all but inaccessible to designers.

While designers analyze the model, activities such as customer response assessment can be performed with the physical prototype. Any changes suggested by customer and designer review can be made to the physical model and converted back to the digital world. The parameters of these features could be used as inputs to the proposed form synthesis model.

5.1.1 Implementation

The cycle has been prototyped (Page, 2000) using the hardware and software detailed in Table 5-1. Upon integration of the form synthesis engine with the cycle, the system responds, as detailed engineering is resolved, using aesthetic, ergonomic, and manufacturing models in combination with the industrial design archetypes to generate styled design variants.

Table 5-1 Industrial Design Cycle function and corresponding hardware and software components.

Function	Component
Rapid prototyping	Z Corporation Z402™ 3-D printer
Digitization	Steinbichler Tricolite™ 3-D optical scanner
Modeling, animation, rendering	Alias|Wavefront Maya® & AutoStudio® software
Digital sculpting	Sensable Phantom™ force-feedback device & FreeForm™ software

5.1.2 Digital clay

A digital clay application based upon the SensAble Technologies Phantom™ force feedback device (see Figure 5-2) is used to either rapidly sculpt approximate geometry or modify scans. This application uses a voxel representation and is limited in its ability to sculpt manufacturable geometry. Rather it is better suited to a rapid form development loop before surfacing. It can also be used for editing scan models.

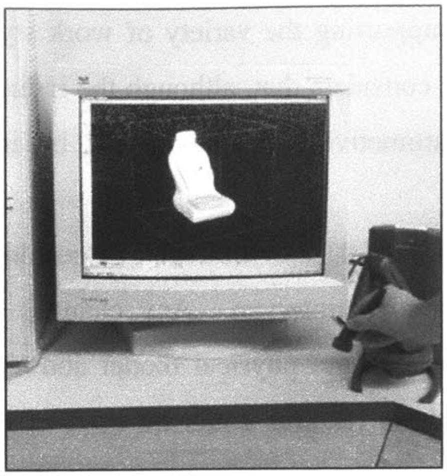


Figure 5-2 Digital clay for coarse geometric modeling.

5.1.3 3-D printing

A Z Corporation Z402™ 3-D printer (see Figure 5-3) is used to rapidly generate physical prototypes from digital models. Parts are post-processed via infiltration with resin, wax or glue.

Figure 5-3 3-D printer for rapid prototyping.

5.1.4 Physical model

The physical model can be evaluated by industrial designers, engineers and potential customers. Changes can be made to the model using traditional forming techniques, as shown in Figure 5-4.

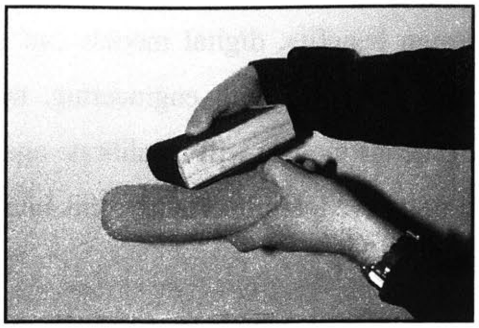


Figure 5-4 Physical prototype made by 3-D printer and edited by hand.

5.1.5 3-D scanning

The Steinbichler Tricolite™ 3-D optical scanner digitizes physical models as shown in Figure 5-5. It is used widely in the automobile industry for clay model scanning because of its ability to achieve a high resolution point spacing on the order of 0.2 mm for up to one-third scale models. It is able to do this by scanning in patches. The trade-off for this flexibility is the increased user time required to merge, or register, the patches using software. This patch metaphor is well-suited to the Industrial Design Cycle as this means that only the edited portion of a physical model must be scanned - the remaining patches are unchanged. It may be possible to generate brand styling guidelines by scanning edited patches as the designer works and translating these edits into a styling feature construction history.

Figure 5-5 3-D scanning of physical model. Insert: 3-D optical scanner.

5.2 DOME

In addition to industrial design benefits, digital models can at all times be supplied to other design participants and integrated with engineering, manufacturing and business models. In this way heterogeneous models will recalibrate against each other to maintain a consistent product system image. Most importantly, this integration permits many more design iterations per unit time.

5.2.1 DOME overview

The Distributed Object-based Modeling Environment (DOME) is a software system that enables distributed heterogeneous mathematical models to exchange parameter data and perform operations such as optimization on that data (Senin *et al.*, 1996). To date, the system has been used in the engineering and business modeling of a projector (Abrahamson *et al.*, 1999) and automobile Moveable Glass Subsystem (MGS) (Abrahamson *et al.*, 2000). A goal of this work is to develop a form synthesis model that supports the integration of engineering, business and aesthetic models. The DOME system was used to connect the parameter sets of various models.

5.2.2 Integration example

Although the ultimate intention is to directly incorporate external hard constraints into the evolution of aesthetic form, integration was first performed between DOME and the Maya application without *3-DNA*. This integration is illustrated by way of an extension of the MGS project initially described in Abrahamson *et al.* (2000). A parameterized surface model of the automobile door is integrated with existing engineering and business models. Parameters were associated with the Maya surface model of the automobile door and then connected via DOME to the SDRC IDEAS® CAD solid model of the door's glass subsystem shown in Figure 5-6.

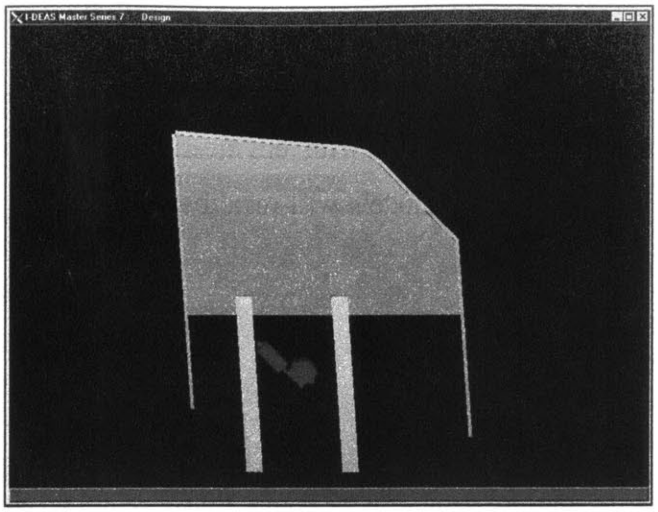
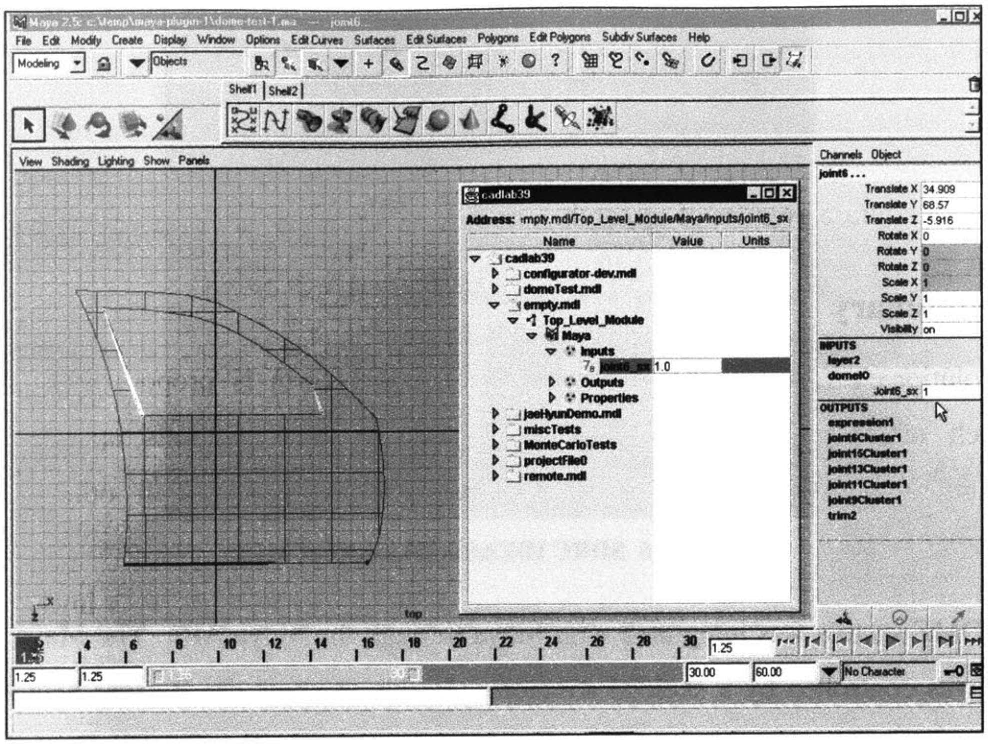


Figure 5-6 SDRC IDEAS solid model of MGS.

The DOME system was integrated with Maya through the Maya API. The Maya surface model was controlled by a skeletal structure. In this example the B-pillar height of the door is adjusted in DOME and then reflected in Maya. The B-pillar height of the door surface model is controlled by a scaleable bone and two joints (see Figure 5-7). Figure 5-7 and Figure 5-8 show both the Maya and DOME models before and after an increase of the B-pillar height respectively. Although skeletal handles have not yet been implemented in *3-DNA*, it is expected that these handles will provide a mechanism for a shared parameterization between surface modelers and other design tools, as illustrated by this example.



cadlab39

Address: ›mpty.mdl/Top_Level_Module/Maya/inputs/joint6_sx

Name	Value	Units
cadlab39		
configurator-dev.mdl		
domeTest.mdl		
empty.mdl		
Top_Level_Module		
Maya		
Inputs		
joint6_sx	1.0	
Outputs		
Properties		
jaeHyunDemo.mdl		
miscTests		
MonteCarloTests		
projectFile0		
remote.mdl		

Channels Object

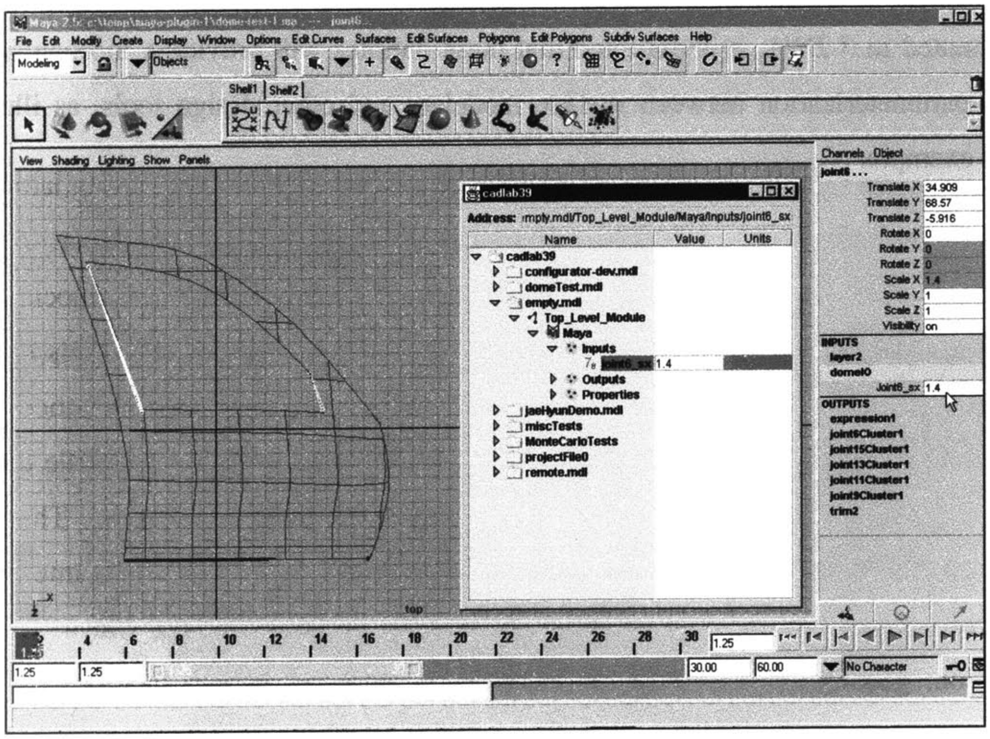
joint6 . . .

Translate X 34.909
Translate Y 68.57
Translate Z -5.916
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on

INPUTS
layer2
dome10
Joint6_sx 1

OUTPUTS
expression1
joint6Cluster1
joint15Cluster1
joint13Cluster1
joint11Cluster1
joint9Cluster1
trim2

Figure 5-7 Surface model of door in Maya prior to changes in B-pillar height. Maya surface model is controlled via a skeletal structure. Cursor points to scale of B-pillar height in Maya, while inserted DOME interface contains a tree view of the model, with scale of B-pillar height in DOME highlighted.



cadlab39

Address: ›mply.mdl/Top_Level_Module/Maya/inputs/joint6_sx

Name	Value	Units
cadlab39		
configurator-dev.mdl		
domeTest.mdl		
empty.mdl		
Top_Level_Module		
Maya		
Inputs		
joint6_sx	1.4	
Outputs		
Properties		
jaeHyunDemo.mdl		
miscTests		
MonteCarloTests		
projectFile0		
remote.mdl		

Channels Object

joint6 . . .

Translate X 34.909
Translate Y 68.57
Translate Z -5.916
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1.4
Scale Y 1
Scale Z 1
Visibility on

INPUTS
layer2
dome10
Joint6_sx 1.4

OUTPUTS
expression1
joint6Cluster1
joint15Cluster1
joint13Cluster1
joint11Cluster1
joint9Cluster1
trim2

Figure 5-8 B-pillar height scaled by 1.4 in DOME. Maya door model deforms as change propagates from DOME to Maya.

6 CONCLUSION

This chapter summarizes the research contribution, highlights limitations of the approach and suggests future avenues of research.

6.1 Summary

An evolutionary model for the synthesis of product form is proposed. This novel approach offers a number of benefits over other form synthesis approaches and deterministic modeling techniques. The principal difference with other synthesis approaches is that the system is intended for designing parametric surfaces. Surfaces are critical to the design of aesthetic form. Other 3-D form synthesis tools invariably use solid model representations because generated geometry can be easily specified in terms of primitive parameters. The proposed approach also compliments existing parametric surface design tools well. The underlying NURBS representation is compatible with these commercially available tools. Therefore the form synthesis engine can be used at any point in the surface design to explore a region of form space. It is important to note that, although the system manipulates NURBS geometry, the designer is not required to understand how to explicitly create surface geometry at each design iteration.

A proof-of-concept implementation, *3-DNA*, is described. An example using lattice form handles is illustrated including the mutation and mating of an automobile door. Although the evolved doors may not possess the symmetry and regularity typically associated with product forms, the use model is indicated. As discussed in Section 6.3.2, this regularity may be improved by creating genetic operators that manipulate subsets of lattice handles and not individual lattice nodes. A skeleton-driven version of the automobile door was integrated using DOME with a CAD solid model of the automobile door. The surface model and the solid model were then able to drive each dynamically. It is intended that the form synthesis model will then be able to communicate through Maya with other simulation, geometric, marketing, manufacturing, cost and human factors models. In addition the synthesis system has increased potential if incorporated into an Industrial Design Cycle that supports the rapid alternation between digital and physical representations of product geometry. Work is underway on validation of the combined form synthesis model and Industrial Design Cycle with an appropriate industrial application. In light of the implementation and integration of the system, several enhancements to the system have been identified and these are detailed in Section 6.3.

6.2 Limitations of the Approach

The choice of geometric representation is invariably a trade-off between many factors. This work has attempted to synthesize parameterized mathematical surfaces. NURBS surfaces are intended for designing freeform surfaces suitable for manufacturing of product interiors and exteriors. However, the drawbacks of this representation include the need to stitch the geometry. To date all attempts to synthesize form known to the author have used voxel or solid representations. Solid and voxel models lend themselves much more to the spatial partitioning problems associated with the generation of form. While those approaches have generality and are valuable for proof-of-concept demonstrations, they do not support the creation of manufacturable freeform geometry. This work has deliberately sacrificed some generality in order to maintain a representation suitable for aesthetic surfaces.

6.3 Future Work

A number of interesting avenues of future research have emerged during the course of this endeavor. These are outlined below. The subsections are arranged from incremental extensions to completely new additions.

6.3.1 Search steering

Currently many generations are required to successfully traverse form space. The form search process can be accelerated in number of ways. Todd and Latham (1992) suggest that ranking of the phenotypes at each generation produces a direction vector analogous to the direction of steepest ascent. This method is described in detail in Section 3.3.3

6.3.2 Regular deformation

One of the drawbacks of the results presented is that the synthesized forms are irregularly freeform and lack visual balance. Although aesthetic product surfaces are often freeform, they still maintain some intangible regularity. Introduction of this regularity may be possible through the manipulation of subsets of handles and not individual handles. Work is underway on the design of genetic operators to achieve this end.

6.3.3 Archetype models

Different handle models may be used to deform the parametric surfaces. These include wire, wrap and cluster deformers. These different handles will provide a variety of deformation effects.

6.3.4 Structural evolution

For reasons of simplicity, the current implementation assumes a fixed length gene vector and does not allow the form variants proposed to vary outside of this initial structure. Although not necessarily a disadvantage, this restricts the structure of the form evolved to the original archetype. It may be interesting to alter the structure of the form during evolution. Sims (1991) experimented with LISP expressions to create variable length gene vectors that allowed new dimensions to be added dynamically to the texture space. Altmann (1995) provides an overview of the differences between fixed parameter space evolution and symbolic expression evolutionary simulation as tried by Sims.

Another approach to dynamic structures may be to allow Boolean operations to be performed on NURBS primitives. The choice of operation can itself be a parameter randomly generated in the process. These extensions may yield interesting new product archetypes.

6.3.5 Evolutionary deformer

The evolutionary model described acts globally. An interesting avenue of investigation may be a more localized approach. It is envisioned that an evolutionary deformer could be developed which has a prescribed region of influence. In this way there could be a certain degree of local control, which would be useful for situations where the designer is satisfied with the overall form, but wishes to make local modifications.

6.3.6 Functional optimization

Although the system presented has been solely concerned with aesthetic concerns, a valuable extension to the system would allow functional optimization. For example, a designer could ask the system to produce several concept variants for aesthetic evaluation which have already been optimized for some function such as wind drag minimization (Bentley, 1999).

The work of the FIORES (Formalization and Integration of an Optimized Reverse Engineering Styling workflow) research consortium may have application to functional optimization of surfaces (Dankwort and Podehl, 1998). The group is attempting to quantitatively define the aesthetic properties of shapes using the analogy of mechanical features and control these properties using objective criteria in an Engineering in Reverse (EiR) workflow. In this context, Bosinco *et al.* (1998) propose a method for the modification of shapes independent of the underlying geometric representation. The findings of the FIORES group may provide the algorithms necessary for integrating functional optimization with the evolutionary synthesis of aesthetic product surfaces.

6.3.7 Procedural modeling

It would appear from the example of the box corner treatments in Section 4.3 that procedural models are too specific for general-purpose modeling because each new product archetype requires a specific 'hard-coded' model. However there may be potential in limiting procedural model to the basic structure of the product and enabling the designer to build the procedural model interactively. In this way, the procedure can be used to describe the skeleton of the form as a sequence of bounding volumes stitched together to form a contiguous surface. Detailed skinning of the underlying procedurally-generated block model could be handled by two dimensional image maps. These maps can be created from cluster deformer weighting maps described in Alias|Wavefront (1999).

6.3.8 Extrusion

Another interesting avenue of research would be to use profile curves to extrude either along a straight or curved axis. The system could be integrated with a sketch input system (Overbeeke *et al.*, 1997; Van Dijk and Mayer, 1997). The designer sketches two types of curves for the seed form: section and profile. These curves can either be sketched directly in a drawing program and exported in a vector representation to a modeling program or the curves can be sketched on paper, scanned and converted from raster graphics to vector graphics in a an image manipulation software package. The section curve is then extruded along the profile curve. The evolutionary model then acts by varying the extrusion profile curves. In this way several variants can be suggested and then design iterations can take place. It should be noted that this approach is best suited to the generation of geometry in two and a half dimensions. Further work needs to be carried out to extend this approach to true three-dimensional space. It is anticipated that topology limitations would be a significant drawback of such a system.

6.3.9 Feature recognition

Existing styling features could be scanned and feature parameters extracted from the point clouds using a combination of reverse engineering and feature detection techniques. Baker *et al.* (1998) succeeded in detecting common parametric features from light intensity information and it seems plausible that such approaches may be extensible to the range images generated by a 3-D optical scanner. The medial axis theorem (Blum, 1967) could be used to generate skeletal spines from the tessellated point cloud data. The set of parameters associated with the transformations of the skeleton constitute the gene vector. This skeleton generation may also benefit from the application of fuzzy logic.

There are several compelling reasons to extend the form synthesis interface in this way. Legacy products and other forms could be used as inputs or seed forms to the evolutionary process. Analogous to the use of reverse engineering in the reconstruction of CAD data, this approach would also allow for the speedy construction of styling features. This approach has the interesting possibility of allowing a library of meta-models of brand features to be rapidly created and subsequently evolved.

7 REFERENCES

Abrahamson, S., Wallace D., Senin, N., Borland, N., (1999) "Integrated Engineering, Geometric, and Customer Modeling: LCD Projector Design Case Study", *Proceedings of the ASME DT Conferences*, DETC/DFM-9084, September 1999, Las Vegas, Nevada.

Abrahamson, S., Wallace, D., Senin, N., Sferro, P., (2000), "Integrated Design in a Service Marketplace", *Computer-aided Design,* **32**(2), 97-107.

Alias|Wavefront, (1999), *Using Maya: Character Setup*, Alias|Wavefront, Toronto, Canada.

Altmann, M. (1995), *Evolutionary Procedural Modeling with Symbolic Expressions*, http://www.cs.wpi.edu/~matt/courses/cs563/talks/evolution/evolut.html.

Bajaj, C., (1993), "The Emergence of Algebraic Curves and Surfaces in Geometric Design", in Martin, R. (Ed.), *Directions in Geometric Computing*, 1-29. Information Geometers Press.

Baker, S., Nayar, S.K., Murase, H., (1998), "Parametric Feature Detection", *International Journal of Computer Vision*, **27**(1), 27-50.

Bedwell, E., Ebert, D., (1998), "Artificial Evolution of Implicit Surfaces", *ACM SIGGRAPH Technical Sketch*, July 1998.

Bedwell, E., Ebert, D., (1999), "Artificial Evolution of Algebraic Surfaces", *Proceedings Implicit Surfaces '99*, September 1999.

Bentley, P. (1999), *Evolutionary Design by Computers*, Morgan Kaufmann, San Francisco, California.

Blum, H., (1967), "A Transformation for Extracting New Descriptors of Shape", *Models for the Perception of Speech and Visual Form*, Wathen-Dunn, W. (Ed.), 362-381, MIT Press.

Bosinco, P., Durand, G., Goussard, J., Lieutier, A., Massabo, A., (1998), "Complex Shape Modifications", *IDMME Conference*, May 1998.

Cagan, J., Argawal M, (1998), "A Blend of Different Tastes: the Language of Coffeemakers", *Environment and Planning B: Planning and Design,* **25,** 205-226.

Coquillart, S., (1990), "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling", *Computer Graphics (Proceedings of SIGGRAPH '90)*, **24**(4), 187-196, August 1990, Dallas, Texas.

Dankwort, C.W., Podehl, G. (1998), "A New Aesthetic Design Workflow - Results from the European Project FIORES", *Dagstuhl Seminar 98461: CAD Tools and Algorithms for Product Design*, International Conference and Research Center for Computer Science, November 1998, Schloss Dagstuhl, Wadern, Germany.

Dawkins, R., (1986), *The Blind Watchmaker*, Penguin Books.

Faux, I.D., Pratt, M.J., (1979), *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., Chichester, England.

Fontana, M., Giannini, F., Meirana, M., (1999), "A Free Form Feature Taxonomy", *Computer Graphics Forum*, **18**(3), 107-118.

Goldberg, D.E., (1989), *Genetic algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts.

Heisserman, J., Woodbury, R., (1994), "Geometric Design with Boundary Solid Grammars", *Ifip Transactions B: Computer Applications in Technology*, **B-18**, 85-105.

Hofstadter, D.R., (1985), *Metamagical Themas*, Basic Books Publishers, New York.

Hoschek, J., Lasser, D., (1993), *Fundamentals of Computer Aided Geometric Design*, A.K. Peters, Wellesley Massachusetts.

Hsiao, S.W., Cheng, M.S., (1996), "Form Creation for Automobile Design", *International Journal of Vehicle Design*, **17**(4), 360-363.

Hsiao, S.W., Wang, H.P., (1998), "Applying the Semantic Transformation Method to Product Form Design", *Design Studies*, **19**(3), 309-330.

Jeanneret-Gris, C.E., (1954), (a.k.a. Le Corbusier), *The Modulor*, M.I.T. Press, Cambridge Massachusetts and London England.

Knoop, W.G., Van Breemen, E.J.J., Horváth, I., Vergeest, J.S.M., Pham, B., (1998), "Towards Computer Supported Design for Aesthetics", *Proceeding of 31st International Symposium on Automotive Technology & Automation Conference,* June 1998, Düsseldorf, Germany, pp.403-412.

Lindenmayer, A., (1968), "Mathematical Models for Cellular Interactions in Development, " *J. Theor. Biol.*, **18**, 280-315.

Overbeeke, C., Kehler, T., Hummels, C., Stappers, P.J., (1997), "Exploiting the Expressive: Rapid Entry of Car Designers' Conceptual Sketches into a CAD Environment", *Proceedings of 30th International Symposium on Automotive Technology & Automation*, Florence, Italy, pp.243-250.

Page, M., (2000), *Blending Engineering Modeling, Industrial Design, and Physical Prototyping in Product Design*, Master of Science Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Prusinkiewicz, P., Lindenmayer, A., Hanan, J., (1988), "Developmental Models of Herbaceous Plants for Computer Imagery Purposes", *Computer Graphics (Proceedings of SIGGRAPH '88)*, **22**(4), 141–150, August 1988, Atlanta, Georgia.

Robinson, D.W., (1996), *Integration of Rapid Prototyping and Reverse Engineering for Complex 3D Shape Design*, Master of Science Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Rowland, K., (1965), "The Shapes We Need: Looking and Seeing 3", Ginn and Company, London, England.

Senin, N., Wallace, D.R., Jakiela, M.J., (1996), "Mixed Continuous Variable and Catalog Search Using Genetic Algorithms", *Proceedings of the ASME Design Automation Conference*, 96-DETC/DAC-1489, August 1996, Irvine, California.

Sims, K., (1991), "Artificial Evolution for Computer Graphics", *Computer Graphics (Proceedings of SIGGRAPH '91), 25(4),* 319-328, July 1991, Las Vegas, Nevada.

Sims, K., (1994), "Evolving Virtual Creatures", *Computer Graphics (Proceedings of SIGGRAPH '94)*, 15-22, July 1994, Orlando, Florida.

Singh, K., Fiume, E., (1998), "Wires: A Geometric Deformation Technique", *Computer Graphics (Proceedings of SIGGRAPH '98)*, 405-414, July 1998, Orlando, Florida.

Smith, A.R., (1984) "Plants, Fractals, and Formal Languages", *Computer Graphics, (Proceedings of SIGGRAPH '84)*, **18**(3), 1-10, July 1984, Minneapolis, Minnesota.

Stiny, G., (1980), "Introduction to Shape and Shape Grammars", *Environment and Planning B* **7,** 343-351.

Taura, T., Nagasaka, I., Yamagishi, A., (1998), "Application of Evolutionary Programming to Shape Design", *Computer Aided Design,* **30**(1), 29-35.

Thompson, D.W., (1961), *On Growth and Form,* Bonner, J.T. (Ed.), Cambridge University Press.

Todd, S., Latham, W., (1992), *Evolutionary Art and Computers*, Academic Press.

Ugail, H., Bloor, M.I.G., Wilson, M.J., (1999), "Techniques for Interactive Design using the PDE Method", *ACM Transactions on Graphics*, **18**(2), 195-212.

Van Breemen, E.J.J., Horváth, I., Knoop, W.G., Vergeest, J.S.M., Pham, B., (1998), "Developing a Methodology for Design for Aesthetics Based on Analogy of Communication", *Proceeding of the ASME 1998 Design Theory and Methodology Conference*, September 1998, Atlanta, Georgia.

Van Breemen, E., Sudijono, S., Horváth, I., (1999), "A Contribution to Finding the Relationship between Shape Characteristics and Aesthetic Appreciation of Selected Products", *International Conference on Engineering Design (ICED '99)*, August 1999, Munich, Germany.

Van Dijk, C.G.C., Mayer, A.A.C., (1997), "Sketch Input for Surface Design", *Computer in Industry*, **34,** 125-137.

Van Elsas, P.A., Vergeest, J.S.M., (1998), "Displacement Feature Modelling for Conceptual Design", *Computer-Aided Design*, **30**(1), 19-27.

Wallace, D.R., (1991), *A Computer Model of Aesthetic Industrial Design,* Master of Science Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Wallace, D.R., Jakiela, M.J., (1993), "Automated Product Concept Design: Unifying Aesthetics and Engineering", *IEEE Computer Graphics & Applications*, **13** (4), 66-75.

Watt, A., Watt, M., (1992), *Advanced Animation and Rendering Techniques*, ACM Press, New York.

Wyvill, G., McRobie, D., Haig, C., McNaughton, C., (1996), "Free Form Modeling with History", *International Journal of Shape Modeling*, **2**(4), 275-282.

Wyvill, G., McRobie, D., Gigante, M., (1997), "Modeling with Features", *IEEE Computer Graphics and Applications*, **17**(5), 40-46.