Evolving product form designs using parametric shape grammars integrated with genetic programming

Ho Cheong Lee and Ming Xi Tang

Artificial Intelligence for Engineering Design, Analysis and Manufacturing / Volume 23 / Issue 02 / May 2009, pp 131 - 158
DOI: 10.1017/S0890060409000031, Published online: 13 November 2008

Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 131–158. Printed in the USA.

doi:10.1017/S0890060409000031

Evolving product form designs using parametric shape grammars integrated with genetic programming

HO CHEONG LEE AND MING XI TANG

Design Technology Research Centre, School of Design, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, People's Republic of China

(RECEIVED January 15, 2007; ACCEPTED December 14, 2007)

Abstract

The two critical issues related to product design exploration are addressed: the balance between stylistic consistency and innovation, and the control of design process under a great diversity of requirements. To address these two issues, the view of understanding product design exploration is first sought. In this view, the exploration of designs is not only categorized as a problem-solving activity but also as a problem-finding activity. A computational framework is developed based on this view, and it encompasses the belief that these two activities go hand in hand to accomplish the design tasks in an interactive design environment. The framework adopts an integration approach of two key computational techniques, shape grammars and evolutionary computing, for addressing the above two critical issues. For the issues of stylistic consistency, this paper focuses on the computational techniques in balancing the conflicts of stylistic consistency and innovation with shape grammars. For the issues of controlling design process, the practical concerns of monitoring the design process through various activities starting from the preparation works to the implementation of shape grammars have been emphasized in the development of this framework. To evaluate the effectiveness of the framework, the experiments have been set up to reflect the practical situations with which the designers have to deal. The system generates a number of models from scratch with numerical analysis that can be evaluated effectively by the designers. This reduces the designers' time and allows the designers to concentrate their efforts on performing higher level of design activities such as evaluation of designs and making design decisions.

Keywords: Configuration Designs; Evolutionary Shape Grammars; Genetic Programming; Interactive Grammar-Based Design Systems; Product Designs

1. INTRODUCTION

1.1. Product design

Product design involves complex activities in which designers, engineers, and manufacturers have to cooperate in specifying design problems and developing flexible solutions. The activities can be classified into physical and decision-making (cognitive) activities from which the customer requirements are transformed into a realized product. The main activity is the determination and specification of the parts of a product and their interrelationships. The assembly of the parts performs specific functions. The assembly itself is packaged with an attractive exterior. Performing these activities imply that product designers have to synthesize new ideas and utilize their knowledge and skills from various domains such as mathematics, sciences, and manufacturing to determine and foresee how the new designs are to be used before the products are manufactured.

The key indispensable tool in product design is the use of computers in assisting designers, engineers, and manufacturers in the design process. Research and developments in the application of computer-aided design systems integrated with artificial intelligence techniques in enhancing the product design process are demanding. Recently, research in exploring the shape grammar approach to product design has received more and more attention by many researchers. For example, Cagan and colleagues developed the coffeemaker grammar, motorcycle grammar, hood panel grammar, and vehicle grammar (Agarwal & Cagan, 1998; Cagan, 2001; McCormack & Cagan, 2002; Pugliese & Cagan, 2002; Orsborn et al., 2006).

This research focuses on the development of a computational framework that integrates two computational techniques: shape grammars and evolutionary computing for

Reprint requests to: Ho Cheong Lee, Design Technology Research Centre, School of Design, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, People's Republic of China. E-mail: jackielee2005@gmail.com

supporting product design activities. The issues related to how such integration should be developed with a philosophical concept to be evaluated with product design experimentations are identified in the following section.

1.2. Problem identification

The computational framework is developed and evaluated based on two critical issues in the product design: stylistic consistency and control of design process.

1.2.1. Stylistic consistency

Because of fierce market competition in the industry, the company faces challenges to launch new products to the market periodically. The launched products may not necessary be major technological breakthrough products, but should have new features that add values to the products. For example, the new features of the products may be designed as an integration to reflect a new particular style. Product designers try to balance stylistic consistency and technical (and or stylistic) innovation for the new designs. The management of stylistic consistency is one of the critical issues in product design. The difficulties arise from maintaining the brand image while introducing new design features to the products. If the new product form has been radically modified, then it may not be consistent with the brand image from consumer perceptions.

The use of shape grammar to generate products with consistent styles is a definite advantage. Shape grammar encodes the stylistic characteristics of a particular product style into different sets of shape grammar rules. With an existing shape grammar, the designers can select a particular rule from each branch of subset of rules. The resulting product design can therefore be generated with certain design characteristics of a particular style. However, the use of a shape grammar to control the evolution of product style is a difficult task. Any radical modification to the shape grammars removes the style consistency of a product that is used originally for deriving them.

In our research, an integrated representation of shape grammars and genetic algorithm is introduced to facilitate the formulation and modification of shape grammars for a particular style. The new representation and control strategies are developed for generating stylistic consistent and new designs.

1.2.2. Control of design process

Solving design problems is so complex in nature that designers have to make decisions on implied information that is not readily available. The implied information refers to inconsistent specifications, over- or underconstrained conditions defined at the beginning of and during the design process. Simon (1984, 1990) determined that the design problems become “ill-defined” in this situation. Most ill-defined problems have uncertain characteristics when it comes to defining the problems, their possible solutions, and even the methodologies in obtaining those solutions. It can be seen that solving and specifying design problems cannot be done easily by designers in a straightforward manner. The successful chance of conquering design problems can become higher if there are methodologies provided to monitor the design process.

Apart from the well-known advantages of the emergent property, which is a key property of shape grammar, described in the research literature and Stiny’s recent book (Stiny, 2006), another significant advantage of applying shape grammar to design applications is the control of design process. A shape grammar is derived systematically to generate designs that fulfill specific requirements. Once a shape grammar is developed, the designers can determine appropriate rules for applications. However, this kind of control process is limited because it assumes that the design problems can be easily solved and specified by designers in a straightforward manner.

A control strategy using genetic programming as a control mechanism to monitor the design process has been developed. The control strategy was developed based on the view that the exploration of design is not only treated as a problem-solving activity but also a problem-finding activity. The control strategy determines the rate of modification for the blueprints of designs. By selecting appropriate control strategies, designers can observe the generation of designs and continuously modify the existing shape grammars interactively. The designers can play a major role in the design process through this control mechanism in an interactive manner.

Section 2 provides an overview of related research on shape grammars. Section 3 presents the development of the evolutionary grammar-based design framework in product design domain. Section 4 illustrates the implementation of this framework using compact digital cameras as examples. Section 5 analyses the implementation results. Section 6 discusses some critical issues. Finally, Section 7 draws conclusions.

2. RELATED WORK

The framework in this research is developed to manage the complex activities involved in product design. The related work of applying shape grammars to design is first appraised as possibilities to enlarge and strengthen the philosophical concept of this framework. The related work is divided into two parts to cover the two critical issues: stylistic consistency and control of design process.

2.1. Stylistic consistency

Before developing a shape grammar to generate product designs with a particular style, the issues related to the definition and creation of product styles have to be addressed. It can be traced from Stiny’s seminal work, which demonstrated in architectural design domain that shape grammars captured styles of designs, generated stylistically consistent designs, and novel designs (Stiny, 1980*a*, 1980*b*). Two activities are involved in his approach: encoding the knowledge of design process into shape grammars by analyzing the existing sets of designs so as to reproduce these designs, and exploration of new designs from the stylistically consistent languages.

According to Stiny's seminal work (Stiny, 1980*a*, 1980*b*), some questions can be raised to give an insight into the problems of stylistic consistency in product design domain. There are questions like how to define a style. What elements constitute a particular style? Are there any concerns about historical and cultural background when defining a product style? What are the views of customers, salesmen, manufacturers, designers for the particular product style? In addition, encoding a "style" into shape grammar rules is subjective to the shape grammar developers or designers. The shape grammar developer and designers cooperate to analyze the existing products and convert the distinctive characteristics into the rules in accordance to their knowledge and experience.

In the architectural design domain, Li (2004) has developed a parametric shape grammar for the enhancement of understanding styles in architecture. The shape grammar rules were developed in accordance to the analysis of a 12th-century Chinese building manual, the *Yingzao Fashi*. To understand a style of a particular architecture, the designers interpreted which designs generated by the shape grammars were stylistically correct. A standard of stylistic correctness can then be developed by refining the shape grammars to eliminate the generation of stylistically incorrect designs and produce only stylistically correct designs.

The issue raised by Li (2004), which is concerned with the cultivation of a standard for maintaining the stylistic consistency, is another insight to product design. A product style represents the image of a brand identity when companies promote their products with certain key identifiable characteristics. Usually, a family of products with a particular style is delivered for particular marketing customers. For example, the strategies of a company may be to launch a series of mobile phones targeted on teenagers. The family of products has a good interface, and can be changed with different color outlooks. Maintaining the stylistic consistency among a family of products can be used as a company strategy in promoting the brand image.

In recent years, research on shape grammars to address every issue of stylistic consistency in brand management has become popular. For example, Pugliese and Cagan (2002) developed a shape grammar for designing motorcycles with historical, contemporary, or a particular style to capture brand identity. Ang et al. (2006) provided another example in which the issues of designing branded products were investigated. Ang et al. (2006) applied an evolutionary algorithm to evolving a set of two-dimensional (2-D) shape grammar rules for the generation of Coca-Cola bottles. The evolved shape grammar rules are executed in sequence and associated with parameters to generate bottles with Coca-Cola styles. The bottles generated fulfill specific volume requirements.

Pugliese and Cagan (2002) and Ang et al. (2006) provided a significant insight to the modification of product styles under the constraints of avoiding the distortion of the brand image for new product development. A style is maintained by means of converging the languages defined by the shape grammar and the language of stylistically correct designs. Conflicts exist when designers try to explore designs by changing the languages defined by the shape grammars. Because the style of the generated results is interpreted subjectively by the designers and customers, this introduces difficulties to the designers. For a simple case like the regular type style, it can easily be identified from the results. For a complex case like cultural and brand identity style, it is hard to determine the structural organization of components and the variation of forms from the results. As a result, designers will have difficulties in balancing the effects of visual change of products among the brand image and customer perceptions of quality, service, and the intangible associations.

This paper addresses the issues related to the formulation of shape grammars for a particular style, and the modification of shape grammars for new defined styles. The focus is on the development of an integrated representation of shape grammars and genetic algorithms that addresses the technical parts of the formulation of shape grammars for a particular product style. The new representation proposed facilities the modification of shape grammars for generating stylistic consistent or new designs.

2.2. Control of design process

The design problems are rather complex and difficult to be managed by designers because of the ill-defined nature of design. Janssen et al. (2002) explained that the nature of ill-defined design problems is unstructured, and the solutions are in a vast multidimensional design space. Different views of design process from the literature that outlines a broad scope for specifying and solving design problems are reviewed. These views range from designating design problems as search problems (Kanal & Kumar, 1988), to exploring alternative possible solutions (Frazer, 2002; Janssen, 2004). These suggest contrary views of finding and solving design problem activities in both simplifying and complicating the design tasks. Specifying appropriate design problems with the right kind of abstractions and correctness, and proper interpretation by designers also lead to diverging the scope of design problems (Dorst, 2006). Relevant shape grammar approaches under each view can be found, and their distinctive characteristics can be identified as an insight to monitoring the design process.

2.2.1. Design as searching process

From the view of design as a searching process, Kanal and Kumar (1988) simplified the design problems as search problems in optimizing the solutions. The improvement of designs is achieved by searching among a selection of some well-known and near-optimal solutions. This kind of searching metaphor aims at elucidating the design process. Under this view, shape grammars incorporated with optimization techniques have shown tremendous impacts on different applications.

In architectural design domain, Çağdaş (1996) employed an integration model of a depth-first search method and a shape grammar for the design of row houses. The generative capability of grammar and the reasoning capabilities of knowledge-based systems are utilized in guiding the generation of design solutions from a high-level abstraction to a low-level abstraction.

Sourav and Michael (1996) presented the integrated network and genetic algorithm model to compute shape grammars. This approach led to the exploitation of aspects of knowledge representation and directed search within the network. Real instances of nonbisymmetric Palladio villas via a shape grammar were generated in the background of cultural expressions.

In urban planning design domain, Duarte et al. (2006) developed a parametric shape grammar for urban planning. The shape grammar captured the knowledge of creating some features of the existing urban fabric. A large amount of work was put into historical analysis and fieldwork for the derivation of useful shape grammar rules.

In structural engineering design domain, Shea and Cagan (1999) used shape annealing, a combination of shape grammar formalism and simulated annealing, to design structures. The concept of search process in simulated annealing was borrowed from physical processes. More details of this approach can be referenced to the Shea's research works (Shea, 1997, 2001, 2002, 2004). Particular examples like traditional geodesic patterns have been constructed using shape grammars. Apart from these particular examples, a wide range of other examples of space frame structures have been constructed using this approach. To illustrate the generative capability of shape grammar, Shea (1997) demonstrated that this approach was capable of generating three space frame roof structures for an octagonal air plane hanger with walls that vary in heights. Another example using this approach was in generating truss structures (Shea & Cagan, 1999).

In mechanical engineering design domain, earlier attempts in merging grammars with optimization techniques have been achieved by Schmidt and Cagan (1998), aiming to direct grammatical generation by design goals. These attempts have led to success in the generation of optimal mechanical systems. Other examples include the generation of optimized process plans for machining designs done by Brown and Cagan (1997). The process plans were defined by a language of machining parts that were derived by Brown et al. (1995).

2.2.2. *Design as exploration process*

From the view of design as an exploration process, Janssen et al. (2002), on the contrary, has criticized the searching metaphors for solving design problems: "However, they (the searching metaphors) do not accurately reflect the reality of the design process and thereby actually result in further confounding the issue" (Janssen, 2004). Frazer (2002) has identified such confounded issues: "This is why it is misleading to talk of design as a problem solving activity—it is better defined as a problem finding activity. This has been very frustrating for those trying to assist the design process with computer-based problem-solving techniques. By the time the problem has been defined, it has been solved. Indeed the solution is often the very definition of the problem" (Frazer, 2002). Under this view, two key approaches are adopted: shape grammars incorporated with exploration techniques are used instead of optimization techniques for different applications, and the use of emergent property of shape grammar.

In the architectural domain, an attempt was made by Gero (1992) and Gero et al. (1994) to provide possible architectural grammars with conventions for configurative mechanisms, to parse historical shape evolution. Shape evolution can be an architectural or a vernacular building style. New rules and rule sets can be evolved in an environment of already given rule sets and procedures of evaluation. The process of the evolution was studied as the simulation of knowledge acquisition.

Other attempts have been made by Rosenman (1996, 2000) and Rosenman and Gero (1999) to extend the generative capability of shape grammar by integrating the evolutionary algorithms with shape grammar-based design systems. Two-dimensional orthogonal plan shape grammar rules for buildings have been derived and encoded as genetic representation. The genetic representation includes genotype, which encodes the selection and application of a set of shape grammar growth rules. The evolutionary algorithms evolve the genetic representation of the shape grammar rules that make small modifications to an existing plan to generate new plans.

With an example of designing a facade, Gero and Ding (1997) applied evolutionary grammars to explore emergent styles in architectural designs. The work was subsequently refined to show that the style can be captured from a language model using genetic representation (Ding & Gero, 2001).

In product design domain, an evolutionary architecture was integrated to an interactive shape grammar-based design system to enhance its generative capability (Lee & Tang, 2004). A parametric 2-D shape grammar with labels was used to generate three-dimensional (3-D) objects by extrusion of 2-D profiles. The shape grammar rules were composed of rule numbers, labeled shapes, and their corresponding parameters. Each shape grammar rule was classified into three main groups: exterior of the product form generation, component generation, and configuration generation. The execution of each shape grammar rule followed an ordered sequence that was determined by a genetic algorithm. The control parameters of the labeled shapes and rule numbers were encoded as code scripts of the genetic algorithm. The genetic algorithm evolved the shape grammar rules to generate both the existing and novel designs.

Agarwal and Cagan (1998) developed the coffeemaker grammar that generated novel designs using function labels to maintain proper function to form sequences. The coffeemaker grammar was further developed by incorporating a decision-making method in which the grammar rules are associated with cost expressions (Agarwal et al., 1999). With this approach, the designers can make decisions to select appropriate rules by evaluating the generated coffeemakers with costing information during the design process. Orsborn et al. (2006) also developed the vehicle grammar, which created different crossover vehicles by defining and combining different vehicle classes. These approaches have potential to integrate with other exploration techniques for extending their generative capability. Other examples include a semantic and shape grammar approach to product design developed by Hsiao and Chen (1997). The shape grammar generates new product forms that satisfy the consumers' physical and psychological requirements.

It is difficult to make a distinctive view of which shape grammar approaches belong to either optimization, exploration, or both. The classification is based on an assumption that the exploration approaches attempt to extend the fixed-shape grammar representation. For example, in our previous research, Lee and Tang (2004) attempted to extend a fixed-shape grammar representation with the research moving in the direction of enhancing the generative capability of shape grammars in product design domain.

With the application of the emergent properties of shape grammars, McCormack and Cagan (2002) developed the hood panel grammar, which generated novel designs with shape-emergent properties. There are many illustrative examples of using the emergent property of shape grammar as described in Stiny's recent book (Stiny, 2006).

2.2.3. Design as coevolution process

From the view of design as a coevolution process, Dorst and Cross (2001) have described an empirical study to analyze and describe the design process as a coevolution of the design problem and the design solution. Taking this view, the research reported in this paper developed a control strategy for monitoring the design process with experimentations in an environment in which designers interact with the system, making decisions about the selection of shape grammars and evaluating whether these shape grammars can generate the desired outcomes.

This research is an extension to our previous work to overcome its two limitations: the representation and the manipulation issues. For the representation issues, the previous work employs parametric 2-D shape grammar with labels to generate 3-D shapes. This limits the modeling capability of the system in generating free-form designs, because the parametric 2-D shape grammar with labels lacks the flexibility for modifying 3-D free-form objects. The generated 2.5-D or 3-D designs are therefore limited in variety. For the manipulation issues, evaluation of the generated designs is limited to artificial selections only. As a result, the complex effects produced by the shape grammar rule modifications during the evolutionary process cannot be fully explored and analyzed.

These varying views contribute to the *ad hoc* nature of various approaches in research of design paradigms in specifying and solving design problems. Because the real problem is not well defined at the beginning of the design process, solutions cannot be well known in advance. The solutions can be progressively found when the real problem is being continuously discovered and refined during the design process. Furthermore, there are no absolute evaluation methods to completely validate the solutions. The evaluation of the solutions is dependent on the designers to determine whether the design problem is sufficiently described (Ozkaya & Akin, 2006). A comprehensive survey that compared the development processes, application areas, and interaction features of different shape grammar approaches is given by Chase (2002).

3. EVOLUTIONARY GRAMMAR-BASED DESIGN FRAMEWORK

A digital camera design case study is introduced to illustrate the approach. Apart from the two critical issues (stylistic consistency and control of design process), the configuration, artificial selection, volume, and parametric constraint issues have been addressed in the approach and illustrated in the case study.

3.1. Parametric shape grammars with labels

A class of compact digital camera forms is defined by two parameters (SH and SP). SH is a specification of a class of shapes and consists of a shape grammar, defining a language of 3-D shapes. SP is a specification of spatial configuration for the shapes defined by SH and consists of a finite list of configuration rules and a limiting shape. Figure 1 illustrates an example of generative specification of the class of compact digital camera forms.

Parametric 3-D shape grammars with labels are developed for the generation of product forms that comprise common engineering shapes. The common engineering shapes are the vocabularies of the shape grammar knowledge base. The common engineering shapes can be classified by their geometric properties such as free-form shapes and primitive shapes including blocks, cylinders, cones, spheres, torus, and their combinations. Nonuniform rational B-spline (NURBS) surfaces are constructed to represent free-form objects in a virtual 3-D spatial environment.

Labels are used to associate the control points of the NURBS surfaces and primitive shapes with the design objects. The labeled control points are used for the identification of NURBS surfaces and primitive shapes. The labels of the design objects are used as functional symbolic notations for the control of generation sequence. Both the labeled control points and the labels of design objects have values to specify their geometric coordinates in X, Y, and Z axes.

In the implementation, both parametric 2-D and 3-D shape grammars with labels are constructed to generate the components and the free-form exterior of the main body, respectively. The reasons for developing these two sets of construction shape grammars are that most of the components are standardized for ease of manufacture in industries. Some standardized form features of the components can therefore be generated by standard mechanical methods such as extrusion and sweeping of 2-D profiles. In contrast, some irregular form features are generated with sophisticated 3-D free form shapes.

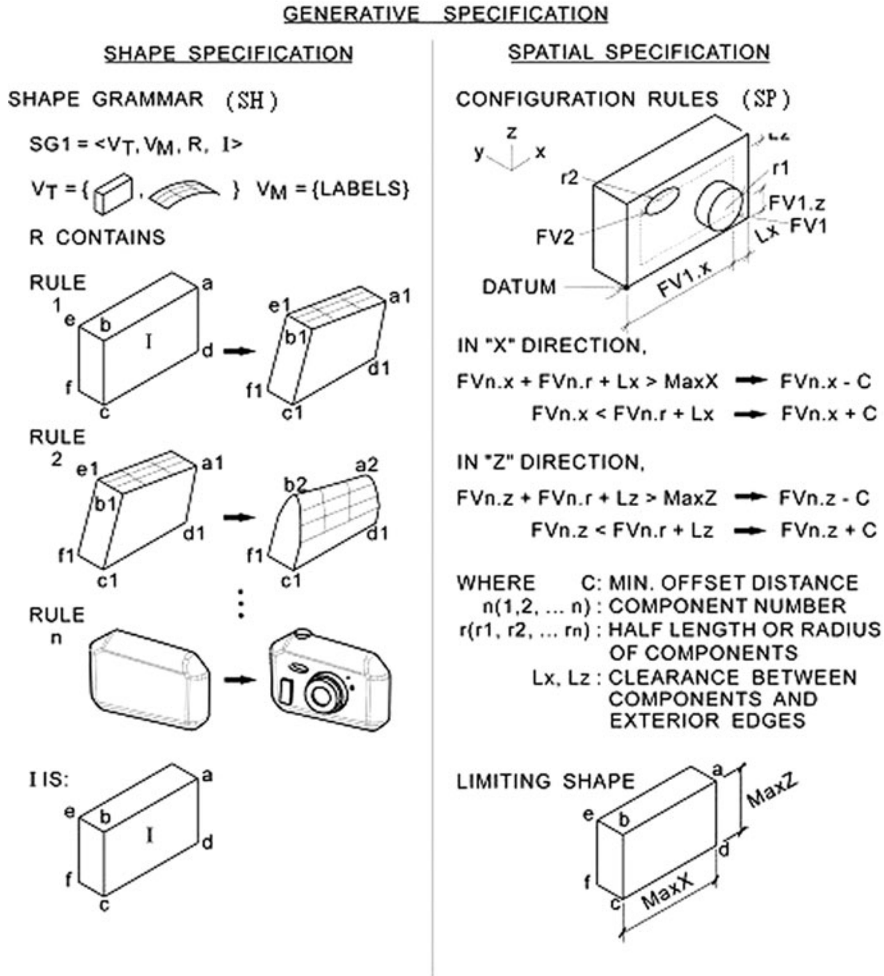

Fig. 1. The generative specification of the class of compact digital camera forms.

The parametric 2-D shape grammars with labels first generate the 2-D profiles of components. The 2-D profiles of components can then be further manipulated with different methods such as coiling, extrusion, lofting, revolving, and sweeping to create 3-D form features of components. In this way, the implementation time can be reduced to generate the form features of the components, whereas longer implementation time is required for the generation of the free-form exterior of the main body.

3.1.1. *Construction rules for free form generation*

An abstracted core variant model representing a class of typical design of compact, durable and all-weather digital cameras is shown in Figure 2. The abstracted core variant model is composed of combined NURBS surfaces and components. Several NURBS surfaces are combined to form the exterior of the main body.

With emphasis on the esthetic quality, the exterior of the main body must be a unique design that attracts users. This can be achieved by modifying the control points of u and v curves of each NURBS surface in the core variant model. A detailed specification of the completed set of parametric shape grammars with labels for the digital camera form design is shown in Appendix A.

3.2. Evolutionary architecture

Most evolutionary algorithms are developed and used as optimization tools in solving engineering problems. Evolutionary algorithms simulate the natural genetic variation and natural selection processes in solving engineering problems. This is achieved by evolving a population of candidate solutions to a given problem using genetic operators such as crossover and mutation, and selection strategies. In this research, genetic programming is selected as the evolutionary algorithm to explore and optimize product form designs.

In building up an evolutionary architecture with genetic programming, there are five preliminary steps to follow: choosing the terminals, the functions, the fitness function, control parameters, and the termination criterion (Koza, 1992). The first two steps can be regarded as representation issues, whereas the last three steps are manipulation issues. Both genetic representation and genetic manipulation are critical in developing the evolutionary shape grammar-based design system, as both

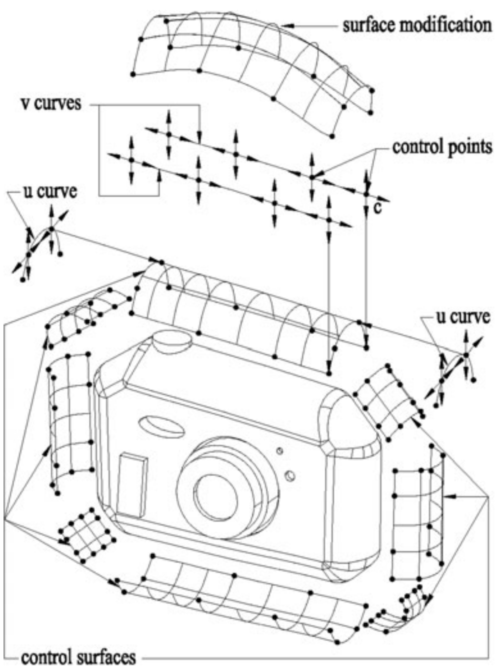

Fig. 2. An abstracted core variant model composed of combined NURBS surfaces and components for the generation of the digital camera form design.

issues will affect the performance of the system in generating product form designs.

3.2.1. "GP-GA-SG" genetic representation

In terms of representation issues, the basic premise in developing the genetic representation for the evolutionary shape grammar-based design system concerns utilizing the power of genetic and shape grammar representations. The genetic representation should facilitate the genetic programming to easily manipulate the shape grammar rules. A three-layer representation interface of phenotypes and genotypes called GP-GA-SG is therefore developed to enhance the performance of the evolutionary shape grammar-based design system (Fig. 3).

The first layer, the "GP interface," is the genotype used by the evolutionary algorithm. The genotype is assigned with a set of modification variables. The modification variables are the genetic programming components organized as tree structures. Each tree represents an evolved program, and can be interpreted as a candidate solution to a given problem. Genotype is coded into one-dimensional (1-D) array data structure in the context of the parameters of genetic programming components.

According to Koza's terminology, the genetic programming components of the evolved programs consist of the terminals and the functions. The functions refer to the junctions in the tree and the terminals the end leaves. For example, Figure 3 shows that a function like "*" takes two arguments. The function branches from the trunk into two branches in the tree. Terminals are the end leaves and can only be used as arguments to a function. Terminals might be assigned a constant such as 4 or an input such as x.

The second layer, the "GA interface," serves two purposes: as a transformation interface interpreting the effects produced by the genetic programming components, and encoding the shape grammar rules in terms of their rule numbers, associated shape parameters, and constraints. The GA interface is coded into a 1-D array data structure in the context of "encoded" shape grammar rule numbers, associated shape parameters, and constraints.

The third layer, the "SG interface," is the phenotype used by both the evolutionary algorithm and shape grammars. The SG interface allows mapping between the "GA elements" and the "SG elements." The phenotype consists of a set of shape grammar rules and parameters that can be used by the shape grammar implementation module to generate the actual design shapes. The SG interface is coded into a 1-D array data structure in the context of the "actual representation" of shape grammar rule numbers, associated shape parameters, and constraints. All of the meanings of the parameters and the relationships among these parameters in the three layers (the GP interface or the genotype, the GA interface, and the SG interface) or the phenotype are interpreted according to the control strategies.

3.2.2. Control strategies

The control strategies are developed in manipulating the new genetic representation and systematically evaluating the evolving designs during the evolutionary process. The control strategies aim at assigning specific sets of terminals and functions to particular types of design problems, and of monitoring the effects of the terminals and functions produced in the generated designs. Based on the choice of genetic programming components of the evolved program (i.e., terminals and functions) and the fitness functions, a search space is then determined for genetic programming to solve particular types of design problems.

The control strategies first define how the control variables in the GP interface should modify the control variables in the GA interface. The control variables in the GA interface, in turn, modify the control variables in the SG interface. Because the control variables in the SG interface are the shape grammar rules and parameters, the modified shape grammar rules and parameters define a new combination of shape features for alternative designs.

An example of the exterior form and component design of a compact digital camera, with the configuration parameters of the components, is shown in Figure 4. A set of parametric 3-D shape grammar rules with labels and parameters are extracted from the shape grammars and put into the SG interface. The second step is to encode the control variables in the SG interface as the "code scripts" of the GA interface by means of the mapping process. The third step is to determine how the control variables in the GP interface should modify the control variables in the GA interface by means of the modification process. Both the mapping and

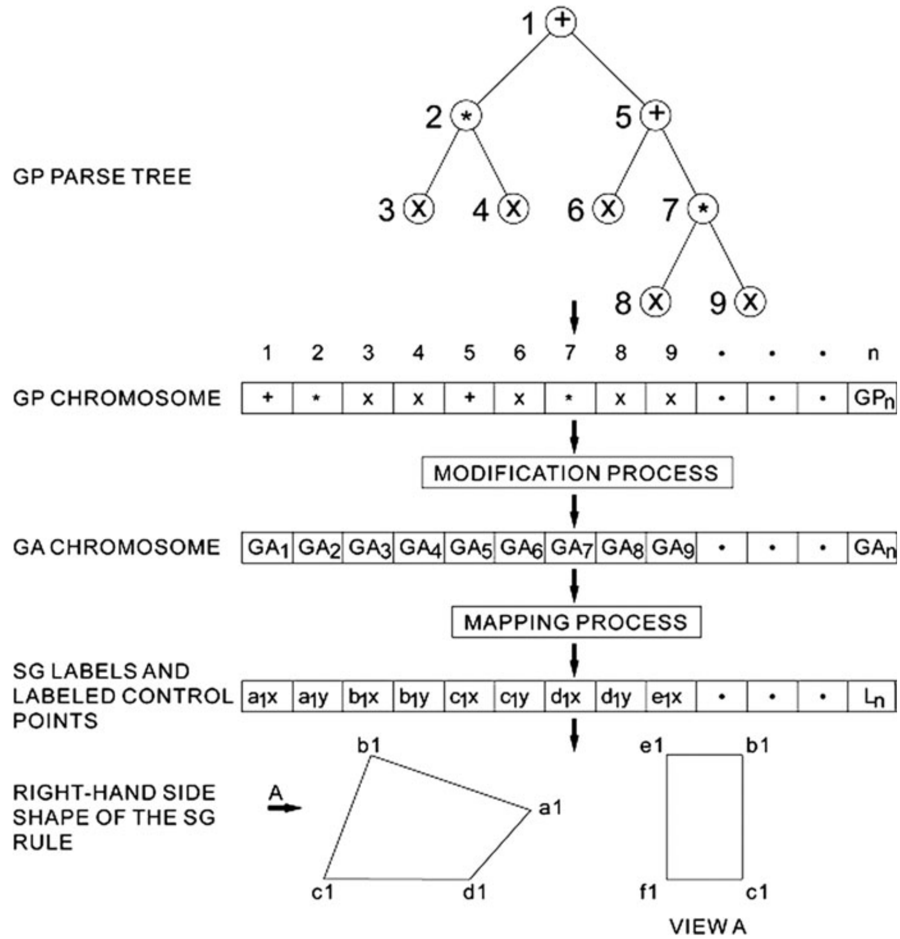

Fig. 3. A genetic representation of the evolutionary algorithm GP–GA–SG interface.

modification processes are regulated by sets of equations that consist of constant-valued parameters and parametric spatial relations among shapes. Table 1 illustrates the details of an example of a particular type of control strategy (the first control strategy) for a compact digital camera form design.

The shape grammar rules are grouped into three categories: exterior of the product form generation, configuration generation, and component generation. Each shape grammar group (SG group) has its corresponding genetic algorithm and genetic programming groups (GA and GP groups) for

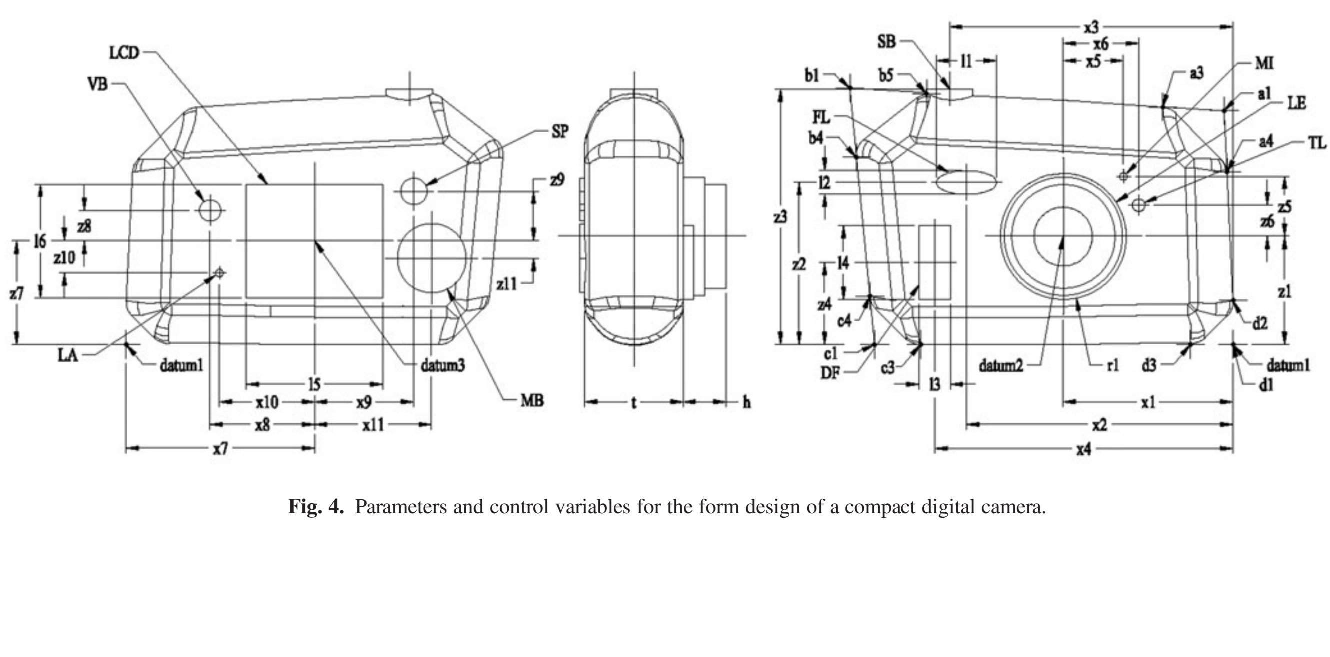

Fig. 4. Parameters and control variables for the form design of a compact digital camera.

Table 1. *Parameters and control variables for the first control strategy*

GP Groups/Elements	Modification to GA Interface	GA Groups/Elements	Mapping to SG Interface	SG Groups/Elements
Group 1		Group 1		Group 1
GP_1–GP_8 Elements: {+, −, *, /, *x*} Notation: {+: +1, −: −1, *: +2, /: −2, *x*: 0}	$GA_n = GP_n + GA_n$ (n = 1–8)	GA_1–GA_8	$a_{p1}x = GA_1 + c_{ap1x}$ $a_{p2}y = GA_2 + c_{ap2y}$ $b_{p3}x = GA_3 + c_{bp3x}$ $b_{p4}y = GA_4 + c_{bp4y}$ $c_{p5}x = GA_5 + c_{cp5x}$ $c_{p6}y = GA_6 + c_{cp6y}$ $d_{p7}x = GA_7 + c_{dp7x}$ $d_{p8}y = GA_8 + c_{dp8y}$	A subset of discrete variables $SG1_p$ ($SG1_p \in N_1$), $N_1 = \{a_1x, \ldots, a_{n1}x, a_1y, \ldots, a_{n2}y, b_1x, \ldots, b_{n3}x, b_1y, \ldots, b_{n4}y, c_1x, \ldots, c_{n5}x, c_1y, \ldots, c_{n6}y, d_1x, \ldots, d_{n7}x, d_1y, \ldots, d_{n8}y\}$, $0 < p < n, 0 < p_1 < n_1, \ldots, 0 < p_8 < n_8$, where N_1 is the total set of the variables for SG1, n_1–n_8 are the total number of variables for each group of control parameters, n is the total number of variables for SG1, which is equal to the sum of n_1–n_8.
Group 2		Group 2		Group 2
GP_9–GP_{12} Elements and notations: same as group 1	$GA_n = GP_n$ (n = 9–12)	GA_9–GA_{12}	$x_1 = GA_9 + c_{x1}$ $z_1 = GA_{10} + c_{z1}$ $x_2 = GA_{11} + c_{x2}$ $z_2 = GA_{12} + c_{z2}$	A subset of discrete variables $SG2_p$ ($SG2_p \in N_2$), $N_2 = \{x_1, \ldots, x_{11}, z_1, \ldots, z_{11}, l_1, \ldots, l_6, r_1, t, h\}$, $0 < p < n$, where N_2 is the total set of the variables for SG2 and n is the total number of variables for SG2.
Group 3		Group 3		Group 3
GP_{13} Elements: {+, −, *, /, *x*} Notation: {+: 1, −: 2, *: 3, /: 4, *x*: 5}	$GA_{13} = GP_{13}$	GA_{13}	$FV_7 = DF_{GA13}$	A subset of discrete variables $SG3_p$ ($SG3_p \in N_3$), $N_3 = \{FV_1, \ldots, FV_7, BV_1, \ldots, BV_8, SV_1\}$, $0 < p < n$, where N_3 is the total set of the variables for SG3 and n is the total number of variables for SG3.

monitoring the effects of the terminals and functions produced in the generated designs. Each (SG group) consists of modifiable elements such as construction parts, configuration, shape grammar rules, structures, and spatial relations. The shape grammar rules in the (SG groups) are specified with their own parameters, for example, the *XYZ* control variables of the labeled point a_1 shown in Figure 4. Table 2 illustrates the details of the properties in each (SG group). An example of the first control strategy is illustrated in Appendix B for the symmetric type of product form designs.

3.2.3. *Genetic programming*

In the evolutionary shape grammar-based design system, the genetic programming performs three main functions: modifying alleles within chromosomes using genetic operators, decoding the genotype to produce the phenotype in accordance to the control strategies, and evaluating the phenotype to identify the fittest solutions (Fig. 5).

At the beginning of running the system, the genetic programming generates an initial population of 500 individuals with random values. Because of the complexity of displaying the virtual models in the limited display area of computer screen, a maximum of 12 individuals are extracted from the population for visualization. However, the designers can choose to keep the displayed selected designs during evolution to trace the modification effects on the selected designs, or to replace the selected designs with the fittest ones during evolution while searching the best designs.

The main loop begins at this stage. Each individual is then evaluated and assigned a fitness value by fitness functions and artificial selection. Based on the scores obtained from each solution, the solutions with higher scores will be selectively copied to a temporary area termed “mating pool.”

Entering to the second loop, two of the solutions are randomly selected as parents from this mating pool. These two parents generate two offspring by random crossover and

Table 2. *Properties in each S group*

SG Groups	Construction Parts/Configuration	Rules	Structures Spatial Relations
Group 1	Exterior	C1–C8	Free from 3-D solids
Group 2	Configuration	F1–F22	Spatial relations
Group 3	Mode dial, shutter button, flash, microphone, self-timer lamp, lens, view button, power switch, zoom button, speaker, strap eyelet, battery compartment, menu button, monitor, lamp, and decorative feature	C9–C24	2.5-D and 3-D solids

mutation operators and replace the parents of the population. The crossover and mutation processes repeat to generate offspring until every parent of the old population is replaced; a new population with fitter solutions is then established.

For each generation, the genotype is converted into the phenotype that represents the solutions. The solutions are a number of individuals, each of which consists of a set of parametric 3-D shape grammar rules with labels and parameters. After execution of the shape grammar rules by the shape grammar implementation module in accordance to the generated rule sequences, both the exterior main bodies and components are generated. The genetic programming repeats the main loop of evaluation and reproduction processes for a specified number of generations, or the genetic programming will stop if satisfactory solutions emerge.

3.2.4. *Multiobjective functions*

Exterior-form generation of compact digital cameras and the configuration of the components are designed to fulfill a set of requirements such as artificial selection, spatial geometric constraints, and desired exterior shell volume. The design requirements can be formulated into objective functions. Objective functions are set up for the evaluation of the generated designs. General objective functions are set up for general requirements, whereas control strategies have their own sets of objective functions for specific requirements. Analysis of the evaluation results will help in the investigation of and understanding of combinatorial effects on the generated designs based on the control strategies. To effectively evaluate the design performance, a metric is formulated as the summation of design objectives and weighted constraint violations.

$$\text{index function} = \text{objective index} + \text{constraint index} = \sum_{i=1}^{l} \text{objective index}_i + \sum_{j=1}^{m} \text{constraint index}_j, \quad (1)$$

where l is the number of objectives and m is the number of constraints. Objective and penalty functions are defined to assign positive and negative fitness scores, respectively. Penalty functions are activated if the generated designs violate the

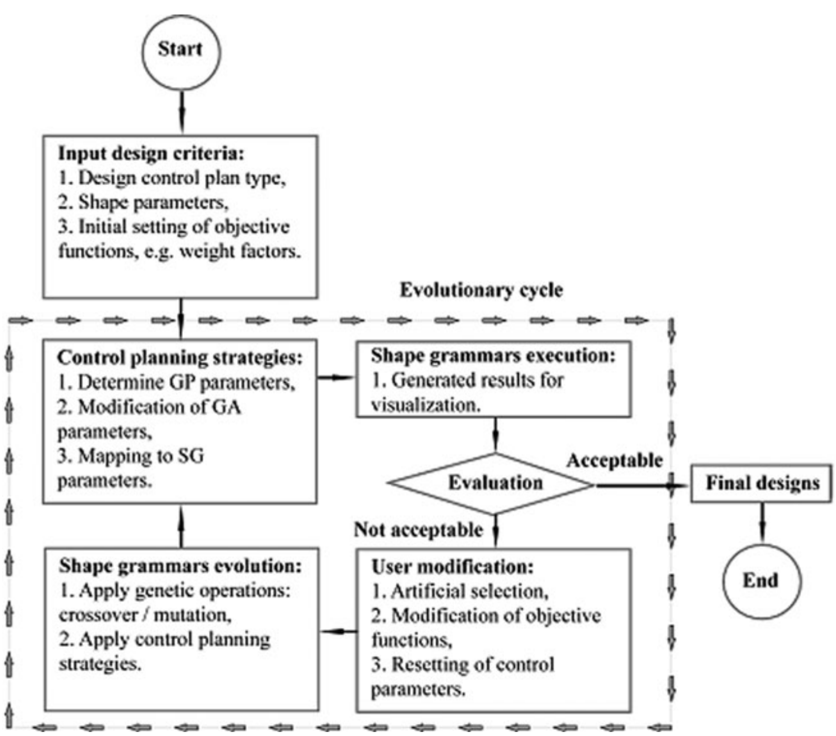

Fig. 5. The evolutionary grammar-based design framework.

constraints. Both design objectives and constraints have weighting factors to determine the relative trade-off among design objectives. The designers can assign different weighting factors on each variable.

$$\text{objective index} = \sum_{i=1}^{l} (\text{objective weight}_i \times \text{objective value}_i), \quad (2)$$

$$\text{constraint index} = \sum_{j=1}^{m} (\text{constraint weight}_j \times \text{constraint violation}_j), \quad (3)$$

where l is the number of objectives and m is the number of constraints. For the artificial selection requirements, objective index_1 is used as the measurement of accumulated effect on selected designs. The selected designs will be assigned with higher fitness scores if they are frequently selected by the designers.

$$\text{objective index}_1 = \sum_{i=1}^{n} (\text{selection weight}_i \times \text{selection value}_i) \quad \{\text{selection value}_i = 0 \text{ or } 1\}, \quad (4)$$

where n is the the number of generations; objective index_1 is the accumulated score for each design; selection weight_i is the weighting factor for each design; selection value_i is assigned with 1 when the designs are selected, otherwise 0. Because the selection cost of each design is the accumulated score from each generation, selection on one or more designs in a particular generation will not significantly impact the whole population. As a result, the population is determined by the accumulated effect on the selected designs.

Under the spatial geometric constraints, the components have to be configured without collision among each other and within the boundary of the exterior of camera body. Geometric variables of the component positions and the boundary positions of the exterior of the camera body are assumed to be configuration design variables, subject to a set of constraints. The objective functions of configuration of components can be determined by the designers with selective options. For example, the selective options of configuration are: to maximize or minimize the total distance (TD_1) among components.

For maximize option selected,

$$\text{objective index}_2 = \text{configuration weight} \times \text{TD}_1. \quad (5)$$

For minimize option selected,

$$\text{objective index}_2 = \text{configuration weight} \times \frac{1}{\text{TD}_1 + C}, \quad (6)$$

$$\text{TD}_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}, \quad \{i \neq j\}. \quad (7)$$

Subject to (a set of constraints)

$$d_{ij} \geq l_i + l_j + l_c, \quad \{i = 1 \text{ or } 2 \text{ or }, \ldots, \text{ or } n\}, \quad \{j = 1 \text{ or } 2 \text{ or }, \ldots, \text{ or } n\}, \text{ and } \{i \neq j\},$$

$$\text{constraint index}_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} (\text{configuration constraint weight} \times \text{constraint violation}_{ij}), \quad (8)$$

$$\{\text{constraint violation}_{ij} = -(l_i + l_j + l_c - d_{ij}), \text{ if the constraints are violated}\},$$

$$\{\text{constraint violation}_{ij} = 0, \text{ if the constraints are not violated}\}.$$

where C is a constant; n is the number of components; l_i, l_j are the half length or radius of components; l_c is the clearance between components; and coefficient d_{ij} is the distance between components i and j. The distance between two components is defined as the distance between the centers of both components as shown in Figure 1. The summation of all the distances between any two components (TD_1) reflects the dispersion among components.

For exterior shell volume calculation, the objective is to minimize the difference between the shell volume and a desired target shell volume of the exterior of camera body.

$$\text{objective index}_3 = (\text{volume weight} \times f(v)), \quad (9)$$

$$\text{minimize } f(v) = \frac{1}{|(v - v_{\text{target}})| + C}, \quad (10)$$

where C is a constant. The value of an exterior shell (v) refers to the approximate volume estimation of the exterior of the camera body. A constant C is added to objective index_2 and $f(v)$ to ensure that the objective indices take only positive values in their domains (Michalewicz, 1996). The addition of constant C to the objective indices also avoids the error arising from dividing zero.

Because multiobjective functions exist in the evolutionary grammar-based design system, a wide number of designs belonging to the paretooptimal front (POF) can be identified. To simplify the implementation, the use of a weighting approach is sufficient to explore different settings of parameters. Further study of an advanced POF technique will lead to performance improvement.

4. IMPLEMENTATION

A software prototype evolutionary grammar-based design system has been developed using Visual C++ and ACIS 3-D modeling kernel, and tested. At the beginning of running the evolutionary grammar-based design system, the designers first input a set of design criteria such as selecting design control plan types, specifying types of components and their corresponding shape parameters, and initial setting of objective functions, for example, weighting factors. Entering the evolutionary cycle, at the first generation, the system applies the

Fig. 6. The initial random generation of designs. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

construction and configuration rules to randomly generate a population of designs (Fig. 6).

The initial form style of each compact digital camera is defined by rule C1 with a pocket-size metal body of rectangular shape. It is deformed to a quadrilateral shape that is the main skeleton of the exterior camera body by rule C1 (Fig. A1). Rules C2 to C8 then modify the main skeleton of the exterior of camera body to a curved profile (Fig. A1).

Rules C9 to C24 generate the components: rotating mode dial (by rule C9), shutter button (by rule C10), flash (by rule C11), microphone (by rule C12), self-timer lamp (by rule C13), optical zoom lens (by rule C14), quick view button (by rule C15), power switch (by rule C16), zoom button (by rule C17), speaker (by rule C18), strap eyelet (by rule C19), battery compartment (by rule C20), menu button (by rule C21), monitor (by rule C22), lamp (by rule C23), and decorative feature (by rule C24), respectively (Fig. A2). The shape grammar implementation module generates the actual design shapes based on the shape grammar parameters.

After all the components have been generated in order, they are positioned in the camera body in accordance to the configuration rules F1 to F22 (Figs. A3 and A4). In the implementation, all the significant components are generated to demonstrate the potential usage of the evolutionary grammar-based design system, as shown in Figures 6 and 7, while leaving some insignificant components to be implemented in the future. The actual design shapes are evaluated by the evaluation module. If the results are not satisfactory, the designers can modify the objective functions, reset the control parameters, or intuitively select the generated designs. Genetic operations such as crossover and mutations will then be applied to evolve the shape grammar rules.

To clearly demonstrate the operations of the system, the first control strategy for designing regular or symmetric type designs is first applied and illustrated with examples. The details of the first control strategy are described in

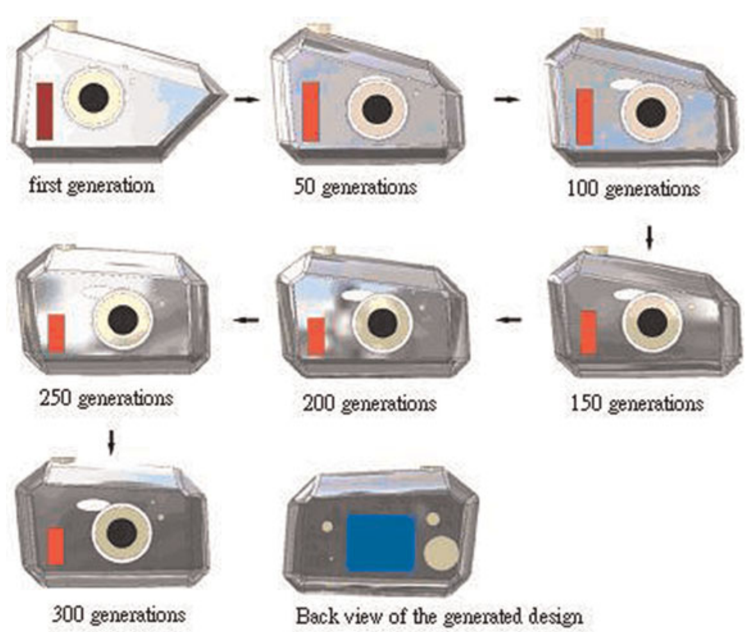

Fig. 7. Regular type designs: results obtained from the first generation, 50 generations, 100 generations, 150 generations, 200 generations, 250 generations, and 300 generations and the back view of the generated design. [A color version of this figure can be viewed online at journals.cambridge. org/aie]

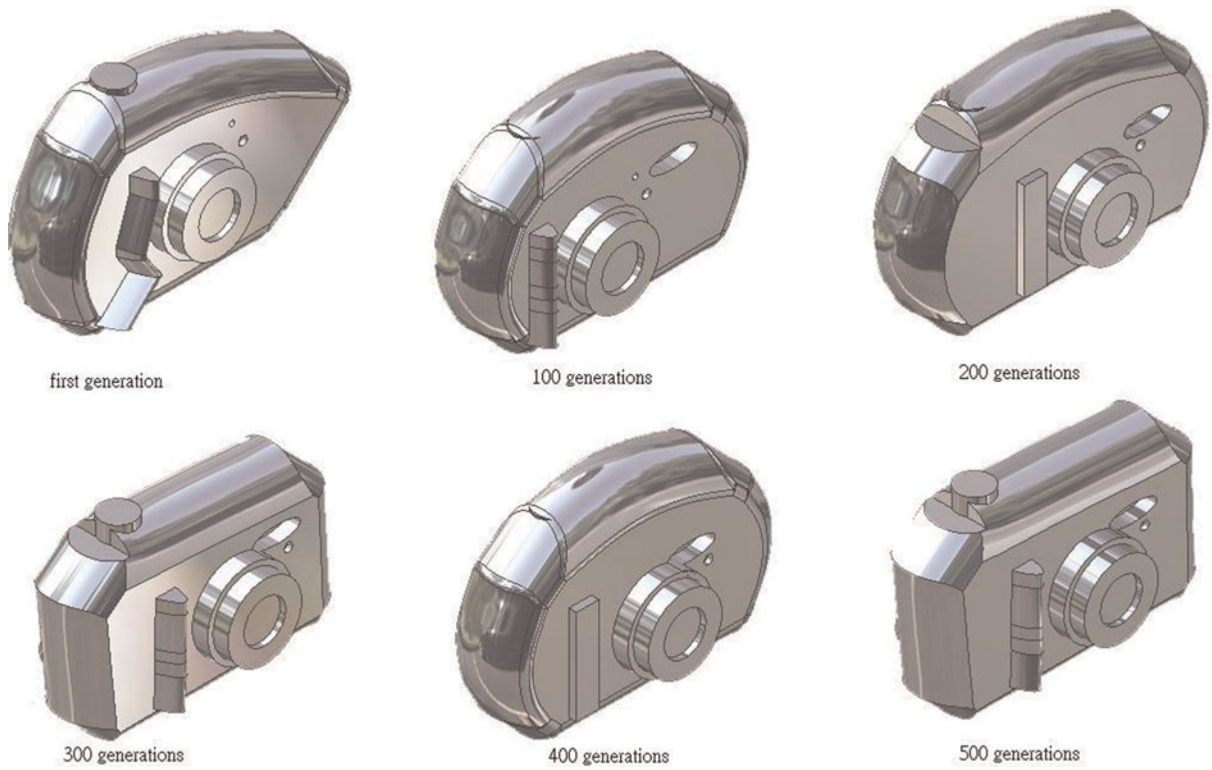

Fig. 8. Results obtained from the first generation, 100 generations, 200 generations, 300 generations, 400 generations, and 500 generations. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

Appendix B. This experiment emphasizes and illustrates the interactions between the designer and the system. Continuing with the operations after the initial running of the system, the designer can select the favorable design intuitively from the 12 designs. The designer can also choose to keep the all displayed designs during evolution for tracing the modification effects on the designs. The modification effects on the selected design are shown in Figure 7.

By adjusting the parameters of the objective functions and selecting the appropriate control strategies, the designer can flexibly study the effects on the generated designs and then determine which strategy is most suitable for a particular application. Other control strategies such as slim, asymmetric, and mixed can also be defined to test the flexibility and effectiveness of the evolutionary grammar-based design approach in product form design generation. Finally, another evolutionary cycle starts and repeats until satisfactory results emerge or maximum generations are reached. Another experiment has been conducted to test the system using other control strategies for new requirements illustrated in the next section.

5. EVALUATION

The setting of the evolutionary grammar-based design system is initialized by the designer prior to the system runs. By setting the population size to 500, the crossover rate to 0.6, and the mutation rate to 0.01, the system generates the designs in accordance to different requirements. Each individual design is assigned with a design number from 1 to 500. Implementation examples are carefully planned to demonstrate how the designers can interact with the system and what the results would be in respect to the requirements. By setting the control parameters of the evolutionary grammar-based design system in each periodically observed generation in a tabular format and by evaluating the corresponding results visually and numerically, a clear picture of the complex effects produced by the objective functions is depicted. Based on the analysis of the results, the designer can select appropriate control parameters and control strategies to explore designs during the evolutionary design process.

Figure 8 shows the implementation results obtained from the system, starting at the first generation and ending at 500 generations. The generated models can be postprocessed by other commercial software for rendering with surface contour patterns. The surface contour patterns allow the designers to evaluate the surface quality of the generated models more effectively. The designers can visually inspect the continuity between surfaces of the generated models.

Together with the aid of a comprehensive table listing all the relevant information of the evolving forms of a product, results can be analyzed numerically. Table 3 depicts the detailed specifications and control parameters of the system.

Table 3. *Implementation results of the evolved parameters of shape grammar*

Detailed Specifications	GP Groups	GA Groups	SG Groups
Generation 1			
1. Design number = 494 2. Control strategy = 3 3. Main objective index (overall fitness) = 0.00 4. Objective index (1 artificial selection fitness) = 0.00, selection weighting factor = 1.00, selection index value = 0.00 5. Objective index 2 (configuration fitness) = 0.00, configuration weighting factor = 0.10, configuration index value = 0.00 6. Objective index 3 (volume fitness) = 0.00, volume weighting factor = 1.00, volume index value = 0.00 7. Constraint index 1 (configuration constraint) = 0.0000, configuration constraint weighting factor = 1.00, configuration constraint index value = 0.0000 8. Shell volume = 28.4564, target volume = 24.5280	9. GP group 1 (GP_1–GP_8): +, *, x, +, x, +, x, + 10. GP group 2 (GP_9–GP_{12}): x, +, −, x 11. GP group 3 (GP_{13}): x	12. GA group 1 (GA_1–GA_8): 34, 44, −68, 63, −75, −1, 8, −1 13. GA group 2 (GA_9–GA_{12}): 39, 31, 49, 31 14. GA group 3 (GA_{13}): 5	15. SG group 1 (a_1x, a_1y, a_3x, a_3y, a_4x, a_4y): 35, 46, 24, 34, 25, 33 16. SG group 1 (b_1x, b_1y, b_4x, b_4y, b_5x, b_5y): −68, 64, −64, 51, −57, 56 17. SG group 1 (c_1x, c_1y, c_3x, c_3y, c_4x, c_4y): −75, 0, −65, 7, −66, 8 18. SG group 1 (d_1x, d_1y, d_2x, d_2y, d_3x, d_3y): 8, 0, −1, 9, −2, 8 19. SG group 2 (lens and flash: x_1, z_1, x_2, z_2): 39, 31, 49, 31 20. SG group 3 (grip): DF5
Generation 100			
1. Design number = 456 2. Control strategy = 3 3. Main objective index (overall fitness) = 3.43 4. Objective index 1 (artificial selection fitness) = 0.00, selection weighting factor = 10.00, selection index value = 0.00 5. Objective index 2 (configuration fitness) = 3.24, configuration weighting factor = 0.10, configuration index value = 32.38 6. Objective index 3 (volume fitness) = 0.195, volume weighting factor = 1.00, volume index value = 0.195 7. Constraint index 1 (configuration constraint) = 0.0000, configuration constraint weighting factor = 10.00, configuration constraint index value = 0.0000 8. Shell volume = 20.3923, target volume = 24.5280	9. GP group 1 (GP_1–GP_8): −, −, x, x, −, −, x, x 10. GP group 2 (GP_9–GP_{12}): x 11. GP group 3 (GP_{13}): nil	12. GA group 1 (GA_1–GA_8): 13, 58, −75, 58, −74, 1, 12, 0 13. GA group 2 (GA_9–GA_{12}): 28, 26, 57, 40 14. GA group 3 (GA_{13}): 3	15. SG group 1 (a_1x, a_1y, a_3x, a_3y, a_4x, a_4y): 12, 57, 1, 45, 2, 44 16. SG group 1 (b_1x, b_1y, b_4x, b_4y, b_5x, b_5y): −75, 58, −70, 45, −64, 50 17. SG group 1 (c_1x, c_1y, c_3x, c_3y, c_4x, c_4y): −75, 0, −65, 7, −66, 8 18. SG group 1 (d_1x, d_1y, d_2x, d_2y, d_3x, d_3y): 12, 0, 3, 9, 2, 8 19. SG group 2 (lens and flash: x_1, z_1, x_2, z_2): 28, 26, 57, 40 20. SG group 3 (grip): DF3

The detailed specifications include: design number, control strategy, main objective index, artificial selection fitness, configuration fitness, volume fitness, configuration constraint index, and volume estimation, whereas the control parameters include the parameters of GP groups (items 9 to 11), GA groups (items 12 to 14), and SG groups (items 15 to 20).

To analyze the generated results, a complete historical record showing how the designers interact with the system is depicted. The second and third control strategies are selected to illustrate the regulation of the generated designs with symmetric properties. Each control strategy has different form features as shown in Figure 8.

- At the first generation, the genetic programming generates an initial population of 500 individuals with random values. All the weighting factors are preset to 1.0 except for the configuration weighting factor, which is preset to 0.1, and the target shell volume, which is preset to 24.528 cm^3. The designers start to modify the control parameters. First, the result of design number: 494 is intuitively selected by the designers. Second, the designers select the third control strategy and modify the selection weighting factor and the configuration constraint weighting factor with a value of 10. Then, the designers evaluate the results at each periodically observed generation (every 100 generations).
- At 100 generations, all the designs are generated based on the third control strategy. Most of the generated designs have similar form features to the previously selected design. This is caused by setting the selection weighting factor with a value of 10. The artificial selection criterion influences the configuration of components and the selection of decorative features. In addition, the adverse effects of constraint violation between components of the generated designs are improved. This is caused by setting the configuration constraint

weighting factor with a value of 10. As an example, the generated design (number: 456) indicates that the configuration constraint is of zero value. It can be visually determined by the designers that the lens and the flash are dispersed far apart from each other on the exterior of the main body.

- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number 456 is intuitively selected by the designers. The designers apply the same settings of control parameters for the next 100 generations.
- At 200 generations, all the designs are generated based on the third control strategy. The previously selected design would probably not appear in the first few periodically observed generations. This is because the selection fitness value is based on the accumulated scores of selected designs. Because the accumulated scores of selected designs in the first few observed generations are not significant, the chances of the selected design appearing in the subsequent observed generations becomes small. Therefore, most of the artificial selection fitness values of the selected designs are zero in the first few observed generations.
- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number 299 is intuitively selected by the designers. The designers would like to explore other types of product form designs by selecting the second control strategy and setting the target volume to be 27 cm^3.
- At 300 generations, all the designs are generated based on the second control strategy. Unexpected outcomes astonish the designers: most of the generated designs get poor volume fitness values. This is caused by the low volume weighting factor with a value of 1.
- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number 124 is intuitively selected by the designers. The designers would like to explore other types of product form designs with a larger distance between the lens and the flash. Therefore, the designers select the third control strategy and set the configuration weighting factor as 1.
- At 400 generations, all the designs are generated based on the third control strategy. Still, most of the designs get poor volume fitness values. However, the designers favor most of the designs in this generation.
- After evaluation of the designs, the designers then continue to modify the control parameters. The result of design number 51 is particularly attractive to the designers. Although the flash is so close to the lens, the designers select this design by their own accord. The designers look for better designs by setting the second control strategy and assigning the volume weighting factor to be 10 for the next 100 generations.
- At 500 generations, the result of design number 418 satisfies the designers. Even though some of the requirements are still not satisfied, the designers could continue to explore better designs by better understanding the complex effects provided by the modification of control parameters.

The assumption that better designs could be obtained is made provided that the requirements have to be refined considerably. If there are conflicting requirements, the designers should report those conflicting criteria to the relevant professionals and ask the professionals to consider modifying the requirements if necessary.

The control parameters of the "GP groups" can be adjusted within appropriate ranges. The larger the range of control parameters as defined, the higher the modification rates to the designs will be. As a result, more dramatic modifications to the designs appear to result from the erratic forms generation. Balance on the rate of modification and the quality of the generated models should be achieved by adjusting appropriate ranges for the control parameters.

Figure 9 shows the variation of objective fitness values with different weighting factors and control strategies. The data are taken from the selected designs in each periodically observed generation. The purpose of this diagram is to record the whole design process historically starting from the first generation to the last of the design process. Because the nature of a real design process is iterative and changes from each periodically observed generation, the diagram can be further analyzed to enhance human computer interaction rather than used for purely computational analysis.

6. DISCUSSION

The discussion section consists of two parts: design research issues and elaboration of the approach.

6.1. Design research issues

This framework supports the design activities in which the two critical issues, stylistic consistency and control of design process, are addressed.

6.1.1. Stylistic consistency

The stylistic consistency of a product becomes more important in industries as the style of a product represents the brand image. Product designers have to keep the key distinguishing characteristics of a product, which are the disposition of components, the type of components used, and the boundary constraints of the component forms. This framework provides a control strategy to manage the complexity of stylistic consistency. The stylistic consistency issue is only part of the design characteristics to be addressed in product design.

Apart from stylistic consistency, further investigation on background support for the development of shape grammar is required. For example, the cultural elements should be studied: "In terms of product design, we should emphasize the

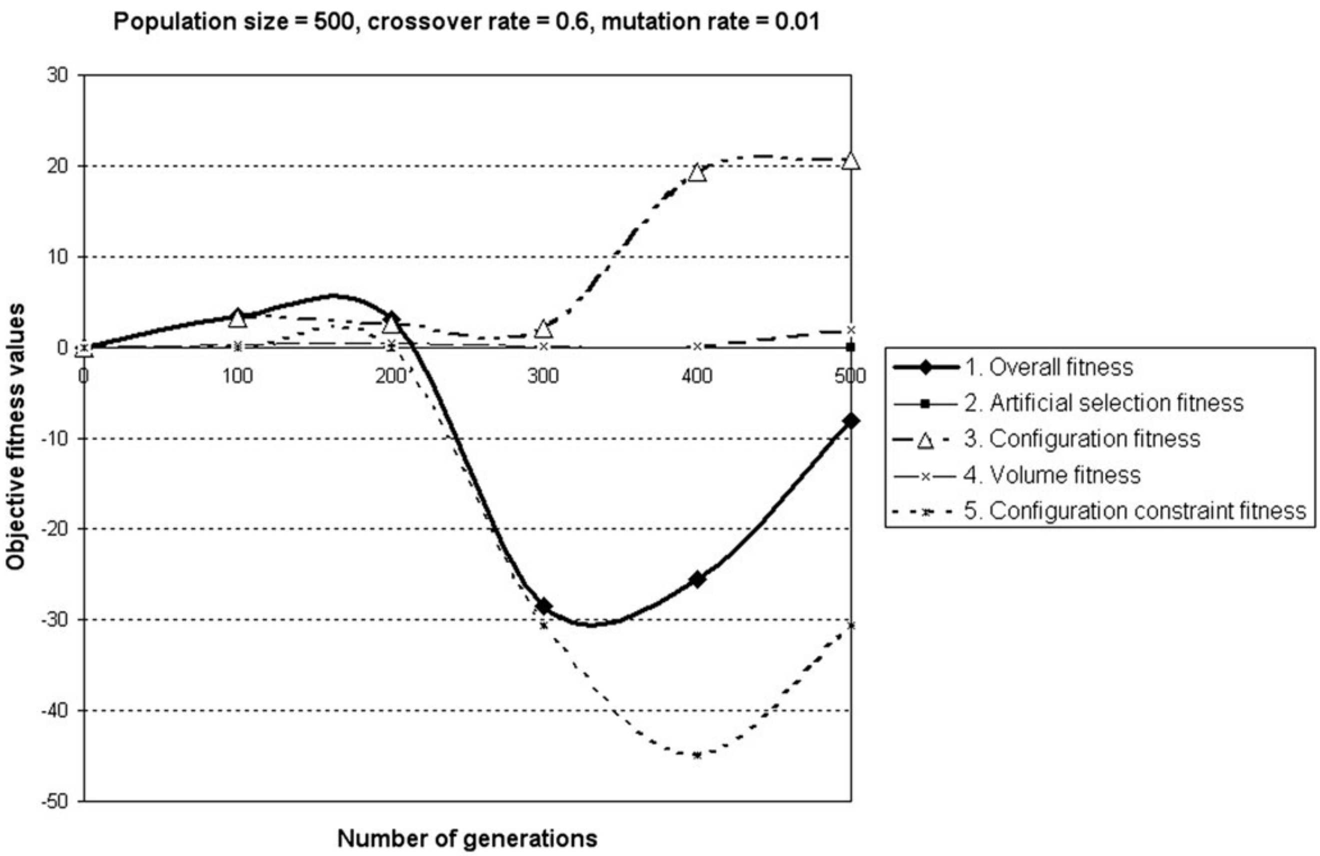

Fig. 9. The variation of the objective fitness values with different weighting factors and control strategies.

uniqueness of eastern culture, such as implicitness, calmness, lightness, as well as the pursuit for natural and humanistic harmonies, in order to stimulate an emotional echo from the users" (Leung, 2006). This framework requires further elaboration of key aspects such as the following:

- embedding the cultural elements in the shape grammar in an elegant manner,
- understanding customer behaviors in choosing the products with their preferences, and
- exploring market trends in the development of new products for specific regions like China and the United States or for global marketing.

6.1.2. Control of design process

The control of the design process can be further exploited from perspectives on two issues: technical issues of control strategy and practical issues of product design.

Technical issues of control strategy. Integrating parametric shape grammars with labels provides a good control strategy in monitoring the design process. However, this approach is limited under three critical constraints: lack of a control mechanism in monitoring the rate of exploration, missing a blueprint of exploration in matching design requirements, and the shape grammar rules are predetermined without dividing the elements of shape grammars into different sets in terms of abstraction level of representation and purposes.

The new genetic representation (GP-GA-SG representation) and control strategies developed in this paper address the relevant issues. The "GP-GA-SG representation" is composed of three layers. The first layer (the GP layer) is a control layer in which the genetic programming operates as a control mechanism to monitor the rate of exploration. This new approach provides a significant change in the way to apply genetic programming to harmoniously integrating the control of design process and generation of designs together. In traditional approaches, the elements of genetic programming represent the ways to generate designs or the elements of designs. The advantage of this representation is that the length of the genetic elements can be dynamically modified during the evolutionary design process. The new approach holds this advantage but shifts the focus to monitoring the design process. During the evolutionary design process, the GP layer with a shorter length of the genetic elements has less effect on the designs compared with the longer one. The GP layer cooperates with the control strategy, which allows the designers to have greater flexibility in monitoring the design process.

The second layer (the GA layer) represents the blueprint of exploration. This new approach provides a plan to solve the conflicts in evolving a shape grammar. One of the key difficulties in integrating a highly detailed shape grammar and an evolutionary computing system is that conflicts exist in the random modification properties of evolutionary computing and the replication and capture of style properties of shape grammar. Random modification of product form designs removes the classical style of the product. The new approach introduces a blueprint that determines the ways to modify the elements of shape grammar rules. Referring to Table 1, any elements of a shape grammar such as shapes and labels can be modified. The ways of modification can be as simple as proportional change or as complex as any sophisticate algorithms. The blueprint itself cannot completely solve the conflicts. The blueprint should cooperate with the control strategy for monitoring the change of shape grammar rules. For example, a control strategy is derived to regulate the shape grammar rules to generate a symmetric exterior body. Assuming that the symmetric property of a design is the only stylistic requirement, the blueprint, which specifies radical modification of shapes, can be selected in this case. Control strategies are derived for particular components with desired design characteristics, whereas multiobjective functions for the evaluation of the overall design.

The third layer (the SG layer) is an implementation layer in which the genetic elements are the same as the shape grammar elements. This new approach divides the shape grammars by parts for matching specific design requirements. In a traditional approach, the shape grammars are derived as a whole without considering dividing the elements of shape grammars into different sets in terms of abstraction level of representation and purposes. The SG layer divides the shape grammars into three different sets. Referring to Table 2, the first set of shape grammar rules (SG group 1) represents the lowest abstraction level of representation and for the purpose of generating free-form 3-D objects, the second set (SG group 2) represents the middle level of representation and for the purpose of configuring the components, and the third set (SG group 3) represents the highest level of representation and for the purpose of selecting and creating different types of components.

In SG group 1, the first set of rules, C1 to C8, is arranged sequentially for generating the exterior of camera body. The arrangement of the shape grammar rules appears as if the right side of a rule is the left side of the subsequent rule. Depending on the control strategy selected, the elements of either one of the rules can be set into this group. For example, in Table 3, the rule C1 with control points (a_1, b_1, c_1, d_1) could be selected and put into the SG group 1. When the values of the control points are modified, the remaining rules (C2 to C8) are executed in accordance to the modified values. This lowest abstraction level of representation is particular useful in manipulating the detailed refinement of designs.

In SG group 2, the second set of rules, F1 to F22, are grouped for the allocation of components in the design space. The configuration is determined by the geometric coordinates of the central axes of the components. The components will be allocated within their corresponding boundary constraints in design space. The components themselves are either generated by another set of rules or predetermined in advance. This middle level abstraction in the representation is particular useful for configuring the spatial relations among the objects.

In SG group 3, the third set of rules, C9 to C24, are grouped based on the functional characteristics of components. Each type of component can be generated with a corresponding set of shape grammar rules. New shapes can be introduced to the rules to extend the fixed-shape grammar representation (Lee & Tang, 2004). This highest abstraction level of representation is particular useful in selecting and creating different types of components.

Practical issues of product design. This paper takes the view that the exploration of designs is not only treated as a problem-solving activity but also a problem-finding activity. The computational framework is developed based on the vision that these two activities go hand in hand to accomplish the design tasks in an interactive design environment during the design process. The results obtained from the experiments reflect real situations that happened in industries that the designers may require to adjust the design requirements during the design process. This may because of the causes of uncertainty in the specification of design problems at the beginning of the design process. The designers need to discuss the intermediate results with the professionals of various departments of an enterprise or clients for compromising the conflicting requirements until satisfactory results are obtained.

In this framework, the design process is monitored through various activities starting from the preparation works to the implementation of shape grammars. In general, the designers can follow a common scenario to use the system to explore designs. For instance, the shape grammar developer prepares a starting grammar, the system evolves the shape grammars that produce designs, the designers evaluate the designs, the system takes account of the evaluations, until the designers are satisfied.

The activities of preparing a shape grammar include converting the empirical skills from experts into shape grammar rules, analyzing the features of existing designs, identifying the interrelationships among various features, suggesting multiply ways to describe the features, converting those features into modifiable elements of rules, organizing the rules into different sets and determining several ways to apply the rules for a topologically diverse set of solutions. Each activity not only costs a tremendous amount of work and time but also is difficult to accomplish in the case of starting from scratch. For example, either theories or empirical skills from experts are difficult to convert into shape grammar rules. It is a time-consuming process to derive theoretical design concepts by means of research. The practical skills of experts are qualitative in nature, and therefore hard to quantify for computation. However, it is worth it for an enterprise to put efforts in formulating a shape grammar approach for monitoring the design process. The amount of work will greatly be reduced once a knowledge base of shape grammar is set up.

The second part of monitoring the design process is to evolve the shape grammar rules by the system. The system adopts an integration approach of two key computational techniques: shape grammars and evolutionary computing for supporting product design activities. The system does all the complicated procedures to construct the complex 3-D models for visualization. The models can be evaluated effectively with numerical analysis by the designers. This reduces the designers' time to do the complicated modeling tasks to construct a number of models from scratch. The designers can concentrate their efforts on performing a higher level of design tasks such as evaluation of designs and making decisions.

Based on this scenario, it can be seen that the design activities in each stage of the design process influence the final results. The understanding of the effects on decision making in each design activity is therefore crucial to the success or failure of the product design. Another significant advantage can be seen from the scenario that designers continuously evaluate the designs and select and modify the objective functions during the evolutionary process. Because the generated designs are continuously monitored interactively by the designers during the evolutionary process, the designs can be improved and refined after many generations. As a result, the design problems are refined from an unclear specification of requirements, through the continuous evaluation and improvement of the solutions, to a clearer specification of requirements. This provides a higher chance for the results generated by the system to be more suitable to fit the expectations of the designers than at starting stage.

6.2. Elaboration of the approach

More complex systems can be achieved by elaborating the four design research issues in scaling up this approach: planning, evolution, control, and evaluation.

6.2.1. Planning

Planning is the starting design activity in which the understanding of cultural issues for the targeted users is a major challenge. Shape grammar developers cooperate with designers to determine shape transformation sequences and shape grammar rules for the users with particular cultural background. It is a critical stage, allowing the developers to build a computational system for the designers. The designers define the design objectives in advance, and discuss their professional opinions and technical concerns of the targeted design objects and users with the developers. The developers research into the related technical and cultural issues and suggest suitable representation scheme of shape grammars to the designers. Both the designers and developers keep verifying their understanding for how the system is created by considering the questions of why they need such a system to design during the design process.

In an attempt to study the questions, the developers should introduce the advantages and limitations of applying shape grammars to design to the designers, especially the factors affecting the quality of shape grammar rules. The quality of shape grammar rules is not simply justified by considering the complexity of the shapes in the shape grammar rules. Technical factors include the level of abstraction in representation, the control of labels, and the way to transformation. Practical factors include the cultural diversity and harmonization such as Chinese and Western comparison, the stylistic consistency, the ways to representing and modifying designs, and the brand image.

For more complex systems, the modification of the basic mechanism of shape grammars is unavoidable. It is the driving force to advance the mechanism, as design is a complex subject that integrates different branches of sciences and users with a diversity of cultural background. For instance, the exterior of a vehicle is developed for both Chinese and Western female users using new material that will change its color and form with reference to temperature changes. The shape grammar rules should be developed for such targeted users, and incorporated with such special physical material characteristics.

6.2.2. Evolution

Evolution is the design activity in which a blueprint of exploration is established and implemented in a well-controlled environment. Shape grammar developers should consider the questions like why do shape grammar rules need to be evolved? This issue is addressed in the section concerning the second layer of the new genetic representation (GP-GA-SG representation) and control strategies. The blueprint should be developed to solve the conflicts in evolving a shape grammar.

The question like which elements of a shape grammar are needed to be evolved should also be considered by the developers. This issue is addressed in the section concerning the third layer (the SG layer) of the GP-GA-SG representation. The issues like what is the difference between a shape grammar and a set of shape rules, what is the nature of the relationship between a shape grammar and design requirements, are some sets of rules applied only sequentially, why are there so many small sets of alternative shapes, how are the rules defined to allow for a topologically diverse set of solutions, are addressed in this section as well. Only when all these questions are clarified by the developers can a clear picture of the blueprint of exploration of design then be developed and implemented, which leads to the success of this approach.

6.2.3. Control

Control is the design activity in which the rate of exploration determines the contraction or expansion of the design space. The design space can be contracted to avoid a radical change in generating the designs. Alternatively, the design space can be expanded for generating new designs without any constraints. This can be achieved by setting the exploration rate under the constraints of modifying the shape grammar rules.

The shape grammar developers should consider the questions like why the design process is required to be controlled. How to control the design process? Which elements are needed to be controlled? What are the advantages and disadvantages of introducing the control methodology in the exploration of designs? These issues are discussed in the section for the first layer (the GP layer) of the GP-GA-SG representation.

6.2.4. *Evaluation*

Evaluation is the design activity in which the results are evaluated by a set of evaluation criteria. Three critical issues leading to the results have to be justified: what the evaluation criteria are and why such criteria are chosen, how the evaluation process is structured and how to justify such method is appropriate, and who are involved and what are the significant differences of the interpretation of the results among different professionals like users, designers, and research scholars.

For the first issue, the evaluation criteria can be so complex, with many multidimensional variables. In the case study, the objective functions include the evaluation of configuration fitness, artificial selection fitness, volume, and parametric constraints. The computational complexity increases as the number of valuables increases and the variables are multidimensional, which cannot be compared directly. The selection of such variables for evaluation is determined by their significance of the results from the users' view and designers' preference.

For the second issue, the evaluation process is structured to effectively test the results with acceptable accuracy. It is necessary to balance between the cost of computational time and resources, and the merits of accuracy. In the case study, the system generates a population of 500 designs in accordance to different requirements for each generation. It is possible to construct all 500 designs and display them on the screen. Higher accuracy of artificial selection fitness can be obtained because the designers can have more choices in selecting the designs based on their preferences. The cost will be the expense of lengthy computational time in generating these designs and the designers' time in waiting the generation of the designs. A balance is obtained in the case study by assigning each individual design with a design number from 1 to 500. Only 12 numbers of the 500 designs are displayed on the screen to demonstrate how the designers can interact with the system and what the results would be in respect to the requirements. By evaluating the results visually and numerically, a clear picture of the complex effects produced by the multiobjective functions is depicted. Based on the analysis of the results, the designers can justify the evaluation criteria. In the case where inappropriate criteria are adopted, the designers can modify the control parameters and control strategies to explore new designs during the evolutionary design process.

For the third issue, different professionals like designers, research scholars, and users can be involved in the evaluation process. In the case study, one senior and two experienced product designers have given their comments to improve the computational approach. Although the designers are not focused in designing cameras, they have expressed in general on what kinds of results they expect after watching the demonstration. Further research can be conducted with professional designers with experience in designing cameras, and a more systematic evaluation approach with statistical analysis can be performed for the enhancement of the approach.

It is of interest that the interpretation of the results is quite different among the professionals in their particular area of expertise. In the early stage of this research, the research scholars have expressed their concerns from the computational point of view on the contradiction of random modification of forms and stylistic consistency, and the generative capability in design exploration (Lee & Tang, 2004; Ang et al., 2006). A significant difference to the designers' concern is found that the designers address more broadly the issues related to brand, culture, humanistic factors, and the flexibility of the computational approach in expressing their ideas.

The case study also seized the opportunities in asking the users' comments on the computational approach. Their concerns are related to functionality, cost, weight, style, and brand. It seemed that the evaluation criteria and the interpretation of the evaluation of the results among users with different cultures were so dispersed. For example, a compact product is considered as an advantage for carriage and as a drawback for the difficulty in grasping it from two users' view. The responses from the users are critical for the designers to justify their ideas and change the ways to use the system.

7. CONCLUSIONS

In this paper, the two critical issues related to product design exploration are addressed: the balance between stylistic consistency and innovation, and the control of design process under a great diversity of requirements. To address these two issues, the view of understanding the product design exploration, the vision, the scenario, and the mission that this paper takes are revisited. This paper takes the view that the exploration of designs is not only categorized as a problem-solving activity but also as a problem-finding activity. Based on this view, a vision is created and encompasses the belief that these two activities go hand in hand to accomplish the design tasks in an interactive design environment. With such belief, a scenario is envisaged when describing the interactions among the designers and the system within a unified framework of shape grammars. According to the scenario, a mission is defined to develop a computational framework that adopts an integration approach of two key computational techniques, shape grammars and evolutionary computing, for addressing the two critical issues in product design exploration.

This framework involved both broad theoretical and practical concerns about the two critical issues. For the issues of stylistic consistency, the theoretical concerns of how to formulate a style and maintain the stylistic consistency of a product using shape grammars have been appraised by the literature. The practical concerns of the impact of stylistic modification of a product employing shape grammars have been justified on the grounds of promoting brand image. These concerns are aware as possibilities to enlarge and strengthen the philosophical concept of product design. Although this paper focuses on the computational techniques in balancing the conflicts of stylistic consistency and innovation with shape grammars, further investigations on addressing stylistic issues like cultural and historical factors are required.

For the issues of control of design process, the theoretical concerns in the development process of shape grammars, including analyzing the existing designs, organizing and representing related design information to build a knowledge base of shape grammars for form generation and configuration, have been illustrated in the newly developed genetic representation and control strategy. The practical concerns for monitoring the design process through various activities starting from the preparation works to the implementation of shape grammars have been emphasized in the development of this framework. The framework provides an interactive environment for the designers to continuously evaluate the generated designs during the design process. To evaluate the effectiveness of the framework, the experiments have been set up to reflect the practical situations with which the designers have to deal. The results obtained from the experiments indicated that the design problems are refined from inconsistent specifications, over- or underconstrained conditions defined at the beginning of the design process to a more clear specification of requirements at the end of the design process. As a result, better solutions can be obtained during the design process.

In conclusion, the integration of shape grammars with evolutionary computing techniques facilitates the formulation of design knowledge from the existing designs with parametric shape grammars. The potential of this framework can be further explored in the future in a product design-oriented environment that involves complex form generation and configuration optimization. The system does all the complicated modeling tasks to construct a number of models from scratch with numerical analysis that can be evaluated effectively by the designers. This reduces the designers' time and allows the designers to concentrate their efforts on performing higher level of design activities such as evaluation of designs and making decisions. Further investigations on designing a user-friendly interface is required to facilitate the designers to master the complexity of control parameters effectively.

ACKNOWLEDGMENTS

This project is supported by a PhD scholarship from Hong Kong Polytechnic University. The first author expresses his gratitude to Professor John Frazer for his valuable comments and to all of the members of the Design Technology Research Centre (DTRC) for their help and support for this project.

REFERENCES

Agarwal, M., & Cagan, J. (1998). A blend of different tastes: the language of coffeemakers. *Environment and Planning B: Planning and Design 25*, 205–226.

Agarwal, M., Cagan, J., & Constantine, K.G. (1999). Influencing generative design through continuous evaluation: associating costs with the coffeemaker shape grammar. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 13*, 253–275.

Ang, M.C., Chau, H.H., McKay, A., & de Pennington, A. (2006). Combining evolutionary algorithms and shape grammars to generate branded product design. *Proc. Design Computing and Cognition*, pp. 521–539.

Bralla, J.G. (1998). *Design for manufacturability handbook*, 2nd ed., pp. 1–17. New York: McGraw–Hill Handbooks.

Brown, K.N., & Cagan, J. (1997). Optimized process by generative simulated annealing. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 11*, 219–235.

Brown, K.N., McMahon, C.A. & Sims, W.J.H. (1995). Features, aka the semantics of a formal language of manufacturing. *Research in Engineering Design 7*, 151–172.

Cagan, J. (2001). Engineering shape grammars. *Formal Engineering Design Synthesis 3*, 65–92.

Çağdaş, G. (1996). A shape grammar model for designing row-houses. *Design Studies 17*, 35–51.

Chase, S.C. (2002). A model for user interaction in grammar-based design systems. *Automation in Construction 11*, 161–172.

Ding, L., & Gero, J.S. (2001). The emergence of the representation of style in design. *Environment and Planning B: Planning and Design 28(5)*, 707–731.

Dorst, K. (2006). Design problems and design paradoxes. *Design Issues 22(3)*, 4–17.

Dorst, C.H., & Cross, N.G. (2001). Creativity in the design process: co-evolution of problem-solution. *Design Studies 22*, 425–437.

Duarte, J.P., Ducla-Soares, G., Caldas, L., & Rocha, J. (2006). An urban grammar for the Medina of Marrakech: towards a tool for urban design in Islamic contexts. *Proc. Design Computing and Cognition*, pp. 483–502.

Frazer, J.H. (2002). Creative design and the generative evolutionary paradigm. In *Creative Evolutionary Systems* (Bentley, P.J., & Corne, D.W., Eds.), pp. 253–274. San Mateo, CA: Morgan Kaufmann.

Gero, J.S. (1992). Creativity, emergence and evolution in design. In *2nd Int. Round Table Conf. Computational Models of Creative Design* (Gero, J.S., & Sudweeks, F., Eds), pp. 1–28, Department of Architectural and Design Science, University of Sydney.

Gero, J.S. & Ding, L. (1997). Exploring style emergence in architectural design. *Proc. CAADRIA '97*, pp. 287–296.

Gero, J.S., Louis, S.J., & Kundu, S. (1994). Evolutionary learning of novel grammars for design improvement. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 8(2)*, 83–94.

Hsiao, S.W., & Chen, C.H. (1997). A semantic and shape grammar based approach for product design. *Design Studies 18(3)*, 275–296.

Janssen, P. (2004). *A design method and computational architecture for generating and evolving building designs*. Doctoral dissertation, Hong Kong Polytechnic University.

Janssen, P., Frazer, J.H., & Tang, M.X. (2002). Evolutionary design systems and generative processes. *The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies 16*(2), 119–128.

Kanal, L., & Kumar, V. (1988). *Search in Artificial Intelligence*. Berlin: Springer–Verlag.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection* . Cambridge, MA: MIT Press.

Lee, H.C., & Tang, M.X. (2004). Evolutionary shape grammars for product design. *Proc. 7th Int. Conf. Generative Art*, pp. 309–322.

Leung, T.P. (2006). Towards a knowledge economy of Hong Kong based on creative and innovative design. *Journal of Designing in China 2*, 42–55.

Li, A.I. (2004). Styles, grammars, authors, and users. *Proc. Design Computing and Cognition*, pp. 197–215.

McCormack, J.P., & Cagan, J. (2002). Designing inner hood panels through a shape grammar based framework. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 16(4)*, 273–290.
Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. London: Springer–Verlag.
Orsborn, S., Cagan, J., Pawlicki, R., & Smith, R.C. (2006). Creating cross-over vehicles: defining and combining vehicle classes using shape grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 20*, 217–246.
Ozkaya, I., & Akin, O. (2006). Requirement-driven design: assistance for information traceability in design computing. *Design Studies 27*, 381–398.
Pugliese, M.J., & Cagan, J. (2002). Capturing a rebel: modelling the Harley-Davidson brand through a motorcycle shape grammar. *Research in Engineering Design: Theory Applications and Concurrent Engineering 13(3)*, 139–156.
Rosenman, M.A. (1996). A growth model for form generation using a hierarchical evolutionary approach. *Microcomputers in Civil Engineering 11(3)*, 163–174.
Rosenman, M.A. (2000). Case-based evolutionary design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 14*, 17–29.
Rosenman, M.A., & Gero, J.S. (1999). Evolving designs by generating useful complex gene structures. In *Evolutionary Design by Computers* (Bentley, P.J., Ed.), pp. 345–364. San Francisco, CA: Morgan Kaufmann.
Schmidt, L.C., & Cagan, J. (1998). Optimal configuration design: an integrated approach using grammars. *ASME Journal of Mechanical Design 120*, 2–9.
Shea, K. (1997). *Essays of discrete structures: purposeful design of grammatical structures by directed stochastic search*. Doctoral dissertation, Carnegie Mellon University.
Shea, K. (2001). An approach to multi-objective optimisation for parametric synthesis. *Proc. 13th Int. Conf. Engineering Design (ICED '01)—Design Methods for Performance and Sustainability, WDK 28*, pp. 203–210.
Shea, K. (2002). Creating synthesis partners. *Architectural Design: Contemporary Techniques in Architecture 72(1)*, 42–45.
Shea, K. (2004). Directed randomness. In *Digital Tectonics* (Leach, N., Turnbull, D., & Williams, C., Eds.) pp. 10–23. New York: Academy Press.
Shea, K., & Cagan, J. (1999). Languages and semantics of grammatical discrete structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 13*, 241–251.
Simon, H.A. (1984). The structure of ill-structured problems. In *Developments in Design Methodology* (Nigel, C., Ed.). New York: Wiley.
Simon, H.A. (1990). *The Sciences of the Artificial*, 2nd ed. Cambridge, MA: MIT Press.
Sourav, K., & Michael, H. (1996). A networks approach for representation and evolution of shape grammars. *Proc. Artificial Intelligence in Design '96*, pp. 291–310.
Stiny, G. (1980*a*). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, *7*, 343–351.
Stiny, G. (1980*b*). Kindergarten grammars: Designing with Froebel's building gifts. *Environment and Planning B: Planning and Design 7*, 409–462.
Stiny, G. (2006). *Shape, Talking About Seeing and Doing*. Cambridge, MA: MIT Press.

Ho Cheong Lee is a PhD student in computational design. He obtained a BSc in computer science from Victoria University of Technology in 1996 and MSc in computers in manufacturing from the University of Hong Kong in 2000. In 2002 he obtained a PhD scholarship from the Hong Kong Polytechnic University. From 1988 to 1998 he worked as a technician and engineer in design, manufacturing, and installation of structural steelworks and electrical and mechanical products. After gaining 10 years' experiences in industry, he then worked as a research assistant in the Design Technology Research Centre (DTRC) in the School of Design of the Hong Kong Polytechnic University. He was involved in the Centre's research initiative in developing artificial intelligence and computational techniques for Engineering and Product Design. He has participated in professional society activities and is a member of ACM (2002), ASME (2003), BCS (2004), IEEE (2003), and IET (2007), and an associate member of ACS (2002) and IMechE (2007). His research interests include computational design, particularly shape grammars and evolutionary design technology, their implementation, and their applications to engineering and product designs.

Ming Xi Tang is a Professor and the Associate Director of the Design Technology Research Centre in the School of Design at Hong Kong Polytechnic University. He received a BSc in 1981 and an MSc in 1984 from Huazhong University of Science and Technology in the People's Republic of China. In 1996 he obtained his PhD from the Department of Artificial Intelligence at the University of Edinburgh. From 1988 to 1997 he worked at the University of Edinburgh, University College of London, and University of Cambridge as a Researcher in the area of developing integrated and intelligent design supporting methods and systems. In 1997 he joined the School of Design of the Hong Kong Polytechnic University. Dr. Tang's research interests include artificial intelligence, generative and evolutionary design, complex form generation and visualization, and collaborative design technologies. He is the Executive Editor of the recently established *Journal of Designing in China*.

APPENDIX A: PARAMETRIC SHAPE GRAMMARS WITH LABELS

A.1. Construction of 3-D shape grammar rules for free-form generation

There are eight construction rules for free-form exterior main-body generation (Fig. A.1). Each rule has constraints applied to the control points with respect to their *XYZ* coordinates. The control points are set in the range [min*X*, max*X*], [min*Y*, max*Y*], [min*Z*, max*Z*]. Special arrangements of the shape grammar rules are allowed in both text and visual descriptions of the construction shape grammars.

Rule C1 starts with a rectangular shape with labeled points: a, b, c, and d. These labeled points specify the maximum boundary of the camera body. Within the boundary, any possible form can be generated provided that the forms generated are under the constraints specified in the subsequent sets of shape grammar rules. Rule C1 deforms the rectangular shape to a quadrilateral shape. This is the most critical transformation rule, which specifies the main skeleton of the exterior camera body.

Rules C2 and C3 define an unconventional camera skeleton based on the conceptual profile of a water droplet. This can be achieved by deforming the quadrilateral shape to a curved profile. The 3-D curved profile with labels c2, b3, and a2 identifies an unique digital lifestyle icon and provides handling comfort.

Rule C4 generates an arc either bending up as a round corner or down as a slot at the upper sharp corner b3. If a slot is generated, a circular shape mode dial button can be placed to the slot. The

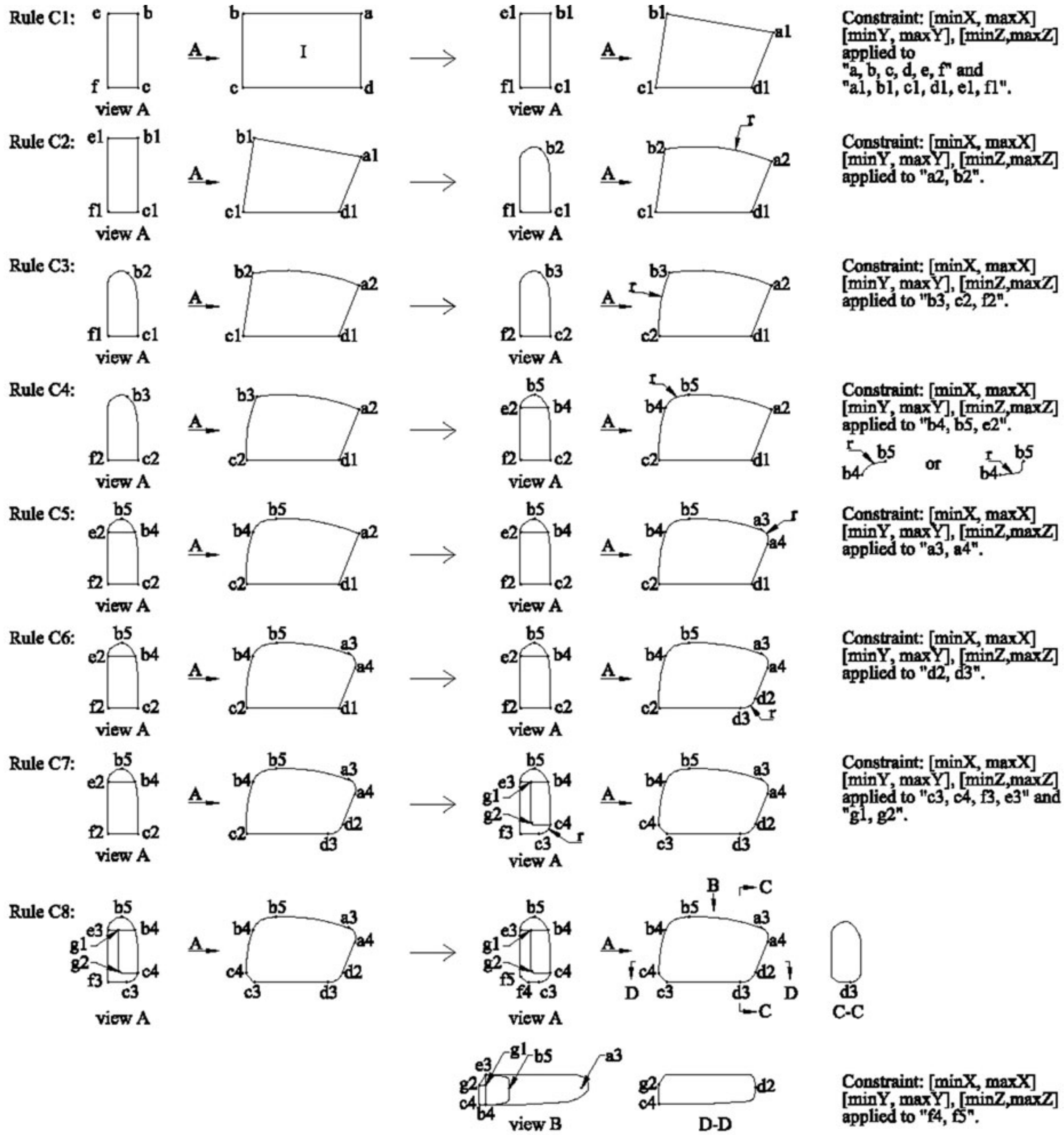

Fig. A.1. Construction shape grammar rules for the free-form exterior of main body generation.

mode dial supports the ergonomic control of digital camera. The two end points of the round corner are labeled with b4 and b5.

Rules C5 and C6 modify the upper sharp corner a2 and lower sharp corner d1 of the exterior to two arcs with a different radius *r*. The end points of the two arcs are labeled with a3, a4, and d2, d3, respectively. Avoiding sharp corners and using generous fillets and radii are a universal design rule for most of the products (Bralla, 1998). Both the manufactured part and tool can have a longer lifetime if generously rounded corners are used. In product design, generous radii and fillets are greatly preferred. The radii and fillets ensure esthetic quality and handling comfort.

Rule C7 makes a radius along the bottom part of the exterior starting from d2 to c4. The radius becomes a fillet from c4 to g2 and continuously extends to g1. The angle and depth of the fillet must be determined to closely match with the radius, not to adversely affect the esthetic quality.

Rule C8 creates another fillet of labels f4 and f5 along the bottom part of the exterior. Large angles and depth of fillet should be avoided as the fillet will reduce the usable area for component placement in the back side of the digital camera. Additional views (view B, Section C-C and D-D) are provided for clear indication of the overall profile of the exterior of the camera body based on this rule.

A.2. Construction of 2-D shape grammar rules for component generation

The second set of construction rules is for the design of components (Fig. A.2). Rule C9 uses the label MD to develop a unique rotating

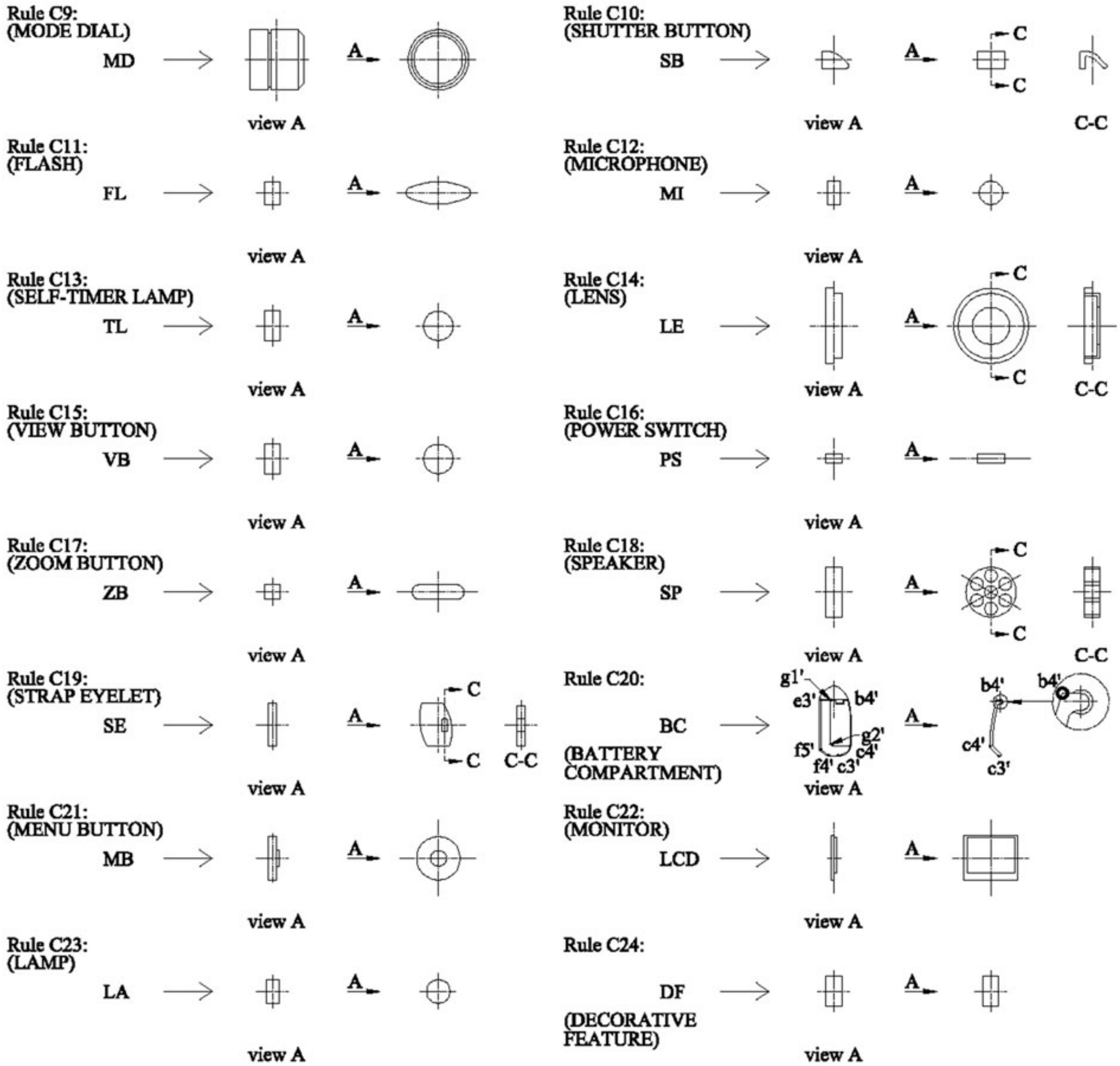

Fig. A.2. Construction shape grammar rules for component generation.

mode dial. The mode is a special feature that facilitates the ergonomic control of the camera. It allows the user to spin to set the camera. For ease of control consideration, a slot can be tailor made by rule C4 at the top corner of the camera's body for the mode dial to be placed. Once the user spins to select the desired camera mode, such as taking pictures, recording movies or reviewing images taken, the camera can be turned on by pressing the power switch. Therefore, the power switch is positioned close to the mode dial. The mode dial is an optional component if other control buttons with the same functionalities are used instead.

Rule C10 produces a shutter button from label SB. For conventional designs, some empirical guidelines can be followed. In designing the shutter button, the basic requirement is to facilitate the user with ease of use and comfortable feeling. Most users are right handed; most of the shutter buttons are allocated at the top right corner of the camera. This allows the user's finger to naturally locate the shutter button. This common practice becomes the rule of thumb in allocating the shutter button position.

Rule C11 changes the label FL to an oval shape for the flash. When designing the flash for a compact size digital camera, the form will usually be designed in a rectangular shape. Other soft and practical forms like a semicircle flash shape can be used to keep uniformity to the camera body. Most of the flashes are positioned at the upper part of the camera body so that the flash is positioned higher than the lens. This arrangement ensures flash coverage by allowing more flash light to be spread across a wider area. However, because of the design constraints of the compact digital camera size, the flash is not positioned far enough from the lens.

Rule C12 constructs a microphone that provides additional features to add value to a digital camera. The first feature is voice memo or voice annotation, which allows the users to describe the photographs (still pictures) either right before or after they shoot. The second feature is movie mode with sound, which allows the users to take small movies, complete with sound, and process them into AVI or QT (QuickTime) files.

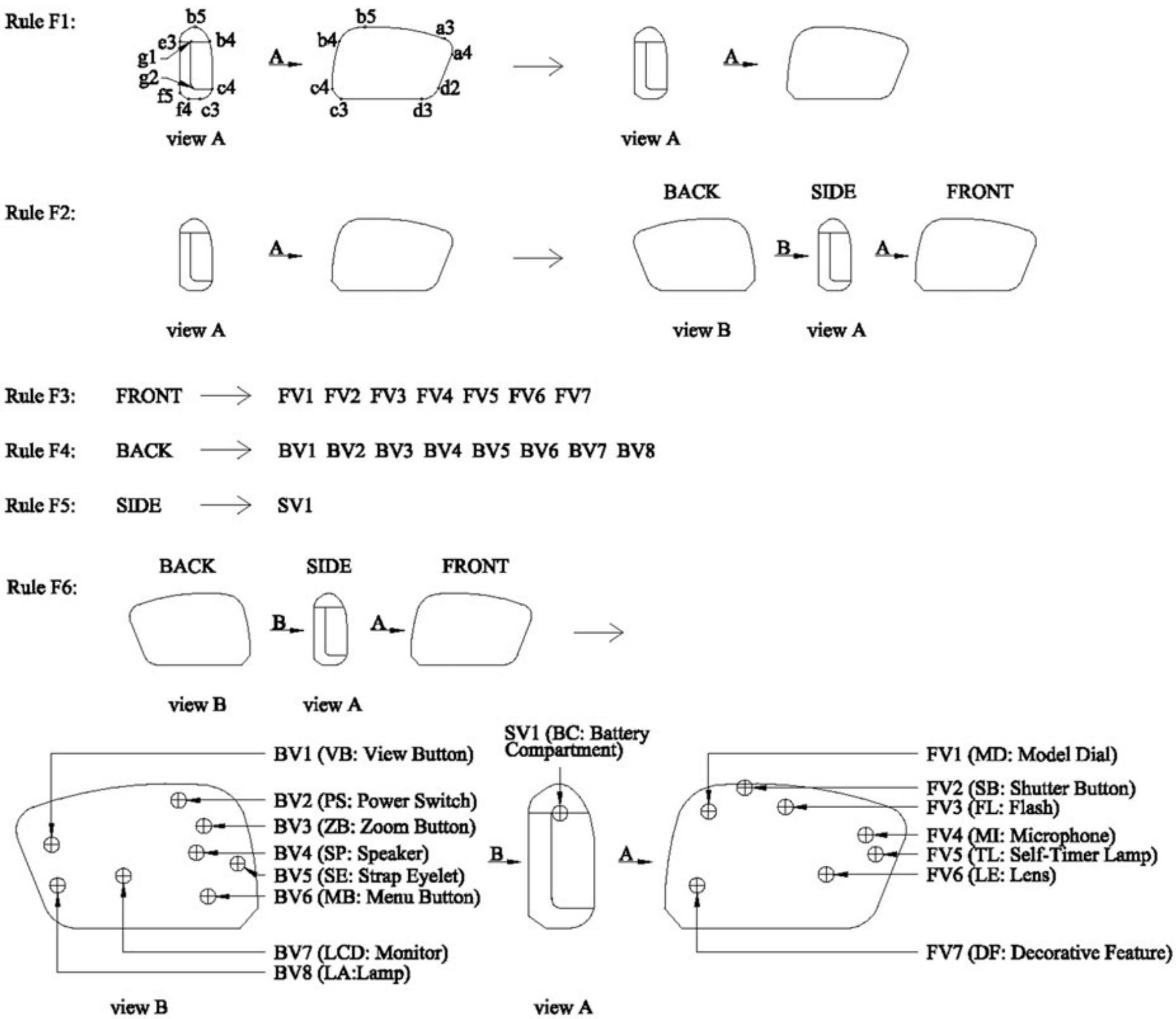

Fig. A.3. Configuration shape grammar rules for assembly.

Rule C13 builds a self-timer lamp that facilitates part of the operations of a self-timer feature. The self-timer feature allows the users who control the camera to include themselves in pictures. The main function of the self-timer lamp is to indicate the time left before the picture is taken by blinking for approximately 10 s. Therefore, when designing the self-timer lamp, it should be positioned on the front of the camera with just enough size for indicating purpose.

Rule C14 generates an optical zoom lens with circular shapes. With the new zoom technology, lens elements can be compressed into a shorter space for a 3× zoom. The optical zoom lens can be retracted into the body and extended from 1× to 3×. A built-in lens cover closes over the front element when the camera is powered down. Some of the camera designs provide lens thread for add-on lens or filters.

Rules C15 to C17 are for the control buttons. Rule C15 generates a quick-view button that displays the last picture taken on the monitor. Rule C16 designs a power switch that turns the camera on. As stated in rule C9, it is placed next to the mode dial. The power switch and mode dial can be viewed as a pair of buttons that perform the selection and switching functions in sequence. Rule C17 generates a zoom button that controls the zoom operation.

Some of the control buttons are designed with small size. This prevents them from being pressed unintentionally. Priority has to be determined when deciding either to increase the space between buttons for shooting comfort or purposely making them small.

Rule C18 generates a speaker that provides startup and shutter sounds. Rule C19 generates a strap eyelet that prevents dropping the camera inadvertently. Rule C20 generates a battery compartment cover for the replacement of the rechargeable battery. Rule C21 generates a menu button that allows the user to select different settings. The menu button is a navigation control pad with four arrow control pads around the button and one OK button in the middle. Rule C22 generates a monitor of rectangular shape for the display of pictures. A bright LCD with a size 1.8-in. display with higher pixel resolution can deliver over 160° angle of viewing and is a typical choice for monitors. Other monitors can be chosen provided that the monitors can be seen clearly from different angles and viewable even in a bright sunlight environment. Rule C23 generates a lamp for indicating purposes when downloading the images to the computer. Rule C24 generates a decorative feature. When there are no grip features specifically designed, the decorative features serve both griping and decorative purposes.

Rule F7: FV1 FV2 FV3 FV4 FV5 FV6 FV7 ⟶ MD FV2 FV3 FV4 FV5 FV6 FV7

Rule F8: MD FV2 FV3 FV4 FV5 FV6 FV7 ⟶ MD SB FV3 FV4 FV5 FV6 FV7

Rule F9: MD SB FV3 FV4 FV5 FV6 FV7 ⟶ MD SB FL FV4 FV5 FV6 FV7

Rule F10: MD SB FL FV4 FV5 FV6 FV7 ⟶ MD SB FL MI FV5 FV6 FV7

Rule F11: MD SB FL MI FV5 FV6 FV7 ⟶ MD SB FL MI TL FV6 FV7

Rule F12: MD SB FL MI TL FV6 FV7 ⟶ MD SB FL MI TL LE FV7

Rule F13: MD SB FL MI TL LE FV7 ⟶ MD SB FL MI TL LE DF

Rule F14: BV1 BV2 BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB BV2 BV3 BV4 BV5 BV6 BV7 BV8

Rule F15: VB BV2 BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB PS BV3 BV4 BV5 BV6 BV7 BV8

Rule F16: VB PS BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB PS ZB BV4 BV5 BV6 BV7 BV8

Rule F17: VB PS ZB BV4 BV5 BV6 BV7 BV8 ⟶ VB PS ZB SP BV5 BV6 BV7 BV8

Rule F18: VB PS ZB SP BV5 BV6 BV7 BV8 ⟶ VB PS ZB SP SE BV6 BV7 BV8

Rule F19: VB PS ZB SP SE BV6 BV7 BV8 ⟶ VB PS ZB SP SE MB BV7 BV8

Rule F20: VB PS ZB SP SE MB BV7 BV8 ⟶ VB PS ZB SP SE MB LCD BV8

Rule F21: VB PS ZB SP SE MB LCD BV8 ⟶ VB PS ZB SP SE MB LCD LA

Rule F22: SV1 ⟶ BC

Fig. A.4. The control of sequential generation of the components.

A.3. Configuration of 2-D shape grammar rules

The third set of shape grammar rules is configuration rules used for the allocation of components in the main body (Fig. A.3). The configuration rules use labels to maintain proper generation sequence. For the sake of clear identification of the components, rule F1 temporarily removes unnecessary labels after the generation of the exterior of the main body. Rule F2 divides the components into three groups: FRONT, BACK, and SIDE according to the spatial arrangement of the components. The labels FRONT, BACK, and SIDE refer to the components positioned with respect to the front, back, and side views. Each group of components is generated sequentially in accordance to a specific generation sequence.

Rules F3 to F5 assign the components for the three groups. Rule F3 assigns seven members of labels FV1 to FV7 to the FRONT group, rule F4 assigns eight labels BV1 to BV8 to the BACK group, and rule F5 assigns one label SV1 to the SIDE group. After the assignment of the components to the three groups, rule F6 allocates the components to the main body with their corresponding positions.

Rules F7 to F22 are used to control sequential generation of the components (Fig. A.4). Rules F7 to F13 generate the components for the FRONT group in sequence by modifying the labels from FV1 to FV7. Rules F14 to F21 generate the components for the BACK group in sequence by modifying the labels from BV1 to BV8. Rule F22 generates one component for the SIDE group by modifying the label SV1.

After the modification of labels, the corresponding construction rules are executed to generate the components. For example, in rule F20, the label BV7 is changed to "LCD," which is matched to the label in the corresponding construction rule C22; rule C22 is then executed to generate the LCD monitor.

APPENDIX B: AN EXAMPLE OF CONTROL STRATEGY APPLICATION

B.1. Setting of the first control strategy

One of the key issues in integrating a highly detailed shape grammar with evolutionary computing is that the random modification properties of evolutionary computing and the capturing style properties of shape grammar conflict with each other. The random modification of product form design removes the style of the product. More conflicts will occur if combining different shape grammar rules to derive new shapes.

The control strategies solve this problem by controlling the modification effects produced in the generated designs. The scope of

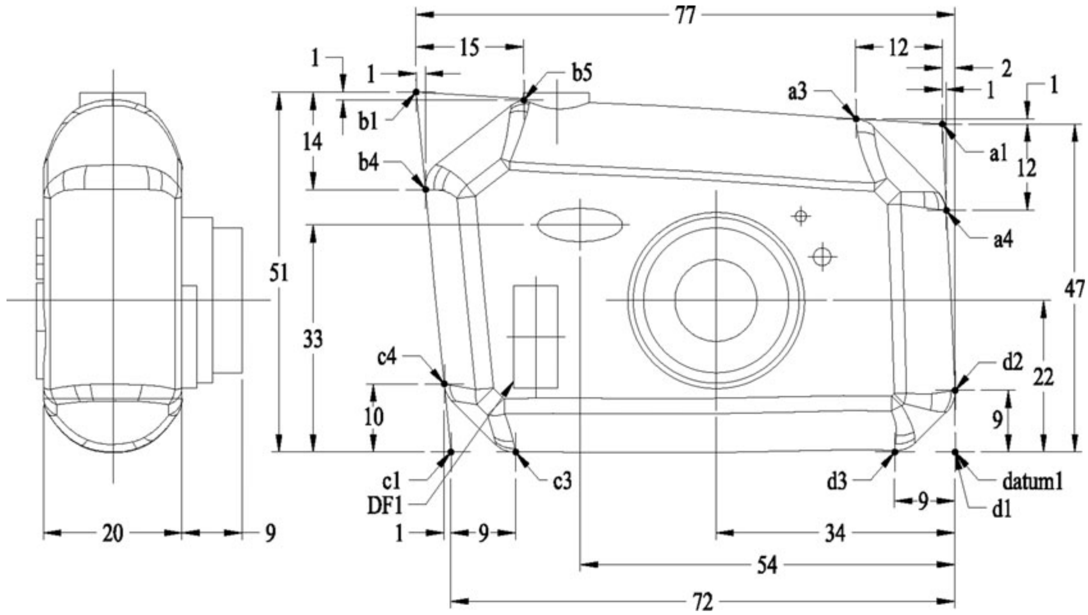

Fig. B.1. The initial setting of parameters and control variables for a compact digital camera.

applying this methodology is similar to objective functions that continuously modify the designs until the designs satisfy the requirements. However, the detailed implementation is different in that the control strategies focus on the modification to every single component. The designs generated can fulfill the general requirements defined by objective functions as well as specific requirements defined by control strategies.

For example, the first control strategy aims to regulate the generated designs with regular or symmetric properties. The generated designs can have different forms generated by different sets of shape grammar rules but all designs appear to have symmetric properties. This can be achieved by regulating the differences among the control points a_1, b_1, c_1, and d_1 (Fig. B.1). The difference pairs (a_1, b_1) and (c_1, d_1) in the x direction and

Table B.1. *Initial setting of parameters and control variables for the first control strategy*

GP Groups/ Elements	Modification to GA Interface	G-Groups/Elements	Mapping to SG Interface	SG Groups/Elements
Group 1		Group 1		Group 1
GP_1–GP_8	$GA_n = GP_n + GA_n$, (n = 1–8)	GA_1–GA_8 {−2, 47, − 77, 51, −72, 0, 0, 0}	$a_1x = GA_1 + 0$, $a_1y = GA_2 + 0$, $a_3x = GA_1 + (-12)$, $a_3y = GA_2 + 1$, $a_4x = GA_1 + 1$, $a_4y = GA_2 + (-12)$, $b_1x = GA_3 + 0$, $b_1y = GA_4 + 0$, $b_4x = GA_3 + 1$, $b_4y = GA_4 + (-14)$, $b_5x = GA_3 + 15$, $b_5y = GA_4 + (-1)$, $c_1x = GA_5 + 0$, $c_1y = GA_6 + 0$, $c_3x = GA_5 + 0$, $c_3y = GA_6 + 0$, $c_4x = GA_5 + (-1)$, $c_4y = GA_6 + 10$, $d_1x = GA_7 + 0$, $d_1y = GA_8 + 0$, $d_2x = GA_7 + 0$, $d_2y = GA_8 + 9$, $d_3x = GA_7 + (-9)$, $d_3y = GA_8 + 0$	$a_1x = -2$, $a_1y = 47$, $a_3x = -14$, $a_3y = 48$, $a_4x = -1$, $a_4y = 35$, $b_1x = -77$, $b_1y = 51$, $b_4x = -76$, $b_4y = 37$, $b_5x = -62$, $b_5y = 50$, $c_1x = -72$, $c_1y = 0$, $c_3x = -63$, $c_3y = 0$, $c_4x = -73$, $c_4y = 10$, $d_1x = 0$, $d_1y = 0$, $d_2x = 0$, $d_2y = 9$, $d_3x = -9$, $d_3y = 0$
Group 2		Group 2		Group 2
GP_9–GP_{12}	$GA_n = GP_n$ (n = 9–12)	GA_9–GA_{12} {−34, 22, 54, 33}	$x_1 = GA_9 + 0$, $z_1 = GA_{10} + 0$, $x_2 = GA_{11} + 0$, $z_2 = GA_{12} + 0$	$x_1 = -34$, $z_1 = 22$, $x_2 = -54$, $z_2 = 33$
Group 3		Group 3		Group 3
GP_{13}	$GA_{13} = GP_{13}$	GA_{13}{1}	$FV_7 = DF_{GA13}$	$FV_7 = DF_1$

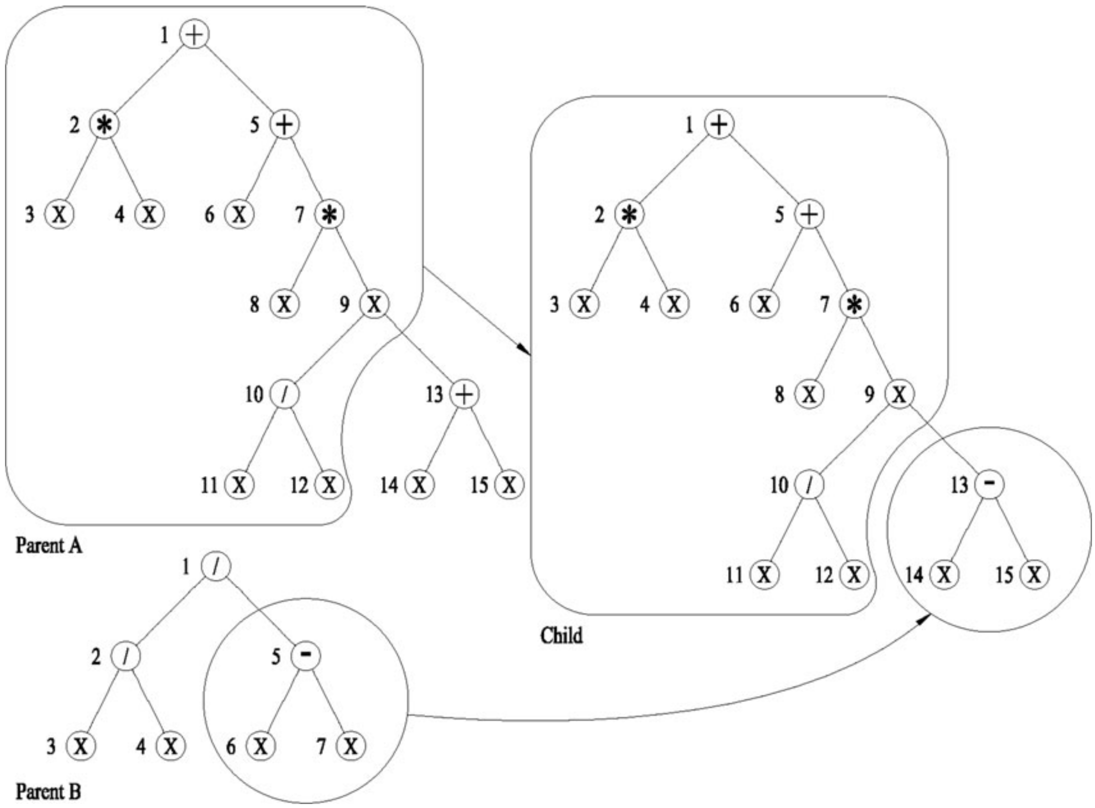

Fig. B.2. Genetic programming crossover: parent A crossed with parent B to produce the child.

(a_1, d_1) and (b_1, c_1) in the z direction are monitoring in the first control strategy. When either one of the difference pairs gets close to zero, the modification effects to the designs produced by the corresponding control variables of the GP-GA-SG interface stabilize. The corresponding control variables of the first layer: the GP interface in the GP-GA-SG interface, have no effects produced in subsequent layers, the GA interface, and the SG interface, except that better designs emerge.

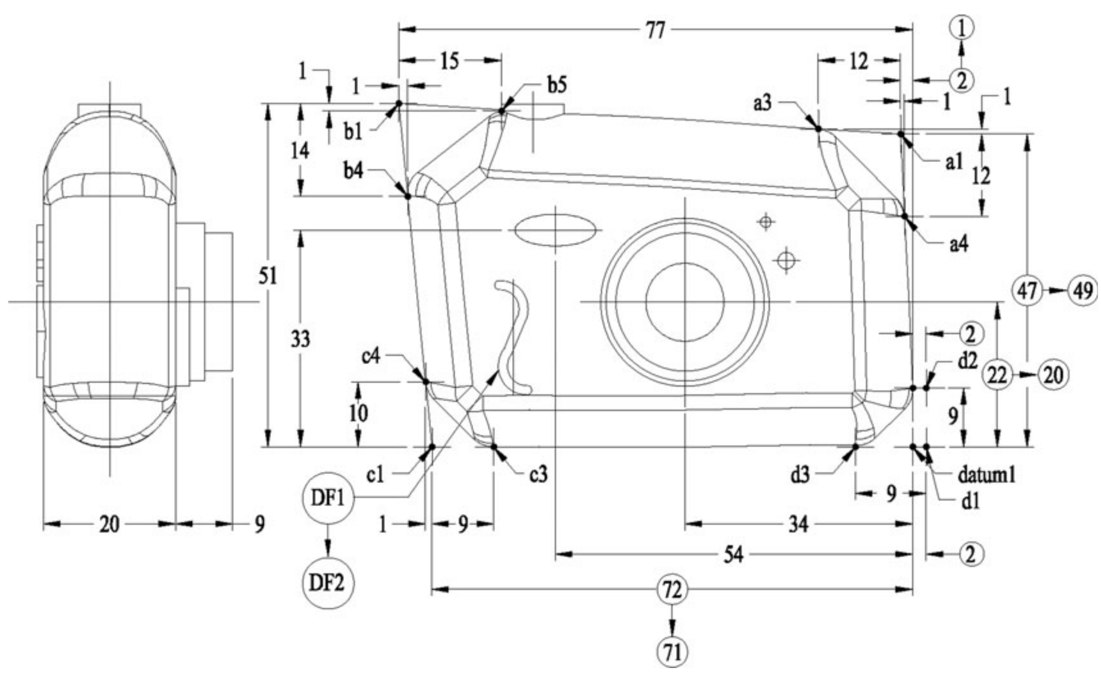

Fig. B.3. The modified setting of parameters and control variables by crossover operation for the first control strategy.

Table B.2. *Modified setting of parameters and control variables by crossover operation for the first control strategy*

GP Group Elements	Modification to GA Interface	GA Groups/Elements	Mapping to SG Interface	SG Groups/Elements
Group 1		Group 1		Group 1
GP_1–GP_8 {+, *, x, x, +, x, *, x}	$GA_1 = GP_1 + (-2) = 1 - 2 = -1$, $GA_2 = GP_2 + 47 = 2 + 47 = 49$, $GA_3 = GP_3 + (-77) = 0 - 77 = -77$, $GA_4 = GP_4 + 51 = 0 + 51 = 51$, $GA_5 = GP_5 + (-72) = 1 - 72 = -71$, $GA_6 = GP_6 + 0 = 0 + 0 = 0$, $GA_7 = GP_7 + 0 = 2 + 0 = 2$, $GA_8 = GP_8 + 0 = 0 + 0 = 0$	GA_1–GA_8 {−1, 49, −77, 51, −71, 0, 2, 0}	Same as initial setting	$a_1x = -1$, $a_1y = 49$, $a_3x = -13$, $a_3y = 50$, $a_4x = 0$, $a_4y = 37$, $b_1x = -77$, $b_1y = 51$, $b_4x = -76$, $b_4y = 37$, $b_5x = -62$, $b_5y = 50$, $c_1x = -71$, $c_1y = 0$, $c_3x = -62$, $c_3y = 0$, $c_4x = -72$, $c_4y = 10$, $d_1x = 2$, $d_1y = 0$, $d_2x = 2$, $d_2y = 9$, $d_3x = -7$, $d_3y = 0$
Group 2		Group 2		Group 2
GP_9–GP_{12} {x, /, x, x}	$GA_9 = GP_9 + (-34) = 0 - 34 = -34$, $GA_{10} = GP_{10} + 22 = -2 + 22 = 20$, $GA_{11} = GP_{11} + (-54) = 0 - 54 = -54$, $GA_{12} = GP_{12} + 33 = 0 + 33 = 33$	GA_9–GA_{12} {−34, 20, −54, 33}	Same as initial setting	$x_1 = -34$, $z_1 = 20$, $x_2 = -54$, $z_2 = 33$
Group 3		Group 3		Group 3
GP_{13} {−}	$GA_{13} = GP_{13} = 2$	GA_{13} {2}	Same as initial setting	$FV_7 = DF_2$

The advantages in separating the scope in the GP-GA-SG interface includes control of modification to the final designs as performed indirectly. This lies in the principle of evolutionary algorithm that the genotype can be evaluated indirectly by evaluating the solutions (phenotype).

Every product form feature has a set of shape grammar rules and parameters put into the third layer of the GP-GA-SG interface: the SG interface, the blueprint to describe how that product form feature, is built up from the shape grammar rules and encoded in the second layer: the GA interface. The control of such blueprint is specified in the first layer: the GP interface. In this arrangement, the modification of specific product form features can be monitored without affecting the optimization search performed on solutions.

The first control strategy is developed to illustrate the methodology for the regular or symmetric type of product form designs. An example of the initial setting of the first control strategy is shown in Figure B.1. The equations for modification and mapping processes between the GP interface to the GA interface, and the GA interface to the SG interface are shown in Table 1. Initial setting of the parameters for the GA interface and SG interface are specified in Table B.1. An illustration of the procedures to apply the first control strategy follows in Section B.2.

B.2. Operation of the first control strategy

An example of crossover operation is demonstrated and shown in Figure B.2. After the application of the crossover operation, the first control strategy determines the GP parameters. The GP parameters then modify the GA parameters, which in turn, map to the SG parameters. The resulting design is shown in Figure B.3, and the modified setting of the parameters and control variables is shown in Table B.2.