PLASTIC SURGERY IN THE EVOLUTIONARY DESIGN OF SPATIAL FORM

MICHAEL A ROSENMAN AND NICHOLAS PREEMA
University of Sydney, Australia

Abstract. This paper presents a methodology for producing good solutions for spatial form for non-routine design more efficiently. The methodology is based on augmenting a conventional evolutionary design approach with a method for improving suboptimal design solutions using domain-specific knowledge. This approach is based conceptually on the practice of plastic surgery, i.e. making minor adjustments to an entity, based on some desired qualities. While a conventional evolutionary design approach can produce reasonably good design solutions in an environment of knowledge uncertainty, plastic surgery, using domain-specific knowledge to manipulate the phenotype, can further improve such solutions in an efficient manner. This paper demonstrates how such a technique can be applied to the generation of spatial form.

1. Introduction

Design is characterized by the generation of form in response to some functional needs In architecture this form can be either spatial or physical (material). In either case, the generation of form can be implemented by placing a number of primitive elements in various combinations. In the case of spatial form, this constitutes the placement of a number of units of space together to produce required spaces. In non-routine design, the form required for a particular problem is not exactly known and a method, which can allow for the generation of many possible forms is advantageous as it can lead to unanticipated solutions.

Non-routine design tasks are characterized by the lack of knowledge available for their immediate solution (Coyne et al. 1990). Thus knowledge-lean approaches, such as evolutionary computation methods, are well suited to the task of non-routine design (Rosenman1996a; Bentley 1999, 2003; Koza et al. 2004). Evolutionary computation methods are characterized by their ability to arrive at reasonable solutions fairly quickly to begin with but

A Dong, A Vande Moere & JS Gero (eds), CAADFutures'07, 139-152.

then needing many generations to make subsequent small improvements, (Goldberg 1989; Parmee and Denham, 1994). A great deal of effort can be expended to make a small (but maybe critical) improvement and, in general, there is no guarantee that such an improvement will be found. In addition, in non-routine design, it is not always possible to perfectly specify the fitness function such that optimal or very good solutions will be found since the design task is not well-known. This paper argues that, even in such conditions, it is possible to obtain reasonable solutions within the bounds given and then, using these resulting solutions as a guide, make improvements to obtain better solutions.

Plastic surgery is a practice whereby features of an entity (generally a human) are altered to improve the appearance of that entity. In all cases, the effect is on the phenotype, i.e. the entity itself, and there is no change to the genotype (DNA). Since evaluation is done on the phenotype, any improvement to the phenotype gives the entity a better chance of survival or attaining its goal, e.g. attaining self esteem, attracting other entities, etc. Plastic surgery is generally done to correct minor features, e.g. making a nose smaller, etc. An entity exists, that is, it has been generated in some way, but is defective in some features and minor corrections are made (to the phenotype) to improve it. Specialized knowledge is required to recognize the defects and to modify the phenotype. Different domains require different specialized knowledge.

A conventional evolutionary method may be used, in situations where the form required is not known a priori, to produce possible solutions which are reasonably good but need some improvement. Once the solutions are produced, they can be examined and, where suitable, improved by making small modifications to the phenotype. This paper presents an approach that has the potential to significantly improve the ability of evolutionary design processes to produce good design solutions. While, an improvement in computational efficiency is important, it is more a case of the ability to actually produce solutions of high quality.

2. Evolutionary Design

While knowledge-rich approaches can solve problems where the problem is well defined and the knowledge and methods required are also known, they operate in specific problem areas with little capacity for producing innovative solutions. On the other hand, while knowledge-lean methods, such as evolutionary design, are good for discovering possible reasonable solutions where little knowledge is known a priori regarding the form of the solution, they are generally computationally expensive and may not be able to make the necessary improvements in a reasonable time with reasonable resources. Additionally, in an environment where there exists little a priori knowledge, it is not always possible to perfectly specify the requirements, i.e. formulate a ‘perfect’ fitness function.

The proposed approach combines knowledge-lean and knowledge-rich approaches to increase the efficiency of producing good design solutions in a non-routine design problem environment. The conventional evolutionary computation approach generates reasonably good solutions within given initial specifications and the proposed plastic surgery makes small modifications to those solutions based on local knowledge of the problem.

3. A Design Representation for Spatial form

In its simplest mode, the construction of form can be thought of as the set of decisions for locating a set of cells of substances, where a substance may be physical (composed of a physical material) or virtual (e.g. composed of graphic entities or pixels). The construction of a spatial entity may be considered as the allocation of a number of cells of a physical substance composed of a ‘space’ material. In an evolutionary design approach, a gene selects a module of substance and allocates it to some location. In the approach of Rosenman (1996a, 1996b, a gene locates a module of substance relative to another module. A gene, GN, is thus (M_1, M_2, L_{12}) where M_1 and M_2 are two modules of some substance and L_{12} is the operator for locating module M_2 relative to module M_1. A module, M_i, may be a single unit cell or a set of unit cells already grouped and, in general, M_1 and M_2 need not be composed of the same substance. In the design of spatial form, S will be a single substance composed of space.

Since spatial form can be represented as the shape of spatial composition, the aim being to produce shapes suitable for various functions. A cellular composition of space units can be represented as a composition of elementary polyhedral or polygonal units (Rosenman 1995, 1999).

4. The Design Solutions

When a number of squares are joined randomly the resulting shapes (polyminos) are not likely to show much regularity, especially if the number of squares is large. Figure 1 shows 40 random generations of 16-unit polyminos.

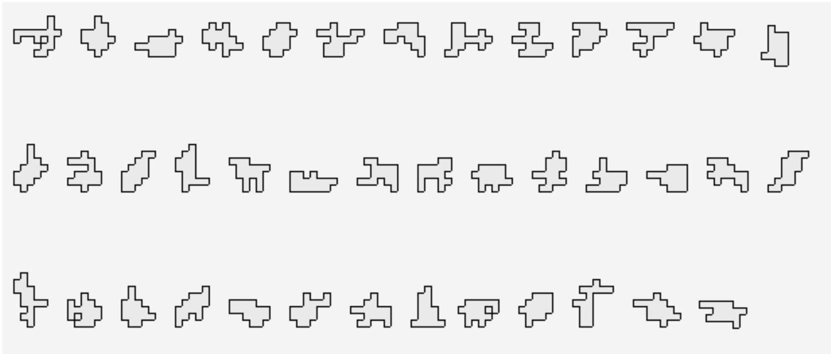

Figure 1. 40 random generations of 16-unit polyminos

The objective for architectural spaces will be to produce a shape with a contour having reasonably smooth edges. Figure 2 (a) shows a configuration of 12 cells that may arise after a number of generations. To produce an improvement such that the protrusion is removed and the indentation is filled (leading to a square in this case), Figure 2 (b), may take a great deal effort from the evolutionary computation process especially where the number of cells is large, e.g. >=100. While we can see that removing the protrusion and filling in the indentation would lead to a good solution, the evolutionary system based on random genetic operations (crossover and mutation) on the genotype may not be able to produce the required solution within a feasible timescale.

(a) an almost ‘perfect’ solution (b) an improved solution

Figure 2. Improvement of a design solution.

As stated previously, in an environment of non-routine design, it will generally not be possible to accurately state the precise requirements nor translate them into a fitness function capable of precisely driving the evolutionary system to perfect solutions. The example in Figure 1 has only 16 cells. However, if the scale of the cells in Figure 1 were reduced by a factor of 10, allowing for increments in length of say 10cms rather than 1m, the total number of cells would be 1600. For a large number of cells, the evolutionary process, will, after a number of generations, give some indication of possible satisfactory shapes but will, usually, not be able to perfectly smooth out all the protrusions and indentations. For example, the fitness function may be based on minimizing the perimeter to area ratio since this will tend to produce compact shapes and tend to minimize long perimeters. However, one will find that, with a large number of cells, the number of cells at the perimeter is small compared to the number of interior cells which are fairly well compacted. Thus most of the solutions will show a fairly high score for that fitness function. The process will not be able to make any significant improvement in any reasonable time.

Although it may be argued that one should find a more precise fitness function, this is not always possible. The approach taken here shows that it is possible to postulate some reasonable fitness function which drives the evolutionary system towards reasonably satisfactory solutions. Then, the strategy is to improve these solutions as required.

5. Plastic surgery in evolutionary design

In an evolutionary system, selection acts with respect to the phenotype. Those members whose phenotypes are judged to be well-suited to their environment will have a better chance of survival and of propagating their genes (Janssen et al. 2002). Thus any improvement in the phenotype, regardless of any change in the genotype, will improve that member's chance of survival and propagation. Of course, this improvement will not be transmitted to the member's descendants. In a design domain the fitness of the design is what counts, how it got to be that way is secondary.

The Merriam-Webster (2006) dictionary states that plastic surgery is:

"surgery concerned with the repair, restoration, or improvement of lost, injured, defective, or misshapen body parts"

Plastic surgery is aimed at improving the organism's survival in its environment, where survival may mean the organism's perceived state of happiness or its improved ability to attract partners as well as its improved ability to function better.

In this work, plastic surgery is proposed as a solution to improving a phenotype (design solution) generated through an evolutionary computation method. The modifications should be limited to relatively small remedial improvements. While it may be possible to make large alterations, this seems to be too large a departure from the solutions found leading to different forms and is not the aim of this work.

While an example in the domain of the generation of smooth polygons will be used to demonstrate the concepts, this paper suggests that the general principles of plastic surgery could be applied to other domains since design is seen generally as a process of locating suitable elements in a certain configuration.

6. Methodology

The implementation of plastic surgery consists of several transformation functions. There exist various smoothing algorithms mainly in image processing, where they are used to produce smoothed surfaces from polygonal or noisy surfaces (Hoppe 1996; Volino and Thalman 1998; Hobby 1998). Algorithms such as Potrace (2006) transform bitmap images into vector graphics. Another process uses sampling for anti-aliassing in ray tracing (Rossignac and Borrel 1992). Sampling works by overlaying a grid of larger cells on the form. Each cell is analyzed to determine what percentage of the cell is occupied. Cells with 50% or more occupation would be filled in completely, while those with less than 50% occupation would be left empty. The small-increment method is closer to the philosophy of making minor repairs rather than large-scale modifications and results in shapes closer to the original shapes than the sampling method which results in 'major reconstructions'.

Modifications can be carried out to various levels of refinement, i.e. with respect to the number of units to be treated. Figure 3 shows the various examples (defects) which may require modification.

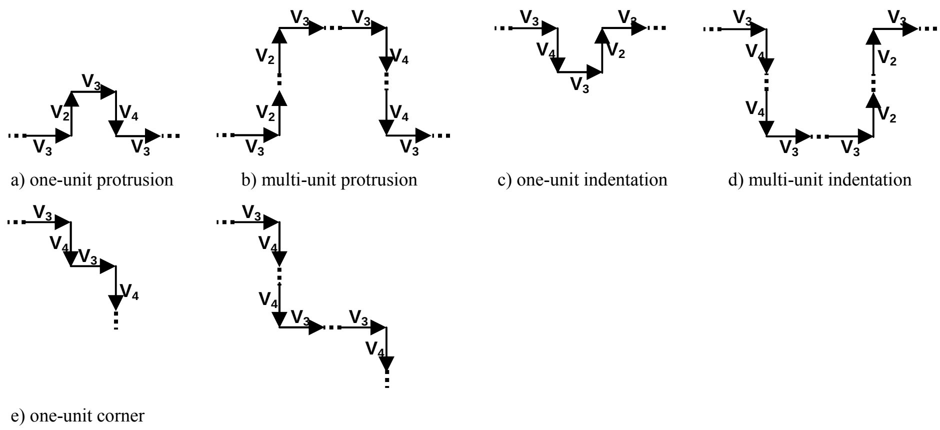

Figure 3. Cases for modification

These include protrusions, indentations and corners, ranging from one unit to several units. The number of units in each direction may depend on the scale, i.e. the total number of units in a shape. While Figure 3 shows defects on one edge or corner only, the defects may occur on any of the four edge or corner directions (for polymino shapes).

Figure 4 shows the rules for plastic surgery, i.e. modifying the phenotype (shape) according to the type of defect (protrusion, indentation or corner) and the number of units to be rectified in the two directions. Again, it should be noted that the defect may occur in any direction so that the depth and width of a defect are local to the particular direction.

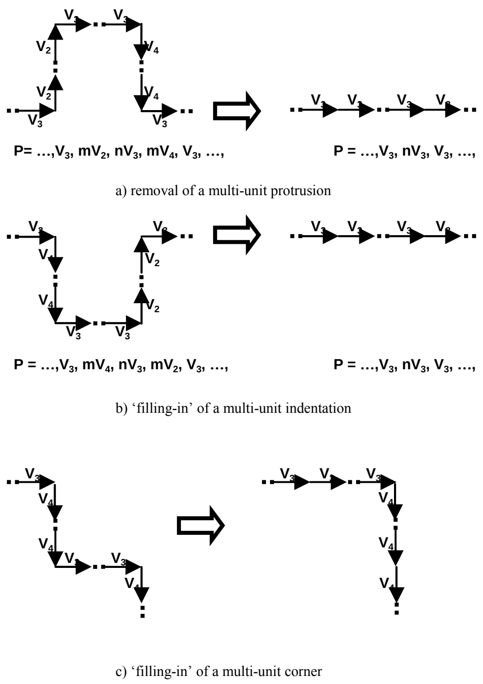

Figure 4. Rules for plastic surgery of defects

The level of refinement is set by setting the depth and width, in terms of number of units, for the plastic surgery to take effect. The degree of refinement and the order of implementation of the operations will determine the final result. Different parameters and sequences will produce different results. In the physical world it is not possible to try several alternatives, whereas in a computational process it is possible to try alternatives and select among them depending on the result. Figure 5 shows two different sequences of operations on a shape of 50 units based on the following operations or rules:

Rule 1:	Defect = protrusion	max depth = n	max width = 1
Rule 2:	Defect = indentation	max depth = 1	max width = 3
Rule 3:	Defect = corner	max depth = 1	max width = 1

Rule 1 states that all protrusions of width 1 unit, no matter their depth, are to be deleted.

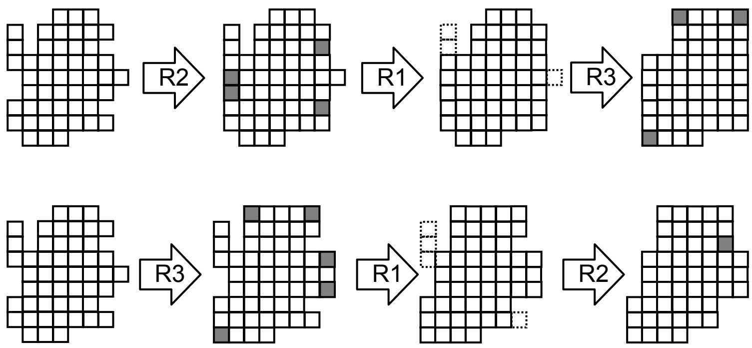

Figure 5. Two different sequences for plastic surgery

The shaded and dotted units show the units added or trimmed. The first solution has grown from 50 units to 54 units whereas the second solution has increased to 52 units. The size of the resulting solution depends on the number of units trimmed or filled. Since the number (size of the element) may be critical, some constraints may need be applied regarding the number of units trimmed or added or the number of elements trimmed may need to be balanced by the number of elements added (and vice versa). In a very large number of units, the number of units adjusted may not makes a significant change to the size of the shape since the number of units on the perimeter is small compared to the total number of units.

A method for recognizing which shapes are suitable for plastic surgery is based on the measure of fitness of the shape as well as on a measure of the number of defective units with respect to the shape's perimeter.

7. Implementation Example

An example in the domain of room designs was implemented. Rooms need not necessarily have rectangular shapes nor do they necessarily have to have 'smooth' walls. They may have recesses but generally these need to be large enough to accommodate furniture such as bookshelves etc. So, in general, small protrusions and recesses in the perimeter are not acceptable. The aim is to generate shapes for a room of 18 m^2. With a square unit of 1 m x 1 m there would be only 18 units and the variations in acceptable dimensions of length and width would be quite limited. Therefore, a variation of 300 mm in each dimension was set to allow for a wide range of possible dimensions. This results in the arrangement of 200 square units of 300 mm x 300 mm.

7.1. FITNESS FUNCTION

The fitness functions used were those used in Rosenman (1996a, b). A function that tends to smooth the perimeter is that of minimizing the perimeter. The minimum perimeter of a polymino shape is ideally a square. While not all given areas (e.g. 200 units) can form squares, using this fitness function will tend to make shapes more compact, thus reducing the length of the perimeter. The aim of room design is not to necessarily produce square or rectangular shapes but to use the fitness function to drive the evolutionary process towards such shapes, generating other suitable shapes in the process. Another measure of the smoothness of the perimeter is that of minimizing the number of corners. The minimum number of corners of a polymino shape is 4. Obviously a square has both the minimum area and the minimum number of corners. This function has a tendency to prefer L-shapes over T-shape. Both these shapes will have the same perimeter to area ratio but the L-shape has six corners compared to eight for the T-shape. While the first function tends to prefer more compact shapes, the second function will assign a maximum value to a rectangular shape no matter what its proportion.

For the first function, minimizing the perimeter to area function, the fitness is given by:

$$f1 = (MaxP - P / MaxP - MinP) \times 100 \quad \text{------------------------} \quad (1)$$

where

- f1 = fitness function wrt minimum perimeter to area
- MaxP = maximum possible perimeter for a shape of n units
- P = perimeter of generated shape
- MinP = (ideal) minimum perimeter of a shape of n units

and

- Min P = $4\sqrt{n}$ (ideal square)
- MaxP = $2n + 2$ (e.g. shape of 1 unit width and n units length)

where

- n = number of units

For the second function, that of minimizing the number of corners, the fitness is given by:

$$f2 = (Max\ C - C / MaxC - 4) \times 100 \quad \text{------------------------} \quad (2)$$

where

- MaxC = maximum possible number of corners for a shape of n units
- C = number of corners of generated shape

and

- MaxC = $2n$ (e.g. fully stepped shape)

Both functions use a ratio of the range of possible values to determine the normalized percentage fitness of the shape. The total fitness is given as:

TF = (f1 + f2) /2 ------------------------ (3)

Different weightings could be used for each fitness function to influence the shape towards one or the other but for this example a simple weighting of 1 for each has been used for simplicity.

7.2. METHOD

A C++ program for Windows was written to generate and evolve a population of polymino shapes using a genetic algorithm based on cell addition using the edge vector representation discussed previously and then to perform plastic surgery. The inputs to the generation and evolution are: the number of units, the number of members of the population and the maximum number of generations to be run. The genetic algorithm may terminate before the maximum number of generations is reached if it converges or remains stable. A run converges if the average fitness is within 5% of the best fitness and remains stable if there is no significant change in the best solution or average fitness over a specified number of generations. Simple one-point crossover was used with the best of the two populations (parent and child) kept to preserve the best solution. The remaining members of the new generation are selected using the roulette wheel method. The inputs to the plastic surgery are the width and length of the three repair cases (protrusion, indentation and corner) specifying the scale of the repair.

The program was run several times with the following parameters:

No. of units 200
Population 40
Max. no. of generations 60
Max. depth 1
Max width 3

7.3. RESULTS

Results were similar over a number of runs. Figure 6 shows the results of one of these runs. Figure 6 shows a typical growth in fitness over the 60 generations using the conventional evolutionary process. The average fitness of the population is 72.6%. As can be seen from the graph, the population has arrived at a fairly stable state and it could take a very large number of additional generations to produce any improvement (if any is possible). After the application of plastic surgery, the average fitness jumps to 89.3%. This is a 23% improvement.

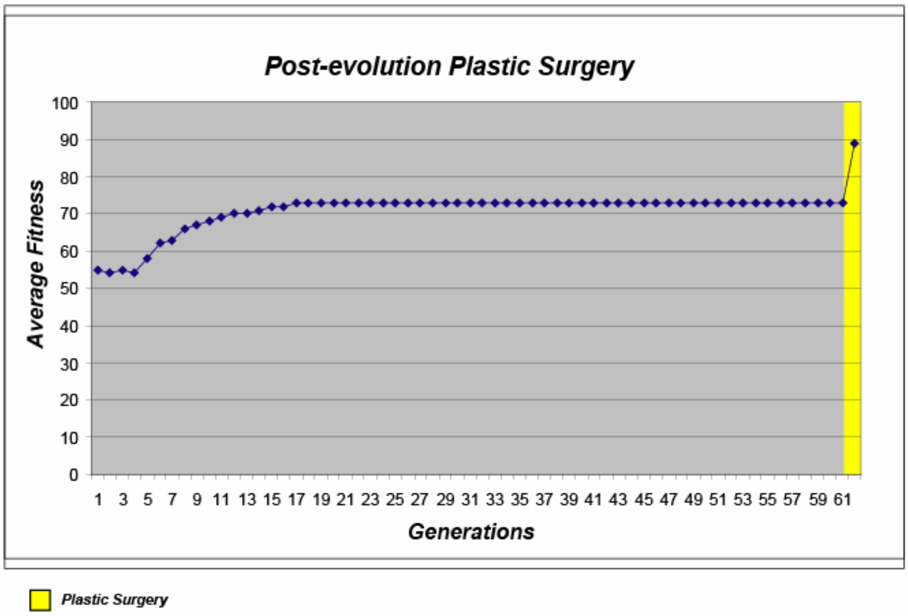

Figure 6. Effect of plastic surgery after the evolutionary process

Figure 7 shows three of the shapes subjected to plastic surgery. It can be seen that these three members, previous to the plastic surgery, had a high fitness (95.6 to 96.6) even though their shapes are not all that good. The first shape is better than the other two but still has some small changes in direction in the upper left-hand part. The relatively high fitness values are due to the fitness function used which, in part measures the compactness of the shape. Since a large proportion of the shapes is indeed compact, the fitness values are high and there is little pressure to improve them. In the previous work (Rosenman 1996a, b) where only relatively small number of units were used (maximum 25) this problem did not exist. It can be seen that while the application of the plastic surgery has improved the fitness values, its main contribution is in producing better shapes, i.e. shapes with fewer small protrusions, etc. No method was used to ensure that the size of the shape (room) remained the same and the first shape has increased to 208 units, the second to 206 and the third to 207, an increase of less than 5% in all cases. Note that while none of the shapes shown are rectangles, nevertheless they could be suitable as rooms in certain instances.

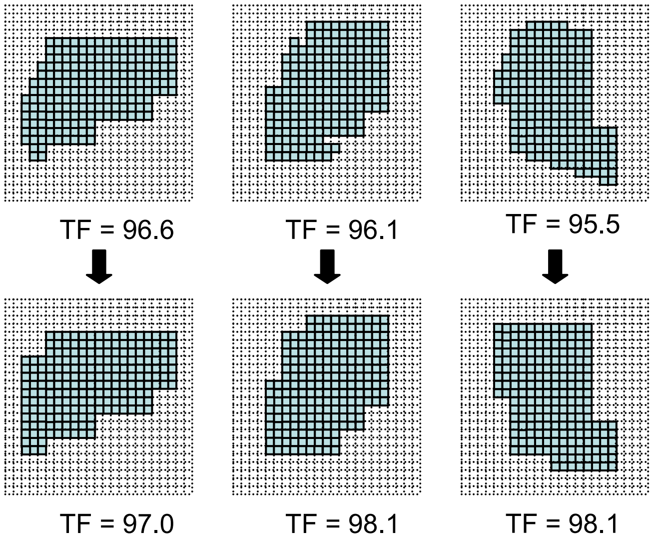

Figure 7. Members from the run before and after plastic surgery

8. Summary

This paper has presented a method of generating spatial forms using an evolutionary approach. It has argued that evolutionary design methods are suitable approaches for non-routine design generation since they are knowledge-lean and hence suitable for situations where there is little a priori knowledge available regarding any associations between the requirements and the form to be generated. An example using square cells was used for simplicity although the approach could be generalized to 3-D polyhedral shapes. However, it was argued that for complex objects with large number of cells, and with possibly imprecise fitness functions, the solutions arrived after a reasonable effort may still need improvement. It was shown that, although, the solutions obtained are reasonable, i.e. they will show the general direction of the form to be achieved, they need minor modifications to achieve more satisfactory results.

The results of the implementation of the example show that plastic surgery is a useful method for efficiently improving design solutions where the evolutionary process has achieved stability. Plastic surgery is seen as a knowledge-based mutation of the form (phenotype). Though illustrated in the context of the 2D cellular formation of shapes and the smoothing of irregular perimeters, it is a general concept applicable to 3D forms and other applications. Other applications will need to use domain specific knowledge for their repair rules.

9. Future Work

It was argued that design, in general, is the allocation of units of substances to create form and that non-routine design suffers from incomplete knowledge. Thus, while this work is specific to space layout, the concept of generate-and-fix is more general. Each domain, however, needs its domain-specific knowledge to specify fitnesses, recognize defects and provide methods for repair. Future work will generalize the approach to other design domains. This will require that one takes into consideration the allocation of units of different substances and the repair of the whole. This will mean deciding not only what form needs to be repaired but what substance should be used.

As in the human example, any modification to the phenotype (design solution) is not transmitted to the genotype. Any 'children' may carry the defective genes and reproduce the same defects. However, in design, if the modified design solution is the final solution required, and no more processing is to take place, then this does not matter as the genotype was just the means to the end and is no longer of any interest. However, if the modified design solution is only a part solution and is required to take part in further evaluation, e.g. as a component in a hierarchical system, a problem exists since all evolutionary operations are carried out on the genotype. In that case, genetic re-engineering of the genotype will be required. Future work will look at the implementation of the re-engineering method for various stages of an evolutionary process.

References

Bentley, PJ (ed): 1999, *Evolutionary Design by Computers*, Morgan Kaufman, San Francisco, CA.

Bentley, PJ: 2003, Natural design by computers, *Proc of the AAAI Symposium on Computational Synthesis*, Stanford University, Palo Alto, CA.

Coyne, RD, Rosenman, MA, Radford, AD, Balachandran, MB and Gero, JS: 1990, *Knowledge Based Design*, Addison-Wesley, Reading, Mass.

Goldberg, DE: 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass.

Hobby, JD: 1998, Smoothing digitized contours, *Theoretical Foundations of Computer Graphics and CAD*, pp. 777-793.

Hoppe, H: 1996, Progressive meshes in computer graphics, *SIGGRAPH'96*, pp. 99-108.

Janssen, P, Frazer, J and Ming-Xi, T: 2002, Evolutionary design systems and generative processes, *Applied Intelligence* **16**: 119-128.

Koza, JR, Jones, LW, Keane, MA, Streeter, MW and Al-Sakran, SH: 2004, Towards automated design of industrial-strength analog circuits by means of genetic programming, *in* U-M Oreilly, RL Riolo, G Yu and W Worzel (eds), *Genetic Programming Theory and Practice II*, Klewer Academic, Boston, Chapter 8, pp. 121-142.

Merriam-Webster: 2006. Retrieved December 4, 2006 from http://www.m-w.com/.

Old, RW and Primrose, SB: 1994, *Principles of Gene Manipulation: An Introduction to Genetic Engineering (studies in Microbiology)*, Blackwell Science 5th ed, Oxford, UK.

Parmee, IC and Denham, J: 1994, The integration of adaptive search techniques with current engineering design practice, *Proc.of Adaptive Computing in Engineering Design and Control '94*, University of Plymouth, Plymouth, pp. 1-13.

Potrace: 2006, Transforming bitmaps into vector graphic. Retrieved December 4, 2006 from http://potrace.sourceforge.net.

Rosenman, MA: 1995, An edge vector representation for the construction of 2-dimensional shapes, *Environment and Planning B:Planning and Design* **22**: 191-212.

Rosenman, MA: 1996a, The generation of form using an evolutionary approach, *in* JS Gero and F Sudweeks (eds), *Artificial Intelligence '96*, Kluwer Academic, Dordrecht, The Netherlands, pp. 643-662.

Rosenman, MA: 1996b, A growth model for form generation using a hierarchical evolutionary approach, *Microcomputers in Civil Engineering*, special issue on Evolutionary Systems in Design, **11**(3): 161-172.

Rosenman, MA: 1999, A face vector representation for the construction of polyhedra, *Environment and Planning B: Planning and Design* **26**: 265-280.

Rossignac, JR and Borrel, P: 1992, Multi-resolution 3D approximations for rendering complex scenes, *in* B Falcidieno and TL Kunii (eds), *Geometric Modelling in Computer Graphics*, Springer-Verlag, Genoa, Italy, pp. 455-465.

Volino, P and Magenat Thalman, T: 1998, The SPHERIGON: A simple polygonal patch for smoothing quickly your polygonal meshes, MIRAlab Copyright Information. Retrieved December 4, 2006 from http://www.miralab.unige.ch/papers/50.pdf.