From Mondrian to Frank Lloyd Wright: Transforming Evolving Representations

Thorsten Schnier
John S Gero

Key Centre of Design Computing
Department of Architectural and Design Science
Faculty of Architecture
University of Sydney, NSW, 2006, Australia
fax: +61-2-9351-3031
email : {thorsten,john}@arch.usyd.edu.au

Abstract:

If a computer is to create designs with the goal of following a certain style it has to have information about this style. Unfortunately, the most often used method of formal representations of style, shape grammars, does not lend itself to automated implementation. However, It has been shown how an evolutionary system with evolving representation can provide an alternative approach that allows a system to learn style knowledge automatically and without the need for an explicit representation. This paper shows how the applicability of evolved representation can be extended by the introduction of transformations of the representation. One such transformation allows mixing of style knowledge, similar to the cross-breeding of animals of different races, with the added possibility of controlling exactly what features are used from which source. This can be achieved through different ways of mixing representations learned from different examples and then using the new, combined representation to create new designs. In a similar manner, information learned in one application domain can be used in a different domain. To achieve this, either the representation or the genotype-phenotype transformation has to be adapted. The same operations also allow mixing of knowledge from different domains. As an example, we show how style information learned from a set of Mondrian paintings can be combined with style information from a Frank Lloyd Wright window design, to create new window designs. Also, we show how the combined style information can then be used to create three-dimensional objects, showing style features similar to the newly designed windows.

1 Introduction

Any design system that is intended to create designs that follow a certain style requires knowledge about the style, given to the system represented either explicitly or implicitly in design data or coded into the design system. Collecting this knowledge is usually done by hand, for example by creating a shape grammar for a representative set of designs. Often, this also involves research about the work and methodology of the designer. In an earlier paper (Schnier & Gero 1996), we have proposed an alternative, machine learning approach. It uses the fact that it is possible to categorize style from its visible features (Chan 1995). The approach creates an implicit representation of style features, requiring only a set of sample designs. The acquired knowledge can be used directly to create new designs that show style similarities to the example designs, but at the same time are adapted towards different design conditions. In this paper, we show how manipulating the learned style knowledge can transform it, allowing the creation of a much wider, possibly more 'interesting' set of designs.

2 Learning Style Feature using Genetic Engineering

The approach used to learn style knowledge is based on evolutionary systems. Evolutionary systems are population-based search algorithms. The population consists of individuals, represented by their genetic code, the 'genotype'. A transformation exists that transforms the genotypes into 'phenotypes', and a measure for the performance of individual phenotypes, the 'fitness' can be calculated. New individuals are created using genetic operations from genetic material from one (mutation) or two (cross-over) genotypes with high fitness; genotypes with low fitness are removed.

To learn style features, the system is programmed to try to create copies of the example or examples given to it; the fitness function is a measure of the distance between the current phenotype and the example design. At the same time, particularly successful combinations of genes in the genotypes are identified and encapsulated into 'evolved' genes. As a result of this 'genetic engineering', the representation evolves, and the search space is transformed in a way that the search is more and more biased towards designs similar to the examples.

To create new designs a second evolutionary system is run with a fitness function evaluating the phenotypes with respect to the design criteria. This evolutionary system uses the evolved representation that has been created in the first step, thus incorporating the knowledge acquired in this step into the new designs. The evolved genes encapsulate sets of basic genes, protecting them from the genetic operations. This is relatively easy if the evolved genes contain only sequences of directly successive genes. If however the evolved genes are allowed to consist of non-successive basic genes (complex evolved genes), the genetic operations can lead to situations where conflicts between different evolved genes arise. A solution, using diploid genetic code with dominant and recessive genes, has been described in Schnier & Gero (1997).

3 Transforming Evolved Representations

Using the evolved representation, it is possible to create new individuals that show similarities with the examples. However, while interesting, the results would usually not be called 'creative'. To make the new designs more 'interesting' and 'surprising' two operations are introduced: combining elements from different sources, and transforming elements into a different domain. Thanks to the flexibility of evolutionary systems, both operations are possible with the evolved representation. It is interesting to contrast this approach with that of Knight (1994). While Knight uses shape grammars as an analytical tool to describe the transformation of styles, we use transformations of style knowledge, represented in an evolved representation, as a generative tool.

3.1 Combining Genetic Material

In nature it is sometimes possible to combine the genetic material from two individuals from two different 'groups' of animals; the resulting offspring includes features from both groups. The most common example is probably cross-breeding between different races, for example in dogs; the results are generally referred to as 'hybrids'. In general, this cross-breeding is not possible, in fact 'species' are defined by the fact that they cannot interbreed. For a successful combination it seems that three conditions have to be met.

- The genetic material of both 'parents' has to be such that the same genotype-phenotype transformation can be used to transform it into a living individual. For life on earth, this is rarely a problem, since the vast majority of life forms use the same universal RNA/DNA-based genetic material.
- The environment in which the 'transformation engine' works has to be compatible. For example, mixing dog breeds of different sizes is generally only successful if the female dog belong to the larger breed, otherwise its womb might not be able to support the developing puppy.
- The genetic materials have to be compatible. In other words, the transformation engine has to be able to transform a genotype consisting of material from both sources into a functioning individual. This, in nature, is the most important obstacle in interbreeding.

While biological systems use a common representation and achieve a huge variety of organisms by a highly interactive multi-level development process, most evolutionary systems use a specialized representation, with a simple, usually linear genotype-phenotype transformation. The three conditions therefore have very different importance for evolutionary systems.

- Contrary to biological systems, evolutionary systems use many different genotype-phenotype transformations, often designed for specific applications. As a result, this condition prevents 'interbreeding' in most cases.
- In the vast majority of evolutionary systems implementations, the genotype-phenotype transformation is very simple, without any interaction with the environment. This point is therefore usually unimportant for evolutionary systems.
- To make the evolutionary search as efficient as possible, the genotype-phenotype transformation is usually designed so that most or all of the possible genotypes can be transformed into phenotypes. If all the genetic material used to create a new individual comes from genotypes that use the same genotype-phenotype transformation, the offspring is equally likely to be a valid individual. However, this does not guarantee that the offspring will have a high fitness.

In evolutionary systems, the most important condition is therefore that the sources of the genetic material are systems that use the same genotype-phenotype transformation.

What to combine?

If, as defined in the previous section, the sources for the genetic material use the same genotype-phenotype translation, they also use the same basic representation. As a result, an initial, random population will look similar in any of the sources. Features specific to an application are only present in the form of certain gene configurations in individuals in later stages of the evolutionary process. The combination of genetic material from different sources is therefore only possible by combining individuals, for example with a cross-over operation. The resulting individual will show features of both individuals, and therefore both sources; however in following generations the genetic operations can destroy any such features.

Evolving representations present an alternative: features of an application are integrated into the representation and, while all applications have the same basic representation, they can have very different evolved representations. As a result, random individuals created from different evolved representations will look different and if the evolved representations are combined, the random initial individuals will show features from all the sources. During the evolutionary process the evolved features are protected: while it is still possible that the evolutionary process leads to individuals that use only evolved genes, and therefore features, from one source, the genetic operations cannot disrupt the evolved genes. This is similar to the case in nature: the 'basic coding' are the base pairs on the DNA (or any variant of RNA), but the units of inheritance are long sequences of base pairs, the genes.

3.2 Extending Genetic Material

Since in evolutionary systems genes and the genotype-phenotype transformation are just data and programs, they can be adapted as required. This allows the use of genetic material in different domains from where it originally was produced. To allow the genotype-phenotype transformation to use the 'foreign' genetic material, we can either modify the genetic material so that it fits into the new domain, or we can modify the transformation so that it is able to directly use the material from a different domain. The same procedures allow us to combine genetic material from two different domains.

4 Example

As an example, we show how style features learned from Mondrian paintings can be combined with style features from Frank Lloyd Wright window designs. For both the paintings and the window design, we use the same basic representation, combining the evolved genes therefore corresponds to cross-breeding between two breeds. To show how a representation can be transformed for a different domain, we use the evolved genes created in the two two-dimensional example applications to create three-dimensional objects.

4.1 Basic Representation

To represent Mondrian paintings a tree-coding is used where every node of a tree corresponds to a division of a rectangle into two smaller rectangles. The position and direction of the division, the thickness of the dividing line, and the colour of one of the two resulting rectangles are encoded in four variables at every node. Every node can also have two subtrees that describe further subdivisions of the rectangles, Figure 1 shows an example[1].

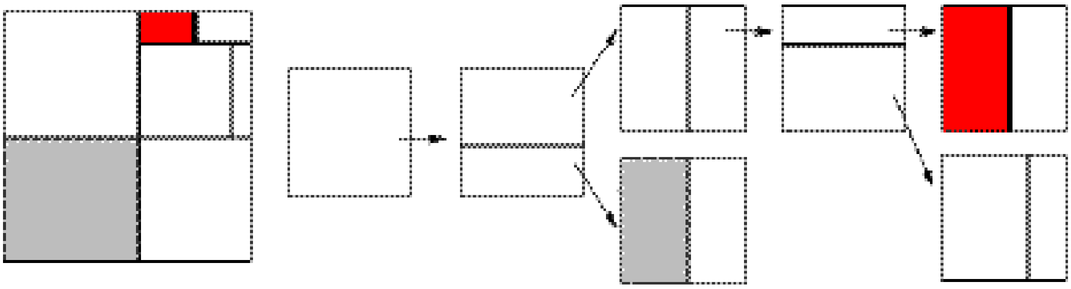

Figure 1: Representation of a Mondrian painting as tree-structured series of rectangle divisions.

This representation allows the creation of a large set of rectangle-based two-dimensional designs. Some additional designs, including for example the 'pinwheel' shapes used in paintings by Vantongerloo (Knight 1989), can be represented as well if 'invisible' lines are allowed, as shown in Figure 2. The invisible line, which can be located anywhere in the painting as long as it intersects the middle rectangle, splits the design into two halves that can be represented.

[1] For colour versions of this and other figures in this paper, please refer to http://www.arch.usyd.edu.au/~thorsten/publications/acdm98.html or contact the authors.

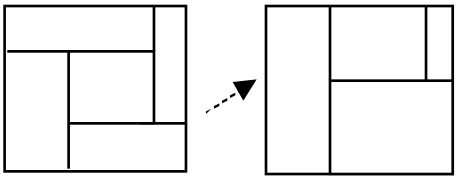

Figure 2: (a) Representing a pin-wheel figure by dividing it using an 'invisible' line, (b) the lower half, which can be represented in the basic representation without problems.

4.2 Learning Representations

Figure 3 shows the examples used to generate an evolved representation based on Mondrian paintings. The paintings have eight (Figures 3(a) and (c) and seven (Figure (b)) areas, therefore a tree with eight nodes is required to completely represent a single painting. At each node, the evolutionary system can choose between 4 different positions (top, bottom, left, right), 15 fractions, 4 line-widths and 12 colours.

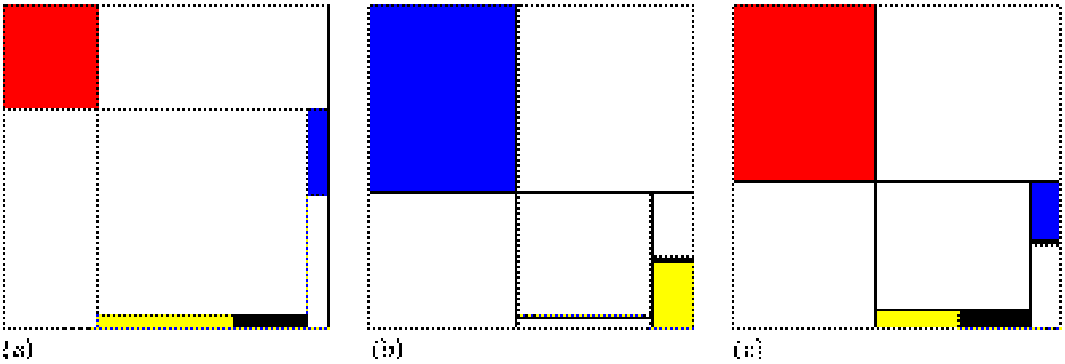

Figure 3: Mondrian paintings used to create evolved representation.

For each phenotype a Pareto fitness vector with fifteen elements is calculated. For each of the three examples five fitness values describe how 'close' the phenotype is to the example in terms of positions of the divisions, correctness of colours and line-widths, completeness and absence of additional divisions.

The gene extraction method is the same as described in Schnier & Gero (1996). However, the gene fitness is calculated differently: in earlier systems, a cumulative fitness value was derived directly from the Pareto fitness vector to calculate the gene fitness. During the evolutionary process individuals with high fitness compared to the current population will still have some wrong colours, line-widths, etc. As a result, it is possible that genes are generated that incorporate these incorrect features. To prevent this, a function has been added that checks if the phenotype is a true subset of the example, ie. if it is possible to convert the phenotype into the example by adding further divisions. Gene combinations that occur in at least one phenotype that is a true subset of at least one example are guaranteed not to have any incorrect features, only those are therefore considered as new evolved genes.

The run produced 110 evolved genes, the first and the last seven genes created are shown in Figure 4.

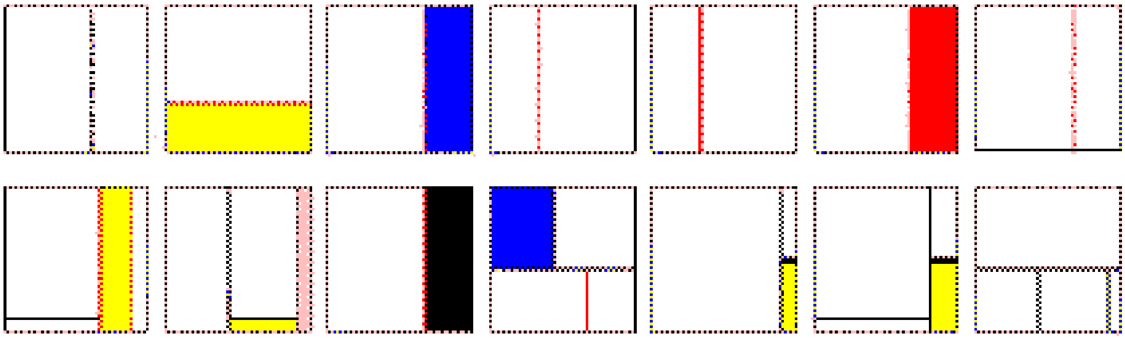

Figure 4: Evolved genes created from the examples in Figure 3. To represent genes with incomplete nodes the following convention was used: no colour: light pink; no line-width : line-width 2; no direction: vertical, and line stippled; no fraction: 1/3, and line coloured red.

As the second example, a window design, created by Frank Lloyd Wright for Hollyhock House (Hanks 1989) was used. Three segments from the centre of the window were coded. Figure 5(a) shows the drawing used to generate the evolved representation. In the original, the light blue rectangles have a wider frame, this had to be changed because it could not be represented in the basic representation. To compare the example with phenotypes, the outer rectangle of the window is transformed into a square, see Figure 5(b). Due to the higher complexity of this example, the system learned more evolved genes, 159. The last 11 evolved genes created are shown in Figure 5(c).

4.3 From Mondrian to Frank Lloyd Wright

In order to make use of evolved representations a set of initial individuals is created using the representation and then an evolutionary system is run with a fitness describing the desired new design. Since the evolved representations from the paintings and the window design use the same basic representations, they can simply be combined by creating a random initial population using evolved genes from both sets. However, it is also possible to add additional control over the way the two representations are used. For example, it is possible to remove some evolved genes from the representations before they are combined. Another possibility is to remove parts from the genes before they are combined. To maximize the influence of the evolved representation the initial individuals are generated so that they do not contain any basic genes. To achieve this the system makes use of a diploid representation, described in more detail in Schnier & Gero (1997)

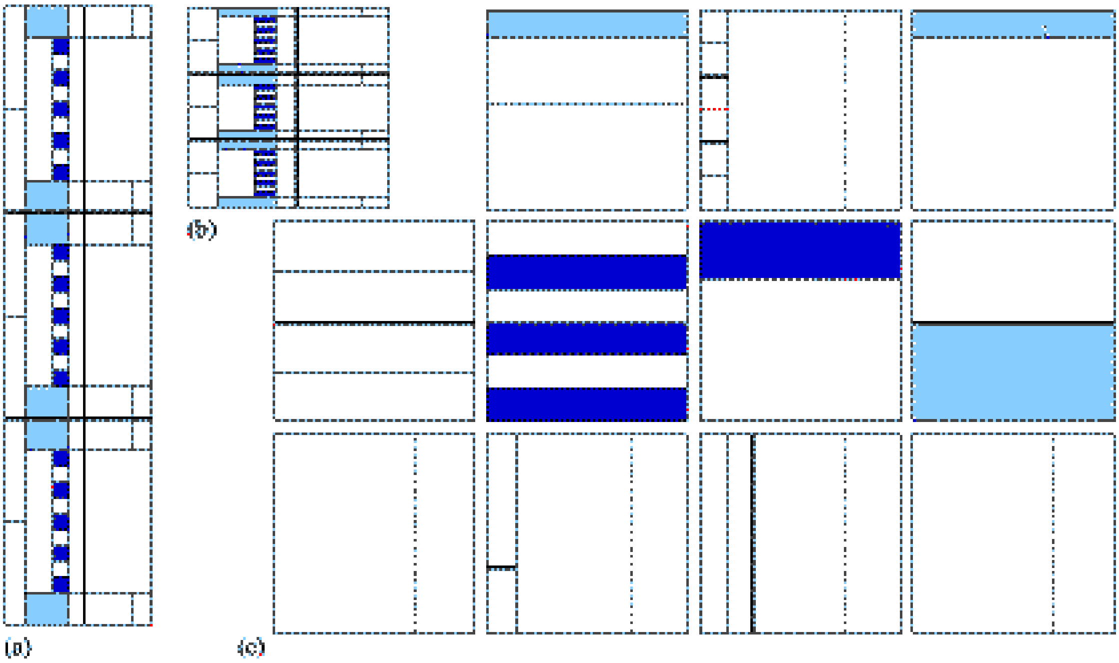

Figure 5: Evolved representation from Frank Lloyd window: (a) part of the design used as example, (b) the design as seen by the evolutionary system, (c) the last 11 evolved genes.

The fitness function used in this example specifies that the new designs have to have a vertical line in the middle of the frame, 22 panes, and that the sides of all panes are longer than 7% of the size of the outer square. The fitness function is intentionally chosen in a way that it neutral to the use of the style features, it neither prevents nor promotes them. In fact, the influence of the evolved representations can easily be seen in the initial, random individuals as well. The fitness function is not very difficult, the system usually finds perfect phenotypes within a few minutes.

For comparison, Figure 6(a) show new designs using only the basic representation and Figures 6(b) and 6(c) show the results using the representations created from only the Mondrian paintings and from only the Frank Lloyd Wright window respectively. Crossover and mutation were able to adapt the designs quickly to the new conditions, while the style information has mostly been preserved through the evolution. Mutation was implemented so that in rare occasions, it would replace an evolved gene with basic genes, this can be noticed in a few places that contain features that were not part of any of the examples.

Figures 6(d) and 6(e) shows the results of combining the evolved representations created from the paintings and the window in two different ways. To create the results in Figures 6(d) the initial population was created choosing random evolved genes from both sources. Features in the designs can easily be identified corresponding to either one of the paintings or to the window. The initial individuals and the final designs contain more genes from the windows than from the paintings, this reflects the fact that more evolved genes from the window design were available.

To create the designs in Figure 6(e) the evolved representations were modified before they were used to create initial individuals: in all evolved genes created from the Mondrian paintings, information about position and fraction of the rectangle division was removed. Similarly, in all evolved genes from the window design, all information about colour and line thickness was removed. The resulting genes were then used to create the initial individuals. The results show, as expected, topological features inherited from the Frank Lloyd Wright window design, and colouring and line thicknesses from the Mondrian paintings. The imbalance in the number of evolved genes does not matter in this case, since the different sets of evolved genes are responsible for different aspects of the new designs.

Figure 7 shows how the fourth design in Figure 6(e) can be assembled into a complete window, similar to the original Frank Lloyd Wright design.

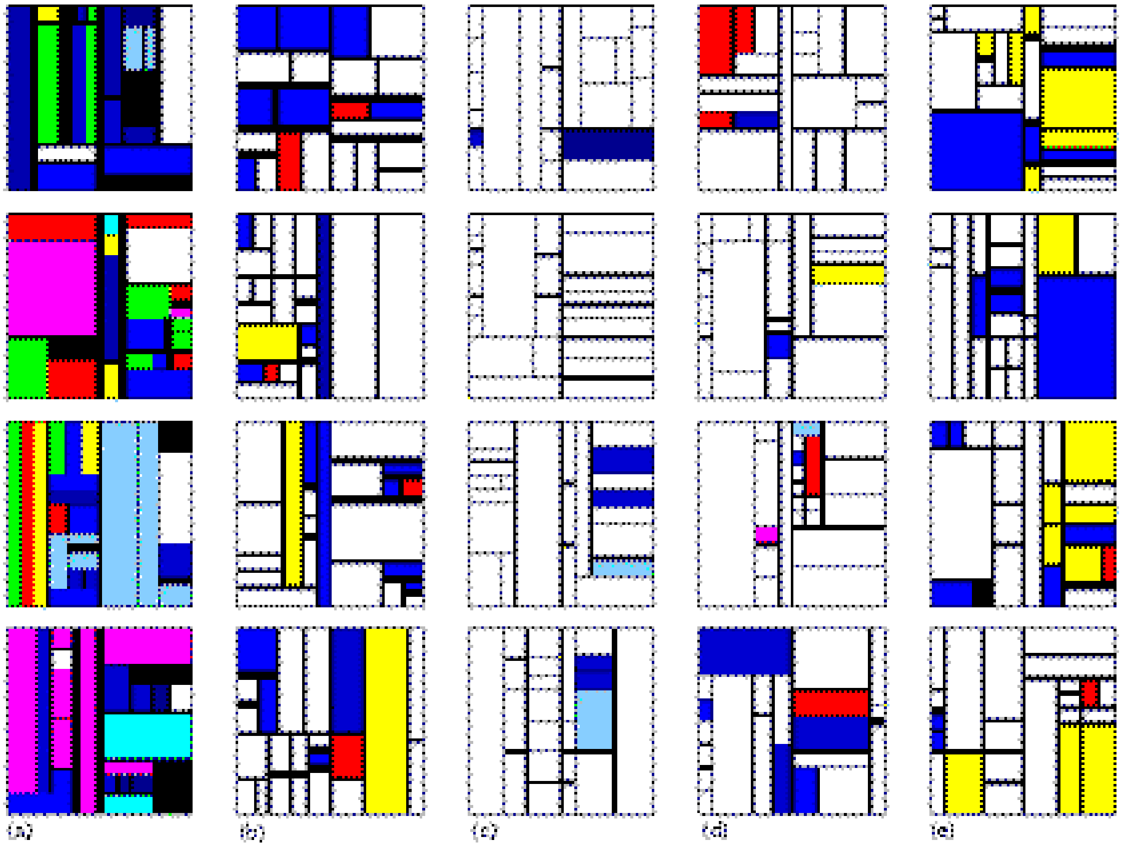

Figure 6: New designs: (a) without evolved representation, (b) using evolved representation from Mondrian paintings, (c) using evolved representation from Frank Lloyd Wright window, (d) using full genes from both representations, (e) using topology from window design and colour and line-width information from paintings.

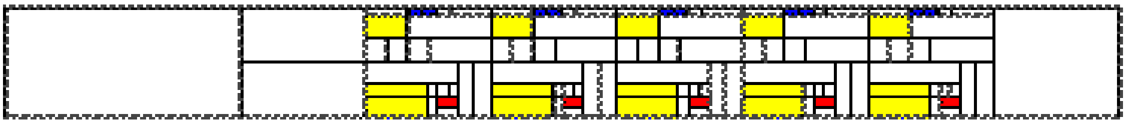

Figure 7: One of the designs in Figure 6(e), assembled into a full window (shown rotated).

4.4 Creating 3-dimensional Objects from 2-dimensional Examples

As mentioned in Section 3.2, it is possible to use evolved representations in domains different from where they were initially created, if either the genotype-phenotype transformation or the evolved representations are adapted for this purpose. The new domain used in this example is the creation of coloured cubes. To create the three-dimensional objects a basic representation is used that is similar to the one used for the paintings. However, instead of rectangles divided by lines, the nodes specify planes intersecting cubes. Each node has an additional variable specifying whether the intersecting plane is perpendicular to the x-y plane, to the y-z plane or to the z-x plane. In other words, to divide a cube, its projection into one of those three planes is used and the resulting two-dimensional shape then cut as in the Mondrian painting.

Due to the similar nature of the representations, it is easy to adapt the evolved genes created from the paintings and the window. Obviously, none of the evolved genes provides a value for the cutting plane. One possibility is to assemble initial individuals without this value and then provide a random value for this at every node. However, this makes the topology features learned from the window less recognizable in the phenotype. The reason is that genes specifying topological features span a number of connected nodes and are only recognizable in the resulting phenotype if they are all perpendicular to the same plane. For example, a series of parallel divisions only remains parallel if all divisions are perpendicular to the same plane, in other words, have the same value for the added variable. Because of the random assignment of this value, this is not the case.

Better results can be produced by adapting the evolved genes before use for the three-dimensional application. For every gene, a value for the intersection plane is randomly chosen and added to every node of that gene. For the example, all topology features were removed from the evolved genes produced form the example paintings and all colour and line-width information removed from the evolved genes created from the window. Then, information about the intersection plane was added to all window-genes in the manner described above. The resulting genes were then used to create the initial population. The fitness function is similar to the one used to create new window designs: 70 sub-cubes, a plane intersecting the resulting cube in the middle, and no edges shorter than 7% of the edge length of the outer cube.

The result is shown in Figure 8(a). As a comparison, Figure 8(b) shows a cube created with the same fitness function and genes, but assigning the planes randomly to the individual nodes. While similarities in colouring and line-width to the Mondrian paintings exist in both cubes, the cube in Figure 8(a) shows more topological features relating to the Frank Lloyd window example.

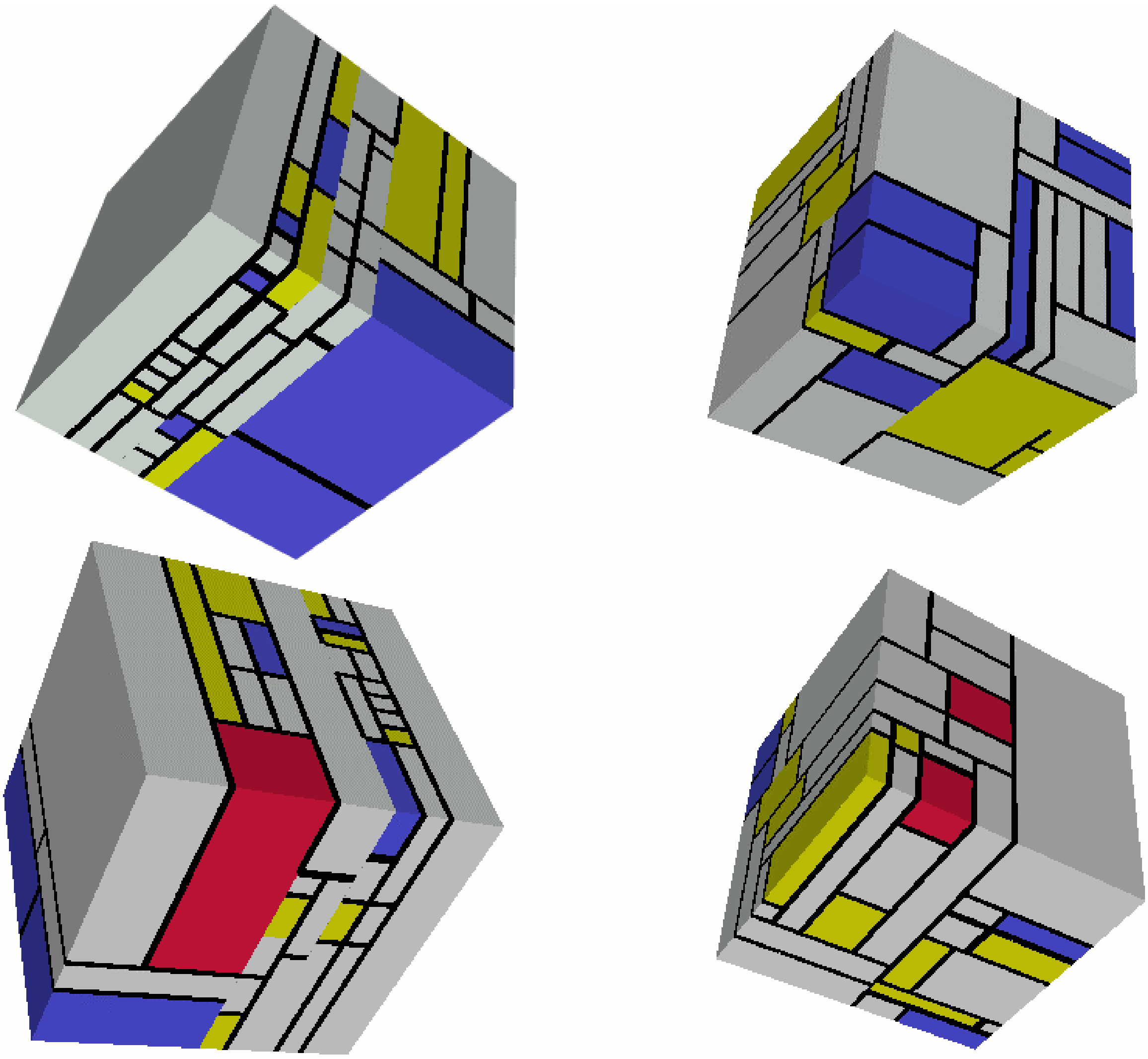

Figure 8: Three-dimensional objects created from 2-d representation, views show opposing corners: (a) plane information added to evolved genes, (b) plane information added to nodes.

5 Conclusions

This work shows that it is possible to use evolutionary systems to produce designs that are 'hybrids', incorporating different styles, by using a mechanism similar to crossbreeding between different races in nature. However, the designer using this computational process has far more control over the mixing process, allowing the inclusion and exclusion of specific features from either of the sources into the new designs. Compared with crossbreeding in nature, it also adds the ability to combine features from different domains, somewhat the equivalent of crossbreeding between different species.

Without evolved representations it still is possible to create single hybrid individuals, for example by using a cross-over between two individuals from different, highly adapted populations. However, the mixing is restricted by the genetic operation used to combine the parents, and the individual features are not protected during further genetic operations. Using evolved representations, it is possible to create high numbers of random initial individuals that show the different style features in a probabilistic distribution. These features are protected through the course of the evolution.

Another way to conceive of this work is to relate it to analogy where the designer draws ideas from a source design and introduces them into the target design (Coyne at al 1990). Of the evolved representations one may be considered the target and the other the source. Whereas in analogically-based design there is considerable difficulty in reformulating the ideas from the source design so that they can be introduced into the target design, there is no such problem in this approach. This work may be considered in terms of creative design processes, ie computational processes which are potentially capable of producing "creative" designs. One computational model of creative design involves the concept of processes which are capable of changing the state space of possible designs (Gero 1994). The combination of two evolved representations fits that model well and is clearly capable of producing novel designs which lie outside the space of designs which could possibly be produced using only either of the evolved representations, implying that the new state space is not simply the union of the previous two state spaces.

To extend this work it would be interesting to create and mix evolved representations from different instances in the same domain used for the examples and from other, different domains. Especially if more complex genotype-phenotype representations are used, together with complex, possibly multi-level evolved genes, the results promise to be very interesting.

6 Acknowledgments

This work is supported by a grant from the Australian Research Council and by a University of Sydney Postgraduate Research Award. Computing resources have been provided by the Key Centre of Design Computing.

References

Chan, C.-S. (1995). A cognitive theory of style, *Environment and Planning B: Planning and Design* **22**: 461-47.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M. & Gero, J. S. (1990). *Knowledge-Based Design Systems,* Addison-Wesley, Reading, Massachusetts.

Gero, J. S. (1994). Computational models of creative design processes, *in* T. Dartnall (ed.), *AI and Creativity*, Kluwer, Dordrecht, pp. 269-281.

Hanks, D. A. (1989). *Frank Lloyd Wright: Preserving an Architectural Heritage*, Studio Vista, London.

Knight, T. W. (1989). Transformations of De Stijl art: the paintings of Georges Vantongerloo and Fritz Glarner, *Environment and Planning B* **16**: 51-98.

Knight, T. W. (1994). *Transformations in Design: a Formal Approach to Stylistic Change and Innovation in the Visual Arts*, Cambridge University Press, Cambridge.

Schnier, T. & Gero, J. S. (1996). Learning genetic representations as alternative to hand-coded shape grammars, *in* J. S. Gero & F. Sudweeks (eds), *Artificial Intelligence in Design '96,* Kluwer, Dordrecht, pp. 39-57.

Schnier, T. & Gero, J. S. (1997). Dominant and recessive genes in evolutionary systems applied to spatial reasoning, *Tenth Australian Joint Conference on Artificial Intelligence* (to be published).

This paper is a copy of: Schnier, T. and Gero, J. S. (1998). From Frank Lloyd Wright to Mondrian: Transforming evolving representations, *in* I. Parmee (ed.), *Adaptive Computing in Design and Manufacture,* Springer, London, pp. 207-219.