Simple Computer Programming to Enhance Spatial Cognition in Architecture and Design

M.C. Burry[1], J.R. Burry[1], G.M. Dunlop[1], A. Maher[1] & S. Anson[2]

[1]Spatial Information Architecture Laboratory
School of Architecture + Design
RMIT University. Melbourne, Australia
Email: mburry@rmit.edu.au; jane.burry@rmit.edu.au
grant.dunlop@rmit.edu.au; andrew.maher@rmit.edu.au
[2]School of Architecture and Building
Deakin University. Geeong, Australia
Email: sima@deakin.edu.au

Presented at SIRC 2001 – The 13th Annual Colloquium of the Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
December 2nd-5th 2001

ABSTRACT

Descriptive geometry and mathematics have had a progressively diminishing presence in architecture and related spatial design education since the late nineteenth century. Despite recent advances in automated manufacturing, virtual prototyping, and a more fluid ability in three-dimensional form making and spatial manipulation in disciplines such as mechanical engineering and product design amongst others, Cartesian measurement is still the principal descriptive approach for constructing buildings.

Without delving into the historical reasons for this separation, or the implications of the haptic intuitive processes facilitated by proprietary software in this post-digital era, the central question posed here is whether the introduction of simple computer programming with a tangible link to figurative space is a way to reintroduce mathematics as a basis for spatial description. How can this process extend the understanding of topology, Cartesian space and morphology?

This paper describes two projects in which morphology and programmatic form finding rather than direct manipulation were used to introduce the spatial opportunities of programming. Here, process, or how the program works, is of greater significance than outcome, or what it does.

In the first of these two parallel assignments students are given a bounded figurative conceptual space comprising three Cartesian axes with a cube at the centre. Six Phileban solids mark the axial limits to the space. Any point in this space represents a hybrid of one, two or three transformations from the central cube towards the various Phileban solids. The first stage is to attempt the prediction of the topological and morphological outcomes of the transforming operations. Subsequently, morphogenesis and hybridization is put into practice through programming and the broad learning opportunities are described below.

The second assignment presents a similarly Cartesian space, this time unbounded with a three-dimensional figure, a column, having its base in the XY plane and its central axis lying on the positive z-axis. It has four similar hyperbolic parabolic surfaces, each with four well-defined vertices and edges, orientated at 90-degree intervals around the column axis. Below the column on the XY plane there are four planar quadrilaterals, each one a progenitor for a four-sided hyperbolic paraboloid warped surface forming part of the column. In this instance it is the movement and morphosis of the rectangular planar figures into the doubly curved ruled surfaces that is explored through programming.

Keywords and phrases**:** architecture, design, geometry, form, information technology, spatial information, topology

1.0 FRAMEWORKS

"To keep this clear, I shall no longer speak of the "real" world but of a "construction world" that encompasses everything that may need to be depicted and reasoned about in the exploration of a particular design." William J Mitchell, 1990

In both the projects detailed below, the problem is clearly framed, the limits, characteristics and constituents of the sphere of operations clearly defined. Mitchell goes on to explore "the function or partial function *D* mapping from physical objects in the construction world to shapes in the design world." Similarly, a depiction of a design world in either of 'our worlds' detailed below is the value of a many-to-one mapping function from shapes and states to the limited geometrical figures and relations of the *Worlds of Discourse* represented in these projects. At the end of the exercise the students have acquired the skills and confidence to investigate the one-to-many mapping from these worlds to a broader design world.

2.0 OUR (PHILEBAN[1]) WORLD

The domain named 'Our World' is a rigorously legislated figurative spatial world, employed to promote the conceptual understanding of a space used to marshal ideas rather than represent occupancy. (Fig. 1)

At the centre of this world is the cube. The limits of the universe also describe a cube. Where the two ends of each of the three Cartesian axes meet the boundaries of this cubic universe, are located six other Phileban forms, each the result of a particular operation on the cube (operation '1'). At other intermediate positions on the axial planes within this space hybrid forms, the result of two transformations (operations '1' and '2') are located. All other Cartesian locations within the space have hybrids resulting from three transformations (operations '1', '2', & '3') associated with them.

This construct exists simply to aid recognition of some possibilities for morphological change. The operation 1→2→3, is a statement of three sequential transformations on the original cube. Using an interactive program, the user has taken the original form partly towards a sphere, then partly towards a wedge, and finished on a path beyond a pyramid.

Participants are first asked to predict the outcome of the three operations. The exercise aims to provide new insights into topology, Cartesian space, morphology, morphogenesis and hybridization. It also introduces the power of this level of computer-user interaction when compared to the simple one-off and direct spatial manipulation, with which most architects, designers and students of these disciplines are familiar. As designers, they experience a completely new and different relationship with the computer. It is a relationship redolent with opportunity for experimentation and innovative form seeking even at this rather trivial level (in terms of actual programming).

---

1. Phileban forms refer to those discussed in the dialogue between Philebus, Socrates and Protarchus by Plato and which may be found in *Philebus*. These forms include the 'primitives' or 'plane solid figures which are formed out of them by turning-lathes and rulers and measurers of angles', such as the cube, sphere, cone and cylinder. (*Philebus*, Plato, Translator Robin H. Waterfield, London: Penguin Books, 1995).

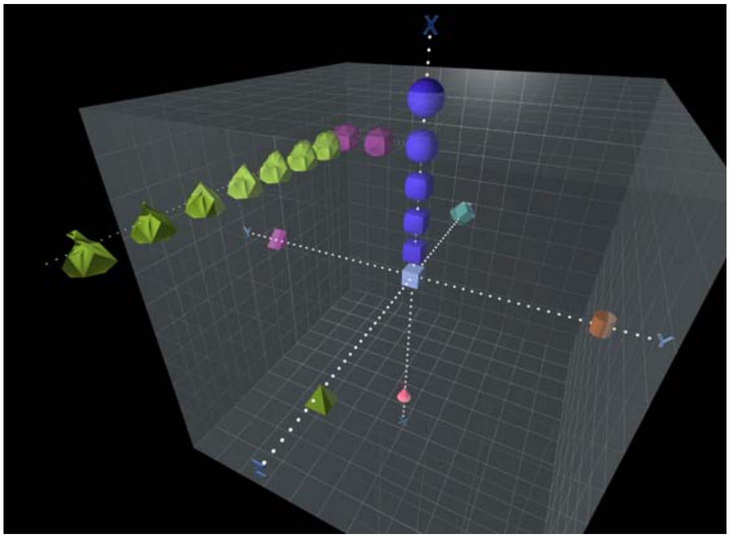


Figure 1: 'Our World', the programming assignment required that a central form (in this case a cube) undergo a series of transformations. As an interactive program, the user can take the original form and steer it partly towards a sphere (in the example above), then partly towards a wedge, and finishing on a path beyond the condition of a pyramid.

The first planning exercise, undertaken by all the students, is to develop the iterative route from cube to sphere. (Fig. 2). This is used to develop a common topological approach to manage a smooth transition from one form to the other. All the other transformations are undertaken as individual solutions. In other words, one person undertakes the transformation of the cube to a pyramid, another to a cone etc. The individual algorithms can be combined subsequently to give the 1-.2-.3 sequencing, which results in any one of a three dimensional matrix of 48 unique combinations or hybrid forms. For this to happen, all individual outputs must be compatible. The transformations must occur iteratively, the number of iterations at the discretion of the program user (rather than designer). To this extent the issue of interface is also part of the brief. Each iteration or morphological change event outputs a file used finally as one of a series of frames in an animation.

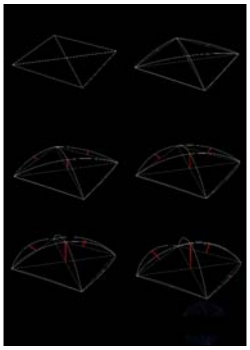


Figure 2: sequence of changes to the twenty-four surfaces that describe both the original cube and the sphere showing the cube tending towards the sphere.

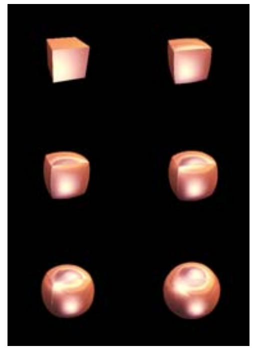


Figure 3: sequence of changes to the twenty-four surfaces that describe both the original cube and the sphere showing the cube tending towards the sphere. The imperfections to the surface of the sphere are deficiencies in the surface algorithms in the chosen software.

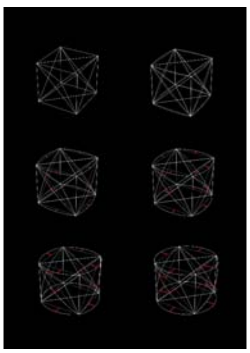


Figure 4: sequence of changes to the twenty-four surfaces that describe both the original cube and the cylinder (see figure 1). This figure shows the cube tending towards the cylinder.

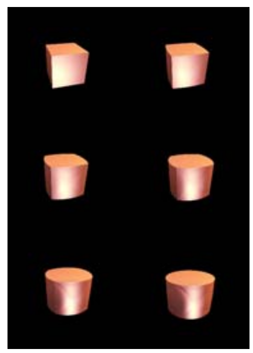


Figure 5: sequence of changes to the twenty-four surfaces that describe both the original cube and the cylinder (see figure 1). This figure shows the cube tending towards the cylinder. The imperfections to the surface of the cylinder are deficiencies in the surface algorithms in the chosen software.

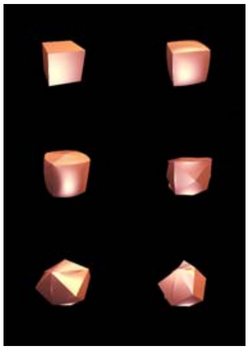


Figure 6: sequence of changes to the twenty-four surfaces that describe both the original cube and a hybrid form (see figure 1 steps '1' '2' and '3'). This figure shows the cube tending towards first a sphere (partway) then partway towards a pyramid and finally a rhomboid.

The cube is the figure with the most parameters, the sphere, the least: six faces versus one single spherical surface. Conceptually a common understanding is required. Points and lines have names conforming to an agreed shared notation to ensure that each algorithm operates with common symbols and is therefore inter-operative. The face of the cube is divided into four triangles (figure 2): these triangles are essentially the foundation class for the project. The program needs to convert the linear curves that bound the triangular cube face components into non-linear curves - in the case of the sphere these curves will be sector arcs. Also, in the case of the sphere, the points at the corners of the cube correspond with their equivalents for the sphere: they are coincident. Therefore a point that describes a mid-point for each curve moves taking the curve with it with the result that an associated surface transforms from a plane to a sector of a sphere. The result is shown in figure 3, and an equivalent transformation from cube to cylinder is shown in figures 4 and 5.

Regardless of the mathematical simplicity, there are many unfamiliar aspects for architects in this procedure. It serves to kindle interest and subsequently more complex problems can be approached with confidence. Reducing a cube to a pyramid, for instance, leaves the problem of what to do with the top face that reduces to nothing while the change from a cube to a cone has even more philosophical and practical difficulties. Here the algorithm reduces the top face of the cube to a point that is the apex to the cone (as for the pyramid). As its area diminishes, the base transforms from a square to a circle. The intermediate iterations show a rounding square at the base with a perfect reducing square at top until the apex of the cone is reached. This is an aesthetic problem, and the programmer is obliged to include a proportional system to compensate for an otherwise inconvenient visual effect.

3.0 OUR (RULED-SURFACE) WORLD

This is a world, despite its title, in which, relative to the (Phileban) world detailed above, the rules have been relaxed. At the centre of this (figurative spatial) world is a wire frame column with four similar orthogonally placed hyperbolic paraboloid surfaces in its composition. The base of the column is centred on the Origin of a Cartesian system that extends infinitely and there are four square 2-dimensional figures located in the four quadrants of the XY plane. The vertices of the squares are joined by four edge lines. Parallel to these edges, an array of straight lines of equal length act as the directrices of the hyperbolic paraboloids. These lines, which do not remain parallel or of equal length but retain their topological identity as straight lines linking points on two opposite edges of the four sided figure. During the transformation and as the vertices and edges move out of their common plane, the square face transforms into a hyperbolic parabolic surface.

The boundaries of this world are temporal. There is a starting state (given) and a final or completion state (implied). The endpoint is a complete and surfaced column where each of the four square figures has undergone transformation and translation into one of the hyperbolic parabolic surfaces making the form of the column.

Figure 7 shows the direct linear translation of each of the four vertices of the square figure to each of the four vertices of the hyperbolic paraboloid via a series of iterations. A simple program to bring about this progressive transformation is developed collectively as an introductory exercise.

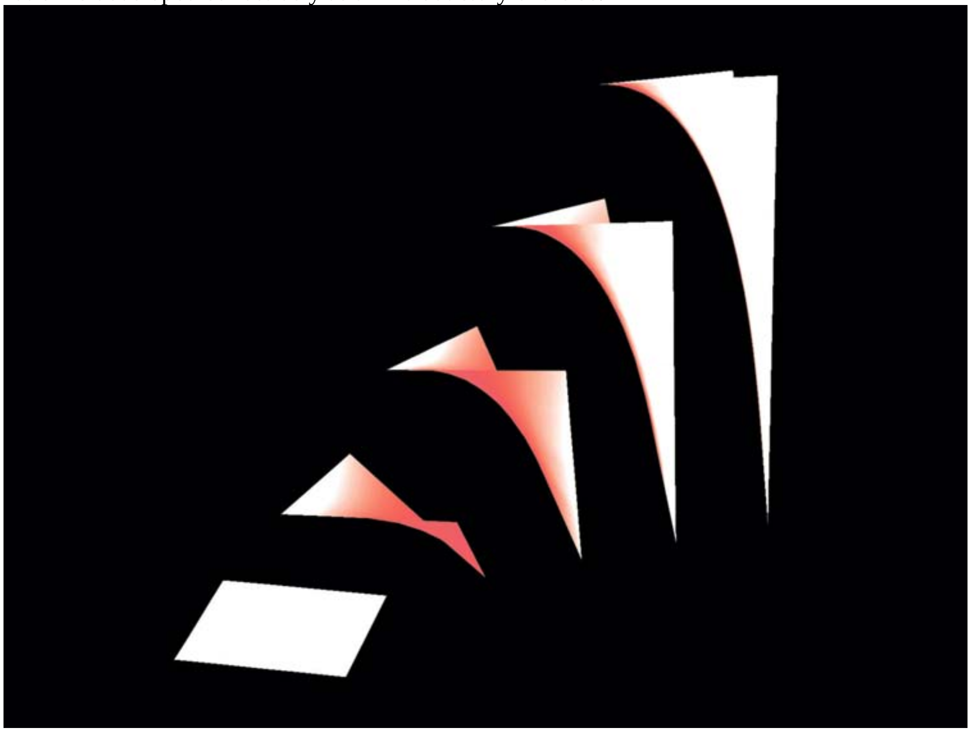


Fig. 7: The direct linear translation of each of the four vertices of the square figure to each of the four vertices of the hyperbolic paraboloid via a series of iterations.

Individually, students then manipulate the sequence, pathways, and intermediate steps in this transformation. Thus while the geometrical starting and endpoints remain the same the full domain of possibilities for the intermediate geometry, relative time sequence, surface characteristics, viewpoint of observer etc may be explored resulting in a many different narratives based on the same underlying script. (Fig. 9)

In this way programming introduces the possibility of translating an idea about form and change into a figurative manifestation in a very controllable and systemic way. As with the cube transformations, the power of the program becomes clear when participants attempt to predict the formal results of an idea, then realize it through their manipulation of the program as generator rather than familiar one-off manipulation of the form itself. Examples are ascribing a particular geometry to the path described by a vertex in the progression from square to hyperbolic paraboloid or introducing a time difference or iterative change difference in the progression of one vertex compared to another.

Fig. 8: The built columns composed of hyperbolic paraboloids on which the project is based, Sagrada Família Church, Barcelona, Antoni Gaudí.

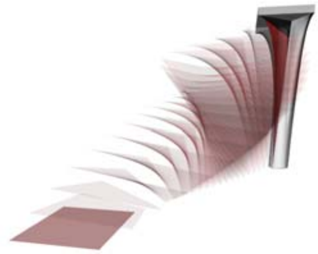


Fig. 9: A series of hyperbolic paraboloid surfaces in the progression from the planar quadrilateral to the surface on the column and shown as transparencies.

4.0 BEYOND 'OUR WORLD'

The fundamentals of programming have now been assimilated, their close association with geometry and the figurative world overcoming some of the potential difficulties that the highly abstracted world of computer code can present to a group with a strong visual/spatial bent but limited mathematical background. Similarly, Cartesian space, convenient for its simple abstraction to 2-dimensional representation and therefore not necessarily well understood even by students of spatial design, is now well grasped. They have also learnt about the complexities of formal problem solving using both logic and mathematics. In the realm of aesthetics, however, they are challenged with quite unfamiliar circumstances: semiautomatic design. Most students can predict the effects of $1 \rightarrow 2$ transformations, if not mentally they are at least able to derive the outcome through sketching. Predicting the results of the third transformation step seems to be especially demanding, and the outcome from a fourth transformation step impossible to visualise it seems. These assignments operate within the boundaries of the 'worlds' defined for them. Within these boundaries the participants are able to act as agents for curious yet computationally (if not intellectually) predictable forms that are repeatable, and therefore of potential use in architectural or formal design.

The iterative computer program, however, creates the opportunity to run the program beyond the boundaries into new and extraordinary territory (Fig. 10). By producing more iterations than proportional steps or negative iterations, the topologically equivalent description generates self-intersecting forms fundamentally challenging spatial conception for the designers, who, in their mathematical adventure, glimpse worlds beyond the known and constructible. (Figs. 11 & 12)

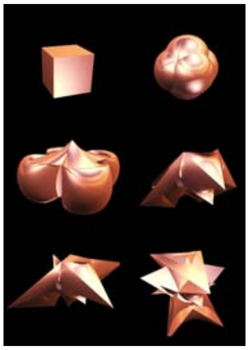
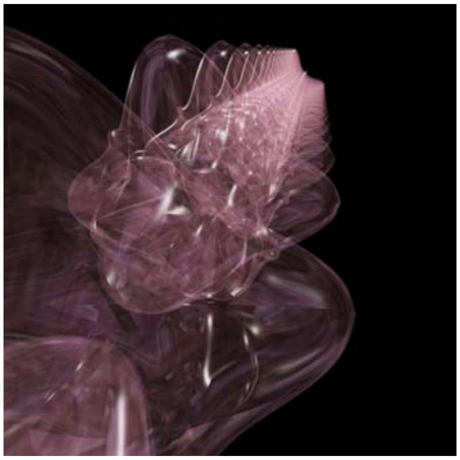


Figure 10: Too much hybrid vigour. The figure shows a sequence of changes that go beyond the boundaries to 'our world' (figure 1). The original cube has developed into a hybrid form (see figure 1 steps '1' '2' and '3') that distorts the original Phileban objects to guided but conceptually unpredictable outcomes.



Figures 11 & 12: Too much hybrid vigour. The designer can experiment with their algorithm and seek formal responses to given architectural problems. The debate centres on whether they have encapsulated a 'design process', or merely produced an effective 'tool' to aid compositional strategies.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the work of the students of SRC 421: Advanced Computer Applications, School of Architecture & Design, Deakin University, Australia; in particular Simon Anson and Andrew Maher.

REFERENCES

Burry, Mark. (2000). "Beyond Animation." In *AD: Architectural Design – Architecture and Animation*, edited by Bob Fear. London: James Wiley editions.

Fox, C. W. (1989). “Integrating computing into an Architectural Undergraduate program.” In *The Electronic Design Studio : Architectural Knowledge and Media in the computer era*. Malcolm McCullough, W.J. Mitchell and P. Purcell (eds). The MIT Press. pp. 377–386.

Frazer, John. (1995). *An Evolutionary Architecture*. London, U.K.: AA Publications.

Frew, R. S. (1989). “The organisation of CAD teaching in design schools.” In *The Electronic Design Studio : Architectural Knowledge and Media in the computer era*. Malcolm McCullough, W.J. Mitchell and P. Purcell (eds) .The MIT Press. pp. 387–392.

Knuth, D. E. (1968). *The art of computer programming*. Addison-Wesley Publishing Company. Reading, Mass.

Lynn, G. (1998). *Animate Form*. Princeton Architectural Press, New Jersey.

Maeda, J. (1999). *Design by Numbers*. MIT Press, Cambridge.

McCullough, M. (1997). *Abstracting Craft : The Practiced Digital Hand*. The MIT Press, Cambridge.

Mitchell, W. J. (1987). *The art of computer graphics programming : a structured introduction for architects and designers.*,Van Nostrand Reinhold, New York

Mitchell, William J., 1990. *The Logic of Architecture*. Cambridge, Mass.: MIT Press.

Streich, B. (1992). “Should We Integrate Programming Knowledge into the Architect's CAAD-Education?” in *CAAD Instruction : The New Teaching of an Architect? eCAADe Conference Proceedings*, Barcelona (Spain). pp. 399–406.

Weizenbaum, J. (c1976). *Computer power and human reason : from judgment to calculation*. W. H. Freeman, San Francisco.

Winograd, T. (c1986). *Understanding computers and cognition : a new foundation for design*. Ablex Pub. Corp., Norwood, N.J.

Winograd, T. (1996). *Bringing design to software*. ACM Press, New York, N.Y.