NSF Project # DMC-8609893
Knowledge Aided Architectural problem Solving and Design
Working Paper 12
June 1989

Transformational Design

Costas Terzides

Abstract

The use of dynamically executable transformations and their orchestration in time is discussed and explored as a design tool. The aim has been to accommodate the dynamic character of architectural design during its form searching stages. The transformation and reformation of architectural elements is executed in real time under the direction of a user, who, by controlling the rythm and the speed, orchestrates the compositional evolution of an architectural parti.

1. Introduction

An almost unique characteristic of Architecture is that it is both dynamic and static. It is dynamic when viewed as the design process which has its roots in historical precedents of culture and the arts and which manipulates entities which are typically of an elastic character. It becomes static when it has to freeze at a certain state so that it may be built. In other words, architecture is static when viewed through individual buildings. It is dynamic when these buildings are viewed as instances of a continuum which derives from the past and projects into the future.

The static nature of architecture is currently served well by the available commercial CAD systems. Whether it is for the drafting of 2-D construction drawings or for the production of 3-D renderings, these systems can only be used after an architectural design has been more or less finalized. To the best of my knowledge there is no CAD system which is capable of dealing with the architectural form while it is still soft and in transition.

Imagine a computer screen where all is in motion. A variety of elements, either of a pure form, such as the cube, or of a symbolic form signifying a memory of the past, are paraded and can be selected and/or operated by a user who enters commands through the mouse. As organizational structures are introduced, they are executed in a continuous mode and in a way such that all the in-between stages are transparent and become part of the compositional experience. Processes may be reversed individually or in groups, may be applied at different speeds, or they may be interrupted and redirected. The morphological evolution of centuries may thus be replayed with accuracy or may be extrapolated into directions which were missed by history. For example, a doric temple may be crossprogrammed with the Barcelona Pavillion, a path for which there is no known historical precedent.

When wire frame representations are used for solids, the computer power to execute multiple transformations of forms is today available even on the personal computers. What have yet to be fully explored are the necessary interfaces and their correspondence to the architectural semantics. The work presented here discusses an experiment in that direction. It was undertaken as a master level thesis project.

The key theme is the transformation of physical entities. When these entities are viewed as parts of an architectural parti, then the theme becomes the transformation of architectural forms. The transformations exploded go beyond the simple symmetry transformations of translation, rotation and scale. They include the transformation of one shape to another. These concepts and their implementation in an experimental CAD system are discussed by the remainder of this paper, which concludes with illustrations of the potential of the system as a design tool.

2. Influences

Transformations of forms have been used rather extensively in the classic Greek architecture, as well as in other periods, but they more dramatically manifest themselves in the work of contemporary designers such as Peter Eisenman. Eisenman regards the idea of transformation as a process of change between architectural types. He considers the history of Architecture as a dynamic continuum of type transformations.

> "...Transformation, while it did not necessarily suggest any ideal order, presumed that the significance of the final form resided, in part, in the process tself; in the capacity of the object to reveal its own origins and processes, to register back to an original type, by a kind of reverse mental process. It was hermetic and internalized."

Furthermore, in the work of Daniel Liebeskind, architectural elements such as beams and columns are placed in space in a way such that they appear as if they are *moving*. It seems that there are some relationships which determine the position of each object in such a way that the moving image can better be described as *a dancing image* (see Figure 1-1).

Fig 1-1: Leakage, silk screen paint, Daniel Libeskind, 1979

An implementation of transformational processes in a CAAD system has been reported by Yessios. The system was used in design exercises which were recorded on video tapes to fully reflect the dynamic essence of the explorations. Yessios's algorithms dealt with 2-D shapes. The work presented here is in essence extending these algorithms into the third dimension and attempts to generalize their semantics.

3. The implementation

A *transformation* is a process in which an object changes its form gradually in order to obtain another form. The operation of transformation consists basically of the selection of two objects and the assignment of *n*, the number of in-between steps. The first object then transforms into the second in *n* steps. This process is illustrated in Figures 3.1 and 3.2. The transformation preserves the structural integrity of the objects involved, that is, an object changes into another object as a single entity. However, in order to increase the possible ways an object can be transformed, the user is allowed to create his/her own transformations by matching pairs of faces, one of each object, an order which deviates from the structural order of the objects involved. This distinction is illustrated in Figures 3.3 and 3.4.

Orchestration is a term used to describe the actions of selecting, assigning, directing and evaluating the performance of objects which participate in a transformation. Transformations can happen concurrently and/or in different speeds. The result is a *moving image* the behavior of which becomes the responsibility of the user. As in an orchestra performance the architect/composer selects a number of objects he/she wants to include, assigns the proper transformation paths and speeds, and then directs the performance through time, form and color.

The essence of such transformational design is not that much in the final form but rather in the intermediate phases these transformations pass through, as well as, in the extrapolations which go beyond the final form. The user has the capability, through the system, to modify and control the flow of the compositional evolution and replay it many times by varying some or all of the transformational parameters.

The user controls the flow of the moving image through the mouse which can be moved in two directions: horizontal (right-left) and vertical (up-down). The horizontal direction represents *time* while the vertical represents *space*. As the cursor is moved upwards or downwards the objects are transformed forwards or backwards, respectively. The speed is relative to the movement of the mouse. Similarly, when the user moves the cursor right or left, the whole scene rotates in space clockwise or counter-clockwise.

4. The Transformation Algorithm

When an object is transformed, it changes its shape gradually in order to match the shape of the object it transforms to. A cube, for example, may be gradually transformed into a pyramid. From the user's point of view, there are always two objects: ***the original***, to which transformation is applied, and the ***destination object***, which is the object one will get at the final step of the transformation. However, internally, there is only one object, which is transformed from one state (original) into another (destination). This object combines characteristics of both objects which are involved in the transformation and will be called a ***hybrid object.*** This object is actually composed of the topology of one object and the geometry of another.

Let O1->O2 be a transformation between two objects O1 and O2. The algorithm to create a hybrid object is as follows:

1) Compare O1 and O2. Mark as Omax the object with the most coordinates. Mark Omin the other.

2) Consider an xyz-axis system which passes through the centroid of Omax and divides it in eight quarter-blocks. Mark each vertex of Omax with a number corresponding to the quarter-block to which it belongs. This number is called the *orientation number*.

3) Do the same for Omin's vertices.

4) Compare each point of Omax and Omin. If the orientation numbers match, substitute the Omax vertex with the corresponding Omin vertex.

5) Interpolate the remaining (unmatched) points of Omax, so that they fall between the previous and the next vertex (found in step 4).

This algorithm constructs a hybrid object which has the topology of Omax, but its vertices match the vertices of Omin. This object appears like Omin, but topologically it is Omax.

After constructing the hybrid object, the increment by which this object will be changed in order to move from one position (original) to the other (destination) needs to be found. Since we know in advance the original and the destination object, we subtract the original object from the hybrid object, or the hybrid from the destination, depending on the direction of transformation and divide each by the number of steps the user has indicated. Those increments are stored in an array. When the increments are progressively added to the hybrid object, it transforms from one shape to another.

4. The Music Analogy

Throughout this discussion, the terms orchestration, composition and synchronization have been frequently used . It seems that, intentionally or not, there is an analogy, a similarity between music and transformations.

Originally, there was no intention to give musical characteristics to the transformation process. However, as the development of the system proceeded, borrowing analogies from music became inevitable. Furthermore, that analogy became even more necessary when more than one objects were in motion. They had to be somehow synchronized whenever, for example, their speeds were different. Thus issues which were more common in music than in architecture had to be addressed. The structural characteristics of music were similar to those of transformation simply because they had a common denominator: *time*.

Consequently, the thought arose whether additional analogies other than time, could also be applicable, such as, melody, rythm or harmony. The idea of orchestration became evident, first taken, of course, from music, and was used to describe the action of selecting objects, assigning tasks and evaluating performances. The conductor's baton became the cursor's movement on the screen indicating time beat and spatial viewpoint. Cords were derived through assignment of parallel movements of objects as if they were parallel voices and a pause was simply the transformation of an object to itself remaining temporarily in it's own position. Possibly, new tasks can be invented to perform similarly to music functions, such as, cadence, harmony or scaling.

What is most interesting in this music comparison is that there is a problem of preferences; it seems that certain motions are preferred to others. Take, for example, a combination of scaling and rotation accelerating in time. It is much more interesting than a simple rotation of the same speed, simply because the first gives more complex feedback to the eyes. This leads to a kind of preference ranking. It may be similar to what harmony is for music. Certain tones, or certain combinations of tones, are preferred to others. Possibly a scale of choices may be constructed of motions, or combinations of motions of objects leading to a harmonic scale of spatial motions. That then can lead to a more complex combination such as a prelude, or a ballet or maybe even a symphony. Again, here it must be repeated, that all those terms refer to the structural similarities of music and architecture and not to their external appearances. The similarity lays between common concepts and not common names.

6. The Performance

A sequence of transformational events is called *performance*. The user orchestrates a performance by selecting objects, assigning tasks and setting them to run under his/her control. Depending on the way the tasks are set, the user can produce rythm, synchronization,
or pause.

A transformational process can be any of the following types:

- transform an object into itself (identity transformation)

- transform object 1 into object 2 and object 2 into object 1
  (bi-directional)

- transform object 1 into object 2 and object 1 into object 3
  (fork)

- transform object 1 into object 2, object 2 into object 3 and
  object 3 into object 1 (circular).

- transform object 1 into object 3 and object 2 into object 4
  in a parallel direction (parallel)

- transform object 1 into object 3 and object 2 into object 4
  crossing each other (cross).

These transformational types are illustrated in Figure 6-1.

Fig 6-1 Different combinations of transformations

In mathematical terms, a transformation can be viewed as a mapping function between one, two or more collections of points. Thus, it can be symbolically represented as ***F(x,y,z)=(f(x,y,z),g(x,y,z),h(x,y,z)).*** This representation is useful because it allows one to explore the possibilities of transformations in a more abstract algebric level.

The following is a list of some transformational formulas:

1. Transformations of a single object:

Movement:

$F(x,y,z) = (x+xoff, y+yoff, z+zoff)$

Magnification:

$F(x,y,z) = (x*sx, y*sy, z*sz)$

Rotation (z-axis):

$F(x,y,z) = (x+\cos(a)+y*\sin(a), y*\cos(a)-x*\sin(a),z)$

Oblique collapse:

$F(x,y,z) = (x+y/2, y/ 2, z+y/2)$

2. Transformations between two objects a and b:

Linear movement:

$F(xa,ya,za) = (xa+(xb-xa)/n, ya+(yb-ya)/n, za+(zb-za)/n)$

where *n* the number of steps

Sinus movement:

$F(xa,ya,za) = (xa+\sin(a)*(xb-xa), ya+\sin(a)*(yb-ya), za+\sin(a)*(zb-za))$

where $0<a<p/2$

The cases shown above, describe transformations which affect the geometry of the object(s) in 3-D space, that is, their coordinates. A transformation can also affect the topology of the object(s) or certain attributes, such as, color, line type, line width, etc. The transformation of those parameters are yet to be explored.

7. Examples

By observing the way forms are flowing in time and space the user can modify the tasks and change the flow. The final step is an interrupt and the user stops the flow and observes the frozen frame. He/she can modify it through editing procedures or render it to produce a real world image. To illustrate the point, the following experiments were constructed using some of the transformational processeses.

7.1. Variations on the Barcelona Pavillion.

The example shown in Figure 7.1 illustates the use of transformations between the structural elements of a single building (Barcelona Pavillion, Mies van der Rohe, Barcelona, 1929). Each wall is transformed into another the direction of which is shown through the arrows. A number of new elements are derived. Some of the new elements are rotated. The same process is repeated several times with the new elements.

7.2. Nine-cube transformation.

The example shown in Figure 7.2 is an experiment where transformations are applied between faces. A nine-cube is used as the basis. Then a number of transformations are assigned between pairs of faces, one on each of two cubes.

7.3. Transforming chopsticks.

This experiment shown in Figure 7.3 aims at exploring possible ways which may suggest dynamic change through relationships established between elements. A number of thin beams are initially constructed in directions which are parallel to the three orthogonal axes of x, y and z. They are then rotated in space by random angles. As the rotations are dynamically executed a number of preassigned transformations are also performed and are allowed to leave traces in 3-D.

7.4. Cross-programming two buildings.

The example of Figure 7.4 shows two buildings, an ancient Greek doric temple and Mies van der Rohe's brick house, transforming into each other. Any number of new buidings can be obtained by selecting any of the in-between transformational steps. Any such new building will be sharing physical features of both.

7.5. Extrapolating beyond the destination form.

The example in Figure 7.5 illustrates transformations where three buildings are involved: two buildings by Mies van der Rohe and one by Frank Lloyd Wright. As one building form transforms into another, it is also allowed to go beyond the destination form, which frequently unfolds some interesting and unexpected results. Some are shown in Figure 7.5.

7. Conclusions

The aim of this paper was to present an experimental design tool implemented in a CAAD system. The main idea is the dynamic transformation of 3-D objects. In brief, the system offers to the user/architect the means to create and edit architectural models, transform them from one position or form into another and finally render them.

This project attempts to introduce the factor of *time* in the architectural design process. It takes advantage of the ability the computer has to dynamically display images in real-time, and uses it as a design tool where the relative position of objects play a much more important role than the objects themselves. The user is placed in the position of a director and the whole design is a process of synchronization and orchestration of moving images.

References

1. **Ammer C. [1972]** Harper's Dictionary of Music, Harper & Row, New York, 1972

2. **Cole H. [1974]**, Sounds and Signs, Oxford University Press, London, 1974

3. **Eisenman P. [1986]** "The Futility of Objects", *Harvard Architecture Review 3*, p. 66.

4. **Evans, R. [1986]** "Not to be used for wrapping purposes", *AAFiles* 10, p. 70

5. **Libeskind D. [1981]** Between zero and infinity, Rizzoli, 1981

6. **Yessios C. [1987],** "A Fractal Studio", *ACADIA 87 Proceedings*, North Carolina State University, 1987.

Fig. 6-2 Transformations between elements of a building

Fig. 6-3 Transformations between the faces of a nine-square

Synthesis with chopsticks

Fig 6-4. Elements leaving traces in 3-D space.

Cross-programming two buildings

Fig. 6-5 transformations between elements of two buildings