

Experiments on Morphing Systems

Kostas Terzidis, Ph.D.
UCLA – Arts and Architecture
Los Angeles CA USA
kostas@ucla.edu

Abstract

This paper presents recent experiments on 3D morphing of buildings. A genealogical tree is created out of cross-morphing buildings showing their children and grandchildren. The resulting children-buildings share characteristics of the formal properties of their parents. There are two methods used here to morph buildings: face-to-face mapping and object-to-object mapping. All morphed buildings are shown as real-time animation. A series of experiments are presented. Some experiments investigate the implementation of architecture or art theories. For example, how would it look like to morph a Graves building into a Le Corbusier building?

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. [1], [3] and [4]

Imagine a computer screen where all is in motion. A variety of elements, either of a pure form, such as a 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, or may be applied at different speeds, or they may be interrupted and redirected.

The key theme is the morphing of physical entities. When these entities are viewed as parts of an architectural parti, then the theme becomes the morphing of architectural forms. The process of morphing goes beyond the simple transformations of translation, rotation and scale. They include the morphing 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. The implementation

Morphing is a process in which an object changes its form gradually in order to obtain another form. The operation of morphing consists basically of the selection of two objects and the assignment of n , the number of in-between steps. The first object then morphs into the second in n steps. This process is illustrated in Figures 2.1. 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 morphed, the user is allowed to create his/her own morphings by matching pairs of faces, one of each object, an order which deviates from the structural order of the objects involved.

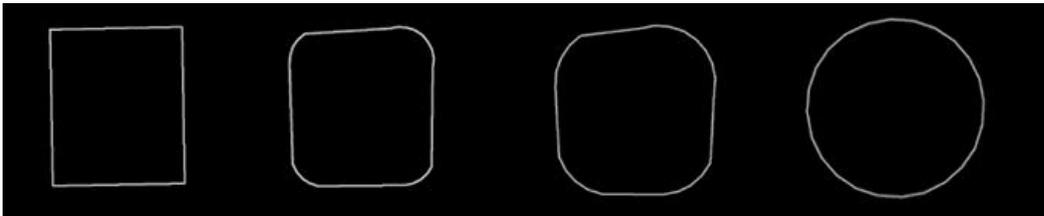


Figure 2.1. The interpolation of a square to a circle in four steps

The essence of such morphing design is not that much in the final form but rather in the intermediate phases these morphings pass through, as well as, in the extrapolations that 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. [2]

The computer system that was developed by the author for this paper is called "zhapes" and is a Java-based 3D-experimentation system. It resides at the address <http://www.zhapes.com> where it can be downloaded for explorations. In zhapes, 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.

3. The Morphing Algorithm

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

Let $O1 \rightarrow O2$ be a morphing between two objects $O1$ and $O2$. The algorithm to create a hybrid object is as follows:

- 1) Compare $O1$ and $O2$. Mark as O_{max} the object with the most coordinates. Mark O_{min} the other.
- 2) Consider an xyz-axis system which passes through the centroid of O_{max} and divides it in eight quarter-blocks. Mark each vertex of O_{max} with a number corresponding to the quarter-block to which it belongs. This number is called the *orientation number*.
- 3) Do the same for O_{min} 's vertices.
- 4) Compare each point of O_{max} and O_{min} . If the orientation numbers match, substitute the O_{max} vertex with the corresponding O_{min} vertex through the formula $O_{min}[i] = \text{mod}(\text{number of points of } O_{max}, i)$.
- 5) Interpolate the remaining (unmatched) points of O_{max} , 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 O_{max} , but its vertices match the vertices of O_{min} . This object *appears like O_{min} , but topologically it is O_{max}* .

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's positions, 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. 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. The user 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 morphing processes.

4.1. *From Le Corbusier to Graves.*

The example of Figure 4.1 shows two buildings, a Le Corbusier building and a Michael Graves house, morphing into each other. Any number of new buildings can be obtained by selecting any of the in-between morphing steps. Any such new building will be sharing physical features of both.

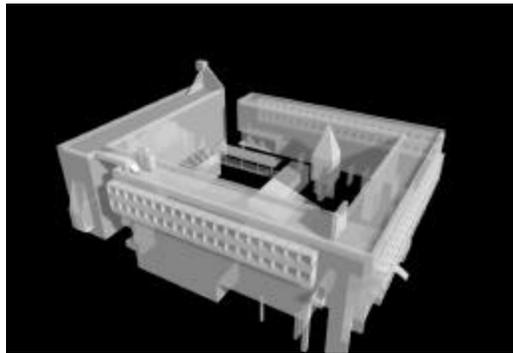


Figure 4.1-1. Parent A of the morphing

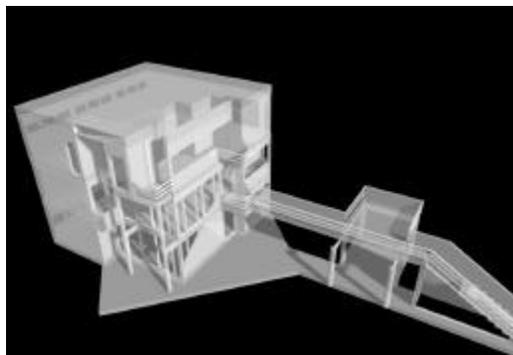


Figure 4.1-2. Parent B of the morphing

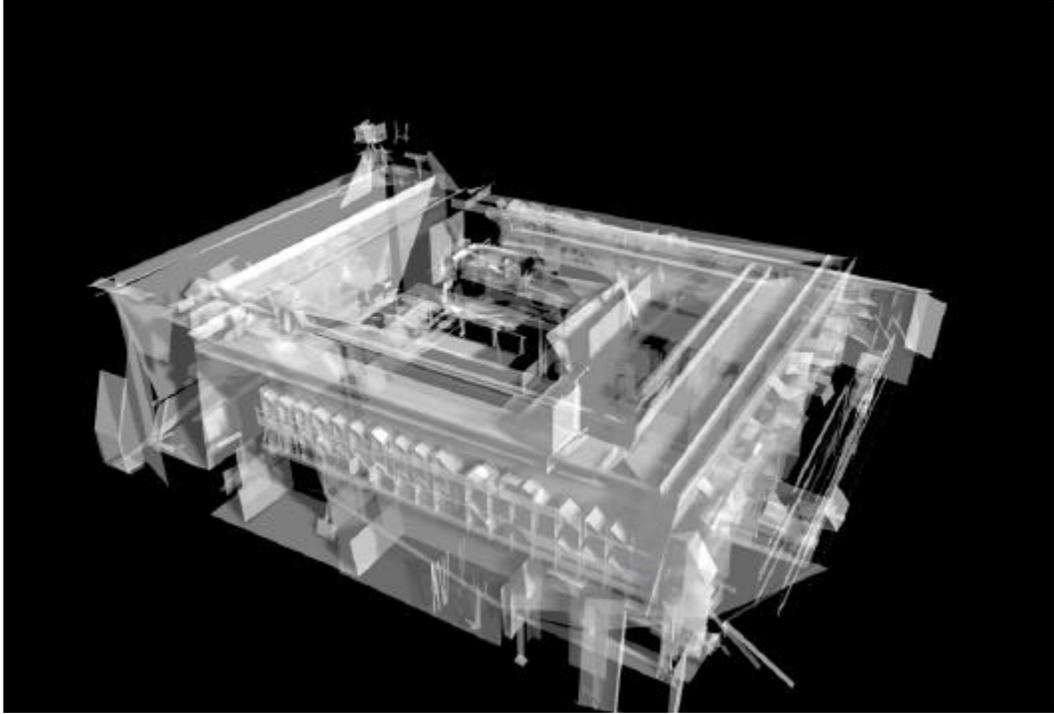


Figure 4.1-3. Morphed object at 25% away from parent A

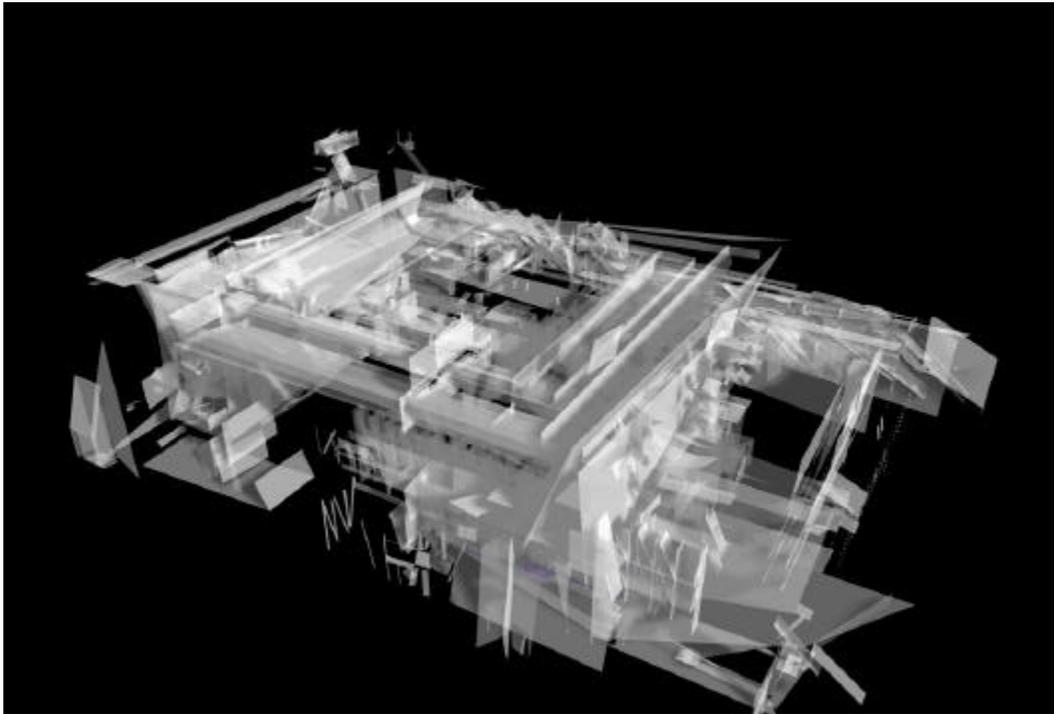


Figure 4.1-4. Morphed object at 50% away from parent A

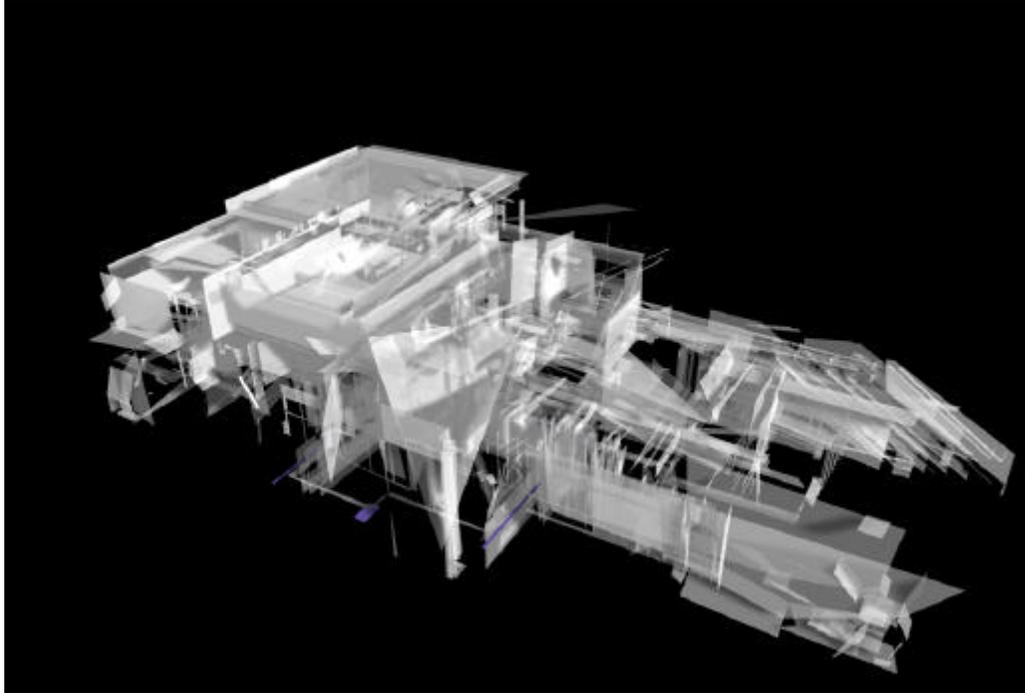


Figure 4.1-5. Morphed object at 75% away from parent A

4.2. Richard Meier altered.

The example shown in Figure 4.2-1 is an experiment where transformations are applied between faces. A Richard Meier house is used as the basis. Then a number of transformations are assigned between pairs of faces, within the same building. In this case parent A and parent B are the same.



Figure 4.2-1. Morphing an object to itself

4.3. Extrapolating beyond the destination form.

The example in Figure 4.3-1 illustrates morphing allowed to go beyond the destination form, which frequently unfolds some interesting and unexpected results.

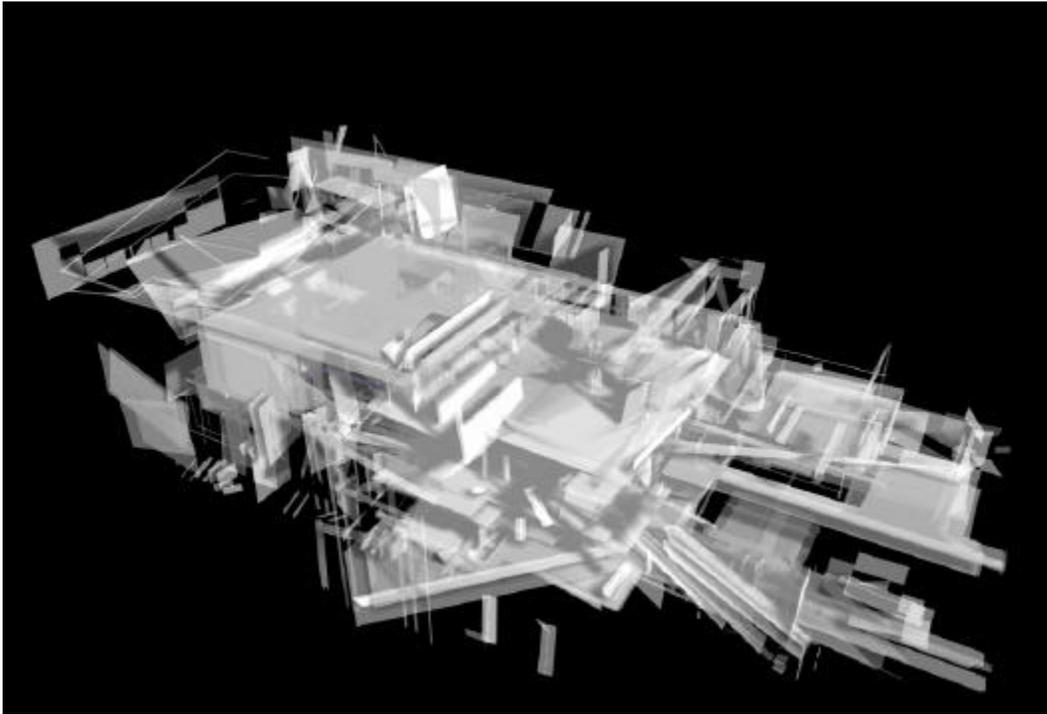


Figure 4.3-1. Morphed object at 125% away from parent A

5. Conclusions

The aim of this paper was to present an experimental design tool implemented in a CAAD system. The main idea is the dynamic morphing 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 shapes.

References

- [1]. **Eisenman P. [1986]** "The Futility of Objects", *Harvard Architecture Review* 3, p. 66.
- [2]. **Evans, R. [1986]** "Not to be used for wrapping purposes", *AAFiles* 10, p. 70
- [3]. **Libeskind D. [1981]** Between zero and infinity, Rizzoli, 1981
- [4]. **Yessios C. [1987]**, "A Fractal Studio", *ACADIA 87 Proceedings*, North Carolina State University, 1987.