

# Experiments on Animated Visual Forms

Kostas Terzidis, Ph.D.  
School of Arts and Architecture  
University of California at Los Angeles

A series of experiments that investigate and demonstrate the visual logic of three-dimensional representation in animated form through the use of computers is presented.

Perspective systems are designed to construct pictures that, when viewed, produce in the trained viewer the experience of depicted objects that match perceivable objects. Panofsky [1] wrote about how our capacities to see are constrained by the perspective system that we use, that is, by our way of depicting what we see. The kind of pictorial spaces are expressed through geometrical models. Each model is expressed as a geometrical transformation applied to Euclidean/Cartesian shapes of the physical environment. These transformations show how shapes are projected in pictorial space [2].

The work of Gauss, Riemann, von Helmholtz, and Clifford [3] showed that geometries could be based on axioms other than Euclidean. This altered the notion of geometry by revealing its abstract postulational character and removing it from any connection with the structure of empirical intuition. Spaces were constructed on the basis of non-Euclidean axioms revealing behaviors closer to our sensations rather than our perception. Luneburg's [4] idea of hyperbolic space was a first attempt to develop a theory of visual space in which geometrical structure is a metric space of constant, probably negative, Gaussian curvature.

On the assumption that visual space is hyperbolic and physical space is Euclidean, it is possible to derive computational numerical models (or mappings) between the two spaces [2]. From a study of these models, we can get a good visual sense of the way our physical environment would manifest itself to hyperbolic vision.

I shall attempt to present below a series of experiments that demonstrate some of these models or mappings for a variety of values for the parameters of the spaces. The experiments address geometric mappings as well as numerical models of projection and their interest lies in the dynamic nature of the continuous computer processing, which is hard to show on a static piece of paper. It is also important to mention that the objects in the physical environment maintain in all experiments their Cartesian coordinates. It is the mappings and models that change.

## Projecting Along Curves

The transformation equations for an orthographic parallel projection are straightforward. If the viewing plane is placed at the origin along the  $z$ -axis, then any point  $(x,y,z)$  in viewing coordinates is transformed to projection coordinates  $(x_p,y_p)$  as:

$$x_p = x, \quad y_p = y \quad (1)$$

In parallel projection the  $z$ -coordinate value is omitted. In contrast, in perspective projection, the  $z$ -coordinate value is involved in the transformation equations of the projection. If the viewing plane is placed along the  $z$  axis, then any point  $(x,y,z)$  in viewing coordinates is transformed to projection coordinates as:

$$x_p = x \cdot t, \quad y_p = y \cdot t \quad \text{where} \quad t = d + \frac{d}{z} \quad (2)$$

and  $d$  is the distance of the viewer to the viewing plane. The orthographic and perspective projections can be seen in figures 1 and 2.

In the first experiment,  $d$  was assigned a negative value, thus the viewer is assumed to be in the opposite position of the viewing plane and the perspective projection equations become:

$$x_p = x \cdot t, \quad y_p = y \cdot t \quad \text{where} \quad t = -d - \frac{d}{z} \quad (3)$$

From the viewer's point of view, the effect of this alteration was an inverted world where faces of objects appear to be inside out and movements appear to shift surprisingly in the opposite direction. This can be seen in figure 3.

In the second experiment, the equations of perspective projection were scaled by a factor of  $n$ . The corresponding equations then became as follows:

$$x_p = n \cdot x \cdot t, \quad y_p = n \cdot y \cdot t \quad (4)$$

From the viewer's point of view, the projected objects appear to be coming from the vanishing point and then as the objects are rotated they seem to be returning back to the vanishing point. This can be seen in figure 4.

Projection of a point  $(x,y,z)$  on a viewing plane does not have to occur along a straight line. In fact, many animals view the world through eyes of different shape and curvature resulting in seeing differently the same objects. The more curved or warped the eye is the more exaggerated the projection becomes. Along this line of thought, the following four experiments employ equations that use exponential and trigonometric functions to map the  $(x,y,z)$  points on the viewing surface. The first uses an exponential function.

Points are projected along an exponential curve. Distances appear greater as we move away from the center of projection. The first set of equations is:

$$x_p = x^n, \quad y_p = y^n \quad (5)$$

From the viewer's point of view, objects appear to be elongated along the borders of the viewing plane. This is shown in figure 5.

The second set of equations has the opposite effect. In this case, a point  $(x,y,z)$  is transformed through a fractional power

$$x_p = \sqrt[n]{x}, \quad y_p = \sqrt[n]{y} \quad (6)$$

From the viewer's point of view, objects appear to be concentrated towards the center of the viewing plane. This is shown in figure 6.

If we rotate the viewing projection lines to become trigonometric curves the projection equations can become as follows:

$$x_p = x \cdot \sin(t) + y \cdot \cos(t), \quad y_p = y \cdot \sin(t) - x \cdot \cos(t) \quad (7)$$

By involving the z-coordinate in the projection curve the equations can become

$$x_p = x \cdot \sin(y) + y \cdot \cos(t), \quad y_p = y \cdot \sin(x) - z \cdot \cos(t) \quad (8)$$

These transformations can be seen in figures 7 and 8.

### **Geometrical Transformations Using Projection Formulas**

In the following set of experiments, the same formulas used to project 3D points on 2D surfaces are used to transform 3D points. For example, the perspective formula is used to transform the  $(x,y,z)$  coordinates of a point in the following way:

$$x' = x \cdot t, \quad y' = y \cdot t \quad \text{and} \quad z' = z \cdot t \quad \text{where} \quad t = d + \frac{d}{z} \quad (9)$$

In the same spirit, the inverted, exponential, and trigonometric formulas used earlier in equations (1) through (8) to map  $(x,y,z)$  point on to  $(x_p, y_p)$  projected points are used to produce instead 3D transformations. Some of these results can be seen in figures 9, 10, and 11.

In the next experiment, a spherical projection is used. A spherical projection is based on distributing points on the surface of a sphere of radius  $r$ . The mapping and projection equations are:

$$x' = x \cdot r \cdot \cos(\mathbf{q}) \cdot \cos(\mathbf{f}) \quad y' = y \cdot r \cdot \sin(\mathbf{q}) \cdot \cos(\mathbf{f}) \quad z' = z \cdot r \cdot \sin(\mathbf{f}) \quad (10)$$

where 
$$\mathbf{q} = \frac{2 \cdot \mathbf{p} \cdot \mathbf{x}}{npoint s} \quad \mathbf{f} = \frac{2 \cdot \mathbf{p} \cdot \mathbf{y}}{npoint s}$$

The result of this mapping is shown in figure 12.

In the next experiment, a hyperbolic projection is used. In hyperbolic space -- one of the non-Euclidean geometries -- the formulas involve the hyperbolic sine and cosine functions (sinh and cosh), that are exponential. Hyperbolic space is infinite in extent, just like Euclidean space. However, it is possible to map the entire infinite space into a finite portion of Euclidean space. It might surprise you that you can map an infinite amount of space with "more room" into a finite piece of a space with "less room", but non-Euclidean geometries have many unexpected consequences for Euclidean intuitions. The equations used are:

$$x_p = x \cdot \cosh(\mathbf{q}) \cdot \cosh(\mathbf{f}) \quad y_p = y \cdot \sinh(\mathbf{q}) \cdot \cosh(\mathbf{f}) \quad z' = z \cdot r \cdot \sinh(\mathbf{f}) \quad (11)$$

where 
$$\mathbf{q} = \frac{2 \cdot \mathbf{p} \cdot \mathbf{x}_i}{\sum i} \quad \mathbf{f} = \frac{2 \cdot \mathbf{p} \cdot \mathbf{y}_i}{\sum i}$$

and 
$$\sinh(\mathbf{q}) = \frac{e^{\mathbf{q}} - e^{-\mathbf{q}}}{2} \quad \cosh(\mathbf{q}) = \frac{e^{\mathbf{q}} + e^{-\mathbf{q}}}{2}$$

The result of this mapping is shown in figure 13.

Finally, random disturbance of the position or the rotation of faces in objects is assigned. The equations are simply:

$$x_p = x \cdot \text{random}(\text{seed}), \quad y_p = y \cdot \text{random}(\text{seed}) \quad (12)$$

These produced the effect of "blurring", "wavy", "fuzzy", or "scratchy". The result of this disturbance is shown in figures 14 and 15.

## Multiple-view Projections

Projection of a point  $(x,y,z)$  on a viewing plane does not have to occur along a single straight or curved line as the previous experiments did. In fact, many creatures view the world by collecting information from different points of view at the same time. The more eyes the viewer and the more offset they are the more collage-like the projection becomes. Along this line of thought, the following two experiments show how a series of objects would appear through multiple views at the same time.

In the first experiment, the viewing plane is divided in areas each one of which projects the scene from a different point of view. As the viewer rotates the scene, the objects appear to be connected and disconnected at the same time. The result of this experiment can be seen in figure 16.

In the second experiment, each face of each object in the scene is projected along a different point of view. Specifically, the equations (2) and (3) are used to project every other face of each object along different values of  $d$ , that is, either  $d$  or  $-d$ . The result of this experiment is shown in figure 17. From the viewer's point of view, faces of objects seem to move in opposite directions at the same time as the scene is rotated.

$$face_{2n+1} \rightarrow view_d \quad face_{2n} \rightarrow view_{-d} \quad (13)$$

### **Dynamic Filtering**

In the next set of experiments, the geometry of the projection is not affected. Instead, the viewing plane is affected by filtering the screen. The effect is dynamic since the shifts happen as the scene rotates. In the first experiment, the screen is divided into small squares and each one copies its neighboring pixels. The result can be seen in figure 18.

In the next three experiments, the screen is divided into random size small shapes and each one copies random neighboring pixels. The result can be seen in figures 19 and 20.

### **Conclusions**

The aim of this paper was to present an experimental tool implemented in a computer system. The main idea is the dynamic transformation of 3-D objects. In brief, the system offers to the user the means to view geometrical models, transform them through a series of different mapping systems and, at the same time, filter them.

The system attempts to introduce the factor of time in the study of visual systems. It takes advantage of the ability the computer has to dynamically display images in real-time, and uses it as a tool where the way objects are viewed plays a much more important role than the objects themselves. The user is placed in the eyes of another creature and visualization is a process of transformation and filtering of moving images.

### **Technical Information**

The experiments presented in this paper are implemented on a computer program called *Zhapes*. The *Zhapes* routines are written in the both C++ and Java programming languages. The decision to use both languages was based on the wish to ensure graphics processing speed and interaction with the World Wide Web. Since the two programs are

tied together, they share common data structures, which are implemented in C++. All routines dealing with network interaction are in Java. Dynamic transformations, such as rotation, translation, and zooming are implemented in C++ in order to take advantage of assembly level code to boost-up the real-time processing.

*Zhapes* can run both as a 32-bit application and as an applet on any web browser. It resides at the address <http://www.cda.ucla.edu/caad/java/x/formProj2/formB.html> where it can be downloaded for testing.

## References

- [1] Panofsky, E. "Perspektive als 'symbolische Form'." In Saxl, (ed.), 1927, pp.258-330
- [2] Heelan, P. "Space-perception and the Philosophy of Science." Berkeley: University of California Press, 1988.
- [3] Clifford, W. K. "The Common Sense of Exact Sciences." New York: Knopf. 1946. First publ. in 1885.
- [4] Lunenburg, R. "The metric of visual space." J. Optic. Soc. Amer. 40: (1950), pp. 627-642.

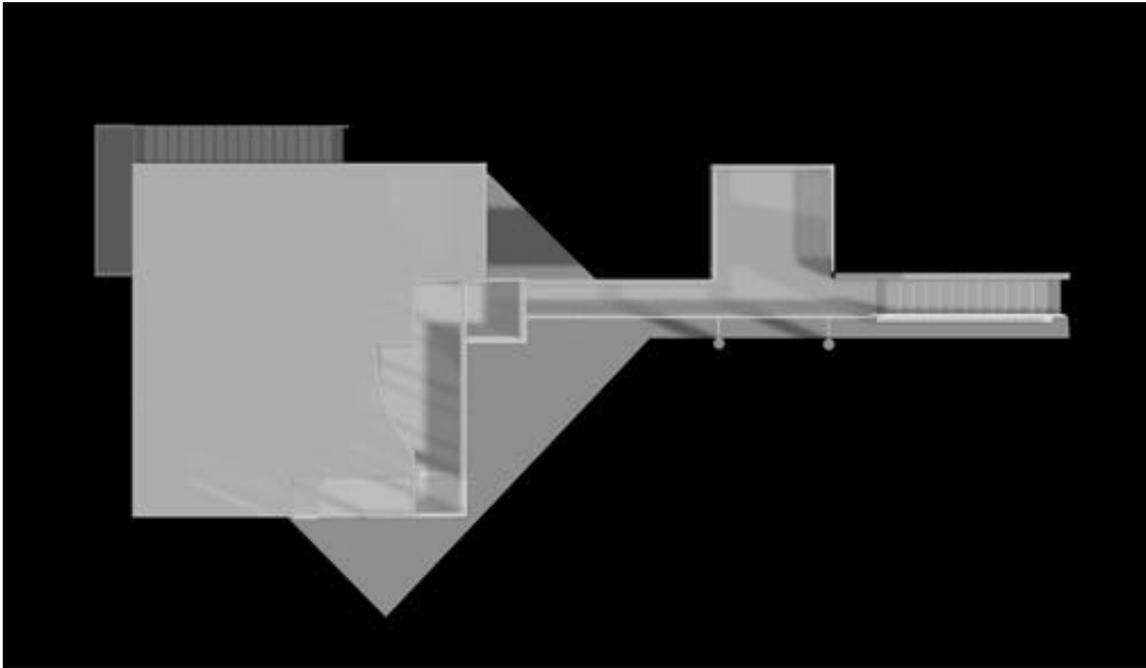


Figure 1. Orthographic View:  $x_p = x$ ,  $y_p = y$

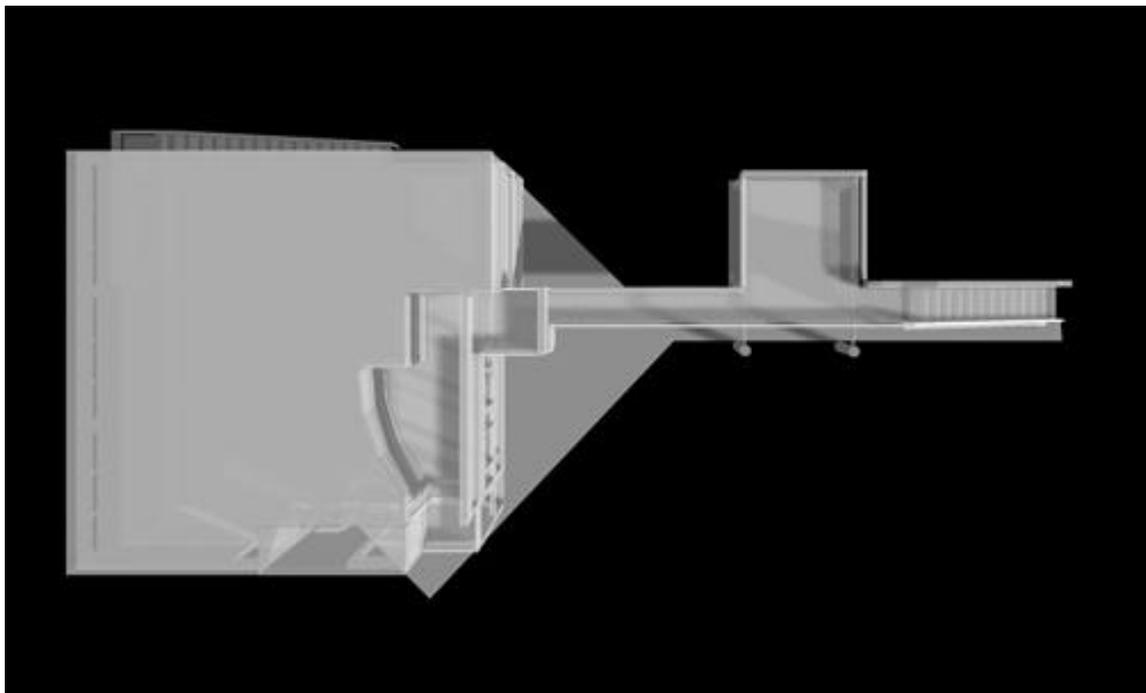


Figure 2. Perspective View:  $x_p = x \cdot t$ ,  $y_p = y \cdot t$   $t = d + \frac{d}{z}$

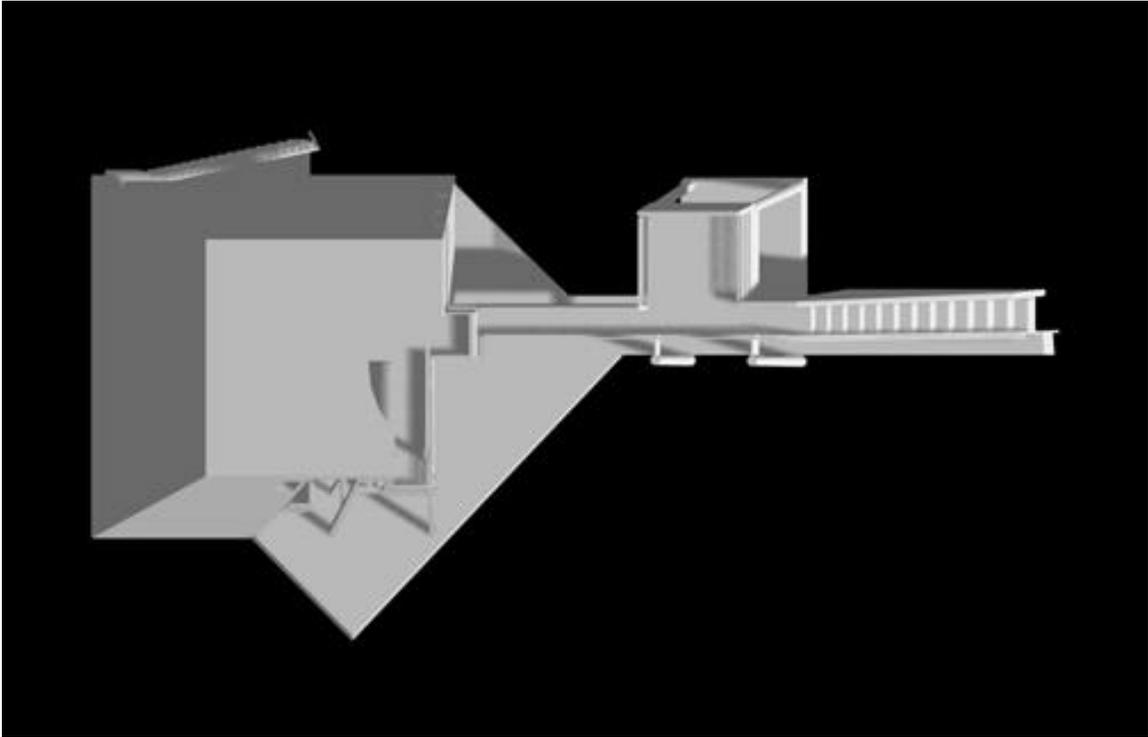


Figure 3. Inverted Perspective:  $x_p = x \cdot t$ ,  $y_p = y \cdot t$   $t = -d - \frac{d}{z}$

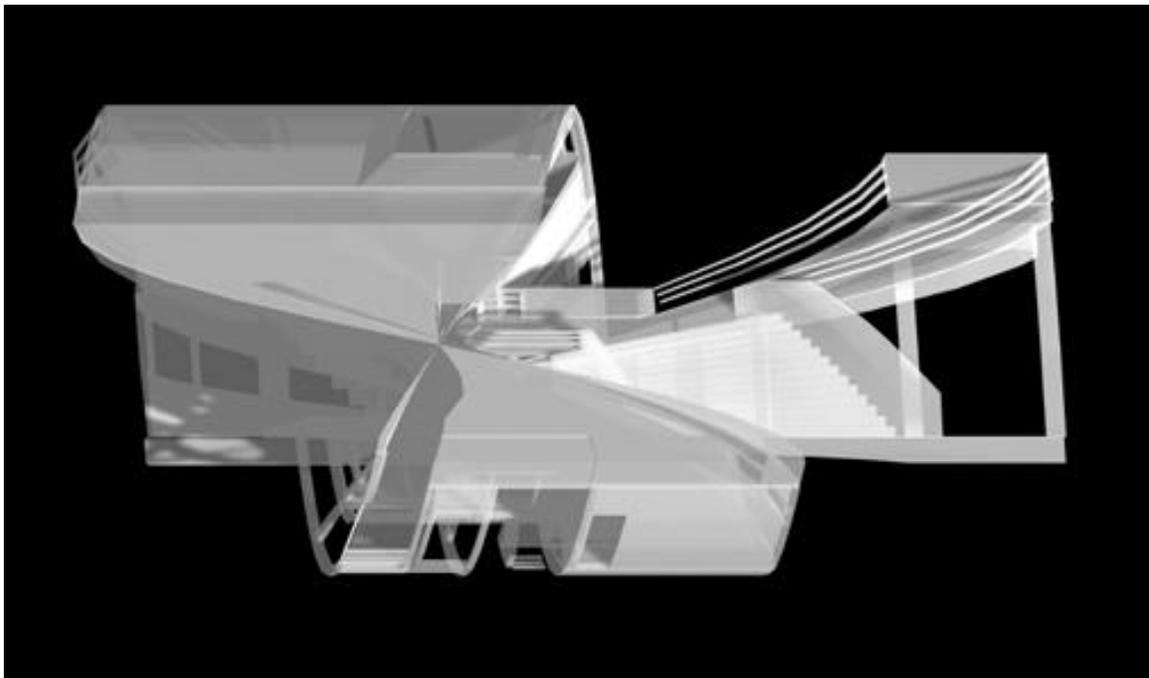


Figure 4. Scaled Perspective:  $x_p = n \cdot x \cdot t$ ,  $y_p = n \cdot y \cdot t$

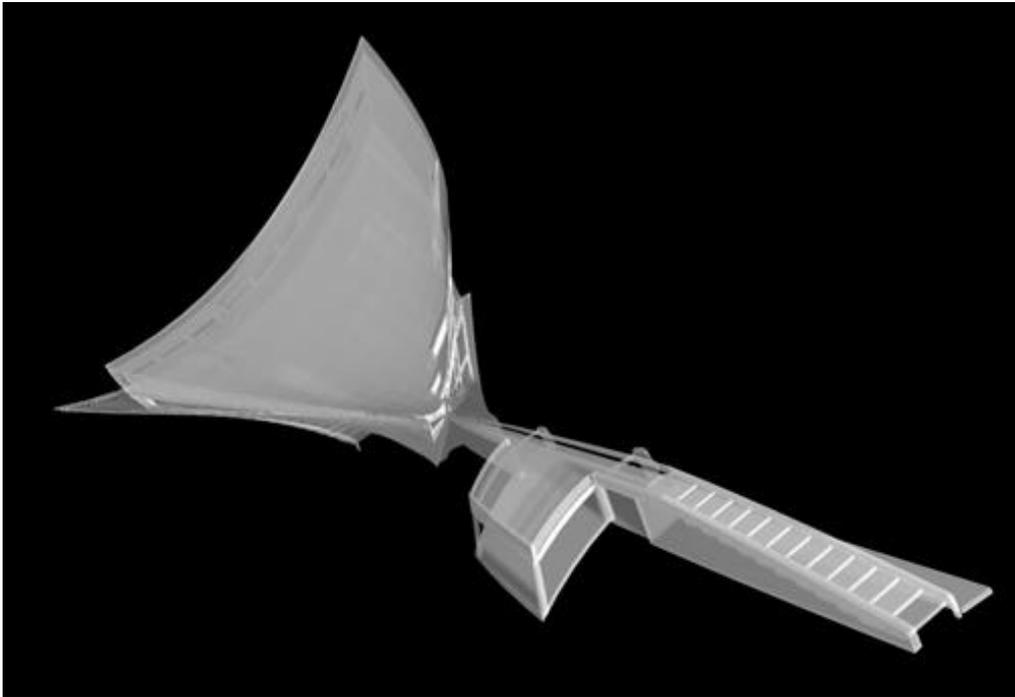


Figure 5. Exponential Projection:  $x_p = x^n$ ,  $y_p = y^n$

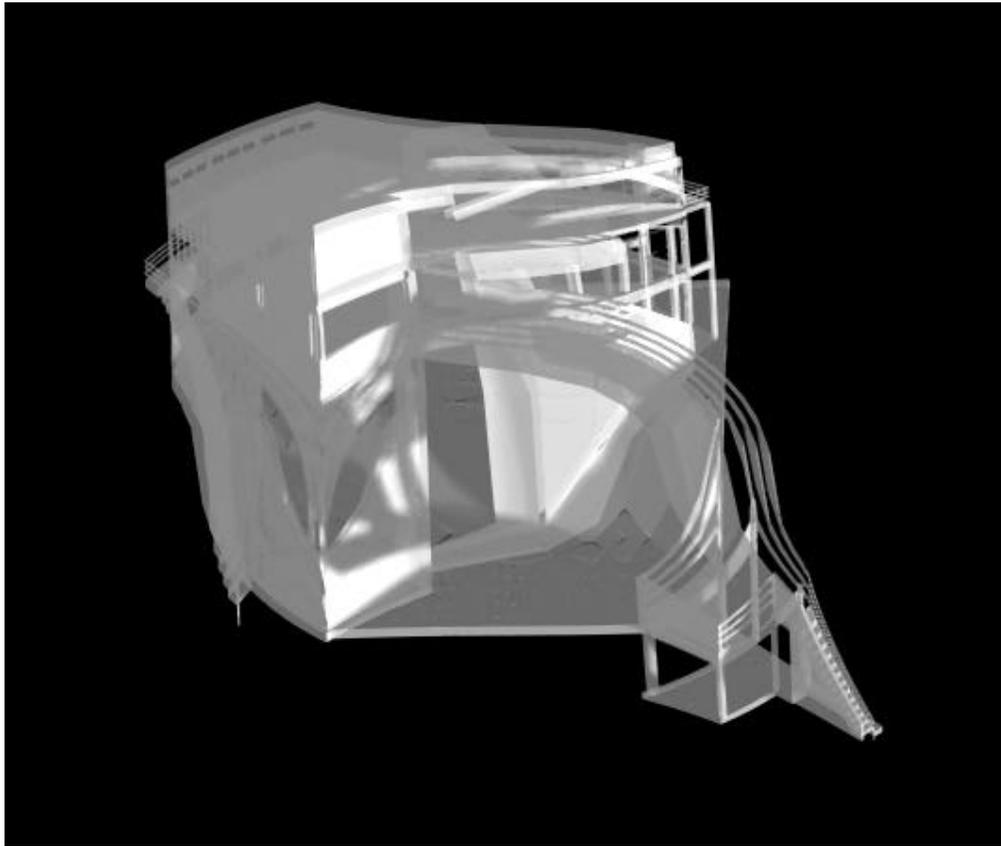


Figure 6. Exponential Projection:  $x_p = \sqrt[n]{x}$ ,  $y_p = \sqrt[n]{y}$

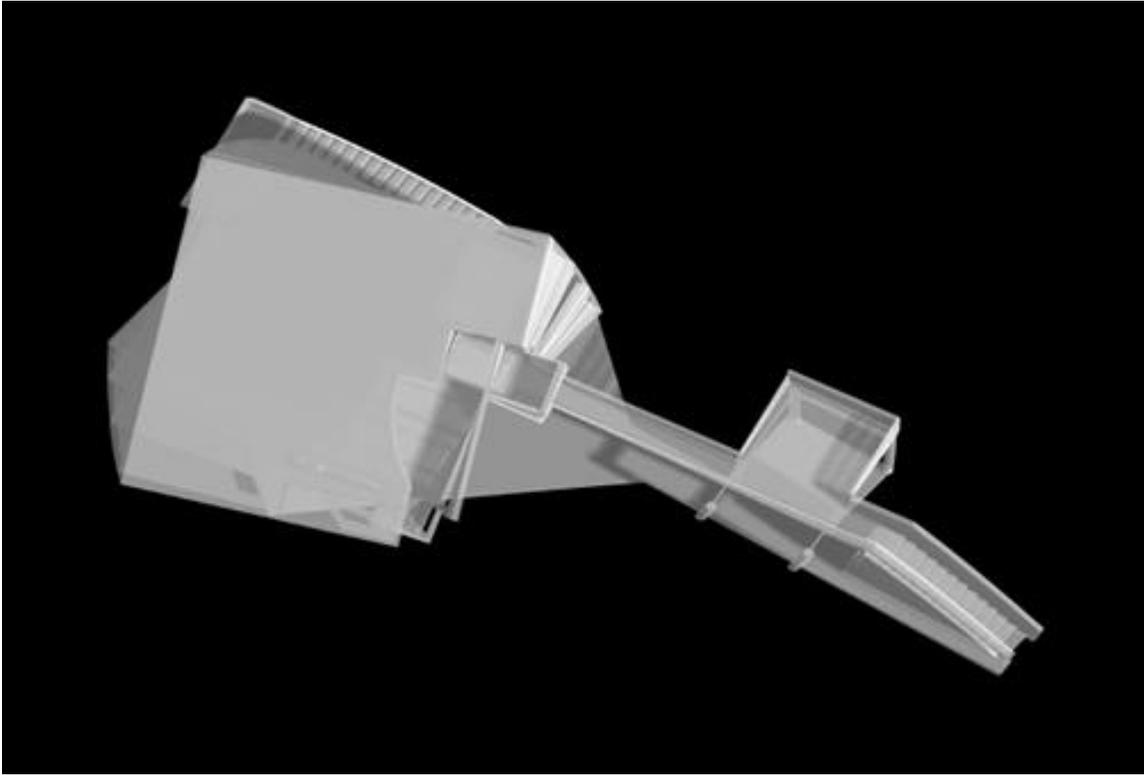


Figure 7. Trigonometric Projection:  $x_p = x \cdot \sin(t) + y \cdot \cos(t)$ ,  $y_p = y \cdot \sin(t) - x \cdot \cos(t)$

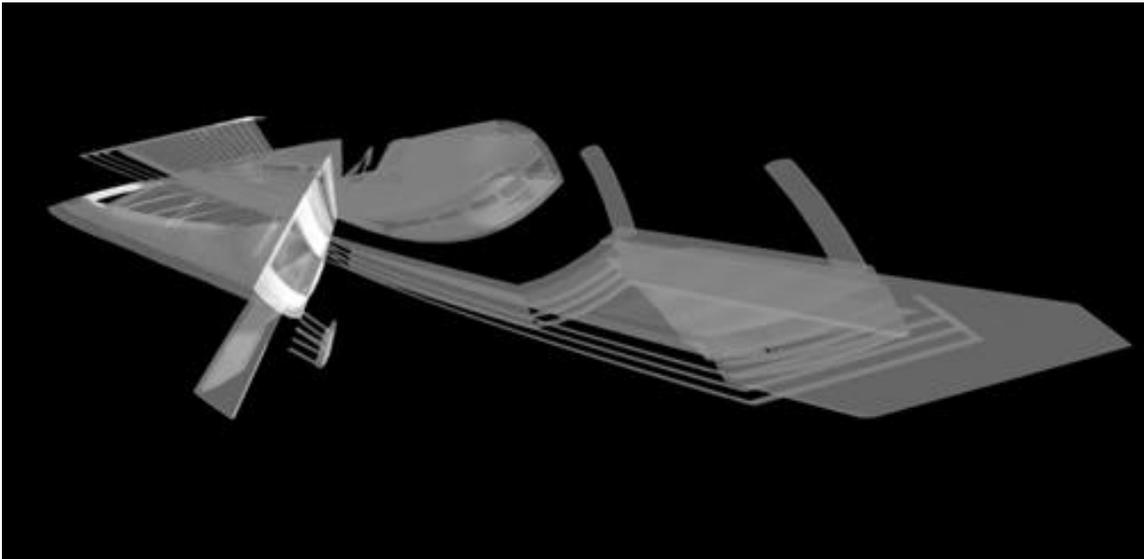


Figure 8. Trigonometric Projection:  $x_p = x \cdot \sin(y) + y \cdot \cos(t)$ ,  $y_p = y \cdot \sin(x) - z \cdot \cos(t)$

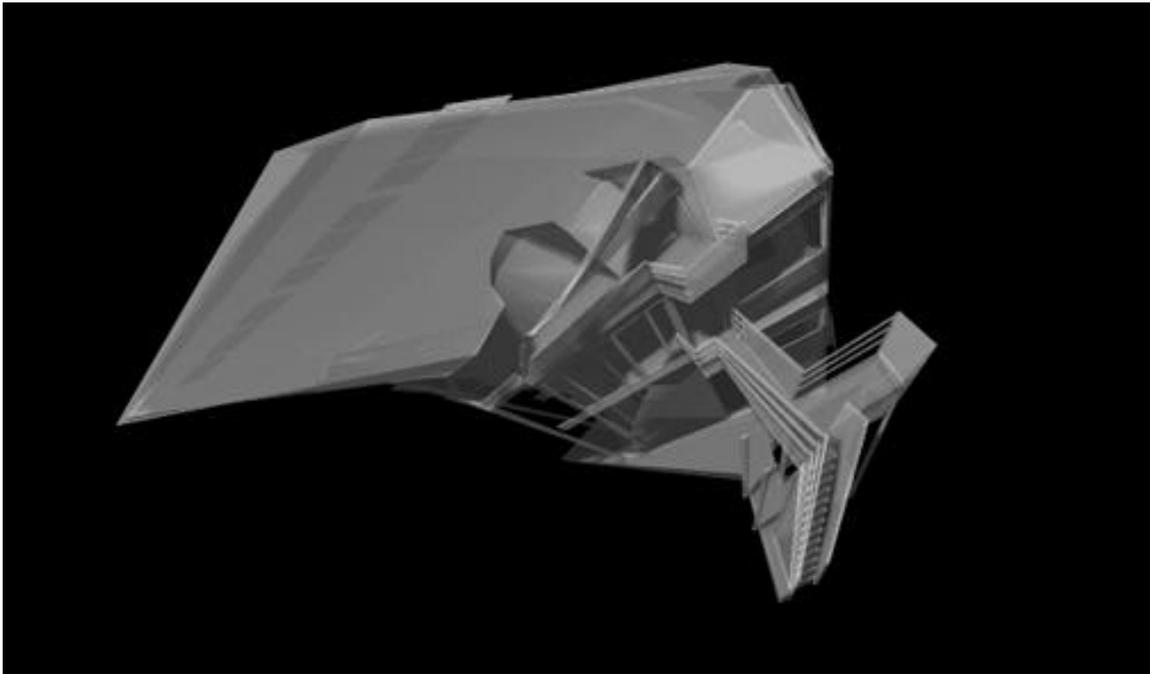


Figure 9. A trigonometric mapping on a trigonometric projection

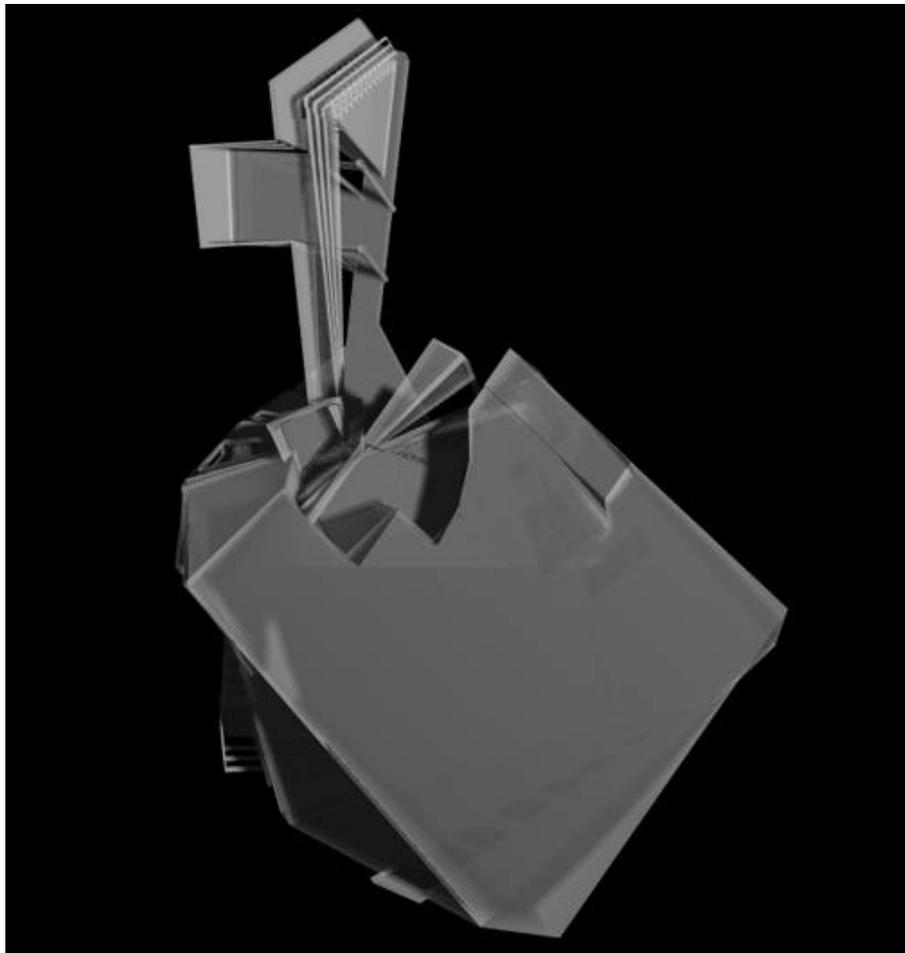


Figure 10. An exponential mapping on an exponential projection

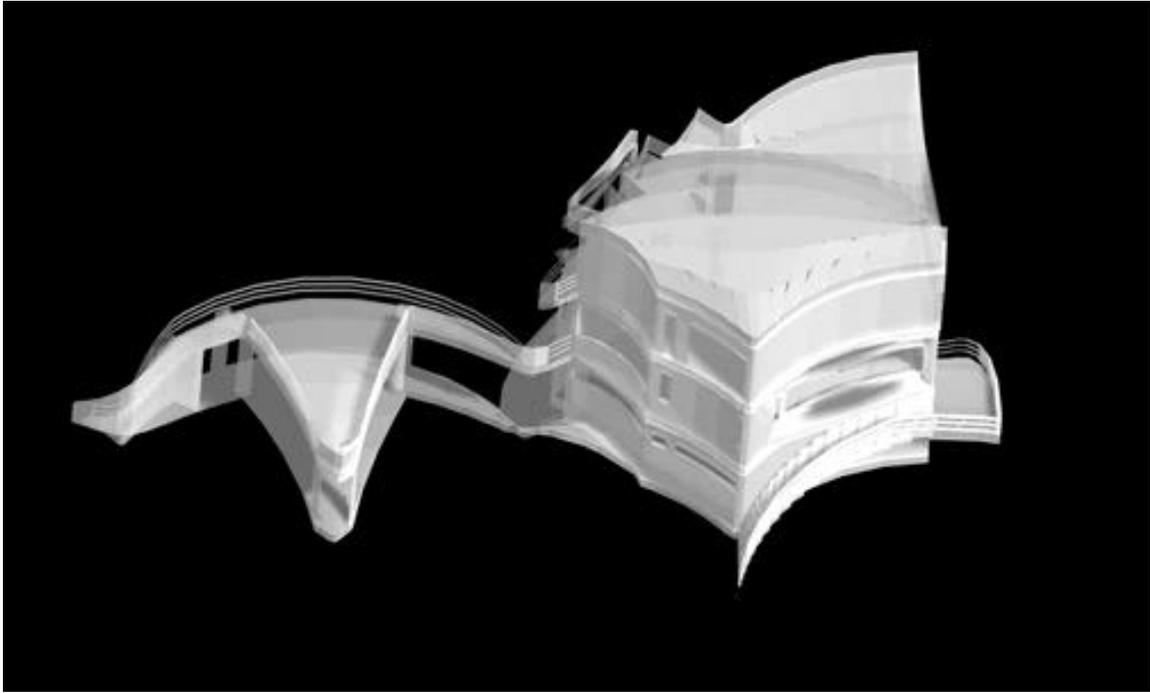


Figure 11. An exponential mapping on an exponential projection

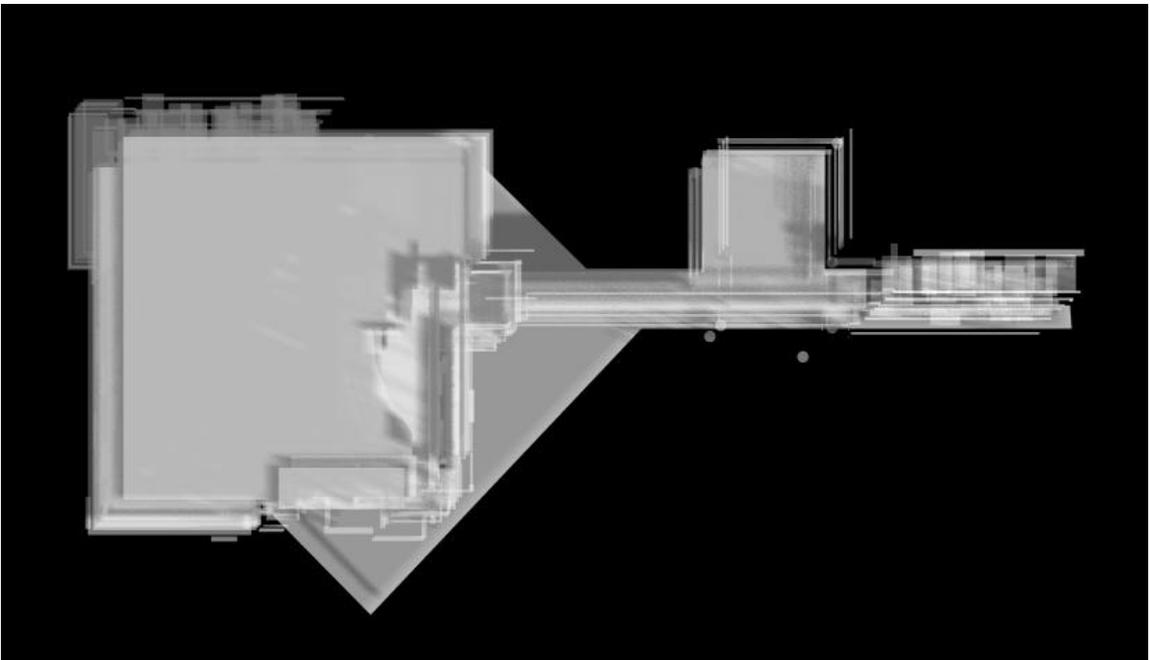


Figure 12. Random Disturbance

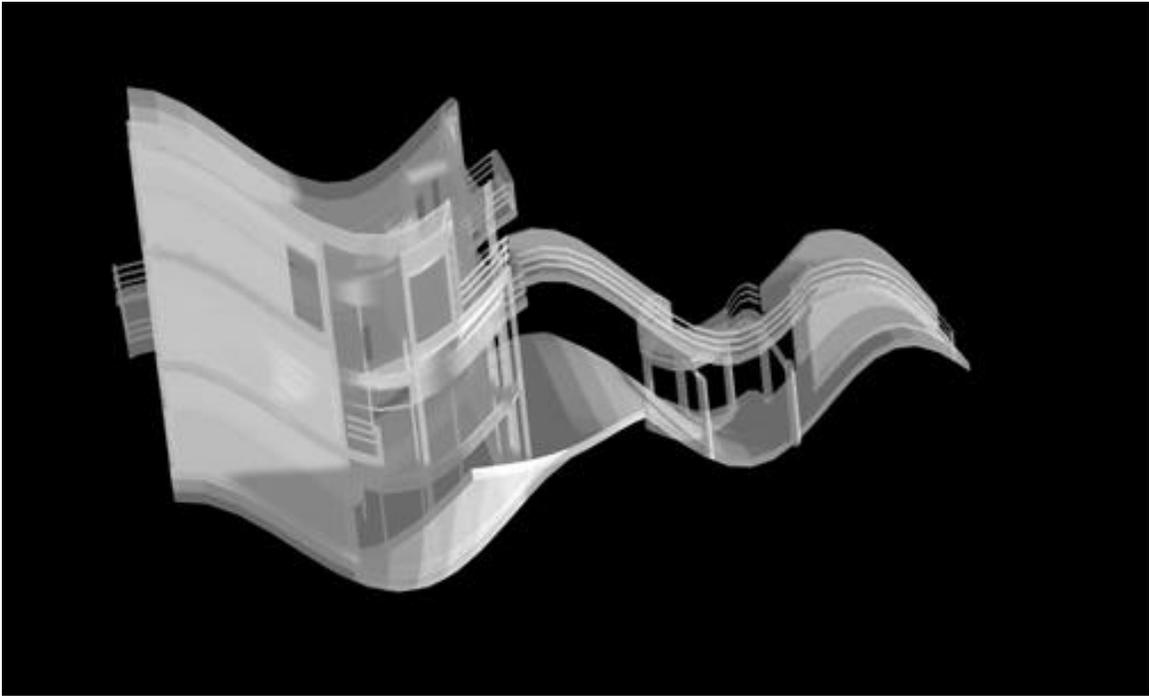


Figure 14. A sine-based disturbance

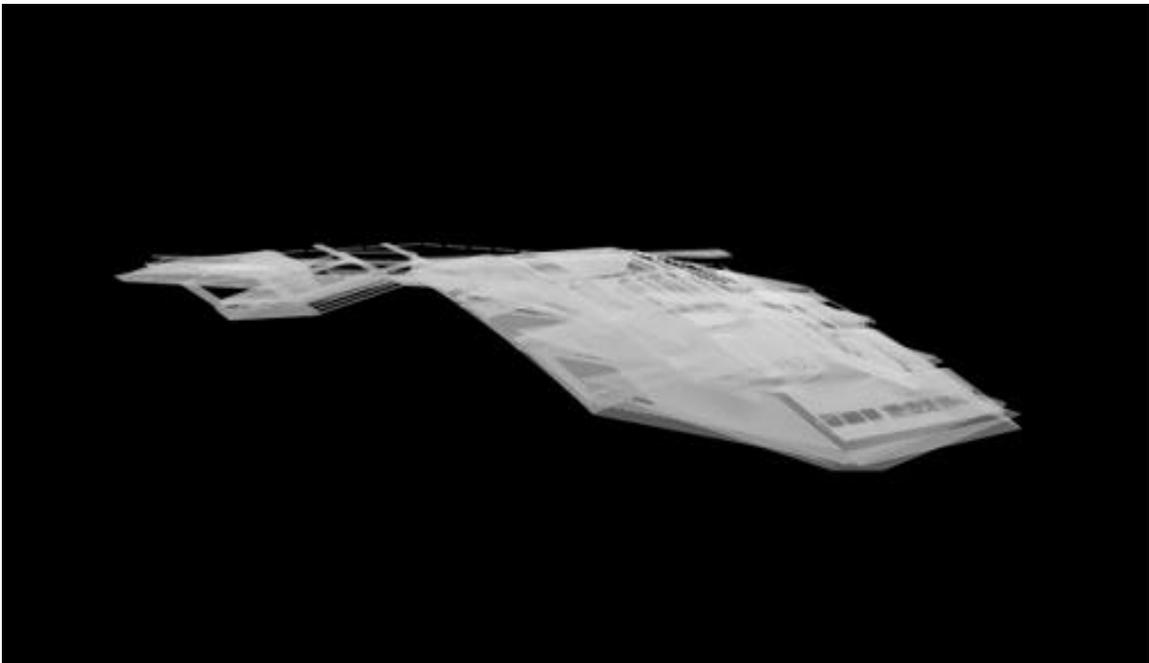


Figure 15. A spherical mapping on a spherical projection

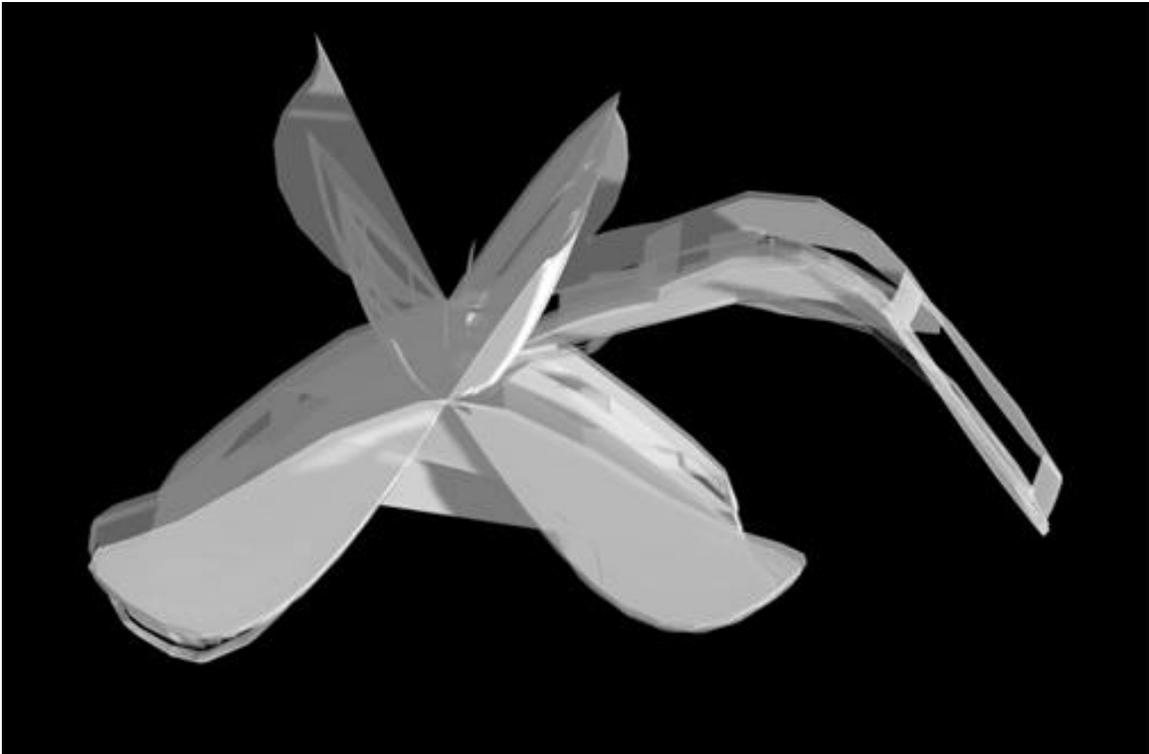


Figure 16. A hyperbolic mapping on a hyperbolic projection

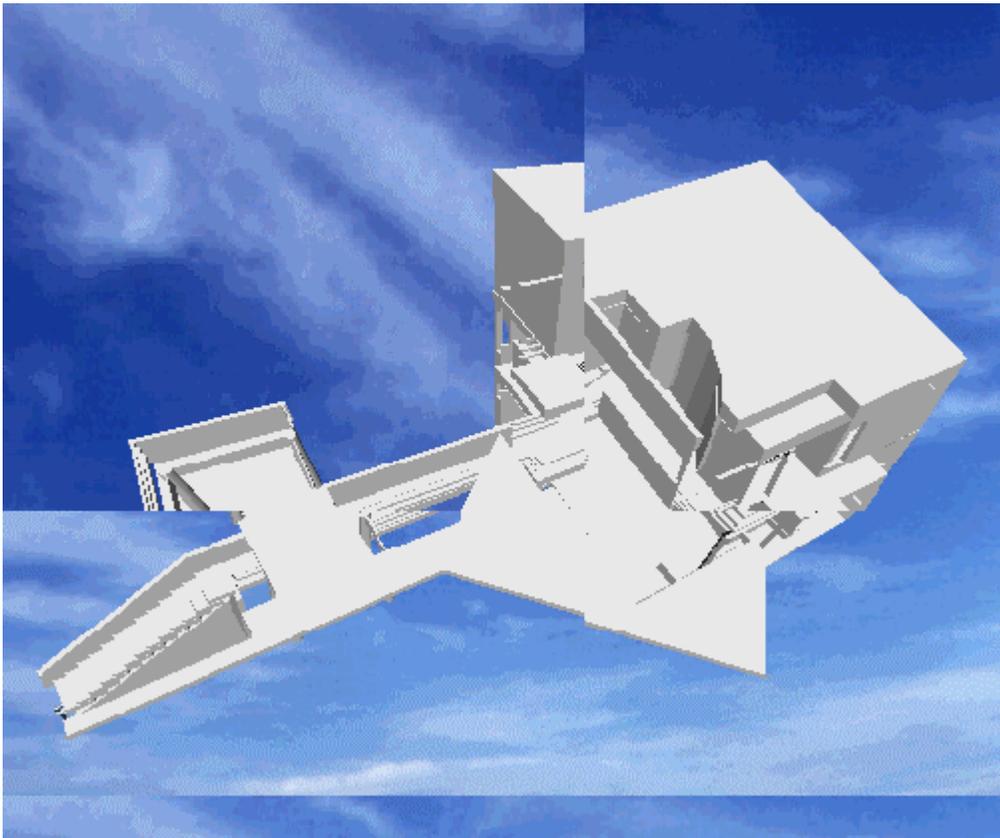


Figure 17. An alteration of perspective view parameters on different portions of the screen

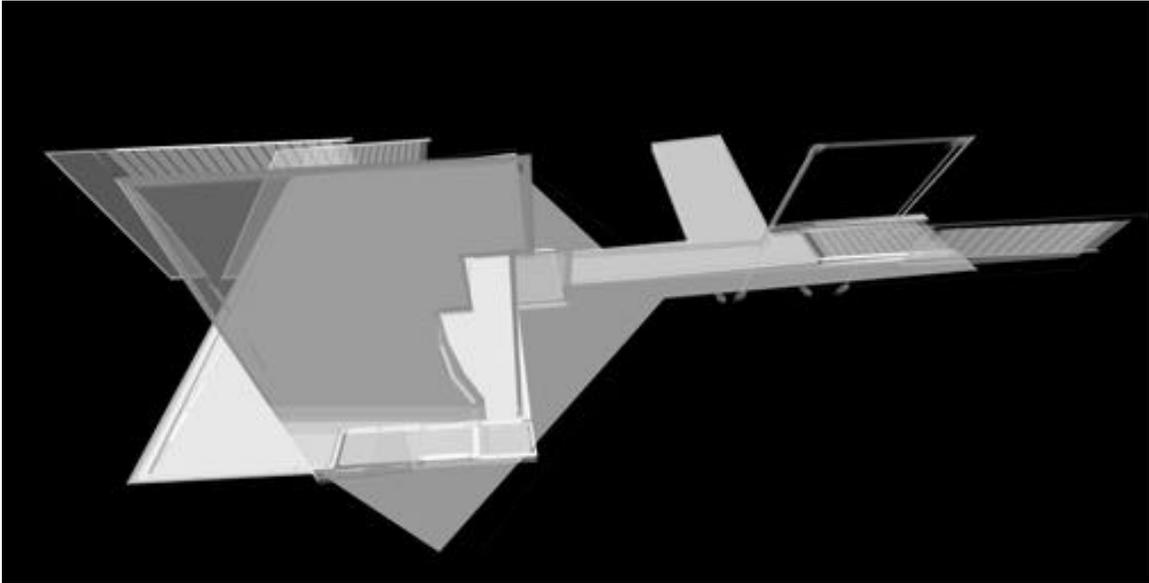


Figure 18. An alteration of perspective view for every face on the object

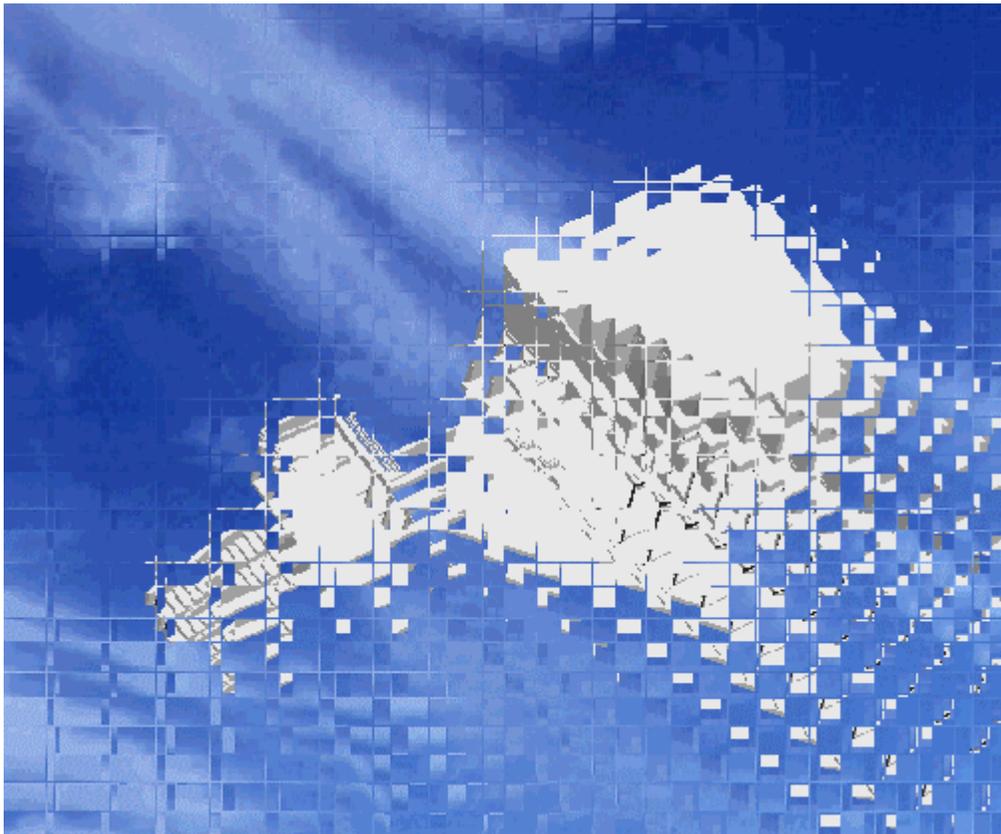


Figure 19. A grid-like filter

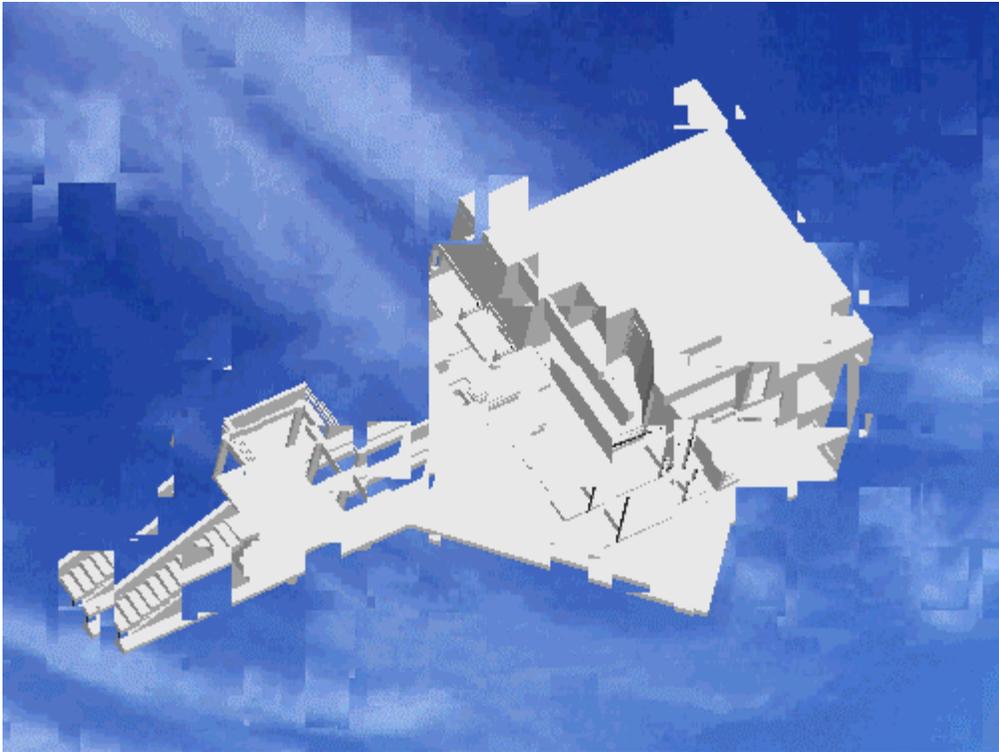


Figure 20. A random size square-based filter

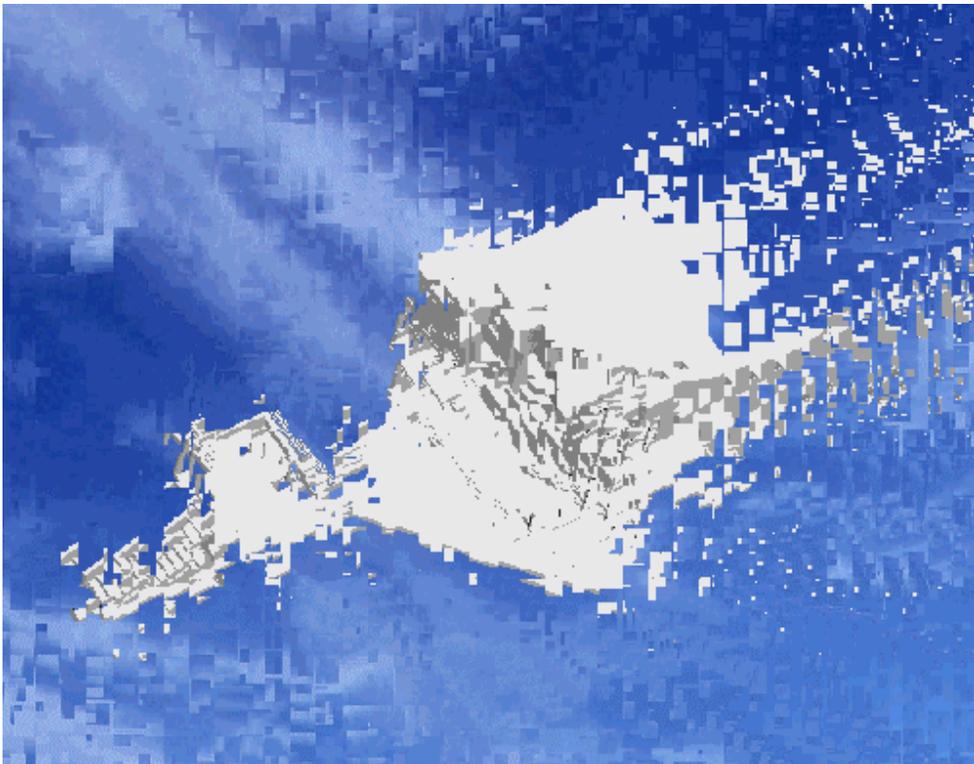


Figure 20. A random size small square-based filter