Modeling and Rendering Escher-Like Impossible Scenes

Guillermo Savransky Dan Dimerman Craig Gotsman

Department of Computer Science
Technion – Israel Institute of Technology
Haifa 32000, Israel

Abstract

Inspired by the drawings of "impossible" objects by artists such as M.C. Escher, we describe a mathematical theory which captures some of the underlying principles of their work. Using this theory, we show how impossible three-dimensional scenes may be modeled and rendered synthetically.

1. Introduction

"A likely impossibility is always preferable to an unconvincing impossibility"
-Aristotle

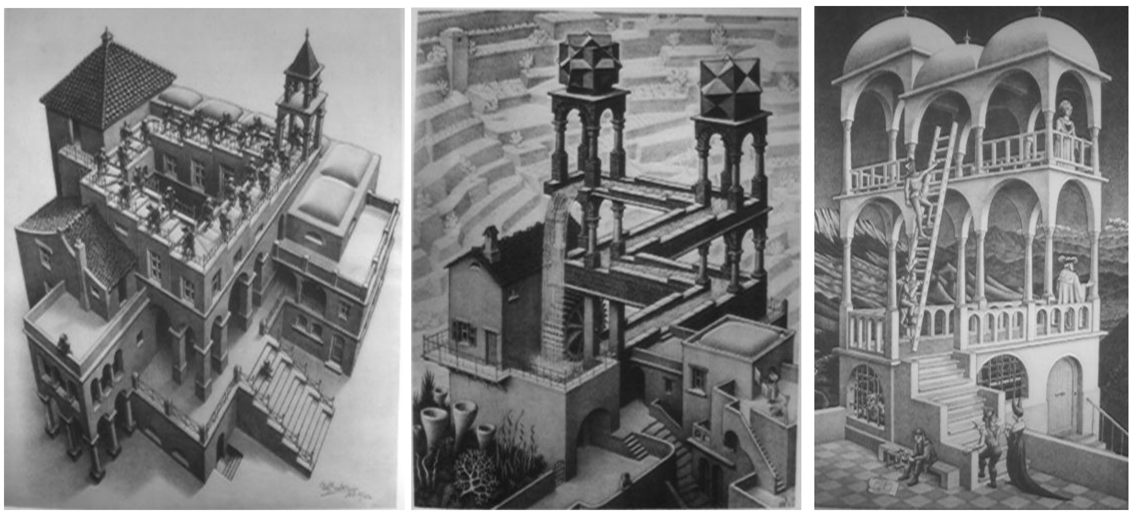
The artist M.C. Escher was not a mathematician, but it is commonly believed that his work was based on sound mathematical principles [1]. Some of his most renowned pictures, e.g. "Waterfall", "Belvedere" and "Ascending and Descending" (see Fig. 1) depict various impossible three-dimensional scenes. These scenes cannot be constructed in their "global" entirety in reality, but yet small "local" portions of the scene do not seem to contradict basic Euclidean geometric laws when viewed individually.

While M.C. Escher is probably the most well-known creator of impossible scenes, drawings such as these date back to Brueghel's 1568 painting "The Magpie on the Gallows". Subsequent works are Duchamp's in 1916 and, more significantly, Reutersvard's in 1934. See the survey of Ernst [2] for a full historical account. The famous "tribar" (see Fig. 2a) was discovered by Reutersvard, and later again by the Penroses [3], when it was first described in a scientific publication. Following correspondence between R. Penrose and M.C. Escher, it later featured in Escher's "Waterfall". Escher's "Belvedere" also contains the impossible "cuboid" (Fig. 2b).

That such impossible scenes may be rendered consistently in one image is due to the fact that the geometric contradictions may be "eliminated" when the scene is viewed from a particular viewpoint in a particular direction, and then projected onto the two-dimensional image plane. When looking at small portions of the result, we have no problem understanding it and reconstructing it in our minds. However, looking at the whole picture leaves us confused. While not exactly what will be advocated in this paper, Fig. 3, due to Ernst, demonstrates how an illusion of an impossible tribar may be created by photographing a real 3D object from a particular angle.

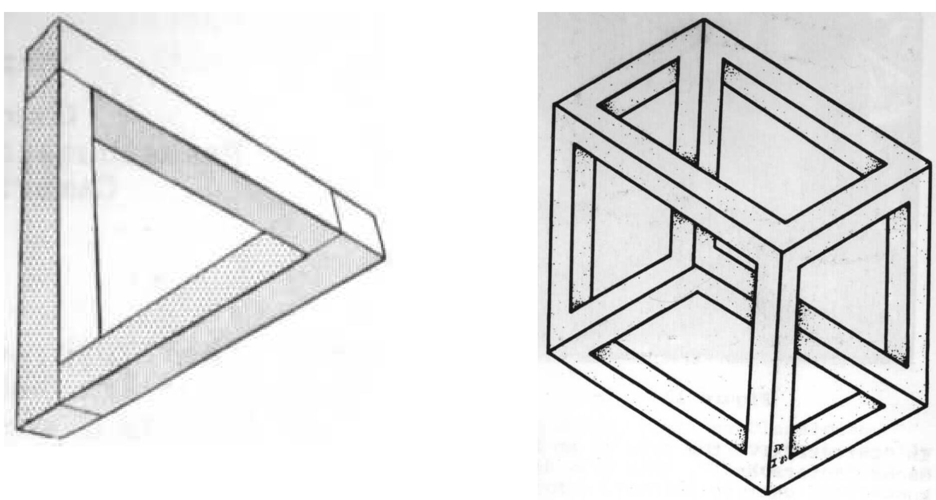
Exploiting the dimension reduction within the projective rendering operation is a way to explain Escher's drawings, but not all impossible pictures are based on these principles. Penrose [4] calls drawings guided by these principles *pure*, in contrast to *impure* pictures that are based on other visual phenomena. Impure images generally take advantage of simple drawing techniques, such as the simplicity of lines, to achieve the impossibility effect (see Fig. 4). From our point of view, impure images are less interesting because they cannot be shaded correctly while still maintaining the impossibility effect. Pure impossible drawings contain "real" objects (such as bars), each of which has an unambiguous consistent interpretation. They can be texture mapped or subjected to any realistic optical effect, hence may be rendered using computer graphics techniques.

In this paper we describe a mathematical theory which captures the intuitive visual illusions present in pure impossible images. A modeling language which enables the description of impossible scenes is proposed and a rendering algorithm which "solves" for a correct viewpoint and image is described. Apart from [5], in which an animation of an Escher-like "Belvedere" was shown, to the best of our knowledge this is the first proposal of an automated method to generate synthetic images of impossible scenes.



(a) **(b)** **(c)**

Figure 1: M.C. Escher's original drawings: **(a)** "Ascending and Descending". **(b)** "Waterfall". **(c)** "Belvedere". Copyright Escher's works : © M.C. Escher Heirs c/o Cordon Art, Baarn, The Netherlands.



(a) (b)

Figure 2: Classic pure impossible scenes: (a) Tribar. (b) Cuboid. Note that the scenes consist of "real" bars.

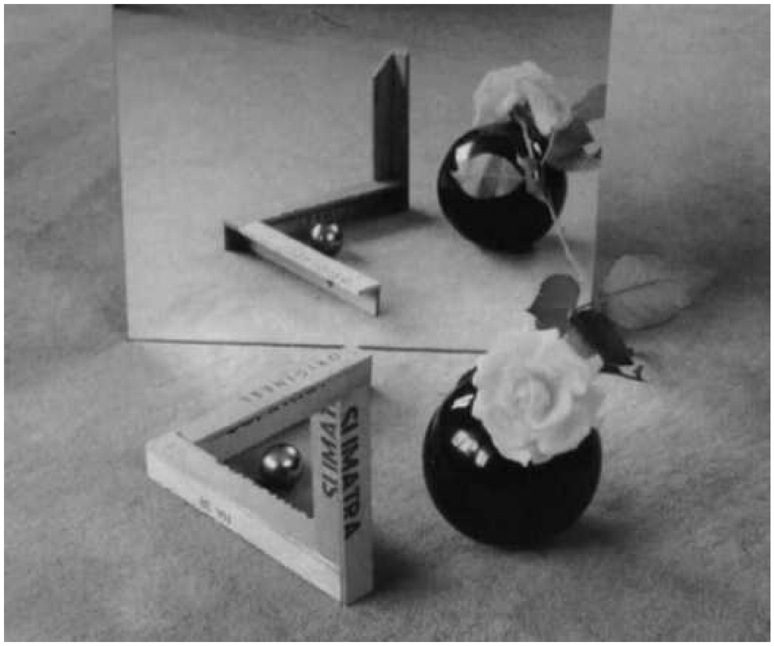


Figure 3: Photograph by Ernst demonstrating how an illusion of an impossible object may be created from a real 3D object (reflected in mirror).

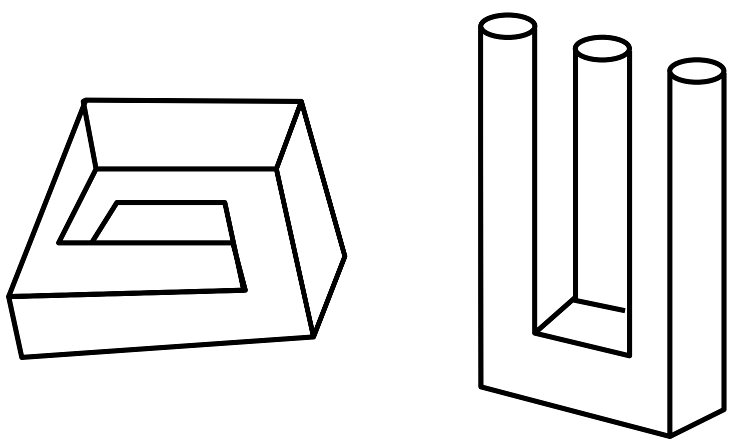


Figure 4: Impure impossible scenes. Note that no "real" physical objects are discernible.

2. Some Mathematics

This section provides some mathematical preliminaries which will prove useful in modeling and rendering impossible scenes. We assume all vectors are row vectors.

Definition: Two linear transformations A and B *are equivalent with respect to (w.r.t.)* P, denoted by $A \approx B$ (P), if AP=BP.

Equivalence w.r.t. P can occur for non-singular $A \neq B$ only if P itself is singular. Indeed, we will use this equivalence relation for projective matrices P. In computer graphics terms, equivalence w.r.t. P means that the same image of a scene results if it is viewed transformed by A and then projected thru P, or transformed by B and then projected thru P.

Definition: A transformation Z is called an *identity with respect to (w.r.t.)* P if for every transformation A: AZ ≈ A (P).

Lemma 1: Z ≈ I (P) ⇔ Z is an identity w.r.t. P. (I is the regular identity matrix).

Proof: One direction is trivial. The other direction follows from:

Z ≈ I (P) ⇒ ZP = IP = P, so for any A, AZP = AP ⇒ (AZ)P = AP ⇒ AZ ≈ A (P).

The next two theorems characterize identities w.r.t. P. The nullspace of a matrix A (denoted null(A)) is the rectangular matrix whose rows are a basis for the orthogonal complement of the linear space spanned by the columns of A. A^T denotes the transpose of A.

Theorem 1: Z is an identity w.r.t. P iff span(Z-I) ⊆ null(P) or span(P^T) ⊆ null($(Z-I)^T$).

Proof: Z is an identity w.r.t P ⇔ Z ≈ I (P) ⇔ ZP = P ⇔ (Z-I)P = 0 ⇔ span(Z-I) ⊆ null(P) or span(P^T) ⊆ null($(Z-I)^T$).

Theorem 3: If Z is an identity w.r.t. P, and A commutes with P, i.e. AP=PA, then:
(One) $A^{-1}ZA$ is an identity w.r.t. P.
(Two) Z is an identity w.r.t PA.

Proof: Z is an identity w.r.t P ⇒ Z ≈ I (P) ⇒ ZP = P ⇒ ZPA = PA ⇒ ZAP = AP. This in turn implies that:
(a) $A^{-1}ZA$ P = P ⇒ $A^{-1}ZA$ ≈ I (P) ⇒ $A^{-1}ZA$ is an identity w.r.t. P.
(Two) Z ≈ I (AP) ⇒ Z is an identity w.r.t. AP.

Theorem 3 implies that identities w.r.t. P are not unique. For example, if P is an orthographic projection through the axis defined by unit vector v, Z an identity w.r.t. P, and A any rotation and scale transformation around v, then $A^{-1}ZA$ is also an identity w.r.t. P.

3. Modeling Impossible Scenes

In most geometric modeling frameworks, a *scene S* consists of a collection of *objects* O_i, each described in its own local coordinate system. The scene is defined by providing a geometric transformation for each of the objects into the global *world* coordinate system, positioning it relative to this fixed reference frame. In this work we generalize by allowing for transformations *between* pairs of objects in the scene, which we call *relations.*

The scene graph

Let $O = \{O_1,..,O_n\}$ be a set of objects, each described by its geometry. The *scene graph* is the directed graph G=(O,T), such that the graph vertices are the scene objects and the edge set are transformations T_{ij}, such that $T_{ij} = T_{ji}^{-1}$. The relation associated with T_{ij} is said to be *satisfied* if $O_i T_{ij} = O_j$. The graph is said to be satisfied if all its relations are satisfied. A scene is said to be *consistent*, or *possible*, if its graph is satisfied, otherwise it is said to be *inconsistent*, or *impossible*.

Describing impossible scenes

In classical scene description, where all objects are related to one global coordinate system, the scene graph has the topology of a star. One object, the center of the star, may be viewed as a global reference, to which all other objects are directly related through some direct transformation.

The key to being able to describe impossible 3D scenes is to describe them using relations between *any* pair of scene objects, instead of relating all the objects to *one* global reference frame. This allows for scene inconsistency, in the sense that not all of the relationships may be satisfied simultaneously. It is precisely this class of inconsistent scenes which are "impossible", and which seem to us unreal. However, in many cases, the use of appropriate viewing and projection transformations may eliminate these inconsistencies. This is where the concept of equivalence with respect to a projection, described in Section 2, proves to be useful.

Making impossible scenes possible

Assume the scene graph G=(O,T). In order to render this scene, the graph must be satisfied. If this is not the case, we require a weaker condition, that the graph be satisfied with respect to some projective viewing transformation V.

If the scene graph does not contain cycles, it is trivially satisfied by any V. The existence of the cycle $O_{i_1} \rightarrow O_{i_2} \rightarrow \ldots \rightarrow O_{i_k}$ implies that $T_{i_1 i_2} T_{i_2 i_3} \cdots T_{i_{k-1} i_k} T_{i_k i_1} = I$ must hold in order that the graph be satisfied. This may not always be the case, as happens for the class of *impossible* scenes. These impossible scenes, however, may be made "possible" if a weaker condition holds:

(1) $$T_{i_1 i_2} T_{i_2 i_3} \cdots T_{i_{k-1} i_k} T_{i_k i_1} \approx I \ \text{(V)}$$

Note that the trivial V=0 is always a solution, reducing the entire scene to a point image. To eliminate this solution, we require that V=BP, where P is a given projective viewing transformation, e.g. the orthographic or perspective transformation, and B is an unknown non-singular matrix. The unknown is, therefore, B. Once a V=BP satisfying (1) for all cycles of G has been found, the actual viewing transformation for each object is determined by calculating it for one object, and then propagating it through the graph.

Solving impossible graphs

Given a viewing transformation P, solving an impossible graph requires finding a transformation B such that (1) holds for V=BP and all cycles in the graph. This may be done by starting with an arbitrary object O, and accumulating (composing) the inter-object transformations through the graph. If O is reached again, and the accumulated transformation is Z, we require that Z ≈ I (BP), or, by Theorem 1, span($P^T B^T$) $\subseteq$ null($(Z-I)^T$). Furthermore, Theorem 2 implies that the solution is not unique, since, if B is a solution, and A commutes with BP, AB is also a solution. Usually, a significantly different effect will not be obtained by using $A \neq I$, other than some image plane transformation.

Finding the nullspace of a matrix is a standard linear algebraic operation, usually performed via a singular value decomposition (SVD) operation. It is available in most numerical matrix manipulation packages (e.g. MATLAB).

Example

Assume that P is the orthographic projection, and that the original and final transformations in a graph cycle define transformations of the canonical axes to axes with parallel XY planes and co-linear Z axes. It is easy to see that a possible solution B corresponds to a rotation mapping the vector connecting the origins of the object spaces to a vector parallel to the canonical Z-axis. The solution BP generates the family of solutions ABP, where A is a rotation of the xy "image" plane. See Fig. 5.

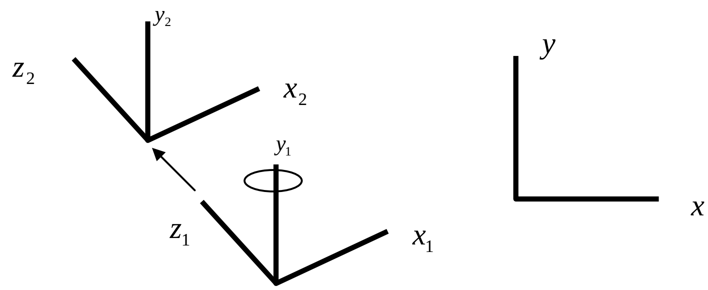


Figure 5: The viewing transformation which bridges the gap between an object's original position and its position after propagation through a scene graph cycle. The rotation aligning the vector v with the canonical Z axis, composed with an orthographic projection thru the Z axis. This may also be rotated afterwards in the xy plane.

4. Rendering Impossible Scenes

To render an image of a given scene involves determining the RGB color of every image pixel. The dominant contribution to a pixel is that from the object closest to the viewer projected on that pixel. In conventional rendering scenarios, determining which object this is requires knowing and ordering the depths of all objects projected to the pixel in the global coordinate system. Since, for impossible scenes, a global coordinate system does not necessarily exist, the rendering of these scenes is first done on a per-object basis. This allows the use of external renderers to calculate the color and depth for every pixel and for every object in its own coordinate space, using its accumulated viewing transformation, producing a separate image for each object. The output of such an external renderer is an image containing a color and a *local* depth for each pixel. Local depth means a depth in the object's local coordinate system. To generate the final image these local depths must be sorted somehow by relating the coordinate systems of the various objects. This is possible for two objects if there is an edge in the scene graph between the two objects (provided by the user), otherwise there is no well-defined relationships between the different coordinate systems. This introduces extra degrees of freedom into the solution, and it is up to the user to specify occlusion relationships resolving these ambiguities.

The algorithm proceeds by rendering all pairs of objects, for which an edge exists in the scene graph, with a Z-buffer compositing algorithm. For pairs of objects for which the user has defined an occlusion relationship, an overwrite operation is used for the occluding object. For pairs of objects for which no relationship is explicitly defined, an arbitrary order is chosen, so long as it is consistent throughout the entire rendering.

For some pixels, a contradicting (or inconsistent) ordering of the pixels will arise. This happens, for example, with the "fat" tribar, whose three bars are so fat that they touch in the center. This center point is unrenderable. This phenomenon seems to be a generalization of the situation arising when trying to render two co-planar polygons with traditional rendering algorithms (the question of who occludes who then arises).

5. Examples

In this section we bring two examples of impossible scenes modeled and rendered using our system, where P is the canonical orthographic projection. The input to the system is a geometric description of each of the 3D scene objects in some local coordinate space, and a set of geometric transformations between some pairs of the objects. The system then solves for the non-singular transformation B, such that the scene graph is consistent w.r.t. BP. The system renders each object separately in in its local coordinate space, and builds an image and depth map for it. It then presents the user with a list of occlusion degrees of freedom which he may set in order to render an image of the scene. The user may experiment with various relationships until a pleasing image is obtained. The inputs for Figs. 6-7 are detailed. Fig. 8, demonstrating the basic motif in M.C. Escher's "Waterfall", was also obtained by our system.

Cool Cube

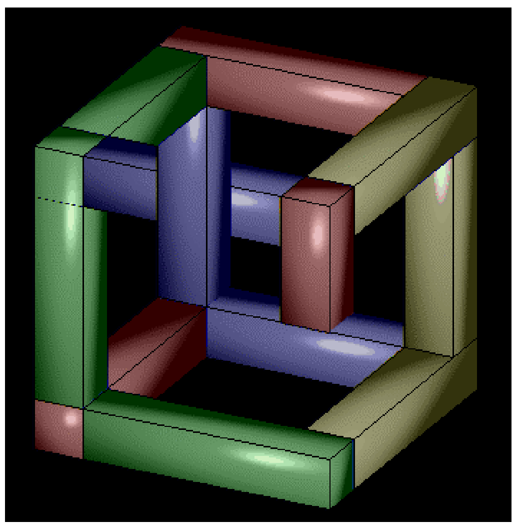


Figure 6: The impossible "Cool Cubes" scene.

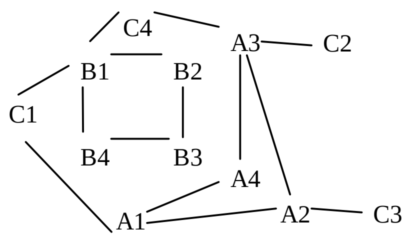
objects:

A1 C1 B4	"bar.3d";
B1	"bar.3d";
C4	"bar.3d";
A3 C2	"bar.3d";
B2 C3 A2	"bar.3d";
B3 A4	"bar.3d";

relations:

A1 C1	rotate x 90;	translate < 0 1 1 >;	
C1 B1	rotate x –90;	translate < 0 1 1 >;	
B1 C4	rotate x 90;	rotate y 180;	translate < 0 -1 1 >;
C4 A3	rotate x -90;		
A3 A4	rotate z –90;	translate < -1 1 0 >;	
A4 A1	rotate z –90;	translate < -1 1 0 >;	
A1 A2	rotate z –90;	translate < -1 1 0 >;	
A2 A3	rotate z –90;	translate < -1 1 0 >;	
B1 B2	rotate z –90;	translate < -1 1 0 >;	
B2 B3	rotate z –90;	translate < -1 1 0 >;	
B3 B4	rotate z –90;	translate < -1 1 0 >;	
B4 B1	rotate z –90;	translate < -1 1 0 >;	
A3 C2	rotate x 90;	translate < 0 1 -1 >;	
A2 C3	rotate x 90;	translate < 0 1 -1 >;	

scene graph:



occlusions: C2 after B1; C2 after A4; C2 before B4; C2 before B3; C1 after A4; C3 before B2;

Star of David

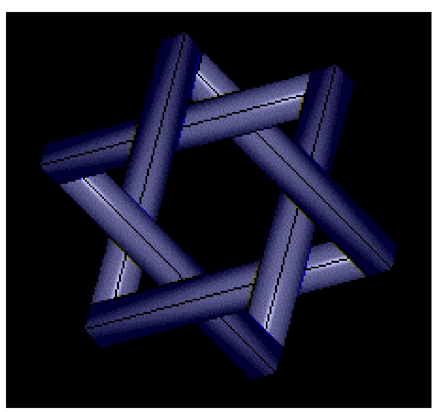


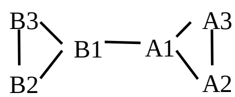
Figure 7: The impossible "Star of David" scene.

objects:
A1 A2 A3 B1 B2 B3: "blue_bar.3d";

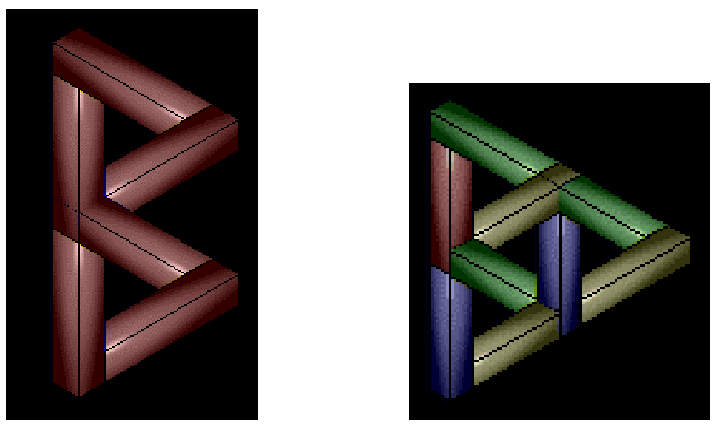
relations:

A1 A2: rotate z 90;	translate < -1 -1 0 >;	
A2 A3: rotate x –90;	translate < 0 -1 -1 >;	
A3 A1: rotate z –90;	rotate y 90;	translate < 0 -1 1 >;
B1 B2: rotate z 90;	translate < -1 -1 0 >;	
B2 B3: rotate x 90;	translate < 0 -1 1 >;	
B3 B1: rotate z –90;	rotate y –90;	translate < 0 -1 -1 >;
A1 B1: rotate y 45;	translate < 1 0 0 >;	rotate y -135;

scene graph:



occlusion: B2 after A2; B2 before A1; B3 after A1;
B3 before A3; B1after A3; B1 before A2;



(a) **(b)**

Figure 8: Synthetic impossible scenes: **(a)** "Waterfall" **(b)** "Super Waterfall".

6. Conclusion

We have shown that impossible scenes, such as those drawn by M.C. Escher, are actually quite possible, given the proper descriptive language and rendering mechanisms. The key to this is that not all geometric scene information is required to render an image of the scene, so some of it may be inconsistent, yet still yield a consistent rendering.

A mathematical theory, based on relations *between* scene objects was described to support our ideas. This formalism generalizes the traditional ways of describing scenes and allows for the description of impossible scenes. A method was proposed for the rendering of such impossible scenes, using a mixed Painter and Z-buffer algorithm.

While our system has proven useful in generate synthetic images of impossible scenes, it still requires considerable user intervention. It would be useful to reduce this to the bare minimum. What that bare minimum is remains an open question.

References

[1] D. Hofstadter, *Godel, Escher, Bach: An eternal golden braid*. Basic Books Inc., New York, 1979.

[2] B. Ernst, *Escher's impossible figure prints in a new context.* In H.S.M. Coxeter, M. Emmer, R. Penrose and M.L. Teuber, Eds. "M. C. Escher: Art and Science", North-Holland, 1986.

[3] L.S. Penrose and R. Penrose, *Impossible objects: A special type of visual illusion*. British Journal of Psychology, 1958.

[4] R. Penrose, *Escher and the visual representation of mathematical ideas.* In H.S.M. Coxeter, M. Emmer, R. Penrose and M.L. Teuber, Eds. "M. C. Escher: Art and Science", North-Holland, 1986.

[5] S. Tsuruno, *The Animation of M.C. Escher's "Belvedere"*. SIGGRAPH '97 Visual Proceedings pp. 237.