Take a Stroll in Another World

Zs. Ruttkay and E.M. Orbons

Dept. of Computer Science, University of Twente, The Netherlands

Abstract

Another World II is a print by the famous Dutch artist M. C. Escher, depicting a cube with arch-like openings on each side. What makes the print interesting is that the depicted views in the arches are inconsistent, leaving the spectator puzzled as to the "impossible assembly" of views from different perspectives. In our work we invite the reader – and more so, the user of our software – to explore the strange geometry of this work by flying through a virtual Another World. The rendering of the inconsistent views is achieved by applying Portal Rendering. We point out the novelty of our work with respect to previous 3D computer models of different "impossible worlds" by M. C. Escher. We also suggest potential application contexts beyond the straightforward 3D reconstruction of Another World.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Computer Modeling, Animation, Cultural heritage

1. Another world

Another World II (figure 1) [Ern76, esc07] is a print by the Dutch artist M. C. Escher. This intriguing work depicts a cube, with six architectural arches opening onto a moon-like landscape.

However, the views through the arches are inconsistent. One can derive the relative position of the cube and the moon based on for example what is seen through the left opening, but then the view through the bottom opening is not the one that is expected: instead of looking down to the surface with craters, we see the sky. It is easy to notice that there are three views altogether, and pairs of neighboring arches share a common view. These pairs are cleverly placed on two adjacent sides of the cube, using altogether five sides. The sixth side, facing the observer has been removed entirely, making it possible to see the views through the arches without obstructions. Escher solved the problem of arranging six arches on five sides by placing two narrow arches on the right side and a single, wide arch on each of the other four sides. The arches are drawn according to the outside view of the moon. This of course makes the cube a very strange architectural piece, which is especially apparent on the right side, where two narrow arches are aligned "with head to bottom" next to each other. One experiences a certain dizziness from observing the three inconsistent views squeezed into a single scene. On the other hand, the airy openings, the uninhabited landscape and even the statues of birds – different views of the same Persian statue, an object dear to Escher and depicted in more of his works – invite the viewer to "fly out" and wander around in this strange world.

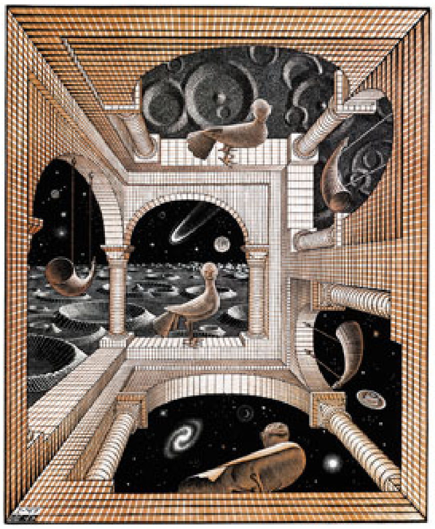

Figure 1: *Escher's Another World II (1947, woodcut)*

We could not resist. In the rest of the paper we invite the reader to take a stroll in a virtual, 3D version of the world depicted by Escher.

First we give an overview of work by others on creating computer graphics reconstructions of impossible worlds in prints by Escher, and relate our problem and approach to theirs. Then, in Section 3 we explain in detail the principle of our implementation of the world and the navigation in it. Finally, we ponder about the intention of Escher, and suggest some scenarios for using such inconsistent views in future applications.

2. Rendering impossible worlds in 3D

Before dwelling on the individual works, we analyze the nature of the impossibility of 3D scenes depicted in 2D in general. We should differentiate between 3D objects which:

(a) may exist in reality, can be physically built, however, they are different from what the viewer perceives when looking at their (specific) view shown on a 2D image;
(b) cannot exist; that is, there is no viewing position from which any 3D object would resemble the given 2D image.

An example of the first type is the *Penrose triangle* [PR58], a basic compositional element in Escher's *Waterfall*. For the second type, we refer to the *Devil's Pitchfork* [Sch64].

When creating 3D models for the first type of objects, the challenge lies in finding a 3D construct and a specific view and projection to produce the 2D image. Such endeavors probe the "3D imagination" of the designer. A further challenge is to be able to walk around the impossible construct in a virtual world in such a way that the trick, the real nature of the object, is not revealed, but the perceived impression is maintained from all views. It is clear that such a tour can be possible only in virtual reality, and some dynamical readjustment of the object is needed depending on the viewing position.

For the second type of objects, 3D modeling already requires some special rendering trick, not the usual linear perspective projection, in order to generate the 2D view of the impossible object. In this case too, animation is possible, but designing and maintaining how different views of a non-existing object would look is an additional intellectual and technical step.

Looking at works where computers have been used to create views of 3D reconstructions and animations of "impossible objects", we find examples of both types. Matthew Hibbs [Hib03] reconstructed Escher's *Endless Staircase* by using a two dimensional approach based on a orthogonal line drawing of the staircase. Consistent different views could be generated automatically. Savransky, Dimerman and Gotsman [SDG99] composed impossible worlds by using their own local coordinate systems for components and expressing their relations pairwise, resulting in a globally inconsistent scene graph in which the objects each get rendered separately. Occlusion relations - to be given explicitly by the user for the object pairs - are used to render the final image. For other examples and principles used to generate views of impossible scenes in Escher's works, see [Ale07].

Khoh and Kovesi [KK99] created animations of impossible objects, by maintaining "what it would be" different views. They animated real objects of complementary halves in such a way that their projected views could be "glued" together to create the illusion of *what a single object would look like* from different views.

Our problem with creating a virtual Another World could be considered as case (a). One could design a scene where the cube is surrounded by some odd 3D construct which, seen from a specific viewpoint, provides the views as depicted on the work by Escher. This may be a technically difficult, but intellectually not very challenging task. A better alternative is to interpret Another World by focusing on the transformed views, which gives rise to a new type of reconstruction task. Start with an ordinary 3D world, define several viewpoints, and develop a mechanism to display different views of this world to be shown in parallel, in a way which could never happen in reality. This may sound like an arbitrary goal, but one can find examples of this all through art history. From the half-profile half-frontal rendering of human figures in ancient Egypt to modern works, such as portraits by Picasso collated together from segments of different views of a woman's head. In our virtual Another World we have thus the task of maintaining three views of a consistent world, and showing these views through the arches. When wandering around, the views need to be updated, with respect to the actual viewing position of the navigator.

After having completed our work, the pioneering animation *Three Worlds* by Ned Greene was brought to our attention [CEPT86]. Greene saw the same possibility of flying out through the arches to the landscape. He created a computer animation along a fixed camera trajectory path, looping in and out through different arches. Along the path, the camera's view was continuously banked between leaving and entering different arches, making the illusion of continuous movement connecting the inconsistent views. Our work not only uses another, more recent computer graphics technique, but the navigation also has more features. First of all, in our case free navigation along any path is possible. Secondly, we maintain the views through the arches while being outside of the cube structure, by updating the through views from any camera position. Finally, we discuss our technique in general terms, making it applicable to use other inconsistent views than the ones in Escher's work.

3. Another world in 3D

3.1. Portal cells and their views

The construction of the cube-like element was made by using the Constructive Solid Geometry [WP01] technique. Portal rendering [WP01] was used to generate the views visible through the arches. Using this technique a scene is divided into different cells that are connected using portals. When the system encounters a visible portal during the rendering process of the cell it traverses into the next cell while limiting the visible area of the cell to the area occluded by the portal. Cells that are connected through portals need not be connected physically. In our case, the different cells correspond to different views of the same landscape. Knowing the transformations between them is sufficient basis to create complete scenes that are consistent from the current viewpoint, yet are physically impossible.

In Another World, an outside and an inside view of the structure exists and the portals that lead from the inside to the outside (and vice versa) are connected to parts of the structure in a seemingly inconsistent fashion. Portals are represented in the scene by thin (invisible) primitives, included in the definition of a cell. One of the primitive's faces is considered the "front" of the primitive and is used as the actual portal. Each portal is assigned a local coordinate system whose orientation matches the orientation of the original portal primitive. When two portals, which are to be connected, that do not share the same location and orientation are paired together, a transformation (consisting of a translation and rotation) is calculated between the two coordinate systems that are defined for the portals.

Three vectors are created by subtracting the center C of the portal primitive from the centers of the *front*, *top* and *right* faces of the portal (assuming that the portal has some small thickness) and normalized. These vectors are used to construct two (orthonormal, given that the portals are rectangular) matrices P_1 and P_2, one for each portal that is part of a matched pair. The rotation R from the first to the second portal is then calculated as $R = P_1^{-1}P_2$, rotating around C_1. One translates from the first portal to the second by using the vector T, calculated as $T = C_2 - C_1$.

The transformations are used for the following purposes:

- When rendering the geometry behind the portal, the transformation that transitions between the two portals in the correct direction is first applied to the current camera position to make it look like as if the two portals are actually connected.
- When the camera moves through a portal the transformation is applied to the current viewing position and orientation. This allows the viewer to move through the portals smoothly, again, as though the portals were connected.

Portal primitives are created by our software and translated and rotated to coincide with the arches in the cube. This is done for both the inside and outside of the cube, each portal on the inside of the cube is paired up with a corresponding portal on the outside. Currently the pairing of portals is fixed, but this process could easily be randomized or made configurable by the user.

3.2. Rendering

The world is rendered using OpenGL by processing visible cells in a depth first recursive manner. Rendering starts with the cell that contains the camera. After rendering each cell the algorithm propagates outwards through visible portals. The geometry of each cell is rendered before testing the portals so that the depth buffer is correctly initialized. After rendering the geometry each portal in the cell is processed in arbitrary order. Each portal is rendered with hardware visibility testing turned on, this operation returns the number of visible fragments that would be generated by the render operation. If the fragment count is non-zero, then the portal is considered visible and the geometry behind the portal is rendered. Before rendering a cell through a portal, the portal is masked using the stencil buffer to ensure that only pixels that lie on the portal can be rendered. In order to create the stencil mask for the portal, the portal is rendered again and pixels that would have been written to the screen are masked in the stencil buffer. After rendering the geometry behind the portal, the stencil buffer is cleared again. To make sure that only geometry *behind* the portal gets rendered, a custom clip plane is temporarily added to the frustum that coincides with the portal plane. Finally the portal's transformation is applied to the current modelview matrix and the rendering process steps into the cell behind the portal.

There is a special case in which the camera is so close to a portal that the near clipping plane of the frustum lies partially behind the portal and the camera is in front. In this case the portal is not fully rendered and it is therefore not possible to completely mask the area occupied by the portal in the stencil buffer. In order to work around this problem, the stencil mask is not used when rendering the geometry behind portals that lie close to the camera.

4. Discussion

We made a demo with the "outside" world as a landscape containing three distinct cubes, connected by some common edges. Figures 2 and 3 show an inside and an outside view rendered by the demo system. The user can move freely through this world, and the camera transition updates smoothly the inconsistent views through the arches as they were part of a real 3D world. The sense of confusion is greatest when entering the cube from the outside: it is hard to predict what the facing view will be after entering the cube. The simulation runs at interactive speed on an average consumer PC using a recent graphics card.

We deviated from the original design in Another World II, and opted for using three cubes that are positioned close

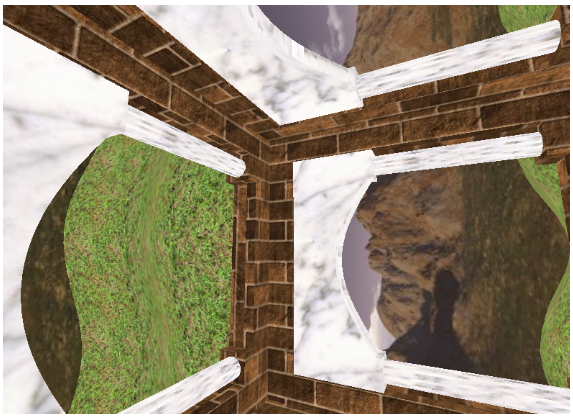

Figure 2: *Rendering of the virtual Another World (inside view)*

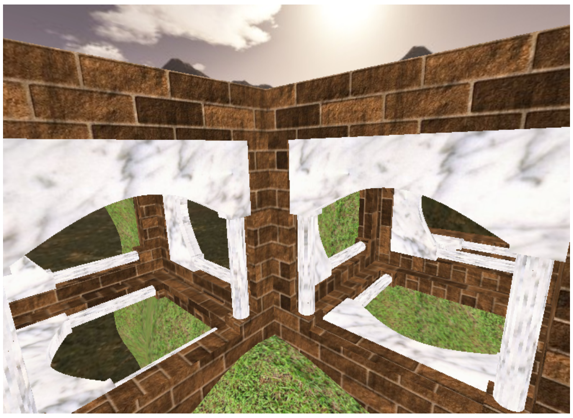

Figure 3: *Rendering of the virtual Another World (outside view)*

together in the same hilly landscape to enhance the sense of confusion. The user can enter the structure through an arch in one of the cubes and exit through another arch to end up in a position that is close to the starting point, yet totally unexpected. Other arrangements of the scene are also feasible and one could easily experiment with various configurations using the software.

One may ponder about the intention of Escher with Another World II. Bruno Ernst [Ern76] discusses how the print was created as an improvement of a first version. He looks at this print as one of several in the genre of more inconsistent views glued together. He also recalls that Escher's intention was to demonstrate the effect of swapping the up and down views: the central point is zenith for the view through the bottom, and nadir for the view through the top arches. While developing our animated 3D version, we wanted to draw the attention to a magic world where one can look out of different windows seeing arbitrarily chosen views of the same surrounding. Our interpretation is a generalization of what Escher had in mind. This interpretation could inspire applications where such a case is a real feature. First of all, our technique could be applied in the entertainment sector to create impossible 3D mazes, or computer games where the player can navigate in a paradoxical scene. Another possible domain is to use the principle in visualization, allowing the user to browse a (real) 3D world in space *and* time. Think of standing in a room in your home, and depending on which window you look (and fly) out of, you find yourself in the town where your house is, but at different historical ages or seasonal times. Such a visualization may sound bizarre, but would be exploited for interactive applications with entertainment and/or educational purposes.

Acknowledgements

We are indebted to Bruno Ernst for his useful comments on Escher's works and for his reference to Greene's animation, and to Lynn Packwood for improving the presentation. The paper was written in the framework of the Dutch national GATE project on gaming and simulation.

References

[Ale07] ALEXEEV V.: Impossible world: Art: Rendered Escher. As visited in April 2007, 2007. http://impossible.info/english/art/rendered_escher/index.html.

[CEPT86] COXETER H. S. M., EMMER M., PENROSE R., TEUBER M.: *M. C. Escher: Art and Sceinece*. North-Holland, 1986.

[Ern76] ERNST B.: *De toverspiegel van M. C. Escher*. Meulenhoff Nederland, 1976.

[esc07] The M.C. Escher company: The official M.C. Escher website, 2007. http://www.mcescher.com.

[Hib03] HIBBS M.: Rendering the "endless staircase" from multiple viewpoints, 2003. http://www.cs.princeton.edu/ mhibbs/class/cs526/ Escher.doc.

[KK99] KHOH C., KOVESI P.: Animating impossible objects. http://www.csse.uwa.edu.au/ pk/Impossible/ impossible.html.

[PR58] PENROSE L. S., R. P.: Impossible objects: A special type of visual illusion. *Brittish Journal of Psychology 49* (1958), 31–33.

[Sch64] SCHUSTER D.: A new ambiguous figure: A three-stick clevis. *American Journal of Psychology 77* (1964).

[SDG99] SAVRANSKY G., DIMMERMAN D., GOTSMAN C.: Modeling and Rendering Escher-Like Impossible Scenes. *Computer Graphics Forum 18*, 2 (6 1999), 173–179.

[WP01] WATT A., POLICARPO F.: *3D Games: Real-time Rendering and Software Technology*. Pearson Education Limited, 2001.