Hollow Mazes

M. Oskar van Deventer

In October 1983 I discovered hollow mazes. A. K. Dewdney presented the concept in *Scientific American*, September 1988. In this article I shall give a more detailed description of the subject. The first part is about *multiple silhouettes* in general, the second part is about their application to *hollow mazes*.

Multiple Silhouettes

Silhouettes are often used in, e.g., mechanical engineering (working drawings) and robotics (pattern recognition). One particular mathematical problem reads: "Which solid object has a circular, a triangular, and a square silhouette?" The solution is the well-known *Wedge of Wallis* (Figure 1).

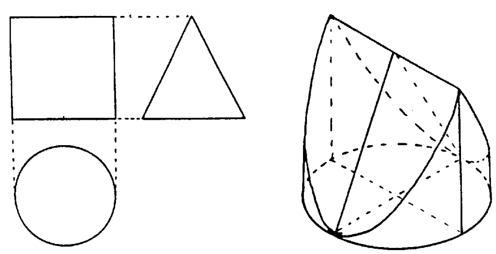

Figure 1. Wedge of Wallis.

A silhouette defines a cylindrical region in space, perpendicular to the silhouette surface. With this meaning the word no longer refers to an object, but only to a region of a flat surface (Figure 2).

Two or more silhouettes (not necessarily perpendicular to each other) define unambiguously a region or object in space that is the cross-section of two or more cylinders.

Dewdney introduced the term *projective cast* for an object defined by a number of silhouettes. Not every object can be defined as the projective

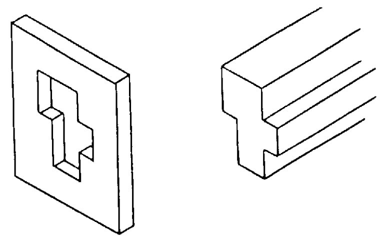

Figure 2. (a) Negative of a silhouette; (b) The cylindrical region in space defined by it.

cast of a (limited) number of silhouettes: One cannot cast a hollow cube or a solid sphere.

There are two interesting topological properties that apply to silhouettes, to projective casts, and to mazes and graphs in general. First, a graph may have either cycles or no cycles. Second, a graph may consist of either one part or multiple parts (Figure 3).

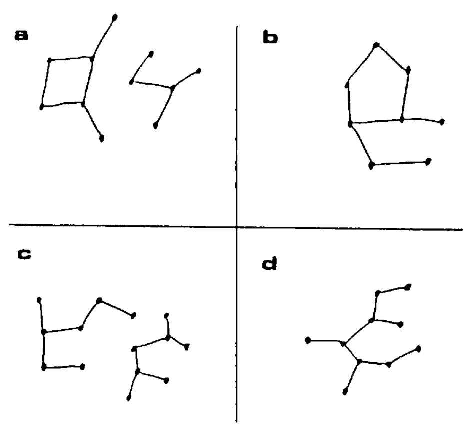

Figure 3. Topological properties of mazes and graphs: (a) multiple parts with cycles; (b) one part with cycles; (c) multiple parts, no cycles; (d) one part, no cycles.

Dewdney used the term *viable* for a silhouette or a maze that consists of one part and has no cycles. Even when silhouettes are viable, their projective cast can still have cycles (Figure 4) or multiple parts (Figure 5).

There is no simple connection between the properties of the silhouettes and the properties of their projective cast. The basic problem is the analysis

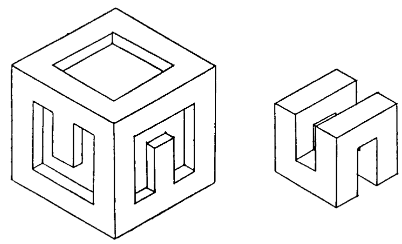

Figure 4. Three viable silhouettes yielding a projective cast with one cycle.

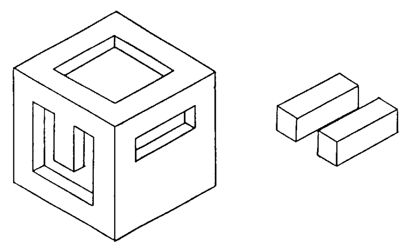

Figure 5. Three viable silhouettes yielding a projective cast with two parts.

and synthesis of silhouettes and projective casts: How can the properties of a projective cast be found without constructing it from its silhouettes, and how can a projective cast be given certain properties by its silhouettes? I do not know whether there is a general answer.

Hollow Mazes

A hollow maze is a rectangular box with six sides (Figure 6). Each side is a two-dimensional *control maze:* a surface into which slots have been cut. A cursor consisting of three mutually perpendicular spars registers one's position in the hollow maze. Each spar passes from one side of the box to the other, sliding along the slots of the control maze on each side. The two control mazes on opposite sides of the box are identical. In this way, a single three-dimensional maze is produced from three pairs of two-dimensional mazes. The resemblance between multiple silhouettes and hollow mazes is evident.

Control mazes are always viable because of their construction. There can be no cycles since the center of a cycle would fall out. Control mazes

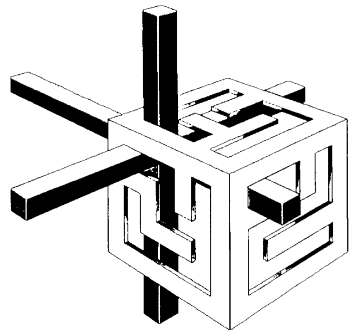

Figure 6. A simple hollow maze.

cannot consist of multiple parts, since a spar cannot jump from one part to another.

I have developed several restrictions for the investigation of hollow mazes:

- There are three (pairs of) perpendicular control mazes;
- the control mazes are viable;
- the control mazes are line mazes (i.e., their paths have zero width);
- a square or cubic grid is used for control mazes and their projective casts.

Now we can take another look at the topological properties of the projective cast. It seems that the projective cast cannot have cycles anymore, because the control mazes are line mazes. (I'm sure that there must be a simple proof of this, but I lack the mathematical tools.)

It can easily be tested whether or not a projective cast is viable, assuming that it cannot have cycles: We count the number of grid points of the projective cast. For example, a $3 \times 3 \times 3$ projective cast has 27 grid points. Then there must be exactly 26 "links" between these points for a viable cast. The number of links can be counted from the control mazes without constructing the projective cast.

Count the links in the hollow mazes of Figure 7. The dotted cross-section of Figure 7(a) has six links through it; the total number of links is found after taking all cross-sections.

There are several ways to construct a hollow maze, viable or not:

- **Trial and Error.** (Working at random.)

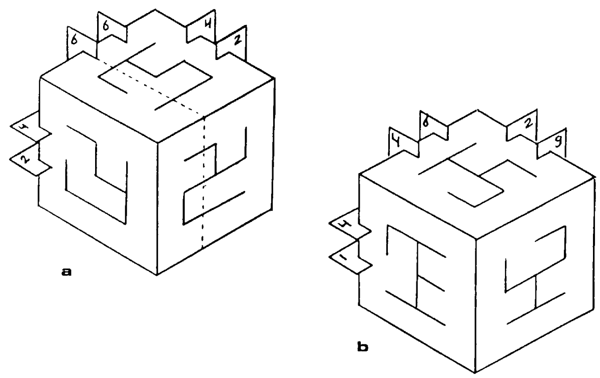

Figure 7. Test of the viability of a projective cast. (a) $2+4+6+6+4+2 < 27-1$; not viable. (b) $1+4+4+6+2+9 = 27-1$; viable!

- **Cut and Connect.** Cut the projective cast several times and link the pieces again (Figure 8). This method guarantees that the eventual hollow maze is viable. Helmut Honig, a German computer science student, showed me that not all viable hollow mazes can be constructed by this method. Figure 9 is a counterexample.

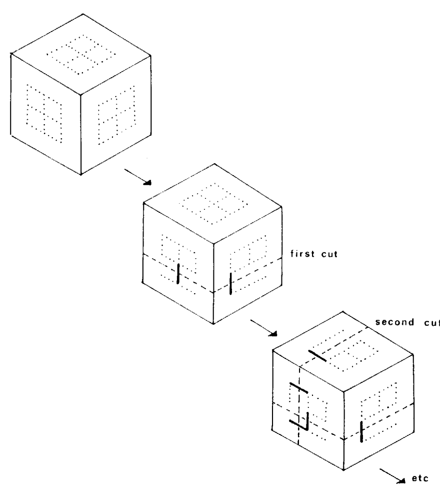

Figure 8. The cut and connect method for constructing hollow mazes.

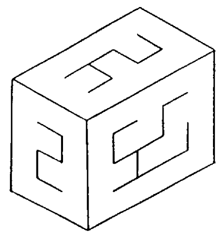

Figure 9. A viable hollow maze that cannot be constructed by the "cut and connect" method.

- **Regular Control Mazes.** See Figure 10. How should identical control mazes of this kind be placed to yield a viable hollow maze?
- **Recursion.** Replace each point by a hollow maze (see Figure 11).

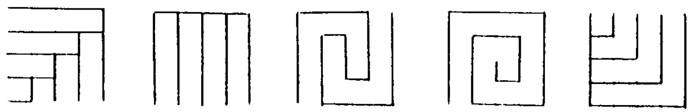

Figure 10. Regular control mazes.

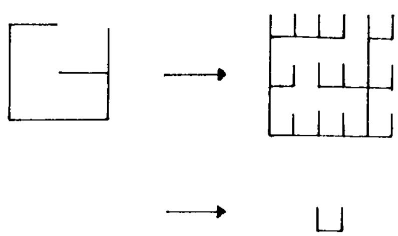

Figure 11. Construction of a hollow maze by recursion.

There is still much research to be done on multiple silhouettes and hollow mazes. I keep receiving letters from readers of *Scientific American*, and perhaps some of the problems mentioned above will be solved in the future.