Journal of Mathematics and the Arts, Vol. 00, No. 00, Month 200x, 1–16

Vortex Maze Construction

Jie Xu and Craig S. Kaplan*

David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, N2L 3G1, Canada

(*v1.0 released July 2006*)

The creation of engaging mazes requires both mathematical and aesthetic considerations. We present an algorithm for producing mazes that we feel are difficult to solve. This algorithm is based on constructing and connecting obfuscating maze devices called vortices. A vortex is a collection of passages that wind around each other in a spiral and converge on a central junction. We explore variations on vortex construction that enable a range of aesthetic opportunities in the design of mazes, and demonstrate our technique with several of our own examples.

Keywords: Maze; Labyrinth; Spiral; Vortex maze; Puzzles

AMS Subject Classification: 65D18; 51N05; 68U05

1. Introduction

Mazes have been a part of human culture for thousands of years [7]. They can form the locus of a powerful legend, as in the journey of Theseus into the labyrinth of Crete. They can serve as a focus for meditation and prayer, most famously on the floor of the cathedral at Chartres. They are a compelling and occasionally frustrating source of entertainment for both children and adults. Countless books of mazes of all kinds are available. Designers such as Adrian Fisher [6] are in constant demand for large-scale maze installations in cornfields, amusement parks, and private residences.

An effective maze is simultaneously a work of art and an engaging puzzle. The process of designing a maze must therefore balance the competing goals of complexity and aesthetics. As an exemplar, consider the remarkable work of Christopher Berg [2]; an example is shown in Figure 1. His images are both challenging mazes and stylized line drawings of real-world scenes.

We are interested in using the computer to assist in the design and rendering of mazes. Although maze design is fundamentally a creative activity, we believe that there is an opportunity to automate some of its more mechanical aspects, freeing the human designer to address the twin goals of complexity and aesthetics at a higher level. Computer-aided maze design is a broad topic; mazes are constructed in a wide variety of styles and media, with more waiting to be discovered. Both complexity and aesthetics rely not only on measurable geometric properties of a maze, but also on poorly understood facts about human perception and psychology. In part, we hope that our computer-based research may provide some insights into these mysteries.

In this paper, we describe an algorithm that produces what we believe are difficult-to-solve mazes. Once our primary goal of complexity is satisfied, we then explore the extent to which we can manipulate the aesthetics of our mazes to increase their visual appeal. The results form an interesting class of abstract geometric designs. Our technique is based on repeated use of a device that Berg calls a *vortex* [3]. Roughly speaking, a vortex is a collection of passages that wind around each other in a spiral and converge on a central junction.

*Corresponding author. Email: csk@cgl.uwaterloo.ca

Journal of Mathematics and the Arts
ISSN: 1751-3472 print/ISSN 1751-3480 online
http://www.tandf.co.uk/journals
DOI: 10.1080/1751347YYxxxxxxxx

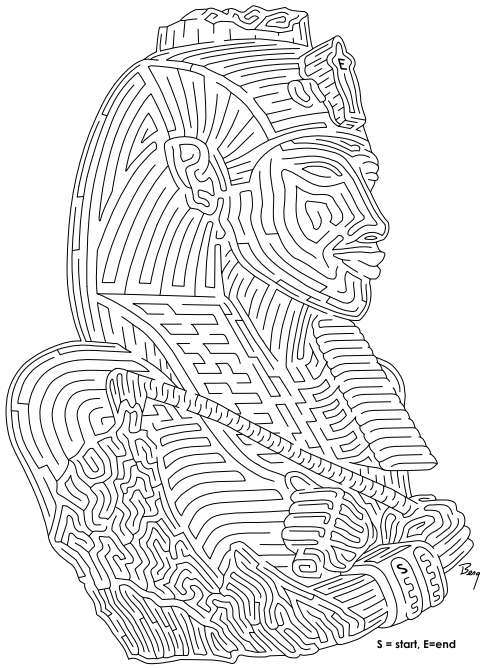
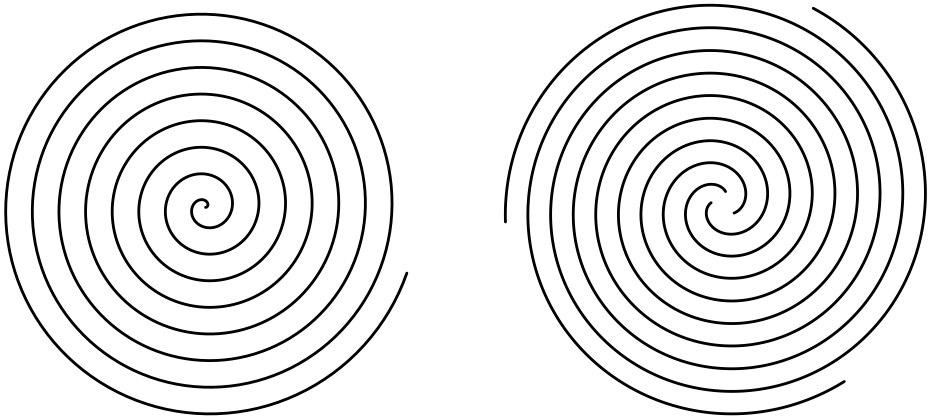


Figure 1. A maze depicting the Pharaoh Akhenaten, drawn by Christopher Berg (`http://www.amazeingart.com`).



(a) (b)

Figure 2. Simple examples of (a) a spiral and (b) a three-way vortex.

The rest of the paper is organized as follows. We proceed by presenting our methods for constructing individual vortices (Section 2) and mazes made from interconnected vortices (Section 3). We present some ways to achieve interesting aesthetic effects within the domain of vortex mazes (Section 4), and show some finished results made with our prototype implementation (Section 5). We then pause to consider the deeper question of the complexity of a maze (Section 6). Finally, we offer some conclusions and discuss opportunities for future work (Section 7).

2. Vortices

A spiral is a passage with a single entrance that leads to a dead end. An example is shown in Figure 2(a). Spirals have the property of moving the dead end very far along the passage from the entrance, with the

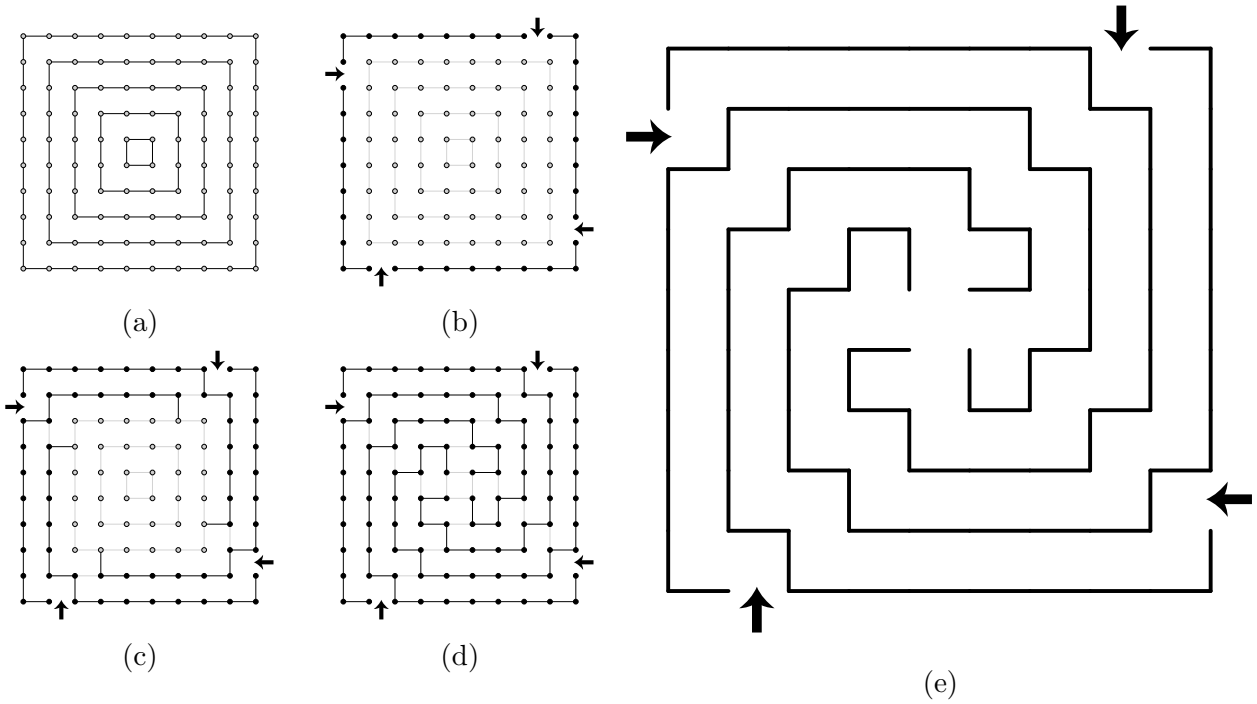


Figure 3. An illustration of the construction of a basic vortex within a square region. The offset polygons and sample locations are shown in (a). The entrances and initial state of the edges is shown in (b). In (c) we begin to grow trails, and in (d) we complete the process. The finished vortex appears in (e).

idea that a solver might be lured into following a long path before discovering it to be fruitless. However, we believe that in practice the solver can easily avoid spirals, at least when viewing the maze from above. The continuity of the spiral's boundary makes the single entrance plainly visible, and any region with only one entrance must be a dead end unless it contains the start or end of the maze.

Berg observes that the vortex, a close cousin of the spiral, is a much more effective device in maze design [3]. A vortex is a collection of three or more passages that spiral together to a central junction. A vortex effectively obfuscates the relationship between its entrances. Upon arriving at the centre, it can be difficult to determine which branch to take to get to a desired exit passage. Figure 2(b) shows an example of a vortex.

In this section, we first give a simple technique for constructing vortices based on convex polygons. We then enrich the technique by adding variations for producing more complicated and more attractive results. Our technique is comprised of two main steps. First we build a vortex in which all the entrances are connected to one another (Section 2.1). Then we modify the vortex slightly so that the entrances are partitioned into mutually interconnected subsets (Section 2.2).

We note that although we developed this technique ourselves, we later discovered that Knoll had previously described a very similar approach [8]. Our technique can be considered as an extension of hers, based on a more general and less structured class of vortex shapes. Furthermore, Knoll does not investigate the question of routing.

2.1. *Vortex construction*

Let $\mathcal{P}$ be a convex polygon in which we wish to construct a vortex, and let $d > 0$ be the width of the passages that will be constructed. We will draw a vortex by connecting points that lie on a sequence of evenly-spaced *offset polygons* within $\mathcal{P}$. In general, an offset curve is the locus of points that lie at some fixed distance from a given curve. Offset curves can be difficult to construct, but the special case of a convex polygon has a simple solution. Because $\mathcal{P}$ is convex, it can be seen as the intersection of the halfplanes defined by its edges. Every offset of $\mathcal{P}$ is then the convex polygon obtained by intersecting the

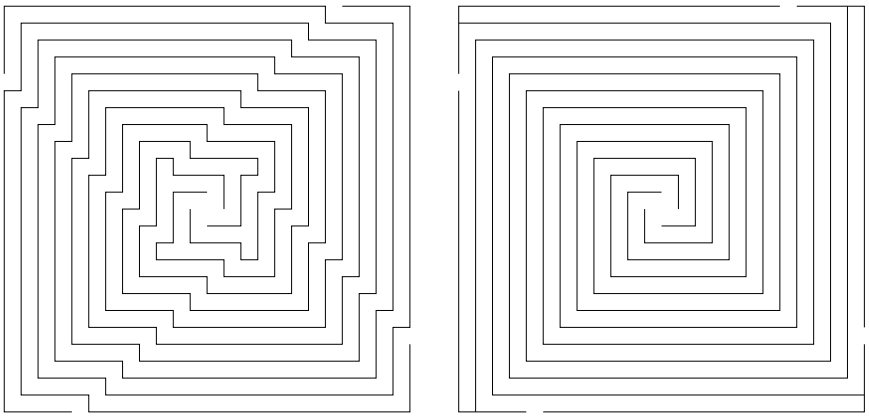


Figure 4. A demonstration of the formation of "ridges" from systematic bending during vortex construction. The vortex in (a) has diagonal ridges. In (b), we fix the problem by temporarily moving entrances to the corners of the region during construction.

corresponding offsets of $\mathcal{P}$'s halfplanes. By iterating, we arrive at a sequence of concentric offset polygons $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$, where $\mathcal{O}_1 = \mathcal{P}$, and each $\mathcal{O}_i$ is the offset at distance d from $\mathcal{O}_{i-1}$. We arrive at a natural stopping place $\mathcal{O}_n$ when the following iteration would yield the empty set.

The next step is to mark every $\mathcal{O}_i$ with a set of evenly spaced points called *samples*. They are placed so that the distance between adjacent samples is d, as measured along the boundaries of the polygons. Figure 3(a) shows a polygon with its offsets and sample positions. The samples divide the $\mathcal{O}_i$ into short curves we will call *segments*. The walls of our vortex will consist of a subset of the segments, together with a few additional lines that connect samples on adjacent offset polygons.

We now designate a subset of the segments on $\mathcal{O}_1$ as *entrance segments*. These represent the openings into the vortex; every entrance will be the mouth of a passage that spirals towards the center of the vortex.

The remaining walls are traced out by building a set of *trails*, one for each entrance. A trail consists of a sequence of sample points as defined above. Consecutive points in the trail are either adjacent on one offset polygon, or on two adjacent offset polygons. We render the trail by drawing the corresponding segment in the former case, and a straight line connecting the samples in the latter.

Let us assume that we wish to construct a vortex that spirals in the clockwise direction (the counterclockwise construction is analogous). For each entrance, we create a trail that initially contains the sample point on the counterclockwise side of that entrance. We can then construct the trails iteratively, consuming one offset polygon at a time on the way to the centre of the vortex. To pass from $\mathcal{O}_{i-1}$ to $\mathcal{O}_i$, we append to each trail the sample point on $\mathcal{O}_i$ that is closest to the trail's current end. Every trail then travels clockwise from this new point, stopping just before it lands on a sample belonging to another trail. This step will consume all the samples on $\mathcal{O}_i$, at which point the iteration continues on the next level inward. This process is illustrated in Figures 3(c) and (d).

When an entrance lies somewhere in the middle of a polygon's side, every layer of the vortex will need to make a small detour around the place where that entrance's trail begins. As shown in Figure 4(a), this systematic pattern of kinks creates the impression of "ridges" in the vortex. Although ridges can be visually interesting, we would also like to provide a way to eliminate them. When the polygon $\mathcal{P}$ has at most one entrance per side, and the sides end in relatively acute angles, we can suppress the ridge artifacts. In the clockwise case, we build the vortex as if the entrances had been given at the counterclockwise limits of the polygon sides they lie on. As shown in Figure 4(b), this modification pushes the kinks into the corners of all the offset polygons, where they disappear.

Figure 5 shows some basic vortices. The circular regions in the top row do not have corners where kinks can be hidden. It is unclear how to suppress the ridge patterns in this case.

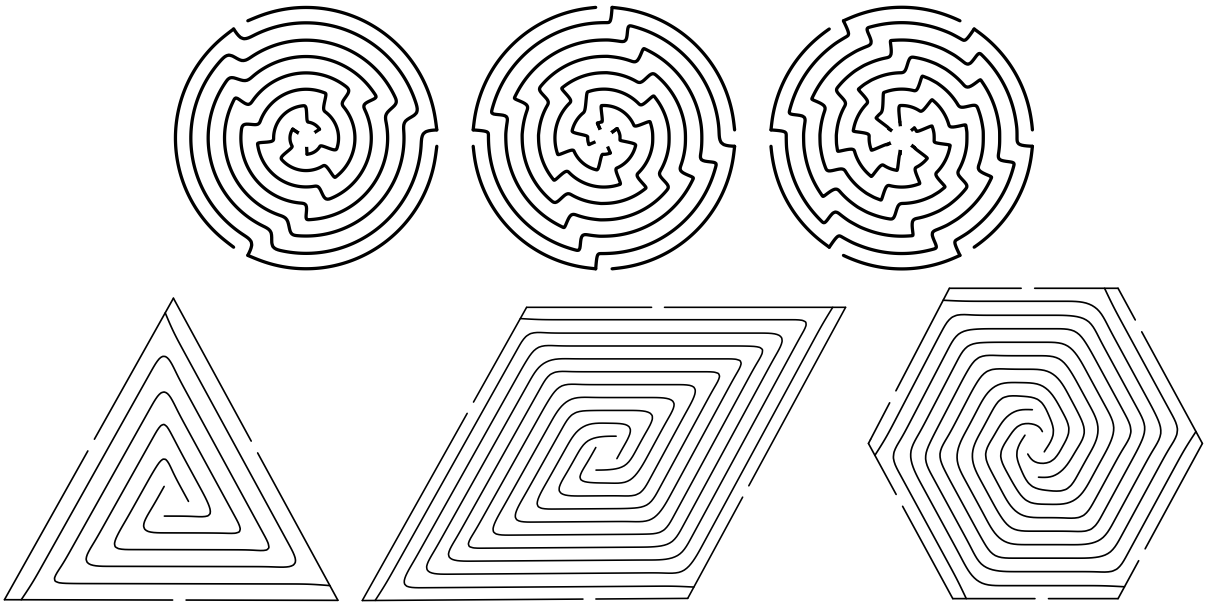


Figure 5. Sample circular vortices with three, four, and six arms are shown in the top row. Because there are no corners in circles, our algorithm produces ridges. The bottom row shows examples of vortices constructed in differently-shaped polygonal regions. All six examples are rendered using curved paths, as described in Section 4.

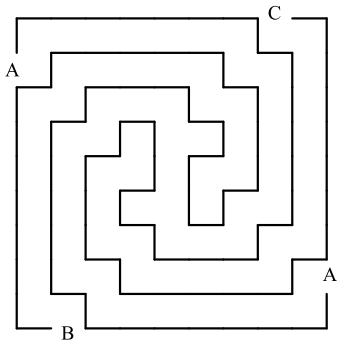


Figure 6. A variation of the vortex in Figure 3 with non-trivial routing. The two A entrances connect to one another, and the B and C entrances lead to dead ends.

2.2. *Routing*

The construction given above will produce a vortex in which all entrances are connected via a central junction. A designer will want to control the *routing* of the vortex, blocking paths between some pairs of entrances. In general, it should be possible to partition the entrances into subsets where two entrances are connected by a passage through the vortex if and only if they are in the same subset. We can record such a partition by labelling the entrances with letters, where each letter corresponds to one of the subsets. Figure 6 gives an example of a vortex with a non-trivial routing and corresponding labels on the entrances.

Not all partitions can be realized as planar mazes, a constraint that can easily be checked by examining the circular sequence of labels around the polygon. It follows from the Jordan curve theorem that the labels are legal if and only if the sequence contains no (possibly non-consecutive) subsequence of the form $\alpha\beta\alpha\beta$, where α and β are distinct labels. Such a subsequence would imply that passages belonging to α and β would have to cross without interacting, an impossibility in a planar drawing.

We describe the routing process in two steps. First, we solve the problem combinatorially; that is, we use the labelling of the entrances to derive which walls in the vortex must be connected to which other walls, without knowing how those connections will be realized. Then, we implement the combinatorial solution within the practical context of the offset polygons and sample positions of our construction method. The

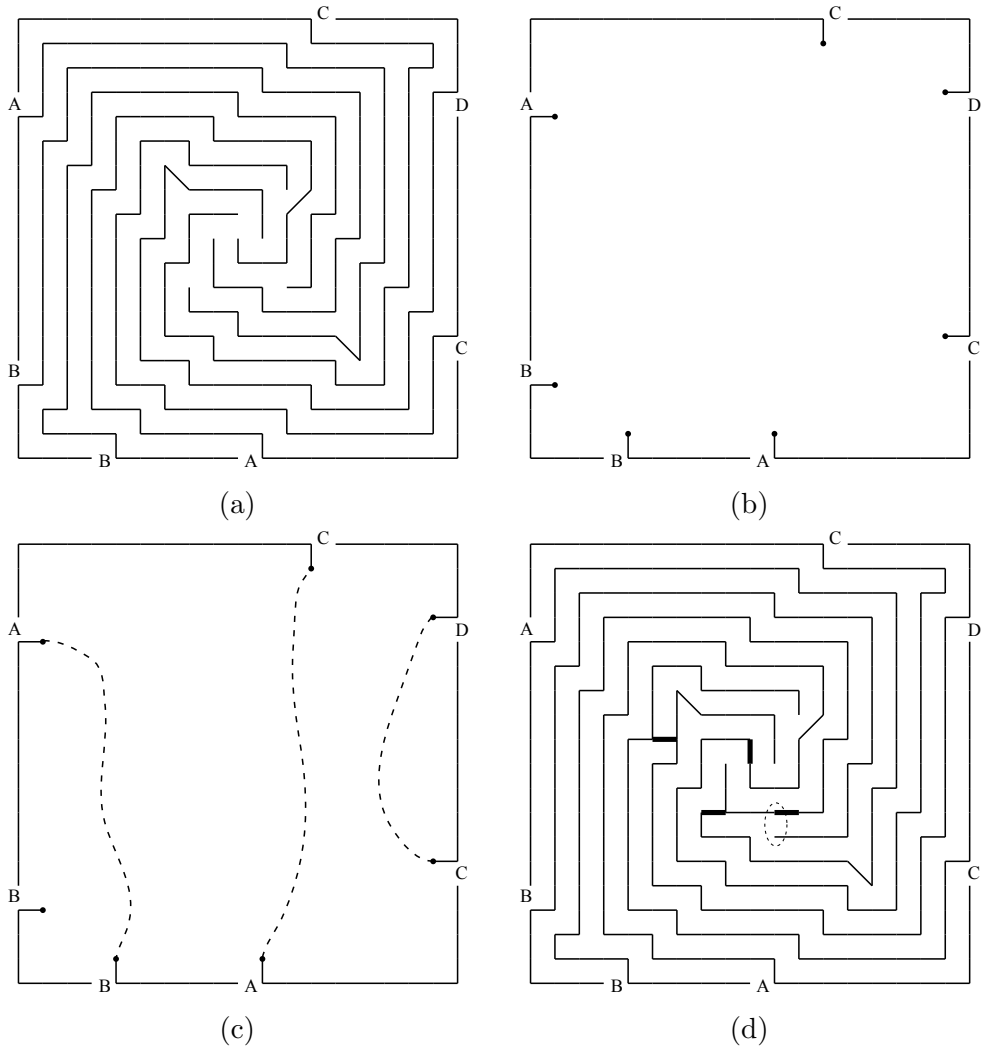


Figure 7. A visualization of the routing process (see Section 2.2 for details). The vortex in (a) is constructed from the given entrances and labels. In order to visualize the topological relationships between the trails, we simplify the drawing in (b). The necessary topological connections are made in (c). As described in the text, we build new connections based on the label sequences $ABBA$, $ACDCA$, and CDC. Note that the new connections the letters into disjoint regions. One possible modification to the walls of the vortex is shown in (d). The walls drawn with thick lines were added to block passages, and a wall was removed in the circled region.

routing process is illustrated in Figure 7.

In order to understand routing as a combinatorial problem, we can greatly simplify our visualization of vortices, as shown in Figure 7(b). For a clockwise vortex, every entrance grows a trail from its counterclockwise side. The trails are all disjoint. Schematically, the trails can be represented as short curves protruding from the entrances; the fact that they wind around one another in the vortex is irrelevant.

We now travel counterclockwise around the boundary of the vortex, looking for any sequences of labels of the form $\alpha\beta_1 \ldots \beta_k\alpha$, where $k > 0$ and $\beta_i \neq \alpha$ for all i. Note that if there is only one occurrence of α in the entrances, the sequence starts and ends at the same entrance.

For any such sequences, we must connect the trails attached to the entrances corresponding to the first α and to β_k. This new connection will isolate all the β_i from the the α entrances. If the labelling is legal, and the curves used to make these connections are not made to intersect one another, the resulting walls will have a connectivity that satisfies the desired routing. An example of these connections is shown in Figure 7(c).

The next step is to realize these topological connections within the geometric structure of the vortex. Some connections can be made easily by connecting the ends of the trail across the empty space at the central junction. Or, if two trails come from consecutive entrances, they will bound a shared passage; a

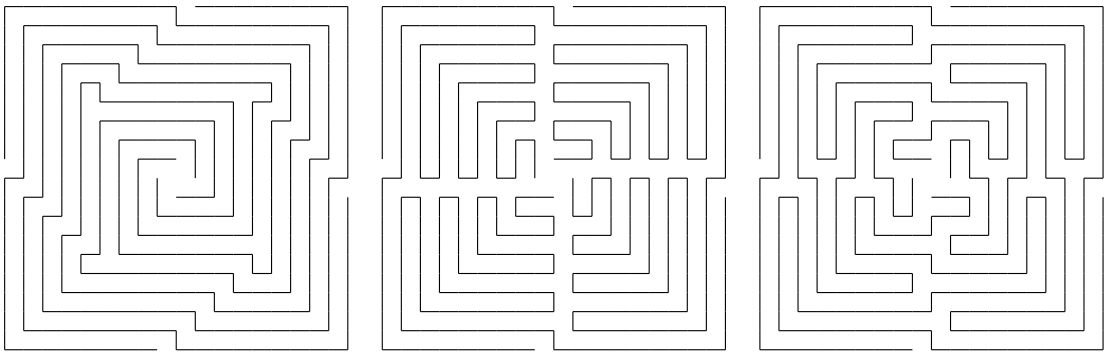


Figure 8. Vortices in which trails are allowed to curve back on themselves at each new offset polygon. A basic vortex is shown in (a). In (b), the trails alternate directions each time they move inwards. A more complicated scheme is given in (c).

dead end can be inserted anywhere along that passage. But there are also less common cases where there is no straight line path from from one trail to another. Here, we use an ad hoc approach. We find any short curve from one trail to the other. Any other trails intersected by that curve are sliced into two pieces. The inner piece is kept attached to the curve and becomes isolated from the entrance that generated it. The other is opened in order to preserve the topology of the vortex. Figure 7(d) shows a vortex in which walls have been added and removed to achieve a complicated routing.

2.3. *Early trail termination*

The vortices produced so far may be confusing for a novice, but with practice they become familiar. For a given pattern of sample points and desired routing, the construction will produce a characteristic pattern in the centre of the vortex. A practiced maze solver could then learn to recognize the structure of the vortex at a glance and pass through the vortex without getting disoriented. We want to thwart this potential shortcut.

An easy way to randomize the centres of vortices is to terminate some trails early. A trail will simply cease to propagate inward, and the remaining trails will continue on without it. In practice, we give a trail a small probability of terminating at every step. The probability grows towards the centre of the vortex. This simple change makes it much more difficult to learn the characteristic signatures of vortices. The vortices of Figure 10 make use of early termination.

It might also be possible to randomize the geometry of a vortex via a more continuous deformation of the walls, using for example the attraction-repulsion technique of Singh and Pedersen [13]. We leave this possibility for future work.

2.4. *Changing directions*

So far, we have decided up front whether a given vortex will spiral in a clockwise or counterclockwise direction. As Knoll shows [8], this decision need not be made uniformly across a vortex. Every offset polygon can be assigned its own direction, so that when a trail drops inward one level it might turn back upon itself. Figure 8 shows a simple vortex and two variations where trail directions are permitted to change.

In our implementation, we have experimented with the simple mechanism of alternating directions at every level, though obviously more complex patterns are possible, as in Figure 8(c). In full generality, every trail can have its own initial direction and mechanism for deciding its new direction at every level change. More experimentation would be required to discover an intuitive way to control that degree of flexibility.

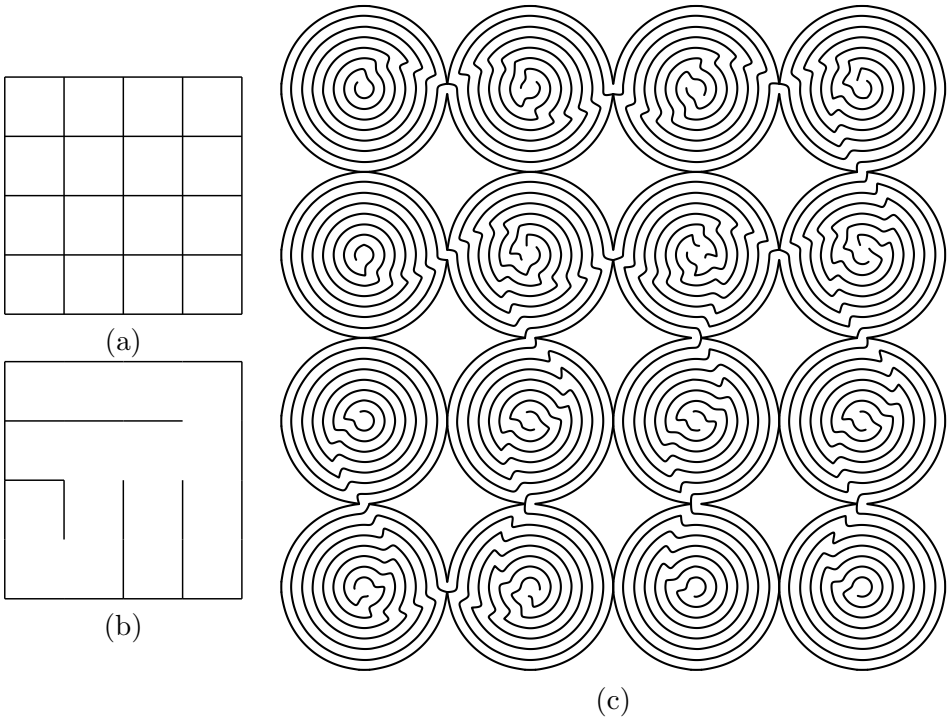


Figure 9. The construction of a network-of-vortices maze from a grid. The original 4x4 grid is shown in (a). the template maze in (b), and the final vortex maze in (c).

Figure 10. A variation of the maze in Figure 9(c) in which every vortex has four (possibly false) entrances and trails are permitted to terminate early.

3. Networks of vortices

Armed with the vortex as a primitive, we can now create difficult mazes by assembling networks of interconnected vortices. Berg mentions this technique as an especially fiendish maze style [3].

Using the construction of the previous section, it is easy to build interconnected vortices. Consider an arrangement of convex polygons with pairwise disjoint interiors. We can construct a vortex in each polygon. If two polygons intersect in a line segment, we can choose to connect the vortices by ensuring that each one has an entrance lying on that portion of its boundary.

The fact that we can choose whether or not to connect adjacent vortices immediately suggests a construction technique for what we call a *network-of-vortices* maze. We use an ordinary maze construction algorithm such as that of Shivers [12] to build a small "template maze" on a square grid. This algorithm begins by randomly generating a spanning tree over a graph formed by the cells of the grid. If two neighbouring cells are connected in the spanning tree, the edge between them is erased. We then render each cell of the grid as a vortex, where there is a shared entrance at each compass heading if and only if the cell's wall in that direction has been erased. Figure 9 gives an example of this construction technique on a small square grid. Because of the obfuscating effect of the vortices, the final maze is more difficult to solve by eye than its template. (By "solve", we mean "find a path from start to finish that does not cross any walls of the maze". *When the start and finish are not labeled, they may be taken as the top left and bottom right corners of the maze.*)

This approach is too simplistic, however. Notice that the only dead ends in the vortex maze are the centres of spirals derived from dead ends in the template. The connectivity of the template maze can therefore easily be determined from the connectivity of the vortices. A skilled solver could deduce the underlying template by eye, and map that solution directly onto the vortices. The obviousness of the spirals can be alleviated by placing an entrance at every compass heading, and closing the entrances when adjacent template cells are not connected, as in Figure 10. This change moves the dead ends from the centres of spirals to the boundaries of vortices, but the template can still be deduced from the presence or absence of connections between adjacent vortices.

An even better solution is to permit more general routing through the template cells. In the case of a square grid, this is made possible by subdividing each cell into five regions as shown in Figure 11(a). Each augmented cell consists of twelve edges arising from an outer square, an inner square, and four diagonal edges. We build a template maze over the resulting arrangement of squares and trapezoids using the same spanning tree algorithm as before, leading to a maze like that of Figure 11(b). The edges in each augmented cell that survive erasure determine the structure of the vortex to place there. In this example, we give every vortex four labeled entrances, but close off those entrances if the corresponding edges of the cell's outer square are part of the template maze. Two entrances share a label if and only if there is a clear path through the template maze from one entrance to the other.

We believe that this final method produces truly difficult mazes. An example is given in Figure 11(c). The more general routing produces dead ends that are much further removed from the decision points that lead to them. As a result, the solver cannot speculatively "look around a corner" to see a nearby dead end and avoid following a branch. Moreover, most vortices remain open to their neighbours, meaning that it is much more difficult to determine the overall connectivity at a glance. This method also produces the entertaining case where the solution to a maze passes though two entrances to a four-way vortex, travels through other cells for a while, and then comes back to consume the other two entrances. The marked start and end positions in Figure 11 illustrate a case where this device occurs. We believe that a solver would be surprised to find themselves returning to a vortex for a second pass.

Network-of-vortices mazes generalize from the simple square grids shown here to arbitrary networks of interconnected cells. A random spanning tree can be found over the graph implied by the augmented cells, and this spanning tree can be turned into a template maze. Alternatively, the connectivity can be decided manually by the designer. Figure 13 illustrates the latter case; the designer has drawn a set of connected paths (shown dotted) on top of the arrangement of cells. The routing within each cell can be derived from the restriction of the designer's drawing to the interior of the cell.

The automatic case does not generalize quite as readily. Our approach relies on augmenting the cells with additional edges. The template maze algorithm will delete some of the augmented cell's edges, and what remains is a diagram that maps onto a routing for that cell. We chose our particular augmented square cell (two concentric squares connected by diagonal edges) because every possible routing between entrances on the square's edges will arise naturally in the construction of the template maze. However, this approach does not generalize to other cell shapes. The labels in Figure 12(a) suggest one possible routing through a hexagon. However, the augmented cell does not have enough faces to allow corresponding labels to connect, and so this routing will never arise in our template maze. It would be necessary to subdivide the augmented cell further in order to allow all possible routings to occur spontaneously.

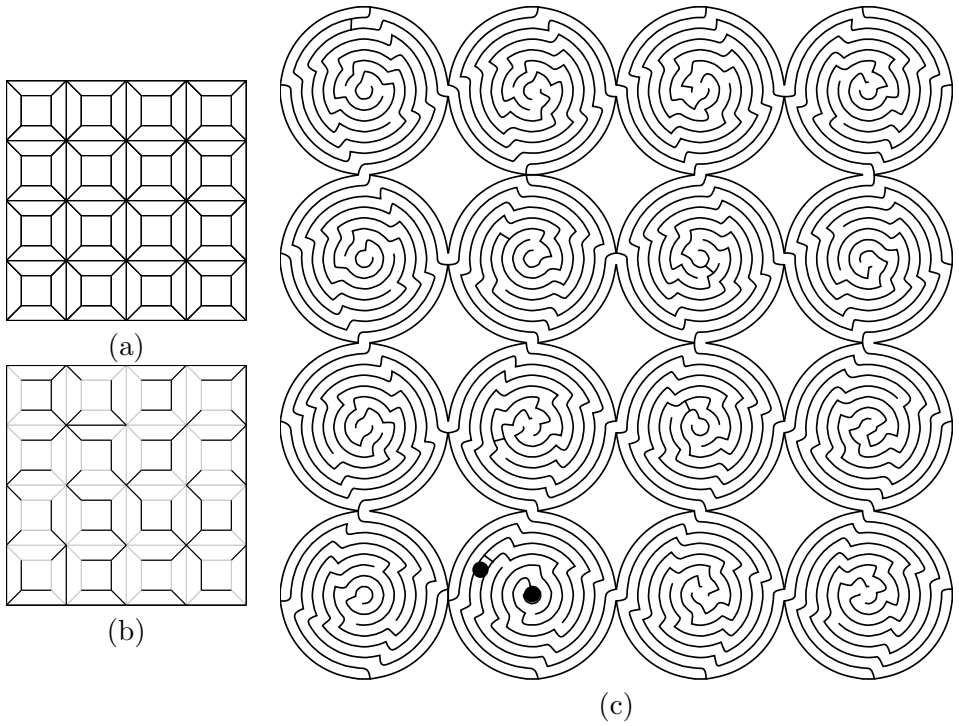


Figure 11. An example of a more complex vortex maze that can be constructed by splitting grid cells. Each cell is augmented with eight extra edges in (a). In (b), a template maze is constructed over the resulting arrangement of squares and trapezoids. The template maze guides the construction of the vortices in (c). The two black dots are joined by a path that passes through half of the vortex immediately above and returns to use the other half later.

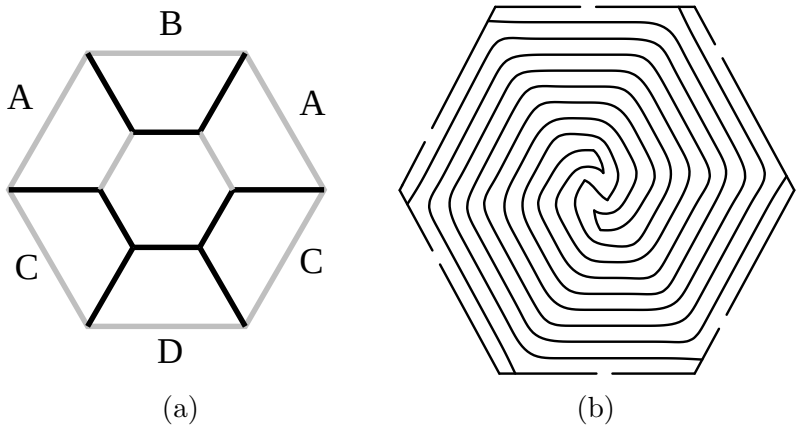


Figure 12. A simple example of how the augmented square cell of Figure 11 does not translate to other cell shapes. An augmented hexagonal cell is shown in (a) with a desired routing. Once edges have been erased to reflect the routing of the labels A, B, and D, there is no room left to connect the C entrances. Nevertheless, a hexagonal vortex with that routing exists, as shown in (b). The vortex was produced by manually editing the example of Figure 5.

4. Aesthetic effects

Originally we focussed on the role of vortices in constructing mazes that are difficult to solve. We have also experimented with the aesthetic possibilities that arise in mazes of interconnected vortices. We have found that even in this very limited, abstract domain, we can still produce mazes that are also attractive drawings.

By default, we render the trails in a vortex as piecewise linear paths. Alternatively, the sample points can be used as control points for cubic splines, giving a smoother, more fluid appearance. Many examples in this paper are drawn using curved walls.

As mentioned in Section 2.1, there are two free parameters in the basic vortex construction algorithm:

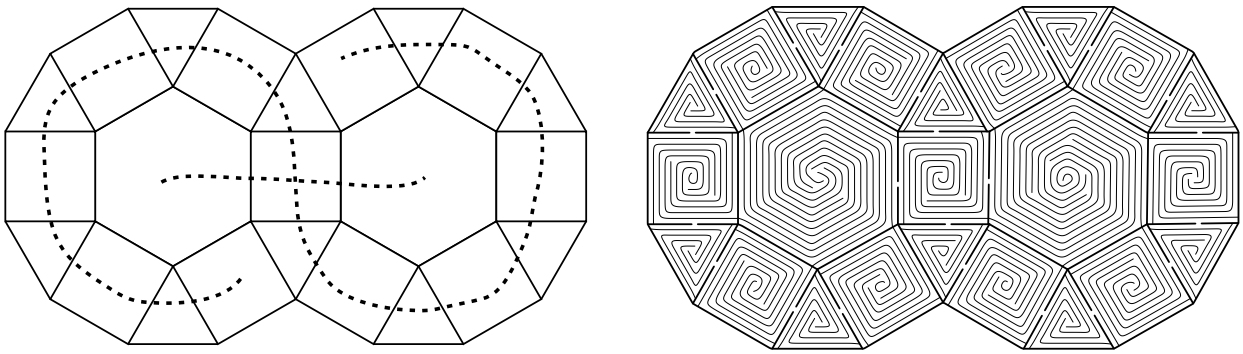


Figure 13. A network-of-vortices maze constructed from a user-supplied solution path. The routing and connectivity of the vortices are decided by the sketched dotted line drawn by the user on the template. In this case, the designer has not chosen start and end positions.

Figure 14. An example of varying the aesthetic parameters of the vortices in a maze. We alternate the orientations and densities of adjacent vortices.

the direction and the spacing between offset polygons. Every vortex can choose different values for these parameters as a way to achieve contrast between neighbouring vortices. The offset distance is also inversely related to the tone of the resulting vortex; a smaller value of d leads to more closely spaced walls and a darker vortex. Figure 14 gives an example of varying both vortex orientation and density.

In the case where we alternate the trail directions on successive offset polygons (see Section 2.4), we obtain attractive empty bands that run outward from the centre of the vortex. We can even arrange these bands so that they form recognizable shapes. Given a planar embedding of a graph with straight edges, we can cover the graph with pairwise disjoint discs whose centres lie at vertices or on edges. If we then place entrances to the discs at their intersections with graph edges, the resulting vortices depict the original graph via channels between trails. See Figure 15 for an example. The resulting maze has the effect of an optical illusion.

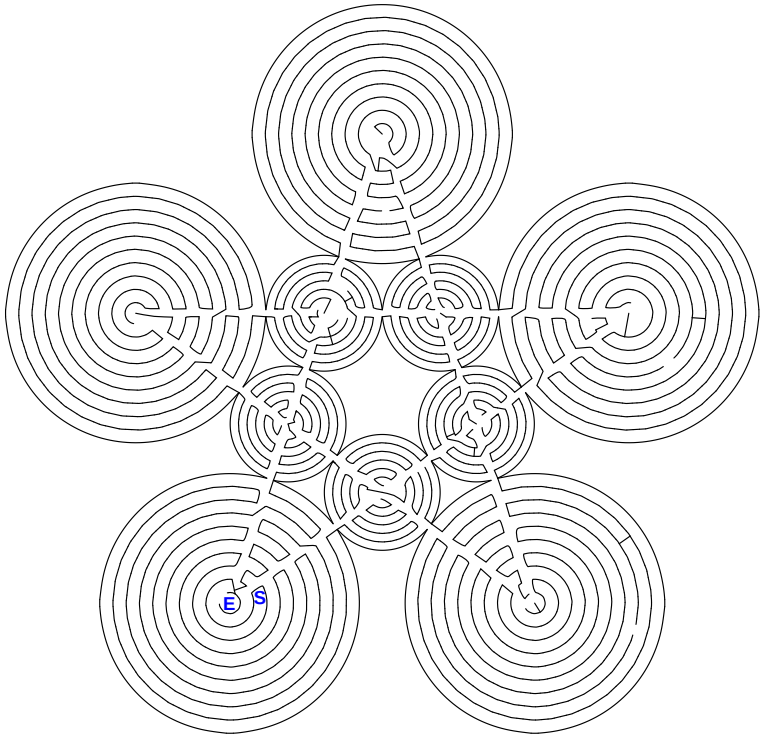


Figure 15. A demonstration of the use of alternating spiral directions to depict a pre-determined line drawing. The target pentagram was covered with discs that were then turned into vortices.

5. Implementation and results

Our algorithm is implemented as a C++ program that produces Postscript output. We use the CGAL library [4] to store and manipulate the geometry of the maze. The program includes an interactive user interface that allows the designer to draw an arrangement of convex polygons, assign them various parameters such as passage width and spiral direction, and optionally sketch a path indicating the connectivity between the vortices. The program uses this information to construct a network-of-vortices maze. Examples such as those in this paper can easily be made with a few minutes of user interaction and less than a minute of computation. We have used our program to produce the finished mazes in this paper, including the additional results in Figures 16 and Figure 17.

6. On the complexity of mazes

In this paper, we claim that our network-of-vortices mazes are difficult to solve. Our evidence is anecdotal, and supported by the opinions of experts like Berg. Still, our informal claim raises the question of whether it is possible to quantify the notion of the complexity of a maze. A measure of complexity would benefit maze designers, who could tailor their work to different audiences.

We believe that the measurement of complexity in mazes is a very deep problem. Clearly, we could begin with any number of topological and geometric properties of a given maze. We could measure the length of the solution path, the number and length of the branches, some notion of "twistiness" of passages, and so on. We are aware of one example of previous work that attempted to classify mazes this way [9].

However, we believe that complexity must ultimately depend to a significant degree on human psychology and perception. There are many ways in which mazes can be deformed to undermine (or assist in) the solving process. Just as the eye seems to get lost among the many parallel passages in a spiral, mazes can be made more difficult by "stretching" them, so that they include long bands of parallel lines. On the other hand, any maze can be laid out so that the solution path is a straight horizontal passage with branches leaving it above and below. Although this maze has the same topological structure as its tangled original,

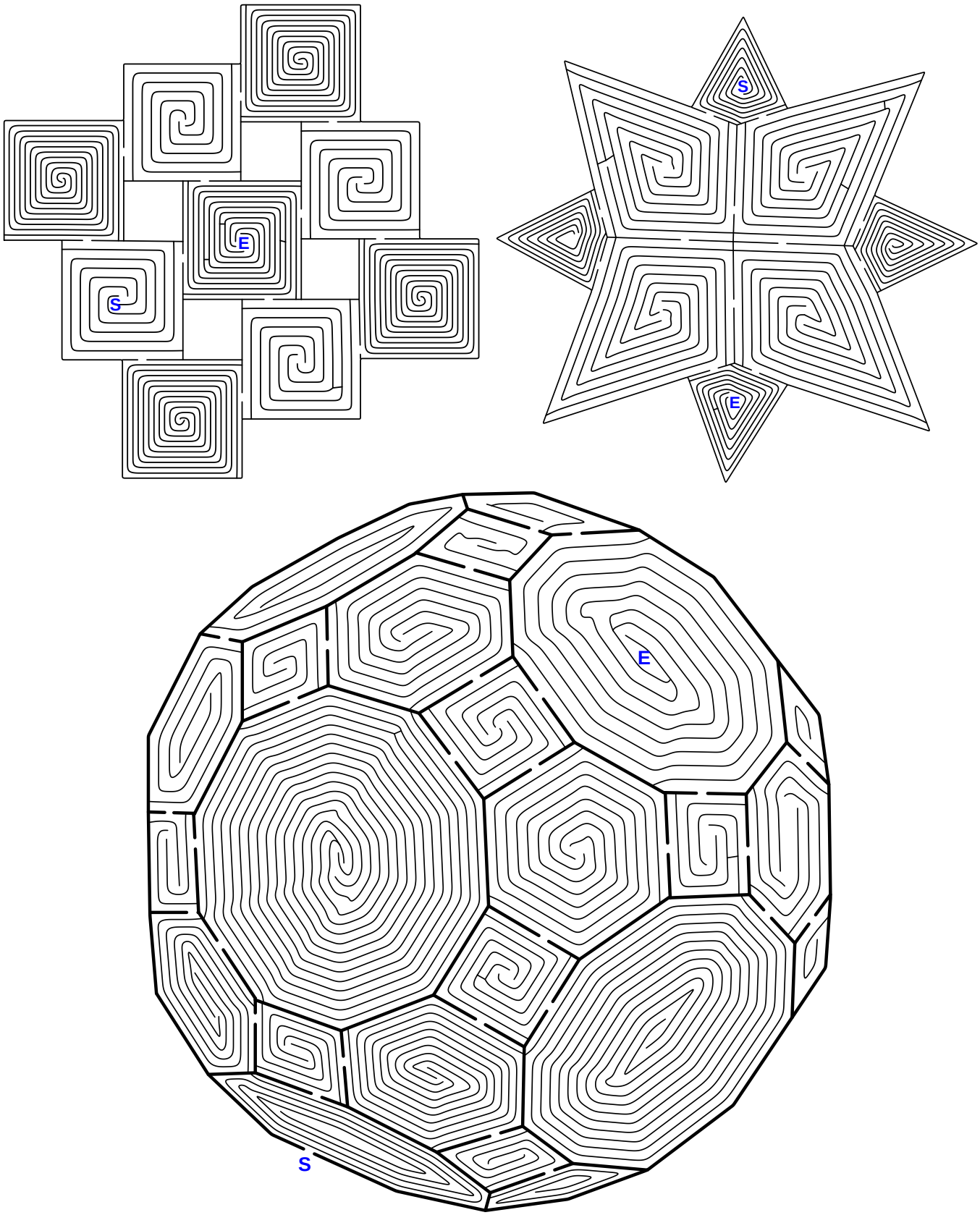


Figure 16. Sample results created using our system.

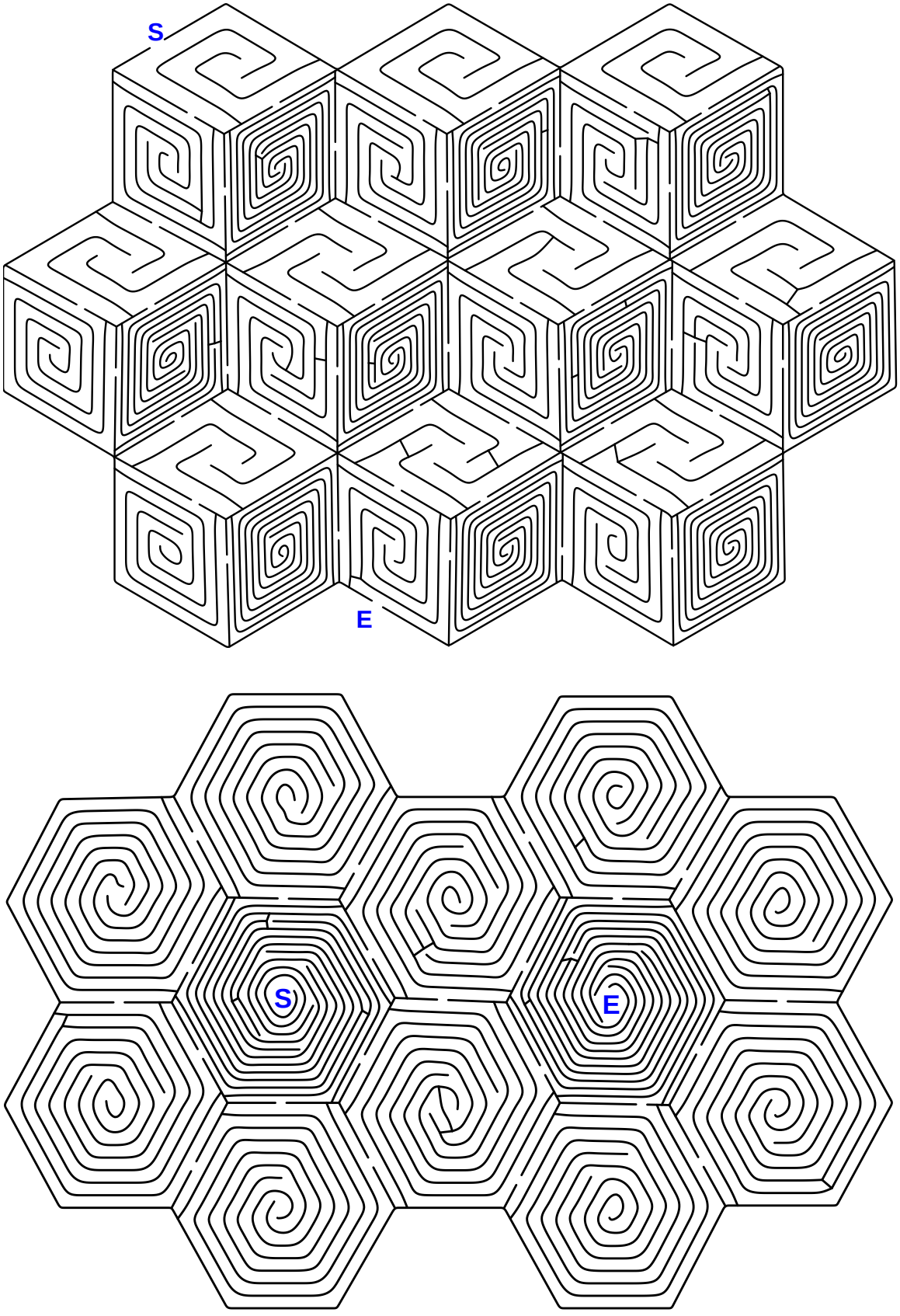


Figure 17. More sample results created using our system.

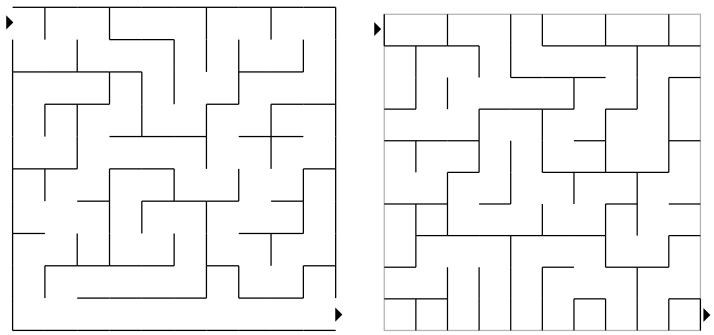


Figure 18. An example of a maze and its corresponding antimaze. To solve the antimaze, move from cell to cell, crossing a wall at every step.

the eye can spot the straight solution path immediately.

One especially devious variation is the *antimaze*. In the example of Figure 18, a maze is developed on a square grid, but drawn with open and closed walls reversed. The solution is never allowed to move through empty space between two cells; it must always pass through a wall. Although the antimaze is trivially equivalent to the original, it is much more difficult to solve, since we are unable to exploit perceptual machinery that allows us to discover open paths.

These considerations all become more or less relevant depending on how the maze is to be solved. Solving by eye alone seems most difficult. Tracing a solution path with a pencil helps keep the eye from getting lost. At the other extreme, solving a corn maze from the inside relies on a much more local act of perception, and also to some extent on dead reckoning. We may require independent notions of complexity for these different cases.

7. Conclusion and future work

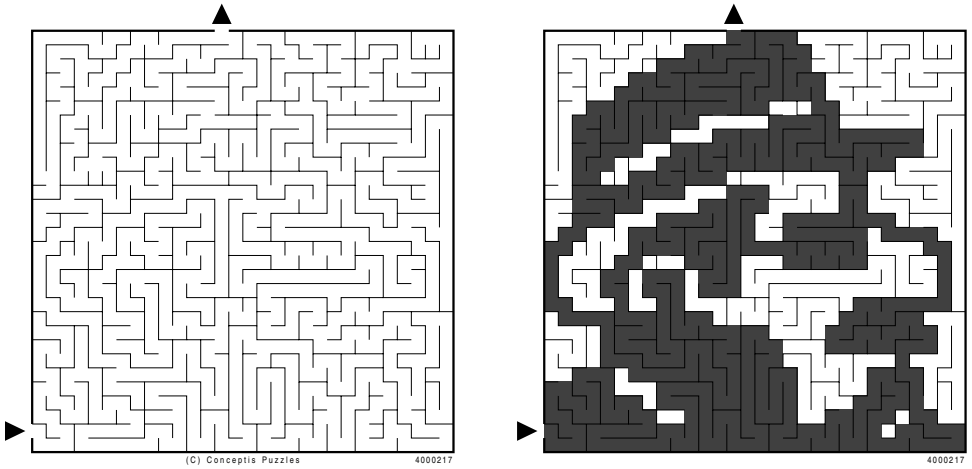
In this paper, we have presented a technique for constructing abstract geometric mazes that are difficult to solve. The technique can easily be implemented as a software tool for computer-aided maze design. The computer handles the construction of individual vortices and the connections between them, freeing the human designer to concentrate on higher level criteria for maze creation. The mazes produced lie within a fairly narrow aesthetic range, but the algorithm could be incorporated into a larger software system that supported maze design across a variety of styles.

An obvious extension to our technique would be the ability to draw vortices inside of non-convex polygons. Our naive approach for computing offset polygons would be inadequate here. Fortunately, more sophisticated algorithms appear in the computational geometry literature [1], with an implementation readily available in the latest version of CGAL. Alternatively, Elber and Wolberg show how to obtain a discrete approximation of offset polygons using computer graphics hardware [5]. Offset polygons in non-convex polygons can break into multiple disjoint pieces; our iteration step would need to be re-worked to spiral inward to multiple centres. We would also need to modify the routing step.

We could improve the appeal of many of the vortices by further suppressing the ridge artifacts. Section 2 claimed that this was impossible for circles, yet the spiral and vortex in Figure 2 have no artifacts. The difference is that in the latter two cases, the region boundaries are not circles! In effect, the kinks have been moved to the boundary. Based on this observation, It may be possible to modify a region's shape systematically to suppress kinks.

It would be interesting to explore other decorative possibilities of our vortex construction method. There is a long history of ornamental patterns based on spiral motifs. For example, bronzework of the Shang dynasty in China and many Chinese jades exhibit characteristic spirals [10]. Our technique could extend to produce patterns, or even mazes, in these styles.

We would like to continue to explore the aesthetic possibilities of mazes. Inspired by the work of Christo-



(a) (b)

Figure 19. An example of a Maze-a-pix. Printed with permission by Conceptis Ltd., `http://www.conceptispuzzles.com`. When the maze's solution path is filled in, the image on the right is revealed.

pher Berg, we ask how the computer might assist in producing mazes that depict real-world scenes. There are many excellent algorithms in the computer graphics literature that approximate photographs with line drawings (see, for example, the work of Salisbury et al. [11]); can such algorithms be adapted to produce mazes?

Maze-a-pix puzzles, introduced in 2004 by Conceptis Ltd., are also aesthetically intriguing. In a Maze-a-pix, the solver fills in the solution path as they go. When the maze is complete, the filled-in path resolves itself into a picture. In this case, the aesthetic qualities of the maze are revealed by the act of solving it. An example is given in Figure 19.

A maze is simultaneously a mathematical abstraction, a work of art, and a window into the human mind. We hope to continue studying both the complexity and aesthetics of mazes. We are interested in investigating the profound ways that these two goals affect each other.

Acknowledgments

We would like to thank Adrian Fisher, George Hart, Eva Knoll, and Morgan McGuire for their helpful comments and feedback during this project. Thanks also to Conceptis Ltd. and Christopher Berg for permission to reprint their puzzles.

References

[1] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761, 1995.
[2] Christopher Berg. *Amazeing Art: Wonders of the Ancient World*. Harper Collins, 2001.
[3] Christopher Berg. Amazeing art. `http://www.amazeingart.com`, 2005.
[4] CGAL: Computational geometry algorithms library. `http://www.cgal.org`, 2005.
[5] Gershon Elber and George Wolberg. Rendering traditional mosaics. *Visual Computer*, 19:67–78, 2003.
[6] Adrian Fisher. Adrian fisher's maize maze website. http://www.maizemaze.com/, 2005.
[7] Hermann Kern. *Through the Labyrinth: designs and meanings over 5000 years*. Prestel, 2000.
[8] Eva Knoll. Mazes and labyrinths. `http://faculty.msvu.ca/evaknoll/teknollogy/projects/labyrinths.html`, 2006.
[9] Michael Scott McClendon. The complexity and difficulty of a maze. In Reza Sarhangi, editor, *Proceedings of Bridges 2001: Mathematical Connections in Art, Music, and Science*, pages 213–220, 2001.
[10] J. Rawson, J. Williams, and D. Gowers. *Chinese Jade from the Neolithic to the Qing*. British Museum Press, 1995.
[11] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 401–406, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
[12] Olin Shivers. Maze generation. `http://www.cc.gatech.edu/~shivers/mazes.html`, 2005.
[13] Karan Singh and Hans Pedersen. Organic labyrinths and mazes. In *The 4th International Symposium on Non-Photorealistic Animation and Rendering(NPAR 2006)*, June 2006.