Organic Labyrinths and Mazes

Hans Pedersen
hp@metris.be
Metris Inc.

Karan Singh
karan@dgp.toronto.edu
University of Toronto

Abstract

This paper addresses the synthesis of labyrinthine and maze structures, which are represented as curves on 2D manifolds. The curves evolve based on a simulation controlled by spatially varying parameters defined by texture maps. We introduce the graphics community to the fascinating area of maze art and present a model for the automatic generation of organic looking labyrinths and mazes. We also present a framework based on regions and patterns for the interactive artistic control of NPR algorithms, such as ours, that evolve in both space and time. In the context of labyrinths, the framework provides the designer with control over both the path complexity and visual aesthetics as the curves evolve. The resulting labyrinths and mazes range from mathematically simple to intricately complex visual structures. Applications of the resulting curves include NPR, difficult to counterfeit imagery, environmental design and architecture, computer games and 1D parameterization of 2D manifolds.

CR Categories: I.3.3 [Computer Graphics]: Curve Generation

Keywords: Labyrinths, Mazes, NPR, Computer Games

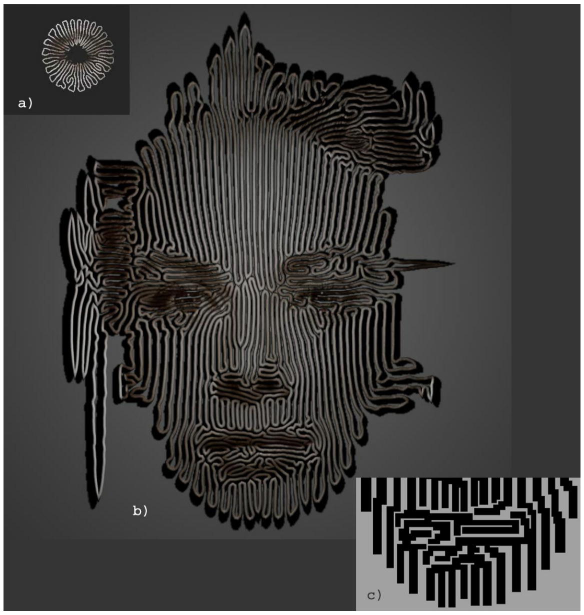


Figure 1: Amazing Grace: A circle evolves a) into a labyrinth b) and postprocessed into a grid-maze c).

1 Introduction

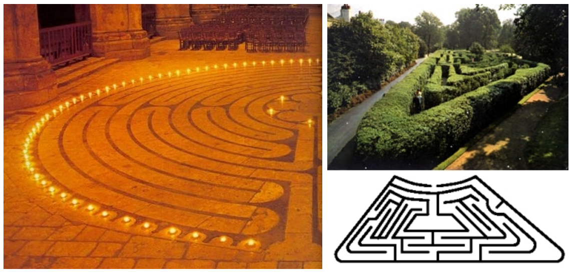
We shall not cease from exploring, and the end of all our exploring,
will be to arrive where we started, and know the place for the first time. *-T.S.Eliot*.

Labyrinths are arguably mankind's first creation borne purely of human imagination. Today, labyrinths and mazes cradle millennia of legend and folklore in their twisted articulations. Labyrinth and maze imagery has at different periods of time in various parts of the world been associated with all aspects of human life. It has been used as a symbol of fertility and birth, as well as one of purgatory and death. It has religious and meditative importance in Hindu, Christian, Islamic, Buddhist and Shamanic rituals [Conty 2002].

Physical labyrinths and mazes have served as athletic or battle fields as much as they have been used as sacrificial altars. Hedge and topiary mazes like the Hampton Court maze (see Figure 2b) have decorated gardens and palaces for centuries [Kern 1995]. The diameter of the famous 11 circuit labyrinth in the Chartres cathedral (see Figure 2a) is 42 feet, a millionth that of the Earth's diameter [Saward 2002]. The annual journal of mazes *Caerdroia*, and a recent book, *"The Unending Mystery"* [McCullough 2004], are comprehensive and entertaining references on labyrinths and mazes.

The words *maze* and *labyrinth* are synonymous in popular parlance. Semantically, however, a labyrinth is a unicursal path that simply winds its way from start to finish, which are often the same place. A maze on the other hand has forks, dead ends and sometimes cycles and multiple paths from start to finish. A maze embedded on a non-planar manifold is often referred to as a *planair* maze [Berg 2001]. Mathematically, a maze on a 2D manifold can be formulated as a planar graph embedding with a vertex at every fork connected by edges along the maze paths (a labyrinth is just a start and finish vertex connected by an edge). Such graphs have been extensively studied in mathematics and robotics [Lovasz 1996]. The answer to what makes a maze difficult to solve lies in a combination of its graph topology and perceptual science. We do not explicitly address solvability in this paper leaving the user to interactively judge and control the solvability of generated mazes.

Philosophically, labyrinths and mazes are looked upon as yin and

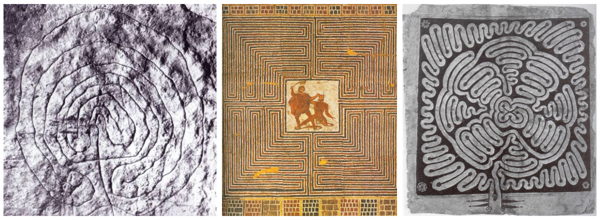


(a) Chartres cathedral labyrinth (b) Hampton court maze

Figure 2: Famous labyrinths and mazes. a) ©2002, Jeff Saward. b) ©1995, Hermann Kern.

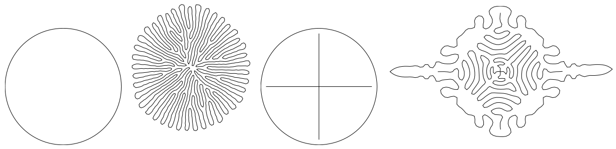
NPAR 2006, Annecy, France, 05–07 June 2006.
© 2006 ACM 1-59593-357-3/06/0006 $5.00

yang [Conty 2002]. The spiritual labyrinth is a meditative path laid out with careful artistic intent to set you free, while the puzzling maze is a challenging structure designed to trick and entrap. This duality is echoed in our design of labyrinths and mazes in this paper using the same curve formulation.



(a) Prehistoric (b) Roman art (c) Tantric diagram

Figure 3: Labyrinthine structures. a) Tomb carving, Sardinia, c. 2500-2000 BC. b) Roman mosaic, Salzburg, c. 300 AD. c) Indian manuscript, c. 18th century. ©2002 Hermann Kern.



(a) (b) (c) (d)

Figure 4: Mazes evolving from curve configurations. Curve (a) grows into the tree maze in (b). The disconnected curves in (c) result in a maze with a cycle (d).

The path defined along a single closed curve with no self intersections is a labyrinth with the same beginning and end (see Figure 4a,b). If we consider the same curve to bound a path and place two openings for a start and a finish, this dual representation defines a tree maze (see Figure 4b), where the strategy of walking with your left shoulder against a wall gets you from start to end. With multiple and intersecting curve configurations we are able to capture general multiply-connected mazes (see Figure 4c,d). Our model also captures labyrinths of multiple paths that are spatially intertwined but topologically disjoint, such as those depicted in Roman mosaics (see Figure 3b) [Kern 1995].
Labyrinths and mazes define the path to travel or environment to navigate in many computer games. Artistically, they have fuelled the imagination of artists from M.C. Escher, Keith Haring and Salvador Dali (see Figure 17) to MAD magazine cartoonists and contemporary maze artists like Christopher Morrison, David Russo, Christopher Berg and Mo Morales (see Figures 5). Meanders thus play an important role in the evolving era of interactive art and entertainment, where the audience is an integral part of the experience. Visually, labyrinths and mazes, could be roughly categorized as structured or organic. Structured mazes, often seen in puzzle books, are constructed on geometric grids and have paths precisely aligned with grid cells. Structured labyrinths, such as the 11-circuit Chartres labyrinth, are also the result of geometric construction algorithms. Organic labyrinths and mazes, seen mostly in maze art (see Figure 5), have an amorphous visual appearance resembling shapes seen in nature. In this paper we generate these organic forms using a dynamic curve simulation.

1.1 Rationale and related work

Organic labyrinths and mazes in art (see Figure 5) motivate the design of our approach. We make the trivial, yet critical, observation that paths in mazes in such artwork are bounded by largely parallel walls. This basic structure, however, is not particularly interesting in itself. Rather, an underlying concept shared by all the inspirational artwork is the exploration of variations over a simple set of path characteristics which include, but are not limited to: *random variations, local smoothness, frequency of branching, scale, self-similarity, anisotropy, regions with distinct patterns, seamless blending between regions, and rendering styles*.

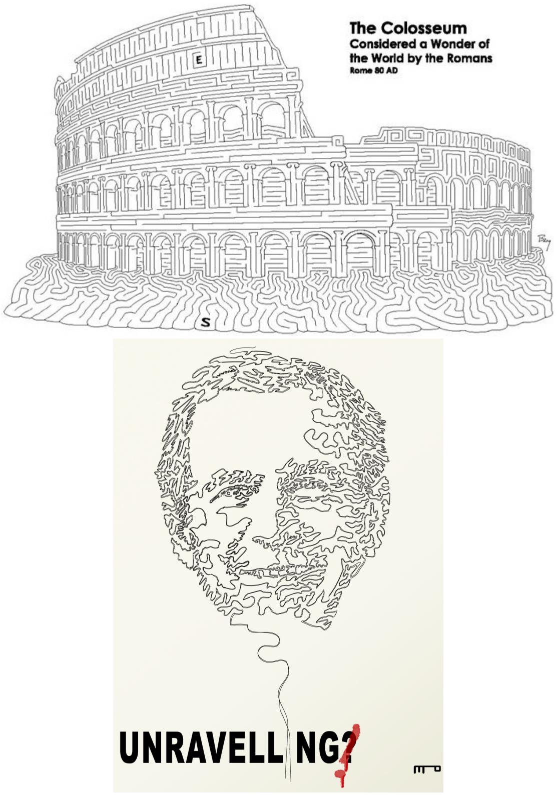


Figure 5: Labyrinth and maze art a) ©2001, Christopher Berg and b) ©2005, Mo Morales.

We couple this analysis with the observation that the fundamental structure of a maze bears resemblance with the point distribution patterns generated by the stochastic sampling algorithm used in [Turk 1991]. We thus choose to represent both labyrinths and mazes as configurations of a set of piecewise linear curves that evolve using an iterative algorithm similar to *point repulsion* [Turk 1991] with parameters designed to capture the maze characteristics listed above. We model *randomness* with *Brownian Motion*, a stochastic process that is the scaling limit of a random walk on a 2D grid [Karatzas and Shreve 1997]. Local smoothness is modeled using a Laplacian fairing term. Branching is controlled by applying the Lennard-Jones potential used to model inter-molecular forces [A. Ya Kipnis 1996]. The long range attraction controls branching frequency and, along with the short range repulsion, ensures that the maze walls remain nearly parallel. Together, these parameters lead to the basic maze synthesis algorithm described in Section 2. Self-similarity and varying scale are accomplished by modulating the basic parameters interactively while the maze is evolving.

The concept and approach motivated above has similarities to space-filling curves and fractals, curve modeling and parameterization, and differential equations for image processing. Space-filling curves, fractals [Gotsman and Lindenbaum 1996] and traveling salesman art [Kaplan and Bosch 2005] share a visual similarity to our labyrinths and mazes in that they cover space and have the ability to capture similar structures at various scales. Primarily, space-filling curves have been used in graphics for dithering images [Velho and Gomes 1995] but also for image compression and 1D parameterization of images [Dafner et al. 2000]. We contribute a technique similar in spirit to the art inspired work on artistic screening [Ostromoukhov and Hersch 1995], engraving [Ostromoukhov 1999] and floral ornaments [Wong et al. 1998]. The generated imagery can capture continuous variations in artistic abstraction of detail along the lines of [DeCarlo and Santella 2002]. Our work also relates to the problem of curve modeling at various resolutions. Research on multiresolution curves [Finkelstein and Salesin 1994] and curve analogies [Hertzmann et al. 2002] can be used within our algorithm to increase, reduce or stylistically vary the geometric detail of the output mazes. [Witkin and Kass 1991] and [Turk 1991] used partial differential equations from developmental biology to synthesize *reaction-diffusion* textures vaguely resembling maze patterns. *Rapidly-exploring Random Trees* have also been applied to produce an interesting range of grid-like mazes [LaValle 2005]. Finally, our curves are influenced by parameter maps in ways similar to *snakes* [Kass et al. 1988] and *level sets* [Sethian 1999].

Artistically, our maze creation approach, like ink diffusion [Chu and Tai 2005], is an example of algorithms that visually evolve over both space and time. Such algorithms transcend being static image filters by offering the opportunity of aesthetically controlling the simulation while watching it grow. We harness this creative potential using an interaction framework of dynamic regions and patterns. Region boundaries that temporally vary in shape and strength specify an artistic partition of algorithmic parameters and render styles. Patterns like preset brushes, define procedural combinations of the basic algorithmic parameters. This framework, in addition to being generally applicable to NPR simulations, allows game designers and physical maze builders to quickly create complex environments with overall control of maze topology and appearance.

1.2 Overview

The contributions of this paper are thus a synthesis algorithm for organic labyrinths and mazes and to a lesser degree a framework built on time-varying regions and patterns for the interactive control of NPR simulations. We begin with a description of our labyrinth and maze generation algorithm in Section 2. Section 3 then builds a high-level interaction framework of regions and patterns upon the basic algorithm. Section 4 presents implementation details and discusses resulting labyrinths, mazes and render styles. Section 5 provides the conclusion and directions for future work.

2 Labyrinth and maze model

As motivated in Section 1.1, the input to our algorithm is a set of curves on a 2D manifold. Assume that the curves have N sample points and that the average distance between these points is D. In an iterative process, the position of each sample point, $\mathbf{p_i}(1 \leq i \leq N)$, is updated by adding force vectors: $\mathbf{p_i}' = \mathbf{p_i} + \mathbf{B_i} + \mathbf{F_i} + \mathbf{A_i}$, where $\mathbf{B_i}$, $\mathbf{F_i}$ and $\mathbf{A_i}$ are the forces corresponding to Brownian Motion, fairing, and attraction-repulsion, respectively.

In order to vary the curve structure spatially, the various parameters used in the force calculations are represented by functions $f : R^2 \mapsto R$. Similarly, a function $\delta : R^2 \mapsto (0,1]$ is used to control the scale of the patterns and support self-similiarity (see Figure 9 for a parameter study of δ). To provide user feedback in our interactive system, all of these functions are conveniently represented as 2D texture maps.

Brownian Motion: To control random structural variations, a random offset vector (chosen stochastically based on a Normal Distribution with mean 0 and variance σ), $\mathbf{z_i}$, is added to each sample point, $\mathbf{p_i}$, using the equation (see Figure 6a):

$$\mathbf{B_i} = f_B(\mathbf{p_i}) \cdot \mathbf{z_i} \cdot \delta(\mathbf{p_i}) \cdot D$$

where $f_B : R^2 \mapsto R$ modulates the amplitude of the offset.

Fairing: To simulate local smoothness, a Laplacian term is added (see Figure 6b):

$$\mathbf{F_i} = f_f(\mathbf{p_i}) \cdot \left(\frac{(\mathbf{p_{i-1}} \cdot \delta(\mathbf{p_{i+1}}) + \mathbf{p_{i+1}} \cdot \delta(\mathbf{p_{i-1}})}{\delta(\mathbf{p_{i-1}}) + \delta(\mathbf{p_{i+1}})} - \mathbf{p_i}\right)$$

where $f_f : R^2 \mapsto [0;1]$ allows the fairing to vary spatially. The weighted average above prevents samples from migrating in the direction of $\nabla\delta$ (i.e. towards more coarsely sampled areas).

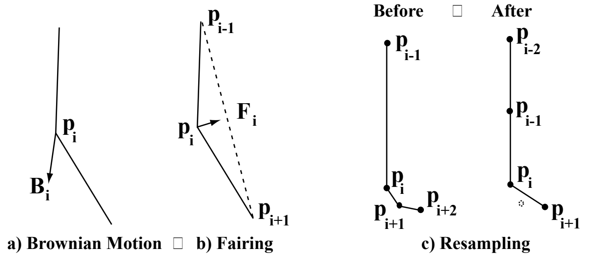


Figure 6: Brownian Motion, fairing, and resampling steps.

Attraction-Repulsion: Sample points along the curves exert attraction-repulsion forces on each other within a radius of influence, R_1 (see Figure 7). Specifically, the attraction-repulsion force, $\mathbf{A_i}$, is the sum:

$$\mathbf{A_i} = f_a(\mathbf{p_i}) \cdot \textstyle\sum_{j=1}^{N} \mathbf{f_{ij}}$$

where $f_a : R^2 \mapsto R$ modulates $\mathbf{A_i}$ spatially and $\mathbf{f_{ij}}$ is the force exerted on $\mathbf{p_i}$ by curve segment $(\mathbf{p_j p_{j+1}})$:

$\mathbf{f_{ij}} = \frac{\mathbf{p_i} - \mathbf{x_{ij}}}{|\mathbf{p_i} - \mathbf{x_{ij}}|} \cdot w(\frac{|\mathbf{p_i} - \mathbf{x_{ij}}|}{D \cdot \delta(\mathbf{p_i})})$ if $|\mathbf{p_i} - \mathbf{x_{ij}}| < min(\delta(\mathbf{p_i}), \delta(\mathbf{x_{ij}})) \cdot R_1$ and $max(|j-i|, |j+1-i|) > n_{min}$; 0 otherwise.

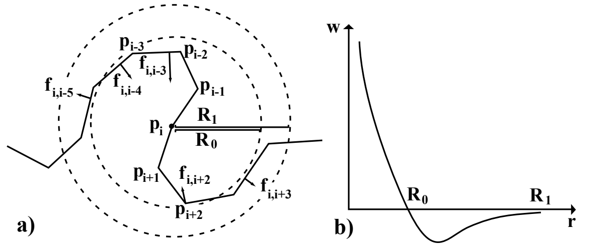


Figure 7: Attraction-Repulsion forces exerted on $\mathbf{p_i}$ by nearby segments.

Here, $\mathbf{x_{ij}}$ is the closest point to $\mathbf{p_i}$ on the segment $(\mathbf{p_j p_{j+1}})$, n_{min} is explained below, and $w : R \mapsto R$ is the Lennard-Jones potential:

$$w(r) = ((\tfrac{\sigma_{LJ}}{r})^{12} - (\tfrac{\sigma_{LJ}}{r})^{6})$$

where σ_{LJ} is a Lennard-Jones specific parameter describing the shape of the function (see e.g. [A. Ya Kipnis 1996] for details). Since the effect of σ_{LJ} can be difficult to visualize, we find it easier

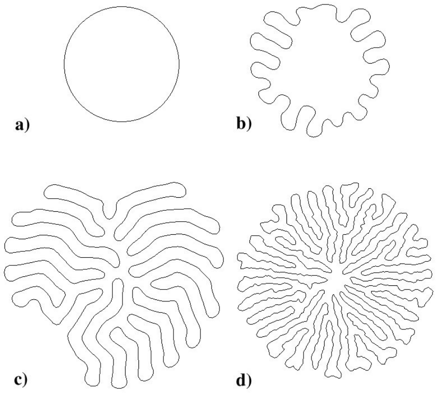


Figure 8: Evolving labyrinths. a) Input. b) Partial evolution (f_B low, f_a high). Result: c) (f_B low, f_a high). d) (f_B high, f_a low).

to specify its 0-crossing, $R_0 = k_0 \cdot D$ (see Figure 7), instead. For efficiency, w is clamped to 0 for $r > R_1$ (we have found this approximation to work well), where $R_1 = k_1 \cdot D$. k_0 and k_1 are global variables that can be controlled by the user.

To prevent strong tangential forces from causing numerical problems, neighboring samples along a curve do not exert forces on each other. An added benefit of the fairing term is that it serves to preserve the structural integrity of the curves locally. Specifically, the n_{min} neighbors of each point are ignored.

Resampling: Adaptive (based on δ), dynamic sampling is used to maintain an optimal distribution of points along the labyrinth. Any segment whose length, $|\mathbf{p_i p_{i+1}}|$, exceeds a threshold, d_{max}, is split:

$$d_{max} = k_{max} \cdot D \cdot \frac{\delta(\mathbf{p_i}) + \delta(\mathbf{p_{i+1}})}{2}$$

where k_{max} is a global constant. Conversely, if the separation between any sample and its neighbors is below an equivalent threshold, d_{min}, the sample is deleted (see Figure 6c) [Sethian 1999], [Witkin and Heckbert 1994]. Note that the sampling rate can be controlled globally by changing D.

The above steps are repeated in each iteration. Figures 8a,b,c) show snapshots in the evolution of a circle with 50 sample points. In this case f_B was set low and f_a high leading to minimal branching and uniform path width. In contrast, the pattern shown in Figure 8d), which also evolved from the circle in Figure 8a), used a higher value for f_B and a lower value for f_a. Note the increased branching, high frequency texture, and ”bulging”, heterogeneous paths.

Figure 9 shows the effect of varying the scale parameter, δ. Note how the width of the branches change based on δ while the overall structure of the pattern remains the same.

Anisotropy: The orientation of the curves can be controlled by introducing a gradient field, $\nabla f_g : R^2 \mapsto R^2$, derived from the *anisotropy function*, $f_g : R^2 \mapsto R$. While anisotropy could be integrated in a number of ways, we choose a simple modification of the attraction-repulsion force, A_i (see Figure 10c):

$$\mathbf{A_i}' = \mathbf{A_i} + \frac{\nabla \mathbf{f_g}(\mathbf{p_i})}{|\nabla \mathbf{f_g}(\mathbf{p_i})|} \cdot (\nabla \mathbf{f_g}(\mathbf{p_i}) \cdot \mathbf{A_i})$$

The force is simply scaled in the direction of the field. The effect can be seen in Figure 10a,b). Note that the magnitude of the gradient affects the characteristics of the anisotropic structure. Figures 1,16 show more advanced applications of anisotropy.

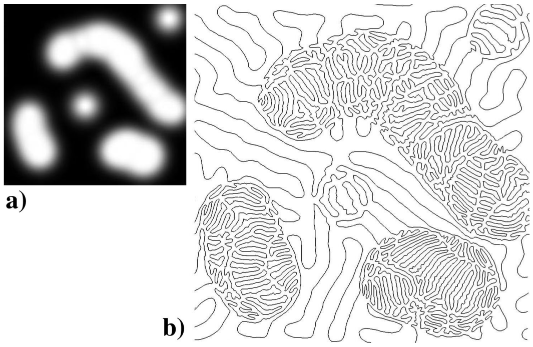


Figure 9: b) Study of the scale parameter, δ. The input curve was a circle with 50 sample points. The greyscale image in (a) was used to vary δ spatially.

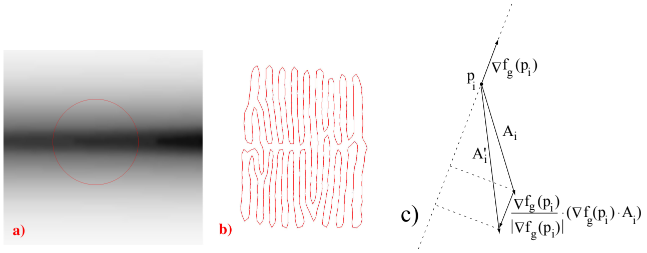


Figure 10: a) Curve with anisotropy function, f_g. b) Resulting Labyrinth. c) Calculation of $\mathbf{A_i}'$.

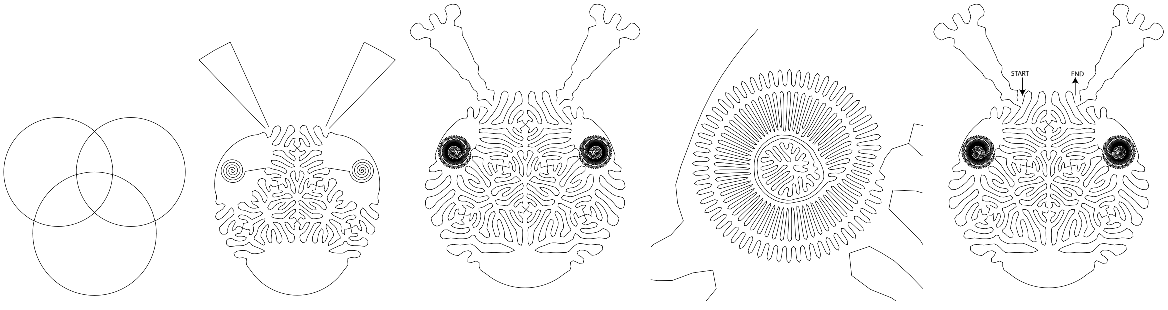
3 User interaction

The equations described above automatically generate mazes based on an initial configuration of curves. The complex relationships between algorithm parameters, however, requires understanding and experimentation to interactively control the visual result by solely adjusting individual parameters. We thus introduce three additional curve types, presets and editing mechanisms to give a user high-level interactive control over the maze topology and appearance.

We first classify curves in our system into the following types:
Labyrinth curves, described in Section 1 and 2, define the visual structure of the labyrinth or maze. The simulation rate of selected points on labyrinth curves can be retarded of frozen interactively.
Boundary curves define regions on a manifold. Their shape and strength can be user animated over time. Boundary curves act like walls for points on labyrinth curves, contributing strength modulated attraction-repulsion forces to the labyrinth curves. Boundary curves themselves, thus, need only be resampled during simulation of the discrete equations in Section 2.
Boundary Gap curves are sections of boundary curves with zero strength, where labyrinth curves can move freely across adjacent regions. More importantly they also denote areas of parameter blending that need special attention to avoid numerical problems.
Solution curves allow users to define the solution paths in a maze. They are resampled similar to boundary curves and attract/repel labyrinth curves, but with a repulsion radius of $R_0/2$. Labyrinth curves thus grow on either side of solution curves without crossing them. In the final maze the solution curve is removed to reveal a path of width R_0 bounded only by labyrinth curves.

3.1 Maze topology

Section 1 shows the planar graph induced by labyrinth curves to define maze topology. Users can interactively create new curves



(a) Initial curves (b) Eyes, ears added (c) Final curves (d) Left eye (e) Final maze topology

Figure 11: Evolution of a maze, *"Biker Bunny Tattoo"*. See text for details.

and delete or reconnect evolving curves to alter maze topology during simulation. Colored tracers placed at key points on the curves allow a user to visually track the evolving maze (see accompanying video) and aid in making topology changes. Figure 11 shows a simple interactive example created with constant parameter values. After the initial curves in Figure 11a) evolve to Figure 11b), curves for the ears are drawn and spirals for eyes connected to the existing curves. Figures 11c,d) shows subsequent maze evolution. The start and finish in Figure 11e) are then created by deleting curve segments to creating openings in the original circular regions.

3.2 Maze appearance

Regions: User defined stencils, regions or masks are commonly used to spatially localize image filters. Our regions dynamically specify a high level artistic partition of algorithmic and render styles. Figure 12a) shows a set of user specified Boundary and Gap curves. Boundary curves, rendered as solid lines and Boundary Gaps as dashed lines together divide the 2D manifold into *regions*. Regions also confine the growth of labyrinth curves contained within. Labyrinth curves migrate to adjacent regions through dynamic gaps where algorithmic parameters of adjacent regions to blend into each other.

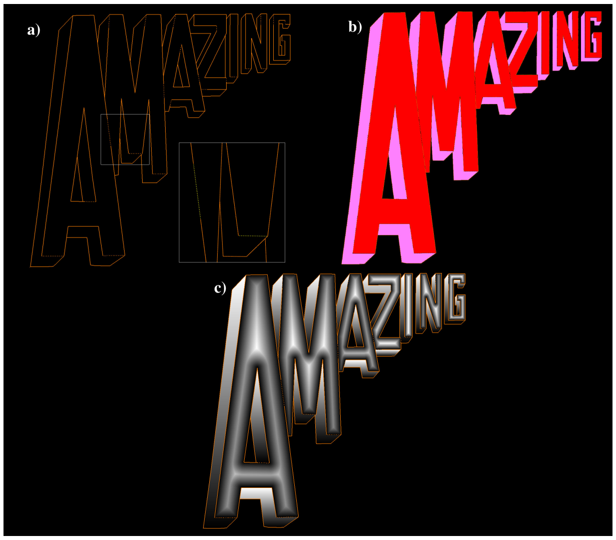


Figure 12: a) Boundaries and Boundary Gaps. b) Regions defined by boundaries. c) Anisotropy function, f_g.

Patterns: Each region is assigned a pattern. A pattern is a combination of time varying parameter values or textures that result in characteristic maze patterns. Generating new patterns is an experimental process but a pattern library allows the simple update and reuse of interesting patterns. The mapping between regions and patterns is conveniently represented by a texture map, which is also used for interactive visualization (see Figure 12b). Each pattern defines a complete configuration of default values for the low-level parameters (along with their variation during the simulation). In addition, each pattern takes its own set of high level input parameters, such as *radius*, *angle* or *noise*, that procedurally generate low-level parameter maps when assigned to regions. Figure 12c) shows an example of an automatically generated parameter map for the anisotropy function, f_g, using two patterns: *Horizontal Stripes* (marked in pink in Figure 12b) and an *Onion Ring* pattern (marked in red) mimicking the iso-contours of the distance to the region boundaries.

Blending: Specifying parameter maps independently for each region can lead to abrupt changes across region boundaries. This poses no problem for strong boundary curves, as labyrinth curves do not cross them. Near boundary gap curves, however, sharp gradients can cause the simulation to become numerically unstable.

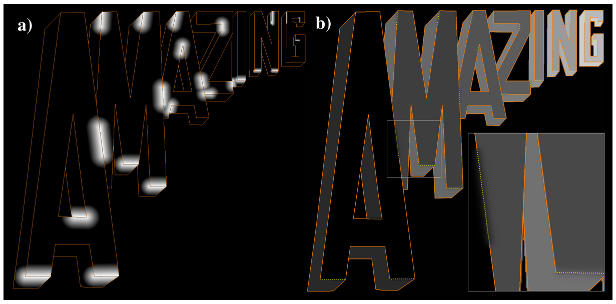


Figure 13: a) Blend map. b) Map applied to scale function (δ).

We address the problem by automatically blending all parameter maps near boundary gaps using a *blend map*. The blend map, shown in Figure 13a) for curves shown in Figure 12, is the sum of distance based implicit functions around each boundary gap curve (decaying over a specified *blend radius*). The blend radius is further modulated by the scale parameter map (δ, shown in Figure 13b) to avoid the loss of detail in areas with lower values of δ. Figure 13b) shows the application of the blend map in Figure 13a) to the scale parameter, causing smooth transitions around the boundary gaps.

Figure 14 shows 3 input circles (one for each connected component) with 30 sample points each evolved into an $\approx 160{,}000$ point labyrinth. The sides of the letters were assigned a horizontal pattern while the letter fronts had an "onion-ring" pattern. Small interactive variations in default parameters caused the variations among regions with the same pattern. The scale parameter was used to increase the sampling (non-linearly) for the smaller letters. Note how the labyrinth curves only cross adjacent regions through boundary gaps with a natural transition of curve pattern. We now present system details and further results.

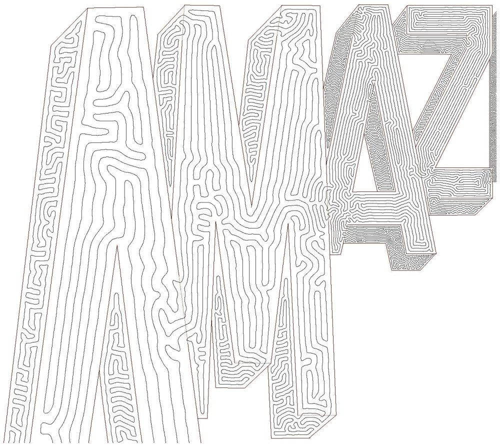


Figure 14: Final labyrinth close-up on "AMAZ".

4 Implementation and results

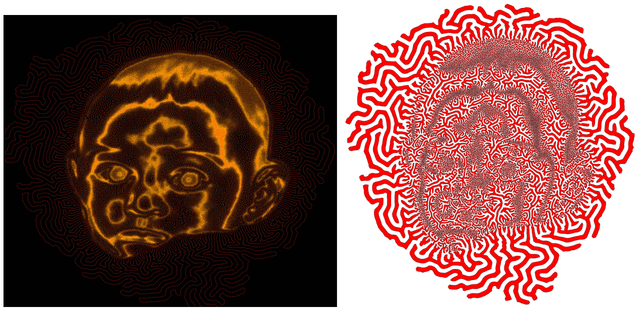
4.1 System details

The system was implemented in openGL using a 2.3 GHz Pentium IV processor. Import and export of curves from commercial modeling software as well as basic curve creation and editing operations was implemented in Section 3. The labyrinths shown throughout the paper grow at interactive rates. The maze shown in Figure 11 has 5,900 points and the interactive session for creating it took a couple of minutes. Larger examples, such as Figure 15, which have upwards of 100,000 points, took around 20 minutes of computing time. When editing the parameters of large examples comprising multiple regions, such as in Figure 14, we found localizing the simulations to selected regions convenient for optimizing performance. The bottleneck of the implementation is the computation of attraction-repulsion forces. We used a quad-tree to find the k nearest neighbors to each sample.

Due to the repeated subdivision of the curves, care must be taken to ensure memory locality. At regular intervals, the sample point vector is defragmented. Further, balancing the low-level parameters is also critical for the performance. For example, high values for f_B and f_a and low for f_f tend to increase the rate of growth of the labyrinth, but with that also the risk of numerical instability. Providing a library of predefined patterns with optimal parameter settings thus also helps improve performance. We found the following parameter ranges in our implementation to be stable and produce interesting results: $f_f \in [0.005; 0.3], f_B \in [0; 0.2], k_0 \in [0.1; 0.3], k_1 \in [1.5 \cdot k_0; 2.5 \cdot k_0], f_a \in [0; 10], n_{min} \in \{1, 2\}, k_{min} = 0.2$, and $k_{max} = 1.2$. δ should start at 1 and be reduced gradually for best results (see Figure 15). We emphasize that interactive editing of parameters is not required and a user may rely on a predefiend pattern library for automatic curve generation.

4.2 Results

Self-similarity. Figure 15 shows the effects of varying the scale parameter, δ, by up to 2 orders of magnitude based on the intensities in an input image. In this simulation, an initial labyrinth was first computed with a constant value for δ. δ was then gradually decreased based on the intensity of the input image.



(a) OpenGL rendering

(b) Self-similarity

Figure 15: Scale and self-similarity. a) OpenGL rendering, $\delta \in [0.01; 1.0]$. b) Alternative visualization, $\delta \in [0.02; 1.0]$.

The labyrinth shown in Figure 15a) has 220,000 points. It evolved from a circle with 50 points. The curve segments are shown in red and the sample points in orange. In Figure 15b) (110,000 points), the interior of the original circle is shown in red.

Anisotropy. Figure 16 shows an application of the anisotropy function obtained by defining f_g as the distance to the region boundaries, reproducing the maze art of Scott Sullivan [Sullivan 2000]. Smooth cretan labyrinths like Figure 2a), 3c) can be built similarly with simple anisotopy maps and a high fairing term. A different example, utilizing a procedurally defined gradient field, is shown in Figure 1. Here, high image intensities were mapped to vertical gradient vectors and low intensities to horizontal gradients. The magnitude of the gradient was scaled by the intensities.

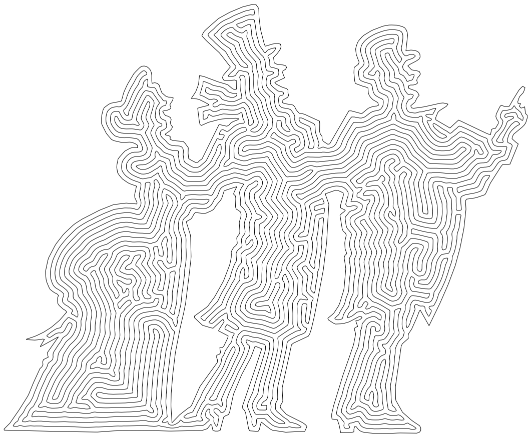


Figure 16: Labyrinth aligned with boundary curves.

Dali's *"Labyrinth"*: The labyrinth in Figure 17 was inspired by the features of a Dali painting. First, boundary curves were drawn to match the painting's feature lines. Patterns were then assigned to the different regions, using presets and procedural parameter maps computed from the boundary curves and the original image. The labyrinth started as a small closed loop with 20 points. Boundary curves initially prevented the labyrinth from crossing features. Near the end of the simulation, boundary curve strength was gradually reduced to zero, causing the sharp boundaries to disappear and the labyrinth curve to organically fill the painting.

Theseus and the Minotaur: Figure 18a) shows a Greek drinking bowl (c. 440-430 BC) depicting a scene from the myth of Theseus and the Minotaur. Figure 18b) shows the corresponding regions.

We chose to replace the meandering pattern near the perimeter of the bowl with one from our system's pattern library. The image shown in Figure 18a) was used to define the scale parameter (δ) and an anisotropy function (f_g) was specified as a blurred version of the same image.

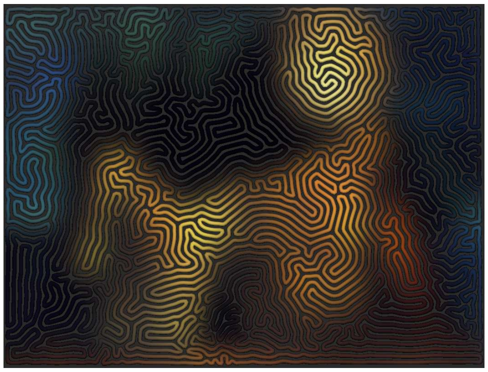


Figure 17: Labyrinth inspired by a Salvador Dali painting.

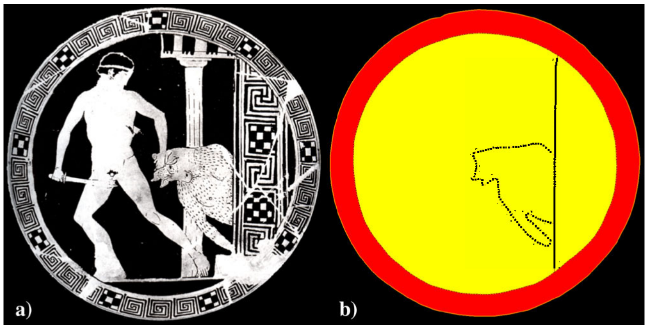


Figure 18: a) ©2002, Hermann Kern. b) Regions and patterns.

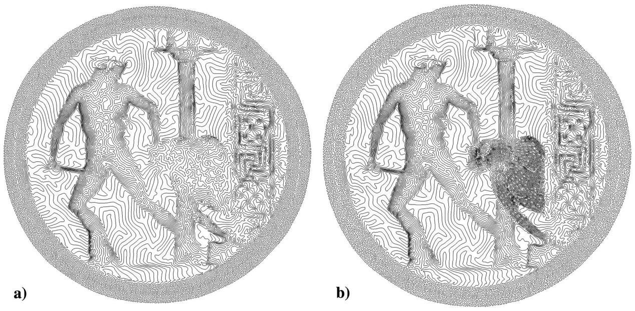


Figure 19: a) Initial result. b) After local modification.

Figure 19a) shows the resulting labyrinth. The magnitude of the gradient was adjusted to get extra curve density near high contrast areas. As can be seen in Figure 18a), the original parameters worked well in most areas but failed to produce sufficient definition of the Minotaur. To address this issue, the problematic region was manually outlined by a boundary gap curve (see Figure 18b) and the parameters for this subregion changed. The curves then evolved from the state in Figure 19a) to the final result in Figure 19b). Figure 20 shows several close-ups of Figure 19.

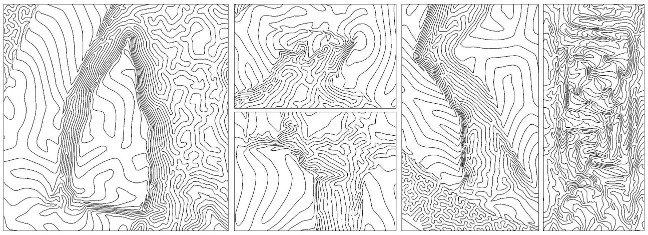


Figure 20: Close-ups of Figure 19.

Planair Mazes: The techniques decribed in this paper are general to 2D manifolds but the quad-tree based optimization and all the examples seen thus far have been either planar or toroidal (Figure 1). Figure 21 shows an example of a *planair maze* embedded on 3D geometry. All steps in the algorithm generalize directly to 3D. We would like to investigate the use of properties like curvature, lighting, depth, etc. as input parameters for planair mazes, in future.

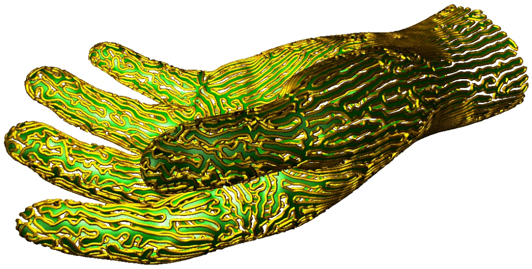


Figure 21: Planair Maze

Render Styles: The resulting curves are conducive to rendering in various ways. They may be used to extrude geometry (Figure 22a), grow particle systems (Figure 22b), or create calligrams and variable micro-lettering (Figure 23). They can also be used as long continuous brush strokes (Figure 22c) that can vary in thickness based on parameter maps (Figure 22d).

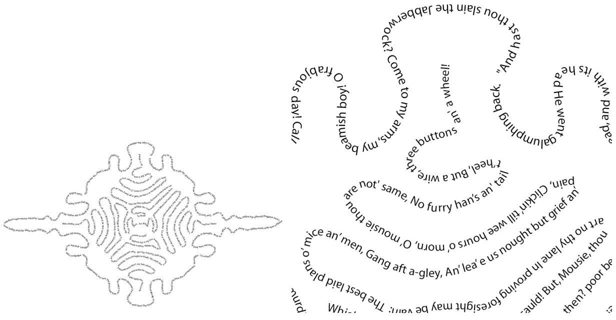
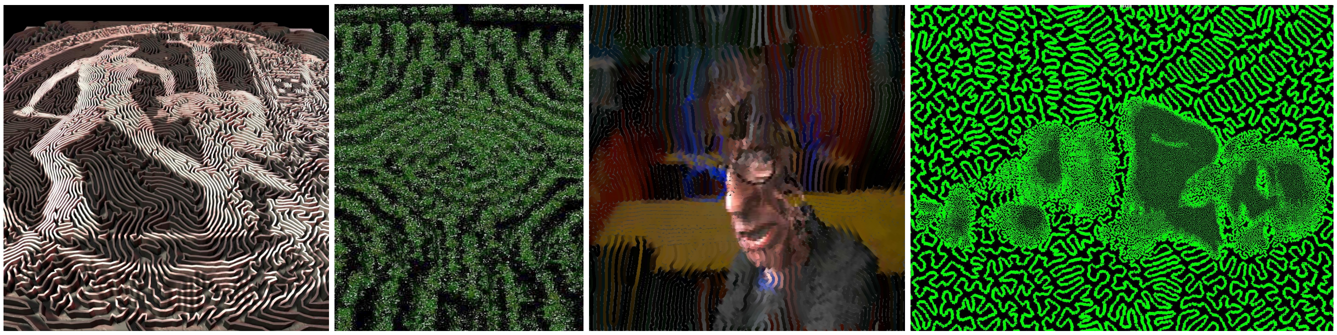


Figure 23: Micro-lettering along labyrinth (apologies to Robbie Burns).

5 Conclusion and future work

This paper introduces readers to a historically fascinating symbiosis of art and science. While a unified theory for maze contruction and solvability remains an open problem, we have presented an algorithm for the generation of organic looking labyrinths and mazes. The approach is not ideally suited to structured mazes but the resulting curves can be pushed by forces towards grid-lines and then postprocessed into a grid-maze by selecting the grid cells traversed by the curves (see Figure 1c). We also embed the algorithm parameters within a general high-level interaction framework, to provide NPR simulation users with greater creative control. This framework of time-varying regions, defined by boundary and gap curves, controls both seperation and mixing of simulation parameters.



(a) Extruded walls (b) Hedge Maze (c) Smudge along vertical strokes (d) Variable line-width

Figure 22: Labyrinth and maze rendering styles

Feedback from game and physical maze designers indicates that interactive programs like ours will allow the creation of richer environments, without sacrificing overall design control. The output patterns also show promise for use in watermarking, image compression and multi-variable visualization. [Dafner et al. 2000] propose a space-filling curve that attempts to follow image gradients. Our curves have similar properties of good autocorrelation and thus could watermark or compress images by encoding the 1D image stream along with the algorithm parameters and initial curves. The visual effect of multiple texture maps can be quite clear on the curve generated by our algorithm. As a result our labyrinthine structures may help in the visualization and understanding of multi-variable texture maps. In the future, we also hope animate our labyrinths and mazes using video textures.

References

A. Ya Kipnis, B. E. Yavelov, J. S. R. 1996. *Van Der Waals and Molecular Science*. Oxford University Press.

Berg, C. 2001. *Amazeing Art*. HarperCollins Publishers Inc.

Chu, N. S.-H., and Tai, C.-L. 2005. Moxi: real-time ink dispersion in absorbent paper. *ACM Trans. Graph. 24*, 3, 504–511.

Conty, P. 2002. *The Genesis and Geometry of the Labyrinth*. Inner Traditions International.

Dafner, R., Cohen-Or, D., and Matias, Y. 2000. Context-based space filling curves. *Computer Graphics Forum 19*, 3 (August). ISSN 1067-7055.

DeCarlo, D., and Santella, A. 2002. Stylization and abstraction of photographs. *ACM Transactions on Graphics 21*, 3 (July), 769–776.

Finkelstein, A., and Salesin, D. H. 1994. Multiresolution curves. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, 261–268.

Gotsman, C., and Lindenbaum, M. 1996. On the metric properties of space-filling curves. *IEEE Transactions on Image Processing 5*, 5, 794–797.

Hertzmann, A., Oliver, N., Curless, B., and Seitz, S. M. 2002. Curve analogies. In *Proceedings of the 13th Eurographics Workshop on Rendering*, Eurographics Association, 233–246.

Kaplan, C., and Bosch, R. 2005. T.s.p. art. In *Bridges*.

Karatzas, I., and Shreve, S. E. 1997. *Brownian Motion and Stochastic Calculus*. Springer-Verlag; 2nd edition.

Kass, M., Witkin, A., and Terzopoulos, D. 1988. Snakes: Active contour models. *IJCV 1*, 4 (Jan.), 321–331.

Kern, H. 1995. *Through the Labyrinth*. Prestel.

LaValle, S. 2005. *The RRT Page - Generating Random Mazes*. UIUC CS Dept.: http://msl.cs.uiuc.edu/rrt/gallery_maze.html.

Lovasz, L. 1996. Random walks on graphs: A survey. *Combinatorics 2*, 353–398.

McCullough, D. W. 2004. In *The unending mystery: A journey through labyrinths and mazes*, Pantheon.

Ostromoukhov, V., and Hersch, R. D. 1995. Artistic screening. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, 219–228.

Ostromoukhov, V. 1999. Digital facial engraving. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 417–424.

Saward, J. 2002. *Magical Paths*. Octopus Publishing Group Ltd.

Sethian, J. A. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

Sullivan, S. 2000. *The Amazing Christmas Maze Book*. Penguin.

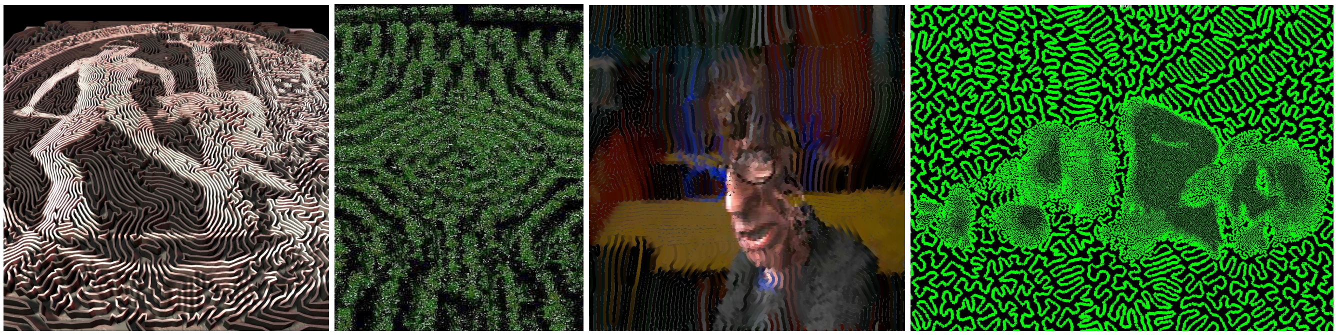
Turk, G. 1991. Generating textures for arbitrary surfaces using reaction-diffusion. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, vol. 25, 289–298.

Velho, L., and Gomes, J. 1995. Stochastic screening dithering with adaptive clustering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, 273–276.

Witkin, A. P., and Heckbert, P. S. 1994. Using particles to sample and control implicit surfaces. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, 269–278.

Witkin, A., and Kass, M. 1991. Reaction-diffusion textures. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, 299–308.

Wong, M. T., Zongker, D. E., and Salesin, D. H. 1998. Computer-generated floral ornament. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 423–434.



(a) Extruded walls

(b) Hedge Maze

(c) Smudge along vertical strokes

(d) Variable line-width

Figure 22: Labyrinth and maze rendering styles

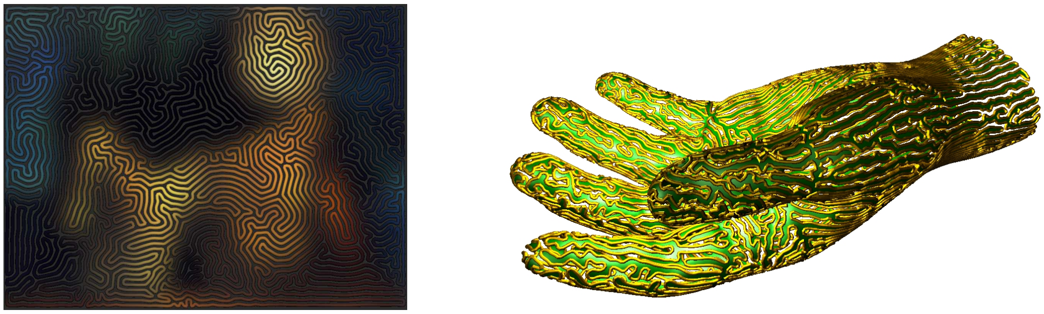


Figure 17: Labyrinth inspired by a Salvador Dali painting, and Figure 21: a Planair maze.

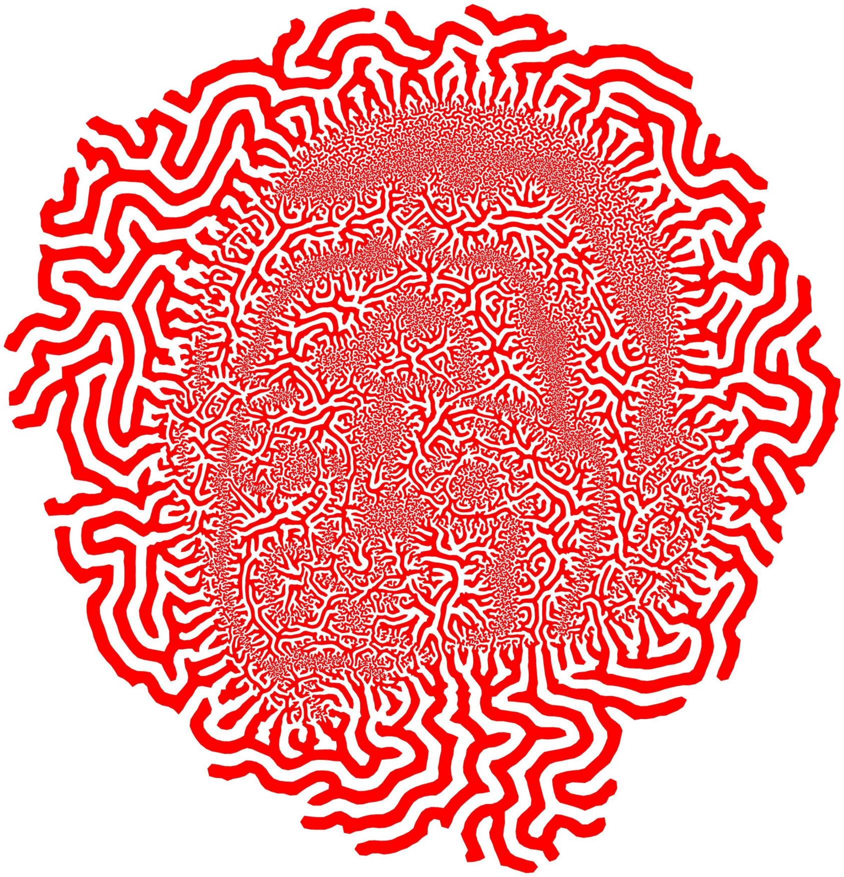


Figure 15: Scale and self-similarity. $\delta \in [0.02; 1.0]$.