WYSIWYG Design of Hypnotic Line Art

Chih-Kuo Yeh [ID], Zhanping Liu [ID], I-Hsuan Lin,
Eugene Zhang [ID], *Senior Member, IEEE*, and Tong-Yee Lee [ID], *Senior Member, IEEE*

Abstract—Hypnotic line art is a modern form in which white narrow curved ribbons, with the width and direction varying along each path over a black background, provide a keen sense of 3D objects regarding surface shapes and topological contours. However, the procedure of manually creating such line art work can be quite tedious and time-consuming. In this article, we present an interactive system that offers a What-You-See-Is-What-You-Get (WYSIWYG) scheme for producing hypnotic line art images by integrating and placing evenly-spaced streamlines in tensor fields. With an input picture segmented, the user just needs to sketch a few illustrative strokes to guide the construction of a tensor field for each part of the objects therein. Specifically, we propose a new method which controls, with great precision, the aesthetic layout and artistic drawing of an array of streamlines in each tensor field to emulate the style of hypnotic line art. Given several parameters for streamlines such as density, thickness, and sharpness, our system is capable of generating professional-level hypnotic line art work. With great ease of use, it allows art designers to explore a wide variety of possibilities to obtain hypnotic line art results of their own preferences.

Index Terms—Non-photorealistic rendering (NPR), hypnotic line art, tensor field design, streamline placement, vector field visualization

✦

1 INTRODUCTION

CANADIAN illustrator Patrick Seymour [1] created a beautiful rhythmic line art style, by which a set of white curves, with each tapering towards the two ends, is drawn over a black backdrop to exhibit high contrast in a 2D image. As the curves vary in the direction, thickness, and density, the resulting pattern of stripes brings 3D shapes and structures out of a planar delineation. Patrick Seymour refers to his line drawings as 'Hypnotic Line Art', a term that we adopt throughout this paper. Compared to computerized synthesis, manual line editing is an arduous and lengthy process, in which artists duplicate curves, adjust the positions, and fine-tune the curvatures through trial and error to attain a desired layout. Our research on hypnotic line art design is intended to significantly reduce the cumbersome workload of manual drawing, by delivering a smart interactive system to ordinary users for semi-automatic production of art work.

There have been many applications of computer graphics to visualization of vector and tensor data. This paper is

- *Chih-Kuo Yeh is with the School of Computer Science and Software, Zhaoqing University, Zhaoqing 526061, China.*
  E-mail: simpson.ycg@gmail.com.
- *Zhanping Liu is with the Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University, Norfolk, VA 23529 USA.*
  E-mail: zhanpingliu@hotmail.com.
- *I-Hsuan Lin and Tong-Yee Lee are with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 701, Taiwan, ROC. E-mail: sherrygottalent@hotmail.com, tonylee@mail.ncku.edu.tw.*
- *Eugene Zhang is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA.*
  E-mail: zhange@eecs.oregonstate.edu.

primarily focused on tensor field design, evenly-spaced streamline placement, and artistic curve drawing for hypnotic line art generation. While this research is motivated by the line art form of Canadian illustrator Patrick Seymour [1], the goal is to provide hobbyists with an intuitive interactive system for ease of use in innovative art design. Fig. 1 shows a line art image created using our method to mimic the illustration style of Patrick Seymour.

Our method builds tensor fields in which integral curves are then generated, distributed, and eventually drawn with a suite of line style parameters. As the input, an image is partitioned into a number of regions, while a tensor field is constructed for each under the guidance of strokes from the user. Tensor fields have seen a broad range of applications in computer graphics and data visualization [2], [3], [4]. One major advantage of tensor fields consists in the ability to encode bi-directional tangential information instead of unidirectional vectors. Specifically, tensor fields facilitate streamline integration by working around the complications of negative-positive signs that are present in vector fields. On the other hand, the geometric shape and the topological structure of a tensor field can be represented by an elegant layout of streamlines with an approximately uniform density. Then, the pattern of streamlines steers interactive modification of the underlying tensor field for advanced iterative design. As for tensor field design in our work, the user just needs to sketch a few strokes to outline the basic appearance of the domain, followed by editing operations if necessary. Tensor field design based on such simple strokes is an interesting research topic, with great promise to art engineering. In this paper, we present an efficient interactive approach to hypnotic line art authoring by incorporating tensor field design with evenly-spaced streamline placement. With only brief user input yet without laborious effort, this method creates tensor fields, one per sub-domain stemming from the segmentation of a 2D image, which are then filled with visually appealing streamlines to convey a
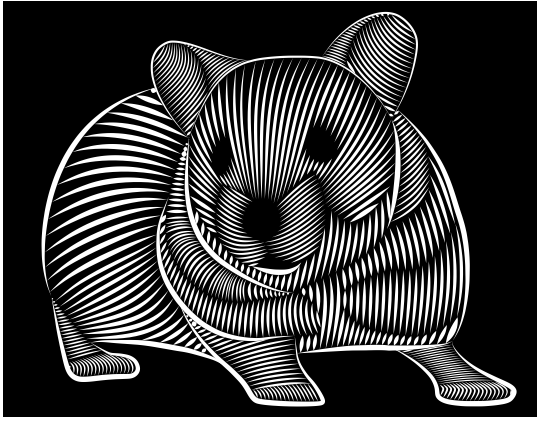
Fig. 1. An image created by using our method in the hypnotic line art form of Patrick Seymour.

3D look and feel about the objects therein. The WYSIWYG feature of the semi-automatic pipeline is well suited for amateurs to design, appreciate, and share hypnotic line art work with fancy effects. The proposed method can be demonstrated by a practically applicable system or tool with the following characteristics:

- Ease of use: intuitive and interactive for hobbyists to design hypnotic line art.
- Customizability: able to satisfy the preferences of a user via the WYSIWYG paradigm.
- Robustness: capable of generating topologically correct tensor fields from strokes and placing uniform streamlines.

Our method differs from previous work in three major aspects. First, users are asked to draw curves instead of painstakingly specifying or marking singularities. Second, evenly-spaced streamlines are employed to populate each region to augment the description of 3D surface shapes. Third, adaptive control is attained over streamlines in the density and width for each region, instead of rigid constraints for the entire image, to help accentuate depth cueing. The main contributions of this paper are as follows.

- By deriving smooth tensor fields from image segmentation, we exploit the tensor field of each region to govern the layout of hypnotic lines therein which can well comply with user strokes.
- We adapt an advanced evenly-spaced streamline placement algorithm, originally designed for vector fields, to tensor fields such that the constituent segments of an image can be filled rapidly and robustly with high quality.
- We propose a novel method that makes use of width-tapering effects to stylize topological contours, mimicking the line art style of Patrick Seymour.

The remainder of this paper is organized as follows. Section 2 reviews previous work related to our approach. In Section 3, we present WYSIWYG design of hypnotic line art in detail. Results and discussions are provided in Section 4 to demonstrate the effectiveness of the proposed method. Section 5 concludes this paper with a brief summary and outlook on future work.

2 RELATED WORK

There has been significant research in non-photorealistic rendering (NPR) for hatch-based illustration of photos or 3D objects. Among the pioneering efforts is a pen-and-ink method, proposed by Salisbury et al. [5], which allows the user to edit a direction field by rotating and placing stroke textures. However, it is not well suited for hypnotic line art because pen-and-ink strokes are relatively short. Zander et al. [6] improved hatch line rendering quality by integrating streamlines in a 2.5-dimensional vector field (i.e., a 2D manifold) that is derived from the point-wise principal curvature direction on the surface. Later, Gerl and Isenberg [7] investigated style transfer for synthesizing the hatch illustration of a 3D mesh, with the drawing style gained from example images through a learning model. Another traditional illustration method is rendering images in a copperplate [8]. Most of these methods rely on 3D models to add illumination to the rendering of hatch lines. Similar to Johnston's Lumo [9], our work does not need explicit shading. Instead, we extract some salient topological information by segmenting a 2D image. Then, 3D geometric shapes are conveyed with depth cues, to attain shading effects, by drawing a layout of evenly-spaced hypnotic curved lines of varying width.

Ostromoukhov [10] presented a very simple technique that produces engravings on digital photos by compositing a stack of black/white and color engraving layers. Cappelli et al. [11] proposed a method for fingerprint synthesis, which iteratively generates ridge-line patterns from a number of randomly distributed spikes. Such fingerprint patterns or zebra striping patterns bear a resemblance to parallel curves utilized in hypnotic line art. The former often exhibit bifurcations which though do not occur in the latter. Hertzmann and Zorin [12] designed a method for automatically generating hatches and cross-hatches to convey the surface shape. Rössl and Kobbelt [13] developed an interactive system that assists the user in creating line art work, with long single or cross-hatching lines of varying thickness to illustrate 3D models. Elber and Cohen [14] devised an adaptive spacing method for rendering a freeform surface with a set of iso-curves. Similar to but more than this strategy, our work provides the user with interactive design. Later, Elber [15] brought forth another approach to emulate Victor Vasarely's art work, demonstrating that 3D illusions may come into being by means of distorted parallel stripes. Also employed in our work, parallel curved stripes are further equipped with width tapering. Kalnins et al. [16] presented a system by which the user can directly apply stylized line strokes to 3D models. Kim et al. [17] proposed an image-space oriented technique in support of real-time pencil rendering and animation of 3D dynamic models. A similar technique by Saputra et al. [18] deforms and assembles ornamental elements to follow the strokes sketched by the user.

Related to our work is an automatic solution by Maharik et al. [19] for digital micrography images, of which each is formed by arranging words or letters at a small scale. Unfortunately, one constraint with this method is that the vector field, constructed for each region, needs to be aligned with part of the boundaries. Then, the boundary condition has to be optimized so as to suppress high-curvature streamlines because the text orientation is controlled by streamlines.

In contrast, our approach is free from this constraint, as shown by the ears, face, and body in Fig. 1. Streamlines in each of these regions are not aligned with any boundary. Instead, they comply with the principal curvature direction of the surface being portrayed by a 2D image. Thus, our method provides the user with great flexibility in that the user just needs to sketch some strokes within several regions or along the boundaries. For each region, we resort to a tensor field such that a wedge or trisector pattern, unavailable in any vector field, can be easily created to meet special needs.

Vector/tensor field design may offer flexible powerful control over topological patterns and surface shapes, with many applications such as painterly rendering, hatch-based illustration, and texture synthesis to achieve a variety of visual effects. Zhang et al. [2] proposed an approach for tensor field design that enables interactive editing of surfaces. Palacios and Zhang [3] demonstrated pen-and-ink sketching as well as quad-dominant re-meshing via operations on the topology of a rotational symmetry field. Xu and Kaplan [20] devised a method for maze generation, by which an image is decomposed into regions and a tangential or orthogonal vector field is specified to govern the placement of the separating walls. Chen et al. [21] interactively modeled a street network by either editing the underlying tensor field or by modifying a graph representation of the street network. Yao et al. [22] presented a paradigm of vector field design by piecing together building blocks called elementary fields that are generated from discrete harmonic functions. However, all these methods are counter-intuitive for amateur users. They require considerable technical skills of the user to employ software programs for design.

Streamlines remain a straightforward and efficient geometric representation [23] for visualizing planar, surface, and volume flows such as ocean currents, wind flows, and CFD fluids. With each spawned and integrated from a seed position, streamlines tend to result in either a coarse view or a cluttered image unless effective control is imposed over seeding and/or density. Turk and Banks [24] presented a seminal method for placing even-spaced streamlines by low-pass filtering the bi-variate image of the layout to measure the uniformity. An initial placement undergoes iterative refinement by inserting, deleting, merging, lengthening, or shrinking streamlines to minimize the non-uniformity that is essentially an energy function. The high placement quality comes with the heavy computational cost.

Jobard and Lefer [25] opened up another line of research on evenly-spaced streamline placement by proposing a geometry-based method. Given a threshold density in the form of a minimum separating distance, uniform samples are generated along each streamline as it is integrated using a fixed step size. Then, inter-streamline distance control is approximated and implemented by inter-sample distance control, which checks whether the advancing front of the streamline being integrated satisfies the threshold separating distance from all samples of each and every existing streamline. Only if the forefront sample is accepted by the inter-sample distance controller, can the streamline continue to grow. Any sample rejection immediately terminates its integration. A greedy placement is driven by an FIFO queue of candidate seeds that survival streamlines introduce. Liu et al. [26] presented a variant of this geometry-

based approach to enhance both placement quality and computational performance. Other extensions involve unsteady flows [27], volume flows [28], [29], multi-resolution representation and exploration [30], and view-dependent layout [29], [31], [32]. In addition, there has been research on seeding mechanisms, e.g., selecting farthest candidate seeds through Delaunay triangulation [33], distributing seeds along virtual streamlines of the orthogonal/dual vector field [34], [35]. Although not intended for uniform streamline placement, some seeding strategies based on flow topology [36], [37] are inspiring, capable of producing an insightful view of the flow pattern. Etienne and Lefebvre [38] proposed an implicit procedural method for arranging and drawing ribbons in a parametric field, instead of explicit integration and placement of streamlines in a vector field, though it cannot produce width tapering effects.

3 WYSIWYG DESIGN OF HYPNOTIC LINE ART

This section presents our method for WYSIWYG design of hypnotic line art, beginning with a high-level description of the pipeline. We then provide a treatise of each of the stages, i.e., image segmentation, a unique characteristic of second-order symmetric tensor fields and generation of such a tensor field for each region, integration and uniform placement of streamlines in a tensor field, artistic rendering of evenly-spaced streamlines with special visual styles for accentuating depth cues, addition of contour lines for strengthening surface shapes, and user interface for convenient interactive design of hypnotic line art work.

3.1 The Pipeline

The pipeline of WYSIWYG design of hypnotic line art is shown in Fig. 2, of which the basic idea is to take a reference image (Fig. 2a) as the input to export a visual representation like Fig. 2d. In the preprocessing step, the user manually partitions the reference image into a set of regions using a pen tool in combination with a graph-cut method. Next, the user sketches several strokes in each region to fulfill a schematic design of the pattern of interest (Fig. 2b). Then, a tensor field is constructed for the region and smooth integral curves (i.e., high-fidelity streamlines) are automatically created (specifically, traced in the tensor field) to conform to those strokes to fill the void (Fig. 3). These curves are clipped against the boundaries of each arbitrarily-shaped region before all regions are connected together (Fig. 2c). Finally, we decorate these boundaries with stylized curves to yield a hypnotic line art image (Fig. 2d). This entire pipeline can be iterated, as is usually the case, for the user to refine the line art design until a satisfactory result is obtained. These steps are detailed, along with necessary results, in the following sub-sections.

3.2 Image Segmentation

Given a picture as the reference, our method begins with image segmentation. The user employs a pen tool to manually trace around the components of an entire object to identify the boundaries (Fig. 4a), which facilitates the subsequent image segmentation process. In fact, we provide a brief user interface for the graph-cut approach [39] by which each region is defined with mark-ups and then extracted from the reference image. In more detail, the user draws two strokes
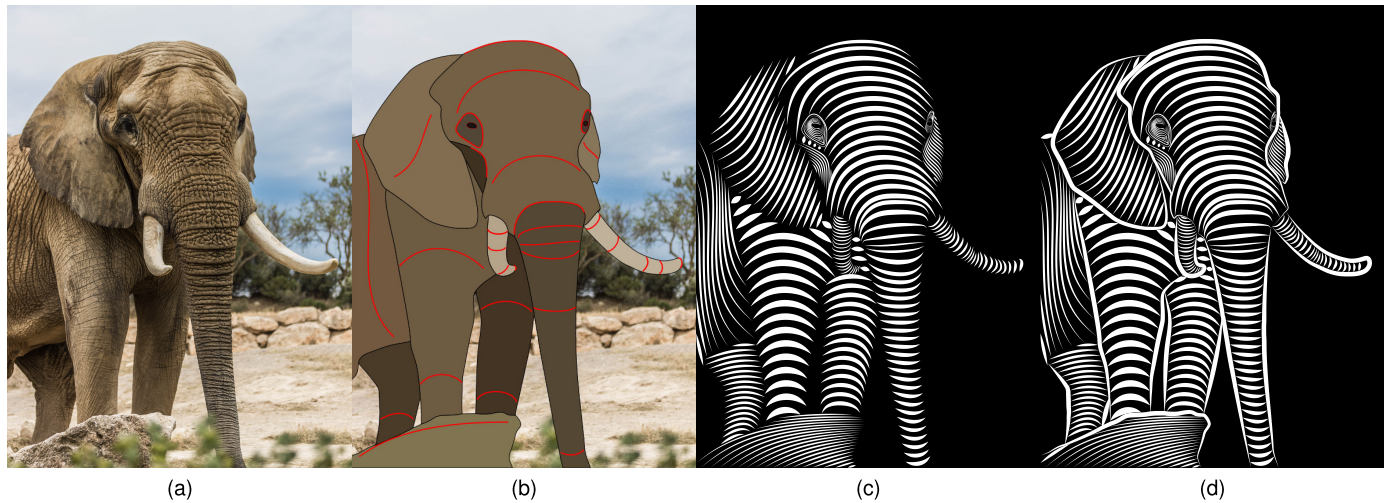
Fig. 2. The pipeline of WYSIWYG design of hypnotic line art. (a) An input image serving as the reference. (b) Image segmentation followed by some strokes (in red) from the user. (c) Construction of a tensor field for each region, based on the strokes, in which streamlines are integrated, equidistantly placed, and then drawn as stylized curves. (d) Addition of stylized contour lines along the boundary of each region to emphasize the surface shape.

to differentiate the foreground from the background for interactive extraction of each region (Fig. 4b). This process divides the image, usually with complex structures, into regions of which each exhibits a simple flow pattern in the associated tensor field. Streamlines are generated in each region, carving out the surface shape with roughly parallel curves and silhouettes. With the optimal number of regions dependent on the input image, it is nontrivial to strike a great balance between this number and the size of each constituent region. A good segmentation means that the boundaries "follow" the silhouettes as well as ridges of the object and reflect the contours within each region such that the surface shape can be easily perceived.

3.3 Tensor Fields

To deliver a WYSIWYG functionality for interactive line art design, we adopt structure tensor instead of deviation tensor or trace-free tensor. A second-order tensor field based on a 2D coordinate system, $\mathbf{x} = (x, y) \in \mathbb{R}^2$, is represented by

$$T(\mathbf{x}) = \begin{pmatrix} T_{11}(\mathbf{x}) & T_{12}(\mathbf{x}) \\ T_{21}(\mathbf{x}) & T_{22}(\mathbf{x}) \end{pmatrix}. \quad (1)$$

A constraint imposed in our work is that the tensor needs to be symmetric, i.e., $T_{12}(\mathbf{x}) = T_{21}(\mathbf{x})$. As elucidated by Delmarcelle and Hesselin [40] in the topological analysis of a second-order symmetric tensor, a real-valued two-dimensional symmetric matrix T always has two real eigenvalues $\lambda_1 \leq \lambda_2$,

with which two mutually orthogonal eigenvectors e_1 and e_2 are associated, respectively. The corresponding eigenvector fields E_1 and E_2 are called minor and major ones, respectively. From these two eigenvector fields, we can understand the structure of the tensor field. A degenerate point occurs if and only if the two eigenvalues $\lambda_1 = \lambda_2$, with the two eigenvector directions undefined. Given a non-zero vector u, we can map it to a tensor by $T = uu^T$. This is a structure tensor, exactly what is required in our work, with two eigenvectors. The principal eigenvector, associated with eigenvalue $\lambda_2 = u^T u$, is parallel to u. The other one or the secondary eigenvector, coupled with eigenvalue $\lambda_1 = 0$, is perpendicular to u. It is worth emphasizing that eigenvectors are bi-directional and hence can be utilized to represent the local structure of an integral curve. Since $uu^T = (-u)(-u)^T$, structure tensor is "robust" in that any interpolation operation between multiple samples with opposite signs in the gradients does not give rise to cancellation effects.
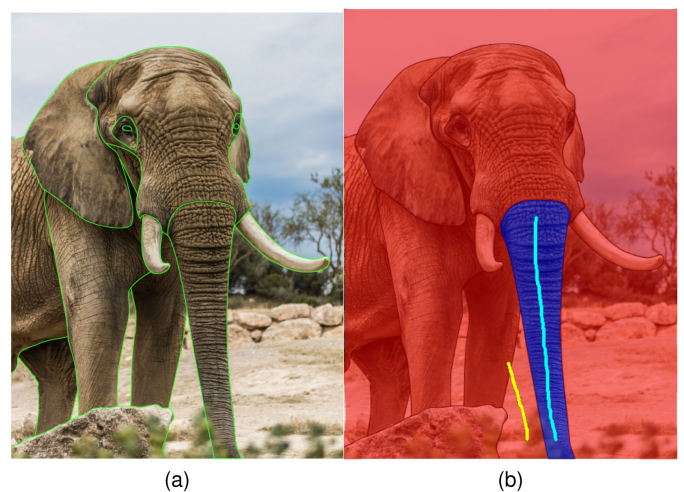
Fig. 4. Interactive segmentation of a 2D image. (a) The user traces around each part of an image of elephant with a pen tool (in green) to specify the boundary. (b) The user tells the foreground with one stroke (in cyan) from the background with another stroke (in yellow) and then the graph-cut method extracts the foreground region (the trunk). This process is repeated until all regions are obtained individually from the image.
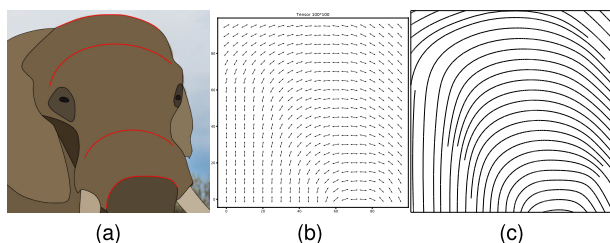

Fig. 3. (a) One region of the segmented image, with a desired pattern illustrated by three strokes (in red). (b) Construction of a tensor field to follow the strokes. (c) Integration of streamlines in the tensor field and an evenly-spaced placement.

The strokes (Fig. 3a) that the user sketches are described by $S(t) = (x(t), y(t))$, $t \in [0, 1]$. We assume that the tensor field T is locally smooth and all of the strokes (curves) S are tangent to the major eigenvector field E_2 (Fig. 3b). Then, we build a structure tensor field to encode these specifications (i.e., the strokes) of the user. Non-linear interpolation may be employed to evaluate the tensor at an arbitrary point in the field. In more detail, we establish an $N \times M$ (e.g., 100×100 in our implementation) Cartesian grid of sample points. $T(\mathbf{x})$ is represented by $T(i\Delta x, j\Delta y)$, where $i, j \in \mathbb{Z}$ are the indices and Δx and Δy are the sampling intervals. We optimize the tensor at each point by minimizing Laplacian energy function $\Delta T = 0$ with respect to the hard constraints derived from the structure tensor of each stroke $T(\mathbf{x}) = u(\mathbf{x})u(\mathbf{x})^T$, where $u(\mathbf{x}) = \frac{\partial S}{\partial t}$ is the tangent vector derived from point $\mathbf{x}$ of the stroke. The region boundaries, for $i = 0$ or M and $j = 0$ or N, are subject to the Neumann boundary conditions: $\nabla T(\mathbf{x}) \cdot \mathbf{n} = 0$, where $\mathbf{n}$ denotes the exterior normal to the boundary. These constraints guarantee that streamlines integrated in this tensor field (Fig. 3c) can precisely follow the strokes of the user. Minimizing the Laplacian energy function adds to the smoothness of the tensor field, reducing the geometric complexity that would not only incur high-curvature streamlines but also increase the number of degenerate points. As a matter of fact, degenerate points damage the coherence that streamlines are expected to exhibit. In our hypnotic line art design, one major advantage of the tensor field is that the user only sketches some strokes, without the need to specify direction and magnitude information. As opposed to a vector field, a structure tensor field does not discriminate between forward and back directions.

3.4 Streamline Generation and Evenly-Spaced Placement in Tensor Fields

Evenly-spaced streamlines may provide an aesthetic as well as informative pattern to help with perceptual assimilation by visual interpolation between neighboring, roughly parallel flow streaks. Such curves are well suited for delineating surfaces in terms of depth, shape, and orientation. We adopt an advanced evenly-spaced streamline placement algorithm, or ADVESS [26], to achieve fast high-quality layout equipped with robust loop detection. Specifically, a fourth-order Runge-Kutta integrator with adaptive step size and error control (RK4-ASSEC) is employed to attain rapid accurate generation of streamlines. Cubic Hermite polynomial interpolation is exploited to create uniform samples along each streamline such that inter-curve distance control can be approximated and then fulfilled by inter-sample distance control. To address visual cluttering, cavities are minimized by adaptive distance control based on the local flow variance. In addition, discontinuities are suppressed by a double-queue seeding strategy that prioritizes seeds distributed via topological templates (e.g., those designed for saddles and centers) and those introduced by the seeds of sufficiently long survival streamlines. Furthermore, centers and foci can be robustly detected, properly distinguished, and unambiguously depicted in a clean manner, without any redundant spiraling self-squeezing curve in the layout. Thus, ADVESS meets the requirements of hypnotic line art design in placement quality and computational performance.

To generate a streamline step by step in a 2D domain, the velocity vector at the advancing-front point needs to be evaluated for a first-order integrator (i.e., the Euler method) to determine the destination of the next "hop" (or advection). For a multi-stage higher-order integrator such as RK-2 (i.e., the mid-point method) or RK-4 (as used in ADVESS), additional vector evaluation is performed at one or several intermediate sample points. Bi-linear interpolation deals well with centers, foci, nodes, and saddles that may be present in a 2D vector field. In more detail, the process of streamline integration is terminated whenever a zero (i.e., vanishing) vector, or one with a very tiny velocity magnitude, is obtained from bi-linear interpolation across a cell in which a critical point resides. In fact, such a cell exhibits mutually opposite vector directions at the corner points, while "direction" refers to the orientation of forward movement, excluding the inverse or backward scenario. Given this definition of direction in combination with bi-linear interpolation, abrupt discontinuities are incurred by wedge or trisector singularities if the domain is still treated as a traditional vector field for tracing streamlines. This problem is demonstrated in Figs. 5a, 5b, 5e, and 5f, which prevents a bunch of curves from running ahead normally and as a consequence, produces wrong patterns into the placement.

To work around this issue, we re-consider the 2D domain from the perspective of a second-order symmetric tensor field. In this context, a streamline is redefined as an integral curve that is everywhere tangent to the major eigenvector field E_2, disregarding the positive (forward) or negative (backward) direction of motion. It is governed by $\nabla \Psi \cdot E_2 = 0$, where Ψ is the streamline function. In other words, the change in Ψ is zero along a streamline. With this re-interpretation, those points deemed as singularities in a traditional vector field are no longer singular in a tensor field, which allows streamlines to avoid premature termination. To integrate a streamline, the advection direction at each step, more precisely at an arbitrary sample point in the 2D domain, is selected between two mutually opposite cases by checking whether the curve is growing roughly along the direction of the local major eigenvector.

$$F(\mathbf{x}, v) = \begin{cases} E_2(\mathbf{x}) & \text{if } E_2(\mathbf{x}) \cdot v \geq 0 \\ -E_2(\mathbf{x}) & \text{otherwise} \end{cases}, \quad (2)$$

where v denotes the line-segment vector pointing from the previous sample of the curve to the current forefront sample, indicating the actual direction in which the curve grows via the previous step. With the advection direction determined at each intermediate sample, the RK-4 integrator below resorts to multiple stages to extend the curve ahead by one step.

$$\begin{aligned} k_1 &= h_n F(\mathbf{x}_n, v_n), \\ k_2 &= h_n F\left(\mathbf{x}_n + \frac{k_1}{2}, v_n\right), \\ k_3 &= h_n F\left(\mathbf{x}_n + \frac{k_2}{2}, v_n\right), \\ k_4 &= h_n F(\mathbf{x}_n + k_3, v_n), \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \left[\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right], \end{aligned} \quad (3)$$
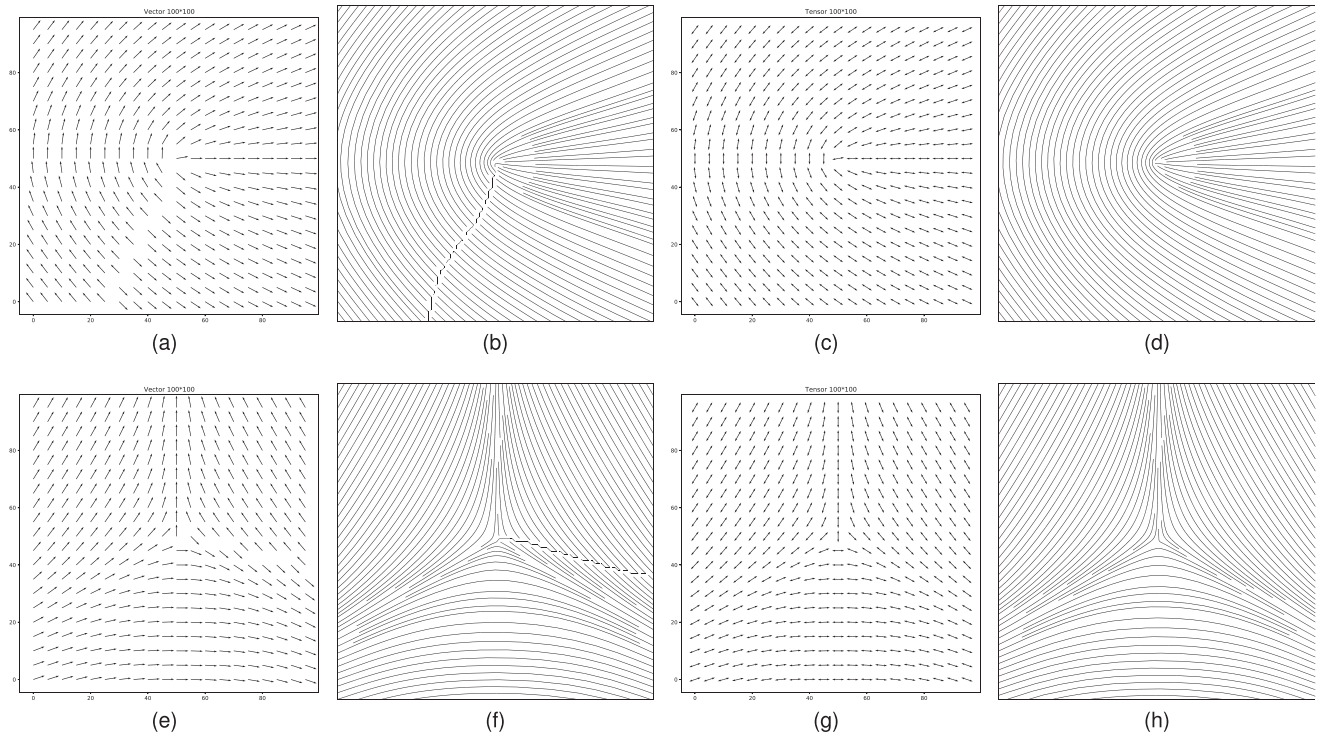
Fig. 5. Comparison of vector fields against second-order symmetric tensor fields in dealing with two singularities, i.e., a wedge (top row) and a trisector (bottom row). (a) A wedge-pattern vector field visualized by uni-directional arrows. (b) Streamlines, with many discontinuities, generated in the wedge vector field. (c) A wedge-pattern tensor field represented by bi-directional arrows. (d) Streamlines, without any discontinuity, integrated in the wedge tensor field. (e) A trisector-pattern vector field shown by uni-directional arrows. (f) Streamlines, with many discontinuities, created in the trisector vector field. (g) A trisector-pattern tensor field illustrated by bi-directional arrows. (h) Streamlines, without any discontinuity, produced in the trisector tensor field.

where $v_n = x_n - x_{n-1}$ refers to the latest line-segment vector of the curve. The initial direction, v_0, is set to $E_2(\mathbf{x}_0)$ for governing the forward advection and to $-E_2(\mathbf{x}_0)$ for guiding the backward advection. The adaptive step size, h_n, is given by

$$h_{n+1} = \begin{cases} h_n/\mu & \text{if } \varepsilon_n > \varepsilon_{max} \\ h_n \cdot \mu & \text{if } \varepsilon_n < \varepsilon_{min} , \\ h_n & \text{otherwise} \end{cases} \qquad (4)$$

where $(\varepsilon_{min}, \varepsilon_{max})$ means a range of error tolerance and $\mu \geq 1$ is a scaling factor by which to enlarge or shrink the step size based on an estimation of the local error ε_n. Through embedded RK formulae, this error is represented in a simple form as follows.

$$\varepsilon_n = \| k_4 - hF(\mathbf{x}_{n+1}, v_n) \|/6. \qquad (5)$$

The customized application of the ADVESS algorithm to a second-order symmetric tensor field allows an evenly-spaced layout of properly integrated streamlines to depict the topological structures that hypnotic line art design may involve. Figs. 5c, 5d, 5g, and 5h shows the ability of this method to cope with wedges or trisectors in a tensor field. The minimum distance between seeds or that between streamlines is specified by a separation parameter s, which is given by the user. The RK-4 integrator with adaptive step size coupled with error control guarantees very fast and highly accurate generation of streamlines, making ADVESS about 5–10 times faster [26] than other geometry-based placement algorithms.

3.5 Line Styles

3.5.1 Interior Lines

Once evenly-spaced streamlines are generated in each region by means of a second-order symmetric tensor field, they can be drawn with diverse line styles. Line styles play an important role in bringing an aesthetic pattern and conveying surface shapes. In particular, we make use of width tapering, i.e., rendering a streamline with the thickness varying from one end to the other, to augment artistic effects. In fact, width tapering is controlled by two parameters, namely base thickness ω and sharpness γ. In more detail, the varying width along a streamline is described by $W(t) = \omega(4(t - t^2))^\gamma$, where $t \in [0, 1]$, ω is a reference thickness, and γ denotes the degree to which the width of a streamline is decreased at the two ends. If $\gamma = 0$, the same width is taken along the entire curve, without any tapering effect at all. If $\gamma = 1$, the width is gradually decreased from the middle point (with the reference thickness) in both directions to reach the ends as the two sharpest tips.

However, streamlines may overlap with one another if the reference thickness ω is excessive (Fig. 6a). To resolve this issue (Fig. 6b), parameter ω needs to be specified relative to the separating distance s that is used in the layout of evenly-spaced streamlines. Specifically, this correlation is governed by $\omega = s\alpha$, where $\alpha \in [0, 1]$ refers to the ratio of ω to s and is hence called the thickness coefficient. Then, the reference thickness can be designated indirectly by the separating distance and the thickness coefficient. This coupling mechanism guarantees that stylized streamlines never overlap as long as the coefficient α falls within the range of $[0, 1]$.
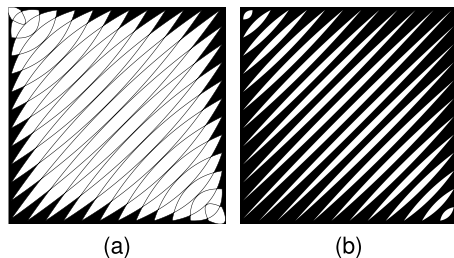
Fig. 6. The varying thickness along streamlines versus the separating distance of the uniform placement. (a) The designation of an absolute value for the reference thickness, if excessive, causes streamlines to overlap with one another. (b) The use of a thickness coefficient correlates the reference thickness with the separating distance in an indirect way to specify the former, able to avoid the overlapping issue.

Fig. 8. Connecting the three U-shaped contours with the vertical boundary (a) is inferior to leaving them disconnected (b) for representing a tiger paw.

A value of 1, i.e., for the extreme case, means that the widest part (i.e., the middle point) of a streamline just occupies the entire separating distance. Fig. 7 shows the same set of anti-orthogonally oriented streamlines drawn with nine different combinations of separating distance s, thickness coefficient α, and sharpness γ.

3.5.2 Contour Lines

Like in those hand-made images of our target style, we place contour lines along region boundaries to portray the 3D shape and emphasize salient features of an object in the 2D image. An inner contour line demarcates different regions, whereas an outer one can enhance the silhouette of the object. Figs. 2c and 2d represent an elephant without and with contour lines, respectively, in hypnotic line art design. In fact, not every contour line has to be drawn. Instead, it depends on the preference of an artist and the specific work. As shown
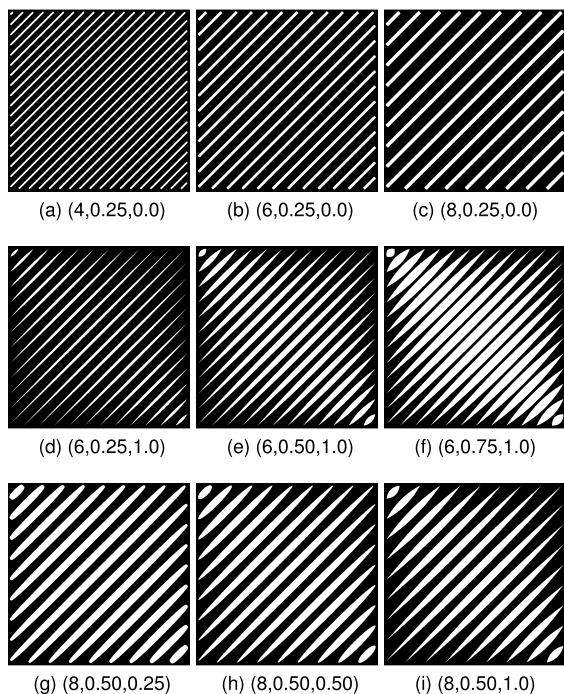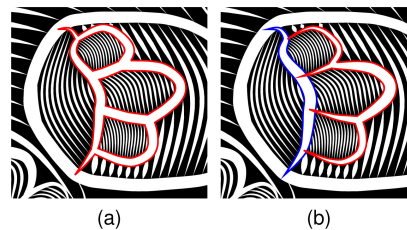
in Fig. 8, the pattern of a tiger paw looks clumsy if the contours there are all connected to form three "loops". An alternative, elegant solution is that the three U-shaped contours are considered irrelevant to and hence not connected with the vertical boundary on the left. We have developed an interactive system by which the user can toggle on or off the boundary of a region. Then, stylized lines are drawn for the contours in the former case but not in the latter. Fig. 9 shows some possible contour topologies that need to be taken into account. The top row illustrates undesirable abrupt branching problems that may occur if a simplistic line width tapering method is separately applied to individual contours. As demonstrated by the middle row, a sophisticated algorithm is needed to address the topology, e.g., loop and branch, when drawing contour lines.

We propose to represent contours using a crossing-free undirected graph $G = (V, E)$, where V is a set of points and E is a set of edges for encoding the connectivity information. The width of a contour line is specified by a deterministic function $W(v) = \omega(H(v))^{\gamma}$, $v \in V$, where ω and γ are the parameters adopted for controlling the line styles (Section 3.5.1). $H(v)$ is a function to be explained below. Suppose each element of set $Q = \{q_1, q_2, \ldots\}$ refers to the index of an endpoint along a contour. Then, we need to take care of three cases when applying line width tapering to the endpoints, namely $|Q| = 0$ (e.g., left column of Fig. 9), $|Q| = 1$, and $|Q| \geq 2$ (e.g., middle and right columns of Fig. 9) in order to avoid the aforementioned branching problem. We begin with a formula for the general solution to case $|Q| \geq 2$. A joint normalized distance from point v_i to all of the endpoints is defined by

(a) (4,0.25,0.0)   (b) (6,0.25,0.0)   (c) (8,0.25,0.0)

(d) (6,0.25,1.0)   (e) (6,0.50,1.0)   (f) (6,0.75,1.0)

(g) (8,0.50,0.25) (h) (8,0.50,0.50) (i) (8,0.50,1.0)

Fig. 7. Three line style parameters (separating distance s, thickness coefficient α, and sharpness γ in order) with nine different combinations (a–i) for drawing the same group of anti-orthogonally oriented streamlines. These combinations illustrate the influence of each parameter and the overall visual effect of the three.

$$H(v_i) = \prod_{j \in Q} \frac{|Q| d_{ij}}{\sum_{j \in Q} d_{ij}}, \forall d_{ij} \neq \infty, \tag{6}$$

where d_{ij} denotes the minimum distance from point v_j to point v_i based on Dijkstra's graph algorithm [41]. As a product falling within [0, 1], $H(v_i)$ considers the relative influence of every minimum distance (i.e., the ratio of each minimum distance to the sum of all such minimum distances). The resulting contour lines are sharpened at the endpoints of the skeleton. Now that Equation (6) works very well for $|Q| \geq 2$, we then need to cope with the other two special cases. Case $|Q| = 0$ implies that the contour consists of only closed loops, without any endpoint, and therefore a constant value is simply assigned: $H(v_i) = 1$. For the case of $|Q| = 1$, we redefine the function as $H(v_i) = d_{ij}/\max_k\{d_{kj}\}$, $j \in Q$. In this way, we can generate topology-aware contour lines to separate regions, show silhouettes, and emphasize surface shapes.
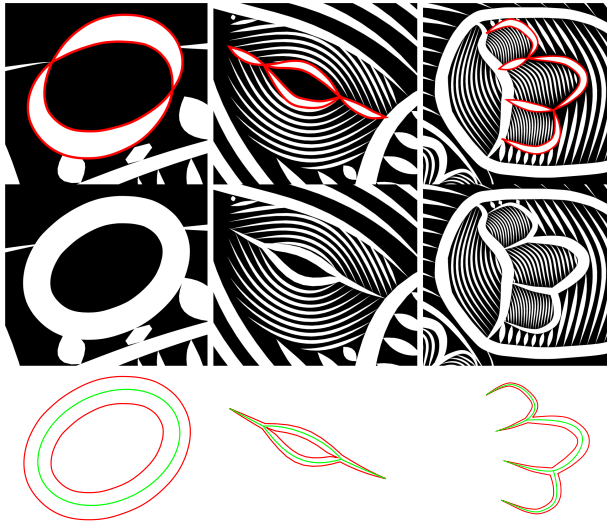
Fig. 9. Contour topology. Tapering the width of each individual contour line, with the topology ignored at all, gives rise to branching artifact (the top row). Width tapering based on the topological structure leads to elegant contour lines (the middle row). Decent contour lines (red, the bottom row) are obtained by tapering the width along the topologically appropriate skeleton (green). The contour topology is characterized by the number of endpoints (0, 2, and 4 for the left, middle, and right columns, respectively).

3.6 User Interaction

Shown in Fig. 10 is the user interface of a system that we provide for amateur artists to conduct interactive design of hypnotic line art. The user can draw and edit a few strokes in each region of a segmented image to outline the work. We approximate each stroke of the user by a cubic Bézier curve to compensate for the smoothness that manual sketches may sacrifice. In addition, a stroke sufficiently near the boundary automatically snaps to the target. We control stroke-boundary snapping by calculating an average ratio $r = s/l$ of the stroke, where s and l are the shortest distance and the longest, respectively, from the points along the stroke being sketched to those along a candidate boundary. If the value is less than a threshold (e.g., 0.02 in our implementation), the stroke is replaced with the associated and actually the closest boundary segment to fulfill the snapping action. Fig. 11 illustrates this snap-to-boundary functionality
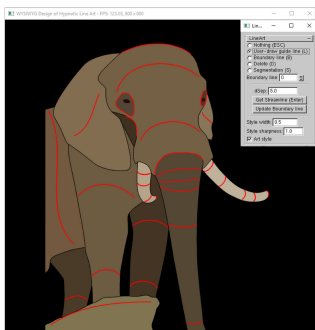

Fig. 10. The user interface of an implementation of our WYSIWYG design of hypnotic line art. The user just needs to draw and edit strokes (red) with mouse clicks to plot a rough sketch for each region of an already segmented 2D image. The system allows for interactive adjustment of a suite of parameters (via a panel at the upper-right part) regarding the placement density and rendering styles of streamlines.
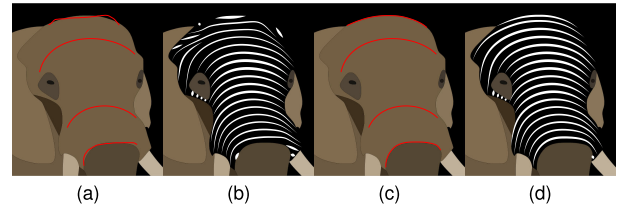

Fig. 11. The snap-to-boundary functionality devised for manual strokes (red) and the influence on placing and rendering streamlines. (a) Two strokes are not snapped to the upper and lower boundaries, respectively. (b) Severe artifacts pop up in the form of short broken line segments (even as elliptic speckles) and jaggy curves. (c) Two strokes are snapped to the upper and lower boundaries, respectively. (d) The region is delineated by a visually appealing placement of evenly-spaced streamlines rendered artistically to mimic the hypnotic line art style.

and the influence on the quality of the output line art. Once manual strokes are collected and processed, a tensor field is constructed for each region and then streamlines are integrated therein to fill the domain. The resulting hypnotic line art complies with the strokes, fleshing out the pattern that the user intends to produce. Offered by the user interface are line style parameters for the user to fine-tune the separating distance, reference thickness, and sharpness of lines (Section 3.5.1). By incorporating interactive input (with strokes) and control (over line styles) with automatic execution (tensor field generation, streamline placement, and contour line decoration), our WYSIWYG system can significantly facilitate the user to create aesthetic hypnotic line art work.

4 RESULTS AND DISCUSSIONS

This section presents a series of results, along with discussions, to demonstrate the effectiveness of our method for hypnotic line art design, involving primitive topological elements treated as building blocks, composite curve patterns that can be represented, real complex objects that can be artistically drawn, the computational time cost, and comparison against other approaches.

4.1 Primitive Topological Elements

Fig. 12 shows a list of primitive topological elements that our method addresses for second-order symmetric tensor fields. They serve as not only building blocks behind the illustrative surface representation of a sophisticated 3D object but also benchmarks for evaluating the sketching capability. Wedge and trisector are two elementary singularities of all tensor fields [40]. In reality, they are well known as core and delta, respectively, in fingerprints [11] to constitute numerous mutually distinct patterns. Wedge and trisector do not exist in vector fields, of which many degenerate points such as saddles, nodes, centers, and foci, though, can be synthesized by combining the two "atomic" elements. Zhang *et al.* [2] provided a comprehensive treatise of these canonical degenerate points of tensor fields with first- and second-order tensor formulae. Unlike previous work [2], our method derives the wedge and trisector elements of a tensor field from schematic sketches of the user.

4.2 Composite Curve Patterns

Fig. 13 shows how primitive topological elements are "woven" together to yield a composite patter of curves for
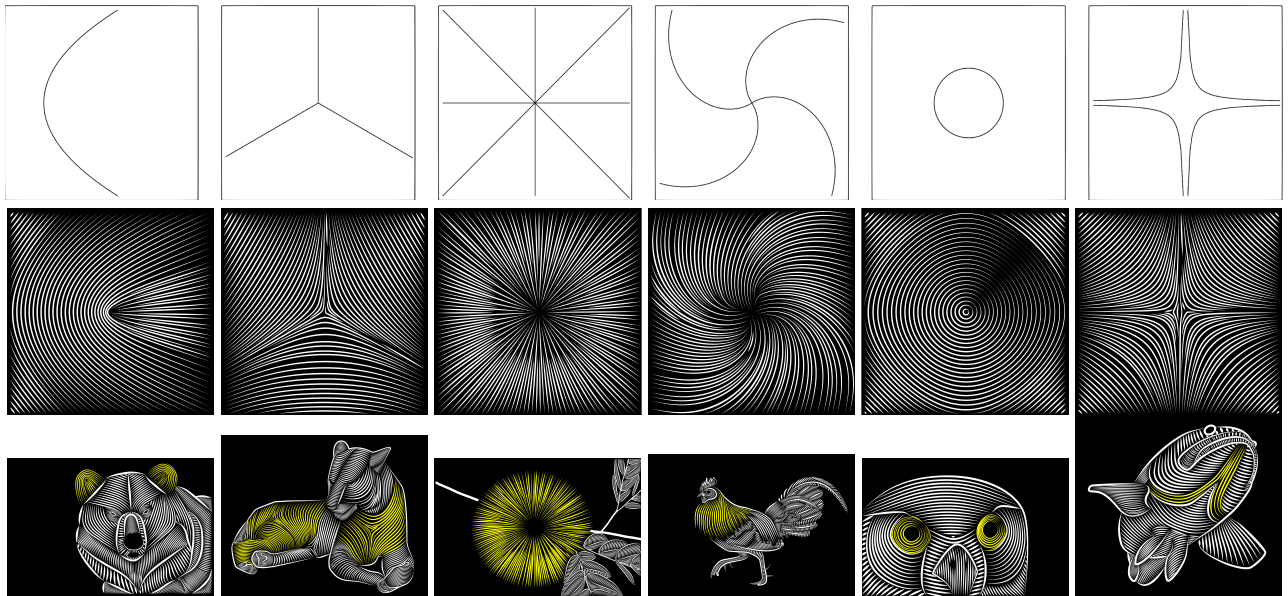
Fig. 12. The primitive topological elements (i.e., wedge, trisector, node, focus, center, and saddle in order from left to right) of tensor fields, underlying the illustrative line art representation. The top row shows the symbolic line structures, one for each topological type, which the user can sketch by freeform strokes. Shown in the middle row are streamlines traced, with an evenly-spaced placement, in each single-element tensor field and rendered with line width tapering effects. The bottom row demonstrates the application of each primitive topological element (clipped against the boundary, marked in yellow) to the hypnotic line art representation of a real complex object.

"extruding" the surface shape of a spinner model. Given the result from 2D image segmentation (the left column), the user may draw two different sets of strokes (the upper row versus the lower row) to try diverse interesting possibilities. Each set is followed by several basic topological structures which are then seamlessly combined to construct a tensor field. The streamlines, uniformly placed and artistically drawn with width tapering, in the rectangular domain (the middle column) are clipped against the 2D boundary of the spinner to attain a hypnotic line art representation (the right column) of the 3D object. The flexibility of sketching strokes based on the user's preference is instrumental to art design by unleashing the imagination.
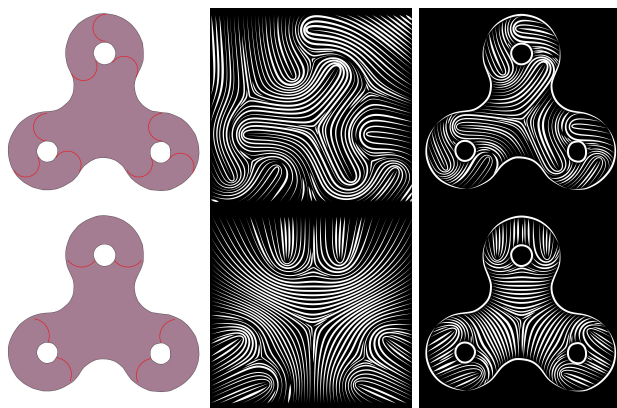

Fig. 13. Composite curve patterns, driven by diverse manual strokes, to describe the surface shape. With two distinct sets of strokes (red) provided for the 2D profile of a spinner model (the left column), primitive topological structures are established and then assembled together to synthesize two tensor fields, which are visualized with evenly-spaced streamline placement and width tapering (the middle column). The intermediate results are clipped against the boundary of the object to export a hypnotic line art representation (the right column).

4.3 Real Complex Objects

Fig. 14 shows the hypnotic line art design of five animals and one plant using our method. Contour lines added along the boundary of each region help accentuate the 3D shape of an object. Given the segmentation result (the middle column) of each reference image (the left column), the user sketches and adjusts strokes to provide one-of-a-kind curves that steer the construction of tensor fields and accordingly the generation as well as placement of streamlines. Specifically, each streamline in the resulting 2D image follows one or multiple strokes manually drawn within a region, representing the local principal curvature direction of the 3D surface being portrayed. By utilizing these strokes as a blueprint, the user does not need to make an arduous effort to deal with the detail. Instead, the automated part of the algorithm processes the interactive yet very coarse input and then fleshes out the pattern with visually compelling effects (the right column). Thus, the entire semi-automatic pipeline enables idea-oriented exploration-driven innovative line art design.

4.4 Computational Time Cost

We have implemented the proposed line art design method in C++ on an ordinary desktop computer equipped with Intel i7 3.4 GHz CPU and 16 GB main memory. We particularly choose scalable vector graphics (SVG) to represent the stylized lines, i.e., the output of the WYSIWYG system, so as to maintain the visual effects that bitmap image file formats inevitably sacrifice due to the aliasing problem of curve rasterization. Based on this platform, timing results are given in Table 1 for illustrative representation and rendering of seven objects. The computational cost of tensor field construction increases with the number of regions in a strictly monotonic manner, so does that of generating, placing, and rendering streamlines. This correlation indeed
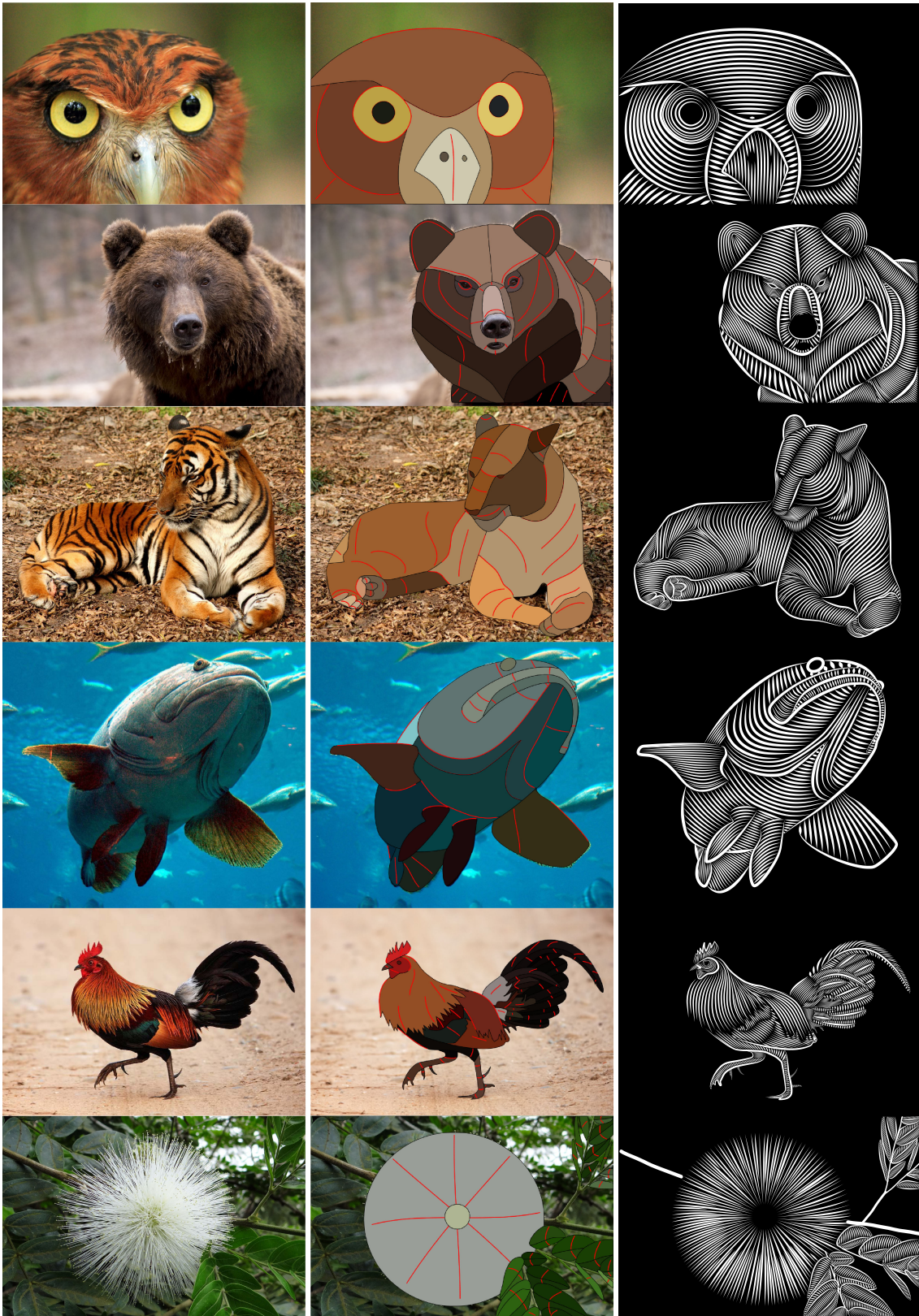
Fig. 14. Hypnotic line art samples generated using our WYSIWYG method. Left column: reference images. Middle column: manual strokes (in red) over the segmented images. Right column: line art results from the system. Shown from top down are an owl, a bear, a tiger, a fish, a cock, and a white powder puff.

makes sense because the number of regions, closely dependent on image segmentation, reflects the complexity of a surface. Needless to say, a complex surface requires many distinct sets of streamlines to capture or approximate intricate features in order to describe the 3D shape. It takes 3.50 seconds on average to process an individual region. This

TABLE 1
The Computational Performance

	Input		Time (in seconds)	
	No. of regions	No. of strokes	Tensor field construction	Streamline placement
Owl (Fig. 14)	13	17	27.11	5.24
Elephant (Fig. 2)	16	33	45.63	8.18
Fish (Fig. 14)	18	35	50.40	9.57
Bear (Fig. 14)	22	44	67.23	12.84
Tiger (Fig. 14)	33	58	100.73	19.31
Plant (Fig. 14)	39	41	120.67	23.23
Cock (Fig. 14)	44	73	132.78	25.85

computational performance hinders the user from immediately obtaining the final result. Fortunately, manual stroke sketching works with real-time response such that the user can be deeply engaged in defining mainstream directions within each region, which is characteristic of interactive design. For each of the seven cases, it takes 10 to 20 minutes on average to complete interactive design, by which the user draws strokes and adjusts parameters (e.g., those regarding line styles). Certainly, the subsequent automated stages (tensor field construction, streamline placement, and artistic rendering of lines) can all be significantly accelerated with a parallel implementation to fully exploit the potential of the proposed WYSIWYG method. The entire semi-automatic pipeline turns out to be efficient for hypnotic line art design.

4.5 Comparison Against Other Methods

Compared in Fig. 15 is our method (b) against the distance transform approach (or DTA for short, a) based on the same image segmentation and the same strokes by the user. Our method resorts to tensor fields (Fig. 5), whereas DTA turns to scalar fields, specifically distance fields (Fig. 16). A distance field is defined by the offset of each sample point in the domain from one or multiple strokes. Then, a bunch of contour loops may be successively generated around the stroke(s), like concentric ripples in a lake. The critical drawback to DTA consists in the rigid parallelism between curves, as opposed to evenly-spaced streamlines which are roughly parallel in our method but "jittered" by the convergent or divergent structures of a tensor field. This rigidness gives rise to sharp curvatures to make jaggy curves, unable to accurately capture the surface shape in an adaptive manner. Thus, artifacts and distractions are considerable in the DTA output, especially near region boundaries. Standing out among the visual disturbances of DTA are cavities, which
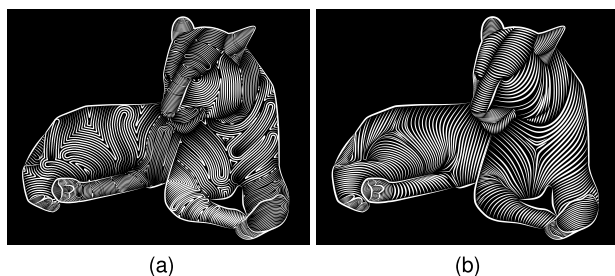

Fig. 15. Comparison between (a) distance transform approach, and (b) our method of hypnotic line art design, in depicting the 3D surface shape of a tiger based on the same 2D reference image, image segmentation, and manual strokes (row 3 of Fig. 12).
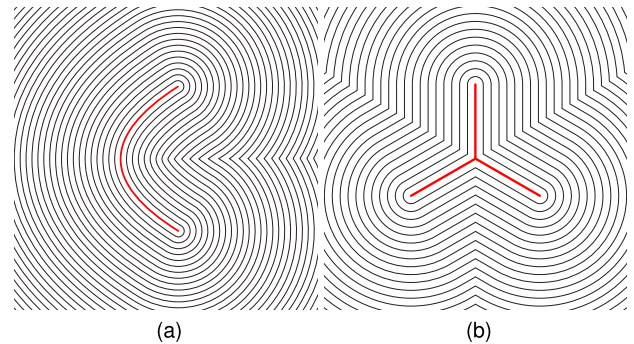

Fig. 16. Two distance fields based on one (a) and three (b) strokes (in red), respectively, each accompanied by exactly equidistant and hence absolutely parallel contour loops for a distance transform.

are though perceptually "filtered" away in our method by the harmony of evenly-spaced but smoothly deviating streamlines. The working mechanism of DTA indicates that distance transform is not good at hypnotic line art design. Our method is essentially advantageous over DTA in that the tensor field built for each spatial region encodes the surface shape (as defined by manual strokes) while every streamline traced within the tensor field exactly delineates this information. As demonstrated by our method in Fig. 15b, accurate shape representation and exquisite line art are supportive to each other. This relationship is reiterated by DTA in Fig. 15a from the opposite perspective.

5 CONCLUSION AND FUTURE WORK

We have presented a novel method for WYSIWYG hypnotic line art design, by which evenly-spaced streamlines are drawn with width tapering to depict the surface shape of a 3D object and accentuate levels of depth across the curved patches. Given a topological segmentation of the surface and a preferred pattern that the user illustrates with strokes over a 2D image, a second-order symmetric tensor field is constructed for each region. By minimizing a Laplacian energy function, a smooth geometrically simple structure tensor field can be obtained to comply with the strokes, avoid high curvatures, and reduce degenerate points. To the best of our knowledge, our work is the first to apply this Laplacian method to the design of smooth tensor fields. Tracing a streamline along the local major eigenvector direction in a tensor field makes it possible to handle wedge and trisector, two topological structures which both prematurely terminate integral curves in a vector field. Controlled by the density and modulated by two line style parameters, i.e., thickness and sharpness, evenly-spaced streamlines constitute a visually pleasing pattern to convey the shape, orientation, and depth of each curved patch, while contour lines added along region boundaries weave all patches into a complete surface. Testing with a rich variety of images (involving animals and plants) and comparison against other methods show that the proposed approach facilitates hobbyists to practice fancy line art design in a WYSIWYG way.

One limitation of our system is that the density of evenly-spaced streamlines is uniform in each region. Given a few strokes sketched to mean a ring-shaped surface (Fig. 17a), one typical goal is to generate a perspective view with streamlines increasingly dense from the center toward the peripheral (Fig. 17b). Unfortunately, this perspective effect may not be
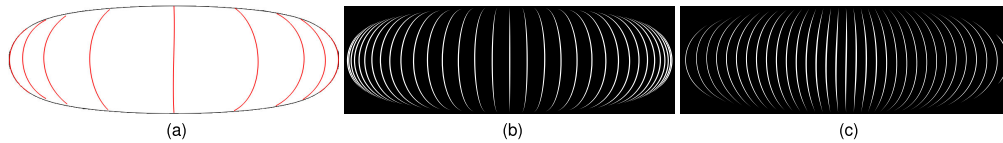
Fig. 17. A limitation of our method. (a) Several strokes (in red) added in a ring-shaped region (e.g., extracted from 2D image segmentation). (b) A perspective view with the depth cueing accentuated by a set of streamlines of which the density progressively increases from the center toward the peripheral. (c) Evenly-spaced streamlines currently adopted in our method may not produce the expected perspective effect.
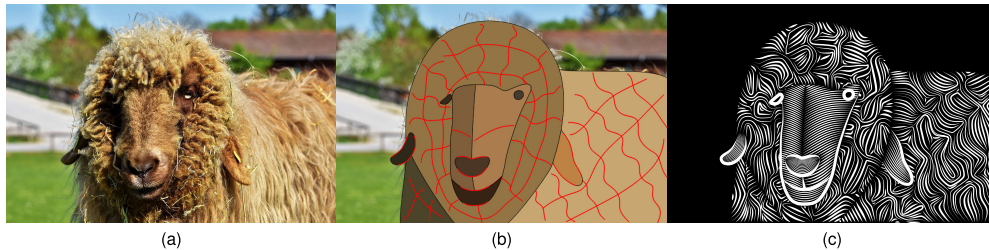

Fig. 18. Basic art design skills required for image segmentation and manual strokes. (a) A reference image. (b) Poor image segmentation and shabby strokes. (c) The output from our system, where the curly hair and wool are not well represented and hence affect the perception of the entire surface shape.

provided by our method, of which the result (Fig. 17c) exhibits depth cues weaker than achieved by non-evenly spaced streamlines. At least for such scenarios, a mixture of multiple progressively increasing densities can strengthen the power of streamline placement in hypnotic line art design. Thus, we will enhance the system with this capability in future work. In addition, we would like to investigate how to generate hypnotic line art work on 3D models. Although our system is able to duplicate Seymour's drawing style very well, image segmentation and illustrative strokes are not trivial tasks for novice users. Instead, they require basic art design skills. Fig. 18 shows a result of arbitrarily drawing high-curvature strokes, from which our system may not effectively delineate the curly hair and wool of the sheep. We plan to analyze the art-making process, of which the rules will help users accomplish original and creative results.

ACKNOWLEDGMENTS

REFERENCES

[1] P. Seymour. [Online]. Available: https://www.behance.net/patrickseymour
[2] E. Zhang, J. Hays, and G. Turk, "Interactive tensor field design and visualization on surfaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 1, pp. 94–107, Jan./Feb. 2007. [Online]. Available: https://doi.org/10.1109/TVCG.2007.16
[3] J. Palacios and E. Zhang, "Rotational symmetry field design on surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 55–es, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276446
[4] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramee, "Asymmetric tensor analysis for flow visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 1, pp. 106–122, Jan./Feb. 2009. [Online]. Available: https://doi.org/10.1109/TVCG.2008.68
[5] M. P. Salisbury, M. T. Wong, J. F. Hughes, and D. H. Salesin, "Orientable textures for image-based pen-and-ink illustration," in *Proc. 24th Annu. Conf. Comput. Graph. Interactive Techn.*, 1997, pp. 401–406. [Online]. Available: https://doi.org/10.1145/258734.258890

[6] J. Zander, T. Isenberg, S. Schlechtweg, and T. Strothotte, "High quality hatching," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 421–430, 2004. [Online]. Available: https://doi.org/10.1111/j.1467-8659.2004.00773.x
[7] M. Gerl and T. Isenberg, "Interactive example-based hatching," *Comput. Graph.*, vol. 37, no. 1, pp. 65–80, 2013. [Online]. Available: https://doi.org/10.1016/j.cag.2012.11.003
[8] W. Leister, "Computer generated copper plates," *Comput. Graph. Forum*, vol. 13, no. 1, pp. 69–77, 1994. [Online]. Available: https://doi.org/10.1111/1467-8659.1310069
[9] S. F. Johnston, "Lumo: Illumination for cel animation," in *Proc. 2nd Int. Symp. Non-Photorealistic Animation Rendering*, 2002, pp. 45–ff. [Online]. Available: https://doi.org/10.1145/508530.508538
[10] V. Ostromoukhov, "Digital facial engraving," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 417–424. [Online]. Available: https://doi.org/10.1145/311535.311604
[11] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic fingerprint-image generation," in *Proc. 15th Int. Conf. Pattern Recognit.*, 2000, vol. 3, pp. 471–474. [Online]. Available: https://doi.org/10.1109/ICPR.2000.903586
[12] A. Hertzmann and D. Zorin, "Illustrating smooth surfaces," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Techn.*, 2000, pp. 517–526. [Online]. Available: https://doi.org/10.1145/344779.345074
[13] C. Rössl and L. Kobbelt, "Line-art rendering of 3D-models," in *Proc. 8th Pacific Conf. Comput. Graph. Appl.*, 2000, pp. 87–96. [Online]. Available: https://doi.org/10.1109/PCCGA.2000.883890
[14] G. Elber and E. Cohen, "Adaptive isocurve-based rendering for free-form surfaces," *ACM Trans. Graph.*, vol. 15, no. 3, pp. 249–263, Jul. 1996. [Online]. Available: https://doi.org/10.1145/231731.231736
[15] G. Elber, "Rendering with parallel stripes," *IEEE Comput. Graphics Appl.*, vol. 21, no. 3, pp. 44–52, May/Jun. 2001. [Online]. Available: https://doi.org/10.1109/38.920626
[16] R. D. Kalnins *et al.*, "WYSIWYG NPR: Drawing strokes directly on 3D models," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 755–762, Jul. 2002. [Online]. Available: https://doi.org/10.1145/566654.566648
[17] Y. Kim, J. Yu, X. Yu, and S. Lee, "Line-art illustration of dynamic and specular surfaces," *ACM Trans. Graph.*, vol. 27, no. 5, Dec. 2008, Art. no. 156. [Online]. Available: https://doi.org/10.1145/1409060.1409109
[18] R. A. Saputra, C. S. Kaplan, P. Asente, and R. Měch, "FLOWPAK: Flow-based ornamental element packing," in *Proc. 43rd Graph. Interface Conf.*, 2017, pp. 8–15. [Online]. Available: https://doi.org/10.20380/GI2017.02
[19] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr, "Digital micrography," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011, Art. no. 100. [Online]. Available: https://doi.org/10.1145/2010324.1964995
[20] J. Xu and C. S. Kaplan, "Image-guided maze construction," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 29–es, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276414

[21] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive procedural street modeling," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, Aug. 2008. [Online]. Available: https://doi.org/10.1145/1360612.1360702

[22] C. Y. Yao, M. T. Chi, T. Y. Lee, and T. Ju, "Region-based line field design using harmonic functions," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 6, pp. 902–913, Jun. 2012. [Online]. Available: https://doi.org/10.1109/TVCG.2011.112

[23] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, "Over two decades of integration-based, geometric flow visualization," *Comput. Graph. Forum*, vol. 29, no. 6, pp. 1807–1829, 2010. [Online]. Available: https://doi.org/10.1111/j.1467–8659.2010.01650.x

[24] G. Turk and D. Banks, "Image-guided streamline placement," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 453–460. [Online]. Available: https://doi.org/10.1145/237170.237285

[25] B. Jobard and W. Lefer, *Creating Evenly-Spaced Streamlines of Arbitrary Density*. Vienna, Austria: Springer, 1997, pp. 43–55.

[26] Z. Liu, R. Moorhead, and J. Groner, "An advanced evenly-spaced streamline placement algorithm," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 965–972, Sep./Oct. 2006. [Online]. Available: https://doi.org/10.1109/TVCG.2006.116

[27] B. Jobard and W. Lefer, "Unsteady flow visualization by animating evenly-spaced streamlines," *Comput. Graph. Forum*, vol. 19, no. 3, pp. 31–39, 2000. [Online]. Available: https://doi.org/10.1111/1467-8659.00395

[28] O. Mattausch, T. Theußl, H. Hauser, and E. Gröller, "Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines," in *Proc. 19th Spring Conf. Comput. Graph.*, 2003, pp. 213–222.

[29] L. Li and H. Shen, "Image-based streamline generation and rendering," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 3, pp. 630–640, May/Jun. 2007. [Online]. Available: https://doi.org/10.1109/TVCG.2007.8093671

[30] B. Jobard and W. Lefer, "Multiresolution flow visualization," in *Proc. Int. Conf. Central Europe Comput. Graph. Vis. Comput. Vis.*, 2001, pp. 34–37.

[31] Z. Liu and R. J. M. II, "Interactive view-driven evenly spaced streamline placement," in *Proc. Vis. Data Anal.*, 2008, pp. 95–106. [Online]. Available: https://doi.org/10.1117/12.765736

[32] B. Spencer, R. S. Laramee, G. Chen, and E. Zhang, "Evenly spaced streamlines for surfaces: An image-based approach," *Comput. Graph. Forum*, vol. 28, no. 6, pp. 1618–1631, 2009. [Online]. Available: https://doi.org/10.1111/j.1467-8659.2009.01352.x

[33] A. Mebarki, P. Alliez, and O. Devillers, "Farthest point seeding for efficient placement of streamlines," in *Proc. IEEE Vis.*, 2005, pp. 479–486. [Online]. Available: https://doi.org/10.1109/VISUAL.2005.1532832

[34] O. Rosanwo, C. Petz, S. Prohaska, H. C. Hege, and I. Hotz, "Dual streamline seeding," in *Proc. IEEE Pacific Vis. Symp.*, 2009, pp. 9–16. [Online]. Available: https://doi.org/10.1109/PACIFICVIS.2009.4906832

[35] K. Wu, Z. Liu, S. Zhang, and R. J. M. II, "Topology-aware evenly spaced streamline placement," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 5, pp. 791–801, Sep./Oct. 2010. [Online]. Available: https://doi.org/10.1109/TVCG.2009.206

[36] V. Verma, D. Kao, and A. Pang, "A flow-guided streamline seeding strategy," in *Proc. Vis.*, 2000, pp. 163–170. [Online]. Available: https://doi.org/10.1109/VISUAL.2000.885690

[37] X. Ye, D. Kao, and A. Pang, "Strategy for seeding 3D streamlines," in *Proc. IEEE Vis.*, 2005, pp. 471–478. [Online]. Available: https://doi.org/10.1109/VISUAL.2005.1532831

[38] J. Etienne and S. Lefebvre, "Procedural band patterns," in *Proc. Symp. Interactive 3D Graph. Games*, 2020, Art. no. 1. [Online]. Available: https://doi.org/10.1145/3384382.3384522

[39] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004. [Online]. Available: https://doi.org/10.1109/TPAMI.2004.60

[40] T. Delmarcelle and L. Hesselink, "The topology of symmetric, second-order tensor fields," in *Proc. Vis.*, 1994, pp. 140–147. [Online]. Available: https://doi.org/10.1109/VISUAL.1994.346326

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

Chih-Kuo Yeh received the PhD degree in computer science and information engineering from National Cheng-Kung University, Taiwan, in 2015. He was a postdoctoral researcher in National Cheng-Kung University, Taiwan, during 2016-2019. He is currently an associate professor in the School of Computer Science and Software, Zhaoqing University, China. His research interests include computer graphics, computer vision, scientific visualization, and machine learning.
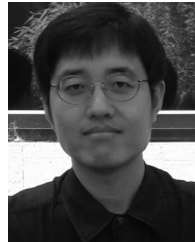
Zhanping Liu received the BS degree in mathematics from Nankai University, China, in 1992, and the PhD degree in computer science from Peking University, China, in 2000. He is an assistant professor with the Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University, Norfolk, Virginia. He was a 2014-summer USRA-sponsored faculty visitor at NASA Ames Research Center and a 2013-summer FRPP-sponsored faculty visitor at Argonne National Lab during his employment as an assistant professor with the Department of Computer Science, Kentucky State University, Frankfort, Kentucky. Much earlier, he was a research staff member at the University of Pennsylvania, Philadelphia, Pennsylvania, Kitware, Inc., Clifton Park, New York, and Mississippi State University, Starkville, Mississippi after a postdoctoral position at the University of Iowa, Iowa City, Iowa. His research interests remain in scientific visualization, particularly vector (flow) visualization and parallel visualization.
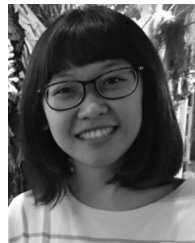
I-Hsuan Lin received the BS degree from the Department of Computer Science and Information Engineering, Cheng-Kung University, Taiwan, in 2015, and the MS degree from the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, in 2017. Her research interest is computer graphics.

Eugene Zhang (Senior Member, IEEE) received the PhD degree in computer science from the Georgia Institute of Technology, Atlanta, Georgia, in 2004. He is currently a professor at Oregon State University, Corvallis, Oregon, where he is a member in the School of Electrical Engineering and Computer Science. During part of 2011 and 2012, he was a guest professor with the Free University of Berlin, Germany and the Max-Planck-Institute in informatics. His research interests include computer graphics, scientific visualization, geometric modeling, and computational topology. He received the National Science Foundation CAREER Award, in 2006. He has also served as a program co-chair for CAD/CG 2013. Currently, he is an association editor of the *ACM Transactions on Graphics*, the *Computer Graphics Forum*, and the *Journal of Graphical Models*. He is a senior member of the ACM.

Tong-Yee Lee (Senior Member, IEEE) received the PhD degree in computer engineering from Washington State University, Pullman, Washington, in May 1995. He is currently a chair professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University, Tainan, Taiwan, ROC (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He is a senior member of the IEEE Computer Society and a member of the ACM.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.