WYSIWYG Design of Hypnotic Line Art

Chih-Kuo Yeh\textsuperscript{a}, Zhanping Liu\textsuperscript{a}, I-Hsuan Lin, Eugene Zhang\textsuperscript{a}, Senior Member, IEEE, and Tong-Yee Lee\textsuperscript{a}, 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 \hspace*{1em} \textbf{INTRODUCTION}

\textbf{CA}nadian illustrator Patrick Seymour \cite{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 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 \cite{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 \cite{2}, \cite{3}, \cite{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
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 streamline algorithms 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 composing 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. Elber and Cohen [14] devised an adaptive spacing method for rendering a freeform surface with a set of iso-curves. 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...
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, $x = (x, y) \in \mathbb{R}^2$, is represented by

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

A constraint imposed in our work is that the tensor needs to be symmetric, i.e., $T_{12}(x) = T_{21}(x)$. 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. 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.

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.
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 $F_2$ (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., $100 \times 100$ in our implementation) Cartesian grid of sample points. $T(x)$ is represented by $T(i \Delta x, j \Delta y)$, where $i, j \in \mathbb{Z}$ are the indices and $\Delta x$ and $\Delta y$ 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 $T(x) = u(x)u(x)^T$, where $u(x) = \frac{\partial F}{\partial x}$ is the tangent vector derived from point $x$ of the stroke. The region boundaries, for $i = 0$ or $M$ and $j = 0$ or $N$, are subject to the Neumann boundary conditions: $\nabla T(x) \cdot n = 0$, where $n$ 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 reconsider 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 $F_2$, disregarding the positive (forward) or negative (backward) direction of motion. It is governed by $\Psi \cdot E_2 = 0$, where $\Psi$ is the streamline function. In other words, the change in $\Psi$ 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.

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

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.

$$
\begin{align*}
k_1 &= h \cdot F(x_n, v_n), \\
k_2 &= h \cdot F\left(x_n + \frac{k_1}{2}, v_n\right), \\
k_3 &= h \cdot F\left(x_n + \frac{k_2}{2}, v_n\right), \\
k_4 &= h \cdot F(x_n + k_3, v_n), \\
x_{n+1} &= x_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4),
\end{align*}
$$
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 $E_2(x_0)$ for governing the forward advection and to $-E_2(x_0)$ for guiding the backward advection. The adaptive step size, $h_n$, is given by

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

where $(\varepsilon_{\text{min}}, \varepsilon_{\text{max}})$ 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 $\varepsilon_n$. Through embedded RK formulae, this error is represented in a simple form as follows.

$$\varepsilon_n = \| k_1 - hF(x_{n+1}, v_n) \| / 6.$$  

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.

### 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 $v$ and sharpness $g$. In more detail, the varying width along a streamline is described by $W(t) = \omega/(4(t - t^2))^\gamma$, where $t \in [0, 1]$, $\omega$ is a reference thickness, and $\gamma$ 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 $\omega$ is excessive (Fig. 6a). To resolve this issue (Fig. 6b), parameter $\omega$ 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 $\omega = sa$, where $s \in [0, 1]$ refers to the ratio of $\omega$ 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 $a$ falls within the range of $[0, 1]$.  

---
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. 7, 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 $\omega$ and $\gamma$ 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

$$H(v_i) = \prod_{q \in Q} \sqrt[|Q|]{\sum_{j \in Q} \frac{d_{ij}}{d_{ij}}} \neq \infty, \quad (6)$$

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 $H(v_i) = d_{ij}/\max\{d_{ij}\}$, $j \in Q$. In this way, we can generate topology-aware contour lines to separate regions, show silhouettes, and emphasize surface shapes.
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 pattern of curves for
“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.

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
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

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.
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 rigidity 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 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

---

**TABLE 1**

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of regions</th>
<th>No. of strokes</th>
<th>Time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owl (Fig. 14)</td>
<td>13</td>
<td>17</td>
<td>27.11</td>
</tr>
<tr>
<td>Elephant (Fig. 2)</td>
<td>16</td>
<td>33</td>
<td>45.63</td>
</tr>
<tr>
<td>Fish (Fig. 14)</td>
<td>18</td>
<td>35</td>
<td>50.40</td>
</tr>
<tr>
<td>Bear (Fig. 14)</td>
<td>22</td>
<td>44</td>
<td>67.23</td>
</tr>
<tr>
<td>Tiger (Fig. 14)</td>
<td>33</td>
<td>58</td>
<td>100.73</td>
</tr>
<tr>
<td>Plant (Fig. 14)</td>
<td>39</td>
<td>41</td>
<td>120.67</td>
</tr>
<tr>
<td>Cock (Fig. 14)</td>
<td>44</td>
<td>73</td>
<td>132.78</td>
</tr>
</tbody>
</table>

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.
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

The authors would like to thank the reviewers for many constructive comments that help improve the article. This work was supported in part by the Ministry of Science and Technology (contracts 107-2221-E-006-196-MY3 and 108-2221-E-006-038-MY3), Taiwan, and NSF award # 1619383.

REFERENCES


