Region-Based Line Field Design Using Harmonic Functions

Chih-Yuan Yao, Ming-Te Chi, *Member*, *IEEE*, Tong-Yee Lee, *Senior Member*, *IEEE*, and Tao Ju

Abstract—Field design has wide applications in graphics and visualization. One of the main challenges in field design has been how to provide users with both intuitive control over the directions in the field on one hand and robust management of its topology on the other hand. In this paper, we present a design paradigm for line fields that addresses this challenge. Rather than asking users to input all singularities as in most methods that offer topology control, we let the user provide a partitioning of the domain and specify simple flow patterns within the partitions. Represented by a selected set of harmonic functions, the elementary fields within the partitions are then combined to form continuous fields with rich appearances and well-determined topology. Our method allows a user to conveniently design the flow patterns while having precise and robust control over the topological structure. Based on the method, we developed an interactive tool for designing line fields from images, and demonstrated the utility of the fields in image stylization.

Index Terms—Field design, line field, singularity, harmonic functions.

1 Introduction

Various graphics and visualization applications utilize a direction field that describes, at each point on an image or surface, a "direction" along which geometric or rendering elements can be placed. In this paper, we are concerned with the design of *line fields*, where each point is associated with the direction of an unoriented line. Such fields find applications in painterly rendering [11], [10], where the placement of brush strokes at each image point is guided by the line directions, in procedure modeling [2], as well as in surface parameterization [17], [23], [12].

An important task in field design is controlling the *topology* of the field, and particularly its *singularities*. Singularities are points in the field where the directions are undefined. The distortion of the field is typically most obvious around these points, and hence avoiding unnecessary singularities is critical for producing a smooth-looking field. On the other hand, a carefully chosen set of singularities is necessary to create interesting, nontrivial flow patterns. Consider the line field example in Fig. 1c, which mimics the flow in Van Gogh's painting Starry Night in Fig. 1a. The field contains more than 30 singularities (green dots in Fig. 1c) located at the centers of vortex-shaped stars and junctions of various streams in the night sky. These singularities are essential in creating the desired flow patterns. To enable the design of a complex field like this while avoiding unwanted singularities, a field design tool should have the following properties:

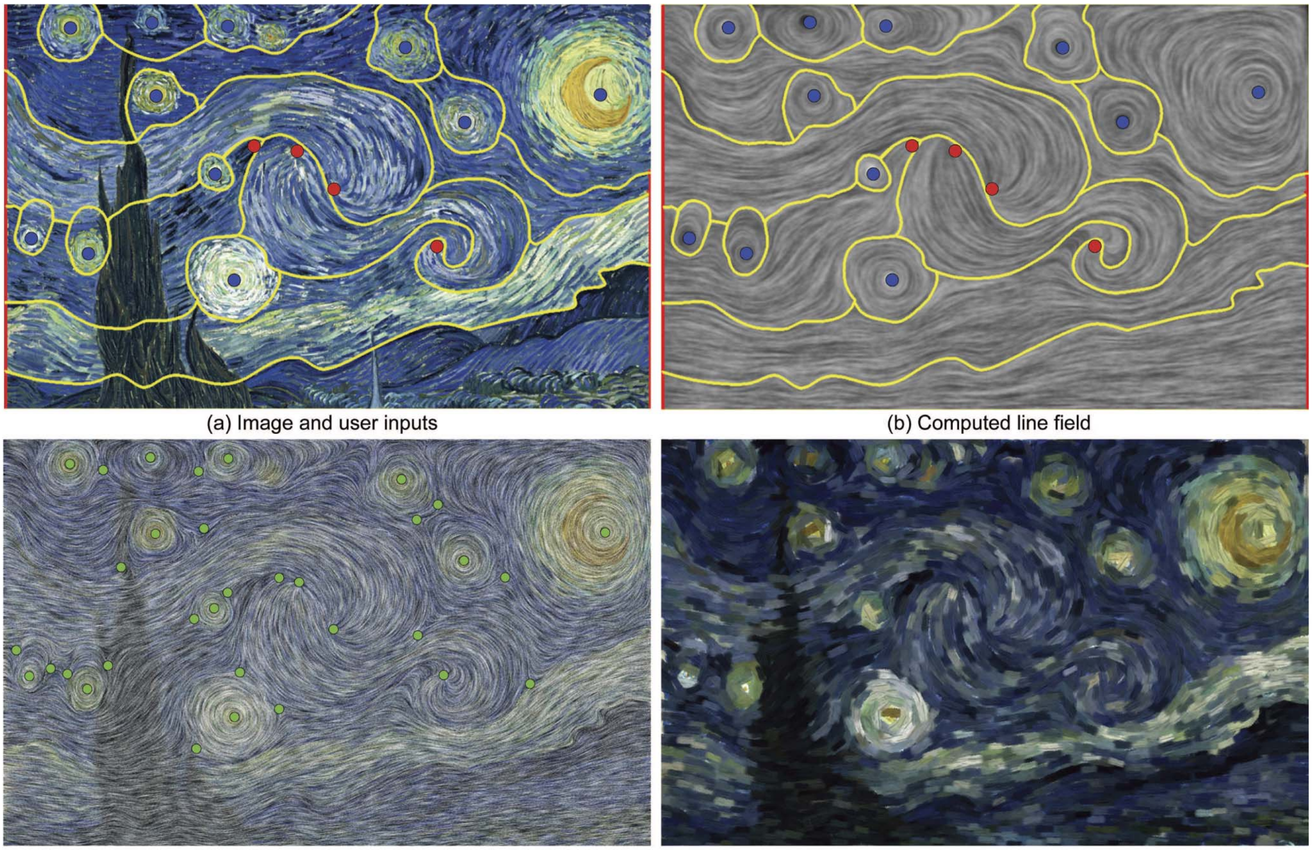
- **Intuitive**. The tool should be easy enough for a novice user to create a field with a possibly nontrivial appearance.
- **Precise**. The user should have exact control over the topological structure of the field, which is described by the singularities, their local topology, and their mutual connectivity.
- **Robust**. The tool always produces a field that meets the user-prescribed topology without producing unwanted singularities.

A number of methods have been proposed to offer topology control in designing various kinds of direction fields (reviewed in Section 2). In particular, several recent methods offer robust topology guarantees [19], [18], [5], [14]. These methods adopt a *singularity-based* design paradigm: the user specifies the locations of singularities, their indices (i.e., the amount of total rotation of the field around the singularity), and/or their quantity. A field is then generated with precisely the given set of singularities.

However, specifying singularities is not the most direct or intuitive way of designing a direction field, as it asks for where the direction "vanishes" rather than what the directions are. In addition, it can be challenging for a novice user to understand and supply the indices of singularities, not to mention that these indices need to satisfy some global constraint (such as the Poincare-Hopf theorem for vector fields). Even for experienced users, correctly identifying all singularities and their indices in a field that has a nontrivial topology can be a daunting task. The second limitation of these methods is that the locations and indices of singularities do not offer precise control over the topological structure of the field. Singularities of dramatically different appearances and local topology can have the same index (see Fig. 5). In addition, the

- C.-Y. Yao and T.-Y. Lee are with the Computer Graphics Group/Visual System Lab, Department of Computer Science and Information Engineering, National Cheng-Kung University, No. 1, University Rd., Tainan 701, Taiwan. E-mail: cyuan.yao@gmail.com, tonylee@mail.ncku.edu.tw.
- M.-T. Chi is with the Department of Computer Science, National Chengchi University, Da-Zen Building, NO. 64, Sec. 2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan, R.O.C. E-mail: mtchi@cs.nccu.edu.tw.
- T. Ju is with Department of Computer Science and Engineering, Washington University, Lopata Hall 517, One Brookings Drive, St. Louis, MO 63130. E-mail: taoju@cse.wustl.edu.

Manuscript received 9 Aug. 2010; revised 10 Apr. 2011; accepted 12 Apr. 2011; published online 16 June 2011.
Recommended for acceptance by J. Stam.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-08-0178.
Digital Object Identifier no. 10.1109/TVCG.2011.112.



(a) Image and user inputs (b) Computed line field (c) Singularities in the field (d) Painterly rendered image

Fig. 1. Designing a line field that mimics the flows in Starry Night: (a) The user provides a partitioning (yellow curves) into regions, and places in each region centers (blue dots), terminals (red dots) and flow-out segments (red lines) to indicate desired flow patterns. (b) Our algorithm computes a continuous line field composed of smooth elementary fields within the regions. (c) The field has singularities (green dots) located exactly at user-placed centers, terminals, and where more than two regions meet. (d) A painterly rendered result guided by the field.

connectivity of singularities by their streamlines (or separatrices) cannot be controlled in these methods, but which can also have a strong impact on the overall appearance of the field.

In this paper, we present a novel design paradigm for line fields in a 2D domain, which gives users intuitive design primitives as well as precise and robust topology control. Unlike existing approaches, here the user designs a complete field by piecing together building blocks called *elementary fields*. Each elementary field resides in a simply connected region, and has a simplistic appearance such as circulating around a center or flowing between two terminals on the region boundary. The "design" process involves the user segmenting the 2D domain into regions, and specifying the type of elementary field within each region by either designating the center of circulation or the terminals of flow. An example of the design result is shown in Fig. 1a. Given the user input, our algorithm computes the elementary fields with the desired flow patterns, as shown in Fig. 1b. Note that the elementary fields are tangential to their region boundaries, so that together they make up a continuous line field over the entire domain. More importantly, the composite field has a well-determined topological structure: the singularities (green dots in Fig. 1c) lie exactly at the user-placed centers, terminals, and junctions of more than two regions, the local topology of a singularity is governed by the incident elementary fields, and the region boundaries form the separatrices connecting the singularities.

In contrast to previous approaches, the user is less concerned with where the field "vanishes" in our method, and instead focuses on how the field is decomposed into simple flow patterns: the topology control emerges as a result of this design. In a sense, the *region-based* design paradigm offers a "middle layer" between the user and the underlying field topology, allowing the user to design the appearance of the field while at the same time creating a complete topological skeleton that can be fully controlled.

The key to our robust topology guarantee is our representation of the elementary fields, which associates a point in each region with the tangent direction of the level curve of some *harmonic function* defined in that region. Singularities in such a field correspond to *critical points* of the harmonic function (where the partial derivatives vanish), which are well studied in the literature in the 2D domain [26]. We consider a particular set of harmonic functions whose corresponding line fields exhibit typical flow patterns, meet tangentially with the region boundaries, and have few and controllable singularities. The representation also gives rise to a simple discrete algorithm for computing a piecewise constant line field as level curves of some *discrete harmonic functions*. We show that the discrete functions computed by our algorithm share the same topological and geometric properties as their continuous counterpart.

1.1 Contributions

In light of previous works on field design with topology control, we see our work make the following contributions:

- We propose a region-based design paradigm for line fields that gives users intuitive design primitives (elementary fields) as well as precise and robust topology control.
- We represent elementary fields by a carefully chosen group of harmonic functions, and show that their composition exhibits rich flow behaviors with well-determined topology.
- We develop an algorithm for computing discrete line fields using discrete harmonic functions, and show that these functions have similar topology and geometry as their continuous counterparts.
- We develop an interactive tool for designing 2D line fields with applications in painterly rendering over images (e.g., Fig. 1d).

This paper focuses on line fields in a 2D domain, where behaviors of harmonic functions are well understood. Unfortunately, theories regarding critical points of harmonic functions are still lacking for curved surfaces in 3D. Nonetheless, we will show that the same design paradigm and discrete computation in 2D can be easily extended onto surfaces in 3D with visually pleasing results.

The rest of the paper is organized as follows: after reviewing previous works in Section 2, we briefly review line field and its topology in Section 3. Section 4 presents the core of our algorithm, the representation of a line field using carefully designed elementary fields. Section 5 presents our discrete algorithm for computing the field over a triangulated domain. Sections 6 discusses an interactive field design interface, and results are presented in Section 7. We conclude in Section 8 where we further discuss the extension of our method onto 3D surfaces.

2 Related Work

There is a sizable literature on interactive means for generating various types of direction fields, such as vector fields, line fields, and direction fields with higher order rotational symmetry. Here, we give a brief review over topology control strategies in these design methods, while referring interested readers for more in-depth reviews in recent articles such as [3] and [18].

As an indirect approach for topology control, *field simplification* removes excessive singularities as a postprocess after the field is created. Typical simplification strategies include geometric smoothing [24], and topological surgery operators like pair annihilation [25] or pair cancellation [6], [31]. While these methods have been useful for cleaning up complex fields with many singularities, it is difficult to rely on simplification to achieve a prescribed topology with a specific set of singularities. Moreover, the simplification process may significantly alter the flow directions in the original field.

In the field of fingerprint recognition, the *zero-pole* method offers a mathematical model for defining line fields with controlled singularities [20], [7]. The model generates an ideal field given locations of "cores" (i.e., wedges) and "deltas" (i.e., tri-sectors). Multiplicity of cores or deltas at a same location would lead to higher order singularities. However, the user has little control over orientations in the field (which is implicitly determined by the relative positions of the singularities), and hence the model is unsuitable at the moment for general field design.

In a series of works [31], [30], [3], [16], Zhang and colleagues pioneered a more versatile design paradigm where the user provides a set of *design elements*, which could be either singularities of particular types or prescribed orientations at certain points. These elements are represented mathematically by basis functions, which are blended to create the entire field. While allowing explicit prescription of singularities and orientations, the blending approach may yield unwanted singularities, which still requires a postprocess simplification. Fig. 2 compares the resulting line fields designed by our method and Zhang et al. approach [30]. Notice that the latter can result in additional singularities other than those specified by the user (e.g., squares in Fig. 2d (top)), which are difficult to predict or control.

A class of methods treats topology as an optimization objective when computing the direction field. To compute a smooth vector field on a surface that follows *unoriented* ridge and valley lines, Xu et al. [28] adopt a greedy heuristic to orient the feature lines, so that the diffused vector field from these oriented lines contains minimal singularities and distortion. Ray et al. [18] propose an optimization framework based on tools from Discrete Exterior Calculus (DEC), which allows the user to intuitively design the appearance of the an N-symmetry direction field on a triangulated surface while ensuring a low quantity of singularities. Editing of the field topology is also allowed by manipulating the location and indices of singularities. However, even though these methods produce good results with low singularities in practice, they do not have theoretical guarantees over the topology of the resulting fields.

Recently, a number of methods offer robust guarantees that their computed field has exactly the specified singularities. On a triangulated surface, Ray et al. [19] compute a direction at each facet as well as a *period jump* across each edge, so that the rotation around each user-specified singularity vertex meets the given index and that the field is as smooth as possible with the option of interpolating directional constraints. To do so, a greedy optimization procedure is formulated, which produces pleasing-looking fields with guaranteed topology. With a similar objective, Crane et al. [5] formulate a convex optimization problem whose variables are *adjustment angles* between directions on neighboring facets. The formulation allows them to obtain an optimally smooth field with the user-constrained topology. Lai et al. [14] treat field design on surfaces as a metric design problem with constrained holomony, and propose a robust algorithm based on flat cone metric. All these methods can be applied to general N-symmetry direction fields. However, as mentioned in the Section 1, these methods require the user to input the complete set of singularities and their indices, which can be tedious and challenging for creating a field with a nontrivial topology (e.g., Fig. 1c). In addition, since the singularity indices do

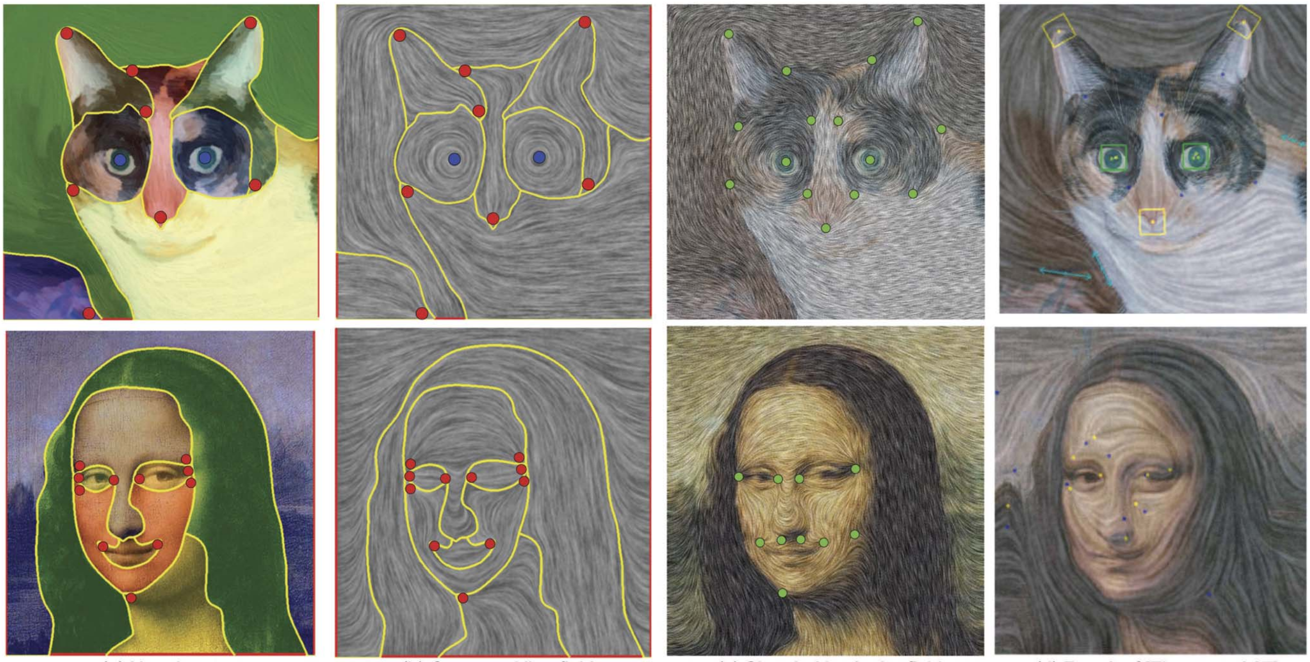


Fig. 2. Comparing the line fields generated by our method (b,c), given user input in (a) (regions boundaries as yellow curves, centers as blue dots, terminals as red dots, and flow-out segments as red lines), with those generated by the method of Zhang et al. [30] (d).

not completely describe the field topology, the user cannot precisely control various topological features such as local topology around a singularity or the separatrices.

The only method we know of that gives the user control over the complete topological structure of the field is that of Theisel [22]. Theisel proposes a piecewise construction of 2D vector field based on a full topology skeleton (singularities and separatrices) provided by the user. The field is interpolated linearly over a triangulation of the domain, which includes specially designed control polygons around the skeleton to ensure proper field behavior. While offering a finer level of topology control than other methods, Theisel's approach also increases the burden of the user: more elements need to be supplied, such as control polygons and separatrices, and the user needs to ensure the correctness of the complete skeleton (such that a field with the prescribed topology exists) in addition to checking the validity of singularity indices. Moreover, the construction of the vector field in Theisel's method highly depends on the triangulation of the domain. Additional singularities may appear if the triangulation does not have sufficient resolution to capture the flow shape, and they need to be resolved in an interactive manner.

At the first sight, our design paradigm is very similar to that of Theisel's [22], as the boundaries of the regions in our method become the topology skeleton of the resulting field. However, our method does not require the user to explicitly *construct* the topology skeleton from the singularities and separatrices. Rather, the topological structure *emerges* as the user creates the partitions and the elementary fields, and its correctness is always guaranteed. In addition, our line field is mathematically defined from the user input, independent of the discretization of the domain. The definition guarantees the prescribed topology without unwanted singularities.

Our region-based design is reminiscent of the work of Tong et al. [23], who present a method for designing cross field on surfaces for the purpose of quadrangulation. The input of the field generation is a decomposition of the surface into patches and specification of how the field should rotate (in multiples of $\pi/2$) across each patch boundary, which they call a *singularity graph*. There are several key differences between the two works. First, the singularity graph in Tong's work does not play the same role as a topology skeleton, as the resulting field does not exactly follow the patch boundaries. Second, there is no guarantee in their construction of the field that additional singularities would not occur away from the desired locations (e.g., patch corners). Finally, while Tong's method assigns a uniform flow pattern within each patch, our method offers a larger variety of flow patterns for the elementary fields, which enables creating fields with rich behaviors.

3 Background: Line Field and Its Topology

A line field over a closed, bounded 2D region $D \subset \mathbb{R}^2$ associates each point in D a unit vector with an unoriented direction (i.e., there is no difference in its forward and backward directions). Such a field can be thought as a mapping from D to the real project space $\mathbb{RP}^1$ consisting of all lines on the 2D plane passing through the origin. An example of line field is the eigenvectors of a symmetric tensor field [6], [30], which arises from many graphics and imaging applications.

The topology of a line field is largely determined by its singularities. A line field l is said to be *continuous* at a point $x \in D$ if $l(x)$ is defined, and if its direction agrees with those lines in x's immediate neighborhood. The latter can be more formally stated as follows: for any angle $\varepsilon > 0$, however small, there exists some number $\delta > 0$, such that for all $y \in D$

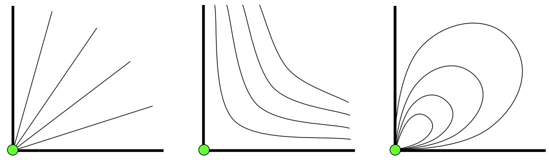


Fig. 3. Types of sectors around a singularity (green dots) with different field behaviors (indicated by thin lines).

within distance δ to x where $l(y)$ is defined, the acute angle between $l(x), l(y)$ is less than ε. If l is continuous at x, then x is said to be a *regular point* of l, otherwise it is a *singularity* of l. We consider in this paper only *isolated* singularities, which are completely surrounded by neighborhoods where l is continuously defined.

A common way to classify singularities is by their *indices* [6], which are defined as the number of counter-clockwise revolutions in the field when traveling once in a counter-clockwise direction along a closed path encompassing the singularity (and including no other singularities). Intuitively, the index captures the amount of "twist" in the field around the singularity. By definition, this index in a line field is always an integer multiple of $\pm\frac{1}{2}$, and zero at any regular point.

However, the index alone does no fully describe the field topology at a singularity. Singularities with drastically different flow patterns around them can have a common index (see examples in the second and fourth row in Fig. 5). The local topology at a singularity can be better characterized by the composition of the singularity's neighborhood into *sectors* [9]. Each sector has a distinct pattern of lines that meet tangentially at the sector boundaries. Fig. 3 depicts three types of sectors: a parabolic sector, where all lines radiate out from the singularity, a hyperbolic sector, where lines sweep pass the singularity, and an elliptic sector, where lines originate and end in the singularity. Note that the index of the singularity, in turn, can be computed from these sectors as

$$1 + \frac{n_e - n_h}{2}, \tag{1}$$

where n_e, n_h are, respectively, the number of elliptic and hyperbolic sectors around the singularity.

Another important component of line field topology is *separatrices*, which are stream lines that connect the singularities. Separatrices come into a singularity via the boundaries of its sectors. Intuitively, the separatrices partition the field into regions within which the field has a similar flow pattern. This is the key intuition that motivates our region-based design and control of field topology.

4 Elementary Fields

We start by introducing our representation of *elementary fields*, which are the design primitives and building blocks in our method. Recall that the user designs the field by partitioning the domain into individual regions and specifying the flow behavior within each region by choosing from a set of elementary fields.

To give users robust control over the field topology without limiting their creativity, this set of elementary fields should possess a number of properties. First, they should provide basic and adjustable flow patterns that can be combined to form rich flow behaviors. Second, continuity among elementary fields in neighboring regions can be easily guaranteed (except at specified singularities). Most importantly, each elementary field should have few and precisely controllable singularities, so that the composed field has a well-determined topology. Toward these goals, we consider three types of elementary fields within a simply connected 2D region, which are depicted in Fig. 4:

- **Circular Type**. The field circulates around a user-specified interior point (called the *center* and drawn as a blue dot), and is tangential along the region border.
- **Elliptic Type**. The field circulates around a user-specified point on the region border (called a *terminal* and drawn as a red dot), and is tangential everywhere else along the border.
- **Flow Types A, B, and C**. The field flows from one user-specified point on the region border (called a *terminal* and drawn as a red dot) or a segment of the border (called a *flow-out segment* and drawn as a red curve) to another terminal or flow-out segment on the border.

These elementary fields cover some of the most common flow patterns, which we have found to be powerful enough to compose a wide variety of 2D line fields. Each elementary field is tangential to their border away from the terminals and the flow-out segments, making it trivial to piece together a continuous field. When continuity is not a concern, such as at the domain boundary, the flow-out segments in Flow Types B and C can be useful for creating a more natural flow. More importantly, as we will see next, we can represent each of the elementary field mathematically to ensure that the singularities in the field are located

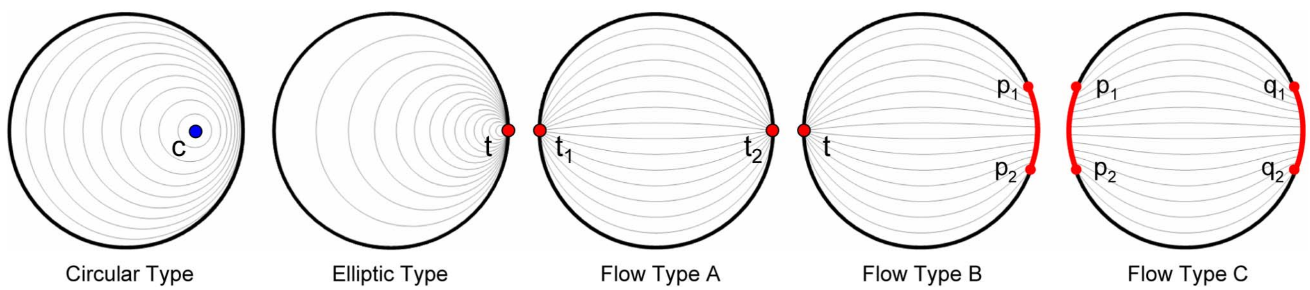


Fig. 4. Five types of elementary fields considered in our method, each defined by level curves of a harmonic function.

precisely at the centers and terminals, and that the topological skeleton of the composed field matches exactly with the region boundaries.

In the following, we first present the definitions of each elementary field. We will then discuss the topology of the composed field.

4.1 Defining Elementary Fields

One way to represent a 2D line field $l(x)$ is by tangent directions on the level curves of some 2D scalar function $w(x)$. That is, l, w satisfy the following relation:

$$l(x) \cdot \nabla w(x) = 0,$$

where $\nabla w(x)$ is the gradient of w at x. If w is sufficiently smooth (e.g., with well-defined partial derivatives), l is continuously defined except at the *critical points* of w where the partial derivatives are all zero. Using this representation, the task of constructing a line field l with controllable singularities becomes finding a scalar function w with deterministic critical points.

To this end, we consider w that are *harmonic functions*, which satisfy the Laplace equation

$$\Delta w = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = 0. \quad (2)$$

The critical points of such functions have been well studied in the literature of differential equations. In particular, a class of harmonic functions have well-determined, isolated critical points in a simply connected region (i.e., free of interior holes) with piecewise smooth boundary [26], which we exploit in our work to define the elementary fields.

Given a simply connected region R with a piecewise smooth boundary B, our choice of the harmonic function w for each elementary field is defined below. We prove in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.112, that the only critical points of these harmonic functions appear at the center in Circular Type, the terminals in Elliptic and Flow Types, as well as tangent discontinuities along B ("corners").

4.1.1 Circular Type

Denote the center as $c \in R$. We define $w(x)$ as the *Green's function* $G(c, x)$ with *pole* at c. Note that $w(c) = -\infty$ and evaluates zero over B, hence creating the concentric look of the line field.

4.1.2 Elliptic Type

Denote the terminal as $t \in B$. We define $w(x)$ as the harmonic function that interpolates the *Dirac delta function* over B with a peak at t. That is, $w(x)$ is zero for every $x \in B$ except at t where $w(t) = \infty$, and the integral of $w(x)$ over B is 1. By Green's third identity, such $w(x)$ uniquely exists, and equals the directional derivative of the Green's function $G(t, x)$ in the normal direction at t. Due to the jump in boundary values, the level curves of w consist of orbits with increasing radius that all touch t.

4.1.3 Flow Types A, B, and C

Denote the terminals as t, t_1, t_2 and the ends of the flow-out segments as p_1, p_2, q_1, q_2, as shown in Fig. 4. We define $w(x)$ as the harmonic function that interpolates the following piecewise, possibly discontinuous boundary conditions on B:

- *Type A.* $w(x)$ is 1 for x on the boundary segment $[t_2, t_1]$ (traveling counter clockwise), and 0 otherwise.
- *Type B.* $w(x)$ is 1 for x on the segment $[p_1, t]$, 0 for x on the segment $[t, p_2]$, and assumes a monotonic function from 0 to 1 with zero derivatives at the two ends as x travels on B from p_2 to p_1.
- *Type C.* $w(x)$ is 1 for x on the segment $[q_1, p_1]$, 0 for x on the segment $[p_2, q_2]$, and assumes a monotonic function from 0 to 1 with zero derivatives at the two ends as x travels on B from p_1 to p_2 or from q_2 to q_1.

The harmonic functions satisfying these discontinuous boundary conditions uniquely exist [27]. Since the boundary values of $w(x)$ exhibit a slope from 1 to 0, the level curves flow from one side to the other, and are tangential to the non-flow-out parts of B where the boundary values are constant.

4.2 Topology Analysis of Composed Field

A line field composed from elementary fields defined above has a topology structure that is completely determined by the user inputs

- Singularity locations. If we assume all region boundaries to be smooth curves except at where more than two boundary curves meet ("joints"), the singularities in the composite field would consist solely of centers, terminals, and joints.
- Singularity topology. The local topology of each singularity is completely governed by the elementary fields. Except for the center in the Circular Type, all singularities lie on some boundary shared by two or more regions. The boundary curves incident to a singularity x divide its neighborhood into sectors, whose types are determined by the type of elementary field l in that sector. If l is the Elliptic Type and x is its terminal, the sector is elliptic. If l is a Flow Type and x is one if its terminals, the sector is parabolic. If x is neither a terminal nor part of a flow-out segment, the sector is hyperbolic regardless of the type of l.
- Separatrices. Since the boundary curves are tangential to the elementary fields and connect the singularities, they are coincident with the separatrices in the composed field.

Our method allows for creating a wide range of topological structures. As examples, Fig. 5 demonstrates a variety of singularities composed from two or three elementary fields. The index of a singularity x in the composed line field can be computed from its neighboring elementary fields as

$$1 + \frac{r_e - r_h}{2}, \quad (3)$$

where r_e is the number of Elliptic Type fields where x is the terminal, and r_h is the number of neighboring fields where x is not a terminal. It is evident from (3) that our method can produce singularities with any indices. It is worth noting that the index only cannot fully characterize the local topology. As seen in Fig. 5, some singularities share the same singularity index and yet having dramatically different local topology.

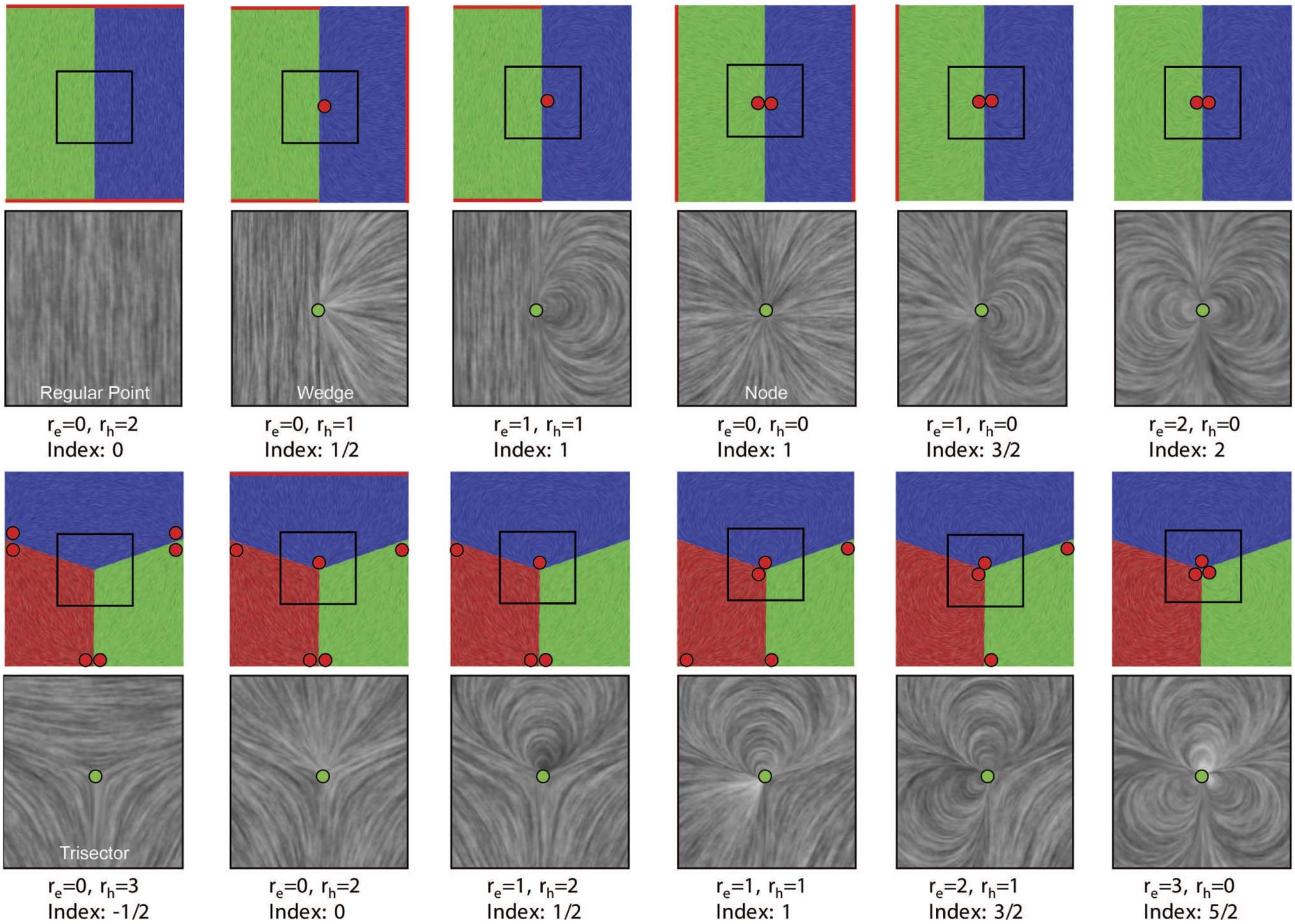


Fig. 5. Composing elementary functions in two and three abutting regions (red dots and lines represent terminals and flow-out segments), and closeups around the singularities (green dots, second and fourth rows). The captions show the number of Elliptic Type regions where the singularity is a terminal (r_e), the number of regions of any type where the singularity is not a terminal (r_h), and the singularity index computed by (3). For visual clarity, the terminals on the boundary of a region are drawn slightly inside the region to distinguish between terminals belonging to neighboring regions that share the same location.

5 Discrete Computation

Since harmonic functions do not have explicit expressions in arbitrarily shaped regions, the piecewise smooth line field defined above does not allow explicit evaluation at a given point in a 2D domain. In the following, we describe a discrete algorithm for approximating the elementary fields on a triangulated domain by computing the level sets of a *discrete harmonic function*. We show that these discrete functions have exactly controllable critical points like their continuous counterparts, but in a discrete sense. The result of the algorithm is a piecewise constant line field where each triangle facet is associated with a line direction.

5.1 Input

The input of our algorithm is a triangulated 2D domain. Each region is represented by a collection of triangles, whose boundary consists of a closed and manifold loop of triangle edges. The centers, terminals, and ends of flow-out segments are located at triangle vertices. In addition, we require that there is no direct edge connecting any two boundary vertices that are not consecutive along a boundary loop. This requirement is important to derive the topology guarantees in the resulting discrete harmonic functions (see details in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.112).

5.2 Setting Boundary Values

The harmonic functions in all five types of elementary fields can be considered as solutions to the Laplace equation subject to certain boundary value conditions. However, some of these boundary values are impossible to realize in a discrete setting, such as an infinite value or a discontinuity at a single boundary point. We, therefore, use the following approximated boundary values for each type of elementary field. In Circular Type, we place 1 at the center vertex and 0 on all vertices on the region boundary. In Elliptic Type, we place 1 at the terminal vertex and 0 on all remaining boundary vertices. In Flow Types, we place 0.5 at each terminal vertex, 1 (0) at all vertices on the top (bottom) portion of the non-flow-out parts of the boundary, and linearly interpolated values between 0 and 1 at vertices on flow-out segments.

5.3 Computing Discrete Harmonic Functions

We use a technique similar to Ni et al. [15] to compute the discrete harmonic functions that satisfy the boundary value conditions as stated above. The Laplace equation in the continuous setting (2) is discretized into a system of linear equations, whose unknowns are scalar values w_i at each *free* vertex v_i that does not have an initial value (which include all interior vertices except for the center in Cyclic type)

$$L(w_i) = w_i - \sum_{j \in N(i)} b_{ij} w_j = 0. \quad (4)$$

Here, $N(i)$ is the 1-ring neighborhood of v_i, and the weight $b_{ij} = \frac{\tan(\alpha_{ij}/2)+\tan(\beta_{ij}/2)}{\|v_i - v_j\|}$ are the mean value weights [8] where α_{ij}, β_{ij} are the angles made by the edge $\{v_i, v_j\}$ and its two immediate neighboring edges at v_i.

The solution to (4) uniquely exists [15]. The solved values at the vertices define a piecewise linear function over the triangulated region, where values within each facet are linearly interpolated from values at its vertices. Ni et al. [15] show that this discrete function is free of local maxima or minima at all free vertices, a fact drawn from the positivity of the mean value weights b_{ij}.

Here, we assert an even stronger statement about the discrete harmonic functions computed using our particular choice of boundary values: these functions are free of not only maxima or minima, but also *saddles*. More precisely, the functions are free of *any* critical points where the level set of the function changes topology, except at the centers and terminals (see proof in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.112). In addition, the level sets of the functions are tangential at the boundary edges except at the edges next to the terminal vertices and those in the flow-out segments. In sum, these discrete functions preserve the key topological and geometric properties of the continuous harmonic functions.

After solving (4), we associate each triangle facet with the level curve direction of the discrete harmonic function within that triangle. Such direction always exists (due to our no-critical-point guarantee), and can be easily computed from the values at the triangle vertices.

6 User Interface

To facilitate the design of line fields, we developed an interactive tool where a user can easily create and edit inputs to our field construction, including the region boundaries and the type of elementary field within each region. The discretely computed field is visualized using the technique in [30]. We next detail the user interaction and the discretization of the user inputs.

Our tool allows the user to create smooth region boundaries as cubic B-spline curves, so that the singularities in the field will lie only at the centers, terminals, and where more than two regions meet. When an underlying image is available (e.g., for painterly rendering), the tool allows the user to create regions via interactive image segmentation. We implement the "live wire" [1] technique which allows the user to sketch out image segments (see an example is Fig. 6a). The segmentation boundaries are then automatically fit with cubic B-spline curves that can be further edited (Fig. 6b). For each region, the user can select the type of elementary field, and specify the locations of centers, terminals, and flow-out segments. Please refer to the attached video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.112, for a demonstration of the interaction process.

(a) (b)

Fig. 6. Creating region boundaries on an image: (a) Sketching using live wire. (b) Editing using cubic B-spline curves.

To discretize the input, we first label the pixels by the region that cover them. The pixel grid is then converted into a triangulation by dividing each pixel into two triangles. If an edge directly connects two nonconsecutive vertices on a same boundary, the edge as well as the incident triangles are split by the midpoint of the edge. To yield smoother boundaries, the triangle mesh is smoothed iteratively using Laplacian-based fairing [21], where each boundary vertex is smoothed using only the locations of its neighboring vertices on the same boundary curve. To avoid folding of the mesh, a small number of iterations is used (e.g., 10). The discrete algorithm described in the previous section is then performed on the smoothed triangle mesh.

7 Results

Here, we present a suite of line fields designed using our interface, and the use of these fields in stylization of the images. For visual clarity, the user-placed terminals in all examples are drawn slightly inside their containing region so as to distinguish between terminals belonging to neighboring regions that share the same location.

7.1 Line Fields

Using the interface, interesting line fields can be created with only a few partitions of the domain, as demonstrated in Fig. 2 and the starfish example in Fig. 7. Our algorithm ensures a simple and low-distortion appearance without unwanted singularities. Moreover, the user has precise control over the topology, which is important for designing fields with high-order singularities (such as the center singularity on the left of Fig. 7).

The strength of our design approach is best demonstrated in creating line fields with complex behaviors, such as the Starry Night example in Fig. 1 and the Rhino example in Fig. 10. Without precise topology control, the field can be easily cluttered with unwanted singularities. On the other hand, while it is possible to create these fields using previous singularity-based design methods, it would be a very tedious and challenging task for the user to specify the complete set of singularities.

Our interface allows users to not only create the field, but also edit the field by modifying the regions and their elementary fields. Using this feature, one can design a complex field in a top-down manner by incrementally refining a coarse field. Fig. 7 right shows an example: the region layout for the bird on the left was modified from an original layout that mirrors the one on the right. The modification adds more details in the head, the wing, and the feet.

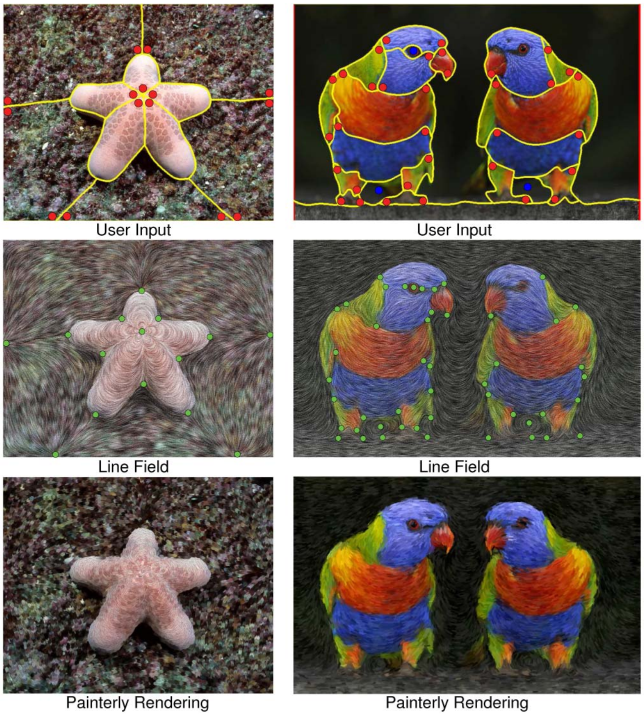


Fig. 7. User input (top), the resulting field (middle), and painterly rendered result (bottom) in two examples.

7.2 Painterly Rendering

Artists often use strokes with orientations to illustrate complex scenes, natural phenomena, or abstract ideas. The "Starry Night" in Fig. 1 is a good example, where Van Gogh employed colorful and long strokes along the cloud flows to express cloud movements. Painterly rendering [11], [10], [29] is a stylization method for images that attempt to imitate artists' painting using directed strokes. As a result, a natural, coherent direction field is essential to this goal.

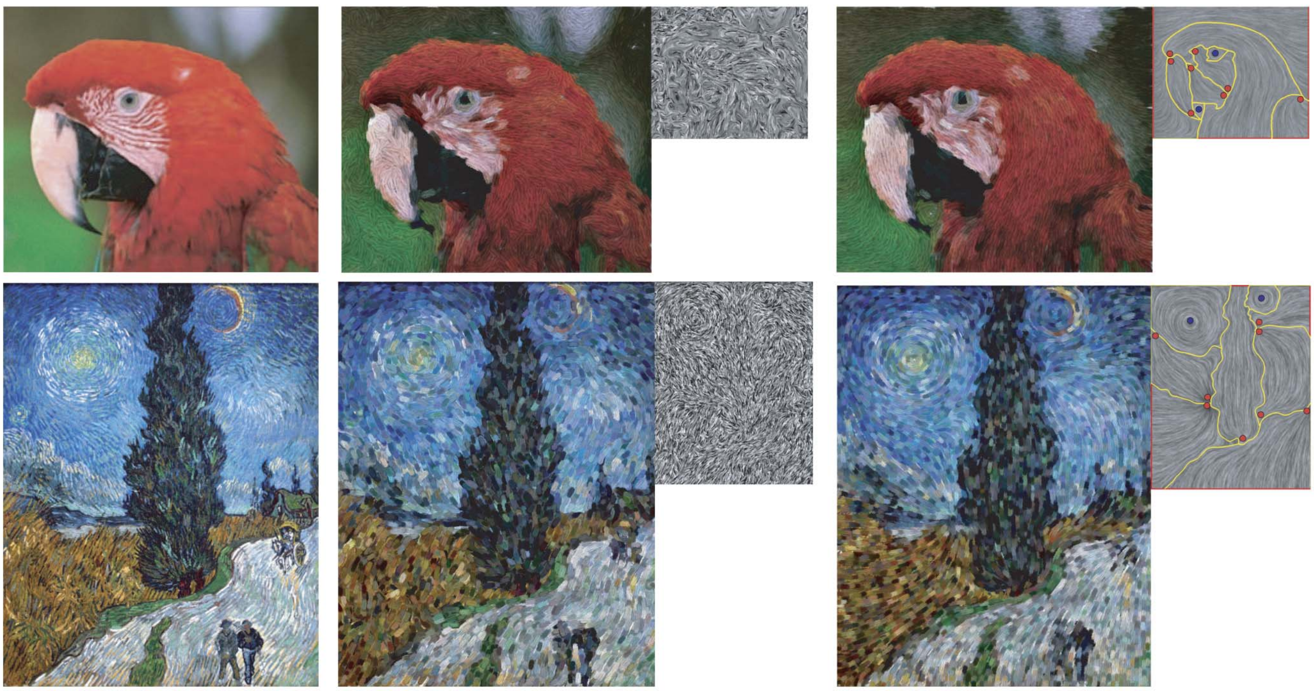
Examples	Resolution	# Regions	Discretization Time(s)	Field Computing Time(s)	Total Time(s)
Starynight	895x561	17	23.46	0.635	24.095
Fish	640x512	10	7.796	0.418	8.214
rhino	640x400	13	5.942	0.316	6.258
Monalisa	557x578	8	15.514	0.414	15.928
Painting Cat	662x661	8	17.9	0.563	18.463
eagle	640x480	8	7.192	0.397	7.589
frog	640x400	12	7.942	0.326	8.268
Starfish	600x480	10	4.595	0.368	4.963
Colorful Bird	800x500	22	31.887	0.582	32.469

Fig. 9. Performance of our implementation.

Existing image stylization techniques often rely on automated approaches for computing a direction field from the images. However, it can be a difficult task to infer a smooth field with low distortions from casual images. In Fig. 8, we compare the painterly rendered results of [29] using our interactively designed line fields with using fields generated automatically from the images by the state-of-art method of Kang et al. [13]. Note that our line fields have a more coherent appearance, yielding in stylized images that "flow" more smoothly (although some details in the original paintings are lost). More painterly rendering results using line fields designed by our tool are shown in Figs. 10 and 11.

7.3 Performance

Our experiments were conducted on a consumer level PC with an Intel i5-750 processor and 4 GB memory. We report the running time of our tool on several examples in Fig. 9. Note that the most time-consuming step is discretizing the



(a) Original image (b) Field by [12] and the rendering (c) Field by our method and the rendering

Fig. 8. Comparing painterly rendered images using line fields (shown in thumbnails) automatically extracted from the images [13] (b) and interactively designed using our method (c).

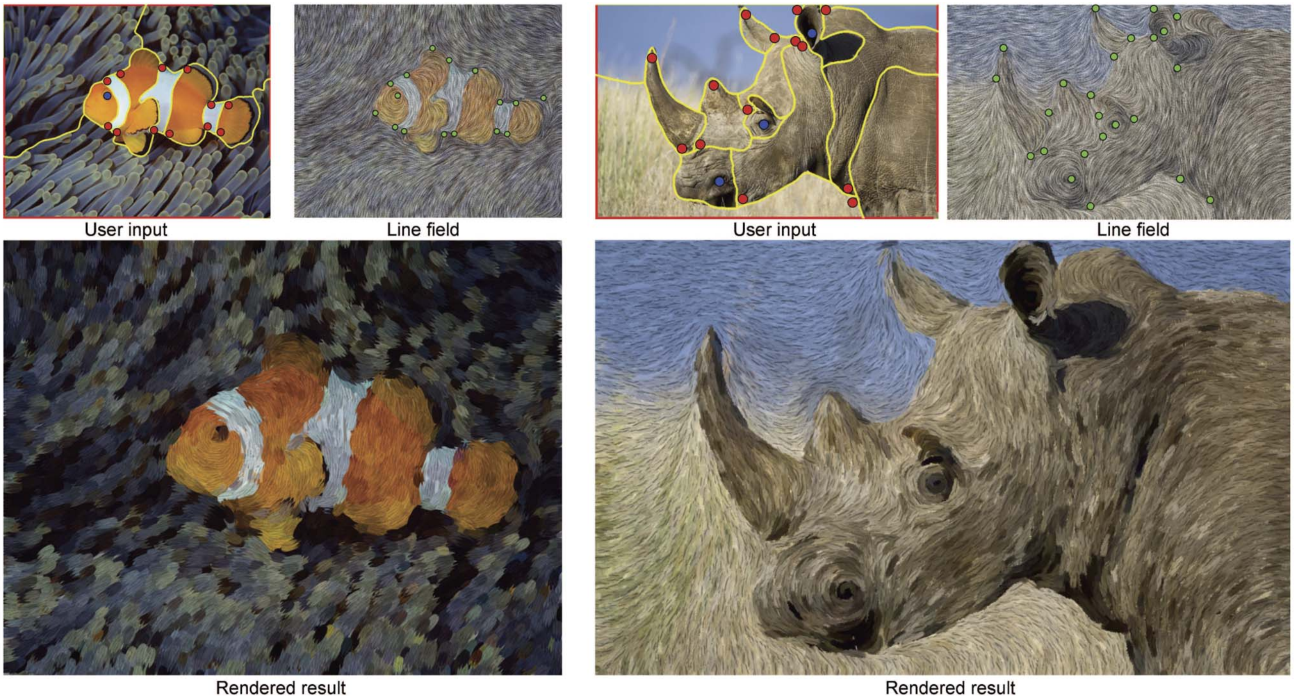


Fig. 10. Painterly rendered results using the designed fields.

Fig. 11. A gallery of painterly rendered images.

domain with user inputs into a labeled, triangulated mesh (see Section 6). The actual computation of the line field on the discretized domain is quite efficient, finishing under 1 second in all cases. The efficient computation allows interactive viewing of the field as the user relocates centers and terminals or changes the type of elementary fields (which do not change the discretization).

8 Conclusion and Discussions

We present in this paper a novel approach for designing 2D line fields using piecewise harmonic functions. We show that a carefully chosen set of harmonic functions can serve as flexible design primitives for creating complex flow patterns while providing exact control over the location and flow behavior of singularities. Based on the mathematical definition, we present a simple and robust discrete algorithm, and developed an interactive tool with the utility of the field in image stylization.

8.1 Limitations

The main drawback of our current design interface is that the user has to manually partition the domain into regions. A potentially much more convenient interaction mode, which we will investigate in the future, is to ask users to provide rough strokes indicating major flow directions, and to automatically create region partitions as well as the locations of centers and terminals from these strokes (with

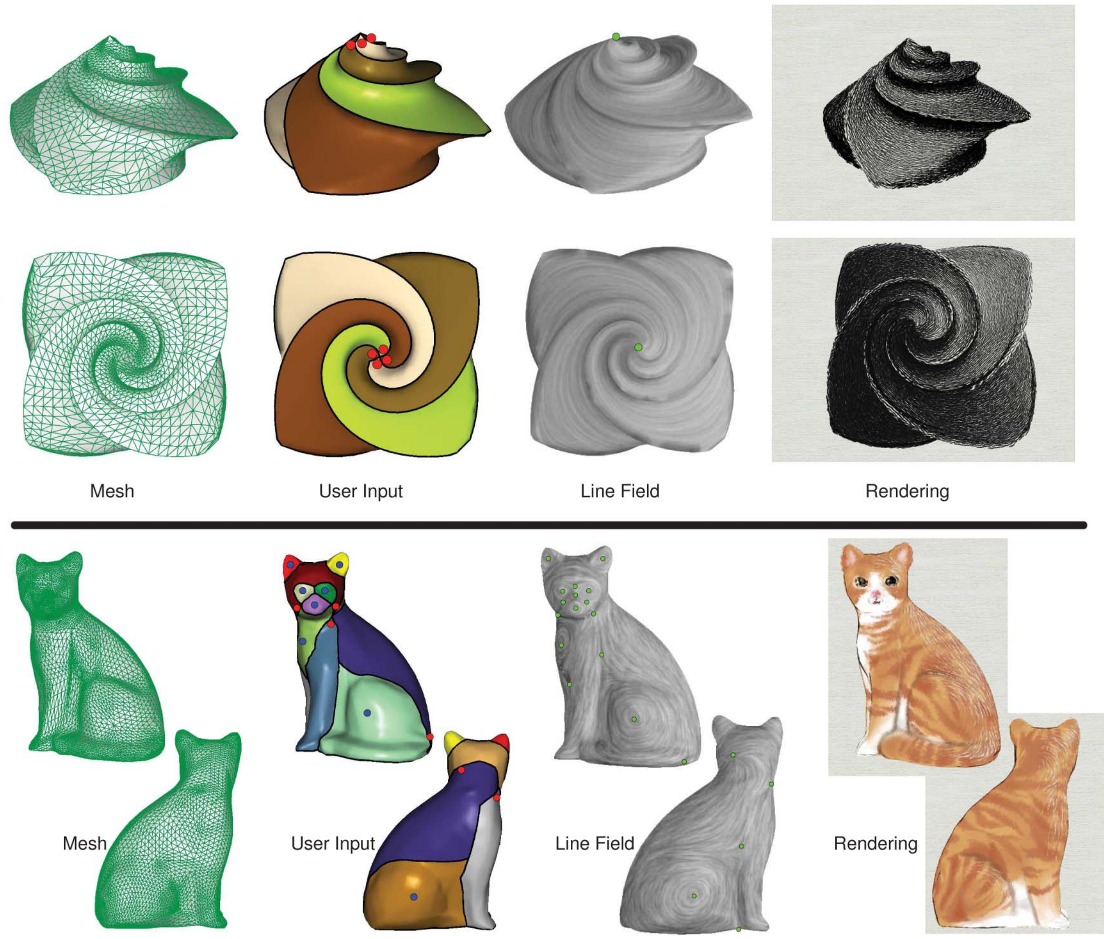


Fig. 12. Designing line fields on two surfaces (top and bottom). Each example shows the mesh structure, the user input (regions boundaries, centers, and terminals), the discretely computed line field, and a painterly rendering result using [4].

or without a background image). Another drawback in our method is that the only orientation control currently is by the shape of the region boundaries. Accepting additional directional constraints interior to the regions, while maintaining the topology correctness, is another direction we would like to address in the future.

8.2 Future Work

It would be most interesting to extend our method onto 3D surfaces. Ideally, we would like to define line fields on surfaces in a piecewise fashion using harmonic functions as we have done in 2D, so that it has the same topological guarantees. Although harmonic functions are well defined on arbitrary Riemannian manifolds, the primary challenge we need to address is extending the theories about critical points from 2D harmonic functions [26] onto Riemannian surfaces, which so far has not been studied to the best of our knowledge.

Despite the lack of theories in the continuous case, our algorithm for computing a discrete line field can be applied to any triangulated domains, including surfaces in 3D. Given a surface partitioned into simply connected patches consisting of triangles (and the centers, terminals, and ends of flow-out segments are located at triangle vertices), the same algorithm in Section 5 computes one direction at each triangle by the level curve of a discrete harmonic function within each patch. Note that the discrete harmonic function in 3D shares the same guarantee of few and controlled critical points as in 2D. As examples, we show in Fig. 12 line fields designed on two closed surfaces, and stylized rendering using the line fields [4]. In these examples, the user manually segmented the surface into patches along the triangle edges in a third-party modeling tool. As in 2D, we would like to explore in our future work more convenient means for segmenting a surface and for locating the centers and terminals, possibly by user strokes.

Another direction of future work is to study other elementary fields that would increase the flexibility of design, such as fields defined in nonsimply connected regions (e.g., a region with holes). These fields will be helpful to reduce the number of partitions, particularly on a surface (e.g., so that the side of a cylinder can stay as a single partition). We would also like to explore means for controlling the flow direction within an elementary field, for example, based on user-provided sketches within a region. Finally, we would like to extend the region-based approach for designing direction fields with higher order symmetry.

Acknowledgments

This work is supported in part by US National Science Foundation (NSF) grant IIS-0846072 and National Science Council (contracts NSC-99-2221-E-006-066-MY3, NSC-99-2221-E-004-014, NSC-100-2628-E-006-031-MY3, and NSC-100-2221-E-006-188-MY3), Taiwan.

REFERENCES

[1] W.A. Barrett and E.N. Mortensen, "Interactive Live-Wire Boundary Extraction," *Medical Image Analysis,* vol. 1, pp. 331-341, 1997.

[2] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive Procedural Street Modeling," *ACM Trans. Graphics,* vol. 27, no. 3, pp. 103:1-103:10, 2008.

[3] G. Chen, K. Mischaikow, R.S. Laramee, P. Pilarczyk, and E. Zhang, "Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 4, pp. 769-785, July 2007.

[4] M.-T. Chi and T.-Y. Lee, "Stylized and Abstract Painterly Rendering System Using a Multiscale Segmented Sphere Hierarchy," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 1, pp. 61-72, Jan./Feb. 2006.

[5] K. Crane, M. Desbrun, and P. Schröder, "Trivial Connections on Discrete Surfaces," *Computer Graphics Forum,* vol. 29, no. 5, pp. 1525-1533, 2010.

[6] T. Delmarcelle and L. Hesselink, "The Topology of Symmetric, Second-Order Tensor Fields," *VIS '94: Proc. IEEE Conf. Visualization '94,* pp. 140-147, 1994.

[7] L. Fan, S. Wang, H. Wang, and T. Guo, "Singular Points Detection Based on Zero-Pole Model in Fingerprint Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 6, pp. 929-940, June 2008.

[8] M.S. Floater, "Mean Value Coordinates," *Computer Aided Geometric Design,* vol. 20, no. 1, pp. 19-27, 2003.

[9] X.T. Gerik, X. Tricoche, G. Scheuermann, H. Hagen, and S. Clauss, "Scaling the Topology of Symmetric, Second-Order Tensor Fields," *Proc. NSF/DOE Lake Tahoe Workshop Hierarchical Approximation and Geometrical Methods for Scientific Visualization,* pp. 171-184, 2002.

[10] J. Hays and I.A. Essa, "Image and Video Based Painterly Animation," *Proc. Third Int'l Symp. Non-Photorealistic Animation and Rendering (NPAR),* pp. 113-120, 2004.

[11] A. Hertzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes," *Proc. SIGGRAPH '98,* pp. 453-460, 1998.

[12] F. Kalberer, M. Nieser, and K. Polthier, "Quadcover - Surface Parameterization Using Branched Coverings," *Computer Graphics Forum,* vol. 26, no. 3, pp. 375-384, 2007.

[13] H. Kang, S. Lee, and C.K. Chui, "Flow-Based Image Abstraction," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 1, pp. 62-76, Jan./Feb. 2009.

[14] Y.-K. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S.-M. Hu, and X. Gu, "Metric-Driven Rosy Field Design and Remeshing," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 1, pp. 95-108, Jan./Feb. 2010.

[15] X. Ni, M. Garland, and J.C. Hart, "Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 613-622, 2004.

[16] J. Palacios and E. Zhang, "Rotational Symmetry Field Design on Surfaces," *ACM Trans. Graphics,* vol. 26, no. 3, pp. 55:1-55:10, 2007.

[17] N. Ray, W.C. Li, B. Lévy, A. Sheffer, and P. Alliez, "Periodic Global Parameterization," *ACM Trans. Graphics,* vol. 25, no. 4, pp. 1460-1485, 2006.

[18] N. Ray, B. Vallet, L. Alonso, and B. Levy, "Geometry-Aware Direction Field Processing," *ACM Trans. Graphics,* vol. 29, no. 1, pp. 1:1-1:11, 2009.

[19] N. Ray, B. Vallet, W.C. Li, and B. Lévy, "N-Symmetry Direction Field Design," *ACM Trans. Graphics,* vol. 27, no. 2, pp. 10:1-10:13, 2008.

[20] B. Sherlock and D. Monro, "A Model for Interpreting Fingerprint Topology," *Pattern Recognition,* vol. 26, no. 7, pp. 1047-1055, 1993.

[21] G. Taubin, "A Signal Processing Approach to Fair Surface Design," *Proc. SIGGRAPH '95,* pp. 351-358, 1995.

[22] H. Theisel, "Designing 2D Vector Fields of Arbitrary Topology," *Computer Graphics Forum (Eurographics),* vol. 21, pp. 595-604, 2002.

[23] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, "Designing Quadrangulations with Discrete Harmonic Forms," *SGP '06: Proc. Fourth Eurographics Symp. Geometry Processing,* pp. 201-210, 2006.

[24] Y. Tong, S. Lombeyda, A.N. Hirani, and M. Desbrun, "Discrete Multiscale Vector Field Decomposition," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 445-452, 2003.

[25] X. Tricoche, G. Scheuermann, and H. Hagen, "Continuous Topology Simplification of Planar Vector Fields," *VIS '01: Proc. Conf. Visualization '01,* pp. 159-166, 2001.

[26] J.L. Walsh, *The Location of Critical Points of Analytic and Harmonic Functions.* Am. Math. Soc., 1950.

[27] N. Wiener, "Discontinuous Boundary Conditions and the Dirichlet Problem," *Trans. the Am. Math. Soc.,* vol. 25, no. 3, pp. 307-314, 1923.

[28] K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong, "Feature-Aligned Shape Texturing," *ACM Trans. Graphics,* vol. 28, no. 5, pp. 108:1-108:7, 2009.

[29] C.-R. Yen, M.-T. Chi, T.-Y. Lee, and W.-C. Lin, "Stylized Rendering Using Samples of a Painted Image," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 2, pp. 468-480, Mar./Apr. 2008.

[30] E. Zhang, J. Hays, and G. Turk, "Interactive Tensor Field Design and Visualization on Surfaces," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 1, pp. 94-107, Jan./Feb. 2007.

[31] E. Zhang, K. Mischaikow, and G. Turk, "Vector Field Design on Surfaces," *ACM Trans. Graphics,* vol. 25, no. 4, pp. 1294-1326, 2006.

Chih-Yuan Yao was born in Tainan, Taiwan in 1980. He received the MS and PhD degree in computer science and information engineering from National Cheng-Kung University, Taiwan, in 2003 and 2010, respectively. His research interests are computer graphics, including mesh processing and modeling, and nonphotorealistic rendering (NPR).

Ming-Te Chi received the MS and PhD degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, in 2003 and 2008, respectively. Since 2008, he has been with the Department of Computer Science, National Chengchi University, as an assistant professor. His research interests include computer graphics and nonphotorealistic rendering. He is a member of the IEEE and the ACM.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a distinguished professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, R.O.C. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He also serves on the editorial boards of the *IEEE Transactions on Information Technology in Biomedicine*, *the Visual Computer and the Computers and Graphics Journal*. He served as a member of the international program committees of several conferences including the IEEE Visualization, the Pacific Graphics, the IEEE Pacific Visualization Symposium, the IEEE Virtual Reality, the IEEE-EMBS International Conference on Information Technology and Applications in Biomedicine, and the International Conference on Artificial Reality and Telexistence. He is a senior member of the IEEE and the member of the ACM.

Tao Ju received the MS and PhD degrees in computer science at Rice University in 2005. He is an associate professor in the Department of Computer Science and Engineering at the Washington University in St. Louis. He conducts research in computer graphics and bio-medical applications, and is particularly interested in geometric modeling and processing.