Customizing Painterly Rendering Styles Using Stroke Processes

Mingtian Zhao* Song-Chun Zhu*
University of California, Los Angeles & Lotus Hill Institute

Figure 1: *Paintings of two customized styles rendered using our method. (a) has lower lightness contrast than (b) thus appears a little more mellow. Zoom to 400% to view details. Their corresponding source photograph is at the top-left of Fig.3.*

Abstract

In this paper, we study the stroke placement problem in painterly rendering, and present a solution named *stroke processes*, which enables intuitive and interactive customization of painting styles by mapping perceptual characteristics to rendering parameters. Using our method, a user can adjust styles (e.g., Fig.1) easily by controlling these intuitive parameters. Our model and algorithm are capable of reflecting various styles in a single framework, which includes point processes and stroke neighborhood graphs to model the spatial layout of brush strokes, and stochastic reaction-diffusion processes to compute the levels and contrasts of their attributes to match desired statistics. We demonstrate the rendering quality and flexibility of this method with extensive experiments.

CR Categories: I.3.4 [Computer Graphics]: Graphics Utilities—Paint Systems; I.4.10 [Image Processing and Computer Vision]: Image Representation—Statistical; J.5. [Computer Applications]: Arts and Humanities—Fine Arts

Keywords: contrast, painterly rendering, perceptual characteristic, point process, reaction-diffusion, stroke-based rendering

*e-mails: {mtzhao|sczhu}@stat.ucla.edu

1 Introduction

Among various techniques of non-photorealistic computer graphics [Gooch and Gooch 2001; Strothotte and Schlechtweg 2002], stroke-based painterly rendering [Hertzmann 2003] simulates the common practices of human painters who create paintings with brush strokes. This is complex since every single stroke depends on many factors, including the scene and objects to depict, the theme and style to express, many previously painted strokes on the canvas, etc. Technically, this problem has two main aspects, namely *brush modeling* and *stroke placement* [Hertzmann 2003; Zeng et al. 2009]. While the former can hopefully be achieved by balancing visual fidelity and computational feasibility, the latter involves subtle subjective factors such as styles and feelings. To paint flexibly like human artists, the computer should capture these factors from users and reflect them in rendering.

For brush modeling, procedural and example-based methods have been proposed, and some can work fairly well. But for stroke placement, progress is less satisfactory. Existing methods simply place strokes sequentially in a greedy manner, or do it by optimizing complex energy functions. In neither way is it convenient to *map intuitive perceptual characteristics to rendering parameters*, for example, "vibrant colors" and "gestural strokes" which appear in many of Vincent van Gogh's paintings. This makes them less convenient to customize and thus unfriendly for interactive usage.

It is noticed that painting styles are usually expressed and recognized through such intuitive perceptual characteristics, which we call *perceptual dimensions*. For style customization in painterly rendering, it is nice to have direct control of these dimensions for each object in the source image. To achieve this, we adopt eight intuitive system parameters defined below which correspond to common perceptual dimensions, and users can interactively control them to achieve desired styles.

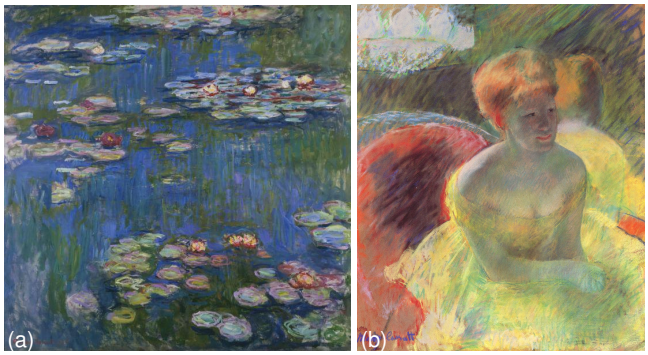

Figure 2: *Two painting masterpieces exhibiting sharp contrast among neighboring strokes: (a) "Water Lilies" by Claude Monet, and (b) "Lydia Leaning on Her Arms" by Mary Cassatt.*

Density: Stroke density is proportional to the number of strokes inside a unit image area.

Non-Uniformity: The degree of unevenness of the spatial density of strokes. A high non-uniformity level means strokes are very dense in some places but very sparse elsewhere.

Local Isotropy: The degree of similarity of stroke orientations in a neighborhood inside an image region. A high local isotropy level means neighboring strokes are usually near-parallel, exhibiting a smoothed style with low contrast in orientation.

Coarseness: The average size of strokes. Generally, the larger the stroke sizes are, the coarser the rendered painting image is.

Size Contrast: The local variance of size, represented by the size differences between each stroke and its neighboring strokes.

Lightness Contrast: The differences in lightness of color between each stroke and its neighbors.

Chroma Contrast: The differences in chroma of color between each stroke and its neighbors.

Hue Contrast: The differences in hue of color between each stroke and its neighbors.

Compared with previous methods focusing on regular features such as stroke size and color, our parameter design explicitly emphasizes the manipulation of spatial *contrast* among neighboring strokes in five of the eight parameters listed above (i.e., local isotropy, and contrasts in size, lightness, chroma and hue). This emphasis is inspired by painters' experience [Cooke 1978] according to which contrast, sometimes called *tempo* by artists, is an intuitive yet powerful tool to depict styles, as we can observe in many famous paintings. For example, the two masterpieces in Fig.2 exhibit vibrant color tempos, which correspond to high contrasts in hue and lightness among neighboring strokes. Our choice to highlight contrast is also supported by the successful use of wavelet and texton features for classifying painting styles [Wallraven et al. 2009; Hughes et al. 2010]. Note that what these features capture are patch-level local contrasts in painting images. Fig.1 displays an example of our rendering results, in which the two paintings are generated with parameters differing only in lightness contrast.

Upon the eight perceptual dimensions, we propose our method named *stroke processes*, which consists of a series of stochastic processes, including point processes [Stoyan et al. 1996] to model the spatial layout of brush strokes, and stochastic *reaction-diffusion* processes [Turk 1991; Zhu and Mumford 1997] to compute the levels and contrasts of their attributes to match desired statistics. Reaction-diffusion is originally used for modeling the physical processes of the chemical reaction among substances and their diffusion in space. In our method, we diffuse attributes among strokes to reduce or enhance (using negative diffusion rates) their contrasts. The reaction with our applied external forces is for preserving information from the source image. In order to simulate the reaction-diffusion, we connect neighboring strokes to build a graph, along whose edge connections we are able to apply the diffusion.

The main contributions of this paper include (1) a parameter design emphasizing contrasts, which are commonly utilized by human painters to reflect styles, (2) a novel stroke neighborhood graph model to represent the relations among strokes, and (3) a fast algorithm to compute stroke attributes, enabling interactive control. Details of the model and algorithm are explained in Section 3.

2 Related Work

Research in stroke-based painterly rendering has achieved encouraging progress. For brush modeling, Strassmann [1986] is among the earliest to study painterly graphical elements. After that, people have developed various improved methods [Cockshott et al. 1992; Meier 1996; Litwinowicz 1997; Hertzmann 1998; Hertzmann 2002; Baxter 2004; Zeng et al. 2009; Chu et al. 2010]. This paper is not going to study brush modeling. We simply adopt the example-based method of Zeng et al., and use a dictionary containing around 200 textured brush strokes. But other models, either procedural or example-based, are also compatible with our method.

For stroke placement, there are greedy and optimization-based methods [Hertzmann 2003]. In a greedy strategy, at each step, the algorithm determines the current stroke according to certain objectives and image/semantic features [Haeberli 1990; Litwinowicz 1997; Hertzmann 1998; Collomosse and Hall 2002; Gooch et al. 2002; Hays and Essa 2004; Zeng et al. 2009; Lu et al. 2010; Zhao and Zhu 2010]. The optimization-based methods compute the entire sequence of strokes together to achieve optimal global energies or desired statistics [Turk and Banks 1996; Hertzmann 2001; Vanderhaeghe et al. 2007; Hurtut et al. 2009]. Theoretically, optimization-based methods have the potential to outperform greedy ones, since they can *explicitly* model the interactions among strokes. These interactions, or high-order statistics among the strokes, essentially control the spatial contrasts mentioned above.

Our method belongs to the optimization-based class, and it improves previous work in two main aspects. (1) It has a parameter design which explicitly emphasizes contrasts or high-order statistics, while parameters in most previous methods only correspond to either individual strokes or global features [Hertzmann 1998; Hertzmann 2001; Hays and Essa 2004; Zeng et al. 2009; Lu et al. 2010] thus lack the power to reflect effects such as "complementary colors in neighboring strokes." (2) Our method decomposes the energies/statistics into separately optimized terms corresponding to different perceptual dimensions. This not only simplifies computation, making it much faster than joint optimization [Hertzmann 2001] and MCMC sampling [Hurtut et al. 2009], but also enables flexible and friendly user customization.

3 Stroke Processes

For clarity, we describe our model and algorithm using a simplified stroke element model, which has a rectangular shape and the following attributes:

1. Position of the rectangle's center $\boldsymbol{p} = (x, y)$,
2. Orientation of its major axis θ,

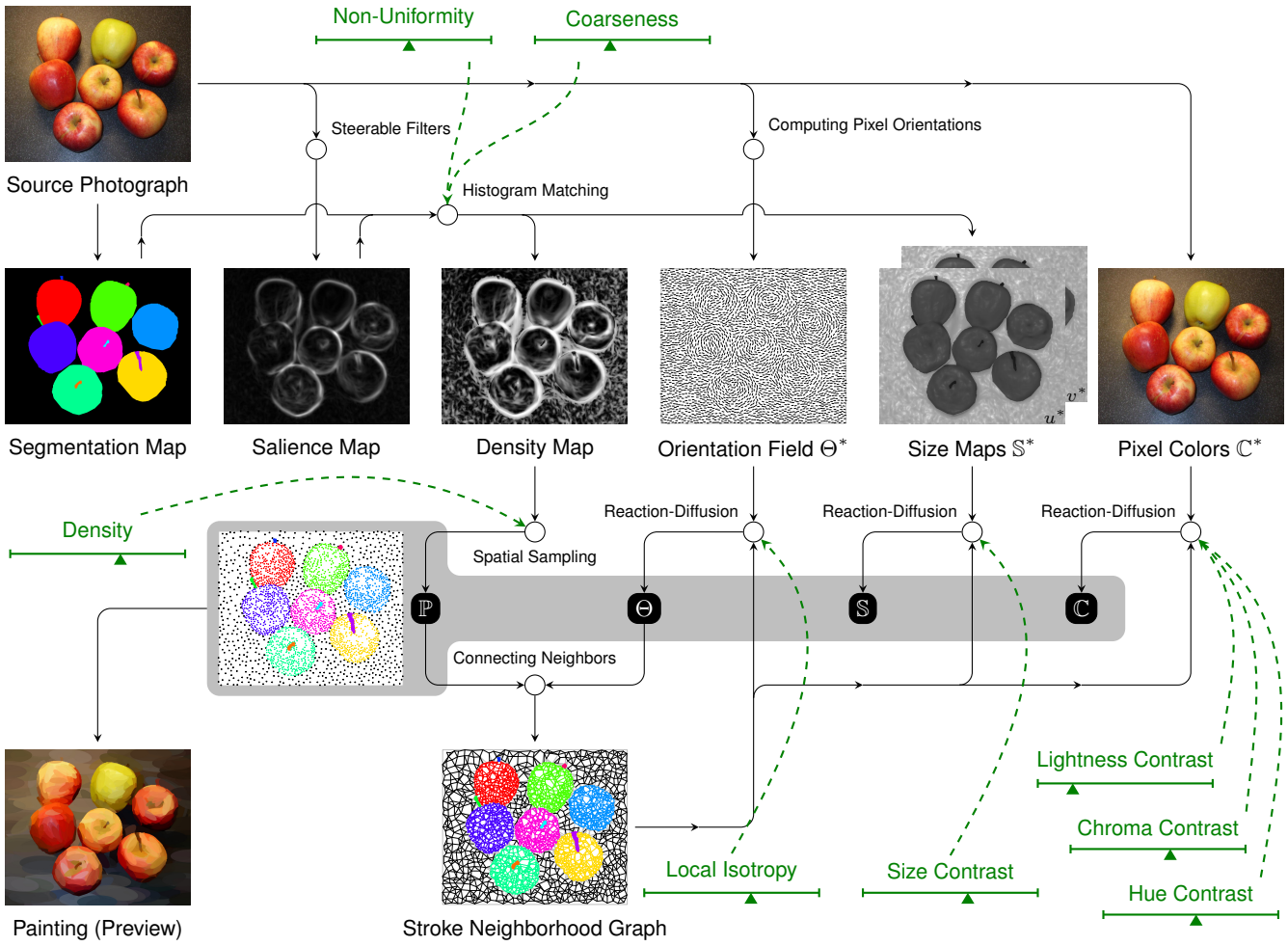

Figure 3: *The pipeline of our stroke processes. Green color and dashed arrows highlight the eight perceptual dimensions that users specify for each image region to the system (slidebars indicate their settings for the regions of apples). $\mathbb{P}$, Θ, $\mathbb{S}$ and $\mathbb{C}$ (black nodes in front of gray background) are the positions, orientations, sizes and colors of strokes to compute, respectively, with which we can render the final painting image or its fast preview. Gray segments in the stroke neighborhood graph (at the bottom) are connections between nodes in different image regions. Zoom to 800% to view details. Source photograph (top-left) courtesy of Evette Murphy @publicdomainpictures.net.*

3. Its size $\boldsymbol{s}$ (i.e., length u and width v), and
4. Its color $\boldsymbol{c}$ in the perceptually relevant CIELCH space, a cylindrical form of the perceptually uniform CIELAB/LUV spaces [Poynton 2002]. The three channels of $\boldsymbol{c}$ are lightness ℓ, chroma k and hue h, respectively.

For simplicity, we use only one of the four types of brush strokes from the dictionary [Zeng et al. 2009] (i.e., the textured type). This basic model can be extended with richer attributes within our framework, for example, multi-color strokes, curved strokes, etc. For an entire collection of M strokes to compose a painting, let $\mathbb{P} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_M)$, $\Theta = (\theta_1, \theta_2, \cdots, \theta_M)$, $\mathbb{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_M)$, and $\mathbb{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_M)$ denote their positions, orientations, sizes and colors, respectively.

We apply a two-level approach of rendering. In the upper level corresponding to the whole image, we adopt the interactive segmentation method used by Zeng et al. [2009] to paint the regions/objects using different parameters for different styles, and also to preserve sharp boundaries and layered effects. In the lower level corresponding to each region/object, instead of hard-coding rendering parameters according to image semantics as done by Zeng et al., we allow user customization by selectively adjusting eight slidebars on the software interface, which correspond to our summarized perceptual dimensions. In this way, styles can be flexibly controlled and even fine-tuned to depict subtle effects, as shown in Fig.1.

According to user customization, we compute the strokes using the following three-phase stroke processes:

I. A global *layout process* for stroke positions, according to the "density" and "non-uniformity" of strokes.

II. Building a *stroke neighborhood graph* to model the neighborhood relations among strokes. Each stroke is a node in the graph. The topology of the graph is not fixed. It keeps changing with stroke positions and orientations during user adjustment.

III. Three local *attribute processes* on the graph, for stroke orientations, sizes and colors, respectively, according to the latter six perceptual dimensions listed in Section 1.

This three-phase method essentially factorizes the stroke collection $(\mathbb{P}, \Theta, \mathbb{S}, \mathbb{C})$ into $\mathbb{P}$ and $(\Theta, \mathbb{S}, \mathbb{C}|\mathbb{P})$, where "|" stands for "given".

In our current implementation, the layout process for $\mathbb{P}$ is a non-stationary hard-core Poisson spatial point process [Stoyan et al. 1996] whose rate is determined by both image features and user customization, and the attribute processes for $(\Theta, \mathbb{S}, \mathbb{C}|\mathbb{P})$ are PDE-based stochastic reaction-diffusion processes [Turk 1991; Zhu and Mumford 1997] defined locally on the stroke neighborhood graph. Assuming the graph topology is determined by only $\mathbb{P}$ and Θ, and $\mathbb{S}$ and $\mathbb{C}$ are conditionally independent given the topology, we can further factorize $(\Theta, \mathbb{S}, \mathbb{C}|\mathbb{P})$ into $(\Theta|\mathbb{P})$, $(\mathbb{S}|\mathbb{P}, \Theta)$ and $(\mathbb{C}|\mathbb{P}, \Theta)$, and compute them separately to match their respective statistics. This is much easier than computing all attributes jointly. Fig.3 visualizes the pipeline of our stroke processes.

3.1 Layout Process for Stroke Positions

In the layout process, with stroke density and non-uniformity specified by user input, we compute the stroke positions in three steps:

1. Computing a salience map of the image by edge and ridge detection using steerable filters [Freeman and Adelson 1991; Collomosse and Hall 2002].

2. Generating a density map of strokes' spatial distribution on the image lattice. Assuming density is positively correlated with salience, we generate the former from the latter by performing a 1D *histogram matching* [Gonzalez and Woods 2002] versus a tail-truncated exponential distribution, whose rate is proportional to the specified non-uniformity. In this way, when the non-uniformity level increases, more pixels on the lattice will have very low probability masses, thus strokes tend to be more clustered around a few salient areas.

3. Sequentially sampling the given (by density) number of stroke positions according to the density map, each inhibiting (through rejection sampling) future strokes within a small radius (i.e., the hard core, inside which other strokes are not allowed to appear) determined by the minimum stroke size (empirically we use half of the minimum stroke width).

See Fig.3 for an example of these maps and sampled stroke positions corresponding to Fig.1a. Note that for the sampling, we use a non-uniform density map but a uniform inhibition radius across the image lattice, which is an exactly opposite design to the popular non-uniform Poisson-disk sampling method (cf. [Stoyan et al. 1996; Deussen et al. 2000; Vanderhaeghe et al. 2007; Gamito and Maddock 2009]). The main advantage of our method is that it can always approximate a full coverage of the canvas with enough strokes given that the inhibition circle is smaller than the minimum stroke size, while in Poisson-disk sampling, strokes with large inhibition radii must also have big enough sizes to cover their surrounding areas, making it less flexible to manipulate stroke sizes for customized styles.

3.2 Stroke Neighborhood Graph

In order to run the attribute processes to match stroke attributes to desired statistics, we construct a Markov stroke neighborhood graph, whose nodes are the strokes at sampled positions, and edges connecting each node with up to four neighbors. Inspired by Guo et al. [2003], we compute the neighborhood structure according to the distances between strokes and their orientations:

1. Initializing each stroke's orientation θ to its reference value θ^* in a reference orientation field Θ^* prepared in advance (e.g., an orientation field computed by diffusing segmentation boundaries and salient sketches [Zeng et al. 2009], or by RBF interpolation of the strongest gradients [Hays and Essa 2004]; we use the former, as visualized in Fig.3).

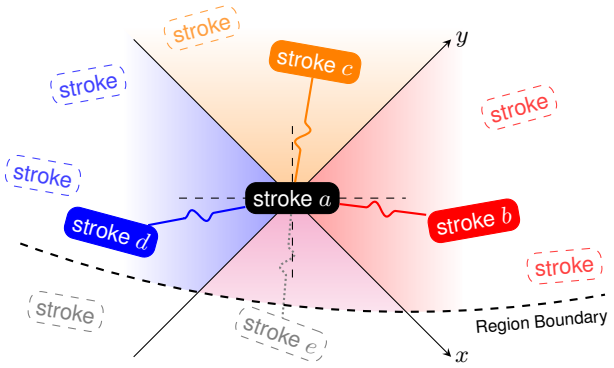

Figure 4: *Edge connections in the stroke neighborhood graph. A stroke's neighborhood includes its nearest neighbor in each of the four quadrants, if there exists one within the predefined distance threshold inside the image region. In this figure, the neighborhood of stroke a is a set $\mathcal{N}(a) = \{b, c, d\}$, and stroke e is excluded because it is not inside the same image region as a.*

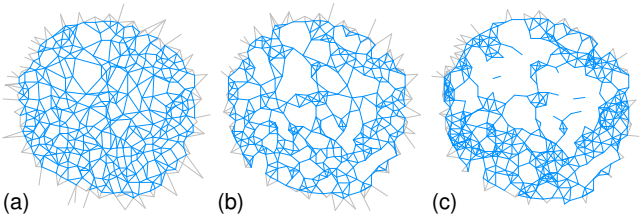

Figure 5: *A visual comparison of three designs of stroke neighborhood graph, for the rightmost apple in Fig.1a: (a) ours using anisotropic 4-nearest neighbors as shown in Fig.4, in which edges are more evenly distributed than in (b) using ordinary isotropic 4-nearest neighbors, and (c) using all neighbors within a fixed radius such that there are approximately the same number of edge connections as in (a) and (b). Gray segments around the boundary indicate connections to nodes in other image regions.*

2. Constructing local two-dimensional Cartesian coordinates as shown in Fig.4, whose origin is anchored at each stroke center, and the orthogonal straight lines $x \pm y = 0$ are aligned with the two axes of the stroke's rectangular area.

3. Connecting the four edges from the stroke to its nearest neighbor in each of the four quadrants. Nearest neighbors too far away (over a predefined distance threshold) are ignored, and strokes near region boundaries or image edges may not have neighbors in every quadrant (i.e., some neighbors may belong to other image regions thus excluded from the neighborhood), so we allow less than four neighbors in such cases (see. Fig.4).

As soon as the stroke orientations are finally computed in the next step, the structure of the stroke neighborhood graph should be updated with refreshed neighborhood connections before we compute the other attributes. An example stroke neighborhood graph is shown at the bottom of Fig.3. In this visualization, some nodes appear to have more than four connections due to our asymmetric neighborhood design, in which if stroke b belongs to the neighborhood of stroke a, i.e., $b \in \mathcal{N}(a)$, it does *not* imply the opposite statement $a \in \mathcal{N}(b)$, and the graph shows superposed neighborhood connections of both a and b.

We expect our anisotropic design of neighborhood structure to be better than those using either ordinary isotropic 4-nearest neighbors or all neighbors within a fixed radius, in the sense that neighbors are

usually more evenly distributed around each stroke, and the entire graph tends to be sparse yet less fragmented, especially when the "non-uniformity" level is high. Fig.5 displays a visual comparison of the three designs. Compared with Delaunay triangulation which can also generate nice meshes [de Berg et al. 2010], our design considers not only stroke positions but also their orientations in determining the graph structure, which we think makes better sense for the case of painting.

3.3 Attribute Processes for Stroke Orientations, Sizes and Colors

In the attribute processes, coarseness as well as contrasts in orientations, sizes and colors are involved. We use stochastic reaction-diffusion equations to perform the computation, in which the diffusion smooths the attributes among neighboring strokes to reduce the contrasts (or enhances the contrasts if we use negative diffusion rates, as explained below), and the reaction preserves information from the source image.

Stroke Orientations. We apply the stochastic reaction-diffusion equation

$$\frac{d\theta}{dt} = R(\theta) + \lambda_\theta D(\theta) + \epsilon_\theta \tag{1}$$

to propagate information across the stroke neighborhood graph to compute the orientations iteratively, in which ϵ_θ is a small stochastic noise added to each iteration to simulate natural randomness. Since θ is periodic over intervals of 2π, we adopt the *orientation diffusion* [Perona 1998] term

$$D(\theta) = \sum_n w_n \sin(\theta_n - \theta), \tag{2}$$

where θ_n are orientations of neighboring strokes of the one currently being updated, and w_n are weights inversely proportional to the spatial distances of these strokes. The *local reaction* term

$$R(\theta) = \sin(\theta^* - \theta) \tag{3}$$

applies the persistent *external force* from the reference orientation field Θ^*. The diffusion rate λ_θ is set to the level of local isotropy specified by user input. Specifically, when $\lambda_\theta > 0$, the diffusion term rotates each stroke's orientation towards those of its neighbors to make them similar, and thus leads to a smoothed style (like van Gogh's). When $\lambda_\theta < 0$, it turns diffusion into *concentration*, which rotates orientations of neighboring strokes away from each other, leading to a cluttered style. The process in Eq.(1), if without ϵ_θ, essentially optimizes the Markov field energy

$$E(\Theta|\mathbb{P};\Theta^*) = \sum_i \phi(\theta_i - \theta_i^*) + \lambda_\theta \sum_i \sum_{j\in\mathcal{N}(i)} w_{ij}\phi(\theta_i - \theta_j), \tag{4}$$

with the kernel $\phi(\cdot) = 1 - \cos(\cdot)$, and each weight w_{ij} inversely proportional to the spatial distance between neighboring strokes i and j. The energy is bounded above and below and converges to a local minimum during reaction-diffusion. With ϵ_θ we can no longer ensure the minimum, but fortunately it is unnecessary, and reaching somewhere close to the minimum is good enough for our purpose.

After a few iterations θ is close to convergence, and we update the edge connections of the stroke neighborhood graph using the computed stroke orientations. The reaction-diffusion and graph updating are both very fast (usually under 100ms and fast enough for interactive adjustment) since the number of strokes is usually much smaller than that of image pixels (the difference is around three orders in our experiments). For common images requiring less than 5000 strokes, 50–100 iterations can work well enough, and even linear search of neighbors has acceptable speed. Both processes can be parallelized on multi-core processors or graphics hardware for even better performance (we use OpenMP for C++).

Stroke Sizes. After the stroke neighborhood graph is updated, we move on to the separate processes for size and color. Similar to the process for orientation, but aperiodic this time, the two-component reaction-diffusion system for size, also involving a stochastic noise $\boldsymbol{\epsilon_s}$, is

$$\frac{d\boldsymbol{s}}{dt} = (\boldsymbol{s}^* - \boldsymbol{s}) + \lambda_{\boldsymbol{s}} \sum_n w_n(\boldsymbol{s}_n - \boldsymbol{s}) + \boldsymbol{\epsilon_s}, \tag{5}$$

subject to predefined ranges $u \in [u_{\min}, u_{\max}]$ and $v \in [v_{\min}, v_{\max}]$. This process corresponds to the quadratic-type Markov field energy

$$E(\mathbb{S}|\mathbb{P},\Theta;\mathbb{S}^*) = \sum_i \|\boldsymbol{s}_i - \boldsymbol{s}_i^*\|^2 + \lambda_{\boldsymbol{s}} \sum_i \sum_{j\in\mathcal{N}(i)} w_{ij}\|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2. \tag{6}$$

$\lambda_{\boldsymbol{s}}$ is inversely proportional to the specified size contrast (large $\lambda_{\boldsymbol{s}}$ leads to locally similar stroke sizes, and thus low size contrast). The coarseness is controlled by reference values $\boldsymbol{s}^* = (u^*, v^*)$ in precomputed maps. The reference size maps of u^* and v^* are generated from the salience map using histogram matching (in similar ways of generating the spatial distribution density map, as introduced in Section 3.1), but here the salience histogram is firstly reversed (since intuitively, large salience corresponds to smaller brush stroke), then matched versus two Laplacian distributions centered respectively at the desired length and width levels (given by coarseness) and truncated by the same predefined ranges of u and v as mentioned above.

Stroke Colors. The reaction-diffusion of stroke colors includes two parts, for the aperiodic lightness and chroma, and the periodic hue, respectively. We use pixel colors $\boldsymbol{c}^* = (\ell^*, k^*, h^*)$ in the source image as local reaction reference values of colors (similar to the usage of θ^* and $\boldsymbol{s}^*$), then the first part is done in a similar way to the case of size, namely,

$$\frac{d\ell}{dt} = (\ell^* - \ell) + \lambda_\ell \sum_n w_n(\ell_n - \ell) + \epsilon_\ell, \tag{7}$$

$$\frac{dk}{dt} = (k^* - k) + \lambda_k \sum_n w_n(k_n - k) + \epsilon_k, \tag{8}$$

and the second part is similar to orientation reaction-diffusion, namely,

$$\frac{dh}{dt} = \sin(h^* - h) + \lambda_h \sum_n w_n \sin(h_n - h) + \epsilon_h. \tag{9}$$

Coupling Among Attributes. In fact, it is usually unnecessary to restart from the very beginning when the user adjusts one or more perceptual dimensions. For example, if only "local isotropy" of orientations is adjusted, the stroke positions $\mathbb{P}$ do not need updating, and if only "hue contrast" is adjusted, nothing except the hue channel of $\mathbb{C}$ needs updating. Compared with some previous optimization-based systems which require complete recomputation once a single parameter is changed, this reduces computational cost significantly.

4 Experiments

We have conducted a large batch of experiments on our painterly rendering method. Fig.1 displays an example of the results, in which (a) is generated with lower lightness contrast than (b), and

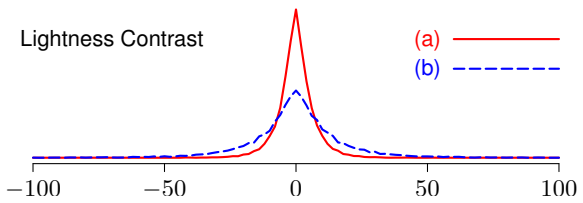

Figure 6: *Histograms of differences in lightness between neighboring strokes for Fig.1, which reflects the actual lightness contrasts we have achieved in the two paintings. The red solid curve corresponds to Fig.1a, and the blue dashed curve corresponds to Fig.1b. With a lower peak and heavier tails, obviously Fig.1b has a higher average difference, and thus a higher contrast in lightness.*

the same configuration of all other attributes. This causes (a) to appear a little more mellow than (b), especially for apples containing wider color ranges (e.g., the one at the bottom). As a quantitative confirmation, Fig.6 plots the histograms of differences in lightness between neighboring strokes for Fig.1, which reflects the actual lightness contrasts we have achieved in the two paintings. (If the contrast is low, neighboring strokes have similar lightness levels, thus most difference values should be close to zero, and the histogram will have a high peak and low tails.) With a lower peak and heavier tails, obviously Fig.1b has a higher average difference, and thus a higher contrast in lightness. Using our method, it is very easy to achieve such subtle difference in style.

Figs.8 through 11 show the flexibility of our method to simulate various styles by simply changing the levels in several perceptual dimensions. (Their figure captions are self-explanatory. Source photographs and reference orientation fields with segmentation are in Fig.7.) In Figs.8b, 9b and 9c we have applied different styles to different image regions. In Figs.8c and 9c, we have used negative λ_h's to apply concentration instead of diffusion, in order to obtain colors not existing in the original photograph. Levels in perceptual dimensions not specified in the captions are set to neutral (i.e., in the middle of the slidebar, corresponding to $\lambda = 0$). Histograms of differences in stroke attributes are also plotted, reflecting the corresponding contrast levels. In these demonstrations, we can see intuitive effects of different perceptual attributes.

We have also experimented curved brush strokes [Hertzmann 1998] with our method, which are more expressive than straight ones for depicting certain objects such as hairs. To build the stroke neighborhood graph upon a spline stroke element model, we simply use the coordinates and tangent at a spline's middle point as its position and orientation. Fig.12 displays a mixed use of straight and curved strokes with color contrasts (corresponding to the source photograph in Fig.7e). Fig.13 includes more results generated using our method.

Our method is friendly for interactive usage. The user only need to slide one or more of the eight bars corresponding to the eight perceptual parameters on the software interface. To help the user do adjustment, the system has real-time previews of the effects (rendered in lower resolution using color blocks as a simplified brush model for faster computation, as shown at the bottom-left of Fig.3). Once satisfied, the user can start the final rendering by clicking a button. In all, it usually takes 3–5 minutes to do adjustment for images with no more than 10–15 regions, and approximately 5–10 minutes to automatically render 2500–5000 textured brush strokes for a 3-million-pixel final painting image (the resolutions of rendered painting images in this paper are around 2000×1500) on a desktop computer (Intel Core 2 Duo E6600 2.4GHz CPU, 8GB DDR2 RAM).

5 Discussions

In this paper, we have presented a method to customize painterly rendering styles in a convenient way, in which users only need to specify desired levels in eight intuitive perceptual dimensions. Compared with previous methods, we emphasize the importance of contrasts as perceptual characteristics among strokes, and explicitly use them as rendering parameters. To take advantage of this design, we propose a novel stroke neighborhood graph model to represent the stroke pattern, and a fast algorithm based on reaction-diffusion for computation according to user specified contrast levels. Our experiments show that this method is able to simulate various styles under intuitive interactive control, and can generate some effects which are difficult to achieve in existing methods, especially when negative diffusion rates are used, as described in Section 3.3.

Perceptual Parameter Space. An intuitive way to improve our understanding of the perceptual dimensions is to analyze the space of these parameters by mapping the styles of some master artists and famous genres to corresponding sub-spaces.

Neighborhood Design. In our formulation of the stroke neighborhood graph, we assume the neighborhood connections depend on stroke positions and orientations, but not sizes, and we include the nearest neighbor in each of the four quadrants, instead of the isotropic 4-nearest neighbors or all strokes within a fixed radius. These empirical designs are preferred because of their balanced properties (see Fig.5) and satisfactory rendering quality. For improvement, sextants or octants can be used, but this requires heavier computation, and we have not seen quantitative justifications for more complex models. For curved strokes with spline backbones, a possibly better neighborhood design is to consider all control points instead of only center points. Neighborhood connections can be first established among these control points, upon which the stroke neighborhood graph can be constructed by connecting strokes with some of their control points connected.

Animation. Based on painterly rendering from still images, there are many studies on painterly animation from videos in recent literature [Meier 1996; Litwinowicz 1997; Hertzmann and Perlin 2000; Hays and Essa 2004; Collomosse et al. 2005; Lin et al. 2010; Kagaya et al. 2011; O'Donovan and Hertzmann 2011]. For this purpose, usually temporal coherence (i.e., smoothness) among strokes is emphasized, including their positions and appearance attributes. Our method may help improve the coherence by adding a temporal domain to the spatial neighborhood graph, and perform reaction-diffusion on this 3D graph. Of course, different or even opposite diffusion rates can be used for the spatial and temporal connections, in order to achieve temporal coherence while preserving sharp spatial contrasts.

Project Website

More results rendered using our stroke processes are available at *http://www.stat.ucla.edu/∼mtzhao/research/stroke-processes/*.

Acknowledgments

We are grateful to Yaling Yang and Xiaolan Ye for their assistance in the experiments. We thank Wenze Hu, Amy Morrow, Brandon Rothrock, Zhangzhang Si, Benjamin Yao, Yibiao Zhao, and the anonymous reviewers for nice suggestions on improving the presentation of this paper. The work at UCLA was supported by an ONR MURI grant N000141010933 and an NSF IIS grant 1018751, and the work at LHI was supported by two NSFC grants 60832004 and 90920009.

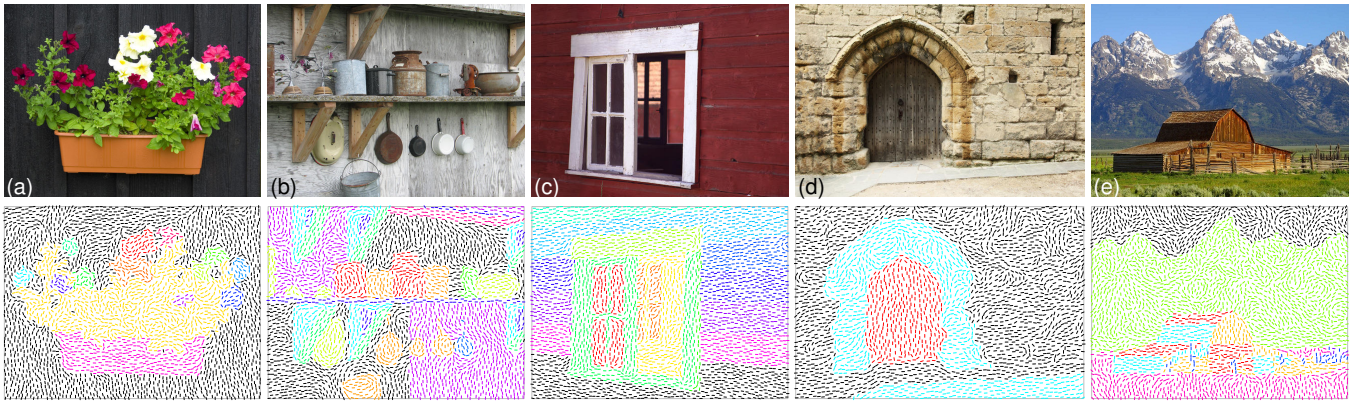

Figure 7: *Source photographs and their reference orientation fields with segmentation visualized using colors, corresponding to the paintings in Figs.8 through 12: (a) Petunias in Pot (courtesy of Petr Kratochvil @publicdomainpictures.net), (b) Vintage Rusty Stuff (courtesy of Kim Newberg @publicdomainpictures.net), (c) Barn Window (courtesy of Jon Sullivan @public-domain-photos.com), (d) Old Castle Gate (courtesy of Petr Kratochvil @publicdomainpictures.net), and (e) Grand Teton Barn (courtesy of pdphoto.org). Zoom to 400% to view details.*

Figure 8: *Painterly rendering close-ups of three different styles corresponding to the source photograph in Fig.7a: (a) neutral, (b) high size contrast (for the leaves) and low local isotropy (for background), and (c) high hue contrast. For the leaves, histograms of differences in size and hue between neighboring strokes are plotted on the right side. Zoom to 400% to view details.*

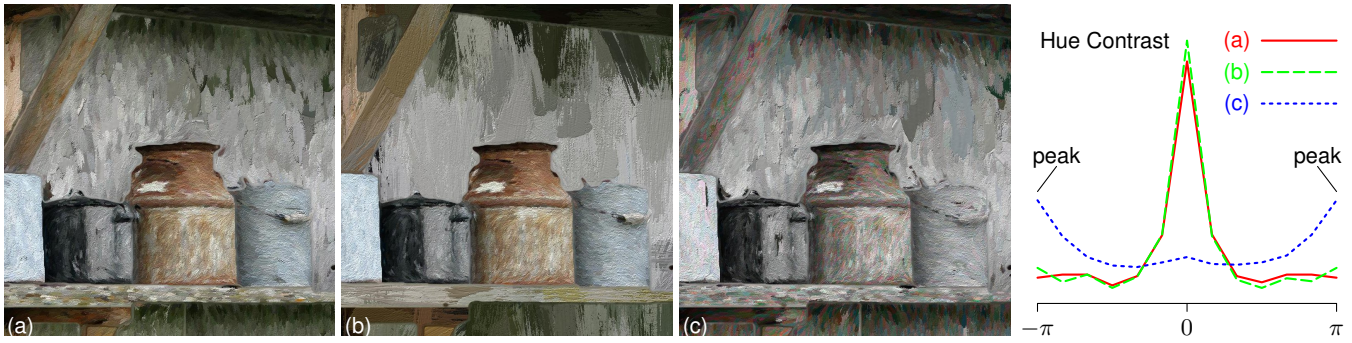

Figure 9: *Painterly rendering close-ups of three different styles corresponding to the source photograph in Fig.7b: (a) neutral, (b) low density and high coarseness (for the wall), and (c) high size contrast and high hue contrast (for the wall, the shelf, and the can in the middle). For the wall, histograms of differences in hue between neighboring strokes are plotted on the right side. In the blue dotted curve for (c), the peaks near $\pm\pi$ (actually they are the same peak since hue is periodic over intervals of 2π) indicate the effect of "complementary colors in neighboring strokes" as we mentioned in Section 2. Zoom to 400% to view details.*

Figure 10: *Painterly rendering close-ups of three different styles corresponding to the source photograph in Fig.7c: (a) neutral, (b) high lightness contrast, and (c) low local isotropy. Histograms of differences in orientation and lightness between neighboring strokes are plotted on the right side. Zoom to 400% to view details.*

Figure 11: *Painterly rendering close-ups of three different styles corresponding to the source photograph in Fig.7d: (a) neutral, (b) high lightness contrast, and (c) high chroma contrast. Histograms of differences in lightness and chroma between neighboring strokes are plotted on the right side. Zoom to 400% to view details.*

Figure 12: *Painterly renderings of two different styles corresponding to the source photograph in Fig.7e: (a) curved strokes for the barn and high lightness contrast, and (b) curved strokes for everything except the sky and low chroma contrast. Zoom to 400% to view details.*

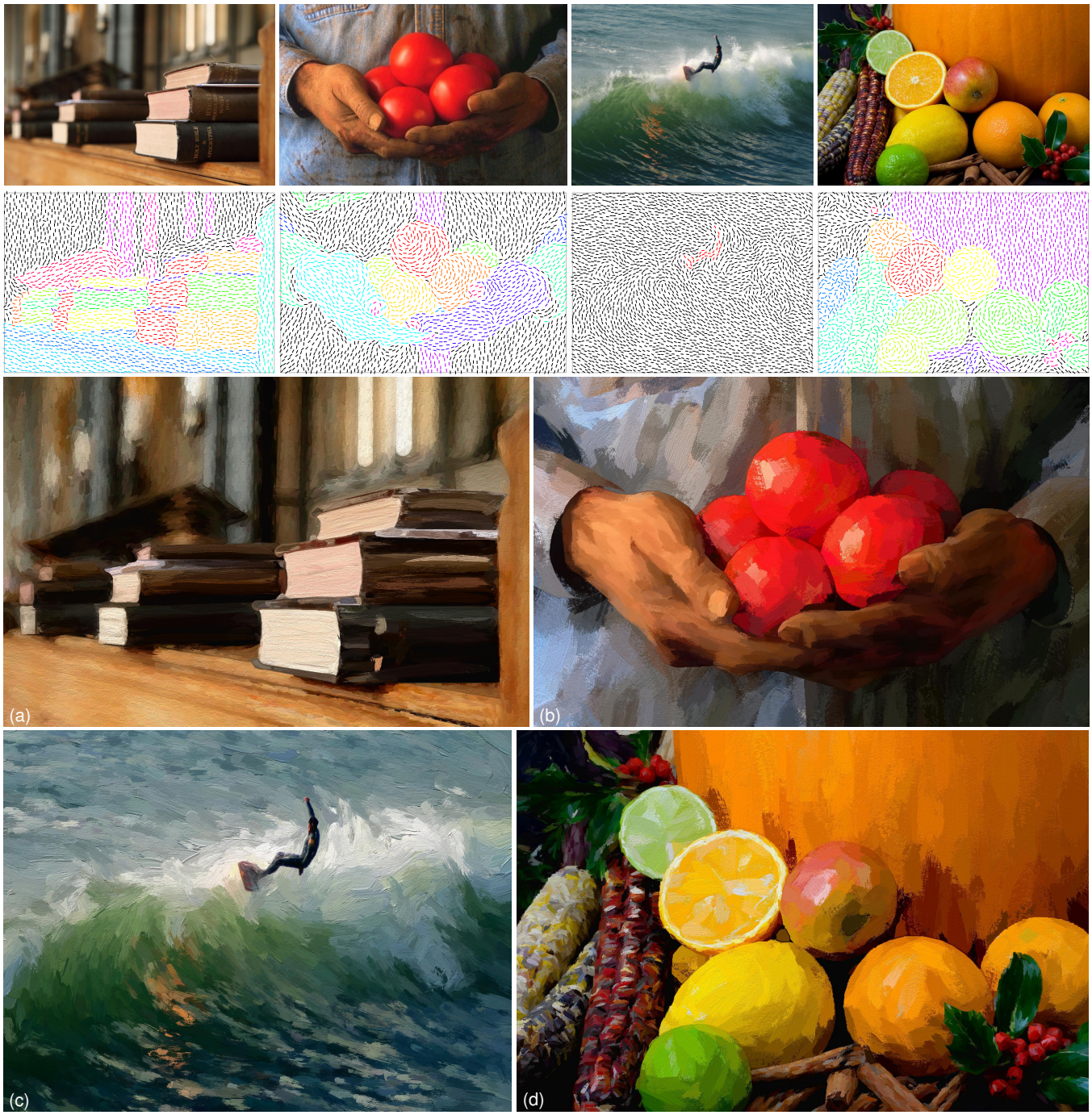

Figure 13: *Four photographs (top row from left to right: books in church, courtesy of Petr Kratochvil @publicdomainpictures.net; hands holding tomatoes, courtesy of Hertzmann [1998, Fig.2a]; longboard surfer on the wave crest, courtesy of Andrew Schmidt @publicdomainpictures.net; and autumn fruits, courtesy of Petr Kratochvil @publicdomainpictures.net), their reference orientation fields with segmentation visualized using colors (second row), and their painterly renderings using our method. Zoom to 400% to view details.*

References

BAXTER, W. V. 2004. *Physically-Based Modeling Techniques for Interactive Digital Painting*. PhD thesis, University of North Carolina at Chapel Hill.

CHU, N., BAXTER, W., WEI, L.-Y., AND GOVINDARAJU, N. 2010. Detail-preserving paint modeling for 3D brushes. In *Proc. NPAR '10*, 27–34.

COCKSHOTT, T., PATTERSON, J., AND ENGLAND, D. 1992. Modelling the texture of paint. *Comput. Graphics Forum 11*, 3, 217–226.

COLLOMOSSE, J. P., AND HALL, P. M. 2002. Painterly rendering using image salience. In *Proc. EGUK '02*, 122–128.

COLLOMOSSE, J. P., ROWNTREE, D., AND HALL, P. M. 2005. Stroke surfaces: temporally coherent artistic animations from video. *IEEE Trans. Visual Comput. Graphics 11*, 5, 540–549.

COOKE, H. L. 1978. *Painting Techniques of the Masters*. Watson Guptill/Pitman Publishing.

DE BERG, M., CHEONG, O., VAN KREVELD, M., AND OVERMARS, M. 2010. *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer.

DEUSSEN, O., HILLER, S., VAN OVERVELD, C., AND STROTHOTTE, T. 2000. Floating points: A method for computing stipple drawings. *Comput. Graphics Forum 19*, 3, 40–51.

FREEMAN, W. T., AND ADELSON, E. H. 1991. The design and use of steerable filters. *IEEE Trans. Patten Anal. Mach. Intell. 13*, 9, 891–906.

GAMITO, M. N., AND MADDOCK, S. C. 2009. Accurate multidimensional poisson-disk sampling. *ACM Trans. Graph. 29*, 1, 8:1–8:19.

GONZALEZ, R. C., AND WOODS, R. E. 2002. *Digital Image Processing*, 2nd ed. Prentice Hall.

GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A K Peters/CRC Press.

GOOCH, B., COOMBE, G., AND SHIRLEY, P. 2002. Artistic vision: painterly rendering using computer vision techniques. In *Proc. NPAR '02*, 83–91.

GUO, C.-E., ZHU, S.-C., AND WU, Y. N. 2003. Modeling visual patterns by integrating descriptive and generative methods. *Int. J. Comput. Vision 53*, 1, 5–29.

HAEBERLI, P. 1990. Paint by numbers: abstract image representations. In *Proc. SIGGRAPH '90*, 207–214.

HAYS, J., AND ESSA, I. 2004. Image and video based painterly animation. In *Proc. NPAR '04*, 113–120.

HERTZMANN, A., AND PERLIN, K. 2000. Painterly rendering for video and interaction. In *Proc. NPAR '00*, 7–12.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH '98*, 453–460.

HERTZMANN, A. 2001. Paint by relaxation. In *Proc. CGI '01*, 47–54.

HERTZMANN, A. 2002. Fast paint texture. In *Proc. NPAR '02*, 91–96.

HERTZMANN, A. 2003. A survey of stroke-based rendering. *IEEE Comput. Graph. Appl. 23*, 4, 70–81.

HUGHES, J. M., GRAHAM, D. J., AND ROCKMORE, D. N. 2010. Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder. *PNAS 107*, 4, 1279–1283.

HURTUT, T., LANDES, P.-E., THOLLOT, J., GOUSSEAU, Y., DROUILLHET, R., AND COEURJOLLY, J.-F. 2009. Appearance-guided synthesis of element arrangements by example. In *Proc. NPAR '09*, 51–60.

KAGAYA, M., BRENDEL, W., DENG, Q., KESTERSON, T., TODOROVIC, S., NEILL, P., AND ZHANG, E. 2011. Video painting with space-time-varying style parameters. *IEEE Trans. Visual Comput. Graphics 17*, 1, 74–87.

LIN, L., ZENG, K., LV, H., WANG, Y., XU, Y., AND ZHU, S.-C. 2010. Painterly animation using video semantics and feature correspondence. In *Proc. NPAR '10*, 73–80.

LITWINOWICZ, P. 1997. Processing images and video for an impressionist effect. In *Proc. SIGGRAPH '97*, 407–414.

LU, J., SANDER, P. V., AND FINKELSTEIN, A. 2010. Interactive painterly stylization of images, videos and 3D animations. In *Proc. I3D '10*, 127–134.

MEIER, B. J. 1996. Painterly rendering for animation. In *Proc. SIGGRAPH '96*, 477–484.

O'DONOVAN, P., AND HERTZMANN, A. 2011. Anipaint: Interactive painterly animation from video. *IEEE Trans. Visual Comput. Graphics*, PrePrints.

PERONA, P. 1998. Orientation diffusions. *IEEE Trans. Image Proces. 7*, 3, 457–467.

POYNTON, C. 2002. *Digital Video and HDTV: Algorithms and Interfaces*, 1st ed. Morgan Kaufmann.

STOYAN, D., KENDALL, W. S., AND MECKE, J. 1996. *Stochastic Geometry and Its Applications*, 2nd ed. Wiley.

STRASSMANN, S. 1986. Hairy brushes. In *Proc. SIGGRAPH '86*, 225–232.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann.

TURK, G., AND BANKS, D. 1996. Image-guided streamline placement. In *Proc. SIGGRAPH '96*, 453–460.

TURK, G. 1991. Generating textures on arbitrary surfaces using reaction-diffusion. In *Proc. SIGGRAPH '91*, 289–298.

VANDERHAEGHE, D., BARLA, P., THOLLOT, J., AND SILLION, F. 2007. Dynamic point distribution for stroke-based rendering. In *Proc. EGSR '07*, 139–146.

WALLRAVEN, C., FLEMING, R., CUNNINGHAM, D., RIGAU, J., FEIXAS, M., AND SBERT, M. 2009. Categorizing art: Comparing humans and computers. *Comput. Graph. 33*, 4, 484–495.

ZENG, K., ZHAO, M., XIONG, C., AND ZHU, S.-C. 2009. From image parsing to painterly rendering. *ACM Trans. Graph. 29*, 1, 2:1–2:11.

ZHAO, M., AND ZHU, S.-C. 2010. Sisley the abstract painter. In *Proc. NPAR '10*, 99–107.

ZHU, S. C., AND MUMFORD, D. 1997. Prior learning and Gibbs reaction-diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 11, 1236–1250.