From Image Parsing to Painterly Rendering

KUN ZENG
Lotus Hill Institute
MINGTIAN ZHAO
Lotus Hill Institute and University of California, Los Angeles
CAIMING XIONG
Lotus Hill Institute
and
SONG-CHUN ZHU
Lotus Hill Institute and University of California, Los Angeles

We present a semantics-driven approach for stroke-based painterly rendering, based on recent image parsing techniques [Tu et al. 2005; Tu and Zhu 2006] in computer vision. Image parsing integrates segmentation for regions, sketching for curves, and recognition for object categories. In an interactive manner, we decompose an input image into a hierarchy of its constituent components in a parse tree representation with occlusion relations among the nodes in the tree. To paint the image, we build a brush dictionary containing a large set (760) of brush examples of four shape/appearance categories, which are collected from professional artists, then we select appropriate brushes from the dictionary and place them on the canvas guided by the image semantics included in the parse tree, with each image component and layer painted in various styles. During this process, the scene and object categories also determine the color blending and shading strategies for inhomogeneous synthesis of image details. Compared with previous methods, this approach benefits from richer meaningful image semantic information, which leads to better simulation of painting techniques of artists using the high-quality brush dictionary. We have tested our approach on a large number (hundreds) of images and it produced satisfactory painterly effects.

Categories and Subject Descriptors: I.3.4 [**Computer Graphics**]: Graphics Utilities—*Paint systems*; I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Hierarchical*; J.5. [**Computer Applications**]: Arts and Humanities—*Fine arts*

General Terms: Algorithms, Human Factors

Additional Key Words and Phrases: Image parsing, nonphotorealistic rendering, orientation field, painterly rendering, primal sketch

ACM Reference Format:

Zeng, K., Zhao, M., Xiong, C., and Zhu, S.-C. 2009. From image parsing to painterly rendering. ACM Trans. Graph. 29, 1, Article 2 (December 2009), 11 pages. DOI = 10.1145/1640443.1640445 http://doi.acm.org/10.1145/1640443.1640445

1. INTRODUCTION

In recent years, *NonPhotorealistic Rendering* (NPR) [Gooch and Gooch 2001; Strothotte and Schlechtweg 2002] and its relevant areas have been attracting growing interest. As one of the major topics of NPR research, *painterly rendering*, especially *Stroke-Based Rendering* (SBR) [Hertzmann 2003] techniques have achieved remarkable success. In general, SBR tries to synthesize nonphotorealistic images by placing and blending strokes of certain visual styles such as stipples or painterly brushes. Towards the solutions of SBR, we are faced with two main tasks:

(1) the modeling and manipulation of brushes, including both shape and appearance factors;
(2) the selection and placement of brush strokes for the result image.

For the first task, previously proposed models can be roughly categorized into two main streams, namely, the physically-based/

K. Zeng and M. Zhao contributed equally to this work and are placed in alphabetical order.
The work at UCLA was supported by an NSF grant on image parsing IIS-0713652, and the work at Lotus Hill Institute was supported by two Chinese National 863 grants 2007AA01Z340 and 2008AA01Z126, and a National Natural Science Foundation of China (NSFC) grant 60672162.
Authors' addresses: K. Zeng, Lotus Hill Institute, City of Ezhou, Hubei Province, China; M. Zhao, Department of Statistics, University of California, Los Angeles, CA 90095-1554; email: mtzhao@stat.ucla.edu; C. Xiong, Lotus Hill Institute, City of Ezhou, Hubei Province, China; S.-C. Zhu, Department of Statistics, University of California, Los Angeles, CA 90095-1554; email: sczhu@stat.ucla.edu.

DOI 10.1145/1640443.1640445 http://doi.acm.org/10.1145/1640443.1640445

motivated models and the image-example-based models. The former stream is supposed to simulate the physical processes involved in stroke drawing or painting, including the models of stroke elements, media, etc. Among this stream, representative works include the *hairy brushes* model proposed by Strassmann [1986] and a graphite pencil and paper model for rendering 3D polygonal geometries studied in Sousa and Buchanan [1999]. Curtis et al. [1997] simulated various artistic effects of watercolor based on shallow-water fluid dynamics. Chu and Tai [2005] developed a real-time system for simulating ink dispersion in absorbent paper for art creation purposes. At the same time, in order to avoid the great computational and manipulative complexity of physically-based methods, image-example-based models are adopted in a few SBR solutions [Litwinowicz 1997; Hertzmann 1998], which usually do not have explicit brush categories or design principles to account for the various types of brush strokes used by artists.

Plenty of work has also been carried out for the second task. A system for 3D NPR was developed in Teece [1998] where users can place strokes interactively on the surfaces of 3D object models. Once the strokes are attached to a geometric model, they can be subsequently replayed from various viewpoints, thus becoming animations in painting styles. Besides, efforts to automatic stroke placement are devoted in two main directions [Hertzmann 2003], namely, the greedy methods and the optimization methods. The greedy algorithms try to place the strokes to match specific targets in every single step [Litwinowicz 1997; Hertzmann 1998], while the optimization algorithms iteratively place and adjust strokes to minimize or maximize certain objective energy functions [Turk and Banks 1996].

Despite the acknowledged success, the current SBR methods, and NPR in general, typically lack semantic descriptions of the scenes and objects to be rendered, while semantics actually play a central role in most drawing and painting tasks as commonly depicted by artists and perceived by audiences [Funch 1997]. Without image semantics, these rendering algorithms capturing only low-level image characteristics, such as colors and textures, are doomed to failure in well simulating the usually greatly flexible and object-oriented techniques of painting. To address this problem, we present a semantics-driven approach for SBR, based on recent image parsing technical advances in computer vision [Tu et al. 2005; Tu and Zhu 2006]. Figure 1 shows the system flowchart of our approach with an example input image and its corresponding final rendering result.

In the rendering flowchart, the input image first goes through a *hierarchical image parsing* phase. As Figure 2(a) illustrates, image parsing decomposes an input image into a coarse-to-fine hierarchy of its constituent components in a *parse tree* representation, and the nodes in the parse tree correspond to a wide variety of visual patterns in the image, including:

(1) generic texture regions for sky, water, grass, land, etc.;

(2) curves for line or threadlike structures, such as tree twigs, railings, etc.;

(3) objects for hair, skin, face, clothes, etc.

We use 18 common object categories in this article.

The nodes in the parse tree are organized with partially ordered occlusion relations which yield a layered representation as shown in Figure 2(b). In our system, the parse tree is extracted in an interactive manner.

Corresponding to their semantics, the diverse visual patterns in the parse tree ought to be painted with different types of brushes, for example, human faces should commonly be painted more carefully using relatively smaller brushes compared with intricate twigs and leaves (see Figure 1). For this purpose, we build a brush dictionary containing a large set (760) of brush examples with varying shapes and texture appearances, which are collected from professional artists. These brushes are aimed at reflecting the material properties and feelings in several perceptual dimensions or attributes, for example, dry versus wet, hard versus soft, and long versus short, as well as four shape and appearance categories (point, curve, block, and texture). These attributes of the brushes are further augmented by four additional attributes (color, opacity map, height map, and backbone geometry), and they are mapped, probabilistically, to the attributes of the visual patterns in the parse tree. Thus the selection of the brushes from the dictionary is guided by the semantics included in the parse tree, with each component and layer painted in various styles.

For each image component, we run the primal sketch algorithm [Guo et al. 2007] to compute a vectorized sketch graph as cues of pixel orientations within the image, and then generate an orientation field through anisotropic diffusion [Perona 1998; Chen and Zhu 2006], as shown in Figure 3. After that, the placement of brush strokes is guided by the shapes of the image components and the orientation field with a greedy algorithm. The scene and object categories also determine the color blending and shading strategies, thus achieve inhomogeneous synthesis for rich image details. In addition, an optional color enhancement process also based on image semantics is provided for more appealing rendering effects.

The main contributions of this article are twofold. First, we introduce rich image semantics to drive painterly rendering algorithms, which leads to better simulation of painting techniques of artists. Second, we build a high-quality image-example-based brush dictionary, which enables vivid synthesis of nice painterly effects. The method described in the article has achieved satisfactory results over hundreds of testing images. Figure 1 includes one of our representative painterly rendering results with diverse painting techniques applied throughout the image. In addition, Figures 8 to 11 show more SBR results generated using our approach.

The rest article of this is planned as follows. Section 2 introduces the formulation and computation of the image parsing process. Section 3 explains our brush dictionary as well as the stroke placement and rendering algorithms. Section 4 displays some examples of our painterly rendering results, and Section 5 includes a brief discussion of possible future improvements.

2. INTERACTIVE IMAGE PARSING

Image parsing refers to the task of decomposing an image into its constituent visual patterns in a coarse-to-fine parse tree representation [Tu et al. 2005; Tu and Zhu 2006]. It integrates image segmentation for generic regions, sketching for curves and curve groups, and recognition for object categories. We develop a software interface to obtain interactive instructions from users for reliable parsing results.

2.1 Hierarchical Decomposition and Recognition

Figure 2(a) shows an example of hierarchical image parsing. The whole scene is first divided into two parts: two people in the foreground and the outdoor environment in the background. In the second level, the two parts are further subdivided into face/skin, clothes, trees, road/building, etc. Continuing with lower levels, these patterns are decomposed recursively until a certain resolution limit is reached. That is, certain leaf nodes in the parse become unrecognizable without the surrounding context, or insignificant for specific tasks.

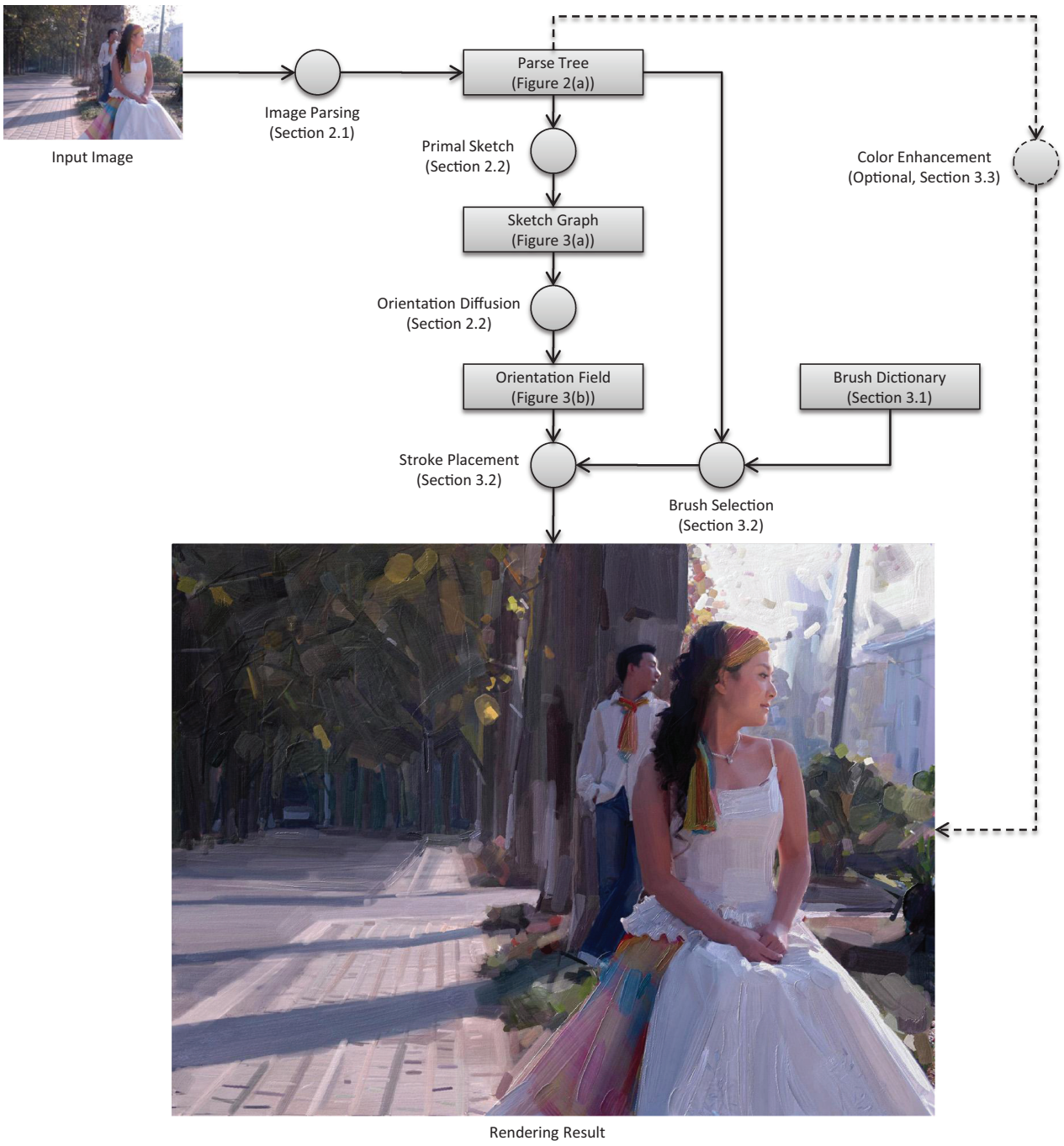


Fig. 1. The flowchart of our painterly rendering system based on image parsing. With the extracted semantic information, the input image is painterly rendered with its constituent components depicted in various styles.

Given an input image, we denote by W the parse tree for the semantic description of the scene, and

$$\mathcal{R} = \{R_k : i = 1, 2, \cdots, K\} \subset W \tag{1}$$

is the set of the K leaf nodes of W, representing the generic regions, curves, and objects in the image. Each leaf node R_k is a 3-tuple

$$R_k = \langle \Lambda_k, \ell_k, \mathcal{A}_k \rangle, \tag{2}$$

where Λ_k is the image domain (a set of pixels) covered by R_k, and ℓ_k and $\mathcal{A}_k$ are its label (for object category) and appearance model, respectively. Let Λ be the domain of the whole image lattice, then

$$\Lambda = \Lambda_1 \cup \Lambda_2 \cup \cdots \cup \Lambda_K \tag{3}$$

in which we do not demand $\Lambda_i \cap \Lambda_j = \emptyset$ for all $i \neq j$ since two nodes are allowed to overlap with each other.

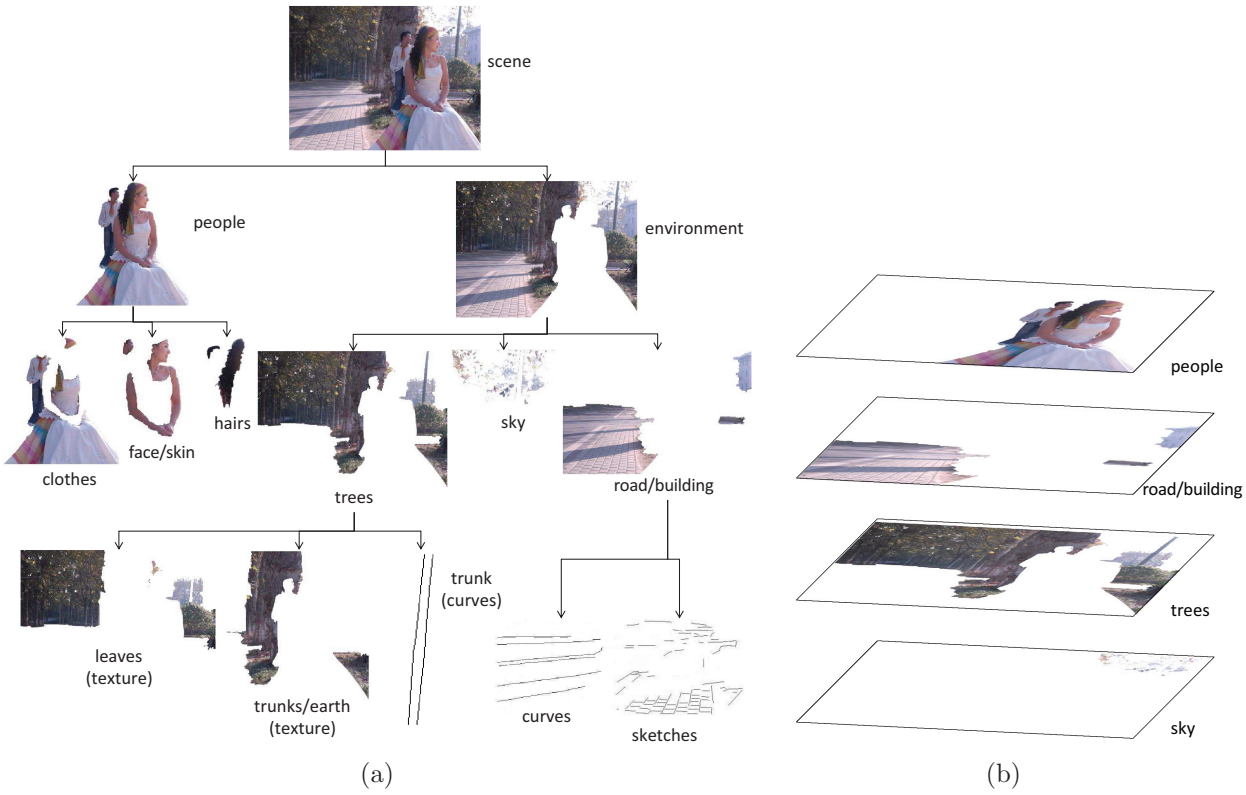


Fig. 2. An illustration of image parsing. (a) The parse tree representation of an image. (b) The occlusion relation between nodes in the parse tree yields a partial order and thus a layered representation.

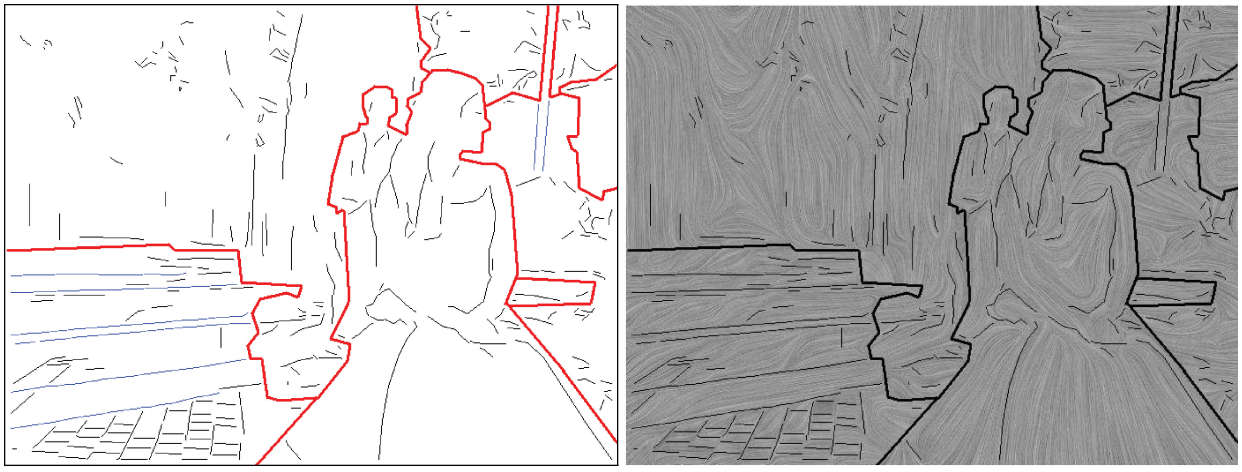


Fig. 3. The layered sketch graph and its corresponding orientation field generated following the example in Figures 1 and 2. In (a), red, blue, and black lines stand for the region boundaries, major curves, and other sketches, respectively. For clarity, the orientation field visualized in (b) is computed without the Gaussian prior energy, which may submerge other factors in some areas (see Section 2.2).

The leaf nodes $\mathcal{R}$ can be obtained with a segmentation and recognition (object classification) process, and assigned to different depths (distances from the camera) to form a layered representation of the scene structure of the image. We use a three-stage, interactive process to acquire the information.

(1) The image is segmented into a few regions by the graph-cut algorithm [Boykov and Jolly 2001] in a real-time interactive manner using foreground and background scribbles [Li et al. 2004] on superpixels generated by mean-shift clustering [Comaniciu and Meer 2002].

(2) The regions are classified by a bag-of-words classifier [Li et al. 2005] with 18 object categories which are common in natural images.

face/skin	hair	cloth	sky/cloud
water surface	spindrift	mountain	road/building
rock	earth	wood/plastic	metal
flower/fruit	grass	leaf	trunk/twig
background	other		

Features including SIFT [Lowe 1999], colors, and region geometries are used for the classification. In case of imperfect recognitions which usually happen, users can correct the category labels through the software interface by selecting from a list of all the labels.

(3) The regions are assigned to layers of different depths, by maximizing the probability of a partially ordered sequence

$$S : R_{(1)} \preceq R_{(2)} \preceq \cdots \preceq R_{(K)} \tag{4}$$

for region $R_{(1)}$ in the same or closer layers of $R_{(2)}$ and so on, which is a permutation of

$$R_1 \preceq R_2 \preceq \cdots \preceq R_K. \tag{5}$$

By assuming all events $R_{(k)} \preceq R_{(k+1)}, k = 1, 2, \cdots, K-1$ are independent, we have an empirical solution

$$\begin{aligned} S^* &= \arg\max_S p(R_{(1)} \preceq R_{(2)}, R_{(2)} \preceq R_{(3)}, \cdots, R_{(K-1)} \preceq R_{(K)}) \\ &= \arg\max_S \prod_{k=1}^{K-1} p(R_{(k)} \preceq R_{(k+1)}) \end{aligned} \tag{6}$$

in which $p(R_{(k)} \preceq R_{(k+1)})$ can be approximated with

$$p(R_{(k)} \preceq R_{(k+1)}) \approx \tilde{f}(R_i \preceq R_j | \ell_i = \ell_{(k)}, \ell_j = \ell_{(k+1)}), \tag{7}$$

where $\tilde{f}$ returns the frequencies of occlusions between different object categories according to previously annotated observations in the LHI image database [Yao et al. 2007]. Note that the independence assumption should fail if $p(S|R_{(1)} \preceq R_{(2)}, R_{(2)} \preceq R_{(3)}, \cdots, R_{(K-1)} \preceq R_{(K)}) = 1$, but we still expect S^* to approximate the mode of $p(S)$. Once S^* is obtained, users can also correct it by swapping pairs of regions through the software interface, and can further compress the sequence to limit the total number of layers, by combining the pairs of $R_{(k)}$ and $R_{(k+1)}$ with relatively low $p(R_{(k)} \preceq R_{(k+1)})$, as shown in Figure 2(b).

2.2 Primal Sketch and Orientation Field

For each leaf node (except curves) in the parse tree, we run the primal sketch algorithm [Guo et al. 2007] to generate a sketch graph and the orientation diffusion algorithm [Perona 1998; Chen and Zhu 2006] for an orientation field.

The concept of primal sketch dates back to David Marr, who conjectured the idea as a symbolic or token representation in terms of image primitives, to summarize the early visual processing [Marr 1982]. A mathematical model of primal sketch was later presented in Guo et al. [2007] which integrates structures and textures.

Given the domain Λ_k of a leaf node R_k, the primal sketch model further subdivides it into two parts: a sketchable part Λ_k^{sk} for salient structures (perceivable line segments) and a nonsketchable part Λ_k^{nsk} for stochastic textures without distinguishable structures, and

$$\Lambda_k = \Lambda_k^{\text{sk}} \cup \Lambda_k^{\text{nsk}}, \quad \Lambda_k^{\text{sk}} \cap \Lambda_k^{\text{nsk}} = \emptyset. \tag{8}$$

The primitives in the sketchable part Λ_k^{sk} provide major pixel orientation information of the image, as shown in Figure 3(a). Using the orientation data of sketchable pixels, we compute an orientation field on Λ_k using a diffusion routine which minimizes an energy function derived within the Markov Random Field (MRF) framework with pair cliques in a 3-layer neighborhood system.

An orientation field $\mathbf{\Theta}_k$ of R_k, defined on Λ_k, is the set of orientations at every pixel $s \in \Lambda_k$

$$\mathbf{\Theta}_k = \{\theta(s) : \theta(s) \in [0, \pi), s \in \Lambda_k\} \tag{9}$$

in which each orientation $\theta(s)$ depends on its neighbors in three layers:

(1) the same pixel s in the initial orientation field

$$\mathbf{\Theta}_k^{\text{sk}} = \{\theta(s) : \theta(s) \in [0, \pi), s \in \Lambda_k^{\text{sk}}\} \tag{10}$$

covering all sketchable pixels of R_k;

(2) the adjacent pixels ∂s of s on the 4-neighborhood stencil of the orientation field $\mathbf{\Theta}_k$;

(3) the same pixel s in the prior orientation field

$$\mathbf{\Theta}_k^{\text{pri}} = \{\theta(s) : \theta(s) \sim G(\mu_k, \sigma_k^2, a_k, b_k), s \in \Lambda_k\} \tag{11}$$

of R_k, in which $G(\mu_k, \sigma_k^2, a_k, b_k)$ is a truncated Gaussian distribution whose parameters depend on the properties of R_k.

Corresponding to the constraints of the three layers, the energy function of the orientation field is defined as

$$E(\mathbf{\Theta}_k) = E_{\text{sk}}(\mathbf{\Theta}_k) + \alpha E_{\text{sm}}(\mathbf{\Theta}_k) + \beta E_{\text{pri}}(\mathbf{\Theta}_k) \tag{12}$$

in which $E_{\text{sk}}(\mathbf{\Theta}_k)$, $E_{\text{sm}}(\mathbf{\Theta}_k)$, and $E_{\text{pri}}(\mathbf{\Theta}_k)$ are terms for the aforementioned three layers, respectively, and α and β are weight parameters assigned by the user. The first term

$$E_{\text{sk}}(\mathbf{\Theta}_k) = \sum_{s \in \Lambda_k^{\text{sk}}} d(\mathbf{\Theta}_k(s), \mathbf{\Theta}_k^{\text{sk}}(s)) \boldsymbol{\rho}_k^{\text{sk}}(s) \tag{13}$$

measures the similarity of $\mathbf{\Theta}_k$ and $\mathbf{\Theta}_k^{\text{sk}}$ at sketchable pixels, in which the weight map

$$\boldsymbol{\rho}_k^{\text{sk}} = \{\rho(s) : \rho(s) = \nabla_{\perp \mathbf{\Theta}_k^{\text{sk}}} \boldsymbol{I}_{\Lambda_k^{\text{sk}}}, s \in \Lambda_k^{\text{sk}}\} \tag{14}$$

is a gradient strength field across the sketches, and d is a distance function between two orientations defined on $[0, \pi) \times [0, \pi)$ as

$$d(\theta, \phi) = \sin|\theta - \phi|. \tag{15}$$

The smoothing term

$$E_{\text{sm}}(\mathbf{\Theta}_k) = \sum_{\langle s,t \rangle} d(\mathbf{\Theta}_k(s), \mathbf{\Theta}_k(t)) \tag{16}$$

measures the similarity between adjacent pixels s and t in $\mathbf{\Theta}_k$, and the prior term is similarly defined homogeneously as

$$E_{\text{pri}}(\mathbf{\Theta}_k) = \sum_{s \in \Lambda_k} d(\mathbf{\Theta}_k(s), \mathbf{\Theta}_k^{\text{pri}}(s)) \tag{17}$$

to apply additional preferences to pixel orientations in $\mathbf{\Theta}_k$, which is especially useful for regions with weak or even no data constraint of $\mathbf{\Theta}_k^{\text{sk}}$ such as the sky.

An orientation diffusion algorithm [Perona 1998; Chen and Zhu 2006] can be applied to minimize $E(\mathbf{\Theta}_k)$ for the objective $\mathbf{\Theta}_k$. With $\mathbf{\Theta}_k, k = 1, 2, \ldots, K$, the orientation field $\mathbf{\Theta}$ of the whole image is eventually computed with

$$\mathbf{\Theta} = \mathbf{\Theta}_1 \cup \mathbf{\Theta}_2 \cup \cdots \cup \mathbf{\Theta}_K. \tag{18}$$

Figure 3(b) visualizes, by Linear Integral Convolution (LIC), an orientation field generated with the sketch graph in Figure 3(a), where the Gaussian prior energy is disabled for clarity. With our layered representation and algorithms, the generated orientation field is determined by only local sketches and boundaries within each region, thus it prevents abnormal flows along boundaries between adjacent regions caused by occlusion, for example, the background flows around the contour of the two people in the example shown in Figure 3(b).

3. PAINTERLY RENDERING DRIVEN BY THE PARSE TREE

3.1 The Brush Dictionary

In the literature, physically-based/motivated brush models have a few common problems.

(1) It is difficult to design parametric physical models to achieve the photoreality that can satisfy human visual perception.
(2) Most physically-based simulations involve dense computations which prevent them from being applied in interactive systems.

On the contrary, image-based models are proved applicable for applications as complex as decomposing and animating a Chinese-styled painting [Xu et al. 2006].

We have developed an example-based model for brushes with a brush dictionary collected from professional artists. Some examples from the dictionary are shown in Figure 4. Brushes in the dictionary are of four different shape/appearance categories: point (200 examples), curve (240 examples), block (120 examples), and texture (200 examples). Approximate opacity and height maps are manually produced for the brushes using image processing software according to pixels' gray levels. Backbone polylines are also manually labeled for all brushes. With variations in detailed parameters, these brushes reflect the material properties and feelings in several perceptual dimensions or attributes, for example, dry versus wet, hard versus soft, long versus short, etc.

Original colors of the brushes in the dictionary are close to green. During the rendering process, they will be dynamically transferred to expected colors, using a color transfer algorithm similar to Reinhard et al. [2001]. The color transfer operation takes place in the HSV color space to keep the psychological color contrast during the transfer. Since the pixels within a brush image are nearly monotone in contrast to the colorfulness of common natural images, this algorithm capturing only means and variances of colors works quite well, as shown in Figure 5. For each brush in the dictionary, we have its opacity and height maps in addition to the shape and color information, allowing painting with different blending methods according to properties of target regions, as well as photorealistic shading effects.

3.2 Stroke Placement and Rendering

We adopt a layered stroke placement strategy. During the rendering process, we start from the most distant layer, and move backwards to the foreground layer. Then the whole stroke placement sequence is determined by the sequences for the layers. For each layer, we use two types of strokes for the processing of curves and regions, respectively. Usually, strokes for curves are placed upon (or after, in time) strokes for regions for an occlusion effect. For example, long strokes for twigs are placed upon texture strokes for the background sky.

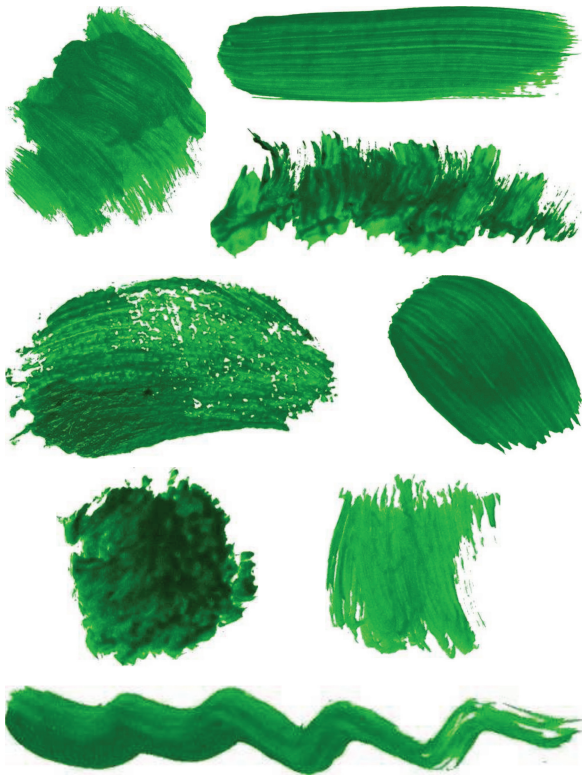


Fig. 4. Some examples from the brush dictionary. The original colors are close to green, and can be dynamically transferred to required colors during synthesis. In additional to the shape and color information, opacity and height maps of the brushes are also available for color blending and shading.

The strokes for curves are placed along the long and smooth curves we acquired during the pursuit of primal sketch (see Figure 3(a)), with morphing operations to bend the brush backbones as well as the attached color pixels according to curve shapes. As for the strokes for regions, we use a simple greedy algorithm to determine the sequence of placement. For each region in a specific layer, we follow the following steps:

(1) Construct a list q to record pixel positions. Randomly select an unprocessed pixel s in this region, and add s to q.
(2) According to the orientation $\Theta(s)$ of s, find pixel t in its 8-neighborhood using

$$t = s + (\text{sign}[\cos\Theta(s)], \text{sign}[\sin\Theta(s)]). \quad (19)$$

(3) If $\cos(\Theta(s) - \Theta(t)) > 1/\sqrt{2}$, add t to q, then let $s = t$ and go to step 2, otherwise go to step 4.
(4) Now q contains a list of pixels, which *trace* the orientation flow to form a streamline. According to the shape and length of the streamline, as well as the object category of the current region, we randomly select a brush B from a set of candidates from the dictionary, then calculate the geometric transformation T to adapt the backbone of B to the streamline. Add stroke $\langle B, T \rangle$ to the stroke sequence for the current region, and mark all pixels covered by this stroke as processed.
(5) Stop if all the pixels in the current region are processed, otherwise go to step 1.

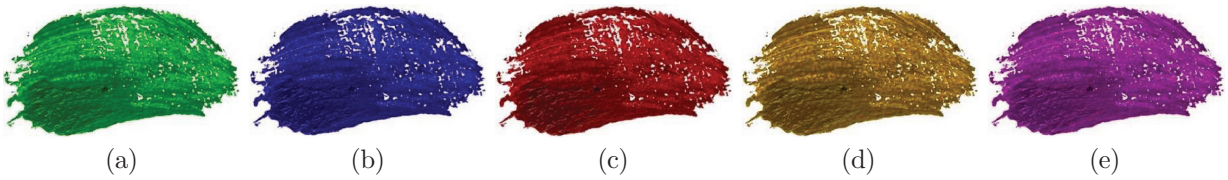


Fig. 5. An example of brush color transfer in the HSV color space. (a) is the original brush image. (b) through (e) are the transferred color brush images.

In order to complete these steps to fulfill the stroke placement task, a few details need to be specified.

(1) In real applications, an orientation field with lower resolution than the original image is often preferred, and the maximum size of list q is limited according to the object category and/or user preferences. The limit depends on the resolution of the discrete orientation field, which corresponds to the size of the result image.

(2) To construct the set of candidate brushes from the dictionary, we have hard-coded the mapping relations between brushes and object categories of regions. Specifically, we divide the four brush categories into more small groups according to the length/width ratios of the brushes, and define probabilities for selection over these groups for each object category. The candidate set is obtained by sampling from the corresponding distribution according to the object category of the region. For example, for an image region labeled as "face/skin", we assign higher probabilities for block brushes with relatively smaller length/width ratios in the dictionary, than the probabilities for very long block brushes and dot, curve, and texture brushes.

(3) To select from the candidate set of brushes, we use the shape parameters obtained from the traced streamline. We select the brush that requires the minimum morphing and scaling to fit the streamline. To achieve this, we adopt a common basis representation for both the backbones of the brushes and the streamline. We first normalize the brushes in the dictionary by placing the start and end points of their backbones at $(x_s, y_s) = (0, 0)$ and $(x_e, y_e) = (1, 0)$, respectively, using affine transformation. After that, we parameterize the backbones as polynomial curves up to the fourth order. For each traced streamline, we fit a similar polynomial curve also with normalization. Then the difference between the streamline and the backbones can be described by the difference between the coefficients of the polynomials, where we usually weigh more for low-order coefficients to emphasize the global shape of the brush stroke. Finally, the brush is selected by minimizing this difference.

After the stroke sequence is determined, the renderer synthesizes the painting image using the high-resolution images from the brush dictionary. Objective colors for color transfer are obtained by averaging over a few random samples from corresponding areas in the source image. Although this method causes loss of information in gradually changing colors, it proves to be no serious problem, especially since the existence of color blocks is one of the observable features of paintings. Depending on the object category of the current region, colors from different brush strokes may be blended using designed strategies, for example, with opacity between zero and one for "face/skin" and "sky/cloud," or without it (i.e., one brush completely covers another) for "flower/fruit" and "grass." Meanwhile, a height map for the region is constructed according to brush properties, for example, the height map accumulates with dry brushes but not with wet brushes. In the end, the photorealistic renderer performs shading with local illumination for the painting image according to the height map.

3.3 Color Enhancement Based on Statistical Analysis

Beside the brush selection and stroke placement algorithms, the system also provides an optional process to transfer and enhance the color of the whole image to match the color statistics of artistic paintings to achieve more appealing rendering effects.

Comparing the colors of natural images and oil-painting images, we found obvious statistical differences between some of their marginal distributions. For example, by defining a psychological color temperature on saturation S and hue H as

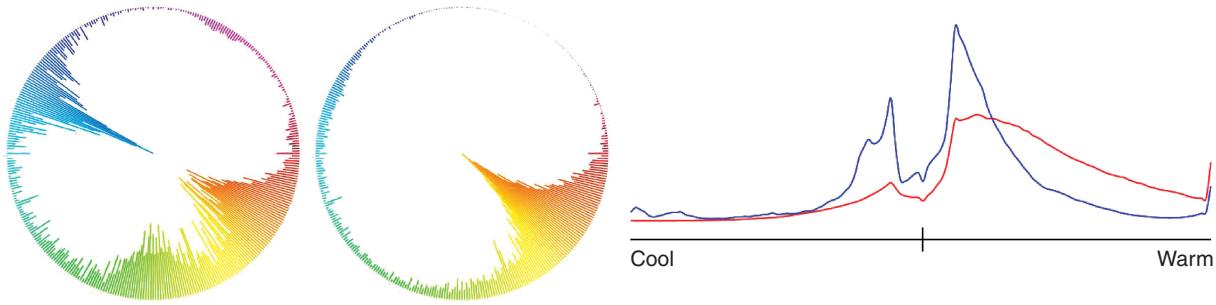
$$\text{ColorTemperature}(S, H) = \frac{S \cdot \sin H}{(S \cdot \cos H)^2 + 1} \tag{20}$$

with orange as the warm pole ($H = \pi/2$) and blue as the cool pole ($H = -\pi/2$) according to human perception, it is observed that oil-paintings by artists tend to appear warmer than natural images, as shown in Figure 6. Also, a study on color statistics by Cohen-Or et al. [2006] shows that the color scheme of an image is *harmony* when its hue follows a V- or L-shape distribution. For a typical warm painting, it does follow the V-shape harmony distribution, as shown in Figure 6(b). In addition, it is observed that color statistics of different images relate directly to the scene category information, for example, portrait, landscape, etc.

We adopt a region-level method to transfer the color of natural images into the color manifolds of painting images. We call this a *color enhancement* operation based on the image-level color transfer algorithm [Reinhard et al. 2001]. By forcing the operation on the region level, we expect better results than the image-level color transfer because the Gaussian assumption is better satisfied. We use a nonparametric strategy to choose color scheme from preanalyzed painting images. We have built a color scheme dictionary including 40 typical schemes collected from painting masterpieces, whose color statistics are computed on the levels of both regions and images. When it comes to select the target scheme from the dictionary, we choose the image with the most similar scene by minimizing an approximate distance between two parse trees

$$\begin{aligned}\delta(W_1, W_2) = &- \sum_{R_i \in W_1} \left(\prod_{R_j \in W_2} \mathbf{1}_{c_i \neq c_j} \right) \log \tilde{f}(c_i) \\ &- \sum_{R_i \in W_2} \left(\prod_{R_j \in W_1} \mathbf{1}_{c_i \neq c_j} \right) \log \tilde{f}(c_i)\end{aligned} \tag{21}$$

in which $\tilde{f}(c_i)$ refers to the empirical prior probability or frequency of object categories in all observations. This is based on the assumption that rare objects tend to be representative scene descriptors.



(a) (b) (c)

Fig. 6. A comparison of distributions of hue and color temperature between natural images and oil-painting images: (a) hue distribution of selected typical natural image samples, (b) hue distribution of selected typical oil-painting image samples, (c) distributions of color temperature (blue curve for natural images and red for oil-painting images). Compared with natural images, oil-painting images mostly have red-orange-yellow colors, and are consequently usually warmer than the former.

(a) (b) (c)

Fig. 7. An illustration of color enhancement. (a) is the source image. (b) is the enhanced image generated according to the scene information. (c) is the final painterly rendering result.

Having the image with the most similar scene and its color statistics, we do color transfer operations for each region according to its corresponding region in the target color scheme. Regions whose object categories are not included in the target scheme are transfered towards the whole image statistics of the scheme. Figure 7 illustrates an example of our color enhancement results.

4. EXPERIMENTAL RESULTS

We have done experiments on hundreds of images of various types of scenes or portraits. Some results are shown in Figures 1 and 8 to 11. Figure 1 includes the final painterly rendering result corresponding to the image parsing example in Figures 2 and 3. Figure 8 is an image of a common landscape scene, with the sky, water surface, rocks, and trees. With our hierarchical parsing and rendering solution, different brushes, strokes, and blending effects are available for different objects. Figure 9 illustrates the use of brushes of multiple sizes by rendering potted flowers in front of an abstract background. Figure 10 displays the rendering result with color enhancement, which is especially useful for daily photos with imperfect colors. Figure 11 shows another landscape painterly rendering.

5. CONCLUSIONS AND FUTURE WORK

The framework proposed previously for painterly rendering based on image parsing can work well on various types of natural images, with satisfactory global and local effects. Compared with previous methods, the improvements in painterly effects are attributed to the fact that our framework takes into consideration the rich semantic information of images stored in hierarchical parse tree structures, and adopts a layered representation for rendering.

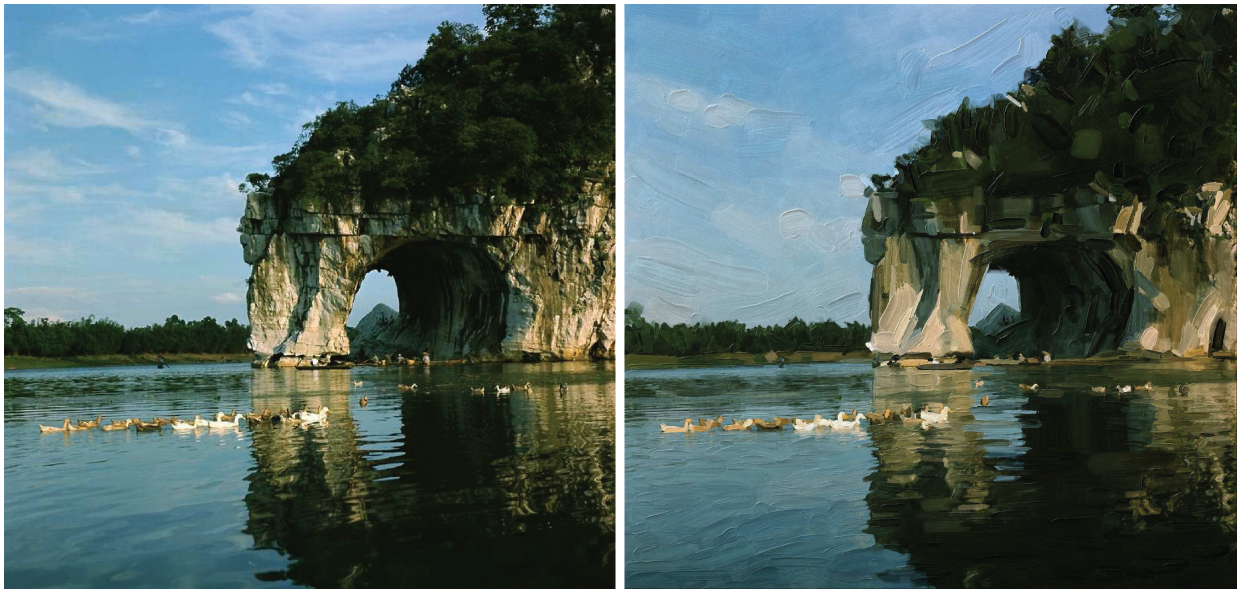
By substituting the brush dictionary with proper graphical elements and adjusting detailed rendering strategies, it is possible to extend this framework for NPR of multiple types and styles, for example, mosaics, stipples, and pencil drawings, etc. In order to achieve such migrations and extensions, we need to eliminate some limits existing in our current framework:

(1) Instead of just tuples of shapes, colors, opacities, heights, etc., a more expressive brush dictionary should be constructed to model some special effects used by artists, and details of the current features can also be richer (e.g., brushes with mixed complementary colors).

(2) More veritable and robust association between brush parameters and object properties should be modeled, with more flexible rendering procedures, for better depicting subtle substances such as human faces and fine fabrics (not necessarily using small and thin brushes according to professional artists).

(3) The stroke placement driven by the primal sketch and orientation field needs to be improved for zigzag or semitransparent boundaries, especially for impressionism paintings.

(4) More advanced vision algorithms should be developed for interactive image parsing with a more friendly software interface providing a better user experience.



(a) (b)

Fig. 8. An example painterly rendering result. (a) is an image of a common landscape scene, including the sky, water surface, rocks, and trees, and (b) is its corresponding painting image.

(a) (b)

Fig. 9. An example painterly rendering result. (a) is a typical scene of still life: potted flowers in front of an abstract background, and (b) is its corresponding painting image.

(a) (b)

Fig. 10. An example painterly rendering result. (a) is an artistic photo of a young lady's upper body, which is dark in its original color, and (b) is its corresponding painting image with color enhancement.

(a) (b)

Fig. 11. An example painterly rendering result. (a) is a computer-generated photorealistic landscape scene, and (b) is its corresponding painting image.

The project Web page with the most updated results: http://www.stat.ucla.edu/~mtzhao/research/parse2paint/.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their suggestions to improve the presentation of this article. We also thank the artist group at Lotus Hill Institute for their help in building the brush dictionary and valuable feedbacks to our painterly rendering effects.

REFERENCES

BOYKOV, Y. AND JOLLY, M.-P. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV'01)*. Vol. 1. 105–112.

CHEN, H. AND ZHU, S.-C. 2006. A generative sketch model for human hair analysis and synthesis. *IEEE Trans. Pattern Anal. Mach. Intell. 28,* 7, 1025–1040.

CHU, N. S.-H. AND TAI, C.-L. 2005. Moxi: Real-Time ink dispersion in absorbent paper. *ACM Trans. Graph. 24,* 3, 504–511.

COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. *ACM Trans. Graph. 25,* 3, 624–630.

COMANICIU, D. AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24,* 5, 603–619.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-Generated watercolor. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. 421–430.

FUNCH, B. S. 1997. *The Psychology of Art Appreciation*. Museum Tusculanum Press.

GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98)*. 447–452.

GOOCH, B., COOMBE, G., AND SHIRLEY, P. 2002. Artistic vision: Painterly rendering using computer vision techniques. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering (NPAR'02)*. 83–90.

GOOCH, B. AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A K Peters, Ltd.

GOOCH, B., SLOAN, P.-P. J., GOOCH, A., SHIRLEY, P., AND RIESENFELD, R. 1999. Interactive technical illustration. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics (I3D'99)*. 31–38.

GUO, C.-E., ZHU, S.-C., AND WU, Y. N. 2007. Primal sketch: Integrating structure and texture. *Comput. Vis. Image Understand. 106,* 1, 5–19.

HAEBERLI, P. 1990. Paint by numbers: Abstract image representations. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'90)*. 207–214.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98)*. 453–460.

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

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*. 327–340.

LI, F.-F., FERGUS, R., AND TORRALBA, A. 2005. Recognizing and learning object categories. *A short course at ICCV 2005*.

LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. *ACM Trans. Graph. 23,* 3, 303–308.

LITWINOWICZ, P. 1997. Processing images and video for an impressionist effect. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. 407–414.

LOWE, D. G. 1999. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV'99), Volume 2*. 1150–1157.

MARR, D. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman.

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

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl. 21,* 5, 34–41.

SOUSA, M. C. AND BUCHANAN, J. W. 1999. Computer-Generated graphite pencil rendering of 3d polygonal models. In *Proceedings of EuroGraphics'99 Conference*. 195–207.

STRASSMANN, S. 1986. Hairy brushes. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'86)*. 225–232.

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

TEECE, D. 1998. 3d painting for non-photorealistic rendering. In *ACM Conference on Abstracts and Applications (SIGGRAPH'98)*. 248.

TU, Z., CHEN, X., YUILLE, A. L., AND ZHU, S.-C. 2005. Image parsing: Unifying segmentation, detection, and recognition. *Int. J. Comput. Vis. 63,* 2, 113–140.

TU, Z. AND ZHU, S.-C. 2006. Parsing images into regions, curves, and curve groups. *Int. J. Comput. Vis. 69,* 2, 223–249.

TURK, G. AND BANKS, D. 1996. Image-Guided streamline placement. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. 453–460.

WINKENBACH, G. AND SALESIN, D. H. 1994. Computer-Generated pen-and-ink illustration. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'94)*. 91–100.

XU, S., XU, Y., KANG, S. B., SALESIN, D. H., PAN, Y., AND SHUM, H.-Y. 2006. Animating chinese paintings through stroke-based decomposition. *ACM Trans. Graph. 25,* 2, 239–267.

YAO, B., YANG, X., AND ZHU, S.-C. 2007. Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. In *Proceedings of the International Conferences on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '07)*. 169–183.

Received November 2008; revised May 2009; accepted July 2009