

Stroke Style Analysis for Painterly Rendering

Yu Zang¹ (臧 戡), Hua Huang^{2,*} (黄 华), *Member, CCF, IEEE*, and Chen-Feng Li³ (李晨风)

¹*School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

²*Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science and Technology
Beijing Institute of Technology, Beijing 100081, China*

³*College of Engineering, Swansea University, Swansea SA2 8PP, U.K.*

E-mail: zangyu7@126.com; huahuang@bit.edu.cn; c.f.li@swansea.ac.uk

Received May 5, 2013; revised August 4, 2013.

Abstract We propose a novel method that automatically analyzes stroke-related artistic styles of paintings. A set of adaptive interfaces are also developed to connect the style analysis with existing painterly rendering systems, so that the specific artistic style of a template painting can be effectively transferred to the input photo with minimal effort. Different from conventional texture-synthesis based rendering techniques that focus mainly on texture features, this work extracts, analyzes and simulates high-level style features expressed by artists' brush stroke techniques. Through experiments, user studies and comparisons with ground truth, we demonstrate that the proposed style-orientated painting framework can significantly reduce tedious parameter adjustment, and it allows amateur users to efficiently create desired artistic styles simply by specifying a template painting.

Keywords non-photorealistic rendering, example-based rendering, style analysis, brush stroke technique

1 Introduction

Painterly rendering techniques automatically create painting-like images and video frames from real photos and videos, and the artistic style of painting results can be adjusted by some parameters. In order to reduce tedious adjustment of puzzling parameters, texture-synthesis based painting techniques have been developed^[1-3], where a template painting is employed to indicate the desired visual effect, and by using some texture synthesis techniques, the result image is rendered to simulate the texture features of the template.

However, comparing with real paintings of different artistic styles, it is clear that defining various painting effects through texture features has some serious limitations: 1) It is difficult, if not impossible, to describe the visual styles for paintings without significant texture features; 2) The flexible brush stroke techniques cannot be well represented and as a result, the rendered results often appear to be mechanical; 3) It cannot describe the uniform artistic style reflected in different paintings that contain totally different texture features (e.g., the representative artistic style associated with the same artist and during the same period).

Artistic styles of many paintings are determined by artists' specific brush stroke techniques. For example, fine brush strokes are used in realist paintings to depict the detailed reality in a "true-to-life" manner, while large visible brush strokes are used in impressionist paintings to emphasize the composition, movement of objects, and light contrast. To simulate this, stroke-based painterly rendering is controlled by a set of parameters related to stroke properties such as size, color, distribution and orientation. With the development of painterly rendering techniques, more painting parameters are included in these more and more complicated rendering frameworks, making it difficult for amateur users to obtain desired results without many trials. However, partially due to the difficulty of brush-stroke detection in real paintings, there has been little progress in automatic style analysis and style-orientated painterly rendering.

In this work, an automatic brush-stroke analysis technique is proposed and the style analysis results are converted to painting parameters in a stroke-based painterly rendering system. It must be admitted that there are much more other style features rather than just stroke properties, but for many paintings stroke

Regular Paper

This work is supported by Fok Ying-Tong Education Foundation of China under Grant No. 131065 and the International Joint Project from the Royal Society of UK under Grant No. JP100987.

*Corresponding Author

©2013 Springer Science + Business Media, LLC & Science Press, China

properties are a big aspect. Hence in this paper we focus on the simulation of high-level style features determined by stroke properties, and it could be a part of any system which would achieve complete style transfer in the future.

2 Related Work

Stroke-based painterly rendering techniques^[4-10] produce painting-like results from an input image or video. In these systems, the painting process is largely controlled with a set of parameters related to stroke properties. By exploiting temporal coherence in video clips, Litwinowicz^[4] designed an automatic filter to produce impressionist animations in a hand-drawing style. Hertzmann^[5] improved this work by proposing a multi-level stroke distribution strategy and a curved stroke model, in which the strokes are fitted with a set of control points, and the painting style can be adjusted by some experimental parameters. Hays and Essa^[6] proposed an RBF-interpolation based technique to compute the stroke orientation and produced impressive painting results. Kagaya *et al.*^[7] focused mainly on video, and expressed the result in multiple styles by using a segmentation-based rendering technique with a new stroke assignment strategy. In order to preserve temporal coherence, Huang *et al.*^[8] proposed a video-layering based painterly rendering technique, which reduces the flickering artifacts considerably. Taking into account the object movement in video clips, Lee *et al.*^[9] proposed a novel technique to compute the stroke orientation field for an input video sequence. Recently, Zeng *et al.*^[10] presented a new rendering framework based on a hierarchical parsing tree to describe the semantic information of an image, and their approach can provide lifelike results in oil-painting style. The aforementioned methods focus mainly on the rendering side and pay little attention to style analysis. As a result, the style-related painting parameters need to be set manually, which may be confusing for amateur users.

To reduce the labor of parameter adjustments, example-based painting techniques^[1-3] were proposed to produce results that have features similar to those of given templates. By learning the relationship of a pair of images A and A' , Hertzmann *et al.*^[1] proposed a method to synthesize the result B' based on the input B . Wang *et al.*^[2] proposed a patch-based texture synthesis technique to speed up the style learning process, where only one input template is needed. Lee *et al.*^[3] added into the traditional fast texture transfer algorithm a new energy term related to the gradient direction, and the new method can produce results with the reference template's texture features while preserving the orientation field of the input image. These

studies treat artistic styles as some low-level texture features. However, artistic styles are essentially determined by painting techniques. Defining it with content-dependent texture features is indirect and as discussed earlier, has some serious limitations. Therefore, in this paper, we focus mainly on analyzing and learning some high-level style features determined by artists' brush stroke techniques.

This work is also related to automatic artist identification^[11-13]. Lyu *et al.*^[11] built a statistical model of an artist from some high-resolution painting scans, where wavelet statistics are used for identification of new paintings. Li and Wang^[12] proposed a two-dimensional (2D) multi-resolution hidden Markov model for Chinese paintings, and by taking the pyramid of wavelet coefficients as features, they achieved an average identification accuracy of about 70%. Yelizaveta *et al.*^[13] proposed a semi-supervised approach for the annotation of brushwork in paintings, which can be helpful for the annotation of the painter and the painting period, etc. These studies focus mainly on artist identification, but some of the tools and concepts are also relevant to our work.

3 Overview

Fig.1 shows some real paintings with different artistic styles. Inspired by these artistic paintings and some previous work^[5,13-15] it can be implied that many artistic styles can be well represented by stroke-related properties, such as stroke visibility, size, orientation and its intrinsic texture (which often arises from the lighting effect of the pigment). In this work, four stroke properties including stroke visibility, stroke size (including the length and width), stroke orientation and stroke texture are chosen to form the style space Γ . These features are selected because they are the most frequently used and the most influencing stroke properties in some latest painterly rendering frameworks (e.g., [6] and [7]). In fact, it is also noticed that other features such as shape and color exaggerations are also significant for artistic styles, while in this work we focus mainly on strokes, the basic unit in painting, and leave other interesting features for further research.

The framework of our style-orientated painting system is shown in Fig.2. Given a style template, its brush strokes are automatically extracted using a novel detection technique, which sequentially enables the construction of a stroke confidence map. Based on this map and the stroke area it indicates, some analysis specific to the style space Γ is performed to automatically initialize the painting parameters for the target image. The details of the new style-analysis based painting technique are explained in the following sections.

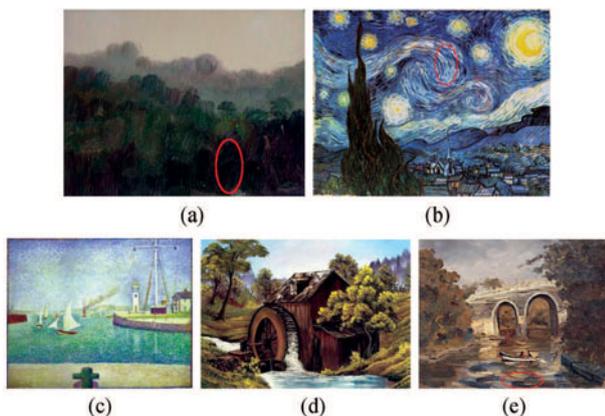


Fig.1. Paintings of different artistic styles. (a) A typical impressionist painting. (b) Painted by Van Gogh and also belonging to impressionism. (c) Painted by Seurat and belonging to pointillism. (d) A typical realist painting. (e) Another impressionist painting with a different style of stroke texture.

4 Stroke Analysis

Brush strokes in a painting always overlap and cross with each other, making it difficult to extract and analyze stroke features. For artist identification^[11,13,16-17], stroke patches are often extracted manually. In this work, in order to facilitate amateur users and minimize the tedious labor in producing desired painting effects, we propose a novel and fully automatic method to detect stroke features in real paintings. The outcome of the automatic stroke detection is a gray map termed stroke confidence map, in which the value of each pixel denotes the confidence level for the pixel to lie in an area with clear stroke features (e.g., the circled areas in Figs.1 (a), 1(b) and 1(e)). This map, which indicates the visibility and distribution of brush strokes in a painting, is then used as the major reference for further analysis of stroke-related artistic styles.

4.1 Criteria for Stroke Identification

As shown in Fig.1, the strokes in some paintings are more visible than others, and even in the same painting, the expression of strokes can differ from place to place. However, it is observed in all these paintings that the visible strokes always appear as some texture areas distinguishing themselves from other significant contours. It is also observed that in local areas containing strong stroke characteristics, the stroke-like line segments gather together roughly aligned. Based on these observations, the basic criteria for stroke identification can be summarized as:

- 1) Stroke features always appear in some texture areas.
- 2) Stroke features have fine local structures with clear directionality.

4.2 Stroke Detection

To transform the intuitive stroke detection criteria discussed above into mathematical models, we analyze the Gabor responses of the template painting. The Gabor filter^[18] has been widely used in the representation of texture features^[19-21], and the 2D case is defined by Daugman^[22] as:

$$g_{\lambda,\theta,\sigma,\phi}(s,t) = e^{-\left(\frac{s'^2}{\sigma_s^2} + \frac{t'^2}{\sigma_t^2}\right)} \cos\left(\frac{s'}{\lambda} + \phi\right), \quad (1)$$

where $s' = s \cos \theta + t \sin \theta$ and $t' = s \sin \theta + t \cos \theta$. The filter response of a signal f can be written as:

$$R_{\lambda,\theta,\sigma,\phi}(x,y) = \iint_W f(x-s,y-t)g_{\lambda,\theta,\sigma,\phi}(s,t) ds dt, \quad (2)$$

where W denotes the filter window, λ is the preferred spatial-frequency, θ is the normal orientation of the Ga-

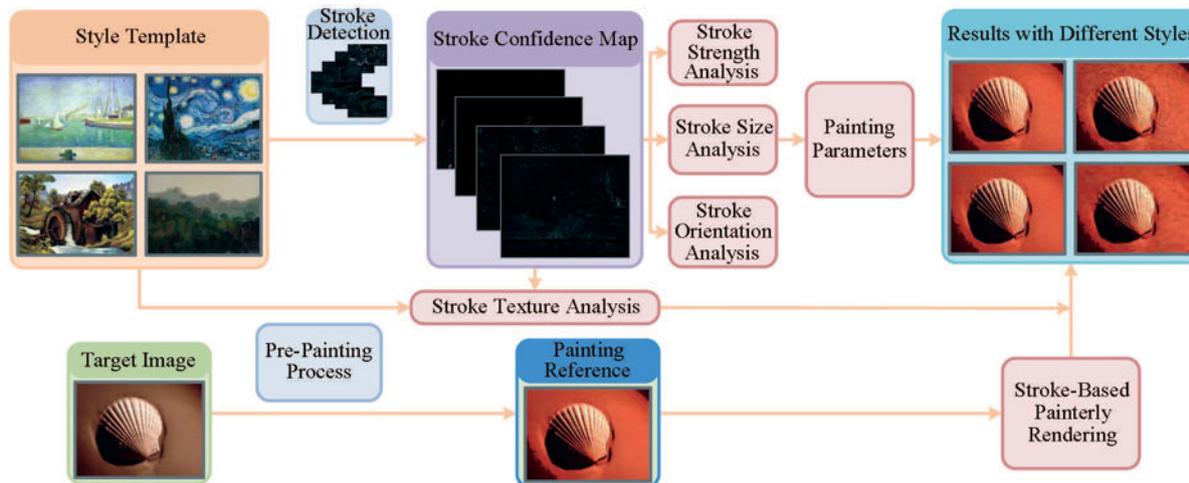


Fig.2. Algorithm framework of our style-oriented painting system.

bor function, σ_s and σ_t are parameters of Gaussian envelop, and ϕ is the phase offset defining the real ($\phi = 0$) and imaginary ($\phi = \frac{\pi}{2}$) parts of the filter response. More details of Gabor filter can be found in [23].

The new stroke detection algorithm operates in two steps corresponding to the two detection criteria described in Subsection 4.1 respectively:

1) A multi-frequency feature analysis is performed to extract texture areas and remove significant semantic contours (as shown in Fig.3(d)) from the template painting.

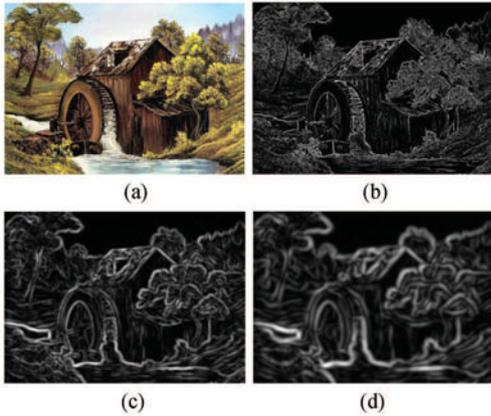


Fig.3. Superposition Gabor-energy responses at different spatial frequencies. (a) Template painting. (b)~(d) Superposition Gabor-energy responses for $\lambda = \sqrt{2}$, $\lambda = 4\sqrt{2}$ and $\lambda = 7\sqrt{2}$, respectively.

2) Through a multi-orientation feature analysis, stroke features with clear directionality are extracted from the remaining texture areas and form a stroke confidence map which indicates the confidence of a pixel belongs to stroke area.

Multi-Frequency Feature Analysis. For each pixel in the template painting, the Gabor-energy is defined as:

$$e_{\lambda,\theta}(x,y) = \sqrt{R_{\lambda,\theta,1,0}(x,y)^2 + R_{\lambda,\theta,1,\frac{\pi}{2}}(x,y)^2}, \quad (3)$$

where a circular Gaussian enveloped with $\sigma_s = \sigma_t = 1$ is adopted in all our experiments. The superposition response at each pixel represents the dominant Gabor-energy among all directions, and it is numerically defined as:

$$e_{\lambda}(x,y) = e_{\lambda,\theta_{max}}(x,y) + e_{\lambda,\theta_{max-1}}(x,y) + e_{\lambda,\theta_{max+1}}(x,y), \quad (4)$$

where $\theta_i = \frac{i\pi}{8}$ for $i = 0, \dots, 7$, the subscript *max* corresponds to the discretized orientation with the maximum Gabor-energy, and the subscripts *max* - 1 and *max* + 1 indicate the neighboring orientations. In (4), the Gabor-energies of each pixel are computed for

eight discretized orientations $\frac{i\pi}{8}$, and adding together the maximum energy response and the neighboring responses ensures a good capture of the dominant Gabor-energy that can occur at any direction in the continuous range $\theta \in [0, \pi)$.

In our implementation, the superposition responses $e_{\lambda}(x,y)$ are calculated for seven different spatial frequencies $\lambda_i = i\sqrt{2}$, $i = 1, \dots, 7$, and these pixel-wise responses are re-scaled to $[0, 255]$ across the image. Fig.3 shows an example of the superposition Gabor-energy responses for different spatial frequencies. The results are consistent with the perception of human eyes: people can see clearly the small-scale details of a complicated object when they are sufficiently close to the target; and as they move away from the object, the texture details start to blur and disappear in the view, leaving only an outline shape (as demonstrated in [18]). It means that with the increase of spatial frequencies, Gabor responses in the texture-like and the contour-like areas vary differently. In texture areas, the response decreases as the scale increases, but for contours, the response always increases or stays roughly unchanged. In addition, it is observed in the superposition response of any fixed frequency that the spatial correlation is also important, i.e., the neighborhood of a texture/non-texture pixel is more likely to be part of a texture/non-texture feature.

Based on the above observations, the texture map $M(x,y)$ of a template painting T can be determined as:

$$M(x,y) = \sum_{i=1}^6 M_i(x,y)\mu(M_i), \quad (5)$$

where $M_i(x,y)$ is the initial estimation of texture areas at the scale λ_i , and $\mu(M_i)$ is the adjustment term due to spatial correlation. Specifically, the texture areas at different scales are calculated by comparing the difference between superposition Gabor-energy responses at neighboring scales, i.e.,

$$M_i(x,y) = (e_{\lambda_i} - e_{\lambda_{i+1}})sign(e_{\lambda_i} - e_{\lambda_{i+1}}), \quad (6)$$

where $sign(z)$ denotes the sign function that equals 1 if $z > 0$ and equals 0 if $z \leq 0$. The spatial correlation adjustment is calculated by counting the texture-map responses in the neighborhood of each pixel, i.e.,

$$\mu(M_i) = \sum_{(x',y') \in \Omega} M_i(x',y'), \quad (7)$$

where Ω denotes the neighborhood of the pixel $p(x,y)$. The radius of the neighborhood Ω is set to 2 in all our experiments.

Multi-Orientation Feature Analysis. The pixel value of the texture map $M(x,y)$ (calculated by (5)) indicates the likelihood of the pixel $p(x,y)$ to be part of a texture

feature. Taking $M(x, y)$ as the input and fixing the spatial frequency at the smallest scale $\lambda = \sqrt{2}$, we compute the Gabor-energy responses of the texture map for eight different orientations, denoted by $E_{\theta_i}(x, y)$ for $\theta_i = \frac{i\pi}{8}$ and $i = 0, \dots, 7$. This is done in a way similar to (3).

Stroke features have clear directionality, and can be approximately detected by comparing the deviation of Gabor-energy responses $E_{\theta_i}(x, y)$ for different orientations. Specifically, we define the directionality map as

$$M_D(x, y) = \frac{E_{\theta_{max}} + E_{\theta_{max-1}} + E_{\theta_{max+1}} - \bar{E}_\theta}{E_{\theta_{max}} + E_{\theta_{max-1}} + E_{\theta_{max+1}}}, \quad (8)$$

where $E_{\theta_{max}}$ denotes the maximum Gabor-energy among eight orientations $\theta_i = \frac{i\pi}{8}$ for $i = 0, \dots, 7$, θ_{max-1} and θ_{max+1} denote the neighboring orientations of θ_{max} , and \bar{E}_θ is the average Gabor-energy of $E_{\theta_i}(x, y)$ without contributions from the orientation θ_{max} or its immediate neighbors (i.e., θ_{max-1} and θ_{max+1}). The directionality map $M_D(x, y)$ examines the orientation-wise variation of Gabor-energy responses, and it is clear that $M_D(x, y)$ takes values in $[0, 1]$. For a fixed pixel $p(x, y)$, a larger value of $M_D(x, y)$ indicates the Gabor-energy of the texture map has a more dominant direction, and consequently the pixel $p(x, y)$ is more likely to be on a stroke feature; a smaller value of $M_D(x, y)$ indicates the Gabor-energy of the texture map has weaker directionality, and hence the pixel $p(x, y)$ is less likely to be part of a stroke feature.

Similar to the texture map defined in (5), the neighborhood of a stroke/non-stroke pixel is more likely to be part of a stroke/non-stroke feature. Thus, we compute the stroke confidence map as

$$S(x, y) = M_D(x, y)\mu(M_D), \quad (9)$$

where $\mu(M_D)$ is the adjustment term due to spatial correlation, and it is calculated in a way similar to (7). The stroke confidence map $S(x, y)$ represents the visibility and distribution of brush strokes in the given template painting, and the value of each pixel $p(x, y)$ indicates the confidence level of the pixel to be part of a stroke feature.

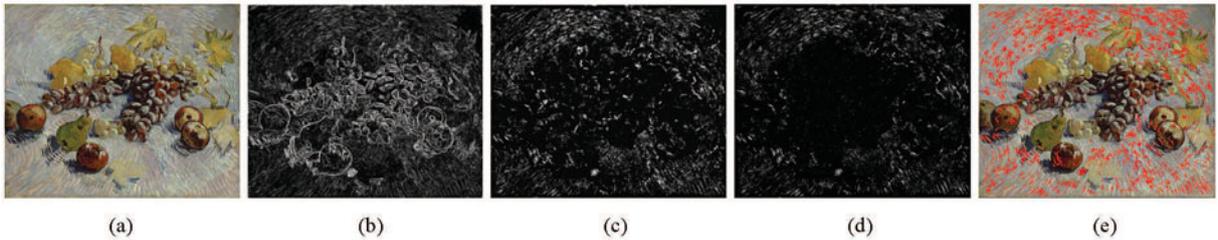


Fig.4. Stroke detection process. (a) Input template painting. (b) Superposition Gabor-energy response with $\lambda = \sqrt{2}$. (c) Texture map detected from the multi-frequency feature analysis. (d) Stroke confidence map obtained from the multi-orientation feature analysis. (e) Visualization of strokes on the original painting.

An example of the stroke detection process is shown in Fig.4. Fig.4(a) is the input style template, Fig.4(b) is the superposition Gabor-energy response corresponding to frequency $\lambda = \sqrt{2}$, Fig.4(c) is the texture map obtained from the multi-frequency feature analysis, Fig.4(d) is the stroke confidence map obtained from the multi-orientation feature analysis, and Fig.4(e) visualizes stroke distribution on the original template painting (using bright red color).

5 Stroke-Based Style Analysis and Rendering

The stroke confidence map provides the reference ground for further analysis of stroke-related style features, and based on the rendering framework of our previous work^[8,24], we develop some adaptive interfaces to automatically connect the style statistics described in Section 5 with the painting parameters for the rendering of different artistic styles.

Stroke Visibility. As shown in Fig.1, brush strokes in Fig.1(d) are almost invisible on the canvas which is significantly different from the other figures. So for a template painting T whose stroke confidence map is $S(x, y)$, the average intensity $\bar{S}(T)$ of the map $S(x, y)$ is designed to measure stroke visibility, which distinguishes realism from other artistic styles painted with expressive brush strokes. For realist paintings, the average intensity $\bar{S}(T)$ will be very small.

On the rendering side, we employ a color blending model^[25]. The main idea of this model is to define the color $C(x, y)$ of a pixel $p(x, y)$ in the canvas by both its current color $C_c(x, y)$ and the stroke color C_s , depending on which color looks closer to the reference color $C_r(x, y)$ in the reference image. That is:

$$C(x, y) = (1 - W) \times C_c(x, y) + W \times C_s. \quad (10)$$

The weight W in the above equation is

$$\begin{cases} (1 - \alpha)G_\sigma(|d_c(p) - d_s(p)|), & \text{if } d_c(p) \leq d_s(p), \\ 1 - (1 - \alpha)G_\sigma(|d_c(p) - d_s(p)|), & \text{if } d_c(p) > d_s(p). \end{cases} \quad (11)$$

where $d_c(x, y)$ is the difference between $C_c(x, y)$ and $C_r(x, y)$, and $d_s(x, y)$ is the difference between C_s and $C_r(x, y)$. Function $G\sigma \in (0, 1]$ is the normalized Gaussian function with mean 0 and standard deviation σ . The parameter α controls the accuracy of the painting result, and a larger/smaller value of α gives a more/less realistic result.

Then to transform the stroke visibility feature to the rendering framework, the average intensity $\bar{S}(T)$ of the template painting T is connected to the painting parameter α in two operational modes: the hard mode and the soft mode. The hard mode aims to find an empirical threshold for $\bar{S}(T)$, distinguishing realism and other artistic styles with expressive brush strokes. The critical threshold is determined by using an SVM-based classification algorithm^[26]. In our experiments, 30 training samples consisting of 15 realist paintings and 15 non-realist paintings are used to determine the critical value for the average stroke intensity, and about 50 test samples are used to verify the method, from which we find that the typical threshold for $\bar{S}(T)$ is 3.54 and the recognition accuracy is about 82%. However, the consequence of potential failure of this hard mode can be severe, because once the template is misrecognized the painting result will appear in a completely different style. Therefore, a soft mode is also designed for critical cases, and it is expressed by a Hyperbolic Tangent function as $\alpha = 1 - \tanh(\phi \times \bar{S}(T))$, where ϕ is a tuning factor set to 0.2 in our implementation. All experiments in this paper are performed in the hard mode because it produces better results when the painting style of the template has been correctly recognized.

Stroke Size. Stroke size is another important factor greatly affecting paintings' artistic styles. Based on the stroke confidence map, we design a statistical character that is related to stroke size. Figs. 5(a) and 5(c) show the stroke confidence maps computed from two template paintings in Fig.4(a) and Fig.1(a), respectively. For Fig.4(a) whose stroke size is relatively small, the corresponding stroke confidence map in Fig.5(a) tends to be discrete and dispersive; and for Fig.1(a) whose stroke size is larger, the corresponding confidence map in Fig.5(c) presents a more continuous pattern with a larger area. Based on this observation, we use the connectivity (which is denoted as $L(T)$) of the stroke map $S(x, y)$ as an indicator for stroke size. First, stroke features in a stroke confidence map are clustered automatically using the mean shift algorithm^[27]. To obtain better results, a binarization processing is applied before clustering, for which we employ a moment-preserving method^[28] to automatically determine the threshold. This scheme selects an optimal threshold such that the resulting binary image preserves the first three mo-

ments of the input image. Corresponding to the stroke confidence maps in Figs. 5(a) and 5(c), the clustering results are shown in Figs. 5(b) and 5(d), respectively. Then, the connectivity $L(T)$ is measured by the average area of each individual cluster, as indicated by the color patches in Figs. 5(b) and 5(d). It is also noticed that the image size may affect the connectivity measurement to some degree, so the template image needs to be processed in a similar size.

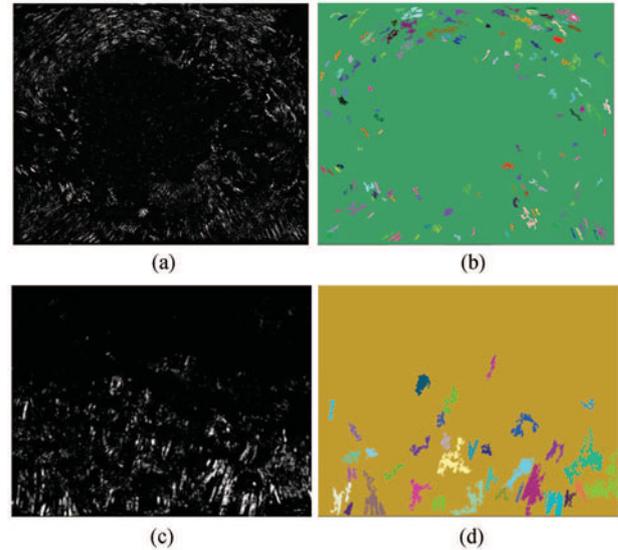


Fig.5. Stroke size analysis. (a) Stroke confidence map of Fig.4(a). (b) Clustering result of (a). (c) Stroke confidence map of Fig.1(a). (d) Clustering result of (c).

Then in the painting system^[24], different areas in the canvas are rendered with different stroke sizes, depending on the levels of details of the target image. Hence, we link the connectivity $L(T)$ to the largest stroke size, which is then used to proportionally determine the sizes of smaller strokes. For the largest stroke, its size l is computed as

$$l = \frac{L(T) - L_{\min}}{L_{\max} - L_{\min}} \times (l_{\max} - l_{\min}), \quad (12)$$

where L_{\min} and L_{\max} denote the empirical range of $L(T)$ with typical values of $L_{\min} = 10$ and $L_{\max} = 470$, and l_{\min} and l_{\max} denote the usual range of stroke sizes in the painting system with typical values of $l_{\min} = 1$ and $l_{\max} = 40$. It is observed in our experiments that the results will be rendered into a pointillism style if the value of l is less than 3. Therefore, the stroke size l is taken as the main reference to recognize such style.

Stroke Orientation. Stroke orientation also has a large impact on the artistic styles of many paintings. However, finding a comprehensive representation for

stroke-orientation style can be very challenging because both the image content and the painter's techniques have strong and complicated influences on the stroke distribution. In this paper, we focus on a small but signal aspect of the stroke orientation, where variance of the stroke orientation field is measured, and such feature is then used as a simple indicator for the level of disorder of stroke orientation.

First the mean shift segmentation^[27] is applied to the template painting T at a large scale to obtain a macroscopical representation $T = \sum_{k=1}^N T_k$ of the whole image. Then, for each sub-image T_k , the stroke orientation field Θ_k is calculated from the corresponding part on the stroke confidence map. Specifically, for each stroke pixel indicated on the stroke confidence map, its orientation corresponds to the maximum Gabor-energy response (3) among eight directions $\theta_i = \frac{i\pi}{8}$, $i = 0, \dots, 7$, with the spatial frequency set to the finest resolution $\lambda = \sqrt{2}$. Next, to reduce the noise from the Gabor-energy responses, the stroke orientation field Θ_k is locally averaged in a circular window (the radius is set to 20 in all our experiments) to form an averaged field $\bar{\Theta}_k$. Finally, for the whole template painting, the variance of the stroke orientation field is determined as

$$V(T) = \frac{\sum_{k=1}^N A_k \times \text{Var}(\bar{\Theta}_k)}{\sum_{k=1}^N A_k}, \quad (13)$$

where N is the total number of sub-images obtained from the image segmentation, A_k is the area of the sub-image T_k , $\text{Var}(\bar{\Theta}_k)$ is the variance of the locally averaged stroke orientation field $\bar{\Theta}_k$ for the sub-image T_k .

To express the stroke orientation with different degrees of exaggeration, we introduce a set of "style points" whose orientations are influenced by the surrounding feature points and the disorder level of the template painting. The stroke orientation field is then interpolated using both feature and style points. The orientations of style points are computed as:

$$O_s(x, y) = \gamma \times O_f(x', y') + (1 - \gamma) \times O_s^0(x, y), \quad (14)$$

where $O_s(x, y)$ denotes the orientation of a style point $p(x, y)$, $O_f(x', y')$ the orientation of the "nearest" feature point $p(x', y')$, $O_s^0(x, y)$ the initial orientation of the style point which is determined by RBF interpolation of feature points, and $\gamma \in [0, 1]$ the weight factor. Increasing the weight factor γ makes the stroke orientation field more chaotic in the painting process. To control the orientation exaggeration, we set the weight factor γ according to the disorder level of the template

painting which is described by:

$$\gamma(x, y) = \tanh\left(\varphi \times \frac{G(x', y') \times V(T)}{\|p(x, y), p(x', y')\|}\right), \quad (15)$$

where $G(x', y')$ is the gradient magnitude (calculated using the Sobel operator) of the nearest feature point $p(x', y')$, $V(T)$ the variance of stroke orientation in the template painting, $\|p(x, y), p(x', y')\|$ the distance between the style point $p(x, y)$ and the nearest feature point $p(x', y')$, and φ is a tuning factor controlling the changing rate between the orientation fields of different styles with typical values $\varphi \in [8.0, 15.0]$.

Stroke Texture. Artistic styles of real paintings are also represented by stroke texture, as shown in Figs. 6(a) and 6(d). Visual effects of stroke texture are mainly observed in two aspects: 1) the texture variation of individual strokes, which is significantly affected by the choice of brushes and pigment materials; and 2) the overlying effect between strokes, which is largely determined by different painting techniques. To extract information of stroke textures, we use empirical mode decomposition (EMD)^[29]. Compared with standard data analysis tools such as Fourier and wavelet decompositions, the EMD technique decomposes the complex signal into a small number of intrinsic mode functions (IMFs), and for efficient implementation, we follow the approach by Gao *et al.*^[30].

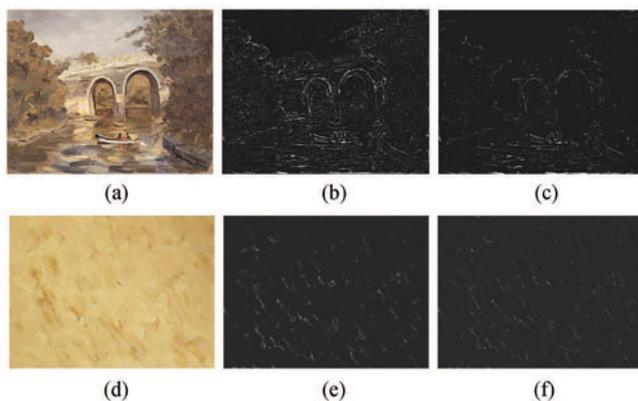


Fig.6. Stroke texture analysis. (a) Template painting. (b) The first IMF of (a). (c) The second IMF of (a). (d) Another template painting. (e) The first IMF of (d). (f) The second IMF of (d).

Specifically, the template painting is converted into a gray image (we use the "L" channel extracted from the Lab color space for easy implementing, and better color to grayscale conversion techniques could be used), which is then decomposed into two IMFs, as shown in Fig.6. In terms of stroke features, the first IMF (IMF_1) (Figs. 6(b) and 6(e)) extracts information at a smaller scale, and both inner- and inter-stroke features can be

observed; the second IMF (IMF_2) (Figs. 6(c) and 6(f)) extracts information at a larger scale, and mainly reflects inter-stroke features.

Hence, for the painting template T with stroke confidence map $S(x, y)$, we define

$$F_1(T) = \frac{\sum_{p(x,y) \in T_1 - T_2} IMF_1(x, y)S(x, y)}{\sum_{p(x,y) \in T_1} IMF_1(x, y)S(x, y)}, \quad (16)$$

to measure the visibility of inner-stroke textures, where T_1 and T_2 denote the pixel sets in the first and the second IMFs respectively whose intensity values are greater than a given threshold. In (16), the stroke pixels are selected by filtering with the stroke map $S(x, y)$, and the stroke-boundary pixels are further removed by taking the set difference between the visible pixel sets T_1 and T_2 . A higher value of $F_1(T)$ indicates a more vibrant characteristic of bristle textures. The inter-stroke features are measured by:

$$F_2(T) = \frac{\sum_{p(x,y) \in T_2} IMF_2(x, y)S(x, y)}{\sum_{p(x,y) \in T_2} S(x, y)}. \quad (17)$$

A higher value of $F_2(T)$ indicates a more distinguishable overlaying effect between strokes.

In rendering side, based on the painting framework of Huang *et al.*^[8,24], a set of nonuniform brush models are employed as shown in Fig.7, where each visible pixel represents a bristle on the brush. The pixel intensity indicates the contact status between the bristle and the canvas, and white means completely touched while black means completely separated. To reflect the template painting's inner-stroke texture features, we adjust each nonuniform brush model B with a random perturbation controlled by the $F_1(T)$ factor, i.e.,

$$I_b(D) = I_b(D) + \beta \times F_1(T), \quad (18)$$

where I_b denotes the intensity of a bristle pixel, $D \subseteq B$ a randomly selected subset of the original brush model B , and $\beta \in [-255, 255]$ a perturbation number initialized randomly for every stroke drawing. The size of the subset D is also set proportionally to the $F_1(T)$ factor

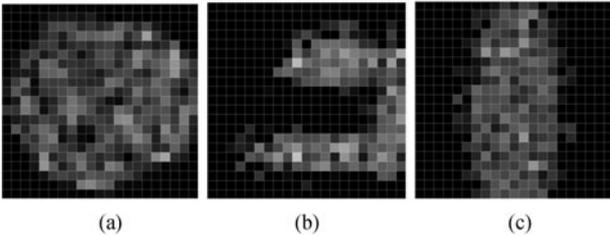


Fig.7. Nonuniform brush models.

such that the larger/smaller value of F_1 is, the more/less bristle pixels will be adjusted.

The overlaying effect between strokes is simulated by a color adjustment scheme based on a height map. To reflect the inter-stroke features of the template painting, we integrate the $F_2(T)$ factor into this color adjustment scheme. Specifically, the pixel intensity of the painted result is adjusted as

$$\begin{aligned} \Delta h(x, y) &= h(x + \cos d, y + \sin d) - h(x, y), \\ I'(x, y) &= I(x, y) + \tanh(\kappa \times F_2(T)) \times \Delta h(x, y) \times \\ &\quad (\min(I(x, y), 255 - I(x, y))/127.5), \end{aligned} \quad (19)$$

where d represents the direction perpendicular to the stroke orientation $O(x, y)$, i.e., $d = O(x, y) + \pi/2$, h denotes the corresponding pixel intensity on the height map, $I(x, y)$ denotes the pixel intensity of the result image, κ is a tuning parameter set as 0.16 in our experiments, and $I'(x, y)$ the pixel intensity after adjustment. Corresponding to larger/smaller value of $F_2(I)$, the intensity adjustment in (19) makes the overlaying effect between strokes more/less visible.

6 Experimental Results and Evaluation

A series of experiments, including parameters detection accuracy, painterly rendering tests, comparisons and user studies, are implemented to demonstrate the performance of the new style-oriented painterly rendering framework. All experiments in this paper are performed on a PC with an Intel® 3.0 GHz Dual Core CPU and a GeForce 9600 GT video card. For an input image with size 800×800 , the analysis process takes about 8~10 seconds and the painting process takes about 9 seconds.

Evaluation of Stroke-Based Style Analysis. The artistic style of paintings is often a subjective concept, depending on the artistic knowledge and preference of the viewer. Hence, to evaluate the effectiveness of the proposed stroke-based style analysis technique, we carry out a user study with 20 art students who do not have knowledge of computer painterly rendering. As shown in Figs. 8(a)~8(o), a variety of painting styles are used for testing and these templates are also with different content, including realism, impressionism, pointillism, landscape, static objects and moving scenes, etc. The automatic style analysis results, including stroke visibility $\tilde{S}(T)$, stroke size $L(T)$, stroke orientation $V(T)$ and stroke texture $F_1(T)$ and $F_2(T)$, are listed in Table 1 as the first value in each table cell. For each template painting, the participant viewers are asked to mark within [1, 10] for five stroke-related style features, with a higher mark indicating clearer stroke visibility, larger average stroke size, more disordered stroke orientation, more varying brush textures and

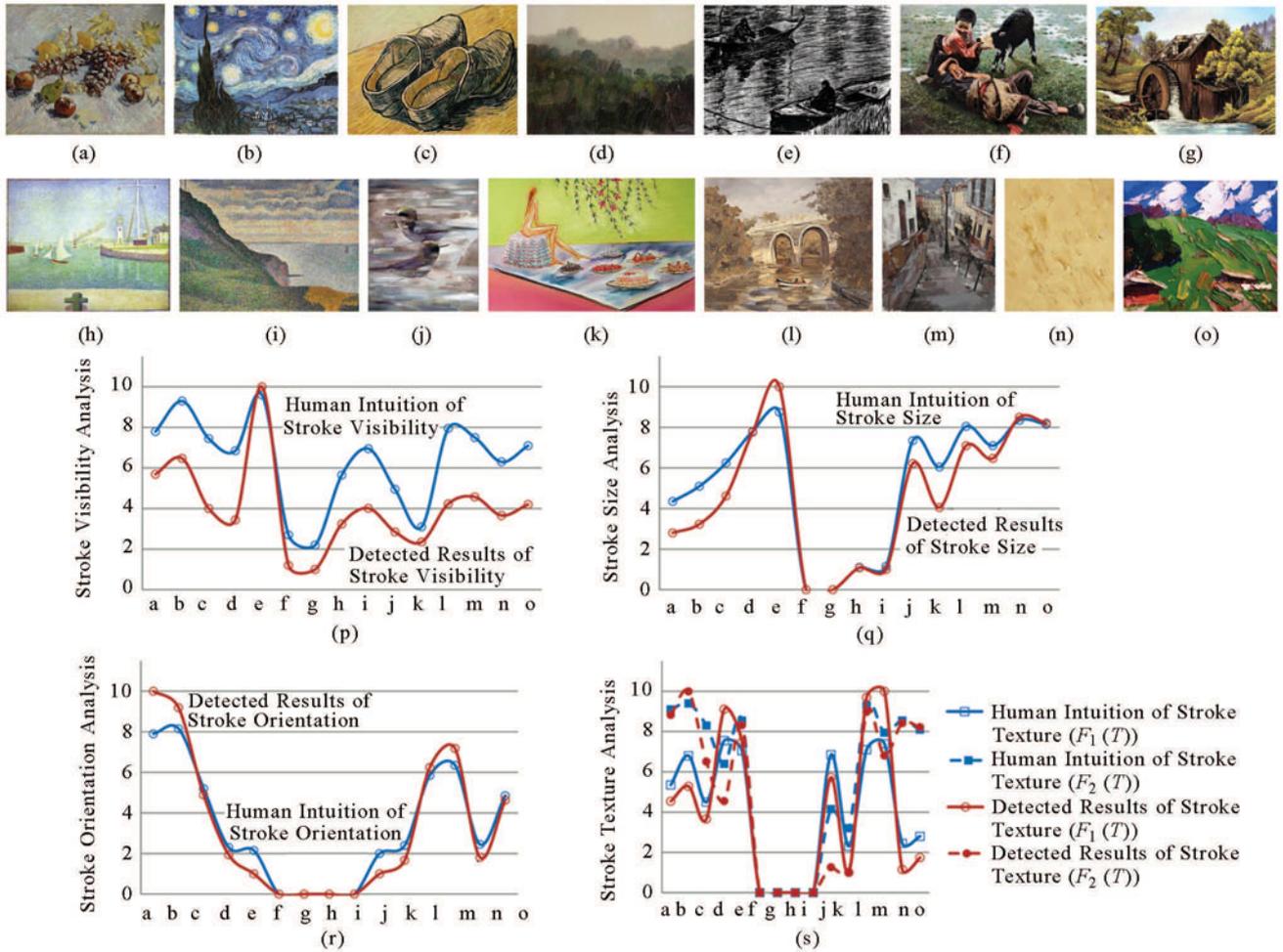


Fig.8. Stroke-based style analysis. (a)~(o) are 15 style templates and (p)~(s) are comparison results between the automatic style analysis and the user evaluation.

Table 1. Comparison Between Our Stroke Detection Results and User Perception

| Image T | Stroke Visibility $\bar{S}(T)$ | Stroke Size $L(T)$ | Stroke Orientation $V(T)$ | Stroke Texture $F_1(T)$ | Stroke Texture $F_2(T)$ |
|-----------|--------------------------------|--------------------|---------------------------|-------------------------|-------------------------|
| a | 11.33 (7.80/1.3) | 116.48 (4.35/1.6) | 0.77 (7.90/1.6) | 0.33 (5.35/1.7) | 17.03 (9.10/0.8) |
| b | 12.84 (9.30/0.8) | 138.92 (5.10/1.1) | 0.71 (8.15/0.9) | 0.38 (6.80/1.7) | 18.87 (9.40/0.5) |
| c | 8.07 (7.45/1.5) | 208.75 (6.25/1.0) | 0.39 (5.20/1.1) | 0.27 (4.50/1.2) | 13.15 (8.30/0.9) |
| d | 7.01 (6.85/1.2) | 365.88 (7.80/0.8) | 0.17 (2.30/0.6) | 0.64 (7.55/1.0) | 9.89 (6.40/1.3) |
| e | 19.48 (9.60/0.7) | 467.88 (8.75/1.9) | 0.10 (2.15/0.6) | 0.53 (7.05/2.0) | 16.17 (8.55/1.1) |
| f | 2.71 (2.70/1.7) | XX | XX | XX | XX |
| g | 2.32 (2.20/1.4) | XX | XX | XX | XX |
| h | 6.63 (5.65/1.6) | 31.33 (1.10/0.5) | XX | XX | XX |
| i | 7.83 (6.95/1.4) | 27.58 (1.15/0.6) | XX | XX | XX |
| j | 5.88 (4.95/2.0) | 288.71 (7.35/0.8) | 0.10 (2.00/0.8) | 0.41 (6.85/1.8) | 4.41(4.15/1.9) |
| k | 4.93 (3.10/0.7) | 179.84 (6.05/1.2) | 0.15 (2.40/0.7) | 0.09 (2.30/1.6) | 3.97 (3.20/1.0) |
| l | 8.54 (7.95/1.3) | 332.39 (8.05/1.1) | 0.49 (5.85/1.6) | 0.68 (7.10/0.9) | 17.27 (9.30/0.8) |
| m | 9.18 (7.50/1.7) | 301.32 (7.10/1.7) | 0.56 (6.35/1.3) | 0.70 (7.40/1.1) | 13.66 (7.95/1.4) |
| n | 7.43 (6.30/1.1) | 402.64 (8.35/1.3) | 0.16 (2.45/1.0) | 0.10 (2.45/1.3) | 16.33 (8.55/1.0) |
| o | 8.48 (7.10/1.4) | 387.85 (8.15/1.2) | 0.37 (4.85/1.2) | 0.14 (2.80/1.4) | 16.01 (8.10/0.6) |

Note: The first value in each table cell: the automatic style analysis result; the first value in each pair of brackets: the corresponding mean value of user evaluations; the second value in each pair of brackets: the standard deviation of user evaluations. For features not relevant to specific template painting, the corresponding cells are filled with “XX”.

more distinguishable individual strokes respectively. The mean value and the standard deviation of the user evaluation are listed in the corresponding table cell, within the bracket. To give a clear comparison, the automatic style analysis results are scaled to [1, 10] and plotted together with the user evaluation in Figs. 8(p)~8(s). It can be seen that for all stroke-based style features, the automatic style analysis is consistent with the user evaluation.

Evaluation of the Whole System. The new method is first applied to render different input images following the same template (Fig.1(b)). The results are shown in Fig.9, where Figs.9(a)~9(c) are different types of inputs, and Figs.9(d)~9(f) are corresponding rendering results, and from which it can be viewed that the style of a specific template can be transformed to different images.

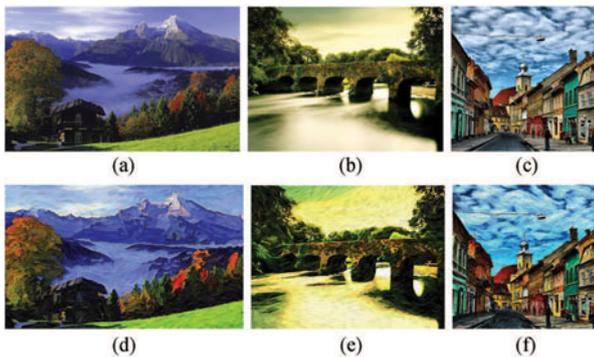


Fig.9. Same template emulated by different inputs, where Fig.1(b) is selected as the template. (a)~(c) Different input images. (d)~(f) Corresponding rendering results.

Then the method is applied to render an image following different style templates. As shown in Figs.10(a)~10(g), seven template paintings are chosen for the test. Fig.10(a) is a realist painting, whose strokes are almost invisible. Fig.10(b) is painted by Vincent Van Gogh, exhibiting a unique curly stroke style. Fig.10(c) is a typical impressionist painting, which uses bold and rough strokes emphasizing the depth-field composition. Painted by Seurat, a representative of pointillism, Fig.10(d) is created by using tiny dot strokes. The strokes in Fig.10(e) are scraped after painting, so that the canvas appears clean and neat. As a result of using dry pigments, strokes in Fig.10(f) appear complicated and show an intense stroke overlaying effect. Such overlaying effect is also visible in Fig.10(g), in which individual strokes are much smoother. Fig.10(h) is the input photo for rendering, and to acquire high quality rendering results, some abstraction^[31-32] and colorization^[33-34] studies are involved as a pre-rendering processing, and the results are shown in Fig.10(i). The remaining images Figs. 10(j)~10(p) show painterly rendering results following the templates Figs. 10(a)~10(g) respectively.

Verification of the Rendering Results. The painterly rendering accuracy is examined by comparing the automatically detected painting parameters with the ground truth. The first part of Table 2 shows the painting parameters and the corresponding style statistics (in brackets), which are automatically obtained by the style analysis algorithm according to the templates in Figs. 10(a)~10(g). For comparison, the stroke-based style analysis is performed again on the rendered results

Table 2. Painting Parameters Recommended by Our System Compared with the Ground Truth

| | Images | Stroke Visibility | Stroke Size | Stroke Orientation | Stroke Texture | Stroke Texture |
|--------------------------------|--------|----------------------|-------------|--------------------|----------------|----------------|
| | | $\alpha(\bar{S}(T))$ | $l(L(T))$ | $V(T)$ | $F_1(T)$ | $F_2(T)$ |
| Template | a | 0.95 (2.32) | XX | XX | XX | XX |
| | b | 0.10 (12.84) | 11 (138.92) | 0.71 | 0.38 | 18.87 |
| | c | 0.10 (7.01) | 31 (365.88) | 0.17 | 0.64 | 9.89 |
| | d | 0.10 (6.63) | 2 (31.33) | XX | XX | XX |
| | e | 0.10 (5.88) | 23 (288.71) | 0.10 | 0.41 | 4.41 |
| | f | 0.10 (8.54) | 26 (332.39) | 0.49 | 0.68 | 17.27 |
| | g | 0.10 (7.43) | 35 (402.64) | 0.16 | 0.10 | 16.33 |
| Results | j | 0.95 (2.43) | XX | XX | XX | XX |
| | k | 0.10 (12.18) | 12 (159.65) | 0.74 | 0.41 | 16.93 |
| | l | 0.10 (6.72) | 28 (349.23) | 0.13 | 0.69 | 9.27 |
| | m | 0.10 (6.88) | 1 (25.25) | XX | XX | XX |
| | n | 0.10 (5.21) | 23 (286.33) | 0.11 | 0.40 | 5.88 |
| | o | 0.10 (8.93) | 28 (348.97) | 0.45 | 0.61 | 16.47 |
| | p | 0.10 (5.98) | 35 (397.69) | 0.13 | 0.09 | 17.23 |
| Average Error (Parameters) (%) | | 5.0 | 4.2 | 8.8 | 6.2 | 7.2 |
| Average Error (Statistics) (%) | | 9.7 | 8.1 | 8.8 | 6.2 | 7.2 |

Note: Shown in the top part of the table are the painting parameters and style statistics obtained from the templates in Figs. 10(a)~10(g). Shown in the center part of the table are the painting parameters and style statistics obtained from the rendered results in Figs. 10(j)~10(p). Shown in the bottom part of the table are the average errors of painting parameters and style statistics, which are calculated from an extended experiment using 20 test images. For statistics not relevant to specific template painting, the corresponding cells are filled with “XX”.

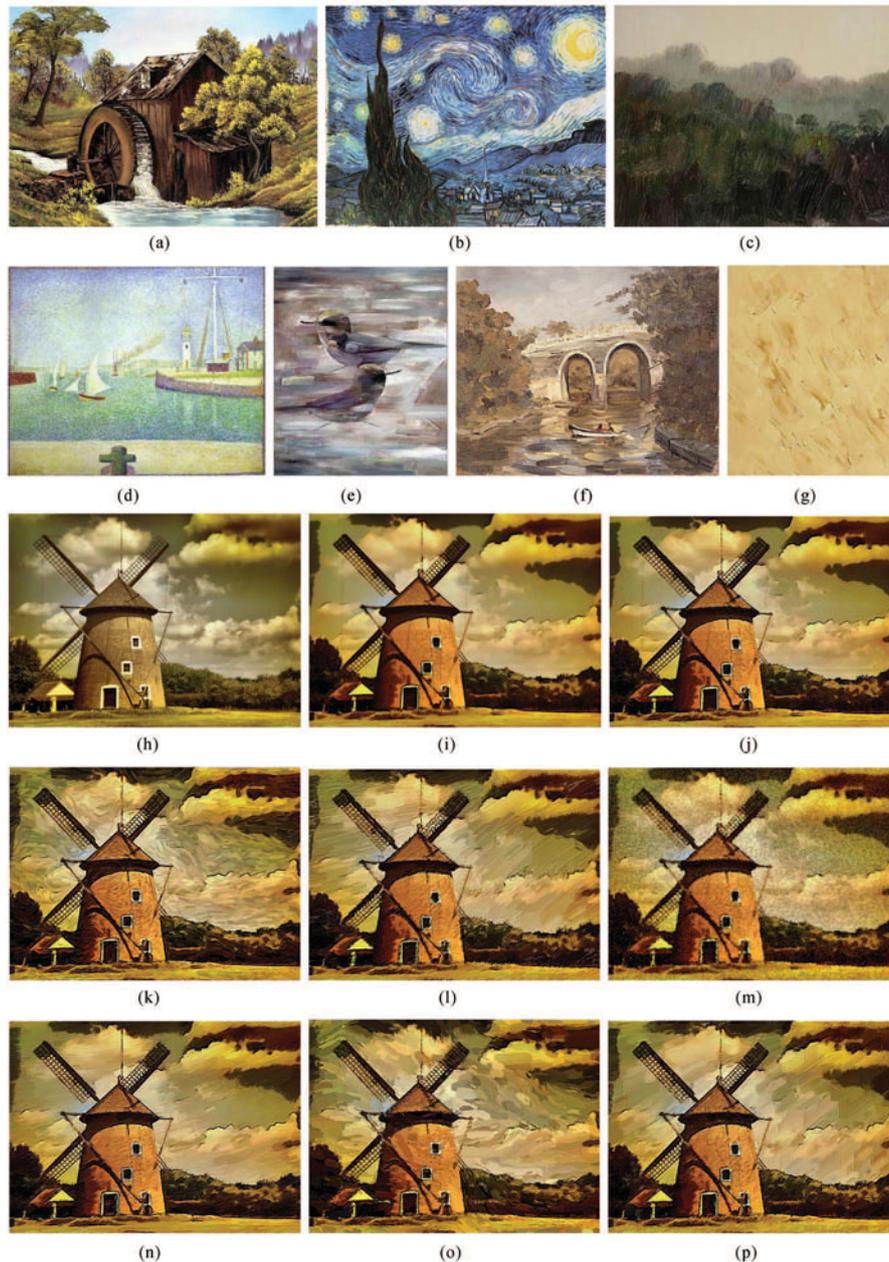


Fig.10. Experimental results. (a)~(g) Style templates. (h) Input photo. (i) Intermediate result after pre-processing. (j)~(p) Painting results corresponding to the templates (a)~(g).

in Figs.10(i)~10(o), and the automatically detected painting parameters and style statistics are shown in the second part of Table 2. It can be observed that the recommended painting parameters and style statistics from the templates are consistent with the detected painting parameters and style statistics from the rendered results. To confirm the accuracy, such experiments are performed for another 20 images to test the average error of our system. The relative errors of both painting parameters and style statistics are shown in the third part of Table 2, which are all under 10%.

To further verify whether the styles of the template are correctly imitated, the structural similarity index (SSIM)^[35], a widely used method for measuring the similarity between two images, along with a user evaluation is employed. For the SSIM investigation, we convert both template paintings and rendered results into gray images (the “L” channel extracted from the Lab color space), and adjust their brightness and contrast to a similar level. Then, according to the stroke confidence map of each template/result image, 2~4 most visible stroke patches are selected from the ad-

justed gray images for calculating SSIM indices. The averaged SSIM indices are listed in Table 3, where the first column shows the self-similarity between stroke patches taken from the same template painting, and columns 2~7 show the similarity between each template painting and each rendered result. It can be seen that the SSIM index for self-similarity ranges in $[0.23, 0.33]$, and among all SSIM indices obtained between template paintings and rendered results, the targeted template-result pairs have the dominant SSIM values as shown in the diagonal entries in the table. Taking the ratio between the SSIM index of the targeted template-result pair and the self-similarity SSIM index of the corresponding template, the relative similarity is computed and listed in the last column of the table. It can be seen that for all template paintings, the corresponding rendered results have achieved a relative similarity above 85%. This confirms that in terms of the structural similarity measure, the proposed style-oriented painterly rendering framework can produce images in similar stroke styles as indicated by template paintings.

An independent user evaluation is also performed to check the effectiveness of the proposed style-oriented painterly rendering framework. Two groups of randomly ordered images, including 10 template paintings and 10 rendered results correspondingly generated by our system, are presented to 20 amateur users (10 males and 10 females) who do not have specialized art knowle-

dge. Each participant is required to find a one-to-one match between the template and result images, according to their stroke-related artistic styles. Fig.11 shows the matching statistics, with 176 successful matches (88%) and 24 failed matches (12%). Although the artistic style of paintings is a subjective concept, the user evaluation confirms that the new style-oriented painterly rendering system can successfully simulate the stroke-related style features of many template paintings.

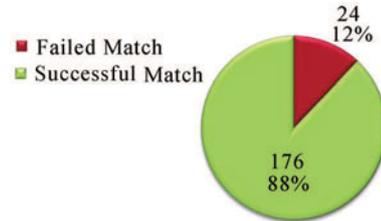


Fig.11. Evaluation of the visual appearance — statistics of user evaluations.

Comparison with Manually-Controlled Painterly Rendering Approach. We also compare our work with previous painterly rendering framework controlled by user interaction. Fig.12(a) is the original input photo; Figs. 12(b)~12(e) are the multi-style painting results of Hays^[6], which are painted using a set of manually specified painting parameters; Figs. 12(f)~12(j) show some similar painting results automatically produced by our

Table 3. Evaluation of the Visual Appearance — SSIM Indices Between the Template Paintings and Rendered Results in Fig.10

| Images (Template) | Self-Similarity | Similarity Between Template Paintings and Rendered Results | | | | | | Relative Similarity (%) |
|----------------------|-----------------|--|---------------|---------------|---------------|---------------|---------------|----------------------------|
| | | Result k | Result l | Result m | Result n | Result o | Result p | |
| b | 0.2878 | 0.2453 | 0.1199 | 0.1721 | 0.1942 | 0.1737 | 0.1902 | 85.2 |
| c | 0.2367 | 0.1974 | 0.2439 | 0.1786 | 0.1772 | 0.1242 | 0.1916 | 103.0 |
| d | 0.2654 | 0.1608 | 0.1179 | 0.2258 | 0.1893 | 0.1789 | 0.1816 | 85.1 |
| e | 0.3267 | 0.1765 | 0.1375 | 0.2006 | 0.3184 | 0.2114 | 0.2001 | 97.5 |
| f | 0.2432 | 0.1391 | 0.1791 | 0.1564 | 0.1889 | 0.2238 | 0.1701 | 92.0 |
| g | 0.2887 | 0.1423 | 0.1232 | 0.1919 | 0.2410 | 0.1982 | 0.2662 | 90.8 |

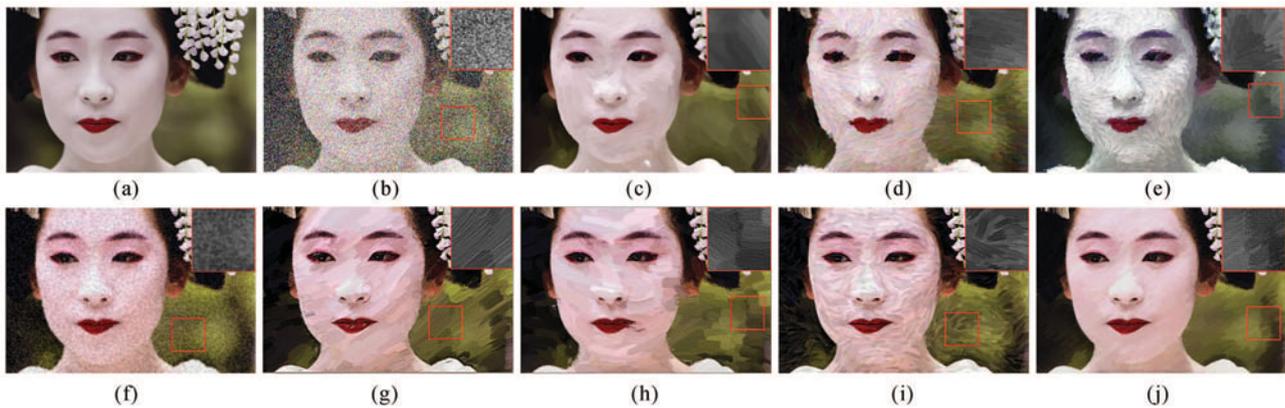


Fig.12. Comparison with the painting results of Hays^[6]. (a) Input image. (b)~(e) Painting results of Hays^[6] based on user-specified painting parameters. (f)~(j) Painting results of our method based on style templates.

system based on style templates. For better comparison, we select some typical stroke patch, and zoom the gray version in top right corner (as highlighted in the red box). From the comparison it can be observed that our new method can produce a variety of painting results with different stroke-based styles, and by specifying the style template it avoids tedious adjustment of tuning parameters.

7 Discussion of Failure Cases

Failures of our system can occur in two cases: 1) the strokes in the template are heavily smeared; 2) when the template painting contains a significant amount of threadlike semantic structures, as the strokes may be wrongly detected. The stroke detection results are not tractable in these two cases, and the misrecognition is particularly misleading for realist paintings. The first case is easy to understand, and a failure example is presented specific to the second case. As shown in Fig.13, the comb and the table top are misrecognized as stroke features in the stroke confidence map. The misrecognized stroke areas then give wrong style statistics, indicating high stroke visibility, which is opposite to the realist template. A practical way to overcome this problem is to mask the threadlike semantic content before the automatic stroke detection.

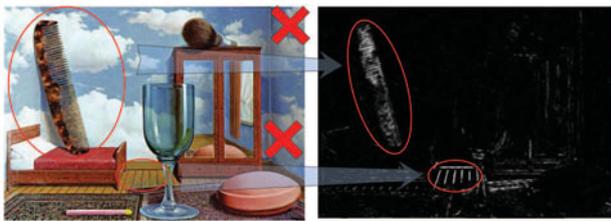


Fig.13. Failed example.

8 Conclusions

This paper presented a novel method that automatically extracts stroke features from paintings and analyze stroke-related artistic styles. The new style analysis is compatible with existing stroke-based painterly rendering algorithms, and can be readily connected to standard painterly rendering platforms as a separate module to automatically initialize stroke-related painting parameters. The effectiveness of the new system was examined and verified through a number of examples, comparisons and user studies. The new style-oriented painterly rendering framework supports the simulation of a variety of stroke-related artistic styles, and is not content-sensitive to the input photo or template painting. Compared with previous user-controlled painterly rendering approaches, the new method is

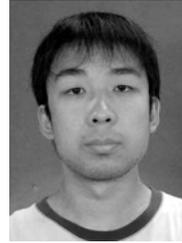
highly efficient and significantly reduces the need for tedious parameter adjustment and trial paintings.

Style transfer by painting features is a challenging problem because it is quite difficult to simulate professional painters' skills. This work focuses mainly on the stroke properties, and many of other aspects are left for further research, such as the overall composition, exaggerated shapes, and the harmony and variety stroke orientation. These aspects are equally important and should be followed up in future research to fully achieve style transfer.

References

- [1] Hertzmann A, Jacobs C E, Oliver N *et al.* Image analogies. In *Proc. the 28th SIGGRAPH*, Aug. 2001, pp.327-340.
- [2] Wang B, Wang W, Yang H *et al.* Efficient example-based painting and synthesis of 2D directional texture. *IEEE Trans. Visualization and Computer Graphics*, 2004, 10(3): 266-277.
- [3] Lee H, Seo S, Ryoo S, Yoon K. Directional texture transfer. In *Proc. the 8th Int. Symp. Non-Photorealistic Animation and Rendering*, June 2010, pp.43-48.
- [4] Litwinowicz P. Processing images and video for an impressionist effect. In *Proc. the 24th SIGGRAPH*, Aug. 1997, pp.407-414.
- [5] Hertzmann A. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. the 25th SIGGRAPH*, Aug 1998, pp.453-460.
- [6] Hays J, Essa I. Image and video based painterly animation. In *Proc. the 3rd Int. Symp. Non-Photorealistic Animation and Rendering*, June 2004, pp.113-120.
- [7] Kagaya M, Brendel W, Deng Q Q *et al.* Video painting with space-time-varying style parameters. *IEEE Transactions on Visualization and Computer Graphics*, 2011, 17(1): 74-87.
- [8] Huang H, Zhang L, Fu T N. Video painting via motion layer manipulation. *Computer Graphics Forum*, 2010, 29(7): 2055-2064.
- [9] Lee H, Lee C H, Yoon K. Motion based painterly rendering. *Computer Graphics Forum*, 2009, 28(4): 1207-1215.
- [10] Zeng K, Zhao M, Xiong C, Zhu S C. From image parsing to painterly rendering. *ACM Transactions on Graphics (TOG)*, 2009, 29(1): Article No. 2.
- [11] Lyu S, Rockmore D, Farid H. A digital technique for art authentication. In *Proc. the National Academy of Sciences of the United States of America*, 2004, 101(49): 17006-17010.
- [12] Li J, Wang J Z. Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Transactions on Image Processing*, 2004, 13(3): 340-353.
- [13] Yelizaveta M, Chua T S, Ramesh J. Semi-supervised annotation of brushwork in paintings domain using serial combinations of multiple experts. In *Proc. the 14th Annual ACM Int. Conf. Multimedia*, Oct. 2006, pp.529-538.
- [14] Hertzmann A. Fast paint texture. In *Proc. the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, June 2002, pp.91-96.
- [15] Kalogerakis E, Nowrouzezahrai D, Breslav S, Hertzmann A. Learning hatching for pen-and-ink illustration of surfaces. *ACM Transactions on Graphics (TOG)*, 2012, 31(1): 1-10.
- [16] Melzer T, Kammerer P, Zolda E. Stroke detection of brush strokes in portrait miniatures using a semi-parametric and a model based approach. In *Proc. the 14th International Conference on Pattern Recognition*, Aug. 1998, pp.474-476.
- [17] Johnson C R, Hendriks E, Berezhnoy I J *et al.* Image processing for artist identification. *IEEE Signal Processing Magazine*, 2008, 25(4): 37-48.

- [18] Gabor D. A new microscopic principle. *Nature*, 1948, 161(4098): 777-778.
- [19] Turner M R. Texture discrimination by Gabor functions. *Biological Cybernetics*, 1986, 55(2/3): 71-82.
- [20] Fogel I, Sagi D. Gabor filters as texture discriminator. *Biological Cybernetics*, 1989, 61(2): 103-113.
- [21] Jain A K, Farrokhnia F. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 1991, 24(12): 1167-1186.
- [22] Daugman J G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 1985, 2(7): 1160-1169.
- [23] Kruijzinga P, Petkov N. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 1999, 8(10): 1395-1407.
- [24] Huang H, Fu T N, Li C F. Painterly rendering with content-dependent natural paint strokes. *The Visual Computer*, 2011, 27(9): 861-871.
- [25] Huang H, Zang Y, Li C F. Example-based painting guided by color features. *The Visual Computer*, 2010, 26(6/8): 933-942.
- [26] Cortes C, Vapnik V. Support-vector networks. *Machine Learning*, 1995, 20(3): 273-297.
- [27] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(5): 603-619.
- [28] Tsai W H. Moment-preserving thresholding: A new approach. *Computer Vision, Graphics and Image Processing*, 1985, 29(3): 377-393.
- [29] Huang N E, Shen Z, Long S R, Wu M C, Shih H H, Zheng Q, Yen N C, Tung C C, Liu H H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 1998, 454(1971): 903-91.
- [30] Gao Y, Li C F, Ren Bo, Hu S M. View-dependent multiscale fluid simulation. *IEEE Transactions on Visualization and Computer Graphics*, 2013, 19(2): 178-188.
- [31] Tang Y, Shi X Y, Xiao T Z, Fan J. An improved image analogy method based on adaptive CUDA-accelerated neighborhood matching framework. *The Visual Computer*, 2012, 28(6): 743-753.
- [32] Li X Y, Gu Y, Hu S M, Martin R. Mixed-domain edge-aware image manipulation. *IEEE Transactions on Image Processing*, 2013, 22(5): 1915-1925.
- [33] Zhang S H, Li X Y, Hu S M, Martin R. Online video stream abstraction and stylization. *IEEE Transactions on Multimedia*, 2011, 13(6): 1286-1294.
- [34] Wang X H, Jia J, Liao H Y, Cai L H. Affective image colorization. *Journal of Computer Science and Technology*, 2012, 27(6): 1119-1128.
- [35] Wang Z, Bovik A C, Sheikh H R, Simoncelli E P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 2004, 13(4): 600-612.



Yu Zang is currently a Ph.D. candidate of School of Electronics and Information Engineering, Xi'an Jiaotong University, China. He received the Bachelors degree in signal and information processing from Xi'an Jiaotong University in 2008. His main research interests include image and video processing.



Hua Huang is a professor in the School of Computer Science and Technology, Beijing Institute of Technology. He received his B.S. degree and Ph.D. degree in signal and information processing from Xi'an Jiaotong University in 1996 and 2006, respectively. His main research interests include image and video processing, computer graphics, pattern recognition, and machine learning.



Chen-Feng Li is currently a lecturer at the College of Engineering of Swansea University, UK. He received his B.Eng. and M.Sc. degrees from Tsinghua University, China, in 1999 and 2002 respectively and received his Ph.D. degree in computational engineering from Swansea University, UK, in 2006. His research interests include computer graphics and numerical analysis.