Creating Watercolor Style Images Taking Into Account Painting Techniques

Henry Johan
The University of Tokyo
henry@is.s.u-tokyo.ac.jp

Ryota Hashimoto
Namco Ltd
hasimoto@nis-lab.is.s.u-tokyo.ac.jp

Tomoyuki Nishita
The University of Tokyo
nis@is.s.u-tokyo.ac.jp

Abstract

Research for creating impressive images like paintings has attained more and more importance because of the recent growth in processing images. Among many painting styles, watercolor style gives a strong impression because of its thin colors and its soft appearance. This paper proposes a method to create watercolor style images from input images, for instance photographs. In contrast with the previously proposed watercolor methods that focuses only on simulating the effects of watercolor medium such as the mottled appearance and the color glazing, the proposed method also considers how the watercolor paintings are painted in order to simulate artists' painting techniques. In the proposed method, artists' painting techniques are represented using *painting rules*. Several painting rules are provided, in addition, user can specify their own painting rules. These painting rules are used for generating strokes to paint the objects. Painting techniques are simulated by first detecting the objects in the input image and generating strokes for each object according to the painting rules. The properties of watercolor medium are simulated by approximating each stroke with sampling points and diffuses their color to nearby pixels. Using the proposed method, a user can interactively create watercolor style images.

Keywords: Watercolor painting, Painting techniques, Non-Photorealistic Rendering.

描画技法を考慮した水彩画風画像の生成

ヘンリー・ジョハン
東京大学

橋本　良太
株式会社ナムコ

西田　友是
東京大学

概要

近年，画像の加工を行うことが身近になりつつあり，絵画のような印象的な画像を生成する研究分野の重要性がますます高まっている．絵画の中でも水彩は淡い色調で柔らかい雰囲気をもっており，印象に残りやすい画風であると言える．本論文では，写真などの画像を入力とし，水彩風に変換された画像を生成する手法を提案する．従来法では，水彩顔料の性質による色の塗りむらや色の重なりのような特徴をシミュレートすることに焦点が置かれる．それに対して，提案法では画家の描画技法をシミュレートするために水彩画はどうやって描かれるのかも考慮する．提案法では描画規則を用いて画家の描画技法を表す．幾つかの描画規則が用意され，またユーザーは独自の描画規則を指定することができる．これらの描画規則を用いて物体を描くためのストロークを生成する．描画技法をシミュレートするために，初めに入力画像内の物体を検出し，そして描画規則に従って各物体を描くためのストロークを生成する．生成されたストロークをサンプリング点で近似し，サンプリング点の色を周囲に拡散させることによって水彩顔料による特徴をシミュレートする．提案法を用いることで，ユーザーは水彩画風画像をインタラクティブに生成することが可能である．

キーワード: 水彩画，描画技法，非写実的レンダリング.

1 Introduction

Recently, computer users can easily produce digital images thanks to the growth in the field of computer graphics and digital cameras. As a result, the need for processing digital images has also increased. Among the exisiting image processing techniques, the technique to convert an input image into artistic style image has gained much attention lately. Artistic rendering, which is to create images that simulate the style of artists, is one of the research field in Non-Photorealistic Rendering (NPR). There is a variety of styles including oil painting style, watercolor painting style, pen-and-ink style and so on.

Among the painting styles, watercolor is one of the most impressive and attractive style because of its thin colors and its soft appearance. The behavior of watercolor pigments in water exhibits a variety of distinctive effects (i.e. the mottled appearance) which make watercolor paintings so attractive. Artists often exploit these effects when creating watercolor paintings. The properties of watercolor medium and the painting techniques of artists make watercolor paintings much more impressive and attractive. However, previous methods to create watercolor style images put emphasis only on the properties of watercolor medium without considering the painting techniques.

We propose a method to create watercolor style images using photographs or images as input, considering both the watercolor medium properties and the artists' painting techniques. To simulate the painting techniques, the proposed method uses the *painting rules* for generating strokes. Strokes are approximated with sampling points and their colors are diffused to nearby pixels in order to simulate the behaviors of watercolor medium. The proposed method can create watercolor style images fast enough. As a result, a user can interactively create watercolor images.

The rest of this paper is organized as follows. Section 2 presents a brief review of related work. Section 3 explains the properties of watercolor paintings to clarify the goal of this paper and describes about the overview of the proposed method. In Sections 4, 5, and 6, the proposed method for creating watercolor-style images is described. We show some example images in Section 7, and finally, conclusion and future work are discussed in Section 8.

2 Related Work

Our work is classified into the field of NPR. NPR techniques create more impressive images than the photorealistic rendering does, thus a promising technique to be used in the computer games, movies, and other entertainment fields which tend to place emphasis on the visual impact. A number of NPR methods have been proposed including silhouette-based methods [5, 14], pen-and-ink methods [8, 20], colored pencil drawing [21], wax crayons [19], stylized rendering [2, 4, 7], mosaics creation [6, 9], and so on.

One of the major interests of NPR researchers is brush stroke-based rendering, which is to create images or animations of an oil-painting-style or other artistic styles that use a brush. Meier [18] proposed a method that renders painterly style animations with temporal coherence by modeling surfaces as 3D particle sets. Litwinowicz [16] created impressionist style animations using an optical flow approach to maintain the coherence between frames. Hertzmann [10] used a series of cubic B-splines to represent strokes, and created painterly images by using a coarse-to-fine painting approach. In these two methods [16, 10], the directions of the strokes are determined by using the directions perpendicular to the gradient of the luminance of the input image. Hertzmann [11] proposed a method to render stroke-based paintings efficiently.

Watercolor paintings can also be categorized into stroke-based paintings but differs from them in the way that watercolor exhibits its own textures and patterns which are exploited by artists. Considering these points, Curtis *et al.* [3] proposed a watercolor painting method using fluid simulation of watercolor pigment in water. Their method creates beautiful images, but takes a long computation time. Lum and Ma [17] created watercolor inspired images from 3D scenes using a new lighting model and generated wash-like textures.

The proposed method creates watercolor-style images from images (e.g. photographs) considering the traits of watercolor paintings. The results of the proposed method was first reported in our previous paper [13]. The method presented by Curtis *et al.* [3] deals with only the properties of watercolor medium and does not consider the painting techniques which artists use when creating watercolor paintings. Furthermore, this method has high computational cost. On the other hand, our method create the output images considering these two points, the properties of watercolor medium and the painting techniques. Moreover, the proposed method is fast.

3 Creating Watercolor Style Images

We categorize the features of watercolor paintings into two categories, *pigment-based features* and *artist-based features*. *Pigment-based features* are a collection of features arisen from the properties of watercolor medium such as watercolor glazing and its mottled appearance. *Artist-based features* are a collection of features arisen from the painting techniques of artists. The goal of the proposed

method is to create watercolor style images considering these features allowing a user to interactively control the image generation process.

3.1 Pigment-based Features

Watercolor pigments are semi-transparent, the apparent color after painting a stroke becomes the mixture of the color of a new pigment layer resulting from the current stroke and ones of the underlying pigment layers resulting from the formerly drawn strokes. The behavior of watercolor pigments in water exhibits a variety of distinctive effects. These effects are fully exploited by artists to create impressive paintings.

These pigment-based features are well categorized by Curtis *et al.* [3] and they simulated these effects using fluid simulation and the Kubelka-Munk color model [15]. Although their approach well captured the various watercolor effects, their method is very time-consuming and thus not suitable for interactive image editing systems. Considering these points, the proposed method adopted a simplified model to simulate the (pigment-based) watercolor effects and deals only with the important ones to reduce the computational cost. We consider the following two effects of watercolor:

- *Mottled appearance*
- Simplified *color glazing*.

We simulate these watercolor effects by diffusing the colors of sampling points to nearby pixels considering the diffuse direction and the image features (i.e. edges).

3.2 Artist-based Features

In contrast with pigment-based features, artist-based features have not been examined thoroughly. Artists paint objects by choosing the most effective painting techniques to express the properties of objects, such as:

- An object like sky is painted with much water to create smooth gradations of thin colors.
- Paint objects several times starting from a rough painting and adding the details successively. For instance, the leaves of trees are painted twice by first roughly painting the overall shapes of the leaves, followed by the sparse placement of strokes to express the complexity of the textures of the leaves of the trees
- In many cases, strokes are put parallel to the object boundaries (or silhouettes).

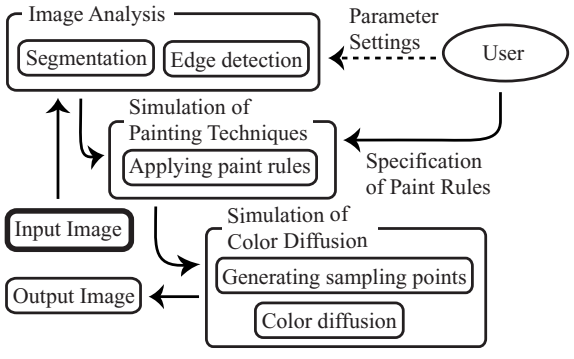


Figure 1: Overview of the proposed method.

- In general, large regions are painted coarsely with large-sized strokes and small regions are painted finely with small-sized strokes.

In addition to these techniques, many artists leave the regions near the boundary of the paper unpainted.

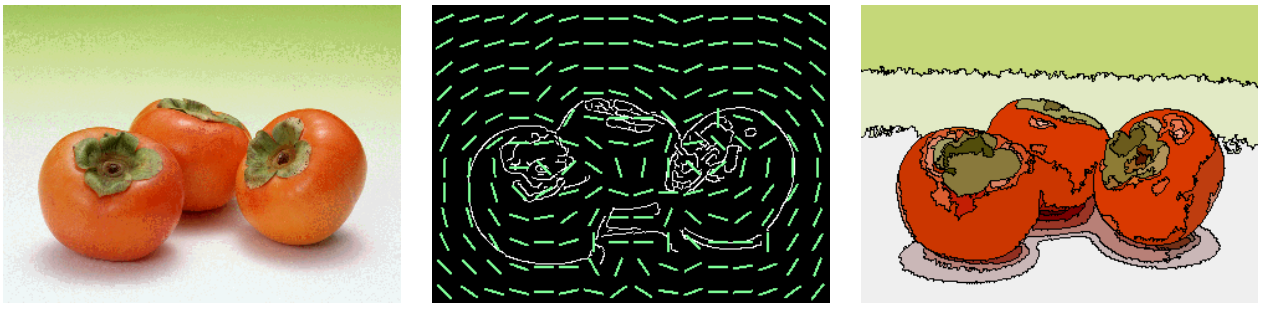
In this paper, we simulate these techniques by first detecting the objects in the input image and then applying the *painting rules* that define the appropriate painting method for each category of objects. The painting rules are specified with the assistance of the user. Most painting rules are reusable and the user burden is not so large. The system determines the most suitable painting rule for each object in the input image, using the result of the analysis of the input image.

3.3 Overview of the Proposed Method

The proposed method consists of three steps, *analysis of the input image*, *simulation of the painting techniques*, and *simulation of color diffusion* (Figure 1). In the first step, the input image is divided into several regions and information to be used in the later process is gathered. In the second step, strokes are generated by applying the painting rules to the regions in the input image, that is the painting techniques are considered. In the final step, strokes are rendered by first selecting sampling points which approximate the strokes and then diffusing their colors to nearby pixels.

4 Analysis of the Input Image

In this step, the properties of the input image are analyzed. The results of the analysis of the input image are used in the rest of the processes of the proposed method. This process is largely automated, but the user can tune the parameters to produce more satisfiable results. We analyze the input image by performing *edge detection* and *image segmentation*.



(a) (b) (c)

Figure 2: The result of image analysis, (a) an input image, (b) the generated vector field of stroke directions, and (c) the result of image segmentation.

4.1 Edge Detection

Edge detection is to find the boundary of objects. The edges in the input image are detected by using the method proposed by Canny [1]. As described before, since artists put strokes parallel to the object boundaries (or silhouettes), we use the result of the edge detection to generate a vector field which specifies the default direction of strokes (Figure 2(b)).

The vector field is generated using an algorithm similar to the method of Hausner [9]. After the edges are detected, a distance image whose pixels contain the distance to the nearest edge is created using the hardware-accelerated method of Hoff [12] that creates the Voronoi diagrams by drawing cones with their apexes at the edge pixels on the screen. The vector field of stroke directions that follow the orientation of the nearby edges is created by calculating the direction perpendicular to the gradient of the distance image.

4.2 Image Segmentation

Image segmentation is to partition an image into meaningful regions (Figure 2(c)). In many cases, each region is closely related to the object in the image, so we use this technique to detect the objects in the input image. We treat a single region as a minimum unit in which the painting rule is applied in the later process. The segmentation process is performed based on the color similarity in order to divide the input image into meaningful regions (objects). After segmentation, the average color and the area of each resulting region, which are used in the later process, are calculated. The area of a region is calculated as the number of the pixels in the region divided by the total number of the pixels in the input image, thus the value is ranging from 0.0 to 1.0.

The input image is segmented using a simple approach which generates regions by connecting neighboring pixels of similar colors, that is a seed pixel is selected randomly from the input image and the region is grown by successively connecting the neighboring pixels whose color differences to the seed pixel are below the user-specified threshold. The input image is segmented by repeating this process until all pixels are classified into some regions. The difference between two colors is measured in the HSV color space by computing the Euclidean distance between the two colors in the HSV color space cone. We also tested the RGB and LUV color spaces to compute the color difference but we found that HSV color space gives the best result.

However, due to the simplicity of the segmentation algorithm, there are some cases where many small regions are generated. In this case, two neighboring regions with similar average colors are merged into a single larger region. Small regions resulting from the segmentation process are then regarded as showing the details of the objects that is to be used in the detailed paint.

5 Simulation of Painting Techniques

Artists use many techniques to create impressive and attractive watercolor paintings. How to paint an object is largely depend on the artist's sense and the property of the object that are both very difficult to process theoretically. Considering these issues, we allow user assistance in determining the painting rules for painting the objects (regions resulting from the segmentation process).

5.1 Definition of Painting Rules

A painting rule consists of the following two parts:

- *A condition* that determines the property of the region to which the painting rule is applied.
- *A painting style* that determines how to paint the region, that is how to generate strokes in the region.

The painting rules are applied to the regions of the input image, one rule per region. Each painting rule has a priority which is used to choose the most appropriate rule when a region satisfies the conditions of multiple rules.

5.1.1 Conditions

In our method, we use two types of conditions for determining the region to which the painting rule is applied.

One type of the conditions is to choose regions based on their average colors and areas. In this case, the user specifies the threshold values for the average color (color (h_t, s_t, v_t) and threshold t_{color}) and area (minimum A_{min} and maximum A_{max}). The color is specified using the HSV color space. A region with its area A and its color (h, s, v) satisfies the condition of the rule when $A_{min} \leq A \leq A_{max}$ and $d < t_{color}$ where d is the color difference between (h, s, v) and (h_t, s_t, v_t). Defining the condition of the painting rule in this way allows the painting rule to be reused in the future.

Another type of the conditions is that the user directly specifies the regions to which the painting rule should be applied. This type of condition is used when the user wants to paint certain regions in a specific painting style.

5.1.2 Painting Styles

A painting style is defined as a set of parameters for controlling the properties of the generated strokes. The properties of strokes are defined as direction, width, length, average interval between strokes, and maximum fluctuation of stroke direction. The direction of strokes can be determined in two ways, (1) a constant direction θ $(0 \leq \theta \leq 2\pi)$ specified by the user, (2) the direction of the vector field that follows the edge direction (see Section 4). The width, the length, and the interval between strokes are specified in the unit of pixel. The fluctuation parameter is specified using a value between 0 and $1/2\ \pi$.

We allow the user to specify multiple sets of parameters for a painting style to deal with the cases when the user wants to paint an object several times. For instance, we can specify three sets of parameters for the first paint (base paint) for drawing the object roughly, the second paint for expressing the shape of the objects, and the final paint for adding the small details by painting the small regions resulting from image segmentation process.

5.2 Generation of Strokes Based on the Painting Rules

The artist-based features are simulated by generating strokes in each region considering the parameters specified in the painting styles of the applied painting rule.

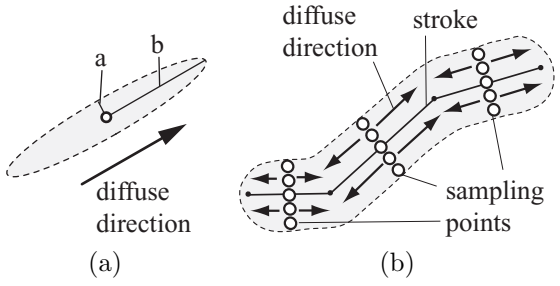


Figure 3: (a) Diffuse area of a sampling point and (b) approximation of a stroke with sampling points.

Each stroke is represented as a polyline (or a set of line segments) and is generated as follows. First, a starting point is selected randomly, then the stroke is extended successively at a specified distance in the direction according to the painting rule. The stroke generation terminates if its length exceeds the specified length or it intersects the region boundary. The inhibition area is set around the generated stroke considering the interval between strokes parameter of the painting rule. The subsequent strokes are generated avoiding the inhibition area.

The color of the stroke is set to the color at the starting point in the input image. However, in order to avoid unnatural color, we use the average color of the pixels in the region as the color of the stroke if the difference between the color at the starting point and the average color exceeds a user given threshold value.

6 Simulation of Color Diffusion

In order to render the generated strokes considering the pigment-based features, the proposed method first approximates the strokes with sampling points and then diffuses their colors to nearby pixels considering the diffuse direction and the features (edges) of the input image. The final color of an pixel is computed by weighted averaging the colors that reached that pixel and the color of the paper.

6.1 Generating Sampling Points for Strokes

After strokes are generated, the proposed method approximates them with sampling points. Diffuse area of a sampling point is roughly the shape of an ellipse (Figure 3(a)). A stroke is approximated by placing sampling points on a cross section at the midpoint of every line segment in the stroke at a specified width (Figure 3(b)).

The diffuse direction of a sampling point is set to be parallel to the line segment. In our experiment, the length

(a) (b)

Figure 4: (a) The generated sampling points for the input image in Figure 2(a), and (b) the final result after diffusing the colors at the sampling points.

of the minor axis a is set to two pixels, and the length of the major axis b is set to half of the length of the corresponding line segment.

Figure 4(a) shows the generated sampling points (black points) for the input image in Figure 2(a). The closed polyline in Figure 4(a) shows one of the generated strokes and the black points inside it are the associated sampling points.

6.2 Rendering Strokes Using Color Diffusion

To render strokes, the proposed method diffuses colors of the sampling points to nearby pixels. The basic idea of the color diffusion process is to diffuse two types of weights: *weight of color* and *weight of shape*. The weight of color is used to calculate the color of each pixel in the output image, which is affected by the roughness of the paper. The weight of shape determines the diffuse area which is affected by the distance from the sampling point and the roughness of the paper.

Both types of weights are influenced by the roughness of the paper. The roughness of the paper is modeled as a height field and is created using the method proposed by Curtis *et al.* [3]. When the weights at pixel p are diffused to its neighboring pixel p', they are scaled by the factor of $(1 + (height(p) - height(p'))$ where $height(p)$ represents the height of paper at p.

The weight of shape is attenuated exponentially based on the distance from the sampling point in addition to the effects of the paper. The diffuse direction is considered by calculating the distance from the sampling point to a pixel after scaling the component perpendicular to the diffuse direction by the factor of b/a in Figure 3(a). The decay d of the stroke is calculated using the equation

$$1 - \left(\frac{w_{limit}}{w_0}\right)^a, \qquad (1)$$

where w_0 and w_{limit} represent the initial weight of shape and the threshold of diffusion, respectively. A special decay is considered for the weight of shape when the weight is diffused to a pixel p across a region boundary and the difference between the color of p and the color of the sampling point exceeds the color threshold used in segmentation.

A stroke is rendered by performing the above process for all sampling points that approximate the stroke. The shape of a stroke is defined to be the union of diffuse area of all sampling points. The weight of color of a stroke at a pixel is defined as the maximum weight of color values that reached the pixel. The output image is created by weighted averaging the colors of all strokes and the paper. Figure 4(b) shows the resulting image after performing color diffusion.

7 Results

The results of the proposed method are shown in Figure 5. The images in the left column are the input images and the images in the right column are the corresponding watercolor style images.

Table 1 shows the examples of painting rules. Each painting rule defines that an object is going to be painted twice. These rules were used to generate the watercolor style of the sunflower image. For the sea example, the user specified the use of thin brush for painting the details of the sea. Moreover, the stroke direction was set to almost horizontal for painting the details. For the rest of the regions, default brush and default vector field were used for generating the strokes. In the flower garden example, the technique where artists intentionally leave some parts unpainted is simulated by controlling the interval between strokes.

In all the examples, we can observe the shape of the strokes. We can also see some granulation and glazing effects in these examples. These effects are achieved be-

Table 1: Painting rules used in the sunflower example.

Painting rules

Priority	Name	Condition	Painting styles
1	Sky	Sky	Base paint, Sky
2	Grass	Grass	Base paint, Random paint
3	Default	Default	Base paint, Default paint

Conditions

Name	Area thresholds		Color thresholds		Description
	Minimum	Maximum	Color (h, s, v)	Threshold	
Sky	0.2	1.0	$(4/3\ \pi, 0.0, 1.0)$	1.0	Large region of washy blue
Grass	0.0	1.0	$(2/3\ \pi, 0.3, 0.5)$	0.3	Dark green region
Default	—	—	—	—	Default condition

Painting styles

Name	Direction	Width	Length	Interval	Fluctuation
Base paint	constant, 0	20	20	20	0
Sky	constant, 0	10	30	30	$1/3\ \pi$
Random paint	constant, $1/3\ \pi$	4	12	20	$2/5\ \pi$
Default paint	vector field	4	10	12	0

cause we take into account the roughness of the paper when performing the diffusion process. Since we consider the color of the paper when generating the images, the resulting watercolor images have washy appearance. The size of the input images are 400×300 pixels. The computational times are around six seconds using a machine with 3 GHz Pentium 4.

8 Conclusion and Future Work

We have proposed a method to create watercolor style images using for instance photographs as input. The proposed method simulated both the effects arisen from the watercolor pigments in water and the painting techniques that artists use to create impressive and attractive images.

To simulate the painting techniques, we use painting rules for generating strokes. Painting rules determine the properties of strokes, such as width, length, interval, direction, and so on, for painting regions. The generated strokes are then approximated with sampling points and their colors are diffused to nearby pixels in order to simulate the behaviors of watercolor medium. The proposed method is fast and thus can be applied in interactive image editing system.

There are some possibilities for extending the proposed method.

- Developing a learning-based approach for automatically selecting the painting rules from training data. This may be a help for the user to reduce the time for specifying the painting rules.

- Extending the proposed method to deal with animations will be useful. To create nice watercolor animations, the coherency between frames of the animation should be considered.

References

[1] John Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679-698, 1986.

[2] John P. Collomosse and Peter M. Hall, "Cubist Style Rendering of Photographs," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 9, No. 4, pp.443-453, 2003.

[3] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin, "Computer-Generated Watercolor," *Proceedings of SIGGRAPH 97*, pp. 421-430, 1997.

[4] Doug DeCarlo and Anthony Santella, "Stylization and Abstraction of Photographs," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2002)*, Vol. 21, No. 3, pp. 769-776, 2002.

[5] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella, "Suggestive Contours for Conveying Shape," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 848-855, 2003.

[6] Yoshinori Dobashi, Toshiyuki Haga, Henry Johan, and Tomoyuki Nishita, "A Method for Creating Mosaic Images Using Voronoi Diagrams," *Proceedings of Eurographics 2002 Short Presentations*, pp.341-348, 2002.

[7] Paul E. Haeberli, "Paint by Numbers: Abstract Image Representations," *Computer Graphics (Proceedings of SIGGRAPH 90)*, Vol. 24, No. 4, pp. 207-214, 1990.

[8] Toshiyuki Haga, Henry Johan, and Tomoyuki Nishita, "Animation Method for Pen-and-Ink Illustrations Using Stroke Coherency," *Proceedings of CAD/Graphics 2001*, pp. 333-343, 2001.

[9] Alejo Hausner, "Simulating Decorative Mosaics," *Proceedings of SIGGRAPH 2001*, pp. 573-580, 2001.

[10] Aaron Hertzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes," *Proceedings of SIGGRAPH 98*, pp. 453-460, 1998.

[11] Aaron Hertzmann, "Fast Paint Texture," *Proceedings of Non-Photorealistic Animation and Rendering 2002*, pp. 91-96, 2002

[12] Kenneth E. Hoff III, Tim Culver, John Keyser, Ming Lin and Dinesh Manocha, "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware," *Proceedings of SIGGRAPH 99*, pp. 277-286, 1999.

[13] Henry Johan, Ryota Hashimoto, and Tomoyuki Nishita, "Creating Watercolor Style Images Using Painting Rules and Color Diffusion," *Proceedings of NICOGRAPH International 2004*, pp. 91-96, 2004.

[14] Robert D. Kalnins, Philip L. Davidson, Lee Markosian and Adam Finkelstein, "Coherent Stylized Silhouettes," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 856-861, 2003.

[15] Paul Kubelka, "New Contributions to to the Optics of Intensely Light-Scattering Material, Part II: Non-Homogeneous Layers," *Journal of the Optical Society of America*, Vol. 44, No. 4, pp. 330-335, 1954.

[16] Peter Litwinowicz, "Processing Images and Video for an Impressionist Effect," *Proceedings of SIGGRAPH 97*, pp. 407-414, 1997.

[17] Eric B. Lum and Kwan-Liu Ma, "Non-Photorealistic Rendering Using Watercolor Inspired Textures and Illumination," *Proceedings of Pacific Graphics 2001*, pp. 322-331, 2001.

[18] Barbara J. Meier, "Painterly Rendering for Animation," *Proceedings of SIGGRAPH 96*, pp. 477-484, 1996.

[19] Dave Rudolf, David Mould, and Eric Neufeld, "Simulating Wax Crayons," *Proceedings of Pacific Graphics 2003*, pp. 163-172, 2003.

[20] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin, "Interactive Pen-and-Ink Illustration," *Proceedings of SIGGRAPH 94*, pp. 101-108, 1994.

[21] Saeko Takagi, Masayuki Nakajima, and Issei Fujishiro. "Volumetric Modeling of Artistic Techniques in Colored Pencil Drawing," *Proceedings of Pacific Graphics 1999*, pp. 250-258, 1999.

Sunflower

Sea

Flower garden

Figure 5: Results: (left) input images, (right) corresponding watercolor style images.