Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Example based painting generation*

GUO Yan-wen[†1,2], YU Jin-hui[1], XU Xiao-dong[1], WANG Jin[1], PENG Qun-sheng[1]

([1]*State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, China*)

([2]*The National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*)

[†]E-mail: ywguo@cad.zju.edu.cn

Received Apr. 7, 2006; revision accepted Apr. 19, 2006

Abstract: We present an approach for generating paintings on photographic images with the style encoded by the example paintings and adopt representative brushes extracted from the example paintings as the painting primitives. Our system first divides the given photographic image into several regions on which we synthesize a grounding layer with texture patches extracted from the example paintings. Then, we paint those regions using brushes stochastically chosen from the brush library, with further brush color and shape perturbations. The brush direction is determined by a direction field either constructed by a convenient user interactive manner or synthesized from the examples. Our approach offers flexible and intuitive user control over the painting process and style.

Key words: Non-photorealistic rendering (NPR), Van Gogh, Painting, Grounding, Brush
doi:10.1631/jzus.2006.A1152 **Document code:** A **CLC number:** TP390

INTRODUCTION

Non-photorealistic rendering (NPR) is a domain of computer graphics that has received more and more attention in the past decade. Different from realistic rendering, NPR concentrates on communicating the main context of an image and explores the rendering effect of the scene with the artistic styles, rather than seeking subtle geometry details and naturally physical attribute.

Many research efforts were dedicated to simulating the traditional media (Hertzmann *et al*., 2001; 2002; Hertzmann, 1998; Hamel and Strothotte, 1999; Jodoin *et al*., 2002; Kalnins *et al*., 2002; Freeman *et al*., 2003; Wang B. *et al*., 2004; Meier, 1996; Nehab and Velho, 2002; Shiraishi and Yamaguchi, 2000; Litwinowicz, 1997), which can be divided mainly into two categories: model based and image based.

The model based NPR techniques (Hamel and Strothotte 1999; Jodoin *et al*., 2002) offer flexible user control over the stroke attributes such as color, texture, etc., as well as stroke orientation. But for faithful simulation of particular painting styles, it often requires careful configuration of parameters in the models. Even so, simulated results still look somewhat mechanical compared with hand-made paintings, especially for painting styles with distinct, expressive strokes, for instance, the style exhibited in Van Gogh's works. The image based NPR techniques (Litwinowicz, 1997; Hertzmann, 1998; Hertzmann *et al*., 2001; Nehab and Velho, 2002; Wang B. *et al*., 2004) usually process photographic images to get information for both brush color and orientation control. Brush texture can be either a solid color or more complex ones. In some recent efforts, brush texture patches are used instead of brush strokes (Wang B. *et al*., 2004). Most of the image based NPR techniques use the object boundary information to control brush orientations, while a few allow users to specify the brush orientation with a single parameter (Hertzmann, 1998). Those systems generate the brush orientations with perturbations added to the orientation parameter.

* Project supported by the National Basic Research Program (973) of China (No. 2002CB312101) and the National Natural Science Foundation of China (Nos. 60403038 and 60373037)

In some paintings, however, stroke orientations may vary in a complex manner, for example the painting of Van Gogh given in Fig.1 (see page 1157). In this picture, the artist painted different regions such as sky, trees and land with different strokes and orientations. Using uniform orientation or object boundary certainly would not be able to represent faithfully the kind of stroke orientations we observe in Fig.1.

In this paper, we present an approach to tackle this problem. Our approach falls into the image based approach. We employ the "divide-and-conquer" strategy, that is, we segment the photographic image into different regions corresponding to the sky, trees, land, etc., and construct directional fields on them. During the painting process, our approach performs layer by layer. After producing the grounding layer, we place representative brush strokes selected from examples over the directional field for several layers. These brushes are endowed with further color and shape perturbations.

The main features of our approach are:

(1) Adding a grounding layer. Most existing image based NPR techniques just place brush strokes to cover the photographic images fully. While in the painting process, artist may first paint different regions with background colors as a rough indication of different objects (as shown by the light blue color in the sky of Fig.1). Our method simulates this painting process by introducing a grounding layer with background colors before adding brush strokes.

(2) Construction of the directional field. The directional field is used to control brush stroke orientations. In our system, the directional field can be constructed in two ways. We may extract brush stroke orientation from example paintings with the compass operator and then fill appropriate regions with extracted directional field patch by use of direction synthesis. Alternatively, we can specify a few key directions in different region, then interpolate these key directions to get the directional field.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. Section 3 describes our algorithm of example based painting generation in detail. Section 4 demonstrates some experimental results and Section 5 concludes the whole paper and highlights future work.

RELATED WORK

NPR is a hot topic in recent years, with many achievements having been made up to now. We mainly review here several representative painterly rendering techniques, which are mostly related to our work.

Many works were devoted to transferring the style of line drawings onto 2D/3D scene (Hertzmann *et al*., 2002; Hamel and Strothotte, 1999; Jodoin *et al*., 2002; Kalnins *et al*., 2002; Freeman *et al*., 2003). Hamel and Strothotte (1999) introduced nonphotorealism templates for geometric model to facilitate this process. Such a template, which encodes the attributes of a rendition, can be automatically extracted from the exemplar model and be reused onto the desired one. Representing the strokes with parametric curves, Jodoin's method considered the stroke set as a vector of random variables following Markov distribution (Jodoin *et al*., 2002). New examples can therefore be converted into the artistic style using curve synthesis, which performs similarly to texture synthesis. The method in (Freeman *et al*., 2003) is based on a set of training lines with various styles drawn by the artists. Any new line can be transformed into a particular style determined by the training set through precise matching.

Hertzmann (1998) proposed an approach for creating an image with a handpainted appearance from a photograph. This approach begins with large scale brush painting, followed by appending progressively smaller brushes to approach the photograph. Some parameters for controlling the desired style is also provided in this method. Similar method was proposed by Shiraishi and Yamaguchi (2000). For each sample point on the given image, it computes the stroke value based on a color difference image, and paints strokes whose sizes are determined automatically by the value. Image analogy presents a new framework for processing images by example (Hertzmann *et al*., 2001). This method needs two diverse exemplars with the same context as input, among which one example serves as the filter version of another. By extracting this filter, a corresponding map is applied onto the given image to produce the similar painting effect. This method achieves remarkable rendering results, although the requirement of two registered exemplars restricts its application. In addition, this method is implemented with low efficiency because of its pixel-wise processing.

Wang B. *et al.*(2004) formulated the problem of painting generation as texture synthesis and enhances greatly the rendering efficiency. Extracting the texture patches from the exemplar paintings, this approach synthesizes textures on the desired image and fuses the synthesized result with the information on the image. But this approach does not adequately reflect the drawing style of the given example in terms of the brush direction, they produce the direction field only by interpolating the medial axis.

Few methods (Litwinowicz, 1997; Wang J. *et al.*, 2004) deal with transforming video sequences into NPR effects, Video Tooning (Wang J. *et al.*, 2004) achieved this by developing an enhanced mean shift method to segment the video volume into semantically meaningful regions, and endowing each region with cartoon effect.

OUR ALGORITHM ON EXAMPLE BASED PAINTING GENERATION

At the high level, our approach works for a given photographic image as follows:

(1) Brush library construction. Representative brushes are extracted from example paintings of the given style to construct a brush library.

(2) Region segmentation. The photographic image is segmented with the mean shift method (Comaniciu and Meer, 2002).

(3) Grounding layer synthesis. Suitable patches in the example paintings are selected to synthesize the grounding layer.

(4) Directional field construction. Directional field is constructed by either interpolating user specified key directions or synthesizing with extracted direction field from example paintings.

(5) Brush painting. Seeds are generated using the direction field and brushes are placed over the seeds with perturbations added to the brush shape and color.

(6) Fusion with image. Fuse the painted result with original photographic image. We introduce here a fusion scheme to prevent color overflow.

In the following subsections, we will describe each step of our approach in detail.

Brush library construction

Our goal is to produce paintings with the style of impressionist—Vincent Van Gogh. We first select interactively with snakes (Kass *et al.*, 1988) some typical brush samples from some paintings of Van Gogh to construct a brush library for different class of objects, such as the sky, trees, land, etc. Fig.2 (see page 1157) shows some brushes selected from Van Gogh's Olive Grove.

Region segmentation

Our approach employs the "divide-and-conquer" strategy. Through the mean shift method (Comaniciu and Meer, 2002), we segment the photographic image into different regions, each of which represents an individual class of objects, e.g. the sky, trees, land, etc. The system then performs painting operation on every region respectively.

Beginning at each pixel in the image, mean shift estimates the local density gradient of similar pixels, and finds the peaks in the local density recursively. All pixels that are drawn upward to the same peak are considered to lie in the same segmentation. To accelerate the segmentation process, we apply mean shift on the down-sampled image. As mean shift may produce many trivial patches, we interactively piece those patches belonging to the same class together if needed. Fig.3 (see page 1157) illustrates the segmentation process.

Throughout this paper we will use the following notations for the segmented regions. $\{C_j, j=1, \ldots, m\}$ represent the initial segmented clusters (Fig.3b) with meanshift. $\{R_i, i=1,\ldots,n\}$ $(n\leq m)$ are defined as the final regions (Fig.3c) after merging, thus each C_j belongs to a certain R_i.

Grounding layer synthesis

Grounding, or priming as it is usually called today, provides the surface suitable for painting. For the region to be painted on the image, we first pick up a nearly textureless sample from the suitable region on the exemplar paintings, and then perform texture synthesis to form the grounding layer. Note that we can apply a low pass filter to the sample in the case that selection of such a textureless sample is difficult. With the TSVQ acceleration (Wei and Levoy, 2000), grounding synthesis can be accomplished in a few seconds. Fig.4 (see page 1157) gives an example of grounding synthesis.

Direction field construction

As addressed in Section 1, in some paintings, brush orientation is an important element forming the painting style of the artist. We propose here two methods to construct the directional field for the current region to be painted.

But first, seed points are produced to anchor the brushes to be painted. We simply divide the current region into regular grids according to a user specified parameter to control the brush density. Centers of these grids with stochastic perturbations are taken as initial seed points.

The two methods we propose here are interpolation of user specified key curves, and direction synthesis with direction sample extracted from example paintings. We now address the two methods in detail.

1. Interactive construction

The user specifies a few curves by simply clicking and dragging the mouse cursor in the current region, after which the curves are automatically discretized into uniformly distributed points. The tangent direction of the curve at each point is chosen as the direction for this point. We then interpolate the directions of the seeds effectively using Radial Basis Functions (Carr *et al*., 2001).

2. Direction synthesis based construction

In this method, we extract the directions of brush strokes on the example paintings for some sample points with the compass operator (Maxwell and Brubaker, 2003). Exploiting robust histogram matching, the compass operator finds the orientation of a diameter that maximizes the difference between two halves of the color distributions in a circular window.

Rather than extracting the brush direction for the whole desired region on the example painting, we only specify interactively a moderate rectangle R_e in the desired region as the representative, and implement detection operator on it. R_e is further divided into regular square grids, whose centers are chosen as the sample points. The Compass operator detects edges for all the points lying in each grid, over which the orientation with the maximal response is regarded as the direction for the sample.

Extracted brush directions can be seen as a kind of textures with which we synthesize the directional field on segmented regions of the image using texture synthesis technique. We first construct a compact bounding box R_p for the current region of interest on the given image, and then divide R_p into uniform grids, each one having the same size as that of the grid in R_e. During the synthesis phase, we take the direction of each sample point as the measure of similarity between the sample directions and those synthesized. The direction field of R_e with corresponding sample points on the example paintings is taken as the texture sample, with R_p being the region to be synthesized. We produce the directions for some grid centers in R_p iteratively by extracting their synthesized neighbors' directions, and matching directions in R_e. The difference between two normalized directions is evaluated by their Euclidean distance in 2D space. Fig.5 (see page 1157) gives an example of direction extraction and synthesis.

After synthesizing the direction field on R_p, we cull the irrelevant region, and only hold the directions lying in the current interest region on the photographic image. For our applications, the synthesis operation runs in real-time because only finite directions, normally about several hundreds, need to be synthesized.

The direction of each seed point is calculated by inverse distance interpolation with its nearby grid centers' directions.

Brush painting

The goal of brush painting is to fill in the current region with brushes chosen from brush library. During this phase, we glue brushes randomly chosen from the brush library onto these seed points, each brush is first rotated with respect to the direction of the current seed, its shape is then zoomed in/out according to a user specified parameter, by taking the size of the desired image into account.

To simulate the hand-crafted look in the generated painting, we perturb further brush shape and color. Brush shape can be changed by a scheme similar to FFD in mesh deformation (Sederberg and Parry, 1986). We build a controlling mesh surrounding the brush (Fig.6a, see page 1158) and perturb positions of controlling points to get deformed brush shapes. Brush color can be changed by adding perturbations to its original color, and Fig.6b gives an example of variation of brush shape and color.

Brushes placed on the grounding layer may appear "floating" above the grounding layer, in order to make brushes stick to the grounding layer, a blending operation is performed on the boundary of each brush. The resulting color of each boundary point on the brush is valued by averaging its original color with the corresponding color on the grounding layer.

In real painting process, the artists usually paint their works with several layers of brushes, each of which with different brush coloring or style. The painting generated in this way therefore looks colorful enough. To imitate this process, brush painting can be done several times. That is, we append a layer of brushes with distinctly different colors or styles at each time. For each layer, the user specifies the direction field, generates seed points, and paints brushes loaded from the library.

Fusion with image

Brushes in the brush library have their own colors, generated painting by just placing them on the grounding layer may look flattening, lacking the depth exhibited by the luminance of the photographic image. In order to reflect the luminance of the photographic image in the generated painting, we fuse the brush color with the photographic luminance as follows.

We apply the fusion operation on YCbCr color space. YCbCr is a well-known color space compliant with the digital video standard, where CbCr mainly represents the hue of the color and Y encodes its brightness.

Define Y, Cb, Cr and Y_e, Cb_e, Cr_e as the color channels of the pixel p on the final result and that of its corresponding pixel p_e on the brush respectively. $\underline{Y_p}$, $\underline{Cb_p}$, $\underline{Cr_p}$ denotes the average color of the mean shift segmented cluster C_j that p belongs to. Then the fusion is performed using the following formulas:

$$Y=m_e\times(Y_e-\underline{Y_e})+\underline{Y_p},$$
$$Cb=m_{cb}\times Cb_e+(1-m_{cb})\times\underline{Cb_p},$$
$$Cr=m_{cr}\times Cr_e+(1-m_{cr})\times\underline{Cr_p},$$

here m_e is the parameter denoting how strong is the brush intensity's effect on the image, $\underline{Y_e}$ represents the average luminance of brushes in the same brush class. m_{cb}, m_{cr} stand for the weights balancing between the hue of the pixel on the photographic image and its corresponding one on the brush, we empirically value them with a small positive 0.2 in our experiment.

In some cases, Y component will overflow the range of 0 to 255 if the parameter m_e is not properly selected. Therefore instead of setting it a constant, we adopt here a non-linear form to value it dynamically, according to the color of the current pixel on the image:

$$m_e=k\times\sin(\pi/2\times\min(\underline{Y_p}, 255-\underline{Y_p})/255),$$

where k is a constant set as 0.7 in our experiments. It is easy to prove with the above formula, that the computed brightness falls in the range 0 to 255.

EXPERIMENTAL RESULTS

We implemented our algorithm on an Intel Pentium IV 1.6 GHz PC with 256 MB memory under the Windows XP operating system.

Fig.7 (see page 1158) shows the painting we generated simulating the style encoded by Van Gogh's painting in Fig.1. Fig.8a (see page 1158) is another painting selected from the series of Van Gogh's Olive Groves, and Fig.8b (see page 1158) is the corresponding painting result produced using our algorithm.

Figs.9c and 9e (see page 1158) show our rendering effects simulating the styles of Van Gogh's painting Wheatfield with Crows (Fig.9a). Our system also supports the user drawn sketches (Fig.9b), and transforms it into the painting style (Fig.9c). Given a photographic image (Fig.9d), we can also specify some auxiliary objects, for example the road in Fig.9d obtained by simply dragging several curves and denoting it as the road. Fig.9e is the corresponding painting we generated. Note that for the sky and wheat fields in Figs.9c and 9e, we paint them with several layers. Normally, it takes about a few minutes to produce a painting with user interaction.

CONCLUSION AND FUTURE WORK

We present a new framework for generating paintings on photographic images with the style of the example paintings. Our framework simulates the hand painting process by first bestowing a grounding, and then appending the brushes with respect to the direction field generated. Brush density and colorintensity over the image can be adjusted with ease.

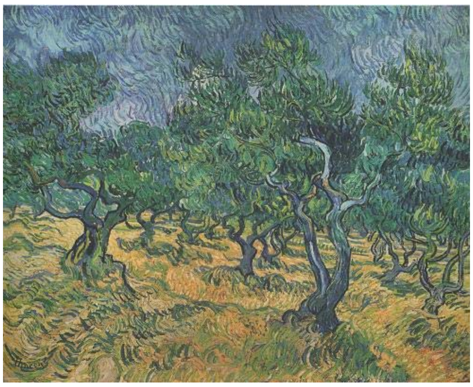
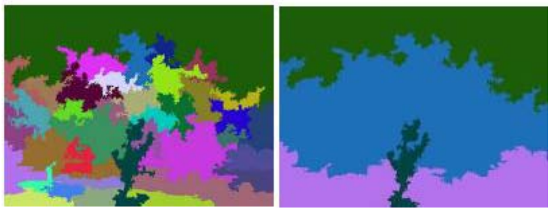


Fig.1 Olive Grove by Van Gogh

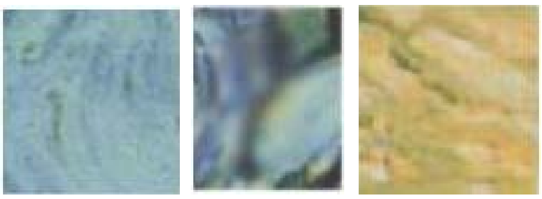
Fig. 2 Brushes selected from Van Gogh's Olive Grove
From top to bottom: brushes for sky, trees and terra

(a)



(b) (c)

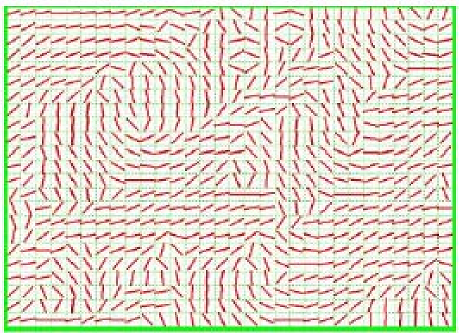
Fig.3 The segmentation process. (a) A desired image; (b) Segmentation result with mean shift, shown in pseudocolor; (c) Final segmentation after merging



Sky Trunk Ground

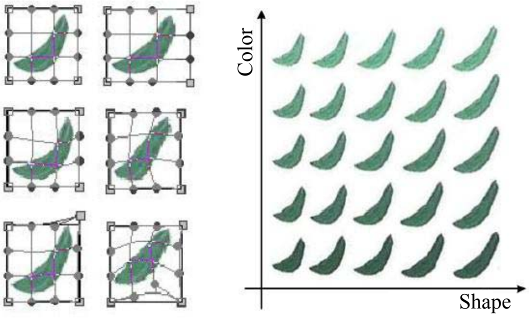
Fig.4 Grounding synthesis for the image in Fig.3. Grounding textures were obtained from Olive Grove (Fig.1). Note that we select the same grounding for the trees with the sky

(a)



(b)

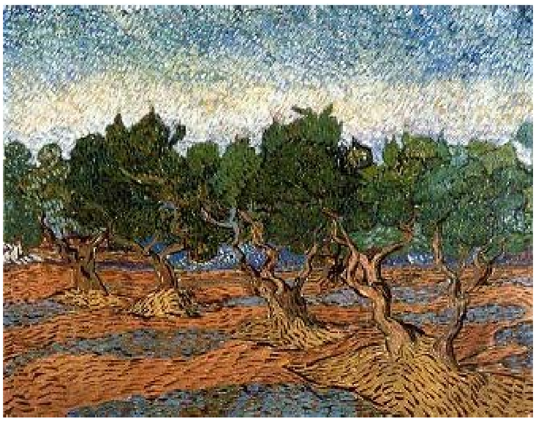
Fig.5 Direction extraction and synthesis. (a) Extracted direction field in a sample rect on Van Gogh's Starry Night; (b) A synthesized direction field



(a) (b)

Fig.6 Shape and color variance of the brush. (a) Shape variance by deforming the controlling mesh; (b) Shape and color variance matrix

Fig.7 Painting generated simulating Olive Grove in Fig.1

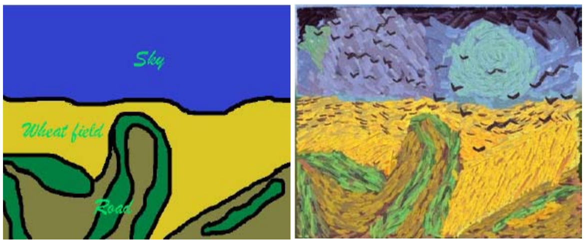


(a)

(b)

Fig.8 Another Olive Grove (a) and result (b)

(a)



(b) (c)

(d) (e)

Fig.9 (a) Van Gogh's Wheatfield with Crows; (b) The user drawn sketches; (c) The painting style; (d) A photographic image; (e) The corresponding painting result we generated

Although initial experiments generated some encouraging results, there still exists a certain difference between our results and the artists' paintings. Also our approach is not yet robust enough to handle all cases. It is more suitable to process the paintings with clear brush structures. For those without clear brushes, texture synthesis based method (Wang B. *et al*., 2004) may be a good choice. In the current phase, brush extraction is preformed by elaborate user interaction, which is inconvenient and time consuming. We intend to enhance the automatization and efficiency of this process based on machine learning and image segmentation. This is a topic for future work.

References

Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R., 2001. Reconstruction and Representation of 3D Objects with Radial Basis Functions. Proc. SIGGRAPH'01, p.67-76.

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, **24**(5):603-619.

Freeman, W.T., Tenenbaum, J.B., Pasztor, E.C., 2003. An example-based approach to style translation for line drawings. *ACM Trans. Graphics*, **22**(1):33-46. [doi:10.1145/588272.588277]

Hamel, J., Strothotte, T., 1999. Capturing and re-using rendition styles for non-photorealsitic rendering. *Computer Graphics Forum (Eurographics'99)*, **18**(3):173-182. [doi:10.1111/1467-8659.00338]

Hertzmann, A., 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. Proc. SIGGRAPH'98, p.453-460.

Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H., 2001. Image Analogies. Proc. SIGGRAPH'01, p.327-340.

Hertzmann, A., Oliver, N., Curless, B., Seitz, S.M., 2002. Curve Analogies. Proc. 13th Eurographics Workshop Rendering, p.233-246.

Jodoin, P.M., Epstein, E., Granger-Piché, M., Ostromoukhov, V., 2002. Hatching by Example: A Statistical Approach. Proc. Int'l Symp. NPAR 2002, Annecy, France, p.29-36.

Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F., Finkelstein, A., 2002. WYSIWYG NPR: Drawing strokes directly on 3D models. *ACM Trans. Graphics (SIGGRAPH'02)*, **21**(3):755-762.

Kass, M., Witkin, A., Terzolpoulos, D., 1988. Snakes: Active contour models. *International Journal of Computer Vision*, **1**(4):321-331. [doi:10.1007/BF00133570]

Litwinowicz, P., 1997. Processing Images and Video for an Impressionist Effect. Proc. SIGGRAPH'97, p.407-414.

Maxwell, B.A., Brubaker, S.J., 2003. Texture Edge Detection Using the Compass Operator. Proc. of BMVC 2003.

Meier, B.J., 1996. Painterly Rendering for Animation. Proc. SIGGRAPH'96, p.477-484.

Nehab, D., Velho, L., 2002. Multiscale Moment-Based Painterly Rendering. Proc. of SIBGRAPI'2002, p.244-251.

Sederberg, T., Parry, S., 1986. Free-form deformation of solid geometric models. *Computer Graphics (SIGGRAPH'86)*, **20**(4):151-160. [doi:10.1145/15886.15903]

Shiraishi, M., Yamaguchi, Y., 2000. An Algorithm for Automatic Painterly Rendering Based on Local Source Image Approximation. Proc. of the First International Symposium on Non-photorealistic Animation and Rendering—NPAR'00, p.53-58. [doi:10.1145/340916.340923]

Wang, B., Wang, W.P., Yang, H.P., Sun, J.G., 2004. Efficient example-based painting and synthesis of 2D directional texture. *IEEE Trans. Visualization and Computer Graphics*, **10**(3):266-277. [doi:10.1109/TVCG.2004.1272726]

Wang, J., Xu, Y.Q., Shum, H.Y., Cohen, M.F., 2004. Video Tooning. *ACM Trans. Graphics (SIGGRAPH'04)*, **23**(3):574-583. [doi:10.1145/1015706.1015763]

Wei, L.Y., Levoy, M., 2000. Fast Texture Synthesis Using Tree-structured Vector Quantization. Proc. SIGGRAPH'00, p.479-488.