SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2013, Vol. 56 012103:1–012103:13
doi: 10.1007/s11432-012-4740-2

Animating Chinese ink painting through generating reproducible brush strokes

YANG LiJie* & XU TianChen*

Faculty of Science and Technology, University of Macau, Macao 999078, *China*

Received June 3, 2012; accepted July 29, 2012; published online December 7, 2012

Abstract By generating reproducible brush strokes of Chinese ink painting, we propose an image-based tool for lively displaying real painting process of Chinese ink painting. Contrasted with previous work where the brush trajectory is explicitly specified as manual input, we automatically extract this trajectory from the outlines of the shapes to paint and model the brush footprints; first complex shape of an input image is decomposed into brush strokes. Second, we estimate the brush trajectory with shape outline and build a brush footprint model to measure the quality of the trajectory constrained by the variation along a stroke of the painting process parameters, such as location, size, and orientation of the footprints. Finally, with our real-time rendering model, the estimated trajectories are dynamically rendered into brush strokes by mapping footprint textures sampled from the input images. It is particularly useful for analysis and appreciation of Chinese ink painting, as well as for art practice and education.

Keywords brush trajectory, non-photorealistic rendering, real-time rendering, Chinese ink painting

Citation Yang L J, Xu T C. Animating Chinese ink painting through generating reproducible brush strokes. Sci China Inf Sci, 2013, 56: 012103(13), doi: 10.1007/s11432-012-4740-2

1 Introduction

Chinese ink painting, also known as “Shui-mo Hua”, is famous for its freehand brushwork and natural aesthetic values, where generally black ink is the only ink used. Unlike western styles, such as water color and oil painting, Chinese ink painting uses few strokes to represent an object. Each stroke conveys information about the scene and painter’s personality through its shape and texture. Its appearance depends on the shape of the object to paint, the brush trajectory, and the distribution of the ink and water in the brush. Over thousands of years, copying (namely “Lin Mo”) masterpieces is the popular tracing exercise for training Chinese ink painting, where people who are not talented in art are given shape outlines and paint strokes inside them to produce paintings without masterly skills. In other words, to be a painter, people must always copy and practice the ancient masterworks to obtain the famous painters brushwork. But the learners can only try to puzzle out the real painting process of an ancient masterpiece based on the general painting knowledge and techniques without any video material which can record the painting process. Besides, painting is a kind of disposable and unduplicated artistic creation. It means that the famous painter cannot produce the same masterpieces in different time. Hence, reproducing the painting process of the still image of Chinese ink painting in a vivid way is not a trivial issue.

*Corresponding author (email: ya77410@umac.mo, X-Universe@live.com)

Previous work (Subsection 1.1) on simulation of Chinese brush strokes require the user to specify the trajectory of the brush which is then rendered into Chinese calligraphy or Chinese ink painting. This simple manual input makes the simulation of Chinese brush stroke highly labor-intensive. Furthermore, the complex interactions based on the input devices, such as pen-like device [1,2], require painting knowledge and the techniques of device control.

In this paper, based on brush trajectories and painting textures, we present a novel image-based tool for generating reproducible brush strokes. After an interactive segmentation on complex shapes of a painting image, the developed tool applies our single-stroke trajectory model to automatically extract the reasonable brush painting path, then renders them with brush footprint model and textures sampled from the original image. We automatically extract brush trajectory from the shape outline to paint, rather than direct manual input.

The remaining parts of the paper are organized as follows. After reviewing the related work (Subsection 1.1), we give an overview of our algorithm and outline the major contributions (Subsection 1.2). Section 2 describes our framework to calculate brush trajectory. We use an elliptic footprint model to emulate the painting process along the trajectory and render each brush stroke dynamically at the level of footprint (Section 3). In the section of experiments and results (Section 4), we show some examples and contrasts to evaluate the correctness and effectiveness. We draw a conclusion in Section 5.

1.1 Related work

In recent years, with aim to reproduce the real painting process, much research work has been done to simulate brush strokes or develop painting tools.

• **Non-physics 2D brush.** With a canvas, brushes, palettes and a great deal of color options provided by digital painting software (e.g. Adobe brush packages, ArtRage, and Corel Painter), users can control a mouse or a digital pen in the same way as they use a real brush. These techniques may apply some simple transformations to the bitmap while stamping the footprint along the stroke. Furthermore, some researchers focus on sketch-based approaches for the stroke placement. This popular approach is to let the user sketch brush trajectories by a mouse or a pen-like device and then automatically convert them into paintings. Strassmann [3] models a stroke as a trajectory of position and pressure specified by the mouse and keyboard. Then an approach for synthesizing rock textures in [4] introduces an ellipse model to simulate variation of stroke width. However users have to input the rocks contours, stroke textures and stroke parameters and this approach just considers the stroke brushwork of Chinese landscape painting. On the other hand, a physically-based system for creating images with watery paint is developed by Van Laerhoven et al. [5]. And Ref. [6–8] propose brush stroke models which define a 2D region of a stroke with splines and control shapes by manual input. Actually these models depend on the specified media axis and stroke width. So they offer a certain flexibility and control over the shape of the stroke, whereas ignoring the artistry and variation of brush strokes, which are considered in our trajectory model and rendering process sincerely. By reproducing the natural and smooth lines in brush strokes, our approach reveals the artistic beauty in Chinese ink painting.

• **3D virtual brush.** A virtual 3D brush provides an intuitive and natural feeling when users hold a pen-like device. These techniques model 3D brush strokes with a series of parameters [9–11]. They improve the flexibility in modeling complex strokes as compared to non-physics 2D brushes but capturing the model parameters to create realistic brush strokes is greatly challenging. On the other hand, in real world, controlling automatically a virtual brush in addition to the dynamics of the tuft is complex and users without professional skills often require simplified environments where paintings can be generated with less interaction and painting skills. Hence, we implement our approach into an accessible system with the minimum user interaction that appropriates for the unskilled people.

• **2D and 3D animation.** During the development of cartoon making, research on 2D and 3D animation mainly focuses on computer techniques in combination with art. For instance, Lake et al. [12] devise real-time algorithms to emulate cartoon styles. In order to save production costs of animations, Zhang et al. [13] introduce a novel algorithm for synthesizing animations of running water in the style of

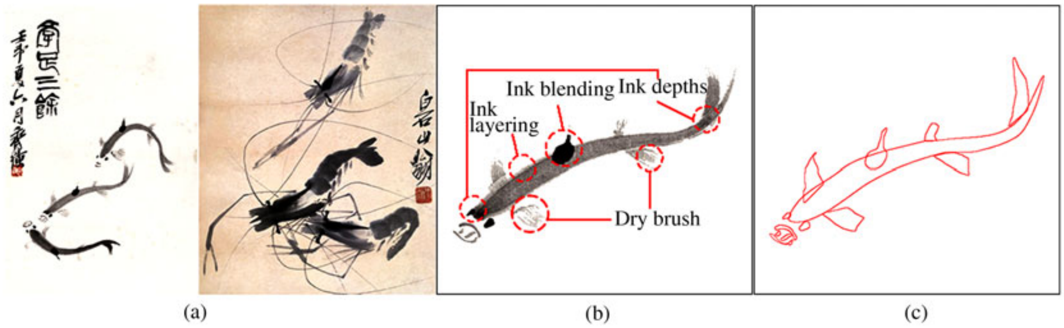

Figure 1 (a) Two masterpieces of Chinese ink painting about fishes and shrimps; (b) special ink effects in the fish painting; (c) complex shape segmentation in fish painting image.

Chinese paintings directly from real videos. Using existing software packages, Chan et al. [14] create 3D Chinese painting animation "Autumn Bamboo" shown in Siggraph Animation Theater. Especially, it is an extremely challenging issue to generate a 2D animation with a static painting image. Xu et al. [15] address a novel approach to animate 2D Chinese painting through stroke decomposition. Moreover, Yuan et al. [16] implement a real-time rendering system to generate a Chinese ink-and-wash cartoon for freeing the animators from laboriously designing Chinese painting appearance. However, these works mainly concentrate on the rendering techniques and rarely involve the traditional ink effects like "dry brush" and "diffused edge" by the special brushwork in Chinese painting (Figure 1(b)). Like our motivation, Yang et al. [17] animate the brush-writing process of Chinese calligraphy characters by a trajectory extraction model, where the trajectory is determined by the skeleton of a stroke and the key points in head and tail part of the stroke according to Chinese calligraphy knowledge. As for a stroke in Chinese ink painting, its painting process should be constrained by its shape. But the skeleton-based trajectory model with thinning algorithm is not suitable to our issue (Section 4). Inspired by these solutions, we reconstruct the painting process into a 2D animation to reappearance art creation of Chinese ink painting with painting trajectories and ink effects in a vivid and distinct way.

1.2 Overview and contributions

One of the most important motivations for our work is that generating reproducible brush strokes will allows Chinese ink paintings to be animated and the generated image will entirely respect the original painting. Given the shape of a painting object, we automatically calculate its trajectory by contour-based method, instead of extracting medial axis or skeleton as the brush trajectory. When rendering brush footprints along the calculated trajectories, first we model the shapes of the footprints as ellipses, and then map the brush footprint textures, which are sampled from the given painting object, onto the positions along the trajectories. Meanwhile, with an ink diffusion process constrained by the shape outlines, our rendering model also presents the expressive ink effects (Figure 1(b)). The overview of our approach is as follows.

• **Brush trajectory estimation.** First based on a image segmentation method in [15], we decompose the complex shapes of a painting image into primary shapes which can be rendered by a single brush stroke. Second we devise contour-based method to calculate the brush trajectory for each stroke (Section 2).

• **Brush stroke rendering.** In this stage, the calculated trajectory is rendered into a brush stroke by mapping footprint textures of the original painting image (Section 3). First we model the brush footprint as an ellipse which is parameterized by the physical parameters of the painting process, including the variation of the footprint location, size, and orientation of the brush along the trajectory. Second, by techniques of multipass texture rendering, we enhance impressive ink effects, such as dry brush, ink layering, ink diffusion and so on (Figure 1(b)).

The novelty proposed in this paper lies in the fact that we estimate the painting trajectory given shapes of the brush strokes. Most previous works require the trajectory to be specified by the user. Our algorithm automatically estimates this trajectory. Compared with the existing approaches, our approach

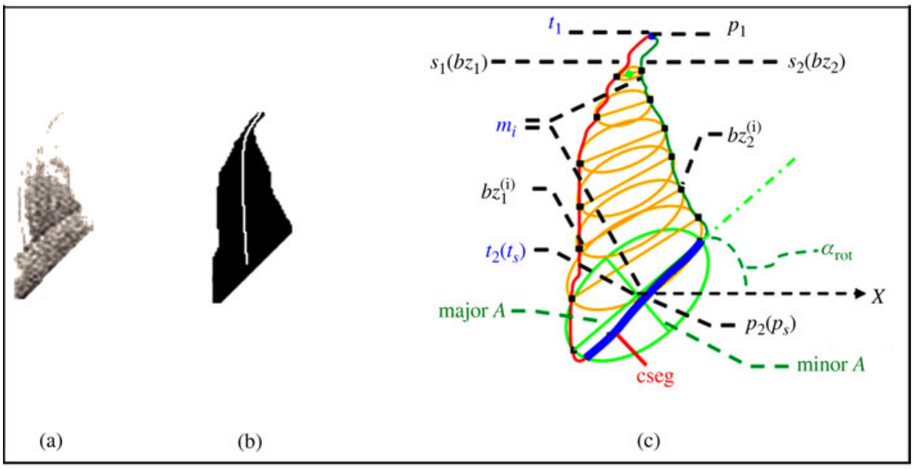

Figure 2 A stroke example in single-stroke trajectory model. (a) Input stroke; (b) brush trajectory estimated by our method (white curve); (c) the enlarged views of trajectory estimation process: $t_1 = p_1$; after measuring the contour segment ("cseg" of a thick line) along each m_i, $t_2(t_s)$ is a certain m_i instead of $p_2(p_s)$.

is more concerned with reconstruction of the artistic painting process of Chinese ink painting (i.e. the brush movement during the sequential painting process). So our work can be of value as a research tool as well for animation production. Moreover, except the applied value in appreciation and education of art, our reproduction work can be applied in digital cultural heritage, which makes the ancient artwork of Chinese ink painting serve all times, all places and all peoples by presenting them as the original in a lively way.

2 Brush trajectory estimation

Drawing Chinese ink painting can be considered as a process of abstracting a complex scene into few strokes, each one communicates key information about the scene and implies painting knowledge and brushwork. When a painter draws a stroke, the brush evolves along a continuous trajectory. Here we propose a contour-based method by which brush footprints along trajectories can represent shape variation. The key to our approach is the use of domain knowledge which is derived from the common brush techniques in Chinese ink painting. As follows we first describe our method on single brush strokes (Subsection 2.2) after decomposing a painting image into a certain number of strokes (Subsection 2.1).

2.1 Brush stroke decomposition

In general, an image of painting object contains complex shape, which is produced by several brush strokes hypothetically. In complex shapes, some of the brush strokes may be overlapped. Hence before estimating brush trajectory takes place, the image with complex shape is decomposed into several single brush strokes. And generating the natural and original brush strokes is a critical procedure in our reproduction work. Inspired by the decomposition method of Xu et al. [15], we develop an interactive tool for stroke extraction. Given an image of a painting, first we segment the image into regions of similar color intensities. This process will produce an oversegmented image. Then we merge contiguous regions that likely belong to the same brush strokes. At last we interactively refine the segmentation results to be more natural. Figure 1(c) shows our segmentation example with a fish painting image.

2.2 Single-stroke trajectory model

By definition, a brush stroke is a continuous sequence of footprints. Any jump can be interpreted as the end of the current stroke and the beginning of a new one. For single brush strokes, we devise a singlestroke trajectory model as shown in Figure 2, where each brush stroke undergoes a more complicated iterative process which consists of two steps as follows.

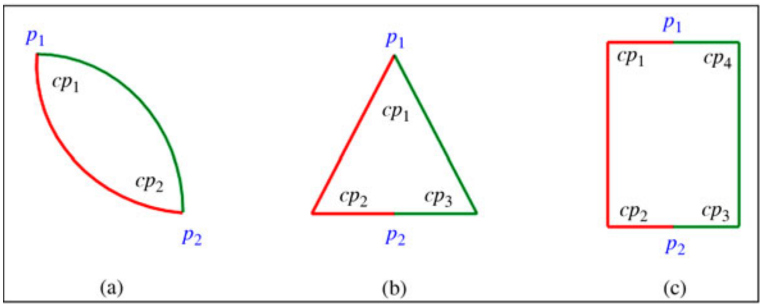

Figure 3 p_1 and p_2 detection of three types of pseudo-strokes for contour partition. (a) cp_1 and cp_2 are p_1 and p_2; (b) cp_2 and cp_3 are merged into p_2 with the shortest arc length between them; (c) in the same way, by comparing the arc lengths between every two corner points iteratively, we merge cp_1 and cp_4, cp_2 and cp_3 into p_1 and p_2 respectively.

2.2.1 *Medial axis of a stroke shape*

Geometrically, a shape can be extracted by its skeleton (i.e. medial axis) using the existing thinning algorithms. However, these thinning algorithms probably produce branches which noise the shape description. So in order to adapt to the changeable lines of strokes in Chinese ink painting, we devise a simple method to find out the medial axis of a shape with its contour. Given a contour of a shape drawn by a single stroke, we proceed:

• **Contour partition.** We segment the contour into two sides s_1 and s_2; we first sample the contour of the shape into n points at one pixel interval and then classify the points into two sides by two special points on the contour p_1 and p_2, which decide the start and end of the stroke when a painter drops and lifts a brush tip to paint the stroke. And partitioned by them, both sides of a contour describe a stroke shape. To determine p_1 and p_2, we proceed:

– We adopt a feature points extraction method proposed by He et al. [18] to find out corner points of the contour as candidates of p_1 and p_2.

– The underlying assumption is that the stroke shapes can be roughly classified into three types, as the pseudo-strokes shown in Figure 3, which are common in Chinese ink paintings. In the case of more than two corner points (Figure 3(b) and (c)), we just iteratively merge two of them along the contour until p_1 and p_2 are identified; we first find out two points with the shortest arc length between them and then approximate a point on the contour to their midpoint.

– We specify p_s from p_1 and p_2 as the start node of the stroke painting process.

• **Contour equal division.** We use piecewise cubic Bézier curve to fit the points inside s_1 and s_2 because a single piece curve hardly approximates to the contour of the stroke shape which is commonly complicated and varies in Chinese ink painting. Additionally, we apply least square method (LSM) to minimize the squared distance between the original data from s_1 and s_2 and the fitted data from cubic Bézier curves. Combining the Bézier approximation with LSM formula, we have

$$\begin{aligned}
&S=\sum_{i=1}^{n}[p_i-(1-t_i)^3P_0+3t_i(1-t_i)^2P_1+3t_i^2(1-t_i)P_2+t_i^3P_3]^2,\\
&\frac{\partial S}{\partial P_1}=0,\quad \frac{\partial S}{\partial P_2}=0,\\
&P_1=(A_2C_1-A_{12}C_2)/(A_1A_2-A_{12}A_{12}),\quad P_2=(A_1C_2-A_{12}C_1)/(A_1A_2-A_{12}A_{12}),
\end{aligned}\tag{1}$$

where

$$\begin{aligned}
&A_1=9\sum_{i=1}^{n}t_i^2(1-t_i)^4,\quad A_2=9\sum_{i=1}^{n}t_i^4(1-t_i)^2,\quad A_{12}=9\sum_{i=1}^{n}t_i^3(1-t_i)^3,\\
&C_1=\sum_{i=1}^{n}3t_i(1-t_i)^2[p_i-(1-t_i)^3P_0-t_i^3P_3],\quad C_2=\sum_{i=1}^{n}3t_i^2(1-t_i)[p_i-(1-t_i)^3P_0-t_i^3P_3].
\end{aligned}\tag{2}$$

Denoting two Bézier approximations of s_1 and s_2 by bz_1 and bz_2, we sample the contour into $n{+}1$ points by dividing bz_1 and bz_2 into n equidistant slices. Let $bz_1^{(i)}$ and $bz_2^{(i)}$ be the ith point on the bz_1 and bz_2 respectively.

• **Midpoints detection.** We calculate the midpoints m_i between $bz_1^{(i)}$ and $bz_2^{(i)}$ to fit the final medial axis by piecewise cubic Bézier curve (Eqs.(1), (2)).

2.2.2 *Conversion of medial axis to trajectory*

In techniques of Chinese ink painting, different brush actions result in different shapes of two ends of a stroke. If a brush paints the end of a stroke with a pause, the shape of the end will be smooth (see two ends shown in Figure 3(a)); otherwise it will be sharp (see two ends in Figure 3(c)). To implement this domain knowledge, for the stroke with two sharp ends, we convert the medial axis to the trajectory directly; and for the stroke with a smooth end, we extract a segment of the medial axis to be trajectory. In other words, we have to revise the end points of a trajectory, denoted by t_1, t_2, by the following procedures (Figure 2):

• If the stroke has a sharp end, we directly remark the corresponding point on the media axis as one of the end points of the trajectory (i.e. $p_1 \rightarrow t_1$ or $p_2 \rightarrow t_2$).

• Using an ellipse centered at m_i, with major-axis length (major A) equal to the distance between $bz_1^{(i)}$ and $bz_2^{(i)}$, minor-axis length (minor A) set to $0.5 \times$ major A, and the rotation $\alpha_{\rm rot}$ measured by an angle between major$A$ and x-axis in coordinate system, we first calculate the length of contour segment inside each ellipse (cseg), and then identify a certain midpoint m_i whose ellipse contains the longest "cseg" as another end point of trajectory (i.e. $m_i \rightarrow t_2$ or $m_i \rightarrow t_1$).

• We fit t_1 and t_2, as well as m_i between them to be a trajectory by piecewise cubic Bézier curve.

In Chinese ink paintings, brush techniques attach great importance to a stroke drawn with smoothness and coherence. Therefore to produce more natural and smooth animation of painting process, we adopt the fitting curve to represent the trajectory rather than several line segments used in Yang's model [17].

3 Brush stroke rendering

In the above section, the trajectory estimation method provides a path of brush footprints during a painting process. In order to generate every reproducible brush stroke, we need to render each trajectory with brush footprints. Hence, here we first emulate brush footprints as the regular and irregular ellipses in a brush footprint model which derives from the brushwork knowledge in Chinese ink painting (Subsection 3.1), and then devise a real-time based rendering model (Subsection 3.2), which maps textures sampled from the original painting image onto a sequence of ellipses with techniques of multipass texture mapping.

3.1 Brush footprint model

As shown in Figure 4(a), at a given time, the brush movement commonly generates an interactive region resembling an ellipse on Xuan paper. In order to present the interaction between a brush and Xuan paper during the painting process, we adopt an ellipse to model a brush footprint which is used for simulating footprints in Chinese calligraphy [9,17]. Chinese ink painting mainly contains two basic painting brushworks: Zhong-feng and Ce-feng. As Yang et al. depicted in [17], Zhong-feng refers to keeping the brush tuft tip always in the middle of the stroke, which can be simulated by the regular ellipses; Ce-feng means that the painter uses the brush tuft tip in a sidelong manner and the brush footprint remains tangent to one boundary of the shape while the tip touches the other, so an irregular ellipse is suitable to display the footprint.

Using a series of ellipses, we model the shapes of brush footprints along a brush trajectory as shown in Figure 4(b). Assuming that an estimated brush trajectory curve is sampled into N points (p_i) at a certain interval, we define a stroke S as a group of footprints $s_i{=}(c_i, \text{ax\_major}_i, \text{ax\_minor}_i, \theta_i)$,$(i = 1, 2, \ldots, N)$, where c_i, ax_major_i, ax_minor_i and θ_i are respectively the center, major-axis length, minor-axis length

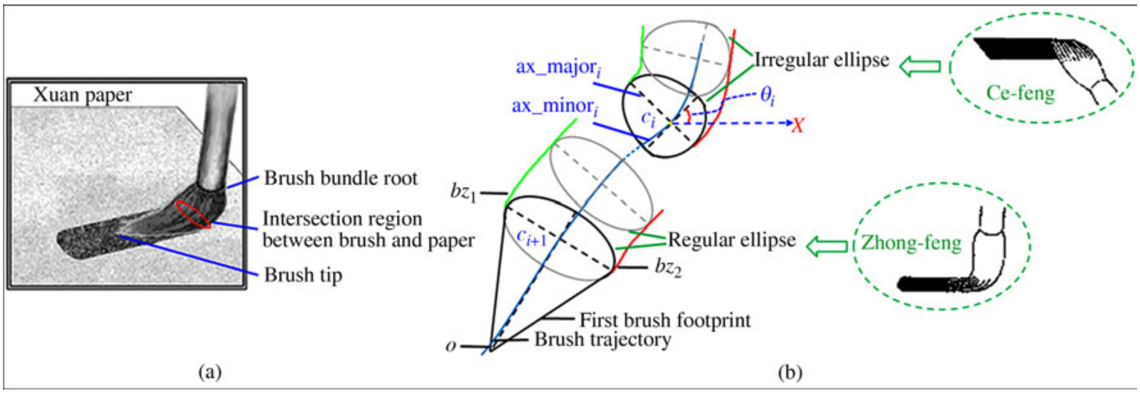

Figure 4 (a) Relationship between a real brush and Xuan paper; (b) effect of each term in our brush footprint model with ellipses of Zhong-feng and Ce-feng. In particular, we model the irregular ellipses as footprints as the trajectory is closer to one side of the stroke contour.

and orientation of the ith footprint s_i. Particularly, we add the tip o to the first ellipse to show a brush tip. In the following, we detail these parameters in our ellipse model (Figure 4(b)):

• **Center (c_i).** We distribute a number of footprints (i.e. ellipses) evenly on a brush trajectory curve. It means that each of our ellipses is centered at the sampled points p_i on the trajectory curve (i.e. $c_i = p_i$).

• **Major-axis (ax_major$_i$).** In the normal direction of the trajectory curve at c_i, major-axis involves two distances (equal or unequal) from c_i to two sides of the contour, which measures the size of the ith footprint (i.e. brush width). In Chinese ink painting, Ce-feng is used to change a brush stroke's direction which causes an obvious difference in length between two sides of a contour. In this way, in local stroke by Ce-feng, the local trajectory estimated by our contour-based method is close to one side of the contour. Hence, by two unequal distances between c_i and bz_1, bz_2 along the major-axis, an irregular ellipse, which is placed on the local trajectory as a footprint, is just amenable to Ce-feng's characteristics. Figure 4(b) shows the above principle for two main brushworks: Zhong-feng and Ce-feng, which demonstrates that our trajectory estimation integrated with footprint model can implement the knowledge of brushwork in Chinese ink painting.

• **Minor-axis (ax_minor$_i$).** We consider that the minor-axis, not mentioned in Yang's model [17], measures the time where the brush footprint pauses on the paper. If the pause time is longer, the length of ax_minor_i will be longer. Generally, the brush pause happens while the orientation of footprint has a sudden change. So given an initial value of the minor axis (ax_minor$_0$), we use the orientations of two adjacent footprints along the trajectory to define ax_minor$_i$.

$$\text{ax\_minor}_i = \frac{\theta_i \cdot \text{ax\_minor}_{i-1}}{\theta_{i-1}}. \tag{3}$$

• **Orientation (θ_i).** The orientation is defined as the angle between the minor-axis and the x-axis of the coordinate system.

3.2 Real-time rendering model

Our brush footprint model provides a sequence of footprint locations, sizes, and orientations, with which we render every brush stroke dynamically at the level of ellipse with artistic ink effects, such as dry brush, ink layering, ink diffusion, etc. (Figure 1(b)), which are reproduced by sampling and footprint reconstruction. As a result, the image generated by animating process is extremely similar to the original one. Besides, to enhance the efficiency of our rendering algorithm we utilize programmable graphics pipeline and implement GPU programs. In this section, due to limitations of space, we only describe the rendering process of one frame (Subsection 3.2.1).

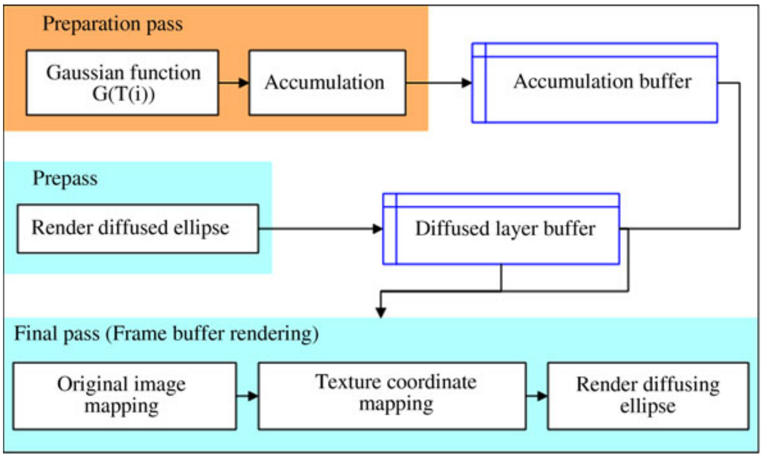

Figure 5 Overview of our rendering pipeline.

Table 1 Buffer allocation

Buffer name	Format	Number	Note
Accumulation	16-bit (R16)	1	Weight computation for footprint layers
Diffused layer	32-bit (R8G8B8A8)	2	The diffused ink into background

3.2.1 *Rendering pipeline*

The brush footprints with the effects of ink and water represent the interaction between a real brush and Xuan paper. And ink and water within the range of footprints undergo a diffusing process. In other words, we suppose every footprint has two rendering states: one is the diffusing state that means the diffusion of ink and water is in progress; the other is the diffused state that represents the diffusion is finished. In order to animate more natural ink effects during the brush movement, we implement three passes in rendering pipeline (Figure 5) with the buffer allocation (Table 1). In particular, except GPU-based vertex shader and pixel shader, we add a preparation pass for layer accumulation to handle the jagged problem of the footprints (Subsection 3.2.2). We describe three passes of our rendering process as follows:

- **Layer accumulation.** The rendering target of this pass is an accumulation buffer for counting the number of footprint layers at one pixel. Subsection 3.2.2 details the principles.
- **Diffused footprint rendering.** The patches which represent the elliptical footprints with the diffused ink and water are rendered into a diffused layer buffer.
- **Diffusing footprint rendering.** The patches which represent the elliptical footprints in the diffusing are rendered directly into the frame buffer with the background of the updated diffused layer buffer.

3.2.2 *Footprints processing*

Concerning reconstruction of ink effects in the painting image, we sample the footprint textures from the original image. Although we model a brush footprint as an ellipse in our brush footprint model (Subsection 3.1), in rendering process, we still apply the rectangular patches mapped with circular binary masks (Figure 6(a)) for simplification of the geometric computation. In order to implement the key features derived from the generation of a brush footprint on Xuan paper, we focus on two crucial issues:

- **Jagged problem.** When we implement a brush stroke with the overlapped footprints, their sampled textures will be rendered more than once. So a rendering artifact caused by a jagged problem must exist in rendering results (Figure 7(a)). To solve this problem, we need to pre-compute the overlapping intensity of ellipses and distribute the intensity to each layer of the ellipses. Therefore, we add a preparation rendering pass, where we output the patches with the elliptical masks.
- **Smoothing problem.** To make the brim of a footprint in the painting process more natural, we adopt Gaussian distribution function to smooth the brim of a footprint, which can also make the layer

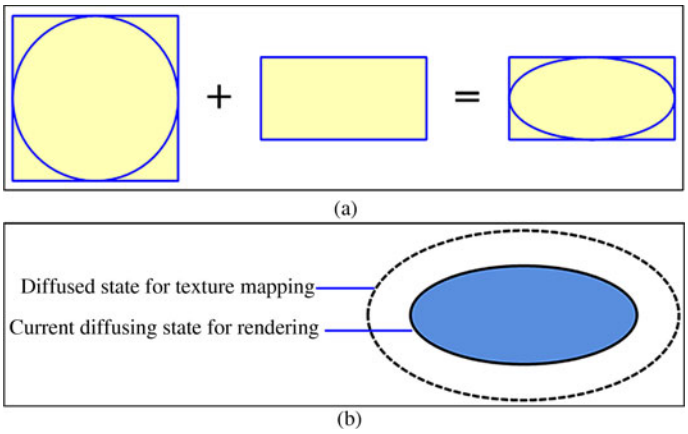

Figure 6 (a) A circular texture is mapped into a rectangular patch to produce an ellipse; (b) state transformation.

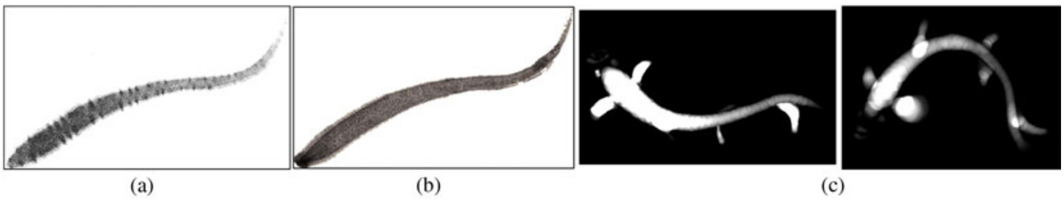

Figure 7 Our solution to the jagged problem. (a) Rendering result with the jagged error; (b) the acceptable result by our solution; (c) accumulation buffer by adding up all layers of ink intensity.

distribution better for the same pixel. So we use the Gaussian kernel as a pattern texture instead of the circular binary mask. Supposing a point $p(x,y)$ in the screen space, we formalize the increment of the overlapping intensity at a certain time t (the patch displaying at t) as Gaussian function $G(T(t))$. And the integrated intensity I at $p(x,y)$ is defined as

$$I = \int_0^t G(T(t))\mathrm{d}t, \tag{4}$$

where $T(t)$ is the corresponding texture coordinate of the patch displaying at t. And a discrete form of I is

$$I = \sum_0^i G(T(i)). \tag{5}$$

Implementation. As shown in Figure 7(c), we distribute the total overlapping intensity on the pixel $p(x,y)$ to each layer of the ellipses by assigning the transparency to the alpha channel. Combining the linear interpolation of color blending with transparency, we devise the blending equation:

$$\begin{cases} A_i = \alpha_i^2 + (1-\alpha_i)\cdot A_{i-1}, \\ A_n = 1, \end{cases} \tag{6}$$

where two terms, α_i and A_i, are the input and output alpha value of the ith patch respectively, and obviously the final alpha value of the overlapped patches A_n is assigned 1. Using Eq. (6), we can solve the input alpha value α_i with the condition of $\alpha \in [0,1]$:

$$1 = A_n \approx \sum_0^n \alpha_i^2. \tag{7}$$

Then as we apply Gaussian distribution function as the weight coefficient, the exact input alpha value α_i is

$$\alpha_i = \sqrt{\frac{G(T(i))\cdot A_n}{I}} = \sqrt{\frac{G(T(i))}{I}}. \tag{8}$$

Finally, the patches are rendered by sampling the color values from the original image with the texture coordinates T_i in the blending mode of Eq. (9). Thus we obtain the reasonable rendering result (Figure 7(b)).

$$C_i = \alpha_i \cdot c_i + (1 - \alpha_i) \cdot C_{i-1}, \tag{9}$$

where c_i and C_i are the input and output full RGBA color value of the ith patch, respectively.

3.2.3 *Geometry-based diffusion*

In order to present the dynamically diffusing effect of ink and water, as well as produce the same ink effects as the input artwork, we need to map the textures sampled from the original artwork onto the geometric patches. In the following, by applying two transformations (Figure 6(b)) we sample and map the texture to the vertices of the patch:

• **Diffused state transformation.** It is to transform the patch into the diffused state in screen space, which is the final scale after diffusing and denoted by M_{dd}. After that, the screen coordinates of the patch vertex within an interval $[-1, 1]$ can be simply transformed into a texture space $[0, 1]$ in the diffused state.

• **Diffusing state transformation.** It is to transform the patch into current diffusing state in screen space. The result is directly inputted to the rasterizer stage to inform GPU how to render an ellipse in a current scale. As this transformation changes with the time, it is denoted by $M(t)$.

When a patch is sent to input assembler, in vertex shader, we apply M_{dd} to the original position of vertex and obtain the texture coordinates for sampling. Then we apply $M(t)$ to obtain the true position of the patch vertex in screen space. The following pseudo codes present the whole procedures in GPU:

Algorithm 1. RenderFlowPatch

1: **In Vertex Shader MainVS**(V_{in}):
2: $V_{\text{out}}.UV \leftarrow M_{\text{dd}} V_{\text{in}}.\text{position} * \text{float2}(0.5, -0.5) + 0.5$
3: $V_{\text{out}}.\text{position} \leftarrow M(t) V_{\text{in}}.\text{position}$
4: **In Pixel Shader MainPS**(P_{in})**:**
5: $C \leftarrow \text{Fish.sample}(P_{\text{in}}.UV)$
6: $C.\alpha \leftarrow \text{sqrt}(G(P_{\text{in}}.UV)/I) * C.\alpha$

4 Experimental results

The approach depicted in this paper is implemented into an animation system which represents a real painting process of an image of Chinese ink painting. Taking the painting image in Figure 1(a) as an example, the processing of our system involves the following main procedures:

• Using the function of image segmentation, we decompose the painting image into several brush strokes with minimum interaction.

• In a preprocessing step of our system, we describe the shape of each brush stroke by binarization, the smoothed contour and skeleton.

• Based on the contour of a stroke's shape, our system estimates the trajectory of the stroke, which records the path of brush movement and presents the key features of brushwork (i.e. Zhong-feng and Ce-feng).

• Finally after modeling the brush footprints along the estimated trajectory as ellipses, the system renders them to be an animation of the painting process by the textures of the original paintings.

4.1 Results

As shown in Figure 8, we demonstrate the results of two animations of painting process with a series of their snapshots, respectively. Besides, the examples in this paper have been produced within 6–15 minutes.

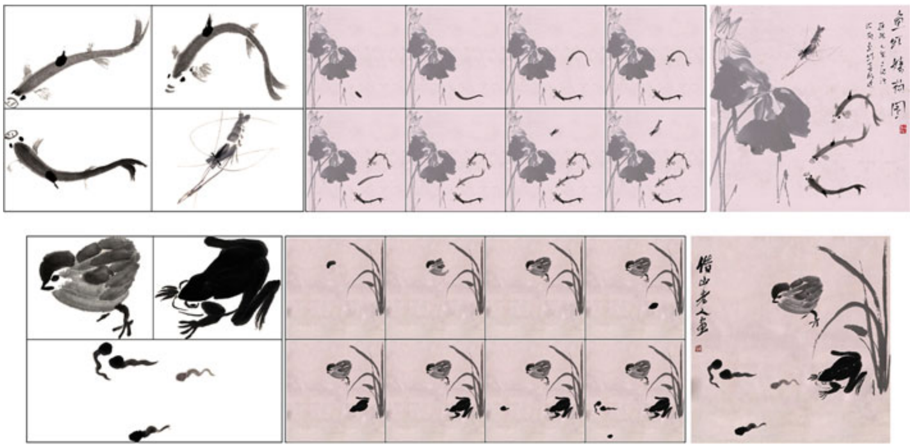

Figure 8 Guided by painting knowledge in Chinese ink painting, we convert several input static painting images (left) into a dramatic animation of a reconstruction of the artistic painting process (middle and right).

4.2 Evaluation

As far as we know, the brush trajectories in most of the existing work are explicitly specified as user interaction, except two remarkable works [17,19]. Through comparison with them, we evaluate the correctness and effectiveness of our approach.

4.2.1 *Animation of Chinese calligraphy*

First we have a similar motivation, that is, to provide a tool to appreciate and educate Chinese traditional arts and culture. Nevertheless, we consider that the skeleton-based method on trajectory extraction is not applicable to brush trajectory in Chinese ink painting for two reasons: some useless branches resulting from skeleton algorithm, except the noises, will interfere shape description of a stroke (Figure 9(b)); and "Feng" patterns with key points in [17] are merely suitable for extracting trajectories of basic and calligraphic strokes in Kai style which contains Ni-feng and Shun-feng. This promotes us to propose our approach to reproducing painting process of Chinese ink painting.

4.2.2 *Shape-driven trajectory approach*

We compare our results with those that can be obtained with the shape-driven approach which uses the minimization of an energy function to estimate the brush trajectory [19]. Figure 9(a) shows an example that is an input shape in [19] and where the strokes are also common in Chinese ink painting. According to the brushwork knowledge of Chinese calligraphy, two basic strokes in this character are mostly written by Zhong-feng except parts where the stroke's direction is changed and the related footprints need Ce-feng. So our trajectory approach is able to recover the brush trajectories of Zhong-feng (yellow curves in Figure 9(e)) and fit the strokes to character's shape while the trajectories by shape-driven approach [19] are greatly deflected to the shape boundaries (red curves in Figure 9(e)).

In Figure 9, for two basic brush strokes Figure 9(f) and (h), the trajectories by the approach in [19] are incoherent as some points stray from the trajectories and their trajectories cannot represent various fengs at the turnings of brush strokes (black points in Figure 9 (f) and (g)). But our trajectories are smooth curves and incline to the shape boundaries that implement the placement of Ce-feng when painting the turning parts of brush strokes (the different blue ellipses shown in Figure 9 (g) and (i)). To evaluate painting effects we compare our orchid paintings with the ones obtained using shaped-driven techniques. Figure 10 illustrates few examples; from left to right, the first painting is obtained with the system in [19] while the second and last ones show the orchid drawings generated by our system. The strokes in our paintings are more natural and show the essence of Chinese ink painting very well.

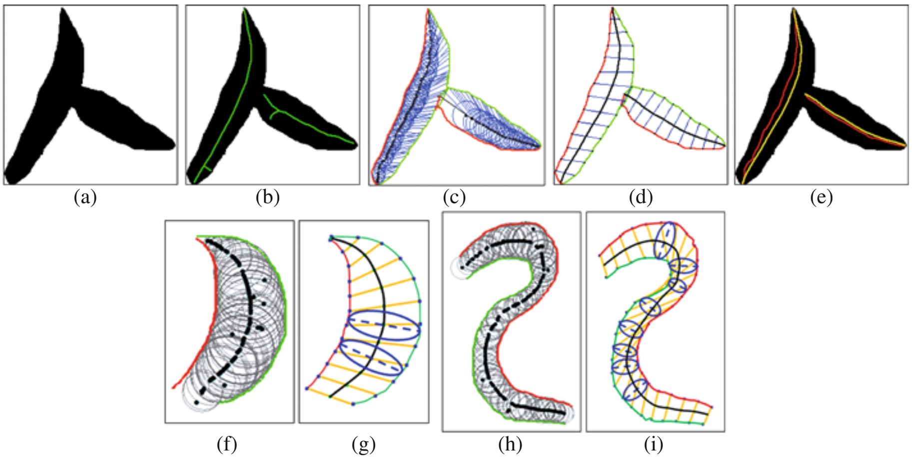

Figure 9 Comparison between Xie's algorithm and ours on estimation of brush trajectory. (a) Input shape of Chinese calligraphic character; (b) Strokes' skeletons with branches; (c)(d) brush trajectory estimation with the energy function in [19] and our contour-based method; (e) two results of trajectories by Xie's algorithm (red curves) and our method (yellow curves); (f)(h) the estimated trajectories (black points) by Xie's algorithm; (g)(i) our smooth trajectories by contour-based method (black curves) and footprints (blue ellipses).

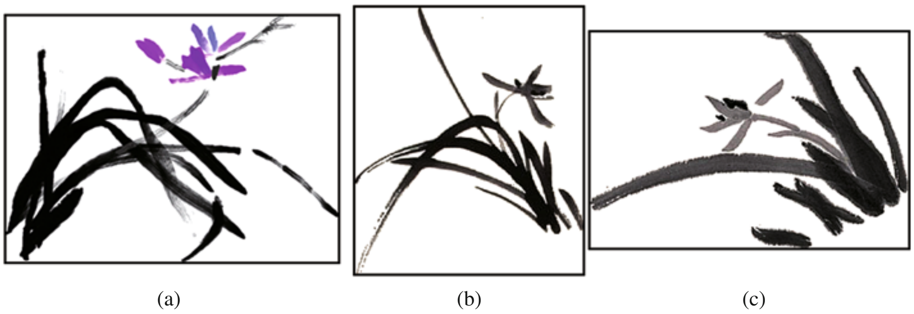

Figure 10 Drawing orchid paintings with our system (b)(c) and with Xie's system (a).

5 Conclusion

The animation of painting process of Chinese ink painting is a challenge work. In this paper, we choose bird-and-flower painting of Qi Baishi as the first starting step for animating the painting process because Qi Baishi is the greatest painter and bird-and-flower painting in great freehand style is a relative simple painting style. Compared to previous work, our method provides automatical brush trajectory estimation and rendering framework instead of the complex manual inputs. In addition, we adequately considered painting expertise and artistic ink effects in painting knowledge, which can make the animating process more natural and vivid. Our experiments show that our approach and animation tool can be suitable for the general brush strokes of complex shapes. Our application would illuminate and help people to understand the complexity of brushwork produced by hand.

However, animation of bird-and-flower painting is just our first step on this topic. Future research will address the following aspects:

• We need to improve some processing with more automation mechanism (e.g. brush stroke decomposition and brush footprint model).

• Until now, our approach limits to the simple Chinese ink paintings in great freehand style, where painting objects mainly cover flowers (plants), fish, insects, birds, etc. and are described by some flowing lines (i.e. brush strokes). In other words, as long as the painting object can be decomposed into brush strokes, our approach can redraw its painting process with the original ink textures of the strokes. From this point of view, we cannot handle the complex shapes with more random strokes and multiple ink layers, which are produced by more difficult brush techniques such as various types of "Cun", also meaning wrinkles, and "splash ink" in Chinese landscape painting. So we would extend our approach to more challenging and professional brush strokes in other varieties of Chinese ink painting.

Acknowledgements

The authors thank the associate editor and all the referees' for their constructive comments to improve our manuscript.

References

1 Chu N S H, Tai C L. MoXi: Real-time ink dispersion in absorbent paper. ACM Trans Graph, 2005, 24: 504–511
2 Yao F H, Shao G F, Yi J Q. Extracting the trajectory of writing brush in Chinese character calligraphy. Eng Appl Artif Intell, 2004, 17: 631–644
3 Strassmann S. Hairy brushes. In: Proceedings of SIGGRAPH. New York: ACM, 1986. 225–232
4 Way D L, Shin Z C. The synthesis of rock textures in Chinese landscape painting. Comput Graph Forum, 2001, 20: 123–131
5 van Laerhoven T, van Reeth F. Real-time simulation of watery paint: natural phenomena and special effects. Comput Animat Virt Worlds, 2005, 16: 429–439
6 Hsu S C, Lee I H H. Drawing and animation using skeletal strokes. In: Proceedings of SIGGRAPH. New York: ACM, 1994. 109–118
7 Su S L, Xu Y Q, Shum H Y, et al. Simulating artistic brushstrokes using interval splines. In: Proceedings of the 5th IASTED International Conference on Computer Graphics and Imaging, Hawaii, 2002. 85–90
8 Seah H S, Wu Z, Tian F, et al. Artistic brushstroke representation and animation with disk B-spline curve. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Valencia, 2005. 88–93
9 Wong H T F, Horace H S. Virtual brush: A model-based synthesis of Chinese calligraphy. Comput Graph, 2000, 24: 99–113
10 Xu S H, Tang M, Lau F, et al. A solid model based virtual hairy brush. Comput Graph Forum, 2002, 21: 299–308
11 Chu N S H, Tai C L. An efficient brush model for physically-based 3D painting. In: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Los Alamitos, 2002. 413–421
12 Lake A, Marshall C, Harris M, et al. Stylized Rendering techniques for scalable real-time 3D animation. In: Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering, Annecy, 2000. 13–20
13 Zhang S, Chen T, Zhang Y, et al. Video-based running water animation in Chinese painting style. Sci China Ser F-Inf Sci, 2009, 52: 162–171
14 Chan C, Akleman E, Chen J. Two methods for creating Chinese painting. In: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, 2002. 403–412
15 Xu S H, Xu Y Q, Kang S B, et al. Animating Chinese paintings through stroke-based decomposition. ACM Trans Graph, 2006, 25: 239–267
16 Yuan M L, Yang X B, Xiao S J, et al. GPU-based rendering and animation for Chinese painting cartoon. In: Proceedings of Graphics Interface, Montreal, 2007. 57–61
17 Yang L J, Li X S. Animating the brush-writing process of Chinese calligraphy characters. In: Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science, Shanghai, 2009. 683–688
18 He S X, Zhang S J, Xiao J L, et al. Extracting profile feature points by a method based on arc length. Opto-Electron Eng, 2004, 31: 59–62
19 Xie N, Laga H, Saito S, et al. IR2s: Interactive real photo to sumi-e. In: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering, Annecy, 2010. 63–71