JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26, 1397-1411 (2010)

Effective Wet-in-Wet Flow Synthesis for Ink Diffusion*

REN-JIE WANG AND CHUNG-MING WANG+
Senior Commercial School of Continuing Education
National Taichung Institute of Technology
Taichung, 404 Taiwan
E-mail: rjwang@ntit.edu.tw
+*Department of Computer Science and Engineering*
Institute of Networking and Multimedia
National Chung Hsing University
Taichung, 402 Taiwan
E-mail: cmwang@cs.nchu.edu.tw

Wet-in-wet flow effect is a famous phenomenon in Chinese ink paintings. In this paper, we propose a new two-stage algorithm to synthesize this renowned effect. Given a reference image, in the first stage, we render the reference image using a new color ink diffusion synthesis algorithm. This physically-based algorithm explores a new and more sensitive Kubelka-Munk (SK-M) equation. As a result, this new algorithm produces an ink diffused image, which has the similar tone to the reference image and offers better visual plausibility than our previous work [1]. In the second stage, we present a controllable flow effect technique in order to synthesize the wet-in-wet flow effect. In particular, the adaptive length line integral convolution is adopted to represent the global flow of the reference image. Given this global flow and luminance of the reference image, a controllable flow map is generated using the desired weight coefficients controlled by the user. Finally, we blend this controllable flow map with the ink diffused image rendered in the first stage. The blending takes advantage of the new SK-M equation again, synthesizing a Chinese ink diffused image with a notable wet-in-wet effect. Our algorithm has four advantages: visually pleasing, controllable, independent, and simple.

Keywords: image-based, physically-based, ink diffusion, wet-in-wet flow, line integral convolution, Kubelka-Munk (K-M) equation

1. INTRODUCTION

Chinese brush painting has developed continuously over a long period of time as an art of expression without using sketches and models. The artist paints with rapid, intuitive movements of the brush that convey an abstraction of the subject.

In computer communities, simulating Chinese brush painting can be classified into two categories. The first one is non-image-based, which focuses on modeling and simulation of an artist's hair brushes. The second category is the image-based approach, which renders a reference image with variant ink painting styles [1].

Ink diffusion is perhaps the most noticeable phenomenon in Chinese brush painting, which is caused by the microscopic capillary effect of absorbent paper [2-5]. Our previous work [1] proposed an image-based painterly rendering algorithm for automatically synthesizing an image with color ink diffusion. We suggested a mathematical model which

---

Received May 9, 2008; revised July 24, 2009; accepted September 23, 2009.
Communicated by Jiebo Luo.

[*] This paper was partially supported by the National Science Council of Taiwan, R.O.C. under grants No. NSC 96-2221-E-005-090-MY2, NSC 95-2221-E-005-046, NSC 94-2213-E-005-022, NSC 93-2213-E-005-018.

includes a K-M equation with a physical base to achieve this goal. Our previous work indicates a success of simulating ink diffusion from an image-based approach. However, one of significant drawbacks of their method is that the rendered results do not show a tone visually similar to the reference image. This is due to the fact that the K-M equation they employed is not sensitive enough to calculate the subtle change of the reflectance value for the mixed result.

Apart from ink diffusion, the wet-in-wet flow effect appears many times in Chinese ink painting works, as shown in Fig. 1. This effect is due to the fact that the surface of the wet paper allows the colloidal ink to spread and follow the direction of the water flow. Chu *et al*. [6] proposed a physically-based method for simulating ink dispersion in absorbent paper. Through a painting system, they demonstrated simulation results of various realistic effects with flow patterns. Their work indicated a succession of wet-in-wet flow simulation from a non-image-based approach. The wet-in-wet flow effect has also been simulated in the work of computer-generated watercolor [7], and is called "fluid flow effect." Unfortunately, most image-based algorithms do not keep pace with the advances of wet-in-wet flow simulation. This has significantly limited the simulation applications of this important and renowned appearance. Therefore, the need to provide wet-in-wet flow simulation from the aspect of image-based approach is clear.

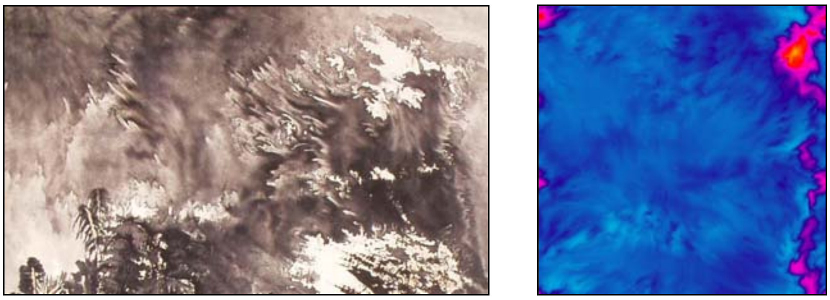

Fig. 1. Famous ink effects-diffusion and wet-in-wet flow; Left picture is real Chinese ink painting (by Feng-Qi Tang); Right picture is a simulated result created by Chu *et al.*'s work [6].

Fig. 2. Taiwan's Yushan (Mount Morrison) [1] (left) and its rendered images created by our new algorithm to simulate the wet-in-wet flow effect with controllable flow strength (middle and right). The image in the middle emphasizes the flow effect in the dark color (the mountain rocks) and the right image accentuates the effect in the light color (the clouds and sky).

In this paper, we propose two approaches to conquer the problem of tone difference and wet-in-wet flow effect. In particular, we present an image-based color ink diffusion synthesis (IBCIDS) algorithm, which employs a more sensitive K-M equation and also a new overlapping equation designed to produce visually pleasing ink-diffused images. These two new equations enable our algorithm to produce a tone similar to the reference image. The resultant ink-diffused image is more visually pleasing than the result of our previous work (see Fig. 4). In regard to simulating the wet-in-wet flow effect, we introduce the concept of the flow map. In particular, we present a controllable flow effect technique (CFET), which is designed to be independent of the IBCIDS algorithm in order to ease the complexity of the algorithm. In this technique, we apply the adaptive length line integral convolution (ALLIC) algorithm in order to depict the global flow of the reference image. Then, we refer to the luminance of the reference image together with the desired weighted coefficient provided by the user to construct a controllable flow map. Finally, we utilize the more sensitive K-M equation which we propose in order to blend the flow map with the resultant diffused image simulated by our IBCIDS algorithm. The final rendered outcome contains both an effective color ink diffusion result and the wet-in-wet flow effect (see Fig. 2). We emphasize that we do not mimic Chinese ink painting styles. Rather, we present a new way to generate another illustrious color ink diffusion effect, the wet-in-wet flow phenomenon. We leave Chinese ink painting styles as a separate and interesting topic to explore in the future.

The overview of this paper is as follows; section 2 takes a closer look at related research; section 3 introduces the proposed algorithm; section 4 shows the simulation results; Conclusions and future work are presented in section 5.

2. RELATED WORK

Research in Chinese ink painting simulation can be classified into two categories: non-image-based and image-based [1]. The first category does not need a reference image, and focuses on modeling and simulating an artist's hair brushes, such as digital painting systems or illustration generators. In this way algorithms generally try to produce realistic brush strokes on canvas [2-4, 8-13]. Some of these approaches emphasize their real-time benefit, and they employ texture mapping or some specific yet non-physical techniques. Lee [3] presented a practical technique to render oriental black ink paintings with realistic diffusion effects. His approach simulated a variety of paper types and black ink properties by use of specific algorithms. Others have employed mathematical or physical models and obtained remarkable results. Kunii *et al.* [2, 14] and Lee *et al.* [5] proposed a multidimensional diffusion model which simulated ink diffusion in absorbent paper. By physical analysis of real ink diffusion images, their model faithfully simulates the ink diffusion phenomenon. According to their approach, diffusion of diffuse ink (colloidal liquid) on the surface of paper should be considered as two separate diffusion processes. In other words, diffusion of water in the paper and diffusion of the motion of solid particles in water should be treated as different (but interlinked) processes. Guo and Kunii [4] proposed an interactive painting system for generating high quality and artistic calligraphy characters and black ink paintings. Their system was a physically-based algorithm which simulated the dynamic diffusion of liquid ink into absorbent painting paper. However, these physically-based approaches merely focus on simulating the diffusion of black ink. Chu *et al*. [6] presented a lattice Boltzmann equation (LBE) method which simulated ink dispersion in absorbent paper for art creation purposes. They demonstrated a digital painting system with various realistic ink dispersion effects, including complex flow patterns observed in real artwork, and other special effects.

The second category is image-based. Picture retouching systems belong to this category. Algorithms utilized in this way generally apply user-defined patterns or texture maps to a reference image in order to render some Chinese ink painting styles [15-17]. Yu *et al*. [16] described a two-stage framework for synthesizing Chinese landscape painting. They mainly employed some brush stroke texture primitivities (BSTP) to mimic the hand-made effects. Farbiz *et al*. [17] described a four step algorithm which automatically generated an Asian ink painting result from photographs. Their method allows the output image objects not to be in the same position as they are in the input image, and they try to convey the inner feelings of the artist just by using a few simple strokes on paper. Given a static calligraphic image, Wong *et al*. [18] developed an automated approach to analyze and synthesize a high quality calligraphic artwork with their virtual brush system. In this category, few researchers have proposed a color ink diffused method based on physical foundations. Some commercial packages produce ink diffusion effects with blur filters, but those packages are not physically-based approaches. Our previous work [1] proposed an image-based painterly rendering algorithm for automatically synthesizing an image with color ink diffusion. We suggested a mathematical model with a physical base to simulate the phenomenon of color colloidal ink diffusing into absorbent paper, and proposed a non-stroke-based (non-SBR) algorithm that allows an input color image to be automatically converted to the color diffused ink style with a visually pleasing appearance. However, our previous work had two defects. First, we do not simulate the wet-in-wet effects, even though artists in Chinese ink paintings always demonstrate more wet-in-wet effects in their works, such as wet-in-wet flow. Secondly, our results do not have a similar tone to the reference images. This inspires us to develop a new, effective algorithm which will be detailed in the next section.

3. WET-IN-WET FLOW EFFECT SYNTHESIS FOR INK DIFFUSION

In this paper, we propose a novel, image-based algorithm to support not only the color ink diffusion effect, but also the wet-in-wet flow effect. Our system consists of two independent parts: the image-based color ink diffusion synthesis (IBCIDS) and the controllable flow effect technique (CFET) (illustrated in Fig. 3). The IBCIDS algorithm simulates color ink diffusion first. Based on this result, the CFET generates a wet-in-wet flow effect, producing the final rendered result. We will describe each part in the following sections.

3.1 Image-Based Color Ink Diffusion Synthesis

The color ink used in Chinese ink painting is a colloidal liquid which mainly consists of water and pigment particles. We employ a physically-based color ink diffusion model which was originally proposed by Kunii *et al*. [2] and extended by our previous

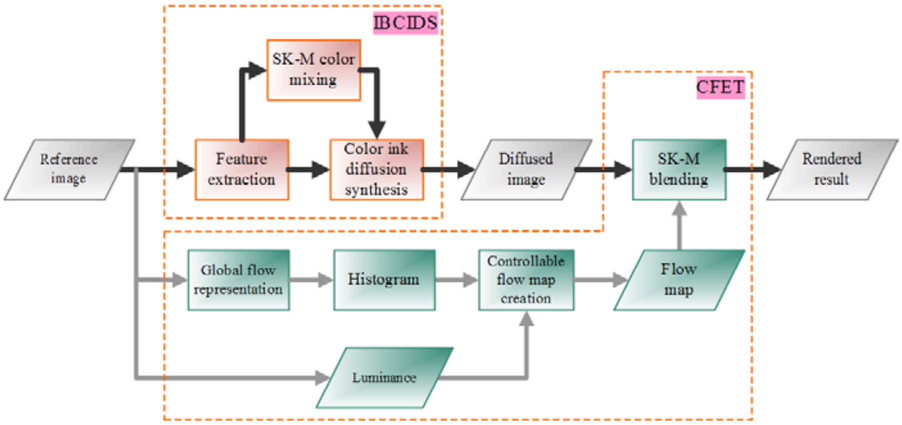

Fig. 3. System architecture.

work [1] which simulated color ink diffusing on absorbent paper. However, we emphasize that our IBCIDS is better than our previous work [1] because we employ a more sensitive K-M equation to mix the pigment color, and use a better equation which overlaps the deposit layer with a diffusion layer. Thus, our rendered result has a similar tone to the reference image.

Our IBCIDS consists of two layers which simulate the process of diffusion: the diffusion layer and the deposit layer. The diffusion layer is used to abstract non-features of the reference image, and the deposit layer reinforces the key features. When a drop of colloidal ink falls on the surface of absorbent paper, the water will spread through the paper structure until it is absorbed. At the same time, the pigment particles are carried by the flow of water. If the accumulated quantity of pigment is greater than some threshold (here we assume that threshold Q_{th} is equal to (1), the pigment particles will be deposited on the deposit layer; otherwise they will be diffused on the diffusion layer. The rendered color is generated by a new overlapping equation, which overlaps the diffusion and deposit layers as follows:

$$R_{i,j} = (\frac{P_{i,j} + G_{i,j}}{d_{i,j} + P_{i,j} + G_{i,j}} C_{i,j} + \frac{d_{i,j}}{d_{i,j} + P_{i,j} + G_{i,j}} \max(C_{i,j}, D_{i,j})). \tag{1}$$

Here $R_{i,j}$ denotes the rendered color of the pixel (i, j); $D_{i,j}$ denotes the color of the deposit layer; $C_{i,j}$ denotes the color of the diffusion layer. $d_{i,j}$ denotes the quantity of the deposit layer; $P_{i,j}$ is the quantity of the pigment; $G_{i,j}$ denotes the quantity of the water. Our new overlapping equation is more effective than our previous work since we take more factors into account, such as water flow and deposit quantity.

When artists paint, they often delineate some features and make others more abstract. Our previous work [1] suggested a two-phase feature extraction: the luminance division and color segmentation to simplify the input information; and the block variation to extract key characteristics. However, our previous work [1] used a fixed threshold to decide the key characteristics. It is not always suitable for all input reference images. Here, we assign a statistic value $\overline{VB}$ – the mean of the block variation $VB_{i,j}$, to the threshold to make the key characteristics more reasonable because every image has its own $\overline{VB}$. Our block variation equation is described as follows:

$$
\begin{aligned}
V_{i,j} &= \frac{1}{n^2}\sum_{i-\frac{n}{2}}^{i+\frac{n}{2}}\sum_{j-\frac{n}{2}}^{j+\frac{n}{2}}(\overline{L'}-L'_{i,j})^2,\\
VB_{i,j} &= \frac{V_{i,j}-VB_m}{VB_M-VB_m},\\
\overline{VB} &= \frac{\sum VB_{i,j}}{M\times N},
\end{aligned}
\tag{2}
$$

where $L'_{i,j}$ is luminance of the reference image on a pixel (i, j); and $\overline{L'}$ denotes the mean. $V_{i,j}$ is the variation of luminance on the pixel (i, j) within a block size of $n \times n$ pixels. Here, we use a block size of 7 × 7 pixels, but users can choose other desired block sizes. Block variation $VB_{i,j}$, which is normalized from the variation $V_{i,j}$, has a value between 0 and 1. VB_M denotes the maximum of the block variation, and VB_m denotes the minimum. If the reference image has a resolution of $M \times N$, we can derive the mean of the block variation $\overline{VB}$ from $VB_{i,j}$. By assigning $\overline{VB}$ to the threshold, block variation can generate adequate key features rather than the gradient magnitude.

The following algorithm can help to decide the quantity of pigment and the quantity of water by the block variation $VB_{i,j}$.

```
Algorithm PaintOnPaper (i, j)
1 if ($VB_{i,j} \geq \overline{VB}$) {            // key characteristic, thick ink
2       assign very little water
3       assign maximum pigment quantity}
4 else {                                         // user-defined ink
5       assign user-defined water
6       assign user-defined pigment quantity}
7       if (pigment quantity $pq_{i,j} \geq Q_{th}$){
8       $d_{i,j} \leftarrow pq_{i,j}$ }           // deposit to deposit layer
```

Line 1, we use the mean of the block variation $\overline{VB}$, a statistic value, in order to decide where the key characteristics are and to assign an adequate quantity to the water and pigment. We render the scene from light pigment to dark pigment in animation. If the user-defined pigment quantity is greater than 0.5 Q_{th} and less than Q_{th}, not only the key characteristics but also the remarkable edges will settle on the deposit layer. Those pigments located on the diffusion layer will diffuse to pale. The others located on the deposit layer still keep their color.

Fig. 4 presents diffused images generated by our IBCIDS (middle column) and the rendered results simulated by our previous work [1] (right column). We can see that our results have similar tones to the reference images rather than our previous work.

Fig. 4. A diffused image generated by our IBCIDS algorithm (middle column) and the rendered result simulated by our previous work [1] (right column). The pictures on the left column are reference images.

3.2 Controllable Flow Effect Technique

The CFET has to be independent of the IBCIDS in order to take advantage of facilitating the complexity of the whole simulation system. In the global flow representation phase, we mainly employ an adaptive length line integral convolution (ALLIC) which represents the global flow of the reference image. Then, we refer to the luminance of the reference image and the weight coefficient given by the user in order to construct a controllable flow map which is used to decide the resultant flow strength. The ALLIC, a technique converted from the line integral convolution (LIC) vector field, is employed to portray the reference image's global flow. The LIC method, originally developed for imaging vector field in scientific visualization, has the potential to produce images with directional characteristics [19, 20]. However, the LIC vector field, which has the same length and sharp visualization, is not suitable if directly used to enhance wet-in-wet effect. Thus, we adapt the LIC equation to an adaptive length line integral convolution (ALLIC) as shown in Eqs. (3.1) and (3.2).

$$l(x_0) = l_M - (l_M - l_m) \times L'_{x_0}, \tag{3.1}$$

$$I(x_0) = \int_{s_0 - l(x_0)}^{s_0 + l(x_0)} k(l(x_0) - s_0) T(\sigma(l(x_0)), \overline{L'}) ds. \tag{3.2}$$

In Eq. (3.1), x_0 represents the current pixel being processed and L'_{x_0} denotes its luminance. The length function $l(x_0)$ is controlled by l_M and l_m, which are user-defined maximal and minimal length, respectively. Eq. (3.2) calculates the intensity of an output pixel $I(x_0)$, where $k()$ denotes a normalized, one dimensional filter kernel with the length

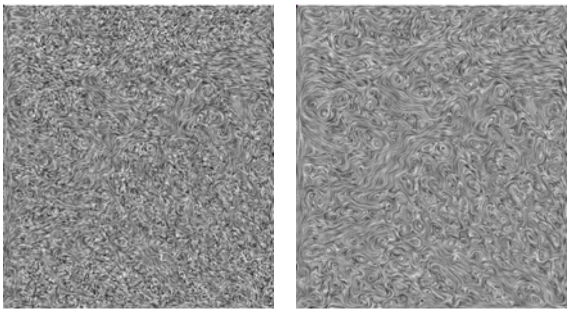

Fig. 5. ALLIC vector field with an adaptive length $L_M = 10$ and $L_m = 2$ (left) and LIC vector field with a fixed length $L = 10$ (right); The ALLIC vector field has variant lengths on different luminance providing sharp, fluent and versatile flow effects for the vector field.

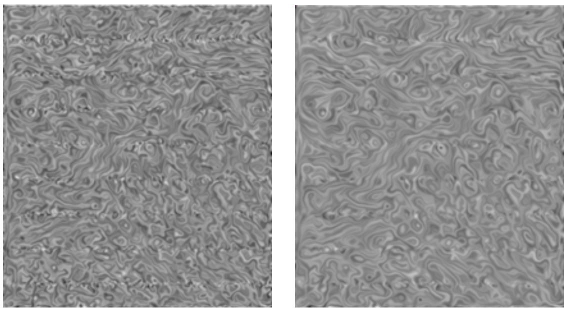

Fig. 6. A smooth ALLIC with $L_M = 10$ and $L_m = 2$ (left) and a smooth LIC vector field with $L = 10$ (right); In contrast to the smooth LIC vector field, the smooth ALLIC has variant lengths distributed on variant luminance providing more vivid and conspicuous flow effects.

of $2l(x_0)$. $T()$ is our proposed noise function, which is modified from the original noise function with two entries including the line of flow $\sigma(l(x_0))$ and the mean of the input image's luminance $\overline{L'}$. Our experiments show that a satisfactory wet-in-wet flow effect can be obtained by using these equations in the final step. Fig. 5 shows the difference between the ALLIC vector field and the LIC vector field. The ALLIC vector field has variant lengths on different luminance providing sharp, fluent and versatile flow effects for the vector field. Our maximum length l_M is 10 (*i.e.* maximum length is 20) and our minimum length l_m is 2 (*i.e.* minimum length is 4).

The ALLIC vector field is too sharp to produce an effective flow image. Thus, we use anisotropic diffusion to smooth it, but still keep the features clear. Anisotropic diffusion [21] is a well-known smooth filter in the digital image processing community. We employ the anisotropic diffusion filter provided by a commercial package, the Adobe Photoshop, to produce a smooth image. Then we smooth the sharp ALLIC vector field to generate a smooth ALLIC vector field. Fig. 6 shows the smooth ALLIC and the smooth LIC vector fields for comparison. The smooth ALLIC vector field has variant lengths distributed on variant luminance providing visually plausible results.

In the histogram phase, the smooth ALLIC vector field will be enhanced by the histogram algorithm to create an effective flow image. Then, the flow map $\varphi_{i,j}$ of pixel (i, j) can be generated by the following equation.

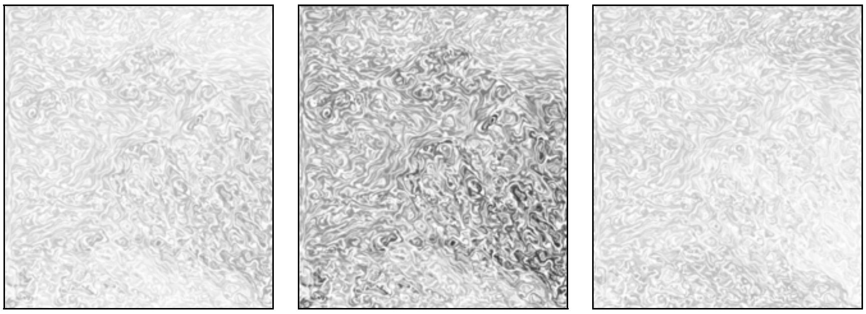

Fig. 7. Different flow maps created by effective flow image with variant α value: with α = 0.44 (left), with α = 1.0 (middle), and with α = 0.36 reverse (right).

$$\varphi_{i,j} = 1 - \alpha \cdot il_{i,j} \cdot \tilde{I}_{i,j} \tag{4}$$

where $\tilde{I}_{i,j}$ denotes the effective flow image; $il_{i,j}$ is the luminance on the reference image (i, j); and α is the weight coefficient (here default α = 0.5~1.0). Fig. 7 illustrates different flow maps generated by the effective flow image with the variant α value. The flow map is controllable. We can use a different flow map to emphasize dark color or light color. We also can choose the variant α value to control flow strength.

3.3 Rendering Algorithm

Our system consists of two independent parts: IBCIDS and CFET. The IBCIDS generates a color ink-diffused image from light pigment to dark pigment in animation. The CFET blends the ink diffusion with wet-in-wet flow effect in interaction. The rendering algorithm is described as follows:

```
Algorithm RenderMain ()
// $il_{i,j}$: luminance on a reference image ($i, j$)
// $iC_{i,j}$: color on a reference image ($i, j$)
// $ol_{i,j}$ : luminance on rendering canvas ($i, j$)
// $oC_{i,j}$: color on rendering canvas ($i, j$)
/* IBCIDS */
while (water not dry) {
    for (all light layer to dark layer k) {
        for (all $il_{i,j}$ in layer $k$) {
            if ($il_{i,j} - ol_{i,j} \geq O_T$ ) {                 // overlap threshold $O_T$
                PaintOnPaper(i, j)                                 // water, pigment
                $oC_{i,j} \leftarrow SK\text{-}M(iC_{i,j}, pq_{i,j}, oC_{i,j}, P_{i,j})$}}}    // SK-M color mixing
    ColorInkDiffusionSynthesis($G$, $P$, $oC$)}
/* CFET */
for (all $i, j$ ) {
    $oC_{i,j} \leftarrow SK\text{-}M(1\text{-}oC_{i,j}, \varphi_{i,j}, 0, 0)$}          // SK-M blending
```

In the IBCIDS phase, we render the diffused image on a rendering canvas from light pigment to dark pigment in animation, and every pigment layer has an interval of time (here we assume 6 iterations of diffused computation) to make diffusion processing. The ink diffusion will then continue computing until the water is dry. In the above algorithm, line 4 decides whether the pixel of the reference image is painted onto the rendering canvas or not. Line 6 is the SK-M color mixing. When a pixel with color $iC_{i,j}$, and pigment quantity $pq_{i,j}$, is painted onto the rendering canvas with color $oC_{i,j}$ and pigment quantity $P_{i,j}$, its resultant color should take account of the color on the rendering canvas. Our sensitive K-M equation is employed to approximate the result of mixing the color. In the CFET phase, line 9 is the SK-M blending. The SK-M equation plays a very important role in this phase. Thus, we introduce our new and sensitive K-M equation in the next section.

3.4 Sensitive K-M Equation

When using the K-M theory for a typical application to mix *Color1* with *Color2*, we need to determine the corresponding scattering coefficient S and the corresponding absorption coefficient A for each color. These coefficients are usually measured with spectral measurements from a layer with the known thickness x. Our previous work [1] has suggested a simplified method to approximate S and A. Here, we propose a more sensitive K-M (SK-M) equation inspired from Curtis' work [7].

We first assume the paper's colors are nearly white. Thus, we set the color *white* to a safe value (here assumed to be 0.9999998) and compute the coefficients (S, a, and b) for the RGB color value C by the following equations:

$$S = \frac{1}{b} \cdot \coth^{-1}\left(\frac{b^2 - (a - white)(a-1)}{b(1 - white)}\right),$$
$$a = \frac{1}{2}\left(white + \frac{C - white + 1}{white}\right), b = \sqrt{a^2 + 1}. \tag{5}$$

Note that for *Color1* and *Color2*, we need to determine its corresponding coefficients (S, a, and b). In our system, we compute the coefficients (S, a, and b) for color $iC_{i,j}$ and $oC_{i,j}$ using the above equations. Then, we compute the reflectance R_1 and transmittance T_1 for color $iC_{i,j}$ by setting the thickness coefficient x to $pq_{i,j}$, as shown in Eq. (6). This was Kubelka's optical compositing equation [7]. Similarly, we compute the R_2 and T_2 of the color $oC_{i,j}$ by setting the thickness coefficient x to $P_{i,j}$. Then, we can determine the overall reflectance R and transmittance T.

$$R = \sinh bSx/c,$$
$$T = b/c, \text{ where } c = a \sinh bSx + b \cosh bSx,$$
$$R = R_1 + \frac{T_1^2 R_2}{1 - R_1 R_2}, T = \frac{T_1 T_2}{1 - R_1 R_2}. \tag{6}$$

Here, we directly use the composite value R to express the result of color mixing. Fig. 8 compares the SK-M equation we propose with the original one in our previous work [1].

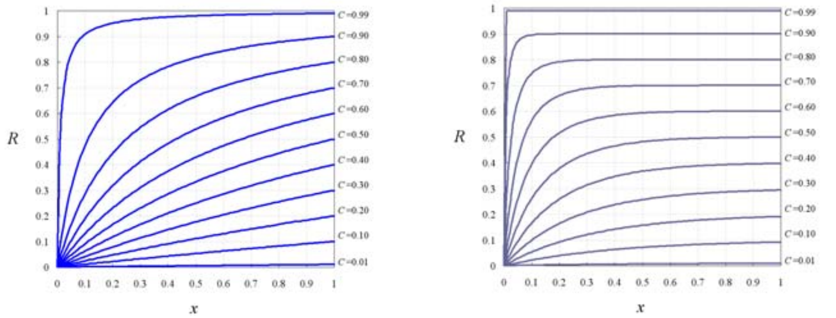

Fig. 8. The relation of the input color C, thickness x, and the output reflectance R using the SK-M equation we propose (left) and the original equation in our previous work [1] (right).

Clearly, our equation demonstrates that the responding reflectance R (left vertical axis) for different input colors C (right vertical axis) is sensitive with respect to various thicknesses x (the horizontal axis). The comparison shows that our new equation can generate more sensitive reflectance R than our previous work [1]. If the real paper color is not white, we can compose the final rendered result with the color of the paper by the above K-M equation. The reflectance R which is calculated by K-M equation is a non-linear curve. This is why the K-M theory is more suitable than the linear interpolation method for pigment-based rendering simulation. Because conventional pigments used in Chinese ink painting are different from those in watercolor, we leave the spectral measurements to pigments used in Chinese ink painting as a separate topic to explore in the future.

In the above rendering algorithm, line 9 uses a flow map φ to enhance a wet-in-wet flow effect by use of the K-M equation. Before being combined with the flow map, the diffused result contains paper texture and a diffusion effect, shown as Fig. 4 in the first row, middle picture. After combining the diffused result with the flow map (Fig. 7, middle picture), it becomes obvious that the color ink diffused within the paper fibers and the wet-in-wet flow effect is visually pleasing (shown as Fig. 2, middle). If we blend the diffused image (Fig. 4, the first row, middle picture) with a reverse flow map (shown in Fig. 7, right), the rendered result (shown in Fig. 2, right) will accentuate its wet-in-wet flow effect on the light color. The K-M blending produces a non-linear and visual pleasing result. This is why the K-M theory is also more suitable than the linear interpolation method for blending the diffused image with the flow map in our algorithm.

4. RESULTS

We implemented our system using C++ programming language and OpenGL on a personal computer. It contains a 3.0 GHz CPU, 1.0 GB RAM, and a video card with 128MB video RAM.

Given a diffused image rendered by our algorithm (Fig. 4, the first row, middle picture), Fig. 9 shows three images rendered by other approaches for comparison, including the alpha blending, directly applying the LIC technique only, and employing a commercial Adobe Photoshop's anisotropic diffusion filter. In comparison with Fig. 2 (middle and right pictures), our new algorithm renders an image demonstrating a visually pleasing

Fig. 9. Some resultant images created by other approaches (diffused image is Fig. 4, the first row, middle picture): using alpha blending on the diffused image (left); directly applying LIC algorithm on the diffused image (middle); employing Adobe Photoshop's anisotropic diffusion filter on the diffused image (right).

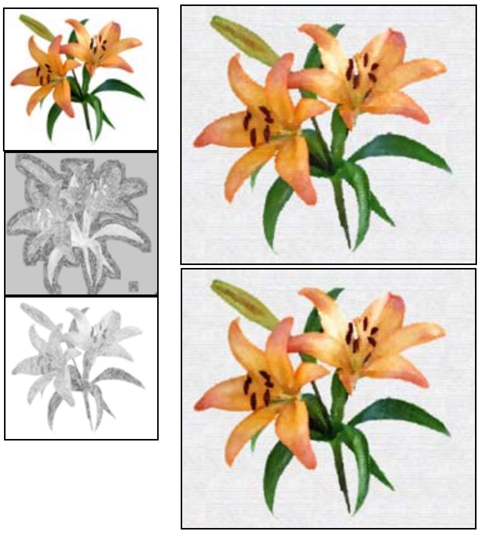

Fig. 10. Two images demonstrating controllable flow strength in simulating the wet-in-wet flow effect; The left top picture is the reference image; The right top image emphasizes the light color (flowers) using a reverse flow map (left middle) with $\alpha = 0.6$. The right bottom image focuses on the dark color (leaves) using the flow map with $\alpha = 0.7$ (left bottom).

Fig. 11. A black ink result simulated by our algorithm; The picture on the left top is the reference image [1] and the one in the left bottom is the reverse flow map ($\alpha = 0.44$); The middle image is our previous work's result and the right one is the result of this current paper. Clearly, our result demonstrates more visible ink diffusion style and the wet-in-wet flow effect in the cloud of the sky.

appearance with diffused ink and wet-in-wet flow style. The results imply that in order to generate the wet-in-wet flow effect, it is inappropriate to simply apply the alpha blending, the LIC algorithm, or the anisotropic diffusion filter on a diffused image. The results also show the power of the flow map and the SK-M blending process in our method.

Fig. 10 presents results generated by different flow maps to demonstrate that our wet-in-wet flow effect is controllable. We can directly use the flow map to emphasize dark color. In contrast, we can also reverse the flow map to emphasize the light color. As a result, a user can fully control the flow strength by changing the weight coefficient α.

Fig. 11 compares the black ink result created by our previous work [1] (middle picture) with our current work (right picture). From these images, we verify that our algorithm renders more visible ink diffusion and variant wet-in-wet flow effects, producing a visually plausible appearance.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we present a novel, image-based algorithm which synthesizes ink diffusion with the tone similar to the reference image and also with the wet-in-wet flow effect. We conclude that our algorithm has four advantages. First, the quality of the resultant images we rendered is more visually plausible than the result of our previous work. This is attributed to the use of our new rendering algorithm, the more sensitive K-M equation, our controllable flow technique, and the new overlapping equation. Secondly, our algorithm is controllable, allowing a user to manipulate the flow strength via variant flow maps when simulating the wet-in-wet flow effect. Thirdly, the algorithm is independent. This feature makes it possible to ingeniously weave the controllable flow effect technique (CFET) not only to our ink diffusion algorithm but also to other ink diffusion approaches. Thus, the integrated color ink diffusion algorithm has the double advantage of having significant features which include a feature-based, non-SBR, physically-based, and also a controllable wet-in-wet flow effect. Finally, it is simple. Without any strokes, a color image can be automatically converted to a visually pleasing appearance with both diffused ink and wet-in-wet flow style.

Although a physically-based approach has the benefit of realistic simulation, in the mean time, it has a problem with computation. Our first future task, therefore, is to speed up the computation. In addition, artists usually use a limited number of conventional pigments in Chinese ink painting; yet, a real image may contain more than a million colors. Our next task is to synthesize an even more realistic result by employing conventional pigments used in Chinese ink painting. This would make the color appearances shown in our results look more like genuine Chinese ink painting.

REFERENCES

1. C. M. Wang and R. J. Wang, “Image-based color ink diffusion rendering,” *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, 2007, pp. 235-246.
2. T. L. Kunii, G. V. Nosovskij, and T. A. Hayashi, “Diffusion model for computer animation of diffuse ink painting,” in *Proceedings of Computer Animation*, 1995, pp. 98-102.

3. J. Lee, "Diffusion rendering of black ink paintings using new paper and ink models," *Computers and Graphics*, Vol. 25, 2001, pp. 295-308.
4. Q. L. Guo and T. L. Kunii, "Nijimi rendering algorithm for creating quality black ink paintings," in *Proceedings of the Computer Graphics International*, 2003, pp. 152-59.
5. S. Lee, H. Y. Lee, I. F. Lee, and C. Y. Tseng, "Ink diffusion in water," *European Journal of Physics*, Vol. 25, 2004, pp. 331-336.
6. N. S. H. Chu and C. L. Tai, "MoXi: Real-time ink dispersion in absorbent paper," *ACM Transactions on Graphics*, Vol. 24, 2005, pp. 504-511.
7. C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischery, and D. H. Salesin, "Computer-generated watercolor," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 421-430.
8. H. T. F. Wong and H. H. S. Ip, "Virtual brush: A model-based synthesis of Chinese calligraphy," *Computer and Graphics*, Vol. 24, 2000, pp. 99-113.
9. C. W. Hsu, D. L. Way, and Z. C. Shih, "The synthesis of rock textures in Chinese landscape painting," *Computer Graphics Forum*, Vol. 20, 2001, pp. C123-C131.
10. C. Chan and E. Akleman, "Two methods for creating Chinese painting," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, 2002, pp. 403-412.
11. J. S. Yeh, T. Lien, and Y. M. Ouhyoung, "On the effects of haptic display in brush and ink simulation for Chinese painting and calligraphy," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, 2002, pp. 439-441.
12. S. H. Chu and C. L. Tai, "Real-time painting with an expressive virtual Chinese brush," *IEEE Computer Graphics and Applications*, Vol. 24, 2004, pp. 76-85.
13. D. L. Way, W. J. Lin, and Z. C. Shih, "Computer-generated Chinese color ink paintings," *Journal of the Chinese Institute of Engineers*, Vol. 29, 2006, pp. 1041-1050.
14. T. L. Kunii and G. V. Nosovskij, "Two-dimensional diffusion model for diffusion ink painting," *International Journal of Shape Modeling*, Vol. 7, 2001, pp. 45-58.
15. S. Z. Wen, Z. C. Shih, and H. Y. Chiu, "The synthesis of Chinese ink painting," in *Proceedings of National Computing Symposium*, 1999, pp. 461-468.
16. J. H. Yu, G. M. Luo, and Q. S. Peng, "Image-based synthesis of Chinese landscape painting," *Journal of Computer Science and Technology*, Vol. 18, 2003, pp. 22-28.
17. F. Farbiz, A. D. Cheok, and P. Lincoln, "Automatic Asian art: Computers converting photos to Asian paintings using humanistic fuzzy logic rules," in *Proceedings of the SIGGRAPH, Sketches and Applications*, 2003, p. 1.
18. S. T. S. Wong, H. Leung, and H. H. S. Ip, "Model-based analysis of Chinese calligraphy images," *Computer Vision and Image Understanding*, Vol. 109, 2008, pp. 69-85.
19. D. H. Stalling and C. Hege, "Fast and resolution independent line integral convolution," in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1995, pp. 249-256.
20. C. M. Wang and J. S. Lee, "Using ILIC algorithm for an impressionist effect and stylized virtual environments," *International Journal of Visual Languages and Computing*, Vol. 14, 2003, pp. 255-274.
21. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, 1990, pp. 629-639.

Ren-Jie Wang (王仁傑) received his Ph.D. and M.S. degrees in Computer Science Department from National Chung Hsing University, Taichung, Taiwan, in 2007 and 2001, respectively, and his B.S. degree in Information and Computer Education Department at the National Taiwan Normal University, Taipei, Taiwan, in 1990. His research interests are in computer graphics, non-photorealistic rendering, data hiding, and multimedia systems.

Chung-Ming Wang (王宗銘) received the B.S. degree in Applied Mathematics from the National Chung Hsing University, Taiwan, in 1984, and the Ph.D. degree in Computer Science from the University of Leeds, Leeds, U.K., in 1992. From August 1986 to August 1988, he was a systems analyst in the President Enterprise Cooperation and the PAL Company. He is currently a professor at the Department of Computer Science and Engineering, Institute of Networks and Multimedia, National Chung Hsing University, Taiwan. His research interests include computer graphics, color science, virtual reality, multimedia systems, and three-dimensional watermarking and steganography. Dr. Wang has won three Dragon Thesis Awards, funded by the Acer Computers, and Outstanding Paper Awards from the Computer Society of the Republic of China. He is a member of ACM, IEEE Computer Society and Eurographics.