AN IMAGE EFFECT CREATION SYSTEM

HSIAU WEN LIN*, HWEI JEN LIN†§, FU WEN YANG†, HSIAO WEI CHANG‡ and YUE SHENG LI†

*Department of Information Management
Chihlee Institute of Technology, New Taipei City, Taiwan, R.O.C.
†Department of Computer Science and Information Engineering
Tamkang University, New Taipei City, Taiwan, R.O.C.
‡Department of Computer Science and Information Engineering
China University of Technology, New Taipei City, Taiwan, R.O.C.
§086204@mail.tku.edu.tw

Received 24 March 2014
Accepted 30 July 2014
Published 12 November 2014

This paper proposes an image effect creation system that can create various types of effects for images, to satisfy different needs of users for specific design goals. Some operations needed for creating image effects, such as flow-based bilateral filter, flow-based Gaussian filter, curve-shaped filters, line drawing, pencil texture generator, and modified shock filter, are first proposed, and various image effects that can be converted by the proposed system are illustrated. The experiments show very rich results in terms of image effects.

Keywords: Image abstraction; bilateral filter; Kuwahara filter; Gaussian blur filter; non-photorealistic rendering; image stitching; shock filter.

1. Introduction

The goal of computational photography is to help users accomplish difficult image and video editing tasks effectively and effortlessly add effects and textures to images. Various methods for performing such works have been proposed in the literature, including segmentation,2,6 anisotropic diffusion,24 nonphotorealistic rendering,13,19,20 colorization,13,19,20 image stitching,1,7 and tone adjustment,21 image abstraction,14–16,27,28,30 pencil drawing,22,29,30 painterly style, cartoon animation,27,28,30 watercolor painting,5,18 and some other types of effects.

Most of these tasks are based on the use of image processing techniques, such as Gaussian blur filtering, DoG, XDoG, edge detection, soft quantization, bilateral filtering,15,27,28,30 Kuwahara filtering, etc.14

Two important visual properties of image abstraction are the absence of texture details and the increased sharpness of edges as compared to photographic images. Winnemöller et al.28 presented an automatic image abstraction framework that
abstracts imagery by modifying the contrast of visually important features. They reduced contrast in low-contrast regions using an approximation to anisotropic diffusion, and artificially increase contrast in higher contrast regions with difference-of-Gaussian edges. Kyprianidis and Döllner\textsuperscript{15} worked on the separated kernel of bilateral filter and proposed to approximate the bilateral filter by a separated implementation that first filters in gradient and then in tangent direction instead of original separated implementation that first filters in horizontal and then in vertical direction. The edge lines generated by DoG are also improved in similar way. As a result, it can obtain more smoothed curves. Bousseau \textit{et al.}\textsuperscript{5} presented a method for creating watercolor-like animation in two main steps: Applying textures that simulate a watercolor appearance; and creating a simplified, abstracted version of the video to which the texturing operations are applied.

Many techniques of generating the pencil sketch effect have been proposed in the literature.\textsuperscript{12-22,29} For designing a pencil filter to simulate the pencil sketch, Xie \textit{et al.}\textsuperscript{29} generated a black noise image from a reference image and used the tone of the input image to guide the distribution of the black noise. Their proposed filter is easy to apply in real-time cases but has the limitation of a fixed shading direction. Lu \textit{et al.}\textsuperscript{22} proposed a pencil drawing method. First, forward difference is applied on the given image to obtain a gradient map. Each pixel is then performed contour enhancement with a line segment filter of the same direction as the tangent direction. A pencil drawing like image is then produced. Papari and Petkov\textsuperscript{23} extended glass patterns (GP)\textsuperscript{9} to the continuous case and showed continuous glass patterns (CGP) can be effectively used for artistic imaging applications. The general idea is to replace the natural texture present in an input image with synthetic painterly texture that is generated by a CGP, whose geometrical structure is controlled by the gradient orientation of the input image. Their proposed method can produce an effect like the painting “The Starry Night” by Van Gogh that contains many flow-like curves.

This paper proposes an image effect converter system that can create various different types of effects to images, to satisfy the different needs of users for specific design goals, including image abstraction, pencil drawing, cartoon animation, watercolor painting, and some other stylistic imaginaries. All this work is accomplished by some extended versions or combinations of existing tools, including DoG, XDoG, bilateral filtering, soft quantization, tone adjustment, CGP, and shock filter.

Section 2 gives a brief review of some existing basic operators used for creating image effects. Section 3 proposes some operators that can be used to produce some image effects, and various image effects are illustrated in Sec. 4. The results are presented in Sec. 5 and conclusions are drawn in Sec. 6.

2. Some Existing Operators

2.1. \textit{Optional color quantization (OCQ)}

In standard quantization, an arbitrarily small luminance change can push a value to a different bin, thus causing a large output change for a small input change, which is
particularly troublesome for noisy input. With soft quantization, such a change is spread over a larger area, making it less noticeable. A significant advantage of soft color quantization implementation is it preserves temporal coherence and also reduces computation time. As a soft quantization, optional color quantization (OCQ)\textsuperscript{28,30} determines a quantized image according to the local variations of pixel \(x\), as shown in (1), where \(\Delta q\) is the bin width, \(q_{\text{nearest}}\) is the bin boundary closest to intensity \(f(x)\), and \(\varphi_q\) is a parameter controlling the sharpness of the transition from one bin to another. To make it a local adjustment method, the user is offered a trade-off between reduced color variation and increased quantization artifacts by defining a target sharpness range \([\Lambda_1, \Omega_1]\) and a gradient range \([\Lambda_d, \Omega_d]\). The calculated gradients were clamped to \([\Lambda_d, \Omega_d]\) and then a new value was generated for \(\varphi_q\) by mapping them linearly to \([\Lambda_1, \Omega_1]\).

\[
Q(x, q, \varphi_q) = q_{\text{nearest}} + \frac{\Delta q}{2} \tanh(\varphi_q \cdot (f(x) - q_{\text{nearest}})).
\]  

(1)

2.2. Tone adjustment

Lu \textit{et al.}\textsuperscript{22} found unlike the highly variable tones in natural images, pencil drawings mainly consist of two basic tones. For very bright regions, artists do not draw anything to show white paper; while heavy strokes are used to accentuate boundaries and highlight dark regions. In between these two tones, mild tone strokes are produced to enrich the layer information. One example is shown in Fig. 1.\textsuperscript{22} With these findings, they proposed a parametric model to represent the target tone distribution.

![Fig. 1. Tone distributions in a natural image and in a pencil drawing. (a) Input natural image, (b) pencil sketch by an artist, (c) histogram of (a) and (d) histogram of (b).\textsuperscript{22}](image-url)
For an image, two values $T_1$ and $T_2$ ($0 < T_1 < T_2 < 255$) are given to divide the image into three portions Low, Mid, and High. If $0 \leq I(x, y) < T_1$ then pixel $(x, y)$ is classified into Low; if $T_1 \leq I(x, y) < T_2$ then pixel $(x, y)$ is classified into Mid; if $T_2 \leq I(x, y) \leq 255$ then pixel $(x, y)$ is classified into High. The mean $\mu_d$ and the standard deviation $\sigma_d$ of the portion Low are used to define the distribution Dark, as given in (2). Another Gaussian distribution Bright is defined in (3), where the value $\mu_b$ set to 255 and $\sigma_b$ is the standard deviation of the portion High. Finally, the uniform distribution Mild defined in (4) is used to enrich the pencil drawing.

Finally, the weighted sum is chosen to integrate the distributions Dark, Bright, and Mild into a tone image. One example is shown in Fig. 2.

$$\text{Dark}(v) = \frac{1}{\sqrt{2\pi\sigma_d}} \exp \left( -\frac{(v - \mu_d)^2}{2\sigma_d^2} \right), \quad 0 \leq v \leq 255,$$

$$\text{Bright}(v) = \frac{1}{\sigma_b} \exp \left( -\frac{(225 - v)}{\sigma_b} \right), \quad 0 \leq v \leq 255,$$

$$\text{Mild}(v) = \begin{cases} 
\frac{1}{T_2 - T_1}, & \text{if } T_1 \leq v \leq T_2, \\
0, & \text{else}.
\end{cases} \quad (4)$$

Fig. 2. An example of tone adjustment. (a) Original image, (b) histogram, (c) distribution Bright, (d) distribution Mild, (e) distribution Dark, (f) combination of (c)–(e) with ratio 5:2:1 and (g) result.
2.3. Continuous glass pattern

A GP\(^9\) is generated by a linear differential equation and corresponds to the trajectories which solve the differential equation of a vector field. An example of GP\(^{23}\) is shown in Fig. 3.

Papari and Petkov\(^{23}\) extended the classical discrete GP\(^9\) to the continuous case, the so-called CGP, and used CGP to produce a nice artistic effect in photographic images. CGP works by tracing and finding the maximum value of an image of random noise along a constructed vector field of the original image, as shown in Figs. 4(a)–4(c). This method can produce image effect similar to the painting “The Starry Night” by Vincent van Gogh, as shown in Fig. 4(d).

Let \(I_{\text{EPS}}\) be the image obtained by performing an edge preserving smoothing filtering on a given image \(I\). Then another image \(\nabla_\sigma I_{\text{EPS}}\) is generated by performing Gaussian blur filtering. The gradient value of each pixel \(r\) in \(\nabla_\sigma I_{\text{EPS}}\) is evaluated and the corresponding angle \(\Theta_\sigma(r)\) of the gradient direction is determined. The vector field \(v(r)\) with initial degree \(\Theta_0\) can then be generated, as shown in (5).

\[
v(r) = \alpha \left( \cos(\Theta_\sigma(r) + \Theta_0), \sin(\Theta_\sigma(r) + \Theta_0) \right).
\]  

(5)

To form a painterly texture image \(B_v\) for image \(I\), an image \(N_s\) of random noise is given, and for each point \(r\) in \(N_s\), a path along the vector \(v(r)\) in \(N_s\) is traced and the maximum value along the path is found and assigned to \(B_v(r)\). The intensity distribution of the CGP image \(B_v\) is then adjusted by performing histogram equalization \(\eta\) to obtain the synthetic painterly texture (SPT) \(U(r)\), as shown in (6). Finally, SPT is added to the smoothed image \(I_{\text{EPS}}\) to form a painterly CGP image \(y(r)\), as shown in (7). An example of CGP operation is given in Fig. 5.

\[
U(r) = \eta[B_v(r)],
\]  

(6)

\[
y(r) = I_{\text{EPS}} + \lambda U(r).
\]  

(7)

![Fig. 3. An example of GP.\(^{23}\) (a) Vector field \(((y^2 - 1) + (1/3)xy, (1/3)(y^2 - 1) - xy))\), (b) the trajectories solving the corresponding differential equation and (c) a corresponding GP.](image-url)
3. Proposed Operations

This section proposes some operations, which can be used to produce some image effects, including flow-based bilateral filter, flow-based Gaussian filter, curve-shaped filter, line drawing, and pencil texture generator.

Fig. 4. Continuous glass pattern. (a) Original image, (b) vector field, (c) corresponding CGP and (d) “The Starry Night”.

Fig. 5. An example of CGP. (a) Original image, (b). SPT and (c) resulting painterly CGP image.
3.1. Flow-based bilateral filter

A bilateral filter is an edge-preserving and noise reducing smoothing filter\textsuperscript{15,27,28,30} which replaces the intensity value $I(x)$ at each pixel $x$ with a weighted average $f(x)$ of intensity values of pixels in the $n \times n$ neighborhood $N(x)$, as shown in (8), where $g_1(x)$ is a Gaussian distribution on Euclidean distance and $g_2(x)$ is a Gaussian distribution on intensity difference.

$$f(x) = \frac{\sum_{x_i \in N(x)} g_1(x_i) g_2(x_i) I(x_i)}{\sum_{x_i \in N(x)} g_1(x_i) g_2(x_i)}.$$ \hfill (8)

The conventional bilateral filter calculates the average of values within an $N \times N$ square block neighborhood of each pixel, which is time consuming. To speed up the process, Pham and van Vliet\textsuperscript{25} proposed a method by performing one-dimensional bilateral filter in the horizontal and vertical directions separately. However, this method poorly matches more complex features such as textured regions, which was then improved by the orientation-aligned bilateral filter (OABF) proposed by Kyprianidis and Dölker.\textsuperscript{15} Instead of the two fixed directions they traced neighbors along the gradient and tangent directions, as shown in Fig. 6(a). To further improve the abstraction result, the so-called flow-based bilateral filter (FBBF) considers the neighbors along the gradient flow and tangent flow instead of those in straight directions is proposed, as shown in Fig. 6(b).

For calculating the gradient and tangent directions for each pixel, the responses of the Sobel filter in three channels of the RGB color space in both the horizontal and vertical directions are first evaluated, namely $H = [R_h, G_h, B_h]^T$ and $V = [R_v, G_v, B_v]^T$. The responses in two directions can then be used to form the

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{Two versions of bilateral filter, where the red dots denote pixel $x$, and the blue lines/curves and black line/curve are the traced paths along gradient direction/flow and tangent direction/flow, respectively. (a) Orientation-aligned bilateral filter and (b) flow-based bilateral filter (color online).}
\end{figure}
covariance matrix $ST$, called the structure tensor, as shown in (9). The eigenvalues and corresponding eigenvectors of $ST$ can be evaluated using (10), where the eigenvectors corresponding to the larger eigenvalue and the smaller eigenvalue indicate the gradient direction and tangent direction of $x$, respectively.

$$ST = \begin{bmatrix} H^T H & H^T V \\ V^T H & V^T V \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix},$$  
(9)

$$\lambda_{1,2} = \frac{E + G \pm \sqrt{(E - G)^2 + 4F^2}}{2}, \quad v_1 = \begin{bmatrix} F \\ \lambda_1 - E \end{bmatrix}, \quad \text{and} \quad v_2 = \begin{bmatrix} \lambda_2 - G \\ F \end{bmatrix}. \quad (10)$$

To trace the tangent flow, we move from $x$ in the tangent direction to a new point $x'$, and evaluate the tangent direction of $x'$, and move in the direction to another new point $x''$. Repeating this process for a certain number of steps can trace a tangent flow of $x$. In a similar way, the gradient flow of each point $x_0$ is traced on the tangent flow of $x$ following the gradient direction of $x_0$ and those of the points traversed.

With the direction $t_i$ and moving step $l_i$ for a pixel $c_{i-1}$ along the gradient/tangent direction, the next sampling pixel $c_i = c_{i-1} + l_it_i$ can be computed, where $t_i$ and $l_i$ can be evaluated using (11) and (12), respectively.

$$t_i = \text{sgn}(v_2(c_{i-1}) \cdot t_{i-1}) \cdot v_2(c_{i-1}),$$  
(11)

$$l_i = \begin{cases} \frac{c_{i-1,x} - [c_{i-1,x}] - \frac{1}{2} \text{sgn}(t_{i,x})}{t_{i,x}}, & \text{if } |t_{i,x}| \leq |t_{i,y}|, \\ \frac{c_{i-1,y} - [c_{i-1,y}] - \frac{1}{2} \text{sgn}(t_{i,y})}{t_{i,y}}, & \text{if } |t_{i,x}| > |t_{i,y}|. \end{cases}$$  
(12)

The neighboring pixels along the gradient/tangent flow are collected in such a way to form a newly defined neighborhood (or mask). The bilateral filter is then applied to this neighborhood taking the pixel distance and pixel difference to the center pixel $x$ as weights. To achieve image abstraction, one may iteratively perform this process on the entire input image until some stop criteria are met.

Since methods based on the bilateral filter preserve high-contrast edges, they generally fail for high contrast images where either no abstraction is performed or too much detail is removed. This typically results in an inconsistent abstraction. To solve this problem, we examine the sum of difference between the current sample point and previous sample point and the difference between filter center point and the current sample point. If the sum is larger than a given threshold $D_{i'}$, we assume that an edge is encountered and the sampling on this direction is stopped.

### 3.2. Flow-based Gaussian filter

The traditional Gaussian filter is applied on a round shaped mask centered at the target pixel to blur an image, including the line texture which is sometimes regarded...
as an important feature and needs to be preserved. Lakshmanan\textsuperscript{17} proposed a separable directional Gaussian filter which enables the Gaussian blur operated along the local direction of the filter center, resulting in directional smoothing. To achieve a better result, the directional Gaussian filter is modified by collecting neighbors of each pixel from its tangent flow and gradient lines, as shown in Fig. 7(b), where the red point is the filter center, black points are collected from the tangent flow, and blue points are collected from the gradient lines.

The advantage of this flow-based Gaussian filter is displayed in Fig. 8. Figure 8(b) shows the hair above the forehead of the girl was blurred and became unclear, but Fig. 8(c) shows that the line textures of the hair were well preserved.

### 3.3. Curve-shaped filters

To simulate strokes in line drawings, Lu \textit{et al.}\textsuperscript{22} performed convolution on the gradient map $G$ of the input image with a line segment. They used a set of eight line segments with directions at $45^\circ$ apart and denoted the line segments as $\{L_i|i=0, \ldots, 7\}$. A line segment filter $L_i$ is considered a block of size $1 \times f$, rotated about degree $\theta_i$ with value 1 in each of $f$ positions, where $\theta_i = i\pi/8$ for $i = 0, 1, 2, \ldots, 7$. The response map for a certain direction is computed as $G_i = L_i * G$. The length $f$ of $L_i$ is set to $1/30$ of the image height or width, empirically, and $*$ is the convolution operator, which groups gradient magnitudes along direction $i$ to form the filter response map $G_i$. The classification is then performed by selecting the maximum value among the responses in all directions, as shown in (13).

$$G_i(p) = \begin{cases} G(p), & \text{if } \arg \max_i \{G_i(p)\} = i, \\ 0, & \text{else} \end{cases} \quad (13)$$
With the map set \( \{ C_1 \} \), lines at each pixel can be also generated by convolution, which is expressed as 
\[ S_0 = \sum_{i=1}^{8} (L_i \odot C_1). \]
The pixel values are then inverted and mapped to \([0, 255]\) to obtain the final pencil stroke map \( S \), as shown in Fig. 9.

To preserve the curvature of the image features, aside from the line segment filters, we provide curve-shaped filters in various directions. Figure 10 shows a line

![Image](image-url)

**Fig. 8.** Comparison of traditional Gaussian filter and the flow-based Gaussian filter. (a) Original image, (b) traditional Gaussian filter \((\sigma = 2)\) and (c) flow-based Gaussian filter \((\sigma_y = 0.33, \sigma_t = 9)\).

![Image](image-url)

**Fig. 9.** Examples of the line segment filters with different radians. (a) \( \theta_0 = 0 \), (b) \( \theta_1 = \pi/8 \), (c) \( \theta_2 = \pi/4 \) and (d) \( \theta_3 = \pi/2 \).

![Image](image-url)

**Fig. 10.** Filters in horizontal direction. (a) Line segment filter, (b) concave-up curve-shaped filter and (c) concave-down curve-shaped filter.
segment filter, a concave-up curve-shaped filter, and a concave-down curve-shaped filter, all in the horizontal direction. Therefore, for $n$ directions, there shall be $3n$ filters.

### 3.4. Line drawing

Lu et al.\textsuperscript{22} convolved the gradient map with line segment filters to introduce line drawings. However, when applied on curviform edges, the resulting line drawing is not smooth as desired. To tackle this problem, three different versions of line drawings are proposed, namely $LDrawing_1$, $LDrawing_2$, and $LDrawing_3$. The first two versions, replace the line segment filters with curve-shaped filters and the flow-based Gaussian filters, respectively. That is, the former convolves the gradient map with curve-shaped filters and the latter convolves the gradient map with flow-based Gaussian filters, to introduce line drawings. The last version $LDrawing_3$ applies image abstraction introduced in Sec. 3.1 to the given image for one iteration, evaluates the gradient map of the resulting image, and convolves the gradient map of the given image with flow-based Gaussian filters introduced in Sec. 3.2. Each of these versions has its own characteristics. Figure 11 shows some results produced by these methods. $LDrawing_3$ is very suitable for images with fine line textures such as hair, whose flow structure needs to be preserved. A result of $LDrawing_3$ is given in Fig. 12.

### 3.5. Pencil texture generator

To generate pencil texture on a tone image, an image $Noise$ for the tone image is produced as follows. For each pixel $(x, y)$ in the tone image, as shown in (14), a random number random is generated, and the $\text{Tone}(x, y)$ is saved in $\text{Noise}(x, y)$ if random is greater than a given threshold $T$, where $T$ is used to control the density of the image Noise. Pencil texture $\text{Texture}_i(x, y)$ in direction $i$ can then be generated by

![Fig. 11. Comparison of line drawings based on different filters. (a) Original image, (b) Lu et al., (c) $LDrawing_1$, (d) $LDrawing_2$, (e) $LDrawing_3$ and (f)–(j) close-ups for (a)–(e).](1454005-11)
convolving the image Noise with the line segment filter $L_i$ mentioned in Sec. 3.3, as shown in (15). Figure 13 shows examples generated pencil texture in directions $\theta = \theta_0 = 0$ and $\theta = \theta_2 = \pi/4$, and a combination of them.

To keep the transparency of the image rendered with the generated pencil texture, we adjust the generated pencil texture using (16), where $0 < \alpha < 1$. Figure 14 shows some results of transparency adjustment with different values of $\alpha$.

$$\text{Noise}(x, y) = \begin{cases} \text{Tone}(x, y), & \text{if random} < T, \\ 0, & \text{else,} \end{cases} \quad (14)$$

![Fig. 12. An example of LDrawing 3. (a) Original image and (b) result.](image1)

![Fig. 13. Examples of generated pencil texture. (a) $\theta = 0$, (b) $\theta = \pi/4$ and (c) combination of (a) and (b).](image2)

![Fig. 14. Transparency adjustment for texture image given in Fig. 13(c) with (a) $\alpha = 0.25$, (b) $\alpha = 0.5$ and (c) $\alpha = 0.75$.](image3)
Texture\(_i(x, y) = L_i \ast \text{Noise}(x, y), \quad i \in \{1, \ldots, N\}, \quad (15)\)

Transparency(I(x), \(\alpha\)) = (1 - \(\alpha\)) \* I(x) + \(\alpha\) \* 255. \quad (16)\)

### 3.6. Modified shock filter

The shock filter\(^{11}\) performs a dilation process near a maximum and an erosion process around a minimum in an image. Whether a pixel belongs to the influence zone of a maximum or a minimum can be decided based on whether the Laplacian is negative or positive. Iterating this procedure produces a sharp discontinuity (shock) at the borderline between two influence zones.

Weickert\(^{26}\) proposed a modified version, called the coherence-enhancing shock filter (CESF). The sign of the Laplacian is replaced by the sign of the second derivative in the direction of the major eigenvector of the smoothed structure tensor.

The CESF produces image with maze-like curves. To create another image effect, like block print, we make some simple modification on the result \(f(x)\) of CESF by \((17)\) in each color channels. As a result, there are only eight colors preserved in the resulting image of high contrast.

\[
B(x) = \begin{cases} 
255, & \text{if } f(x) \geq 128, \\
0, & \text{else.}
\end{cases} \quad (17)
\]

### 4. Proposed Image Effect Creation Methods

This chapter illustrates various image effects, including image abstraction, pencil sketch, watercolor painting, and some other artistic imaging. Each of these effects is described in the following subsections.

#### 4.1. Image abstraction and cartoon animation

To create an image abstraction effect, an image pyramid\(^{8,14}\) is used to operate the flow-based bilateral filter on multiple scales. A pyramid processing is proposed that is simpler than the one presented in Ref. 14. Suppose there are \(n\) levels in the pyramid, where level \(i - 1\) is the downsampled result of level \(i\). Therefore, the top level (level 1), has the smallest size and the bottom level (level \(n\)) with the original image has the largest size. Once the pyramid is constructed, Gaussian blur is performed on each image in the pyramid.

The processing of the pyramid is performed in a coarse-to-fine manner, with intermediate results being propagated up the pyramid. Initially, flow-based bilateral filtering and upsampling are carried out on image \(I_1\) on the top level. The result is then combined with image \(I_2\) on the level below (level 2) to form an intermediate image \(Layer_2\). Again, flow-based bilateral filtering and upsampling are performed on the \(Layer_2\). The result is then combined with image \(I_3\) on the level below (level 3) to form another intermediate image\(Layer_3\). The procedure is continuously performed
downward until $Layer_n$ is obtained. Finally, flow-based bilateral filtering is carried out on $Layer_n$ to obtain the resulting image $I_p$. The processing of the pyramid is given in Fig. 15.

For image abstraction, the image pyramid processing is iteratively performed $t$ times with $n$ levels of pyramid. A soft color quantization, such as OCQ, can be carried out or edges can be added to the abstracted image if a paint-like or cartoon animation is desired.

4.2. Pencil sketching

For the pencil sketching effect, first three steps are performed: (1) one of the three versions of the line drawing presented in Sec. 3.4, (2) tone adjustment introduced in Sec. 2.2, and (3) pencil texture rendering presented in Sec. 3.5. Let $Lin$, $Ton$, and $Tex$ be the results of the three steps mentioned above, respectively. Finally, the three images are combined by the use of image multiplication operation to form the resulting pencil sketching like image $Mtp(Lin, Multi(Ton, Tex))$, where the image multiplication operator $Mtp$ is defined in (18).

$$Mtp(I, J)(x) = I(x) \cdot J(x)/255.$$  

4.3. Watercolor painting

Watercolor paintings possess characteristics of abstraction. Besides, their main characteristics is the pigments are mixed with water but do not dissolve totally. This leads to a nonhomogeneous repartition of water on the canvas after drying and darkened edges due to pigment migration. To create an effect of watercolor painting according to these properties, abstraction on the given image is carried out with a large standard deviation, say $\sigma_r = 20$, for smoothing color variation between regions of different colors. The result of abstraction $Abs$ is then combined with the line drawing $Lin$ by image multiplication given in (20) to obtain the resulting image $Wat = Mtp(Abs, Lin)$.

4.4. Artistic imaging with CGP

Since a painterly CGP image was rendered with a synthetic texture image obtained from the vector field of the original image, the consistency of the flow is affected by the image content. Images with more noise lead to ordered flow, while images with less noise lead to disordered flow. The resulting images of CGP can be modified with
some other operations to produce some artistic styles, such as flow-based Gaussian blurring and shock filtering.

5. Experimental Results

This section presents results of some effects created by the proposed image effect creation system, which was built on Windows environment. Basically when the input image and the effect are specified, the system can generate an image with the effect automatically with a set of default values. Also, the system preserves the interface for the users. Users are allowed to set some specified parameters interactively to produce images of different styles.

Figure 16 shows a result of image abstraction by performing three iterations of the image pyramid processing, using two levels of pyramid with parameters setting $\sigma_r = 10$ and $\sigma_d = 10$ for the flow-based bilateral filter. Figure 16(c) shows a result of cartoon animation, by performing optional color quantization and edge adding on the abstracted image given in Fig. 16(b).

Figures 17(b) and 17(c) show the result of $LDrawing$ and the result of pencil sketch with $LDrawing$, where before performing image multiplication the tone
image $T_{\text{on}}$ was produced by performing image abstraction twice and tone adjustment with $T_1 = 160$, $T_2 = 200$, and weights = 10:1:2, and the pencil texture image $T_{\text{ex}}$ was adjusted by using transparency adjustment with $\alpha = 0.25$. To make a more natural result, Fig. 17(c) is modified by processing its line drawing image $L_{\text{in}}$ using Gaussian filter with $\sigma = 1$ and then using transparency adjustment with $\alpha = 0.5$. We may make a color pencil sketch effect by simply adding color in the original image to a pencil sketch result. Figure 17(d) shows an example of color pencil sketch.

Figure 18 gives a result of water painting effect. Figure 19 shows some results of van Gogh’s painting style effect, which was done by performing abstraction and CGP. Figure 20 shows some results of block print effect by performing modified shock filtering and flow-based Gaussian filtering. Figure 21 shows more results of combinations of different operators. Figure 22 shows a page of interface of the proposed system.

Fig. 18. An example of watercolor painting effect. (a) Original image and (b) result.

Fig. 19. Examples of van Gogh’s painting style effect. (a) and (c) Original images and (b) and (d) results.
Fig. 20. Example of block print effect. (a) and (c) original images and (b) and (d) results.

Fig. 21. Examples of modified CGP. (a), (e) and (i) Original images. (b), (f) and (j) CGP. (c), (g) and (k) CGP + flow-based Gaussian filtering. (d), (h) and (l) CGP + flow-based Gaussian filtering + shock filtering.
6. Conclusion and Future Works

This paper proposes an image effect creation system. First, some operations, which can be used to produce some image effects, are presented, including flow-based bi-lateral filter, flow-based Gaussian filter, curve-shaped filter, line drawing, pencil texture generator, and modified shock filter. Based on the proposed operations and some existing operations, various image effect convertors are then presented, including image abstraction, pencil sketch, watercolor painting, van Gogh’s painting style, block print, and some other artistic imaging effects. Our system can automatically produce a variety of image effects, and a friendly interface for users outlined here was built.

Aside from the above mentioned effects, there are many more other effects that have not yet been classified and named. This will be accomplished in our future work. In the future, computation of Graphic Processing Unit (GPU) to accelerate the implementation speed will be also considered and further extension of the work outlined here will also be sought.

Acknowledgment

This work was supported in part by the National Science Council of Taiwan, R.O.C. under the grant MOST 103-2221.E-032-044.

References


**Hsiau Wen Lin** received her M.S. degree from the Department of Informatics at Fo Guang University, Taiwan in 2003 and her Ph.D. from the Department of Computer Science and Information Engineering of Tamkang University, Taiwan, in 2007. She is currently an Assistant Professor at the Department of Information Management at Chihlee Institute of Technology, Taiwan. Her research interests include pattern recognition, adaptive learning, eLearning and data mining.

**Hwei Jen Lin** received her B.S. degree in Applied Mathematics from National Chiao Tung University, Hsinchu, Taiwan in 1981 and her M.S. degree and the Ph.D. in Mathematics from Northeastern University, Boston, USA in 1983 and 1989, respectively. She is currently a Full Professor at the Department of Information Engineering of Tamkang University, Taipei, Taiwan. Her current research interests include pattern recognition, image processing, computer algorithms, and intelligent computation.

**Fu-Wen Yang** received his B.S. degree in Industrial Education from National Taiwan Normal University, Taipei, Taiwan in 1988 and his M.S. degree and Ph.D. in Information Engineering from Tamkang University, Taipei, Taiwan in 1998 and 2005, respectively. He is currently a teacher at the Department of Information Engineering of New Taipei Municipal Tam-Shui Vocational High School, and a part-time assistant professor at the Department of Computer Science and Information Engineering of Tamkang University as well. His research interests include image processing, pattern recognition and computational intelligence.

**Hsiao-Wei Chang** received his B.S. degree in Applied Mathematics from Fu-Jen Catholic University, Taipei, Taiwan in 1980, his M.S. degree in Computer Science from Texas A&M University, Texas, U.S.A. in 1986 and his Ph.D. in Computer Science and Information Engineering from Tamkang University, Taipei, Taiwan, R.O.C. in 2011. He is currently an associate professor at the Department of Computer
Science and Information Engineering of China University of Science and Technology, Taipei, Taiwan. His current research interests include pattern recognition, image processing and logic design.

Yue Sheng Li received his B.S. and M.S. degrees from the Department of Information Engineering of Tamkang University, in 2010 and 2012, respectively. He is now a Ph.D. student in the Department of Information Engineering of Tamkang University. His research interests include pattern recognition and iris recognition.