



Article

# Tangent-Based Binary Image Abstraction

Yi-Ta Wu <sup>1</sup>, Jeng-Sheng Yeh <sup>2</sup>, Fu-Che Wu <sup>3,\*</sup> and Yung-Yu Chuang <sup>1</sup>

<sup>1</sup> Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan; dada@cmlab.csie.ntu.edu.tw (Y.-T.W.); cyy@ntu.edu.tw (Y.-Y.C.)

<sup>2</sup> Computer and Communication Engineering, Ming Chuan University, Taoyuan County 333, Taiwan; jsyeh@mail.mcu.edu.tw

<sup>3</sup> Computer Science and Communication Engineering, Providence University, Taichung City 43301, Taiwan

\* Correspondence: fcwu@pu.edu.tw; Tel.: +886-04-2632-8001

Academic Editor: Gonzalo Pajares Martinsanz

Received: 17 December 2016; Accepted: 14 April 2017; Published: 26 April 2017

**Abstract:** We present a tangent flow-based image abstraction framework that turns a color or grayscale image into a two-tone image, which contains only black and white color, but retains sufficient information such that the viewer can still recognize the main content of the original image by observing the output of a two-tone image. Usually, the relevant visual information is presented by the edge. Thus, the algorithm will enhance the edge content first by the tangent flow to preserve its detail. We use filters to smooth the input image to reduce the contrast in low contrast regions and enhance some important features. Then we turn the image into a two-tone image as output. We can reduce the size of the image significantly but retain sufficient information to capture the main point of the image through our method. At the end of our process, we provide a smoothing step to smooth the two-tone image. Through this step, we can get an artistic black-and-white image.

**Keywords:** tangent flow; image abstraction; image simplification; image stylization; two-tone image; binary image

## 1. Introduction

When we look at a picture, the visual system works together with the brain to interpret what we see by performing sophisticated inference to organize the information and make it reasonable. Of course, the ability of our visual intelligence has limits. Tufte [1], for example, suggested making detail as light as possible to keep the main point of a presentation perceptually salient and avoiding adding any detail that does not contribute to the argument of a presentation. Thus, drawing an image in an abstract way is helpful to reduce the detail while keeping the main point of the image.

Image abstraction or image simplification is an interesting problem in the field of non-photorealistic rendering (NPR). Through the methods of image abstraction, we can get a simplified illustration which re-represents the original image; the information is reduced, but the main content of the original image is still recognizable. Furthermore, because the abstraction of images is non-photorealistic, they somewhat provide an artistic or comic-like version of the original images.

Converting an image into a binary image is a widely-studied task in computer graphics. Thus, there are many kinds of image binarization methods for different purposes, such as thresholding, line drawing, halftoning, etc. Because those methods are designed for different types of tasks, the resulting binary images are significantly different.

In this paper, we present a binary image abstraction method based on image tangent flow and turn the input colored image or grayscale image into a binary image containing only black and white color. We smooth the input image with a bilateral filter, which is a non-linear filter that can reduce the noise while preserving the edges of an image. Then we use the smoothed image to compute the image

tangent flow and use it to smooth the image with line integral convolution further. Finally, we enhance the important details and turn the processed image into a binary illustration. It is optional to smooth the binary image further to make the result more abstract. More details are discussed in Section 3.

## 2. Related Work

Abstraction is considered a technique for effective visual communication that attracts the viewer's attention to the most meaningful parts and allows the viewer to understand the primary purpose without much effort. This skill is widely used in art and design. Abstracting an image often involves reducing the information of content. If it is the main point that we want to present, then it will be preserved, and other less essential components will be concealed. Thus, clarifying the significant structures in an image is an important task for image abstraction. Decarlo and Santella [2] proposed a computational approach to stylizing and abstracting photographs. They used a model of human perception and recorded users' eye movements to identify the significant elements in a picture. Winnemöller et al. [3] presented an automatic, real-time abstraction framework that abstracted images by modifying the contrast of visually important features. They also provided a flexible quantization method on the abstracted images, which resulted in a cartoon or paint-like effects and became another artistic illustration style. Kyprianidis and Döllner [4] presented a framework, which used oriented filters, creating simplified stylistic illustrations automatically. Such works aim at providing a system that creates colorful abstractions and gets pretty good results (see Figure 1). Now we further want to present a framework that abstracts images using the minimal colors, just in black and white.

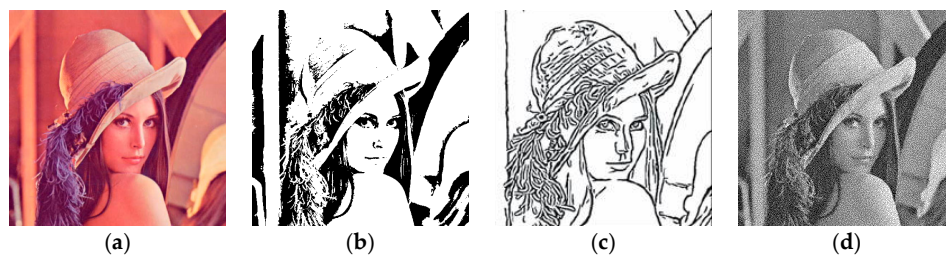


**Figure 1.** Image abstraction. (a) input (b) DeCarlo and Santella [2] (c) Winnemöller, et al. [3] (d) Kyprianidis and Döllner [4].

Many algorithms aim at creating black-and-white representations of a continuous-tone image, such as thresholding, line drawing, and halftoning. Those methods are designed for different purposes, so the resulting black-and-white images are significantly different. We show an example in Figure 2.

Image thresholding is the most well-known technique that turns an image into a binary image, and it is a long-studied problem in computer vision. Thresholding is a common task in many computer vision and graphics applications, such as document processing, part inspection, medical image processing, counting function, and so on. The goal of image thresholding is to classify every pixel as either dark or light or saying black or white. A simple method is to choose a threshold based on a certain characteristic (most of the cases are based on the intensity of pixels) so that pixels whose measured value are below the chosen threshold would become black, and the other pixels would

become white. Unfortunately, it is not easy to decide a suitable threshold that works well to get the desired result. It is more likely that such a good threshold does not even exist. Instead of choosing a fixed global threshold, determining a local threshold for each pixel by observing its intensity and the intensities of its neighbors is a common and straightforward solution for improving the quality of the results. Those algorithms which use the local information of being processed pixels are classified as local thresholding methods or adaptive thresholding methods. There are numbers of adaptive thresholding methods, such as Chan, FHY, et al. [5], Sauvola and Pietikäinen [6], and Bradley and Roth [7]. Some further methods and comparisons can be found in Yu-Ting Pai et al. [8] and Singh, T. Romen, et al. [9]. It is considered that using local information to decide a suitable threshold for each pixel would get a better result than choosing a global threshold arbitrarily. Although adaptive thresholding works well for some specific tasks, it does not produce an attractive abstraction of images, since the noises may stand out in a local window. Thus, without any other processes, the results of thresholding may look dotted and noisy, confusing the main point.



**Figure 2.** Different kinds of binary images. (a) input (b) thresholding (c) line drawing (d) halftone of Lena.

Line drawing, just as its name implies, aims at extracting the edges of the images. Those algorithms use the small amount of information, i.e., lines, to represent the original continuous-tone image. Canny edge detection [10] is one of the most common algorithms and is considered to be a standard method to find edges. Canny edge detection has been used by many applications—for example, Salisbury, Michael P., et al. [11] and DeCarlo and Santella [2]—and the resulting edges are suitable for computer vision applications. Though canny edge detection has been proposed for decades and there are some various modifications such as Meer and Georgescu [12], the resulting edges are often unsuitable for stylistic applications without any further processing, since the extracted edges often have equal thickness. On the other hand, the difference-of-Gaussians (DoG) filter, which is an approximated of Laplacian-of-Gaussians proposed by Marr and Hildreth [13], produces a set of stylistic edges. Gooch, et al. [14] presented a method for creating black-and-white illustrations for photos of human faces based on a DoG filter. Compared with canny edge detection, the DoG filter models the edges in a more pleasing way and is more suitable for pictorial illustrations. However, due to the nature of Gaussian filters, which have isotropic filter kernels, the resulting pixels that represent the edges may not clearly reveal the sense of directedness and thus may look less like lines. Kang et al. [15] presented a method that automatically generated a line drawing from a photograph. A result of their work is shown in Figure 2c. The main idea of their approach is to make the DoG filter anisotropic. They presented a flow-based anisotropic DoG (FDoG) filter which takes the directions of the local image structures in DoG filter into account. Thus, the extracted edges are smoother, stylistic, and enhance the coherence of the lines. Wang, Shandong, et al. [16] further extended the flow-based filters. They improved the quality of line drawings and made the resulting artistic expressions more similar to hand-drawing styles. Compared with other edge detections, flow-based filters—such as the FDoG filter—are more likely to generate the edges that suitable for pictorial illustrations. Although those algorithms can create a good line drawing of an image, it is sometimes too detailed to be an abstraction of images. In some cases, we want some areas to be colored black or white to conceal less important information.

Halftoning is a method for creating an illusion of grayscale images with many black and white dots, imitating the sense of continuous-tone by dithering the black dots on the white areas with different densities. The human eye perceives the diffusion as a mixture of the colors, and such dithering techniques take advantage of the human eye's tendency to mix colors to create continuous-tone-like images. Thus, halftoning can create grayscale-like images by using only black and white colors and is widely used in printing. It is considered that error-diffusion-based dithering is the most efficient technique because of its short computational time while keeping a good quality. Floyd-Steinberg dithering [17] is the most famous error-diffusion dithering technique, and there are some other error-diffusion methods such as Jarvis, Judice, and Ninke Dithering [18] and Stucki Dithering [19]. Furthermore, Ostromoukhov [20] proposed an error-diffusion matrix which restricted the choice of the distribution coefficients by pre-analysis work. Some other algorithms aim at providing continuous-tone-like but also well-structured halftone images, such as Pang, Wai-Man, et al. [21], Jianghao Chang et al. [22] and Li, Hua, and David Mould [23]. Although halftoning can simulate the continuous-tone well and is widely used in printing, it still suffers from some problems. For example, halftone images are difficult to edit. One should re-compute a new halftone image after the modifications. Furthermore, halftone images may look kind of noisy on some displays because of how halftoning simulates the continuous-tone: dithering many dots with different densities. For those reasons, halftone images are not suitable for creating abstract illustrations.

Mould and Grant [24] presented an automatic algorithm for converting a grayscale image into a black-and-white illustration. They propose a base plus detail algorithm for black and white stylization, where graph cuts is used to obtain a coarse (base) level, local adaptive segmentation is used to get a fine (detail) level, and the two are combined to produce a final image that both preserves details and has large flat-colored regions where warranted. Compared with their work, we make use of full RGB color channels and provide some intuitive parameters to adjust the results. We will discuss these later in Section 3.

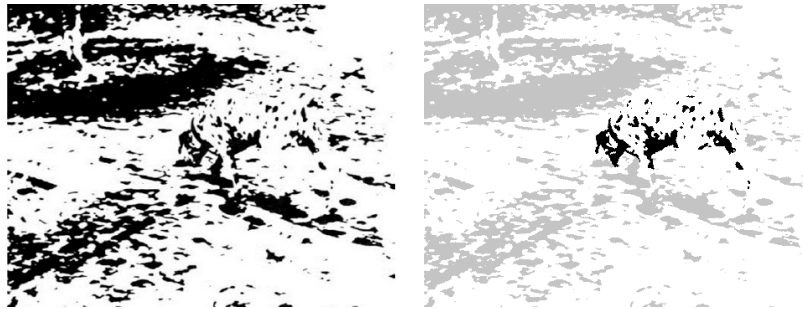
Xu and Kaplan [25] proposed a framework that attempted to depict continuous-tone images using only black and white colors. Unlike the methods mentioned in previous, they based on image segmentation and assigned each segment with a color. By optimizing the energy function, one can get an artistic black-and-white illustration, and the results can be distinct with different coefficients. However, the relation between the resulting illustrations and the coefficients in the energy function are not intuitive. Hence it was difficult to figure out the suitable coefficients for the desired results.

Our work is similar to the previous two projects. All our goals are abstracting arbitrary input images into black and white images also to preserve details while, as much as possible, producing a large region of a solid color in the output. For fitting the conflict requirements, previous approaches try to find an adaptive threshold from global or local regions to form the final result. However, we provide an alternative approach that focuses on the content instead of on its threshold. The algorithm will calculate their tangent flow first to enhance the detail.

Japanese manga is another field of two-tone illustrations, and there is some literature about it, such as Yingge Qu et al. [26], Herr, C. M., et al. [27]. Japanese manga creates lots of fantastic illustrations. However, it usually uses plenty of texture patterns to enrich the visual perception. These techniques may obscure the key point in the image which makes it unsuitable for image abstraction.

Emergence refers to the unique mental skill of humans to aggregate information from seemingly meaningless pieces. An example is shown in Figure 3. Niloy J. Mitra, et al. [28] proposed a framework to generate binary illustrations by spreading numerous black spots with varied sizes on the white background. The resulting black-and-white images seem disordered and disorganized at first glance, but an important object pops out after a few seconds. However, Niloy J. Mitra et al. [28] do not aim at proposing an abstraction method, but rather at challenging our attention so that we need to make more efforts to figure out the main content of the images, which contradicts the main idea of image abstraction.

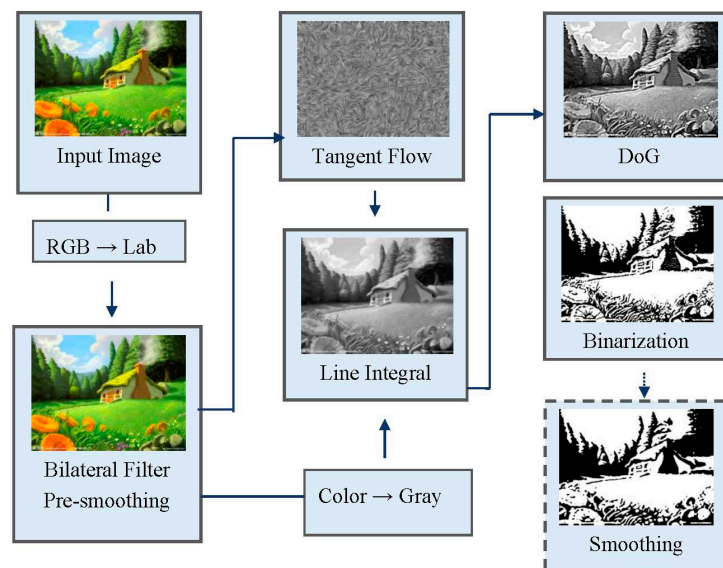




**Figure 3.** A classic example of an emergent image. One can perceive a primary object in the right half image as a Dalmatian dog pops out suddenly.

### 3. Method

In this section, we will derive our work of making black-and-white abstract illustrations from continuous-tone images. The overview of our framework is shown in Figure 4. Our goal is to abstract images by simplifying their visual contents while preserving the primary structure of the main regions of interest. This is the main idea of image abstraction. Then we use just two colors to represent the abstract images as black-and-white illustrations. Shape plays an important role in convey information for a picture. Thus, edge extraction and region smoothing become very critical for image abstraction. Kang et al. [15,29] developed a vector field called the Edge Tangent Flow as guidance for shape/color filtering. Our approach is similar to their method. Both of our methods are based on a vector field to enhance the important features. However, their abstraction is in a parallel manner which line drawing and region smoothing are two independent operators. We are doing that in a sequence manner which smooths first then enhance the edge areas. By this process, the final presentation form between theirs and ours also has a little difference. Their image retains color information. Ours is reduced to a binary image. The details are discussed in the following sections.



**Figure 4.** Framework overview. Each step of our framework, along with the performed function and the stage result, is shown in the figure. The dotted line implies an optional step.

#### 3.1. Pre-Smoothing

For abstraction, smoothing is a critical operator. Based on the smoothing, the level of detail will reduce. The similar intensity can merge into the same value. However, if without consider the

boundary of shape, it also will be blurred. Therefore, in the pre-smoothing stage, we just want to remove the effect of small noise. Thus, we first use the bilateral filter, which is an edge-preserving and noise reducing smoothing filter, to smooth the input image. We want to reduce the possibility of unpleasant patterns caused by noise and want to eliminate less important information in the images. Tomasi and Manduchi [30] suggested that it is better to compute the bilateral filter on a perceptually uniform feature space, such as CIELab, so that image contrast is adjusted depending on just noticeable differences.

### 3.2. Image Tangent Flow

We estimate the local orientation of images based on the eigenvalues of structure tensor, which is a matrix derived from the gradient of a function. We regard a continuous-tone image (supposed with  $m$  channels) as a function  $f: \mathbf{R}^2 \rightarrow \mathbf{R}^m$ , which maps a point  $p = (x, y)$  in the image plane to an  $m$ -dimensional color space. Here we use the Lab color space; this idea can be seen in Di Zenzo [31]. Thus, the images can be denoted as  $f(x, y) = (L(x, y), a(x, y), b(x, y))$ . For convenience, we rewrite  $f(x, y), L(x, y), a(x, y), b(x, y)$  as  $f, L, a, b$  respectively. We calculate the convolutions of the image with the Sobel filter.

$$\text{Sobel}_x = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{Sobel}_y = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

To approximate the partial derivative along x-direction and y-direction.

$$\begin{aligned} \frac{\partial f}{\partial x} = f_x &= \begin{bmatrix} \frac{\partial L}{\partial x} & \frac{\partial a}{\partial x} & \frac{\partial b}{\partial x} \end{bmatrix}^T = \begin{bmatrix} L_x & a_x & b_x \end{bmatrix}^T \\ \frac{\partial f}{\partial y} = f_y &= \begin{bmatrix} \frac{\partial L}{\partial y} & \frac{\partial a}{\partial y} & \frac{\partial b}{\partial y} \end{bmatrix}^T = \begin{bmatrix} L_y & a_y & b_y \end{bmatrix}^T \end{aligned} \quad (2)$$

The Jacobian matrix is

$$J = \begin{bmatrix} f_x & f_y \end{bmatrix} = \begin{bmatrix} L_x & L_y \\ a_x & a_y \\ b_x & b_y \end{bmatrix} \quad (3)$$

and the structure tensor of  $f$  is

$$S_{2 \times 2} = J^T J = \begin{bmatrix} L_x & a_x & b_x \\ L_y & a_y & b_y \end{bmatrix} \begin{bmatrix} L_x & L_y \\ a_x & a_y \\ b_x & b_y \end{bmatrix} = \begin{bmatrix} f_x^T f_x & f_x^T f_y \\ f_y^T f_x & f_y^T f_y \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \quad (4)$$

Now, we can further define a squared local contrast of  $f$  at a point  $p = (x, y)$  for a given vector  $n = (n_x, n_y)$  as

$$S(p, n) = E \cdot n_x^2 + 2F \cdot n_x n_y + G \cdot n_y^2 \quad (5)$$

It is well known that such a quadratic form has a maximum and minimum value as  $n$  varies, and the extremal values correspond to the eigenvalues of  $S_{2 \times 2}$ . By solving  $\det(S_{2 \times 2} - \lambda I) = 0$ , one can find the eigenvalues to be

$$\lambda_{1,2} = \frac{E + G \pm \sqrt{(E - G)^2 + 4F^2}}{2} \quad (6)$$

Without losing of generality, we assume that  $\lambda_1 \geq \lambda_2$ . By solving  $S_{2 \times 2} v = \lambda v$ , one can find the corresponding eigenvectors are

$$v_1 = (F, \lambda_1 - E) \quad v_2 = (\lambda_2 - G, F) \quad (7)$$

which are two orthogonal vectors. The two eigenvectors correspond to the directions with maximal and minimal local contrast respectively. Thus, we consider  $v_1$  as the gradient direction and  $v_2$  as the tangent direction, which is perpendicular to the gradient direction, of points in the image. More details can be found in Di Zenzo [31] and Cumani [32]. We can construct a vector field which represents the local orientation of an image by computing the eigenvector  $v_2$  of each point of the image. Such a vector field, which is induced by the tangents, is called a tangent field. Unfortunately, the vector field, which is constructed by assigning the eigenvector  $v_2$  to each point in the image, has discontinuities and may degrade the quality of the results. To fix the discontinuities, we can smooth the vector field by applying a box filter or a Gaussian filter on it.

### 3.3. Line Integral Convolution

We further simplify the boundary of objects and smooth the image by line integral convolution [33]. The convolution process is guided by the vector field computed in Section 3.2. Before calculating line integral convolution, we first convert the pre-smoothed image into grayscale image and normalize each vector in the vector field such that all vectors are unit vectors, i.e.,  $\|v\| = \|(n_x, n_y)\| = 1$ . The local behavior of a point  $p = (x, y)$  in the vector field can be approximated by extracting a streamline that starts at the center of the point and traveling in both positive and negative directions. We use the following equations to extract a streamline:

$$\begin{aligned} P_{start} &= P_0 = (x + 0.5, y + 0.5) \\ P_i &= P_{i-1} + V_{\pm}(\lfloor P_{i-1} \rfloor) \cdot ds, \forall i \in \mathbf{N} \end{aligned} \quad (8)$$

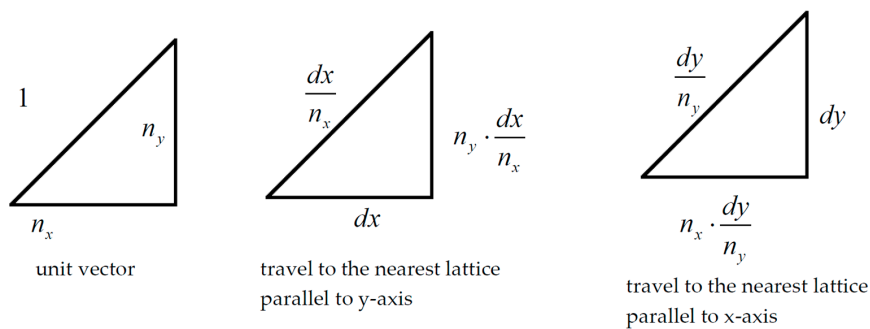
where  $V_{\pm}(\lfloor P_{i-1} \rfloor)$  are the vector in the vector field at the lattice point  $\lfloor P_{i-1} \rfloor = (\lfloor P_x \rfloor, \lfloor P_y \rfloor)$  and  $\pm$  represents traveling in positive and negative directions respectively. Where  $\lfloor P \rfloor, \lceil P \rceil$  are the floor function and the ceiling function, respectively. In each traversal step, we reach the nearest lattice axis, either parallel to x-axis or y-axis, so that the term

$$ds = \min\left(\left|\frac{dx}{n_x}\right|, \left|\frac{dy}{n_y}\right|\right) \quad (9)$$

measures the smallest distance between the current point and the closest lattice axis.

$$\text{Let } V_{\pm}(\lfloor P_{i-1} \rfloor) = (n_x, n_y)$$

$$dx = \begin{cases} \lceil P_x \rceil - P_x & \text{if } n_x > 0 \\ P_x - \lfloor P_x \rfloor & \text{if } n_x < 0 \\ \infty & \text{if } n_x = 0 \end{cases} \quad dy = \begin{cases} \lceil P_y \rceil - P_y & \text{if } n_y > 0 \\ P_y - \lfloor P_y \rfloor & \text{if } n_y < 0 \\ \infty & \text{if } n_y = 0 \end{cases} \quad (10)$$

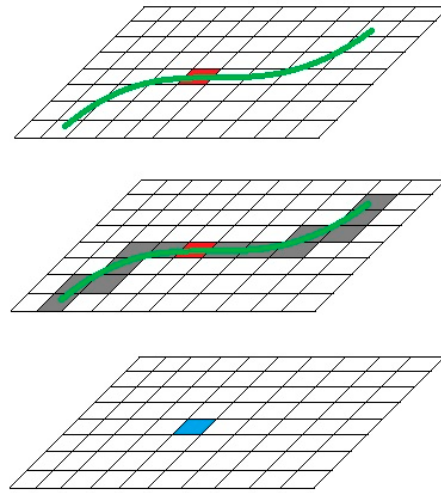


After extracting a streamline, we apply a Gaussian filter to re-compute a new weighted average of the pixel value at the point  $p = (x, y)$ .

$$f(p) = \frac{\sum_{i=1}^n \omega_i \cdot f(p_i)}{\sum_{i=1}^n \omega_i} \quad (11)$$

where  $\omega_i$  is the coefficient (weight) of the Gaussian filter and the set  $\{p_i\}$  stores all pixels that the extracted streamline passed through.  $f(p)$  is the pixel value of a point  $p = (x, y)$ . For convenience to calculate, the coefficient will look upon a prepared matrix. The size of the matrix is  $10 \times 10$  and produced by sampling the Gaussian filter kernel (with  $\sigma = 1$ ) at the midpoint of each pixel.

Figure 5 shows the concept of line integral convolution. Since we smooth the image along the tangent flow, we can reduce the noise while preserving the primary structure of the image. We only need to decide the traversal step, which determines how far we traverse along the tangent flow when we compute a streamline. Compared to other content-aware smooth filters, our parameter is more intuitive and hence simpler to adjust.



**Figure 5.** Line integral convolution overview. We extract a streamline (green) starting from a pixel (red) in both positive and negative directions, and compute a weighted average (blue) of pixels that streamline passed through (gray).

### 3.4. Difference-of-Gaussians Enhancement

The Gaussian filter is a well-known low-pass filter, which allows low-frequency signals to pass through and reduces the amplitude of the high-frequency signals. One can get a band-pass filter, which passes signals within a certain frequency range and attenuates frequencies outside the range, by subtracting two Gaussian filters with different kernels. Thus, a difference-of-Gaussians (DoG) filter can extract image features whose frequencies are within a particular frequency band. Those features tend to correspond to the edges of the image. It is believed that DoG filter gives a good feature line extraction and has stylistic versatility. There are many artistic works use the DoG filter in their frameworks, such as human face illustrations [14], coherent line drawing [15], real-time video abstraction [3], image abstraction by structure adaptive filtering [4], and style and abstraction in portrait sketching [34].

We first look at the DoG filter suggested by Winnemöller [3]:

$$D(\sigma, \tau, k, \mathbf{I}) = G(\sigma) \times \mathbf{I} - \tau \cdot G(k\sigma) \times \mathbf{I} \quad (12)$$

where  $G(\sigma)$  is the Gaussian filter with variance  $\sigma^2$  and  $\mathbf{I}$  is the image convolved with  $G(\sigma)$ . We can rewrite the formula of the DoG kernel above in more intuitive way:



$$\begin{aligned}
D(\sigma, \tau, k) &= G(\sigma) - \tau \cdot G(k\sigma) \\
&= G(\sigma) + \tau \cdot G(\sigma) - \tau \cdot G(k\sigma) - \tau \cdot G(\sigma) \\
&= (1 - \tau) \cdot G(\sigma) + \tau \cdot (G(\sigma) - G(k\sigma)) \\
&= (1 - \tau) \cdot G(\sigma) + \tau \cdot D(\sigma, k)
\end{aligned} \tag{13}$$

As you can see, the proposed DoG filter was a linear combination of the Gaussian-blurred image and the standard DoG. Thus, decreasing the  $\tau$  reduce the effect of the edge and become blurry. The parameter  $\tau$  determines the sensitivity of the feature detector. As  $\tau$  increases, more details are presented and thus the resulting image becomes darker. One can adjust the parameter to get the desired result. For most of the cases,  $\tau \in [0.95 \ 0.99]$ . Typically,  $k$  is set to be 1.6 to obtain an approximation of the Laplacian-of-Gaussian filter  $\nabla^2 G$  [13].  $k$  is defined as the ratio of the area under the inhibitory Gaussian to the area under the excitatory Gaussian. The problem with using a DoG to approximate  $\nabla^2 G$  is to find a space constant that keeps the bandwidth of the filter small and yet allows the filter adequate sensitivity: for, clearly, as the space constants approach one another, the contributions of the excitatory and inhibitory components become identical, and the sensitivity of the filter is reduced.

We use the line-integral-convolution-smoothing image as the base of our result, and use the standard DoG filter as our detail enhancement kernel:

$$\text{Enhance}(\sigma, k, \mathbf{I}_{\text{LIC}}) = \mathbf{I}_{\text{LIC}} + w_{\text{enhance}} \cdot D(\sigma, k) \times \mathbf{I}_{\text{LIC}} \tag{14}$$

By applying this enhancement kernel, we can get a smoothed but feature-preserved image, where the line-integral-convolution-smoothing image is the base of the result and a DoG filter enhances the level of detail by adjusting the weight  $w_{\text{enhance}}$ .

### 3.5. Binarization

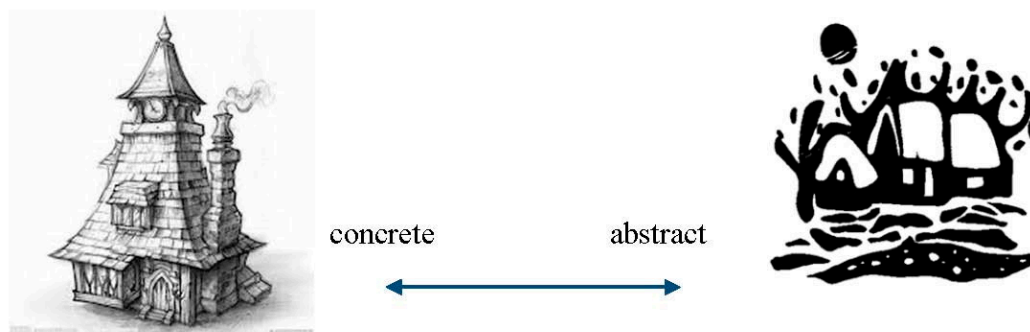
After getting an enhanced image as proposed in the previous section, the binarization step is straightforward and fast: just pick up a threshold. Although it is not easy to get the desired result for just choosing a global threshold, since such a suitable threshold may not even exist, our work works well with global thresholding method in most cases. Our smoothing method, which uses line integral convolution guided by tangent flow, eliminates the noise, and details are enhanced by a DoG filter, letting the important features stand out. Thus, our framework of splitting the image into the important and unimportant parts and just applying a global thresholding method can get a pleasant result.

### 3.6. Optional Smoothing Process

Since there are many kinds of art, some artists want to express feelings by drawing vivid paintings and some drawing images in the more abstract way (see Figure 6). Artists may emphasize different information; this concept can be seen in style and abstraction in portrait sketching [34], they forced their artists to sketch a face within a small amount of time, which makes the resulting sketches abstract. Our goal is to mimic this abstraction process, which reduces the level of detail but the primary object is still recognizable. We extract each connected black block in the binary image generated in the previous step, then apply a Gaussian filter with variance  $\sigma^2$  and smooth for  $n_{\text{smooth}}$  iterations. With bigger differences and running more iterations, we can get a more abstract black-and-white illustration.

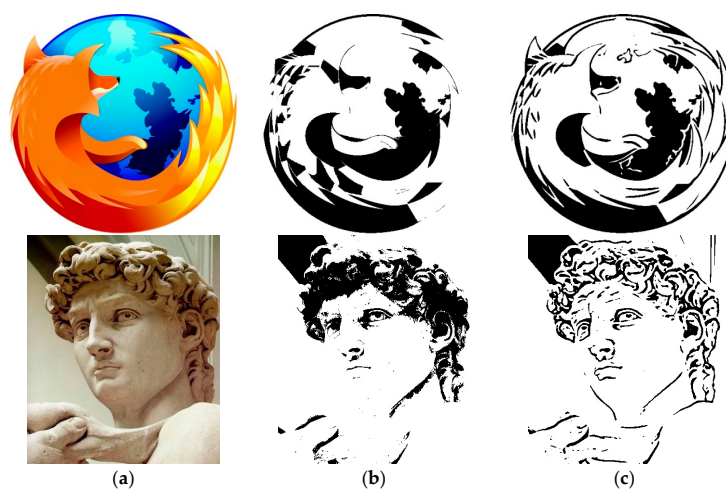
## 4. Results

The quality of art is not easy to compare with a quantitative analysis. It is not obvious to say that our result is better than others. However, it is an option. We provide a stylization of abstraction for an image that keeps the photo as simple as possible but still reserves its detail.

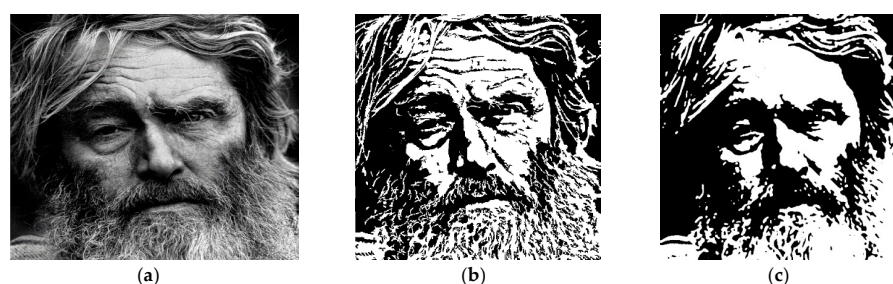


**Figure 6.** Different types of art.

In our algorithm, the key steps are the line integral convolution and the Difference-of-Gaussians enhancement. In the previous step focus on the smoothing. Let the similar color become a vast region. Increasing the size of the filter will provide a more blurred image and lose the detail. However, the tangent field will help to keep the boundary of different color regions clear. The next step differentiates between the bright side and darker side. Increasing the weight of DoG will improve the contrast of them. The user is allowed to adjust these parameters to get a result that they desire. In the following, we compare our results with other previous works. Figure 7 demonstrates that it is easier to find a suitable threshold after the context is enhanced. Figures 8 and 9 show the results among different stylization approaches, compared with the works of Mould et al. and Xu et al. respectively. Figure 10 shows the interim results for the various input images. Finally, we show some results further utilizing the optional smoothing process in Figure 11.



**Figure 7.** Compared with thresholding method. (a) input (b) thresholding (c) our result.



**Figure 8.** Compared with Stylized Black and White Images from Photographs [24]. (a) input (b) Mould and Grant [24] (c) our result.

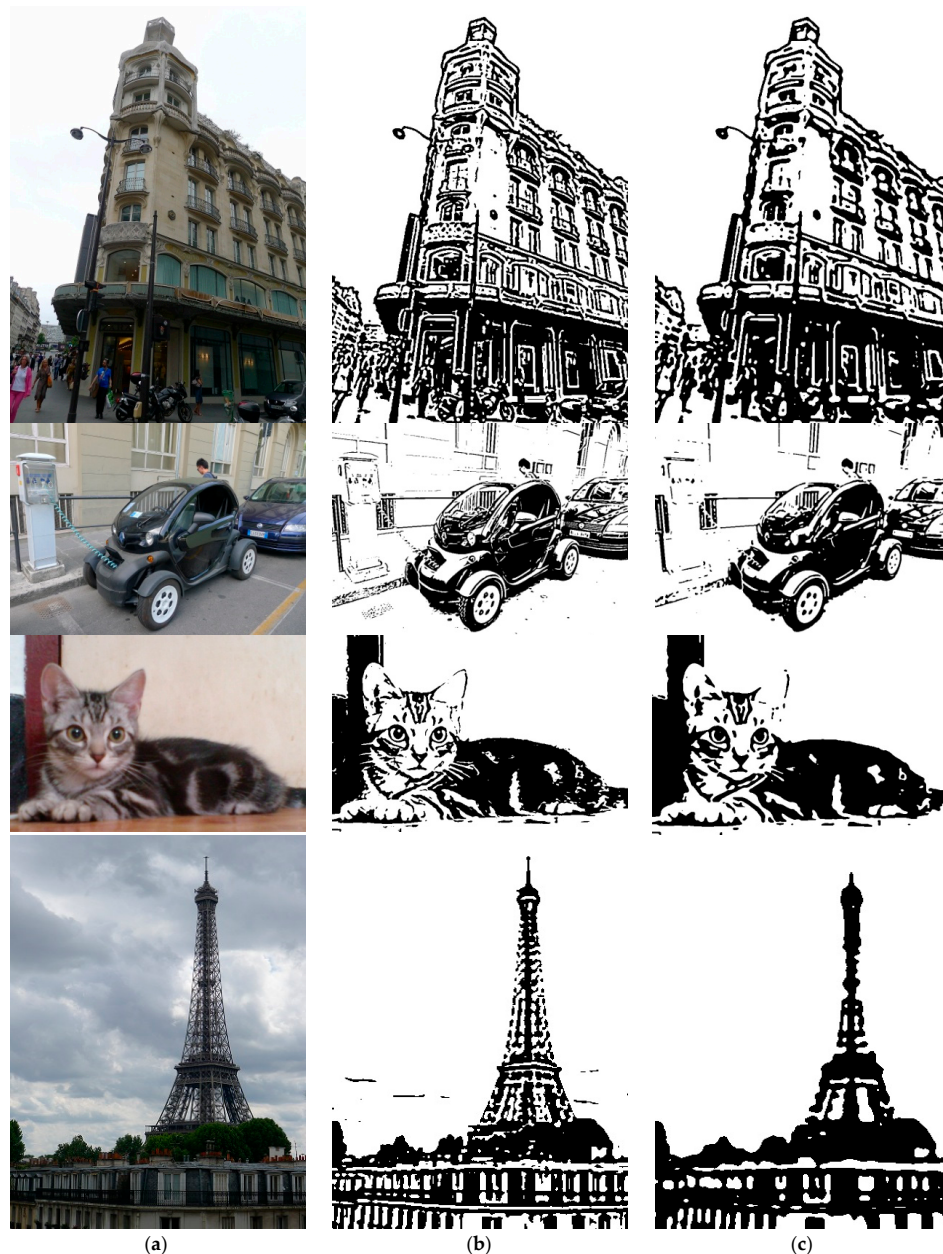


**Figure 9.** Compared with Artistic Thresholding [25]. (a) input (b) Xu and Kaplan [25] (c) our result.



**Figure 10.** Some other results applying our framework. (a) input (b) line integral convolution smoothing (c) DoG enhancement (d) binarization.





**Figure 11.** Some results of applying optional smoothing process. (a) input (b) binarization (c) optional smoothing result.

## 5. Conclusions

We present a practical and adjustable binary image abstraction framework that creates a simplified version of the input image and turns it into a black-and-white illustration. Our parameters are intuitive, and we can see the modification after adjusting the settings immediately. Hence, it is easy to find out the proper parameters for the desired result. The computational time of our framework is short, and the user can get the results in minutes, depending on how long the user spends on adjusting the parameters. The optional smoothing process makes the results more abstract and smoother. It may help the viewer to capture the key point in the image since more details are eliminated. The storage size of the results are extremely decreased, only one bit per pixel, but the primary structure of the images would still be recognizable since our abstract images reduce the information while preserving the main content of the input images. Although there are a lot of different approaches discussing on the image

abstraction, we believe that the result of our approach still has its unique quality. It preserves the edge of principle shape and merges the similar color into the same area. Even in the binary representation, many details still can be discriminated.

There are two parts that we can further discuss. First, we only apply the simplest binarization method to make the processed image into a binary illustration. As far as we know, the results of adaptive thresholding methods are better than those of global thresholding methods in most cases. Although our framework works fine, it is considered that developing a suitable adaptive thresholding method may provide a more pleasant result. Second, we use Gaussian filters as the kernels of optional smoothing step. As we mentioned in Section 2, Gaussian filters are isotropic. It may make sense to use anisotropic filters or guide the smoothing step by a vector field to smooth the result in particular directions. Another smoothing method that we take into consideration is curvature-based smoothing. Nieuwenhuis [35] proposed an energy term on curvatures and provided some results of their work. It is worth figuring out a curvature-based smoothing method to further stylize our binary illustrations.

**Acknowledgments:** This work was supported in part by the National Science Council under the grants of MOST 105-2221-E-126-011.

**Author Contributions:** All of the authors contributed extensively to this work presented in the paper. Yi-Ta Wu has coordinated the work and participated in all section. Yung-Yu Chuang is as an advisor guiding the designing of image processing methods. Jeng-Sheng Yeh and FuChe Wu provide suggestions to improve the algorithm and reorganize and revise the whole paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tufte, E.R. Envisioning information. *Optom. Vis. Sci.* **1991**, *68*, 322–324. [CrossRef]
2. Doug, D.; Santella, A. Stylization and Abstraction of Photographs. *ACM Trans. Graph. (TOG)* **2002**, *21*, 769–776.
3. Winnemöller, H.; Olsen, S.C.; Gooch, B. Real-Time Video Abstraction. *ACM Trans. Graph. (TOG)* **2006**, *25*, 1221–1226. [CrossRef]
4. Kyprianidis, J.E.; Jürgen, D. Image Abstraction by Structure Adaptive Filtering. Available online: [https://hpi.de/fileadmin/user\\_upload/fachgebiete/doellner/publications/2008/KD08b/jkyprian-tpcg2008.pdf](https://hpi.de/fileadmin/user_upload/fachgebiete/doellner/publications/2008/KD08b/jkyprian-tpcg2008.pdf) (accessed on 17 April 2017).
5. Chan, F.H.Y.; Lam, F.K.; Zhu, H. Adaptive Thresholding by Variational Method. *IEEE Trans. Image Process.* **1998**, *7*, 468–473. [CrossRef] [PubMed]
6. Sauvola, J.; Pietikäinen, M. Adaptive Document Image Binarization. *Pattern Recognit.* **2000**, *33*, 225–236. [CrossRef]
7. Bradley, D.; Gerhard, R. Adaptive Thresholding Using the Integral Image. *J. Graph. GPU Game Tools* **2007**, *12*, 13–21. [CrossRef]
8. Pai, Y.T.; Chang, Y.F.; Ruan, S.J. Adaptive Thresholding Algorithm: Efficient Computation Technique Based on Intelligent Block Detection for Degraded Document Images. *Pattern Recognit.* **2010**, *43*, 3177–3187. [CrossRef]
9. Singh, T.R.; Roy, S.; Singh, O.I.; Sinam, T.; Singh, K.M. A New Local Adaptive Thresholding Technique in Binarization. *IJCSI Intern. J. Comput. Sci. Issues* **2011**, *8*, 271–277.
10. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698.
11. Salisbury, M.P.; Anderson, S.E.; Barzel, R.; Salesin, D.H. Interactive Pen-And-Ink Illustration. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, FL, USA, 24–29 July 1994; pp. 101–108.
12. Meer, P.; Georgescu, B. Edge Detection with Embedded Confidence. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1351–1365. [CrossRef]
13. Marr, D.; Ellen, H. Theory of Edge Detection. *Proc. R. Soc. Lond. Ser. B* **1980**, *207*, 187–217. [CrossRef]
14. Gooch, B.; Reinhard, E.; Gooch, A. Human Facial Illustrations: Creation and Psychophysical Evaluation. *ACM Trans. Graph. (TOG)* **2004**, *23*, 27–44. [CrossRef]



15. Kang, H.; Lee, S.; Chui, C.K. Coherent Line Drawing. In Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering, New York, NY, USA, 4–5 August 2007; pp. 43–50.
16. Wang, S.; Wu, E.; Liu, Y.; Liu, X.; Chen, Y. Abstract Line Drawings from Photographs Using Flow-Based Filters. *Comput. Graph.* **2012**, *36*, 224–231. [CrossRef]
17. Floyd, R.W.; Steinberg, L. An Adaptive Algorithm for Spatial Grey Scale. *Proc. Soc. Inf. Disp.* **1976**, *17*, 75–77.
18. Jarvis, J.F.; Judice, C.N.; Ninke, W.H. A Survey of Techniques for the Display of Continuous Tone Pictures on Bi-level Displays. *Comput. Graph. Image Process.* **1976**, *5*, 13–40. [CrossRef]
19. Stucki, P. MECCA—A multiple error correcting computation algorithm for bi-level image hard copy reproduction. Research report RZ1060. IBM Research Laboratory: Zurich, Switzerland, 1981.
20. Ostromoukhov, V. A Simple and Efficient Error-Diffusion Algorithm. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001.
21. Pang, W.M.; Qu, Y.; Wong, T.T.; Cohen-Or, D.; Heng, P.A. Structure-Aware Halftoning. *ACM Trans. Graph. (TOG)* **2008**, *27*, 89. [CrossRef]
22. Chang, J.; Benoît, A.; Victor, O. Structure-Aware Error Diffusion. *ACM Trans. Graph. (TOG)* **2009**, *28*, 162. [CrossRef]
23. Li, H.; Mould, D. Contrast Aware Halftoning. *Comput. Graph. Forum* **2010**, *29*, 273–280. [CrossRef]
24. Mould, D.; Grant, K. Stylized Black and White Images from Photographs. In *NPAR '08 Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 9–18 June 2008*; ACM: New York, NY, USA; pp. 49–58.
25. Xu, J.; Kaplan, C.S. Artistic Thresholding. In Proceedings of the *NPAR '08 6th International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 9–11 June 2008*.
26. Qu, Y.; Pang, W.M.; Wong, T.T.; Heng, P.A. Richness-Preserving Manga Screening. *ACM Trans. Graph. (TOG)* **2008**, *27*, 155. [CrossRef]
27. Herr, C. M.; Gu, N.; Roudavski, S.; Schenabel, M.A. Digital Manga Depiction. 2011. Available online: [http://papers.cumincad.org/data/works/att/caadria2011\\_070.content.pdf](http://papers.cumincad.org/data/works/att/caadria2011_070.content.pdf) (accessed on 17 April 2017).
28. Mitra, N.J.; Chu, H.K.; Lee, T.Y.; Lior, W.; Hezy, Y.; Cohen-Or, D. Emerging Images. *ACM Trans. Graph. (TOG)* **2009**, *28*, 163. [CrossRef]
29. Kang, H.; Lee, S.; Chui, C.K. Flow based Image Abstraction. *IEEE Tran. Vis. Comp. Graph.* **2009**, *15*, 62–78. [CrossRef] [PubMed]
30. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the 1998 6th IEEE International Conference on Computer Vision, Bombay, India, 7 January 1998.
31. Silvano, D.Z. A Note on the Gradient of a Multi-Image. *Comput. Vis. Graph. Image Process.* **1986**, *33*, 116–125.
32. Cumani, A. Edge Detection in Multispectral Images. *CVGIP Graph. Models Image Process.* **1991**, *53*, 40–51. [CrossRef]
33. Cabral, B.; Leith, C.L. Imaging Vector Fields Using Line Integral Convolution. In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 2–6 August 1993.
34. Berger, I.; Shamir, A.; Mahler, M.; Carter, E.; Hodgins, J. Style and Abstraction in Portrait Sketching. *ACM Trans. Graph. (TOG)* **2013**, *32*, 55. [CrossRef]
35. Nieuwenhuis, C.; Toeppe, E.; Lena, G.; Veksler, O.; Boykov, Y. Efficient Squared Curvature. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Available online: [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2014/papers/Nieuwenhuis\\_Efficient\\_Squared\\_Curvature\\_2014\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Nieuwenhuis_Efficient_Squared_Curvature_2014_CVPR_paper.pdf) (accessed on 17 April 2017).

