IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 23, 2019, accepted December 16, 2019, date of publication December 19, 2019, date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960877

A Novel Pencil Drawing Algorithm Based on Non-Symmetry and Anti-Packing Pattern Representation Model

YUNPING ZHENG[1], YUMING CHEN[1], AND MUDAR SAREM[2,3]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]School of Software Engineering, Huazhong University of Science and Technology, Wuhan 510006, China
[3]General Organization of Remote Sensing, Damascus, Syria

Corresponding author: Yunping Zheng (zhengyp@scut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant No. 61300134, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120172120036, the Natural Science Foundation of Guangdong Province of China under Grant No. S2011040005815, No. S2013010012515, No. 2015A030313206, and No. 2017A030313349, and the Fundamental Research Funds for the Central Universities of China under Grant No. 2015ZM133.

ABSTRACT The artistic stylization transformation of images is an important part in the research of Non-Photorealistic Rendering (NPR). This kind of non-realistic stylized research has a very strong realistic romantic temperament and it can express the artistic pursuit of researchers very well. It is a good example of practical application combining computer science and art. In the technology of image style conversion, the simulation of the realistic art style includes mainly cartoon style, oil painting style, ink painting style and pencil drawing style. Previously, some researchers proposed a style conversion method based on the prior knowledge that there might be crosses at the junction of two lines where the tone distribution has a certain rule. However, in this paper, we have focused on the use of an edge extraction algorithm based on Non-symmetry and Anti-packing pattern representation Model (NAM) to enhance the pencil strokes and highlight the details of the image. Then, by combining the line pencil strokes with the tone drawing, we can transform the image into an image with pencil sketch style. Finally, compared with the most important previous methods, the experimental results presented in this paper showed that our proposed method showed more clearly the scene and the content and more details of the image than the previous methods. We also verified our method and calculated Structural Similarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR) of original images and color pencil drawings with different algorithms on the 7 nature images and the BSDS500 data set.

INDEX TERMS Image pencil drawing, non-photorealistic rendering (NPR), non-symmetry and anti-packing pattern representation model (NAM), peak signal-to-noise ratio (PSNR), structural similarity (SSIM).

I. INTRODUCTION

Pencil is one of the most accessible and easy-to-use painting tools. As one of the most fundamental drawing languages to abstract the understanding of the nature scene [1], pencil drawing has a unique attraction. With the increasing popularity of the image capture devices and the Internet, the artistic style of converting nature images into pencil drawings has become increasingly popular. At the same time, the Non-Photorealistic Rendering (NPR) [1]–[5] of images has increased in line with the public aesthetics, since it pays attention to the graphic art and the personalized expression. In the text illustration, film production, and advertisement design, the applications of pencil drawing are often noticed, which can accurately depict the content of the image, while express a concise and accurate artistic effect. Drawing pencil drawings by hand requires certain professional skills, and it takes a lot of time and effort. Therefore, it is a feasible method to draw a pencil drawing by using computer, and it is also an important form of non-photorealistic rendering of images.

Since the non-photorealistic rendering technology has started, many pencil drawing algorithms have been proposed.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

These methods can be divided into 3D model-based rendering [6], [7] and 2D image-based rendering [8]–[15]. The drawing algorithms based on 3D models required a complex 3D model. However, with the development of digital camera technology and the popularity of network sharing, obtaining high-quality 2D images was much easier than building 3D models based on a nature scene. Therefore, the demand for rendering pencil drawings from nature images has greatly increased. This paper studies the algorithms of generating pencil drawings based on 2D images.

Gatys *et al.* [10] and Selim *et al.* [11] introduced an image style conversion technique based on a convolutional neural network that can specify a painting style and then dynamically transformed images into the needed style. In the recent years, a lot of face sketch synthesis algorithms [12]–[15] have been developed due to their widespread utilities in digital entertainment. Peng *et al.* [13] proposed a novel multiple representations-based face sketch-photo synthesis algorithm. This algorithm could adaptively combine multiple representations to represent an image patch. Li *et al.* [14] proposed a novel adaptive representation-based face sketch synthesis algorithm, where different regions were represented with different features. Their algorithm combined multiple features generated from the face images, utilized several filters, and deployed Markov networks to exploit the interacting relationships between the adjacent image patches.

In the real art work, the artists usually use hand-drawn strokes to depict the global shape and the main outline, and then repeatedly render the pencil strokes in different areas to depict the tone and the shadow. That is, the pencil drawing is composed of a line drawing with strokes representing the shape, and a tonal texture representing the shadow and the texture. Based on this fact, the 2D image-based pencil rendering algorithm consists of two main steps: generating the pencil strokes and drawing the pencil tonal textures.

In the literature [15], Kim *et al.* proposed a novel sketch filter based on the maximum filter to extract the edges feature and the textures which have hand-drawn style. And then, they applied the sketch filter for edge feature extraction, color pencil drawing, and animation filtered image. The sketch filter could implicitly calculate the complexity of the local area, and produce an image that displays the edges feature and the textures from the nature image, just as same as an artist draws a pencil drawing with a pencil. Way *et al.* [16] implemented a simulation algorithm based on the YUV color mode and completed the drawing of the color pencil drawing by mapping the YUV color back to the RGB color mode.

Lu *et al.* [17] proposed an approach of generating pencil drawings that combined pencil tone drawing and line drawing with pencil strokes. The pencil tone drawing depicted the tone, the shapes, the shadow, and the shading, while the line drawing with pencil strokes focused more on the general structures of the scene. Lu *et al.* simulated the process of generating the tonal pencil texture in human drawing by using multiplication of strokes. On the other hand, the line drawing with pencil strokes was extracted based on a convolution framework.

Based on the previous method of Lu *et al.* [17], in this paper, we use the edge extraction algorithm based on Non-symmetry and Anti-packing pattern representation Model [18]–[22] to enhance the pencil strokes and highlight the details of the image. And then, by combining the enhanced line drawing with the tone drawing, we can obtain the final pencil drawing. The experimental results presented in this paper show the effectiveness of our algorithm over the previous state of the art methods. We also verified our method and calculated Structural Similarity (SSIM) [23] and Peak Signal-to-Noise Ratio (PSNR) of original images and color pencil drawings with different algorithms on the 7 nature images. The Berkeley Segmentation Dataset 500 (BSDS500) [24] is an extension of the BSDS300, where the original 300 images are used for training and validation and 200 images are used for testing. We also calculate the average PSNR and SSIM of all 500 images from the BSDS500.

II. RELATED WORK

In the realistic literature, pencil drawings consist of the pencil strokes that depict the global shapes and the tone with the pencil textures. In the process of generating a pencil drawing from an existing 2D image-based rendering, many algorithms have simulated the pencil strokes and the tonal pencil textures to generate the corresponding line drawing and the tonal pencil texture respectively. Then, such algorithms combined the corresponding line drawing and the tonal pencil texture to generate the final pencil drawing.

A. LINE DRAWING WITH PENCIL STROKES

Through the observation of many pencil drawings, we can notice that the artist cannot always draw a very long and continuous curve, but he can put a set of less lengthy edges to border upon sketchy lines, which will produce a cross at the junction of two lines, as same as in the human drawing. In the previous years, many researches have been done to extract meaningful sketchy lines from a nature image. The lines obtained by using boundary detection algorithms were difficult to reflect the hand-drawing effects, or the lines were not smooth enough, or there were isolated less lengthy lines, and they were extremely sensitive to noise. For obtaining more accurate lines that describe the general structures of the scene, Kang *et al.* [25] discussed the use of the Edge Tangent Flow (ETF) and used the Flow-based Difference-of-Gaussians method to obtain continuous line drawings. Wang *et al.* [26] constructed a smooth and direction-enhanced edge flow field and used the flow field to guide the process of drawing sketchy lines. What's more, they incorporated an anisotropic nonlinear filter into their algorithm to extract the line edges, which may create an image for conveying a hand painting style. Kim *et al.* [15] proposed a novel sketch filter based on the maximum value of the filter to extract the edges features and the textures which have hand-drawn style.

Lu *et al.* [17] proposed an approach of generating line drawing with pencil strokes based on a convolution framework.

B. TONE DRAWING WIHT PENCIL TEXTURE

The second step of generating the pencil drawing is generating a tone drawing with pencil texture. In fact, many researches have been conducted to generate tonal pencil shadow and textures. Lee *et al.* [6] and Sousa and Buchanan [7] proposed approaches which generate the pencil texture with directional strokes for expressing the shading. Their approaches relied on the lighting and the material in the 3D models. In the approach of generating pencil texture based on 2D image, Li and Huang [27] simulated the direction of the strokes according to the directional feature from the local region of the input image, and used Line Integral Convolution (LIC) to generate a pencil texture. In the literature [28], Yamamoto *et al.* combined the image segmentations to divide the source image into intensity layers in different ranges and generate pencil-style shading lines based on the tone of the image. Yang *et al.* [29] proposed a Swing Bilateral LIC (SBL) approach to generate a controllable pencil texture by extracting the color from the input image and using them to describe the object in details.

The Line Integral Convolution (LIC) was a texture visualization technique based on the texture of the vector field [30]. It requested a 2D vector field and a white noise image as the input, and then convolved along a specific vector field. The white noise image was generally set to Gaussian white noise related to the hue of the image. For the acquisition of the vector field, Li and Huang [27] calculated the moment of the image to obtain the texture direction of the region. In the literature [29], the vector field consisted of different types of stroke streams.

The disadvantages of these LIC approaches were that the amount of calculation was large, and since the result was affected by the vector field and the noise image, the algorithms used in these LIC approaches had certain limitations in the expansion. For example, a vector field was distributed throughout the image, regardless of the tone of the image description. The entire image was convolved to generate a texture. However, the background area and the object area didn't use the same convolution approach to generate the texture when the artists draw a pencil drawing.

Unlike these LIC methods for generating pencil textures, Lu *et al.* [17] simulated the process where the tonal pencil texture was generated in human drawing using multiplication of strokes from a simple texture image that already existed.

C. DESCRIPTION OF THE NAM

Inspired by the concept of the packing problem, a novel Non-symmetry and Anti-packing pattern representation Model (NAM) was proposed in our previous work [18]–[22] in order to represent the pattern more effectively.

The idea of the NAM can be described as following: Giving a packed pattern and n predefined subpatterns with different shapes, pick up these subpatterns from the packed pattern and then represent the packed pattern with the combination of these subpatterns. The following is an abstract description of the NAM.

Suppose an original pattern is Γ, two reconstructed non-distortion and distortion patterns are Γ' and Γ'', respectively. Then, the NAM is either a non-distortion model from Γ to Γ' or a distortion one from Γ to Γ''. The procedure of the transform can be written as follows:

$$\Gamma' = \mathrm{T}(\Gamma), \quad \Gamma'' \approx \mathrm{T}(\Gamma),$$

where $T()$ is a transform or encoding function.

The procedure of the non-distortion encoding can be obtained by the following expression.

$$\Gamma' = \bigcup_{j=1}^{n} p_j(v, A \left| A = \{a_1, a_2, \cdots, a_{m_i}\}\right.) + \varepsilon(d),$$

where Γ' is the reconstructed pattern; $P = \{p_1, p_2, \ldots, p_n\}$ is a set of some predefined subpatterns; n is the type number of the subpatterns; p_j $(1 \leq j \leq n)$ is the j^{th} subpattern; v is the value of p_j; A is a parameter set of p_j; $a_i(1 \leq i \leq m_i)$ is a parameter set of shapes of p_j; m is the serial number of p_j; $\varepsilon(d)$ is a residue pattern, and d is a threshold of $\varepsilon(d)$.

If the residue pattern $\varepsilon(d)$ is removed from the non-distortion pattern, then the distortion pattern can be obtained as follows:

$$\Gamma'' = \bigcup_{j=1}^{n} p_j(v, A \left| A = \{a_1, a_2, \cdots, a_{m_i}\}\right.).$$

It is obvious that the following expression is true.

$$\Gamma \propto \Gamma' = \Gamma'' + \varepsilon(d).$$

Since the residual pattern $\varepsilon(d)$ is only used in the distortion pattern, $\varepsilon(d)$ is NULL for the non-distortion pattern. Therefore, the following expression can be obtained when the residual pattern is NULL:

$$\Gamma = \Gamma' = \Gamma''.$$

III. PENCIL DRAWING ALGORITHM BASED ON NAM

Lu *et al.* proposed an algorithm of generating pencil drawings that combined the pencil tone drawing and the line drawing with pencil strokes. The pencil tone drawing depicted the tones, the shapes, the shadows, and the shading, while the line drawing with pencil strokes focused more on the general structures of the scene. In fact, they simulated the process where the tonal pencil texture was generated by human drawing using multiplication of strokes. On the other hand, the line drawing with pencil strokes was extracted based on an unfilled convolution framework. Inspired by the method of Lu *et al.* [17], we propose a pencil drawing algorithm based on Non-symmetry and Anti-packing Pattern Representation Model (NAM) for image segmentation.

A. ALGORITHM DESCRIPTION

Line Drawing with Pencil Strokes: In the real pencil drawing, the first step is to generate pencil strokes that focus more on the general structures of the scene. Then, there is a need to compute the gradients on the grayscale version of the original image. The gradient map G is defined as follows:

$$G = \left((\partial_x I)^2 + \left(\partial_y I\right)^2\right)^{\frac{1}{2}}, \tag{1}$$

where I is the grayscale version of the original image and ∂_x and ∂_y are the gradient operators for I in the x and y directions respectively, implemented by the followed difference. However, the gradient map for a nature image can only give a rough representation of the outline of the scene for the nature image. It is typically noisy and doesn't contain continuous edges which are immediately ready for stroke generation.

Actually, in a real pencil drawing, a sketchy line consists of a set of less lengthy edges, rather than a very long continuous curve, which will produce a cross at the junction of two lines, as in human drawing. To achieve this effect, in literature [31], Ji *et al.* used the directional convolution operation for the gradient map to obtain the contribution value of each pixel for the defined direction in the gradient map. The direction at a certain point relies on the maximum contribution at the point. Therefore, eight directions at 45 degrees apart are defined. The response map for a certain direction is computed as the following:

$$G_i = D_i \otimes G, \tag{2}$$

$$C_i(x) = \begin{cases} G(x) & if \;\; \arg\max_i \{G_i(x)\} = i \\ 0 & other, \end{cases} \tag{3}$$

where D_i is considered as a convolution kernel, which expresses a line segment at the i^{th} direction, $\otimes$ expresses the convolution operator, G_i is the response map formed by the convolution of the gradient map G and the kernel D_i, which groups the gradient magnitudes along the direction i to form G_i, and C_i is the gradient magnitude map for the direction, which is performed by selecting the maximum value among the response map in all directions.

Also, generating lines at each pixel can be done by the convolution, which aggregates the nearby pixels along direction D_i. The convolution can link the edges of the pixels that are even not connected in the original gradient map, and thus we can achieve the effect of enhancing the edge and the stroke crossing. This can be expressed as in the following written formula:

$$S' = \sum_{i=1}^{8} (D_i \otimes C_i) \tag{4}$$

In the stroke map S' the edge areas have higher pixel value (i.e. the bright area). The final pencil stroke map S is generated by inverting the pixel values and mapping them to the range [0, 1].

Inspired by the method of Lu *et al.* [17], this paper proposes a pencil drawing algorithm based on Non-symmetry and Anti-packing Pattern Representation Model (NAM) [18]–[22] for image segmentation.

This paper uses the edge extraction algorithm based on NAM to enhance the pencil strokes and highlight the details. Let the enhanced pencil stroke combine with the tonal pencil texture to obtain a pencil drawing. We convert the input image to a binary image for a threshold. We select an adaptive threshold via maximum entropy threshold segmentation method for different images. In section 3.3, we will introduce the maximum entropy threshold segmentation method in detail. The binary image is divided by the NAM-based image segmentation algorithm into a set of sub-patterns (the homogeneous blocks). And then, each sub-pattern is traversed to scan its northern boundary and western boundary, and to update the overall boundary information of sub-pattern when there are adjacent sub-patterns that are homogeneous blocks. The edge map E of the original image is obtained after the traversal. The NAM-based edge extraction algorithm is expressed as follows:

$$E = NAM_E\,(I)\,, \tag{5}$$

where E is the edge map, which displays only the main outline features of the scene and the objects without a lot of detail, and I is the input image.

It will be very abrupt to directly superpose the pencil stroke S and the edge map E. Therefore, in this paper, the effects are not performed by directly combining the pencil stroke S and the edge map E by multiplying their values for each pixel. Instead, a random dilution matrix r with a range of [0.5, 0.6] is introduced to attenuate the intensity of the edge map E. Considering the phenomenon of color diffusion in the edge region of pencil drawing, we use Gaussian filter to deal with the attenuated edge map. The final pencil stroke map S_{NAM} is computed through the following operator:

$$E' = g \otimes (r * E)\,, \tag{6}$$

$$S_{NAM} = S * E', \tag{7}$$

where g is the Gaussian filter, and r is the random desorption matrix with a range of [0.5, 0.6].

Tone Drawing with Pencil Texture: After the pencil stroke is constructed, the following is a simulation of the tone drawing with pencil texture. By researching and counting a large number of pencil drawings, it has been found that the sketch tone histograms usually follow certain patterns [17]. The tone of pencil drawing mainly consisted of three layers: the bright layer, the dark layer, and the mild tone layer, which are certain by partitioning the pixels. Consequently, Lu *et al.* [17] proposed a parametric model to fit the tone distribution of the pencil drawing. They used different distribution functions to simulate the sketch tone histograms for the bright layer, the mild tone layer, and the dark layer. Among them, the bright layer, the mild tone layer, and the dark layer were modeled by using the Laplacian distribution, the uniform distribution and the Gaussian distribution respectively.

For the brighter layer, the Laplacian distribution is used and it is written as follows:

$$h_1(v) = \begin{cases} \frac{1}{\sigma_b} e^{-\frac{1-v}{\sigma_b}} & if\ v \le 1 \\ 0 & other, \end{cases} \tag{8}$$

where σ_b is the scale of the Laplacian distribution.

For the mild tone layer, the uniform distribution is used and it is written as follows:

$$h_2(v) = \begin{cases} \frac{1}{u_b - u_a} & if\ u_a \le v \le u_b \\ 0 & other, \end{cases} \tag{9}$$

where u_a and u_b are two controlling thresholds that represent the tone range of the distribution.

For the dark layer, the Gaussian distribution is used and it is written as follows:

$$h_3(v) = \frac{1}{\sqrt{2\pi\sigma_d}} e^{-\frac{(v-\mu_d)^2}{2\sigma_d^2}}, \tag{10}$$

where μ_d is the mean value of the dark strokes and σ_d is the scale of the Gaussian distribution.

The controlling parameters in equations (8), (9), and (10) will determine the shape of the tone histogram. We obtain the standard tone map with tone distribution of the pencil drawing by adjusting w_i of the three distributions. A parametric model is proposed to represent the target tone distribution, and it is expressed as follows:

$$h(v) = \frac{1}{N} \sum_{i=1}^{3} w_i * h_i(v), \tag{11}$$

where w_i are the weight coarsely that is corresponding to the number of the pixels in the corresponded tonal layer, and for each layer, the parameters w_i are estimated using the Maximum Likelihood Estimation (MLE), N is the normalization factor to make $\int_0^1 h(v)dv = 1$, where v is the tone value, and $h(v)$ represents the probability that a pixel value is equal to v in a pencil drawing.

Finally, based on the parametric components h_1, h_2, and h_3, we adjust the tone maps using the histogram matching in the three tone layers and superposed them again to obtain the final tone map J.

In the human drawing, the tonal pencil texture is generated by repeatedly drawing at the same place. In order to simulate the process using the dense pencil strokes, the reference [17] introduced an existing pencil texture map H, which was generated by hand drawing, and defined an logarithm combination $\beta(x) \ln H(x) \approx \ln J(x)$ that expressed rendering β times using H to approximate the local tone in the tone map $J \cdot \beta$ is computed by the following formula:

$$\beta^* = \arg\min \|\beta \ln H - \ln J\|_2^2 + \lambda \|\nabla\beta\|_2^2, \tag{12}$$

where λ is equal to 0.2 in our experiments, and β^* would be obtained by transforming β to a standard linear equation, which could be solved by using a conjugate gradient. The final pencil texture map T was generated by an exponential operator as follows:

$$T = H^{\beta^*}. \tag{13}$$

Finally, combining the line drawing with the pencil strokes S_{NAM} and the tone drawing with the pencil texture T to generate the final pencil drawing by multiplying S_{NAM} and T values for each pixel. It can be expressed as follows:

$$R_{NAM} = S_{NAM} \cdot T. \tag{14}$$

Obtain the color pencil drawing by taking the generated grayscale pencil sketch R_{NAM} as the Y channel in YUV color space, and re-mapping YUV back to the RGB color space.

B. ALGORITHM STEPS

The formal description of our algorithm is presented in the following steps:

Step 1: Use equation (1) to compute the gradients on the grayscale version of the input image I. And obtain the gradient map G.

Step 2: Define eight directions at 45 degrees apart as eight convolution kernels $\{D_i\}, i \in \{1, 2, \ldots, 8\}$. These eight convolution kernels are respectively convolved with the gradient map G along the direction to generate the response map $G_i, i \in \{1, 2, \ldots, 8\}$.

Step 3: Select the maximum value among the response map G_i in all directions by using Eq.(3), and obtain eight maps $\{C_i\}, i \in \{1, 2, \ldots, 8\}$.

Step 4: Given the eight maps $\{C_i\}, i \in \{1, 2, \ldots, 8\}$, the convolution operator between C_i and D_i is applied to link the edges of the pixels in the gradient map G. And then, generate the pencil strokes map S by inverting the pixel values and mapping them to the range [0, 1].

Step 5: Convert the input image to a binary image for a threshold. The edge map E will be obtained by the edge extraction algorithm NAM-based image segmentation, as given in Eq. (5).

Step 6: Introduce a random dilution matrix with a range of [0.5, 0.6] to attenuate the intensity of the edge map E, and define a Gaussian filter g which is used to deal with the edge map attenuated. The final pencil stroke map S_{NAM} is computed through the operators by using equations (6) and (7).

Step 7: By applying equations (8), (9), and (10), simulate the bright layer, the mild tone layer, and the dark layer, which make the components of the tone in the pencil drawing, by using the Laplacian distribution, the uniform distribution and the Gaussian distribution respectively. Based on the parametric components h_1, h_2 and h_3, Adjust the weights w_i in the three tone layers and superpose them again to obtain the standard histogram $h(v)$ in pencil drawing. The tone map J is obtained by adjusting the tone histogram of the input image using the histogram matching in the three tone layers according to the standard histogram $h(v)$.

Step 8: Introduce an existing pencil texture map H, which is generated by hand drawing, and define a logarithm combination $\beta(x) \ln H(x) \approx \ln J(x)$, that expresses to render β

times using H to approximate the local tone in the tone map J. The β^* is computed by equation (12) using a conjugate gradient.

Step 9: Obtain the pencil texture map T by using equation (13).

Step 10: Combine the line drawing with pencil strokes S and the tone drawing with pencil texture T by multiplying the S_{NAM} and the T for each pixel to generate the final grayscale pencil drawing R_{NAM}.

Step 11: Finally, obtain the color pencil drawing by taking the generated grayscale pencil sketch R_{NAM} as the Y channel in YUV color space, and re-mapping YUV back to the RGB color space.

C. ADAPTIVE THRESHOLD

It is very difficult to set the threshold of image binarization. How to select a reasonable threshold is a very important problem. Generally, the same default value is adopted for different images, but it will lead to good processing effect for one kind of image and bad processing effect for other images. Therefore, we select an adaptive threshold via maximum entropy threshold segmentation method for different images. The threshold value is defined as the pixel value that maximizes the entropy of the input image. In other words, after binarization of the image with this threshold, the entropy value of the whole image is the maximum. The so-called maximum entropy value means that the amount of information contained in the image is the maximum. This is called ''maximum entropy threshold segmentation'' method. The principle of this method is summarized as follows: another smooth image is obtained by averaging neighborhood of the original image, and a two-dimensional histogram is constructed from the original image and smooth image. Using the two-dimensional histogram obtained above to find the optimal threshold according to the maximum entropy principle.

In order to implement the maximum entropy segmentation method, the entropy model, used to calculate the entropy value of the input image, is realized by histogram, and is defined as:

$$\begin{aligned} H(x) &= E(I(x_i)) = E(\log_2(1/p(x_i))), \\ &= \sum p(x_i)(\log_2(1/p(x_i))), \end{aligned} \tag{15}$$

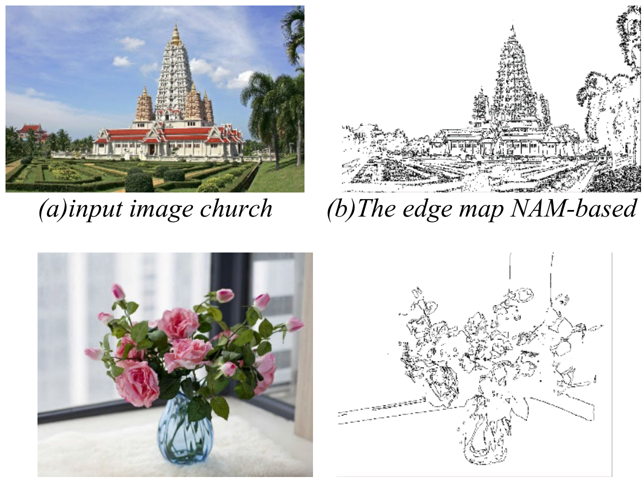
where I is the input image, $p(x_i)$ represents the ratio of the current pixel in the histogram of the input image.

$$\max \Big(\sum\nolimits_0^{T-1} p(x_i)(\log_2(1/p(x_i))) + \sum\nolimits_T^{255} p(x_i)(\log_2(1/p(x_i)))\Big), \tag{16}$$

In equation (16), T is in $[0, 1, 2, \ldots, 255]$. We select the T as threshold that maximizes the entropy of the input image. In Table 1, we show the thresholds of some images that are calculated by the maximum entropy segmentation method. Also, in Fig. 1, we show the edge map obtained by the edge extraction algorithm NAM-based image segmentation, as given in Eq. (5).

TABLE 1. The thresholds of some nature images calculated by the maximum entropy segmentation method.

Images	Thresholds
church	120
flower	117
flowers	139
salad	149
road	131
susu	126
woman	118



(a)input image church *(b)The edge map NAM-based*

(c)input image flowers *(d)The edge map NAM-based*

FIGURE 1. The edge map obtained by the edge extraction algorithm using NAM-based image segmentation.

D. PSNR AND SSIM

Peak Signal to Noise Ratio (PSNR) is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation, is an image quality evaluation metric. It is calculated as follows:

$$MSE(x, y) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (x(i, j) - y(i, j))^2,$$

$$PSNR(x, y) = 10 \log_{10}\Big(\frac{(2^n - 1)^2}{MSE}\Big),$$

where MSE represents the mean square error of the current image x and the reference image y, and H and W are respectively the height and width of the image; n is the number of bits per pixel, it generally is 8, that is, the gray level of the pixel is 256. The unit of PSNR is dB. The larger the value, the smaller the distortion.

For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences (now for each color, i.e. three as much differences as in a monochrome image) divided by image size and by three.

The structural similarity (SSIM) index is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE).

Given two images x and y, the structural similarity of the two images can be found as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ_x is the average of x, μ_y is the average of y. σ_x^2 is the variance of x, σ_y^2 is the variance of y. σ_{xy} is the covariance x and y. $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ is two variables to stabilize the division with weak denominator, L is the dynamic range of the pixel-values, $k_1 = 0.01$ and $k_2 = 0.03$ by default. The SSIM value range is [0, 1], the larger the value, the smaller the image distortion.

IV. EXPERIMENTAL RESULTS

Our experiments were performed by using Matlab 2016. And we have compared our results with the results presented by Lu *et al.* [17] and Kim *et al.* [15] respectively. In Fig. 2, 3, 4, and 5, we show the pencil strokes of some nature images with Lu *et al.*'s and our method. In Fig. 2, 3, 4, and 5, all the (a) are original images, all the (b) are the pencil stroke results of Lu *et al.*, and all the (c) are our results based on NAM. From these figures, it is obvious that our results not only retain the effects of the strokes cross, but also enhance the edges of the pencil drawing. As one can see, our method shows better the general structures of the scene and the objects than Lu *et al.* and Kim *et al.* methods.

(a)Input image church *(b) Lu et al. result* *(c) Our result*

FIGURE 2. Comparison of pencil stroke for the image church.

(a)Input image flowers *(b) Lu et al. result* *(c) Our result*

FIGURE 3. Comparison of pencil stroke for the image flowers.

(a)Input image flower *(b) Lu et al. result* *(c) Our result*

FIGURE 4. Comparison of pencil stroke for the image flower.

(a)Input image salad *(b) Lu et al. result* *(c) Our result*

FIGURE 5. Comparison of pencil stroke for the image salad.

(a) Input image flower *(b) Kim et al. result* *(c) Lu et al. result* *(d) Our result*

FIGURE 6. Comparison of color pencil drawing based on three algorithms for image flower.

(a) pencil stroke *(b) tonal pencil texture* *(c) pencil drawing*

FIGURE 7. The process of generating pencil drawing based on our algorithm for the image flower.

What's more, our algorithm improves the phenomenon in which some results of Lu *et al.*, have too many crosses in the edge regions, and it makes the effects of strokes cross softer.

In Fig. 6, 8, 10, 12, 14, 16, and 18, we present three groups for comparison of color pencil drawing with Kim *et al.* method [15], Lu *et al.* method [17], and our method. The results of Kim *et al.*, Lu *et al.* and ours are shown in (b), (c), and (d) respectively. It can be seen that our color pencil sketches have more vivid color and sharper outlines. Our results are also more layered. Fig. 7, 9, 11, 13, 15, 17, 19 shows the two steps of generating the pencil drawing (i.e., the pencil stroke and the tonal pencil texture), and the final pencil drawing which is generated by multiplying them based on our NAM-based pencil drawing algorithm.

(a) Input image salad (b) Kim et al. result (c) Lu et al. result (d) Our result

FIGURE 8. Comparison of color pencil drawing based on three algorithms for image salad.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 9. The process of generating pencil drawing based on our algorithm for image salad.

(a) Input image road (b) Kim et al. result (c) Lu et al. result (d) our result

FIGURE 10. Comparison of color pencil drawing based on three algorithms for image road.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 11. The process of generating pencil drawing based on our algorithm for image road.

In order to prove the effectiveness of our method, we show the experimental results of 7 nature images, as shown in the Fig. 6-19.

From an intuitive visual experience, we can see that our results, shown in Fig. 6(d), 8(d), 10(d), 12(d), 14(d), 16(d) and 18(d) have better artistic beauty subjective and they

(a) Input image church (b) Kim et al. result (c) Lu et al. result (d) our result

FIGURE 12. Comparison of color pencil drawing based on three algorithms for image church.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 13. The process of generating pencil drawing based on our algorithm for image church.

(a) Input image flowers (b) Kim et al. result (c) Lu et al. result (d) our result

FIGURE 14. Comparison of color pencil drawing based on three algorithms for image flowers.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 15. The process of generating pencil drawing based on our algorithm for image flowers.

are closer to the realistic color pencil drawing than the other methods. Also, compared with the results of Kim *et al.* which shown in Fig. 6(b), 8(b), 10(b), 12(b), 14(b), 16(b) and 18(b), our results are better either on the pencil stroke or on the tone drawing. It can be seen from the results of Kin et al. that the direct processing by using the maximum filter to generate the pencil drawing does not deal well with the bright-dark relationship in the pencil drawing. It can be noticed that in Fig. 8(b), in the salad region, the tonal distribution is very uneven, and there is a large gap between the tone and the original image, which does not intuitively show the effect of the pencil drawing. While by comparing with the result of Kim *et al.*, the strokes of our results are more obvious, and the scenes are clearer. Compared with the results of Lu *et al.* which shown in Fig. 6(c), 8(c), 10(c), 12(c), 14(c), 16(c) and 18(c), our color pencil sketches have more

(a) Input image woman (b) Kim et al. result (c) Lu et al. result (d) our result

FIGURE 16. Comparison of color pencil drawing based on three algorithms for image woman.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 17. The process of generating pencil drawing based on our algorithm for image woman.

(a) Input image susu (b) Kim et al. result (c) Lu et al. result (d) our result

FIGURE 18. Comparison of color pencil drawing based on three algorithms for image susu.

(a) pencil stroke (b) tonal pencil texture (c) pencil drawing

FIGURE 19. The process of generating pencil drawing based on our algorithm for image susu.

vivid color and sharper outlines. Our results are also more layered.

We also verified our method and calculated Structural Similarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR) of original image and color pencil drawings with different algorithms on the above 7 nature images and the BSDS500 data set. Table 2 and Table 3 show the SSIM and PSNR on the above 7 nature images and the BSDS500 data set, respectively. From the Table 2, we can see that the average values of the SSIM for Kim, Lu, and our proposed algorithms on the above 7 nature images are 0.49, 0.57, and 0.56, respectively. Also, the average values of the PSNR for Kim, Lu, and our proposed algorithms are 9.25, 12.93, and 13.21, respectively. Therefore, our algorithm has more visual effect than Kim's and Lu's algorithms while simultaneously retaining the original image structures.

TABLE 2. **The SSIM and PSNR on the 7 nature images of original image and color pencil drawings using different algorithms.**

	SSIM			PSNR		
Image	Kim	Lu	Our	Kim	Lu	Our
church	0.43	0.40	0.40	10.83	12.02	12.74
flower	0.37	0.58	0.58	6.48	9.12	9.41
flowers	0.66	0.73	0.72	10.14	14.87	15.10
salad	0.57	0.51	0.51	9.06	11.94	12.31
road	0.41	0.60	0.59	11.04	15.73	15.54
susu	0.56	0.55	0.53	8.53	14.18	14.46
woman	0.43	0.60	0.60	8.66	12.61	12.91
Average	0.49	0.57	0.56	9.25	12.93	13.21

TABLE 3. **The average SSIM and PSNR on BSDS500 data set.**

Method	SSIM	PSNR
Kim	**0.44**	7.31
Lu	0.43	8.64
Our	0.43	**9.36**

Also, it can be seen from the Table 3, we calculate the average PSNR and SSIM of all 500 images from BSDS500. We can see that the average values of the SSIM for Kim, Lu, and our proposed algorithms on BSDS500 data set are 0.44, 0.43, and 0.43, respectively. Also, the average values of the PSNR for Kim, Lu, and our proposed algorithms are 7.31, 8.64, and 9.36, respectively. Therefore, our algorithm has also more visual effect than Kim's and Lu's algorithms while simultaneously retaining the original image structures.

V. CONCLUSION

This paper mainly proposes an algorithm to generate pencil drawing based on NAM. We use the edge extraction algorithm based on NAM for image segmentation to extract the edge map from the original image. And then, we let the edge map enhance the pencil stroke to obtain a new pencil stroke with more obvious strokes and clearer details, which will improve the visual effects of the pencil drawing. Combined with the prior knowledge, where the sketch tone histograms usually follow certain patterns, the tonal histogram distribution of the pencil drawing is established, and the image is converted into a tone drawing using the histogram matching. The tone drawing with pencil texture is generated by simulating as to repeatedly draw the strokes at the same place. Finally the final pencil drawing is generated by multiplying the enhanced pencil stroke and the tone drawing with pencil texture values for each pixel.

The main contributions of this paper are embodied in the following two aspects:

Firstly, by applying the idea of NAM to the pencil drawing algorithm, a pencil drawing algorithm based on NAM is proposed.

Secondly, for the first time, the pencil drawing algorithms are tested and verified on the BSDS500 data set.

Since we use the global pencil texture to generate the tonal images, our pencil drawing algorithm based on NAM cannot be directly applied to video sequences. Pencil drawings are almost static. But video's foreground may move, that results in the inconsistent with the foreground and background. Also, applying our pencil drawing algorithm to video sequences may be good work in future.

REFERENCES

[1] G. Li, S. Bi, J. Wang, Y. Xu, and Y. Yu, "ColorSketch: A drawing assistant for generating color sketches from photos," *IEEE Comput. Graph. Appl.*, vol. 37, no. 3, pp. 70–81, May/Jun. 2017.

[2] J.-W. Yeh and M. Ouhyoung, "Non-photorealistic rendering in Chinese painting of animals," *Acta Simulata Systematica Sinica*, vol. 14, no. 9, pp. 1220–1224, Sep. 2002.

[3] A. Agrawal, "Non-photorealistic rendering: Unleashing the artist's imagination," *IEEE Comput. Graph. Appl. Mag.*, vol. 29, no. 4, pp. 81–85, Jul./Aug. 2009.

[4] R. Mukundan, "Multi-level stroke textures for sketch based non-photorealistic rendering," in *Proc. Int. Conf. Workshop Comput. Commun. (IEMCON)*, Oct. 2015, pp. 1–7.

[5] P. Lu, B. Sheng, S. Luo, X. Jia, and W. Wu, "Image-based non-photorealistic rendering for realtime virtual sculpting," *Multimedia Tools Appl.*, vol. 74, no. 21, pp. 9697–9714, 2015.

[6] H. Lee, S. Kwon, and S. Lee, "Real-time pencil rendering," in *Proc. 4th Int. Symp. Non-Photorealistic Animation Rendering*, Annecy, France, Jun. 2006, pp. 37–45.

[7] M. C. Sousa and W. J. Buchanan, "Computer generated graphite pencil rendering of 3D polygonal models," *Comput. Graph. Forum*, vol. 18, no. 3, pp. 195–207, 1999.

[8] C. Zhao, B. Gao, W. Deng, and H. Zhang, "A pencil drawing algorithm based on wavelet transform multiscale," in *Proc. Prog. Electromagn. Res. Symp.—Fall (PIERS—FALL)*, Nov. 2017, pp. 73–79.

[9] J. Zhang, R.-Z. Wang, and D. Xu, "Automatic generation of sketch-like pencil drawing from image," in *Proc. IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*, Jul. 2017, pp. 261–266.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[11] A. Selim, M. Elgharib, and L. Doyle, "Painting style transfer for head portraits using convolutional neural networks," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–18, 2016.

[12] N. Wang, X. Gao, L. Sun, and J. Li, "Bayesian face sketch synthesis," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1264–1274, Mar. 2017.

[13] C. Peng, X. Gao, N. Wang, D. Tao, X. Li, and J. Li, "Multiple representations-based face sketch–photo synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2201–2215, Nov. 2016.

[14] J. Li, X. Yu, C. Peng, and N. Wang, "Adaptive representation-based face sketch-photo synthesis," *Neurocomputing*, vol. 269, pp. 152–159, Dec. 2017.

[15] G. Kim, T. Kim, D.-H. Lim, and C. Yim, "Sketch filter for feature extraction and rendering applications," *Electron. Lett.*, vol. 49, no. 13, pp. 805–806, 2013.

[16] D. L. Way, M. K. Yang, Z. C. Shih, and R. R. Lee, "A colored pencil non-photorealistic rendering for 2D image," *Int. J. Innov. Comput. Inf. Control*, vol. 10, no. 1, pp. 233–241, 2014.

[17] C. Lu, L. Xu, and J. Jia, "Combining sketch and tone for pencil drawing production," in *Proc. Symp. Non-Photorealistic Animation Rendering*, 2012, pp. 65–73.

[18] Y. Zheng and M. Sarem, "A novel binary image representation algorithm by using NAM and coordinate encoding procedure and its application to area calculation," *Frontiers Comput. Sci.*, vol. 8, no. 5, pp. 763–772, 2014.

[19] Y. Zheng and M. Sarem, "A fast region segmentation algorithm on compressed gray images using non-symmetry and anti-packing model and extended shading representation," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 153–166, Jan. 2016.

[20] Y. Zheng and M. Sarem, "An improved ORNAM representation of gray images," *Int. J. Comput. Sci. Eng.*, vol. 17, no. 2, pp. 234–243, 2018.

[21] Y. Zheng, Y. Chang, and M. Sarem, "Accurate computation of geometric moments using non-symmetry and anti-packing model for color images," *Int. J. Comput. Commun. Eng.*, vol. 6, no. 1, pp. 19–28, 2017.

[22] Y. Zheng, Z. Yu, J. You, and M. Sarem, "A novel gray image representation using overlapping rectangular NAM and extended shading approach," *J. Vis. Commun. Image Represent.*, vol. 23, no. 7, pp. 972–983, 2012.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[24] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

[25] H. Kang, S. Lee, C. K. Chui, "Coherent line drawing," in *Proc. 5th Int. Symp. Non-photorealistic Animation Rendering*, 2007, pp. 43–50.

[26] S. Wang, E. Wu, Y. Liu, X. Liu, and Y. Chen, "Abstract line drawings from photographs using flow-based filters," *Comput. Graph.*, vol. 36, no. 4, pp. 224–231, 2014.

[27] N. Li and Z. Huang, "A feature-based pencil drawing method," in *Proc. 1st Int. Conf. Comput. Graph. Interact. Techn. Australasia South East Asia*, New York, NY, USA, 2003, pp. 135–140.

[28] S. Yamamoto, X. Mao, and A. Imamiya, "Enhanced LIC pencil filter," in *Proc. Int. Conf. Comput. Graph. Imag. Vis.*, Washington, DC, USA, Jul. 2004, pp. 251–256.

[29] H. Yang, Y. Kwon, and K. Min, "A stylized approach for pencil drawing from photographs," *Comput. Graph. Forum*, vol. 31, no. 4, pp. 1471–1480, 2012.

[30] B. Cabral and C. Leedom, "Imaging vector fields using line integral convolution," in *Proc. Conf. Comput. Graph. Interact. Techn. SIGGRAPH*, 1993, pp. 263–270.

[31] N. Ji, S. Shan, Z. Ma, and X. Chen, "Artist-drawing inspired automatic sketch portrait rendering," in *Proc. Nicograph Int. (NicoInt)*, Jul. 2016, pp. 16–156.

YUNPING ZHENG was born in Hubei, China, in 1979. He received the B.S. degree in computer science from Airforce Radar Academy, Wuhan, in 2001, and the M.S. and Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, in 2005 and 2008, respectively.

He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He has published over 70 articles in refereed journals and conferences. His interests include computer graphics, fractal image compression, image processing, and pattern recognition.

Dr. Zheng is a member of the ACM.

YUMING CHEN was born in Guangdong, China, in 1993. He received the B.E. degree in network engineering from South China Normal University, Guangzhou, in 2017. He is currently pursuing the master's degree with the Department of Computer Technology, South China University of Technology.

His research interests include image processing and computer vision.

MUDAR SAREM received the B.S. degree in electronic engineering from Tishreen University, Lattakia, Syria, in 1989, and the M.S. and Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2002, respectively.

He is currently an Associate Professor with the School of Software Engineering, Huazhong University of Science and Technology. He has published over 50 articles in refereed conferences and journals. His research interests include computer graphics, multimedia databases, and image processing.