DOI: 10.1111/cgf.13334

COMPUTER GRAPHICS *forum*
Volume 37 (2018), number 6 pp. 395–409

PencilArt: A Chromatic Penciling Style Generation Framework

Chengying Gao[1], Mengyue Tang[1], Xiangguo Liang[1,2], Zhuo Su[1,2] and Changqing Zou[3]

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China
mcsgcy@mail.sysu.edu.cn, tmy1107@126.com, 553429515@qq.com, suzhuo3@mail.sysu.edu.cn
[2]National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
[3]UMIACS, University of Maryland, College Park, MD, USA
aaronzou1125@gmail.com

Abstract

Non-photorealistic rendering has been an active area of research for decades whereas few of them concentrate on rendering chromatic penciling style. In this paper, we present a framework named as PencilArt for the chromatic penciling style generation from wild photographs. The structural outline and textured map for composing the chromatic pencil drawing are generated, respectively. First, we take advantage of deep neural network to produce the structural outline with proper intensity variation and conciseness. Next, for the textured map, we follow the painting process of artists to adjust the tone of input images to match the luminance histogram and pencil textures of real drawings. Eventually, we evaluate PencilArt via a series of comparisons to previous work, showing that our results better capture the main features of real chromatic pencil drawings and have an improved visual appearance.

Keywords: non-photorealistic rendering, image/video editing, image processing

ACM CCS: I.4.9 [Image Processing and Computer Vision]: Applications—

1. Introduction

Chromatic pencil drawing is an essential artistic style for the painting beginner with charming and aesthetic appearance (see Figures 1 and 2). However, it is challenging to make the computer produce the style generation automatically, especially in faithfully representing the rich colours and the various pencil textures involved in the art. As illustrated in the bottom row of Figure 2, in the creative process of a real painting, the chromatic pencil work is usually generated step by step: outline sketching, underpainting and detail compensation. Computer-aided chromatic pencil drawing rendering can be processed in a similar way.

For the chromatic penciling from natural photos, we observe that three factors are crucial: outline generation, tone mapping and texture rendering. The outlines describe the contours of the salient objects in the fashion of hand drawing, expressed as various thickness, grey-scale and curvature in different locations; Unlike oil paintings or watercolour arts, visual perception elements, including high brightness, low saturation and contrast awareness, are usually required to express the chromatic penciling style; moreover, stroke directions (hatching, dotting, crossing, etc.) and various material properties such as paper texture should also be considered.

There are some penciling solutions proposed to achieve these key factors in different ways. Yamamoto *et al.* [YMI04a] and Matsui *et al.* [MJN05] selected the colours of pencils to present the tone for each region, and created some stroke directions with the local information for texture rendering. Yang *et al.* [YKM12] controlled the stroke direction by determining pencil stroke flow from the objects in the input images. Lu *et al.* [LXJ12] proposed a monochrome penciling framework with a structural outline generation scheme and extended it to achieve the chromatic result. However, these methods have not adequately considered the three key factors all sidedly. For example, outline generation was ignored in [YMI04a, MJN05] and [YKM12]; unique tone features of chromatic pencil drawings were not considered in [LXJ12], which probably leads to inharmonious results or obvious artefacts.

In this paper, we present a new chromatic penciling style generation framework named as PencilArt. Our framework aims to integrate a series of image operations to generate a chromatic penciling style which adequately considers outline generation, tone mapping and texture rendering. The framework mimics the procedure of real chromatic penciling creation. It has three main modules as illustrated in Figure 3: (1) a convolutional neural network (CNN)

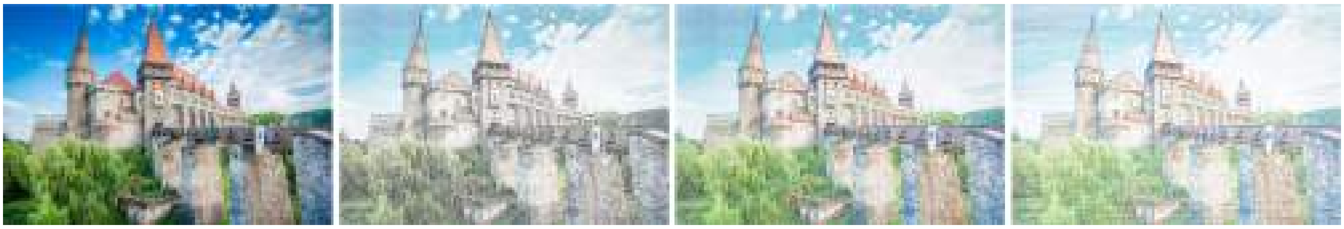


Figure 1: *Chromatic penciling style generation. From left are input image and three typical chromatic penciling results: dotted texture pattern and monochrome outline style, dotted texture pattern and inherent-hue outline style and horizontal hatching texture pattern and inherent-hue outline style.*

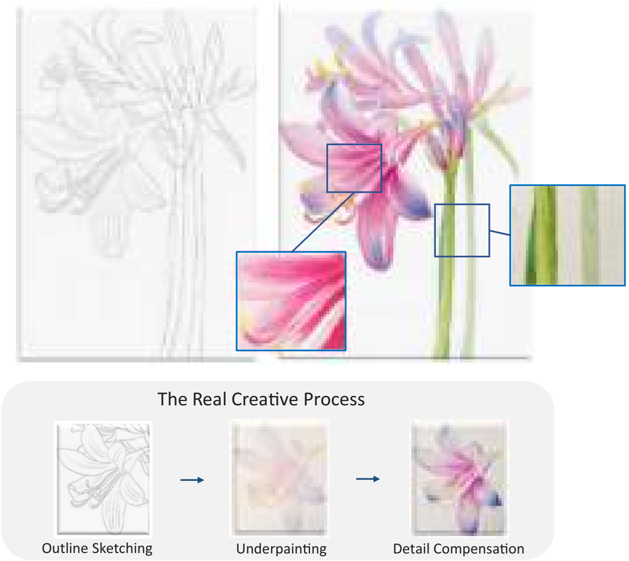


Figure 2: *Top row are a typical real chromatic pencil work and its structural outline drawing. The close-ups show the unique chromatic penciling textures. Bottom row is to illustrate the procedure for chromatic pencil drawing creation in a real painting scene.*

to produce the proper pencil-like structural outline which provides sufficient generalization ability for various types of input images; (2) a penciled stylization model which mimics the colouring process of artists and can produce a soft and light appearance similar to real drawings and (3) a fusion module which fuses the outputs of the former two modules to the final art.

We have evaluated PencilArt via comprehensive experiments. We also conducted a serious study to compare PencilArt with existing state-of-the-art work. Our experiments and user study show that our results can capture the main features of the real chromatic pencil drawings and have better visual appearances.

Our main contributions are summarized as follows:

- A new framework for the chromatic penciling style generation achieving better results than present state-of-the-art methods.
- A CNN-based structural outline generation model producing pencil-like outline drawings with great adaptability.
- A global penciled stylization model generating material texture appearance with high brightness and low saturation close to real chromatic drawings.

The rest of this paper is organized as follows. Section 2 reviews previous work related to penciling style rendering. Section 3 details the method of CNN-based structural outline generation. Section 4 describes the penciled stylization model. Section 5 presents our results and makes a comprehensive comparison with several state-of-the-art methods to demonstrate the effectiveness of PencilArt. Section 6 introduces two other potential applications. Section 7 concludes the work.

2. Related Work

Pencil drawing rendering is generally associated with Non-Photorealistic Rendering. In the past three decades, a great deal of research has been proposed to transform images/videos into artistic-style rendered content. A full review of this topic is beyond the scope of this paper. A recent survey can be referred to [KCWI12]. The following are several representative work. Durand *et al.* presented an interactive system which can simulate a class of styles including pencil drawing in [DOM*01]. Semmo *et al.* [SRT*16] presented a GPU-based framework that parametrizes image filters at three levels of control. In addition to traditional methods, image analogies [HJO*01] can also produce pencil drawing results from input images. This method requires extra training image pairs. More recently, Gatys *et al.* [GEB16] introduced an artificial system which applies deep neural networks to create the artistic images.

Most of existing work mainly aims at general art styles and has no specific module tailored to chromatic pencil drawing. Therefore, it is not easy for these methods to obtain decent penciling style rendering results. Next, we briefly review the prior work related to image-based pencil drawing rendering from three aspects.

2.1. Line drawing

Apart from traditional edge detection algorithms, researchers have proposed extensive line drawing methods in recent years. [SKLL07] imitated the human line drawing process with two modules: line extraction and line rendering. This technique can generate various kinds of line drawings from 2D images through the control over detail, focus and style. Bhat *et al.* [BZCC10] presented an optimization framework which unifies multiple ideas in the gradient-domain

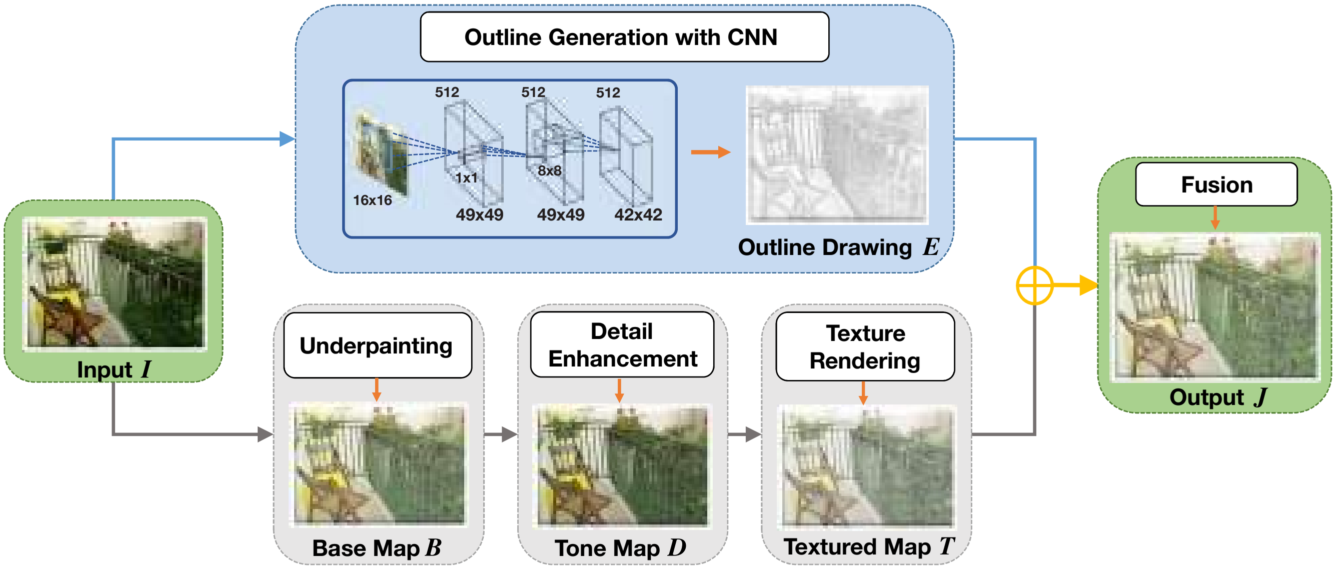


Figure 3: *The framework of PencilArt. Convolutional neural network is employed to produce the structural outline; penciled stylization model including three sequential processes, underpainting generation, detail enhancement and pattern-based pencil texture rendering, is utilized to satisfy the key features of chromatic pencil drawings in terms of tone mapping and texture rendering; outline and textured map fuse together and generate the final art.*

literature through a single optimization formulation. They also proposed a novel metric for measuring local gradient saliency that gives rise to long coherent edges. Another class should be mentioned is Difference-of-Gaussians (DoG) filter-based method, which is generally considered to be initiated by Marr and Hildreth *et al.* [MH80]. Among this line of work, Winnemoller *et al.* [WLOG06] generalized DoG for general colour images and videos. Kang *et al.* [KLC07] presented a flow-based anisotropic DoG (F-DoG) filtering technique. Winnemoller *et al.* [WKO12] extended DoG to achieve results spanning a variety of styles and a range of subtle artistic effects. Generally, the above work is not quite applicable to shape hand-drawn-style pencil outlines. More recent deep learning-based methods such as [SWW*15, BST15, XT15, ZXSY16] and [YPC*16], focus on contour detection, not pencil-like outline generation in this paper.

There are also a couple of work focused on pencil outline sketching. Zhou and Li [ZL05] proposed a computationally efficient algorithm for gradient estimation to automatically generate pencil-like structural outlines from personal photos. The convolution framework in [LXJ12] generates line strokes with the essential characteristics of pencil sketch. Both of these two work is not strong at producing satisfactory outline details.

2.2. Pencil texture rendering

Physical models that attempt to mimic the physical properties of materials such as graphite and paper were widely used in early attempts to model the texture of real pencil drawing. Sousa and Buchanan [SB99] presented a blender and eraser model to enhance the rendering results, producing realistic looking graphite pencil tones and textures. Takagi *et al.* [TNF99] proposed a volume graphics model for chromatic pencil drawing, describing the microstructure of paper, pigment distribution on paper and pigment redistribution in a volumetric fashion. But physical models require a lot of computation, and it is difficult to control it to produce satisfying results.

Beyond the physical models, Line Integral Convolution (LIC) plays an important role in most previous work about pencil texture generation. Mao *et al.* [MNI01] first used LIC to produce the pencil shading effect based on the local structure orientation. Li and Huang [LH03] generated a pencil drawing using the feature geometric attributes obtained by analysis of image moment and texture of each region. Yamamoto *et al.* [YMI04b] proposed an extension to LIC to solve the inconsistent segmentation problem, by dividing a source image into layers of successive intensity ranges, generating a stroke image for each layer and adding them together to obtain the final pencil drawing. Chen *et al.* [CZG*08] alternatively considered superposition of edges, unsharp masked image and texture for region-based LIC. Yang *et al.* [YKM12] developed a swing bilateral LIC (SBL) filter to express the styled pencil drawings. But LIC is unstable for highly textured or noisy regions, as the right convolution direction is hard to decide.

2.3. Pencil drawing colourization

A few studies consider the colour processing of chromatic pencil drawings. The method [YMI04a] automatically selects the best colour set for individual regions in a source image and then builds the target colour through optical blending. [MJN05] first determines which colours of pencils to use and how deep to push the pencils, then draws several types of strokes such as strokes for outlines, basecoats and shading. This method allows drawn strokes to overlap each other. [YKM12] achieves the colour customization of the pencil stroke flow by executing a bilateral convolution filter to a colour noise. Kim *et al.* [KWY14] proposed a coloured pencil filter with small amount of computations to create images in coloured pencil style. In contrast, our penciled stylization model is much more efficient and has better visual quality.

3. Pencil-Like Outline Generation

The artists often draw outlines at the first stage of penciling. An example of a penciling outline is provided in the upper left corner of Figure 2. With reference to [ZL05], the essential properties of a chromatic penciling outline include: (1) Only the most characteristic lines are drawn. (2) The grey-scale and thickness of outlines are used to depict various types of transitional boundaries and are roughly proportional to the local gradient. (3) The strokes are long, continuous and significant. (4) Textured regions are depicted by a collection of short lines.

Unfortunately, existing N-order edge detectors (e.g. Canny, Sobel) are not capable to satisfy these requirements. Results produced by these edge detectors are usually noise sensitive (i.e. it is hard to obtain desired clean outlines steadily) and have broken lines. Some improved DoG algorithms and machine learning–based contour detectors can generate clean outlines, but they tend to neglect the proper grey-scale and thickness variations. It results the loss of the penciling appearance in practices.

As aforementioned, the convolution framework proposed by Lu *et al.* [LXJ12] can produce structural outlines with a good imitation of human drawing, but this framework is a parameter-dependent solution where specific parameters are required for individual input images. Moreover, its parameters are non-local (not adaptable to the data). For example, [LXJ12] fits fixed-length line segments, whose length is specialized by a parameter, to all contour lines in the input image's gradient map, which could easily lead to overlength intersections at lines with large curvature (see the number '40' in the first picture of Figure 4b). Although shorter line segments may alleviate this phenomenon, it may make the outlines with small curvature lose pencil-like appearances (see the outer contours in the middle picture of Figure 4b).

In this paper, we propose a CNN-based outline generation model to produce pencil-like structural outline drawings. It has three major characteristics. First, our model is end-to-end (i.e. the model directly outputs structural outlines from the input image, without pre- or post-processing). Secondly, the result generated by our model is not parameter-dependent (i.e. our model is completely adaptable to input images). It learns the optimal parameters implicitly from the data, thus does not require the user to manually set up parameters. Lastly, our model has trained multiple convolution kernels in various sizes and can produce outlines with more pencil-like details for various image contexts.

3.1. Vectorized convolutional network

The pencil-like outline generation is transformed as an image-to-image problem. We adopt an efficient vectorized convolutional neural network (VCNN) model [RX15] to generate structural outlines. The convolution operation of VCNN can be formulated as follows:

$$[y_i]_i = \sigma([\phi(x)]_i * [W_i]_i + [b_i]_i), \quad (1)$$

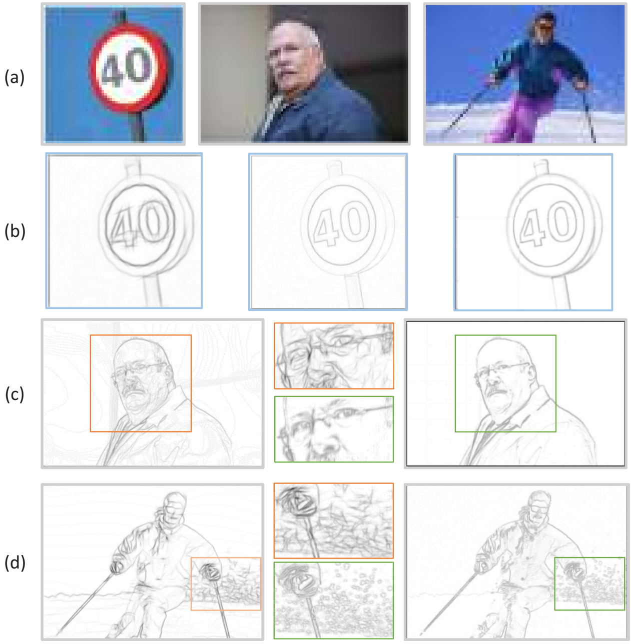


Figure 4: *(a) Input images. (b) The two images on the left are the results of [LXJ12] with long and short line segments, respectively, and the right is ours. (c) and (d) The left are the results of [LXJ12], while the right are ours. The middle are the comparisons of close-ups. This figure shows that our deep learning model can generate more stable and satisfactory results.*

where x is the input of convolution layer and y_i is the ith output. σ is a non-linear transfer function, which is ReLU in our model. W_i is the ith convolution kernel, b_i is the ith bias vector, ϕ refers to the vectorization operator and the operator $[.]_i$ assembles vectors with index i to form a matrix. This operation is a simplification of extracting matrix from original data or feature maps. The generative process of the structural outline E from an input image is defined as

$$E = F(I), \quad (2)$$

where I is the input colour image, F is the mapping from I to E that needs to be learned by the neural network. Three convolutional layers and a mapping layer are established in our implementation (see Figure 3). In the first layer of the network, 512 16 × 16 convolutional kernels are applied to detect the outline features. In the second layer, there are totally 512 1 × 1 convolutional kernels, which are expected to select appropriate lines so that they can be combined to generate desired results. In the third layer, we use 512 kernels with size of 8 × 8 to generate suitable grey-scale variation, which plays an important role in making the result close to real pencil drawings.

3.2. Training network

The most reliable training data for our vectorized network should be the real structural outlines drawn by artists. But it is hard to collect a

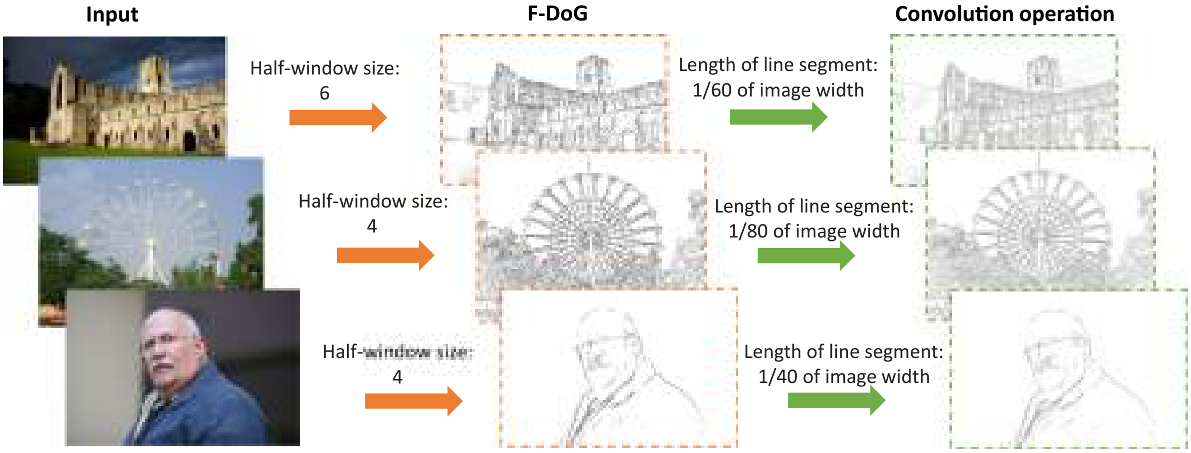

Input
F-DoG
Convolution operation
Half-window size: 6
Length of line segment: 1/60 of image width
Half-window size: 4
Length of line segment: 1/80 of image width
Half-window size: 4
Length of line segment: 1/40 of image width

Figure 5: *The flowchart of the training data generation. For each input image in the data set, we manually adjusted the half-window size of the bilateral filter in F-DoG [KLC07] and the length of line segment in [LXJ12], while the rest of algorithm parameters remain unchanged.*

large size of such data in practice. The structural outlines generated by [LXJ12] are closest to the real case. Therefore, we would like to leverage [LXJ12] to generate the training set.

The forward difference operation is one part of the outline generation framework of [LXJ12] to produce the gradient map for input image. However, this operation is sensitive to noises and the changing of light in images, and is likely to produce messy contour lines on areas with rich textures. We therefore replace the forward difference operation of [LXJ12] with F-DoG [KLC07], which could not only reduce noise but also enhance the gradient response.

Since the accuracy of the training data is very important to the neural network's result, for each image in the data set, we manually adjust the two parameters which have the biggest impact on the outline appearance, until a visually satisfactory outline is achieved. These two parameters include the half-window size of the bilateral filter in F-DoG and the length of line segment in [LXJ12]. The flowchart of our training data generation is provided in Figure 5. We selected 800 satisfactory structural outlines of various images and got 16 000 patches in 227 × 227 size by randomly cut to compose our training set. Though [LXJ12] is unable to achieve satisfactory outline appearance in training images' every location, with a large number of training data, the CNN model will learn appropriate treatments for different local regions, resulting in a network with a global-level better versatility for all images.

For the network training, the loss function L is defined in the following:

$$L(E, G) = \min \sum \|E - G\|_2, \tag{3}$$

where E is the result of the network. The loss is the L_2 norm between the network result E and the expected result G, which can help improve the overall appearance of the output structural outline. We apply the backpropagation algorithm to propagate the loss through the network and the stochastic gradient descent algorithm to update the weights, and finally obtain the trained model.

Here, we show the effectiveness of our CNN model through several typical examples. In Figure 4(a), [LXJ12] is hard to obtain a result which has no curve intersection at contours with large curvature and simultaneously keep the hand-drawn sense at contours with small curvature. Our CNN model has no problem in this case (see the third picture in Figure 4b). In Figures 4(c) and (d), we used the default algorithm parameters for [LXJ12]. It is easy to find that [LXJ12] is relatively more sensitive to shadow and weaker to preserve tiny structures compared to our CNN model. In Figure 4(c), because of shadow, many unnecessary wrinkles on the face are produced by [LXJ12]; in Figure 4(d), the snowflakes with tiny edges lose the circular shapes. These limitations may result in detail damage. In contrast, our method is robust in these cases (see the results of our CNN model in Figure 4). Another significant benefit is that the outlines generated by our CNN model have appropriate grey-scale and thickness variations. Moreover, although there are some regions become discontinuous in some training examples (like the third row of Figure 5), the self-learning ability of CNN makes up for this problem. Readers can see more evidence in Section 5.

4. Penciled Stylization Model

A chromatic pencil drawing has a special tonal appearance that is distinguished from other artistic styles. By observing a large number of chromatic pencil drawings, we found their main colour characteristic is presented through brightness and the saturation. To further investigate this characteristic of the chromatic pencil drawings, we have done some experiments on several real chromatic penciling samples with various scenes. Figure 6 illustrates three samples and their brightness distributions in the first and second columns. Obviously, these examples have similar brightness distributions and tend to aggregate at high intensities. The saturation distributions of these examples have the similar circumstance, i.e. pixels of all these examples aggregate at low saturations as shown in the third column of Figure 6. We therefore conclude that the chromatic pencil drawings have that common characteristic of high brightness and

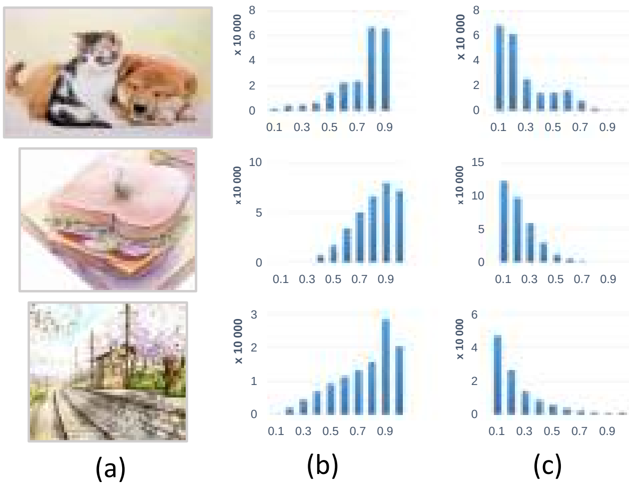


Figure 6: *Three real chromatic pencil works are shown in (a), with the corresponding brightness and saturation distributions in (b) and (c), which share the common features of high brightness and low saturation. In (b) and (c), horizontal axes denote brightness and saturation, respectively, and vertical axes denote pixel number.*

low saturation. This observation is the base of the proposed penciled stylization model.

We build our penciled stylization model according to the real colouring process, with modules: underpainting generation, detail enhancement and texture rendering. The first step is to set a homogeneous overall tone with high brightness. The second step makes dark compensation for underpainting, which uses the contrast information between the light and shade of the input image. The third step is to render the texture of chromatic pencil drawing and reduce the colour saturation. Next, we give the details of the steps of our penciled stylization model. All pixel values are normalized to the range [0, 1].

4.1. Underpainting generation

Artists usually create an underpainting as the basic tone in the first step of the colouring process. Our penciled stylization model has also simulated this step. Specifically, we get a base map B from the input image I by enhancing the global brightness while emphasizing the edges.

As it is a brightness-related operation, we first convert the input image I from RGB to YUV colour space. Next, we apply the sketch filter [KWY14] to the I_Y channel as following to obtain the M_Y channel of an excessive bright map M:

$$M_Y(x, y) = I_Y(x, y)/\mathcal{R}(x, y), \tag{4}$$

where $\mathcal{R}$ denotes the response map through using the maximum filter with kernel size of 5 × 5 to I_Y, and (x, y) is the pixel location. In the response map $\mathcal{R}$, the edges will be sharply modified, but the non-edge areas are less affected.

After that, the triplet (M_Y, U, V) is reconverted from YUV to RGB colour space. Since the values of I_Y and $\mathcal{R}$ are similar in the areas without edges, the intensities of pixels at the corresponding locations of M will aggregate within a narrow range of high brightness. Meanwhile, since I_Y and $\mathcal{R}$ are dissimilar on edges, the intensities of pixels at the corresponding locations of M will become darker, as shown in Figure 7(b). After taking the average of I and M for each RGB channel, the resulting base map B will have a soft and pastel colour style as an underpainting image (see Figure 7c).

4.2. Detail enhancement using gamma expansion

Representing shading is a critical issue in the penciling procedure. The artists usually achieve the effect by enhancing the dark cues in the drawing. In general, shading information can be extracted from the input image. However, the underpainting generation step which enhances the global brightness will weaken the shading information. Therefore, it is necessary to compensate the dark details of base map B to obtain the tone map D.

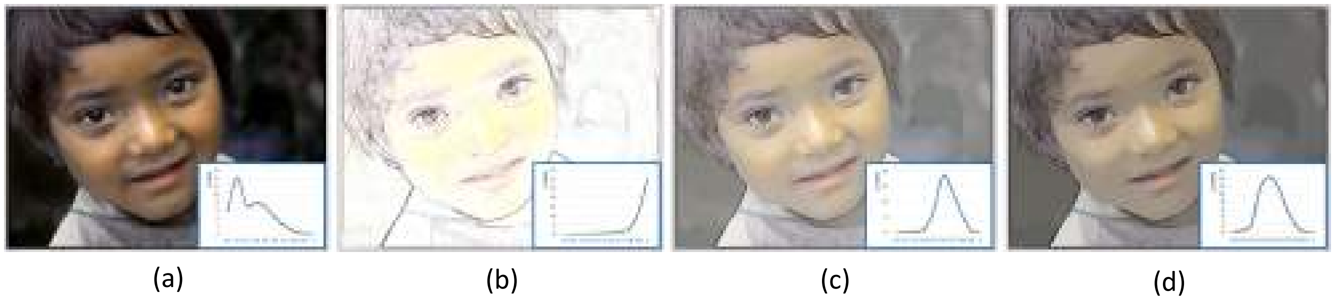


Figure 7: *Tone adjustment of our penciled stylization model. (a) Input image, (b) excessive bright map M, (c) base map B and (d) tone map D. The corresponding brightness curves are shown in the bottom right corner of each image (horizontal axes denote brightness, and vertical axes denote pixel number). The brightness curve of base map B produced by taking the average of input image and excessive bright map M has a high global brightness. After dark detail enhancement, the brightness curve of tone map D is expanded towards darker intensity, i.e. there are more dark pixels in tone map D.*

Instead of providing an accurate physical simulation to superimpose the dark colour to the base map B [TNF99], we apply gamma expansion to the intensity of B with

$$D = B^{\gamma}. \tag{5}$$

This operation will expand the pixel distribution curve towards darker intensities, and hence enhance the contrast in bright areas. This is equivalent to adding shading on the underpainting. In our experiments, we find that a visual satisfactory result is generated with $\gamma = 1.4$. If there are some noises in the result, an edge-preserving filtering can be used for post-processing. We illustrate this detail enhancement in Figure 7(d).

4.3. Pattern-based texture rendering

The material properties of the coloured pencil and the drawing paper offer a unique texture appearance for chromatic pencil drawing. Some previous methods [MNI01, YKM12, YMI04a, MJN05] simulate this appearance by generating the stroke directions of textures.

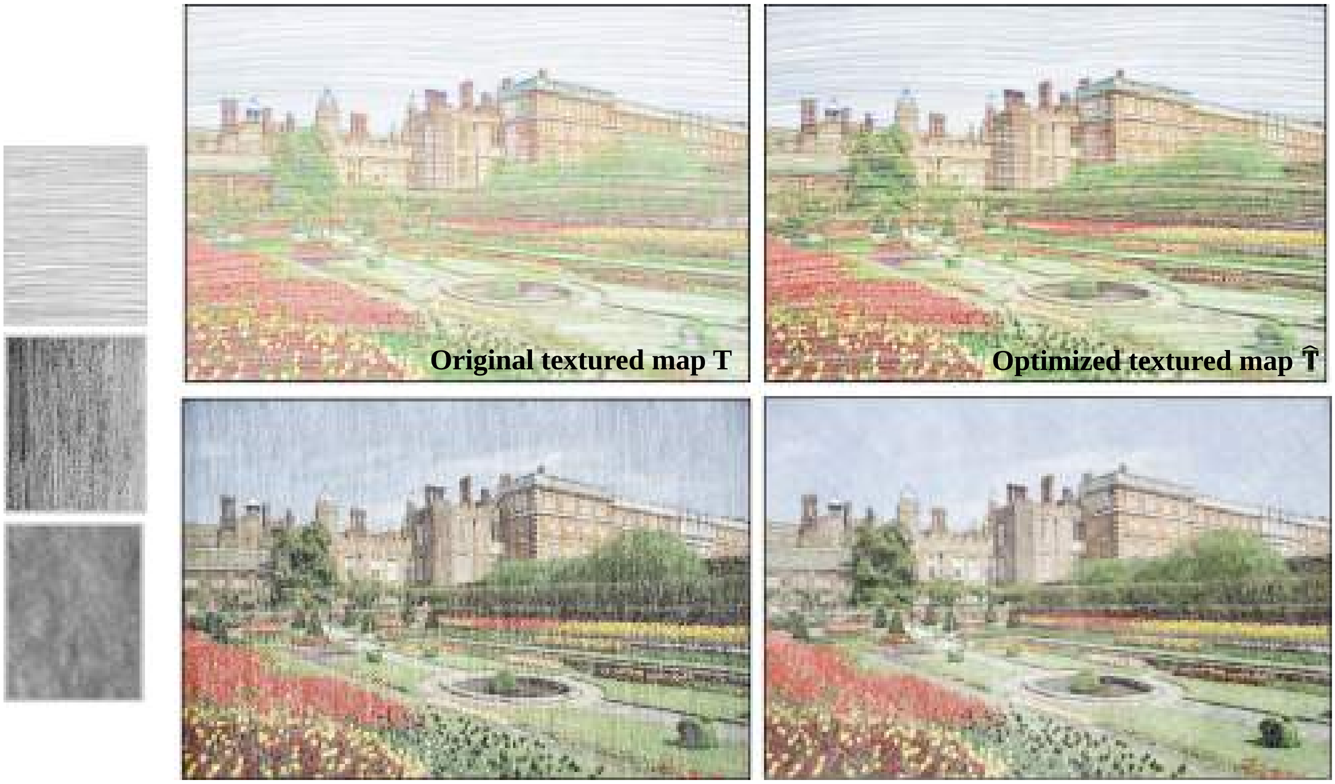


Figure 8: *Texture rendering. (Left) Three pencil texture patterns. (Right) Four textured maps. On the right side, we show the original textured map (upper left), the optimized textured map (upper right) generated by Equations* (6) *and* (7) *using the top texture pattern on the left side and the optimized textured maps (bottom left and bottom right, respectively) using the middle and bottom patterns on the left side.*

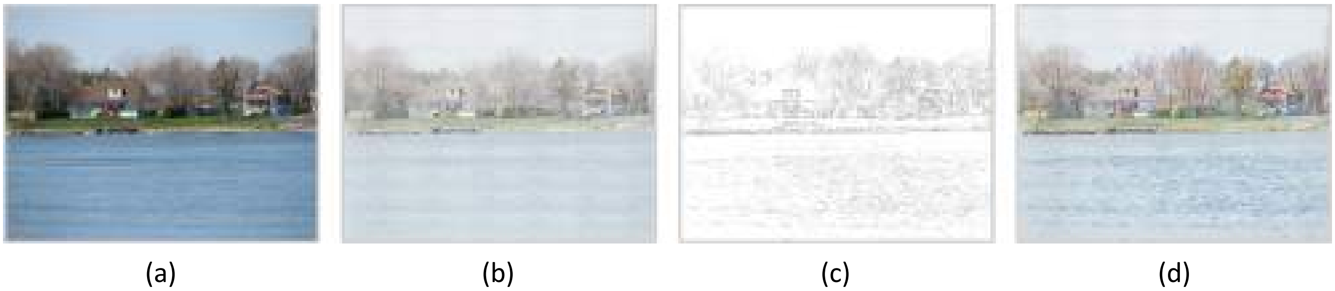


Figure 9: *An example for illustrating final result generation by fusing the outline and the textured map. (a) Input image. (b) Textured map. (c) Structural outline. (d) Resulting image.*

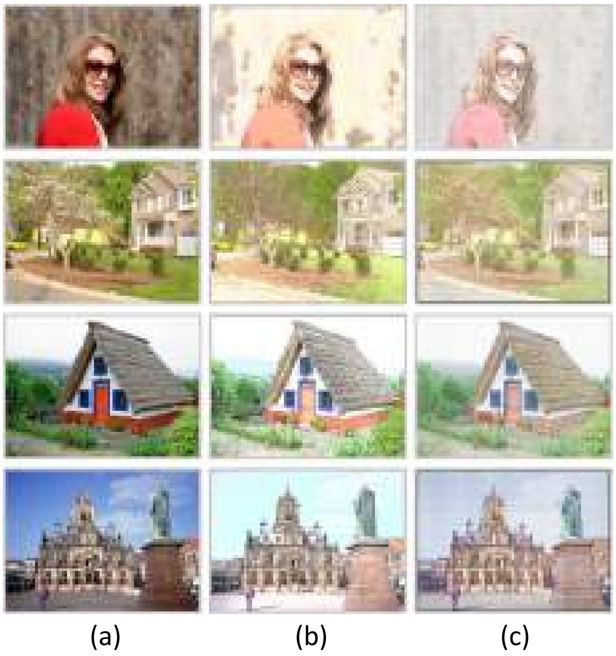


Figure 10: *Comparison with [LXJ12]. (a) Input images. (b) The results of [LXJ12]. (c) Our results. See texts in Section 5.1 for detailed explanation.*

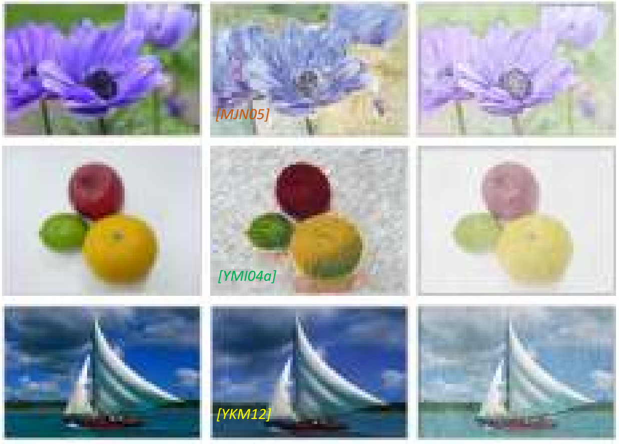


Figure 11: *Comparisons with [MJN05, YMI04a] and [YKM12].*

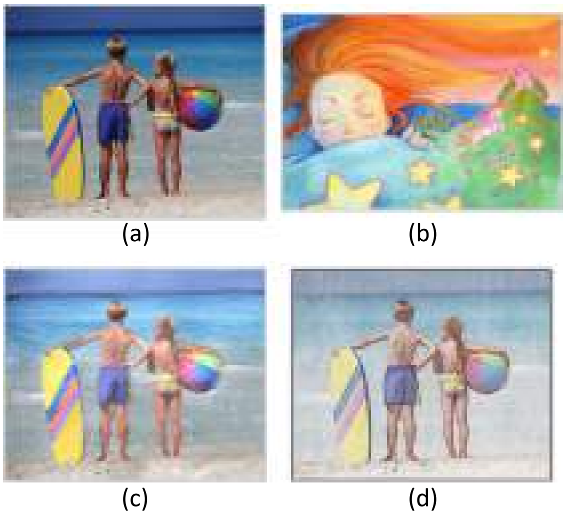


Figure 12: *Comparisons with [GEB16]. (a) Input image. (b) The target style image of a real chromatic pencil drawing used for [GEB16]. (c) The result of [GEB16]. (d) Our result.*

[LXJ12] produces tonal textures without obvious stroke direction by applying global exponential operation of selected pencil texture patterns. In this step, our goal is not only rendering the pencil texture but also reducing the colour saturation of tone map D. We leverage an efficient texture rendering algorithm to generate textured map T with sound texture appearance and globally low saturation.

Due to the noise and complex structures in input images, the texture generation model based on local gradient is not always reliable. We therefore render out a textured map T by merging the tone map D and a texture template P. P is set to be in the size equal to the input image. It is synthesized from an appropriate monochrome pencil texture pattern of real pencil drawing. Some texture pattern examples, including two hatching patterns on the drawing paper with fine and coarse textures and a crossing pattern, are shown in the left side of Figure 8.

To retain the special appearance and ‘white-noise’ look of the original texture patterns as well as reduce the saturation of the tone map D, We blend every pixel in the tone map D and the white colour (also known as the value of 1) for each RGB channel to obtain the textured map T. The proportion of blending is decided by the pixel value at corresponding location of texture template P. It can be formulated as

$$T(x, y) = D(x, y)(1 - P(x, y)) + P(x, y). \tag{6}$$

In Equation (6), the intensity of $P(x, y)$ controls the amount of white colour blended with $D(x, y)$. The lighter the intensity of $P(x, y)$ is, the lower saturated the colour of $T(x, y)$ is and vice versa. It not only allows the textured map T to have the similar special pattern as the texture template P, but also reduces the colour saturation of the whole tone map.

Directly applying Equation (6) will produce an output with the lack of colour contrast compared to the original input image I. It can be seen in the upper left image on the right side of Figure 8. The reason is that the operation above only uses the original texture template P to control the amount of mixed white colour. It does not consider the colour contrast of the input image I. Hence, to take the contrast information of the input image into account, we modify the intensity of P as follows:

$$\widehat{P}(x, y) = P(x, y)^{\beta(I_Y(x,y))}, \tag{7}$$

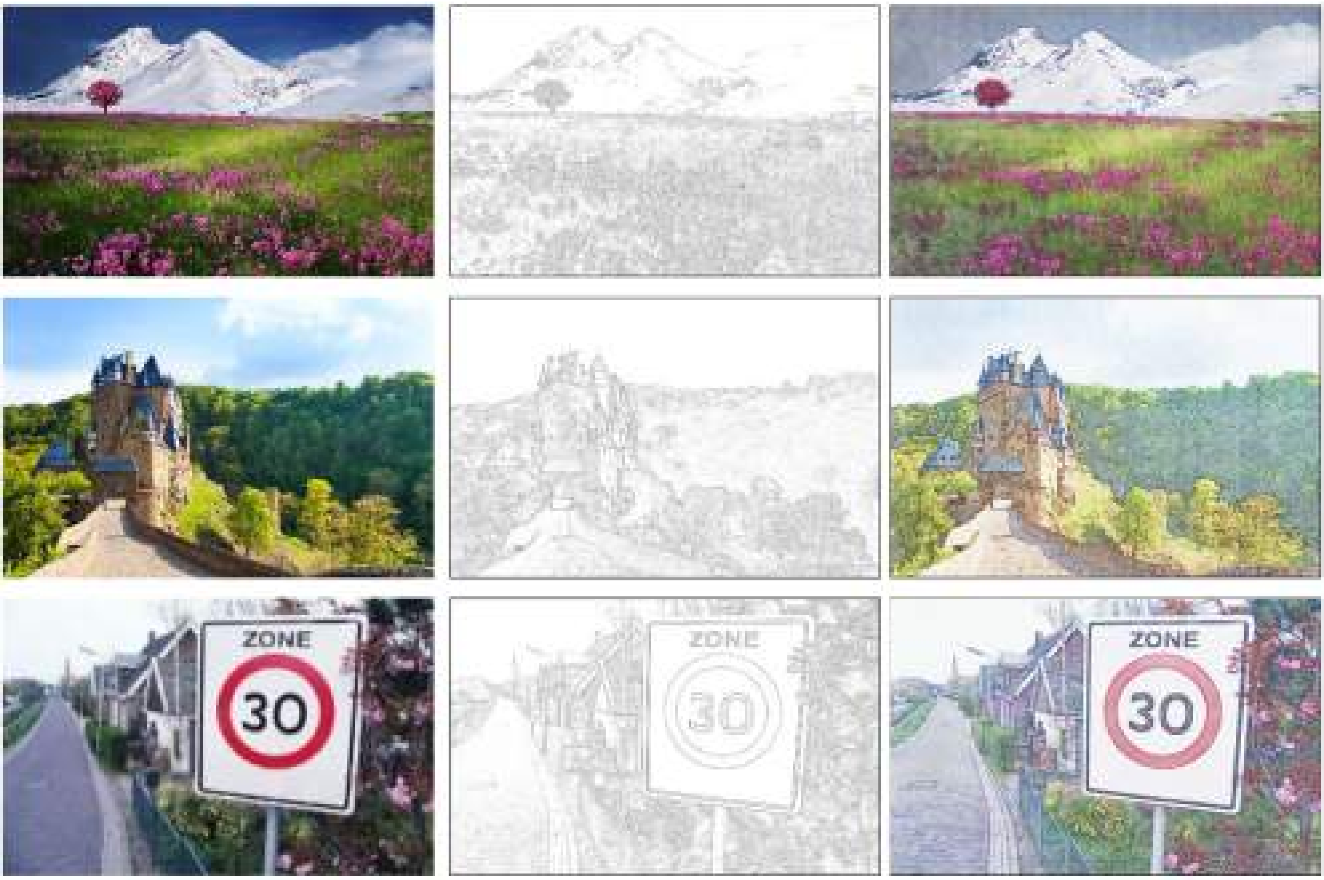


Figure 13: *More structural outlines and chromatic pencil drawing rendering results on outdoor natural landscape and architecture. The example in row 3 was collected from the website of [LXJ12], the other examples were collected from Google image.*

where I_Y refers to the Y channel of I in the YUV colour space, and β is a mapping function, taking I_Y as the parameter. For dark areas of I, we would like to blend less white colour (i.e. the pixel value of the corresponding location in $\widehat{P}$ should be much smaller than the original P), which could be obtained by a high value of β. The function β is defined as

$$\beta(i) = \log(1/i) + 1. \tag{8}$$

If the parameter i is equal to zero, we add a bias $\delta = 0.001$ to i to avoid the zero division. β works as a monotonically decreasing function to map the value of i in the range of $[0, 1]$ into the range of $[1, \log(1/\delta) + 1]$.

Replacing the texture template P with $\widehat{P}$ in Equation (6), we obtain an optimized textured map $\widehat{T}$, as shown in the upper right image on the right side of Figure 8. This modification of the texture template P helps retain original colour in D for darker areas, thus increasing the colour contrast. In addition, our texture rendering approach allows users to select favourite texture patterns. The two images in the bottom lines of Figure 8 are results produced using other two kinds of texture patterns.

4.4. Outline-texture fusion

At last, we combine the structural outline E and textured map $\widehat{T}$ to obtain the final output J. We intend to assign colour to E as well as preserve its original intensity variation. For this purpose, we convert the textured map $\widehat{T}$ into HSL colour space, and modify H, S and L channels to embed E in $\widehat{T}$. Let the $\widehat{T}_H$, $\widehat{T}_S$ and $\widehat{T}_L$ represent the H, S and L components of $\widehat{T}$, respectively. In the real painting, it is a common way to choose the inherent hues of objects as the hues of their contours. We thus set J_H equal to $\widehat{T}_H$,

$$J_H(x, y) = \widehat{T}_H(x, y). \tag{9}$$

Furthermore, to reveal and emphasize the structural outline from $\widehat{T}$, for each pixel, we increase the saturation of $\widehat{T}$ by $1 - E(x, y)$ in S channel,

$$J_S(x, y) = \widehat{T}_S(x, y) + 1 - E(x, y), \tag{10}$$

and decrease the colour brightness of $\widehat{T}$ by multiplying $E(x, y)$ in L channel,

$$J_L(x, y) = \widehat{T}_L(x, y)E(x, y). \tag{11}$$

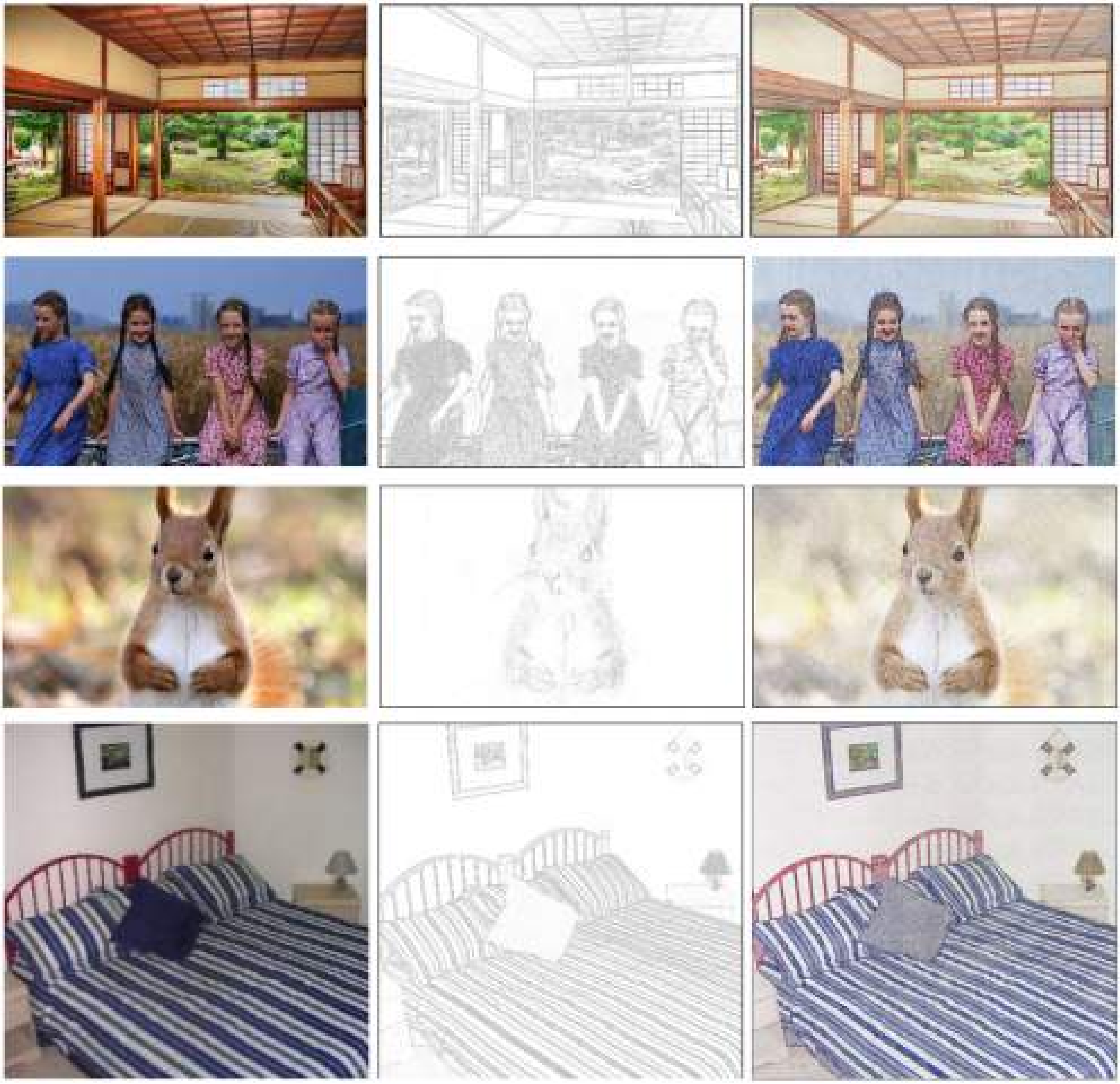


Figure 14: *More structural outlines and chromatic pencil drawing rendering results on indoor scene, people and animal. The examples in rows 1 and 2 were collected from the website of [LXJ12], the other examples were collected from Google image.*

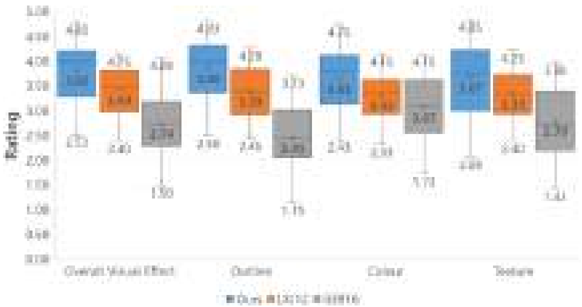
The above two operations only modify the colour of pixels belonging to the structural outlines E, while the colours of the rest pixels remain unchanged. A fusion result is shown in Figure 9.

5. Experiments and Analysis

In this section, we compare our framework with the state-of-the-art methods. Followed by a user study, where we provide a quantitative analysis for PencilArt and other two baseline methods. In addition, we have demonstrated the efficiency of PencilArt by a time analysis.

5.1. Results and comparisons

Five most representative work has been compared with PencilArt. Since [LXJ12] is the most related work, the major comparison



Standard Deviation	Overall Visual Effect			Outline			Colour			Texture		
	Ours	LXJ12	GEB16	Ours	LXJ12	GEB16	Ours	LXJ12	GEB16	Ours	LXJ12	GEB16
	0.59	0.49	0.63	0.65	0.52	0.64	0.62	0.49	0.66	0.74	0.47	0.73

User statistics: 72% male, 28% female. 72% twenties, 21% thirties, 7% forties.

Figure 15: *Statistical analysis of user investigation. The horizontal axis of the box plot represents the four evaluation aspects. The vertical axis represents the statistical score given by participants. Standard deviation of statistical results and participants' attribute information have also been provided.*

has been made between [LXJ12] and PencilArt in this section. To achieve a fair comparison, the experimental results of [LXJ12] were downloaded from its website. Some visual comparison results can be seen in Figure 10. The three earlier work has not published the code, therefore we used the best results from their papers for the comparison. The selected examples can be seen in Figure 11. The result of [GEB16] was obtained via its author's code, which is shown in Figure 12. We used different texture patterns for each image. Figure 10 used dotted pattern. The first row of Figure 11 used hatching pattern, and the others used crossing pattern.

In Figure 10, we can see that our results have a style different from [LXJ12] in terms of structural outline and tone. Relatively, structural outlines in our results have clearer structure in highly textured regions (see the hair and face in row 1, the messy leaves and grass in rows 2 and 3 and the buildings with complex contours in row 4) and richer colour. This may be benefitted by the vectorized convolutional network and outline-texture fusion employed in PencilArt. Moreover, our results have the colour style perceptually closer to the real chromatic pencil drawing than [LXJ12]. It is because our penciled stylization model is tailored for coloured pencil, rather than applying the original tone adjustment mechanism used for monochrome pencil drawing in [LXJ12].

In Figure 11, the representative example from [MJN05] and [YMI04a] are, respectively, shown in the first and second rows. Both of these two work is segmentation-based method. From the results, we can see it is hard to produce a desired segmentation when the input image has rich details (see the image in the first row). Although segmentation-based method could highlight the stroke directions within each region, it easily results in the loss of structural information. In addition, both the two work generates tone through the colour information of each region (e.g. [MJN05] uses the average colour, and [YMI04a] blends stroke layers with two colours). Unsmooth tonal change [MJN05] and unpleasant colour appearance [YMI04a] can be observed in the examples due to poor region segmentation and inappropriate colour blending. The result generated by [YKM12] in the third row has smooth directional textures. However, its tone has a significant gap with real chromatic pencil drawing. It is because [YKM12] only makes a simple combination of the original photo and the textures, ignoring, to some degree, the colour style difference between the natural photo and chromatic pencil drawing.

In addition, we have also compared our results with some general style transfer methods. In our experiment, we compared PencilArt with the state-of-the-art style transfer method [GEB16]. We show the comparison in Figure 12. We can find the result of [GEB16] has chaotic textures and weaker structural outline. Its style is closer to oil painting rather than chromatic pencil drawing, which may be partially caused by the fact that [GEB16] does not make any optimization specialized for chromatic penciling style.

By contrast, our results have sharper contours and more perceptually acceptable tones. Our CNN-based outline generator and penciled stylization model make partial contribution. Moreover, using texture patterns globally also helps PencilArt produce more natural texture. All these features enable PencilArt to output results with clear structure outlines and stable tones and textures, thus make the results more like real chromatic pencil drawings.

Figures 13 and 14 demonstrate structural outlines and chromatic pencil drawings produced by PencilArt on several representative examples depicting various scenes. In Figure 13, the images of outdoor natural landscape and architecture contain richer textures and shorter line structures but have relatively simpler tones; especially, the image in the third line contains both simple and complex textures. In Figure 14, the images of indoor scene (the first and the bottom lines) have regular object structures and large regions with single colours; people in the second line have a variety of facial expressions and complex patterns of clothes; the animal in the third image has fine hair with blurred background. The results on these various examples indicate our CNN model is powerful to generate desirable structural outlines and our framework is effective to produce high-quality chromatic penciling style.

5.2. User study

In this section, we present the quantized comparison results. We evaluated PencilArt by comparing it with two baseline methods: [LXJ12] and [GEB16]. The target style for [GEB16] is provided by the same style image in Figure 12. PencilArt was implemented with Matlab, and the results of [LXJ12] and [GEB16] were produced via the source codes from the authors of these two papers ([LXJ12] and [GEB16] used Matlab and Lua, respectively).

Total 40 test examples were randomly chosen. Some were collected from the experimental data of the comparison methods, and the others were collected from Google. These images have various styles such as figure painting, landscape painting and still life. Fifty people participated in the user study, contributing a total of 47 valid user ratings. A small percentage of participates major in art; the other participates major in computer science. More user attribute information is listed at the bottom of the Figure 15.

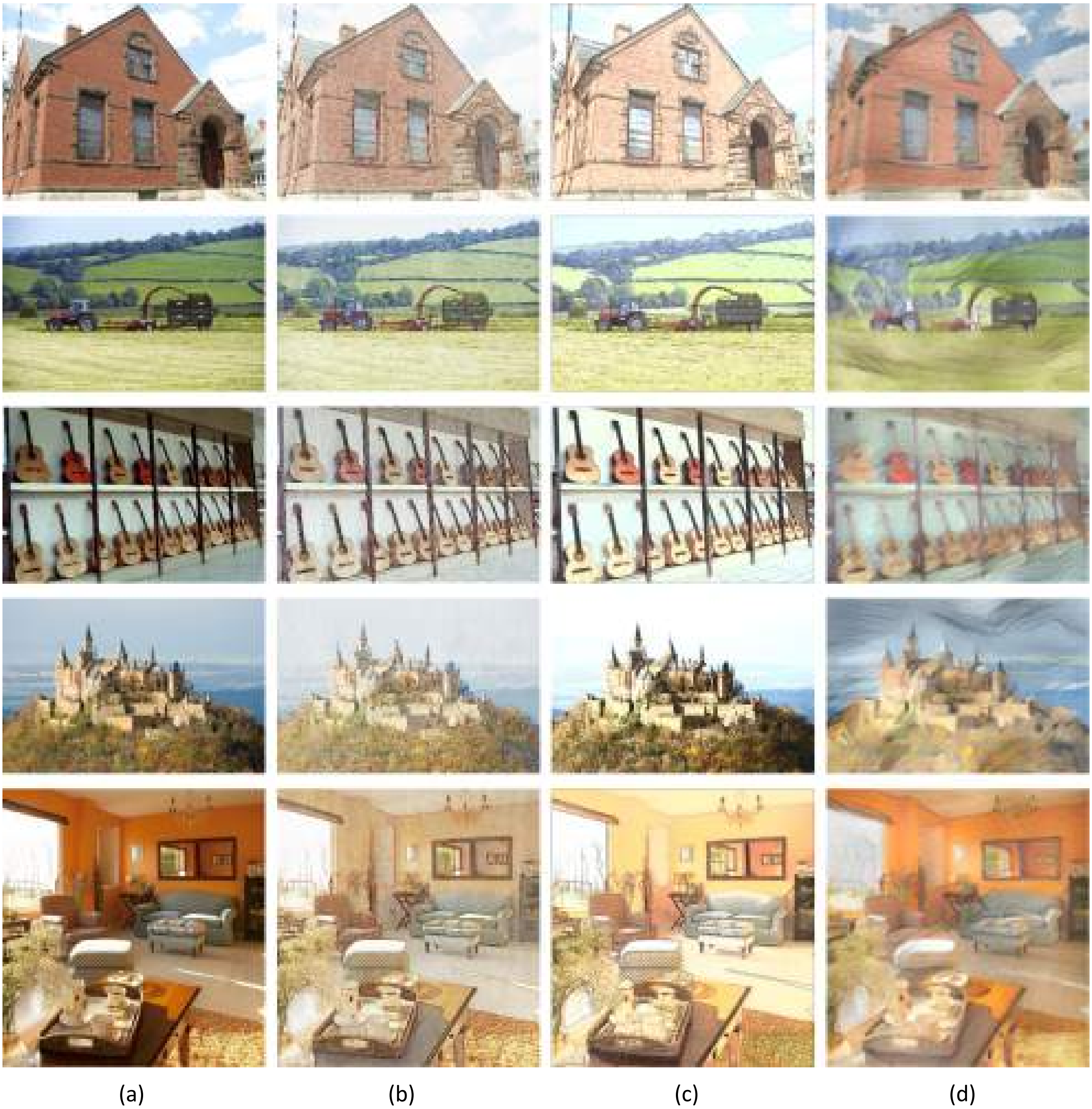


Figure 16: *Five examples for user study. (a) Input images. (b), (c) and (d) are the results of PencilArt, [LXJ12] and [GEB16], respectively.*

Four aspects have been evaluated in the user study. (1) Overall visual effect. It is an overall visual quality scale. (2) Outline quality. Unlike those brush-based artworks such as oil painting and watercolour painting, chromatic pencil drawing prefers clear outline structure and strong hand-drawn feeling. Results having clearer outlines and stronger hand-drawn feelings were rated with higher scores. (3) Colour quality. The colour of chromatic pencil drawing is characterized with high brightness and low saturation. Better results in terms of colour quality should have higher brightness and lower saturation. (4) Texture quality. Better results should have stronger pencil textures reflecting the characteristics of coloured pencils and drawing paper such as stroke overlapping and 'white noise' effects.

In our study, rating satisfaction increases by a scale from 1 to 5 for each aspect. To avoid bias, the results of the three methods were randomly arranged for each test example. In Figure 15, the statistical analysis of the user investigation data is shown through boxplot. The standard deviations for each method in the four aspects are also provided. The quantized result of the study demonstrates that PencilArt has achieved significantly better effects in every aspect against the other two methods. Although the standard deviation of PencilArt is slightly higher than [LXJ12], most of the ratings assigned to PencilArt are higher than those of [LXJ12]. Figure 16 shows some examples used in the user study.

5.3. Efficiency analysis

We tested the total running time of PencilArt and [LXJ12]. Both two methods were implemented by Matlab. We ran the two methods on a PC with Intel i5 CPU (2.20 GHz, dual-core), and tested six representative examples. The results are shown in the line chart of Figure 17. We can see that our method takes less time in all the examples. Besides, as the size of the test image increases, the time of PencilArt grows much slower than that of [LXJ12].

Table 1 lists the running time for each step of [LXJ12] and ours. Among the three steps, outline generation takes the most computational cycles in PencilArt. The time is close to that taken by [LXJ12]. It indicates that CNN used for outline generation achieves better results and does not sacrifice much efficiency. For tone adjustment and texture rendering steps, [LXJ12] uses stroke multiplication to simulate repeatedly sketching at the same place in hand drawings. It needs a large amount of computation to optimize each stroke point. As the size of the image increases, the amount of calculation will greatly increase. In contrast, due to our light-weight algorithm design, our method spends much less time than [LXJ12] in tone adjustment and texture rendering steps, especially on large-size images.

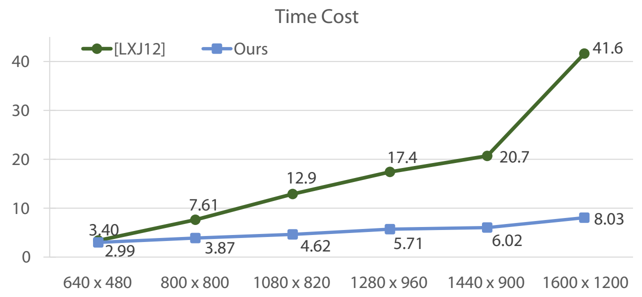


Figure 17: *The time analysis on input images in different sizes. The green line represents the time performance of [LXJ12], and the blue line represents ours.*

Table 1: *Comparison of the running time for each step (unit: seconds).*

	Outline		Tone		Texture	
Resolution	[LXJ12]	Ours	[LXJ12]	Ours	[LXJ12]	Ours
640 × 480	0.23	1.41	1.37	0.64	1.80	0.94
800 × 800	0.97	2.03	2.61	0.83	4.03	1.01
1080 × 820	1.41	2.43	3.98	1.12	7.46	1.07
1280 × 960	2.11	2.92	5.32	1.57	10.0	1.22
1440 × 900	2.25	3.05	5.67	1.71	12.8	1.26
1600 × 1200	3.82	3.89	8.44	2.56	29.4	1.58

6. Extended Applications

There are several potential applications could be explored based on PencilArt. Apart from user-defined texture pattern, colour customization and outline stylization may also enhance the user's creativity on chromatic penciling.

6.1. Dominant colour customization

The proposed method can be combined with existing techniques to produce more interesting applications. For instance, recent colour transfer methods such as [CFL*15] and [WZL*17] can be combined with our method to facilitate the art creation with more diverse visual senses. We have made such an attempt. As illustrated in Figure 18, we generated a chromatic pencil drawing shown in Figure 18(b) by applying PencilArt on the input image in Figure 18(a). Assisted by a dominant colour palette, we changed some dominant colours and produced a new chromatic pencil drawing shown in Figure 18(c). Note that existing colour transfer methods like [WZL*17] can usually complete the colour transfer process within several seconds. The combination of these two techniques can much help the artists in inspiring their imaginations.

6.2. Chromatic outline stylization

PencilArt also allows chromatic outline stylization where artists may use the outlines in uniform hue to express some specific feelings. For instance, they may use brown outlines to express warmness, use pink outlines to express romance. This can be achieved by replacing Equation (9) with the following term:

$$J_H(x,y) = \begin{cases} h, & (x,y) \in E, \\ \widehat{T}_H(x,y), & \text{otherwise}, \end{cases} \qquad (12)$$

where h is the colour hue specified by the user, and J_H, $\widehat{T}_H$ are the H component in HSL colour space for J, $\widehat{T}$. This term will change the

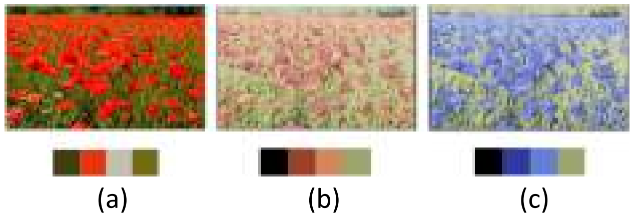


Figure 18: *Dominant colour customization via palette editing. (a) Input image and its dominant colours. (b) Chromatic penciling output of PencilArt; the bottom palette shows its dominant colours. (c) New art generated by changing the middle two colours of the palette of (b) to two blue colours shown in the palette at the bottom of (c).*

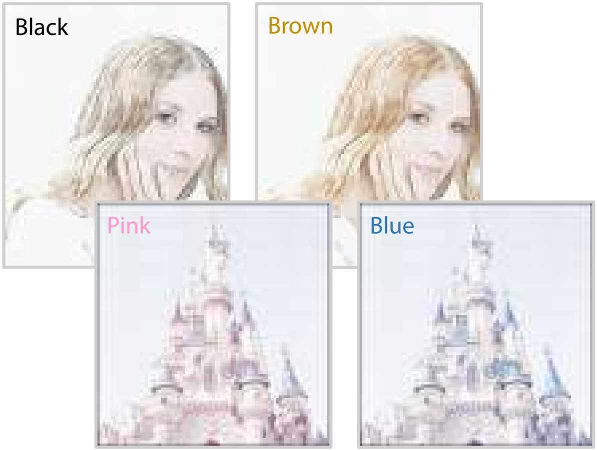


Figure 19: *Specifying the colour style of structural outlines. From top to bottom and from left to right: results with black, brown, pink and blue outlines.*

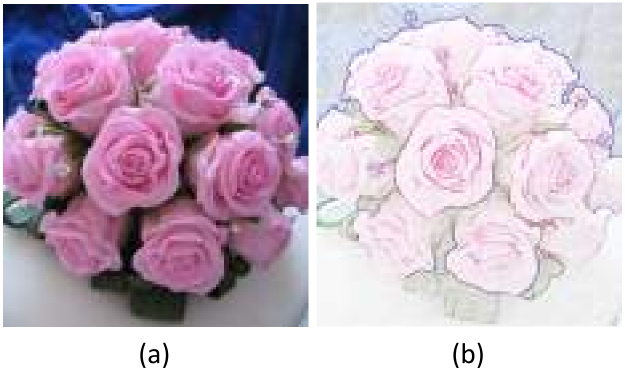


Figure 20: *Limitations. Input and output examples are, respectively, shown in (a) and (b). Some inappropriate colour is assigned to some structural outlines of the foreground object (the flower) because edges are detected from the background (the blue curtain).*

hue of outlines to the specified colour. Note that if the h is set to 0 (it means that the user uses black outline), the corresponding $J_S(x, y)$ in Equation (10) should be set to 0. Four examples in Figure 19 are shown to illustrate this feature.

7. Conclusions

We have presented a new non-photorealistic rendering framework named as PencilArt to generate chromatic pencil drawings from natural photos. PencilArt can well render the structural outlines, tones and textures much closer to real drawings painted by coloured pencils than other existing methods. In addition to basic chromatic penciling features, PencilArt can also offer the user several extra customization features. Our implementation uses CNN to generate structural outlines, and produces tones and textures by approximating the major features of real chromatic pencil drawings. We have compared our results with several state-of-the-art work. Comprehensive experiments demonstrate the effectiveness of our method.

Limitations and future work. PencilArt currently has some minor limitations. For instance, as the outline's hue is directly extracted from the input image, it sometimes happens that some background colours are assigned to the structural outlines of the foreground objects. We illustrate this case in Figure 20 where the pink contour of the flower is contaminated by the colour of the curtain. In our future work, we would like to take advantage of semantic information to improve the outline fusion result. Another limitation is the training data for outline generation. We have not used real outline drawings created by artists in the training process. We believe that our result quality can be improved if real artist-drawn data can be used as the training data.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (61502541, 61772140, 61502546), the Guangzhou Science Technology and Innovation Commission (GZSTI16EG14/201704030079), and the Science and Technology Planning Project of Guangzhou (201604016136). Zhuo Su is the corresponding author of this paper (E-mail: suzhuo3@mail.sysu.edu.cn). The authors thank all the data providers for our experiments.

References

[BST15] Bertasius G., Shi J., Torresani L.: High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (Santiago, Chile, 2015), pp. 504–512.

[BZCC10] Bhat P., Zitnick C. L., Cohen M., Curless B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics 29*, 2 (2010), 655–661.

[CFL*15] Chang H., Fried O., Liu Y., Diverdi S., Finkelstein A.: Palette-based photo recoloring. *ACM Transactions on Graphics 34*, 4 (2015), 139:1–139:11.

[CZG*08] Chen Z., Zhou J., Gao X., Li L., Liu J.: A novel method for pencil drawing generation in non-photo-realistic rendering. In *Proceedings of Advances in Multimedia Information Processing - PCM 2008, Pacific Rim Conference on Multimedia, December 9–13* (Tainan, Taiwan, 2008), pp. 931–934.

[DOM*01] Durand F., Ostromoukhov V., Miller M., Duranleau F., Dorsey J.: Decoupling strokes and high-level attributes for interactive traditional drawing. In *Proceedings of Eurographics Workshop on Rendering Techniques* (London, United Kingdom, 2001), pp. 71–82.

[GEB16] Gatys L. A., Ecker A. S., Bethge M.: Image style transfer using convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, 2016), pp. 2414–2423.

[HJO*01] Hertzmann A., Jacobs C. E., Oliver N., Curless B., Salesin D. H.: Image analogies. In *Proceedings of the 28th*

Annual Conference on Computer Graphics and Interactive Techniques (Los Angeles, CA, USA, 2001), ACM, pp. 327–340.

[KCWI12] KYPRIANIDIS J. E., COLLOMOSSE J., WANG T., ISENBERG T.: State of the 'art': A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (2012), 866–885.

[KLC07] KANG H., LEE S., CHUI C. K.: Coherent line drawing. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (San Diego, CA, USA, 2007), pp. 43–50.

[KWY14] KIM G., WOO Y., YIM C.: Color pencil filter for non-photorealistic rendering applications. In *Proceedings of International Symposium on Consumer Electronics* (JeJu Island, South Korea, 2014), pp. 1–2.

[LH03] LI N., HUANG Z.: A feature-based pencil drawing method. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (Melbourne, Australia, 2003), ACM, pp. 135–140.

[LXJ12] LU C., XU L., JIA J.: Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France, 2012), Eurographics Association, pp. 65–73.

[MH80] MARR D., HILDRETH E.: Theory of edge detection. *Proceedings of the Royal Society of London 207*, 1167 (1980), 187–217.

[MJN05] MATSUI H., JOHAN H., NISHITA T.: Creating colored pencil style images by drawing strokes based on boundaries of regions. In *Proceedings of Computer Graphics International* (Stony Brook, NY, USA, 2005), pp. 148–155.

[MNI01] MAO X., NAGASAKA Y., IMAMIYA A.: Automatic generation of pencil drawing from 2D images using line integral convolution. In *Proceedings of International Conference on Computer Aided Design and Graphics* (Dallas, TX, USA, 2001), pp. 147–159.

[RX15] REN J. S., XU L.: On vectorization of deep convolutional neural networks for vision tasks. In *The 29th AAAI Conference on Artificial Intelligence (AAAI-15)* (Austin, Texas, USA, January 2015).

[SB99] SOUSA M. C., BUCHANAN J. W.: Observational model of blenders and erasers in computer-generated pencil rendering. In *Proceedings of Graphics Interface 1999 Conference, June 2–4* (Kingston, Ontario, Canada, 1999), pp. 157–166.

[SKLL07] SON M., KANG H., LEE Y., LEE S.: Abstract line drawings from 2D images. In *PG '07: Proceedings of Pacific Conference on Computer Graphics and Applications* (Maui, HI, USA, 2007), pp. 333–342.

[SRT*16] SEMMO A., RSCHMID T., TRAPP M., KLINGBEIL M., LLNER J., RGEN PASEWALDT S.: Interactive image filtering with multiple levels-of-control on mobile devices. In *Proceedings of ACM SIGGRAPH Asia Symposium on Mobile Graphics and Interactive Applications* (Macau, 2016), pp. 2:1–2:8.

[SWW*15] SHEN W., WANG X., WANG Y., BAI X., ZHANG Z.: Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of Computer Vision and Pattern Recognition* (Boston, USA, 2015), pp. 3982–3991.

[TNF99] TAKAGI S., NAKAJIMA M., FUJISHIRO I.: Volumetric modeling of colored pencil drawing. In *Proceedings of Seventh Pacific Conference on Computer Graphics and Applications* (Seoul, South Korea, 1999), pp. 250–258, 329.

[WKO12] WINNEMOLLER H., KYPRIANIDIS J. E., OLSEN S. C.: XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers and Graphics 36*, 6 (2012), 740–753.

[WLOG06] WINNEMOLLER, H., OLSEN S. C., GOOCH B.: Real-time video abstraction. *ACM Transactions on Graphics 25*, 3 (2006), 1221–1226.

[WZL*17] WANG D., ZOU C., LI G., GAO C., SU Z., TAN P.: *L*0 gradient-preserving color transfer. *Computer Graphics Forum 36*, 7 (2017), 93–103.

[XT15] XIE S., TU Z.: Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision* (Santiago, Chile, 2015), pp. 1395–1403.

[YKM12] YANG H., KWON Y., MIN K.: A stylized approach for pencil drawing from photographs. *Computer Graphics Forum 31*, 4 (2012), 1471–1480.

[YMI04a] YAMAMOTO S., MAO X., IMAMIYA A.: Colored pencil filter with custom colors. In *Proceedings of 12th Pacific Conference on Computer Graphics and Applications, PG 2004* (Seoul, South Korea, 2004), IEEE, pp. 329–338.

[YMI04b] YAMAMOTO S., MAO X., IMAMIYA A.: Enhanced LIC pencil filter. In *Proceedings of International Conference on Computer Graphics, Imaging and Visualization* (Penang, Malaysia, 2004), pp. 251–256.

[YPC*16] YANG J., PRICE B. L., COHEN S., LEE H., WU Y.: Object contour detection with a fully convolutional encoder-decoder network. In *Proceedings of Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, 2016), pp. 193–202.

[ZL05] ZHOU N. J., LI N. B.: Automatic generation of pencil-sketch like drawings from personal photos. In *IEEE International Conference on Multimedia and Expo* (Amsterdam, The Netherlands, 2005), pp. 1026–1029.

[ZXSY16] ZHANG Z., XING F., SHI X., YANG L.: Semicontour: A semi-supervised learning approach for contour detection. In *Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, 2016), pp. 251–259.