Computational Visual Media
https://doi.org/10.1007/s41095-022-0320-6 Vol. 9, No. 4, December 2023, 807–825

Research Article

Feature-preserving color pencil drawings from photographs

Dong Wang[1], Guiqing Li[2] (✉), Chengying Gao[3], Shengwu Fu[1], and Yun Liang[1]

Abstract Color pencil drawing is well-loved due to its rich expressiveness. This paper proposes an approach for generating feature-preserving color pencil drawings from photographs. To mimic the tonal style of color pencil drawings, which are much lighter and have relatively lower saturation than photographs, we devise a lightness enhancement mapping and a saturation reduction mapping. The lightness mapping is a monotonically decreasing derivative function, which not only increases lightness but also preserves input photograph features. Color saturation is usually related to lightness, so we suppress the saturation dependent on lightness to yield a harmonious tone. Finally, two extremum operators are provided to generate a foreground-aware outline map in which the colors of the generated contours and the foreground object are consistent. Comprehensive experiments show that color pencil drawings generated by our method surpass existing methods in tone capture and feature preservation.

Keywords non-photorealistic rendering; pencil drawings; image editing; feature preservation

1 Introduction

Color pencil drawing, a popular kind art, has diverse styles while preserving almost all natural scene details [1]. It has been widely used in animation, advertisement, film, and television. Unfortunately, making real color pencil drawings is not only time-consuming but also expensive [2], and requires artists to be well trained. Image-based color-pencil-style drawing generation, as an important supplement to manual work, can greatly improve efficiency and reduce production costs. However, it is challenging to generate high-quality results that can both capture the stylistic characteristics of color pencil drawings and preserve the structure and features of the original image.

Color tone computation and object outline detection are two key ingredients for automatically producing color pencil drawings. The former describes visual features such as lightness and saturation, while the latter defines shape features such as contours and feature lines of objects in the scene.

Color tone appears as a harmonic composition of lightness and saturation. Early tone mappings usually established a mathematical relationship between the new tone and the old tone. Because these relationships have no reasonable descriptions, such mappings (Gao et al. [3] and Tong et al. [4]) either fail to capture the pencil drawing style or suffer from structural distortion, as shown in Figs. 1(b) and 1(d). Deep learning models such as Li et al. [5] only inefficiently learn a correct tone mapping between photographs and pencil drawings due to a lack of exact training pairs (see Fig. 1(c)).

Outlines play a key role in highlighting the structural conception of a pencil drawing. Previous pencil drawing approaches employ the gradient operator or DoG (difference of Gaussians) for edge detection, and usually do not consider whether the outline pixels come from the foreground or the background. Outlines (contours) are used to highlight the boundaries of foreground objects in color pencil drawings. So, a good outline for an object should not only describes the shape of the object but also has the same color as the object itself. This becomes a

1 College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China. E-mail: D. Wang, wdngng@163.com; S. Fu, 1225148057@qq.com; Y. Liang, sdliangyun@163.com.

2 School of Computer Science & Engineering, South China University of Technology, Guangzhou, China. E-mail: ligq@scut.edu.cn (✉).

3 School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. E-mail: sgcy@mail.sysu.edu.cn.

Manuscript received: 2022-06-10; accepted: 2022-10-03

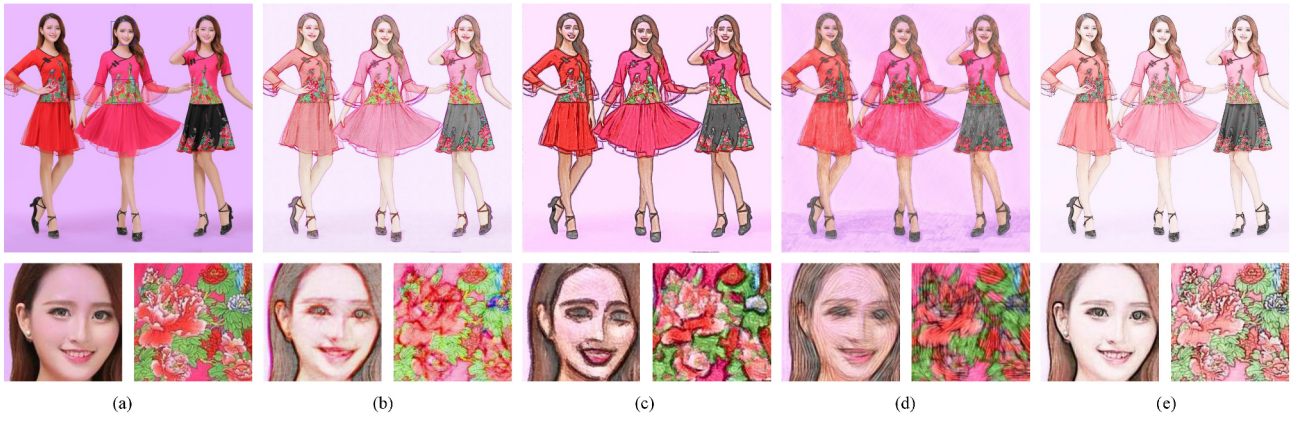

Fig. 1 Comparison of generated color pencil drawings using various methods. (a) Source image, (b) Gao et al., (c) Li et al., (d) Tong et al., and (e) our method.

problem for color pencil drawing generation because the colors of the outline and the corresponding object may be incoherent, as the border between the hair and background in Fig. 1(b). Existing deep learning-based algorithms can not solve the problem due to training data from traditional methods, as illustrated by Fig. 1(c). Unfortunately, this issue has not yet received attention.

To address these issues, we present a feature preserving approach to convert photographs into color pencil drawings. First, following Gao et al. and Lu et al. [6], we also model a pencil drawing style using a tone of high lightness and low saturation, as shown by Fig. 2 for several real pencil drawings. Thus, we design a universal tone mapping consisting of lightness mapping and saturation mapping. Given a pixel in the photograph, the former mapping warps its lightness to a larger value, while the latter mapping explicitly reduces the saturation in terms of its new lightness. In addition, to address color inconsistency of object contours, we use extremum operators to generate a foreground-aware outline map in which the outline pixels are located in the foreground.

In summary, our contributions are an approach for generating pencil drawings from photographs comprising a lightness mapping with monotonically decreasing and bounded derivatives, a lightness dependent saturation mapping, and two gray difference operators for correctly detecting outline pixels from the foreground. The lightness mapping is carefully devised to greatly enhance the total lightness of the drawings and preserve features to some extent, while the saturation mapping decreases the saturation of a pixel in a manner proportional to its increase in lightness, to capture the tone of color pencil drawings. Extreme gray difference operators for generating a foreground-aware outline map further guarantee the coherence of soft object contours and feature lines in color and shape. A user study demonstrates that our algorithm distinctly surpass previous approaches in overall appearance, tone, and outline coherence, as well as structure and feature preservation.

Fig. 2 Real pencil drawings with high lightness and low saturation

2 Related work

Pencil drawing generation from real photos is as a type of image style transfer. The core problem is to model the statistics of the given target style [7]. Pencil drawing style modeling usually considers several key ingredients: tone mapping, outline generation, and texture rendering. Our approach focuses on improving the first two; relevant research is summarised below.

2.1 Tone modeling of pencil drawings

Pencil drawing tone modeling adjusts the tone from photographs to that of real pencil drawings. To achieve a tone of high lightness and low saturation, painters usually mix the current color with a high lightness color (such as white) [8]. The combination increases lightness and simultaneously decreases saturation of the current color due to purity reduction.

Lu et al. decomposed the lightness of a pencil drawing into low, medium, and high layers to match a Gaussian distribution, uniform distribution, and Laplace distribution, respectively. Lightness enhancement in this approach comes at the expense of detail. Gao et al. conducted a local maximum filter [9] on the lightness component of the original photo to generate an image with increased lightness. The new image is then blended with the original photo to generate an image with lower saturation and higher lightness compared with original photo. Unfortunately, such a lightness mapping is neither monotonic nor continuous, and can easily yield artifacts in which color regions with low lightness become gray.

Although the acquisition of effective training data pairs is challenging for artistic style tasks due to the difficulty of establishing an accurate spatial correspondence between photographs and artistic images [10], data-driven methods have also been employed to generate pencil drawing tone. Li et al. [11] designed a so-called ArtPDGAN model based on a generative adversarial network [12] to learn pencil drawing tone from artists. Because of the large gap in content between each image training pair, it is not easy to preserve scene details and object structures. Li et al. proposed a deep learning framework to fuse the structural features of pencil drawings such as hatching or stippling into the input photograph. They created training data from pencil-shaded drawings and a tone map by the guided filter described in Ref. [13], which does not have cues concerning the color tone differences between pencil drawings and photographs.

The pencil drawing tone generated by the aforementioned approaches is far from optimal. In particular, the saturation is unsatisfactory. Ma et al. [14] separated the saturation range into low, medium, and high levels and dealt with each level in isolation to avoid regions with high saturation being oversaturated. Although the new saturation may fall into the valid range, the corresponding new color may go outside the color space: HSI space is constrained to a double cone. Chiang et al. [15] divided HSL space into six regions and used lightness variation to find the region in which a color falls. The slope of the characteristic triangle edge for a specific region was employed to adjust the saturation to ensure validity of the new HSL triple. The idea of the correlation between lightness and saturation motivates us to explicitly establish a reasonable saturation mapping, even though the issue they address is to enhance saturation, while we instead need to decrease the saturation.

2.2 Outline generation

Traditional line drawing generation methods can be divided into two categories: gradient operator-based and DoG-based approaches.

Zhou and Li [16] automatically produced portrait contours in pencil drawing style from portrait photos based on the gradient operator. Son et al. [17] presented a model composed of a contour detector and a contour renderer to obtain a variety of hand-painted outline styles by controlling the details and regions of interest in images. The optimization framework by Bhat et al. [18] unified multiple gradient-based edge detectors to generate line drawings. Lu et al. proposed a gradient-based convolution framework to achieve a piecewise linear effect of pencil outlines. However, the gradient operator may induce double edges for an object feature line, leading to a ghosting effect.

The DoG operator was first introduced in Ref. [19]. Winnemöller et al. [20] utilized an iterative DoG to extract object contours for image abstraction. Kang et al. [21] proposed a flow-based anisotropic DoG filter to extract image outlines. The technique was further adopted by Spicker et al. [22] to generate depth-aware line drawings. Winnemöller et al. [23] introduced an extended DoG filter to generate diverse line drawing effects. A common issue is that DoG-based edge detection algorithms yield many discontinuous edges.

In recent years, various data-driven approaches have been proposed for generating line drawings. Gao et al. devised a conventional convolution network to generate outline maps that were trained by using manually selected ideal pencil outlines from the results of the method by Lu et al., and Jin et al. [24] further introduced a hierarchical network to obtain refined outlines. Li et al. [25] gave an image semantic aware sketch line generation technique, fusing segmented boundaries to boost weak boundaries. In the pencil style outline generation model of Li et al., training pairs were obtained by

extracting outlines from a set of pencil drawings using the extended DoG filter by Winnemöller et al. The training data for these methods were created using traditional algorithms, so inevitably inherit the defects of these algorithms. To obtain important line drawings for accurately depicting overall object structures, Inoue et al. [26] detected expressive edges and created a clean line drawing using a convolutional neural network. This approach also exhibits the problem of generating many discontinuous lines.

As a special case of line drawings, portrait generation has been widely investigated. Chen et al. [27] proposed a line synthesis method for portrait generation, which combines a local model and a global model where the local model converts each element (such as nose and mouth) of the face into lines and the global model locates the element position in the image. The idea is further developed by Yi et al. [28, 29], where global and local structure-driven GANs are adopted to automatically generate portraits. The local GANs transfer the face elements into the artistic style, while the global GAN fuses the elements together. Methods described in Refs. [30–33] also achieved similar results. Nevertheless, as object outlines do not exactly match object edges, these approaches also lead to incompatible color outlines when applied to color pencil drawings.

Although many approaches have been developed for generating line drawings, none addresses the color coherence of the generated outlines. The outlines generated by these methods are usually close to image edges, but it is uncontrollable whether the pixel position of the outline is on the background or the foreground. This results in line color inconsistency with the foreground color, and motivates us to design new operators to constrain line pixels within the foreground area.

3 Approach

3.1 Overview

Given a photograph I, our approach creates its color pencil drawing P in two stages. It first separately conducts a feature-preserving tone mapping to generate a tone map T and a coherent outline mapping to obtain an outline map C from I, and then fuses the two maps into the final result P. The feature-preserving tone mapping is performed in HSL color space from input photograph I to the tone map T. The pipeline of the proposed method is shown in Fig. 3. Let triples $(I_{\mathrm{H}}, I_{\mathrm{S}}, I_{\mathrm{L}})$ and $(T_{\mathrm{H}}, T_{\mathrm{S}}, T_{\mathrm{L}})$ be respective hue, saturation, and lightness components of I and T. In tone mapping, first an optimized lightness mapping on I_{L} generates a high lightness map T_{L^*}, followed by a lightness dependent saturation mapping on I_{S} to get a low saturation map T_{S}. T_{L^*}

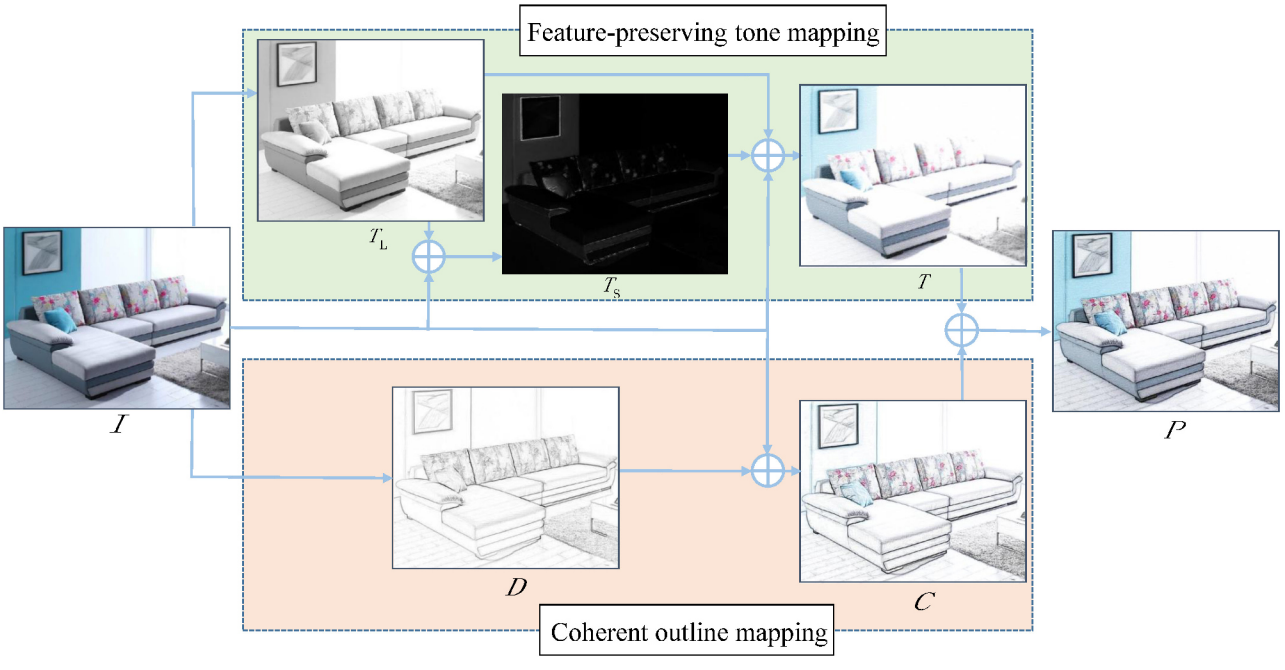

Fig. 3 Pipeline of our proposed method.

is further processed by texture mapping to generate a textured lightness map T_{L}. Components T_{L}, T_{S}, and I_{H} form the tone map T. In outline mapping, an extreme difference operator on the gray map G of I yields an outline map D. It is then used to blend the white image and I to obtain the coherent color outline map C. Fusing T and C finally creates the color pencil drawing P. We next describe the two maps in detail and then briefly address compositing the maps.

3.2 Feature-preserving tone mapping

3.2.1 Basis

Feature-preserving tone mapping generates a tone map T from a photograph I. T has the popular color pencil drawing style of higher lightness and lower saturation compared to I, while preserving the features of I. Tone mapping includes a lightness mapping and a saturation mapping. The former evaluates the lightness of T from I_{L} and the latter calculates the saturation of T from I_{S}. The two mappings are universal in the sense that they are independent of the content of I. Given the HSL triple (h, s, l) of a pixel in I, let (h, s', l') be the corresponding mapped result in T; the hue channel remains unchanged during the mapping.

3.2.2 Lightness mapping with monotonically decreasing derivative

Contrary to the tone mapping of HDR images which compresses a large lightness range into a smaller one with lower lightness, our lightness mapping $f(l) : [0,1] \to [0,1]$ should be subjective and warps low lightness to higher value. Specifically, it is expected to satisfy the following properties:

(1) *Interpolation* $f(0) = 0$ and $f(1) = 1$ guarantee that T has pixels with 0 or 1 lightness values if black or white pixels exist in I.

(2) *Additivity* $f(l) > l$ for $l \in (0,1)$ forces an increase in total lightness after mapping.

(3) *Positivity* $f'(l) > \epsilon$, where $\epsilon > 0$ is small and positive, guarantees preservation of detailed features in I.

We choose the exponential function to formulate this mapping

$$f(l) = (1 - \mathrm{e}^{-l/\sigma})/\gamma, \qquad l \in [0,1] \tag{1}$$

where $\sigma \in (0,1]$ is a parameter to control the speed of growth of $f(l)$, and $\gamma = 1 - \mathrm{e}^{-1/\sigma}$ is a normalized weight to force $f(l)$ to satisfy the interpolation constraint.

Figure 4 depicts four shapes of $f(l)$ defined by Eq. (1) for $\sigma = 0.1, 0.2, 0.5, 0.9$. It shows that smaller σ leads to a larger increment $f(l) - l$. However, if σ is set very small (say 0.1), $f(l)$ grows rapidly when l is small and approximates 1 quickly. This implies that $f(l) \approx 1$ for all l over a large range (say $[0.5, 1]$), which inevitably results in loss of detail. On the other hand, large σ helps preserve image detail well, but fails to provide sufficient lightness increment, particularly when l is small. It is not easy to achieve a good balance with fixed σ.

A good mapping should grow faster for small l in order to quickly get high lightness increments while increasing relatively slowly for large l to guarantee monotonicity and boundedness. This motivates us to increase σ as l grows. We therefore introduce a linear relation $\sigma(l) = al + b$, $0 < a, b \leqslant 1$. Equation (1) is then written as

$$f(l) = (1 - \mathrm{e}^{-l/(al+b)})/\gamma \tag{2}$$

where $\gamma = 1 - \mathrm{e}^{-1/(a+b)}$, and a and b are two unknown parameters.

Observing Fig. 4(b), we find that the total lightness gain:

$$\int_0^1 \big(f(l) - l\big)\,\mathrm{d}l$$

is proportional to the maximum of $f(l) - l$. It is equivalent to solving the constrained optimization problem in Eq. (3):

$$\begin{aligned} &\arg\max_{\{a,b,l\}} f(l) - l \\ &\text{such that} \quad a, b, l \in (0,1], \quad f'(1) = \kappa \end{aligned} \tag{3}$$

where κ is the derivative of $f(l)$ at $l = 1$ and $\kappa > 0$ is given by the user to ensure that function $f(l)$ is monotonic when $l \in [0,1]$. Once κ is given, we then determine a and b in Eq. (2) such that $f(l)$ achieves

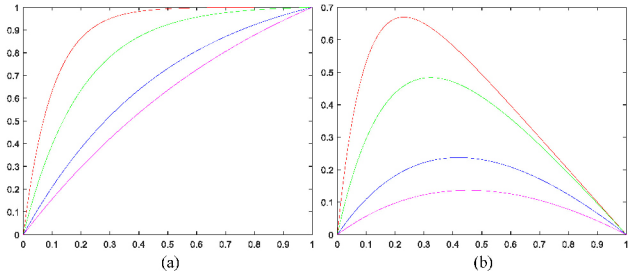

Fig. 4 (a) Lightness mapping $f(l)$, and (b) increment $f(l) - l$, for different values of parameter σ: red, green, blue, and pink curves correspond to $\sigma = 0.1, 0.2, 0.5$, and 0.9 respectively.

the largest lightness gain. This function satisfies the following proposition:

Proposition 1. For $0 < a, b \leqslant 1$ and $0 \leqslant l \leqslant 1$, we always have: (i) $f'(l) > 0$, (ii) $f''(l) < 0$, and (iii) $f'(l) \geqslant \kappa$.

Proposition 1 implies that $f(l)$ in Eq. (2) satisfies the aforementioned third property if we set $\kappa > 0$. For image edges caused by lightness differences, this condition can guarantee that edges in the source image do not vanish in the color pencil drawing.

Figure 5 illustrates two optimal solutions of Eq. (3) for $f(l)$ which respectively correspond to $\kappa = 0.1$ (red) and $\kappa = 0.2$ (green). We also depict the shape of $f(l)$ in Eq. (1) for $\sigma = 0.2$ (blue). It is easy to show that the gain for the red curve is greater than for the blue one. Moreover, the derivative of the blue curve almost vanishes near $l = 1$ while that of the red one does not. This implies that Eq. (3) is superior to Eq. (1) both in total lightness gain and detail-preserving effect.

3.2.3 Lightness dependent saturation mapping

In HSL color space, lightness and saturation are two highly correlated quantities. For images, lightness and saturation of almost all pixels are inversely correlated: pixels with high lightness usually have low saturation, and vice versa. This motivates us to establish a saturation mapping dependent on lightness. Moreover, the mapping should not change original image features; the relation between original lightness and original saturation should be preserved.

Let the general form of the saturation mapping be $s' = g(s, l, l')$ where $l' = f(l)$ is defined in Eq. (2). It should obey the following three constraints at least:

(1) Pixel value (h, s', l') must fall into the valid range of HSL space.

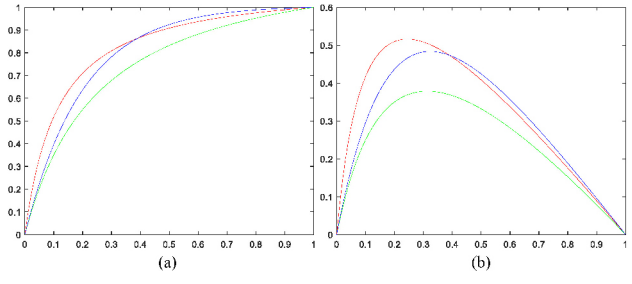

Fig. 5 (a) Lightness mapping $f(l)$, and (b) increment $f(l) - l$, for different parameter settings: red, green curves correspond to $\kappa = 0.1$ and 0.2 respectively. The blue curve corresponding to $\sigma = 0.2$ increases more slowly for small l and has smaller derivative for large l than the curve for $\kappa = 0.1$.

(2) Very approximately, the new saturation s' should be inversely proportional to the new lightness l'.

(3) $0 \leqslant s' \leqslant s$ to meet the low saturation characteristics of color pencil drawings.

To simplify the mapping model, we assume that the relationship between s' and s is linear:

$$s' = g(s, l, l') = \lambda(l, l')s \tag{4}$$

where $\lambda(l, l')$ is a proportionality coefficient related to original lightness and new lightness.

According to constraint (1), $s' = 0$ when $l' = 1$, so $g(s, l, 1) = 0$, which implies $\lambda(l, 1) = 0$. This, combined with constraint (2), indicates that λ contains the factor $1 - l'$. Accordingly, the factor $1 - l$ is also introduced to produce a symmetrical relationship. In addition, constraint (3) suggests $0 \leqslant \lambda(l, l') < 1$. This inspires us to define

$$\lambda(l, l') = \frac{1 - l'}{1 - l} \tag{5}$$

which leads to the final saturation mapping:

$$s' = g(s, l, l') = \frac{1 - l'}{1 - l}s \tag{6}$$

Note that we take $\frac{0}{0} = 1$. With s and l fixed, the greater the new lightness l' is, the smaller the generated saturation s' becomes.

Equation (6) actually reveals a geometric relationship in HSL space: the old right triangle with right sides s and $1 - l$ is similar to the new right triangle with right sides s' and $1 - l'$. This guarantees the validity of (h, s', l'), in that the triple can be smoothly converted to RGB space without truncation. Moreover, the similarity preserves original image features.

3.2.4 Tone map composition

Lightness and saturation mappings act on lightness and saturation components of each pixel in input photograph I respectively to produce a tone map T with higher lightness and lower saturation. The mapping between the HSL triple $(I_{\mathrm{H}}, I_{\mathrm{S}}, I_{\mathrm{L}})$ in I and its counterpart $(T_{\mathrm{H}}, T_{\mathrm{S}}, T_{\mathrm{L}})$ in T is established via Eq. (7):

$$\begin{cases} T_{\mathrm{H}} &= I_{\mathrm{H}} \\ T_{\mathrm{L}^*} &= f(I_{\mathrm{L}}) \\ T_{\mathrm{S}} &= g(I_{\mathrm{S}}, I_{\mathrm{H}}, T_{\mathrm{L}}) \end{cases} \tag{7}$$

where T_{L^*} is further processed using the method by Lu et al. to generate a textured lightness component T_{L}. Figure 6 illustrates some immediate component maps during the generation of tone map T. Compared to the original component maps, the new saturation

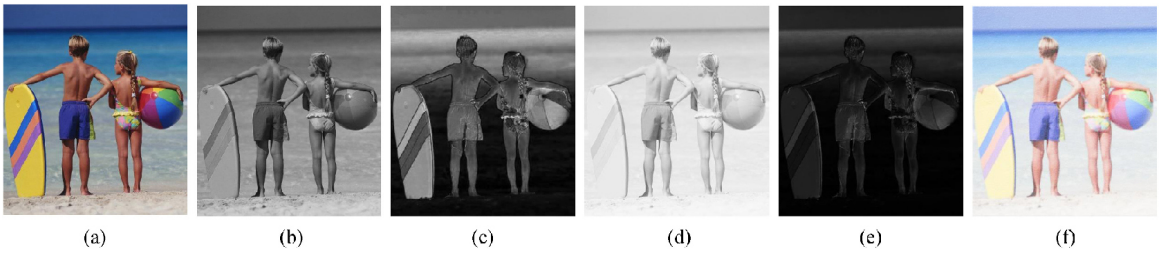

Fig. 6 Tone map composition. (a) Original image I, (b) original lightness component I_{L}, (c) original saturation component I_{S}, (d) increased lightness component T_{L}, (e) decreased saturation component T_{S}, and (f) generated tone map T.

is obviously reduced while the new lightness is considerably augmented.

3.3 Coherent outline map

3.3.1 Basis

Enhanced outlines are a key ingredient of pencil drawings. Outlines include object contours and feature lines inside an object. If an outline includes pixels from different objects or the background, it will exhibit appearance artifacts due to their color difference from those of the corresponding foreground object. In pencil drawing, such artifacts are further magnified because of edge enhancement.

To address the issue, we propose a coherent contour generation approach based on the gray map G of photograph I, which is inspired by the following observations:

(1) Compared to the difference between adjacent pixels, the difference between the current pixel and its local extreme can better express an image edge.

(2) Forcing contour pixels to lie on the foreground object may help address coherence of the object contour.

(3) For a feature line inside an object, coherence can be enhanced by selecting pixels with smaller gray values in their local regions.

The approach includes two extreme operators: maximal gray difference operator and minimal gray difference operator.

3.3.2 Maximal gray difference operator

Let p be an arbitrary pixel in G. We use $N(p)$ to denote its local neighborhood (for example, a 5×5 patch) and evaluate the difference between the maximal gray level in $N(p)$ and the gray value of p:

$$D_{\mathrm{M}}(p) = \max\{G(q), q \in N(p)\} - G(p) \qquad (8)$$

We call D_{M} the maximal gray difference operator. It is easy to show that $D_{\mathrm{M}}(p) \in [0, 1]$. Feature line pixels have smaller gray values in their local neighborhoods, so have larger $D_{\mathrm{M}}(p)$ values and receive maximum reinforcement. For object contour pixels within the region of an object, if their gray levels are smaller in their local neighborhood, like feature line pixels, they also will have maximum reinforcement. Thus, the generated feature lines are compatible with the ground truth and the positions of generated object contours are within the region of the corresponding foreground object. This is desirable for color pencil drawings.

However, if the object gray value is greater than that of the background gray value, the maximal gray difference operator will strengthen the object contour pixels on the side of the background region. The generated outline will be incompatible with the object in color tone. A minimal gray difference operator is a better substitute to deal with this case.

3.3.3 Minimal gray difference operator

The minimal difference operator D_{m} is defined as Eq. (9):

$$D_{\mathrm{m}}(p) = G(p) - \min\{G(q), q \in N(p)\} \qquad (9)$$

Also, $D_{\mathrm{m}}(p) \in [0, 1]$ holds.

A good object outline should not only describe the shape of the object but also have the same color as the object itself. However, two feature lines are extracted for each original feature line, in Fig. 7(b), by the gradient operator, and in Fig. 7(c), by the minimal gray difference operator. The partial or whole contour color of the flower in Fig. 7(b) and in Fig. 7(d), using maximal gray difference operator, is black, which is inconsistent with the flower's color. The ideal result combines the contour in Fig. 7(c) with the feature lines in Fig. 7(d), as shown in Fig. 7(e).

3.3.4 Combined gray difference operator

In most cases, the object gray value is lower than or almost the same as the background gray value. D_{M}

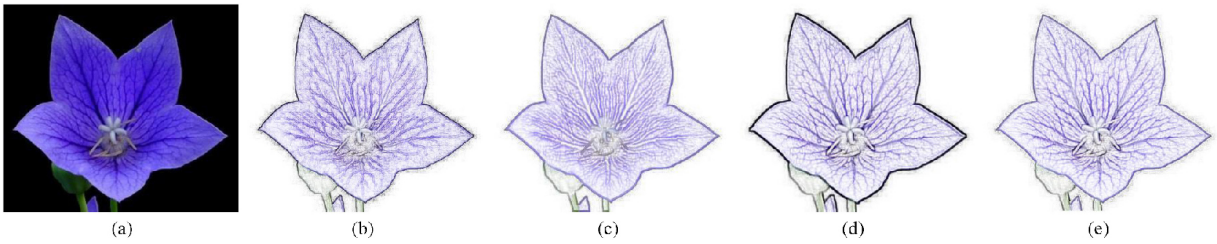

Fig. 7 Contour extraction. (a) Original image, (b) gradient operator, (c) minimum gray difference operator, (d) maximum gray difference operator, and (e) operator combining minimum and maximum gray differences.

is a good choice for selecting the foreground contour pixels as outlines. Unfortunately, in some cases, the foreground gray value is much larger than that of the background, as shown in Fig. 8. D_{M} is inclined to select the background contour pixels as outlines. In this case, D_{m} can be used to select foreground pixels as outlines. As the pixel gray value of feature lines is always smaller in their local regions, D_{M} is appropriate.

Therefore, we need a mask map ρ to indicate which of the two operators to select. $D_{\mathrm{M}}(p)$ is activated for pixel p if $\rho(p) = 1$ while $D_{\mathrm{m}}(p)$ is chosen if $\rho(p) = 0$. By introducing ρ to combine operators D_{M} and D_{m}, the gray outline generation can be written as

$$D(p) = \rho(p)D_{\mathrm{M}}(p) + (1 - \rho(p))D_{\mathrm{m}}(p) \qquad (10)$$

$\rho(p) = 1$ for each pixel p by default. When the foreground gray value is much larger than the background gray, a map ρ like Fig. 8(b) is required. There are many methods to get ρ. In this paper, we regard the salient regions in the input image as the foreground and adopt the approach in Ref. [34] to calculate a saliency map. We then utilize the color transfer method in Ref. [35] to change the foreground color greatly. Comparing the original image and color transferred image, pixels with a large gradient variation lie at the boundary between foreground and background, so can help to generate ρ.

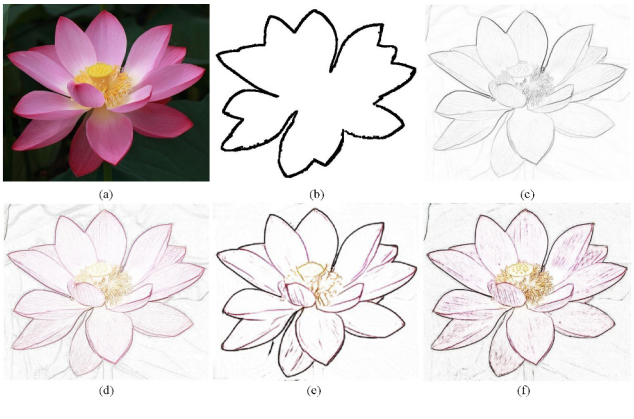

Fig. 8 Outline map combining the maximum and minimum gray operators. (a) Original image, (b) ρ mask, (c) inverse of D, (d) corresponding C, (e) C generated by Chan et al., and (f) C generated by sketchKeras.

3.3.5 Coherent color outline map

We assume color pencil drawings are made on white paper. Let I^{w} be a white image with the same resolution as I, i.e., $I^{\mathrm{w}}(p) = (1, 1, 1)$ for all p. We then evaluate the pencil drawing color outline map C pixel by pixel:

$$C(p) = (1 - D(p))I^{\mathrm{w}}(p) + D(p)I(p) \qquad (11)$$

This equation indicates that $C(p) = I(p)$ for $D(p) = 1$, the case when p lies on a strong edge, and $C(p) = I^{\mathrm{w}}(p)$ for $D(p) = 0$, the case when p lies within trivial regions, and $C(p)$ blends the two images for $0 < D(p) < 1$, the case when p is located in a weak feature region.

Figure 8 presents an example of the generated outline map D (Fig. 8(c)) and color outline map C (Fig. 8(d)) using the combined gray difference operator. The maximal gray difference operator provides feature lines and the minimal gray difference operator provides the flower's contour. The combined gray difference operator combines the advantages of the maximal and minimal gray difference operators. We also compare our outlines with those provided by Chan et al. [36] and sketchKeras [37]. The contour color in their result is generated from the background, which is inconsistent with the real colors of the lotus. Clearly, our contour is located on the side of the foreground, and possesses the color of the flower.

3.4 Fusion of outline map and tone map

To make a color pencil drawing, artists usually first sketch object outlines and then add scene detail. Our color pencil drawing P for I is similarly calculated by fusing the color outline map C and the textured tone map T in RGB color space as Eq. (12):

$$P = C^{\gamma}T \qquad (12)$$

where γ plays the role of enhancing the image outline and is set to 2 in our experiment. Figure 9 illustrates a color pencil drawing generated by our approach. The tone map and outline map are naturally combined to produce a vivid color pencil drawing.

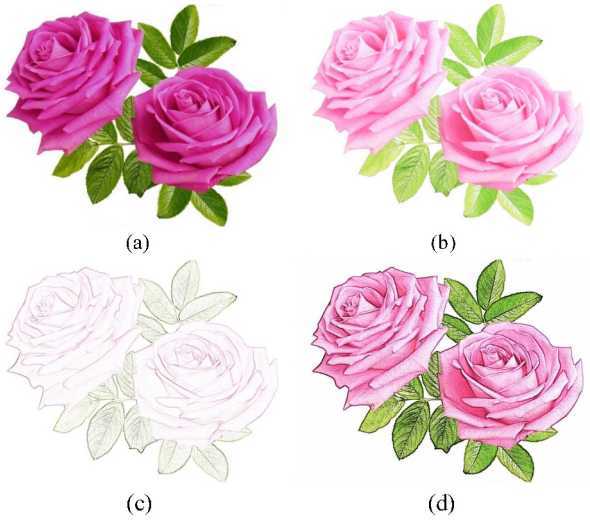

Fig. 9 Color pencil drawing and its components. (a) Original image I, (b) textured tone map T, (c) color outline map C, and (d) generated color pencil drawing P.

4 Experimental results and discussions

This section demonstrates the effectiveness of our framework by comparing it with state-of-the-art methods in terms of overall appearance, tone mapping quality, and outline coherence of the output.

4.1 Overall comparison

4.1.1 Visual comparison

We evaluated our overall success by comparing our work with the four most representative methods by Lu et al., Gao et al., Li et al., and Tong et al. Results for these methods were produced using the authors' source code or directly downloaded from the authors' websites.

Figures 10 and 11 present four examples of generated color pencil drawings and grayscale pencil drawings using each method in turn. The grayscale pencil drawings were obtained by converting color pencil drawing images to grayscale images. Compared with other methods, our results have fewer defects and finer detail. At the same time, the generated style is perceptually closer to real pencil drawings for both color and grayscale output. Further discussions of generated tone and outlines will be provided in Sections 4.2 and 4.3.

We also compared our method with the zero-shot style transfer method of Sheng et al. [38]; see Fig. 12. Their results were generated using a reference image provided by a real color pencil drawing; see Fig. 12(d). It is easy to see that their results (Fig. 12(c)) exhibit blurring of detail and shape distortion artifacts. In comparison, our results (Fig. 12(b)) look closer in style to the color pencil drawings of Fig. 12(d).

4.1.2 User study

Even when applied to pairs of real and generated images, quantitative metrics still lack consistency with human perception [39]. A user study was thus adopted for qualitative human assessment of the generated color pencil drawings. We collected 30 photographs, mostly from the aforementioned four papers. Each approach generated a color pencil drawing for each photograph. We then designed a questionnaire to evaluate each result in terms of the following four considerations:

(1) Similarity to real pencil drawings in overall visual effect.
(2) Quality of outline structures and colors: significant feature lines or contours are extracted with corrected shape and their colors are consistent with the colors of corresponding foreground objects.
(3) Harmony of overall tone: local tone is smooth and detail-preserving. The whole tone mimics the style of pencil drawing well.
(4) Preservation of image features. The content of the color pencil drawing follows the original image.

For each consideration, satisfactoriness of results is indicated by increasing discrete scores from 1 to 5.

We divided the 30 photographs into 6 groups with 5 test examples per group and then invited non-art major users to appraise five groups and fine arts users to evaluate one group. Before commencing evaluation, participants were given a group of high-quality real pencil drawings as reference. The user study was conducted on Questionnaire Star (`https://www.wjx.cn/`), a Chinese public platform. In total, 78 valid ratings were collected from non-art major users, and 13 valid ratings were collected from fine arts users. Figure 13 summarizes the results in pairs of bars, from fine arts users (left bar) and non-art major users (right bar). These histograms indicate our method

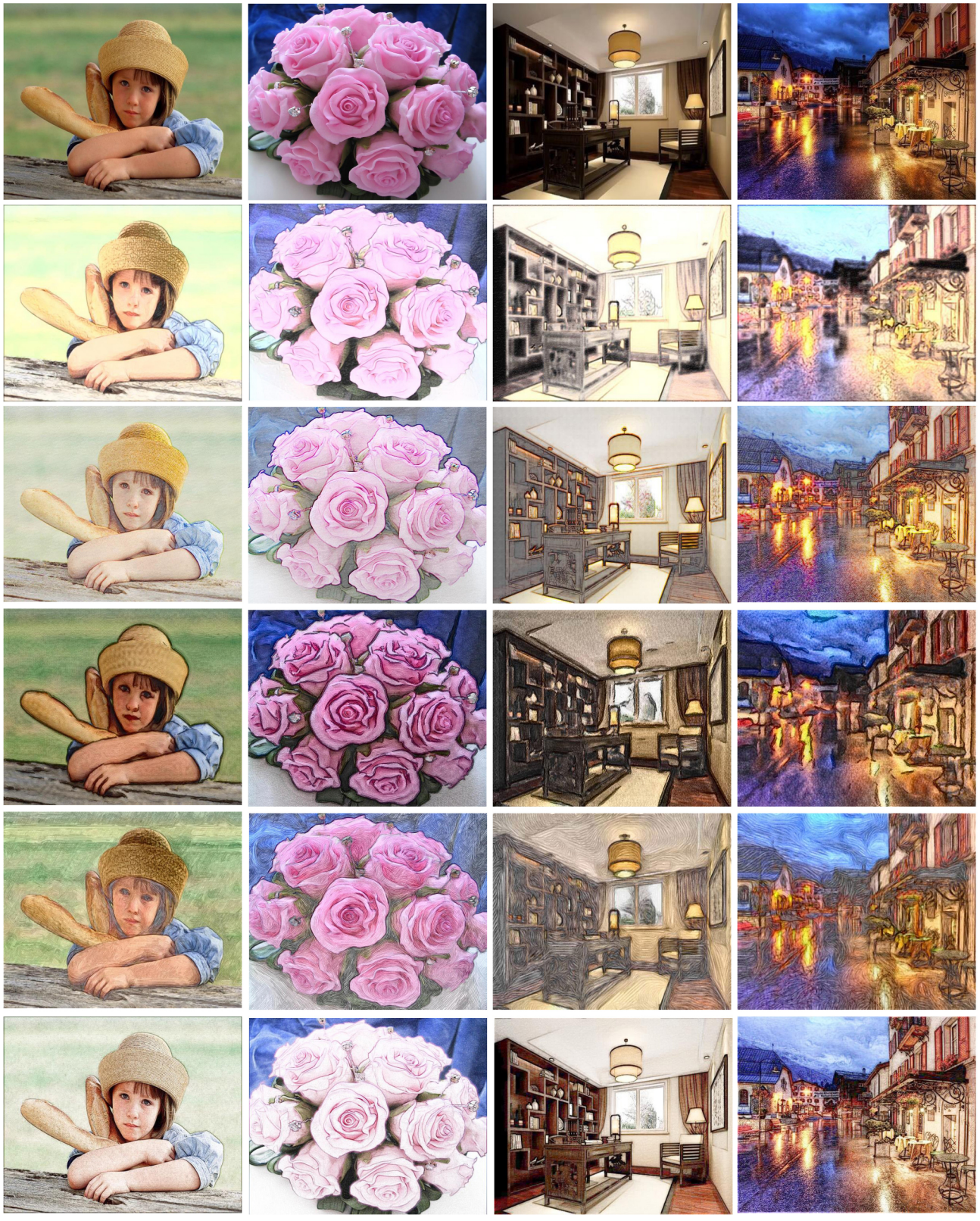

Fig. 10 Four example color pencil drawings using different approaches. Top to bottom: source image, outputs from Lu et al., Gao et al., Li et al., Tong et al., and our method.

Fig. 11 Four example gray pencil drawings using different approaches. Top to bottom: source image, outputs from Lu et al., Gao et al., Li et al., Tong et al., and our method.

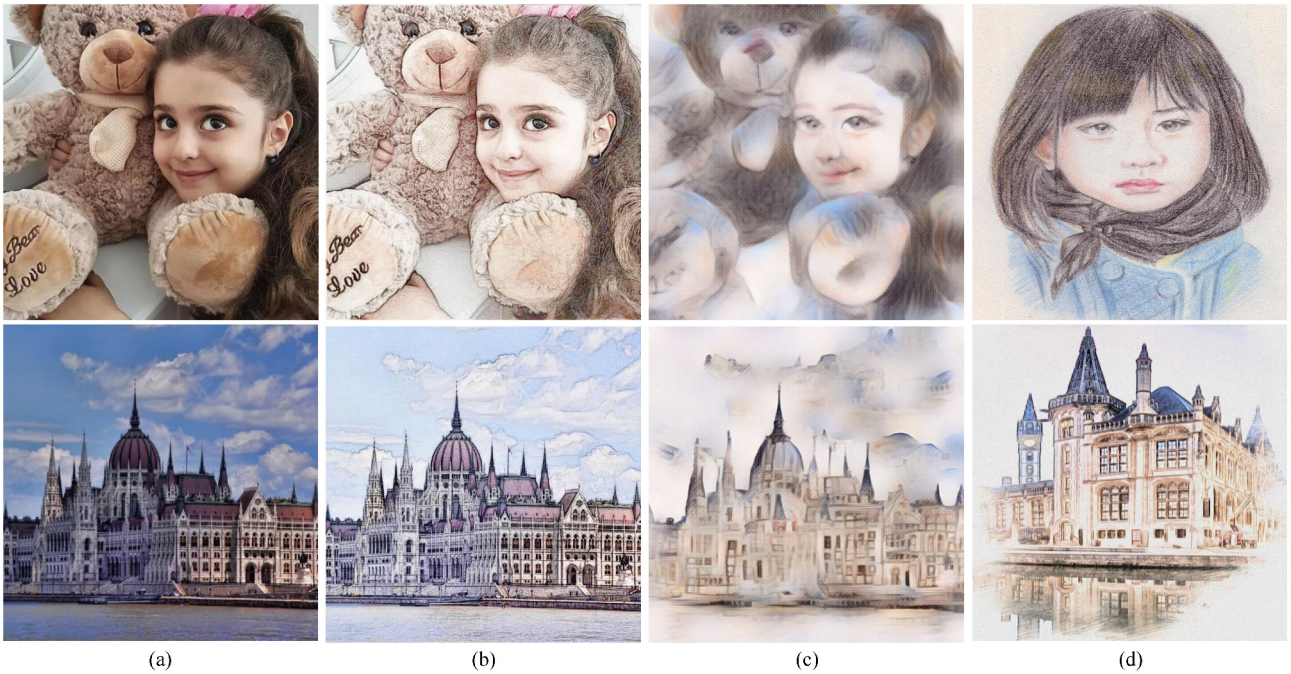

Fig. 12 Comparison to the style transfer method. (a) Source images, (b) our results, (c) Sheng et al., and (d) real color pencil drawings.

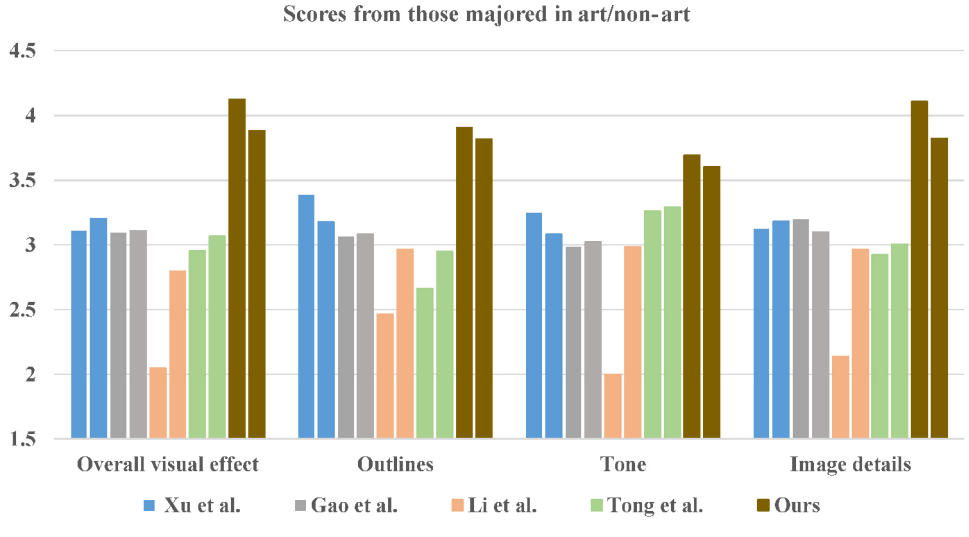

Fig. 13 Statistical analysis of results. Four representative approaches were compared to our proposed method, in terms of overall visual effect, outlines, tone, and image details. Bars (the higher, the better) show average score for a factor under consideration; colors indicate approach.

to be significantly preferred to previous approaches for all considerations.

4.1.3 Speed

We tested the efficiency of different approaches by running examples with different resolution images on a PC with an i5-8600K CPU. As the code for generating the outlines is unavailable for the approach by Gao et al., we used the algorithm by Lu et al. to generate outlines for the results of the method of Gao et al. (in fact, Gao et al. stated that their outline network is based on and slower than the approach of Lu et al.). Table 1 demonstrates that the speed of our method is comparable with those of Gao et al. and Lu et al. Our method for generating the tone mapping and outline mapping consumes less time than the approach by Gao et al. and almost the same as the approach by Lu et al. The end-to-end network of Li et al. is the fastest, while the approach by Tong et al. takes the longest time due to generation of a stroke sequence.

Table 1 Running time (in second)

	Resolution	800×600	1024×768	1280×960	1440×1080	1600×1200
Xu et al.	Outline	0.13	0.21	0.30	0.40	0.47
	Tone	0.01	0.02	0.02	0.04	0.02
	Total	**0.64**	**1.02**	**1.57**	**2.03**	**2.50**
Gao et al.	Outline	0.13	0.21	0.30	0.40	0.47
	Tone	0.28	0.45	0.67	0.86	1.02
	Total	**0.70**	**1.09**	**1.59**	**2.06**	**2.47**
Ours	Outline	0.04	0.07	0.10	0.14	0.16
	Tone	0.10	0.15	0.23	0.30	0.38
	Total	**1.85**	**2.65**	**3.99**	**4.99**	**5.98**
	Auto mask	7.28	11.12	18.31	23.46	29.15
Li et al.	Total	**0.02**	**0.03**	**0.04**	**0.04**	**0.05**
Tong et al.	Total	**120.05**	**217.65**	**455.91**	**665.28**	**818.37**

4.2 Tone mapping evaluation

To further show the advantage of our tone mapping, we conducted an analysis of tone generation by comparing our method with the approaches of Lu et al., Gao et al., and Li et al. Figure 14 compares tone maps for the four approaches using two examples. We did not consider the approach of Tong et al. because it focuses on texture generation for pencil drawings instead of tone computation.

The tone editing techniques of Lu et al. and Li et al. work on monochrome images. Lu et al. model the tone of high lightness pixels using a Laplacian distribution, which inevitably results in excessive lightness and missing details. See the floor in the first image and people in the second image in Fig. 14(b) for example. The deep learning model of Li et al. for image shading or tone mapping fails to learn the differences in lightness and saturation between a photograph and a real pencil drawing because the training data pairs only come from real pencil drawings. Figure 14(d) shows that the tone maps remain similar to those of the corresponding original photographs.

Although tone editing by Gao et al. works on multiple color components, high lightness and low saturation are generated by blending a high lightness

(a) (b) (c) (d) (e)

Fig. 14 Tone comparison. (a) Source images, (b) tone mappings by Lu et al., (c) tone mappings by Gao et al., (d) tone mappings by Li et al., and (e) ours.

image. The tone maps of Gao et al. lack color, especially in low lightness areas. See the table in the first image and the dark clothes in the second image in Fig. 14(c). Figure 14(e) shows that the tone of our results not only has the characteristics of high lightness and low saturation, but also preserves details and color better.

The overall tone of these images can be further analyzed by statistics. Figure 15 shows the mean lightness (above) and mean saturation (below) of the images in Fig. 14. The five vertical bars labelled c1–c5 in Fig. 15(a) are respectively the mean lightness and mean saturation components of the five images in Fig. 14(above). Similarly, the five vertical bars in Fig. 15(b) correspond to the five images in Fig. 14(below). Apart from the approach of Li et al., which yields results with mean lightness close to the original values, other methods lead to higher lightness and lower saturation than the original values. While the approaches of Lu et al. and Gao et al. yield results with higher mean lightness, they come at a cost of loss of detail. This is caused by greatly increasing high brightnesses for Lu et al. (Fig. 14(b)), but by greatly increasing the low brightnesses for Gao et al. (Fig. 14(c)). In comparison, our approach constrains rate of change of lightness by limiting the derivatives of the two endpoints of the mapping curve. Our results look better.

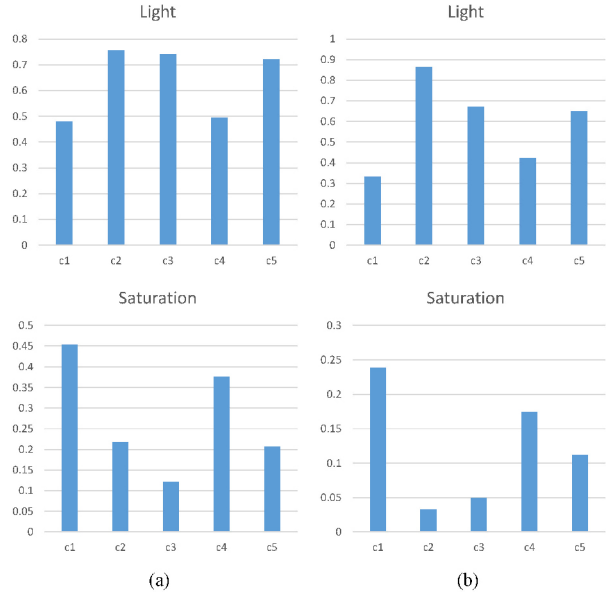

Fig. 15 Comparisons of light and saturation components. Above: mean lightness. Below: mean saturation. (a, b) refer to the images in Fig. 14(above) and Fig. 14(below) respectively.

Finally, we show the robustness of our tone mapping scheme by using images with extreme lightness distributions. Our tone mapping improves all lightness values and reduces all saturation values except 0 and 1. This means, both for high and low lightness regions, the lightness component is always enhanced and the saturation component becomes smaller to meet the tone characteristic of color pencil drawings. In Fig. 16, we present three examples; the first includes both bright and dark regions, the second is almost globally bright and the third one is dark. The results in Fig. 16(c) look satisfactory, especially for low brightness areas with a larger increase in lightness and a larger decrease in saturation.

4.3 Outline coherence evaluation

To show the outlining qualities of our operator, we compare our generated outlines with those of Gao et al. and Li et al. The training data for outline generation by Gao et al. were collected using a gradient operator. Outline pixels may be uncontrollably located either in the background or the foreground. If they belong to the background, the colors of outlines and corresponding objects will not match: see the treetop outline in Fig. 17(b). The training data for outline generation model by Li et al. are provided by extended DoG filtering, which may produce discontinuous edges, as in the feature lines of the corn and the wall in Fig. 17(e). Moreover, the

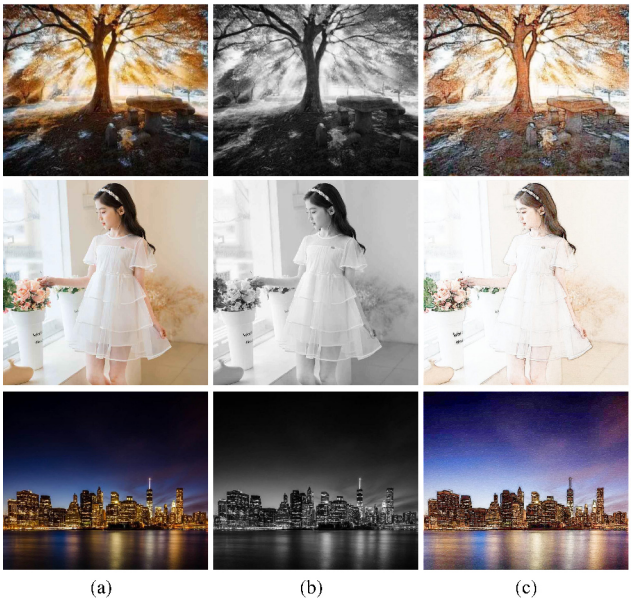

Fig. 16 Tone mappings of our approach for extreme lighting. (a) Source images, (b) lightness components, and (c) our results.

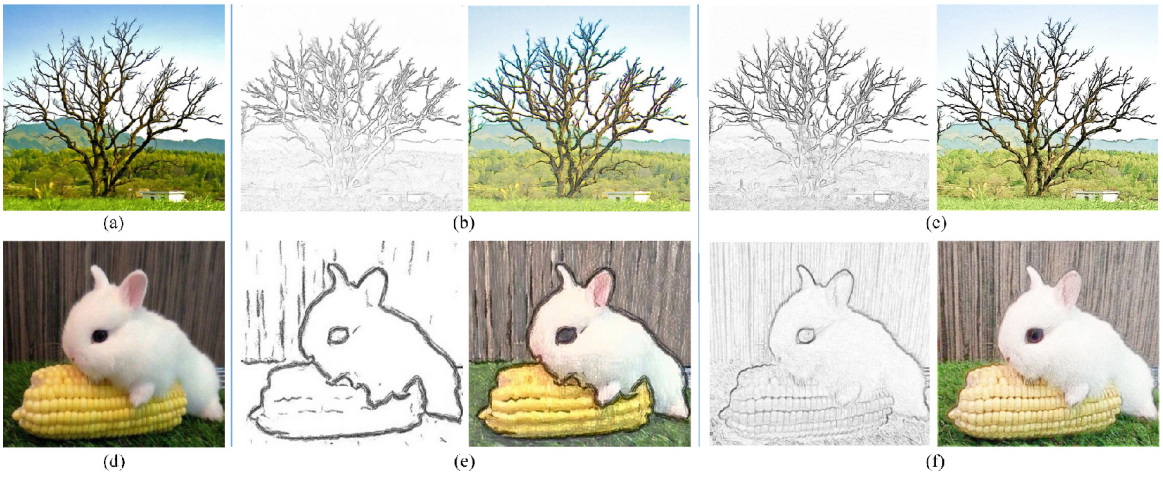

Fig. 17 Outline maps for approaches by Gao et al. and Li et al., and for our method. (a, d) Source images, (b, e) outline maps (left) and color pencil drawings (right) by Gao et al. and Li et al., and (c, f) outline maps (left) and color pencil drawings (right) for our method.

contour colors also differ from those of foreground objects (see the contour colors of the animal and the corn). In contrast, as our outline extraction technique controls outline positions to be in the foreground, its results present more realistic pencil outlines in both shape and color, as can be seen in Figs. 17(c) and 17(f).

Based on low-level features of pixel values, our outline extraction scheme is sensitive to noise. For noise-free images, the gray values usually smoothly change within non-edge regions. However, the differences between gray values of a pixel and its local extrema significantly increase for noisy images. This leads to many non-edge pixels being recognized as outline pixels. Figure 18(b) shows many discontinuities on object contours in the outline map of the noisy image, which was generated from the noise-free image in Fig. 18(a) by adding Gaussian white noise of mean 0 and variance 0.01.

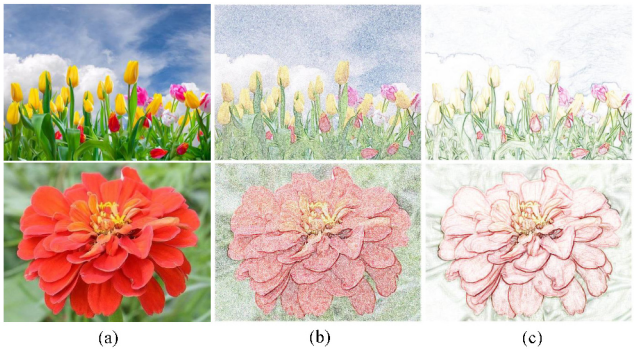

Fig. 18 Outline generation for noisy images. (a) Original images, (b) outline maps for noisy images, and (c) outline maps for noise-free images.

4.4 Parameters

In our method, κ in Eq. (3), the slope at the endpoint of the lightness mapping curve, controls the brightness adjustment and plays an important role in preserving the features of the original image and balancing the total lightness of the generated pencil drawing. A smaller κ forces $f(x)$ to grow faster and consequently leads to larger lightness. Examples with different κ are depicted in Fig. 19. It is easy to observe that as κ increases, the total brightness gradually decreases while saturation and features are gradually strengthened. In our experiments, all results were generated by setting $\kappa = 0.15$, which makes a good trade-off between feature preservation of the original image, and the style of high lightness and low saturation of color pencil drawing.

5 Conclusions

We have presented a new image style transfer framework for creating color pencil drawings from photographs. Lightness increase mapping with a monotonically decreasing derivative and lightness-dependent saturation mapping were presented to generate a specific color pencil drawing tonality in which lightness is usually much higher and saturation is lower than in photographs. Furthermore, two extremal filters were developed for generating a coherent outline map of the given photograph. A variety of experiments demonstrated that our results are closer to real color pencil drawings than existing solutions.

(a) (b) (c) (d) (e)

Fig. 19 Impact of the slope κ on the final color pencil drawings. (a) Source image, (b) $\kappa = 0.01$, (c) $\kappa = 0.05$, (d) $\kappa = 0.15$, and (e) $\kappa = 0.3$. As κ increases, total lightness decreases and saturation grows.

Theoretical analysis has verified that the values in the lightness map, saturation map, and outline map generated by our model naturally fall in the expected bounded range of $[0, 1]$. Moreover, while the algorithm involves a few parameters, only fixed values are required for the whole framework. Experiments also showed its robustness.

Because of the difficulty of acquiring effective pairs of photographs and corresponding ground truth pencil drawings, evaluation of generated pencil drawings is still very subjective. Quantitative methods are necessary to provide scientific evaluation criteria. It is also desirable to explore the essential characteristics of pencil drawings and design more competitive texture generation methods to produce highly aesthetic color pencil drawings. In addition, generating the outline map involves a mask which labels a set of pixels on the boundaries separating the foreground objects and background regions. When multiple overlapping objects appear in the original image, our saliency-based approach may fail to obtain correct contour colors along boundaries between objects. Figure 20 shows an example with five people in which the third person is in front of the second and the fourth persons (from left to right). Similarly, the second and fourth persons are in front of the first and fifth people. Our approach cannot determine these complex relationships and readily yields incorrect contour colors as shown in Fig. 20(b): the color of the contour pixels of arms and hands come from their related backgrounds. Image analysis may be of use in generating a multi-level mask in future work.

Acknowledgements

We thank the reviewers for their valuable comments and constructive suggestions. This work was supported in parts by GD Natural Science Foundation (2021A1515012301, 2022A1515011425), and the Key Research and Development Project of Guangzhou (202206010091, SL2022B03J01235).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

(a) (b)

Fig. 20 Limitations. (a) Source image, and (b) generated color outline map. Some inappropriate colour is assigned to contours of a foreground object.

References

[1] Lewis, D. *Pencil Drawing Techniques*. Watson-Guptill Publications, 1984.

[2] Jing, Y. C.; Yang, Y. Z.; Feng, Z. L.; Ye, J. W.; Yu, Y. Z.; Song, M. L. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics* Vol. 26, No. 11, 3365–3385, 2020.

[3] Gao, C. Y.; Tang, M. Y.; Liang, X. G.; Su, Z.; Zou, C. Q. PencilArt: A chromatic penciling style generation framework. *Computer Graphics Forum* Vol. 37, No. 6, 395–409, 2018.

[4] Tong, Z.; Chen, X.; Ni, B.; Wang, X. Sketch generation with drawing process guided by vector flow and grayscale. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 609–616, 2021.

[5] Li, Y.; Fang, C.; Hertzmann, A.; Shechtman, E.; Yang, M. Im2Pencil: Controllable pencil illustration from photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 1525–1534, 2019.

[6] Lu, C.; Xu, L.; Jia, J. Combining sketch and tone for pencil drawing production. In: Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering, 65–73, 2012.

[7] Chen, D. D.; Yuan, L.; Liao, J.; Yu, N. H.; Hua, G. StyleBank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2770–2779, 2017.

[8] Véliz, Z. Francisco Pacheco's comments on painting in oil. *Studies in Conservation* Vol. 27, No. 2, 49–57, 1982.

[9] Kim, G.; Woo, Y.; Yim, C. Color pencil filter for non-photorealistic rendering applications. In: Proceedings of the 18th IEEE International Symposium on Consumer Electronics, 1–2, 2014.

[10] Cole, F.; Golovinskiy, A.; Limpaecher, A.; Barros, H. S.; Finkelstein, A.; Funkhouser, T.; Rusinkiewicz, S. Where do people draw lines? In: Proceedings of the ACM SIGGRAPH 2008 Papers, Article No. 88, 2008.

[11] Li, S.; Li, K.; Kacher, I.; Taira, Y.; Yanatori, B.; Sato, I. ArtPDGAN: Creating artistic pencil drawing with key map using generative adversarial networks. In: *Computational Science – ICCS 2020. Lecture Notes in Computer Science, Vol. 12143*. Springer Cham, 285–298, 2020.

[12] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Wardefarley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2672–2680, 2014.

[13] He, K. M.; Sun, J.; Tang, X. O. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 6, 1397–1409, 2013.

[14] Ma, S. P.; Ma, H. Q.; Xu, Y. L.; Li, S. A.; Lv, C.; Zhu, M. M. A low-light sensor image enhancement algorithm based on HSI color model. *Sensors* Vol. 18, No. 10, 3583, 2018.

[15] Chiang, J.; Hsia, C.; Peng, H.; Lien, C. Color image enhancement with saturation adjustment method. *Journal of Applied Science and Engineering* Vol. 17, No. 4, 341–352, 2014.

[16] Zhou, J.; Li, B. X. Automatic generation of pencil-sketch like drawings from personal photos. In: Proceedings of the IEEE International Conference on Multimedia and Expo, 1026–1029, 2005.

[17] Son, M.; Kang, H.; Lee, Y. J.; Lee, S. Abstract line drawings from 2D images. In: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, 333–342, 2007.

[18] Bhat, P.; Zitnick, C.; Cohen, M.; Curless, B. GradientShop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics*, Vol. 29, No. 2, Article No. 10, 2010.

[19] Marr, D.; Hildreth, E. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences* Vol. 207, No. 1167, 187–217, 1980.

[20] Winnemöller, H.; Olsen, S. C.; Gooch, B. Real-time video abstraction. *ACM Transactions on Graphics* Vol. 25, No. 3, 1221–1226, 2006.

[21] Kang, H.; Lee, S.; Chui, C. K. Coherent line drawing. In: Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering, 43–50, 2007.

[22] Spicker, M.; Kratt, J.; Arellano, D.; Deussen, O. Depth-aware coherent line drawings. In: Proceedings of the SIGGRAPH Asia 2015 Technical Briefs, Article No. 1, 2015.

[23] Winnemöller, H.; Kyprianidis, J. E.; Olsen, S. C. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics* Vol. 36, No. 6, 740–753, 2012.

[24] Jin, Y. X.; Li, P.; Sheng, B.; Nie, Y. W.; Kim, J.; Wu, E. H. SRNPD: Spatial rendering network for pencil drawing stylization. *Computer Animation and Virtual Worlds* Vol. 30, Nos. 3–4, e1890, 2019.

[25] Li, T.; Xie, J. Y.; Niu, H. L.; Hao, S. J. Enhancing pencil drawing patterns via using semantic information. *Multimedia Tools and Applications* Vol. 81, No. 24, 34245–34262, 2022.

[26] Inoue, N.; Ito, D.; Xu, N.; Yang, J.; Price, B.;

Yamasaki, T. Learning to trace: Expressive line drawing generation from photographs. *Computer Graphics Forum* Vol. 38, No. 7, 69–80, 2019.

[27] Chen, H.; Liu, Z. Q.; Rose, C.; Xu, Y. Q.; Shum, H. Y.; Salesin, D. Example-based composite sketching of human portraits. In: Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering, 95–153, 2004.

[28] Yi, R.; Liu, Y. J.; Lai, Y. K.; Rosin, P. L. APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10735–10744, 2019.

[29] Yi, R.; Xia, M. F.; Liu, Y. J.; Lai, Y. K.; Rosin, P. L. Line drawings for face portraits from photos using global and local structure based GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 10, 3462–3475, 2021.

[30] Liao, J.; Yao, Y.; Yuan, L.; Hua, G.; Kang, S. B. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 120, 2017.

[31] Isola, P.; Zhu, J. Y.; Zhou, T. H.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5967–5976, 2017.

[32] Zhu, J. Y.; Park, T.; Isola, P.; Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2242–2251, 2017.

[33] Huang, Z. Y.; Peng, Y. C.; Hibino, T.; Zhao, C. Q.; Xie, H. R.; Fukusato, T.; Miyata, K. DualFace: Two-stage drawing guidance for freehand portrait sketching. *Computational Visual Media* Vol. 8, No. 1, 63–77, 2022.

[34] Zhou, L.; Yang, Z. H.; Yuan, Q.; Zhou, Z. T.; Hu, D. W. Salient region detection via integrating diffusion-based compactness and local contrast. *IEEE Transactions on Image Processing* Vol. 24, No. 11, 3308–3320, 2015.

[35] Wang, D.; Zou, C. Q.; Li, G. Q.; Gao, C. Y.; Su, Z.; Tan, P. $\mathcal{L}0$ gradient-preserving color transfer. *Computer Graphics Forum* Vol. 36, No. 7, 93–103, 2017.

[36] Chan, C.; Durand, F.; Isola, P. Learning to generate line drawings that convey geometry and semantics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7905–7915, 2022.

[37] sketchKeras. Available at https://github.com/lllyasviel/sketchKeras.

[38] Sheng, L.; Lin, Z. Y.; Shao, J.; Wang, X. G. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8242–8250, 2018.

[39] Xue, Y.; Guo, Y. C.; Zhang, H.; Xu, T.; Zhang, S. H.; Huang, X. L. Deep image synthesis from intuitive user input: A review and perspectives. *Computational Visual Media* Vol. 8, No. 1, 3–31, 2022.

Dong Wang is an associate professor in the Department of Computer Science and Engineering in South China Agricultural University, Guangzhou, China. She received her Ph.D. degree in the Department of Computer Science in City University of Hong Kong in 2012. From 2016 to 2017, she worked in Simon Fraser University. Her research interests include computer vision, image computation, and computer graphics.

Guiqing Li is a professor in the School of Computer Science and Engineering, South China University of Technology (SCUT). Before joining SCUT, he worked as a postdoctoral researcher in the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University. From 2002 to 2008, he visited City University of Hong Kong several times as a research associate and research fellow. His research interests include dynamic geometry processing, image and video editing, and digital geometry processing.

Chengying Gao is an associate professor in the School of Computer Science and Engineering, Sun Yat-sen University. She received her Ph.D. degree in computer science from the School of Information Science and Technology, Sun Yat-sen University in 2003. Her research interests include computer graphics and image processing.

Shengwu Fu is a master student in the Department of Computer Science and Engineering in South China Agricultural University. He majored in computer science and technology and received his undergraduate degree from South China Agricultural University in 2021. His research interests include computer vision and machine learning.

Yun Liang is a professor in the Department of Computer Science and Engineering in South China Agriculture University. She received her M.Sc. and Ph.D. degrees from the School of Information Science and Technology, Sun Yat-sen University in 2005 and 2011, respectively. From 2016 to 2017, she worked in Simon Fraser University. Her research interests include computer vision, image computation, and machine learning.