Machine Vision and Applications (2018) 29:503–512
https://doi.org/10.1007/s00138-018-0906-2

ORIGINAL PAPER

Image-based pencil drawing synthesized using convolutional neural network feature maps

Xiuxia Cai[1] · Bin Song[1]

Received: 8 November 2016 / Revised: 13 November 2017 / Accepted: 3 January 2018 / Published online: 27 January 2018

Abstract
In most cases, the conventional pencil-drawing-synthesized methods were in terms of geometry and stroke, or only used classic edge detection method to extract image edge characters. In this paper, we propose a new method to produce pencil drawing from natural image. The synthesized result can not only generate pencil sketch drawing, but also can save the color tone of natural image and the drawing style is flexible. The sketch and style are learned from the edge of original natural image and one pencil image exemplar of artist's work. They are accomplished through using the convolutional neural network feature maps of a natural image and an exemplar pencil drawing style image. Large-scale bound-constrained optimization (L-BFGS) is applied to synthesize the new pencil sketch whose style is similar to the exemplar pencil sketch. We evaluate the proposed method by applying it to different kinds of images and textures. Experimental results demonstrate that our method is better than conventional method in clarity and color tone. Besides, our method is also flexible in drawing style.

Keywords Deep learning · Pencil sketch drawing · Feature maps · CNN

1 Introduction

Pencil drawing is a kind of visual arts for human beings to perceive natural scene. Pencil sketch drawings are a very popular form of art. Sketch and hatching are two styles of pencil drawing. Among them, sketch is used by artists to depict the global shape or contours. Besides, hatching is used to depict tone or shading. Recently, automatic method for generating pencil drawings image directly from photos becomes popular. The automatic method needs no 3D model and no requirement for human interaction, which is appealing in various applications targeting.

There has already existed a lot of works on sketching [1–4] and hatching [5,6]. In recent years, paper Lu et al. [7] has made a great improvement in pencil-drawing-synthesized method. However, the conventional pencil-drawing-synthesized methods are almost in terms of geometry and stroke, or only use classic edge detection method to extract image edge characters which is difficult to produce satisfactory result. It is because the existence of texture, noise, and illumination variation makes the extracting and manipulating structures difficult. Besides, how to well depict hatching style of artist's work is a challenge to most of the researches. Gatys et al. [8] transfer style of artist work to photo picture, which makes learning the style of artist work become possible. Cai et al. [9] proposed using convolutional feature maps to combine inconsistent textures. Convolutional neural network is a good tool for extracting features for the texture. Inspired by this work, we use convolutional neural network (CNN) as a tool to extract hatching feature from artist's work. In this paper, we propose a new framework for pencil drawing. We compute gradients on the grayscale version of the input, yielding magnitude and draw line which will obtain an initial sketch. Then, we learn the hatching style from one artist's work that is represented by the statistical of CNN feature maps. Finally, we use large-scale bound-constrained optimization (L-BFGS) method to optimize the initial sketching and hatching features distance functions and obtain the ultimate pencil drawing image. The distance functions between initial sketch and hatching features are calculated by the Gram matrix of feature maps. In the tone drawing, we can save the colors of the original natural image.

✉ Bin Song
bsong@mail.xidian.edu.cn

Xiuxia Cai
caixiuxia@stu.xidian.edu.cn

[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

In view of the above description, the work described in this paper focuses on the following points. First, we propose a new framework for automatically generating pencil drawing image; the loss function is calculated using the statistical information of the feature maps of CNN. Then, L-BFGS optimization is used to get the final drawing. Additionally, the color tones of the original image can be preserved by our method.

The remainder of this paper is organized as follows. Section 2 reviews related work and symmetry group theory. In Sect. 3, we first present the framework of our method of image-based pencil drawing. Section 3.1 shows the way we generate edge image for sketching. In Sect. 3.2, we present the details of representation features for hatching based on the work of artist. Section 3.3 explains how we can generate pencil drawing image. Section 3.4 shows the way we save the color of original image. Section 4 provides several experiment results in our method. Section 5 is the conclusion of this paper.

2 Previous work

2.1 Image-based sketching

Most of pencil-drawing-synthesized works are about portrait sketch in comparison with natural image sketching [10]. Compared with the natural scenery sketch, portrait sketch has more regularity characteristics in the process of analysis. Xu et al. [11] advocated an L0 smoothing filter which is applicable to pencil stroke generation. Based on these previous works in our framework, we compute gradients on the grayscale version of the input, yielding magnitude and draw line which will obtain an initial sketch.

2.2 Convolutional neural network

In recent years, there are numerous interesting research results based on the convolutional neural network [12–18]. One of interesting research is Gatys et al. [19], which generates texture using the convolutional neural network (CNN) features at different layers, by matching the CNN features of an input texture. Another way of generating images is Simonyan [20,21], maximizing the responses of CNN units. Inspired by the prosperity of texture synthesis based on CNN features, we consider inpainting image on the basis of convolutional neural network. Moreover, in light of the similarities between performance of convolutional neural networks and biological vision [22–25], our work takes the feature maps in forward processes of convolutional neural network as the important information about how humans understand and perceive image content. Papers Cimpoi et al. [26,27] have provided a fruitful new analysis tool for studying visual perception with CNN. Based on the work of Cimpoi et al., the feature representation in this paper is represented with CNN. The VGG network was extensively used and introduced in recent object recognition research works which are based on convolutional neural network. Considering the well design of VGG, we also use VGG-16 (VGG-19 is suitable in this paper. Besides, we are just randomly choosing one of them) network in our work.

Inspired by VGG-16 network's architecture, our convolutional neural network computations are mainly based on linearly rectified convolution and average pooling. The convolution filters are in the size of $3 \times 3 \times k$ where k is the number of input feature maps. The size of pooling windows is 2×2 in non-overlapping regions, and the convolutional layer is followed by a average pooling layer.

2.3 Large-scale bound-constrained optimization

Large-scale bound-constrained optimization which is a limited memory algorithm for solving large nonlinear optimization problems subjects to simple bounds on the variables developed from Quasi-Newton method. Before using Quasi-Newton method, there were gradient descent method, Newton method, and conjugated gradient method. All the methods mentioned before are designed to solve the optimization problem. In addition, these methods are through calculating Hessian matrix or gradient descent to find the optimized result. In Quasi-Newton method, the $f(x)$ to be optimized is

$$\begin{aligned} f(x) = {} & f(x_{i+1}) + (x - x_{i+1})^{\mathrm{T}} \nabla f(x_{i+1}) \\ & + \frac{1}{2}(x - x_{i+1})^{\mathrm{T}} H_{i+1}(x - x_{i+1}) \\ & + o(x - x_{i+1}), \end{aligned} \tag{1}$$

where x_{i+1} is one value in domain, and H_{i+1} is the Hessian matrix in x_{i+1}. The (1) formula derivation and ignoring high small order item we can get

$$\nabla f(x) \approx \nabla f(x_{i+1}) + H_{i+1}(x - x_{i+1}).$$

Let $x = x_i$, then

$$H_{i+1}^{-1}(\nabla f(x_{i+1}) - \nabla f(x_i)) \approx (x_{i+1} - x_i).$$

If we let $B_{i+1} = H_{i+1}^{-1}$, then we need to calculate the B_i for the optimized result. Let $t_i = \nabla f(x_{i+1}) - \nabla f(x_i)$ and $s_i = x_{i+1} - x_i$; we save the nearest number of m t_i and s_i. So in L-BFGS has

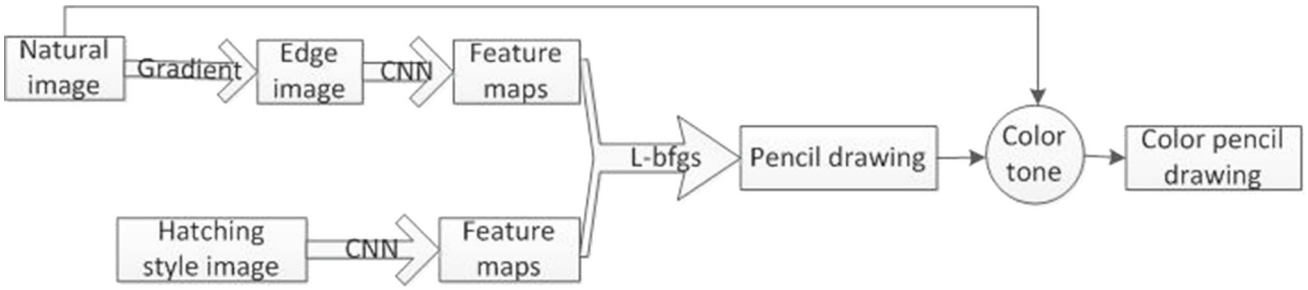

Fig. 1 Proposed framework of pencil drawing image synthesized using CNN features

$$
\begin{aligned}
B_i &= V_{i-1}^{\mathrm{T}} B_{i-1} V_{i-1} + r_{i-1} s_{i-1} s_{i-1}^{\mathrm{T}} \\
&\vdots \\
&= (V_{i-1}^{\mathrm{T}} \cdots V_{i-m}^{\mathrm{T}}) B_i^0 (V_{i-m} \cdots V_{i-1}) \\
&+ r_{i-m} (V_{i-1}^{\mathrm{T}} \cdots V_{i-m+1}^{\mathrm{T}}) s_{i-m} s_{i-m}^{\mathrm{T}} (V_{i-m+1} \cdots V_{i-1}) \\
&+ \cdots + r_{i-1} s_{i-1} s_{i-1}^{\mathrm{T}},
\end{aligned} \tag{2}
$$

where $r_i = \frac{s_{i-1}^{\mathrm{T}} t_{i-1}}{t_{i-1}^{\mathrm{T}} t_{i-1}}$, $V_i = I - r_(i-1) t_{i-1} s_{i-1}$. We set the initial value $B_i^0 = r_i I$. For the earlier methods, when the problem is large, not only the calculation is large, but also the demand for storage space is large. The basic idea of the L-BFGS algorithm is that the algorithm only preserves and uses the information of the recent m iterations to construct the approximate matrix of the Hessian matrix. This approach can greatly reduce the amount of computing and storage space. So in this paper, we will use Eq. (2) to calculate the new generated sketching image.

3 Our method

The proposed framework is shown in Fig. 1. We first compute several directions gradients on the grayscale version of original natural image, which will generate an edge image. This step of process will be introduced in Sect. 3.1. At the same time, we compute the Gram matrix of feature maps from one artist's work. Then, L-BFGS method [28–30] is used to synthesize the pencil drawing. These two steps will be introduced in Sects. 3.2 and 3.3. If we want to get a color pencil drawing, we keep the original nature image tone in the ultimate result. The details of this step will be introduced in Sect. 3.4.

3.1 Obtaining sketch

To calculate the edge of the natural image, the gradient is calculated for natural image in our framework. Set I be the gray image of the input natural image. ∂_x and ∂_y are gradient operators in directions x and y. Then, the magnitude of the image is

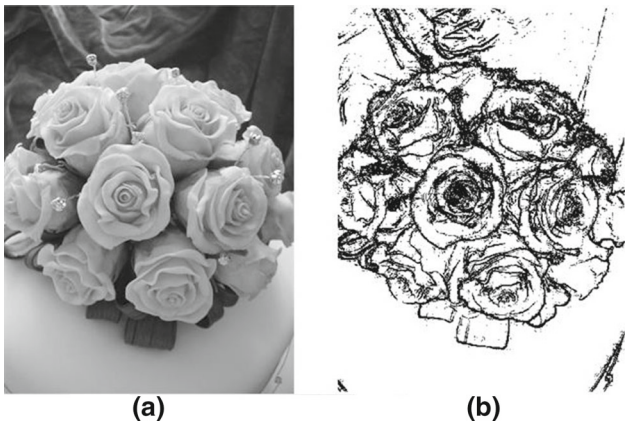

Fig. 2 Our method of getting edge image. **a** input image, **b** edge image

$$G(p) = ((\partial_x I(p))^2 + (\partial_y I(p))^2)^{1/2}, \tag{3}$$

where p is the p-th pixel of the image. In order to get a robust edge image, we calculate several gradients corresponding to different directions. Suppose there are K magnitudes $G_i(p)$, where i is the i-th direction, $G_i(p)$ equals the $G(p)$ which calculated on the rotated image. The rotation angle is $180 * i/K$. Here, we set K to be 8. Then, our edge response in pixel p is

$$E(p) = \sum_{i=1}^{K} G_i(p). \tag{4}$$

Our one example result of edge image is shown in Fig. 2.

3.2 Obtaining hatching style

To learn the hatching style of the sample artwork, we first extract the feature maps of the artwork using caffe. This will be the input for the convolutional neural network. Based on the VGG network [20,21], we use alternating convolution

layers and polling layers, the convolution layers starting with the size of the artwork to analyze and (due to the polling) each successive convolution layer half the size of the previous one. We use the feature space provided by the convolutional layers (layers of Conv1-1, Conv2-1, Conv3-1, Conv4-1, and Conv5-1). The fully connected layers are not used in our method. The caffe model and trained network can be publicly available and be explored in the convolutional architecture for fast feature embedding caffe framework [31].

To characterize the artist's work in our model, the artist's work is transferred to a vector $\mathbf{x}$. Then, $\mathbf{x}$ is passed through the convolutional neural forward network and we compute the activations for each layer l in the network. If a layer has N_l distinct filters, then there will be N_l feature maps in one layer. We set the size of feature map to be S_l which is the product of height and width of the feature map. The activation of the ith filter at position j in layer l is f_{ij}^l. Therefore, the response in a layer l is a matrix $F^l \in R^{N_l \times M_l}$, and the element is f_{ij}^l.

When we synthesize the pencil drawing image, on the one hand, we want to partially preserve the contour of the natural image; on the other hand, we synthesize hatching style based on artist's work style information. So, the edge image extracted from natural image is treated as the content control information. Besides, the feature maps extracted from artist's work are treated as the style control information. Correspondingly, two kinds of loss functions are calculated. One is the feature map-based loss function, which aims to let the initial image be similar with the edge image extracted from natural image in contour. Another one is the Gram-matrix-based loss function, which aims to let the initial image be similar with the artist's work in style.

Suppose that we have one initial image which is randomly initialized and one edge image containing the contour we want to generate. We set $\mathbf{c}$ to be the edge image which can control the contour for the pencil drawing image, and set $\mathbf{t}$ to be the initial image (pencil drawing image) that is to be optimized. F_c and F_t are their respective feature maps representation in layer l. Then, the squared-error loss between the two feature map representations in layer l can be defined in the following equation:

$$E_{c,l}(\mathbf{c}, \mathbf{t}) = \frac{1}{2} \sum_{i,j} (F_{cij}^l - F_{tij}^l)^2. \tag{5}$$

The feature map-based total loss is

$$\ell_c(\mathbf{c}, \mathbf{t}) = \sum_{l=0}^{L} w_{c,l} E_{c,l}, \tag{6}$$

where $w_{c,l}$ are weighting factors of the contribution of each layer to the feature map-based total loss.

On top of the CNN responses, in each feature map of the network a style representation is built, which computes the spatial correlations. The correlations reflect the image information of the feature map to some extent. These feature correlations are given by the Gram matrix $G^{sl} \in R^{N_l \times M_l}$, where G_{ij}^{sl} is the inner product between the vectorized feature map i and j in layer l:

$$G_{ij}^{sl} = \sum_k F_{ki}{}^l F_{kj}{}^l, \tag{7}$$

where K represents the feature map which filtered by the kth filter. To compute the correlations between the different filter responses, the feature correlations are given by the $G^{fl} \in R^{N_l \times M_l}$ either, where G_{ij}^{fl} is the inner product between the vectorized feature map i and j in layer l:

$$G_{ij}^{fl} = \sum_k F_{ik}{}^l F_{jk}{}^l, \tag{8}$$

where K is the kth feature map in one group maps which filtered by the same filter. We can find that:

$$G_{ij}^{sl} = (G_{ij}^{fl})^{\mathrm{T}}. \tag{9}$$

G_{ij}^{sl} is the transpose of G_{ij}^{fl}. So if we set $G^l = G^{sl} = (G^{fl})^{\mathrm{T}}$, the G^l not only represents the spatial correlation information, but also represents the correlation information between the different filter responses.

To generate pencil drawing image that is similar to the artist's work in style, we use gradient descent from one initial image to find another image which matches the style representation of the artist's work. This is conducted by minimizing the mean-squared distance between the entries of the Gram matrix [32–34]. The Gram matrix represents the statistical information of the feature maps. So the synthesized image is statistically similar to the art image on the basis of Gram-matrix-based loss function. The correlation between two features is calculated by the inner product. In our paper, Gram matrix is the inner product between feature maps in each layer. The Gram matrix is the statistic information that is not only the correlation in spatial, but also the correlation of filtered values from different filters in the identical location. Therefore, let $\mathbf{s}$ and $\mathbf{t}$ be the exemplar artist's work image and the pencil drawing image to be generated, and set $S^l = S^{sl} = (S^{fl})^{\mathrm{T}}$ and $T^l = T^{sl} = (T^{fl})^{\mathrm{T}}$ are their respective Gram matrixes in layer l. The contribution of that layer to the loss is then

$$\begin{aligned} E_{s,l} &= \frac{1}{8{N_l}^2{M^2}_l} \sum_{i,j} \left(\left(S_{ij}^{sl} - T_{ij}^{sl}\right)^2 + \left(S_{ij}^{fl} - T_{ij}^{fl}\right)^2 \right) \\ &= \frac{1}{8{N_l}^2{M^2}_l} \sum_{i,j} \left(\left(S_{ij}^{sl} - T_{ij}^{sl}\right)^2 + \left(\left(S_{ij}^{sl}\right)^{\mathrm{T}} - \left(T_{ij}^{sl}\right)^{\mathrm{T}} \right)^2 \right) \end{aligned}$$

(a) (b)

Fig. 3 Style extracted from the exemplar, then be synthesized. In **b**, we only synthesize the style without contour information. **a** An exemplar of artist's work, **b** style extracted from the exemplar

$$= \frac{1}{8N_l{}^2M^2{}_l}\sum_{i,j}\left(\left(S^l_{ij}-T^l_{ij}\right)^2+\left(\left(S^l_{ij}\right)^{\mathrm{T}}-\left(T^l_{ij}\right)^{\mathrm{T}}\right)^2\right)$$
$$= \frac{1}{4N_l{}^2M^2{}_l}\sum_{i,j}\left(S^l_{ij}-T^l_{ij}\right)^2. \tag{10}$$

The Gram-matrix-based total loss is

$$\ell_s(\mathbf{s},\mathbf{t}) = \sum_{l=0}^{L} w_{s,l}E_{s,l}, \tag{11}$$

where $w_{s,l}$ are weighting factors of the contribution of each layer to the style that control total loss. One example of only learning the style of an exemplar artist's work is shown in Fig. 3.

where $\alpha = 0.0001$, $\beta = 1$, c is edge image, s is the artist's work, and t is the pencil drawing image we want to generate.

From Eq. (12), we can find that the object function is weighting sum of two finite distortion function. The boundaries are in the control of edge image and artist's work image. So, finding the best $\mathbf{t}$ for $f(\mathbf{t})$ is a convex optimization problem. Then, our target is to find $\mathbf{t}$ to minimize $f(\mathbf{t})$, which is

$$\min\{f(\mathbf{t})\} = \min\left\{\frac{\alpha}{\alpha+\beta}\ell_c(\mathbf{c},\mathbf{t})+\frac{\beta}{\alpha+\beta}\ell_s(\mathbf{s},\mathbf{t})\right\}. \tag{13}$$

We perform gradient descent on $f(\mathbf{t})$ to find the Hessian matrix satisfied the minimization. The gradient of $f(\mathbf{t})$ is

$$\frac{\partial f}{\partial \mathbf{t}} = \frac{\alpha}{\alpha+\beta}\frac{\partial \ell_c}{\partial \mathbf{t}}+\frac{\beta}{\alpha+\beta}\frac{\partial \ell_s}{\partial \mathbf{t}}$$
$$= \frac{\alpha}{\alpha+\beta}\frac{\partial \ell_c}{\partial F^l_{cij}}+\frac{\beta}{\alpha+\beta}\frac{\partial \ell_s}{\partial S^l_{ij}}. \tag{14}$$

From Eq. (5), the derivative of content control total loss $E_{c,l}$ with respect to the activations in layer l equals

$$\frac{\partial E_{c,l}}{\partial F^l_{cij}} = \begin{cases}(F^l_c - F^l_t)_{ij}, & \text{if } F^l_{cij} > 0\\ 0, & \text{if } F^l_{cij} \le 0\end{cases}. \tag{15}$$

and from Eq. (10), the derivative of Gram-matrix-based total loss $E_{s,l}$ regarding the activations in layer l equals

$$\frac{\partial E_{s,l}}{\partial S^l_{ij}} = \begin{cases}\frac{1}{N_l{}^2M^2{}_l}\left((S^l)^{\mathrm{T}}(S^l-T^l)\right)_{ij}, & \text{if } S^l_{ij} > 0\\ 0, & \text{if } S^l_{ij} \le 0\end{cases}. \tag{16}$$

Then, Eq. (14) becomes

$$\frac{\partial f}{\partial \mathbf{t}} = \alpha\sum_{l=0}^{L} w_{c,l}\frac{\partial E_{c,l}}{\partial F^l_{cij}}+\beta\sum_{l=0}^{L} w_{s,l}\frac{\partial E_{s,l}}{\partial S^l_{ij}} = \begin{cases}\alpha\sum_{l=0}^{L} w_{c,l}((F^l_c-F^l_t)_{ij}) & \\ +\beta\sum_{l=0}^{L} w_{s,l}\frac{1}{N_l{}^2M^2{}_l}((S^l)^{\mathrm{T}}(S^l-T^l))_{ij} & \text{if } F^l_{cij}>0,\ S^l_{ij}>0\\ \alpha((F^l_c-F^l_t)_{ij}), & \text{if } F^l_{cij}>0,\ S^l_{ij}\le 0\\ \beta(\frac{1}{N_l{}^2M^2{}_l}((S^l)^{\mathrm{T}}(S^l-T^l))_{ij}), & \text{if } F^l_{cij}\le 0,\ S^l_{ij}>0\\ 0, & \text{if } F^l_{cij}\le 0,\ S^l_{ij}\le 0\end{cases}. \tag{17}$$

3.3 Pencil drawing image generation

The core idea of generating pencil drawing image is that the initial image is synthesized similar in contour with edge image and similar in style with artist's work image. Based on this concept, we define the similar distance as:

$$f(\mathbf{t}) = \frac{\alpha}{\alpha+\beta}\ell_c(\mathbf{c},\mathbf{t})+\frac{\beta}{\alpha+\beta}\ell_s(\mathbf{s},\mathbf{t}), \tag{12}$$

In order to use L-BFGS to optimize $f(\mathbf{t})$, we save the 20 (this value is usually between 3 and 20) times iterative values of $f(\mathbf{t})$ and $\frac{\partial f}{\partial \mathbf{t}}$. The maximum number of iterations is 1000. The inverse Hessian matrix $B_i = H_i^{-1}$ has the iterative relation

$$B_{i+1} = V_i^{\mathrm{T}}B_iV_i + r_iS_iS_i^{\mathrm{T}}, \tag{18}$$

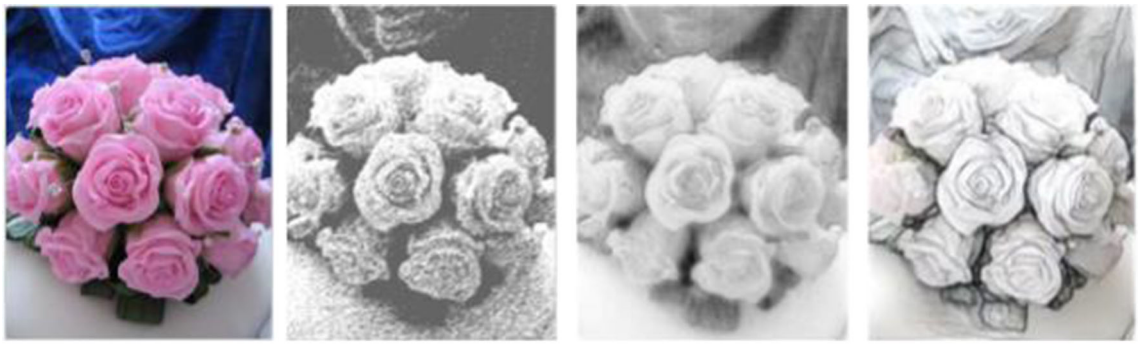

Fig. 4 Comparison with commercial software and method of Lu et al. **a** input, **b** Photoshop CS5, **c** Lu et al., **d** our method

where

$$V_i = I - r_i \left(\frac{\partial f}{\partial \mathbf{t}_{i+1}} - \frac{\partial f}{\partial \mathbf{t}_i} \right) S_i^{\mathrm{T}},$$

where $S_i = f(\mathbf{t}_{i+1}) - f(\mathbf{t}_i)$, and $r_i = \frac{1}{(\frac{\partial f}{\partial \mathbf{t}_{i+1}} - \frac{\partial f}{\partial \mathbf{t}_i})^{\mathrm{T}} S_i}$ is the learn rate. According to the equation we described before, we send the total loss function $f(\mathbf{t})$ and gradient value $\frac{\partial f}{\partial \mathbf{t}}$ to the L-BFGS model to calculate the best value t to satisfy Eq. (13).

c is edge image extracted from original natural image and **s** is the exemplar of artist's work in Eq. (12) when we synthesize the pencil drawing image. $\mathbf{t_1}$ is random initialization in the size of the **c**. The corresponding total loss function $f(\mathbf{t_1})$ and gradient value $\frac{\partial f_1}{\partial \mathbf{t_1}}$ are used to calculate B_1 in Eq. (18). According to Eq. (2) in L-BFGS, we calculate the $\mathbf{t_2}$. Then, $\mathbf{t_2}$ is used to calculate B_2. This iterative process will continue until it reaches the maximum steps of the iteration or the value of B_i is nearly unchanged. The last value of $\mathbf{t_i}$ is the ultimate result we needed. We transfer the $\mathbf{t_i}$ to an image which is the pencil drawing image. Our method results are shown in Sect. 4.

3.4 Color tone generated

In Sect. 3.3, we can generate a pencil drawing image without color tone. In this section, we will save the color tone of original natural image for the pencil drawing image. Here, we use color histogram matching to transform the pixel. Let x_c be a RGB vector pixel of original natural image and x_p be a pixel of pencil drawing image obtained in Sect. 3.3. The pixel transformed between these two images is:

$$x_p = Ax_c + b, \tag{19}$$

where A is a 3×3 matrix and b is a 3-vector. So, if we can find A and b to satisfy Eq. (19) in color mean and covariance, then we can transform the color in original natural image to pencil drawing image. This is satisfied by:

$$u_p = Au_c + b \tag{20}$$

$$A\Sigma_p A^{\mathrm{T}} = \Sigma_c, \tag{21}$$

where u_p and u_c be the mean colors of the pencil drawing and original natural images, and Σ_p and Σ_c be the pixel covariances. The mean and covariance of the color pixels are given by $u = \sum_i x_i/N$ and $\Sigma = \sum_i (x_i - u)(x_i - u)^{\mathrm{T}}/N$. We make a Cholesky decomposition on Σ. Then, Eq. (21) has

$$\begin{aligned} & A\Sigma_p A^{\mathrm{T}} = \Sigma_c \\ \Rightarrow\ & AL_p L_p^{\mathrm{T}} A^T = L_c L_c^{\mathrm{T}} \\ \Rightarrow\ & AL_p = L_c \\ \Rightarrow\ & A = L_c L_p^{-1}, \end{aligned} \tag{22}$$

where $L_p L_p^{\mathrm{T}}$ and $L_c L_c^{\mathrm{T}}$ are Cholesky decompositions of Σ_p and Σ_c. We take Eq. (22) into Eq. (20), then we can get b. The synthesized results are shown in Sect. 4.

4 Results and analysis

We used VGG filters of Simonyan and Zisserman [20] as well as the platform of caffe framework [31]. We implemented our algorithm in python and ipython. Our implementation is on graphic processing unit (GPU) of NVIDIA GTX980. The images in our paper are derived from Internet. We apply our algorithm to the images. In our experiments, the original images included people, buildings, and landscape trees, which are typical examples used by the artist.

After surveying, we find that the method of Lu et al. is the best compared the traditional methods (as shown in Fig. 4, which is the comparison with commercial software and method of Lu et al. [7]). Therefore, in our paper, we just compare with the best traditional method (Lu et al.).

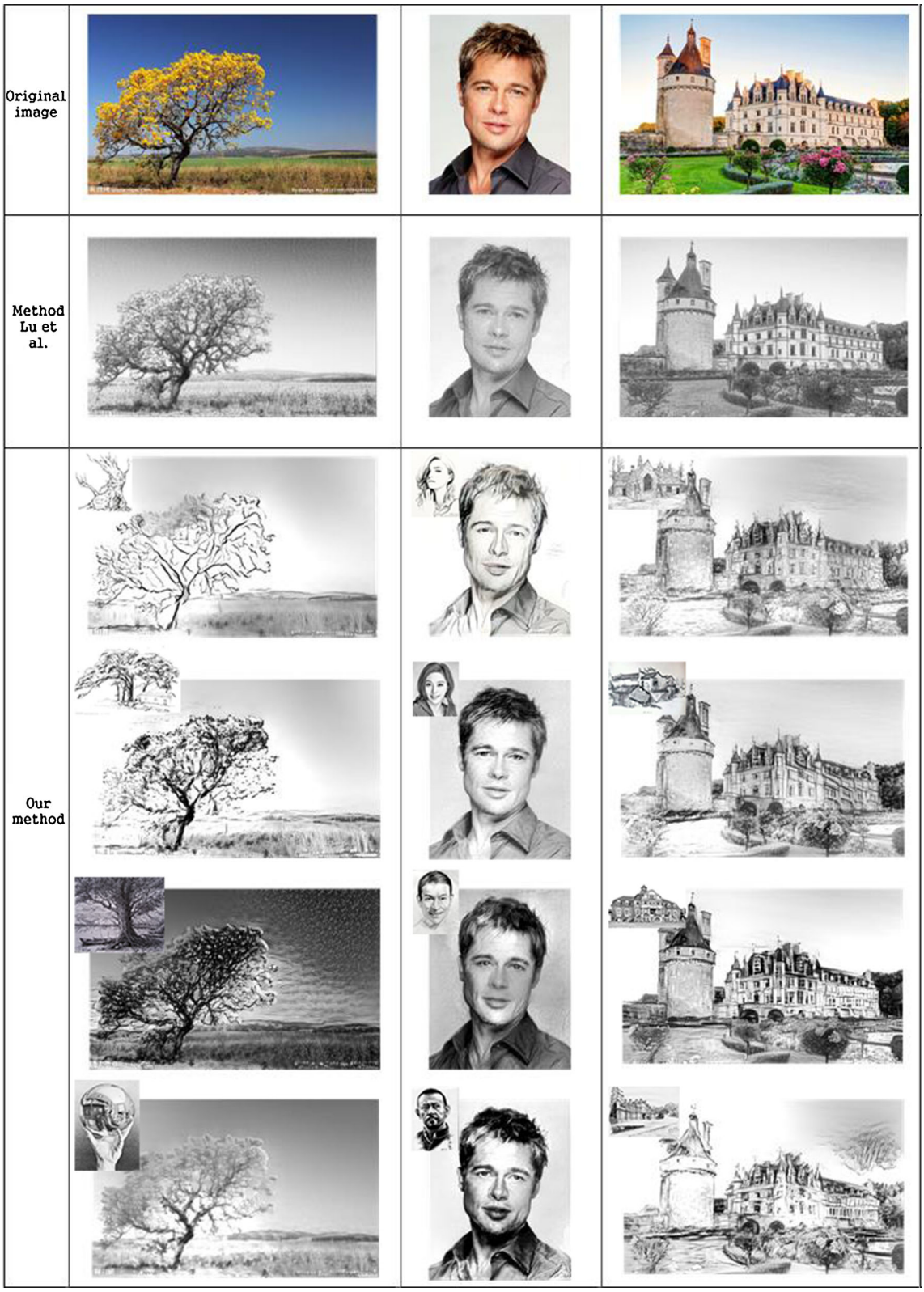

Fig. 5 First row is the original images. The second row is the synthesized results of method Lu et al. We randomly show out four kinds of our synthesized results

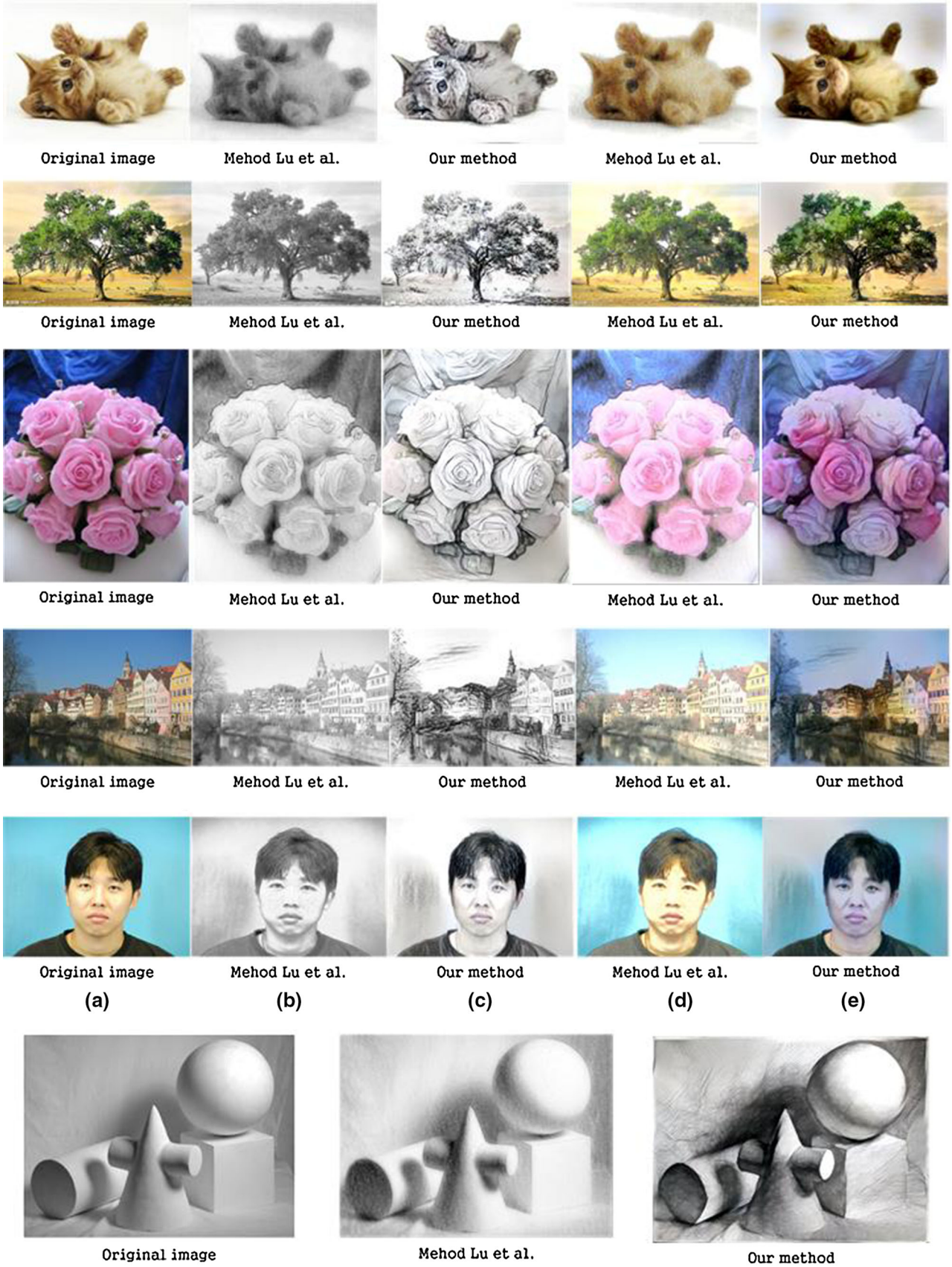

Fig. 6 Synthesized results of our method compared with method of Lu et al.

Fig. 7 Example of artifact synthesized in our result

In Fig. 5, the first row is the original images. The second row is the synthesized results of method Lu et al. There is only one kind result of pencil drawing image for the method of Lu et al. Until now, all the pencil-drawing-synthesized methods can only generate one kind of result. In our method, we can generate several kinds of pencil drawing image if there are several artist's works for learning. In Fig. 5, we randomly show four kinds of our synthesized results proving that our method are flexible in drawing style. The image in the upper left corner is the work of the corresponding artist. The size of the original image and the art image is arbitrary, without any restrictions. And the size of pencil drawing image is the same as the original image. From this picture we can see that our synthesis results can be somewhat similar to the artist's style, and can preserve the outline of the original image.

In Fig. 6, except the last row, column (a) is the original image; column (b) is the result synthesized by method of Lu et al. in which blur problem exists; column (c) is the result synthesized by our method. Compared with the method of Lu et al., our method is more clear. Column (d) shows the color pencil-drawing-synthesized results of Lu et al. Column (e) shows the color pencil-drawing-synthesized results of ours. In comparison with the method of Lu et al., the color of our method is brighter. In the last row, the second picture is the synthesized result of method Lu et al. and the third picture is the synthesized result of ours. Compared with method of Lu et al., our method result is more like pencil drawing image in biological visualization. Because the original image is gray picture, we do not compare the color-synthesized result in this row.

From Figs. 5 and 6, we can find that compared with method of Lu et al., our method is able to draw in different styles which can be chosen to give the result visually most pleasing or best suited to the purpose. Above all, our method is flexible in drawing style and can generate any kind style of pencil drawing image by learning the existing styles of artists.

However, our approach lacks some robustness in style control, which is reflected in the fact that artifacts occur when the texture of the art style changes more strongly and the original image texture is processed more slowly. Artifact shown in Fig. 7 appears when the sample image and the artwork strongly differ in style, due to the relatively low robustness of the style control. This also implies an increased processing time.

5 Conclusion

In this conceptual paper, we explore a new framework for pencil drawing image synthesis based on CNN feature maps and illustrate that it is robust for most kinds of images. Our method can save the color of the source image and generate color pencil drawing image. The next step of our works maybe focus on the quality improvement on semantic segmentation. Besides, the optimization efficiency of L-BFGS is not high. We will consider optimizing this step in the following work.

Acknowledgements We thank the anonymous reviewers and the editor for their valuable comments. This work has been supported by the National Natural Science Foundation of China (Nos. 61772387 and 61372068), the Research Fund for the Doctoral Program of Higher Education of China (No. 20130203110005), the Fundamental Research Funds for the Central Universities (No. K5051301033), the 111 Project (No. B08038) and also supported by the ISN State Key Laboratory.

References

1. Decarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A.: Suggestive contours for conveying shape. ACM Trans. Graph. **22**(3), 848–855 (2010)
2. Judd, T., Durand, F., Adelson, E.H.: Apparent ridges for line drawing. ACM Trans. Graph. **26**(3), 19 (2007)
3. Lee, Y., Markosian, L., Lee, S., Hughes, J.F.: Line drawings via abstracted shading. ACM Trans. Graph. **26**(3), 18 (2007)
4. Gao, X., Zhou, J., Chen, Z., Chen, Y.: Automatic generation of pencil sketch for 2D images. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, pp. 1018–1021 (2010)
5. Hertzmann, A., Zorin, D.: Illustrating smooth surfaces. In: Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co. pp. 517–526 (2004)

6. Praun, E., Hoppe, H., Webb, M., Finkelstein A.: Real-time hatching. In: Proceedings of the ACM Siggraph, p. 581 (2004)
7. Lu, C., Xu, L., Jia, J.: Combining Sketch and Tone for Pencil Drawing Production, pp. 65–73. Eurographics Association, Geneve (2012)
8. Gatys, L.A., Ecker, A.S., Bethge, M.A.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423 (2016)
9. Cai, X., Song, B.: Combining inconsistent textures using convolutional neural networks. J. Vis. Commun. Image Represent. **40**, 366–375 (2016)
10. Wang, N., Zhang, S., Gao, X., Song, B., Li, J., Li, Z.: Unified framework for face sketch synthesis. Signal Process. **130**, 1–11 (2017)
11. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L0 gradient minimization. ACM Trans. Graph. (TOG) **30**(6), 61–64 (2011)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp.1097–1105 (2012)
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Berg, A.C.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
15. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708 (2014)
16. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the CVPR (2015)
17. Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: Feedforward semantic segmentation with zoom-out features. In: Proceedings of the CVPR (2015)
18. Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 328–335 (2014)
19. Gatys, L.A., Ecker, A.S., Bethge, M.A.: Neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
21. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: ICLR (2015)
22. Cadieu, C.F., Hong, H., Yamins, D.L.K.: Deep neural networks rival the representation of primate IT cortex for core visual object recognition. PLoS Comput. Biol. **10**(12), e1003963 (2014)
23. Gl, U., van Gerven, M.A.J.: Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. J. Neurosci. **35**(27), 10005–10014 (2015)
24. Yamins, D.L.K., Hong, H., Cadieu, C.F.: Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. **111**(23), 8619–8624 (2014)
25. Khaligh-Razavi, S.M., Kriegeskorte, N.: Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS Comput. Biol. **10**(11), e1003915 (2014)
26. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Computer Vision and Pattern Recognition (CVPR), pp. 3606–3613 (2014)
27. Cimpoi, M, Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and description. In: Proceedings of the CVPR (2015)
28. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. IJCV **81**(1), 24–52 (2013)
29. Zhu, S., Ma, K.-K.: A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. Image Process. **9**(2), 287–290 (2000)
30. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Trans. Math. Softw. (TOMS) **23**(4), 550–560 (1997)
31. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678 (2014)
32. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 229–238. ACM (1995)
33. Portilla, J., Simoncelli, P.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int. J. Comput. Vis. **40**(1), 49–71 (2000)
34. Xie, X., Tian, F., Seah, H.S.: Feature guided texture synthesis (FGTS) for artistic style transfer. In: Proceedings of the 2nd International Conference on Digital Interactive Media in Entertainment and Arts, pp. 44–49 (2007)

Xiuxia Cai received the B.S. degree in Department of mathematics and application from Shandong University of technology, and received the M.S. degree in Department of information and signal processing from Xi'an Jiaotong University. Now, she is a Ph.D. student in Xidian University. She is currently focussing on image processing and deep learning.

Bin Song received the B.S. degree, M.S. degree and Ph.D. degree in Communication and Information System from Xidian University, Xian, China, in 1996, 1999 and 2002, respectively. He is currently a professor of the School of Telecommunications Engineering Xidian University. He has authored over 50 journal papers and conference papers. His research interests and areas of publication include image and video compression, error- and packet-loss-resilient video coding, video transcoding, distributed video coding, video signal processing based on compressed sensing, and multimedia communications.