Direction-aware Neural Style Transfer

Hao Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
knightwuhao@outlook.com

Zhengxing Sun*
State Key Laboratory for Novel Software Technology, Nanjing University, China
szx@nju.edu.cn

Weihang Yuan
State Key Laboratory for Novel Software Technology, Nanjing University, China
dereky0322@gmail.com

ABSTRACT

Neural learning methods have been shown to be effective in style transfer. These methods, which are called NST, aim to synthesize a new image that retains the high-level structure of a content image while keeps the low-level features of a style image. However, these models using convolutional structures only extract local statistical features of style images and semantic features of content images. Since the absence of low-level features in the content image, these methods would synthesize images that look unnatural and full of traces of machines. In this paper, we find that direction, that is, the orientation of each painting stroke, can capture the soul of image style preferably and thus generates much more natural and vivid stylizations. According to this observation, we propose a Direction-aware Neural Style Transfer (DaNST) with two major innovations. First, a novel direction field loss is proposed to steer the direction of strokes in the synthesized image. And to build this loss function, we propose novel direction field loss networks to generate and compare the direction fields of content image and synthesized image. By incorporating the direction field loss in neural style transfer, we obtain a new optimization objective. Through minimizing this objective, we can produce synthesized images that better follow the direction field of the content image. Second, our method provides a simple interaction mechanism to control the generated direction fields, and further control the texture direction in synthesized images. Experiments show that our method outperforms state-of-the-art in most styles such as oil painting and mosaic.

CCS CONCEPTS

• **Computing methodologies → Image-based rendering**;

KEYWORDS

Neural Style Transfer, Neural Networks, Direction Field

ACM Reference Format:
Hao Wu, Zhengxing Sun, and Weihang Yuan. 2018. Direction-aware Neural Style Transfer. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240629

*Corresponding author(szx@nju.edu.cn).

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240629

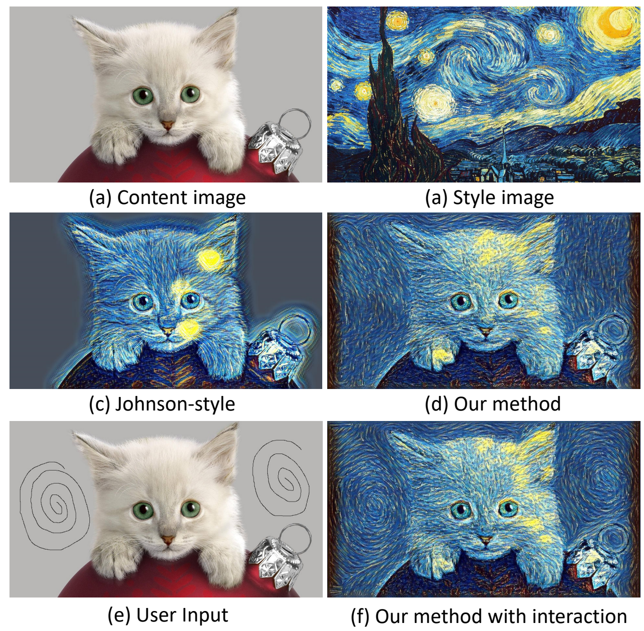


Figure 1: An example of our DaNST method and its result with interaction. (a)(b) is content and style image. (c)(d) are the results from Johnson[5] and our method . These figure show that our method can synthesize directional stroke in the background while Johnson-style can not. Since the style image are full of strokes, the result of ours are much more appealing. (e) is the user input for the interaction. (f) is corresponding results from our method according to the interaction. The result show that our method can change the orientation of the texture in the image through interaction.

1 INTRODUCTION

Image stylization is a classical problem in computer graphics and vision[10]. Image-based Artistic Rendering (IBAR) provides a direct method to transfer style to given image. This kind of methods uses example template (e.g., an oil painting) to extract primitives to represent texture, and the stylized image is synthesized from a real image (e.g., a portrait photo) [4]. In recent years, deep convolutional neural networks have demonstrated dramatic improvements in performance for computer vision task. In the area of IBAR, Neural Style Transfer methods [1, 5, 19, 20] have improved the transfer quality of Classical Image-based artistic rendering dramatically.

Recently, Gatys et al. [1] propose parametric synthesis models for IBAR, which utilize CNNs to synthesize style image that retains the high-level structure of a content image and keeps the low-level texture of a style image . For some kind of artworks, particularly for oil painting, these models can result in quite successful, state-of-the-art results. Johnson et al.[6] propose an feed-forward network based method to improve the efficiency of NST methods. Existing deep neural network based image style transfer methods often produce impressive results on varies artworks. However, as shown in the survey of Amir et al. [19], local effects and phenomena of traditional artistic media such as oil paint, pencil or embroidery are still hard to reproduce by NSTs. By introducing new constraints in the loss function like Edge Loss or Depth loss, a few works [12][14] improved local effects of Neural Style Transfer results. These works show that, in traditional optimization objective, low-level features of the content image are absent, and the low-level features of the style image dominate the low-level detail structures of the synthesized images. This will cause the loss of structure information of content image and produce some unpleasing results. In this paper, we find that the low-level feature direction, that is, the orientation of each painting stroke, can capture the soul of image style preferably and thus generates much more natural and vivid stylizations

The limitation of existing NST methods is that they do not consider the relationship between directional strokes, such as oil painting strokes and embroidery threads, in style images and directional texture in content images. This leads to the fact that they cannot produce synthesized images with corresponding directions according to the direction of texture in content images (e.g. red boxes of fig.1 b and c). To our knowledge, none of the existing NST methods concern the direction of texture in their transfer process. But in the area of Classical IBAR methods, many works have shown that[7, 15, 16], when transfer the style, the quality of the result can be improved significantly when the algorithm takes into account the direction of texture. Especially for some art, the direction of texture plays a vital role in visual effect, such as oil painting and embroidery. In our method, we introduce a novel directional field loss to steer the direction of synthesized image by comparing the differences between the direction field of content image and synthesized image. In order to construct this loss, we build direction field loss networks to generate direction field of content image and synthesized image. We combine our directional field loss with content loss and style loss from [1, 5] to build our novel direction-aware optimization objective.

In addition, another disadvantage of the existing NST methods is that they regard the style transfer process as an end-to-end process, thus turning the whole process into 'black box'. In the survey of [19], Amir et al. show that how to add controllable mechanism into NST will become the trend of NST in the future, but how to achieve this interaction is still very difficult. In this paper, we take the first step to explore the way of introducing interaction into NST. Through simple interaction with the content image, we can control the direction field of content image generated by our direction loss networks. By this way, we can easily control the direction of texture in synthesized image.

Finally, we use the our direction-aware optimization objective to synthesize style image[1]. Compared with existing methods, the contributions of our work can be summarized as follow:

(1) We are the first to propose a novel direction-aware loss for neural style transfer, which can synthesize style image with proper directional texture according to the direction field of content image, which is extremely useful in some artworks, such as oil painting and embroidery. Extensive experiments show that our method generates much more vivid synthesized images with accurate directional texture.

(2) By interaction in the direction field generation process, our method can control the orientation of the texture in the synthesized image. The experiments show that our method provides an easy way for interaction and can even synthesize texture in the smooth area of content image.

(3) Our Direction Field Loss network can be incorporated into almost any existing NST methods for preserving directional texture according to the direction field of content image.

2 RELATED WORK

Direction-aware Classical Image-based Artistic Rendering: In the area of classical IBAR, the IBAR methods can be divided into two categories, namely, Heuristics-based Algorithms and Image Statistics based style transfer. Heuristics-based Algorithms are based on rendering functions, which are implemented by a domain expert who explicitly models individual artistic styles and its correspondent design aspects or mechanisms. These methods render strokes according to the region or direction field informations of the content image. Image Statistics based style transfer are based on the idea to copy image patches from a style image to a content image in a way that locally shares and minimizes pixel differences in the content image. In the literatures of both methods, there are methods that rendering or transferring the direction strokes or direction texture along the direction field of the content image in a direction-aware way. Many works have proved that the direction-aware method has better effect on the art created by directional strokes, such as oil painting and embroidery[7, 15, 16, 18, 23, 24].

In the area of Heuristics-based Algorithms, many works propose direction-aware rendering methods to render strokes according to the direction field of the content image. Kang et al.[8] propose an method to extract smooth direction field that preserves the flow of the salient image feature from the content image using an edge tangent flow field method. Then they introduce a flow-guided anisotropic filtering for detecting highly coherent lines to produce a coherent line drawing effect. Amir et al. [18, 20] propose a similar direction field extraction method based on adaptively smoothed structure tensor. They use direction field to generate paint texture that looks like directional painting strokes. Thus this method can produce amazing oil painting like results. Zeng et al. [24] generate direction field for the content image and use the direction field to guide the placement of the pre-collected directional oil painting strokes.

In the area Image Statistics based style transfer methods, several works propose direction-aware style transfer process to transfer directional texture from style image to the content image according to direction field of content image. These direction-aware methods manage to discover patches with direction information from the style image. In these patches, the texture is painted by strokes that have dominant direction (e.g. strokes of line drawing[7]). For example, Wang et al.[23] rotate style image into 24 directions and use dense sampling at each rotated image to extract patches. These patches construct a repository with discrete direction information. The repository is then used to generate texture in the synthesize image according to the direction field of content image. Instead of rotating the style image discretely, Luká et. al [15] propose a patch extraction method based on patches which can rotate continuously to sample the directional texture according to the dominant direction. This method is effective to extract the primitive with complex direction field (e.g., oil painting of hair or fur) and can produce better style transfer results through a direction-aware way.

All the works above produce an appealing stylized result using a direction-aware method. they can produce such a good result, because they consider the direction of texture and strokes.

Neural Style Transfer With the development of both theory and hardware capability, deep learning offers a novel alternative for style transfer. As ground truth is generally unavailable for style transfer, training a model that extracts features dedicated for style transfer is challenging. As a first attempt, Gatys et al. [1] use Gram matrices of the neural activations from different layers of a Convolutional Neural Network (CNN) to represent the artistic style of an image, and generate a new image from a white noise initialization followed by an iterative optimization process. This novel method attracted many follow-up works aimed at improving different aspects of the approach. To reduce the computational burden, Johnson et al. [6] and Ulyanoy et al. [22] train a feed-forward network to quickly approximate solutions to the optimization problem. To improve the transfer results, researchers have developed different complementary schemes, e.g. by incorporating novel spatial constraints through gain maps [17] and semantic maps [2]. Li and Wand et al. [11] introduced a framework based on Markov Random Field (MRF) in the deep feature space to enforce local patterns. Among all the works, Klingbeil et al. use a combination of NST with user-controlled state-of-the-art nonlinear image filtering and greatly improve the synthesized results of [1] at a local scale. More recently, Li et al.[12] introduce a Laplacian loss constraint in the loss function of neural style transfer to measure the detail structure differences between synthesized image and content image. This method removes the artifacts in the synthesized image and preserves the detail structure of content image.

But none of the existing NST methods have considered the direction of texture in the content image and synthesized image. Last but not least, these methods lack the mechanism of interaction to control the synthesized image.

3 METHOD

3.1 Direction-Aware Neural Style Transfer

Given a content image y_c and a style image y_s. Our Direction-aware Neural Style Transfer process can synthesize image $\hat{y}$ through minimizing a optimization objective. Our method is constructed by three loss net ϕ_0 , ϕ_1 and ϕ_2. These three loss networks are used to define three loss functions: l_1, l_2 and l_3, where l_1 and l_2 are based on ϕ_0, and correspond to the style loss and content loss, also denoted as $l^{\phi_0}_{style}$ and $l^{\phi_0}_{content}$ respectively. l_3 is the direction field loss based on ϕ_1 and ϕ_2. The direction-aware neural style transfer generates image $\hat{y}$ by solving the following optimization objective:

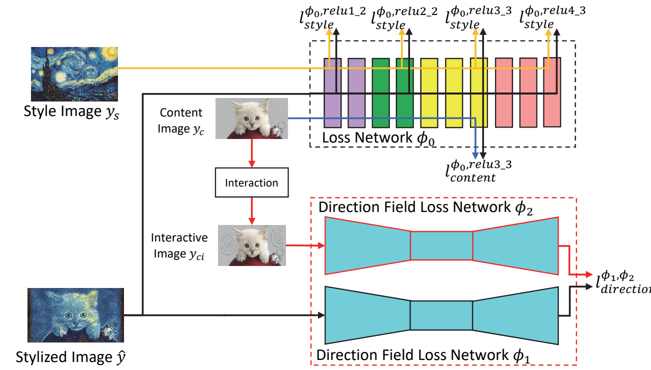


Figure 2: Method overview: We propose a triple loss networks based NST called DaNST. In our method, our direction field loss networks is build to construct direction loss. Loss Network ϕ_0 is based on VGG-16 follow Gatys et al. [1]. And we use ϕ_0 to construct style loss and content loss. Given style image and content image, we generate synthesized image by minimizing the losses described above. Moreover, the user can interact with DaNST by drawing curves in the content image. See Sec.3.1 for detail.

$$W^* = \arg\min_{W} E_{x,\{y_i\}} \sum_{i=1}^{3} [\lambda_i l_i(f_W(x), y_i)] \tag{1}$$

The three loss functions fall into two categories: perceptual loss functions $l^{\phi_0}_{style}$ and $l^{\phi_0}_{content}$ and per-pixel loss function $l^{\phi_1,\phi_2}_{direction}$. Compared with the traditional neural style transfer optimization objective [1, 5], we add directional field loss $l^{\phi_1,\phi_2}_{direction}$:

$$l_{total} = \lambda_1 l^{\phi_0}_{style} + \lambda_2 l^{\phi_0}_{content} + \lambda_3 l^{\phi_1,\phi_2}_{direction} \tag{2}$$

Perceptual loss functions based on high-level features extracted from pre-trained networks, are used to measure high-level perceptual and semantic differences between images. Compared with per-pixel losses, perceptual losses measure image similarities more robustly. This is because the convolutional neural networks pretrained for image classification can learn to encode the perceptual and semantic information according to the recent works [3, 21]. In contrast, when we have a ground-truth image that the network is expected to match, per-pixel loss is more suitable . This is suitable for the direction loss, direction fields can be estimated from the content and synthesized images. In our method, ϕ_0 is a pre-trained image classification network (VGG-16), and ϕ_1, ϕ_2 are single-image direction field loss networks. In the following sections we will introduce our direction field loss networks in detail.

In the synthesis process, the user supplies style image y_s and content image y_c. Then a optimization process is used to generate synthesize image $\hat{y}$ iteratively by minimizing the total loss l_{total}. The style loss $l^{\phi_0}_{style}$ is produced by comparing each $\hat{y}$ with y_s in the loss network ϕ_0, and the content loss $l^{\phi_0}_{content}$ is produced by comparing each $\hat{y}$ with y_c in the same loss network ϕ_0. The direction loss $l^{\phi_1,\phi_2}_{direction}$ is produced by two additional direction field loss networks ϕ_1 and ϕ_2 through comparing the output of $\hat{y}$ and y_c in ϕ_1 and ϕ_2 respectively, with the aim of making the synthesized image retain a direction field output consistent with the content.

Moreover, the user can interactive with the content image y_c to generate an interactive content image y_{ci} before the synthesis process. The interactive content image y_{ci} will feed the direction field loss network ϕ_2 while the non-interactive content image still feed the loss network ϕ_0. Though this interactive mechanism, we can force the direction field of synthesized image $\hat{y}$ to be consistent with the direction field of the interactive content image y_{ci}.

3.2 Content Loss

The content loss function is used to measure the dissimilarity between content image y_c and synthesized image $\hat{y}$. Let $F^l_i(x)$ denote the i-th feature map in the l-th layer of the VGG-16 loss network applied to image x. the content loss is the squared-error loss between the two feature representations at layer l.

$$l^{\phi_0}_{content} = \sum_{i=1}^{N_l} \|F^l_i(y_c) - F^l_i(\hat{y})\|^2_2 \tag{3}$$

That is, the content loss directly compares the feature maps computed from the corresponding layers and thus is suitable for characterizing spatial content similarity.

3.3 Style Loss

Gatys et al. propose that the correlations between feature maps in each layer of the loss network can be seen as texture representations of an image [1][5]. Those correlations are given by the Gram matrix, whose elements are pairwise scalar products between those feature maps:

The Gram matrix $G^{\phi_0}_j$ is defined as the inner product of every two filter responses. It is given by:

$$G^{\phi_0}_j(x)_{c,c'} = \frac{1}{C_j \times H_j \times W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi^j_0(x)_{h,w,c} \phi^j_0(x)_{h,w,c'} \tag{4}$$

The contribution of layer j to the total style loss is defined as the squared Frobenius norm of the difference between the Gram matrices of the stylized image $\hat{y}$ and style image y_s:

$$l^{\phi_0,j}_{style}(\hat{y}, y_s) = \|G^{\phi_0}_j(\hat{y}) - G^{\phi_0}_j(y_s)\|^2_F \tag{5}$$

J is the set of selected layers and the total style loss is:

$$l^{\phi_0}_{style}(\hat{y}, y_s) = \sum_{j \in J} l^{\phi_0,j}_{style}(\hat{y}, y_s) \tag{6}$$

Details about the selected layers J is described in our experiments.

3.4 Direction Field Loss

Given content image y_c and synthesized image $\hat{y}$, we use a direction field $l^{\phi_1,\phi_2}_{direction}$ to measure the difference between their direction field:

$$l^{\phi_1,\phi_2}_{direction} = \sum_{i=1}^{N} \|\phi_1(y_c) - \phi_2(\hat{y})\|^2_2 \tag{7}$$

That is, the direction loss directly compares the direction field generated by the direction field loss networks ϕ_1 and ϕ_2 from the content image y_c and the synthesized image $\hat{y}$. The detail description of the direction field loss networks are provided in the following section 3.5 and 3.6.

3.5 Architecture of Direction Field Loss Networks

Inspired by the excellent function fitting ability of neural network, we use neural networks to generate direction field loss. Because the input image and the output direction field are matrices of same size, we use fully-convolutional neural networks (fCNNs) as the basic architecture of our **Direction Field Loss Networks**. Our image **Direction Field Loss Networks** roughly follow the architectural of image transform net in Johnson et al.'s work[5]. Our network body comprises five residual blocks. All non-residual convolutional layers are followed by batch normalization and ReLU nonlinearities with the exception of the output layer, which uses a scaled tanh to ensure that the output has pixels in the range $[0, 180]$ instead of $[0, 255]$. This ensures that the output of our network is consistent with the output value of direction field generation process, which is in the range $[0°, 180°]$. The first and last layers use 9×9 kernels; all other convolutional layers use 3×3 kernels.

Input and Outputs: For direction field loss networks the inputs is gray image of shape $1 \times 512 \times 512$. The pixels of input image are in the range of $[0, 255]$. The output is the corresponding direction field matrix of the same shape as the input image. But each item in the output matrix is in the range of $[0, 180]$. This is because the range of orientation angle of the direction field is in the range of $[0°, 180°]$. Since the Direction Field Loss Networks are fully-convolutional, at test-time they can be applied to images of any resolution.

Downsampling and Upsampling: We first use two downsampling layers and then two upsampling layers, each of stride 2, to process the input. Between the sampling layers are several residual blocks. After these processing steps, the size of the image is preserved, but this procedure comes with two advantages: On the one hand, after downsampling, we can use a larger network for the same computational cost. For instance, the computational cost of a 3×3 convolution with C filters on an input of size $H \times W \times C$ is equal to a 3×3 convolution with DC filters on an input of shape $\frac{H}{W} \times \frac{W}{D} \times DC$, where D is the downsampling factor. On the other hand, downsampling gives a larger effective receptive fields with the same number of layers. For instance, without downsampling, each additional 3×3 convolutional layer increases the effective receptive field size by 2. After downsampling by a factor of D, the effective receptive field size increases to $2D$. In general, the larger the receptive fields, the better the style transfer results are.

Residual Connections: He et al.[3] point out that residual connections make it easy for the network to learn the identity function. It can be observed that when performing style transfer, in

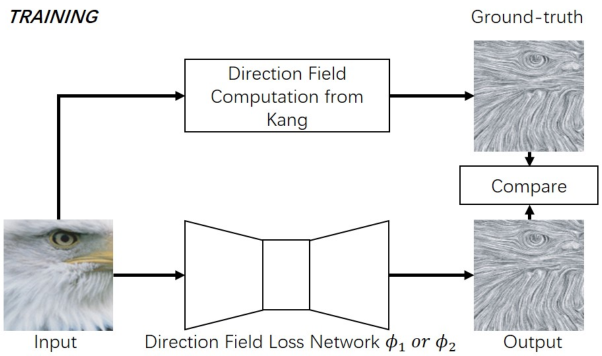
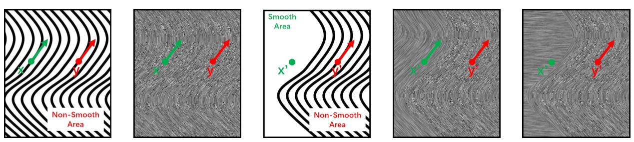


Figure 3: Training Process of Direction Field Loss Networks



(a) image k (b) $\phi_1(k)$ (c) image l (d) $\phi_1(l)$ (e) $\phi_2(l)$

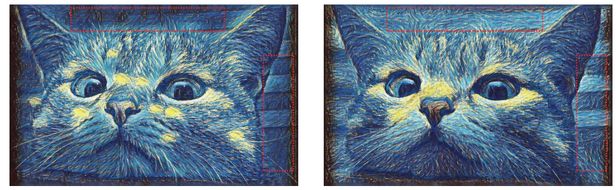
Figure 4: Example image k and l. and its corresponding direction field generated by direction field networks ϕ_1 and ϕ_2.

many cases, the output image should share structure with the input image, so we include several residual blocks in our network to enhance this ability.

3.6 Direction Field Loss Networks Construction and Training

In this section, our goal is to construct direction field loss networks to compute a smooth feature-preserving direction field. The architecture of the direction field loss networks is described in section 3.5. The direction field must satisfy the following requirements: (1) The direction field must describe the salient edge direction in the neighborhood; (2) The neighboring direction vector must be smoothly aligned except at sharp corners; (3) Important edges must retain their original directions; (4) The computation process must has the ability to be integrated into the loss function, and the computation process must be derivable for backpropagation optimization. The easiest way to produce direction field is to define a direction field $t(x)$ which is perpendicular to the image gradient $g(x) = \nabla I(x)$. But this method can not accurately solve the direction field in some cases, especially in an area with image noise or low contrast (smooth) area. The training process is described in fig.3.

Kang et al.[7] have proposed an effective method to compute direction field. But unfortunately, their method is too complex to be derived for backpropagation optimization and could hardly be integrated into the loss function. Inspired by the strong function fitting ability of deep neural networks (DNNs), we propose two direction field loss networks ϕ_1,ϕ_2 to compute the direction field for content image y_c and synthesized image $\hat{y}$ respectively. The direction field loss network ϕ_1 follows the method proposed by [7]



(a) Both using ϕ_1 (b) Using ϕ_1 and ϕ_2

Figure 5: Example synthesized image. (a) is the result using only ϕ_1 and (b) is the result using ϕ_1 and ϕ_2. The red box areas show that although the directional strokes in the foreground are synthesized accurately, the directional strokes in the background be synthesized with directional texture.

to generate direction field for content image y_c. The direction field loss network ϕ_2 generate direction field for synthesized image $\hat{y}$ by a modified method of [7]. We train our direction field loss networks ϕ_1 using the output direction field of [7]. And for the training of ϕ_2, we just adjust the process range parameter to remove the direction field generated in the smooth area. Then most of the direction field of the smooth area will be set to zero. The results can be find in the fig.4 a,b and fig.4 c,e.

One may argue that why don’t we use the same direction field loss network ϕ_1 to generate direction field for both images $\hat{y}$ and y_c and compute the direction loss $l_{direction}$ through comparing the $\phi_1(y_c)$ and $\phi_1(\hat{y})$. This is because the method of [7] would generate direction field follow the nearest edge (fig.4d) in regions with smooth texture (no obvious direction), such as the smooth area of fig.4c. Thus, under some circumstances, it is possible to generate the same or similar direction field $\phi_1(y_c) \approx \phi_1(\hat{y})$ whether the texture of a region of the image is smooth or not. For example, assume the texture of a region of y_c is not smooth (fig.4a) and the corresponding region of $\hat{y}$ is smooth (fig.4c), the output $\phi_1(y_c)$ (fig.4b) and $\phi_1(\hat{y})$ (fig.4d) could generate the same value (green arrow in both fig.4b and fig.4d). This will cause the loss of texture in the synthesized image. The algorithm are seeking the texture consistency of y_c and $\hat{y}$ while it can’t decided whether the fig.4a and fig.4c are consistent or not, since the direction field fig.4b and fig.4d are similar. These failures could be found in fig.5. To avoid these circumstances, we propose direction field loss network ϕ_2 which only generate direction field in non-smooth area. Due to the limitation of space, we will not introduce the direction field generation method in detail. Readers can find this detailed generation process in the original paper [7].

3.7 Interaction With Direction Field Loss Networks

As described in section 1, the user can interactive with the content image y_c to generate an interactive content image y_{ci} before the synthesis process. The interactive content image y_{ci} will feed the direction field loss network ϕ_2 while the non-interactive content image still feed the loss network ϕ_0. Though this interactive mechanism, we can force the direction field of synthesized image $\hat{y}$ to be consistent with the direction field of the y_{ci}.

To interactive with the direction field loss network ϕ_2, the user can simply add straight line or curves on the input content image

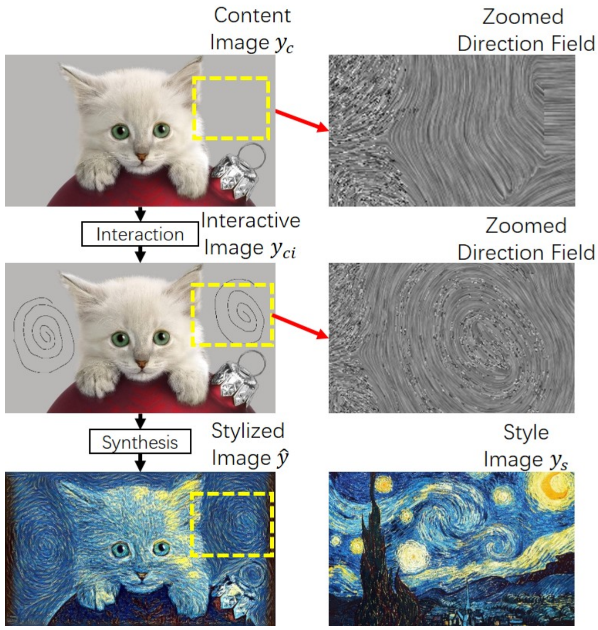
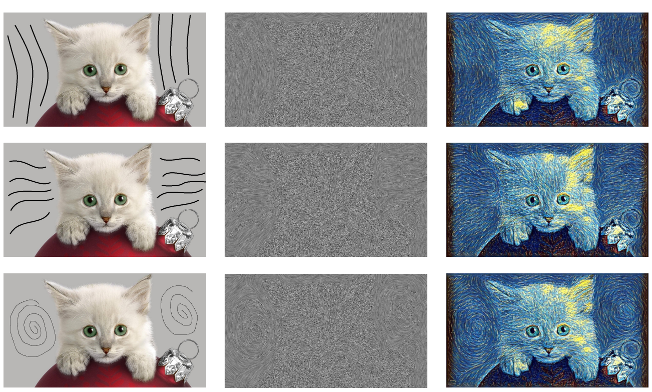


Figure 6: Interaction process and the corresponding result. It can be find in this figure that the user input interaction (yellow box in interactive image) will influence the direction field of content image and will generate the texture in yellow box of stylized image. In the stylized image, the strokes from style image follow the user input accurately.



(a) Interactive image (b) Direction Field (c) Synthesized image

Figure 7: Example of interactive results. (a) are the image of user interactive input y_{ci} . (b) are the results of corresponding direction field $\phi(y_{ci})$. (c) are the results of corresponding interactive style transfer. The results show that our method can control the directional texture in the synthesized images according to the user interaction.

to generate image y_{ci}. This interactive content image will be fed to the direction field loss network ϕ_2. The process can be found in the framework fig.2 and fig.6. Then the interactive content image will be used to generate direction field $\phi_2(y_{ci})$ which is corresponding to the user input y_{ci}. Thus the synthesized image will generate the directional texture at the interactive area according to the user. More results can be found in fig.7.

4 EXPERIMENTS

In this section, we show the results of our DaNST on various content and style images and evaluate the results with interaction. Compared to the NST methods proposed by Gatys et al. [1] , Johnson et al.[5] and MRF-CNN proposed by [11], DaNST yields more appealing images that better preserve the details of directional texture. In the experiments, we use VGG-16 to compute perceptual loss for the traditional NST method like Gatys et al. [1] , Johnson et al.[5] and our DaNST. We compute content loss at layer relu3_3 and style loss at layer relu1_2, relu2_2, relu3_3 and relu4_3 of the VGG-16 loss network. Our Direction Field Loss Networks work only on the original content image and synthesized image since the direction field may be severely distorted on the higher layers of VGG-16 loss network. The weights of three losses are 1(content), 5(style) and 3 (direction). For the MRF-CNN, we keep the original setting of the paper to yield best results.

4.1 Training Details of Direction Field Loss Networks

We choose Microsoft COCO [13], including 80k images, as the training dataset. All the training images are converted to gray-scale and resized to 512×512. We generate the corresponding direction fields of training images with Kang et al.'s method[7]. The generated direction fields are stored in the format of HDF5 which is a high performance storage solution for large matrix. The direction fields serve as the ground-truth. Then we train our Direction Field Loss Networks with a batch size of 7 for 300 iterations. We use Adam [9] with a learning rate of 1×10^{-3} , because this method is straightforward, computationally efficient, has few requirements and is well suited for problems that involve large amounts of data.

4.2 Interactive Results

In fig.7, we demonstrate how the user interaction will influence the synthesized image. This is done by generating direction field according to the user input. The user can easily interact with the content image through drawing a few lines or curves, then the synthesized image will follow the lines or curves inputted by the user. With our method, the user can even draw circles to control the strokes in the synthesized image. It can be observed that the synthesized image will follow the input from the user accurately. This means our interaction mechanism is not only effective but also simple and easy to use.

4.3 Qualitative Results

In fig.8 and 9, we qualitatively compare our DaNST with Gatys-style, Johnson-style, MRF-CNN on different style and content images.

It can be observed that images synthesized by our DaNST consistently improve on those by Gatys-style. Images synthesized by DaNST improve on those by Gatys-style. The details of directional strokes and texture in the foreground are preserved better and leave much less traces of machines. For example, the eyes of the

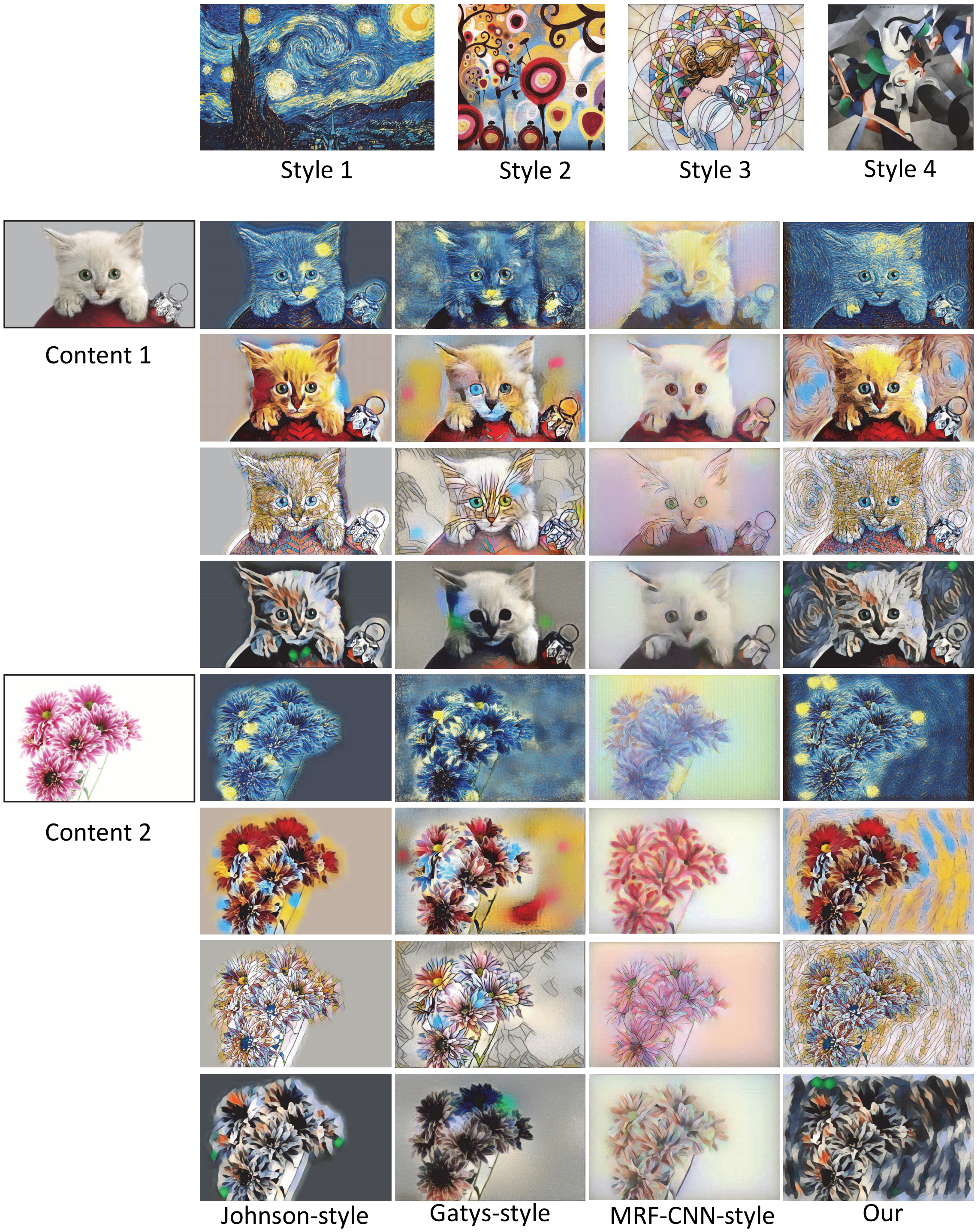


Figure 8: Result (a).

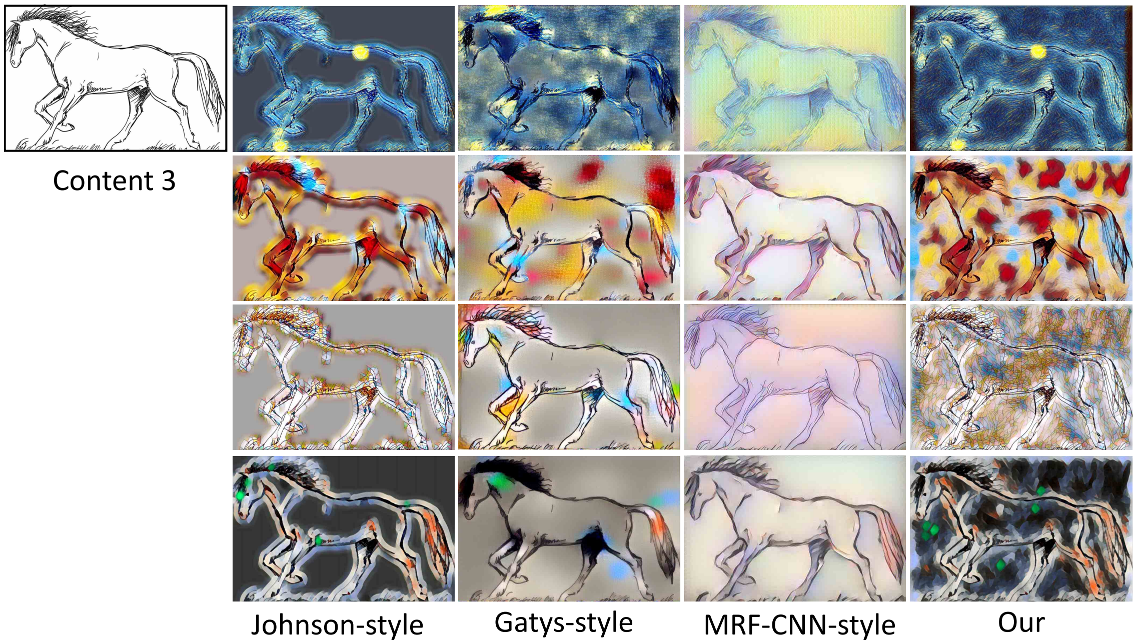


Figure 9: Result Result (b) Comparisons of Johnson-style, Gatys-style, MRF-CNN, and Our DaNST.Compared to these methods, our DaNST produces images with finer details and directional stroke according to the content image (Even for sketch). Our method can produce textures in the background while Johnson-style lose all the details. Gatys-style produce few details in the background but still not good enough. And details in the foreground are lost compared to our method. MRF-CNN seems to lose all the details in both background and foreground. Zoom 400% for detail.

cat of the $4th$ are all black in the synthesized image by Gatys-style. Our method and Johnson-style can generate details of the eyes.

Compare to the Gatys-style, Johnson-style can produce more details in the foreground of the image. But the details of the background are totally missing. The strokes are all over the style image while the strokes of synthesized images by Johnson-style are all disappeared. This makes the synthesized image much more unnatural. Our DaNST can generate directional strokes according to the direction field produce by the direction field loss networks in the background and thus produce much appealing synthesized images.

MRF-CNN is built to generate more details in some synthesized images. In the both foreground an background, the details of strokes are lost and the overall synthesized results are quite pool. For style 3, MRF-CNN synthesized mosaic with only edges while our method can generate more appealing results.

Our DaNST outperform all these method in details of directional strokes and can generate vivid synthesized image.

5 CONCLUSIONS

Neural Style Transfer uses an optimization objective that is concerned to preserve only high-level semantic CNN features of content image. Since the absence of low-level feature, the synthesized images will look unnatural and full of traces of machines. In this paper we have proposed a method called DaNST that is able to steer the direction of strokes and texture in the synthesized image. Based on the ability of controlling directional strokes in the synthesized image, we also have proposed a simple but effective way for the users to interact with the DaNST process. These are all based on a novel direction field loss which are build by direction field loss networks. The loss encourages the synthesized image to have a similar direction field as that of the content image. The direction field loss networks captures the direction of texture of the content image and guides the optimization towards a synthesized image that more retains the details of directional texture. We have validated our method and conclude that the images synthesized by our method look more natural and vivid. We also validate our interactive mechanism in our DaNST and show the effectiveness of interaction. User can control the direction of strokes and textures in the synthesized image through a simple interaction.

In the future, we seek to be able to apply our direction loss to progressive neural style transfer with multiple steps to further improve the detail effects and enhance the ability to synthesize high resolution images.

6 ACKNOWLEDGEMENT

This work was supported by National High Technology Research and Development Program of China (No. 2007AA01Z334), National Natural Science Foundation of China (Nos. 61321491 and 61272219), Program for New Century Excellent Talents in University of China (NCET-04-04605), the China Postdoctoral Science Foundation (Grant No. 2017M621700) and Innovation Fund of State Key Laboratory for Novel Software Technology (No. ZZKT2013A12, ZZKT2016A11 and ZZKT2018A09).

REFERENCES

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2414–2423. https://doi.org/10.1109/CVPR.2016.265

[2] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3730–3738. https://doi.org/10.1109/CVPR.2017.397

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[4] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. 2001. Image analogies. *SIGGRAPH* 2001, August (2001), 327–340. https://doi.org/10.1145/383259.383295

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*. 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. (2016), 694–711. https://doi.org/10.1007/978-3-319-46475-6-43

[7] Henry Kang, Seungyong Lee, and Charles K. Chui. 2007. Coherent line drawing. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering 2007, San Diego, California, USA, August 4-5, 2007*. 43–50. https://doi.org/10.1145/1274871.1274878

[8] Henry Kang, Seungyong Lee, and Charles K Chui. 2007. Coherent line drawing. *NPAR* 1, 212 (2007). https://doi.org/10.1145/1274871.1274878

[9] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[10] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. 2013. State of the 'Art: A taxonomy of artistic stylization techniques for images and video. *TVCG* 19, 5 (2013), 866–885. https://doi.org/10.1109/TVCG.2012.160

[11] Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2479–2486. https://doi.org/10.1109/CVPR.2016.272

[12] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. 2017. Laplacian-Steered Neural Style Transfer. In *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. 1716–1724. https://doi.org/10.1145/3123266.3123425

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. 2014. Microsoft COCO: Common Objects in Context, In ECCV. https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/

[14] Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L. Rosin. 2017. Depth-aware neural style transfer. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, NPAR 2017, Los Angeles, CA, USA, July 29-30, 2017*. 4:1–4:10. https://doi.org/10.1145/3092919.3092924

[15] M. Lukáč, J. Fišer, P. Asente, J. Lu, E. Shechtman, and D. Sýkora. 2015. Brushables: Example-based Edge-aware Directional Texture Painting. *CGF* 34, 7 (2015). https://doi.org/10.1111/cgf.12764

[16] Michal Lukác, Jakub Fiser, Jean Charles Bazin, Ondrej Jamriska, Alexander Sorkine-Hornung, and Daniel Sýkora. 2013. Painting by feature: texture boundaries for example-based image creation. *ACM Trans. Graph.* 32, 4 (2013), 116:1–116:8. https://doi.org/10.1145/2461912.2461956

[17] Ahmed Selim, Mohamed A. Elgharib, and Linda Doyle. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph.* 35, 4 (2016), 129:1–129:18. https://doi.org/10.1145/2897824.2925968

[18] Amir Semmo, Jürgen Döllner, and Frank Schlegel. 2016. BeCasso: Image Stylization by Interactive Oil Paint Filtering on Mobile Devices. In *Proceedings SIGGRAPH Appy Hour*. ACM, New York, 6:1–6:1. https://doi.org/10.1145/2936744.2936750

[19] Amir Semmo, Tobias Isenberg, and Jürgen Döllner. 2017. Neural style transfer: a paradigm shift for image-based artistic rendering? (2017), 5:1–5:13. https://doi.org/10.1145/3092919.3092920

[20] Amir Semmo, Matthias Trapp, Jürgen Döllner, and Mandy Klingbeil. 2017. Pictory: Combining Neural Style Transfer and Image Filtering *(Proceedings SIGGRAPH Appy Hour)*. ACM, New York, 5:1–5:2. https://doi.org/10.1145/3098900.3098906

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR* abs/1312.6034 (2013). arXiv:1312.6034 http://arxiv.org/abs/1312.6034

[22] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 1349–1357.

[23] Bin Wang, Wenping Wang, Huaiping Yang, and Jiaguang Sun. 2004. Efficient example-based painting and synthesis of 2D directional texture. *TVCG* 10, 3 (2004), 266–77. https://doi.org/10.1109/TVCG.2004.1272726

[24] Kun Zeng, Mingtian Zhao, Caiming Xiong, and Song-Chun Zhu. 2009. From image parsing to painterly rendering. *TOG* 29, 1 (dec 2009), 1–11. https://doi.org/10.1145/1640443.1640445