Shape-Matching GAN++: Scale Controllable Dynamic Artistic Text Style Transfer

Shuai Yang, *Member, IEEE*, Zhangyang Wang, *Member, IEEE*, and Jiaying Liu, *Senior Member, IEEE*

Abstract—Dynamic artistic text style transfer aims to migrate the style in terms of both the appearance and motion patterns from a reference style video to the target text to create artistic text animation. Recent researches have improved the usability of transfer models by introducing texture control. However, it remains an important open challenge to investigate the control of the stylistic degree with respect to shape deformation. In this paper, we explore a new problem of dynamic artistic text style transfer with glyph stylistic degree control. The key idea is to build multi-scale glyph-style shape mappings through a novel bidirectional shape matching framework. Following this idea, we first introduce a scale-ware Shape-Matching GAN to learn such mappings to simultaneously model the style shape features at multiple scales and transfer them onto the target glyph. Furthermore, an advanced Shape-Matching GAN++ is proposed to animate a static text image based on the reference style video. Our Shape-Matching GAN++ characterizes the short-term consistency of motion patterns via shape matchings within consecutive frames, which are propagated to achieve effective long-term consistency. Experiments show that the proposed method outperforms previous state-of-the-arts both qualitatively and quantitatively, and generate high-quality and controllable artistic text.

Index Terms—Text style transfer, structure transfer, scale control, temporal consistency

1 Introduction

ARTISTIC text is highly appreciated and widely used in many visual designs such as posters and websites. Recent works have investigated the automatic generation of artistic text based on two kinds of references. The first one is to render text in the style specified by well-designed reference text effects [1], while the second one is more flexible and creative by simulating the style features from more general free-form reference style images [2].

Considering text is highly different from natural images in terms of structures, for free-form style as reference, more attention ought to be paid to match the glyph to the style during the stylization. Fig. 1b shows an example where the glyph needs careful deformation to better resemble the style subject *flames*. As the deformation increases, the text demonstrates more artistry but with the cost of legibility. Hence, there is a trade-off between legibility and artistry. However, such a subtle balance is subjective and difficult to achieve automatically. Therefore, providing users with a user-friendly tool to adjust the stylistic degree of the glyph is of great application value. Furthermore, to obtain desired effects, users are inclined to try various settings before making the final selection, thus an immediate response to the scale adjustment is expected.

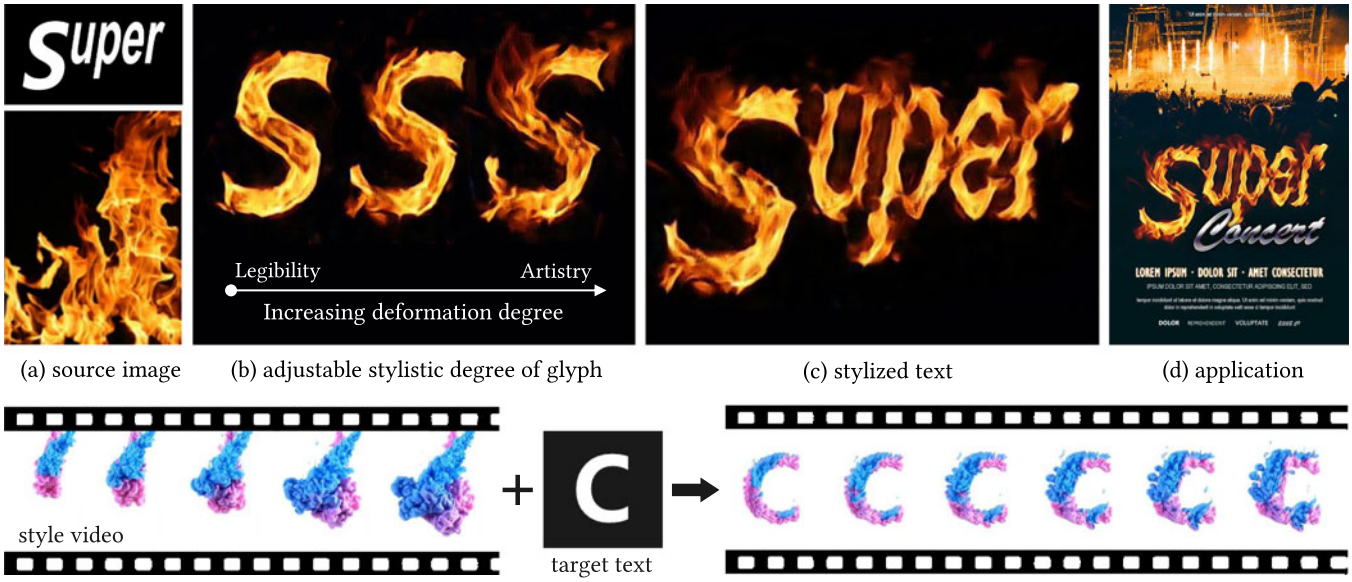
- *Shuai Yang and Jiaying Liu are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China. E-mail: {williamyang, liujiaying}@pku.edu.cn.*
- *Zhangyang Wang is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA. E-mail: atlaswang@utexas.edu.*

Manuscript received 13 May 2020; revised 9 Nov. 2020; accepted 19 Jan. 2021. Date of publication 28 Jan. 2021; date of current version 3 June 2022. (Corresponding author: Jiaying Liu.)
Recommended for acceptance by E. Shechtman.
Digital Object Identifier no. 10.1109/TPAMI.2021.3055211

In search for an efficient solution to style scale control, namely, *fast scale-controllable style transfer*, recent researches have proposed to train feed-forward networks to adjust texture scales such as the texture strength [3] and the size of texture patterns [4]. However, the *real-time control of glyph deformations* has been less investigated, which is essential for text stylization.

This practical requirement motivates our work to explore a new problem of fast controllable artistic text style transfer from a single style image/video. As illustrated in Figs. 1b and 1c, we focus on the efficient stylistic degree control in terms of the crucial glyph deformation, which allows users to select the artistic text of the best visual quality by navigating across different rendered results. Our problem has two challenges. First, different from the aforementioned well-defined texture scales that can be directly modeled by hyper-parameters, glyph deformation is subjective and not clearly defined. How to parameterize it remains an open question. Second, each style usually has only one available image/video for reference, lacking large-scale paired datasets to provide mappings between the text and its stylized versions under various deformation degrees. Thus we cannot directly learn such multi-scale glyph deformation using popular data-driven models.

In this paper, we develop a novel Shape-Matching GAN to meet the aforementioned challenges. The key idea is to model the glyph deformation as the shape mappings of the style image/video between the coarse level and the fine level, and manipulate the deformation degree with the coarse level. We show that such mappings can be robustly established by the proposed bidirectional shape matching framework with backward and forward transfers. Specifically, we first build a sketch module to forward simplify the style image to match the glyph features in different coarse levels. The obtained coarse-fine image pairs offer effective



(a) source image (b) adjustable stylistic degree of glyph (c) stylized text (d) application

(e) dynamic artistic text style transfer

Fig. 1. Illustration of the dynamic text style transfer with glyph stylistic degree control. We propose a novel Shape-Matching GAN++ to render artistic text based on reference (a) style images or (e) style videos, and allow users to (b) control the glyph deformation in a fast and continuous manner to effectively adjust the stylistic degree and select the most desired one. Our network provides users with a practical tool for (d) poster design and (e) artistic text animation rendering.

mappings to train the date-driven Shape-Matching GAN. Then, we propose a scale-aware Controllable ResBlock to equip our network with the ability to simultaneously characterize the style features of various scales. Finally, scale-controllable style transfer is achieved by forward transferring the learned style features of an arbitrary specified scale.

Compared with our previous work [5], we further explore dynamic text style transfer with glyph stylistic degree control. We extend ShapeMatching GAN to dynamic text style transfer by establishing effective spatial-temporal structural mappings within consecutive frames. As shown in Fig. 1e, the improved ShapeMatching GAN++ realizes appearance and motion pattern transfer between the output and the reference dynamic styles with nice temporal consistency. In addition, comprehensive experiments are conducted to analyze the style transfer performance of the proposed model, including additional comparison results for quantitative evaluation, results for dynamic text style transfer, ablation studies to explore the submodule and parameter settings of our proposed techniques, and new application for dynamic text transition. In summary, our contributions are threefold:

- We raise a new controllable artistic text style transfer problem for efficient glyph deformation control, and design a novel bidirectional shape matching framework to resolve it.
- We propose a sketch module to simplify the style shape to match the glyph, and convert a single style image/video into paired multi-scale training data to provide robust glyph-style mappings.
- We develop Shape-Matching GAN with a scale-controllable module to stylize the text and manipulate its stylistic degree in a fast and continuous manner for flexible user customization to balance legibility and artistry.
- We present Shape-Matching GAN++ to transfer dynamic styles onto plain text, which generates artistic text animation that characterizes large-scale motion patterns while preserving temporal consistency.

The rest of this paper is organized as follows. In Section 2, we review related works in image stylization, artistic text stylization, and scale control in style transfer. Section 3 defines the fast scale-controllable style transfer problem and the dynamic text style transfer problem, and gives an overview of the proposed bidirectional shape matching framework. Sections 4 and 5 introduce the details of the proposed Shape-Matching GAN for static text style transfer and Shape-Matching GAN++ for dynamic text style transfer, respectively. In Section 6, the superiority of our method is validated via extensive experiments and comparisons with state-of-the-art style transfer methods. Finally, the conclusion of our work is presented in Section 7.

2 Related Work

2.1 Image/Video Style Transfer

For image style transfer, Gatys *et al.* [6] proposed the first deep-based method of Neural Style Transfer, where the image style was represented as the correlation between deep features in form of Gram matrix [7]. The style is transferred by matching these statistics from the output image to the style image in an iterative optimization way. To speed up the method, feed-forward StyleNet [8] was trained using the loss in [6]. In terms of style representation, besides the Gram matrix, similar statistics like means, variances [9], covariance [10] and even learnt convolution kernels [11] are explored. These global statistics are shown to be effective in modeling textures but are hard to characterize image structures. On the other hand, the style is viewed as local neural patches in [12], [13] based on Markov random fields, which can better match semantic structures for photorealistic style

transfer. Recently, with the in-depth study of Generative Adversarial Network (GAN) [14], some researches applied image-to-image translation models [15], [16] to the image style transfer task. Driven by the big data, the specialized styles such as artistic paintings [17], cartoons [18] and make-ups [19], [20] are precisely learned and convincingly transferred. In this paper, we combine the idea of the local model and GAN. The structure changes are modeled in a local manner and are learned through the powerful GAN.

To achieve temporal consistency for video style transfer, Ruder *et al.* [21] incorporated optical-flow-based temporal loss into Neural Style Transfer. In [22] and [23], the authors further used the previous stylized frame as input to constrain the feed-forward stylization process. Later, Chen *et al.* [24] proposed to warp and fuse the stylized frames in the feature domains, so that the short-term consistency can be propagated to achieve the long-term ones. Recently, Wang *et al.* [25] proposed a compound temporal regularization from two perspectives of both local jitters and global motions, which better balances the spatial and temporal performance. However, our scenario is very different from video style transfer and the aforementioned methods are not fit for our problem. In video style transfer, the input style is a static image and the content is a video, which requires that the output preserves the temporal consistency as in the content video. Our dynamic text style transfer is the opposite, where the content is a static text image and the style is a video. Our problem mainly focuses on capturing the motion patterns in the style video and transferring them onto the text.

2.2 Artistic Text Style Transfer

In [1], Yang *et al.* first put forward the problem of artistic style transfer on text, where the authors focused on the style of well-designed text effects. The text effects are characterized by image patches along with its correlated spatial information to the glyph, which helps achieve spatially consistent style transfer. Later, Azadi *et al.* [26] proposed a deep-based MC-GAN for fast text effect transfer. However, it is limited to 26 English letters of small image size. A large dataset with high-quality text effects images is built in [27] to support the training of a feature disentangling TET-GAN on more diversified glyph and styles. Wang *et al.* [28] further considered the separation, transfer, and recombination of exquisite decorations over the text effects.

Up to our best knowledge, DynTypo [29] is the most related work of our problem. It aims to animate a static text image based on a well-designed text effects video. This method exploited example-based texture synthesis technology and optimized the texture across keyframes as a whole. Although achieving good performance, DynTypo [29] required the input style to be dynamic text effects rendered on a static text, and assumed its unstylized text image is given for guidance. This strict requirement limits DynTypo's application scenarios and makes this method less competent for glyph deformations. Also, it suffered a time-consuming optimization process.

In addition to the text effects, more general texture images can also be used as the reference style. UT-Effect [2] explored artistic text style transfer with arbitrary textures, enjoying wider application scenarios. This method exploited shape synthesis [30] to deform the glyph to match the style shape. Compared with UT-Effect [2], our method additionally investigates a more challenging problem of fast and continuous control over the glyph stylistic degree.

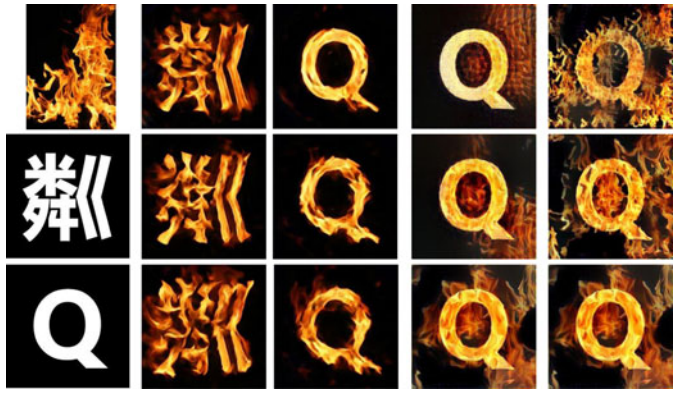


Fig. 2. Illustration of three style scales in text style transfer. (a) Style image and text image. (b) From top to bottom, our style transfer results under an increasing glyph deformation. (c) and (d) From top to bottom, stylization results by Neural Style Transfer [6] under an increasing texture strength and stroke size, respectively.

2.3 Multi-Scale Style Control

Image style in terms of textures is extensively studied in recent years, thus the research on style scale control mainly focuses on the texture. In the literature, two kinds of scales are explored. The first one is the *strength* of the texture, determining the richness and prominence of the textures over the content image (Fig. 2c). It is effectively parameterized by the weight between the style loss and content loss [6] in the mainstream neural style transfer framework. To avoid retraining the network for different weights, Babaeizadeh *et al.* [3] introduced an auxiliary network to modulate the style transfer network based on a texture strength parameter, achieving fast texture strength control. The second scale is the *stroke size* of the texture, which describes the scale of the texture patterns as illustrated in Fig. 2d and can be controlled by the input image size. It is first studied in [31], where the coarse-scale and fine-scale strokes are sequentially rendered in the downsampled and the original images. To achieve fast stroke size control, a stroke-controllable network [4] is proposed to train adaptively on images of different scale factors. In this paper, we explore a less explored but important dimension of "scale": the glyph deformation degree (Fig. 2b).

3 Problem Analysis

3.1 Multi-Scale Glyph Deformation

This section defines the multi-scale glyph deformation problem and gives an overview of our bidirectional shape matching framework. We start by stipulating two concrete goals for this new problem. First, taking the style *maple* in Fig. 12 for example, rendering leaf textures on glyph without leaf-like shapes produces unnatural results. It reveals our first goal of shape deformation and matching at all possible scales in addition to the texture transfer. Second, as

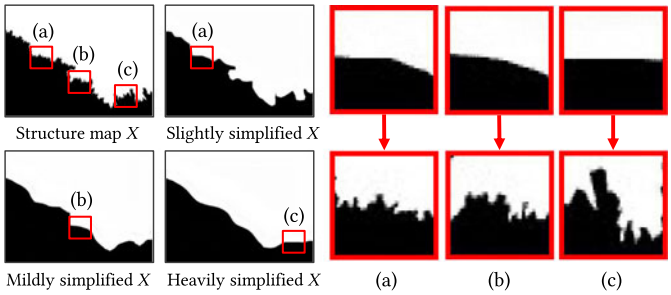


Fig. 3. Overview of bidirectional shape matching. Left: we backward simplify a leaf-like structure map into three coarse levels. Right: The resulting coarse-to-fine image pairs constitute forward shape mappings in (a) slight, (b) moderate, and (c) heavy deformation degrees.

shown in Fig. 2b, the legibility of complex glyph is more susceptible to large glyph deformation [32], which means the ideal scale that balances artistry and legibility is greatly affected by the glyph, let alone the more diversified style and people's subjective evaluation. Hence, users would prefer navigating across the possible scale space rather than retraining the model for each scale. To sum up, a controllable text style transfer should satisfy:

- *Artistry*: The stylized text should imitate the shape characteristics of the style reference, at any scale.
- *Controllability*: The glyph deformation degree needs to be adjusted in a quick and continuous way.

These two goals make our problem distinguished from previous multi-scale style control problems, which either do not deal with the shape deformation [3], [4] as in Figs. 2c and 2d, or are unable to control it efficiently [2].

To solve this problem, we proposed a novel bidirectional shape matching strategy, whose key idea is displayed in Fig. 3. We first backward simplify the reference structure map into various coarse levels to match the glyph. Then the forward style transfer is achieved by learning the obtained coarse-to-fine shape mappings that characterize the reference structural features. In Figs. 3a, 3b, and 3c, under coarser level, the similar horizontal strokes are mapped to more irregular shapes, thus learning greater glyph deformation. By doing so, *Artistry* is achieved because input shapes are all mapped to the original fine-level stylish shapes. Meanwhile, these mappings could be learned using a single feed-forward network to meet *Controllability*.

In summary, this paper formulates the new scale-controllable glyph deformation problem as learning *the function to map the style image from different coarse levels back to itself in a fast feed-forward way*. Given the framework, we still have two technical obstacles to clear. First, a reasonable method to simplify the shape needs to be figured out so that the acquired mappings are well applied to the text images. Second, how to prevent model collapse in learning the aforementioned complicated mappings on only a single style image is to be explored. Section 4 will explain our network design to meet these challenges.

3.2 Dynamic Text Style Transfer

Classic video style transfer stylizes a target video based on one static style image, which focuses on reducing the discrepancy of corresponding pixels between frames. Our dynamic text style transfer aims to animate a static text image based on dynamic styles such as the dancing flames and flowing liquid, which deals with the modeling of the motion patterns in the style. The two have completely different research focuses. And it is not straightforward to exploit commonly used video style transfer techniques such as optical flows to solve our problem.

Our solution is to model the motion patterns through short-term shape matchings. Instead of dealing with the long-term motion patterns of the entire video, we focus on the short-term motion patterns of short video clips. We define the short-term motion pattern as the shape changes between the front and back frames within a video clip. Then it can be naturally modeled as the shape mappings from the previous frames to the last frame, i.e., frame prediction. Finally, by repeatedly predicting the next frame based on previous frames, the short-term motion patterns can be propagated to achieve long-term motion patterns. Section 5 will detail our frame prediction network.

4 Shape-Matching GAN for Static Text Style Transfer

The controllable static text style transfer studies the problem of developing a feed-forward Shape-Matching GAN G to synthesize artistic text, whose deformation degree is controlled by a parameter $\ell \in [0, 1]$ and is positively related to ℓ. We further disentangle the stylization procedure into sequent structure transfer and texture transfer steps, and model them using generators G_S and G_T, respectively. Then we have $G = G_T \circ G_S$. As we will show later in Section 6.4, such disentanglement helps two generators better focus on their own tasks to boost the overall performance. Let I and Y be the target text image and reference style image, respectively, and the stylization process is formulated as

$$I_\ell^Y = G_T(G_S(I, \ell)), \quad I_\ell^Y \sim p(I_\ell^Y | I, Y, \ell), \tag{1}$$

where the target statistic $p(I_\ell^Y)$ of the stylized image I_ℓ^Y is characterized by the text image I, the style image Y, and the controllable parameter ℓ.

As analyzed in Section 3, we realize text style transfer through a novel bidirectional shape matching strategy. Let X denote the structure map of Y to indicate the shape of its style subject, which can be readily acquired through existing image matting methods or image editing tools like Photoshop. During backward structure transfer, X is simplified to a coarse version with the shape style of the glyph and coarse level ℓ, which we refer to as the sketch structure map $\tilde{X}_\ell$. $\{\tilde{X}_\ell, X\}$ forms a training pair for G_S. Then, during forward structure transfer, G_S learns the shape characteristics of X under various deformation degrees from the mappings between $\tilde{X}_\ell$ and X. Fig. 4 illustrates our overall framework with G_S and G_T:

- *Glyph Network* G_S: It learns the mapping from $\tilde{X}_\ell$ under deformation degree ℓ to X in the training phase. During testing, it transfers the structure style of X onto I, yielding the structure transfer result I_ℓ^X.
- *Texture Network* G_T: It learns the mappings from the structure map X to the style image Y in the training

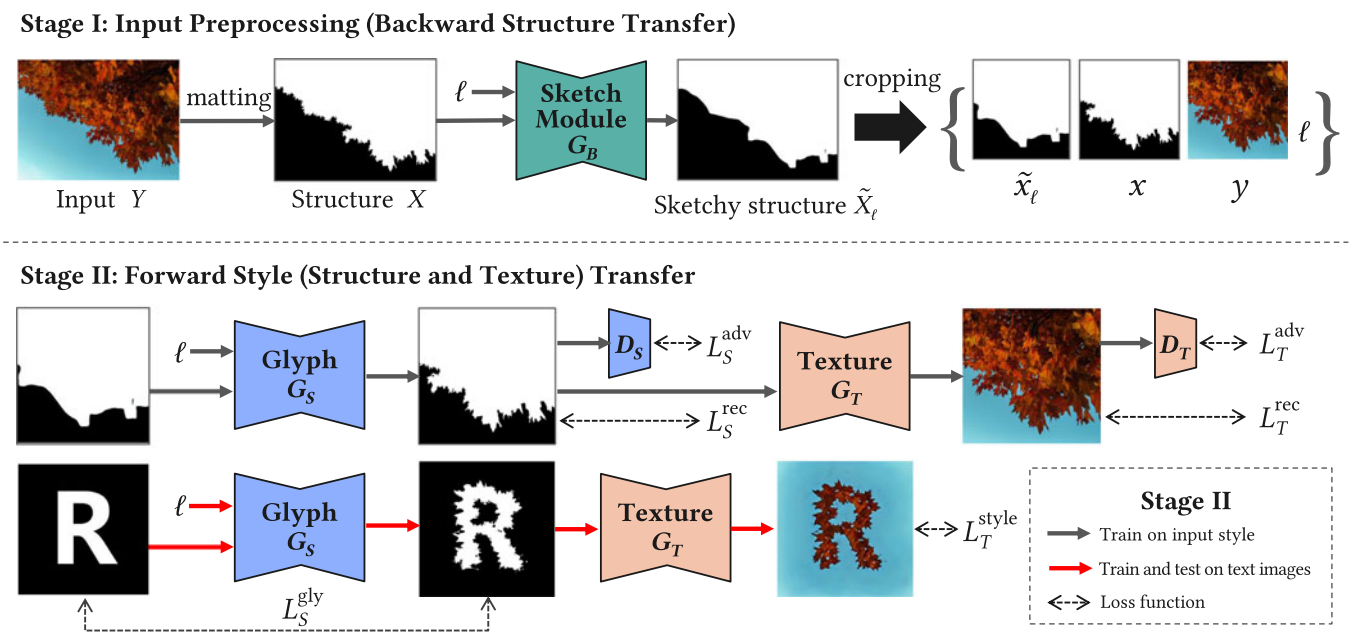


Fig. 4. Framework of shape-matching GAN.

phase. During testing, it transfers the texture style of Y onto I_ℓ^X to produce the final result I_ℓ^Y.

We couple our generators with discriminators D_S and D_T to introduce the adversarial loss to enhance the quality of stylized images. Then in Section 4.1, we are going to introduce the details of the proposed bidirectional shape matching to train the structure transfer network G_S. Section 4.2 next details the texture transfer network G_T.

4.1 Bidirectional Structure Transfer (G_S)

Backward Structure Transfer. To simply X in various coarse levels to match the glyph characteristics, we design a sketch module G_B. Fig. 5a shows an overview of G_B containing a smoothness block and a transformation block. Motivated by the multi-scale image simplification via Gaussian scale-space representation [33], [34], we build our smoothness block as a fixed convolutional layer with Gaussian kernel and control its standard deviation by ℓ as $\sigma = 16\ell + 8$. Then, the smoothness block blurs the text image and X, mapping them into a shared smooth domain, where all shapes have similar blurry contours. Finally, the transformation block conditioned by ℓ via label concatenation is trained to restore the text image from its smoothed version so that it learns to capture the glyph characteristics. By using the smooth domain as a bridge between the source style domain and target glyph domain, we can transfer the glyph characteristics onto X by feeding the smoothed X into the transformation block. The advantages are twofold. First, the coarse level is naturally parameterized by σ, in other words, the deformation degree is thus controlled by ℓ; and second, only text images, which are easy to collect, are required to train G_B. Once trained, we can apply G_B to arbitrary styles.

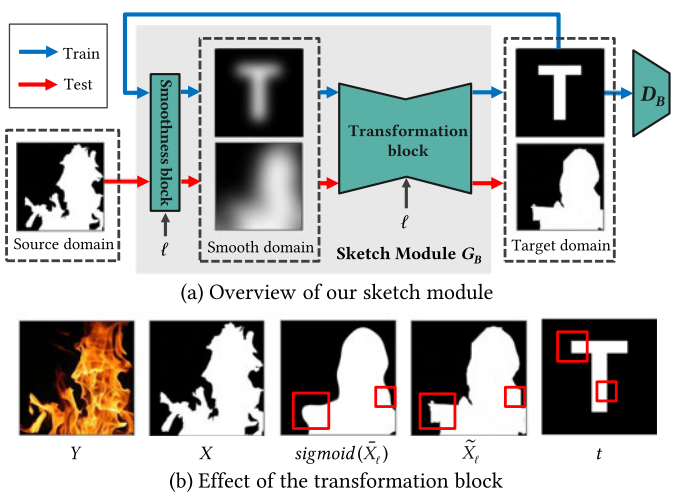


Fig. 5. Illustration of sketch module G_B for backward structure transfer.

During the training of G_B, text images t are sampled from the TE141K dataset [35] with parameter ℓ sampled within [0, 1]. G_B is tasked to restore t using L_1 loss

$$\mathcal{L}_B^{\text{rec}} = \mathbb{E}_{t,\ell}[\|G_B(t,\ell) - t\|_1]. \tag{2}$$

A conditional adversarial loss is further imposed to improve the quality of the reconstructed image

$$\begin{aligned}\mathcal{L}_B^{\text{adv}} = \; & \mathbb{E}_{t,\ell}[\log D_B(t,\ell,\bar{t}_\ell)] \\ & + \; \mathbb{E}_{t,\ell}[\log(1 - D_B(G_B(t,\ell),\ell,\bar{t}_\ell))],\end{aligned} \tag{3}$$

where D_B learns to discriminate generated images from real images given the parameter ℓ and smoothed image $\bar{t}_\ell$ as conditions. Hence, the total loss function is defined as

$$\min_{G_B} \max_{D_B} \lambda_B^{\text{adv}} \mathcal{L}_B^{\text{adv}} + \lambda_B^{\text{rec}} \mathcal{L}_B^{\text{rec}}. \tag{4}$$

Once trained, we can finally obtain the sketchy shape of X at various levels ℓ as $\tilde{X}_\ell = G_B(X,\ell)$. Fig. 5b shows an example of $\tilde{X}_\ell$, which is compared with a naïve thresholded Gaussian simplification result $sigmoid(\bar{X}_\ell)$ by substituting a sigmoid layer for the transformation block. It can be seen that the result of our sketch module better matches the

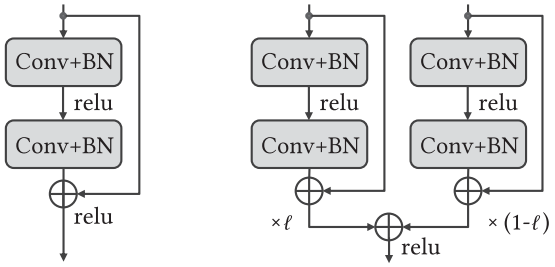


Fig. 6. Illustration of controllable ResBlock.

strokes of the text in the red boxes, which offers more accurate mappings for the glyph network.

Forward Structure Transfer. Given X and $\{\tilde{X}_\ell\}$ under various ℓ, the glyph network G_S is then trained to learn their mappings so as to capture and transfer the shape characteristics of X onto the target text. Now we are facing the challenge of learning many-to-one mappings with only a single example X. As we will show later in Section 6.4, directly exploiting standard image-to-image translation models would easily fall into model collapse, namely, just memorizing the target X during training and producing almost identical results ignoring the conditional ℓ during testing.

To solve this issue, we put forward two strategies: *data augmentation* and *Controllable ResBlock*. First, we expand a single image into a dataset by randomly cropping X and $\tilde{X}_\ell$ into abundant sub-image pairs $\{x, \tilde{x}_\ell\}$. Second, inspired by Deep Network Interpolation [36], we design an effective scale-aware Controllable ResBlock to constitute the middle layers of G_S. As displayed in Fig. 6, Controllable ResBlock is composed of two ResBlocks [37] with the linear weighting factor ℓ. When $\ell = 1$ (0), half path of G_S is blocked and Controllable ResBlock degrades into a standard ResBlock to learn a well-defined one-to-one mapping for the greatest (tiniest) shape deformation. Meanwhile, when $\ell \in (0, 1)$, G_S learns to compromise between the two extremes. Thus G_S is effectively controlled by ℓ.

In the loss aspect, G_S aims to approach the target X and compete with the discriminator D_S

$$\mathcal{L}_S^{\text{rec}} = \mathbb{E}_{x,\ell}[\|G_S(\tilde{x}_\ell, \ell) - x\|_1], \tag{5}$$

$$\begin{aligned}\mathcal{L}_S^{\text{adv}} = {} & \mathbb{E}_x[\log D_S(x)] \\ & + \mathbb{E}_{x,\ell}[\log(1 - D_S(G_S(\tilde{x}_\ell, \ell)))].\end{aligned} \tag{6}$$

For irregular styles, text t could become nearly illegible under large deformation degree ℓ. A glyph legibility loss is further optionally imposed to preserve the trunk of t in the result $G_S(t, \ell)$. Specifically, we first compute a weighting map $W(t)$ with pixel value increasing as its distance from the text contour increases. Then $G_S(t, \ell)$ is tasked to approach t in the area far from the text contour by element-wisely multiplying with $W(t)$

$$\mathcal{L}_S^{\text{gly}} = \mathbb{E}_{t,\ell}[\|(G_S(t, \ell) - t) \otimes W(t)\|_1]. \tag{7}$$

Hence, the overall loss function of G_S is

$$\min_{G_S} \max_{D_S} \lambda_S^{\text{adv}} \mathcal{L}_S^{\text{adv}} + \lambda_S^{\text{rec}} \mathcal{L}_S^{\text{rec}} + \lambda_S^{\text{gly}} \mathcal{L}_S^{\text{gly}}. \tag{8}$$

4.2 Texture Transfer (G_T)

With the structure transfer result $I_\ell^X = G_S(I, \ell)$, we formulate texture transfer as an image analogy problem such that $X : Y :: I_\ell^X : I_\ell^Y$ [38]. Considering that existing models to solve this problem such as Image Analogy [38] and Neural Doodle [39] are mainly based on less efficient optimization, we directly train a feed-forward texture transfer network G_T to build a fast end-to-end Shape-Matching GAN. Given image pairs $\{x, y\}$ randomly cropped from X and Y, G_T is trained to map x to y with the reconstruction loss and conditional adversarial loss

$$\mathcal{L}_T^{\text{rec}} = \mathbb{E}_{x,y}[\|G_T(x) - y\|_1], \tag{9}$$

$$\begin{aligned}\mathcal{L}_T^{\text{adv}} = {} & \mathbb{E}_{x,y}[\log D_T(x, y)] \\ & + \mathbb{E}_{x,y}[\log(1 - D_T(x, G_T(x)))].\end{aligned} \tag{10}$$

To further promote the overall performance on real text images, we sample text images t and improve the style similarity between $G_T(G_S(t, \ell))$ and X using the style loss $\mathcal{L}_T^{\text{style}}$ introduced in Neural Style Transfer [6]. Thus, the final loss function for texture transfer is

$$\min_{G_T} \max_{D_T} \lambda_T^{\text{adv}} \mathcal{L}_T^{\text{adv}} + \lambda_T^{\text{rec}} \mathcal{L}_T^{\text{rec}} + \lambda_T^{\text{style}} \mathcal{L}_T^{\text{style}}. \tag{11}$$

5 Shape-Matching GAN++ for Dynamic Text Style Transfer

In dynamic text style transfer, we are given a style video $\mathbf{Y} = \{Y^i | i = 1, 2, \ldots, T_Y\}$ containing T_Y consecutive frames for style reference and a text image I for content reference. We study the problem of rendering dynamic artistic text $\mathbf{I}_\ell^{\mathbf{Y}} = \{I_\ell^{\mathbf{Y},i} | i = 1, 2, \ldots, T\}$ that characterizes both spatial structure/texture features and temporal dynamic features of $\mathbf{Y}$. The total frame number and the glyph deformation degree is controlled by user-specified T and ℓ, respectively.

Our solution is to repeatedly predict the next frame according to a few previously generated frames. Let N be the number of previous reference frames required to synthesize the next frame. Let $\mathbf{I}^{a:b}$ denote the subset of $\mathbf{I}$ with indexes $i \in [a, b]$. As in Section 4, we decompose the style transfer process into structure transfer and texture transfer, and the latter is still modeled by G_T. For structure transfer, we use a new glyph network G_S^{pre} to predict the ith structure frame $I_\ell^{\mathbf{X},i}$ based on its previous N structure frames $\mathbf{I}_\ell^{\mathbf{X},i-N:i-1}$. In the beginning, there are no readily stylized frames for G_S^{pre}. Thus, we incorporate the original glyph network for static style to generate the first N structure frames. This glyph network is denoted as G_S^{ini} for frame initialization in Shape-Matching GAN++. Therefore, the proposed Shape-Matching GAN++ is built upon three main components, and Fig. 7 illustrates the framework

- *Frame Initialization Glyph Network* G_S^{ini}: It learns the mappings from $\tilde{X}_\ell^i$ under deformation degree ℓ to X^i in the training phase. During testing, it transfers the structure style of $\mathbf{X}$ onto I repeatedly to obtain the initial N structurally stylized frames.
- *Frame Prediction Glyph Network* G_S^{pre}: It learns to map $\tilde{X}_\ell^i$ with deformation degree ℓ and the previous N

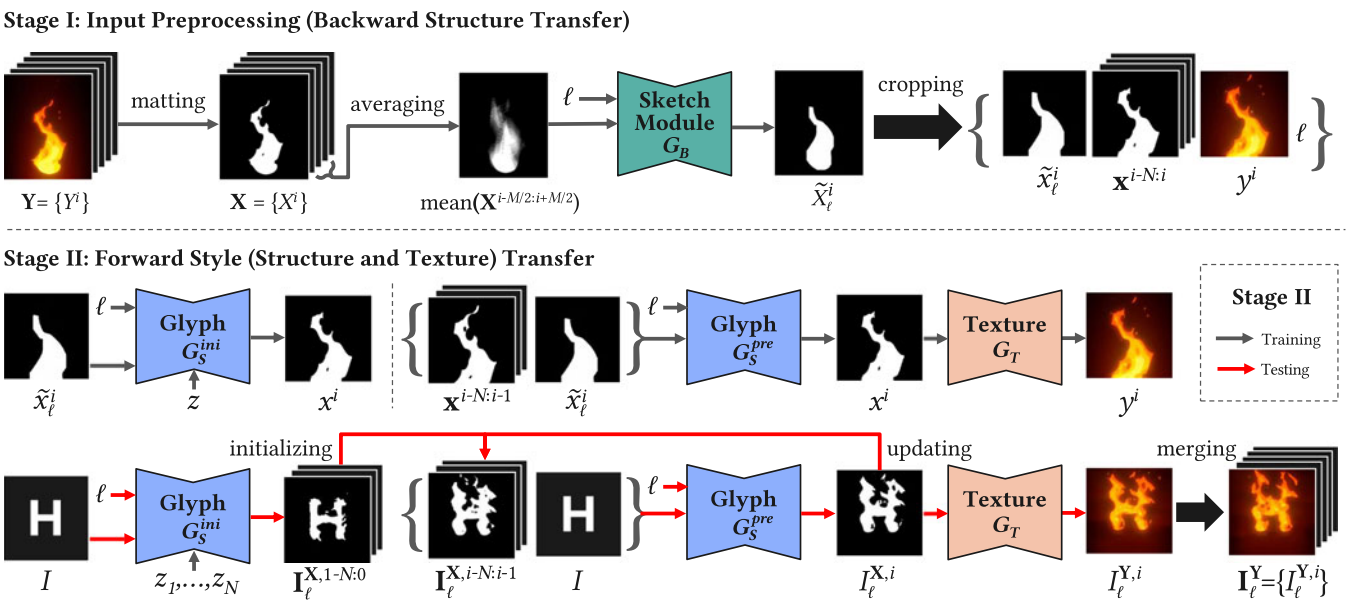


Fig. 7. Framework of Shape-Matching GAN++. For simplicity, we omit the discriminators and loss functions.

frames $\mathbf{X}^{i-N:i-1}$ to the current frame X^i during training. In testing, it repeatedly predicts the next frame $I_\ell^{\mathbf{X},i}$ based on the target text image I and the previously synthesized N frames $\mathbf{I}_\ell^{\mathbf{X},i-N:i-1}$.

- *Texture Network* G_T: It learns the mappings from the structure map X^i to the style image Y^i in the training phase. During testing, it transfers the texture style of $\mathbf{Y}$ onto $\mathbf{I}_\ell^{\mathbf{X}}$ to produce the final result $\mathbf{I}_\ell^{\mathbf{Y}}$.

As with Shape-Matching GAN, we randomly crop frames into abundant sub-images to augment training data. In the following, we present the details of the backward structure transfer, frame initialization, and frame prediction.

5.1 Backward Structure Transfer With Frame Fusion

In static text style transfer, a sketch module is designed to simplify the structure map X into different coarse levels to generate paired training data for the glyph network. For dynamic text style transfer, we can also perform backward structure transfer frame-by-frame to form $\{\mathbf{X}, \tilde{\mathbf{X}}_\ell\}$. However, we found that the simplification of our sketch module mainly focuses on the contour adjustment of the style, which means only small-scale motion patterns near the contours are characterized. Key global motion patterns such as the whole flame's left and right swing cannot be transferred.

To tackle this issue, we propose to apply frame fusion to further simplify the structure map in a more global way. Specifically, for X^i and ℓ, we first take the mean of M frames around X^i to obtain the fused frame $mean(\mathbf{X}^{i-M/2:i+M/2})$. Hence, the unique global motion of X^i is neutralized. Then the sketchy shape is calculated as $\tilde{X}_\ell^i = G_B(mean(\mathbf{X}^{i-M/2:i+M/2}), \ell)$. Intuitively, a large M means large structural changes. Thus we can associate M and ℓ as $M = 1 + \lfloor m\ell \rfloor$, where $\lfloor\rfloor$ is the round down operator, and m is the maximum allowable frame number for fusion. In this way, Shape-Matching GAN++ will still adjust local contours for small ℓ, but will pay more attention to the global structural adjustment to fit the motion patterns for large ℓ.

5.2 Frame Initialization Glyph Network

The frame initialization glyph network G_S^{ini} follows the training of G_S in Section 4.1. The only difference is that the training data are sampled from all frames $\{\mathbf{X}, \tilde{\mathbf{X}}_\ell\}$ rather than a single image pair.

To generate initial N diverse frames with temporal consistency, we propose to incorporate random noises into Shape-Matching GAN++ to diversify its output and interpolate frames through noise interpolation. Specifically, Gaussian noises are added onto the input of G_S^{ini}. In addition, inspired by StyleGAN [40], Gaussian noises are also fed into the Controllable ResBlocks through AdaIN [9]. This strategy empowers our network to generate different results according to the sampled noise during testing. Another advantage is that the structure map contains many saturated areas, adding noises can alleviate the ambiguity problem. Then, we sample two noises, which are interpolated and fed into the network to generate N initial frames. The frame initialization is summarized in Algorithm 1.

We will show later that although achieving temporal consistency, trained on independent frames, G_S^{ini} cannot depict accurate motion patterns. Thus these initial frames are not included in the final output $\mathbf{I}_\ell^{\mathbf{X}}$. They are only used for the prediction of the first N frames.

5.3 Frame Prediction Glyph Network

G_S^{pre} shares similar network architecture as G_S^{ini} except that it receives additional reference frames as input. To train G_S^{pre}, $\mathbf{X}^{i-N:i}$ and $\tilde{X}_\ell^i$ are first sampled from $\{\mathbf{X}, \tilde{\mathbf{X}}_\ell\}$. Then they are cropped into sub-image triplet $\{x^i, \mathbf{x}^i, \tilde{x}_\ell^i\}$ to gather as a training set, where we use $\mathbf{x}^i$ to refer to the reference frames $\mathbf{x}^{i-N:i-1}$ concisely. G_S^{pre} is trained using the reconstruction loss and conditional adversarial loss

$$\mathcal{L}_S^{\text{rec}} = \mathbb{E}_{x,\ell,i}[\|G_S^{pre}(\tilde{x}_\ell^i, \mathbf{x}^i, \ell) - x^i\|_1], \tag{12}$$

$$\begin{aligned}\mathcal{L}_S^{\text{adv}} = \ & \mathbb{E}_{x,i}[\log D_S^{pre}(x^i, \mathbf{x}^i)] \\ & + \mathbb{E}_{x,\ell,i}[\log(1 - D_S^{pre}(G_S^{pre}(\tilde{x}_\ell^i, \mathbf{x}^i, \ell), \mathbf{x}^i))].\end{aligned} \tag{13}$$

Fig. 8. Comparison with state-of-the-art methods on four styles of *fire*, *maple*, *lightning*, *water*. (a) Input style and its structure map. (b) Target text. (c) Image Analogy [38]. (d) Neural Style Transfer [6] with spatial control [31]. (e) Neural Doodle [39]. (f) T-Effect [1]. (g) UT-Effect [2]. (h) Results of the proposed Shape-Matching GAN. For UT-Effect [2] and Shape-Matching GAN, the deformation degrees are manually selected.

In the testing phase, the frame prediction process is summarized in Algorithm 1.

Algorithm 1. Dynamic Text Style Transfer

Input: Text image I, frame number T, deformation degree ℓ
Output: Stylized text frames $\mathbf{I}_\ell^{\mathbf{Y}} = \{I_\ell^{\mathbf{Y},i} | i = 1, 2, \ldots, T\}$

$\triangle$ *Initial structure frame synthesis*:
sample two random noises z_1 and z_N
for $i = 1 \rightarrow N$ **do**
  $z_i = ((N-i) * z_1 + (i-1) * z_N)/(N-1)$
  $I_\ell^{\mathbf{X},i-N} = G_S^{ini}(I, \ell, z_i)$
end for
for $i = 1 \rightarrow T$ **do**
  $\triangle$ *Structure frame prediction*:
  $I_\ell^{\mathbf{X},i} = G_S^{pre}(I, \mathbf{I}_\ell^{\mathbf{X},i-N:i-1}, \ell)$
  $\triangle$ *Texture transfer*:
  $I_\ell^{\mathbf{Y},i} = G_T(I_\ell^{\mathbf{X},i})$
end for

6 Experimental Results

6.1 Implementation Details

Network Architecture. Our generators are built upon the StyleNet [8] with ResBlocks as middle layers, except that G_S utilizes the proposed Controllable ResBlock instead. The patch-based discriminators introduced in pix2pix [15] is used to better preserve image details. The architecture details are provided in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3055211.

Network Training. The style image/video is randomly cropped to 256×256 to constitute the training set. We adopt Adam optimizer and use a learning rate of 0.0002. During training our Controllable ResBlocks in the three glyph networks, we first fix $\ell = 1$ to make the network learn the greatest deformation. The parameters of the trained half part in Controllable ResBlocks are copied to the other half part. Then, we sample $\ell \in \{0, 1\}$ to train the network on the greatest and the tiniest deformation. Finally, the network is fine-tuned on $\ell \in \{i/K\}_{i=0,\ldots,K}$. We use $K = 3$ on static style and $K = 4$ on dynamic style. The number of reference frames is $N = 5$. Unless stated otherwise, the maximum allowable number of frame fusion is $m = 16$. For all experiments, we set $\lambda_B^{\text{rec}} = \lambda_S^{\text{rec}} = \lambda_T^{\text{rec}} = 100$, $\lambda_B^{\text{adv}} = \lambda_T^{\text{adv}} = 1$. λ_S^{adv} is set to 0.1 and 1 for static style and dynamic style, respectively. We manually choose λ_S^{gly} from $\{0, 1\}$ and λ_T^{style} from $\{0, 0.01\}$ based on the style types for better performance.

6.2 Comparisons With State-of-the-Art Methods

Static Text Style Transfer. We first qualitatively and quantitatively analyze the performance of Shape-Matching GAN on static text style transfer through comparative experiment. Image Analogy [38], Neural Style Transfer [31], Doodle [39], T-Effect [1], and UT-Effect [2] are selected for comparison. For unsupervised Neural Style Transfer and UT-Effect, we adapt them to a supervised manner by directly feeding our extracted structure map to these methods for spatial control [31] or structure transfer [2]. By doing so, all the methods follow a one-shot supervised stylization paradigm that transfers styles based on one style image and its unstylish counterpart for a fair comparison. Fig. 8 shows the visual comparison results on four styles. Please refer to the supplementary material for full results on eighteen styles, available online.

As illustrated in Figs. 8c and 8f, Image Analogy [38] and T-Effect [1] only transfer textures without adjusting

TABLE 1
User Preference Ratio of Image Analogy [38], Neural Style Transfer [6], Doodle [39], T-Effect [1], UT-Effect [2], and Shape-Matching GAN on Eighteen Different Static Styles

Style	[38]	[6]	[39]	[1]	[2]	Ours
fire	0.30	0.54	0.48	0.30	**0.70**	0.68
maple	0.26	0.40	0.72	0.06	0.64	**0.92**
smoke	0.56	0.18	0.44	0.46	0.64	**0.72**
water	0.70	0.16	0.44	0.40	0.44	**0.86**
sketch	**0.76**	0.38	0.22	0.26	0.62	**0.76**
lightning	0.68	0.24	0.36	0.42	0.48	**0.82**
ink	0.42	**0.82**	0.38	0.14	0.54	0.70
sakura	0.48	0.56	0.70	0.04	0.48	**0.74**
cloud	0.38	0.22	0.76	0.20	0.66	**0.78**
water2	0.78	0.56	0.26	0.32	0.24	**0.84**
island	0.36	0.56	0.64	0.02	0.56	**0.86**
flower	0.52	0.26	0.64	0.24	0.54	**0.80**
flower2	0.50	0.54	0.56	0.22	0.46	**0.72**
fissure	0.74	0.06	0.50	0.36	0.52	**0.82**
ivy	0.62	0.42	0.52	0.26	0.44	**0.74**
snowflake	0.36	0.18	0.66	0.08	0.78	**0.94**
rime	0.42	0.28	0.68	0.06	0.62	**0.94**
wall	0.40	0.40	0.70	0.30	0.40	**0.80**
Average	0.513	0.376	0.537	0.230	0.542	**0.802**

For each row, we show the best preference ratio in bold and the second underlined.

the text contour, thus generating rigid and unnatural results. Meanwhile, the deep-based methods of Neural Style Transfer [31] and Doodle [39] model image styles as deep features, which implicitly characterize both the shape and texture patterns. Therefore, they can transfer both structure and texture styles. But they often excessively distort the text and produce color deviations and checkerboard artifacts, which makes the results less legible as shown in the third row. UT-Effect [2] builds upon a patch-based structure transfer algorithm to match the glyph to style. However, patching matching and blending are not robust enough, and some structural details are not fully transferred. By comparison, thanks to the proposed bidirectional shape matching strategy, our method successfully learns precise shape and texture patterns, which are vividly transferred through adversarial learning. Therefore, our method produces highly natural and visually pleasing artistic text.

To quantitatively measure the performance of the compared methods, we conducted a user study on the Amazon Mechanical Turk platform where observers were given image pairs and tasked to choose which one is of the best style similarity with the reference style image while maintaining legibility. The preference ratio is utilized as our evaluation metric. It measures the percentage of times a method is selected in all of its related selections. According to the definition, if a method performs as normal as other methods, then its mean preference ratio will be 0.5; if it is significantly better than all other methods, then its mean preference ratio can reach 1.0. The preference ratio over 18 test styles is shown in Table 1. For each style, 15 image pairs were rated by 10 observers. As shown in Table 1, our method obtains a steady preference from the users with preference ratios surpassing 0.5 in all cases. The best average preference ratio of 0.802 quantitatively verifies the superiority of our method.

Scale-Controllable Style Transfer. As the most related work that focuses on the text deformation, we further compare with UT-Effect [2] in Fig. 9. UT-Effect [2] models structure patterns as boundary patches at multiple resolutions and controls the deformation degree with the resolution level. It has four drawbacks: First, the patch blending step inevitably blurs the structure and texture details. Second, because patches are locally and greedy matched, the globally consistent stylization is not guaranteed. Third, the transformation is discontinuous due to the independent patch matching process for each scale. Finally, the iterative optimization process in UT-Effect [2] has a high computational burden. It takes UT-Effect [2] about 100 s for the 256×256 image in Fig. 9 on Intel Core i7-6500U CPU. Shape-Matching GAN, on the contrary, trained simultaneously on all possible scales, is able to fast and continuously adjust the deformation degree while preserving fine details. For the same image, our feed-forward method requires about 0.43 s and *16 ms* on Intel Xeon E5-2650 CPU and a GeForce GTX 1080 Ti GPU, respectively.

Dynamic Text Style Transfer. We study the performance of Shape-Matching GAN++ on dynamic text style transfer in

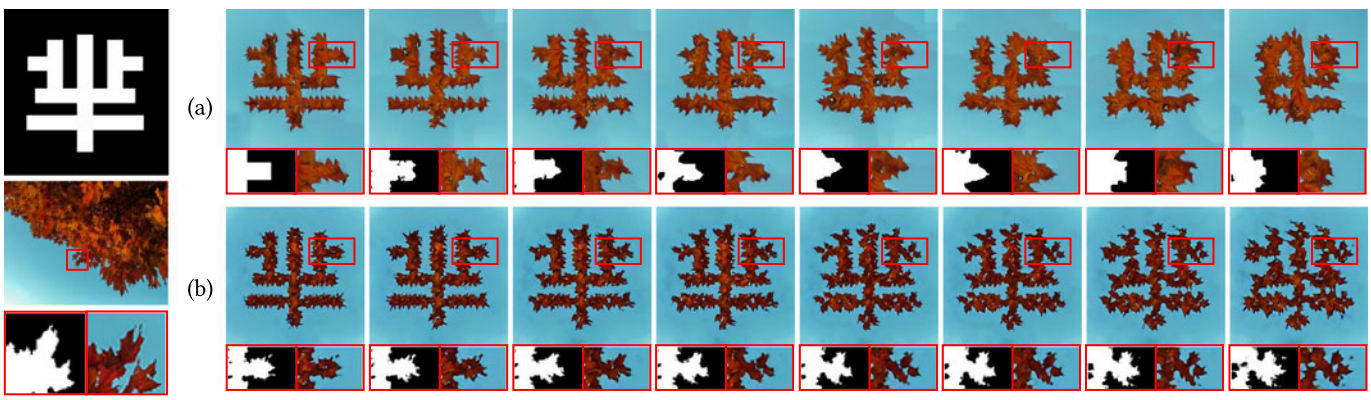


Fig. 9. Qualitative comparison between Shape-Matching GAN and UT-Effect [2]. For the first column, from top to bottom: target text, style image, the enlarged patches from the style image, and their corresponding structure maps. Remaining columns: Results by (a) UT-Effect [2] with resolution level evenly increasing from 1 to 7; (b) the proposed method with ℓ evenly increasing from 0 to 1. The red box region is shown enlarged at the bottom with the corresponding structure map for better visual comparison.

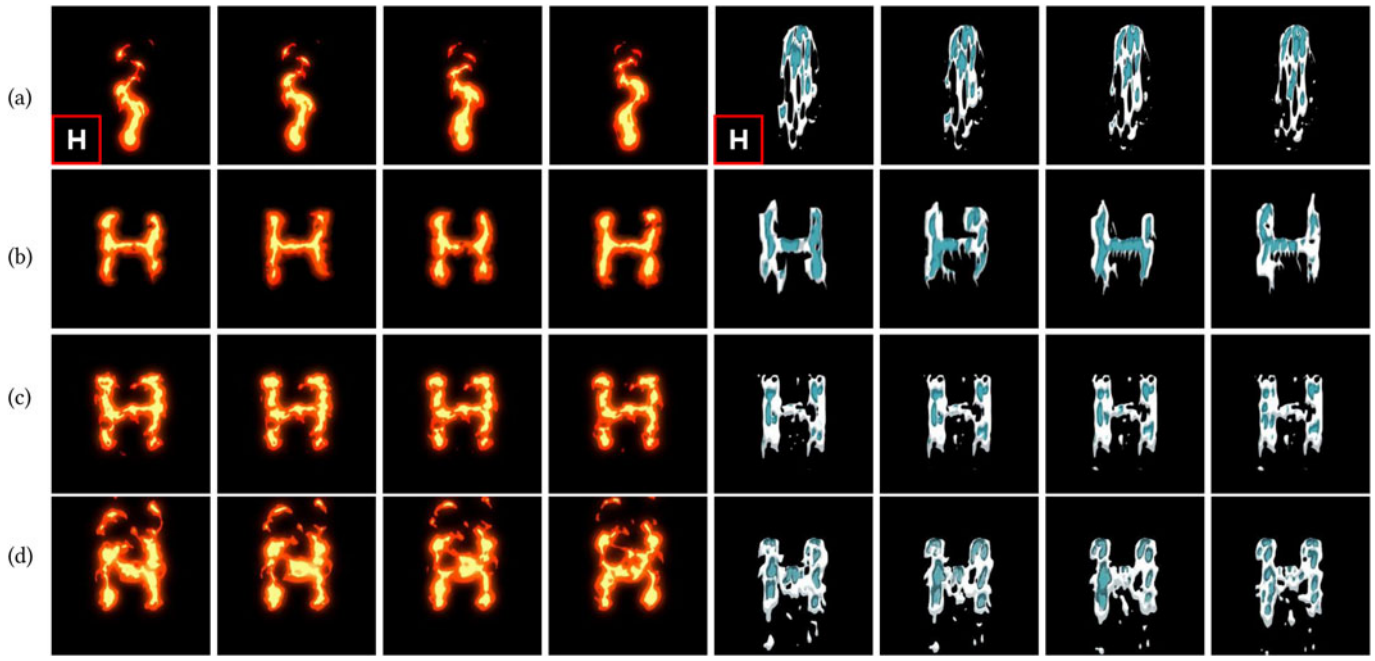


Fig. 10. Comparison with state-of-the-art methods on dynamic text style transfer. For each group: (a) Four consecutive frames of the reference style with the target text in the lower-left corner. (b) Results of UT-Effect [2]. (c) Results of Shape-Matching GAN. (d) Results of Shape-Matching GAN++.

Fig. 10. Since there is no other work exactly handling our task, we choose the most related UT-Effect [2] and Shape-Matching GAN for comparison in terms of motion pattern transfer and temporal consistency. UT-Effect [2] generates the ith frame by using the ith reference style frame. Hence it fails to preserve the temporal coherence and suffers severe flickers. For Shape-Matching GAN, we directly use the frame initialization strategy introduced in Section 5.2 to generate artistic text animation. It can be seen that noise interpolation introduces temporal consistency between adjacent frames. However, the structural motions derived from the noise changes do not reflect the real motion patterns of the reference style. By comparison, Shape-Matching GAN++ preserves realistic fluid motion patterns, while achieving satisfactory temporal consistency.

For dynamic text style transfer, a user study is conducted for quantitative evaluation. Besides structure/texture similarity and text legibility, the observers were asked to additionally consider the temporal consistency and motion pattern similarity to the reference style video. Three methods on six styles including the two styles in Fig. 10 (full results are included in the supplementary material, available online) are rated by 15 observers. Table 2 reports the preference ratio where UT-Effect [2] obtains low scores due to bad temporal consistency. Shape-Matching GAN++ is much more preferred than its previous version, indicating our frame prediction glyph network achieves better motion pattern transfer.

We further compare with DynTypo [29]. Since this method only supports text effects as style with no text deformation, we adapt Shape-Matching GAN++ to dynamic text effect transfer by fixing $\ell = 0$. The provided source text image is directly used as the sketchy structure map $\tilde{X}_\ell^i$ for all frames. Fig. 11 shows the results. In this example, we use a large sub-image size of 400×296 to better capture the shape of the flame above the text. As can be seen, the performance of our adapted Shape-Matching GAN++ is comparable to DynTypo [29]. Meanwhile, our method can handle more general styles and is superior in time efficiency. As reported in [29], for 159 frames with a size of 496×360,

TABLE 2
User Preference Ratio of UT-Effect [2], Shape-Matching GAN [5] and Shape-Matching GAN++ on Six Different Dynamic Styles

Style	[2]	[5]	Shape-Matching GAN++
fire1	0.17	0.55	**0.76**
fire2	0.33	0.50	**0.67**
water1	0.03	**0.80**	0.67
water2	0.07	0.56	**0.87**
smoke1	0.37	0.40	**0.73**
smoke2	0.27	0.30	**0.93**
Average	0.206	0.522	**0.772**

For each row, we show the best preference ratio in bold and the second underlined.

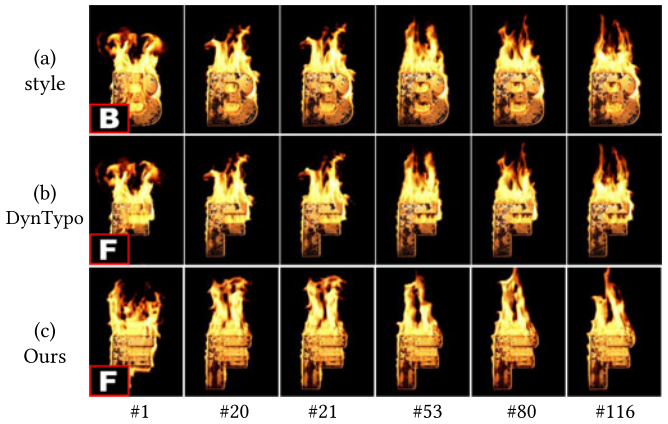


Fig. 11. Comparison with DynTypo [29] on dynamic text effect transfer.

TABLE 3
Running Time of the Proposed Method on 256×256 Sub-Images and Testing Frames

Method	Training time (per style)			Testing time (per frame)
	sketch	glyph	texture	glyph + texture
Shape-Matching GAN	0.75 h	1.71 h	0.75 h	16 ms
Shape-Matching GAN++	0.75 h	2.47 h (ini) + 2.16 h (pre)	0.75 h	48 ms

DynTypo uses about 1,200 s while the testing time of our method is only 194 s on Intel Xeon E5-2650 CPU and 15 s on a GeForce GTX 1080 Ti GPU.

6.3 Running Time

In Table 3, we report the running time of the proposed method on a GeForce GTX 1080 Ti GPU. In the training phase, we crop training images into 256×256 sub-images, thus our method is essentially invariant to practical resolutions of the style image. The training of our sketch module is independent of the style. After about 0.75 hours of training, it can be applied to arbitrary styles. For a new style image/video, it takes about 2.46 hours and 5.38 hours to train Shape-Matching GAN and Shape-Matching GAN++, respectively. In the testing phase, our feed-forward method requires less than 50 ms to process a 256×256 frame, which implies a potential of nearly real-time user interaction.

6.4 Ablation Study on Shape-Matching GAN

Network Architecture. To investigate each component of Shape-Matching GAN, the following four configurations are designed and compared:

- Baseline: The baseline model is a texture network to render textures based on the mapping between the structure map X and the style image Y.
- W/o CR: This model contains a naïve glyph network and a texture network. The naïve glyph network is controlled by ℓ via conventional label concatenation instead of the Controllable ResRlock (CR).
- W/o TN: This model contains a single glyph network without the Texture Network (TN) and is trained to directly map the sketchy structure map $\tilde{X}_\ell$ to Y.
- Full model: The proposed model with both the glyph network and the texture network.

Fig. 12 exhibits the style transfer results of these configurations. As expected, the baseline model does not consider structure transfer, thus its results have rigid and unnatural contours. The naïve glyph network learns to synthesize leaf-like contours, but conventional label concatenation is not powerful enough to characterize the challenging many-to-one mapping. Thus it generates very similar results under different ℓ. As illustrated in Fig. 12d, the proposed Controllable ResBlock effectively solves this issue: our glyph network learns accurate multi-scale structure transfer, and can even simultaneously render textures. Finally, by transferring the texture transfer task to a dedicated texture network, full Shape-Matching GAN is able to render more exquisite texture details, sharing better style consistency with the reference style.

We further compare with the closely related Deep Network Interpolation (DNI) [36] in Fig. 13 to verify the effectiveness of Controllable ResBlock. DNI trains two correlated networks on two related mappings, and interpolates network parameters to smoothly control the output's imagery effects between these two mappings, which is proven to work well in interpolating color and texture styles. However, when applying DNI to structure transfer, its interpolated result is poor as shown in the middle row of Fig. 13b. It might be because the structure deformation is much more non-linear than texture blending. Inspired by DNI, we design two paths in Controllable ResBlock to break down the challenging multi-scale mapping into two extreme one-to-one mappings to avoid model collapse. To model the challenging structure deformations, we propose to train our

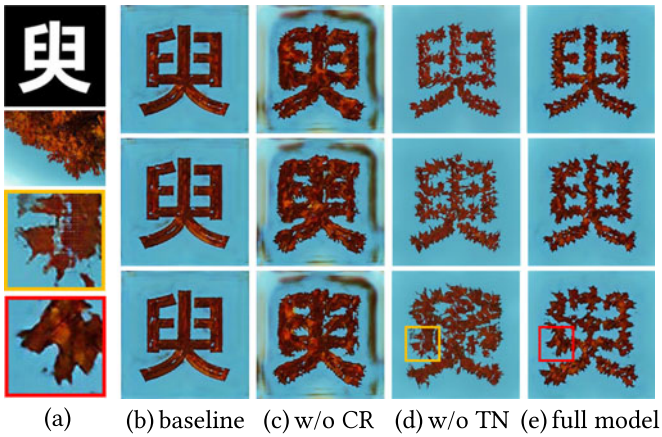


Fig. 12. Ablation study on four different network configurations. (a) From top to bottom: target glyph, reference style, the enlarged patches from (d) and (e), respectively. (b)-(e) From top to bottom: Results under $\ell = 0.0, 0.5, 1.0$, respectively.

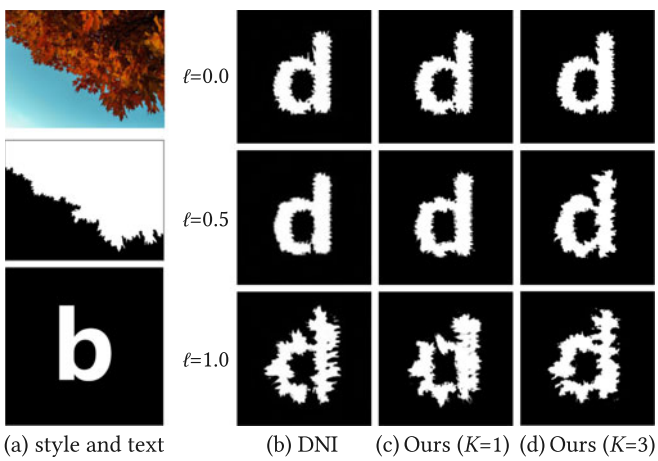


Fig. 13. Comparison with DNI [40] on multi-scale structure transfer.

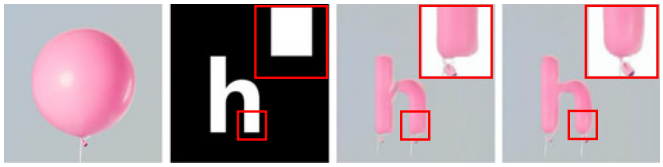


Fig. 14. Performance of the proposed sketch module. The foreground stylized text in the red box is enlarged for better visual comparison.

network with $\ell \in (0, 1)$ to compromise between the two extremes. Instead of interpolating network parameters as DNI, Controllable ResBlock interpolates feature maps, which allows easier training when $\ell \in (0, 1)$. Compared with only training on two extremes (Fig. 13c), the proposed training strategy is more effective in teaching our network to infer moderate deformations (Fig. 13d). Note that we only sample $\ell \in \{0, 1/3, 2/3, 1\}$ for training, and our network infers reasonable results of unseen $\ell = 0.5$.

Sketch Module. As analyzed in Section 4.1, the task of the sketch module G_B is to offer robust mappings between the text and style domain by simplifying the style image to match the glyph. We study the effect of G_B in Fig. 14, where the Transformation Block (TB) in G_B is replaced by a single sigmoid layer to simplify the style image but without imitating the glyph contours. Fig. 14c shows that this configuration cannot provide robust mappings, and the stylized text still has rigid contours. By contrast, our full model successfully synthesizes a rounded h-shaped balloon.

Loss Function. In Fig. 15, we investigate the performance of our glyph legibility loss (Eq. (7)) through a comparative experiment. Our method successfully renders a rigid Chinese character into trickles of wafting smoke. However, under large deformation ($\ell = 0.75$), its strokes become extremely irregular with fractures as in Fig. 15c, making the character less recognizable. This issue can be solved effectively through our glyph legibility loss by setting $\lambda_S^{\text{gly}} = 1$. As shown in Fig. 15d, the trunk region of the strokes is well preserved with other regions highly stylized, thus striking a good balance between the artistry and legibility.

6.5 Ablation Study on Shape-Matching GAN++

Number of Reference Frames. We examine the effect of the number N of reference frames in Fig. 16. In this experiment, we train Shape-Matching GAN++ on the style video *fire*. During testing, the network aims to reconstruct the video *fire* as accurately as possible based on only its first N frames and its sketchy structure map. As can be seen in Figs. 16a and 16b, a small number of reference frames cannot characterize adequate temporal motion patterns. Thus, the generator tends to rely more on the input sketchy structure map and generate less dynamic flames in the 21st and 27th

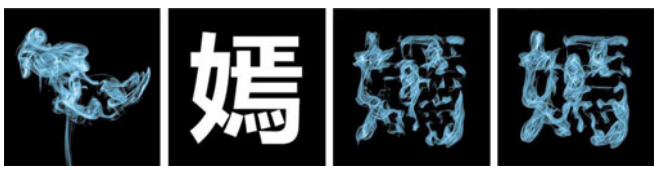


Fig. 15. Performance with and without the glyph legibility loss $\mathcal{L}_S^{\text{gly}}$.

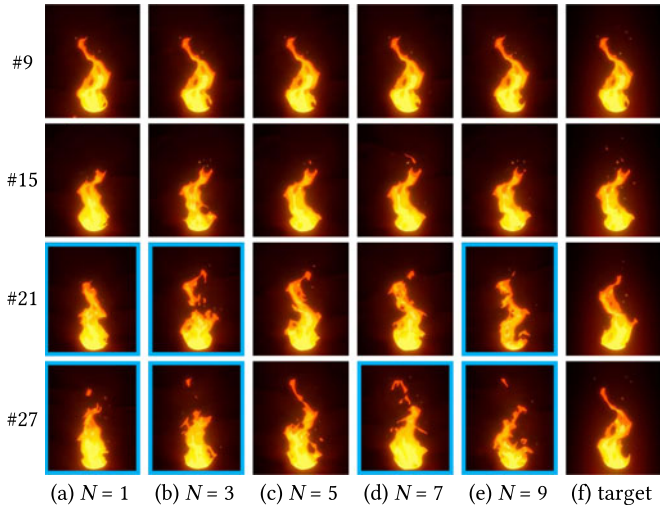


Fig. 16. Effect of the number N of reference frames. (a)-(e): Each row shows the generated ninth, 15th, 21st, and 27th frames. (f) The target style frames. Blue boxes indicate a large divergence between the generated frame and the target frame.

frames. For a large N, on the other hand, reference frames contain too much spatial-temporal structural information, making it hard to match and transfer. Thus the results gradually deviate from the reference video. The best reconstruction result is obtained with an intermediate number of 5. It is also verified that although out network is trained with only short-term temporal consistency within 5 frames, our network can effectively propagate short-term temporal consistency to achieve long-term temporal consistency.

Number of Frame Fusion. In Fig. 17, we investigate the impact of the number m of frames for fusion. Although using the maximum deformation degree $\ell = 1$, without frame fusion, i.e., $m = 1$, the structures are only transferred in a limited region near the text contour. As m increases, flames gradually emerge from the contour of the text and sway. It verifies that frame fusion could effectively guide the network to capture large-scale global motion patterns.

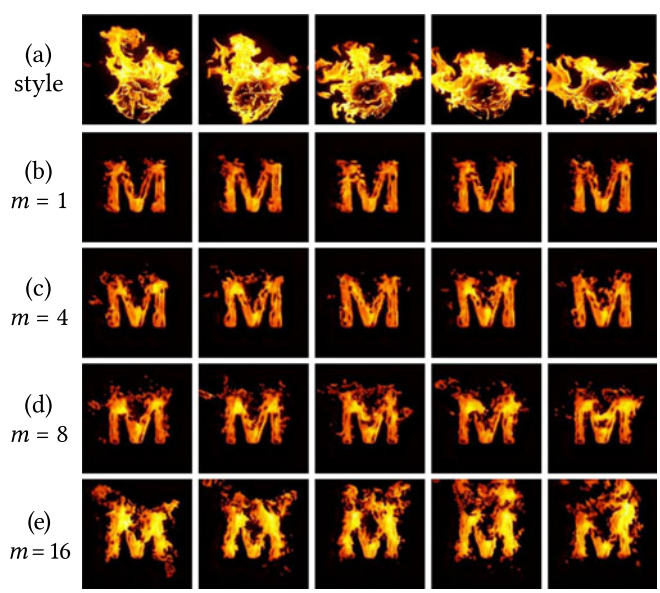


Fig. 17. Effect of the number m of frames for fusion. (a) The reference style. (b)-(e): The generated five consecutive frames.

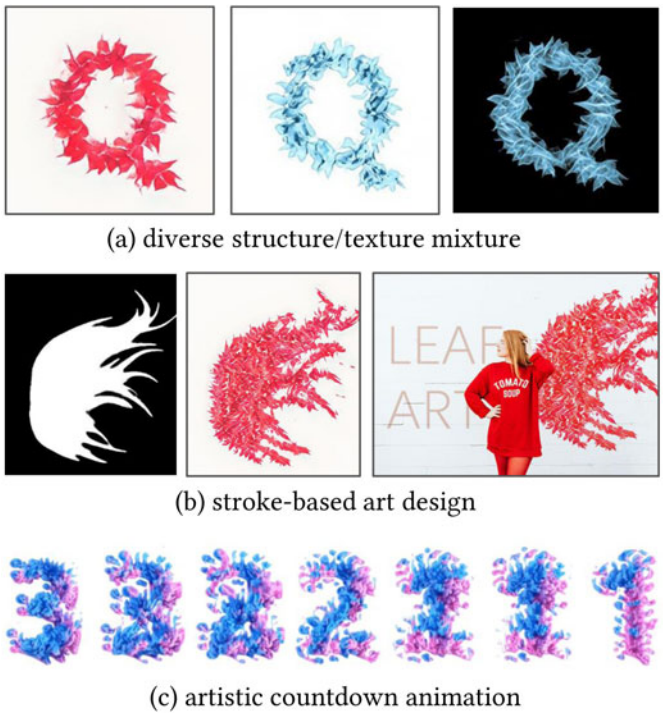


Fig. 18. Applications of the proposed method.

6.6 Applications

In this section, we demonstrate three applications of our method: structure/texture mash-up, stroke-based art design, and artistic countdown.

Structure/Texture Mash-Up. By mixingly using the glyph and texture networks trained on different styles, we can obtain a mash-up of structure style and texture style, thus creating brand-new text styles. Fig. 18a shows a standard stylization result and two mash-ups. The text in the shape feature of *maple* is rendered by the *maple*, *water*, and *smoke* textures, respectively.

Stroke-Based Art Design. Our method can extend to more general shapes such as icons and symbols without additional modifications. It is shown in Fig. 18b that our method successfully renders wings made of maple leaves based on an icon, facilitating the following graphic design.

Artistic Countdown. By interpolating between text images, Shape-Matching GAN++ can render dynamic transitions between text. We find those fluid styles such as water and smoke, are especially suitable to render the shape changes naturally. Fig. 18c presents an example of a countdown animation composed of colored inks.

7 Conclusion

In this paper, we explore a new problem of fast controllable text style transfer and propose Shape-Matching GAN++ that renders dynamic artistic text animation and enables continuous control of the stylistic degree of the glyph. The multi-scale glyph deformation task is expressed as learning a coarse-to-fine shape mapping problem and a corresponding bidirectional shape matching framework is introduced. We present a sketch module to narrow the style's structure discrepancy to the glyph and to offer robust mappings. Our model leverages the proposed Controllable ResBlock to learn the multi-scale shape mappings for effective scale control. The temporal consistency is further modeled as shape mappings within consecutive frames to achieve the transfer of motion patterns for dynamic stylization. The experimental results verify the superiority and robustness of Shape-Matching GAN++ on both glyph deformation control and dynamic text style transfer.

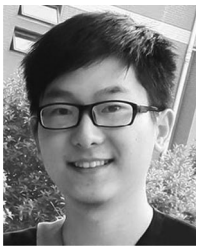
Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities and the National Natural Science Foundation of China under contract No. 61772043.

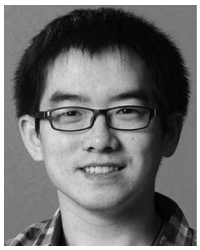
References

[1] S. Yang, J. Liu, Z. Lian, and Z. Guo, "Awesome typography: Statistics-based text effects transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7464–7473.
[2] S. Yang, J. Liu, W. Yang, and Z. Guo, "Context-aware text-based binary image stylization and synthesis," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 952–964, Feb. 2019.
[3] M. Babaeizadeh and G. Ghiasi, "Adjustable real-time style transfer," in *Proc. Int. Conf. Learn. Representations Workshop DeepGenStruct*, 2019, pp. 1–12.
[4] Y. Jing *et al.*, "Stroke controllable fast style transfer with adaptive receptive fields," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 238–254.
[5] S. Yang, Z. Wang, Z. Wang, N. Xu, J. Liu, and Z. Guo, "Controllable artistic text style transfer via shape-matching GAN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4442–4451.
[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.
[7] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 262–270.
[8] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
[9] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1510–1519.
[10] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 386–396.
[11] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1897–1906.
[12] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2479–2486.
[13] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 702–716.
[14] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[15] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.
[16] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.
[17] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time HD style transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 698–714.
[18] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9465–9474.
[19] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.
[20] H. Chang, J. Lu, F. Yu, and A. Finkelstein, "PairedcycleGAN: Asymmetric style transfer for applying and removing makeup," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 40–48.

[21] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Proc. German Conf. Pattern Recognit.*, 2016, pp. 26–36.
[22] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei, "Characterizing and improving stability in neural style transfer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4067–4076.
[23] H. Huang *et al.*, "Real-time neural style transfer for videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 783–791.
[24] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1105–1114.
[25] W. Wang, J. Xu, L. Zhang, Y. Wang, and J. Liu, "Consistent video style transfer via compound regularization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1–8.
[26] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content GAN for few-shot font style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7564–7573.
[27] S. Yang, J. Liu, W. Wang, and Z. Guo, "TET-GAN: Text effects transfer via stylization and destylization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 1238–1245.
[28] W. Wang, J. Liu, S. Yang, and Z. Guo, "Typography with decor: Intelligent text style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5889–5897.
[29] Y. Men, Z. Lian, Y. Tang, and J. Xiao, "DynTypo: Example-based dynamic text effects transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5870–5879.
[30] A. Rosenberger, D. Cohen-Or, and D. Lischinski, "Layered shape synthesis: Automatic generation of control maps for non-stationary textures," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–9, 2009.
[31] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3985–3993.
[32] Z. Wang *et al.*, "DeepFont: Identify your font from an image," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 451–459.
[33] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 1, pp. 26–33, Jan. 1986.
[34] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
[35] S. Yang, W. Wang, and J. Liu, "TE141K: Artistic text benchmark for text effect transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 1, 2020, doi: 10.1109/TPAMI.2020.2983697.
[36] X. Wang, K. Yu, C. Dong, X. Tang, and C. C. Loy, "Deep network interpolation for continuous imagery effect transition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1692–1701.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[38] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. 28th Annu. Conf. Comput. Graph. Interactive Techn.*, 2001, pp. 327–340.
[39] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," 2016, *arXiv:1603.01768*.
[40] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.



Shuai Yang (Member, IEEE) received the BS and PhD degrees (Hons.) in computer science from Peking University, Beijing, China, in 2015 and 2020, respectively. He is currently a postdoctoral research fellow with the NTU AI Corporate Laboratory, Nanyang Technological University, Singapore. He was a visiting scholar with the Texas A&M University, College Station, Texas, from September 2018 to September 2019. He was a visiting student with the National Institute of Informatics, Japan, from March 2017 to August 2017. He received the IEEE ICME 2020 Best Paper Awards and IEEE MMSP 2015 Top10% Paper Awards. His current research interests include image stylization and image inpainting.



Zhangyang Wang (Member, IEEE) received the BE degree from the University of Science and Technology of China (USTC), China, in 2012, and the PhD degree from the Electrical and Computer Engineering (ECE) Department, University of Illinois at Urbana-Champaign (UIUC), Champaign, Illinois, working with Professor Thomas S. Huang. He is currently an assistant professor of ECE at The University of Texas at Austin, Austin, Texas. He was an assistant professor of CSE with the Texas A&M University, Austin, Texas, from 2017 to 2020. His research has been addressing machine learning, computer vision and optimization problems, as well as their interdisciplinary applications. He has co-authored more than 100 papers, published two books and one chapter. He has received more than 30 research awards. He is an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*.

Jiaying Liu (Senior Member, IEEE) received the PhD degree (Hons.) in computer science from Peking University, Beijing China, 2010. She is currently an associate professor, Peking University, China, Boya Young fellow with the Wangxuan Institute of Computer Technology, Peking University, China. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 50 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. He is a senior member of CSIG and CCF. She was a visiting scholar with the University of Southern California, Los Angeles, California, from 2007 to 2008. She was a visiting researcher with Microsoft Research Asia, in 2015 supported by the Star Track Young Faculties Award. She has served as a member of Multimedia Systems and Applications Technical Committee (MSA TC), and Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society. She received the IEEE ICME-2020 best paper awards and IEEE MMSP-2015 Top10% Paper Awards. She has also served as the associate editor of the *IEEE Transaction on Image Processing*, the *IEEE Transaction on Circuit System for Video Technology* and *Journal of Visual Communication and Image Representation*, the technical program chair of IEEE ICME-2021/ACM ICMR-2021, the publicity chair of IEEE ICME-2020/ICIP-2019, and the area chair of CVPR-2021/ECCV-2020/ICCV-2019. She was the APSIPA distinguished lecturer (2016-2017).