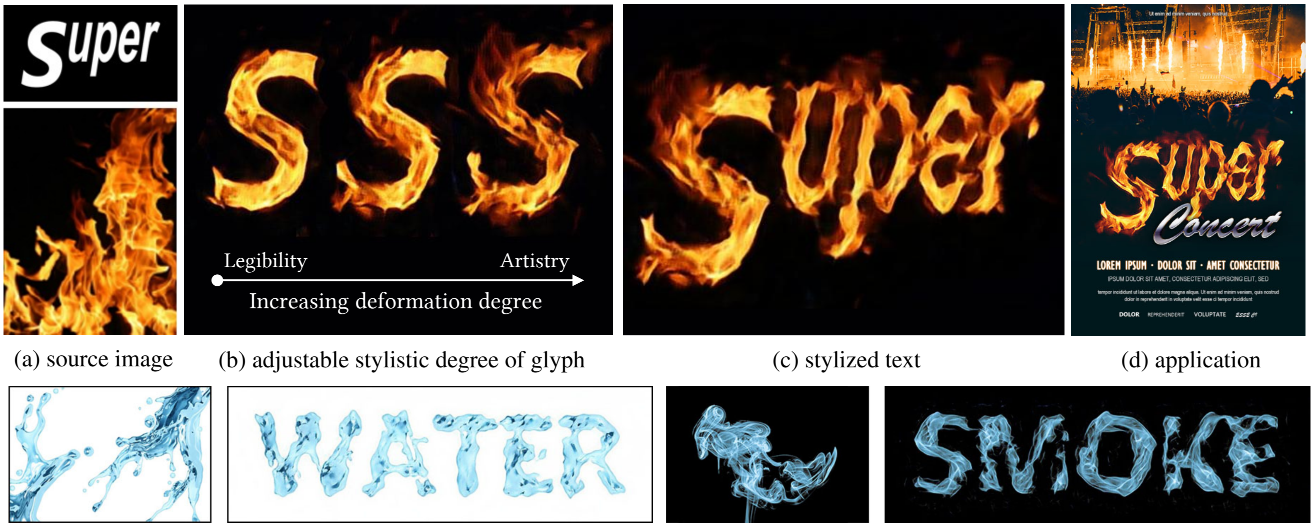
This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Controllable Artistic Text Style Transfer via Shape-Matching GAN

Shuai Yang[1,2], Zhangyang Wang[2], Zhaowen Wang[3], Ning Xu[3], Jiaying Liu[*1] and Zongming Guo[1]
[1] Institute of Computer Science and Technology, Peking University
[2] Texas A&M University [3] Adobe Research


Super
Legibility
Artistry
Increasing deformation degree
Super
Concert
LOREM IPSUM · DOLOR SIT · AMET CONSECTETUR
WATER
SMOKE

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

(e) liquid artistic text rendering (f) smoke artistic text rendering

Figure 1: We propose a novel style transfer framework for rendering artistic text from a source style image in a scale-controllable manner. Our framework allows users to (b) adjust the stylistic degree of the glyph (*i.e.* deformation degree) in a continuous and real-time way, and therefore to (c) select the artistic text that is most ideal for both legibility and style consistency. The generated diverse artistic text will facilitate users to design (d) exquisite posters and (e)(f) dynamic typography.

Abstract

Artistic text style transfer is the task of migrating the style from a source image to the target text to create artistic typography. Recent style transfer methods have considered texture control to enhance usability. However, controlling the stylistic degree in terms of shape deformation remains an important open challenge. In this paper, we present the first text style transfer network that allows for real-time control of the crucial stylistic degree of the glyph through an adjustable parameter. Our key contribution is a novel bidirectional shape matching framework to establish an effective glyph-style mapping at various deformation levels without paired ground truth. Based on this idea, we propose a scale-controllable module to empower a single network to continuously characterize the multi-scale shape features of the style image and transfer these features to the target text. The proposed method demonstrates its superiority over previous state-of-the-arts in generating diverse, controllable and high-quality stylized text.

1. Introduction

Artistic text style transfer aims to render text in the style specified by a reference image, which is widely desired in many visual creation tasks such as poster and advertisement design. Depending on the reference image, text can be stylized either by making analogy of existing well-designed text effects [28], or by imitating the visual features from more general free-form style images [30]: the latter provides more flexibility and creativity.

For general style images as reference, since text is significantly different from and more structured than natural images, more attention should be paid to its stroke shape in the stylization of text. For example, one needs to manipulate the stylistic degree or shape deformations of a glyph to resemble the style subject flames in Fig. 1(b). Meanwhile, the glyph legibility needs to be maintained so that the stylized text is still recognizable. Such a delicate balance is subjective and hard to attain automatically. Therefore, a

[*] Corresponding author
The work was done when Shuai Yang was a visiting student at TAMU.

practical tool allowing users to control the stylistic degree of the glyph is of great value. Further, as users are prone to trying various settings before obtaining desired effects, real-time response to online adjustment is important.

In the literature, some efforts have been devoted to addressing *fast scale-controllable style transfer*. They trained fast feed-forward networks, with the main focus on the scale of textures like the texture strength [2], or the size of texture patterns [17]. Up to our best knowledge, there has been no work discussing **the real-time control of glyph deformations**, which is rather crucial for text style transfer.

In view of the above, we are motivated to investigate a new problem of fast controllable artistic text style transfer from a single style image. We aim at the real-time adjustment for the stylistic degree of the glyph in terms of shape deformations. It can allow users to navigate around different forms of the rendered text and select the most desired one, as illustrated in Fig. 1(b)(c). The challenges of fast controllable artistic text style transfer lie in two aspects. On one hand, in contrast to well-defined scales such as the texture strength that can be straightforwardly modelled by hyper-parameters, the glyph deformation degree is subjective, neither clearly defined nor easy to parameterize. On the other hand, there does not exist a large-scale *paired* training set with both source text images and the corresponding results stylized (deformed) in different degrees. Usually, only one reference image is available for a certain style. It is thus also not straightforward to train data-driven models to learn multi-scale glyph stylization.

In this work, we propose a novel Shape-Matching GAN to address these challenges. Our key idea is a bidirectional shape matching strategy to establish the shape mapping between source styles and target glyphs through both backward and forward transfers. We first show that the glyph deformation can be modelled as a coarse-to-fine shape mapping of the style image, where the deformation degree is controlled by the coarse level. Based on this idea, we develop a sketch module that simplifies the style image to various coarse levels by backward transferring the shape features from the text to the style image. Resulting coarse-fine image pairs provide a robust multi-scale shape mapping for data-driven learning. With this obtained data, we build a scale-controllable module, *Controllable ResBlock*, that empowers the network to learn to characterize and infer the style features on a continuous scale from the mapping. Eventually, we can forward transfer the features of any specified scale to target glyphs to achieve scale-controllable style transfer. In summary, our contributions are threefold:

- We investigate the new problem of fast controllable artistic text style transfer, in terms of glyph deformations, and propose a novel bidirectional shape matching framework to solve it.
- We develop a sketch module to match the shape from the style to the glyph, which transforms a single style image to paired training data at various scales and thus enables learning robust glyph-style mappings.
- We present Shape-Matching GAN to transfer text styles, with a scale-controllable module designed to allow for adjusting the stylistic degree of the glyph with a continuous parameter as user input and generating diversified artistic text in real-time.

2. Related Work

Image style transfer. Leveraging the powerful representation ability of neural networks, Gatys *et al.* pioneered on the Neural Style Transfer [10], where the style was effectively formulated as the Gram matrix [9] of deep features. Johnson *et al.* trained a feed-forward StyleNet [18] using the loss of Neural Style Transfer [10] for fast style transfer [26, 15, 22, 21, 6]. In parallel, Li *et al.* [19, 20] represented styles by neural patches, which can better preserve structures for photo-realistic styles. Meanwhile, other researchers regard style transfer as an image-to-image translation problem [16, 31], and exploited Generative Adversarial Network (GAN) [12] to transfer specialized styles such as cartoons [7], paintings [25] and makeups [8, 5]. Compared to Gram-based and patch-based methods, GAN learns the style representation directly from the data, which can potentially yield more artistically rich results.

Artistic text style transfer. The problem of artistic text style transfer was first raised by Yang *et al.* [28]. The authors represented the text style using image patches, which suffered from a heavy computational burden due to the patch matching procedure. Driven by the progress of neural network, Azadi *et al.* [1] trained an MC-GAN for fast text style transfer, which, however, can only render 26 capital letters. Yang *et al.* [29] recently collected a large dataset of text effects to train the network to transfer text effects for any glyph. Unlike the aforementioned methods that assume the input style to be well-designed text effects, a patch-based model UT-Effect [30] stylized the text with arbitrary textures and achieved glyph deformations by shape synthesis [24], which shows promise for more application scenarios. Compared to UT-Effect [30], our GAN-based method further enables the continuous adjustment of glyph deformations via a controllable parameter in real-time.

Multi-scale style control. To the best of our knowledge, the research on the multi-scale style control currently focuses on two kinds of scales: the *strength* and the *stoke size* of the texture. The texture strength determines the texture similarity between the result and the style image (Fig. 2(c)). It is mainly controlled by a hyper-parameter to balance the content loss and style loss [10]. As a result, one has to re-train the model for different texture strengths. Babaeizadeh *et al.* [2] performed efficient adjustment of

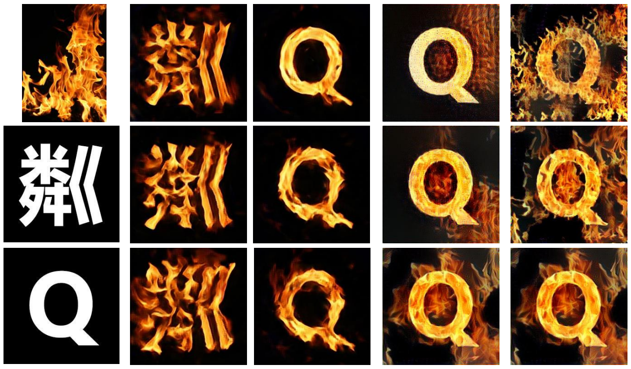


Figure 2: Comparing different scale effects in text style transfer. (a) shows the reference style and target text. In remaining columns, each shows the results with increasing (b) glyph deformation degree; (c) texture strength; and (d) stroke size. Results in (b) are generated by our proposed method, while (c) and (d) are generated by Neural Style Transfer [10].

the texture strength, with an auxiliary network to input additional parameters to modulate the style transfer process. Meanwhile, the stroke size depicts the scale of texture patterns (Fig. 2(d)), *e.g.*, size or spatial frequency. Jing *et al.* [17] proposed a stroke-controllable neural style transfer network (SC-NST) with adaptive receptive fields for stroke size control. Our work explores the glyph deformation degree (Fig. 2(b)), a different and important dimension of "scale" that is unexplored in prior work.

3. Problem Overview

We start by giving operative requirements for our new task. Considering the *maple* style for instance, it will look weird to have artistic text with the texture of leaves, but without the leave-like shapes (see an example in Fig. 9), demonstrating the need of shape deformation and matching in addition to merely transferring texture patterns. Meanwhile, the optimal scale to balance the legibility and artistry can vary a lot for different styles and text contents, not to mention the subjective variation among people. Taking Fig. 2(b) for example, one may see that the glyph with more complex strokes is more vulnerable to large glyph deformation [27]. Therefore, users will enjoy the freedom to navigate through the possible scale space of glyph deformations, without the hassle of re-training one model per scale. Concretely, a controllable artistic text style transfer shall ensure:

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

The two requirements distinguish our problem from those studied by previous multi-scale style transfer methods, which are either unable to adjust the shape at all [2, 17] (*e.g.*, Fig. 2(c)(d)) or fail to do so efficiently [30].

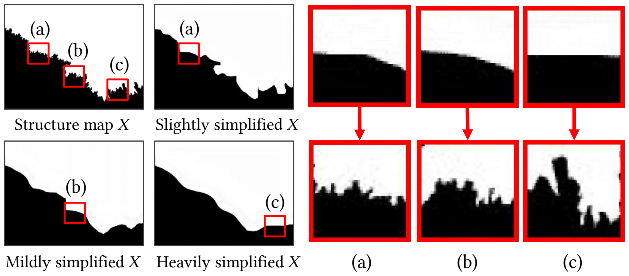


Figure 3: Illustration of bidirectional shape matching. Left two columns: a leaf-shaped structure map and its three backward simplified versions. Right columns: forward shape mappings under (a) slight, (b) moderate, and (c) heavy deformations.

Our solution to this problem is a novel bidirectional shape matching strategy. As illustrated in Fig. 3, the target structure map is first (backward) simplified to different coarse levels, and then its stylish shape features can be characterized by the (forward) multi-level coarse-to-fine shape mappings, to realize multi-scale transfer. As shown in Fig. 3(a)-(c), similar horizontal strokes at different levels are mapped to different shapes, and the coarser the level, the greater the deformation between the mapped shapes. *Artistry* is met since the targets in these mappings are exactly the reference fine-level stylish shapes while *Controllability* can be achieved via training a feed-forward network.

To sum up, we formulate the new task of scale-controllable artistic text style transfer as learning *the function to map the style image from different coarse levels back to itself in a fast feed-forward way*. Still, two technical roadblocks remain to be cleared. First, how to simplify the shape to make the obtained mapping applicable to the text images. Second, how to learn the many-to-one (multiple coarse levels to a fine level) mapping without model collapse. Sec. 4 will detail how we address these challenges with our network design.

4. Shape-Matching GAN

Assume that Y and I denote the style image and text image provided by users, respectively. We study the problem of designing a feed-forward stylization model G to render artistic text under different deformation degrees controlled by a parameter $\ell \in [0, 1]$, where larger ℓ corresponds to greater deformations. We further decompose the style transfer process into two successive stages: structure transfer and texture transfer, which are modelled by generators G_S and G_T, separately. The advantage of such decomposition is that we can disentangle the influence of textures and first focus on the key shape deformation problem. We denote $G = G_T \circ G_S$, and formulate the stylization process as:

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

where the target statistic $p(I^Y_\ell)$ of the stylized image I^Y_ℓ is characterized by the text image I, the style image Y and the

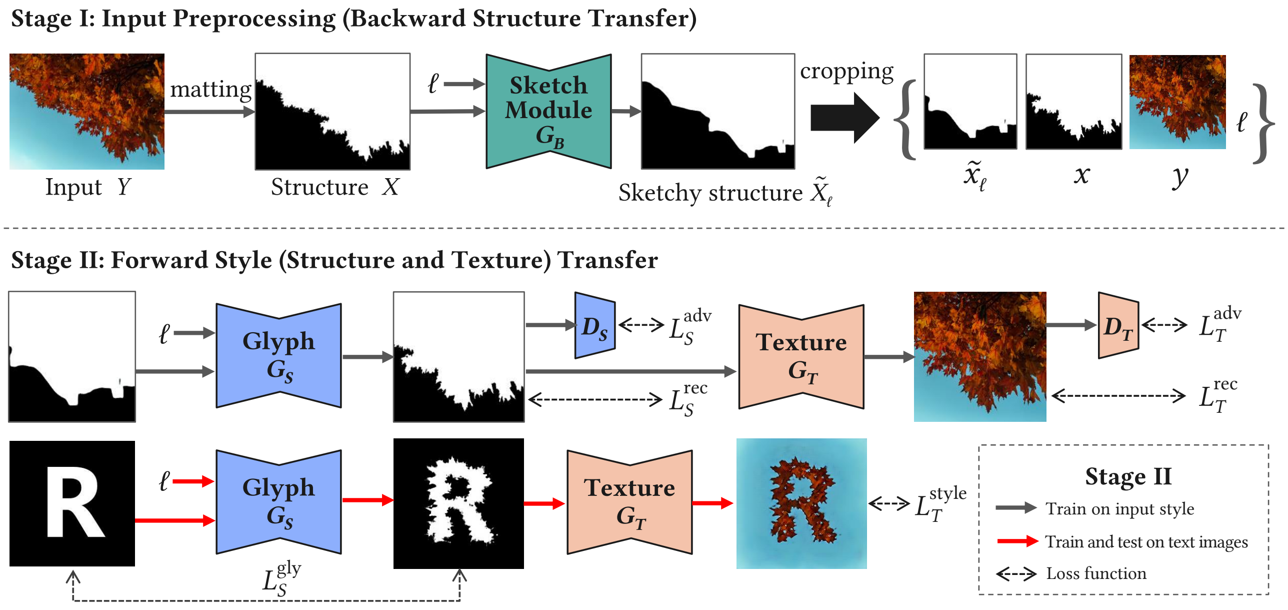


Figure 4: Overview of our bidirectional shape matching framework.

controllable parameter ℓ.

As outlined in Sec. 3, our solution to structure transfer is bidirectional shape matching. Assume the structure map X to mask the shape of the style subject in Y is given, which can be easily obtained by image editing tools such as Photoshop or existing image matting algorithms. In the stage of backward structure transfer, we preprocess X to obtain training pairs $\{\tilde{X}_\ell, X\}$ for G_S, where $\tilde{X}_\ell$ is a sketchy (coarse) version of X with the shape characteristics of the text, and ℓ controls the coarse level. In the stage of forward structure transfer, G_S learns from $\{\tilde{X}_\ell, X\}$ to stylize the glyph with various deformation degrees. Fig. 4 summarizes the overall framework built upon two main components:

- **Glyph Network** G_S: It learns to map $\tilde{X}_\ell$ with deformation degree ℓ to X during training. In testing, it transfers the shape style of X onto the target text image I, producing the structure transfer result I_ℓ^X.
- **Texture Network** G_T: It renders the texture in the style image Y on I_ℓ^X to yield the final artistic text I_ℓ^Y.

The generators are accompanied with corresponding discriminators D_S and D_T to improve the quality of the results through adversarial learning. In the following, we present the details of our bidirectional shape matching and the proposed controllable module that enables G_S to learn multi-scale glyph deformations in Sec. 4.1. The texture transfer network G_T is then introduced in Sec. 4.2.

4.1. Bidirectional Structure Transfer (G_S)

Backward structure transfer. To transfer the glyph characteristics to X at different coarse levels, we propose a sketch module G_B composed of a smoothness block and a transformation block, as shown in Fig. 5(a). Inspired by the Gaussian scale-space representation [3, 23] for simplifying images at different scales, our smoothness block is set as a fixed convolutional layer with Gaussian kernel, whose standard deviation $\sigma = f(\ell)$ is controlled by ℓ and a linear function $f(\cdot)$. Our key idea is to bridge the source style domain and the target text domain using the smoothness block that maps the text image and X into a smooth domain, where the details are eliminated and the contours demonstrate similar smoothness. Structure transfer is then achieved by training the transformation block to map the smoothed text images back to the text domain to learn the glyph characteristics. The advantages of our sketch module are twofold: 1) the coarse level (and thus the deformation degree) can be naturally parameterized by σ; and 2) the training process of G_B only requires easily accessible text images. Once trained, it can be applied to arbitrary input styles.

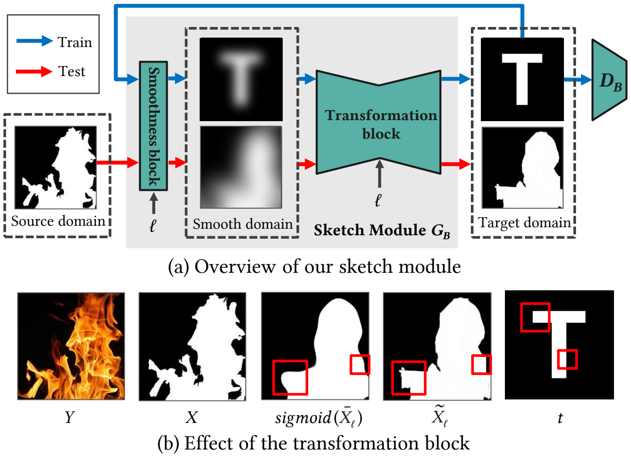


Figure 5: Backward structure transfer by sketch module G_B

For training G_B, we sample a text image t from the text dataset provided by [29] and a parameter value ℓ from $[0, 1]$. G_B is tasked to reconstruct t:

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

In addition, we impose a conditional adversarial loss to force G_B to generate more text-like contours:

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

where D_B learns to determine the authenticity of the input image and whether it matches the given smoothed image $\bar{t}_\ell$ and the parameter ℓ. Thus, the total loss takes the form of

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

Finally, by applying trained G_B to X with various level ℓ, we can obtain the corresponding sketchy shape $\tilde{X}_\ell = G_B(X, \ell)$. An example is shown in Fig. 5(b). The simply thresholded Gaussian representation $sigmoid(\bar{X}_\ell)$ (by replacing the transformation block with a sigmoid layer) does not match the shape of the text. In contrast, our sketch module effectively simplifies the flame profile to the shape of strokes in the red box regions, thus providing a more robust shape mapping for the glyph network.

Forward structure transfer. Having obtained $\{\tilde{X}_\ell\}$, $\ell \in [0, 1]$, we now train the glyph network G_S to map them to the original X so that G_S can characterize the shape features of X and transfer these features to the target text. Note that our task is a many-to-one mapping, and we only have a single example X. The network should be carefully designed to avoid just memorizing the ground truth X and falling into model collapse, namely, yielding very similar results regardless of the parameter ℓ during testing.

To tackle this challenging task, we employ two strategies: *data augmentation* and *Controllable ResBlock*. First, X and $\tilde{X}_\ell$ are randomly cropped into sub-image pairs $\{x, \tilde{x}_\ell\}$ to gather as a training set. Second, we build G_S upon the architecture of StyleNet [18], and propose a very simple yet effective Controllable ResBlock to replace the original ResBlock [13] in the middle layers of StyleNet. Our Controllable ResBlock is a linear combination of two ResBlocks weighted by ℓ, as shown in Fig. 6. For $\ell = 1$ (0), G_S degrades into the original StyleNet, and is solely tasked with the greatest (tiniest) shape deformation to avoid the many-to-one problem. Meanwhile for $\ell \in (0, 1)$, G_S tries to compromise between the two extremes.

In terms of the loss, G_S aims to approach the ground truth X in an L_1 sense and confuse the discriminator D_S:

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

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

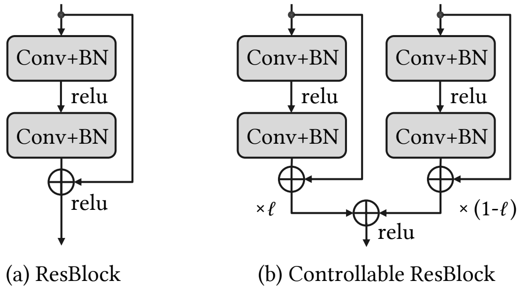


Figure 6: Controllable ResBlock

For some styles with large ℓ, the text t could be too severely deformed to be recognized. Thus we propose an optional glyph legibility loss to force the structure transfer result $G_S(t, \ell)$ to maintain the main stroke part of t:

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

where $\otimes$ is the element-wise multiplication operator, and $M(t)$ is a weighting map based on distance field whose pixel value increases with its distance to the nearest text contour point of t. The overall loss for G_S is as follows:

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

4.2. Texture Transfer (G_T)

Given the structure transfer result $I_\ell^X = G_S(I, \ell)$, the texture rendering task can be formulated as a standard image analogy problem such that $X : Y :: I_\ell^X : I_\ell^Y$ [14], which can be well solved by existing algorithms like the greedy-based Image Analogy [14] and the optimization-based Neural Doodle [4]. To build an end-to-end fast text stylization model, we instead train a feed-forward network G_T for texture rendering. Similar as in training G_S, we first use random cropping to obtain adequate training pairs $\{x, y\}$ from X and Y. Then we train G_T using the reconstruction loss and conditional adversarial loss:

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

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

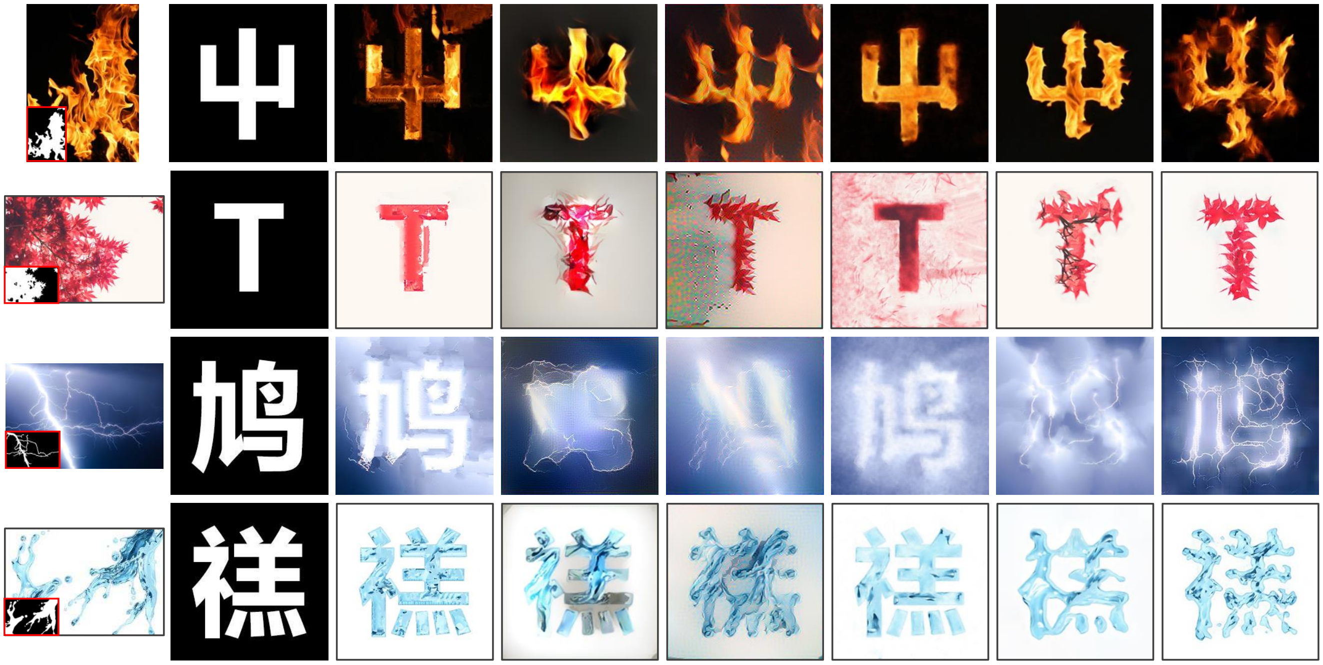
The overall style rendering performance on the sampled text image t is further taken into account by adding the style loss $\mathcal{L}_T^{\text{style}}$ proposed in Neural Style Transfer [10]. Finally, the objective of texture transfer can be defined as:

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

5. Experimental Results

5.1. Implementation Details

Network architecture. We adapt our generators from the Encoder-Decoder architecture of StyleNet [18] with



(a) Input style (b) Target text (c) Image Analogy (d) NST-mask (e) Doodle (f) T-Effect (g) UT-Effect (h) Ours

Figure 7: Comparison with state-of-the-art methods on various styles. (a) Input style with its structure map in the lower-left corner. (b) Target text. (c) Image Analogy [14]. (d) Neural Style Transfer [10] with spatial control [11]. (e) Neural Doodle [4]. (f) T-Effect [28]. (g) UT-Effect [30]. (h) Our style transfer results. We manually select the suitable deformation degrees for UT-Effect [30] and out method.

six ResBlocks, except that G_S uses the proposed Controllable ResBlock instead. Our discriminators follow PatchGAN [16]. Since the structure map contains many saturated areas, we add gaussian noises onto the input of G_S and G_T to avoid ambiguous problem. It also empowers our network to generate diversified results during testing. Code and pretrained models are available at: `https://github.com/TAMU-VITA/ShapeMatchingGAN`.

Network training. We randomly crop the style image to 256×256 sub-images for training. The Adam optimizer is adopted with a fixed learning rate of 0.0002. To stabilize the training of G_S, we gradually increase the sampling range of ℓ. Specifically, G_S is first trained with a fixed $\ell = 1$ to learn the greatest deformation. Then we copy the parameters from the trained half part in Controllable ResBlocks to the other half part and use $\ell \in \{0, 1\}$ to learn two extremes. Finally, G_S is tuned on $\ell \in \{i/K\}_{i=0,\dots,K}$. We find that $K = 3$ is sufficient for G_S to infer the remaining intermediate scales. The linear function to control the standard deviation of the Gaussian kernel is $f(\ell) = 16\ell + 8$. For all experiments, we set $\lambda_B^{\text{rec}} = \lambda_S^{\text{rec}} = \lambda_T^{\text{rec}} = 100$, $\lambda_B^{\text{adv}} = \lambda_T^{\text{adv}} = 1$, $\lambda_S^{\text{adv}} = 0.1$, and $\lambda_T^{\text{style}} = 0.01$.

5.2. Comparisons with State-of-the-Art Methods

Artistic text style transfer. In Fig 7, we present the qualitative comparison with five state-of-the-art style transfer methods: Image Analogy [14], NST [11], Doodle [4], T-Effect [28] and UT-Effect [30][1]. These methods are selected because they are all one-shot supervised (or can be adapted to a supervised version) for a fair comparison, which transfer styles with a single style image and its structure map.

Image Analogy [14] and T-Effect [28] directly copy the texture patches to the text region, yielding rigid and unnatural contours. NST [11] and Doodle [4] are deep learning-based methods, where the shape characteristics of the style are implicitly represented by deep features. Thus these methods could modify the glyph contours but often lead to unrecognizable text. In terms of texture, they suffer from color deviations and checkerboard artifacts. UT-Effect [30] explicitly matches the glyph to the style at a patch level. However, image patches are not always robust. For example, in the *maple* style, the leaf shapes are not fully transferred to the vertical stroke. In addition, texture details are lost due to the patch blending procedure. By comparison, our network is able to learn accurate shape characteristics through the proposed bidirectional shape matching strategy, and transfers vivid textures via adversarial learning, which together leads to the most visually appealing results.

Scale-controllable style transfer. In Fig. 8, we present the qualitative comparison with two scale-controllable style transfer methods: SC-NST [17] and UT-Effect [30]. SC-

[1] For NST, We build upon its public model and implement the spatial control function introduced in [11]. Meanwhile, results of other methods are obtained by public models or provided by the authors.

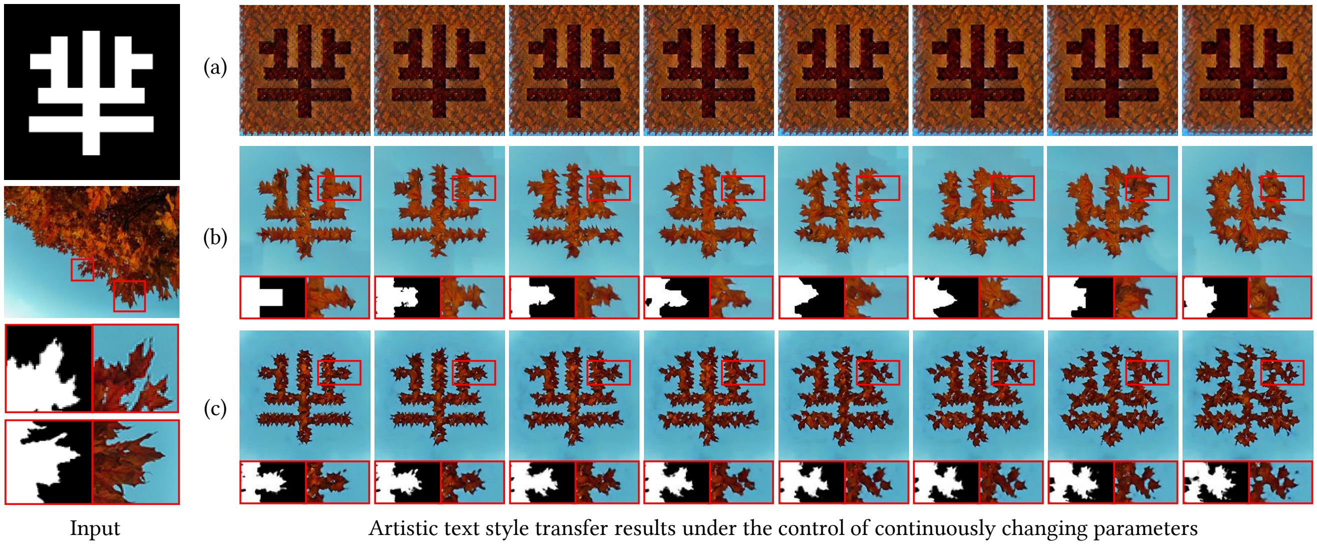

(a)
(b)
(c)
Input
Artistic text style transfer results under the control of continuously changing parameters

Figure 8: Qualitative comparison between the proposed method and other scale-controllable style transfer methods. 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) stroke-controllable neural style transfer (SC-NST) [17] with stroke size evenly increasing from 256 to 768; (b) UT-Effect [30] with resolution level evenly increasing from 1 to 7; (c) the proposed method with ℓ evenly increasing from 0 to 1. All results are produced by one single model for each method. For UT-Effect [30] and our method, the red box region is shown enlarged in the bottom with the corresponding structure map provided for better visual comparison.

NST [17] does not synthesize the textures in the correct region due to its unsupervised setting. Regardless of this factor, it can adjust the texture size, but is ineffective in controlling the glyph deformation. UT-Effect [30] matches boundary patches at multiple resolutions for structure transfer, which has several drawbacks: First, as shown in Fig. 8(b), the greedy-based patch matching fails to global consistently stylize the glyph. Second, the patch blending procedure inevitably eliminates many shape details. Third, the continuous transformation is not supported. On the contrary, the proposed method achieves continuous transformation with fine details, showing a smooth growing process of the leaves as they turn more luxuriant. In terms of efficiency, for 256×256 images in Fig. 8, the released MATLAB-based UT-Effect [30] requires about 100 s per image with Intel Core i7-6500U CPU (no GPU version available). In comparison, our feed-forward method only takes about 0.43 s per image with Intel Xeon E5-2650 CPU and **16 ms per image** with a GeForce GTX 1080 Ti GPU, which implies a potential of nearly real-time user interaction.

5.3. Ablation Study

Network architecture. To analyze each component in our model, we design the following experiments with different configurations:

- Baseline: Our baseline model contains only a texture network trained to directly map the structure map X back to 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 the commonly used label concatenation instead of using 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. 9 displays the stylization results of these models. Without structure transfer, the contours of the stylized text by baseline model are rigid, showing poor shape consistency with the reference style. The naïve glyph network could create leaf-like shapes, but fails to learn the challenging many-to-one mapping. It simply ignores the conditional ℓ, and generates very similar results. This problem is well solved by the proposed Controllable ResBlock. As shown in the fourth column of Fig. 9, our glyph network can even learn the multi-scale structure transfer and texture transfer simultaneously, although the rendered texture is flat and has checkerboard artifacts. By handing the texture transfer task over to a separate texture network, our full model can synthesize high-quality artistic text, with both shape and texture consistency w.r.t. the reference style.

Sketch module. In Fig. 10, we examine the effect of the sketch module G_B through a comparative experiment. As introduced in Sec. 4.1, our sketch module aims to transfer the shape characteristics of the text to the style image to provide a robust mapping between the source and target domains. To make a comparison, we replace the Trans-

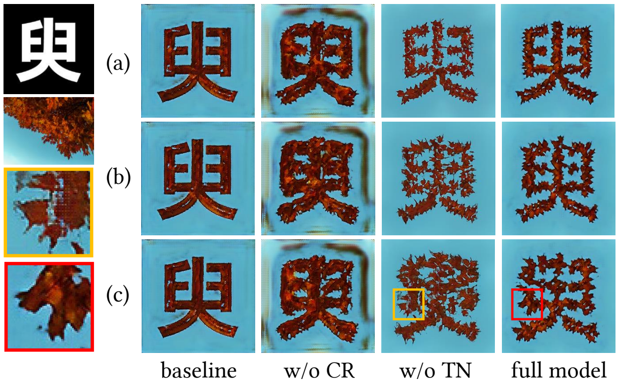


Figure 9: Analysis for the network configurations in controllable artistic text style transfer. For the first column, from top to bottom: target text, style image, the enlarged patches from the results without and with the texture network, respectively. Remaining columns: (a)-(c) Results with $\ell = 0.0, 0.5, 1.0$, respectively.

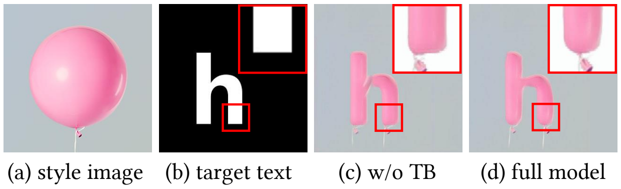


Figure 10: Effect of the proposed sketch module. The red box area is enlarged and the contrast is enhanced in the top right corner for better visual comparison.

formation Block (TB) in G_B with a simple sigmoid layer. The resulting naïve sketch module is still able to simplify the shape, but cannot match it with the glyph. Without robust mappings, the shape of the stylized text is not correctly adjusted and is as rigid as the input text as shown in Fig. 10(c). By contrast, our full model successfully synthesizes a rounded h-shaped balloon in Fig. 10(d).

Loss function. We study the effect of the glyph legibility loss (Eq. (7)) in Fig. 11. When transferring a trickle of wafting smoke onto a rigid Chinese character with a high deformation degree $\ell = 0.75$, the strokes of this character demonstrate irregular shapes, uneven thickness, and even fractures in Fig. 11(c). Although very similar to the style, the character is unrecognizable. As shown in Fig. 11(d), by setting $\lambda_S^{\text{gly}} = 1$, our glyph legibility loss effectively preserves the trunk of the strokes, while allowing a high freedom to deform the contours of the strokes, thus achieving a balance between legibility and artistry.

5.4. Applications

In addition to the poster and dynamic typography design shown in Fig. 1(d)-(f), we further present two other applications of our method as follows.

Structure/texture mash-up. The disentanglement of structures and textures enables us to combine different styles to create some brand-new text styles. Some examples are shown in Fig. 12(a), where we apply the textures of *maple*, *water* and *smoke* to the text with the shape characteristics of *maple*, respectively.

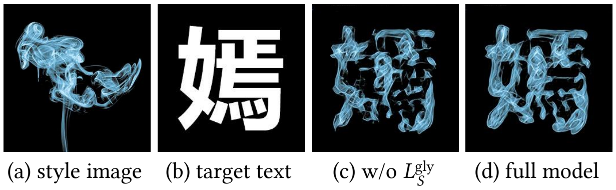


Figure 11: Effect of the glyph legibility loss $\mathcal{L}_S^{\text{gly}}$.

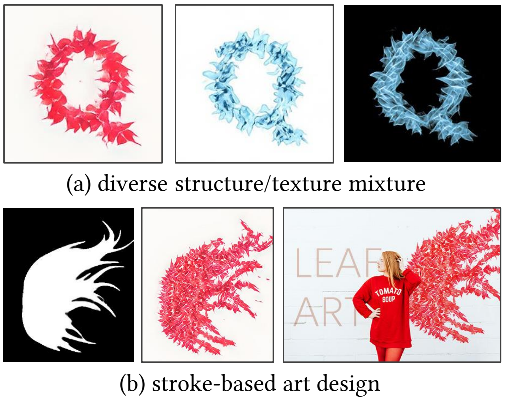

(a) diverse structure/texture mixture
LEAF
ART
TOMATO
SOUP
(b) stroke-based art design

Figure 12: Applications of our method.

Stroke-based art design. Since no step specially tailored for the text is used, our method can be easily extended to style transfer on more general shapes such as symbols and icons. In Fig. 12(b), we show an example for synthesizing wings made of maple leaves from a user-provided icon.

6. Conclusion

In this paper, we present a fast artistic text style transfer deep network that allows for flexible, continuous control of the stylistic degree of the glyph. We formulate the task of glyph deformation as a coarse-to-fine mapping problem and propose a bidirectional shape matching framework. A novel sketch module is proposed to reduce the structural discrepancy between the glyph and style to provide robust mappings. Exploiting the proposed Controllable ResBlock, our network is able to effectively learn the many-to-one shape mapping for multi-scale style transfer. We validate the effectiveness and robustness of our method by comparisons with state-of-the-art style transfer algorithms.

Acknowledgement. This work was supported in part by National Natural Science Foundation of China under contract No. 61772043, and in part by Beijing Natural Science Foundation under contract No. L182002 and No. 4192025. This work was supported by China Scholarship Council. We thank the Unsplash users (Aaron Burden, Andre Benz, Brandon Morgan, Brooke Cagle, Florian Klauer, Grant Mccurdy and Stephen Hocking) who put their photos under the Unsplash license for public use.

References

[1] Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018. 2

[2] Mohammad Babaeizadeh and Golnaz Ghiasi. Adjustable real-time style transfer. 2018. arXiv:1811.08560. 2, 3

[3] Jean Babaud, Andrew P Witkin, Michel Baudin, and Richard O Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):26–33, 1986. 4

[4] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. 2016. arXiv:1603.01768. 5, 6

[5] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 40–48, 2018. 2

[6] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017. 2

[7] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 9465–9474, 2018. 2

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018. 2

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015. 2

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 2, 3, 5, 6

[11] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 3985–3993, 2017. 6

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[14] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proc. Conf. Computer Graphics and Interactive Techniques*, pages 327–340, 2001. 5, 6

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. Int'l Conf. Computer Vision*, pages 1510–1519, 2017. 2

[16] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 2, 6

[17] Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke controllable fast style transfer with adaptive receptive fields. In *Proc. European Conf. Computer Vision*, pages 238–254, 2018. 2, 3, 6, 7

[18] Justin Johnson, Alexandre Alahi, and Fei Fei Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conf. Computer Vision*, pages 694–711, 2016. 2, 5

[19] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2479–2486, 2016. 2

[20] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. European Conf. Computer Vision*, pages 702–716, 2016. 2

[21] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017. 2

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017. 2

[23] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. 4

[24] Amir Rosenberger, Daniel Cohen-Or, and Dani Lischinski. Layered shape synthesis: automatic generation of control maps for non-stationary textures. *ACM Transactions on Graphics*, 28(5):107, 2009. 2

[25] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *Proc. European Conf. Computer Vision*, pages 698–714, 2018. 2

[26] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017. 2

[27] Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S Huang. Deepfont: Identify your font from an image. In *Proc. ACM Int'l Conf. Multimedia*, pages 451–459, 2015. 3

[28] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 7464–7473, 2017. 1, 2, 6

[29] Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. Tet-gan: Text effects transfer via stylization and destylization. In *AAAI Conference on Artificial Intelligence*, 2019. 2, 5

[30] Shuai Yang, Jiaying Liu, Wenhan Yang, and Zongming Guo. Context-aware text-based binary image stylization and synthesis. *IEEE Transactions on Image Processing*, 2019. 1, 2, 3, 6, 7

[31] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. Int'l Conf. Computer Vision*, pages 2242–2251, 2017. 2