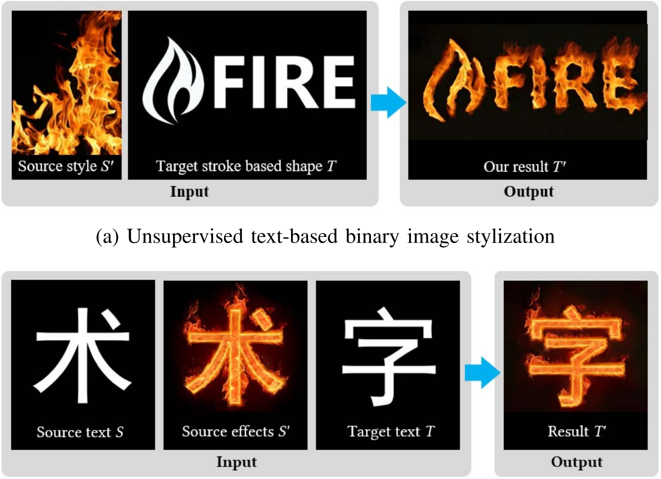
Context-Aware Text-Based Binary Image Stylization and Synthesis

Shuai Yang, Jiaying Liu, *Senior Member, IEEE*, Wenhan Yang, *Student Member, IEEE*, and Zongming Guo, *Member, IEEE*

***Abstract*— In this work, we present a new framework for the stylization of text-based binary images. First, our method stylizes the stroke-based geometric shape like text, symbols, and icons in the target binary image based on an input style image. Second, the composition of the stylized geometric shape and a background image is explored. To accomplish the task, we propose legibility-preserving structure and texture transfer algorithms, which progressively narrow the visual differences between the binary image and the style image. The stylization is then followed by a context-aware layout design algorithm, where the cues for both seamlessness and aesthetics are employed to determine the optimal layout of the shape in the background. Given the layout, the binary image is seamlessly embedded into the background by texture synthesis under a context-aware boundary constraint. According to the contents of binary images, our method can be applied to many fields. We show that the proposed method is capable of addressing the unsupervised text stylization problem and is superior to state-of-the-art style transfer methods in automatic artistic typography creation. Besides, extensive experiments on various tasks, such as visual-textual presentation synthesis, icon/symbol rendering, and structure-guided image inpainting demonstrate the effectiveness of the proposed method.**

***Index Terms*— Texture synthesis, structure synthesis, context-aware, style transfer, image inpainting.**



(a) Unsupervised text-based binary image stylization

(b) Supervised text stylization [8]

Fig. 1. Supervised text stylization requires registered raw text S and text effects S' as input. We instead handle a more challenging unsupervised text-based image stylization problem with an arbitrary source image S'. (a) Our unsupervised stylization result. (b) Supervised stylization result by [8].

I. Introduction

STYLE transfer is the task of migrating a style from an image to another to synthesize a new artistic image. It is of special interest in visual design, and has applications such as painting synthesis and photography post-processing. However, creating an image in a particular style manually requires great skills that are beyond the capabilities of average users. Therefore, automatic style transfer has become a trending topic both in academic literature and industrial applications.

Text and other stroke-based design elements such as symbols, icons and labels highly summarize the abstract imagery of human visual perceptions and are ubiquitous in our daily life. The stylization of text-based binary images as in Fig. 1(a) is of great research value but also poses a challenge of narrowing the great visual discrepancy between the binary flat shapes and the colorful style image.

Style transfer has been investigated for years, where many successful methods are proposed, such as the non-parametric method Image Quilting [1] and the parametric method Neural Style [2]. Non-parametric methods take samples from the style image and place the samples based on pixel intensity [1], [3], [4] or deep features [5] of the target image to synthesize a new image. Parametric methods represent the style as statistical features, and adjust the target image to satisfy these features. Recent deep learning based parametric methods [2], [6], [7] exploit high-level deep features, and thereby have the superior capability of semantic style transfer. However, none of the aforementioned methods are specific to the stylization of text-based binary images. In fact, for non-parametric methods, it is hard to use pixel intensity or deep features to establish a direct mapping between a binary image and a style image, due to their great modality discrepancy. On the other hand, text-based binary images lack high-level semantic information, which limits the performance of the parametric methods.

Manuscript received December 3, 2017; revised July 18, 2018; accepted September 26, 2018. Date of publication October 1, 2018; date of current version October 22, 2018. This work was supported by the National Natural Science Foundation of China under Contract 61772043. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Catarina Brites. *(Corresponding author: Jiaying Liu.)*

The authors are with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: williamyang@pku.edu.cn; liujiaying@pku.edu.cn; yangwenhan@pku.edu.cn; guozongming@pku.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes a pdf. Contact williamyang@pku.edu.cn for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2873064

As the most related method to our problem, a text effects transfer algorithm [8] is recently proposed to stylize the binary text image. In that work, the authors analyzed the high correlation between texture patterns and their spatial distribution in text effects images, and modeled it as a distribution prior, which has been proven to be highly effective at text stylization. But this method strictly requires the source style to be a well-structured typography image. Moreover, it follows the idea of Image Analogies [9] to stylize the image in a supervised manner. For supervised style transfer, in addition to the source style image, its non-stylized counterpart is also required to learn the transformation between them, as shown in Fig. 1(b). Unfortunately, such a pair of inputs is not readily available in practice, which greatly limits its application scope.

In this work, we handle a more challenging unsupervised stylization problem, only with a binary text-based binary image and an arbitrary style image as in Fig. 1(a). To bridge the distinct visual discrepancies between the binary image and the style image, we extract the main structural imagery of the style image to build a preliminary mapping to the binary image. The mapping is then refined using a structure transfer algorithm, which adds shape characteristics of the source style to the binary shape. In addition to the distribution constraint [8], a saliency constraint is proposed to jointly guide the texture transfer process for the shape legibility. These improvements allow our unsupervised style transfer to yield satisfying artistic results without the ideal input required by supervised methods.

Furthermore, we investigate the combination of stylized shapes (text, symbols, icons) and background images, which is very common in visual design. Specifically, we propose a new context-aware text-based binary image stylization and synthesis framework, where the target binary shape is seamlessly embedded in a background image with a specified style. By "seamless", we mean the target shape is stylized to share context consistency with the background image without abrupt image boundaries, such as decorating a blue sky with cloud-like typography. To achieve it, we leverage cues considering both seamlessness and aesthetics to determine the image layout, where the target shape is finally synthesized into the background image. When a series of different styles are available, our method can generate diverse artistic typography, symbols or icons against the background image, thereby facilitating a much wider variety of aesthetic interest expression. In summary, our major technical contributions are:

- We raise a new text-based binary image stylization and synthesis problem for visual design and develop the first automatic aesthetic driven framework to solve it.
- We present novel structure and texture transfer algorithms to balance shape legibility with texture consistency, which we show to be effective in style transition between the binary shape and the style image.
- We propose a context-aware layout design method to create professional looking artwork, which determines the image layout and seamlessly synthesizes the artistic shape into the background image.

The rest of this paper is organized as follows. In Section II, we review related works in style transfer and text editing. Section III defines the text-based binary image stylization problem, and gives an overview of the framework of our method. In Section IV and V, the details of the proposed legibility-preserving style transfer method and context-aware layout design method are presented, respectively. We validate our method by conducting extensive experiments and comparing with state-of-the-art style transfer algorithms in Section VI. Finally, we conclude our work in Section VII.

II. Related Work

A. Color Transfer

Pioneering methods transfer colors by applying a global transformation to the target image to match the color statistics of a source image [10]–[12]. When the target image and the source image have similar content, these methods generate satisfying results. Subsequent methods work on color transfer in a local manner to cope with the images of arbitrary scenes. They infer local color statistics in different regions by image segmentation [13], [14], perceptual color categories [15], [16] or user interaction [17]. More recently, Shih *et al.* [18], [19] employed fine-grained patch/pixel correspondences to transfer illumination and color styles for landscape images and headshot portraits. Yan *et al.* [20] leveraged deep neural networks to learn effective color transforms from a large database. In this paper, we employ color transfer technology [11] to reduce the color difference between the style image and the background image for seamless shape embedding.

B. Texture Synthesis

Texture synthesis technologies attempt to generate new textures from a given texture example. Non-parametric methods use pixel [21] or patch [1] samplings in the example to synthesize new textures. For these methods, the coherence of neighboring samples is the research focus, where patch blending via image averaging [22], dynamic programming [1], graph cut [23] and coherence function optimization [24] is proposed. Meanwhile, parametric methods build mathematic models to simulate certain texture statistics of the texture example. Among this kind of methods, the most popular one is the Gram-matrix model proposed by Gatys *et al.* [25]. Using the correlations between multi-level deep features to represent textures, this model produces natural textures of noticeably high perceptual quality. In this paper, we adapt conventional texture synthesis methods to dealing with binary text images. We apply four constrains of text shape, texture distribution, texture repetitiveness and text saliency to the texture synthesis method of Wexler et al. [24] to build our novel texture transfer model.

C. Texture Transfer

In texture transfer, textures are synthesized under the structure constraint from an additional content image. According to whether a guidance map is provided, texture transfer can be further categorized into supervised and unsupervised methods.

Supervised methods, also known as image analogies [9], rely on the availability of an input image and its stylized result.

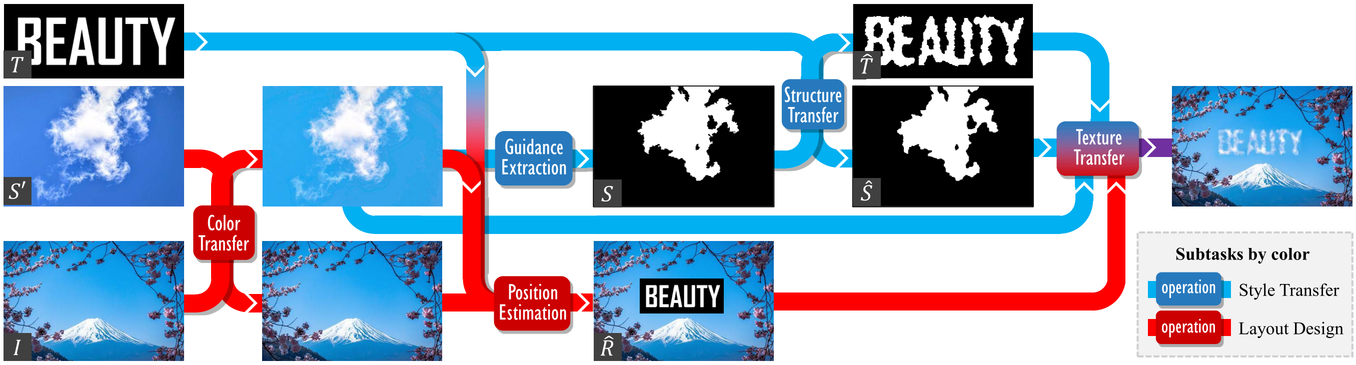


Fig. 2. Overview of the proposed algorithm. Our method consists of two subtasks: style transfer for stylizing the target text-based binary image T based on a style image S' and layout design for synthesizing the stylized image into a background image I. We first extract a guidance map S from S' to establish a mapping with T. Then a structure adjustment is adopted to bridge the structural gap between T and S. Under the guidance of $\hat{S}$ and $\hat{T}$, the stylized image is generated via texture transfer. To embed the target shape of T into the background image, we develop a method to determine the image layout $\hat{R}$. The colors of I and S' are adjusted and the contextual information of I is used to constrain the texture transfer for visual seamlessness. This flowchart gives an example of synthesizing a visual-textual presentation when T is a text image. The operation modules are colored to show the subtasks they belong to, where dark blue and dark red represent style transfer and layout design, respectively. Best viewed in color. *Image credits: Unsplash users JJ Ying, Tim Gouw.*[1]

These methods learn a mapping between such an example pair, and stylize the target image by applying the learned mapping to it. Since first reported in [9], image analogies have been extended in various ways such as video analogies [26] and fast image analogies [27]. The main drawback of image analogies is the strict requirement for the registered example pair. Most often, we only have a style image at hand, and need to turn to the unsupervised texture transfer methods.

Without the guidance of the example pair, unsupervised methods directly find mappings between different texture modalities. For instance, Efros and Freeman [1] introduced a guidance map derived from image intensity to help find correspondences between two texture modalities. Zhang *et al.* [28] used a sparse-based initial sketch estimation [29] to construct a mapping between the source sketch texture and the target image. Frigo *et al.* [3] put forward a patch partition mechanism for an adaptive patch mapping, which balances the preservation of structures and textures. However, these methods attempt to use intensity features to establish a direct mapping between the target image and the style image, and will fail in our case where the two input images have huge visual differences. By contrast, our method proposes to extract an abstract binary imagery from the style image, which shares the same modality as the target image and serves as a bridge.

Fueled by the recent development of deep learning, there has been rapid advancement of deep-based methods that leverage high-level image features for style transfer. In pioneering Neural Style [2], the authors adapted Gram-matrix-based texture synthesis [25] to style transfer by incorporating content similarities, which enables the composition of different perceptual information. This method has inspired a new wave of research on video stylization [30], perceptual factor control [31] and acceleration [32]. In parallel, Li and Wand [6] introduced a framework called CNNMRF that exploits Markov Random Field (MRF) to enforce local texture transfer. Based on CNNMRF, Neural Doodle [33] incorporates semantic maps for analogy guidance, which turns semantic maps into artwork. The main advantage of parametric deep-based methods is their ability to establish semantic mappings. For instance, it is reported in [6] that the network can find accurate correspondences between real faces and sketched faces, even if their appearances differ greatly in pixel domain. However, in our problem, the plain text image provides little semantic information, making these parametric methods lose their advantages in comparison to our non-parametric method, as demonstrated in Fig. 12.

D. Text Stylization

In the domain of text image editing, several tasks have been addressed like calligrams [34]–[36] and handwriting generation [37], [38]. Lu *et al.* [39] arranged and deformed pre-designed patterns along user-specified paths to synthesize decorative strokes. Handwriting style transfer [40] is accomplished using non-parametric samplings from a stroke library created by trained artists or parametric neural networks to learn stroke styles [38]. However, most of these studies focus on text deformation. Much less has been done with respect to the fantastic text effects such as shadows, outlines, dancing flames (see Fig. 1), and soft clouds (see Fig. 2).

To the best of our knowledge, the work of Yang *et al.* [8] is the only prior attempt at generating text effects. It solves the text stylization problem using a supervised texture transfer technique: a pair of registered raw text and its counterpart text effects are provided to calculate the distribution characteristics of the text effects, which guide the subsequent texture synthesis. In contrast, our framework automatically generates artistic typography, symbols and icons based on arbitrary source style images, without the input requirements as in [8]. Our method provides a more flexible and effective tool to create unique visual design artworks.

[1] Unsplash (https://unsplash.com/) shares copyright-free photography from over 70,000 contributing photographers under the Unsplash license. We collect photos from Unsplash for use as style images and background images.

III. Problem Formulation and Framework

We aim to automatically embed the target text-based binary shape in a background image with the style of a source reference image. To achieve this goal, we decompose the task into two subtasks: 1) Style transfer for migrating the style from source images to text-based binary shapes to design artistic shapes. 2) Layout design for seamlessly synthesizing artistic shapes in the background image to create visual design artwork such as posters and magazine covers.

Fig. 2 shows an overview of our algorithm. For the first subtask, we abstract a binary image from the source style image, adjust its contour and the outline of the target shape to narrow the structural difference between them. The adjusted results establish an effective mapping between the target binary image and the source style image. Then we are able to synthesize textures for the target shape. For the second subtask, we first seek the optimal layout of the target shape in the background image. Once the layout is determined, the shape is seamlessly synthesized into the background image under the constraint of the contextual information. The color statistics of the background image and the style image are optionally adjusted to ensure visual consistency.

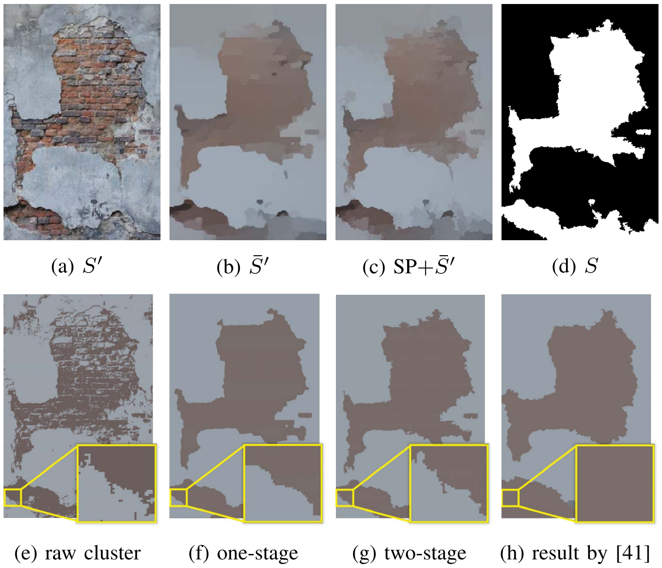
A. Style Transfer

The goal of text-based image style transfer is to stylize the target text-based binary image T based on a style image S'. In previous text style transfer method [8], distribution prior is a key factor to its success. However, this prior requires that S' has highly structured textures with its non-stylized counterpart S provided. By comparison, we solve a tougher unsupervised style transfer problem, where S is not provided and S' contains arbitrary textures. To meet these challenges, we propose to build a mapping between T and S' using a binary imagery S of S', and gradually narrow their visual discrepancy by structure and texture transfer. Moreover, a saliency cue is introduced for shape legibility.

In particular, instead of directly handling S' and T, we first propose a two-stage abstraction method to abstract a binary imagery S as a bridge based on the color features and contextual information of S' (Section IV-A). Since the textons in S' and the glyphs in T probably do not match, a legibility-preserving structure transfer algorithm is proposed to adjust the contours of S and T (Section IV-B). The resulting $\hat{S}$ and $\hat{T}$ share the same structural features and establish an effective mapping between S' and T. Given $\hat{S}$ and S', we are able to synthesize textures for $\hat{T}$ by objective function optimization (Section IV-C). In addition to the distribution term [8], we further introduce a saliency term in our objective function, which guides our algorithm to stylize the interior of the target shape (white pixels in $\hat{T}$) to be of high saliency while the exterior (black pixels in $\hat{T}$) of low saliency. This principle enables the stylized shape to be highlighted from the background, thereby increasing its legibility.

B. Layout Design

The goal of context-aware layout design is to seamlessly synthesize T into a background image I with the style of S'. We formulate a optimization function to estimate the optimal embedding position $\hat{R}$ of T. And the proposed text-based image stylization is adjusted to incorporate contextual information of I for seamless image synthesis.


(a) S' (b) $\bar{S}'$ (c) SP+$\bar{S}'$ (d) S

(e) raw cluster (f) one-stage (g) two-stage (h) result by [41]

Fig. 3. Guidance map extraction. Our two-stage abstraction method generates guidance maps that well match the texture contours in S'. (a) Input S'. (b) Rough structure abstraction $\bar{S}'$. (c) Super pixels colored by their mean pixel values in (b). (d) Extracted guidance map S. (e)-(g) K-means clustering results of (a)-(c), respectively. (h) Result by multi-scale label-map extraction [41]. Cropped regions are zoomed for better comparison.

In particular, the color statistics of S' and I are first adjusted to ensure color consistency (Section V-A). Then, we seek the optimal position of T in the background image based on four cues of local variance, non-local saliency, coherence across images and visual aesthetics (Section V-B). Once the layout is determined, the background information around T will be collected. Constrained by this contextual information, the target shape is seamlessly synthesized into the background image in an image inpainting manner (Section V-C).

IV. Text-Based Binary Image Style Transfer

A. Guidance Map Extraction

The perception of texture is a process of acquiring abstract imagery, which enables us to see concrete images from the disordered (such as clouds). This inspires us to follow human's abstraction of the texture information to extract the binary imagery S from the source image S'. S serves as a guidance map, where white pixels indicate the reference region for the shape interior (foreground) and black pixels for the shape exterior (background). The boundary of foreground and background depicts the morphological characteristics of the textures in S'. We propose a simple yet effective two-stage method to abstract the texture into the foreground and the background with the help of texture removal technologies.

In particular, we use the Relative Total Variation (RTV) [42] to remove the color variance inside the texture, and obtain a rough structure abstraction $\bar{S}'$. However, texture contours are also smoothed in $\bar{S}'$ (see Fig. 3(b)(f)). Hence, we put forward a two-stage abstraction method. In the first stage, pixels in

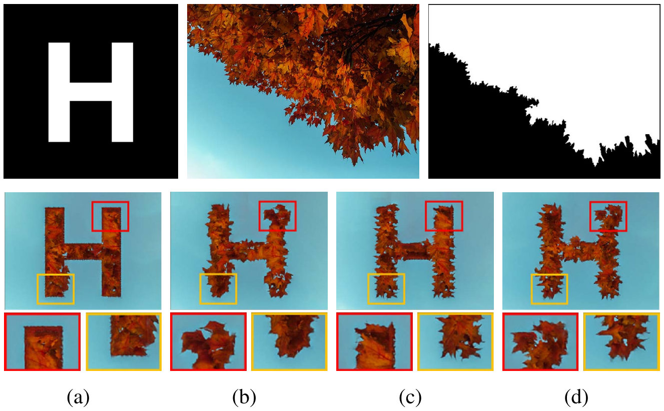


Fig. 4. Benefits of bidirectional structure transfer. The forward transfer simulates the distribution of leaves along the shape boundary, while the backward transfer generates the fine details of each leaf shape. Their combination creates vivid leaf-like typography. The first row: from left to right, T, S' and S. The second row: the text stylization results using (a) original $T + S$, (b) forward transfer $\hat{T} + S$, (c) backward transfer $T + \hat{S}$, and (d) bidirectional transfer $\hat{T} + \hat{S}$. *Image credits: Unsplash users Aaron Burden.*

S' are abstracted as fine-grained super pixels [43] to precisely match the texture contour. Each super pixel uses its mean pixel values in $\bar{S}'$ as its feature vector to avoid the texture variance. In the second stage, the super pixels are further abstracted as the coarse-grained foreground and background via K-means clustering ($K = 2$). Fig. 3 shows an example where our two-stage method generates accurate abstract imagery of the plaster wall. In this example, our result has more details at the boundary than the one-stage method, and fewer errors than the state-of-the-art label-map extraction method [41] (see the zoomed region in Fig. 3(h)).

Finally, we use the saliency as a criterion to determine the foreground and background of the image. Pixel saliency in S' is detected [44] and the cluster with higher mean pixel saliency is set as the foreground. Compared with the commonly used brightness criterion in artistic thresholding methods [45], [46] to retrieve artistic binary images, our criterion helps the foreground text find salient textures.

B. Structure Transfer

Directly using S extracted in Section IV-A and the input T for style transfer results in unnatural texture boundaries as shown in Fig. 4(a). A potential solution could be employing the shape synthesis technique [47] to minimize structural inconsistencies between S and T. In Layered Shape Synthesis (LSS) [47], shapes are represented as a collection of boundary patches at multiple resolution, and the style of a shape is transferred onto another by optimizing a bidirectional similarity function. However, in our application such an approach does not consider the legibility, and the shape will become illegible after adjustment as shown in the second row of Fig. 6. Hence we incorporate stroke trunk protection mechanism into LSS and propose a legibility-preserving structure transfer method.

The main idea is to adjust the shape of the stroke ends while preserving the shape of the stroke trunk, because the legibility of a glyph is mostly determined by the shape of its trunk. Toward this, we extract the skeleton from T and detect the stroke end as a circular region centered at the endpoint of the skeleton. For each resolution level l, we generate a mask $M^{(l)}$ indicating the stroke end regions as shown in Fig. 5. At the top level $l = L$, the radius of the circular region is set to the average radius of the stroke calculated by the method of [8]. The radius increases linearly as the resolution increases. And at the bottom level $l = 0$ (original resolution), it is set to just cover the entire shape. Let $T^{(l)}$, $S^{(l)}$ and $\hat{T}^{(l)}$ denote the downsampled T, downsampled S and the legibility-preserving structure transfer result at level l respectively. Given $M^{(l)}$, $T^{(l)}$, $S^{(l)}$ and $\hat{T}^{(l+1)}$, we calculate $\hat{T}^{(l)}$ by

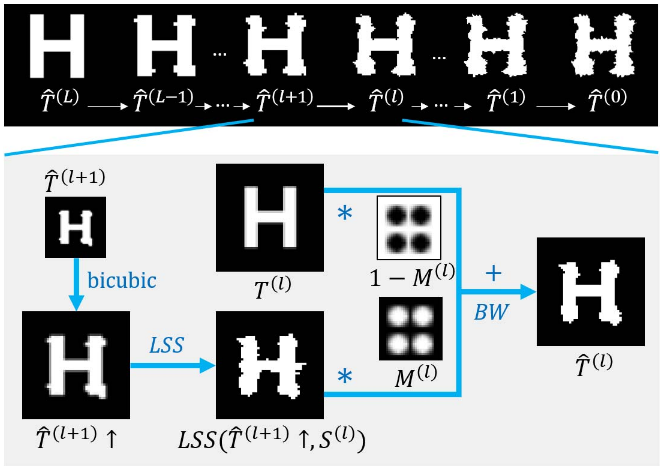


Fig. 5. Preserving the stroke trunks by weighted combination of the trunk region from $T^{(l)}$ and the stroke end region from the shape synthesis result.

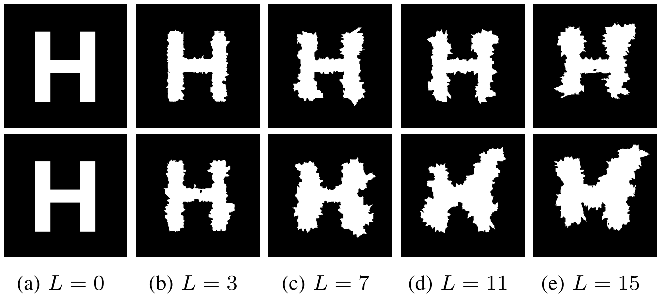

(a) $L = 0$ (b) $L = 3$ (c) $L = 7$ (d) $L = 11$ (e) $L = 15$

Fig. 6. Effects of our stroke trunk protection mechanism and the number L of image pyramid layers. Top row: our legibility-preserving structure transfer result $\hat{T}^{(0)}$. Bottom row: structure transfer result $\hat{T}^{(0)}$ without our stroke trunk protection mechanism. As L increases, the deformation degree also increases. Without the stroke trunk protection mechanism, the shape legibility is severely reduced. 'H' is deformed like 'X' in (d) and 'M' in (e) in the bottom row. In this example, we use S in Fig. 4 as the reference.

$$\hat{T}^{(l)} = BW\Big(M^{(l)} * LSS\big(\hat{T}^{(l+1)}\uparrow, S^{(l)}\big) + (1-M^{(l)}) * T^{(l)}\Big), \tag{1}$$

where $*$ is the element-wise multiplication operator and $\uparrow$ is the bicubic upsampling operator. $LSS(T, S)$ is the shape synthesis result of T given S as the shape reference by LSS, and $BW(\cdot)$ is the binarization operation with threshold 0.5. The pipeline of the proposed structure transfer is visualized in Fig. 5. In our implementation, the image resolution at the top level L is fixed. Therefore the deformation degree is solely controlled by L. We show in Fig. 6 that our stroke trunk protection mechanism effectively balances structural consistency with shape legibility even under very large L.

For characters without stroke ends (*e.g.* "O"), our method automatically reverts to the baseline LSS algorithm. We use an eclectic strategy to keep these characters consistent with the average degree of adjustment of characters with stroke ends. Specifically, during our L-level hierarchical shape transfer, these characters are masked out during the first $L/2$ levels, and then are adjusted using LSS in the following $L/2$ levels.

In addition, we propose a bidirectional structure transfer (Fig. 4(a)-(d)) to further enhance the shape consistency, where a backward transfer is added after the aforementioned forward transfer. The backward transfer migrates the structural style of the forward transfer result $\hat{T}^0$ back to S to obtain $\hat{S}^0$ using the original LSS algorithm. The results $\hat{T}^{(0)}$ and $\hat{S}^{(0)}$ will be used as guidance for texture transfer. For simplicity, we will omit the superscripts in the following.

C. Texture Transfer

In our scenarios, S' is not well-structured text effects, and thus the distribution prior used in [8] to ensure shape legibility takes limited effect. We introduce a saliency cue for compensation. We augment the texture synthesis objective function in [8] with the proposed saliency term as follows,

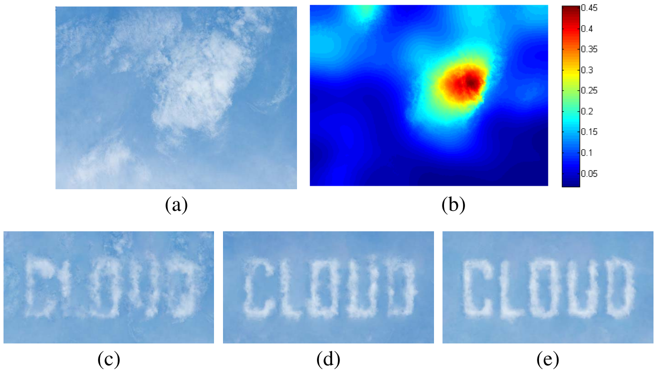
$$\min_q \sum_p E_a(p,q) + \lambda_1 E_d(p,q) + \lambda_2 E_p(p,q) + \lambda_3 E_s(p,q), \tag{2}$$

where p is the center position of a target patch in $\hat{T}$ and T', q is the center position of the corresponding source patch in $\hat{S}$ and S'. The four terms E_a, E_d, E_p and E_s are the appearance, distribution, psycho-visual and saliency terms, respectively, weighted by λs. E_a and E_d constrain the similarity of local texture pattern and global texture distribution, respectively. E_p penalizes texture over-repetitiveness for naturalness. We refer to [8] for details of the first three terms. For the distribution term E_d, we truncate the distance maps of $\hat{S}$ and $\hat{T}$ to a range of $[0.5, 2]$ where distance 1 corresponds to the shape boundaries. By doing so, we relieve the distribution constraint for pixels far away from the shape boundary. And these pixels are mainly controlled by our saliency cue,

$$E_s(p,q) = \begin{cases} W(p)\cdot Sal(q), & \text{if } T(p)=0 \\ W(p)\cdot(1-Sal(q)), & \text{if } T(p)=1 \end{cases} \tag{3}$$

where $Sal(q)$ is the saliency at pixel q in S'. $W(p) = 1-\exp(-dist(p)^2/2\sigma_1^2)/2\pi\sigma_1^2$ is the gaussian weight with $dist(p)$, the distance of p to the shape boundary. The saliency term encourages pixels inside the shape to find salient textures for synthesis and keeps the background less salient. We show in Fig. 7 that a higher weight of our saliency term makes the stylized shape more prominent.

Similar to [8], we take the iterative coarse-to-fine matching and voting steps as in [24]. In the matching step, PatchMatch algorithm [48] is adopted to solve (2).



(a) (b)

(c) (d) (e)

Fig. 7. Effects of the saliency term in texture transfer. The saliency term makes the foreground character more prominent and the background cleaner, thereby enhancing the legibility. *Image credits: Unsplash users Ashim D'Silva.* (a) S'. (b) Saliency map of S'. (c) $\lambda_3 = 0.00$. (d) $\lambda_3 = 0.01$. (e) $\lambda_3 = 0.05$.

V. Context-Aware Layout Design

A. Color Transfer

Obvious color discontinuities may appear for style images S' that have a different color from the background I. Therefore we employ color transfer technology. Here we use a linear method introduced by Image Analogies color transfer [11]. This technique estimates a color affine transformation matrix and a bias vector which match the target mean and standard deviation of the color feature with the source ones. In general, color transfer in a local manner is more robust than the global method. Hence, we employ the perception-based color clustering technique [15] to divide pixels into eleven color categories. The linear color transfer is performed within corresponding categories between S' and I. More sophisticated methods or user interactions could be optionally employed to further improve the color transfer result.

B. Position Estimation

In order to synthesize the target shape seamlessly into the background image, the image layout should be properly determined. In the literature, similar problems in cartography are studied to place text labels on maps [49]. Viewed as an optimization problem, they only consider the overlap between labels. In this paper, both the seamlessness and aesthetics of text placement are taken into account. Specifically, we formulate a cost minimization problem for context-aware position estimation by considering the cost of each pixel x of I in four aspects,

$$\hat{R} = \arg\min_R \sum_{x\in R} U_v(x) + U_s(x) + U_c(x) + \lambda_4 U_a(x), \tag{4}$$

where R is a rectangular area of the same size as T, indicating the embedding position. The position is estimated by searching an $\hat{R}$ where pixels have the minimum total costs. U_v and U_s are local variance and non-local saliency costs, concerning the background image I itself, and U_c is a coherence cost measuring the coherence between I and S'. In addition, U_a is the aesthetics cost for subjective evaluation, weighted by λ_4. Here all terms are normalized independently. We use equal

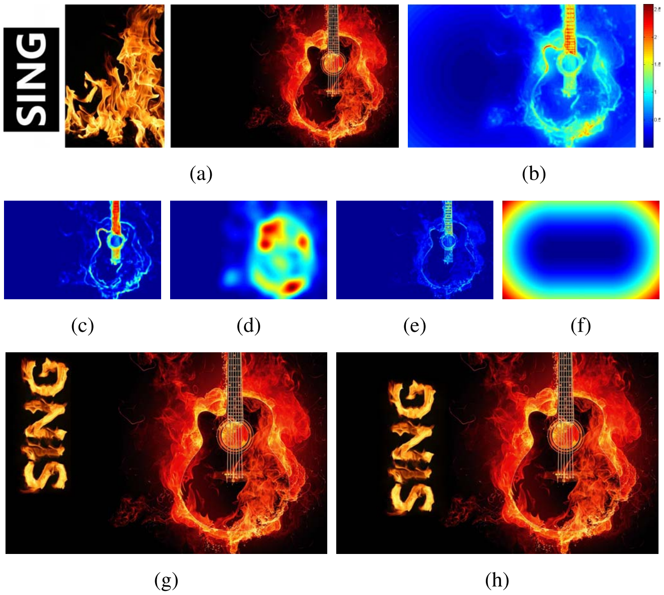


Fig. 8. Image layout is determined by jointly considering a local variance cost, a non-local saliency cost, a coherence cost and an aesthetics cost. *Image credits: Unsplash users Dark Rider.* (a) Input T, S' and I. (b) Total placement cost. (c) Variance. (d) Saliency. (e) Coherence. (f) Aesthetics. (g) Result without the aesthetics cost. (h) Result with the aesthetics cost.

weights for the first three terms, and a lower weight $\lambda_4 = 0.5$ for the aesthetics term.

We consider the local and non-local cues of I. First, we seek flat regions for seamless embedding by using $U_v(x)$ as the intensity variance within a local patch centered at x. Then, a saliency cost $U_s(x) = Sal(x)$ which prefers non-salient regions is introduced. These two internal terms preclude our method from overlaying important objects in the background image with the target shape.

In addition, we use the mutual cost U_c to measure the texture consistency between I and S'. More specifically, $U_c(x)$ is obtained by calculating the $L2$ distance between the patch centered at x in I and its best matched patch in S'.

So far, both the internal and mutual cues are modeled. However, a model that considers only the seamlessness may find unimportant image corners for the target shape, which is not ideal for aesthetics as shown in Fig. 8(g). Hence, we also model the centrality of the shape by U_a

$$U_a(x) = 1 - \exp\left(-dist(x)^2/2\sigma_2^2\right), \tag{5}$$

where $dist(x)$ is the offset of x to the image center, and σ_2 is set to the length of the short side of I. Fig. 8 visualizes these four costs, which jointly determine the ideal image layout.

As for the minimization of (4), we use the box filter to effectively solve the total costs for every valid R throughout I, and choose the minimum one.

We further consider the scales and rotations of text. In addition, when the target image contains multiple shapes (which is quite common for text), we investigate the placement of each individual shape rather than considering them as a whole, which greatly enhances the design flexibility.

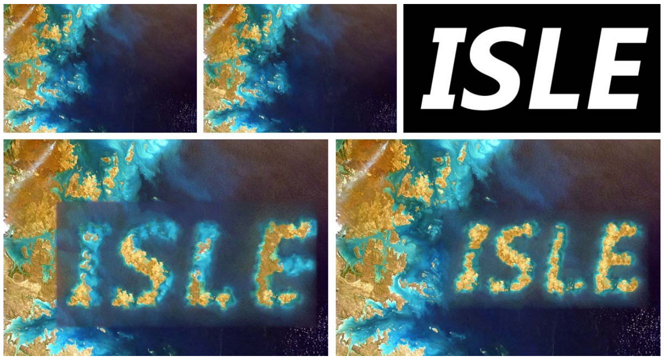


Fig. 9. Text scale in position estimation. Top row: Three images from left to right are I, S' and T, respectively. Bottom row: the result image layout using T of the original size (left) and the result image layout using T of the optimal size (right). *Image credits: Unsplash user NASA.*

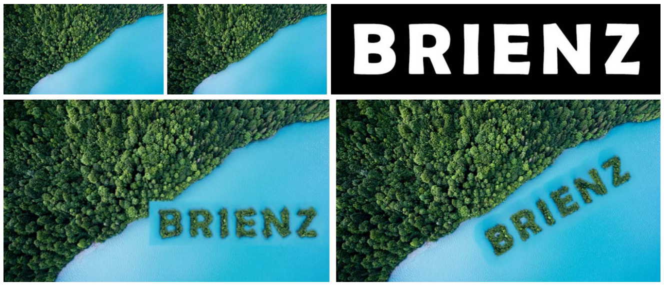


Fig. 10. Text rotation in position estimation. Top row: Three images from left to right are I, S' and T, respectively. Bottom row: the result image layouts without text rotation (left) and with text rotation (right). *Image credits: Unsplash user Andreas Gucklhor.*

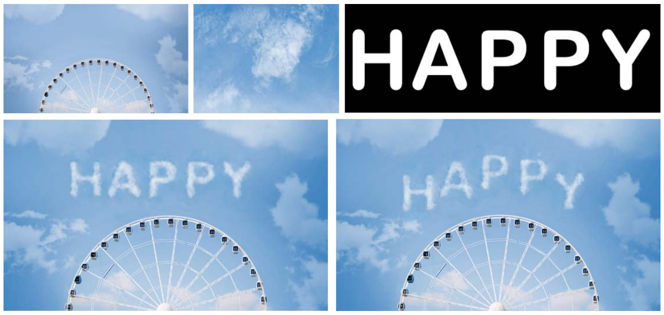
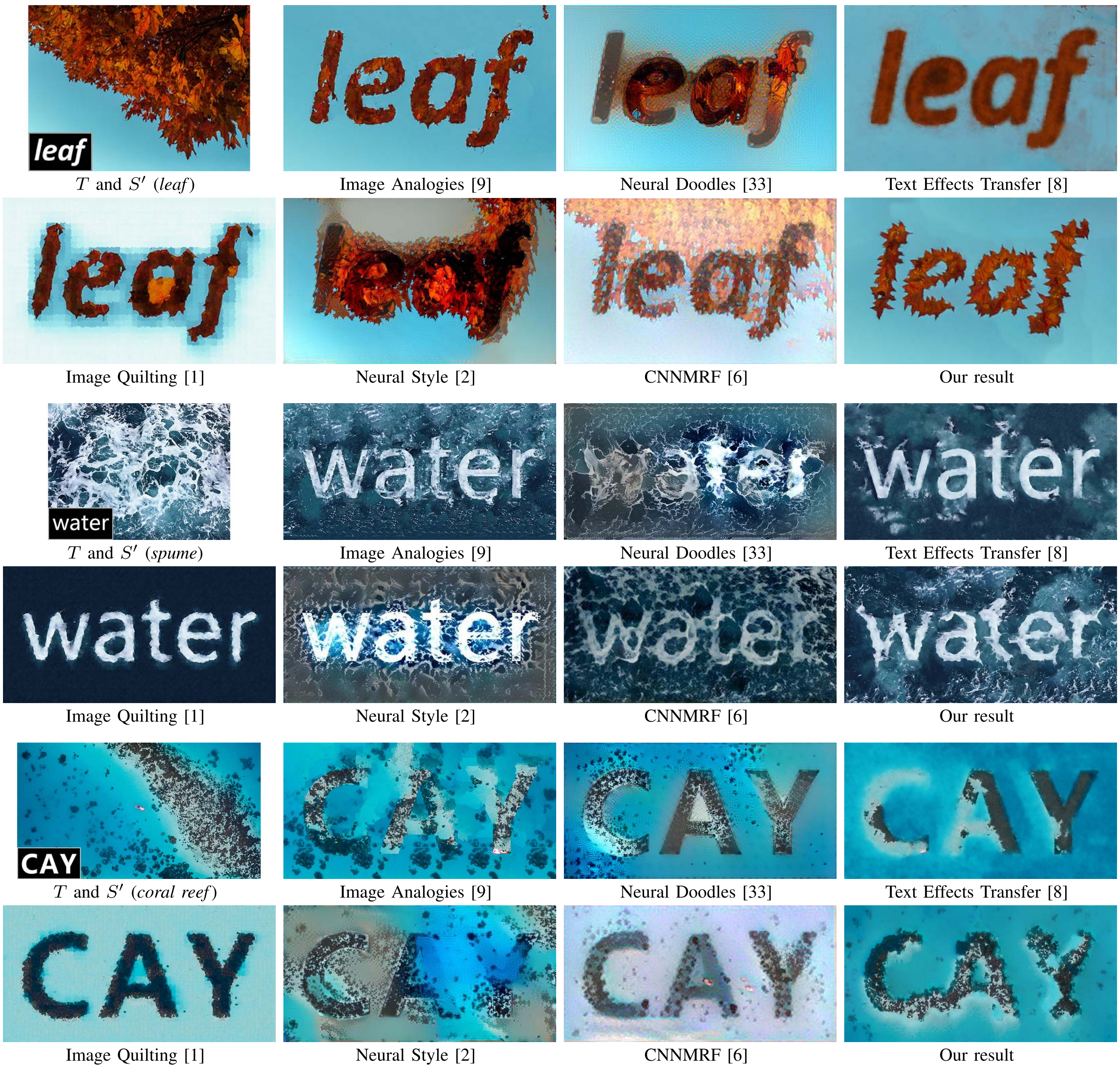


Fig. 11. Multi-shape placement in position estimation. Top row: Three images from left to right are I, S' and T, respectively. Bottom row: the result image layouts without layout refinement (left) and with layout refinement (right).

1) Text Scale: Our method can be easily extended to handle different scales. During the box filtering, we enumerate the size of the box and then find the global minimum penalty point throughout the space and scale. Specifically, we enumerate a scaling factor ψ_s over a range of [0.8, 1.2] in steps of 0.1. Then the text box R is zoomed in or out based on ψ_s to obtain $\psi_s(R)$. Finally, the optimal embedding position and scale can be detected by

$$\hat{R}, \hat{\psi_s} = \arg\min_{R,\psi_s} \sum_{x\in\psi_s(R)} \frac{U_v(x)+U_s(x)+U_c(x)+\lambda_4 U_a(x)}{|\psi_s(R)|}. \tag{6}$$

Fig. 9 shows an example where the target image T is originally too large and is automatically adjusted by the proposed method so that it could be seamlessly embedded into the background.



T and S' (*leaf*) | Image Analogies [9] | Neural Doodles [33] | Text Effects Transfer [8]

Image Quilting [1] | Neural Style [2] | CNNMRF [6] | Our result

T and S' (*spume*) | Image Analogies [9] | Neural Doodles [33] | Text Effects Transfer [8]

Image Quilting [1] | Neural Style [2] | CNNMRF [6] | Our result

T and S' (*coral reef*) | Image Analogies [9] | Neural Doodles [33] | Text Effects Transfer [8]

Image Quilting [1] | Neural Style [2] | CNNMRF [6] | Our result

Fig. 12. Visual comparison of text stylization. For each result group, the first one is the input source style and target text. Other images are results by supervised Image Analogies [9], Neural Doodles [33], Text Effects Transfer [8] (the upper row) and unsupervised Image Quilting [1], Neural Style [2], CNNMRF [6], our methods (the lower row). For supervised methods, the structure guidance map extracted by our method is directly given as input. More examples can be found in the supplementary material. *Image credits: Unsplash users Aaron Burden, Cassie Matias, Ishan@seefromthesky.*

2) Text Rotation: Similar to the text scale, we enumerate the rotation angle ψ_r over a range of $[-\pi/6, \pi/6]$ in steps of $\pi/60$, and find the global minimum penalty point in the entire space and angle. To use the box filter for fast solution, instead of rotating R, we choose to rotate the cost map $\mathbf{U} = U_v(x) + U_s(x) + U_c(x) + \lambda_4 U_a(x)$ by $-\psi_r$, and perform box filter on the rotated $\mathbf{U}$, followed by minimum point detection. Fig. 10 shows an example where the target image T is automatically rotated to match the direction of the coastline.

3) Multiple Shapes: To deal with multiple shapes, we first view them as a whole and optimize (4) to search an initial position and then refine their layouts separately. In each refinement step, every shape searches for the location with the lowest cost within its small spatial neighborhood to update its original position. After several steps, all the shapes converge to their respective optimal positions. In order to prevent the shapes from overlapping, the search space is limited to ensure the distance between adjacent shapes is not less than their initial distance. Fig. 11 shows that after layout refinement, the characters on the left and right sides are adjusted to a more central position in the vertical direction, making the overall text layout better match the shape of the Ferris wheel.

It is worth noting that the above three extensions can be combined with each other to provide users with more flexible layout options.

C. Shape Embedding

Once the layout is determined, we synthesize the target shape into the background image in an image inpainting manner. Image inpainting technologies [50]–[52] have long been investigated in image processing literature to fill the unknown parts of an image. Similarly our problem sets $\hat{R}$ as the unknown region of I, and we aim to fill it with the textures of S' under the structure guidance of $\hat{T}$ and $\hat{S}$. We first enlarge $\hat{R}$ by expanding its boundary by 32 pixels. Let the augmented frame-like region be denoted as $\hat{R}^+$, and the pixel values of I in $\hat{R}^+$ provide contextual information for the texture transfer. Throughout the coarse-to-fine texture transfer process described in Section IV-C, each voting step is followed by replacing the pixel values of T' in $\hat{R}^+$ with the contextual information $I(\hat{R}^+)$. This manner will enforce a strong boundary constraint to ensure a seamless transition at the boundary.

VI. Experimental Results and Analysis

A. Comparison of Style Transfer Methods

In Fig. 12, we present a comparison of our method with six state-of-the-art supervised and unsupervised style transfer techniques on text stylization. For supervised methods, the structure guidance map S extracted by our method in Section IV-A is directly given as input. Please enlarge and view these figures on the screen for better comparison.

1) Structural Consistency: In comparison to these approaches, our method can preserve the critical structural characteristics of textures in the source style image. Other methods do not consider to adapt the stroke contour to the source textures. As a result, they fail to guarantee structural consistency. For example, text boundaries in most methods are rigid in the *leaf* group of Fig. 12. By comparison, Neural Style [2] and CNNMRF [6] implicitly characterize the texture shapes using deep-based features, while our method explicitly transfers structural features. Therefore only these three approaches create leaf-like letters. Similar cases can also be found in the *spume* and *coral reef* groups of Fig. 12. The structural consistency achieved by our method can be better observed in the zoomed regions in Fig. 13, where even Neural Style [2] does not appear to transfer structure effectively.

2) Text Legibility: For supervised style transfer approaches, the binary guidance map can only provide rough background/foreground constraints for texture synthesis. Consequently, the background of the *island* result by Image Analogies [9] finds many salient repetitive textures to fill, which confuses itself with the foreground. Text Effects Transfer [8] introduces an additional distribution constraint for text legibility, which, however, is not effective for pure texture images. For example, due to the scale discrepancy between S' and T, the distribution constraint forces Text Effects Transfer [8] to place textures compactly inside the text, leading to textureless artifacts in *leaf* and *spume* results. Our method further proposes a complimentary saliency constraint, resulting in the creation of the artistic text that highlights from clean backgrounds. We show in Fig. 14 that when the foreground and background colors are not contrasting enough, our approach demonstrates greater superiority.

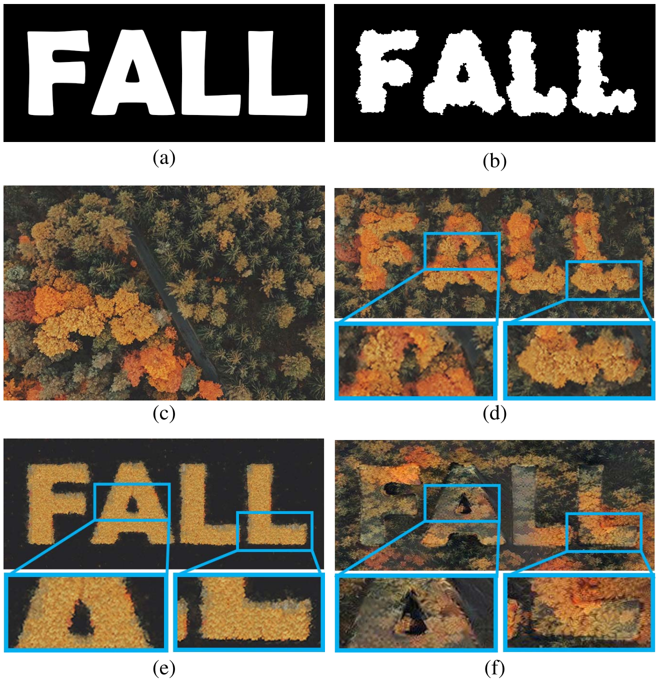


Fig. 13. Visual comparison with unsupervised style transfer methods. By structure transfer, our result better characterizes the shape of the forest canopy. Cropped regions are zoomed for better comparison. *Image credits: Unsplash users Jakub Sejkora.* (a) Target text T. (b) Adjusted text $\hat{T}$. (c) Source style S'. (d) Our result. (e) Result of Image Quilting [1]. (f) Result of Neural Style [2].

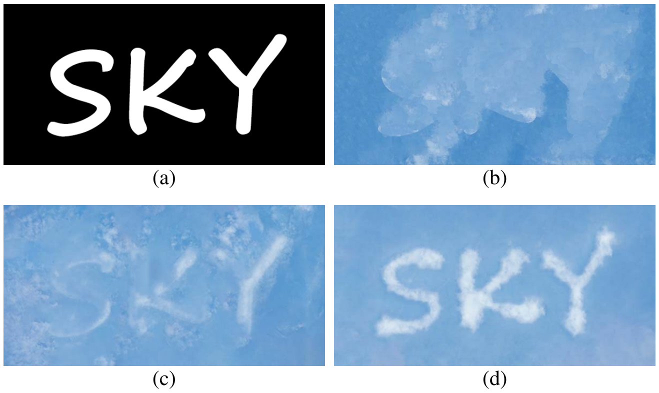


Fig. 14. Visual comparison with supervised style transfer methods. Our method yields the most distinct foreground against the background, thus well preserving shape legibility. In this example, S' in Fig. 7 is used. (a) Target text T. (b) Result of Image Analogies [9]. (c) Result of Text Effects Transfer [8]. (d) Our result.

3) Texture Naturalness: Compared with other style transfer methods, our method produces visually more natural results. For instance, our method places irregular coral reefs of different densities based on the shape of the text in the example *coral reef* of Fig. 12, which highly respects the contents of S'. This is achieved by our context-aware style transfer to ensure structure, color, texture and semantic consistency. By contrast, Image Quilting [1] relies on patch matching between two completely different modalities S' and T, thus its results are

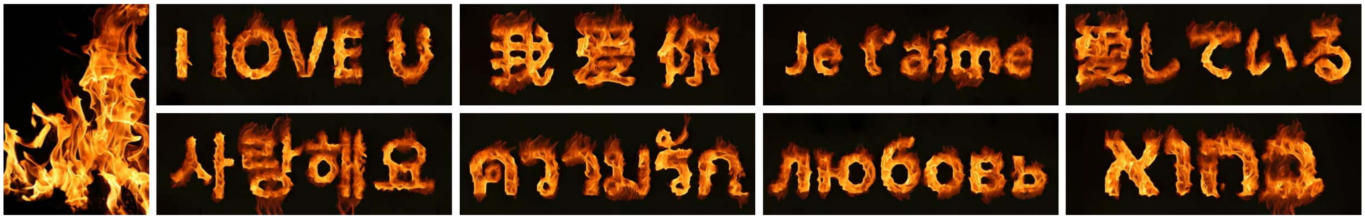


Fig. 15. Performance on different languages. The leftmost image is the source style S'.

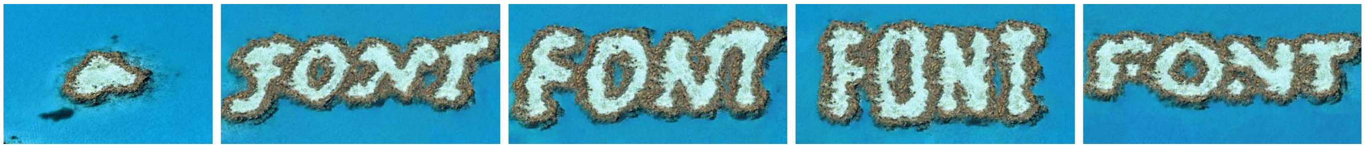
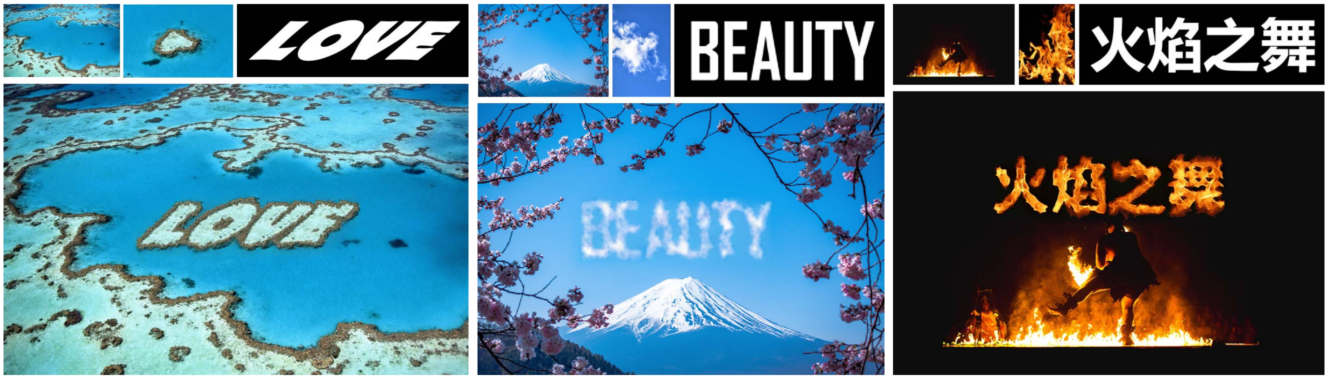


Fig. 16. Performance on different fonts. The leftmost image is the source style S'. *Image credits: Unsplash users Yanguang Lan.*



barrier reef *cloud* *flame*

Fig. 17. Visual-textual presentation synthesis. For each result group, three images in the upper row are I, S' and T, respectively. The lower one is our result. More examples can be found in the supplementary material. *Image credits: Unsplash users Yanguang Lan, JJ Ying, Tim Gouw and Thomas Kelley.* (a) *barrier reef*. (b) *cloud*. (c) *flame*.

just as flat as the raw text. Three deep-learning based methods, Neural Style [2], CNNMRF [6] and its supervised version Neural Doodles [33], transfer suitable textures onto the text. However, their main drawbacks are the color deviation and checkerboard artifacts (see example *coral reef* in Fig. 12).

B. Generating Stylish Text in Different Fonts and Languages

We experimented on text in various fonts and languages to test the robustness of our method. Some results are shown in Figs. 15-16. The complete results can be found in the supplementary material. In Fig. 15, characters are quite varied in different languages. Our method successfully synthesizes dancing flames onto a variety of languages, while maintaining their local fine details, such as the small circles in Thai. In Fig. 16, the rigid outlines of the text are adjusted to the shape of a coral reef, without losing the main features of its original font. Thanks to our stroke trunk protection mechanism, our approach balances the authenticity of textures with the legibility of fonts.

C. Visual-Textual Presentation Synthesis

We aim to synthesize professional looking visual-textual presentation that combines beautiful images and overlaid stylish text. In Fig. 17, three visual-textual presentations automatically generated by our method are provided. In the example *barrier reef*, a LOVE-shaped barrier reef is created, which is visually consistent with the background photo. We further show in the example *cloud* that we can integrate completely new elements into the background. Clouds with a specific text shape are synthesized in the clear sky. The colors in the sky of S' are adjusted to match those in the background, which effectively avoids abrupt image boundaries. Please note the text layout automatically determined by our method is quite reasonable. Therefore, our approach is capable of artistically embellishing photos with meaningful and expressive text and symbols, thus providing a flexible and effective tool to create original and unique visual-textual presentations. This art form can be employed in posters, magazine covers and many other media. We show in Fig. 20 a poster design example, where its stylish headline is automatically generated by our method and the main body is manually designed. A headline made of clouds effectively enhances the attractiveness of the poster.

D. Symbol and Icon Rendering

The proposed method has the ability to render textures for text-based geometric shapes such as symbols and icons.

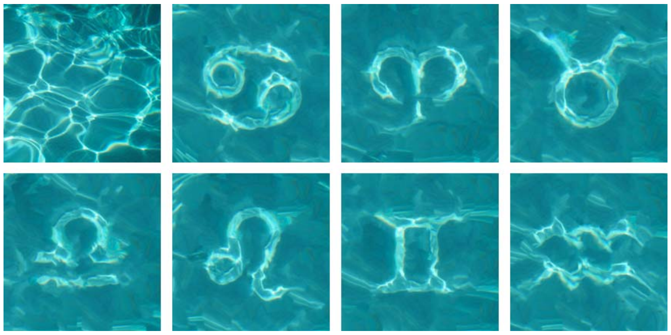


Fig. 18. Rendering rippling Zodiac symbols using a photo of water on the top left. *Image credits: Unsplash users Raphaël Biscaldi.*

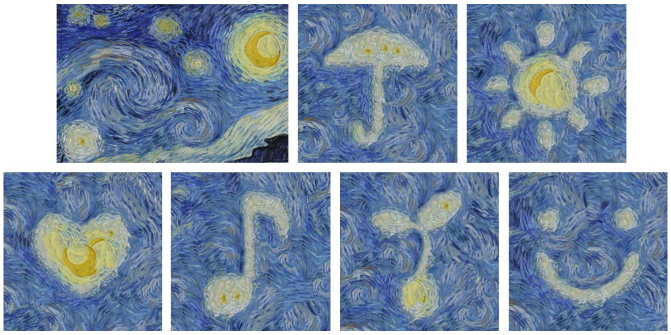


Fig. 19. Rendering emoji icons with the painting style of Van Gogh using "*The Starry Night*" on the top left.

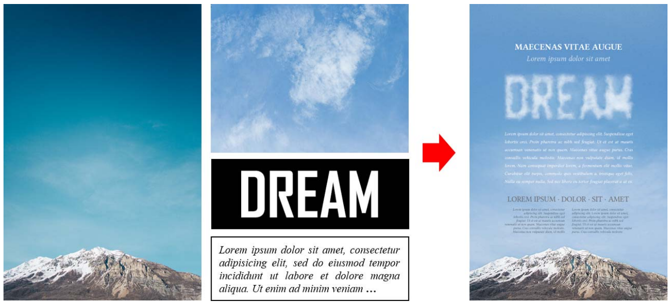
Fig. 18 shows that our method successfully transfers rippling textures onto the binary Zodiac symbols. It seems that the proposed method is also capable of stylizing more general shapes, like the emoji icons in Fig. 19. Meanwhile, we notice that our saliency term selects the prominent orange moon to be synthesized into the *sun* and *heart*, which enriches the color layering of the results.

E. Structure-Guided Image Inpainting

Our method can be naturally extended to structure-guided image inpainting. Fig. 21 demonstrates the feasibility of controlling the inpainting result via user-specified shapes. The input photo in Fig. 21(a) is used as both a style image and a background image, where its red mask directly indicates the embedding position. The user sketches in white the approximate shapes of the branches (shown in the upper right corner of the result), and the resulting sketch serves as target T for stylization. Figs. 21(b)(c) are our inpainting results, which are quite plausible and the filled regions blend well with the background.

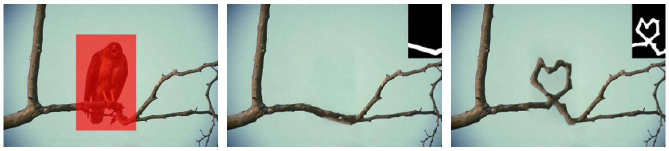
F. Running Time

When analyzing the complexity of the proposed method, we consider the time of guidance map extraction, position estimation and color/structure/texture transfer. To simplify the analysis, we assume the target image T has N pixels, and the image resolution of S' and I have the same magnitude $O(N)$ as T. In addition, the patch size and the number of iterations are constants that can be ignored in computational complexity.



DREAM
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ...
MAECENAS VITAE AUGUE
Lorem ipsum dolor sit amet
DREAM
LOREM IPSUM · DOLOR · SIT · AMET

Fig. 20. Application: Computer aided poster design. *Image credits: Unsplash users Evan Kirby, Ashim D'Silva.*



(a) (b) (c)

Fig. 21. Structure-guided image inpainting. (a) Input. (b) Inpainting result. (c) Inpainting result.

1) Guidance Map Extraction: According to [42]–[44], the complexity of RTV, super pixel extraction and saliency detection is $O(N)$. K-means has a practical complexity of $O(NKt)$ [53], where $K = 2$ and t is the number of iterations. Ignoring K and t, the total complexity of guidance map extraction is $O(N)$.

2) Position Estimation: The complexity of calculating U_v, U_s, U_a and box filter is $O(N)$. For coherence cost U_c, we use FLANN [54] to search matched patches between S' and I, which has a complexity of $O(N \log N)$. Therefore, the overall complexity of the proposed position estimation is $O(N \log N)$.

3) Style Transfer: According to [11], color transfer has an $O(N)$ complexity. During structure transfer, patches along the shape boundary are matched using FLANN [54]. The upper bound of the patch number is $O(N)$ and thus the proposed structure transfer is $O(N \log N)$ complex. As reported in [48], PatchMatch in texture transfer has a complexity of $O(N \log N)$.

In summary, the overall computational complexity of the proposed method is $O(N \log N)$.

Table I shows the running time of our method on three test sets (Fig. 17) with Intel Xeon 3.00 GHz CPU E5-1607. The proposed method is implemented on MATLAB platform. Texture transfer is our major computational bottleneck, which accounts for about 85% of the total time. This is because matching patches in mega-pixel (Mp) images can be slow at finer scales. As our method is not multithreaded, it just uses a single core. Our method can be further speeded up by implementing a well-tuned and fully parallelized PatchMatch algorithm.

G. Limitation

While our approach has generated visually appealing results, some limitations still exist. Our guidance map extraction relies

TABLE I

RUNNING TIME (SECONDS) OF THE PROPOSED METHOD

Test set	T	S'	I	Guidance extraction	Position estimation	Color transfer	Structure transfer	Texture transfer	Total
barrier reef	975×308	566×377	1920×1200	11.08	28.11	-	6.43	255.93	301.55
cloud	477×197	500×750	900×600	16.86	21.76	15.23	6.44	247.37	307.85
flame	916×300	400×575	1556×1024	15.87	13.63	-	17.37	314.15	361.02
Averaged	0.22Mp	0.27Mp	1.5Mp	14.60	21.17	15.23	10.15	272.48	323.47

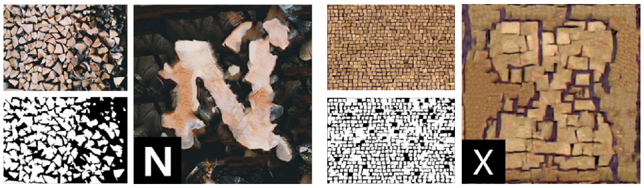


Fig. 22. Some examples of failure cases. *Image credits: Unsplash users Brigitte Tohm, Edouard Dognin.*

on simple color features, and is not fool-proof. Main structure abstraction from complex scenarios might be beyond its capability. The problem may be addressed by either employing high-level deep features or user interactions. Moreover, even with precise guidance maps, our method may fail when the source style contains tightly spaced objects. As shown in Fig. 22, our method yields interesting results, but they do not correctly reflect the original texture shapes. The main reason is that it is hard for our patch-based method to find pure foreground or background patches in the dense patterns for shape and texture synthesis. Therefore, when synthesizing the foreground (background) region, shapes/textures in the background (foreground) region will be used, which causes mixture and disorder.

VII. CONCLUSION AND FUTURE WORK

In this paper, we demonstrate a new technique for text-based binary image stylization and synthesis to incorporate binary shape and colorful images. We exploit guidance map extraction to facilitate the structure and texture transfer. Cues for seamlessness and aesthetics are leveraged to determine the image layout. Our context-aware text-based image stylization and synthesis approach breaks through a barrier between images and shapes, allowing users to create fine artistic shapes and to design professional looking visual-textual presentations.

There are still some interesting issues for further investigation. A direction for future work is the automatic style image selection. S' that shares visual consistency and semantic relevance with the background image will contribute more to seamless embedding and aesthetic interests. Recommendation of S' could be achieved by leveraging deep neural networks to extract semantic information.

REFERENCES

[1] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. ACM Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 341–346.

[2] 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.*, Jun. 2016, pp. 2414–2423.

[3] O. Frigo, N. Sabater, J. Delon, and P. Hellier, "Split and match: Example-based adaptive patch sampling for unsupervised style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 553–561.

[4] M. Elad and P. Milanfar, "Style transfer via texture synthesis," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2338–2351, May 2017.

[5] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 120.

[6] 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.*, Jun. 2016, pp. 2479–2486.

[7] 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.*, Jul. 2017, pp. 1897–1906.

[8] 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.*, Jul. 2017, pp. 7464–7473.

[9] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 327–340.

[10] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.

[11] A. Hertzmann, "Algorithms for rendering in artistic styles," Ph.D. dissertation, Dept. Comput. Sci., New York Univ., New York, NY, USA, 2001.

[12] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Comput. Vis. Image Understand.*, vol. 107, nos. 1–2, pp. 123–137, 2007.

[13] Y. W. Tai, J. Y. Jia, and C. K. Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 747–754.

[14] Y.-W. Tai, J. Jia, and C.-K. Tang, "Soft color segmentation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1520–1537, Sep. 2007.

[15] Y. Chang, S. Saito, K. Uchikawa, and M. Nakajima, "Example-based color stylization of images," *ACM Trans. Appl. Perception*, vol. 2, no. 3, pp. 322–345, 2006.

[16] Y. Chang, S. Saito, and M. Nakajima, "Example-based color transformation of image and video using basic color categories," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 329–336, Feb. 2007.

[17] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 277–280, Jul. 2002.

[18] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, "Data-driven hallucination of different times of day from a single outdoor photo," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 2504–2507, 2013.

[19] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, "Style transfer for headshot portraits," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 148.

[20] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Trans. Graph.*, vol. 35, no. 1, 2016, Art. no. 11.

[21] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1033–1038.

[22] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127–150, 2001.

[23] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, 2003.

[24] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[25] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 262–270.

[26] P. Bénard *et al.*, "Stylizing animation by example," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 119.

[27] C. Barnes, F.-L. Zhang, L. Lou, X. Wu, and S.-M. Hu, "PatchTable: Efficient patch queries for large datasets and applications," *ACM Trans. Graph.*, vol. 34, no. 4, p. 97, 2015.

[28] S. Zhang, X. Gao, N. Wang, and J. Li, "Robust face sketch style synthesis," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 220–232, Jan. 2016.

[29] S. Zhang, X. Gao, N. Wang, J. Li, and M. Zhang, "Face sketch synthesis via sparse representation-based greedy search," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2466–2477, Aug. 2015.

[30] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1105–1114.

[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.*, Jul. 2017, pp. 3985–3993.

[32] 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.

[33] A. J. Champandard. (Mar. 2016). "Semantic style transfer and turning two-bit doodles into fine artworks." [Online]. Available: https://arxiv.org/abs/1603.01768

[34] X. Xu, L. Zhang, and T.-T. Wong, "Structure-based ascii art," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 52-1–52-9, Jul. 2010.

[35] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir and N. Carr, "Digital micrography," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 100:1–100:12, 2011.

[36] C. Zou *et al.*, "Legible compact calligrams," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 122-1–122-12, Jul. 2016.

[37] T. S. F. Haines, O. M. Aodha, and G. J. Brostow, "My text in your handwriting," *ACM Trans. Graph.*, vol. 35, no. 3, pp. 26-1–26-18, May 2016.

[38] Z. Lian, B. Zhao, and J. Xiao, "Automatic generation of large-scale handwriting fonts via style learning," in *Proc. ACM SIGGRAPH ASIA Tech. Briefs*, 2016, pp. 12-1–12-4.

[39] J. Lu, C. Barnes, C. Wan, P. Asente, R. Mech, and A. Finkelstein, "Decobrush: Drawing structured decorative patterns by example," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 90.

[40] J. Lu, F. Yu, A. Finkelstein, and S. Diverdi, "Helpinghand: Example-based stroke stylization," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 13–15, 2012.

[41] Y. D. Lockerman, B. Sauvage, R. Allègre, J.-M. Dischler, J. Dorsey, and H. Rushmeier, "Multi-scale label-map extraction for texture synthesis," *ACM Trans. Graph.*, vol. 35, no. 4, p. 140, 2016.

[42] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, p. 139, Nov. 2012.

[43] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[44] J. Zhang and S. Sclaroff, "Saliency detection: A Boolean map approach," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 153–160.

[45] D. Mould and K. Grant, "Stylized black and white images from photographs," in *Proc. Int. Symp. Non-Photorealistic Animation Rendering*, 2008, pp. 49–58.

[46] J. Xu and C. S. Kaplan, "Artistic thresholding," in *Proc. Int. Symp. Non-Photorealistic Animation Rendering*, 2008, pp. 39–47.

[47] 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. 107-1–107-9, 2009.

[48] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 341–352, Aug. 2009.

[49] J. Christensen, J. Marks, and S. Shieber, "An empirical study of algorithms for point-feature label placement," *ACM Trans. Graph.*, vol. 14, no. 3, pp. 203–232, 1995.

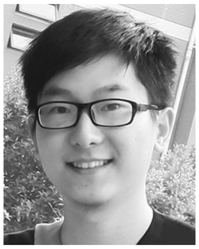
[50] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM Trans. Graph.*, 2000, pp. 417–424.

[51] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[52] O. Le Meur, M. Ebdelli, and C. Guillemot, "Hierarchical super-resolution-based inpainting," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3779–3790, Oct. 2013.

[53] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proc. IEEE Int. Conf. Mach. Learn.*, Jun. 2003, pp. 147–153.

[54] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

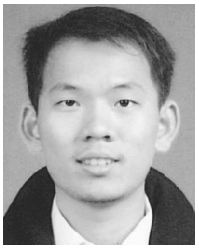


Shuai Yang received the B.S. degree in computer science from Peking University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology, Peking University, Beijing, China.

His current research interests include image inpainting, depth map enhancement, and image stylization.

Jiaying Liu (S'08–M'10–SM'17) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, and the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2005 and 2010, respectively. She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. She has authored over 100 technical articles in refereed journals and proceedings and holds 28 granted patents. Her current research interests include image/video processing, compression, and computer vision.

Dr. Liu was a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, from 2007 to 2008. She was a Visiting Researcher at Microsoft Research Asia supported by the Star Track for Young Faculties in 2015. She has also served as a TC member in the IEEE CAS-MSA/EOT and APSIPA IVM, and APSIPA Distinguished Lecturer from 2016 to 2017. She is a CCF Senior Member.



Wenhan Yang (S'17) received the B.S. and Ph.D. degrees (Hons.) in computer science from Peking University, Beijing, China, in 2012 and 2018, respectively. He is currently a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. He was a Visiting Scholar with the National University of Singapore from 2015 to 2016. His current research interests include deep-learning based image processing, bad weather restoration, related applications and theories.

Zongming Guo (M'09) received the B.S. degree in mathematics, and the M.S. and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1987, 1990, and 1994, respectively.

He is currently a Professor with the Institute of Computer Science and Technology, Peking University. His current research interests include video coding, processing, and communication.

Dr. Guo is an Executive Member of the China Society of Motion Picture and Television Engineers. He was a recipient of the First Prize of the State Administration of Radio Film and Television Award in 2004, the First Prize of the Ministry of Education Science and Technology Progress Award in 2006, the Second Prize of the National Science and Technology Award in 2007, and the Wang Xuan News Technology Award and the Chia Tai Teaching Award in 2008. He received the Government Allowance granted by the State Council in 2009. He received the Distinguished Doctoral Dissertation Advisor Award from Peking University in 2012 and 2013, respectively.