Context-Aware Unsupervised Text Stylization

Shuai Yang
Institute of Computer Science and Technology, Peking University
williamyang@pku.edu.cn

Jiaying Liu*
Institute of Computer Science and Technology, Peking University
liujiaying@pku.edu.cn

Wenhan Yang
Institute of Computer Science and Technology, Peking University
yangwenhan@pku.edu.cn

Zongming Guo
Institute of Computer Science and Technology, Peking University
guozongming@pku.edu.cn

ABSTRACT

In this work, we present a novel algorithm to stylize the text without supervision, which provides a flexible and convenient way to invoke fantastic text expressions. Rather than employing the fixed pair of target text and source style images, our unsupervised framework establishes an implicit mapping for them by using an abstract imagery of the style image as bridges. Based on the mapping, we progressively narrow the visual discrepancy between text and style images by the proposed legibility-preserving structure transfer and texture transfer algorithms, which effectively balance the text legibility and style consistency. Furthermore, we explore a seamless composition of the stylized text and a background image, in which the optimal text layout is determined by a context-aware layout design algorithm utilizing cues for both seamlessness and aesthetics. Given the layout, the text can be seamlessly embedded into the background by texture synthesis under a context-aware boundary constraint. Experimental results demonstrate the effectiveness of the proposed method in automatic artistic typography creation and visual-textual presentation synthesis.

CCS CONCEPTS

• **Computing methodologies** → **Texturing**; *Image processing*; • **Applied computing** → Text editing;

KEYWORDS

Texture synthesis; structure synthesis; context-aware; style transfer; unsupervised method

ACM Reference Format:
Shuai Yang, Jiaying Liu, Wenhan Yang, and Zongming Guo. 2018. Context-Aware Unsupervised Text Stylization. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240580

*Corresponding author

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

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240580

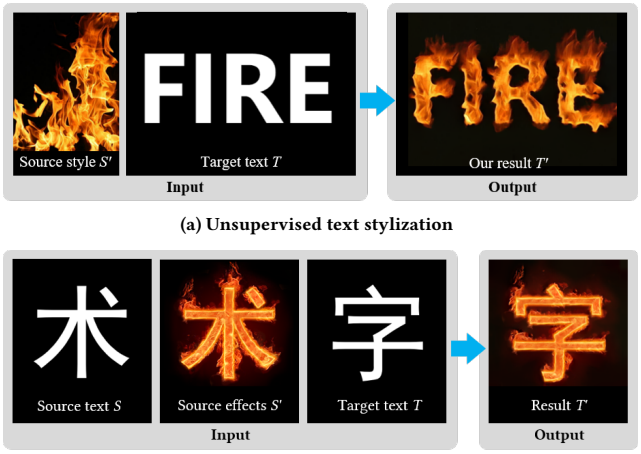


Figure 1: Supervised text stylization requires registered raw text S and text effects S' as input. We instead handle a more challenging unsupervised problem with an arbitrary source image S'.

1 INTRODUCTION

Style transfer [10, 15, 19, 22] is the task of migrating styles from a style image to a content image 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. Style transfer for content images of specified types is studied including human faces [30, 31], urban cities [32], and text [37]. Among them, text images highly summarize concepts in the real world. Text decorated by well-designed textures can convey much more visual information and better reflect the meaning of the text as in Figure 1(a). Thus the stylization of text images is of great research value. But it also poses a challenge of narrowing the great visual discrepancy between the binary flat text 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 [10] and the parametric method Neural Style [15]. Non-parametric methods take samples from the style image and place the samples based on pixel intensity [10, 12, 13] or deep features [26] of the target image to synthesize a new image. Parametric methods represent the style as statistical features, and adjust the

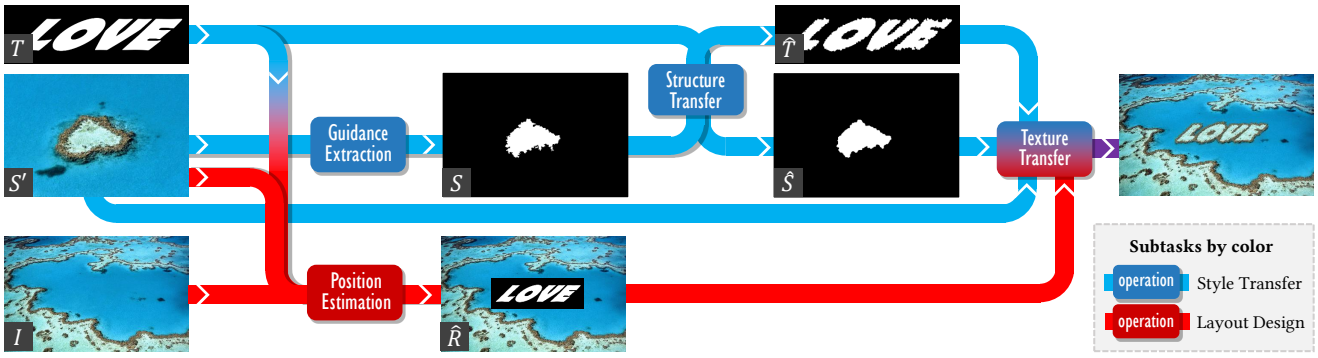


Figure 2: Framework of the proposed method. ***Image credits: Unsplash user Yanguang Lan*** [1].

target image to satisfy these features. Recent deep learning based parametric methods [8, 15, 22] 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 images. In fact, for non-parametric methods, it is hard to use pixel intensity or deep features to establish a direct mapping between binary text and a style image, due to their great modality discrepancy. On the other hand, text images lack high-level semantic information, which limits the performance of the parametric methods.

As the most related method to our problem, a text effects transfer algorithm [37] 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. However, this method is based on supervised learning [19] and requires a corresponding non-stylized image in addition to the style image to learn the transformation between them, as shown in Figure 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 text stylization problem, only with a binary text image and an arbitrary style image as in Figure 1(a). More specifically, our problem consists of two tasks as shown in Figure 2. In the first task of text stylization, we aim to bridge the distinct visual discrepancies between the text image and the style image. We first extract the main structural imagery of the style image to build a preliminary mapping to the text. The mapping is then refined by a legibility-preserving structure transfer algorithm, which carefully adds shape characteristics of the source style to the text while maintaining the main structure of the text stroke. Based on the mapping, a novel texture transfer guided by saliency constraints for text legibility is proposed. These improvements allow our unsupervised method to yield satisfying results without the ideal input required by supervised methods.

Furthermore, in the second task we investigate the combination of stylized text and background images, which is very common in visual design. Specifically, we propose a new context-aware visual-textual presentation 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. For example, decorating a blue sky with cloud-like typography as in Figure 2. To achieve it, we leverage cues for both seamlessness and aesthetics to determine the image layout, where the target text is finally synthesized into the background image. When various styles are available, our method can generate diverse artistic typography against the background image, thereby facilitating a much wider variety of aesthetic interest expression.

In summary, the contributions of this work are threefold:

- We raise a new unsupervised text stylization problem for visual design and develop the first automatic aesthetic driven framework to address it. Extensive experiments show that our method creates more appealing stylized text.
- We present novel structure and texture transfer algorithms to balance text legibility with texture consistency, which we show to be effective in style transition between the binary text and the style image.
- We propose a context-aware layout design method to determine the image layout and seamlessly synthesize the artistic text into the background image, which creates professional looking visual-textual presentations.

2 RELATED WORK

2.1 Texture Synthesis

Texture synthesis technologies attempt to generate new textures from a given texture example. Non-parametric methods use pixel [11] or patch [10] 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 [25], dynamic programming [10], graph cut [21] and coherence function optimization [34] is proposed. Meanwhile, parametric methods build mathematic models to simulate certain texture statistics of the texture example. The recent most popular model is the Gram-matrix model proposed by Gatys *et al.* [14]. Using the correlations between multi-level deep features to represent textures, this model produces natural textures of noticeably high perceptual quality.

[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.

2.2 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 [19], rely on the availability of an input image and its stylized result. 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 [19], image analogies have been extended in various ways such as video analogies [4] and fast image analogies [3]. The main drawback of image analogies is the strict requirement for the registered example pair. More often, we only have a style image at hand, and need to turn to the unsupervised methods.

Without the guidance of the example pair, unsupervised methods directly find mappings between different texture modalities. For instance, Efros and Freeman [10] introduced a guidance map derived from image intensity to help find correspondences between two texture modalities. Zhang *et al.* [40] used a sparse-based initial sketch estimation [39] to construct a mapping between the source sketch texture and the target image. Frigo *et al.* [13] put forward a patch partition mechanism for an adaptive patch mapping, which balances the preservation of structures and textures.

Driven 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 [15], the authors adapted Gram-matrix-based texture synthesis [14] 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 [7], perceptual factor control [16], structure preservation [23] and acceleration [20]. In parallel, Li and Wand [22] introduced a framework called CNNMRF that expoits Markov Random Field (MRF) to enforce local texture transfer. Based on CNNMRF, Neural Doodle [6] incorporates semantic maps for analogy guidance, which turns semantic maps into artwork.

2.3 Text Stylization

In the domain of text image editing, several tasks have been addressed like calligrams [28, 36, 41] and handwriting generation [17, 24]. Zou *et al.* [41] arranged and deformed letters to fit a word into a 2D shape. Handwriting style transfer is accomplished using letter samplings from annotated handwriting documents [17] or neural networks to learn stroke styles [24]. 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 Figure 1), and soft clouds (see Figure 2).

To the best of our knowledge, the work of [37] is the only prior attempt at generating text effects. It solves the text stylization problem in a supervised manner: 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 based on arbitrary source style images, without the input requirements as in [37]. Our method provides a more flexible and effective tool to create unique visual design artworks.

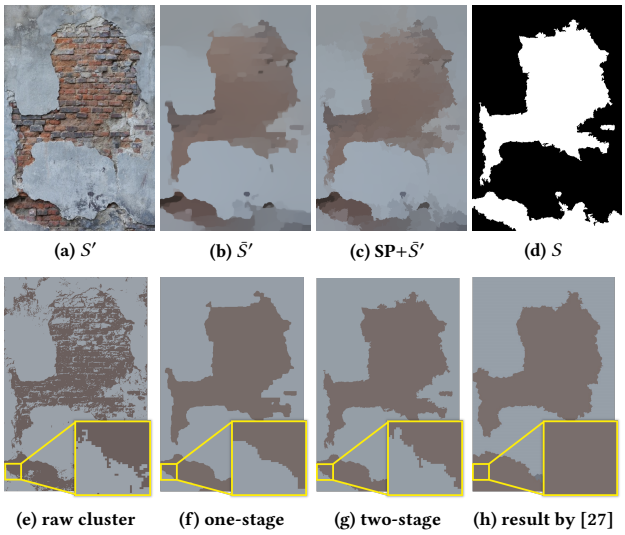


Figure 3: Guidance map extraction. (a) Input S'. (b) Structure image $\bar{S}'$. (c) Super pixels colored by their mean pixel values in (b). (d) Extracted guidance map S. (e)-(g) K-means clustering results of (a)-(c), respectively. (h) Result by [27]. Cropped regions are zoomed for better comparison.

3 TEXT STYLE TRANSFER

In this section, we describe our text style transfer method to migrate the style from a source image S' to a text image T. As shown in Figure 2, we abstract a binary imagery S from the source style image (Section 3.1), adjust its contour and the outline of the target text to narrow the structural difference between them (Section 3.2). The adjusting results $\hat{S}$ and $\hat{T}$ establish an effective mapping between the target text image and the source style image. Then we are able to synthesize textures for the target text (Section 3.3).

3.1 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 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 text interior (foreground) and black pixels for the text 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 that well preserve the morphological characteristics of source textures.

In particular, we use the Relative Total Variation (RTV) [35] to remove the color variance inside the texture, and obtain a structure image $\bar{S}'$. However, the details on the texture contour are also smoothed out in $\bar{S}'$ (see Figure 3(b)(f)). Hence, we put forward a two-stage abstraction method. In the first stage, pixels in S' are abstracted as fine-grained super pixels [1] 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,

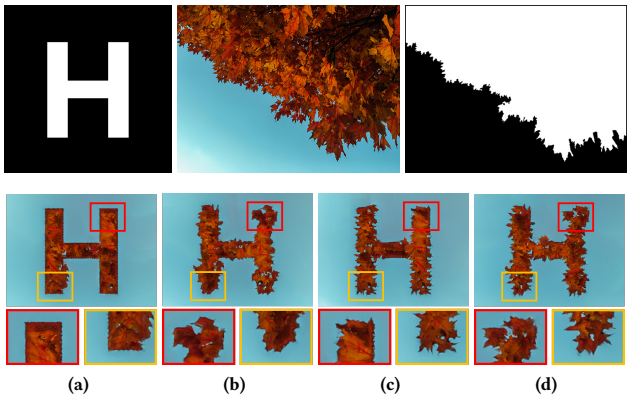


Figure 4: Effect of bidirectional structure transfer. Top row: from left to right, T, S' and S. Bottom row: the text stylization results using (a) original $T + S$, (b) forward transfer $\hat{T} + S$, (c) backward transfer $T + \hat{S}$, and (d) bidirectional transfer $\hat{T} + \hat{S}$. *Image credits: Unsplash users Aaron Burden.*

the super pixels are further abstracted as coarse-grained foreground and background via K-means clustering ($K = 2$). Figure 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 [27] (see the zoomed region in Figure 3(h)).

Finally, pixel saliency in S' is detected [38] and the cluster with higher mean pixel saliency is set as the foreground.

3.2 Structure Transfer

Directly using S extracted in Section 3.1 and T for style transfer results in unnatural texture boundaries as shown in Figure 4(a). A potential solution is to employ shape synthesis technique [29] to minimize structural inconsistencies between S and T. In Layered Shape Synthesis (LSS) [29], shapes are represented as a collection of boundary patches at multiple resolutions, and the style of a shape is transferred onto another by optimizing a bidirectional similarity function. However, in our task, such an approach does not consider the legibility, and the text will become illegible after adjustment. As shown in the bottom row of Figure 6, 'H' is deformed like 'X' and 'M'. 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 Figure 5. 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

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

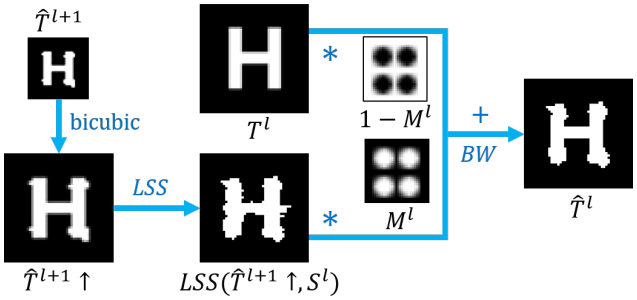


Figure 5: Preserving the stroke trunks by weighted combination of the trunk region from T^l and the stroke end region from the shape synthesis result.

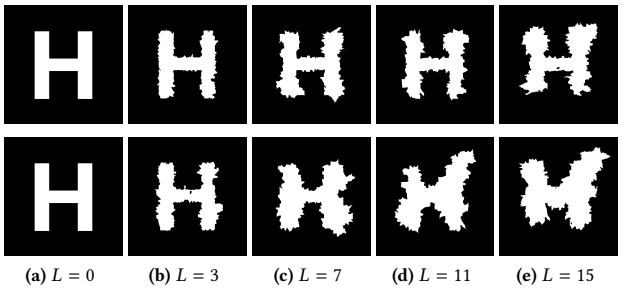
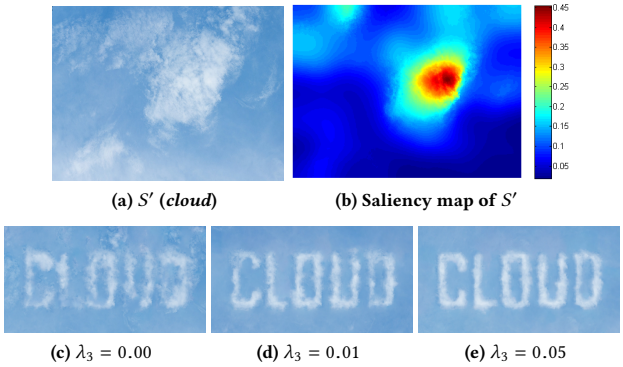


Figure 6: Effect of stroke trunk protection mechanism and the number of image pyramid layers L. Top row: our legibility-preserving structure transfer result. Bottom row: structure transfer result without stroke trunk protection. In this example, we use S in Figure 4 as the reference.

where $*$ is the element-wise multiplication operator and $\uparrow$ is the bicubic upsampling operator. $LSS(T, S)$ is the shape synthesis result of T obtained using LSS with S as the shape reference, and $BW(\cdot)$ is the binarization operation with threshold 0.5. The pipeline of the proposed structure transfer is presented in Figure 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 Figure 6 that the deformation degree increases as L increases, and our stroke trunk protection mechanism effectively balances structural consistency with text legibility even under very large L.

In addition, we propose a bidirectional structure transfer 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. As shown in Figure 4(a)-(d), 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 results $\hat{T}^0$ and $\hat{S}^0$ will be used as guidance for texture transfer. For simplicity, we will omit the superscript in the following.



(a) S' (*cloud*) (b) Saliency map of S'

(c) $\lambda_3 = 0.00$ (d) $\lambda_3 = 0.01$ (e) $\lambda_3 = 0.05$

Figure 7: Saliency term enhances text legibility. *Image credits: Unsplash users Ashim D'Silva.*

3.3 Texture Transfer

We follow the optimization-based texture synthesis method [37] to transfer textures from S' to $\hat{T}$ with $\hat{S}$ as guidance. In order to make the text stand out in the textures, we augment the texture synthesis objective function in [37] with a new saliency term as follows,

$$\min_{q} \sum_{p} E_a(p,q) + \lambda_1 E_d(p,q) + \lambda_2 E_p(p,q) + \lambda_3 E_s(p,q), \quad (2)$$

where p is the center position of a target patch in $\hat{T}$ and T', and 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 [37] for details of the first three terms. Our novel saliency term is defined as

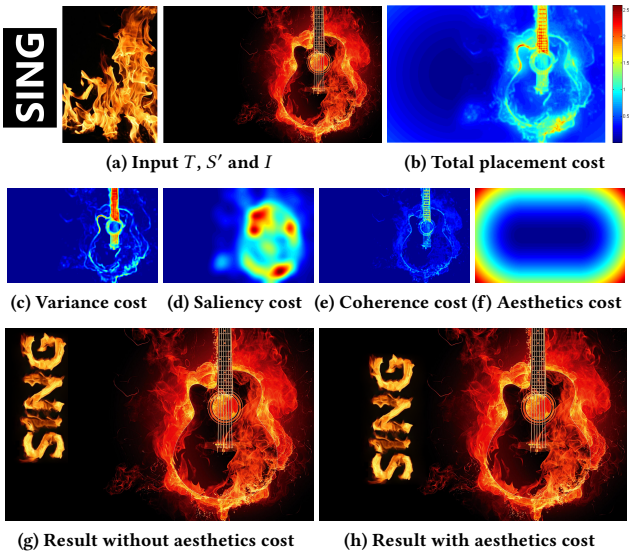
$$E_s(p,q) = \begin{cases} W(p) \cdot Sal(q), & \text{if } \hat{T}(p) = 1, \\ W(p) \cdot (1 - Sal(q)), & \text{if } \hat{T}(p) = 0, \end{cases} \quad (3)$$

where $Sal(q)$ is the saliency at pixel q in S' detected using [38]. $W(p) = 1 - \exp(-dist(p)^2/2\sigma_1^2)/2\pi\sigma_1^2$ is a gaussian-based weight with $dist(p)$ the distance of p to the text boundary. The saliency term encourages pixels inside the text to find salient textures for synthesis and keeps the background less salient. We show in Figure 7 that a higher weight of our saliency term makes the stylized text more prominent, thereby enhancing text legibility.

Similar to [37], we take the iterative coarse-to-fine matching and voting steps as in [34] to solve Eq. (2). In the matching step, PatchMatch algorithm [2] is adopted to accelerate the algorithm.

4 CONTEXT-AWARE LAYOUT DESIGN

In this section, we describe our context-aware layout design method to synthesize T into a background image I with the style of a source image S'. As shown in Figure 2, we formulate an optimization function to estimate the optimal position $\hat{R}$ for T to embed (Section 4.1). Once the layout is determined, the background information around T will be collected. Constrained by this contextual information, the target text is seamlessly synthesized into the background image in an image inpainting manner (Section 4.2).



(a) Input T, S' and I (b) Total placement cost

(c) Variance cost (d) Saliency cost (e) Coherence cost (f) Aesthetics cost

(g) Result without aesthetics cost (h) Result with aesthetics cost

Figure 8: Image layout is determined by jointly considering a variance cost, a saliency cost, a coherence cost and an aesthetics cost. *Image credits: Unsplash users Dark Rider.*

4.1 Position Estimation

In order to synthesize the target text seamlessly into the background image, the image layout should be properly determined. We formulate an optimization function for context-aware position estimation by considering the cost of each pixel x of I in four aspects,

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

where R is a rectangular area of the same size as T, indicating the embedding position. The position is estimated by searching an area $\hat{R}$ 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 weighted by $\lambda_4 = 0.5$ for subjective evaluation.

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 terms preclude our method from overlaying important objects in the background image with the target text.

We further use U_c to measure the texture consistency between I and S'. Specifically, $U_c(x)$ is defined as the L_2 distance between the patch centered at x in I and its best matched patch in S'.

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

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

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. Figure 8 visualizes these four costs, which jointly determine the ideal image layout.

Figure 9: 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 (the upper row) and unsupervised (the lower row) methods. For supervised methods, the structure guidance map extracted by our method is directly given as the input. *Image credits: Unsplash users Aaron Burden and Ishan@seefromthesky.*

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

4.2 Text Embedding

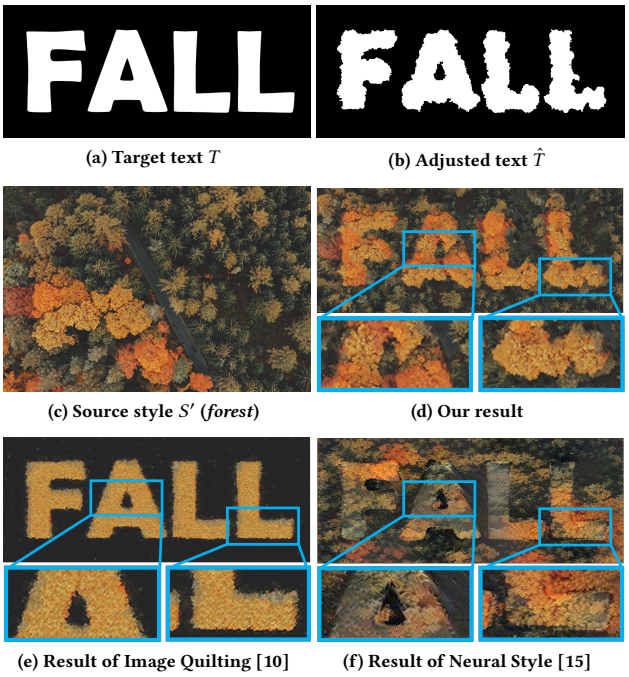
Once the layout is determined, we synthesize the target text into the background image in an image inpainting manner. Image inpainting technologies [5, 9, 33] have long been investigated in image processing literature to fill the unknown parts of an image. Similarly our problem sets $\hat{R}$ as the 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 3.3, 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.

5 EXPERIMENTAL RESULTS

5.1 Comparison of Style Transfer Methods

In Figure 9, we present a comparison of our method with six state-of-the-art supervised and unsupervised style transfer methods on text stylization. For supervised methods, the guidance map S extracted by our method in Section 3.1 is directly given as the input.

Structural consistency. In comparison to these approaches, our method can preserve critical structural characteristics of the textures in source style images. Other methods do not consider to adapt the text 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 Figure 9. By comparison, Neural Style [15] and CNNMRF [22] 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 *coral reef* groups. The structural consistency achieved by our method can be better observed in the zoomed regions in Figure 10, where even Neural Style [15] cannot transfer structures effectively.

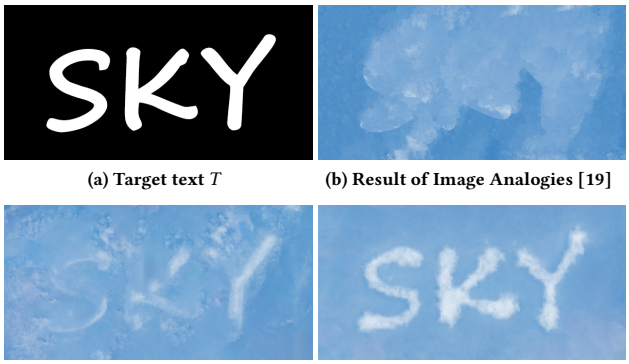


(a) Target text T (b) Adjusted text $\hat{T}$

(c) Source style S' (*forest*) (d) Our result

(e) Result of Image Quilting [10] (f) Result of Neural Style [15]

Figure 10: 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) Result of Image Analogies [19]

(c) Result of Text Effects Transfer [37] (d) Our result

Figure 11: 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' (*cloud*) in Figure 7 is used.

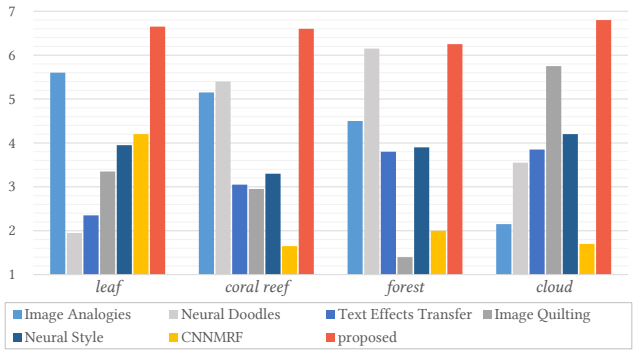


Figure 12: Average evaluation scores of different methods for four test cases in Figures 9-11 from our 20-person study.

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 [19] is filled by many salient repetitive textures, which confuses itself with the foreground. Text Effects Transfer [37] 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 [37] to place textures compactly inside the text, leading to textureless artifacts in *leaf* results. Our method further proposes a saliency constraint for complement, resulting in the creation of the artistic text that highlights from clean backgrounds. We show in Figure 11 that when the foreground and background colors are not contrasting enough, our approach demonstrates greater superiority.

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 Figure 9, which highly respects the contexts of S'. It is achieved by our context-aware style transfer to ensure structure, color, texture and semantic consistency. By contrast, Image Quilting [10] relies on patch matching between two completely different modalities S' and T, thus its results are just as flat as the raw text. Three deep-based methods, Neural Style [15], CNNMRF [22] and its supervised version Neural Doodles [6], transfer suitable textures onto the text. However, their main drawbacks are the color deviation and checkerboard artifacts generated by deep networks.

User study. For quantitative evaluation, we conducted user studies where twenty participants were shown four test cases in Figures 9-11 and were asked to assign 1 to 7 scores to the seven methods in each case (a higher score indicates that the stylized result is more consistent in style with the style image). Figure 12 shows that our method outperforms other methods in all cases, obtaining the best average score of 6.58, significantly higher than 4.35, 4.26, 3.26, 3.36, 3.84 and 2.39 of Image Analogies [19], Neural Doodles [6], Text Effects Transfer [37], Image Quilting [10], Neural Style [15] and CNNMRF [22], respectively.

5.2 Stylish Text Generation in Different Fonts and Languages

We experiment on text in various fonts and languages to test the robustness of our method. Some results are shown in Figures 13-14. In Figure 13, characters are quite varied in different languages. Our method successfully synthesizes dancing flames onto a variety of

Figure 13: Visual effects for text stylization on different languages.

Figure 14: Visual effects for text stylization on different fonts. ***Image credits: Unsplash users Yanguang Lan.***

barrier reef *cloud* *flame*

Figure 15: 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. ***Image credits: Unsplash users Yanguang Lan, JJ Ying, Tim Gouw and Thomas Kelley.***

languages, while maintaining their local fine details, such as the small circles in Thai. In Figure 14, the rigid outlines of the text are adjusted to the shape of a coral reef, but 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.

5.3 Visual-Textual Presentation Synthesis

We aim to synthesize professional looking visual-textual presentation that combines beautiful images and overlaid stylish text. In Figure 15, 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 by color transfer algorithm [18], 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, thus providing a flexible and effective tool to create original and unique visual-textual presentations.

6 CONCLUSION AND FUTURE WORK

In this paper, we raise a new unsupervised text stylization problem to incorporate binary text and colorful images, and develop a novel automatic aesthetic driven framework to solve it. We exploit structure and texture transfers to balance text legibility with texture consistency. Cues for contextual seamlessness and aesthetics are leveraged to determine the image layout. Our context-aware text stylization approach breaks through a barrier between images and text, allowing users to create fine artistic text and professional looking visual-textual presentations. In future work, we would like to explore style image recommendation based on the background image, which will contribute more to seamless embedding.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under contract No. 61772043 and CCF-Tencent Open Research Fund.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 11 (2012), 2274–2282.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. 2009. PatchMatch: a Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics* 28, 3 (August 2009), 341–352.

[3] Connelly Barnes, Fang Lue Zhang, Liming Lou, Xian Wu, and Shi Min Hu. 2015. PatchTable: efficient patch queries for large datasets and applications. *ACM Transactions on Graphics* 34, 4 (2015), 97.

[4] Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. 2013. Stylizing animation by example. *ACM Transactions on Graphics* 32, 4 (2013), 119.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. *ACM Transactions on Graphics* (2000), 417–424.

[6] Alex J. Champandard. 2016. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. (2016). arXiv:1603.01768.

[7] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *Proc. Int'l Conf. Computer Vision.*

[8] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2017. StyleBank: An Explicit Representation for Neural Image Style Transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

[9] A. Criminisi, P. Pérez, and K. Toyama. 2004. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions on Image Processing* 13 (September 2004), 1200 – 1212.

[10] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proc. ACM Conf. Computer Graphics and Interactive Techniques.* 341–346.

[11] Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *Proc. IEEE Int'l Conf. Computer Vision.*

[12] Michael Elad and Peyman Milanfar. 2017. Style Transfer via Texture Synthesis. *IEEE Transactions on Image Processing* 26, 5 (2017), 2338–2351.

[13] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. 2016. Split and Match: example-based Adaptive Patch Sampling for Unsupervised Style Transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

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

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

[16] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

[17] Tom S. F. Haines, Oisin Mac Aodha, and Gabriel J. Brostow. 2016. My Text in Your Handwriting. *ACM Transactions on Graphics* 35, 3 (May 2016), 26:1–26:18.

[18] Aaron Hertzmann. 2001. *Algorithms for rendering in artistic styles.* Ph.D. Dissertation. New York University.

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

[20] Justin Johnson, Alexandre Alahi, and Fei Fei Li. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proc. European Conf. Computer Vision.*

[21] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics* 22, 3 (2003), 277–286.

[22] Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

[23] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat Seng Chua. 2017. Laplacian-Steered Neural Style Transfer. In *Proc. ACM Int'l Conf. Multimedia.*

[24] Zhouhui Lian, Bo Zhao, and Jianguo Xiao. 2016. Automatic Generation of Large-scale Handwriting Fonts via Style Learning. In *SIGGRAPH ASIA 2016 Technical Briefs.* ACM, Article 12, 12:1–12:4 pages.

[25] Lin Liang, Ce Liu, Yingqing Xu, Baining Guo, and Heungyeung Shum. 2001. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics* 20, 3 (2001), 127–150.

[26] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Singbing Kang. 2017. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics* 36, 4 (2017), 120.

[27] Yitzchak David Lockerman, Basile Sauvage, Rémi Allègre, Jean-Michel Dischler, Julie Dorsey, and Holly Rushmeier. 2016. Multi-scale label-map extraction for texture synthesis. *ACM Transactions on Graphics* 35, 4 (2016), 140.

[28] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr. 2011. Digital Micrography. *ACM Transactions on Graphics* (2011), 100:1–100:12.

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

[30] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics* 35, 4 (2016), 1–18.

[31] Yichang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. 2014. Style transfer for headshot portraits. *ACM Transactions on Graphics* 33, 4 (2014), 1–14.

[32] Yichang Shih, Sylvain Paris, Frédo Durand, and William T. Freeman. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics* 32, 6 (2013), 2504–2507.

[33] O. Le Meur, M. Ebdelli, and C. Guillemot. 2013. Hierarchical Super-Resolution-Based Inpainting. *IEEE Transactions on Image Processing* 22 (October 2013), 3779 – 3790.

[34] Y. Wexler, E. Shechtman, and M. Irani. 2007. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (March 2007), 463–476.

[35] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. 2012. Structure extraction from texture via relative total variation. *ACM Transactions on Graphics* 31, 6 (2012), 139.

[36] Xuemiao Xu, Linling Zhang, and Tien-Tsin Wong. 2010. Structure-based ASCII Art. *ACM Transactions on Graphics* 29, 4 (July 2010), 52:1–52:9.

[37] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. 2017. Awesome Typography: Statistics-Based Text Effects Transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.*

[38] Jianming Zhang and Stan Sclaroff. 2013. Saliency detection: A boolean map approach. In *Proc. Int'l Conf. Computer Vision.* 153–160.

[39] Shengchuan Zhang, Xinbo Gao, Nannan Wang, and Jie Li. 2015. Face Sketch Synthesis via Sparse Representation-Based Greedy Search. *IEEE Transactions on Image Processing* 24, 8 (2015), 2466–2477.

[40] S Zhang, X Gao, N Wang, and J Li. 2016. Robust Face Sketch Style Synthesis. *IEEE Transactions on Image Processing* 25, 1 (2016), 220–232.

[41] Changqing Zou, Junjie Cao, Warunika Ranaweera, Ibraheem Alhashim, Ping Tan, Alla Sheffer, and Hao Zhang. 2016. Legible Compact Calligrams. *ACM Transactions on Graphics* 35, 4 (July 2016), 122:1–122:12.