Computer Vision and Image Understanding 174 (2018) 43–56

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

# Text effects transfer via distribution-aware texture synthesis

Shuai Yang, Jiaying Liu[1,*], Zhouhui Lian, Zongming Guo

*Institute of Computer Science and Technology, Peking University, Beijing 100080, China*

ARTICLE INFO

*Keywords:*
Text effects
Texture synthesis
Spatial distribution
Multi-scale
Style transfer

ABSTRACT

In this paper, we explore the problem of fantastic special-effects synthesis for the typography. The main challenge of this problem lies in the model diversities to illustrate varied text effects for different characters. To address this issue, we exploit the key analytics on the high regularity of the texture spatial distribution for text effects to guide the synthesis process. Specifically, we characterize the stylized patches by their normalized positions relative to the text skeleton and the optimal scales to depict their style elements. Our method first estimates these two features and derives their correlation statistically. They are then converted into soft constraints for texture transfer to accomplish adaptive multi-scale texture synthesis and to make style element distribution uniform. It allows our algorithm to produce artistic typography that well consists with both local texture patterns and the global spatial distribution in the source example. Furthermore, stroke similarities are considered to control the varieties of text effects among multiple characters in a word. Experimental results demonstrate the superiority of our distribution-aware method for various text effects over conventional style transfer methods. In addition, we validate the effectiveness of our algorithm with extensive artistic typography library generation and apply our method to a general application of special effects transfer for stroke-based graphics.

## 1. Introduction

Text stylization is the technology to design the special text effects to render the character into an original and unique artwork. These amazing text styles include basic effects such as *shadows, outlines, colors* and sophisticated effects such as burning *flames*, multi-layered *denims*, multi-colored *neons*, as shown in Fig. 1. Texts decorated by well-designed special effects become much more attractive. It can also better reflect the thoughts and emotions from the designer. The beauty and elegance of text effects are well appreciated, making it widely used in the publishing and advertisement. However, creating vivid text effects requires a series of subtle processes by an experienced designer using editing softwares: determine color styles, warp textures to match text shapes and adjust the transparency for visual plausibleness, *etc*. These advanced editing skills are far beyond the abilities of most casual users. This practical requirement motivates our work: We investigate an approach to automatically transfer various highly stylized text effects onto raw plain texts, as shown in Fig. 1.

Text effects transfer is a brand new sub-topic of style transfer. Style transfer can be related to color transfer and texture transfer. Color transfer matches global (Reinhard et al., 2001) or local (Tai et al., 2005) color distributions of the target and source images. Texture transfer relies on texture synthesis technologies, where the texture generation is constrained by guidance images. Texture synthesis can be divided into two categories: non-parametric methods (Efros and Freeman, 2001; Efros and Leung, 1999; Kwatra et al., 2003) and parametric methods (Julesz and Bergen, 1983; Versteegen et al., 2016). The former generates new textures by resampling pixels or patches from the original texture, while the latter models textures using statistical measurements and produces a new texture that shares the same parametric results with the original one.

From a technical perspective, it is quite challenging and impractical to directly exploit the traditional style transfer methods to generate new text effects. The challenges lie in:

- The extreme diversity of the text effects and character shapes: The style diversity makes the transfer task difficult to model uniformly. Further, the algorithm should be robust to the tremendous variety of characters.
- The complicated composition of style elements: Text effects often contain multiple intertwined style elements (we call them *text sub-effects*) that have very different textures and structures (see *denim*

* Corresponding author.
*E-mail address:* liujiaying@pku.edu.cn (J. Liu).
[1] This work was supported by National Natural Science Foundation of China under contract No. 61772043 and CCF-Tencent Open Research Fund.

https://doi.org/10.1016/j.cviu.2018.07.004
Received 6 October 2017; Received in revised form 11 April 2018; Accepted 27 July 2018
Available online 30 July 2018

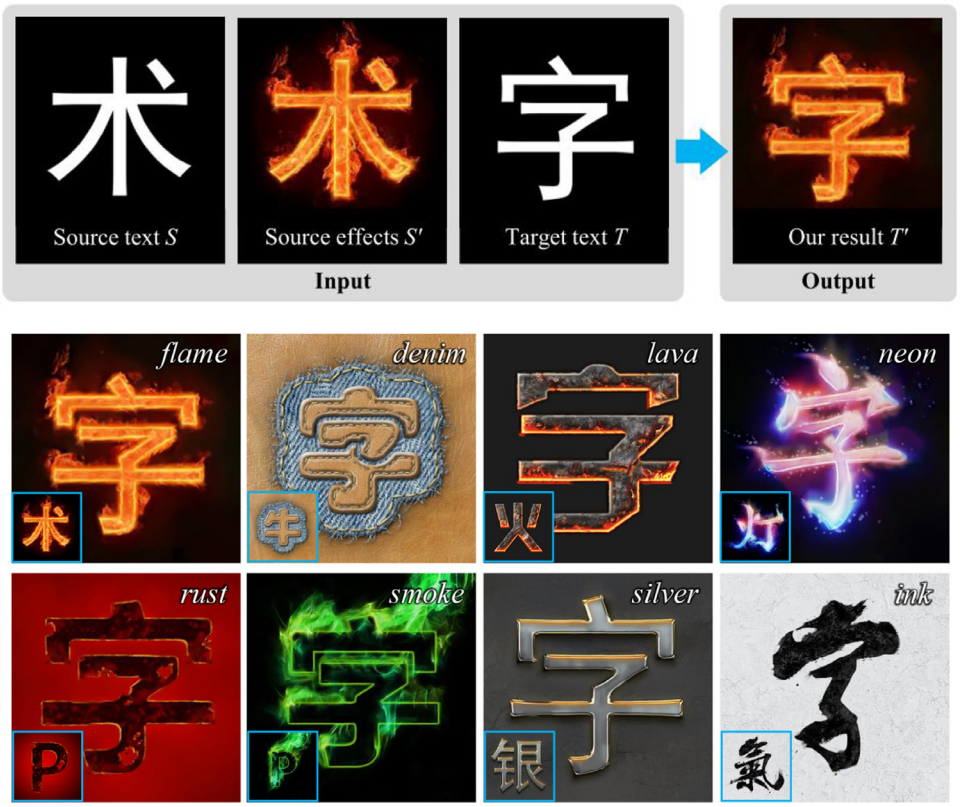



**Fig. 1.** Overview. The top row: Our method takes as input the source text image $S$, its counterpart stylized image $S'$ and the target text image $T$, then automatically generates the target stylized image $T'$ with the special effects as in $S'$. The bottom two rows: Our stylization results $T'$ with their reference style $S'$ in the low-left corner. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

example in Fig. 1, which has two sub-effects: brown cowhide and blue denim fabric) and need specialized treatments.

- The simpleness of guidance images: The raw plain text as guidance gives few hints on how to place different sub-effects. Textures in the white text and black background regions may not hold the stationarity. It makes the traditional non-parametric texture-by-numbers method (Hertzmann et al., 2001) fail, which has assumed textures to be stationary in each region of the guidance map. Meanwhile, the plain text image provides little semantic information. This makes the recent successful parametric deep-based style transfer methods (Gatys et al., 2016; Li and Wand, 2016a) lose their advantages of representing high-level semantic information.

For these reasons, conventional style transfer methods for general styles perform poorly on text effects.

In this paper, we propose a novel text effects transfer algorithm to address these challenges. The key idea is to analyze and model the spatial distribution-based essential characteristics of high-quality text effects and to leverage them to guide the synthesis process. The characteristics are summarized based on the analytics over dozens of well-designed text effects into a general prior. This prior guides our style transfer process to synthesize different sub-effects adaptively and to simulate their spatial distribution as in the source example. We further consider the psycho-visual factor to enhance image naturalness. All measurements are carefully designed to achieve a certain robustness to character shapes.

Compared with our previous work (Yang et al., 2017), we expand the text effects synthesis on a single character to words by considering the relationships of text effects among multiple characters. We introduce a new stroke term to regulate the synthesis to be more consistent or diverse for multiple characters. In addition, we refine our experiments with augmented test images for visual comparisons and a user study for quantitative comparisons. The quantitative comparisons over a wider variety of text effects verify that the proposed method has obvious advantages for high-quality exquisite text effects transfer. We further provide a running time comparison, which validates the efficiency of the proposed method. Finally, we extend our method to special effects transfer between general stroke-based graphics. In summary, our contributions are threefold:

- We raise a brand new topic of text effects transfer that turns plain texts into highly stylized artworks, which enjoys wide application scenarios such as picture creation on social networks and commercial graphic design.
- We perform analysis on well-designed text effects and summarize their key spatial distribution-based characteristics. We model these characteristics mathematically to form a general prior that can be used to significantly improve the style transfer process for texts.
- We propose the first method to generate compelling text effects, which share both similar local texture patterns and the global spatial distribution with the source example, while preserving image naturalness. Our method also provides a flexible mechanism to control the texture consistency for multi-character text effects synthesis.

The rest of this paper is organized as follows. In Section 2, we review related works in color transfer and texture transfer. Section 3 defines the text effects transfer problem and analyze the spatial distribution-based characteristics for text effects. In Section 4, the details of the proposed distribution-aware algorithm is presented. We validate our method by comparing it with state-of-the-art style transfer algorithms and generating extensive artistic typography library in Section 5. Finally, we conclude our work in Section 6.

## 2. Related work

### 2.1. Color transfer

Pioneering color transfer methods (Pitié et al., 2007; Reinhard et al., 2001) transfer color between images by matching their global color distributions. Subsequently, local color transfer is achieved based on

segmentation (Tai et al., 2005; 2007), perceptual color categories (Chang et al., 2007; 2005) or user interaction (Welsh et al., 2002). Color transfer is further improved using fine-grained patch or pixel correspondences. Shih et al. (2013) considered the problem of hallucinating daytime of an image by learning local color affine transforms in a time-lapse database based on patch matching. The authors latter proposed a color transfer method for headshot portraits (Shih et al., 2014) through pixel-level luminance and contrast statistics. Song et al. (2017) further investigated the combination of multiple headshot color references. In Park et al. (2016), sparse pixel correspondences are used to estimate white balance and gamma correction parameters, which successfully unify the color style of photo collections. A graph regularization for color processing is proposed in Lezoray et al. (2007). Recently, color transfer (Yan et al., 2016) and colorization (Larsson et al., 2016; Zhang et al., 2016) using deep neural networks have drawn people's attentions.

### 2.2. Non-parametric texture synthesis and transfer

Pioneering non-parametric approaches create new textures one pixel a time in an inside-out (Efros and Leung, 1999) or scanline (Wei and Levoy, 2000) order. The subsequent works improve pixel-by-pixel approaches in quality and speed by synthesizing patches rather than pixels. To handle the overlapped regions of neighboring patches for seamlessness, Liang et al. (2001) proposed to blend patches, and Efros and Freeman (2001) used dynamic programming to find an optimal separatrix in overlapped regions, which is further improved via graph cut (Kwatra et al., 2003). Unlike previous methods that synthesize textures in a local manner, recent techniques synthesize globally using objective functions. In Kwatra et al. (2005), the authors determine pixel values by optimizing a global quadratic energy function, which minimizes the mismatches of input/output neighborhoods, leading to better output quality. Base on Kwatra et al. (2005), Elad and Milanfar (2016) proposed a style transfer approach, which emphasizes keeping the content intact in selected regions, while producing hallucinated and rich style in others. Besides, a coherence-based function (Wexler et al., 2007) is proposed to synthesize textures in an iterative coarse-to-fine fashion. This method performs patch matching and voting operations alternately and achieves good local structures. It is then accelerated using PatchMatch algorithm (Barnes et al., 2009) and is extended to adapt to non-stationary textures through patch geometric and photometric transformations (Barnes et al., 2010; Darabi et al., 2012).

Texture transfer generates textures but also preserves the structure of the target image. It can be computed in a supervised or unsupervised fashion. Given a pair of guidance maps, the supervised methods, also known as Image Analogies (Cheng et al., 2008; Hertzmann et al., 2001; Okura et al., 2015), preserve the structure by reducing the mismatches between the source and target guidance maps. By investigating temporal coherence, the subsequent work of Bénard et al. (2013) extents image analogies to video analogies. Meanwhile, the recent work of Barnes et al. (2015) effectively accelerates the analogy process using a lookup table. In Lukáč et al. (2013), texture boundaries are synthesized in priority to further constrain the structure.

The unsupervised methods deal with the scenario where guidance map pairs are not available. Therefore, finding good mappings between different texture modalities is the crux. Frigo et al. (2016) proposed an adaptive patch partition to precisely capture source textures and preserve target structures, followed by a Markov Random Field (MRF) objective function for global texture synthesis.

### 2.3. Parametric texture synthesis and transfer

The idea of modeling textures using statistical measurements has led to the development of textons (Julesz and Bergen, 1983; Xu et al., 2012). Nowadays, deep-based texture synthesis starts trending due to the great descriptive ability of deep neural networks. Gatys *et al.* proposed to use Gram-matrix in the Convolutional Neural Networks (CNNs) feature space to represent textures (Gatys et al., 2015) and adapted it to style transfer by incorporating content similarities (Gatys et al., 2016). This work presented the remarkable generic painting transfer technique and attracted many follow-ups in loss function improvement (Lin and Maji, 2016; Selim et al., 2016) and algorithm acceleration (Johnson et al., 2016; Ulyanov et al., 2016). Recently, methods that replace the Gram-matrix by MRF regularizer is proposed for photographic synthesis (Li and Wand, 2016a) and semantic texture transfer (Champandard, 2016). Meanwhile, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) provide another idea for texture generation by using discriminator and generator networks, which iteratively improve the model by playing a minimax game. Its extension, the conditional GANs (Mirza and Osindero, 2014), fulfills the challenging task of generating images from abstract semantic labels. Li and Wand (2016b) further showed that their Markovian GANs has certain advantages over the Gram-matrix-based methods (Gatys et al., 2016; Ulyanov et al., 2016) in coherent texture preservation.

## 3. Problem formulation and analysis

In this section, we first formulate our text effects transfer problem. Visual analytic is then presented on our observation of the high correlation between patch patterns (*i.e. color* and *scale*) and their spatial distributions relative to the text skeleton in text effects images.

Text effects transfer takes as input a set of three images, the source raw text image $S$, the source stylized image $S'$ and the target raw text image $T$, then automatically produces the target stylized image $T'$ with the text effects such that $S: S' :: T: T'$ (Hertzmann et al., 2001).

It is a quite challenging task to transfer arbitrary text effects automatically, due to the variety of text effects, the complex composition of text sub-effects and the simpleness of guidance maps. To address this problem, we investigate the preferable text effects on the following two aspects: (i) how to determine the essential characteristics of text effects and (ii) how to characterize them mathematically. We start with a basic observation on text effects that the patch patterns are highly dominated by their locations. We develop to represent the pattern of a patch by two optimal factors: the patch *color* and the patch *scale*. As shown intuitively in Figs. 2(a)–(d), the patches at similar locations (marked with the same color) tend to have similar patterns.

To quantitatively evaluate the locations of patches, we divide a text effects image into $N = 16$ classes, namely, $N$ partitions. The modes of partition are extremely diverse and thus it is impractical to compare all of them. In this work, we compare five typical partition modes:

- Random: pixels are randomly divided into $N$ equal parts;
- Grid: all partitions are evenly distributed according to their horizontal and vertical coordinates on the image;
- Angle: all partitions are evenly distributed according to their angular coordinate, where the center of polar coordinate system is at the geometric center of the image;
- Ring: all partitions are evenly distributed according to their radial coordinate, where the center of polar coordinate system is at the geometric center of the image;
- Distance: all partitions are evenly distributed according to their geometric distance (the distance calculation will be introduced in Section 4.1.2) to the skeleton of the text.

In Fig. 2(e), the partitions modes of grid, angle, ring and distance have been intuitively illustrated, where all partitions are tinged differently.

Then for each partition mode, we investigate the relationship between these partitions and the distributions of corresponding patterns. For the factor of *color*, we represent its reliability by its classification accuracy of partitions:


(a) *flame* (b) Pattern distribution (c) *denim* (d) Pattern distribution (e) Partition modes

**Fig. 2.** Correlation between patch patterns and distances. (a)(c) *flame* and *denim* text effects. (b)(d) Textures with similar distances to the text skeleton (in white) tend to have similar patterns. (e) Pixels are divided into $N = 16$ classes using different partition modes.

$$r_{\text{color}} = 1 - \epsilon, \tag{1}$$

where $\epsilon$ is the training error or empirical risk obtained by training SVM (Chang and Lin, 2011) to classify the color given a type of partition. To simplify the analysis, we use the center pixel color to represent the whole patch color in this section. We have tested on 30 text effects images created by designers to obtain their reliability on color classification. The average reliability is then shown in Table 1, where only the relative values are instructive in our design. From this table, the distance is demonstrated to be the most reliable factor to depict pixel colors, with a value of 0.147 on average. In Figs. 3(b)–(f), pixels of the *flame* image are tinged according to their partition modes (see the top left image of Fig. 2(e)) in RGB space. We note that in partition mode of distance (Fig. 3(f)), the points with the same class-color appear in the neighborhood, while points in different classes are mixed together for all other partition modes. It is also intuitively shown that the color and distance are highly correlated in text effects.

The distance has also shown its importance in characterizing the scale of patterns. Firstly, for different patch sizes, we calculate the average patch difference (the Sum of the Squared Differences, SSD) between all patches in a partition and their best matches on the same image, which forms a response curve of scale. Then, for all the $N$ partitions with the same partition mode, we have $N$ response curves that show the impacts of scales. Two examples of response curves for *denim* image are shown in Figs. 4(a) and (b), where each point shows its average and standard deviation of patch differences under the same partition and scale. To compare the reliability of all partition modes, two terms are utilized: (i) inter curve standard deviation $\sigma_{\text{inter}}$: the average of the scale-wise standard deviations of average responses at same partitions; and (ii) intra curve standard deviation $\sigma_{\text{intra}}$: the average of point-wise standard deviations for all scales and partitions. A higher $\sigma_{\text{inter}}$ implies that sub-effects are easier to be distinguished by their locations, while a lower $\sigma_{\text{intra}}$ implies patches in the same partition react uniformly to scale changing and possibly share common optimal scale for description. Therefore, we evaluate the reliability by

$$r_{\text{scale}} = \sigma_{\text{inter}}/\sigma_{\text{intra}}. \tag{2}$$

The reliability of all the five partition modes is then given in Table 1 where the factor of distance achieves highest to characterize the patch scales.

As a conclusion, there exist *high correlations between patch patterns (i.e. color and scale) and their distances to text skeletons*. These are reasonable essential characteristics for high-quality text effects.

**Table 1**
Reliability between patch patterns and different partitions.

| $r$ | rand | grid | angle | ring | dist |
|---|---|---|---|---|---|
| color | 0.063 | 0.106 | 0.119 | 0.105 | **0.147** |
| scale | 0.153 | 0.793 | 0.486 | 0.590 | **0.950** |

## 4. Proposed method

### 4.1. Text effects statistics estimation

We now convert the aforementioned analysis into patch statistics that can be directly used as the transfer guidance. For our patch-based algorithm, in the following we use $p$ and $q$ to denote the pixels in $T/T'$ and $S/S'$, respectively, and use $P(p)$ and $P'(p)$ to represent the patches centered at $p$ in $T$ and $T'$, respectively. The same goes for patches $Q(q)$ and $Q'(q)$ in $S$ and $S'$.

#### 4.1.1. Optimal patch scale detection

Inspired by Frigo et al. (2016), we propose a simple yet effective approach to detect the optimal patch scale scal($q$) to depict texture patterns round $q$. Considering large patches can better depict texture styles than small patches that capture limited contextual information, patch size is encouraged to be large enough. However, large size will make it hard to find precisely matched patches, and will cause blurring problem for our patch-based algorithm. Therefore, we define the optimal scale as the largest scale where the target patch is still under a given criterion designed to prevent blurring. For each patch, we emulate its scale from large to small until it begins to satisfy the criterion to find its optimal scale.

Specifically, we use a fixed patch size of $m \times m$ and resize the image to accomplish multiple scales. Given a predefined downsample factor $s$ and the max scale $L$, let $S_\ell$ be the downsampled source $S$ with a scale rate of $1/s^{\ell-1}$ and $Q_\ell(q)$ be the patch centered at $q/s^{\ell-1}$ in $S_\ell$. $S'_\ell$ and $Q'_\ell(q)$ are similarly defined. If $\hat{q} \neq q$ is the best correspondence of $q$ at scale $\ell$ that minimizes

$$d_\ell(q, \hat{q}) = \|Q_\ell(q) - Q_\ell(\hat{q})\|^2 + \|Q'_\ell(q) - Q'_\ell(\hat{q})\|^2, \tag{3}$$

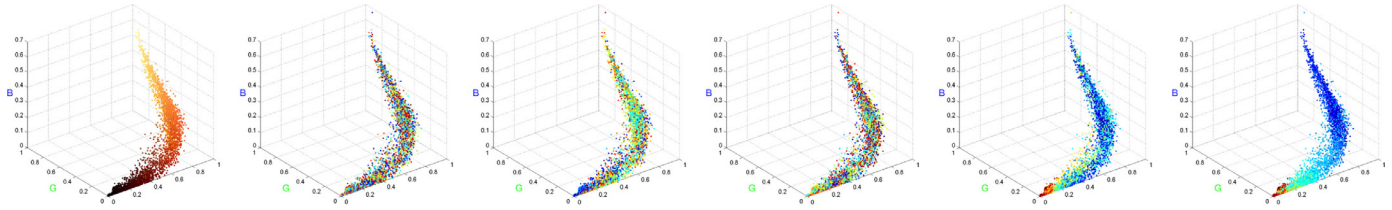
then a criterion at scale $\ell$ is defined as

$$\zeta_\ell(q, \hat{q}) = (\sigma_\ell + \sqrt{d_\ell(q, \hat{q})} \leq \omega), \tag{4}$$

where $\sigma_\ell = \sqrt{Var(Q'_\ell(q))}/2$. The criterion suggests that at the ideal scale, the target patch should find good matches and avoid having complex textures that are susceptible to blurring issues. Patches that satisfy the criterion set $\ell$ as their optimal scales, while others pass through to finer scale $\ell - 1$. The optimal patch scale detection is summarized in Algorithm 1. An example of the optimal scales for the *flame* image is shown in Fig. 5(a). It is found that the textured region near the text requires finer patch scales than the outer flat region. For better visualization, we show the receptive field of patch $Q(q)$ by resizing it at a scale rate of $s^{\text{scal}(q)-1}$ in Fig. 5(b).

#### 4.1.2. Robust normalized distance estimation

Here we first define some concepts. In the text image, its text region is denoted by $\Omega$. The skeleton skel($\Omega$) is a kernel path within $\Omega$. We use dist($q$, $A$) to denote the distance between pixel $q$ and its nearest pixel in set $A$. We are going to calculate dist($q$, skel($\Omega$)). For $q$ on the text contour $\delta\Omega$, the distance is also known as the text radius $r(q)$. Fig. 6(b)



(a) RGB (b) random (c) grid (d) angle (e) ring (f) distance

**Fig. 3.** Statistics of the text effects images: High correlation between pixel colors and distances. (a) Pixels in RGB space. (b)-(e) Pixels are mixed together for partition modes of random, grid, angle and ring in RGB space. (f) Pixels are distinguished from each other by their distances in RGB space.

gives the visual interpretation.

We extract skel(Ω) from $S$ by means of the standard morphology hit-or-miss transform operator using Lantuéjoul's formula (Lantuéjoul, 1977). To ensure the distance invariant to the text radius, we aim to normalize the distance so that the normalized text radius equals to 1. Simply dividing the distance by the text radius is unreliable because the inaccuracy of the obtained skel(Ω) leads to errors both in the numerator and denominator as well. To address this issue, we estimate corrected text radius $\tilde{r}(q)$ based on text statistics and use the accurate dist($q$, $\delta\Omega$) to derive normalized $\widetilde{\text{dist}}(q, \text{skel}(\Omega))$.

Specifically, we sort $r(q)$, $\forall q \in \delta\Omega$ and obtain their rankings $rank(q)$. We observe that the relation between $r(q)$ and $rank(q)$ can be well modeled by linear regression, as shown in Figs. 6(d). From Figs. 6(b)(d), we discover that outliers (tinged with red color) assemble at small values. We empirically assume the leftmost 20% points are outliers and perform linear regression on the remaining 80% boundary points to obtain the regression coefficients $k$ and $b$. Then the corrected text radius $\tilde{r}(q)$ are calculated by

$$\tilde{r}(q) = \max(\text{dist}(q, \text{skel}(\Omega)), 0.2k|\delta\Omega| + b), \tag{5}$$

where $|\delta\Omega|$ is the pixel number of $\delta\Omega$. Finally, the normalized distance is obtained,

$$\widetilde{\text{dist}}(q, \text{skel}(\Omega)) = \begin{cases} 1 - \text{dist}(q, \delta\Omega)/\tilde{r}(q_\perp), & \text{if } q \in \Omega \\ 1 + \text{dist}(q, \delta\Omega)/\bar{r}, & \text{other} \end{cases}, \tag{6}$$

where $q_\perp \in \delta\Omega$ is the nearest pixel to $q$ along $\delta\Omega$ and $\bar{r} = 0.5k|\delta\Omega| + b$ is the mean text radius.

**Input:** Image $S, S'$, parameters $L, s, \omega$
**Output:** Optimal scale scal($q$) for each pixel $q$

Initialize $Z = \{q | q \in S\}$ and scal($q$) = 1, $\forall q \in Z$
**for** $\ell = L, \ldots, 2$ **do**
  **for all** $p \in Z$ **do**
    Compute $\hat{q} = \arg\min_{\hat{q}} d_\ell(q, \hat{q})$
    **if** $\zeta_\ell(q, \hat{q})$ is true **then**
      scal($q$) = $\ell$
      $Z = Z \setminus \{q\}$
    **end if**
  **end for**
**end for**

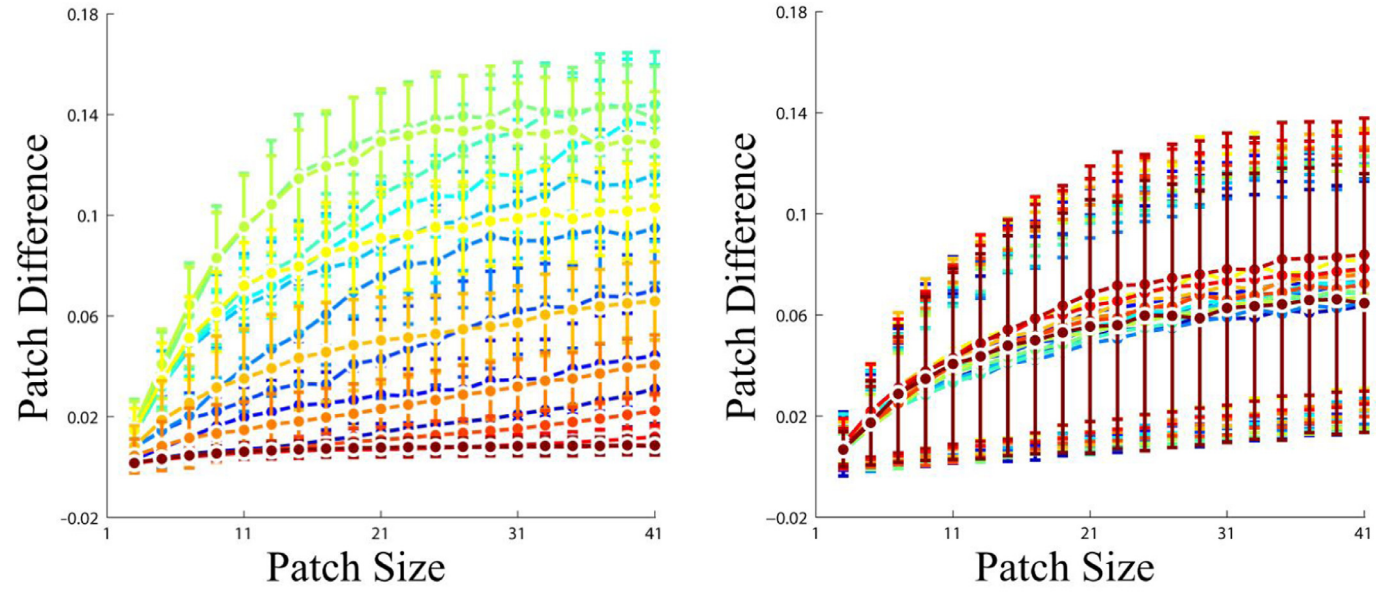
**Algorithm 1.** Optimal patch scale detection.

For simplicity, we omit skel(Ω) and use dist($q$) to refer to $\widetilde{\text{dist}}(q, \text{skel}(\Omega))$ in the following.

#### 4.1.3. Optimal scale posterior probability estimation

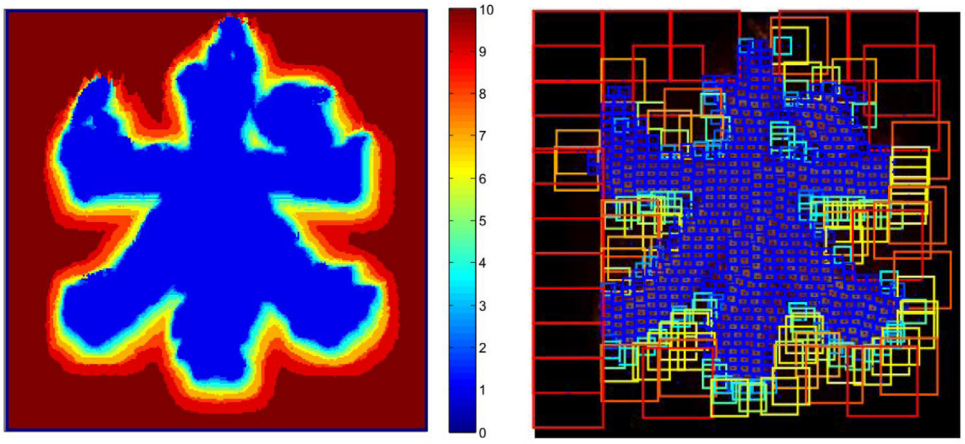
In this section, we derive the posterior probability of the optimal patch scale to model the aforementioned high correlation between patch patterns and their spatial distributions.

We uniformly quantify all distances into 100 bins and denote bin($q$) as the bin $q$ belongs to. Then, a 2-d histogram $hist(\ell, x)$ is computed:




(a) Response curves (distance mode) (b) Response curves (random mode)

**Fig. 4.** Statistics of the text effects images: High correlation between patch scales and distances. Patches with similar distances have uniform responses to changes of their size.




(a) Optimal scale map (b) Visualized patch scale

**Fig. 5.** Detected optimal patch scales for the *flame* image.

$$hist(\ell, x) = \sum_{q} \psi(\text{scal}(q) = \ell \wedge \text{bin}(q) = x), \tag{7}$$

where $\psi(\cdot)$ is 1 when the argument is true and 0 otherwise. And the joint probability of the distance and the optimal scale can be estimated as,

$$\mathcal{P}(\ell, x) = hist(\ell, x) / \sum_{\ell, x} hist(\ell, x). \tag{8}$$

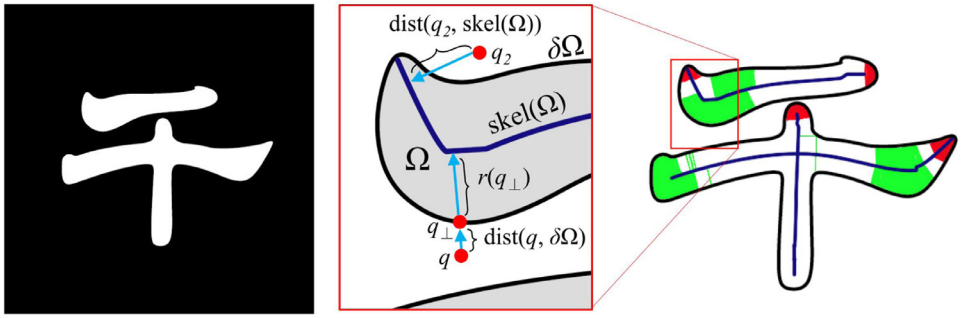
Finally, the posterior probability $\mathcal{P}(\ell|\text{bin}(q))$ for $\ell$ being the appropriate scale to depict the patches with distances corresponding to bin($q$) can be deduced:

$$\mathcal{P}(\ell|\text{bin}(p)) = \mathcal{P}(\ell, \text{bin}(p)) / \sum_{\ell} \mathcal{P}(\ell, \text{bin}(p)). \tag{9}$$

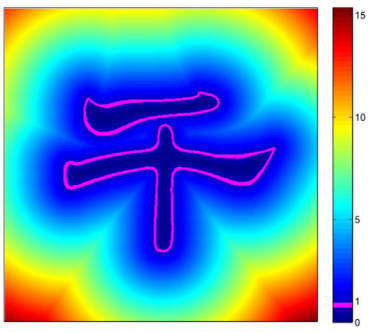
We argue that optimal scale posterior probability is one of the characteristics of the text effects. To make $T'$ and $S'$ share exactly the same text effects, we assume the target image has the same posterior probability as the source image. And we will use this probability to select patch scales statistically for texture synthesis to adapt extremely various text effects.

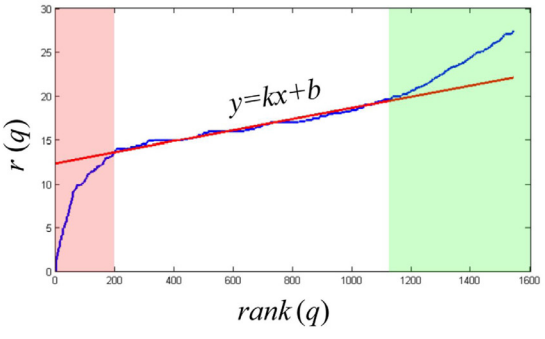
### 4.2. Text effects transfer

In this section, we describe how we adapt conventional texture synthesis method to dealing with the challenging text effects. We build on the texture synthesis method of Wexler et al. (2007) and its variants (Darabi et al., 2012) using random search and propagation as in PatchMatch (Barnes et al., 2009; 2010). We refer to these papers for




(a) Text image (b) Notation definition




(c) $\widetilde{\text{dist}}(q, \text{skel}(\Omega))$ (d) Linear relation of $r(q)$ and $rank(q)$

**Fig. 6.** Robust normalized distance estimation. (a) The text image. (b) Our detected text skeleton and the notation definition. (c) The estimated normalized distance. The distance of the pixels on the text boundary to the text skeleton are normalized to 1 (colored by magenta). (d) The statistics of the text radius. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

details of the base algorithm.

We apply character shape constrains to the patch appearance measurement to build our baseline, and further incorporate estimated text effects statistics to accomplish adaptive multi-scale style transfer (Section 4.2.2). Then a distribution term is introduced to adjust the spatial distribution of the text sub-effects (Section 4.2.3). Finally, we propose a psycho-visual term that prevents texture over-repetitiveness for naturalness (Section 4.2.4).

#### 4.2.1. Objective function

We augment the texture synthesis objective function in Wexler et al. (2007) by including a distribution term and a psycho-visual term. And our objective function takes the following form,

$$\min_{q} \sum_{p} E_{\text{app}}(p, q) + \lambda_1 E_{\text{dist}}(p, q) + \lambda_2 E_{\text{psy}}(p, q), \tag{10}$$

where $p$ is the center position of a target patch in $T$ and $T'$, $q$ is the center position of the corresponding source patch in $S$ and $S'$. The three terms $E_{\text{app}}$, $E_{\text{dist}}$ and $E_{\text{psy}}$ are the appearance, distribution and psycho-visual terms, respectively, which are weighted by $\lambda_1$ and $\lambda_2$ to together make up the patch distance.

#### 4.2.2. Appearance term (texture style transfer)

The original texture synthesis algorithm of Wexler et al. (2007) minimizes the SSD of two patches sampled from texture image pair $S'/T'$. We adapt it to text effects transfer tasks by applying additional SSD of two patches sampled from the text image pair $S/T$:

$$E_{\text{app}}(p, q) = \lambda_{\text{text}} \|P(p) - Q(q)\|^2 + \|P'(p) - Q'(q)\|^2, \tag{11}$$

where $\lambda_{\text{text}}$ is a weight that compromises between the color difference and text shape difference. We take the objective function that only minimizes the appearance term in Eq. (11) as our baseline.

Stylized texts often contain multiple sub-effects with different optimal representation scales. Thus, in addition to the baseline, we propose the adaptive scale-aware patch distance by incorporating the estimated posterior probability,

$$\begin{aligned} E_{\text{app}}(p, q) \;=\; & \lambda_{\text{text}} \sum_{\ell} \mathcal{P}(\ell|\text{bin}(p)) \, \|P_\ell(p) - Q_\ell(q)\|^2 \\ & + \sum_{\ell} \mathcal{P}(\ell|\text{bin}(p)) \, \|P'_\ell(p) - Q'_\ell(q)\|^2. \end{aligned} \tag{12}$$

The posterior probability helps to explore patches through multiple appropriate scales for better textures synthesis.

#### 4.2.3. Distribution term (spatial style transfer)

The distribution of sub-effects highly correlates with their distances to the text skeleton. Based on this prior, we introduce a distribution term,

$$E_{\text{dist}}(p, q) = (\text{dist}(p) - \text{dist}(q))^2 / \max(1, \text{dist}^2(p)), \tag{13}$$

which encourages the text effects of the target to share similar distribution with the source image, thereby realizing a spatial style transfer. To ensure that the cost is invariant to the image scale, we add the denominator $\max(1, \text{dist}^2(p))$.

#### 4.2.4. Psycho-visual term (naturalness preservation)

Texture over-repetitiveness can seriously reduce human subjective evaluation in the aesthetics. Therefore, we aim to penalize certain source patches to be selected repetitiously.

Let $\Phi(q)$ be the set of pixels that currently finds $q$ as its correspondence and $|\Phi(q)|$ be the size of the set. We define the psycho-visual term as,

$$E_{\text{psy}}(p, q) = |\Phi(q)|. \tag{14}$$

From the perspective of $q$, we can better understand the repetitiveness penalty:

$$\sum_{p} |\Phi(q)| = \sum_{q} \sum_{p \in \Phi(q)} |\Phi(q)| = \sum_{q} |\Phi(q)|^2. \tag{15}$$

Since $\sum_q |\Phi(q)| = |T|$ is constant, Eq. (15) reaches the minimum when all $|\Phi(q)|$ equals. It means our psycho-visual term encourages source patches to be used evenly.

#### 4.2.5. Function optimization

We follow the iterative coarse-to-fine matching and voting steps as in Wexler et al. (2007). In the matching step, PatchMatch algorithm (Barnes et al., 2009) is adopted. We fix $\Phi(q)$ during the search and propagation stages, and update $\Phi(q)$ after each iteration of these stages for the psycho-visual term. Meanwhile, the initialization of $T'$ plays an important role in the final results, since our guidance map provides very few constraints on textures. We vote the source patches that are searched to only minimize Eq. (13) to form our initial guess of $T'$. This simple strategy improves the final results significantly as shown in Fig. 9.

### 4.3. Transfer for words

For multiple characters in a word, we additionally consider the relationships of text effects among characters. Specifically, the texture similarity of matched strokes within multiple characters is controlled. We focus on *stroke ends*, namely, protruding shape features on the input characters, as shown by red boxes in Fig. 7. The shape of a stroke, its legibility, is mostly determined by its trunk. Intuitively, an experienced designer tends to put more stylish elements upon stroke ends rather than the trunk to find a compromise between legibility and aesthetics. As a result, stroke ends usually characterize a large part of the style of each stroke.

First, we detect stroke ends in $T$ as the patches centered at the endpoints of the text skeleton, with a size of $R \times R$, where $R = 4\bar{r} + 1$ approximates double the text width. $\bar{r}$ is the text radius as in Eq. (6). Stroke ends that find good matches (their SSD is less than a threshold of $0.05{*}R^2$) among characters form a set, denoted as $\Psi$. Supposing pixel $p$

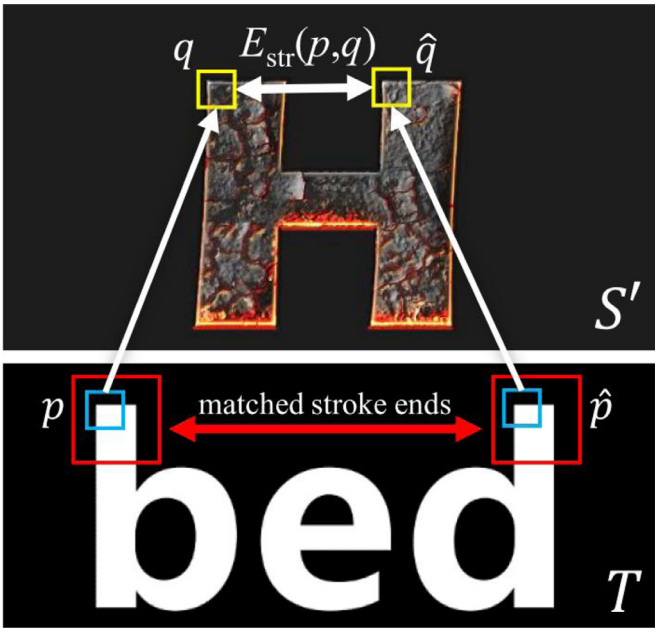
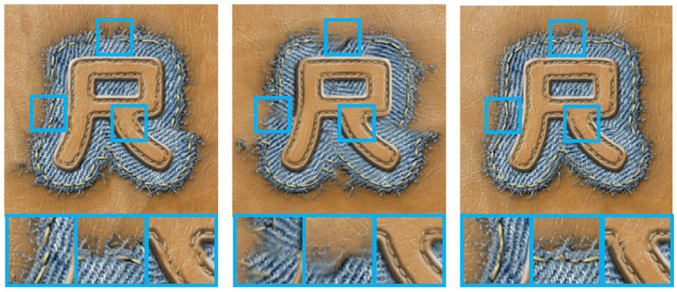



**Fig. 7.** Stroke term $E_{\text{str}}(p, q)$ controls the texture similarity of similar strokes. The red patches are two matched stroke ends. $p$ and $\hat{p}$ are corresponding pixels in each stroke end. The blue target patches $P(p)$ and $P(\hat{p})$ find yellow patches $Q'(q)$ and $Q'(\hat{q})$ for texture synthesis. $E_{\text{str}}(p, q)$ measures the Euclidean distance between $q$ and $\hat{q}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) $m = 5, L = 1$ (b) $m = 15, L = 1$ (c) $m = 5, L = 5$

**Fig. 8.** Effects of the multi-scale strategy. (a) Results using single-scale $5 \times 5$ patches. (b) Results using single-scale $15 \times 15$ patches. (c) Results using joint $5 \times 5$ patches over 5 scales.

(a) $\lambda_1 = 0.0$ (b) $\lambda_1 = 0.1$ (c) $\lambda_1 = 0.1$+ Init

**Fig. 9.** Effects of the distribution term. (a) Results without distribution term. (b) Results obtained by random initialization and optimization with distribution term. (c) Results obtained by both initialization and optimization with distribution term.

in one stroke end corresponds to pixel $\hat{p}$ in its matched stroke end, and the source patches of $P(p)$ and $P(\hat{p})$ are centered at $q$ and $\hat{q}$, respectively, then we augment our objective function with an additional term $E_{\text{str}}$ to control the texture similarity of similar strokes,

$$E_{\text{str}}(p, q) = \begin{cases} \min(1, \|q - \hat{q}\|/R), & \text{if } p \in \Psi \\ 0, & \text{other} \end{cases}. \tag{16}$$

Fig. 7 gives a visual interpretation of $E_{\text{str}}(p, q)$. We finally optimize the total energy, defined as

$$\min_q \sum_p E_{\text{app}}(p, q) + \lambda_1 E_{\text{dist}}(p, q) + \lambda_2 E_{\text{psy}}(p, q) + \lambda_3 E_{\text{str}}(p, q), \tag{17}$$

where $\lambda_3$ is a relative weighting coefficient. The minimization of $E_{\text{str}}(p, q)$ is solved in a similar manner as that of $E_{\text{psy}}(p, q)$. We fix $\hat{q}$ during the search and propagation stages of PatchMatch, and update $\hat{q}$ after each iteration of these stages.

## 5. Experimental results

In the experiment, the patch size is $5 \times 5$ and the max scale $L = 5$. We build an image pyramid of 10 levels with a fixed coarsest size (length of the image short edge is 32). At level $\ell$, joint patches over scales from $\ell$ to $min(10, \ell + L - 1)$ are used. Unless stated otherwise, the weights $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_{\text{text}}$ to balance different terms are empirically set to 0.1, 0.005, 5.0 and 10, respectively. The parameter $\omega$ for the filter criterion is 0.3. In addition to the examples in this paper, all the results are included in the supplementary material.

### 5.1. Effect of the three terms

#### 5.1.1. Appearance term

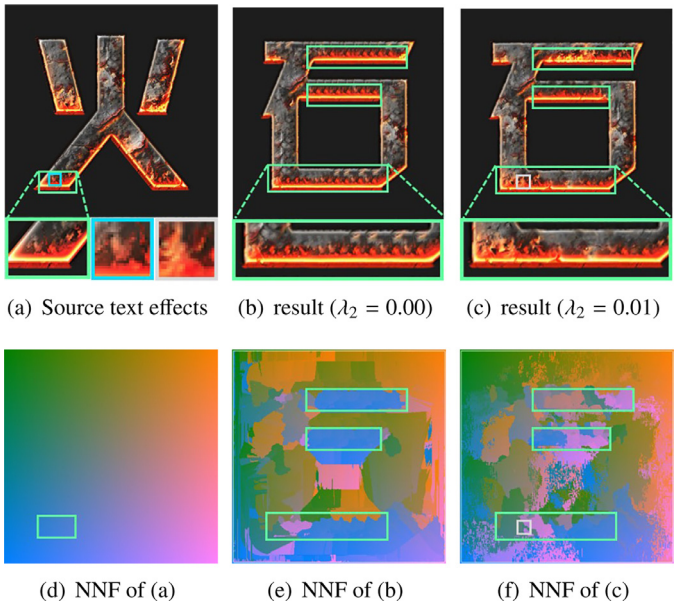
The advantages of the proposed appearance term lie in two aspects: (i) Preserve coarse grained texture structures. (ii) Preserve texture details. We show in Figs. 8(a) and (b) the *denim* style generated using single-scale $5 \times 5$ and $15 \times 15$ patches, respectively. Small patches capture very limited contextual information, thus it cannot guarantee the structure continuity. As can be seen in Fig. 8(a), sewing threads look cracked and are not along the uniform directions. However, choosing large patches leads to smoothing out tiny thread residues as in Fig. 8(b). These problems are well solved by jointly using $5 \times 5$ patches over 5 scales as in Fig. 8(c), where the overall shape is well preserved and the details like sewing threads look more vivid.

#### 5.1.2. Distribution term

The distribution term ensures the sub-effects in the target image and the source example are similarly distributed, which is the basis of our assumption in Section 4.1.3 that the posterior probabilities $\mathcal{P}(\ell|x)$ in $T'$ and $S'$ are the same. Fig. 9 shows the effects of the distribution term on the *flame* style. Without distribution constraints, the flames appear randomly in the black background. The distribution term adjusts the flames to better match their spatial distribution as that in the source example.

#### 5.1.3. Psycho-visual term

The effects of our psycho-visual term are shown in Fig. 10. The *lava* textures synthesized without the psycho-visual penalty densely repeat the red cracks (see green rectangle in Fig. 10(a)) in three regions highlighted by green rectangles in Fig. 10(b), which causes obvious unnaturalness. By increasing the penalty, the reuse of the same source texture is greatly restrained (Fig. 10(c)). It is better illustrated by the nearest neighbor fields (NNF) of Fig. 10(f) that source patches used to synthesize textures in green rectangular regions are more widely distributed. For example, the grey patch in Fig. 10(c) composes of textures from two areas (tinted with pink and green in NNF). We show its best matching patch in $S'$ by blue rectangle and they look distinctly different. This means our method agilely combines different source patches to create brand-new textures. Thus, the psycho-visual term can effectively penalize texture over-repetitiveness and encourage new texture creation.



(a) Source text effects (b) result ($\lambda_2 = 0.00$) (c) result ($\lambda_2 = 0.01$)

(d) NNF of (a) (e) NNF of (b) (f) NNF of (c)

**Fig. 10.** Effects of the psycho-visual term, which penalizes texture over-repetitiveness and encourages new texture creation. Top row: Source text effects and our results with and without the psycho-visual term. The blue patch in (a) is the best matching patch of the grey patch in (c). For visual inspection, regions highlighted by green, blue and grey rectangles are enlarged. Bottom row: The visualized nearest neighbor fields (NNF). (d) NNF tints each pixel in different positions of $S'$ with a specific color. (e)-(f) NNF shows where each pixel in the results comes from $S'$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(a) $(S, S')$ (b) $T$ (c) Analogies (d) Split & match (e) Neural doodle (f) Baseline (g) Proposed

**Fig. 11.** Comparison with state-of-the-art methods on various text effects. From top to bottom: *neon, smoke, denim, flame, neon2, smoke2, denim2, denim3*. (a) Input source text effects with their raw text counterparts in the lower-left corner. (b) Target text. (c) Results of Image Analogies (Hertzmann et al., 2001). (d) Results of Split and Match (Frigo et al., 2016). (e) Results of Neural Doodle (Champandard, 2016). (f) Results of our baseline method. (g) Results of the proposed method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.) Image credits: http://www.zcool.com.cn/work/ZMTcwNjEwMTI=.html.

#### 5.1.4. Combination of the three terms

It is worth noting that *the proposed three terms are complementary*: First, the appearance and distribution terms emphasize local texture patterns and global sub-effects distributions, respectively. The former depicts the relationship of low-level color features between $S'$ and $T'$, while the latter exploits the relationship of complementary mid-level position features between $S'$ and $T'$. Second, the appearance and distribution terms jointly evaluate objective patch similarities. Meanwhile,

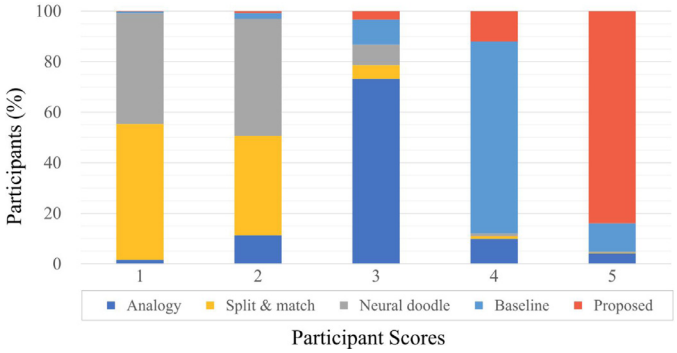



**Fig. 12.** Score distribution for each method from our 90-person study. The figure gives the percentage of users who gave a certain score to a given style transfer technique.

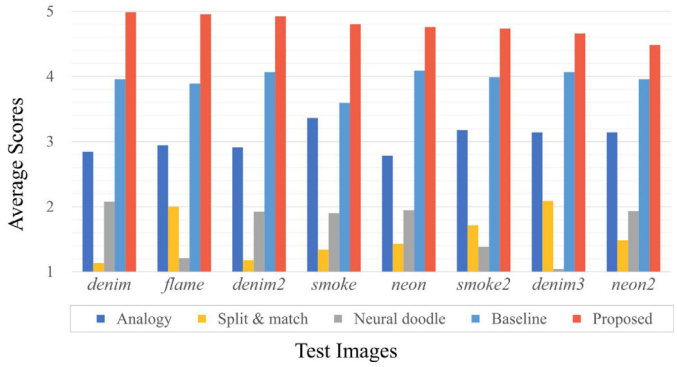



**Fig. 13.** Average evaluation scores for each test image in Fig. 11 from our 90-person study, sorted by our score. The proposed method outperforms other state-of-the-art methods in all cases.

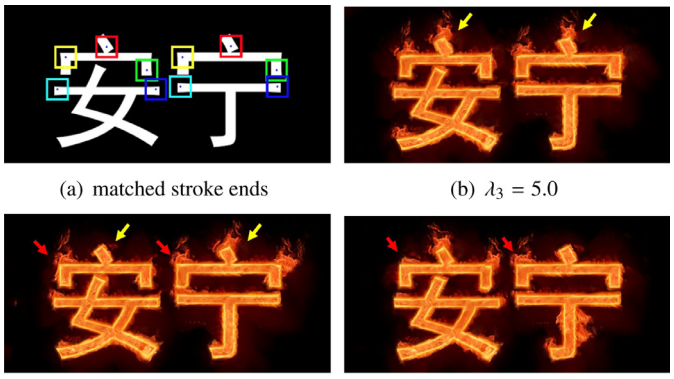



**Fig. 14.** Texture effects transfer for words of *flame2*. (a) Matched stroke ends. (b) Results with positive stroke similarity constraint. As shown by the yellow arrows, matched strokes are stylized in a more consistent way. (c) Results without stroke term. (d) Results with negative stroke similarity constraint. As shown by the red arrows, matched strokes are stylized in a more diverse way. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the psycho-visual term complements these two terms by incorporating aesthetic subjective evaluations.

### 5.2. Comparisons for text effects transfer

#### 5.2.1. Visual comparison

In Fig. 11, we present a comparison of our algorithm with three state-of-the-art style transfer techniques as well as our baseline. The first method is the pioneering non-parametric Image Analogies (Hertzmann et al., 2001). The textures in their results repeat locally and look disordered globally with evident patch boundaries. The second method is our implementation of non-parametric Split and Match (Frigo et al., 2016), which synthesizes textures using adaptive patch sizes. The original method directly transfers the style in $S'$ to $T$ without the help of $S$. To make a fair comparison, we incorporate the guidance by using $S$ instead of $S'$ in the split stage. This method fails to generate textures in the background and produces plain stylized results. The third method, parametric Neural Doodle (Champandard, 2016), is based on the combination of MRF and CNN (Li and Wand, 2016a) and incorporates semantic maps for analogy guidance. While the color palette of the example text effects is transferred, fine textures are poorly synthesized. The text shape is lost as well. The fourth method is our baseline. We take the objective function that only minimizes the appearance term in Eq. (11) as our baseline. Without any spatial constraints, the baseline method transfers fine textures but fails to keep the overall sub-effects distribution and generates artifacts in the

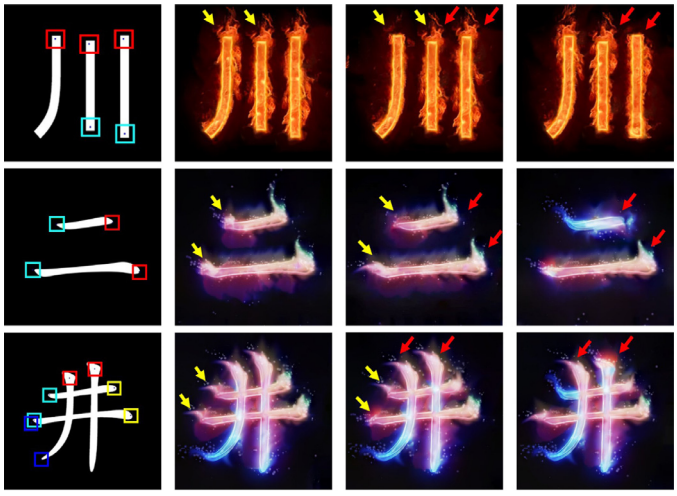



**Fig. 15.** Effect of the stroke term on single characters. From top to bottom: *flame3, neon2, neon3*. (a) Matched stroke ends. (b) Results with positive stroke similarity constraint. As shown by the yellow arrows, matched strokes are stylized in a more consistent way. (c) Results without stroke term. (d) Results with negative stroke similarity constraint. As shown by the red arrows, matched strokes are stylized in a more diverse way. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 2**
The average score of results obtained with different $\lambda_3$.

| Image | $\lambda_3 = 5.0$ | $\lambda_3 = 0.0$ | $\lambda_3 = -5.0$ |
|---|---|---|---|
| *flame2* | 1.80 | 2.00 | 2.20 |
| *flame3* | 1.20 | 2.10 | 2.70 |
| *neon2* | 1.70 | 1.30 | 3.00 |
| *neon3* | 1.60 | 2.00 | 2.40 |
| Average | 1.575 | 1.850 | 2.575 |

**Table 3**
The average running times (seconds) of different methods.

| Image size | Analogies | Split & match | Neural doodle | Proposed |
|---|---|---|---|---|
| 256 × 256 | 26.0 | 88.0 | 77.0 | 34.0 |
| 320 × 320 | 57.0 | 158.0 | 125.0 | 49.5 |
| 400 × 400 | 237.6 | 281.6 | 186.8 | 79.4 |
| Average | 166.0 | 226.5 | 157.6 | 66.3 |


(a) Target text (b) *flame* (c) *lava* (d) *rust* (e) *drop* (f) *pop* (g) *blink*

**Fig. 16.** Apply different text effects to representative characters (Chinese, alphabetic, handwriting).

background. The proposed method improves our baseline by incorporating adaptive scale-aware patch distance (Eq. (12)) and adding a distribution term and a psycho-visual term. The adaptive scale-aware patch distance allows our method to synthesize textures at their optimal scale. The distribution term mimics the sub-effects distribution in $S'$ and the psycho-visual term boosts local synthesis variety, as we have already mentioned in Section 5.1. Thus, the proposed method outperforms all other methods, preserving both local textures and the global sub-effects distribution.

#### 5.2.2. User study

To better understand the performance of these methods, we perform user studies for quantitative evaluations. Participants are shown the eight stylization cases in Fig. 11. Each subject is asked to score the style similarity between the reference and the stylized results from 5 to 1 (5 being most similar and 1 most different). To ensure the fairness, the orders of five methods randomly change every round. A total of 90 subjects participate in this study and a total of 3600 scores are tallied. The number of female participants is 30 while the age range is from high school to retirees. Most participants are not experts in art and therefore provide relatively objective judgments on style. Their professions are diverse, including computer science, education, law, management. The proposed method obtains the best average score of 4.79, while the average scores of Image Analogies (Hertzmann et al., 2001), Split and Match (Frigo et al., 2016), Neural Doodle (Champandard, 2016) and our baseline are 3.04, 1.54, 1.68 and 3.95, respectively. Fig. 12 shows the distribution of the participants' scores for five techniques. Overall, the scores of our method are mainly distributed over 4 and 5.

Simultaneously, we investigate the average scores of individual test images. As shown in Fig. 13, most of our results are rated above 4.7 and are significantly higher than the other four methods. We observe that for styles that well match the spatial distribution-based characteristics like *denim* and *flame*, our results are most favored. Meanwhile, our method obtains slightly lower scores for *denim3* and *neon2*. The simple less-structured texture in *denim3* is easy to synthesize. As a result, the proposed method does not show much more advantages compared to other methods. In *neon2*, our method uses the horizontal textures in $S'$ to synthesize the horizontal strokes in $T'$, which creates plausible results but fails to cover blue neon colors. We will show in Section 5.3 that this problem can be well solved using the proposed stroke term.

### 5.3. Effect of the stroke term

The effects of the stroke term $E_{str}(p, q)$ are shown in Fig. 14, where two characters share similar components (radicals of Chinese characters) in their upper part. For a positive $\lambda_3$, $E_{str}(p, q)$ attempts to stylize the radicals of two characters in a more consistent way, which may possibly be favored in applications like uniform typography generations. For a negative $\lambda_3$, $E_{str}(p, q)$ attempts to transfer more diverse flame textures onto two radicals, which well meets the requirement of design flexibility.

In some languages like Chinese, a single character may also contain similar strokes. By allowing similar stroke ends to be found inside characters, we can apply our stroke term to control style diversity within single characters. Fig. 15 shows the effect of $E_{str}(p, q)$ on single characters. As in the case of words, positive $\lambda_3$ effectively unifies *flame* and *neon* sub-effects within single characters, while negative $\lambda_3$ diversifies them. Note that in Fig. 15(d), the term positively impacts the *neon2* result for covering all the colors, making its color style more consistent with the source text effects.

Furthermore, we conduct a user study for quantitative comparisons. Ten participants are shown four stylization cases in Fig. 14 and Fig. 15. Each subject is asked to score the style diversity of each results from 1 to 3 (1 being most unified and 3 most diversified). The average scores for the four cases are shown in Table 2. As the weight of the stroke term decreases, the score of the result gets higher, verifying that the stroke term can well regulate the synthesis to be more consistent or diverse.

**Fig. 17.** An overview of our *flame* typography library. The bigger image at the top left corner serves as the example to generate the other 774 characters. The whole library as well as the other stylized libraries can be found in the supplementary material.

#### 5.3.1. Running time

We compare the running times of different methods on the test images in Fig. 11, including one $256 \times 256$ image, two $320 \times 320$ images and five $400 \times 400$ images. Table 3 shows the average running times on these images using a GeForce GTX 1080 GPU for deep-based Neural Doodle (Champandard, 2016) and an Intel Xeon E5-1607 CPU for other methods. Even with a high-performance GPU, Neural Doodle (Champandard, 2016) is still expensive to run due to the optimization loop. Meanwhile, it can be observed that the running time of Image Analogies (Hertzmann et al., 2001) increases significantly as the image size increases. Moreover, the efficiency of Split and Match (Frigo et al., 2016) is limited because the patch size used in this method can be large (the maximum size is $32 \times 32$ in our implementation of it). On the contrary, the proposed method achieves promising visual effect with only small $5 \times 5$ patches in our multi-scale strategy, suffering fewer computational burdens. And it is further sped up by parallelization using four threads. As a result, the proposed method is advantageous in computational efficiency.

(a) $(S, S')$ (b) $(T, T')$

**Fig. 18.** We extend our method to texture rendering for icons. We transfer the special effects from text in (a) to icons in (b).

### 5.4. Transfers between styles, languages and fonts

In Fig. 16, we present an illustration of style transfer from six very different text effects to three representative characters (Chinese, alphabetic, handwriting). This experiment covers challenging transformations between styles, languages and fonts. Thanks to distance normalization and multi-scale strategy, our algorithm accomplishes to transfer the text effects regardless of character shapes and texture scales, providing a solid tool for artistic typography.

### 5.5. Typography library generation

We show our *flame* typography library including as much as 775 frequently used Chinese characters. Due to the space limitation, only the first 32 of them are presented in Fig. 17. The whole library as well as the other typography libraries are included in our supplementary material. The extensive synthesis results demonstrate the robustness of our method to varied character shapes.

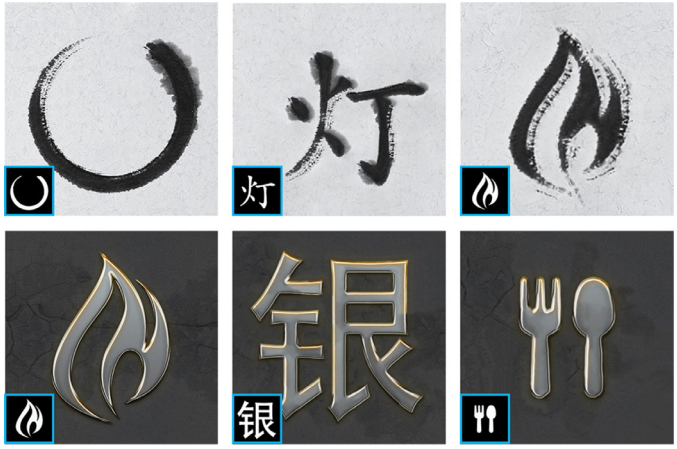
### 5.6. Extensions for stroke-based graphics

Our method can be extended to stroke-based graphics such as icons. Fig. 18 shows that our method can transfer text effects onto binary icons to render a rusty heart or a blazing flame. Besides text effects, the input of our method for style references can also be rendered basic shapes or complex icons. In Fig. 19, the ink/metal effects are successfully transferred from a ring shape/flame icon to text and icons. To obtain satisfying results, we only tune the repetitiveness weight $\lambda_2$ for these icons.

### 5.7. Limitations and future work

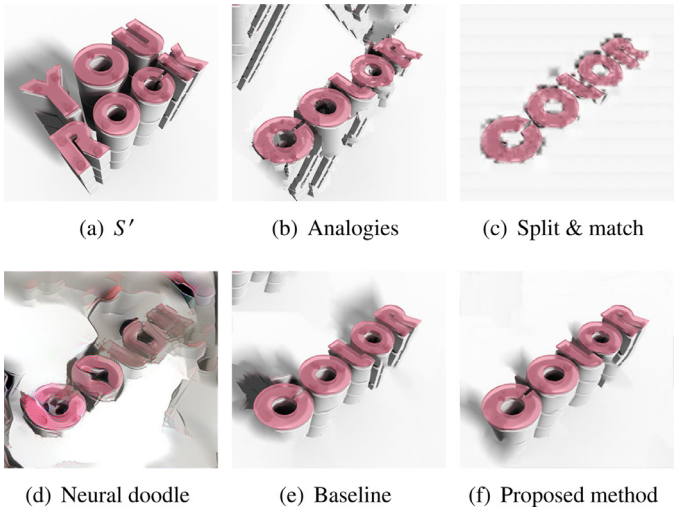
While the proposed method is able to generate visually appealing results, some cases still pose challenges to our approach. As discussed in the user study, since we focus on synthesizing each target stroke using the best matching source textures, the exhaustiveness of the source texture usage is not ensured (*e.g.* the blue textures in *neon2* of Fig. 11 are not transferred). Stylization results will be improved if the algorithm could diversify the usage of source strokes. It guides us onto an interesting avenue of future work.

We would like to point out that our method is designed to deal with texture distributions within 2D image planes. The distribution-aware scheme could provide a certain degree of robustness to simple 3D textures like the *pop* example in Fig. 16 and the *metal* example in Fig. 19 . However, for challenging 3D cases as in Fig. 20, our method struggles to create more plausible textures compared to other methods, and still cannot keep the accurate spatial relationship of the complex 3D structures. In the future, 3D analysis and reconstruction technologies may be introduced to make the synthesis process more reliable in 3D cases.



(a) $(S, S')$ (b) $(T, T')$ (c) $(T, T')$

**Fig. 19.** We extend our method to stroke-based graphics. We transfer the ink/metal effects from stroke-based shapes in (a) to text in (b) and other shapes in (c).



(a) $S'$ (b) Analogies (c) Split & match

(d) Neural doodle (e) Baseline (f) Proposed method

**Fig. 20.** Performance on complex 3D text effects. (a) Input source text effects. (b) Results of Image Analogies (Hertzmann et al., 2001). (c) Results of Split and Match (Frigo et al., 2016). (d) Results of Neural Doodle (Champandard, 2016). (e) Results of our baseline method. (f) Results of the proposed method. Image credits: http://www.zcool.com.cn/work/ZMTgwMDc4ODQ=.html.

## 6. Conclusion

In this paper, we raise the text effects transfer problem and propose a novel statistics-based method to solve it. We convert the high correlation between the sub-effects patterns and their relative spatial distribution to the text skeletons into soft constraints for text effects generation. An objective function with three complementary terms is proposed to jointly consider the local multi-scale texture, global sub-effects distribution and visual naturalness. The relationship of the text effects among multiple characters is further considered. We validate the effectiveness and robustness of our distribution-aware method by comparisons with state-of-the-art style transfer algorithms and extensive artistic typography generations.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cviu.2018.07.004.

## References


Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B., 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graphics 28 (3), 341–352.

Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A., 2010. The generalized patchmatch correspondence algorithm. Proc. European Conf. Computer Vision. pp. 29–43.

Barnes, C., Zhang, F.L., Lou, L., Wu, X., Hu, S.M., 2015. Patchtable: efficient patch queries for large datasets and applications. ACM Trans. Graphics 34 (4), 97.

Bénard, P., Cole, F., Kass, M., Mordatch, I., Hegarty, J., Senn, M.S., Fleischer, K., Pesare, D., Breeden, K., 2013. Stylizing animation by example. ACM Trans. Graphics 32 (4), 119.

Champandard, A. J., 2016. Semantic style transfer and turning two-bit doodles into fine artworksArvix preprint; https://arxiv.org/abs/1603.01768.

Chang, C.C., Lin, C.J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chang, Y., Saito, S., Nakajima, M., 2007. Example-based color transformation of image and video using basic color categories. IEEE Trans. Image Process. 16 (2), 329–336.

Chang, Y., Saito, S., Uchikawa, K., Nakajima, M., 2005. Example-based color stylization of images. ACM Trans. Appl. Percept. 2 (3), 322–345.

Cheng, L., Vishwanathan, S., Zhang, X., 2008. Consistent image analogies using semi-supervised learning. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition. pp. 1–8.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P., 2012. Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graphics 31 (4), 82:1–82:10.

Efros, A.A., Freeman, W.T., 2001. Image quilting for texture synthesis and transfer. Proc. ACM Conf. Computer Graphics and Interactive Techniques. pp. 341–346.

Efros, A.A., Leung, T.K., 1999. Texture synthesis by non-parametric sampling. Proc. IEEE Int'l Conf. Computer Vision.

Elad, M., Milanfar, P., 2016. Style-transfer via texture-synthesisArvix preprint; https://arxiv.org/abs/1609.03057.

Frigo, O., Sabater, N., Delon, J., Hellier, P., 2016. Split and match: example-based adaptive patch sampling for unsupervised style transfer. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Gatys, L.A., Ecker, A.S., Bethge, M., 2015. Texture synthesis using convolutional neural networks. Advances in Neural Information Processing Systems.

Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Goodfellow, I., Pougetabadie, J., Mirza, M., Xu, B., Wardefarley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Advances in Neural Information Processing Systems. pp. 2672–2680.

Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H., 2001. Image analogies. Proc. Conf. Computer Graphics and Interactive Techniques. pp. 327–340.

Johnson, J., Alahi, A., Li, F.F., 2016. Perceptual losses for real-time style transfer and super-resolution. Proc. European Conf. Computer Vision.

Julesz, B., Bergen, J.R., 1983. Textons, the fundamental elements in preattentive vision and perception of textures. Bell Labs Tech. J. 62 (6), 243–256.

Kwatra, V., Essa, I., Bobick, A., Kwatra, N., 2005. Texture optimization for example-based synthesis. ACM Trans. Graphics 24 (3), 795–802.

Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A., 2003. Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graphics 22 (3), 277–286.

Lantuéjoul, C., 1977. Sur le modèle de Johnson-Mehl généralisé. Technical Report. Technical report, Centre de.

Larsson, G., Maire, M., Shakhnarovich, G., 2016. Learning representations for automatic colorization. Proc. European Conf. Computer Vision.

Lezoray, O., Elmoataz, A., Bougleux, S., bastien, 2007. Graph regularization for color image processing. Comput. Vision Image Understanding 107 (12), 38–55.

Li, C., Wand, M., 2016. Combining markov random fields and convolutional neural networks for image synthesis. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Li, C., Wand, M., 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. Proc. European Conf. Computer Vision.

Liang, L., Liu, C., Xu, Y., Guo, B., Shum, H., 2001. Real-time texture synthesis by patch-based sampling. ACM Trans. Graphics 20 (3), 127–150.

Lin, T., Maji, S., 2016. Visualizing and understanding deep texture representations. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Lukáč, M., Fišer, J., Bazin, J.C., Jamriška, O., Sorkine-Hornung, A., Sýkora, D., 2013. Painting by feature: texture boundaries for example-based image creation. ACM Trans. Graphics 32 (4). 96–96

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. Comput. Sci. 2672–2680.

Okura, F., Vanhoey, K., Bousseau, A., Efros, A.A., Drettakis, G., 2015. Unifying color and texture transfer for predictive appearance manipulation. Comput. Graphics Forum 34 (4), 53–63.

Park, J., Tai, Y., Sinha, S.N., Kweon, I.S., 2016. Efficient and robust color consistency for community photo collections. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.

Pitié, F., Kokaram, A.C., Dahyot, R., 2007. Automated colour grading using colour distribution transfer. Comput. Vision Image Understanding 107 (1), 123–137.

Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P., 2001. Color transfer between images. IEEE Comput. Graphics Appl. 21 (5), 34–41.

Selim, A., Elgharib, M., Doyle, L., 2016. Painting style transfer for head portraits using convolutional neural networks. ACM Trans. Graphics 35 (4), 1–18.

Shih, Y., Paris, S., Barnes, C., Freeman, W.T., Durand, F., 2014. Style transfer for headshot portraits. ACM Trans. Graphics 33 (4), 1–14.
Shih, Y., Paris, S., Durand, F., Freeman, W.T., 2013. Data-driven hallucination of different times of day from a single outdoor photo. ACM Trans. Graphics 32 (6), 2504–2507.
Song, Y., Bao, L., He, S., Yang, Q., Yang, M.-H., 2017. Stylizing face images via multiple exemplars. Comput. Vision Image Understanding PP (99), 1–11.
Tai, Y.W., Jia, J., Tang, C.K., 2005. Local color transfer via probabilistic segmentation by expectation-maximization. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition. pp. 747–754.
Tai, Y.W., Jia, J., Tang, C.K., 2007. Soft color segmentation and its applications. IEEE Trans. Pattern Anal. Mach.Intell. 29 (9), 1520–1537.
Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V., 2016. Texture networks: feed-forward synthesis of textures and stylized images. Proc. IEEE Int'l Conf. Machine Learning.
Versteegen, R., Gimel'Farb, G., Riddle, P., 2016. Texture modelling with nested high-order Markov–Gibbs random fields. Comput. Vision Image Understanding 143, 120–134.
Wei, L.Y., Levoy, M., 2000. Fast texture synthesis using tree-structured vector quantization. Proc. ACM SIGGPRAH. pp. 479–488.
Welsh, T., Ashikhmin, M., Mueller, K., 2002. Transferring color to greyscale images. ACM Trans. Graphics 21 (3), 277–280.
Wexler, Y., Shechtman, E., Irani, M., 2007. Space-time completion of video. IEEE Trans. Pattern Anal. Mach.Intell. 29 (3), 463–476.
Xu, Y., Huang, S., Ji, H., 2012. Scale-space texture description on sift-like textons. Comput. Vision Image Understanding 116 (9), 999–1013.
Yan, Z., Zhang, H., Wang, B., Paris, S., Yu, Y., 2016. Automatic photo adjustment using deep neural networks. ACM Trans. Graphics 35 (1).
Yang, S., Liu, J., Lian, Z., Guo, Z., 2017. Awesome typography: statistics-based text effects transfer. Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition.
Zhang, R., Isola, P., Efros, A.A., 2016. Colorful image colorization. Proc. European Conf. Computer Vision.