Awesome Typography: Statistics-Based Text Effects Transfer

Shuai Yang, Jiaying Liu[*], Zhouhui Lian and Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing, China

Abstract

In this work, we explore the problem of generating fantastic special-effects for the typography. It is quite challenging due to the model diversities to illustrate varied text effects for different characters. To address this issue, our key idea is to exploit the analytics on the high regularity of the spatial distribution for text effects to guide the synthesis process. Specifically, we characterize the stylized patches by their normalized positions 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 fits for both local texture patterns and the global spatial distribution in the example. Experimental results demonstrate the superiority of our 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.

1. Introduction

Typography 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*, flowing *smokes*, multicolored *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 the editing software: 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 fantastic text effects designed by artists onto raw plain texts, as shown in Fig. 1.

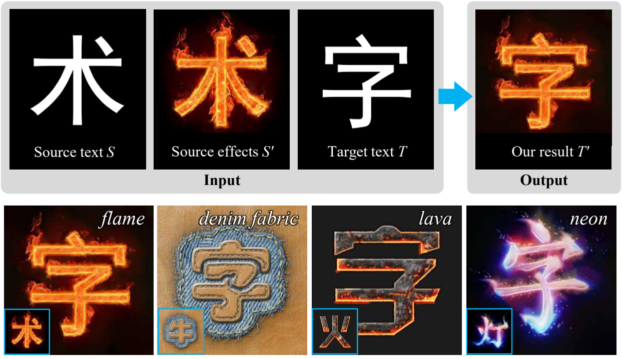


Figure 1. Overview: Our method takes as input the source text image S, its counterpart stylized image S' and the target text image T, then automatically generates the target stylized image T' with the special effects as in S'.

Text effects transfer is a brand new sub-topic of style transfer. Style transfer can be related to color and texture transfer, respectively. Color transfer matches global [27] or local [31] color distributions of the target and source images. Texture transfer relies on texture synthesis technologies, where the texture generation is constrained by guidance images. Meanwhile, texture synthesis can be divided into two categories: non-parametric methods [7, 6, 15, 35] and parametric methods [14, 9, 17, 11]. 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 three aspects: (i) 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. (ii) The complicated composition of style elements: A text effects image often contains multiple intertwined style elements (we call them *text sub-effects*) that have very dif-

[*] Corresponding author

This work was supported by National Natural Science Foundation of China under contract No. 61472011 and No. 61472015.

ferent textures and structures (see *denim fabric* example in Fig. 1) and need specialized treatments. (iii) 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. This makes the traditional non-parametric texture-by-numbers method [12] 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 [10, 17] 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 distance-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. All measurements are carefully designed to achieve certain robustness to the character shape. In addition, we further consider the psycho-visual factor to enhance image naturalness. In summary, our contributions are threefold:

- We raise a brand new topic of text effects transfer that turns plain texts into fantastic artworks, which enjoys wide application scenarios such as picture creation on social networks and commercial graphic design.
- We perform investigation and analysis on well-designed typography and summarize the key distance-based characteristics for high-quality text effects. 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.

2. Related Work

Color Transfer. Pioneering color transfer methods [27, 25] transfer color between images by matching their global color distributions. Subsequently, local color transfer is achieved based on segmentation [31, 32] or user interaction [34] and it is further improved using fine-grained patch [30] or pixel [29, 24] correspondences. Recently, color transfer [36] and colorization [16, 37] using deep neural networks have drawn people's attentions.

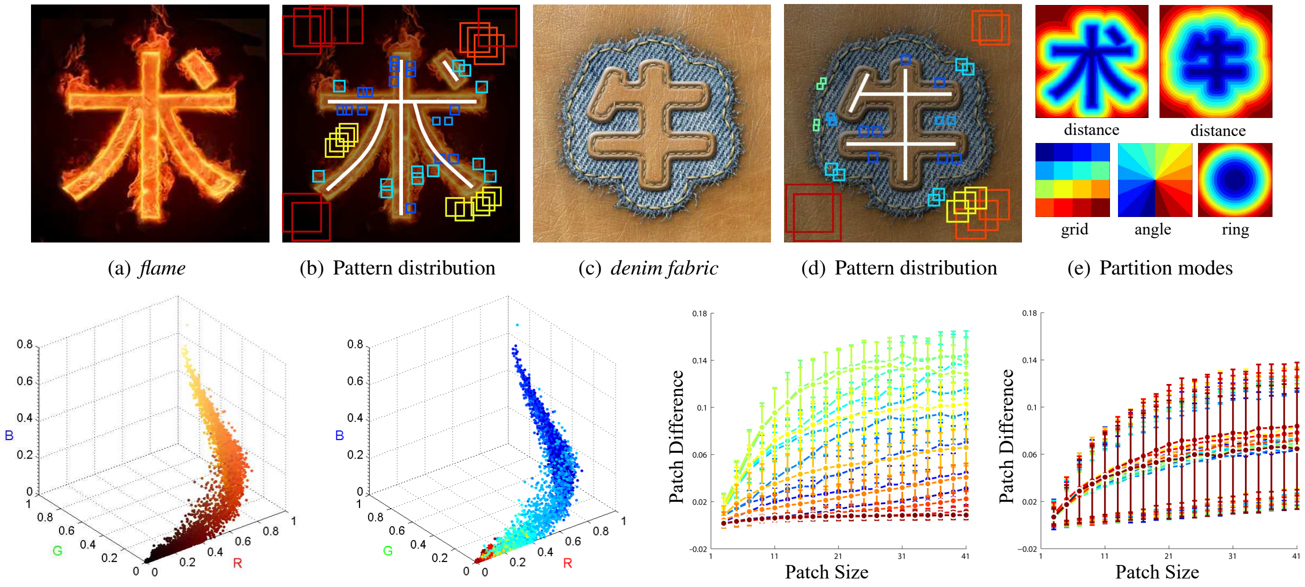
Non-Parametric Texture Synthesis and Transfer. Efros and Lueng [7] proposed a pioneering pixel-by-pixel synthesis approach based on sampling similar patches. The subsequent works improve it in quality and speed by synthesizing patches rather than pixels. To handle the overlapped regions of neighboring patches for seamlessness, Liang *et al.* [19] proposed to blend patches, and Efros and Freeman [6] used dynamic programming to find an optimal separatrix in overlapped regions, which is further improved via graph cut [15]. Unlike previous methods that synthesize textures in a local manner, recent techniques synthesize globally using objective functions. A coherence-based function [35] 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 extended to adapt to non-stationary textures through patch geometric and photometric transformations [2, 5].

Texture transfer, also known as Image Analogies [12], generates textures but also keeps the structure of the target image. Structures are usually preserved by reducing the differences between the source and target guidance maps [12, 23]. In [21], texture boundaries are synthesized in priority to constrain the structure. Frigo *et al.* [8] proposed an adaptive patch partition to precisely capture source textures and preserve target structures, followed by a Markov Random Field (MRF) function for global texture synthesis.

Parametric Texture Synthesis and Transfer. The idea of modelling textures using statistical measurements has led to the development of textons and its variants [14, 26]. Nowadays, deep-based texture synthesis [9] 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 [9] and adapt it to style transfer by incorporating content similarities [10]. This work presented the remarkable generic painting transfer technique and attracted many follow-ups in loss function improvement [20, 28] and algorithm acceleration [13, 33]. Recently, methods that replace the Gram-matrix by MRF regularizer is proposed for photographic synthesis [17] and semantic texture transfer [3]. Meanwhile, Generative Adversarial Networks (GANs) [11] 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 [22], fulfils the challenging task of generating images from abstract semantic labels. Li and Wand [18] further showed that their Markovian GANs has certain advantages over the Gram-matrix-based methods [10, 33] in coherent texture preservation.

3. Proposed Method

In this section, we first formulate our text effects transfer problem. Visual analytic is then presented on our observa-



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

(f) Pixels in RGB space (g) Pixel distance distribution (h) Response curves (distance mode) (i) Response curves (random mode)

Figure 2. Statistics of the text effects images. (a)(c) *flame* and *denim fabric* 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. (f)-(g) High correlation between pixel colors and distances: Pixels are distinguished from each other by their distances in RGB space. (h)-(i) High correlation between patch scales and distances: Patches with similar distances have uniform responses to changes of their size.

tion of the high correlation between patch patterns (*i.e. color* and *scale*) and their spatial distributions in text effects images (Sec. 3.1). Based on this observation, we extract text effects statistics from the source images (Sec. 3.2) and employ it to adapt the texture synthesis algorithm for high-quality text effects transfer (Sec. 3.3).

3.1. Problem Formulation and Analysis

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 $S : S' :: T : T'$. 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'.

To transfer arbitrary text effects automatically, we have to face a variety of text effects, the complex composition of text sub-effects and the simpleness of guidance maps. Thus, it is a quite challenging task. How to synthesize preferable text effects is still an open question. That is, (i) what is the essential factor to make text effects preferable? and (ii) how can we capture these characteristics for synthesis?

Fortunately, after mathematically analyzing dozens of text effects created by designers, we find the clue: *the high correlation between patch patterns (i.e. color and scale) and their distances to text skeletons*. We note that textures with similar distances to the text skeleton tend to share similar patterns. It is schematically illustrated in Figs. 2(b)(d) where the patches with the same distance to the skeleton (in white) are marked by the same color. This observation is consistent with the texture adaptation based on text shapes for readability conducted by designers in the real world.

To quantitatively verify this correlation, we divide pixels/patches in text effects images into N classes based on their distance to the text skeleton (distance calculation will be given in Sec. 3.2.2) and use the differentiation of N partitions as the measurement. Two examples of the distance-based partition are shown in the top row of Fig. 2(e), where we exploit different colors to denote each class.

For the pixel color, taking *flame* image for example, by marking each point in RGB space (Fig. 2(f)) with its class-color, we note that the points in Fig. 2(g) with the same class-color appear in the neighborhood, which means pixel colors have strong correlation with their distance values. Thus, we quantify it as the classification accuracy,

$$corr(\text{color}, \text{dist}) = 1 - \epsilon, \tag{1}$$

where ϵ is the training error obtained by training SVM [4] to classify pixel colors given our distance-based partition. The mean correlation between pixel colors and the distance values on 30 text effects images created by designers is 0.147. We also provide other three partition modes (as shown in the bottom row of Fig. 2(e)) as well as the random partition mode for comparison in the experiments. Their mean cor-

Table 1. Correlations between patch patterns and different modes.

corr	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**

relations with pixel colors over 30 test images are shown in the second row of Table 1. As expected, the distance is the most important factor to depict pixel colors.

For the patch scale, we enumerate patch sizes and calculate the difference between each patch and its best match, which forms a response curve at different scaling. Taking *denim fabric* image for example, Figs. 2(h) and (i) show N response curves of patch sizes in distance and random modes, respectively. Each point on the curve gives the mean and standard deviation of patch differences at a certain patch size in the corresponding class. We find that: (i) Response curves in distance mode are more diverse (high inter-curve standard deviations σ_{inter}), which means different sub-effects are well distinguished by their distance values. (ii) Points on response curves in distance mode have lower standard deviations (low intra-curve standard deviations σ_{intra}), which means patches with similar distances react uniformly to scale changing and possibly share common optimal scale for description. Considering these two aspects, we evaluate the correlation with the patch scale by,

$$corr(\text{scale}, \text{dist}) = \sigma_{\text{inter}}/\sigma_{\text{intra}}. \qquad (2)$$

Here σ_{inter} and σ_{intra} are defined as sum of standard deviations σ with different patch sizes. Given a patch size, σ is standard deviation of mean patch differences of N points in different curves for σ_{inter} and is mean of standard deviations of N points for σ_{intra}. The bottom row of Table 1 shows the mean correlations between the patch scale and different modes on 30 images where distance owns the highest value.

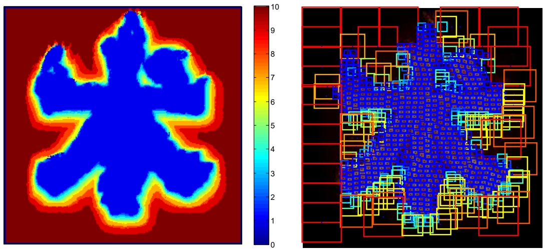
Based on the results for pixel colors and patch scales, we obtain the conclusion that the high correlation between patch patterns and their distances is reasonable essential characteristics for high-quality text effects.

3.2. Text Effects Statistics Estimation

We now convert the aforementioned analysis into patch statistics that can be directly used as the transfer guidance. Specifically, we detect the optimal scales for source patches, and estimate their normalized distances to the text skeleton. Then we are able to derive the posterior probability of the optimal scale for each patch based on its spatial position.

3.2.1 Optimal Patch Scale Detection

Inspired by [8], we propose a simple yet effective approach to detect the optimal patch scale $\text{scal}(q)$ to depict texture patterns round q. Given a predefined downsample factor s, we start from the max (roughest) scale L to filter source patches and let the screened patches pass to a finer scale.



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

Figure 3. Detected optimal patch scales for the *flame* image.

We use a fixed patch size of $m \times m$ and resize the image to accomplish multiple scales. Let S_ℓ 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_ℓ. S'_ℓ and $Q'_\ell(q)$ are similarly defined. If $\hat{q}$ is the correspondence of q at scale ℓ such that

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

then our filter criterion at scale ℓ is

$$\zeta_\ell(q, \hat{q}) = \big(\sigma_\ell + \sqrt{d_\ell(q, \hat{q})} > \omega\big), \qquad (4)$$

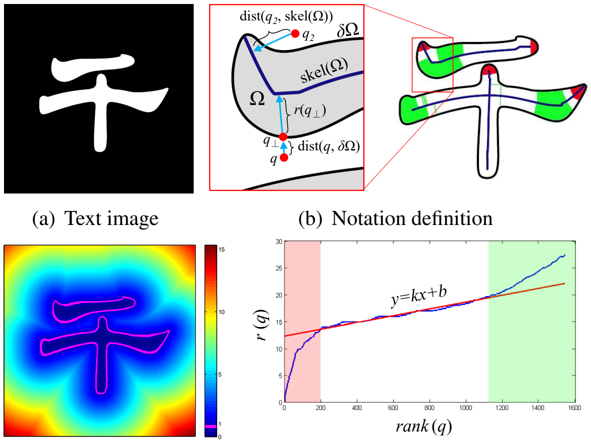
where $\sigma_\ell = \sqrt{Var(Q'_\ell(q))}/2$. Patches that satisfy the filter criterion pass through to finer scale $\ell - 1$, while the filter residues set ℓ as their optimal scales. An example of the optimal scales for the *flame* image is shown in Fig. 3(a). It is found that the textured region near the character requires finer patch scales than the outer flat region. For better visualization, we show the optimal scale of the patch $Q(q)$ by resizing it at a scale rate of $s^{\text{scal}(q)-1}$ in Fig. 3(b).

3.2.2 Robust Normalized Distance Estimation

Here we first define some concepts. In the text image, its text region is denoted by Ω. The skeleton $\text{skel}(\Omega)$ is a kernel path within Ω. We use $\text{dist}(q, A)$ to denote the distance between q and its nearest pixel in set A. We are going to calculate $\text{dist}(q, \text{skel}(\Omega))$. For q on the text contour $\delta\Omega$, the distance is also known as the text width or radius $r(q)$. Fig. 4(b) gives the visual interpretation.

We extract $\text{skel}(\Omega)$ from S using morphology operations. To ensure the distance invariant to the text width, we aim to normalize the distance so that the normalized text width equals to 1. Simply dividing the distance by the text width is unreliable because the inaccurate of the obtained $\text{skel}(\Omega)$ leads to errors both in the numerator and denominator as well. To address this issue, we estimate corrected text width $\widetilde{r}(q)$ based on statistics and use the accurate $\text{dist}(q, \delta\Omega)$ 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 modelled by linear regression, as shown in Figs. 4(d). From Figs. 4(b)(d), we discover that



(a) Text image (b) Notation definition

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

Figure 4. 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 width.

outliers assemble at small values. We empirically assume the leftmost 20% points are outliers and eliminate them by

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

where k, b are linear regression coefficients, $|\delta\Omega|$ is the pixel number of $\delta\Omega$. Finally, the normalized distance is obtained,

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

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

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

3.2.3 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:

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

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 ℓ 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 assume the target images share the same posterior probability with the source image. And we will use this probability to select patch scales statistically for texture synthesis to adapt extremely various text effects.

3.3. Text Effect 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.* [35] and its variants [5] using random search and propagation as in PatchMatch [1, 2]. We refer to these papers for 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 (Sec. 3.3.2). Then a distribution term is introduced to adjust the spatial distribution of the text sub-effects (Sec. 3.3.3). Finally, we propose a psycho-visual term that prevents texture over-repetitiveness for naturalness (Sec. 3.3.4).

3.3.1 Objective Function

We augment the texture synthesis objective function in [35] 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_{app}, E_{dist} and E_{psy} are the appearance, distribution and psycho-visual terms, respectively, which are weighted by λ_1 and λ_2 to together make up the patch distance.

3.3.2 Appearance Term: Texture Style Transfer

The original texture synthesis algorithm of Wexler *et al.* [35] minimizes the Sum of the Squared Differences (SSD) of two patches sampled from texture image pair S'/T'. We adapt it to texture transfer tasks by applying additional SSD of two patches sampled from the text image pair S/T:

$$E_{\text{app}}(p, q) = \lambda_3 \|P(p) - Q(q)\|^2 + \|P'(p) - Q'(q)\|^2, \tag{11}$$

where λ_3 is a weight that compromises between the color difference and character 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,

$$\begin{aligned}E_{\text{app}}(p,q) =&\lambda_3 \sum_{\ell} \mathcal{P}(\ell|\text{bin}(p))\|P_\ell(p) - Q_\ell(q)\|^2 \\ &+ \sum_{\ell} \mathcal{P}(\ell|\text{bin}(p))\|P'_\ell(p) - Q'_\ell(q)\|^2.\end{aligned} \tag{12}$$

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

3.3.3 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,

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

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 $\max(1, \text{dist}^2(p))$.

3.3.4 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 find q as their correspondence and $|\Phi(q)|$ be the size of the set. We define the psycho-visual term as,

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

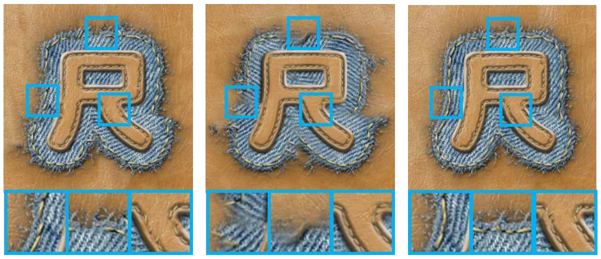
From the perspective of q, we can better understand this repetitiveness penalty:

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

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.

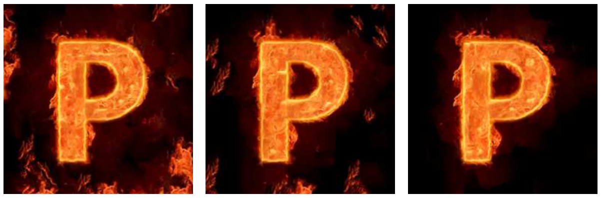
3.3.5 Function Optimization

We follow the iterative coarse-to-fine matching and voting steps as in [35]. In the matching step, PatchMatch algorithm [1, 2] is adopted. We update $\Phi(q)$ after each iteration of search and propagation 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. 6.



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

Figure 5. Effects of the multi-scale strategy. (a) Results using single-scale 5×5 patches. (b) Results using single-scale 15×15 patches. (c) Results using joint 5×5 patches over 5 scales.



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

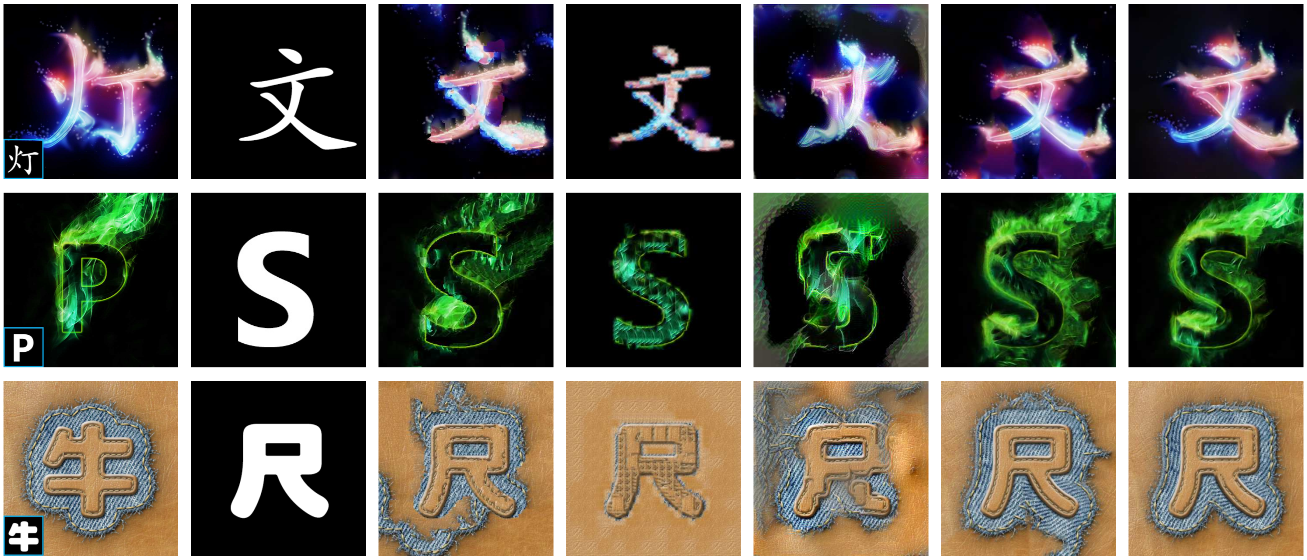
Figure 6. Effects of the distribution term. (a) Results without distribution term. (b) Results obtained by random initialization and optimization with distribution term. (c) Results obtained by both initialization and optimization with distribution term.

4. Analysis

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. 5(a) and (b) the *denim fabric* style generated using single-scale 5×5 and 15×15 patches, respectively. Small patches capture very limited contextual information, thus it cannot guarantee the structure continuity. As can be seen in Fig. 5(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. 5(b). These problems are well solved by jointly using 5×5 patches over 5 scales as in Fig. 5(c), where the overall shape is well preserved and the details like sewing threads look more vivid.

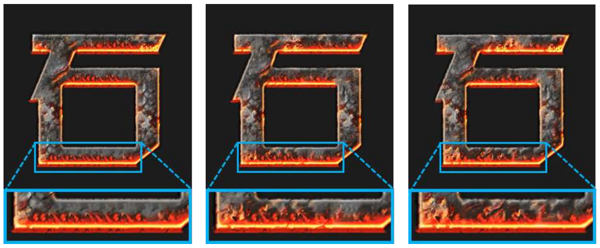
Distribution Term. The distribution term ensures the sub-effects in the target image and the source example are similarly distributed, which is the basis of our assumption in Sec. 3.2.3 that the posterior probabilities $\mathcal{P}(\ell|x)$ in T' and S' are the same. Fig. 6 shows the effects of the distribution 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.

Psycho-Visual Term. The effects of our psycho-visual term are shown in Fig. 7. The *lava* textures synthesized without the psycho-visual penalty (Fig. 7(a)) densely repeat the red cracks in several regions, which causes obvious unnaturalness. By increasing the penalty, the reuse



(a) (S, S') (b) T (c) Analogy [12] (d) Split & match [8] (e) Neural doodle [3] (f) Baseline (g) Proposed method

Figure 8. Comparison with state-of-the-art methods on various text effects. From top to bottom: *neon*, *smoke*, *denim fabric*. (a) Input source text effects with their raw text counterparts in the lower-left corner. (b) Target text. (c) Results of Image Analogies [12]. (d) Results of Split and Match [8]. (e) Results of Neural Doodles [3]. (f) Results of our baseline method. (g) Results of the proposed method.



(a) $\lambda_2 = 0.0$ (b) $\lambda_2 = 0.005$ (c) $\lambda_2 = 0.01$

Figure 7. Effects of the psycho-visual term, which penalizes texture over-repetitiveness and encourages new texture creation.

of the same source textures is greatly restrained (Fig. 7(b)) and our method tends to agilely combine different source patches to create brand-new textures (Fig. 7(c)). Thus, the psycho-visual term can effectively penalize texture over-repetitiveness and encourage new texture creation.

Combination of the Three Term. It is worth noting that *the proposed three terms are complementary*: First, the appearance and distribution terms emphasize local texture patterns and global text sub-effects distributions, respectively. The former depicts low-level color features while the latter exploits complementary mid-level position features. Second, the appearance and distribution terms jointly evaluate objective patch similarities. Meanwhile, the psycho-visual term complements these two terms by incorporating aesthetic subjective evaluations.

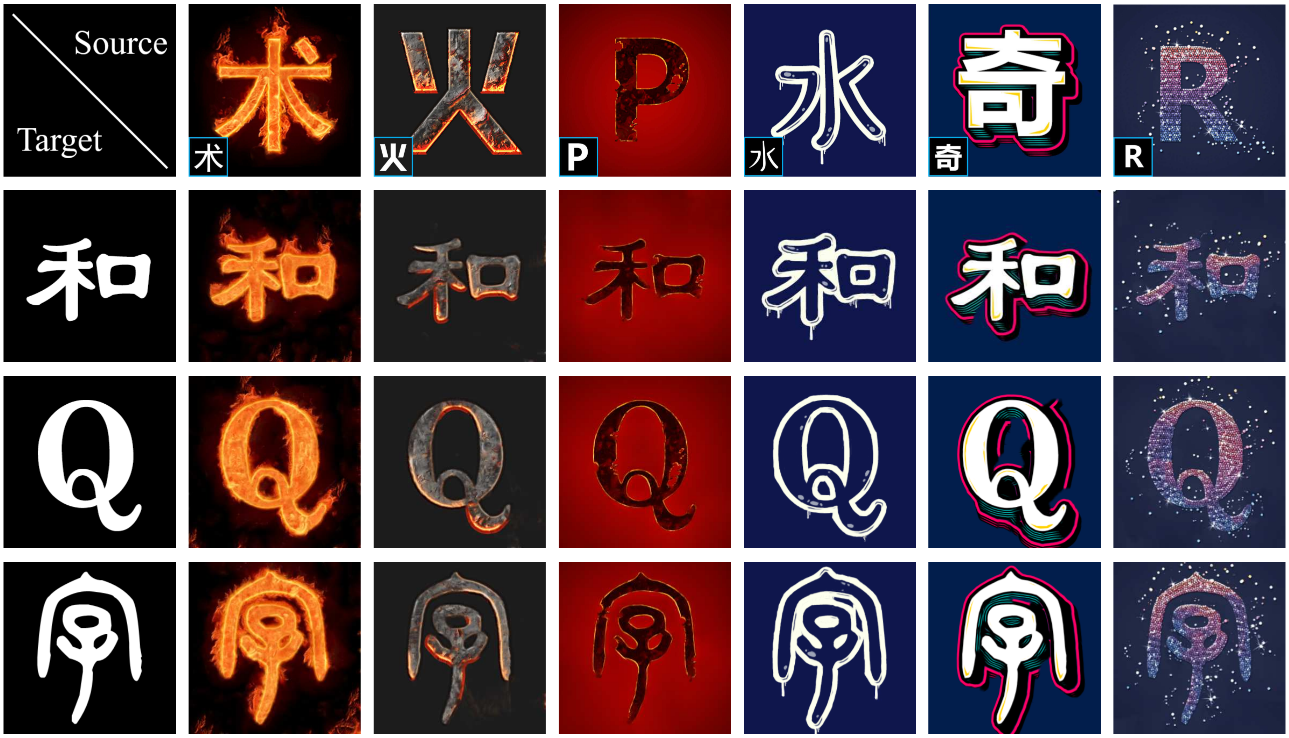
5. Experimental Results

In the experiment, the patch size is 5×5 and the max scale $L = 5$. We build an image pyramid of 10 levels with a fixed coarsest size (32×32). At level ℓ, joint patches over scales from ℓ to $min(10, \ell + L - 1)$ are used. The weights λ_1, λ_2 and λ_3 to balance different terms are set to 0.01, 0.005 and 10, respectively. The parameter ω for the filter criterion is 0.3 in our implementation[1].

In Fig. 8, 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 Image Analogies [12]. The textures in their results repeat locally and look disordered globally with evident patch boundaries. The second method is our implementation of Split and Match [8], 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, Neural Doodle [3], is based on the combination of MRF and CNN [17] 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. Our baseline transfers fine textures but fails to keep the overall sub-effects distribution and generates artifacts in the background. By comparison, the proposed method outperforms state-of-the-art methods, preserving both local textures and the global sub-effects distribution.

In Fig. 9, we present an illustration of style transfer from

[1] We release our Matlab implementation at: https://github.com/williamyang1991/Text-Effects-Transfer



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

Figure 9. Apply different text effects to representative characters (Chinese, alphabetic, handwriting).

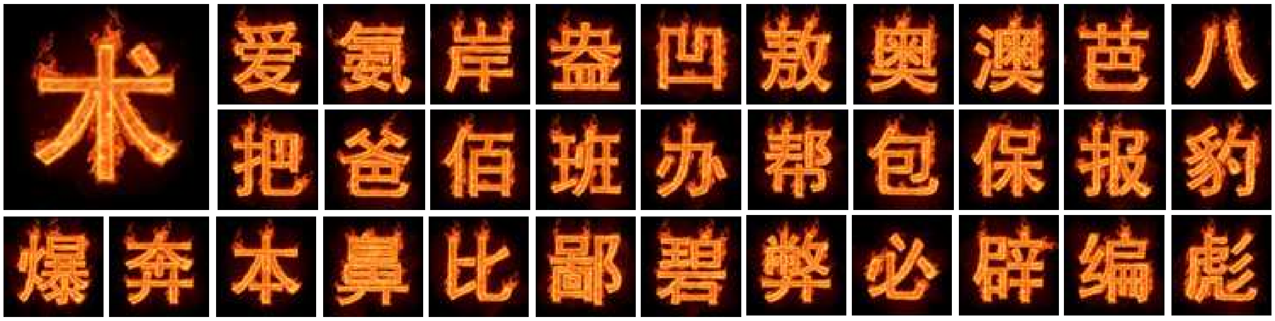

术
爱 氨 岸 盎 凹 敖 奥 澳 芭 八
把 爸 佰 班 办 帮 包 保 报 豹
爆 奔 本 鼻 比 鄙 碧 弊 必 辟 编 彪

Figure 10. 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.

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.

Finally, 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. 10. 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.

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. We validate the effectiveness and robustness of our method by comparisons with state-of-the-art style transfer algorithms and extensive artistic typography generations. Future work will concentrate on the composition of the stylized texts and the background photos.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):341–352, August 2009. 5, 6

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *Proc. European Conf. Computer Vision*, pages 29–43, 2010. 2, 5, 6

[3] A. J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. 2016. Arvix preprint; https://arxiv.org/abs/1603.01768. 2, 7

[4] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 3

[5] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, 31(4):82:1–82:10, July 2012. 2, 5

[6] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proc. ACM Conf. Computer Graphics and Interactive Techniques*, pages 341–346, 2001. 1, 2

[7] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. IEEE Int'l Conf. Computer Vision*, 1999. 1, 2

[8] O. Frigo, N. Sabater, J. Delon, and P. Hellier. Split and match: example-based adaptive patch sampling for unsupervised style transfer. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016. 2, 4, 7

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2015. 1, 2

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016. 2

[11] I. Goodfellow, J. Pougetabadie, M. Mirza, B. Xu, D. Wardefarley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 2

[12] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. Conf. Computer Graphics and Interactive Techniques*, pages 327–340, 2001. 2, 7

[13] J. Johnson, A. Alahi, and F. F. Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conf. Computer Vision*, 2016. 2

[14] B. Julesz and J. R. Bergen. Textons, the fundamental elements in preattentive vision and perception of textures. *Bell Labs Technical Journal*, 62(6):243–256, 1983. 1, 2

[15] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3):277–286, 2003. 1, 2

[16] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proc. European Conf. Computer Vision*, 2016. 2

[17] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016. 1, 2, 7

[18] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. European Conf. Computer Vision*, 2016. 2

[19] L. Liang, C. Liu, Y. Xu, B. Guo, and H. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, 2001. 2

[20] T. Lin and S. Maji. Visualizing and understanding deep texture representations. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016. 2

[21] M. Lukáč, J. Fišer, J. C. Bazin, O. Jamriška, A. Sorkine-Hornung, and D. Sýkora. Painting by feature: texture boundaries for example-based image creation. *ACM Transactions on Graphics*, 32(4):96–96, 2013. 2

[22] M. Mirza and S. Osindero. Conditional generative adversarial nets. *Computer Science*, pages 2672–2680, 2014. 2

[23] F. Okura, K. Vanhoey, A. Bousseau, A. A. Efros, and G. Drettakis. Unifying color and texture transfer for predictive appearance manipulation. *Computer Graphics Forum*, 34(4):53–63, 2015. 2

[24] J. Park, Y. Tai, S. N. Sinha, and I. S. Kweon. Efficient and robust color consistency for community photo collections. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2016. 2

[25] F. Pitié, A. C. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1):123–137, 2007. 2

[26] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision*, 40(1):49–70, 2000. 2

[27] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. 1, 2

[28] A. Selim, M. Elgharib, and L. Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics*, 35(4):1–18, 2016. 2

[29] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4):1–14, 2014. 2

[30] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics*, 32(6):2504–2507, 2013. 2

[31] Y. W. Tai, J. Jia, and C. K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 747–754, 2005. 1, 2

[32] Y. W. Tai, J. Jia, and C. K. Tang. Soft color segmentation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1520–1537, 2007. 2

[33] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *Proc. IEEE Int'l Conf. Machine Learning*, 2016. 2

[34] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Transactions on Graphics*, 21(3):277–280, 2002. 2

[35] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, March 2007. 1, 2, 5, 6

[36] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics*, 35(1), 2016. 2

[37] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. European Conf. Computer Vision*, 2016. 2