A Common Framework for Interactive Texture Transfer

Yifang Men, Zhouhui Lian*, Yingmin Tang, Jianguo Xiao
Institute of Computer Science and Technology, Peking University, China

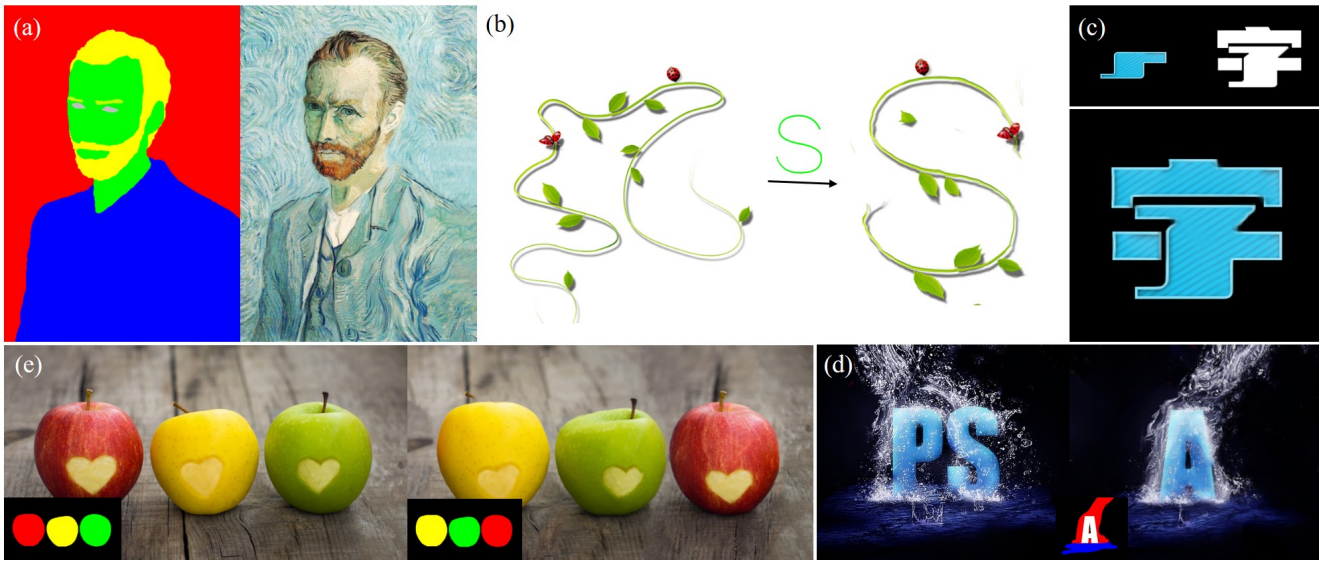


Figure 1: Representative results generated using our interactive structure-guided texture transfer framework. The stylized images are synthesized with the guidance of corresponding user-specified semantic maps. The proposed common framework is capable of multiple challenging user-controlled texture transfer tasks: (a) turning doodles into artworks, (b) editing decorative patterns, (c) generating texts in special effect, (d) controlling effect distribution in text images, (e) swapping textures. *Source image credits: (a) Van Gogh; (b,c,d) Zcool [1]; (e) Luan* et al. *[35]*

Abstract

In this paper, we present a general-purpose solution to interactive texture transfer problems that better preserves both local structure and visual richness. It is challenging due to the diversity of tasks and the simplicity of required user guidance. The core idea of our common framework is to use multiple custom channels to dynamically guide the synthesis process. For interactivity, users can control the spatial distribution of stylized textures via semantic channels. The structure guidance, acquired by two stages of automatic extraction and propagation of structure information, provides a prior for initialization and preserves the salient structure by searching the nearest neighbor fields (NNF) with structure coherence. Meanwhile, texture coherence is also exploited to maintain similar style with the source image. In addition, we leverage an improved PatchMatch with extended NNF and matrix operations to obtain transformable source patches with richer geometric information at high speed. We demonstrate the effectiveness and superiority of our method on a variety of scenes through extensive comparisons with state-of-the-art algorithms.

1. Introduction

Texture transfer is a classic problem in areas of Computer Vision and Computer Graphics. With this technique, we can automatically transfer the stylized texture from a given sample to the target image. A number of algorithms capable of creating impressive stylization effects have been reported over these years. Champandard *et al*. [8] proposed Neural Doodle, a technique turning doodles painted by users into fine artworks with provided samples. DecoBrush [33], an extension of Realbrush [32] and Helpinghand [34] allows designers to draw structured decorative patterns simply with selected styles. More recently, Yang *et al*. [43] achieved text effect transfer which enables to migrate the effect from a

* Corresponding author. E-mail: lianzhouhui@pku.edu.cn

stylized text image to a plain text image. However, existing approaches seem to be isolated from each other due to specific usage scenarios. In fact, they share a common notion of transferring textures under user guidance, namely, users should be able to transfer the texture from source to anywhere in target as they want.

The aim of this paper is to establish a general framework of user-guided texture transfer for multiple tasks, including turning doodles into artworks, editing decorative patterns, generating texts in special effect as well as controlling effect distribution in text images, and swapping textures (see Figure 1).

Due to the diversity of tasks and the simplicity of user guidance, it is challenging to achieve the goal mentioned above using existing methods. Some approaches [33, 43] perform well but they are tailored to specific domains. Hertzmann *et al*. [21] proposed a more general solution called Image Analogy. However, due to the lack of enough guidance of structural distribution, it suffers inner texture dislocation and fails to preserve local high-frequency structures. Painting by Feature [36] allows users to utilize the line and contour to guide texture transfer. It presents an improvement by treating line feature and area feature using brush tool and fill tool separately. Yet, the method is more suitable for filling nearly-stationary textures, since it does not provide the directional control for internal texture generation. The neural doodle reported in [8] using convolutional neural networks fails to reproduce clear and high-quality images with low-level texture details. The recently proposed Deep Image Analogy [31] produces particularly compelling results via a combination of image analogy [21] and neural networks [27, 41]. While when we feed a doodle image to the network, since the semantic labels have very low neural activation it is difficult to establish correspondence in textureless regions and thus unable to generate satisfactory synthesis results.

In this paper, we propose a common framework for user-guided texture transfer that is able to handle various challenging tasks. Interactive structure-based image synthesis is guided by both semantic map and structure information. Semantic channels are annotated by the user who can control the spatial distribution of stylized textures in the target image. The structure channels are then extracted automatically by content-aware saliency detection and propagated from the source style image to the target as a prior. Specifically, the propagation step acquires inner structure correspondences via the registration of key contour points between the source and target images. Combining semantic and structure information for dynamic guidance enables the transfer process to produce high-quality textures with content-awareness and low-level details. In addition, an improved PatchMatch algorithm with the extended nearest neighbor fields and matrix operations is adopted to provide richer source patches without speed reduction. Major contributions of this paper can be summarized as follows:

- We design a general framework to handle interactive texture transfer issues with the challenge of task diversity and guided-map simplicity, and show the effectiveness of our framework in multiple tasks.
- We propose a method that extracts salient structure regions and conveys structure information in the source image to the target. The structure information is then utilized as a prior to guide better synthesis procedure.
- We present some novel scenarios of user-controlled texture transfer in which, by incorporating the improved texture synthesis method, finer detailed synthesis images can be generated with higher speed.

2. Related Work

Up to now, a number of texture transfer methods have been proposed, which can be roughly categorized as classic texture transfer or neural-based techniques. Here, we briefly review some representative works.

2.1. Classic Texture Transfer

Classic texture transfer method is a variant of texture synthesis with given texture examples. For instance, most early transfer algorithms as pioneered by Efros and Freeman [13] are based on example-based texture synthesis methods [14, 2]. They utilized a correspondence map with some corresponding quantities such as intensity to constrain the synthesis process. A later work by Criminisi *et al*. [10] used patch priorities for region-filling to preserve the structure. Komodakis *et al*. [26] adopted Belief Propagation as the optimization scheme to avoid greedy patch assignments.

Optimization-based texture transfer technique, firstly proposed by Kwatra *et al*. [28], is also a follow-up work of example-based method. This technique develops to a successful texture synthesis method due to its high visual quality outcome and wide application in different scenes. Kwatra *et al*. [28] regarded texture synthesis problem as a global optimization task and used Expectation Maximization (EM)-like algorithm to iteratively minimize the energy function. Wexler *et al*. [42] alleviated the completion issue with multi-level synthesis to avoid being stuck in local minima. Barnes *et al*. [3, 4] introduced a PatchMatch algorithm to accelerate the nearest-neighbor search process leveraging random search and the natural coherence in the image. The optimization-based method was extended to image melding [11], stylized 3D renderings [15], and text effects transfer [43] using adaptive patch partitions [16]. However, these methods fail to synthesize the texture with salient structure and are prone to wash-out effect caused by

overusing low-frequency texture [38]. Our method shares the common baseline with these techniques and overcomes the challenges using multi-channel dynamic guidance.

Analogy-based method is another alternative for texture transfer. Image Analogy, originally proposed in [21], utilizes the availability of the input exemplar pair (source image A and stylized result A') to acquire the stylized image B' of target image B. The method finds the best correspondence in source image for each pixel in target image. Cheng *et al*. [9] improved this method with semi-supervised learning and image quilting model, which aims to ensure both local and global consistency. This approach has also been extended to solve animation stylization problems [20, 6] and construct efficient queries for large datasets [5]. Unfortunately, it does not provide a directional control and easily results in inner texture dislocation, which leads to the lost of structure information.

2.2. Neural-based Style Transfer

Gatys *et al*. [17] proposed a neural style transfer method leveraging pre-trained deep convolutional networks such as VGG-19 [41]. Their method is effective for stylizing the context image with a given style image, due to the ability of decomposing and recombining the content and style of images. Johnson *et al*. [25] later utilized perceptual loss functions to train feed-forward networks for real-time texture transfer tasks. Li and Wand [29] combined the Markov Random Fields model with deep neural networks, which was later extended to semantic style transfer [8]. Despite the great success of neural-based method, it is not suitable for our scenarios where source images are not limited to artistic works, photographs and photorealism images are also included. For those kinds of data, results of neural-based methods often contain many low-level noises. Moreover, no intuitive way is provided to control the synthesis process and thus results become unpredictable.

The recently-proposed Generative Adversarial Networks (GANs) [19, 12, 39] provided a potential alternative to generate texture via an adversarial process. GANs train a discriminator to distinguish whether the output is real or fake and a generator is trained simultaneously to fool the discriminator. More recently, image-to-image translation [24] was proposed using 'U-Net'-based architecture [40] for generator and convolutional 'PatchGAN' classifier [30] for discriminator. It is a general framework for translating an input image into the corresponding output image, such as turning semantic labels, edges, or segments into realistic images. Although this technique produces impressive results, it requires collecting thousands of related images to train a model for a specific category. On the contrary, our method only needs one exemplar for generating the target stylized image from a corresponding semantic map.

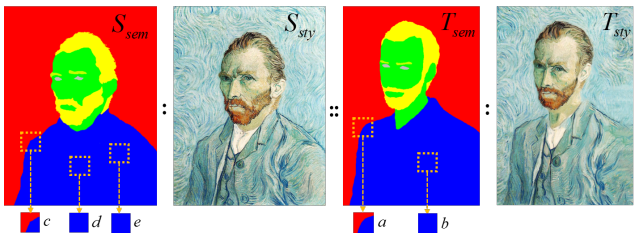


Figure 2: Overview of the interactive texture transfer problem. With three input images S_{sem} (semantic map of source image), S_{sty} (stylized source image aligned to S_{sem}) and T_{sem} (semantic map of target image), stylized target image T_{sty} with the style in S_{sty} can be generated.

3. Method Description

Interactive texture transfer aims to generate the stylized target image from a given source image with user guidance. Users can control the shape, scale and spatial distribution of the objects to be synthesized in the target image via semantic maps. With three input images S_{sem} (semantic map of source image), S_{sty} (stylized source image aligned to S_{sem}) and T_{sem} (semantic map of target image), the stylized target image T_{sty} could be automatically synthesized such that $S_{sem} : S_{sty} :: T_{sem} : T_{sty}$ (see Figure 2 for an overview).

Reproducing a structural image with stylized textures by using a semantic map that contains little information is a challenging task. In our method, we search the best correspondences between the source and target in a patch-wise manner. From the semantic map shown in Figure 2 we can see that patches in the boundary of color labels contain more abundant features than those located in internal positions. For patches in T_{sem}, patch a can find its best correspondence (patch c) more easily than patch b that is hard to choose its best-suited partner among internal source patches, such as patch d and e, which are completely identical (both full-blue) in the semantic map. Thus, it is difficult for internal patches with salient structure to be correctly synthesized by only relying on semantics. To solve the problem, we introduce a structure guidance based on the shape similarity of semantic labels.

The basic idea is that boundary patches are forward to be synthesized roughly correctly with more characteristics in the semantic map, then we find the best correspondences of inner patches mainly based on the structure guidance and coherence with the source stylization. Actually, once the boundary patches have been correctly synthesized, this interactive texture transfer problem could almost be degenerated into an image completion task [42, 23, 22] with a large hole to be filled via boundary propagation. We have tried many state-of-the-art inpainting methods [11, 23] but our experimental results (see supplementary materials) show that all of them fail to synthesize structural textures with such a large hole. One major reason is that a patch in the in-

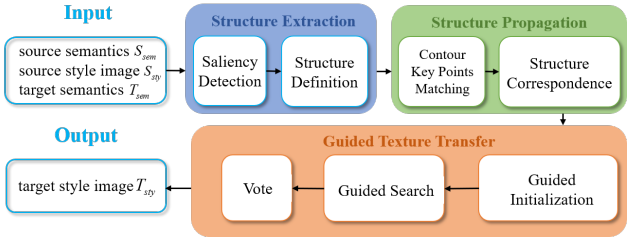


Figure 3: The pipeline of our framework.

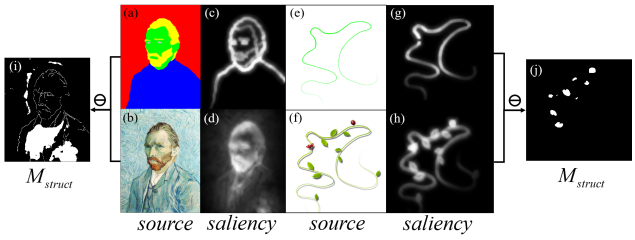


Figure 4: Illustration of structure information extraction. The structure mask (i) or (j) is acquired by the computation of saliency maps (c), (d) or (g), (h).

ternal region receives conflict information propagated from four directions due to the difference in shapes. We alleviate this issue by using structure information to provide a prior in the initialization stage and guide the synthesis process.

As shown in Figure 3, three main steps constitute our pipeline including salient internal structure extraction, structure propagation and guided texture transfer.

3.1. Internal Salient Structure Extraction

Some salient texture details in the internal region of a source semantic map are prone to being lost or suffering disorder in the synthesized target image. This step aims to extract detailed structure information within the semantic map for the following propagation and synthesis.

Saliency Detection. Saliency detection is performed to mark the salient regions of the source stylized image, which contain complex textural structure or curvilinear structure such as an edge or contour in an image. Goferman *et al*. [18] proposed a saliency detection with content-awareness. Following their method, we compute a saliency map for the source semantic map as M_{sem} and the other one for the source stylized image as M_{sty}.

Structure Definition. There exist some structural textures which are easily lost in the target because they contain salient structure information in the source stylized image but are not marked in the semantic map. These structural patches mainly locate in the internal region of the semantic map, such as the cloth region below the neck in Figure 4 (b) and leaves in Figure 4 (f). In this paper, we define these salient internal textures as structure and the structure mask

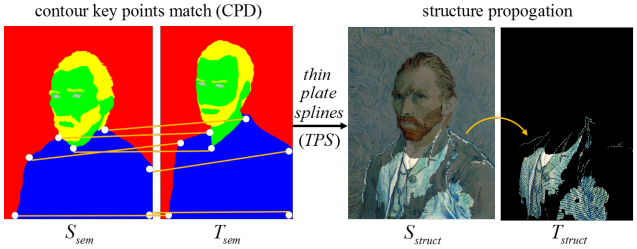


Figure 5: Overview of our structure propagation process. Structure information in a target image is obtained by finding a planar transformation, which enables to project the structural pixels in S_{struct} to T_{struct}. The transformation is computed with the TPS algorithm based on contour key point matching and the correspondence of contour points established via the CPD method.

as a binary image which can be computed by

$$M_{struct}(p) = \begin{cases} 1, & M_{sty}(p) - \ell M_{sem}(p) > \delta \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where p is the pixel in saliency map M_{sty} and M_{sem}, we set ℓ as 10 for a sharp saliency decrease of boundary pixels. δ is a threshold to discriminate the structure information. Figure 4 shows two examples of extracted structure regions.

The structure map of source image is acquired by the multiplication of colocated elements in matrix

$$S_{struct} = S_{sty} \circ M_{struct}. \quad (2)$$

3.2. Structure Propagation

After extracting internal structure of source labels, as shown in Figure 5, the structure information is propagated from source to target to guide the texture transfer process.

Matching Contour Key Points. With only the semantic map given for a target image, we want to propagate the structure from source to target via a planar transformation $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, by which each structural pixel in S_{struct} is projected to T_{struct}. We compute the planar transformation ψ by using key points on the contour, which are more reliable to represent the shape features. To establish correspondence between two sets of points, we use the coherent point drift (CPD) [37], a powerful point set registration algorithm to match each target contour point $\widehat{cp} \in \Omega'_{con}$ to source contour point $cp \in \Omega_{con}$. We choose this algorithm mainly due to the fact that it is capable of both rigid and nonrigid spatial transformation and is more robust to assign contour correspondences as a queue. Then, the contour points with top n_c curvature values are picked up as key points.

Structure Correspondence. Once the matching of contour key points is completed, we compute the planar transformation ψ using thin plate splines (TPS) [7], which is often used to build elastic coordinate transformation. The dense

correspondences for structural pixels are acquired by

$$\Omega'_{struct}(\widehat{sp}) = \psi \cdot \Omega_{struct}(sp), \tag{3}$$

where Ω_{struct} is the point set of structural pixels (sp) in S_{struct} and Ω'_{struct} is the point set of transformed points in T_{struct}. Afterwards, the structure map of target image is computed by

$$T_{struct}(q) = \begin{cases} S_{struct}(sp), & q \in \Omega'_{struct} \text{ and } q = \widehat{sp} \\ 0, & q \notin \Omega'_{struct} \end{cases}. \tag{4}$$

T_{struct} provides a prior for predicting positions of structural pixels in the target image. We introduce the structure correspondences $< sp, \widehat{sp} >$ and target structure map T_{struct} for guided initialization and guided search.

3.3. Guided Texture Transfer

In this section, we describe how the extracted structure information and user-specified semantic annotations are used to guide the texture transfer process. Our structure-guided texture transfer approach is designed based on an optimization-based texture synthesis [42] and utilizes an improved PatchMatch to search the nearest neighbor fields.

More specifically, we incorporate customized guidance by modifying the original energy function (Section 3.3.1) and three guiding channels (semantics, structure and coherence) are introduced for dynamical guidance using changeable weights (Section 3.3.2-3.3.4). Finally, after initialization with structure information, we optimize the energy function by performing guided search and vote iteratively (Section 3.3.5).

3.3.1 Energy Function

Our goal is to synthesize the target stylized image using stylized textures in source. We pose this problem as a patch-based optimization task with the following energy function

$$E = \sum_{q \in T} \min_{p \in S} (\lambda_1 E_{sem}(p,q) + \lambda_2 E_{struct}(q) + E_{coh}(p,q)), \tag{5}$$

where p denotes the center coordinate of the source patch in S_{sem} and S_{sty}, and q is the center coordinate of the target patch in T_{sem}, T_{sty} and T_{struct}. λ_1 and λ_2 are the two weight parameters of semantic and structure guidance terms, respectively. We define λ_1 as a linear variable decreasing with iteration times and λ_2 as a constant based on the shape similarity between source and target:

$$\lambda_1 = \frac{t_e - t}{t_e - t_s}\beta, t_s \leqslant t \leqslant t_e, \tag{6}$$

$$\lambda_2 = exp\{-\frac{1}{|\Omega'_{con}|} \sum_{\widehat{cp} \in \Omega'_{con}} d(\widehat{cp}, cp)\}, \tag{7}$$

where t_s and t_e denote the starting and ending times of iteration, respectively. λ_1 will be changeable from β to 0. $d(\widehat{cp}, cp)$ is the Euclidean distance between the contour point in target and its aligned correspondence. During initial iterations, with β set to a large value, semantic guidance is in dominant position leading boundary patches to find reliable correspondences first. The influence of semantic guide is gradually weakened with reduction in λ_1. Structure and textural coherence terms weighted by λ_2 guide synthesis together in the later stage.

3.3.2 Semantic Guide

The semantic map specified by users introduces manual control to the texture transfer process. Same color labels in S_{sem} and T_{sem} manifest the similar objects with identical stylized texture. We manually produced labels via the brush and quick selection tool of photoshop in about 30 seconds for each image. A semantic label should cover an object to naked eyes to avoid textures in one label being synthesized in another. We define the semantic guidance term using L2-norm of two sampled patches in RGB space

$$E_{sem}(p,q) = \|T_{sem}(N_q) - S_{sem}(f(N_p))\|^2, \tag{8}$$

where $T_{sem}(N_q)$ is a $\omega \times \omega$ patch sampled around the center position q in target semantic map T_{sem}, and $S_{sem}(f(N_p))$ is a $\omega \times \omega$ patch centered at source pixel p with geometric transformation f applied. The transformation f encompasses transform, rotation and reflection. The i-th pixel in N_p is transformed as

$$f(N_p^i) = \gamma H \Delta N_p^i + p, \tag{9}$$

where $H = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$ denotes the rotation matrix, $\gamma \in \{1, -1\}$ represents the reflection parameter, and ΔN_p^i is the coordinate of pixel i related to p.

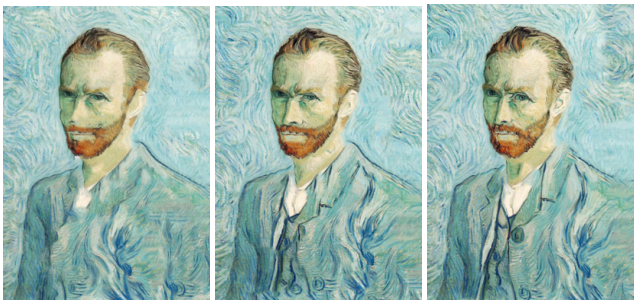
3.3.3 Structure Guide

The salient structure in S_{sty} ignored by semantic map is pre-projected as T_{struct}. With this prior we describe E_{struct} as the similarity of the target structural patch and temporary stylized patch. Structural pixels are constrained with the following equation

$$E_{struct}(q) = \sum_{i=0 \ldots \omega^2 - 1} \frac{(T_{struct}(N_q^i) - T_{sty}(N_q^i))^2 \kappa(N_q^i)}{\tau(N_q)}, \tag{10}$$

where $\kappa(N_q^i)$ denotes whether the i-th pixel in N_q is a structural pixel or not, defined by

$$\kappa(N_q^i) = \begin{cases} 1, N_q^i \in \Omega'_{struct} \\ 0, N_q^i \notin \Omega'_{struct} \end{cases}, \tag{11}$$



(a) no structure guide (b) struct init (c) struct init+search

Figure 6: The effects of structure guide. (a) Results without structure guide. (b) Results obtained by initialization with structure prior. (c) Results obtained by both initialization and search with structure guide.

and $\tau(N_q) = \sum_{i=0...\omega^2-1} \kappa(N_q^i)$ denotes the number of structural pixels in patch N_q. The structure guidance term affects the synthesis of T_{sty} by leveraging EM iterations since the weighted average based on energy is used in the vote step and thus results in the search step can also be improved iteratively.

3.3.4 Coherence Guide

The coherence term aims to synthesize the target image using the consistent stylized textures in source. We define this term similar to semantic term using distance in RGB space

$$E_{coh}(p,q) = \|T_{sty}(N_q) - S_{sty}(f(N_p))\|^2, \qquad (12)$$

where T_{sty} is the temporarily-generated image, which will be iteratively improved.

3.3.5 Function Optimization

Our optimization approach is modified from the one originally proposed by Wexler *et al*. [42] with the main difference for the customized guidance and improved PatchMatch. To be specific, the energy function is optimized by EM-like iterations with two steps (guided search and vote) performed alternatively. Here, we mainly describe the difference of our method against the original one, whose more details can be found in [42].

Guided Initialization. In the coarsest level, structure correspondences acquired in section 3.2 are utilized to initialize the NNF that assigns the source patch to each target patch. Then the initialization of T_{sty} is synthesized via the vote step. Projecting structural patches to roughly correct positions in the initial stage is beneficial for strengthening the structure information, which will be propagated to neighbors later. In the finer level, the NNF and T_{sty} are both upsampled from coarser one as the initialization of current level. Meanwhile, we construct the image pyramid for target structure map T^l_{struct} with $l \in [1, L]$ and L is the number of pyramid levels.

Guided Search. In the search step, we mainly leverage Equation (5) to search better correspondences between source and target with the given T_{sty}. Specially, the structure guidance uses multi-scale T^l_{struct} in coarse-to-fine resolution. Low-frequency structure map T^L_{struct} provides a roughly correct projection guide while the details are missing, and high-frequency structure map T^1_{struct} contains clearer detailed textures but suffers from severe corruption with cracks. Multi-scale texture synthesis integrates them together for better synthesis. The effects of initialization and search with structure guide are shown in Figure 6.

Moreover, inspired by [23] we use the PatchMatch (PM) algorithm [3] with extended NNF and matrix operations. The NNF is extended to $[x, y, \theta, \gamma]$ containing position (x, y), rotation θ and reflection γ. With matrix operations, target patches are not processed in scanning order and the neighbor information is propagated in target patches simultaneously. Thus, we do not need to search geometric transformation space explicitly. Instead, the four directional propagations are performed alternately until no patch is updated. Geometric transformable patches are provided in the random search step. In this manner, we accelerate the retrieval of nearest neighbors while obtaining more abundant source patch for synthesis. This improved PM method also reduces the mistake accumulation in one-by-one fashion and encourages the correct correspondences scattered in multi-places to be better preserved and propagated.

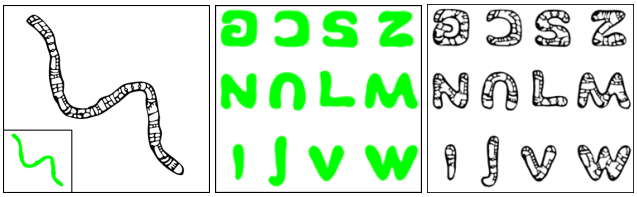
Vote. In the vote step, we reconstruct the target stylized image T_{sty} with given NNF. T_{sty} is produced by computing the weighted average color of co-located pixels in neighbor patches as mentioned in [22].

4. Implementation Details

We use a fixed patch size 5×5. Parameter ℓ and δ in Equation (1) control the salient degree of structural pixels. ℓ is set to a higher value to decrease boundary saliency since boundary patches mainly depend on semantic annotations rather than structure guidance. δ is a saliency threshold between 0 and 1. Parameters n_c controls the number of key contour points. Parameters λ_2 and λ_1 determined by β control the balance among global semantic consistency, structure completeness and local texture coherence. In this paper, we set the values of ℓ, δ, β and n_c as 10, 0.2, 10 and 20, respectively. The rotation angle θ ranges from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ and γ takes value from $\{-1, 1\}$. In synthesis process, we use ten levels of the image pyramid with φ optimization iterations on each level, where φ linearly decreases when synthesizing from coarse to fine.

(a) Input (source) (b) Input (semactics) (c) Output (target)

Figure 7: Doodles-to-artworks. Image courtesy of Champandard [8] and Liao *et al*. [31]



(a) Input (source) (b) Input (semactics) (c) Output (target)

Figure 8: Decorative Pattern Editing. Image courtesy of Lu *et al*. [33]

5. Experimental Results

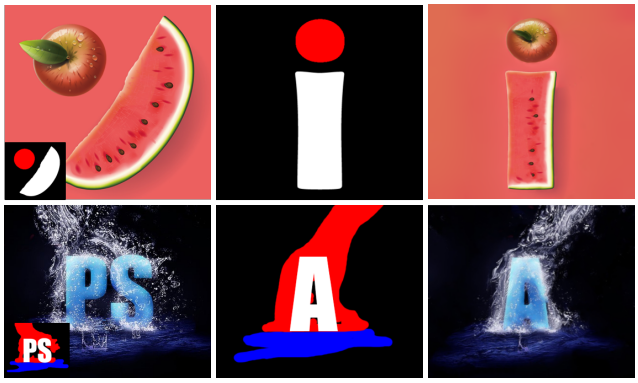
We implemented our method in Matlab with a 4 GHz quad-core CPU. It takes around 2 minutes to synthesis a target stylized image with 500×400 pixels. To validate the performance of our general framework, we applied the proposed technique to a wide variety of interactive texture transfer applications and illustrated that it performs better than other state-of-the-art methods.

5.1. Applications

Our approach can be effectively used for multiple tasks of interactive texture transfer such as turning doodles into artworks, editing decorative patterns with user guidance, generating special effect texts and swapping textures.

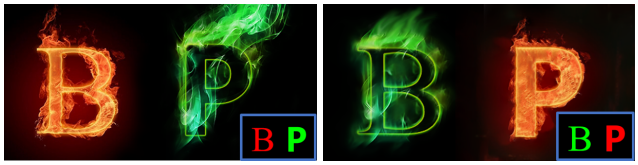
Doodles-to-artworks In this scenario, two-bit doodles annotated by users can be turned into fine paintings with similar styles as corresponding exemplars. When users force fine-grained guidance into semantic map, this task is more like an image morphing problem with object deformation. While with multiple objects in the picture, this task becomes an image retargeting process. Results are shown in Figure 7 and more can be found in supplemental materials.

Decorative Pattern Editing. As depicted in Figure 8, given an exemplar, the decorative patterns can be synthesized naturally along with the user-specified path. To be specific, our method first automatically cuts the stroke into several sections based on the curvature and then performs the structure



(a) Input (source) (b) Input (plain text) (c) Output (target)

Figure 9: Special Effect Text Generation. Image courtesy of Zcool [1]



(a) Input (source) (b) Output (target)

Figure 10: Texture Swap. Image courtesy of Yang *et al*. [43]

projection for each section to ensure the accuracy of propagated structure information.

Special Effect Text Generation. As shown in Figure 9, our method is also effective for generating texts with various textures such as the skin of object and the stylization designed by artists. We can also control the effect distribution in complex texts and complete text effect transfer with fragile decorative textures. The proposed method performs better when the shapes are more similar between source and target, but the structure can still be well preserved even with large shape difference.

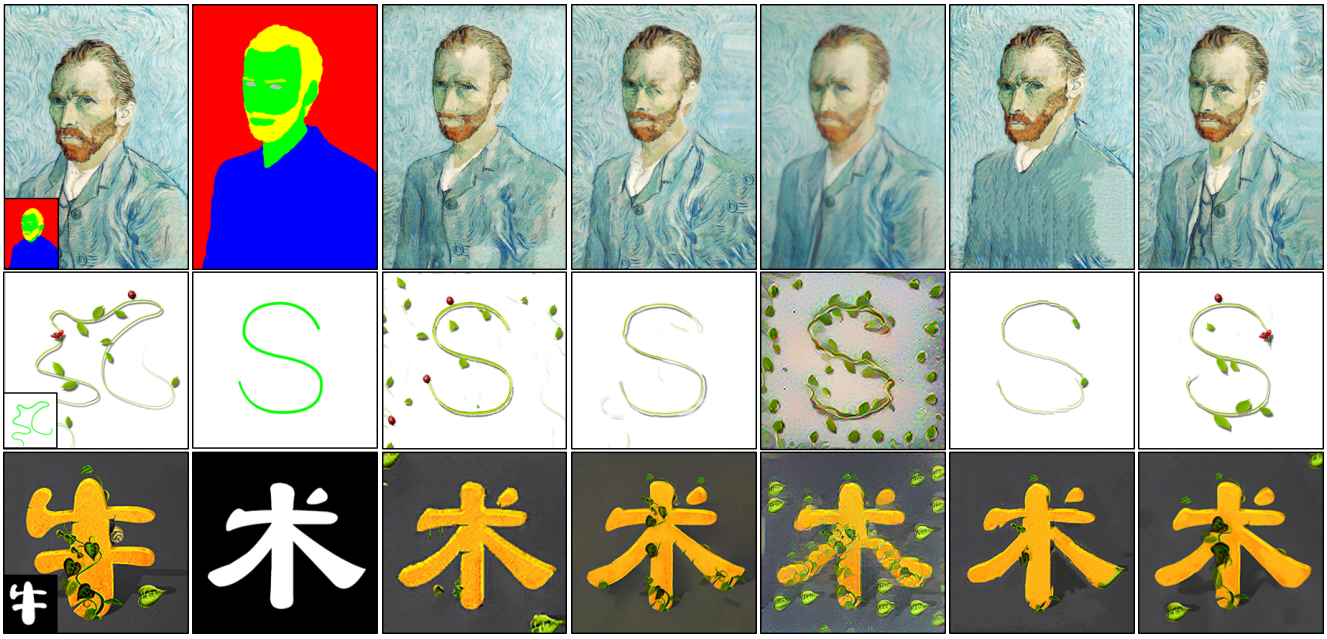
Texture Swap. From Figure 10 we can see that our method is also capable of texture swap. For instance, apples can swap the skin with each other (see Figure 1 (e)) and special effect texts can swap the effects among them.

5.2. Comparison

We compared our algorithm with state-of-the-art interactive texture transfer methods in different scenarios mentioned in Section 5.1. See Figure 11 for the results and more can be found in supplemental materials.

Image analogy [21], a pioneering approach, fails to maintain local structures in the target stylized image, such as incomplete leaves in the second row and missing vine in the bottom row. It is also unable to preserve high-frequency details for the nose and collar in the top row.

Text effect transfer [43], tailored to special effect text generation, uses a spatial distribution model based on the



(a) (S_{sem}, S_{sty}) (b) T_{sem} (c) Image Analogy (d) Text Transfer (e) Neural Doodle (f) Deep Analogy (g) Our method

Figure 11: Comparison with state-of-the-art texture transfer methods. Image courtesy of Van Gogh and Zcool [1]

high correlation between patch patterns and their distances to text skeleton. The method performs better than other previous approaches in our experiments of special effect text generation, but it still fails to preserve the vine's structure (vine effect in the third row) whose effect patterns do not distribute according to the distances. It also suffers from dislocation for inner textures in other scenarios.

Neural doodle [8] based on the combination of CNN and MRF methods [29] does not guarantee a high-quality image with low-level details (the first row). It produces color noise for photorealism images and messy background with leaves appearing randomly (the second and third row).

Deep image analogy [31] achieves attractive results with two stylized images as input. However, in our scenarios one stylized image must be replaced with a semantic map and the other stylized image needs to be automatically synthesized. With low neural activation in the semantic map, it is difficult to find correct correspondence in textureless regions using the VGG network. As we can see from the first row in Figure 11 (f), although fine-grained controls have been performed to the face, synthesized facial features are still more similar as the source stylized image with little characteristic of the target content. If we increase the content weight, it will fill regions with pure color patches repeatedly (such as the body part). No internal structure is preserved due to the simplicity of semantic guidance.

From the last column of Figure 11, we can see that the proposed framework is effective for multiple tasks, synthesizing higher-quality content-specific stylization with well-preserved structures. Furthermore, under the same experimental settings, our method runs much faster (≈2 mins per image) than other existing approaches such as image analogy (≈15 mins) and neural doodle (≈40 mins).

6. Conclusion

This paper presented a general framework to interactive texture transfer with structure guidance. Our method can automatically migrate style from a given source image to a user-controlled target image while preserving the structure completeness and visual richness. More specifically, we introduced a structure guidance acquired by automatically extracting salient regions and propagating structure information. By incorporating the structure channels with semantic and textural coherence, guided texture transfer can be achieved. Experimental results showed that the proposed framework is widely applicable for many texture transfer challenges. Despite the current tendency to use neural-based methods to style transfer, our results demonstrated that a simple conventional texture synthesis framework can still achieve state-of-the-art performance.

Acknowledgements

This work was supported by National Natural Science Foundation of China (61672043, 61472015 and 61672056), National Key Research and Development Program of China (2017YFB1002601) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

[1] Zcool. http://www.zcool.com.cn.

[2] M. Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226. ACM, 2001.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24–1, 2009.

[4] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, pages 29–43. Springer, 2010.

[5] C. Barnes, F.-L. Zhang, L. Lou, X. Wu, and S.-M. Hu. Patchtable: Efficient patch queries for large datasets and applications. *ACM Transactions on Graphics (TOG)*, 34(4):97, 2015.

[6] P. Bénard, F. Cole, M. Kass, I. Mordatch, J. Hegarty, M. S. Senn, K. Fleischer, D. Pesare, and K. Breeden. Stylizing animation by example. *ACM Transactions on Graphics (TOG)*, 32(4):119, 2013.

[7] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[8] A. J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

[9] L. Cheng, S. N. Vishwanathan, and X. Zhang. Consistent image analogies using semi-supervised learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[10] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[11] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82–1, 2012.

[12] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[13] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[14] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[15] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sỳkora. Stylit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)*, 35(4):92, 2016.

[16] O. Frigo, N. Sabater, J. Delon, and P. Hellier. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–561, 2016.

[17] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[18] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[20] R. Hashimoto, H. Johan, and T. Nishita. Creating various styles of animations using example-based filtering. In *Computer Graphics International, 2003. Proceedings*, pages 312–317. IEEE, 2003.

[21] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

[22] J. Ho Lee, I. Choi, and M. H. Kim. Laplacian patch-based image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2735, 2016.

[23] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Image completion using planar structure guidance. *ACM Transactions on Graphics (TOG)*, 33(4):129, 2014.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[25] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[26] N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)*, 24(3):795–802, 2005.

[29] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

[30] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[31] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

[32] J. Lu, C. Barnes, S. DiVerdi, and A. Finkelstein. Realbrush: painting with examples of physical media. *ACM Transactions on Graphics (TOG)*, 32(4):117, 2013.

[33] J. Lu, C. Barnes, C. Wan, P. Asente, R. Mech, and A. Finkelstein. Decobrush: drawing structured decorative patterns by example. *ACM Transactions on Graphics (TOG)*, 33(4):90, 2014.

[34] J. Lu, F. Yu, A. Finkelstein, and S. DiVerdi. Helpinghand: example-based stroke stylization. *ACM Transactions on Graphics (TOG)*, 31(4):46, 2012.

[35] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. *arXiv preprint arXiv:1703.07511*, 2017.

[36] M. Lukáč, J. Fišer, J.-C. Bazin, O. Jamriška, A. Sorkine-Hornung, and D. Sỳkora. Painting by feature: texture boundaries for example-based image creation. *ACM Transactions on Graphics (TOG)*, 32(4):116, 2013.

[37] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[38] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[39] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[40] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3), 2007.

[43] S. Yang, J. Liu, Z. Lian, and Z. Guo. Awesome typography: Statistics-based text effects transfer. *arXiv preprint arXiv:1611.09026*, 2016.