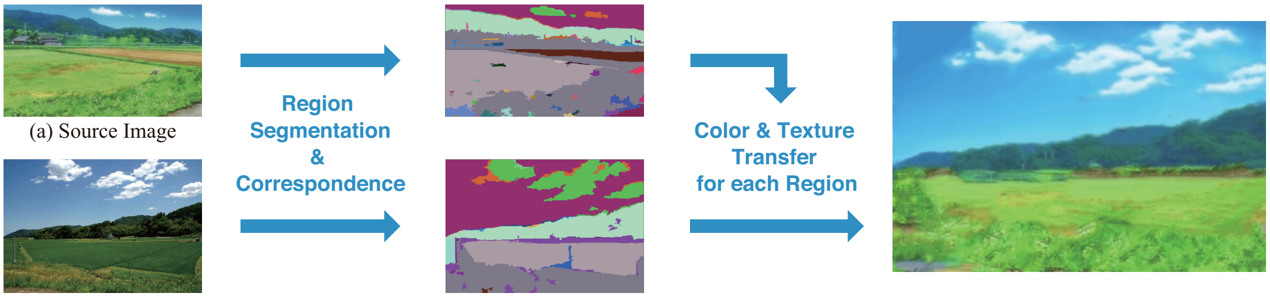
Region-Based Painting Style Transfer

Shugo Yamaguchi[1,3] * Takuya Kato[1,3] Tsukasa Fukusato[1] Chie Furusawa[1,3] Shigeo Morishima[2,3]†

[1]Waseda University [2]Waseda Research Institute for Science and Engineering [3]JST CREST



(a) Source Image

(b) Target Photograph

(c) Region Correspondence Map

(d) Our Result

Figure 1: *Process overview. First, we divide the source (a) and target (b) images into regions by Graph-based method and compute the correspondence between regions based on the color and texture features (The corresponding region is filled with a same color in (c)). Finally, we transfer the color and texture for each region by PatchMatch method to yield the final result (d).*

Abstract

In this paper, we present a novel method for creating a painted image from a photograph using an existing painting as a style source. The core idea is to identify the corresponding objects in the two images in order to select patches more appropriately. We automatically make a region correspondence between the painted source image and the target photograph by computing color and texture feature distances. Next, we conduct a patch-based synthesis that preserves the appropriate source and target features. Unlike previous example-based approaches of painting style transfer, our results successfully reflect the features of the source images even if the input images have various colors and textures. Our method allows us to automatically render a new painted image preserving the features of the source image.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms I.4.10 [Image Processing and Computer Vision]: Image Representation—Statistical J.5 [Computer Applications]: Arts and Humanities—Fine arts;

Keywords: NPR, style transfer, texture synthesis

1 Introduction

Painted images are pleasing to look at and useful for various purposes. Excellent painting is unique in color usage and detail. However, painting effectively requires a great deal of skill. Therefore, an automatic conversion of a photograph into a image that appears painted has an enormous merit as a way of generating a painted image quickly and easily. Accordingly, such applications have been developed and mounted in various software titles, such as Adobe Photoshop, Adobe illustrator, and GIMP. These software titles are able to automatically generate a painted image from a photograph using different setting parameters. Such methods usually use an image-filtering technique However, single filters are applied equally to the whole image. As a result, the appearance of the filtered image seems to be the same throughout, which makes the resulting image look like a photograph with different color patterns. In addition to this problem, such methods have to change the algorithm according to the desired painting style.

Accordingly, many patch-based techniques have been proposed to synthesize painted images from photographs using a single algorithm. The core assumption of these approaches is that once finding a correspondence between the patches (tiny squares of a few pixels) of the source and target image, any style of painting can be reproduced. Certainly, this assumption is true for many styles such as impressionist paintings, and most patch-based approaches are built on top of this core assumption. However, this assumption does not always apply because many modern paintings, including anime background images and surrealist images, use multiple color combinations and strokes. In these cases, simple patch-based algorithms do not offer plausible results because the correct correspondence becomes a difficult task to achieve.

Considering this problem, we set our core assumption as follows: once finding the correspondence between the source and target *regions*, any style of painting can be reproduced. This may seem very similar to the assumption of existing approaches (e.g. [Hertzmann et al. 2001], [Efros and Freeman 2001]). Nevertheless, consideration of the corresponding regions prior to conducting a patch-based synthesis offers a better initialization of where to search for the patches. This has a powerful result with better synthesis when transferring styles with multiple color combinations and strokes while searching for a optimal patch from only similar regions.

Thus, region segmentation and correspondence are indispensable. We conduct a painting style transfer through the following processes (Fig. 1). First, we divide the input images into regions using a graph-based technique such that each segmented region has a single texture (Section 3.1). Second, we automatically correlate the source and target regions. Region correspondence is established by calculating the feature distance of the colors and textures after ap-

*e-mail:wasedayshugo@suou.waseda.jp

†e-mail:shigeo@waseda.jp

SA'15 Technical Briefs, November 02 – 06, 2015, Kobe, Japan.
© 2015 ACM. ISBN 978-1-4503-3930-8/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2820903.2820917

plying histogram matching (Sections 3.1.1 and 3.1.2). Third, the target region colors are transferred to the source regions colors to preserve the source color feature and target shade (the first half of Section 3.2). Finally, the source textures are synthesized in the target regions (the latter half of Section 3.2). Additionally, we set the region constraint of this process to eliminate any errors in the region boundaries of the output image caused by the region segmentation, and we preserve the coherence in the region boundaries of the source image.

Our primary contributions in this paper are as follows:

- automatically computing the correspondence between the source and target regions (different colors and textures)
- preserving coherence in the region boundaries and regions (particularly the gradation).

These allow us to generate a newly painted image while preserving the features of the source image. Our method can be naturally extended to correspondence between still and moving images to create a stylized video.

2 Related Work

Example-based approaches. Such approaches use existing images as a reference, and transfer the colors and textures onto photographs. These approaches can transfer the painting styles of any artist appropriately. For example, Image Analogies [Hertzmann et al. 2001] conducts a nearest-neighbor search and pixel-based synthesis. Image Quilting [Efros and Freeman 2001] also searches for and synthesizes the nearest-neighbor patch. A nearest-neighbor search is a significant process for example-based approaches. However, a nearest-neighbor search has a problem in that it takes an enormous amount of time. The PatchMatch algorithm [Barnes et al. 2009] solved this problem, and enables example-based approaches to be easily applied. However, a nearest-neighbor patch is not always the optimal patch for a painting style transfer because photographs and paintings differ in color and texture. For example, real-world colors need to be converted into specific colors for a painted image; further, the type of brushwork is different according to the target elements (e.g., sky, mountain, or grass). Therefore, we apply region segmentation and a texture transfer for each region individually.

Region-Based Approaches. Such an approach [Zeng et al. 2009] divides an image into regions and synthesizes the brush-strokes from a dictionary. Although an appropriate synthesis is conducted for each region, this type of method requires a brush-dictionary. Corpus-based visual synthesis [Mital et al. 2013] is a suggested technique for extracting brush strokes from a painting image region and synthesizing it into a photograph region. However, even if an example image is drawn with a brush, the extracted brush cannot reproduce the details of the source image. The synthesized results are not clear, and the painting style cannot be preserved. Wang et al. [2004] suggested applying a region correspondence and transfer texture similar to our suggestion. This method succeeds in a texture transfer preserving the source features, although the correspondence requires manual control and the region boundary includes artifacts because of ignoring the coherence on the boundary. In this paper, we thus apply a texture transfer preserving the detailed features and region boundary coherence, in addition to considering the particular features of the region.

3 Our Approach

3.1 Region Correspondence

We conduct region segmentation and correspondence.

First, we divide both the source and target images into regions. It is necessary for a divided region to have a single kind of texture so that we may conduct a texture transfer between the corresponding regions successfully. In addition, it is necessary that a gradation region, such as the sky, not be divided into two different regions (eliminating false edges). Therefore, we divide the regions using a Graph-Cut-based automatic region segmentation technique [Felzenszwalb and Huttenlocher 2004].

We then compute the correspondence between the source image regions R^S and the target photograph regions R^T. In this process, we conduct a global color transfer (3.1.1) before calculating the feature distance (3.1.2) in order to compute any impartial correspondence.

3.1.1 Modified Color Transfer

A common feature for corresponding regions that have similar properties is to calculate the mean of each region color and its Euclidean distance. This method applies the correspondence for the minimum distance. However, regions with inappropriate correspondence often have a problem because the colors of the appropriately corresponding regions often have similar but different features. The destinations of the correspondence can be concentrated in a particular region when the input images are quite different in color. A partial correspondence causes different objects to be filled in with the same texture.

Therefore, we apply a fitting of the color histogram between the input images before calculating the feature distance. To do so, we use the ColorTransfer [Reinhard et al. 2001] technique. This technique replaces the target mean and standard deviation with the source to transfer the global color features.

However, ColorTransfer cannot be applied to an image that has various regions (Fig. 2 (a)) because other peaks affect the transferring peak when a histogram has various peaks. We conduct histogram clustering to apply ColorTransfer to a specific range of histogram with similar color features. We employ the color clustering technique introduced by Chang et al. [2006] to apply a cluster based on human perception. We divide the pixels into nine color categories C = (black, white, gray, red, orange, yellow, green, blue, and purple) based on the HSV color system. We then apply ColorTransfer to each color group $j \in C$ independently.

$$\begin{aligned} L^j_{dst} &= \frac{\sigma^{Lj}_{src}}{\sigma^{Lj}_{tar}}(L^j_{tar} - \langle L^j_{tar}\rangle) + \langle L_{src}\rangle \\ a^j_{dst} &= \frac{\sigma^{aj}_{src}}{\sigma^{aj}_{tar}}(a^j_{tar} - \langle a^j_{tar}\rangle) + \langle a^j_{src}\rangle \qquad (1) \\ b^j_{dst} &= \frac{\sigma^{bj}_{src}}{\sigma^{bj}_{tar}}(b^j_{tar} - \langle b^j_{tar}\rangle) + \langle b^j_{src}\rangle \end{aligned}$$

where L^j_{tar}, a^j_{tar}, and b^j_{tar} are the pixel values of the target image in the L*a*b* color space; $\sigma^j_{tar}, \sigma^j_{tar}$,and$\sigma^j_{tar}$ are the standard deviations of the target image; L^j_{src}, a^j_{src}, and b^j_{src} are the pixel values of the source image; $\sigma^j_{src}, \sigma^j_{src}$, and σ^j_{src} are the standard deviations of the source image; L^j_{dst}, a^j_{dst}, and b^j_{dst} are the pixel values of the destination image; and $\langle L\rangle$ is the mean of color value. The results of a color transfer are shown in Fig. 2 (b).

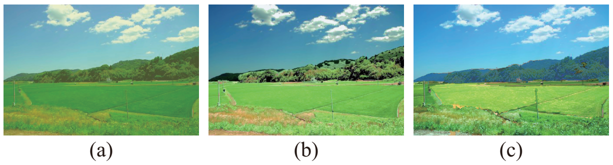


Figure 2: *(a) ColorTransfer [Reinhald et al.]. (b) Our modified ColorTransfer (Source: Fig. 1(a), Target: Fig.1(b)). (c) The result of color transfer for each region (Section 3.2).*

3.1.2 Feature Distance

After applying a modified ColorTransfer, we take the region correspondence by calculating the feature distance. We calculate the color feature distance d_{col} and texture feature distance d_{tex} (discussed below), and determine the source region R^S corresponding to the target region R^T .

$$\begin{aligned} R^S &= \arg\min_{R^S} d_{col}(R^T, R^S) \\ & \quad \text{subject to } d_{tex}(R^T, R^S) < \gamma \end{aligned} \tag{2}$$

where γ is a threshold.

Here, d_{col} is established by calculating the moment of inertia of the difference between the color histogram of the source region $\mathbf{H}^S_{lab}(x)$ and a target region $\mathbf{H}^T_{lab}(x)$. The histograms are in the L*a*b color space.

$$d_{col} = \int |\mathbf{H}^S_{lab}(x) - \mathbf{H}^T_{lab}(x)| \cdot |x - \mu| dx \tag{3}$$

where

$$\mu = \frac{\int |\mathbf{H}^S_{lab}(x) - \mathbf{H}^T_{lab}(x)| x dx}{\int |\mathbf{H}^S_{lab}(x) - \mathbf{H}^T_{lab}(x)| dx} \tag{4}$$

We employ the local binary pattern (LBP) feature distance to avoid taking a correspondence with an entirely different texture region. The LBP feature value is calculated by converting the relation between the center pixel and the neighborhood pixels into an eight-digit binary number. The histogram of the LBP feature value over the pixels is a 256-dimension LBP feature vector. The LBP feature vector can describe a texture feature without being affected by the color. Below, d_{tex} is the LBP feature distance between the source and target region, and is described as

$$d_{tex} = ||\mathbf{F}^S_{lbp} - \mathbf{F}^T_{lbp}||^2. \tag{5}$$

where $\mathbf{F}^S_{lbp}$ is the LBP feature vector of the source region and $\mathbf{F}^T_{lbp}$ is that of the target region.

3.2 Color and Texture Transfer for each Region

We transfer the color of the target region to the color of the corresponding source regions (by Eq.(1), j is the region number here.) in order to preserve the shading condition of the target (Fig. 2 (c)).

For the texture transfer, we use a technique of texture optimization [Kwatra et al. 2005]. This technique synthesizes a texture by repeating the nearest-neighbor search and the optimization of the output color. Let X denote the output image and Z denote the input image. Let $\mathbf{x}$ be the vectorized version of X. Let $\mathbf{x}_p$ be a patch at coordinate $p \in X$ and let $\mathbf{z}_p$ be the nearest-neighbor patch of $\mathbf{x}_p$ in Z. In addition, $\mathbf{x}$ is optimized to minimize the texture energy

$$E(\mathbf{x}; \{\mathbf{z}_p\}) = \sum_{p \in X} ||\mathbf{x}_p - \mathbf{z}_p||^2 \tag{6}$$

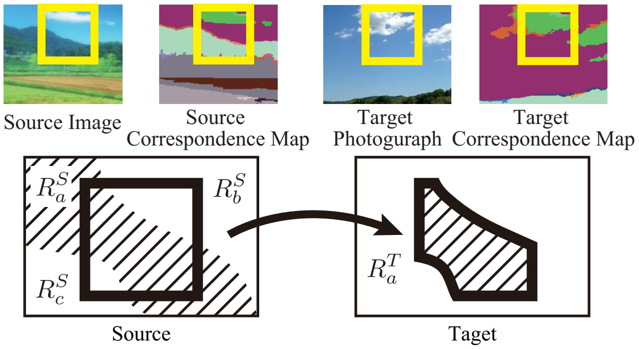


Figure 3: *Region constraint for low-resolution patch synthesis.*

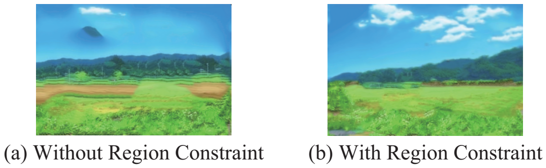


Figure 4: *Comparison of multi-level resolution synthesis - (a) without region constraint and (b) our method.*

In addition, we conduct a multi-level resolution synthesis. We first synthesize the texture at a coarse resolution, and then up-sample it to a higher resolution. Our multi-level resolution synthesis is able to apply a texture synthesis that stores both the global (low-frequency) and local (high-frequency) features of the source image. Furthermore, we use the PatchMatch [Barnes et al. 2009] technique to accelerate the nearest-neighbor search. PatchMatch speeds up the nearest-neighbor search using the natural coherence of the images as follows: (1) all patches initially have random assignments, (2) every patch checks the neighbors above and to the left to search for good matches, and (3) every patch searches randomly for improvements in the concentric neighborhoods.

However, multi-level resolution synthesis cannot preserve the target region shape (Fig. 4(a)). Therefore, we expand the texture optimization and PatchMatch technique to preserve the target shape. We set the constraint of the search range in steps (1) and (3). During the initialization in (1), we obtain random assignments from only the corresponding region. In the random neighborhood search in (3), we do not select any location other than the corresponding region. At a low-resolution for a multi-level synthesis, we only synthesize the pixels of a corresponding region in order to preserve the target region shape (Fig. 3). At a high resolution, we synthesize the original patch to preserve the source region boundary.

4 Results

The results of a painting style transfer from a source image to a target photograph are shown in Fig. 5. The figure also shows the results of Image Analogies [Hertzmann et al. 2001] and Image Quilting [Efros and Freeman 2001] as a comparison. Image Analogies was unable to transfer the textures of the source image. Image

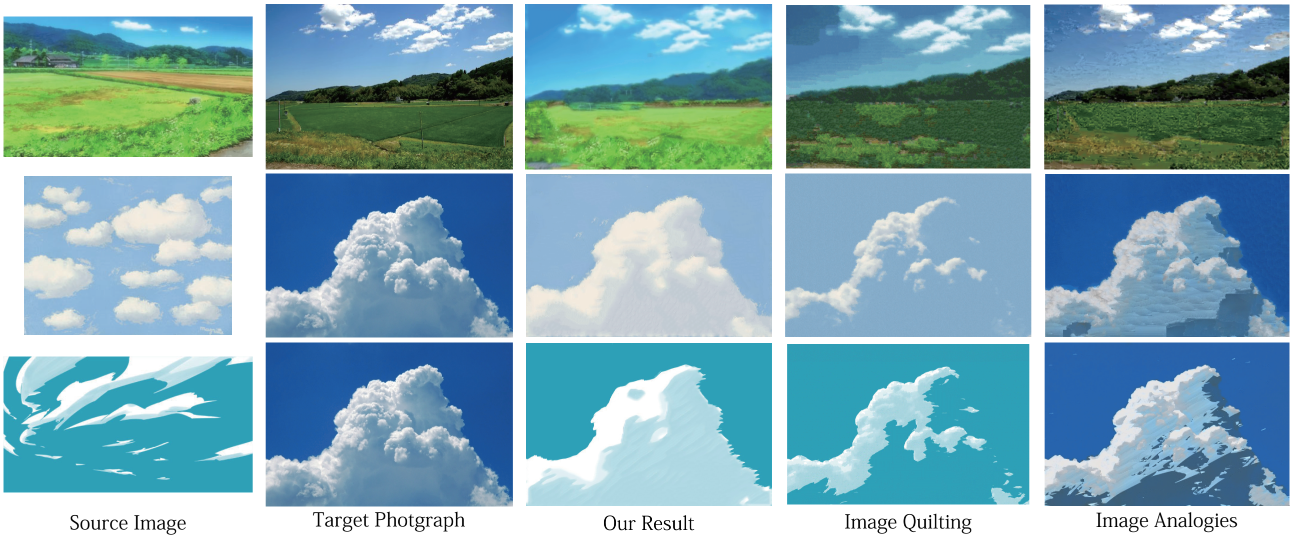


Figure 5: *Comparison the results between our approach and previous approaches. The source images were referenced from: "Nonnonbiyori", Rene Magritte "The Curse", and Pixiv (by inkwataame) from top to bottom.*

Quilting mixed the textures and could not preserve the source textures. On the other hand, our approach achieves results that reflect the painting style (color and texture) of the source images. Moreover, our method can input more than one painting style using a region corresponding with more than one source image.

We used a 2.3 GHz Intel ®Core i7 CPU in this paper, and the execution time was approximately 0.7 s for region segmentation, 0.8 s for region correspondence, and 5.0 s for a texture transfer of an image 640 pixels x 360 pixels in size.

5 Application for Video

Furthermore, our method can be extended to video style transfers. In 2D-animation production, moving backgrounds are frequently created by hand drawing each frame, mapping the textures onto 3D models, or rendering a physical simulation. However, it is difficult for these techniques to generate background with stable quality or time consuming. Our method allows us to achieve this by transferring the painting style to real video frames or a simulated animation. In this paper, we applied our method to each frame of a video. For coherence, we first obtained a region mask in one frame and then established a region mask for all other frames by calculating the pixel correspondence using the PatchMatch method.

6 Conclusion and Future Work

In this paper, we suggested a painting style transfer using each region feature. Our method presented the transfer of a painting style, which is unachievable using previous approaches. In addition, our method can be naturally extended to computing a correspondence with painted images to video frames in order to create a stylized film. A direction for future work is an optimization of the parameters used in region segmentation and region correspondence. In addition, we aim to speed-up the texture synthesis in order to edit images and videos interactively.

References

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG 28*, 3, 24.

Chang, Y., Saito, S., Uchikawa, K., and Nakajima, M. 2006. Example-based color stylization of images. *ACM Transactions on Applied Perception 2*, 3, 322–345.

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

Felzenszwalb, P. F., and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2, 167–181.

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

Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. 2005. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (TOG)*, vol. 24, ACM, 795–802.

Mital, P. K., Grierson, M., and Smith, T. J. 2013. Corpus-based visual synthesis: an approach for artistic stylization. In *Proceedings of the ACM Symposium on Applied Perception*, ACM, 51–58.

Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. 2001. Color transfer between images. *IEEE Computer graphics and applications 21*, 5, 34–41.

Wang, B., Wang, W., Yang, H., and Sun, J. 2004. Efficient example-based painting and synthesis of 2d directional texture. *Visualization and Computer Graphics, IEEE Transactions on 10*, 3, 266–277.

Zeng, K., Zhao, M., Xiong, C., and Zhu, S.-C. 2009. From image parsing to painterly rendering. *ACM Trans. Graph 29*, 1, 2.