Blending Texture Features from Multiple Reference Images for Style Transfer

Hikaru Ikuta[1, 2*] Keisuke Ogaki[2†] Yuri Odagiri[2‡]

[1]The University of Tokyo [2]DWANGO Co., Ltd.

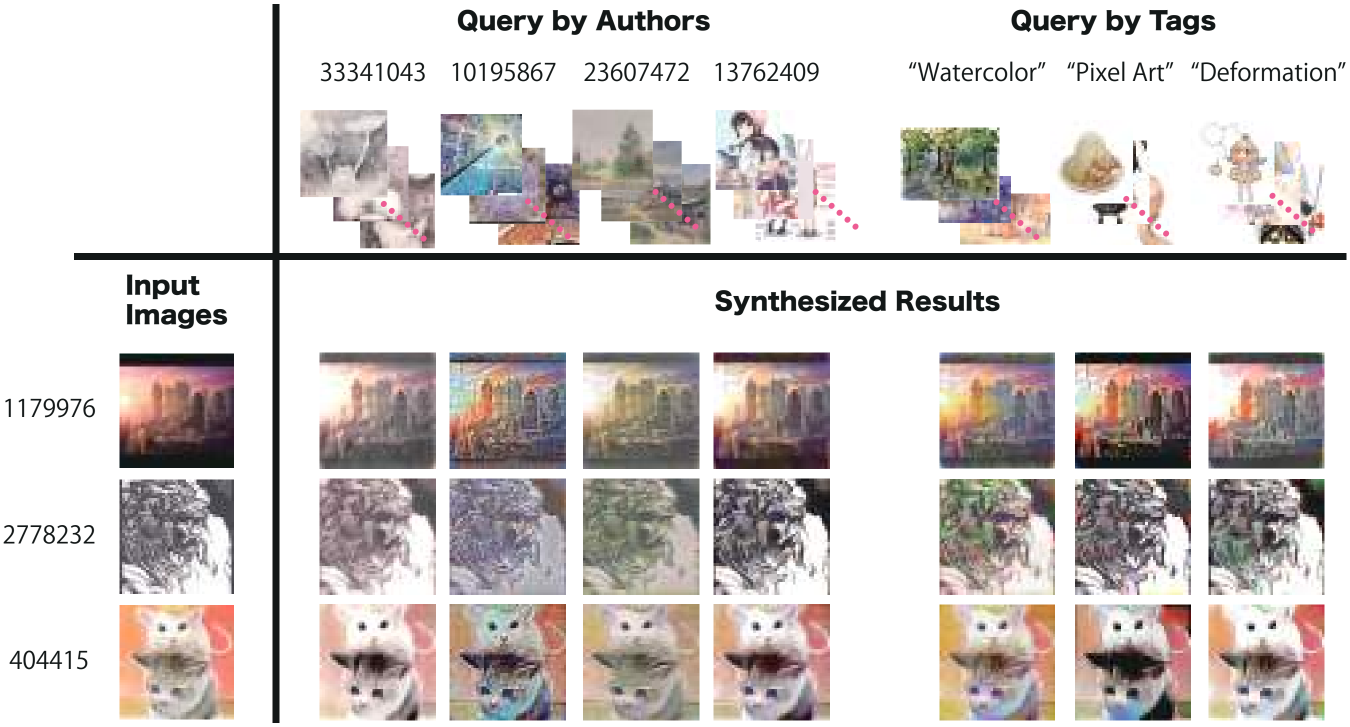

Figure 1: *Image synthesis results.*

Abstract

We present an algorithm that learns a desired style of artwork from a collection of images and transfers this style to an arbitrary image. Our method is based on the observation that the style of artwork is not characterized by the features of one work, but rather by the features that commonly appear within a collection of works. To learn such a representation of style, a sufficiently large dataset of images created in the same style is necessary. We present a novel illustration dataset that contains 500,000 images mainly consisting of digital paintings, annotated with rich information such as tags, comments, etc. We utilize a feature space constructed from statistical properties of CNN feature responses, and represent the style as a closed region within the feature space. We present experimental results that show the closed region is capable of synthesizing an appropriate texture that belongs to the desired style, and is capable of transferring the synthesized texture to a given input image.

1 Introduction

Re-creating the style of an image is a basic process in the field of art and design. For example, in animation production, backgrounds are often based on photographs of an actual location, redrawn in the animation's style. The goal of our research is to create a system that captures a certain style of artwork from a set of reference images, and re-creates an arbitrary input image in the desired style, as shown in Figure 1.

There are approaches that share the motivation of transferring a certain style. Image analogies, proposed by A. Hertzmann et al. [2001], is a method based on patch matching. The major restriction of this system is that it requires a pair of images with strict region correspondences. L. A. Gatys et al. [2015] uses the image features of a pre-learned convolutional neural network (CNN) for style transfer. Their method takes an arbitrary pair of an input image and a reference texture image and transfers the texture of the reference texture image to the input image, eliminating the need of region correspondences. The major drawback of this method is that the color distribution of the output image is restricted by the input texture image. For example, if an input texture image is mainly drawn by a blue color such as the sky, the output image tends to be painted in blue. This often causes unnatural results as shown in Figure 3, where some regions in the original input image (Figure 3(b)) is unnaturally drawn in a blue color (Figure 3(c)).

Such a problem occurs in the method by L. A. Gatys et al., since their algorithm does not distinguish the style of an artwork from the texture of a specific artwork. For example, Vincent Van Gogh's most prominent style of art is the use of small repetitive strokes, which is captured by the frequency of a certain edge pattern, in terms of features. However, since the color used in Gogh's series of artwork depends on each piece of art, features representing color may vary, therefore making it not a common style. From this exam-

*e-mail:hikaru_ikuta@ipc.i.u-tokyo.ac.jp
†e-mail:keisuke_ogaki@dwango.co.jp
‡e-mail:yuri_odagiri@dwango.co.jp

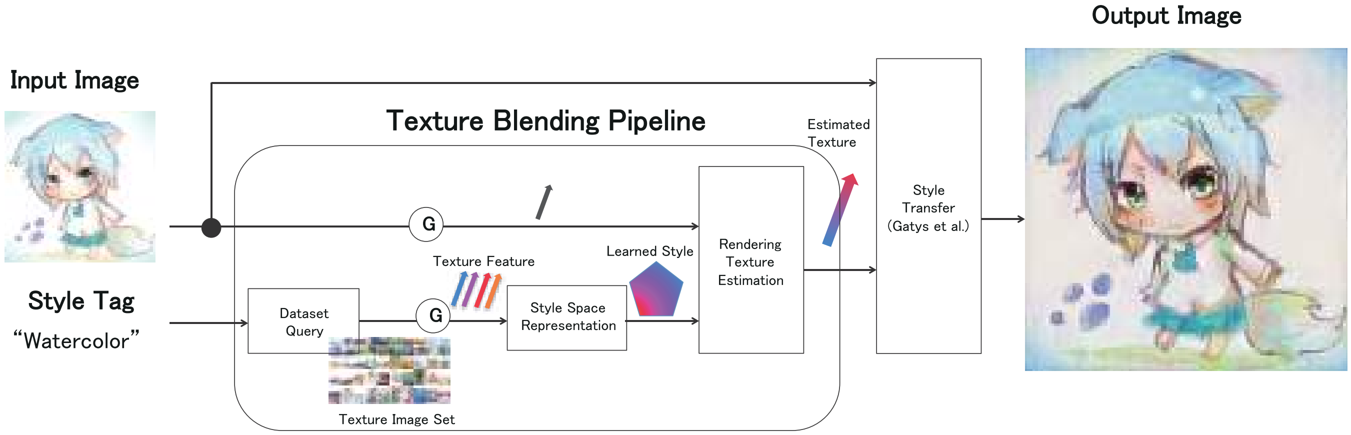

Figure 2: *Illustration of our style transfer pipeline. The system receives an input image and a style tag as the input. The system then queries the tag within the nico-illust dataset to collect 50 images annotated with the given tag. The collected images are then used to learn the desired style as a representation in a feature space constructed from CNN filter responses. The learned style representation is then used to estimate the appropriate texture for drawing the input image in the desired style. The estimated texture feature is finally passed on to the style transfer to obtain the final results.*

ple, we learn that an artwork contains two types of visual features: (1) the features that are common in all of the images that share the same style, and (2) features that are special in the certain instance of the artwork. Gogh's repetitive strokes are classified as (1), the common feature, and the different colors used in each of the artworks are classified as (2). From then on, we will refer to (2) as the style, and will refer to both (1) and (2) in general as textures.

In this paper, we propose a style transfer algorithm that transfers the common features of a certain style, by generating a texture feature that optimally fits the input image from reference texture images. Our algorithm takes a set of images created in a certain style, and learns the style as a representation in a feature space constructed by statistical amounts of CNN filter responses. We then show that the extracted style could be transferred to an arbitrary input image, using the learned representation of the style.

For evaluation, we have applied our algorithm to a novel image dataset, the *nico-illust* dataset, which we present in this paper. The nico-illust dataset contains 500,000 images, mainly consisting of digital paintings, each annotated with rich text information. To the extent of our knowledge, our dataset is the first dataset with such a volume of images mainly consisting of digital paintings, a popular style of artwork.

Our contributions are summarized as follows: (1) we propose a method of learning a given style from multiple reference images, and then transferring the artist's style to a given input image, by generating an appropriate texture from the artist's style to transfer to the image, and (2) we present a novel dataset that containing 500,000 images largely consisting of modern-style illustrations, annotated with rich text information. All code and data required for reproducing the experiments are available online [1].

2 The Method

2.1 Overview

Figure 2 illustrates the overview of our system. The input to our system is (1) the input image, for the style to be transferred, and (2) a text query. The novelty of the system is the style pipeline, where the appropriate texture for rendering the output image with the given style, is estimated. This system is constructed from two modules: (1) creating the model of the desired style, and (2) estimating the choice of texture to render the input image in the desired style. In (1), the system first constructs a representation of the desired style from a collection of images taken from the nico-illust dataset. The collection of images could be simply be derived from a single text query. The style is represented by a closed region in a feature space constructed from CNN filter responses.

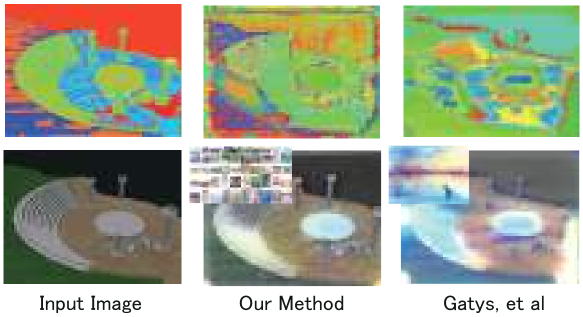

Figure 3: *Evaluation of color correspondence by clustering.*

Once the style representation is learned, it is used in (2), where the style model is used to estimate an appropriate texture to render the input image. We use [Gatys et al. 2016] to perform style transfer to the input image.

2.2 The Texture Feature

In our method, we learn the style model using a texture space that is a normalized version of the style feature G, derived from filter responses of comvolutional neural networks (CNNs), used in [Gatys et al. 2016]. [Gatys et al. 2016] indicates that the Grammian G could be used to represent the "style" of an input image, and used the style representation of G to transfer the style. In this paper, we first show that the G of a concatenated image is approximately equal to the linear combinations of the G of each of the concatenated images. We then exploit this property to show that the G of different images in the same style could be linearly blended to represent the texture of a well-defined image, belonging to a class of images created in the desired style as well.

[1] https://nico-opendata.jp/en/index.html

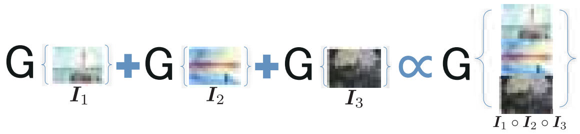

Figure 4: *The sum of G of each image I is linear to G of concatinated image* $\boldsymbol{I}_1 \circ \boldsymbol{I}_2 \circ \cdots \circ \boldsymbol{I}_N$.

To construct G, we first prepare a pre-learned CNN used for image recognition. Given an image $\boldsymbol{I}$, let $F^l_{ij}(\boldsymbol{I})$ be the jth element of the response of the ith filter in the lth layer of the CNN. The texture feature G is then represented using F^l_{ij} as

$$G^l_{ij}(\boldsymbol{I}) = \frac{1}{M_l} \sum_{k}^{M_l} F^l_{ik}(\boldsymbol{I}) F^l_{jk}(\boldsymbol{I}). \tag{1}$$

Here, M_l is the dimentionality of the responses of each of the filters in the lth layer of the CNN.

The difference of this definition from [Gatys et al. 2016] is the normalization factor $1/M_l$. In the previous definition of the texture feature, because the normalization factor $1/M_l$ was absent, the magnitude of the texture feature was dependent on the image size, as $G^l_{ij}(\boldsymbol{I})$ is a summation of the filter responses along the pixels k in the CNN filter response. Our definition of G^l_{ij} normalizes the feature with the image size $1/M_l$, and thus we are able to compare G^l_{ij} of different images $\boldsymbol{I}$. Each element of $G^l_{ij}(\boldsymbol{I})$ then describes a statistical amount that could be viewed as the frequency of a certain local feature among the entire image, or the co-occurence of certain pairs of local features among the entire image.

2.3 Linear Combinations of the Texture Feature

Now, let $\mathcal{I} = \{\boldsymbol{I}_n\}^N_{n=1}$ be a set of reference images, and $\cdot \circ \cdot$ be the concatenation of two images. Then, the texture feature of the concatenated image $\boldsymbol{I}_1 \circ \boldsymbol{I}_2 \circ \cdots \circ \boldsymbol{I}_N$ is written as

$$\begin{aligned} & G^l_{ij}(\boldsymbol{I}_1 \circ \boldsymbol{I}_2 \circ \cdots \circ \boldsymbol{I}_N) \\ &= \frac{1}{M_{l,all}} \sum_{k=1}^{M_{l,all}} F^l_{ik}(\boldsymbol{I}_1 \circ \boldsymbol{I}_2 \circ \cdots \circ \boldsymbol{I}_N) F^l_{jk}(\boldsymbol{I}_1 \circ \boldsymbol{I}_2 \circ \cdots \circ \boldsymbol{I}_N) \\ &\approx \frac{1}{M_{l,all}} \sum_{n=1}^{N} \sum_{k=1}^{M_{l,n}} F^l_{ik}(\boldsymbol{I}_n) F^l_{jk}(\boldsymbol{I}_n) \\ &= \sum_{n=1}^{N} \frac{M_{l,n}}{M_{l,all}} G^l_{ij}(\boldsymbol{I}_n). \end{aligned} \tag{2}$$

Here, $M_{l,n}$ is the number of pixels for each filter response for the nth image, and $M_{l,all} = \sum_n M_{l,n}$. The approximation on the second line is due to the image padding that appears in the calculation of the convolution. When calculating $G^l_{ij}(\boldsymbol{I}_{all})$, the window of the convolution could lie across different images, which in $G^l_{ij}(\boldsymbol{I}_n)$ corresponds to the image padding. Thus, the approximation becomes strictly equal in the limit where the padding size becomes negligible against the total image size.

Therefore, by considering only large images as the input, we can use Eq. (2) to say that G is linear against image concatenation (Figure 4). Using this fact, for every weighted linear combination of G, such that the weight is rational number, there exists an image that realizes its value of the texture feature.

2.4 The Style Model

Preparing the Texture Space In the proposed method, we use linear combinations of G to construct a representation of the desired style in the texture feature space. Since arbitrary concatenations of images $\boldsymbol{I}_n$ drawn in a given style also belong to the same style as well, any linear combinations of the texture feature of these images are also a valid texture features that belong to the given style. In other words, any point in the following closed region

$$\left\{ r_n G^l_{ij}(\boldsymbol{I}_n) \middle| \sum_{n=1}^{N} r_n = 1,\ 0 \le r_n \le 1 \quad (n = 1, \cdots, N) \right\} \tag{3}$$

could be chosen as a valid texture feature that belongs to the desired style. We define this closed region in the feature space as the "style space".

As mentioned in the introduction, each instance of an artwork created in a certain style contains two types of features: (1) the features common in all of the images that share the style, and (2) features that are special in the certain instance of the artwork. The restriction of the choice of texture features within the style space corresponds to (1), and the degrees of freedom of the textures within the style space corresponds to (2).

If a sufficient number of texture reference images is collected, the method could synthesize a new texture that absorbs the differences of each sample, appropriate for the input image.

Estimating the Appropriate Texture to Transfer Since we have obtained a set of texture features that represent an instance of an artwork created in a certain style, the next task is to find the appropriate texture to render a given input image. Let I_{content} be the input content image. Since I_{content} is not drawn in the target style, $G(I_{\text{content}})$ does not necessarily lie within the representation of the desired style. We estimate the appropriate texture within the desired style representation by finding the closest point in the style representation from $G(I_{\text{content}})$:

$$\begin{aligned} &\arg\min_{\boldsymbol{r}} \sum_{l,i,j} \left(G^l_{ij}(\boldsymbol{I}_{\text{content}}) - \sum_{n=1}^{N} r_n G^l_{ij}(\boldsymbol{I}_n) \right)^2 \\ &\text{s.t.} \sum_{n=1}^{N} r_n = 1,\ 0 \le r_n \le 1 \quad (n = 1, \cdots, N). \end{aligned} \tag{4}$$

Here, $\boldsymbol{r} = [r_1, \ldots, r_n]$ is the blending ratio of the texture images. This optimization problem could be easily transformed into a quadratic programming problem of $\boldsymbol{r}$. We solved the optimization problem using CVXOPT [Andersen et al. 2012], a Python package for convex optimization.

Texture Transfer Using the optimized weight $\boldsymbol{r}$, we construct a linearly weighted texture feature,

$$\tilde{G}^l_{ij} = \sum_{k=1}^{K} r_k G^l_{ij}(\boldsymbol{I}_k), \tag{5}$$

where K is the number of images in the reference texture images. We then apply the method of Gatys et al. [Gatys et al. 2016] for texture transfer. We incorporate the estimated texture feature into the texture loss term, i.e., the second term in the loss function $\mathcal{L}$, and modify it as

$$\begin{aligned} \mathcal{L} &= \alpha \sum_{i,j} \frac{1}{2N_l} \left(F^l_{ij}(\boldsymbol{I}) - F^l_{ij}(\boldsymbol{I}_{content}) \right)^2 \\ &+ \beta \sum_{l} \sum_{i,j} w_l \left(G^l_{ij}(\boldsymbol{I}) - \tilde{G}^l_{ij} \right)^2. \end{aligned} \tag{6}$$

Here, w_l is the scalar weight for each CNN layer used to construct the texture feature. The difference here is that the synthesized texture feature is used in place of the texture feature of the input texture image. We then use probabilistic gradient descent to find an image $\boldsymbol{I}$ that gives the local optimum of $\mathcal{L}$. The obtained image $\boldsymbol{I}$ becomes the final output image, where the estimated texture is transferred to the input image.

3 The Dataset

As mentioned in the introduction, learning a specific style of artwork requires a large collection of annotated image data. Although Wikipaintings [Karayev et al. 2014] is a dataset that contains 100,000 high-art images, to our knowledge, there is no dataset that contains such a number of digital paintings.

We provide a new dataset, the nico-illust dataset, for analyzing the style of modern-style illustrations. The dataset includes 500,000 images largely consisting of modern-style illustrations. All images in this dataset are in illustration-sharing service "Niconico Seiga" [2]. We obtained permission from the authors to use them for academic research purposes. Each image is associated with annotations such as the author id, tags.

Among these annotations, our proposed method mainly uses the author ids and tags. The tags that are annotated include information such as the author, the motif, and the style of the image (watercolor, pencil, acrylic, etc.). It is also noteworthy that there are many examples where the same motif is drawn by various artists.

4 Result and Discussion

Application to Images Figure 1 shows an example of our method applied to various pairs of input images and style queries. The choice of style queries consists of 4 authors with IDs 33341043, 10195867, 23607472, and 13762409, and two styles, watercolor and pixel art. The input images were chosen from the nico-illust dataset as well. To construct the feature space, we chose the VGG model proposed by Simonyan et al. [Simonyan and Zisserman 2015]. In the figure, we ans see that the algorithm (1) does not corrupt the color of the original image, and (2) effectively captures and transfers the queried style. For example, for the user 23607472, the algorithm effectively captures both the pointillism-like style and the colors of the artist.

Application to Videos We have also applied our method to each frame of a video using "watercolor" as the style query. Figure 5 shows the plot of the series of weights of the 50 texture images in the texture image set, and the two images with the largest weights. The results show that the algorithm effectively chooses texture images that describes the input frame. The results are available in the supplemental material.

5 Conclusion

We have presented a system that learns a desired style of artwork from a collection of images, and transfers the style to an arbitrary image. Our novel illustration dataset contains 500,000 images, consisting mainly of digital paintings annotated with rich text information. We constructed our system using this presented dataset. Experimental results show that our system is capable of effectively transferring various styles, and our system is capable of smoothly transferring styles to videos as well.

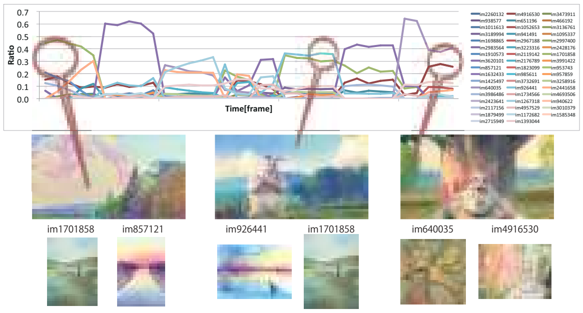

Figure 5: *Series of texture blending weights across the Big Buck Bunny ©Blender Foundation, 2008.*

Acknowledgements

We would like to thank all of the illustrators for their inspiring artwork, which has motivated us for this research, and was essential for creating the dataset. We would also like to thank Yusuke Matsui for the valuable suggestions and fruitful discussions.

References

ANDERSEN, M. S., DAHL, J., AND VANDENBERGHE, L., 2012. Cvxopt: A python package for convex optimization. http://cvxopt.org/.

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2015. Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*, 262–270.

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2016. Image style transfer using convolutional neural networks. In *IEEE Computer Vision and Pattern Recognition*.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *ACM SIGGRAPH*, 327–340.

KARAYEV, S., TRENTACOSTE, M., HAN, H., AGARWALA, A., DARRELL, T., HERTZMANN, A., AND WINNEMOELLER, H. 2014. Recognizing image style. In *British Machine Vision Conference*.

SIMONYAN, K., AND ZISSERMAN, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

YAMAGUCHI, S., FURUSAWA, C., KATO, T., FUKUSATO, T., AND MORISHIMA, S. 2015. Bgmaker: Example-based anime background image creation from a photograph. In *ACM SIGGRAPH Posters*.

[2] http://seiga.nicovideo.jp/