Supplementary Material for "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: *Relationship between the number of reference images and output image quality. Each row shows the style transfer results for the same author, and each column shows the number of reference images used for style transfer. The number of reference images used in each columns are 1, 5, 10, 30, and 50 images, starting from the left.*

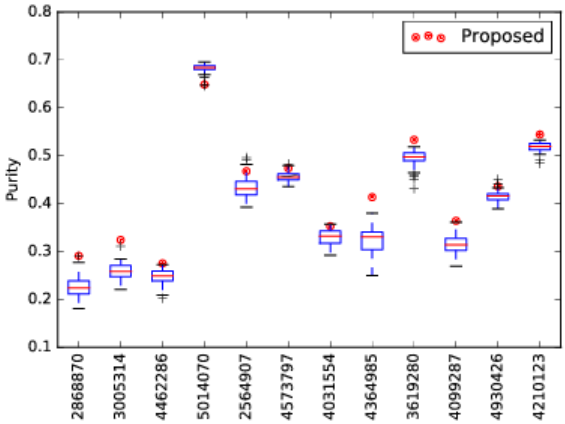

Figure 2: *Color correspondence evaluation results.*

1 Relationship Between the Number of Reference Images and Output Quality

Here we study the differences of the output image quality when the number of reference images are changed. 3 keywords, "sunset," "spring," and "night sky" were chosen as the style keyword input. We then let the number of reference images to be used for constructing the style space, to 1, 5, 10, 30, and 50 images. We performed our algorithm with each of these numbers of reference images and observed the output results. Note that, only one reference image, our method is equivalent with the style transfer algorithm by Gatys, et al. [2016].

Figure 2 shows the results of the experiment. The results of 10 reference images has a largely different color from the original image. The color discrepancy from the original image becomes smaller as the number of images increase. Still, we are able to observe the different styles of each of the authors.

2 Quantitative Evaluation of Color Correspondence

For quantitative evaluation of the color correspondence of the output image, we constructed an evaluation factor of a style transfer result based on pixel clustering as follows.

First, for the input content image $\boldsymbol{I}_{\mathrm{in}}$ and the transferred output image $\boldsymbol{I}_{\mathrm{out}}$, we perform pixelwise clustering in $\mathrm{L^*a^*b^*}$ color space using the k-means method. The number of clusters were set to 20. Let $\mathcal{C}_{\mathrm{in}}$, $\mathcal{C}_{\mathrm{out}}$ be the clustering results of the content image and the transferred image, respectively. Using these clustering results, we evaluate the color correspondence of the output image using purity [Weyand et al. 2012], a evaluation factor of clustering results. Let p be the purity of $\mathcal{C}_{\mathrm{out}}$ when $\mathcal{C}_{\mathrm{in}}$ is taken as the ground truth of the clustering results, where $p_{\mathrm{out}} \in [0, 1]$. In images such that $p = 1$, every pixel that belong to the same color cluster in the original image, also belong to the same color cluster in the output image as well. Therefore, p could be used as an evaluation index of color correspondence preservation, where larger p indicates a better preservation of color correspondence.

To compare our method with [Gatys et al. 2016], we calculated p for (1) our result using "watercolor" as the style query, and (2) using each of the texture images in the "watercolor" texture image set with [Gatys et al. 2016].

Figure 1 shows the results of the evaluation. The blue plot shows the distribution of p for each input texture image by Gatys et al. [2016]. The red plot shows the result of our method. The evaluation results indicate that the proposed method preserves the color correspondence of the image better than Gatys et al. [2016].

3 Implementation Details

Our algorithm mainly consists of the texture estimation part, which is the novelty of this paper, and the texture transfer part, which performs the style transfer. The entire algorithm was mainly implemented using Chainer, a Python framework for machine learning [Tokui et al. 2015]. For the texture transfer part, we use the Adam optimizer [Kingma and Ba 2015] to find the optimal image. For texture feature extraction, we use a modified version of the VGG model [Simonyan and Zisserman 2015], where the padding from every convolution layer is removed. This modification helps removing the artifacts in the resulting images. The number of iterations for the texture transfer process was set to 1000 for every image in the main paper and the supplementary material. The computation time for image generation was about 6 minutes, on a single machine using a GTX Titan X GPU.

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

4 Further Comparison with Previous Studies

Here we show further comparisons of our algorithm and Gatys, et al. [Gatys et al. 2016]. Figure 3 compares the output images of our algorithm with Gatys, et al. [Gatys et al. 2016]. From Figure 3 we see that the results of Gatys, et al. [Gatys et al. 2016] tends to modify the color from the original image, while our algorithm preserves the color of the original image.

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. 2016. Image style transfer using convolutional neural networks. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

KINGMA, D., AND BA, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*.

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

TOKUI, S., OONO, K., HIDO, S., AND CLAYTON, J. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

WEYAND, T., HOSANG, J., AND LEIBE, B. 2012. An evaluation of two automatic landmark building discovery algorithms for city reconstruction. In *Proceedings of the 11th European Conference on Trends and Topics in Computer Vision*, 310–323.

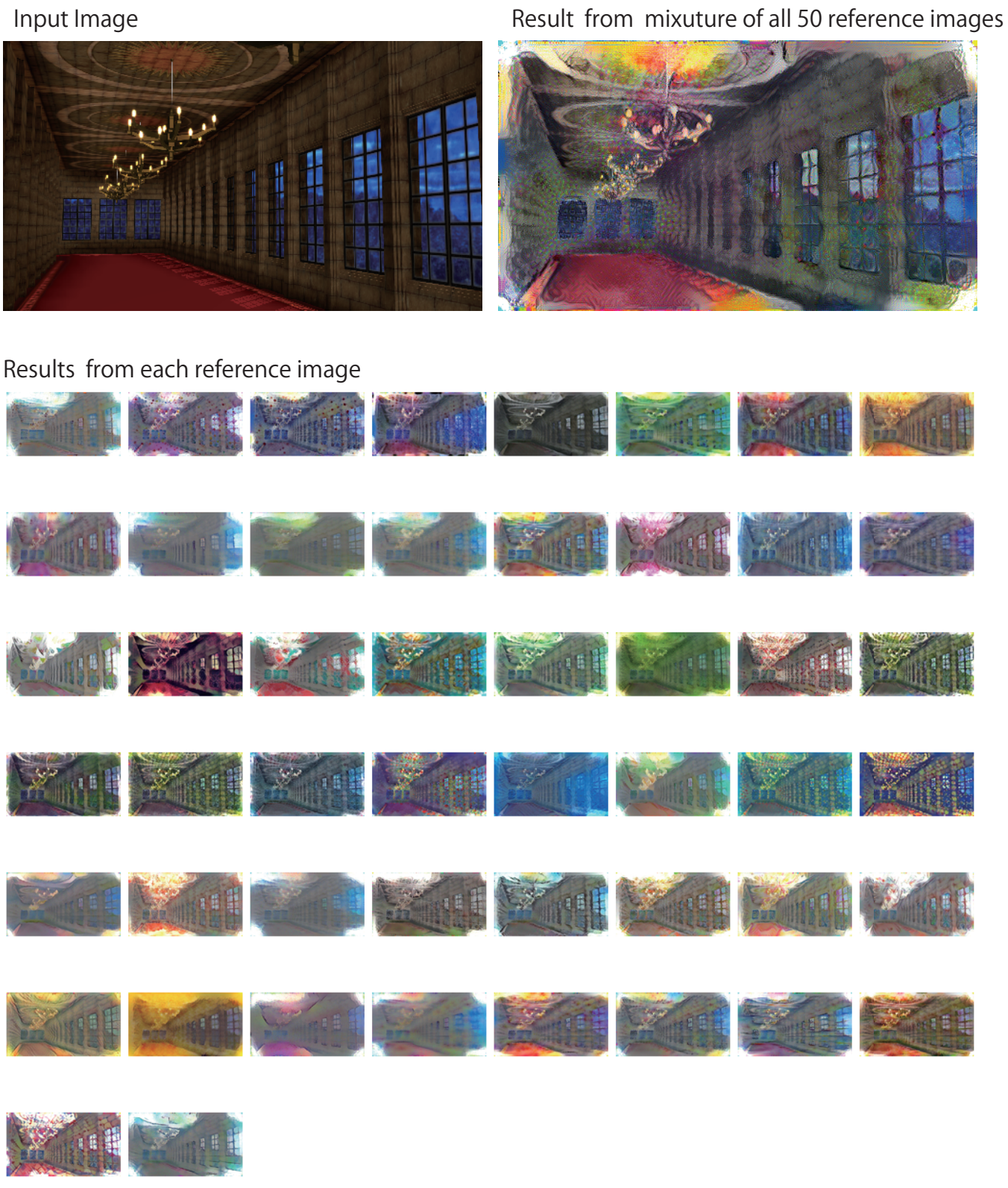

Figure 3: *Comparison of our algorithm with Gatys, et al. Image id is 2868870, shown as first column in Figure 2.*