Zero-shot Font Style Transfer with a Differentiable Renderer

Kota Izumi
The University of Electro-Communications, Tokyo, Japan
izumi-k@mm.inf.uec.ac.jp

Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
yanai@cs.uec.ac.jp

ABSTRACT

Recently, a large-scale language-image multi-modal model, CLIP, has been used to realize language-based image translation in a zero-shot manner without training. In this study, we attempted to generate language-based decorative fonts for font images using CLIP. By the existing image style transfer methods using CLIP, stylized font images are usually only surrounded by decorations, and the characters themselves do not change significantly. On the other hand, in this study, we use CLIP and vector graphics image representation using a differentiable renderer to achieve a style transfer of text images that matches the input text. The experimental results show that the proposed method transfers the style of font images to match the given texts. In addition to text images, we confirmed that the proposed method was also able to transform the style of simple logo patterns based on the given texts.

CCS CONCEPTS

• **Computing methodologies** → **Appearance and texture representations**.

KEYWORDS

font style transfer, neural style transfer, differentiable renderer, CLIP, large-scale text-image model

ACM Reference Format:
Kota Izumi and Keiji Yanai. 2022. Zero-shot Font Style Transfer with a Differentiable Renderer. In *ACM Multimedia Asia (MMAsia '22), December 13–16, 2022, Tokyo, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3551626.3564961

1 INTRODUCTION

In recent years, a method of image style transfer using CLIP [9], a text-image multi-modal embedding model, has been proposed for zero-shot text-based image style transfer. However, when CLIPstyler [5], an existing style transfer method using CLIP, is applied to font images, only the color around the text changes or decorations appear, as in Figure 1 "CLIPstyler", but the font region itself does not change significantly.

To solve this problem, in this study, we propose a method to realize a style transfer of character font images by combining a method to optimize the parameters of Bézier curves using a differentiable renderer [7] with a text-based style transfer using CLIP.

MMAsia '22, December 13–16, 2022, Tokyo, Japan

ACM ISBN 978-1-4503-9478-9/22/12…$15.00
https://doi.org/10.1145/3551626.3564961

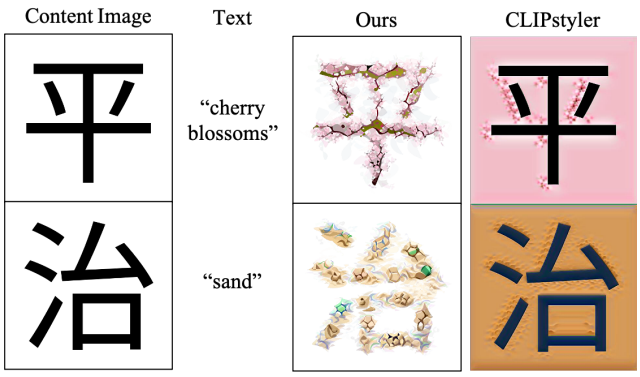


Figure 1: Comparison between the proposed method and CLIPstyler. In CLIPstyler, the contours of the letters did not change. In our results, the contours changed according to the style texts.

The proposed method employs the loss function proposed by CLIPstyler [5]. In addition, we introduce the distance transform loss which Atarsaikhan *et al.* [1] used to successfully change the font style of only the character parts.

We confirmed that the proposed method successfully transformed the character itself into a style that follows the given text, as shown in Figure 1 "Ours". The proposed method is also applicable to black-and-white binary logo images and can transform the style of a logo of arbitrary shape.

2 RELATED WORKS

Unlike ordinary neural image style transfer, style transfer on font images requires the transfer of the style only to the character part. Atarsaikhan *et al.* [1] introduced distance transform loss into neural style transfer [4], and succeeded in performing style transfer while preserving the shape of text and logo patterns. In Typography with Decor [11], they proposed a deep neural network for transferring the style of decorated text images, and Shape Matching GAN [12] can perform style transfer of text images by learning a single style image.

Recently, a method to manipulate images by arbitrary texts in a zero-shot training manner has been proposed using CLIP published by OpenAI [9]. Since CLIP was trained with four hundred million image-text pairs, CLIP enables image translation without training, that is, zero-shot image translation.

StyleCLIP [8] combines StyleGAN [6] and CLIP to manipulate images by text. StyleGAN-NADA [3] can transform a domain image into a new domain based on text without adding domain images. CLIPstyler [5] allows the transfer of text-matched styles to an input image.

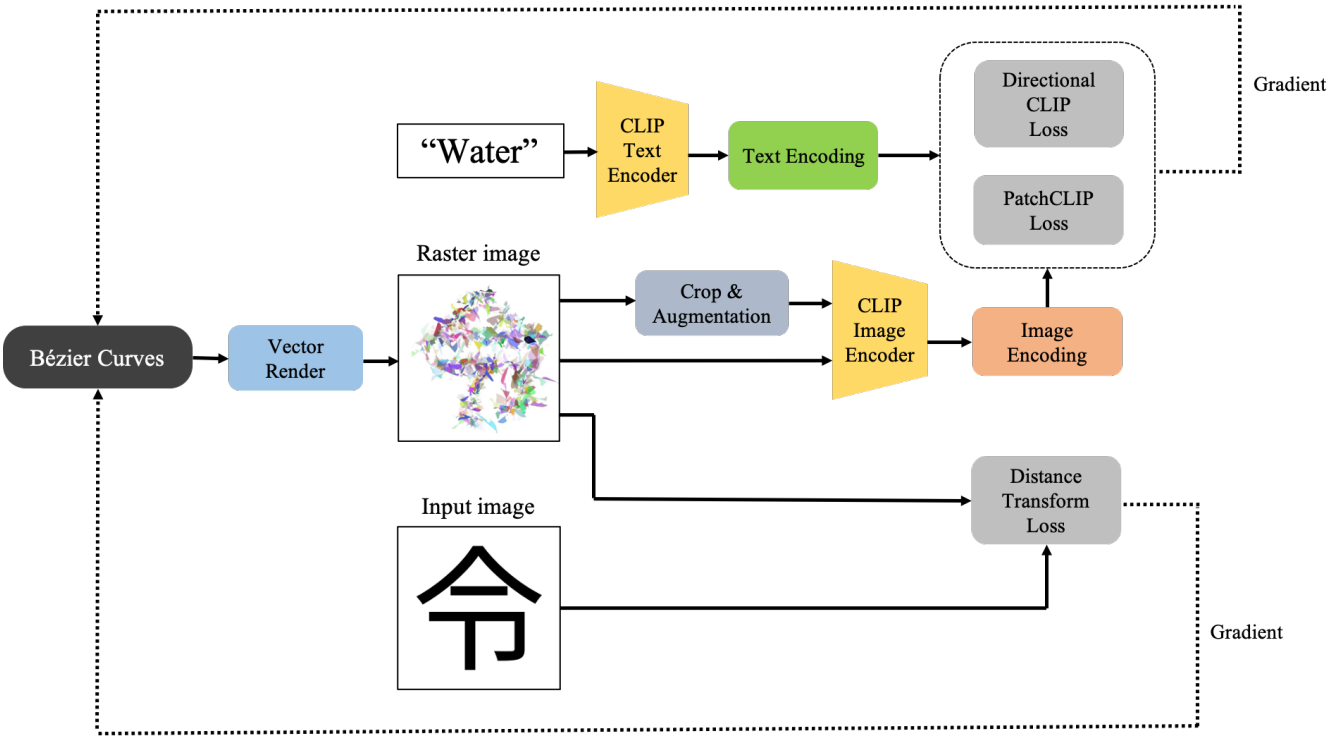


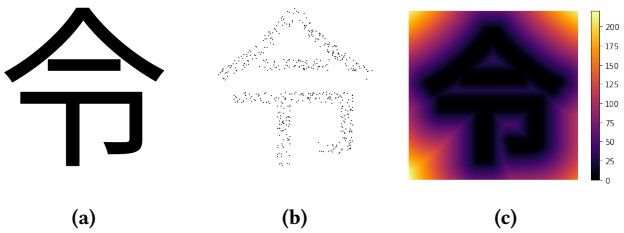
Figure 2: Overview of the proposed method.

CLIP and Vector Graphics have also been studied in combination; CLIPDraw [2] combines a differentiable renderer [7] and CLIP to generate a drawing from text alone by optimizing the Bézier curve parameters. StyleCLIPDraw [10] can transfer the style of a style image to a drawing generated by CLIPDraw.

Based on these studies, we combine CLIPDraw [2] and CLIPstyler [5] to achieve natural style transfer by arbitrary texts and then introduce distance transform loss used by Atarsaikhan *et al.* [1] to achieve style transfer of only the font regions.

3 METHOD

The proposed method is based on the work of Li *et al.* [7] who introduced a differentiable renderer for vector graphics, and CLIPDraw [2], which combines the differentiable renderer of Li *et al.* [7] with CLIP [9]. A schematic diagram of the method is shown in Figure 2. In the method, stylized fonts are represented by Bézier curves with control point positions, colors, and line thicknesses as parameters, instead of raster images represented by pixels. The parameters of the Bézier curves are then optimized to generate stylized fonts. During optimization, Bézier-based vector representation is converted to a raster image using a differentiable renderer. Then, after augmentation, the loss is calculated using image and text encoders of CLIP. The loss calculation is based on the loss used in CLIPstyler. In addition, to preserve the shape of the text, the distance transform loss introduced by Atarsaikhan *et al.* [1] for font style transfer is also used.



(a) (b) (c)

Figure 3: (a) Input image (b) The initial positions of Bézier curves. Control points are distributed near these points. (c) The distance transform map. The further away from the text area, the larger the value.

3.1 Font Representation

Stylized fonts are represented by a set of differentiable RGBA closed Bézier curves based on the method of Li *et al.* [7]. Each curve is represented by a line thickness, position of control points, color, and opacity. Closed Bézier curves have three to five segments per shape, and the interior is filled. While optimizing, the number of curves and control points are fixed, and only the positions of control points, line thickness, color, and opacity are optimized by the gradient descent method. The initial control points are placed around the points randomly sampled by the number of strokes from the coordinates of the black area in an input font image, as shown in Figure 3b. This is because if a random position is used as the initial position, Bézier curves will be drawn even on the background.

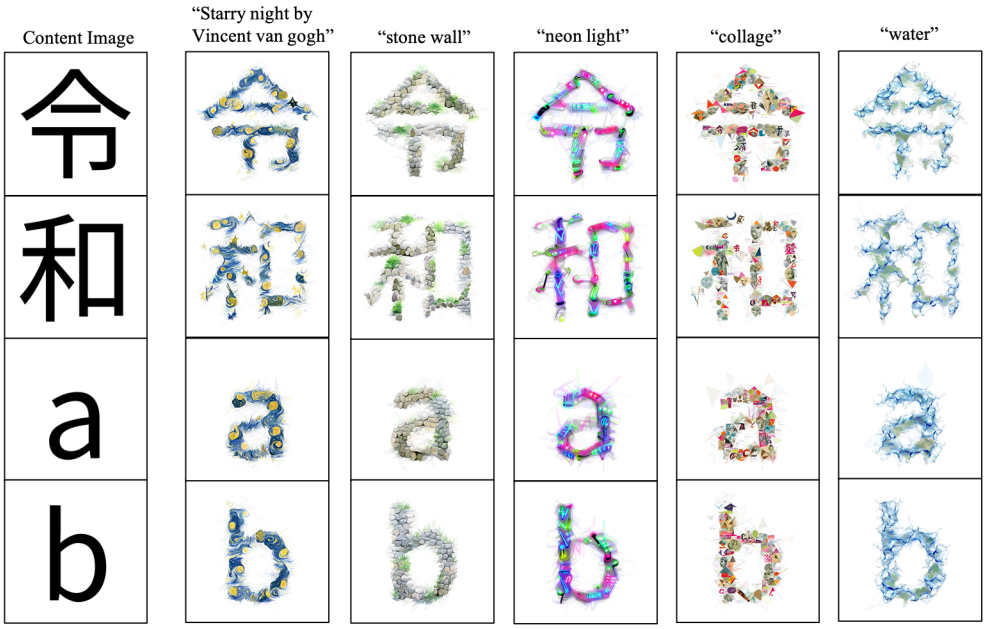


Figure 4: Results of font style transfer using our method.

3.2 Directional CLIP loss

CLIPstyler [5] introduced the directional CLIP loss proposed in StyleGAN-NADA [3] to stabilize optimization and improve output image quality. We employ this loss function in our method. The directional CLIP loss can be defined as:

$$\Delta T = E_T(T_{sty}) - E_T(T_{src})$$
$$\Delta I = E_I(I_{draw}) - E_I(I_c)$$
$$L_{dir} = 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I||\Delta T|}, \tag{1}$$

where I_c is an input image and I_{draw} is the image rendered by a differentiable renderer. E_I and E_T denote the image and text encoders of CLIP, respectively. T_{sty} is an input text which represents the style to which the input font image is transferred, while T_{src} is the text which is expected to represent non-stylized images. For T_{src}, we tried "a photo of text", "letters", and "logo" in addition to "a photo" used in CLIPStyler [5]. Since we found that the results are not so different from any of these four texts in the preliminary experiments, we used "a photo" in the same way as CLIPstyler in the experiments.

3.3 PatchCLIP loss

CLIPstyler [5] introduced a new loss for texture transfer, Patch CLIP loss, inspired by Gatys *et al.* [4] and Frans *et al.* [2]. We adopt this loss in our method as well. Specifically, a sufficient number of raster images are cropped from a raster image rendered by a differentiable renderer, and random perspective augmentation is performed on the cropped patches to calculate the directional CLIP loss. A threshold τ is set to reject gradient optimization for high-scored patches. Thus, PatchCLIP loss can be defined as:

$$\Delta T = E_T(T_{sty}) - E_T(T_{src})$$
$$\Delta I_i = E_I(\hat{I}^i_{draw}) - E_I(I_c)$$
$$l^i_{\text{patch}} = 1 - \frac{\Delta I_i \cdot \Delta T}{|\Delta I_i||\Delta T|}$$
$$L_{\text{patch}} = \frac{1}{N}\sum_i^N R(l^i_{\text{patch}}, \tau) \tag{2}$$
$$\text{where } R(s,\tau) = \begin{cases} 0, & \text{if } s \le \tau \\ s, & \text{otherwise} \end{cases}$$

and $\hat{I}^i_{draw}$ is the i-th patch randomly cut from the rendered raster image.

3.4 Distance Transform loss

When rendering a font image, the background should remain white like the input image. If the shape of the stylized font is too distorted, the font will be difficult to read, which is not desirable. To solve this problem, Atarsaikhan *et al.* [1] used distance transform loss for font style transfer. We also employ the distance transform loss. To calculate this loss, first, a distance transform is applied to an input font image to create a distance transform map. The distance transform map is then multiplied by the input and output images, and the mean square error is calculated. As shown in Figure 3c, the distance transformation sets the value of the silhouette pixel in the black-and-white image to 0, and the farther away from the silhouette pixel, the larger the value. Distance transform loss can be defined as:

$$L_{\text{distance}} = \frac{1}{2}\left(I_c \circ I_d - I_{draw} \circ I_d\right)^2, \tag{3}$$

where I_d denotes the distance transform map.

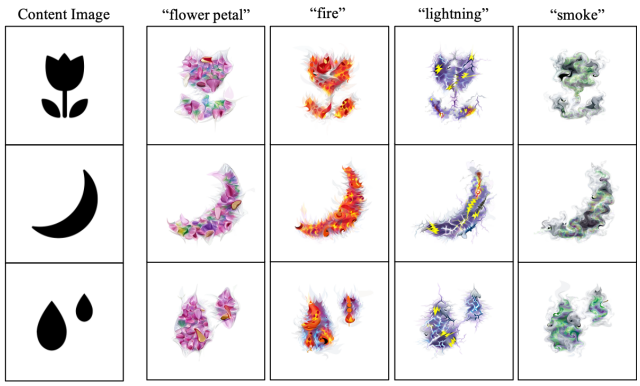


Figure 5: The results on simple logos.

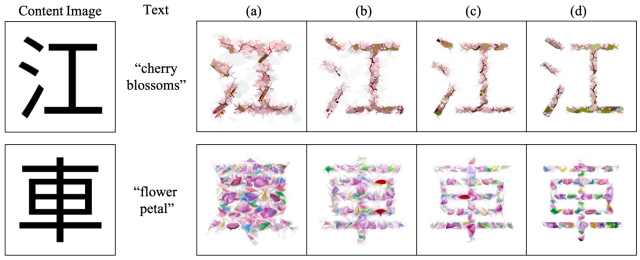


Figure 6: The results with different weights of distance transform loss, $\lambda_{\text{distance}}$. (a) $\lambda_{\text{distance}} = 1.5{\times}10^2$. (b) $\lambda_{\text{distance}} = 1.5{\times}10^3$. (c) $\lambda_{\text{distance}} = 1.5 \times 10^4$. (d) $\lambda_{\text{distance}} = 1.5 \times 10^5$.

3.5 Total loss

In addition to the above three losses, the total variation regularization loss L_{tv} is introduced as in CLIPstyler [5] to reduce side artifacts. Thus, the loss during optimization is defined as:

$$L_{\text{total}} = \lambda_d L_{\text{dir}} + \lambda_p L_{\text{patch}} + \lambda_{distance} L_{\text{distance}} + \lambda_{tv} L_{\text{tv}} \quad (4)$$

4 EXPERIMENTAL RESULTS

4.1 Experimental settings

The resolution of the input image is 512×512. A distance transform map is generated using the function in OpenCV library. The weights λ_d, λ_p, $\lambda_{distance}$, and λ_{tv} are set to 5×10^2, 9×10^3, 1.5×10^3, 2×10^{-3}, respectively. The weight of the distance transform loss is set to a large value so that even complex fonts do not collapse. The optimizer is Adam, and the learning rates are set to 1, 1×10^{-1}, and 1×10^{-2} for the position, line thickness, and color of the control point, respectively. The number of iterations is 200. The size of the cropped image for the PatchCLIP loss is selected to be 160×160, which indicates the best quality. The number of cropped images is 64. The function in PyTorch libraries is used for perspective augmentation. The threshold τ is set to 0.7. As in CLIPstyler [5], we use the prompt engineering method proposed by Radford *et al.* [9]. The training time is about 100s per image on the RTX2080Ti.

4.2 Qualitative evaluations

Some results are shown in Figure 4. It can be seen that font images are successfully stylized so that the style of the rendered images matched the given style texts. It also shows that the shape of the stylized fonts keeps the shape of the input fonts because of using the distance transform loss. Figure 5 shows the results when simple logos are used as input images. It can be seen that the style can be transferred not only to font images but also to simple silhouette images.

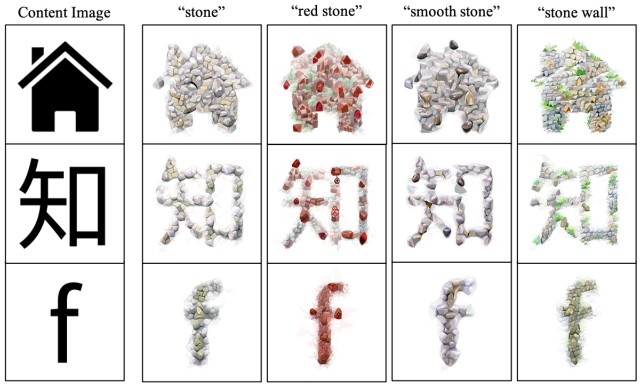


Figure 7: The results with different adjectives.

4.3 Weight change of Distance Transform Loss

Figure 6 shows the results when applying different weights to the distance transform loss. The larger the weight, the less the letters protrude from the contour. Furthermore, in case (d), the letter thickness is closer to that of the input image. If the input image is of more complex characters, the shape of the characters can be preserved by increasing the weight.

4.4 Effect of adjectives

The results in the case of adding different adjectives to "stone" are shown in Figure 7. When the text is "red stone", the color of the stone changes according to the text. When the adjective "smooth" is added, it is clear that it expresses a smoother texture than "stone". Furthermore, "stone wall" shows that unlike "stone", a plant appears in the image.

5 CONCLUSIONS

In this paper, we proposed a method for zero-shot text-based font style transfer. The proposed method combined a differentiable renderer [7], the loss proposed in CLIPStyler [5] and Distance transform loss to preserve a font shape. The generated images have been stylized with the style represented by input style texts while preserving the shape of the input fonts. The method was also confirmed to be effective for simple logos.

The proposed method is at an initial stage of development, and further improvements are planned. Future work includes comparisons with existing methods and quantitative evaluations such as user studies.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Numbers, 21H05812, 22H00540, 22H00548, and 22K19808.

REFERENCES

[1] Gantugs Atarsaikhan, Brian Kenji Iwana, and Seiichi Uchida. 2018. Contained Neural Style Transfer for Decorated Logo Generation. *In 13th IAPR International Workshop on Document Analysis Systems* (2018).

[2] Kevin Frans, LB Soros, and Olaf Witkowski. 2021. CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843* (2021).

[3] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. 2021. StyleGAN-NADA: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946* (2021).

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. *Proc.of IEEE Computer Vision and Pattern Recognition* (2016), 2414–2423.

[5] Kwon Gihyun and Ye Jong Chul. 2022. CLIPStyler: Image style transfer with a single text condition. *Proc.of IEEE Computer Vision and Pattern Recognition* (2022), 18062–18071.

[6] Tero Karras, Samuli Laine, , and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. *Proc.of IEEE Computer Vision and Pattern Recognition* (2019), 4401–4410.

[7] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable Vector Graphics Rasterization for Editing and Learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39, 6 (2020), 193:1–193:15.

[8] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249* (2021).

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Jack Clark Pamela Mishkin, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).

[10] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. 2021. StyleCLIPDraw: Coupling Content and Style in Text-to-Drawing Translation. *arXiv preprint arXiv:2202.12362* (2021).

[11] Wenjing Wang, Jiaying Liu, Shuai Yang, and Zongming Guo. 2019. Typography with decor: Intelligent text style transfer. *Proc.of IEEE Computer Vision and Pattern Recognition* (2019), 5889–5897.

[12] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. 2019. Controllable artistic text style transfer via shape-matching GAN. *Proc.of IEEE International Conference on Computer Vision* (2019), 4442–4451.