Performance Optimization of Neural Style Transfer for Animated Characters in Real-Time Rendering

Katsushi Suzuki, Takeshi Okuya, Misumi Hata
{katsushi.suzuki,takeshi.okuya,misumi.hata}@delightworks.co.jp

Research & Development Office,
DELiGHTWORKS Inc.

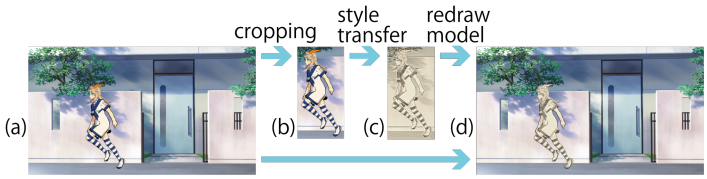

Figure 1: Overall Processing Flow of the Proposed Method.

Layer	Layer Size	Stride	Output Size
Encoder			
Input			3×1920×1080
Conv + InsNorm + ReLU	8×9×9	1	8×1920×1080
Conv + InsNorm + ReLU	16×3×3	2	16×960×540
Conv + InsNorm + ReLU	32×3×3	2	32×480×270
(Res + InsNorm + ReLU)×2	32×3×3	1	32×480×270
Decoder			
Up-sample		1/2	32×960×540
Conv + InsNorm + ReLU	16×3×3	1	16×960×540
Up-sample		1/2	16×1920×1080
Conv + InsNorm + ReLU	8×3×3	1	8×1920×1080
Conv + Tanh	3×9×9	1	3×1920×1080

Table 1: Architecture of the proposed model

1. Introduction: The study of neural style transfer, which extracts the style of a picture and transfers it to another one by deep learning, was first proposed by Gatys et al. [5]. In recent years, inferring neural networks in game engines such as Unity has become more accessible. We believe that the demand for neural style transfer is increasing as one of the post-effects processing in real-time CG. However, there is a high load problem in processing neural style transfers during real-time rendering. Therefore we focused on an animation technique in which the foreground and background created separately, moving only the characters in the foreground. In this paper, we propose an optimization method for real-time rendering on a game engine, focusing on the style transfer to the animated characters in the foreground. There are two proposed methods: the technique to crop moving character in the rendered image and the optimization by shaping up ReCoNet proposed by Gao et al. [4], which is a time-consistent style transfer model.

2. Related Work: The feed-forward perceptual loss model proposed by Johnson et al. [7] achieved the best quality in image style transfer. However, when the model is applied to videos, temporal flickers appear. Anderson et al. [1] introduced a time loss function to assure temporal consistency, but it takes several minutes to process each frame. Hence, it is not suitable for real-time rendering. Methods using optical flow (Chen et al.[2][3]), inputting additional previous frames that have been style transferred (Gupta et al. [6]), and introducing new time loss methods at the feature map level [4] have been proposed as style transfer methods for videos. Though they solved the time consistency problem, further work is needed for adopting it to full HD resolution in real-time rendering.

3. Method: Figure 1 [1] shows the overall processing flow of the proposed method. First, we render the entire scene containing the target of the style transfer using normal shaders (Fig.1a). Next, we crop the region containing the target character (Fig.1b). Then, we apply a style transfer model to this cropped region (Fig.1c). Finally, the character is re-rendered using the result of the style transfer as a texture, resulting in an image with style transfer applied to a specific model (Fig.1d).

3.1. Region extraction for an animated character: In this step, only the region of the character is extracted from the entire rendered screen image. This makes the computation faster than inputting the entire screen image into the neural network in the next step. To extract the character regions, bone positions are used. By calculating 2D screen position of all the bones and obtaining the maximum and minimum values in each direction, the rectangle containing the character is obtained. There are two advantages to using bones for region extraction. Firstly, skin meshes are always calculated its bone 3D positions every frame. Therefore, it is only necessary to convert them to 2D screen position by matrix multiplication. Secondly, it is not necessary to scan the entire image to extract the region.

[1] In the figures in this paper, the 3D model was used under the terms of the Unity-Chan license. (c) UTJ/UCL https://unity-chan.com/contents/license_jp/

Figure 2: left: original rendering result. center: results of style transfer of the ReCoNet. right: Optimized ReCoNet (ours).

	full screen	cropped
ReCoNet [4]	383.43 (case1)	134.59 (case2)
ReCoNet Optimised	38.29 (case3)	12.62 (case4:ours)

Table 2: Processing time of inference on the GPU[ms]. The measurement processing environment is as follows. CPU: Intel Core i9-9980XE / GPU: Nvidia RTX 2080 Ti / Resolution: 1920×1080 / Rendering: Unity 2019.3.0f1, Barracuda 1.0.0.

3.2. The performance-optimized ReCoNet model: The architecture of the proposed model is shown in Table 1: We achieved to scale down the size of the model by reducing the number of Convolution layers, the number of channels, and the number of Residual layers, reducing the size of the model from 12,113 KB in the original version to 214 KB in the reduced version.

4. Result and Conclusion: Table 2 shows the results of measuring the inference processing time on the GPU for four cases. Case 4 with our proposed method shows the lowest processing load. As shown in figure 2, we believe that the optimized ReCoNet model can be qualitatively transferred. These results show that the proposed method is effective for applying style transfer to animated characters. One of the limitations of the proposed method is that it only considers models with bones, such as humanoid models. In addition, due to the constraint of Barracuda, if the size of the conversion region is changed, the processing stops for several frames to synchronize between CPU and GPU, so the region size is fixed to the initial value and only the position of the rectangle is computed per frame. The neural network model is limited in its ability to express styles. In the future, we would like to investigate approaches to optimize performance when applying style transfer to multiple characters and specific regions.

[1] Alexander G. Anderson, Cory P. Berg, Daniel P. Mossing, and Bruno A. Olshausen. Deepmovie: Using optical flow and deep neural networks to stylize movies, arXiv:1605.08153, 2016.

[2] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua. Coherent online video style transfer. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1114–1123, 2017.

[3] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stereoscopic neural style transfer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6654–6663, 2018.

[4] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. Reconet: Real-time coherent video style transfer network. In *Computer Vision – ACCV 2018*, pages 637–653, 2019.

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

[6] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei. Characterizing and improving stability in neural style transfer. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4087–4096, 2017.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision – ECCV 2016*, pages 694–711, 2016.