Minecraft-ify: Minecraft Style Image Generation with Text-guided Image Editing for In-Game Application

Bumsoo Kim[1], **Sanghyun Byun**[2], **Yonghoon Jung**[2],
Wonseop Shin[2], **Sareer UI Amin**[3], **Sanghyun Seo**[*]
[1,*]School of Art and Technology, Chung-Ang University
[2]GSAIM, Chung-Ang Univeristy
[3]Department of Computer Science and Engineering, Chung-Ang University
[1]bumsookim00@gmail.com, {[2]egoist12276, [2]dydgns2017,
[2]wonseop218, [3]sarrer2021, [*]sanghyun}@cau.ac.kr

Abstract

In this paper, we first present the character texture generation system *Minecraft-ify*, specified to Minecraft video game toward in-game application. Ours can generate face-focused image for texture mapping tailored to 3D virtual character having cube manifold. While existing projects or works only generate texture, proposed system can inverse the user-provided real image, or generate average/random appearance from learned distribution. Moreover, it can be manipulated with text-guidance using StyleGAN and StyleCLIP. These features provide a more extended user experience with enlarged freedom as a user-friendly AI-tool. Project page can be found at https://gh-bumsookim.github.io/Minecraft-ify/

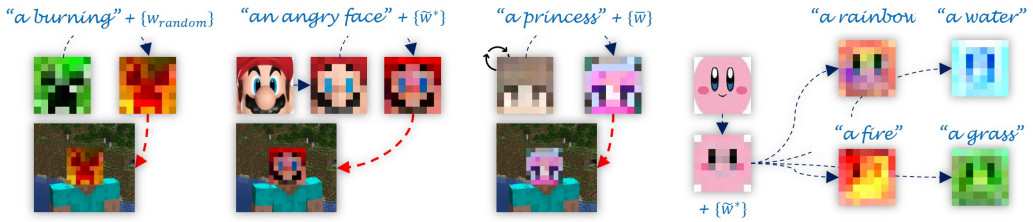

Figure 1: Rendered 3D character in Minecraft-World using our generated frontal character texture.

1 Introduction

Non-photorealistic image generative models with editability leads AI-based creation tools for comic book, animation and video game. Recently, for specific domain with applicability, generative model have been tailored with data-centric approach [4, 8, 14, 13, 15, 11, 3, 7, 10]. In this paper, we present creation tool, specified the 3D character texture of Mincraft-World[2] based on StyleGAN [5, 6] and StyleCLIP [9] including text-guided manipulation. With elaborately refined large Minecraft-World character texture dataset, game player can generate the frontal face texture of 3D character and manipulate it via text description with extended user experience and freedom.

[*]Corresponding author
[2]https://www.minecraft.net/

Preprint. Under review.

2 Method

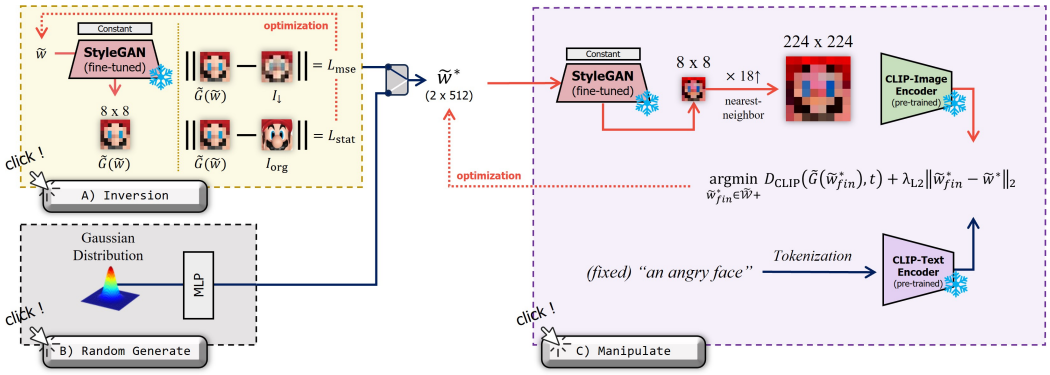

Figure 2: Overview of our *Minecraft-ify* system.

Our proposed system aims to generate and manipulate Minecraft-World character image having texture format. From that, we can provide wider user freedom for character creation with two paths: A) generate the frontal texture from learned distribution, or B) inverse the user-provided real image. Finally, they can manipulate generated image with text description. For inversion, our inversion objective, was originally proposed from Image2StyleGAN [1], designed with simple modification:

$$\underset{\tilde{w}}{\operatorname{argmin}} \frac{\lambda_{\text{mse}}}{N} \left\| \tilde{G}(\tilde{w}) - I_{\downarrow} \right\|_2^2 + \lambda_{\text{stat}} L_{\text{stat}}(\tilde{G}(\tilde{w}), I_{\text{org}}), \tag{1}$$

where $\tilde{G}(\cdot)$ is fined-tuned generator trained in preprocess with our large dataset which output the 8 by 8 image, $\tilde{w} \in \mathbb{R}^{2\times 512}$ is limited latent vector in limited space $\tilde{\mathcal{W}}+$ for specified Minecraft-World texture, $I_{\downarrow}$ is downsampled real image with same size as $\tilde{G}(\tilde{w})$ and L_{stat} is statistics loss obtained by:

$$L_{\text{stat}}(\tilde{G}(\tilde{w}), I_{\text{org}}) = \frac{1}{3} \sum_{c\in\{\mathbf{R},\mathbf{G},\mathbf{B}\}} \left(|\mu_c(\tilde{G}(\tilde{w}) - \mu_c(I_{\text{org}})| + |\sigma_c(\tilde{G}(\tilde{w}) - \sigma_c(I_{\text{org}})| \right), \tag{2}$$

where μ_c and σ_c are mean and standard deviation of c channel, respectively. With L_{stat}, we explicitly force the generated texture to have similar image statistics with real image I_{org} inspired by [2]. After inversion, we apply the StyleCLIP [9] via text using latent optimization method without identity loss:

$$\underset{\tilde{w}^*_{fin}\in\tilde{\mathcal{W}}+}{\operatorname{argmin}} \; D_{\text{CLIP}}(\tilde{G}(\tilde{w}^*_{fin}), t) + \lambda_{\text{L2}} \left\| \tilde{w}^*_{fin} - \tilde{w}^* \right\|_2, \tag{3}$$

where $\tilde{w}^*$ is fixed vector obtained by inversion process, D_{CLIP} output the similarity between image and text using CLIP [12] image-, text-encoder and t is tokenized vector from text description. From Eq. 3, we can finalize the manipulation process for in-game texture generation and editing through StyleCLIP-based [9] optimized vector $\tilde{w}^*_{fin}$ as $\tilde{G}(\tilde{w}^*_{fin})$. Player also can utilize average vector $\bar{w}$ or random vector w_{random} instead of inversed vector $\tilde{w}^*$ in Eq. 3 without considering real image input.

3 Conclusion

To generate and manipulate the Minecraft-World texture toward in-game application, we proposed *Minecraft-ify* that can fully support the functions for enhanced user-freedom as user-friendly AI-tool using StyleGAN [5, 6] and StyleCLIP [9]. From experimental results, we demonstrated that the text-guided manipulation can enough provide semantically plausible appearance although it was derived from user-wanted real sample by inversion. Additionally, we also showed that user can generate seamless random or average appearance texture from the learned distribution without considering the input images.

4 Ethical Implications

Our large dataset originally obtained from `here` using Public Domain license. Our system generate the image via text with CLIP [9]. CLIP is known to have unwanted data-bias issues by training dataset. Thus, it is important that the user do not use this work for generating harmful or unpleasant things. Note that this work is proposed for entertainment purposes only to easily create diverse character texture to enrich the in-game play experience.

References

[1] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4431–4440, 2019.

[2] M. Afifi, M. A. Brubaker, and M. S. Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7941–7950, 2021.

[3] J. Back, S. Kim, and N. Ahn. Webtoonme: A data-centric approach for full-body portrait stylization. *SIGGRAPH Asia 2022 Technical Communications*, 2022.

[4] Z. Hao, A. Mallya, S. J. Belongie, and M.-Y. Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14052–14062, 2021.

[5] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2019.

[7] Z. Li, Y. Xu, N. Zhao, Y. Zhou, Y. Liu, D. Lin, and S. He. Parsing-conditioned anime translation: A new dataset and method. *ACM Transactions on Graphics*, 42:1 – 14, 2023.

[8] J. Lin, Y. Yuan, and Z. Zou. Meingame: Create a game character face from a single portrait. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[9] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[10] M. Pesko, A. Svystun, P. Andruszkiewicz, P. Rokita, and T. Trzciński. Comixify: Transform video into a comics. *ArXiv*, abs/1812.03473, 2018.

[11] J. N. M. Pinkney and D. Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *ArXiv*, abs/2010.05334, 2020.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[13] T. Shi, Y. Yuan, C. Fan, Z. Zou, Z. Shi, and Y. Liu. Face-to-parameter translation for game character auto-creation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 161–170, Los Alamitos, CA, USA, 2019. IEEE Computer Society.

[14] Z. Wu, L. Chai, N. Zhao, B. Deng, Y. Liu, Q. Wen, J. Wang, and S. He. Make your own sprites. *ACM Transactions on Graphics (TOG)*, 41:1 – 16, 2022.

[15] R. Zhao, W. Li, Z. Hu, L. Li, Z. Zou, Z. X. Shi, and C. Fan. Zero-shot text-to-parameter translation for game character auto-creation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21023, 2023.