TexSR: Image Super-Resolution for High-Quality Texture Mapping

Jae-Ho Nah
Sangmyung University
Seoul, South Korea
jaeho.nah@smu.ac.kr

Hyeju Kim
Sangmyung University
Seoul, South Korea
hyeju724@naver.com

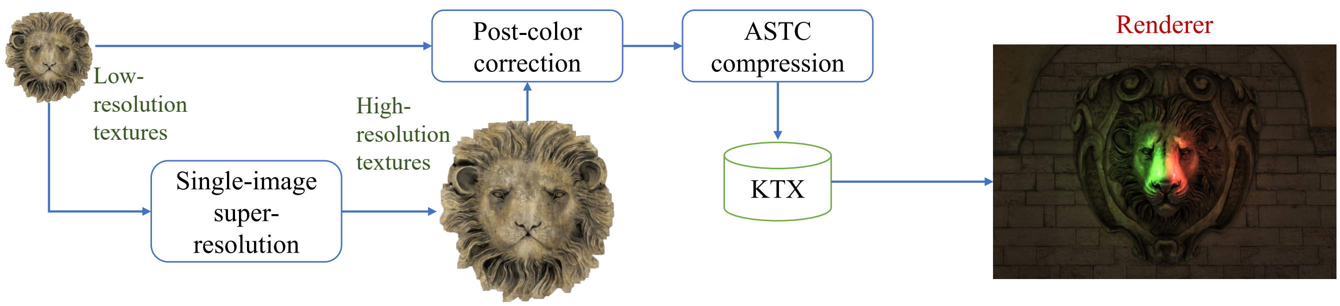

Figure 1: The process of our texture super-resolution. ©Crytek.

ABSTRACT

We introduce an image super-resolution technique for high-quality texture mapping in this poster. We first get upscaled textures from an existing image super-resolution (SR) method. We then perform a post-color correction algorithm to restore color tones and details lost in the SR algorithm. Finally, we compress the textures with variable compression ratios to reduce storage and memory overheads caused by the increased resolution. As a result, TexSR can improve the image quality of a state of the art, Real-ESRGAN.

CCS CONCEPTS

• **Computing methodologies → Image processing**.

KEYWORDS

Image super-resolution, texture mapping, ASTC

ACM Reference Format:
Jae-Ho Nah and Hyeju Kim. 2022. TexSR: Image Super-Resolution for High-Quality Texture Mapping . In *SIGGRAPH Asia 2022 Posters (SA '22 Posters), December 06-09, 2022.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3550082.3564204

1 INTRODUCTION

With the increasing number of high-resolution (up to 8K) displays, the rendering resolution of games and other graphical apps has been proportionally increased. Some game vendors have distributed HD texture packs to upgrade the texture quality of their game apps. Recreating high-resolution textures requires designers' efforts. As an alternative, a deep-learning-based image super-resolution technique can automate this task [Takemura 2018].

Although recent advances in single-image super-resolution (SISR) have made impressive progress with deep neural networks [Liang et al. 2021; Wang et al. 2021; Zhang et al. 2021], direct use of these techniques may not be suitable for increasing the resolution of textures. Because the SISR techniques train networks for restoring low-quality noisy images, upscaled results from noise-free input images like textures may be too clean. In other words, the filtered output may either lose some details or look unrealistic.

High-resolution textures also bring out storage and memory overheads. If we upscale a texture by a factor of two (e.g., from 1Kx1K to 2Kx2K), the size of the texture is quadrupled. As a result, the size of an HD texture pack ranges from tens to hundreds of gigabytes. Thus, it is necessary to reduce the size of upscaled textures if possible.

2 PROPOSED METHOD

To tackle the problems about the image quality and size, we present a texture super-resolution (TexSR) technique (Figure 1). Note that we aim at 2x texture upscaling for practical use.

The first stage of our technique is to perform an upscaling using a SISR method. We investigated three state-of-the-art SISR methods: Real-ESRGAN [Wang et al. 2021], SwinIR [Liang et al. 2021] and BSRGAN [Zhang et al. 2021], and we chose Real-ESRGAN as the base algorithm among them because it showed better SR results in our texture set than the others. We then retrained Real-ESRGAN_x2plus with the modified parameters. First, we have removed the additive Gaussian and Poisson noise because textures usually do not include noise. Second, we have not considered the perceptual loss for training the network. We found that this loss was not highly effective for reconstructing details if the scaling factor was two, and it also often generated color distortion.

SA '22 Posters, December 06-09, 2022, Daegu, Republic of Korea

ACM ISBN 978-1-4503-9462-8/22/12.
https://doi.org/10.1145/3550082.3564204

The second stage is a post-color correction to improve image quality further. The output images from the first stage may tend to either be over-smooth or show different color tones from the original images due to inaccurate estimations of the neural network. To recover the details and color tones, we compare four pixels in the upscaled SR image and a pixel in its original image on the HSV space. If the highest difference among hue, saturation, and values between them is higher than 22%, we maintain the reconstructed pixels to represent clean edges. If not, we blend the two images by adding the difference to the pixels in the SR image and clipping the values to the HSV range. Note that we set the threshold value according to our internal experiments.

The last stage is texture compression. Among the standard texture compression formats, ASTC [Nystad et al. 2012] is the only codec providing a fine-grained trade-off between quality and size. Thus, ASTC is suitable for TexSR if the target platform supports ASTC. Instead of applying the same block size to all textures, we adopt a PSNR-based approach suggested by Nah [2022] to apply the appropriate size to each texture with different characteristics. This approach can minimize quality scarification caused by increased block sizes because compressed textures result in similar or higher PSNR values than the predefined target PSNR value.

3 EXPERIMENTAL RESULTS

To measure the image quality of our approach, we used the texture set used in Nah [2020] (excluding seven 8K textures) in our experiments. We scaled the textures in the set by half, upscaled the textures with super-resolution, and calculated the PSNR and ꟻLIP [Andersson et al. 2020] values of the results by comparing with the original textures. As described in Table 1, our training parameter adjustments showed 0.14 dB higher PSNR and 14% less ꟻLIP mean values on average, compared to the unmodified Real-ESRGAN. After we applied the post-color correction, the PSNR value increased by 1.24 dB on average, and the ꟻLIP mean values decreased by 28% on average. The zoomed-in images in Figure 2 support the results in Table 1; TexSR successfully enhanced texture details, such as holes, feathers, cloth, and tiles in the first to fourth textures.

To measure the space efficiency of our approach, we compare the total texture sizes of the Crytek Sponza textures in Vulkan Sponza [Willems 2018]. The size of all the DXT3 textures included in Vulkan Sponza is 82 MB, and that of our 2x-upscaled, ASTC-compressed textures is 164 MB (at 37.3 dB of the target PSNR [Nah 2022]). Our approach reduced the storage and memory requirements of 2x upscaling by half.

Table 1: Quantitative image quality comparison with the entire texture set. Higher PSNR and lower ꟻLIP values are better, respectively.

	Avg. PSNR (dB)	Avg. ꟻLIP (mean)
Real-ESRGAN (base)	26.79	0.125
+ modified parameters	26.93	0.108
+ post-color correction	28.17	0.078

4 FUTURE WORK

It is possible to find a more optimal threshold in the post-color correction algorithm with further statistical analysis. We are also interested in real-time texture upscaling at the driver level because this implementation can affect all graphical apps as with real-time texture resizing [Nah et al. 2018].

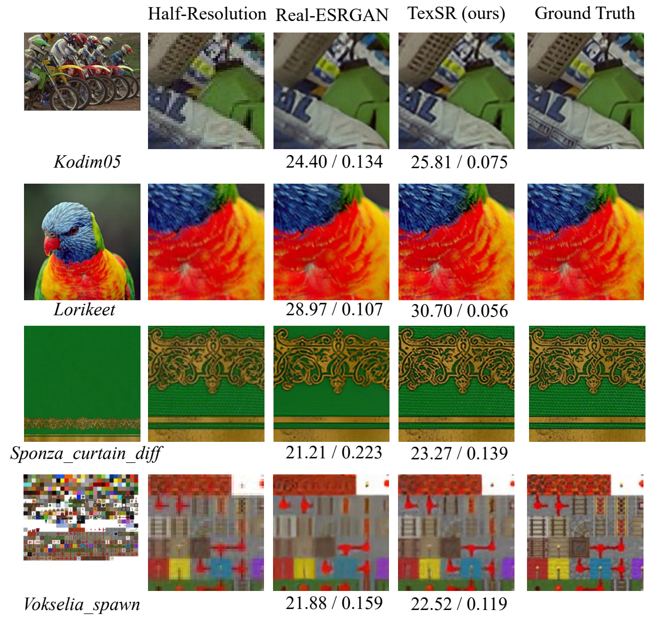

Figure 2: Quality comparison of the four textures. The two numbers below each magnified image represent PSNR and ꟻLIP values of the upscaled texture, respectively. ©Kodak, Simon Fenney, Crytek and Vokselia

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1093404).

REFERENCES

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, and Kalle Åströmand Mark D. Fairchild. 2020. ꟻLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (HPG 2020)* 3, 2, Article 15 (2020), 23 pages.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. SwinIR: Image Restoration Using Swin Transformer. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*. 1833–1844.

Jae-Ho Nah. 2020. QuickETC2: Fast ETC2 Texture Compression using Luma Differences. *ACM Transactions on Graphics (SIGGRAPH Asia 2020)* 39, 6, Article 270 (2020).

Jae-Ho Nah. 2022. ASTC Block-Size Determination Method based on PSNR Values. *Journal of the Korea Computer Graphics Society* 28, 2 (2022), 21–28.

Jae-Ho Nah, Byeongjun Choi, and Yeongkyu Lim. 2018. Classified texture resizing for mobile devices. In *ACM SIGGRAPH 2018 Talks*. Article 73.

Jorn Nystad, Anders Lassen, Andy Pomianowski, Sean Ellis, and Tom Olson. 2012. Adaptive scalable texture compression. In *Proceedings of the conference on High-Performance Graphics (HPG)*. 105–114.

Shintaro Takemura. 2018. Optimize Deep Super-Resolution and Denoising for Compressed Textures. In *ACM SIGGRAPH Asia 2018 Posters*. Article 58, 2 pages.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-ESRGAN: Training Real-World Blind Super-Resolution With Pure Synthetic Data. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*. 1905–1914.

Sascha Willems. 2018. *Vulkan Sponza*. https://github.com/SaschaWillems/VulkanSponza

Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. 2021. Designing a Practical Degradation Model for Deep Blind Image Super-Resolution. In *Proceedings of the International Conference on Computer Vision (ICCV)*. 4791–4800.