Supplemental materials: Neural Super-Resolution for Real-time Rendering with Radiance Demodulation

1. Details of Radiance Demodulation

In this section, we first introduce the specific bidirectional reflectance distribution function (BRDF) used in the implementation and then go into detail about the pre-computation of the material component F_β. One example of radiance demodulation is shown in Figure 1.

Bidirectional Reflectance Distribution Function. We use Disney physically-based material model [7] as our bidirectional reflectance distribution function (BRDF), which is widely used nowadays. And we can split the BRDF into diffuse and specular terms in real-time rendering. The formula is as follows:

$$\rho(\omega_i, \omega_o) = \rho_{\text{diff}}(\omega_i, \omega_o) + \rho_{\text{spec}}(\omega_i, \omega_o), \tag{1}$$

where ω_i and ω_o represent the incoming direction and outcoming direction, respectively. ρ_{diff} and ρ_{spec} represent the diffuse term and specular term of the BRDF.

For the diffuse term, we directly use the Lambertian model, the formula is as follows:

$$\rho_{\text{diff}}(\omega_i, \omega_o) = k_{\text{d}} \frac{c}{\pi}, \tag{2}$$

$$= (1 - m)\frac{c}{\pi}, \tag{3}$$

where k_{d} represents the diffuse ratio which can be calculated by metallic value m, and c is the albedo of the object.

For the specular term, we use the Cook-Torrance [4] microfacet specular shading model. The general formula is as follows:

$$\rho_{\text{spec}}(\omega_i, \omega_o) = \frac{D(\omega_h)F(\omega_o, \omega_h)G(\omega_i, \omega_o)}{4(\omega_o \cdot \omega_h)(\omega_i \cdot \omega_h)}, \tag{4}$$

where ω_h represents the half vector between ω_i and ω_o. $D(\omega_h)$ is the normal distribution function (NDF), $F(\omega_o, \omega_h)$ is the Fresnel term and $G(\omega_i, \omega_o)$ is the shadowing-masking function.

We use the GGX/Trowbridge-Reitz model [17] as our normal distribution function:

$$D(\omega_h) = \frac{\alpha^2}{\pi((\omega_n \cdot \omega_h)^2(\alpha^2 - 1) + 1)^2}, \tag{5}$$

where α and ω_n represent the roughness and normal of the object surface, respectively.

For the Fresnel term, we use Schlick's approximation [15]:

$$F(\omega_o, \omega_h) = F_0 + (1 - F_0)(1 - (\omega_o \cdot \omega_h)) \tag{6}$$

and

$$F_0 = \text{lerp}(0.04, c, m), \tag{7}$$

where F_0 is the specular reflectance at normal incidence, which can be obtained by linear interpolation using the metallic value m from plastic Fresnel coefficient (0.04) to albedo c.

Finally, we use Smith method [19] and Schlick model [15] to formulate our shadowing-masking function:

$$G(\omega_i, \omega_o) = G_1(\omega_i)G_1(\omega_o), \tag{8}$$

where

$$G_1(\omega_o) = \frac{\omega_n \cdot \omega_o}{(\omega_n \cdot \omega_o)(1 - k) + k}, \tag{9}$$

$$k = \frac{(\alpha + 1)^2}{8}. \tag{10}$$

Pre-computation. Following Zhuang et al. [24] and the equation 1, the material components F_β in the radiance demodulation module can be rewritten as

$$F_\beta(\omega_o) = \int \rho(\omega_i, \omega_o) \cos\theta_i \text{d}\omega_i, \tag{11}$$

$$= \int \rho_{\text{diff}}(\omega_i, \omega_o) \cos\theta_i \text{d}\omega_i + \int \rho_{\text{spec}}(\omega_i, \omega_o) \cos\theta_i \text{d}\omega_i, \tag{12}$$

where θ_i is the angle between the incoming direction and the shading normal. We split the integral into two terms, the diffuse term and the specular term.

The integral of the diffuse term can be calculated directly:

$$\int \rho_{\text{diff}}(\omega_i, \omega_o) \cos\theta_i \text{d}\omega_i = k_{\text{d}} c, \tag{13}$$

$$= (1 - m)c. \tag{14}$$

Figure 1. Visualization of the material component, lighting component and the radiance image. The texture details are shown on the material component, and the lighting component is much smoother than the radiance.

Because the integral of the specular term is angular-dependent, we need to pre-compute this part to convert it into a simple function. However, if we directly pre-compute the entire integral, we need to store many parameters. Inspired by Karis et al. [7], we extract F_0 out of the Fresnel term and convert the integral into a simple linear function. After a series of derivations, the integral of the specular term can be converted to the following form:

$$\int \rho_{\text{spec}}(\omega_i, \omega_o) \cos\theta_i \mathrm{d}\omega_i = F_0 A + B, \tag{15}$$

where

$$A = \int \frac{\rho_{\text{spec}}(\omega_i, \omega_o)}{F(\omega_o, \omega_h)} (1 - F_{\text{c}}) \cos\theta_i \mathrm{d}\omega_i, \tag{16}$$

$$B = \int \frac{\rho_{\text{spec}}(\omega_i, \omega_o)}{F(\omega_o, \omega_h)} F_{\text{c}} \cos\theta_i \mathrm{d}\omega_i, \tag{17}$$

$$F_{\text{c}} = (1 - (\omega_o \cdot \omega_h))^5. \tag{18}$$

The two resulting integrals represent a scale (denoted by A) and a bias (denoted by B) to F_0, respectively. We perform importance sampling on the incident direction vector ω_i, resulting in a 2D lookup table with respect to $\cos\theta_o$ and roughness α. In our implementation, the resolution of the pre-computation lookup table is 512×512 with 1024 samples per pixel, as shown in Figure 2.

After performing the pre-computation above, we can avoid complex integral of material component F_β in real-time inference. We can directly use the albedo map (c) and metallic map (m) to obtain the integral value of the diffuse term. And use the specular map (F_0), NoV map (the dot product of normal and view direction, i.e., $\cos\theta_o$) and the roughness map (α) to query the pre-computation lookup table to get the integral value of the specular term. The result of adding the two terms is the final material component F_β.

2. Details of Dual Motion Vector

In order to alleviate the ghosting problem caused by warping with the traditional motion vector (TMV) due to object occlusion in dynamic scenes, Zeng et al. [22] proposed

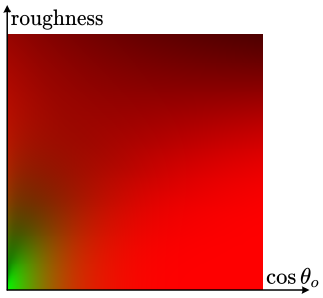

Figure 2. Pre-computation lookup table. The horizontal axis is $\cos\theta_o$, and the vertical axis is roughness α. The first and second channels are the scale (denoted by A) and the bias (denoted by B) to F_0, respectively.

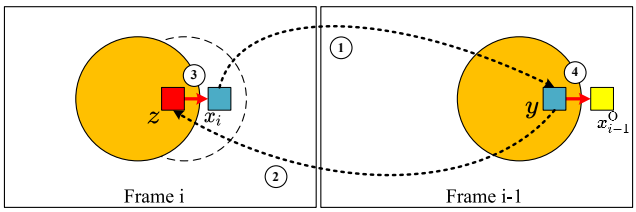

Figure 3. Illustration of the implementation of dual motion vectors for occlusions. For a pixel x_i that is visible now but was occluded in the previous frame at y, we find where the occluder y is in the current frame at z. Then we find x_i's correspondence x^{O}_{i-1} in the previous frame using the relative motion vector from z to x_i.

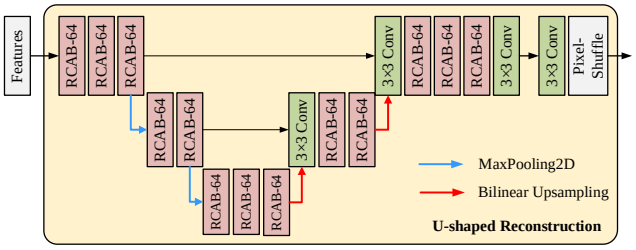

Figure 4. The structure of our U-shaped reconstruction module.

the dual motion vector (DMV), and its implementation process is shown in Figure 3. First, for the visible point x_i in the current frame, find the occluded point y in the previous frame through back projection, where $y - x_i$ is considered as TMV. In the second step, find the point z in the current frame through forward projection from point y in the previous frame. The third step is to get the relative motion vector from point z to point x_i. Finally, use this relative motion vector to find point x^{O}_{i-1} in the previous frame, and set $x^{\text{O}}_{i-1} - x_i$ as DMV. In our method, we can accurately obtain the occluded area by subtracting DMV with TMV.

3. Details of Reconstruction Module

In the frame-recurrent super-resolution module, we use the U-shaped reconstruction module to reconstruct and upsample the intermediate features to obtain the high-resolution lighting components. The structure of the U-shaped reconstruction module is shown in Figure 4. The residual channel attention blocks (RCAB) [23] in the module have been widely used in super-resolution due to the ability to improve reconstruction quality, and we choose the U-shaped structure to connect them in order to reduce the network computation as much as possile. We use max-pooling layer for downsampling, bilinear interpolation for upsampling and channel-wise connections for preserving the shallow features. Finally, the pixel-shuffle operation [16] is used for upsampling reconstruction.

4. Experiments

4.1. Datasets

We use the Unity [18] rendering engine to generate our dataset. We select seven representative scenes, namely Bistro [11], Square [13], San Miguel (San_M) [12], Bar [11], ZeroDay [20], Airplane and Pica. The example images are shown in Figure 13. Similar to previous work [21], we uniformly distribute different fast-moving cameras in each scene to generate multiple sequences of 100 frames each, containing as different objects and materials as possible to enhance diversity. We randomly divided the training, validation and testing datasets from these sequences, and the exact number of sequences is shown in Table 1. We also generate a set of rendering G-buffers as the additional inputs to the network. An example is shown in Figure 5.

4.2. Implementation Details

We demodulate the LR radiance into the lighting component with the LR and HR material G-buffer generated in the deferred rendering pipeline, as described in Section 1. Then, the light component, together with the LR depth, normal, motion vector, and dual motion vector into the network, is fed into the network.

In our super-resolution neural network, we set the convolutional output channel number as 32 in the radiance demodulation and reliable warping modules. The convolutional output channel number in the frame-recurrent reconstruction module is set as 64 (except for the last channel number is $3 \times s \times s$ for the pixel-shuffle operation, where s is the SR factor). The breakdown of our network is shown in Table 2.

The output of the network – the reconstructed HR light component, is modulated with the HR material component to form the current output frame and aids the reconstruction for the next frame.

Table 1. The sequence number of training, validation and testing dataset for each scene. Each sequence contains 100 frames.

Scene	Training Sequences	Validation Sequences	Testing Sequences
Bistro	54	6	12
San_M	54	6	6
Square	42	6	6
Bar	45	3	6
ZeroDay	24	3	3
Pica	30	3	3
Airplane	24	3	3

Table 2. The parameters and GFLOPs of each module.

		Params (K)	GFLOPs
Radiance Demodulation		2.05	0.26
Reliable Warping		9.34	1.20
Frame-Recurrent Reconstruction	First conv	13.86	1.79
	ConvLSTM	442.62	57.33
	Ushaped-Net	1142.92	84.78
Total		1610.79	145.36

Table 3. Comparison among our method, DLSS 2.0 and FSR 2.0 on the Bistro scene. The SR factor is set as 2×2.

	PSNR(dB)	SSIM	LPIPS ↓
DLSS 2.0	28.35	0.9104	0.136
FSR 2.0	28.90	0.9117	0.137
Ours	**30.11**	**0.9405**	**0.080**

Table 4. Comparison between our method and EDSR with four error measurements on the Bistro scene. The SR factor is set as 4×4.

	PSNR(dB)	SSIM	LPIPS ↓	VMAF
EDSR	24.08	0.7625	0.327	33.25
Ours	**26.43**	**0.8739**	**0.141**	**53.82**

Table 5. Reconstruction quality versus the SR factor on the Bistro scene with the target resolution set as 1920 × 1080. The metrics are averaged over all test data (1200 frames).

SR factor	2 × 2	4 × 4	6 × 6
PSNR	30.11	26.43	25.18
SSIM	0.9405	0.8739	0.8349

4.3. More Comparison Results

We provide additional qualitative comparisons of the results on seven scenes, as shown in Figure 10 and Figure 11. From the results, our method not only produces results with richer texture details (Bistro Scene) but also recovers the view-dependent highlights (ZeroDay scene) well. It can also be seen from the Pica scene that our method successfully elim-

inates the obvious ghosting problem in NSRR [21]. Furthermore, as can be seen from the video in the supplementary material, our method has better temporal stability compared to other methods.

We adopt the epipolar plane image (EPI) [1] to evaluate the temporal consistency visually by plotting the transition of the dotted red horizontal scanlines over time, as shown in Figure 12. By comparison, our results look sharper and are the closest to the ground truth, while the other methods exhibit blurred results and flickering artifacts, demonstrating that our method is more temporal consistent than other methods.

The quantitative comparison results among our method, DLSS 2.0 [5] and FSR 2.0 [6] are shown in Table 3. Our method outperforms the other two methods.

4.4. Comparison of Generalization Ability

We have compared the generalization results of FRVSR [14], TecoGAN [3] and NSRR [21] quantitatively in the main paper, and we also provide a comparison of qualitative results in Figure 9. It can be seen from the results that other methods cause excessive blurring of details. Although TecoGAN can get a slightly sharp result, it is still quite different from GT. However, our method can still preserve complex texture details, which proves that our method has generalization ability.

4.5. Varying SR Factors Results

Table 5 and Figure 8 show the quantitative and qualitative reconstruction results under different SR factors, respectively. We keep the target resolution (1920 $\times$ 1080) the same and modify the input image resolution according to the SR factors. As the SR factor increases, the error becomes larger, since the SR reconstruction becomes more difficult. Our method can still preserve rich texture details thanks to the radiance demodulation module.

4.6. Comparison with SISR Method

We compare our method with a SISR method, i.e., EDSR [9], in Table 4 and Figure 6. Our method shows higher quality than EDSR on all metrics, and the details are better preserved. Furthermore, EDSR leads to poor temporal stability, as demonstrated in the video, since they do not consider the consistency between consecutive frames.

4.7. Limitations

Although our method produces high-fidelity results in most scenarios, we have still identified some limitations, including complex indirect reflections and moving shadows, which are known to be challenging, as shown in Figure 7. These high-frequency effects are due to the lighting rather than the material component. Therefore, our method shows subtle benefits.

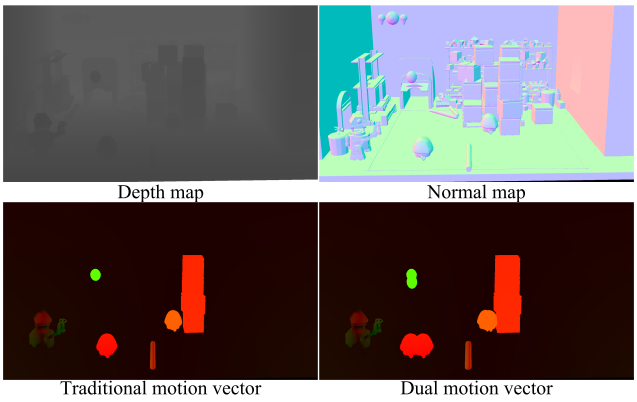

Figure 5. Additional network inputs of the Pica scene.

Figure 6. Comparison with EDSR on the Bistro scene. The target resolution is set as 1920 $\times$ 1080 and the SR factor is set as 4 $\times$ 4.

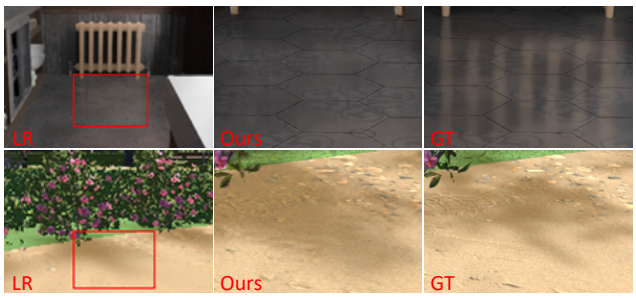

Figure 7. Failure cases on high-frequency indirect reflections and shadow boundaries.

References

[1] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining

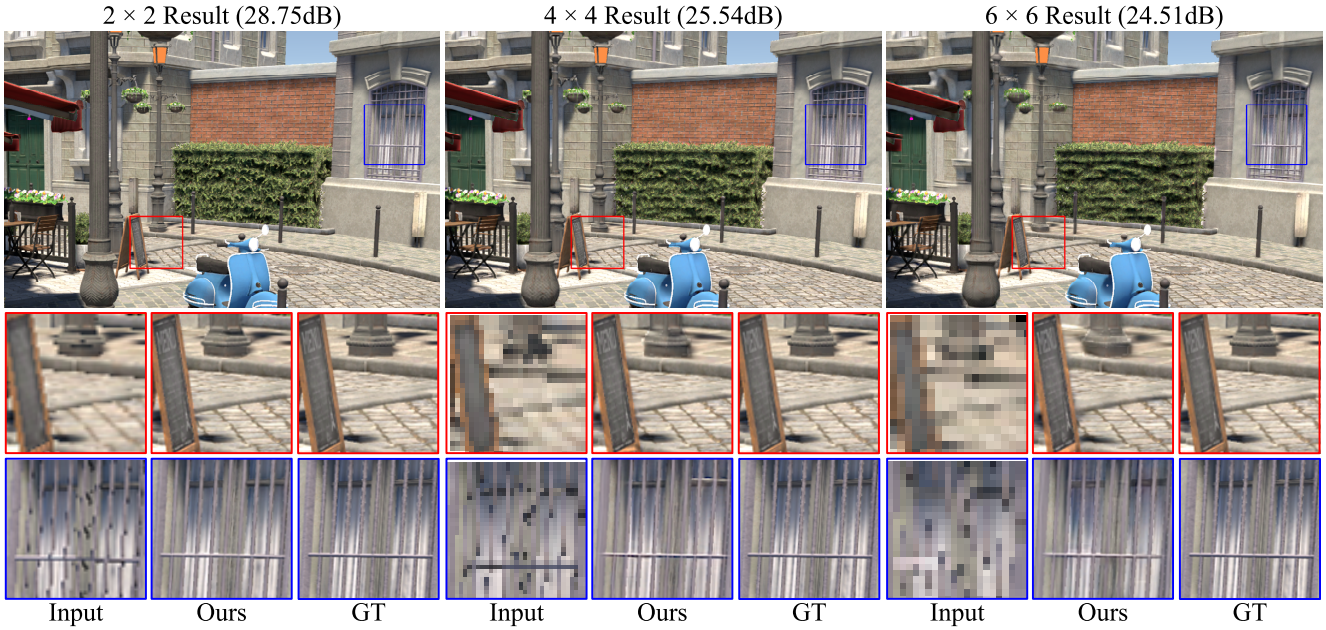

Figure 8. Reconstruction quality versus the SR factor on the Bistro scene with the target resolution set as 1920 × 1080. The results of PSNR are tested on the single full-resolution image.

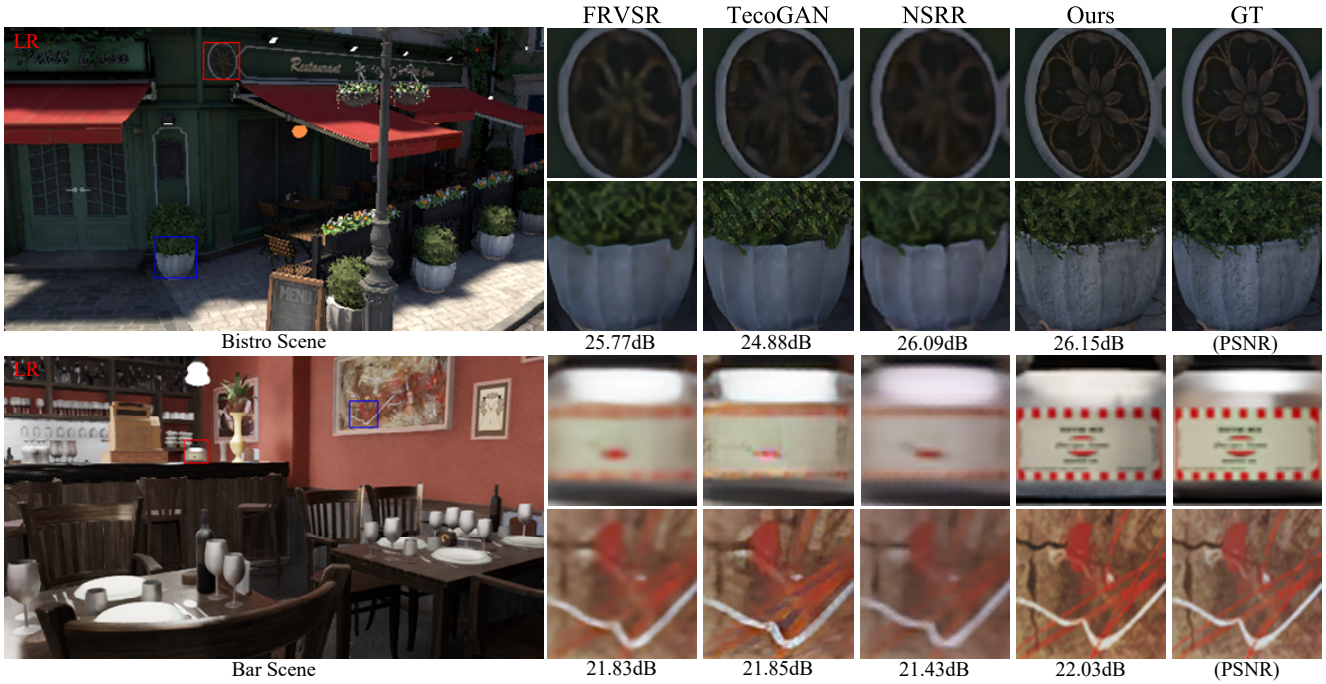

Figure 9. Comparison of generalization ability among our method, FRVSR [14], TecoGAN [3] and NSRR [21]. The target resolution is set as 1920 × 1080 and the upsampling ratio is set as 4 × 4.

structure from motion. *International journal of computer vision*, 1(1):7–55, 1987. 4, 7

[2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5972–5981, 2022. 6, 7

[3] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transac-*

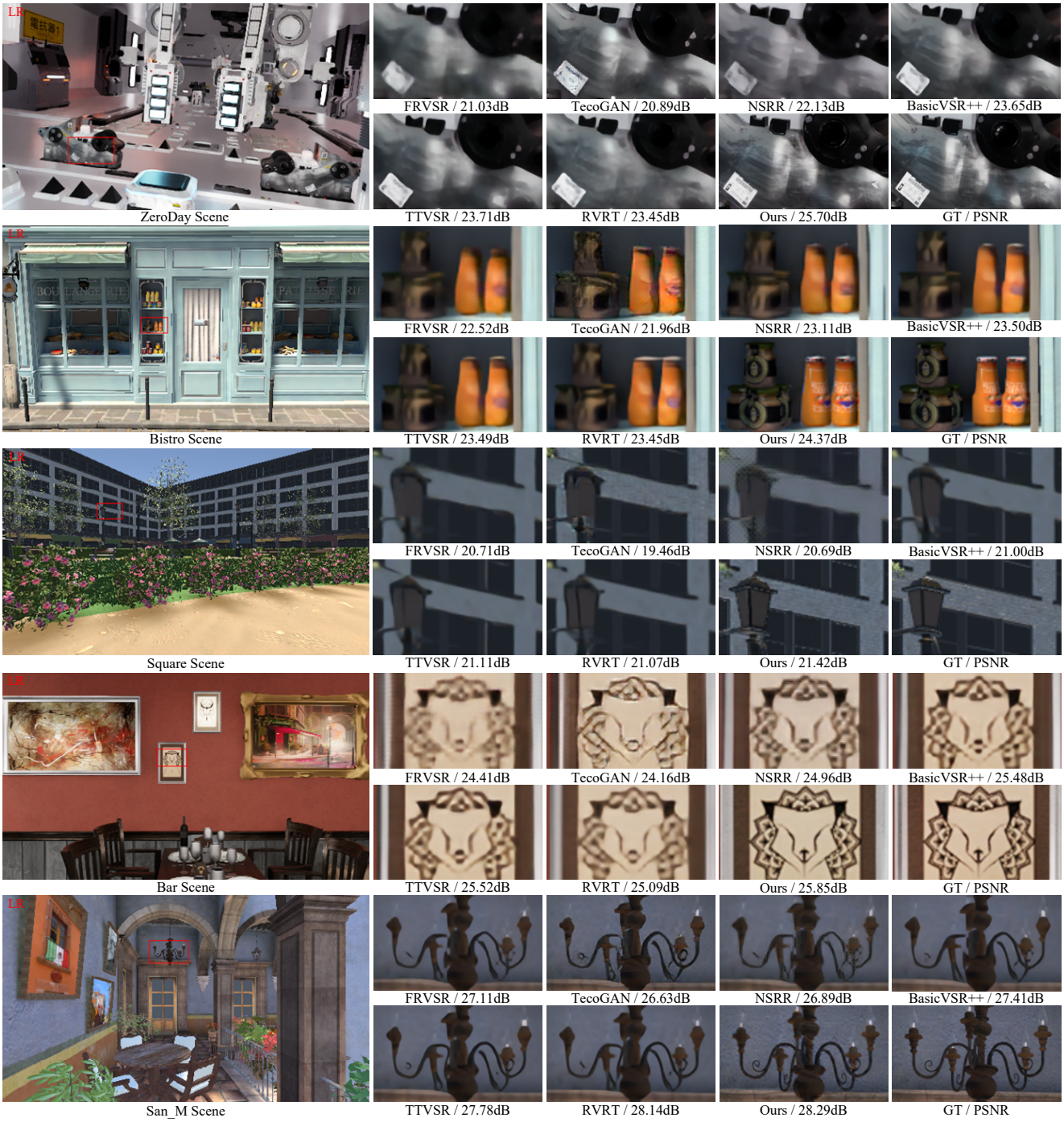

Figure 10. Comparison among our method, FRVSR [14], TecoGAN [3], NSRR [21], BasicVSR++ [2], TTVSR [10] and RVRT [8]. The target resolution is set as 1920×1080 and the upsampling ratio is set as 4×4.

tions on Graphics (TOG), 39(4):75–1, 2020. 4, 5, 6, 7

[4] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982. 1

[5] Andrew Edelsten, Paula Jukarainen, and Anjul Patney. Truly next-gen: Adding deep learning to games and graphics. In *In NVIDIA Sponsored Sessions (Game Developers Conference)*, 2019. 4

[6] Amd fidelityfx super resolution, 2022. 4

[7] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 1, 2

[8] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu

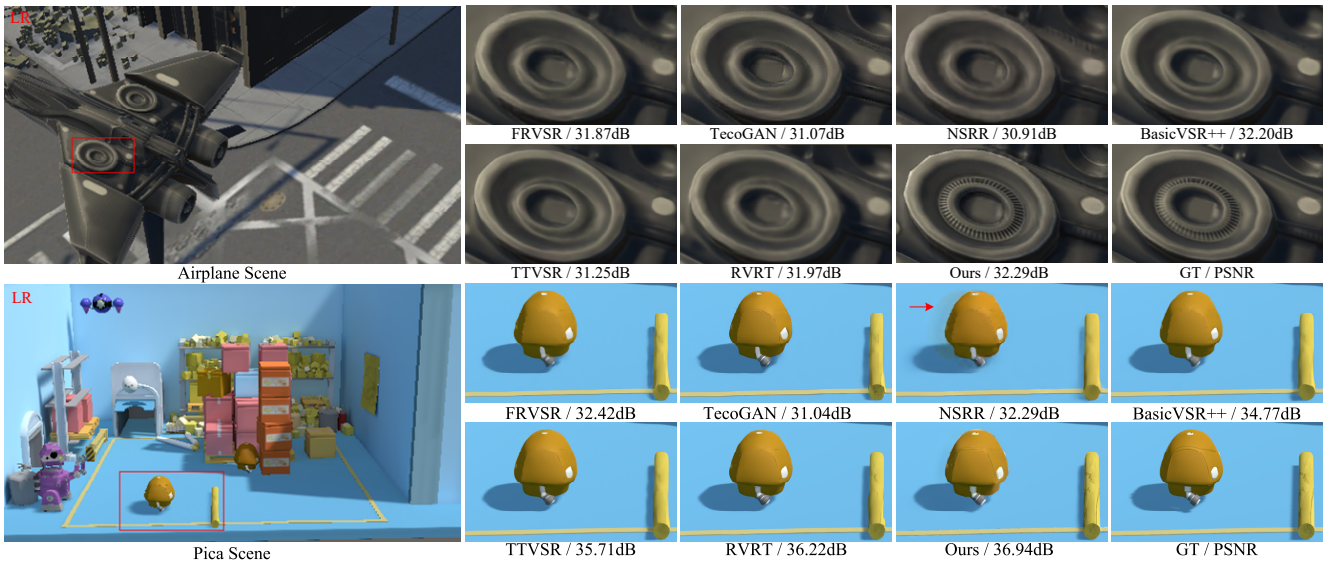

Figure 11. Comparison among our method, FRVSR [14], TecoGAN [3], NSRR [21], BasicVSR++ [2], TTVSR [10] and RVRT [8]. The target resolution is set as 1920×1080 and the upsampling ratio is set as 4×4.

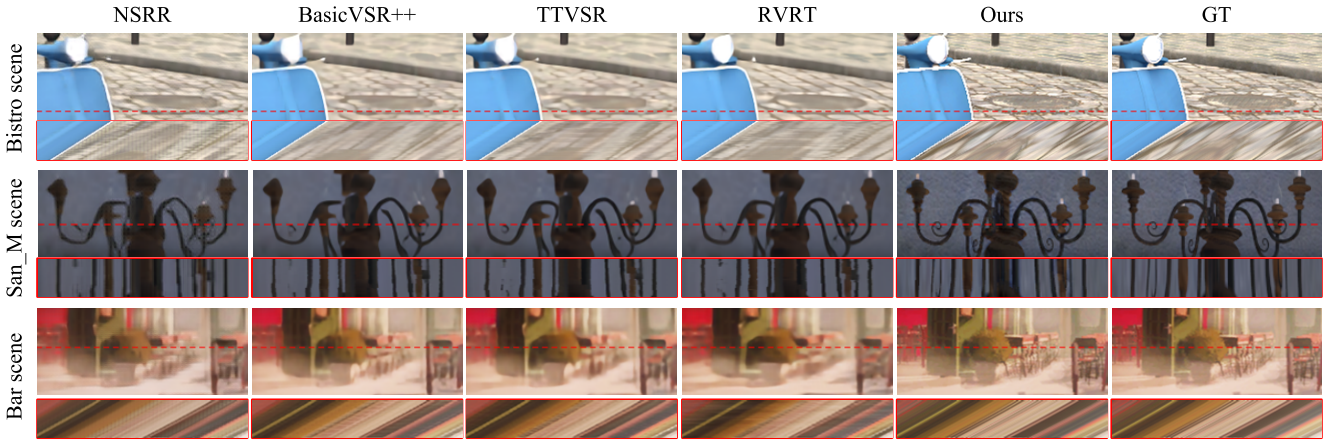

Figure 12. The EPI [1] comparison among NSRR, BasicVSR++, TTVSR, RVRT and our method on three scenes by plotting the transition of the dotted red horizontal scanline over time in the red box. The results which are sharper and closer to GT are better.

Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *Advances in Neu-*

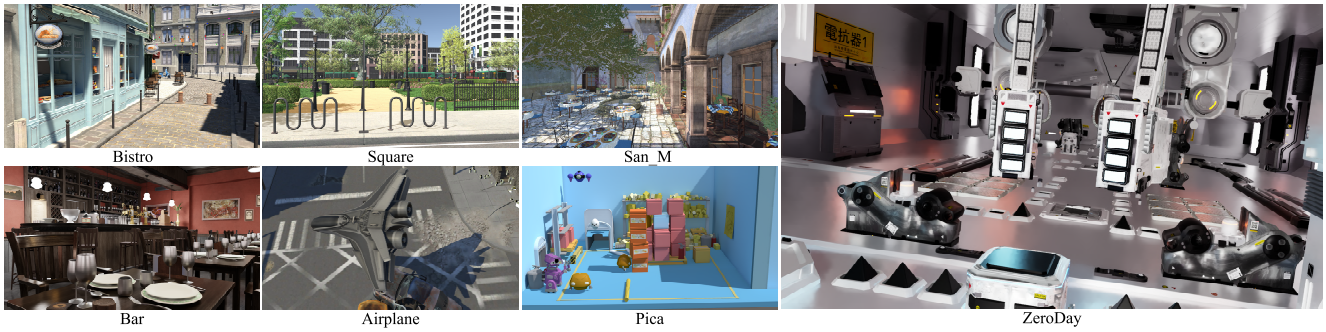

Figure 13. Example images of seven scenes.

ral Information Processing Systems, 35:378–393, 2022. 6, 7

[9] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1132–1140. IEEE Computer Society, 2017. 4

[10] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian. Learning trajectory-aware transformer for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5687–5696, 2022. 6, 7

[11] Amazon Lumberyard. Amazon lumberyard bistro, July 2017. 3

[12] Morgan McGuire. Computer graphics archive, July 2017. `https://casual-effects.com/data`. 3

[13] Kate Anderson Nicholas Hull and Nir Benty. Nvidia emerald square, July 2017. 3

[14] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6626–6634, 2018. 4, 5, 6, 7

[15] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994. 1

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1874–1883. IEEE Computer Society, 2016. 3

[17] TS Trowbridge and Karl P Reitz. Average irregularity representation of a rough surface for ray reflection. *JOSA*, 65(5):531–536, 1975. 1

[18] Unity, 2022. 3

[19] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 1

[20] Mike Winkelmann. Zero-day, open research content archive (orca), November 2019. 3

[21] Lei Xiao, Salah Nouri, Matt Chapman, Alexander Fix, Douglas Lanman, and Anton Kaplanyan. Neural supersampling for real-time rendering. *ACM Trans. Graph.*, 39(4), jul 2020. 3, 4, 5, 6, 7

[22] Zheng Zeng, Shiqiu Liu, Jinglei Yang, Lu Wang, and Ling-Qi Yan. Temporally reliable motion vectors for real-time ray tracing. *Computer Graphics Forum*, 40(2):79–90, 2021. 2

[23] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2018. 3

[24] Tao Zhuang, Pengfei Shen, Beibei Wang, and Ligang Liu. Real-time denoising using brdf pre-integration factorization. In *Computer Graphics Forum*, volume 40, pages 173–180. Wiley Online Library, 2021. 1