ExtraSS: A Framework for Joint Spatial Super Sampling and Frame Extrapolation – Supplementary Document

1 DATASET AND SCENE CONFIGURATION

Our dataset is generated from a modified Unreal Engine 4.25. We choose different types of scenes to test our system. *BUNKER* and *FOREST* are third-person games, which are mainly used for test the ability in extrapolating with correct shading and shadow movement as well as disoccluded area. Furthermore, *BUNKER* contains more metallic materials to demonstrate glossy reflection effects, while *FOREST* has complex geometries of leaves and trees. *SEQUENCER* and *INFILTRATOR* are first-person games and we use them to demonstrate the full ability of our ExtraSS framework. The resolutions of different scenes are varing in order to show the quality and performance under different settings. Table 1 shows the number of frames in each scene for training and the settings of input and target resolutions. We select two 120-frames clips from each scene for testing. The test sequences are not chosen from training sequences and don't overlap with training sequences.

Our dataset contains high resolution alias-free rendered images using SSAA as the ground truth and TAA as additional reference, and low resolution aliased rendered images with corresponding G-buffers as the input. Specifically, we render our data several times with the same sequencer in Unreal Engine 4.25 through following steps:

- The cinematic sequence in Unreal Engine can generate the exact same images except for the jittering offset. Thus, we use 64 Halton jitter sequence [Berblinger and Schlier 1991] to render the images in high resolution 64 times and average them in order to generate aliasing-free images.
- We use multiple rendering passes to generate the low resolution rendered images along with G-Buffers together. The low resolution images are rendered with 128 Halton jitter patterns.

The captured rendered images are not tone-mapped so we use μ-Law [Kalantari et al. 2017] to transfer linear HDR frames and inverse tone-map the output predictions back to the linear HDR space with μ equal to 8. All losses are calculated before the inverse tone-mapping. For G-buffer guided warping and FRNet, all rendered images are first demodulated and then modulated back before sending to the ExtraSS network. We follow the same demodulation method as ExtraNet [Guo et al. 2021]:

$$d = i/(\alpha + s \times 0.08 \times (1 - m))$$

and the modulation is

$$i = d * (\alpha + s \times 0.08 \times (1 - m))$$

where i is the rendered image, α is the *basecolor*, m is the *metallic*, s is the *specular* and d is the demodulated image. The tonemapping should be applied after the demodulation and inverse tonemapping should be applied before the modulation.

Table 1: Data configuration of each scene in our dataset. BK refers to *BUNKER*, FR refers to *FOREST*, IF refers to *INFILTRATOR*, and SQ refers to *SEQUENCER*.

	BK	FR	IF	SQ
Training Frames	4000	6000	6000	3500
Input	540p	540p	720p	1080p
Output	1080p	1080p	1440p	2160p

Table 2: We compare with different spatial super sampling quality in SSIM of TAAU, NSR, TAA with our methods. Ours-SS refers to spatial super sampling only results. Ours-ESS refers to extrapolated and super sampled high resolution results. Note that TAA is an additional reference instead of a spatial SS method, and our pipeline only takes half input frames comparing to baselines.

Scenes	TAAU	NSR	Ours-SS	Ours-ESS	TAA
BUNKER	0.872	0.912	0.895	0.888	0.923
FOREST	0.637	0.652	0.642	0.640	0.731
SEQUENCER	0.988	0.988	0.986	0.986	0.994
INFILTRATOR	0.963	0.969	0.967	0.966	0.976

Table 3: SSIM values on different test scenes. Ours-W refers the results with only G-buffer guided warping. Ours-E refers to Ours-W + shading refinement module. All method runs in the same input and output resolution without anti-aliasing.

	BK	FR	SQ	IF	Mean
ExtraNet	0.956	0.777	0.979	0.968	0.920
IFRNet	0.843	0.555	0.977	0.943	0.830
Ours-W	0.963	0.803	0.989	0.982	0.934
Ours-E	0.968	0.817	0.990	0.982	0.939

Table 4: SSIM values on different test scenes. We compare with ExtraNet and IFRNet in the same output resolution. We apply temporal anti-aliasing to ExtraNet and IFRNet so all outputs are anti-aliased. Note that our input resolution is 2X smaller than other baselines.

	BK	FR	SQ	IF	Mean
ExtraNet	0.910	0.778	0.992	0.979	0.915
IFRNet	0.802	0.703	0.969	0.990	0.866
Ours-ESS	0.888	0.641	0.986	0.966	0.870

Table 5: Comparison with NSR + ExtraNet. Note that since we are testing on consecutive sequences, it also has spatial super sampling only frames and joint spatial super sampling with extrapolation for NSR + ExtraNet baseline.

		NSR + ExtraNet		Ours	
	Scenes	SS	ESS	SS	ESS
PSNR	BUNKER	28.08	27.27	28.25	27.81
	FOREST	18.77	18.68	19.89	19.86
	SEQUENCER	37.24	36.59	37.75	37.57
	INFILTRATOR	29.97	29.55	30.04	29.91
SSIM	BUNKER	0.905	0.893	0.895	0.888
	FOREST	0.644	0.632	0.642	0.640
	SEQUENCER	0.985	0.983	0.986	0.986
	INFILTRATOR	0.964	0.961	0.967	0.966

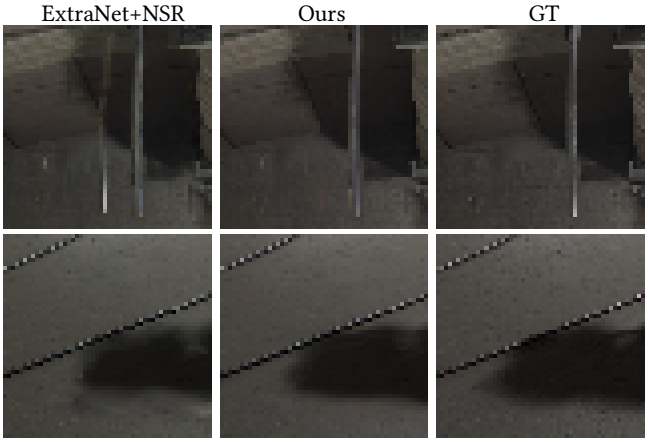

Figure 1: Comparison with ExtraNet + NSR. ExtraNet+NSR has worse shadow and ghosting artifacts.

2 ADDITIONAL METRICS

We report the results of our pipeline comparing to the baselines in an additional metric structural similarity index measure (SSIM). Table 2 shows the comparison with different spatial super sampling baselines. Note that although NSR achieves the best result, it requires more computation resource and is less temporal stable (please refer to the supplementary video).

Table 3 shows the SSIM results of the frame generation baselines compared with our modules. Table 4 shows the SSIM results of frame generation baselines compared with our full pipeline, which means we only require half resolution as inputs than the baselines. Although our SSIM is slightly worse than baselines, we show less artifacts in visual comparison. Please refer to the supplementary video.

3 COMPARISON AGAINST SEQUENTIALLY APPLYING EXTRANET AND NSR

We compare with naively combining a spatial super sampling method (NSR [Xiao et al. 2020]) and a temporal super sampling method (ExtraNet [Guo et al. 2021]). Note that there is no trivial way to sequentially apply NSR first and then apply ExtraNet since NSR doesn't upscale the G-buffers and the upscaled rendered images are anti-aliased while ExtraNet requires aliased inputs. Thus, we sequentially apply ExtraNet first and then apply NSR as our baseline.

Table 5 shows the quantitative results of our method and the baseline, and Figure 1 shows the visual comparison. Since the frames generated from ExtraNet are not reliable and there is no special design for temporal coherence. Our method has less artifacts in disoccluded areas (metal string on the top) and shadows and more temporal stable (thin wires on the ground). Besides, NSR+ExtraNet is slower than ours (Ours (13.7ms) and NSR+ExtraNet (79.6ms) for 1080p's inputs). Please refer to the supplementary video for more details.

Table 6: Runtime (milliseconds) breakdown of our framework and rendering time. The resolution refers to the output resolution of our whole framework. The inference time of our framework is independent to the scene complexity. Since the G-buffer generation time and rendering time vary from scene to scene, we report the corresponding time in *BUNKER* as an example.

	1080p	1440p	2160p
Low-res Shading	10.3	17.2	33.2
G-Buffers Generation	0.28	0.31	0.55
G-buffer guided Warping	0.4	0.7	1.7
Shading Refinement	1.2	1.7	2.9
ExtraSS Network	2.5	4.1	9.1
Ours-ExtraSS total	4.38	6.81	14.25
Ours-SS total	12.8	21.3	42.3
High-res Shading	33.2	60.5	136.5

Table 7: Runtime (milliseconds) of NSR inference time.

	1080p	1440p	2160p
Low-res Shading	10.3	17.2	33.2
NSR	17.1	30.1	67.6
NSR total	27.4	37.3	100.8
Ours-SS total	12.8	21.3	42.3
Ours-ExtraSS total	4.38	6.81	14.25

4 NETWORK STRUCTURE

We report the details of our ExtraSS network and FRNet in the Figure 2 and Figure 3.

5 PERFORMANCE

The runtime breakdown of our framework under different output resolutions is reported in Table 6. Note that Ours-SS is slower than Ours-ExtraSS because the time is the total time of generating one frame, where SS needs low resolution rendered frames while ESS doesn't.

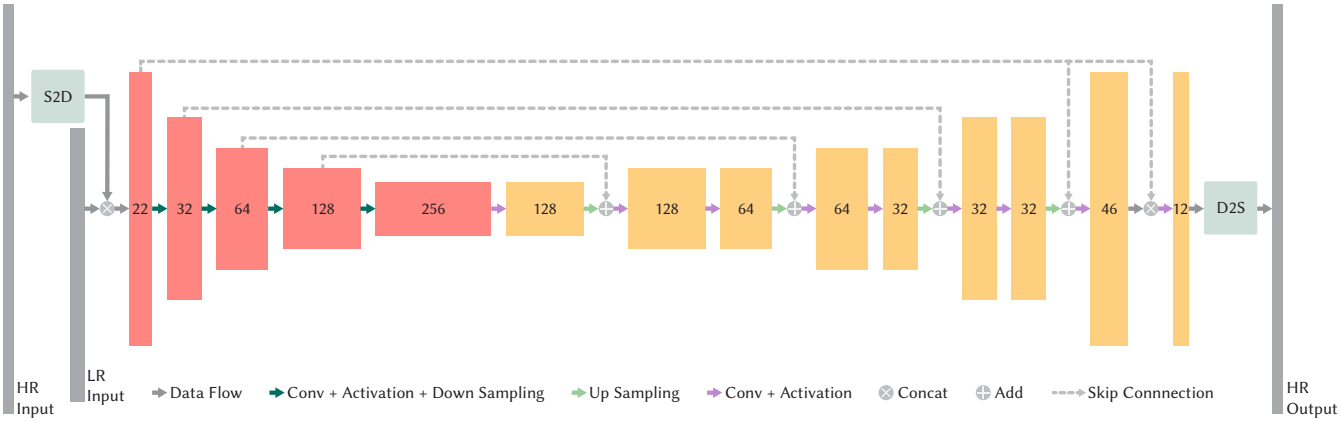

Figure 2: ExtraSS Net structure. S2D and D2S means Space-to-Depth [Redmon and Farhadi 2017] and Depth-to-Space respectively.

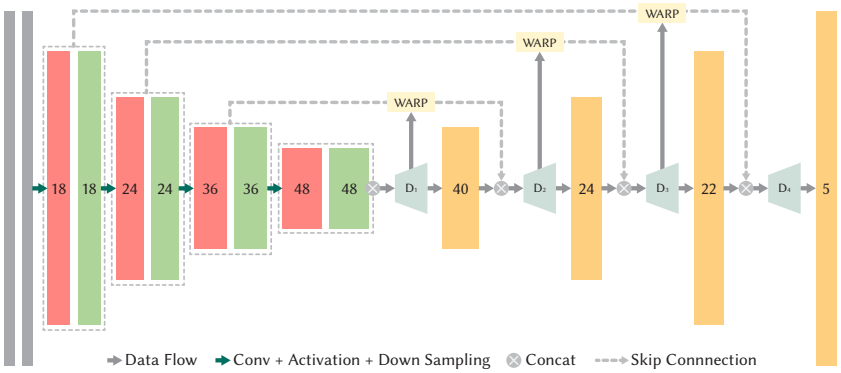

Figure 3: FRNet.

We also report the performance of NSR [Xiao et al. 2020] in table 7 under different output resolution for 2X spaital super sampling, which is significantly slower than our method.

6 ABLATION STUDIES

In this part we evaluate our designs on providing initialization for joint temporal and spatial super sampling.

G-buffer guided warping. Comparing to the previous methods, we propose a new warping method to fill the disoccluded areas and out-of-screen areas with sharper warped results. In the previous sections, the effectiveness of G-buffer guided warping method has already been demonstrated in the main paper with the comparison of frame generation methods and prior warping methods. We further evaluate the importance of our G-buffer guided warping in our full pipeline. Table 8 shows the quantitative results of replacing G-buffer guided warping with regular warping.

Flow based shading refinement. Flow based shading refinement is one key design to fix the incorrect shadings after G-buffer guided warping. IFRNet [Kong et al. 2022] already shows that directly predicting optical flows for all motions will generate blurry results in geometries and also missing some details. Thus, we only evaluate other alternatives of refine shading after G-buffer guided warping.

Table 8: Ablation study of our G-buffer guided warping. Ours refers to our full pipeline and Ours (Regular Warping) refers to replacing G-buffer guided warping with regular warping.

	Scenes	Ours (Regular Warping)	Ours
PSNR	BUNKER	25.84	27.81
	FOREST	19.27	19.86
	SEQUENCER	35.21	37.57
	INFILTRATOR	28.15	29.91
SSIM	BUNKER	0.847	0.888
	FOREST	0.591	0.640
	SEQUENCER	0.975	0.986
	INFILTRATOR	0.949	0.966

Figure 4 shows the final results of using flow based neural network to refine the shading and directly using Unet to refine the shading. Our flow based shading refinement generates sharper shadows, which leads to better temporal consistency.

Temporal Loss. Temporal Loss $\mathcal{L}_t$ is used for increasing temporal coherence of our pipeline between SS frames and ESS frames. It is also useful to improve the quality of SS and ESS frames since

FRNet Unet

Figure 4: A comparison of our flow model and Unet. Our method generates sharper shadows.

Table 9: Ablation study of temporal loss in terms of PSNR and SSIM.

		w/o Temporal Loss		w Temporal Loss	
	Scenes	Ours-SS	Ours-ESS	Ours-SS	Ours-ESS
PSNR	BUNKER	27.81	27.30	28.25	27.81
	FOREST	19.82	19.76	19.89	19.86
	SEQUENCER	37.44	37.28	37.75	37.57
	INFILTRATOR	29.99	29.89	30.04	29.91
SSIM	BUNKER	0.885	0.878	0.895	0.888
	FOREST	0.635	0.631	0.642	0.640
	SEQUENCER	0.964	0.964	0.986	0.986
	INFILTRATOR	0.984	0.984	0.967	0.966

Table 10: The memory usage (Gigabytes) of neural networks for our method and baseline methods under different output resolution.

	1080p	1440p	2160p
ExtraNet	5.26	9.28	20.83
IFRNet	4.60	7.83	18.24
NSR	9.88	17.56	39.44
Ours	1.07	1.90	4.20

it explicitly constrains the high level features similarity between them. Table 9 shows the results of removing temporal loss.

7 MEMORY

We report memory usage of networks for our pipeline and baselines in Table 10 under different resolutions.

8 MORE DISCUSSIONS

Distributing samples over spatial or temporal side. With the same rendering computation cost, the rendering samples could be distributed into either more spatial side (higher resolution but lower frame rates) or more temporal side (lower resolution but higher frame rates). Since ours is the first one to combine spatial/temporal super sampling, to the best of our knowledge, there is no experiment/theory of what is the optimal way to distribute samples over spatial and temporal side. However, under a given target resolution and frame rate, we show better or comparable quality than distributing all over temporal side (with spatial super sampling only methods and 2X more samples) or distributing all over spatial side (with temporal super sampling only methods and 4X more samples). Therefore, a better strategy of distributing rendering sample may lead to a better quality and further boost our method in the future.

Generalization ability. We currently select 4 representative scenes with glossy materials (Bunker), complex geometries (Forest), a large-scale game-like scene (Infiltrator) and a close-up view of a moving character (Sequencer), which cover wide ranges of data characteristics. However, to demonstrate generalization ability, it requires to collect a large scale dataset with more data characteristics and rendering styles like DLSS and XeSS. In the future, it will be interesting to train our models over more scenes to make it be a general model.

REFERENCES

Michael Berblinger and Christoph Schlier. 1991. Monte Carlo integration with quasi-random numbers: some experience. *Computer physics communications* 66, 2-3 (1991), 157–166.

Jie Guo, Xihao Fu, Liqiang Lin, Hengjun Ma, Yanwen Guo, Shiqiu Liu, and Ling-Qi Yan. 2021. ExtraNet: real-time extrapolated rendering for low-latency temporal supersampling. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–16.

Nima Khademi Kalantari, Ravi Ramamoorthi, et al. 2017. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.* 36, 4 (2017), 144–1.

Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. 2022. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1969–1978.

Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7263–7271.

Lei Xiao, Salah Nouri, Matt Chapman, Alexander Fix, Douglas Lanman, and Anton Kaplanyan. 2020. Neural supersampling for real-time rendering. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 142–1.