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

Songyin Wu
s_wu975@ucsb.edu
University of California, Santa Barbara
USA

Sungye Kim
sungye.kim@intel.com
Intel Corporation
USA

Zheng Zeng
zhengzeng@ucsb.edu
University of California, Santa Barbara
USA

Deepak Vembar
deepak.s.vembar@intel.com
Intel Corporation
USA

Sangeeta Jha
sangeeta.jha@intel.com
Intel Corporation
USA

Anton Kaplanyan
anton.kaplanyan@intel.com
Intel Corporation
USA

Ling-Qi Yan
lingqi@cs.ucsb.edu
University of California, Santa Barbara
USA

ABSTRACT

We introduce ExtraSS, a novel framework that combines spatial super sampling and frame extrapolation to enhance real-time rendering performance. By integrating these techniques, our approach achieves a balance between performance and quality, generating temporally stable and high-quality, high-resolution results. Leveraging lightweight modules on warping and the ExtraSSNet for refinement, we exploit spatial-temporal information, improve rendering sharpness, handle moving shadings accurately, and generate temporally stable results. Computational costs are significantly reduced compared to traditional rendering methods, enabling higher frame rates and alias-free high resolution results. Evaluation using Unreal Engine demonstrates the benefits of our framework over conventional individual spatial or temporal super sampling methods, delivering improved rendering speed and visual quality. With its ability to generate temporally stable high-quality results, our framework creates new possibilities for real-time rendering applications, advancing the boundaries of performance and photo-realistic rendering in various domains.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Image manipulation**; *Antialiasing*.

KEYWORDS

extrapolation, super resolution, low latency, warping

ACM Reference Format:
Songyin Wu, Sungye Kim, Zheng Zeng, Deepak Vembar, Sangeeta Jha, Anton Kaplanyan, and Ling-Qi Yan. 2023. ExtraSS: A Framework for Joint Spatial Super Sampling and Frame Extrapolation. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3610548.3618224

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12.
https://doi.org/10.1145/3610548.3618224

1 INTRODUCTION

With the recent breakthrough of graphics hardware, real-time rendering has made significant progress and is widely used in movies, animations, and games. However, with the increasing demand for photo-realism in quality, the challenge remains to achieve real-time performance. Balancing quality and performance becomes a trade-off – enhancing one aspect often leads to a deterioration in the other aspect.

There are various methods trying to address this problem by decreasing the computation cost for a single frame to increase the performance while keeping similar quality. One approach commonly used in real-time rendering engines is spatial super sampling (Spatial SS), which leverages temporal information from previous frames to generate the current frame from a low-resolution rendered image. Many commercial productions have already achieved great quality in real-time rendering such as DLSS [Liu 2020], XeSS [Intel 2022] and FSR [AMD 2021]. Although they utilize temporal information, spatial super sampling only focuses on increasing spatial resolution. On the other hand, temporal super sampling (Temporal SS), a technique also known as frame generation, reduces the computational cost by generating new frames based on existing frames. Unlike spatial super sampling, temporal super sampling actually hallucinates more frames over time.

Although these methods improve performance by employing temporal or spatial super sampling individually, none of them have attempted to unify them into a single framework[1]. Combining extrapolation and spatial super sampling poses challenges. It uses less rendered inputs comparing to individual spatial (fewer frames) or temporal (smaller resolution) super sampling methods while

[1]Specifically note that DLSS 3's details are not released to the general public. For completeness, we still discuss with DLSS 3 in our work, treating it as spatial super sampling plus optical flow interpolation as generally believed.

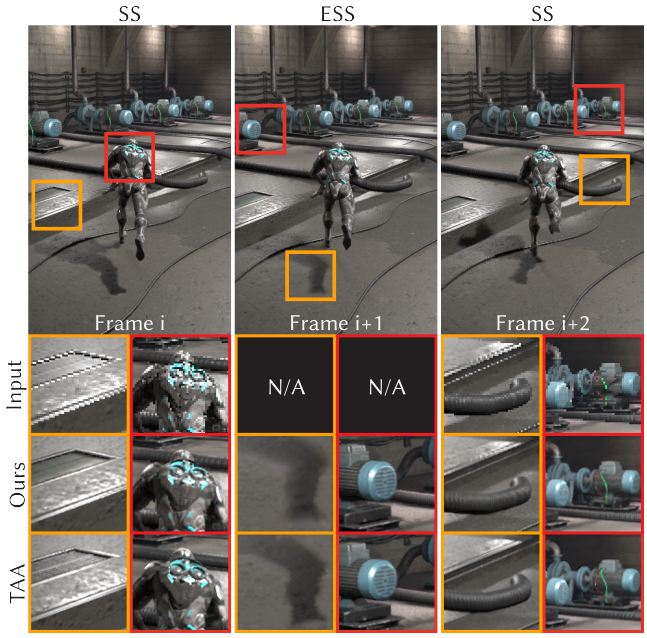

Figure 1: We propose a framework for joint spatial super sampling and frame extrapolation. With only low resolution inputs every other frame, we are able to perform spatial super sampling (for the frames shown on the left and right), or joint spatial and temporal super sampling (for the frame in the middle). Our method not only generates more frames over time but also results in comparable quality against Temporal Anti-Aliasing (TAA) at native high resolutions. The resolution are 540P for Input and 1080P for Output and TAA here. SS means spatial super sampling only frame and ESS means joint extrapolation and spatial super sampling.

generating the same output. Additionally, simply combining spatial super sampling methods with frame generation methods leads to bad performance because of redundant computation.

We analyze the possibilities of building a unified pipeline for both spatial super sampling and three modules to solve the problem. Our key idea is using lightweight modules to provide a good extrapolated initialization and use a neural network to further refine the results of extrapolation and generate final alias-free high resolution images. For initialization, we use motion vectors with G-buffer guided information to re-use a large area of spatial-temporal information to generate sharper warped results and solve the problem in disoccluded areas[2]. Then we design a lightweight flow based model to further refine the moving shading including shadows, highlights and reflections, to ensure consistent movement on both geometries and shading. Lastly, thank to our previous two designs, our ExtraSS network takes unified inputs for the frames with low resolution rendered images and without low resolution rendered images, to generate high resolution rendered images and refine the warped results.

[2]Disoccluded area refers to parts of the frame that were occluded but then, in the next frame, are no longer occluded.

We evaluate our method on Unreal Engine 4 [Epic Games 2022] and demonstrate better performance and comparable (or better) quality comparing to traditional spatial or temporal only super sampling methods.

2 RELATED WORK

2.1 Motion Vectors and Warping

Since both spatial and temporal super sampling re-use the temporal information in the existing frames, motion vectors and warping algorithms are key parts of these methods. Similar to the optical flow that tracks the movement of each pixel, motion vectors generated from the rendering engine are usually more accurate. Yang et al. [2020] have studied on various warping methods in order to use the temporal information more accurately. However, the warped methods usually contain incorrect shading in disoccluded areas and some indirect shadings, which causes ghosting and lagging artifacts. To solve this problem, it requires additional information from current rendered images to reject and rectify the pixels' values, which are not available in our extrapolated frames. Zeng et al. [2021] proposed a more reliable motion vector to solve disoccluded areas and shading movement. But it requires complex modification in calculating the motion vectors, which is hard to be integrated into a rendering engine, and it still fails in the disoccluded areas when the background becomes complicated.

2.2 Spatial Super Sampling and Anti-aliasing

Spatial SS shows significant improvement in real-time rendering by taking low resolution aliased input to generate high resolution alias-free results. Heuristic methods such as TAAU [Yang et al. 2020], FSR [AMD 2021] generate high resolution results by using handcraft weights to blend low resolution jittered images with warped high resolution images from previous output. Guo et al. [2022] train a classification network and a composition network to more wisely blend the results to have better temporal stability. Xiao et al. [2020] directly use a trained neural network to predict the current high resolution frame from five consecutive low resolution images. But it usually generates flickering results. DLSS 2.0 [Liu 2020] and XeSS [Intel 2022] are commercially deployed products for spatial super sampling, but the details are not publicly available.

Anti-aliasing removes the aliasing of rendered image while keeping in the same output resolution. It can be done by simply increasing the sample numbers of each pixel (SSAA, MSAA [Akeley 1993; Sousa et al. 2011]), or using spatial information nearby (FXAA and SMAA [Jimenez et al. 2011; Navarro and Gutierrez 2011]) or re-using temporal information (TAA [Karis et al. 2014]). The spatial super sampling usually removes the aliasing when increasing the spatial resolution. Since our method contains both spatial and temporal super sampling, unlike ExtraNet [Guo et al. 2021], we don't need additional anti-aliasing modules after our outputs.

2.3 Frame Interpolation

Frame interpolation is one approach of frame generation or temporal SS. It depends on previous frames and latter frames to generate the new frames between them. There are several approaches to generate a new frame including optical flow based methods[Baker et al. 2011; Jiang et al. 2018; Shen et al. 2021; Xue et al. 2019], kernel

based methods[Niklaus et al. 2017, 2021; Park et al. 2020], phase based methods[Meyer et al. 2015; Vogels et al. 2018] and direct prediction [Choi et al. 2020; Kalluri et al. 2023; Long et al. 2016]. Flow based methods show some promising results in rendering pipelines such as Neural Frame Interpolation (NFI) [Briedis et al. 2021] and DLSS 3 [Liu 2020]. Some video interpolation methods [Bao et al. 2019; Kong et al. 2022; Reda et al. 2022] also have been proposed to increase the frame rate but often have worse quality due to lack of G-buffers information. Furthermore, since interpolation based method rely on the latter rendered frames, it brings inevitable latency when generating new frames.

2.4 Frame Extrapolation

Frame extrapolation is another way to increase the frame rate by only using the information from prior frames. Li et al. [2022] proposed an optical flow based method to predict flow based on previous flows and then warp the current frame to the next frame. ExtraNet [Guo et al. 2021] uses occlusion motion vectors with neural networks to handle disoccluded areas and shading changes with G-buffers information. Their methods fail when the scene becomes complex and generate artifacts in the disoccluded areas. Furthermore, it requires higher resolution inputs since they only generate new frames. We are the first one to propose a joint framework to solve both spatial super sampling and frame extrapolation together while staying efficient and high quality.

Note that NVIDIA DLSS 3 is a combination of super sampling and *interpolation* since it generates intermediate frames[3]. The interpolation based method will introduce extra latency for the rendering pipeline, so it requires an additional modules NVIDIA Reflex to decrease the latency. However, NVIDIA Reflex decreases the latency by reducing the bottleneck between CPU and GPU, and it doesn't eliminate the latency from the frame interpolation.

3 PROBLEM ANALYSIS

Before we introduce our approach for joint spatial super sampling and frame extrapolation, we first specify the problems and goals and analyze the challenges to motivate our design choices.

3.1 Design Choices

Problem Formulation. Our method aims to increase the spatial resolution (spatial SS) and generate new extrapolated frames (temporal SS) together in a single framework. Specifically, given a stream of rendered images $\{i_t\}$ and an optional auxiliary buffer (usually refers to G-buffer) stream $\{g_t\}$, Spatial SS aims to generate higher resolution image stream $\{I_t\}$ where I_t is δ times larger than i_t in both height and width, and Temporal SS aims to generate additional frames among original frames $\{i_{t+\varepsilon}\}$. Temporal Super Sampling can be categorized into frame interpolation if $\{i_{t+\varepsilon}\}$ depends on both previous and latter rendered frames, and frame extrapolation if $\{i_{t+\varepsilon}\}$ does not depend on the future rendered frames.

Our task here is jointly increasing spatial resolution and extrapolating frames with twice larger in width and height, and one extrapolated frame for every rendered frame. The input is the low resolution rendered image stream $\{i_t\}$ and the G-buffer stream $\{g_t, g_{t+1}\}$ and output is a stream of high resolution rendered images with extra frames $\{I_t, I_{t+1}\}$. Our extrapolated super sampled results $\{I_{t+1}\}$ depends on previous rendered images $\{i_t\}$, warped previous high resolution images $\{I_t\}$ and current G-buffers $\{g_{t+1}\}$. And our spatial super sampled results $\{I_t\}$ depend on warped previous high resolution images $\{I_{t-1}\}$ and current rendered low images $\{i_t\}$. Since we don't use any information from the next frames, our method is an extrapolating method and we will discuss the advantage of extrapolation below.

Interpolation vs. Extrapolation. Frame interpolation and extrapolation are two key methods of Temporal Super Sampling. Usually frame interpolation generates better results but also brings latency when generating the frames. Note that there are some existing methods such as NVIDIA Reflex [NVIDIA 2020] decreasing the latency by using a better scheduler for the inputs, but they cannot avoid the latency introduced from the frame interpolation and is orthogonal to the interpolation and extrapolation methods. The interpolation methods still have larger latency even with those techniques. Frame extrapolation has less latency but has difficulty handling the disoccluded areas because of lacking information from the input frames. Our method proposes a new warping method with a lightweight flow model to extrapolate frames with better qualities to the previous frame generation methods and less latency comparing to interpolation based methods.

G-buffers. Generating G-buffers is much cheaper than rendering a frame and they provide strong clues in the extrapolation and the use of G-buffer has been proven to increase the quality of frame generation [Briedis et al. 2021]. But the time of generating them and the memory cost cannot be ignored as discussed in ExtraNet [Guo et al. 2021]. We only use low resolution G-buffers in our pipeline for efficiency and better memory usage.

3.2 Challenges

Disoccluded and Out-of-screen Areas. Temporal Super Sampling usually re-uses the information from the history frames by motion vectors. Therefore, it becomes harder when the temporal information is not available, where we call them disoccluded areas and out-of-screen areas. Frame interpolation [Baker et al. 2011; Choi et al. 2020; Kong et al. 2022; Meyer et al. 2015; Niklaus et al. 2017] usually fills the disoccluded areas from the latter frames but it increases the latency as discussed before. ExtraNet utilizes more reliable occlusion motion vectors proposed by Zeng et al. [2021] and an in-painting network to fill in those areas but still fails when shading becomes complex. There are some other hole-filling methods for image in-painting which are usually too slow for real time rendering as discussed by Guo et al [2021]. Although the exact position in the previous frame is occluded, the shading information nearby is still useful to predict the shading for the disoccluded pixels. Therefore, we propose a G-buffer guided warping method to re-use the temporal and spatial information to generate better warped frames.

Temporal Stability. Temporal stability is a crucial part of rendering. Inconsistency between frames causes flickering artifacts in the rendered results. Motion vectors can be used for tracking objects to keep consistency but accurate shading and shadow's motion

[3] https://www.nvidia.com/en-us/geforce/news/dlss3-ai-powered-neural-graphics-innovations/

tracking is not perfect. Optical flow based methods[Baker et al. 2011; Jiang et al. 2018; Xue et al. 2019] are able to track the motion of shadows and some other shading changes but they are usually interpolation based methods and require heavy computation to estimate the optical flow per pixel which is not practical in real time rendering. ExtraNet proposes a history encoder to handle the shading and shadow's movement but generates blurry shadows and sometimes are inaccurate. Our method utilizes the G-buffer guided warping to first decouple the motion of the shading directly corresponding to the geometries and then use a lightweight flow-based model to handle the remaining shading motions. Furthermore, we design temporal constraints for our ExtraSS network to improve the stability in our joint spatial and temporal super sampling process.

Joint Temporal and Spatial Super Sampling. Our method combines both spatial and temporal super sampling to generate new frames with high resolution. It requires two different types of inputs: half frames with low resolution rendered images and the other half frames only contain low resolution G-buffers. Simply using two individual methods on frame generation and super resolution is redundant, since both of them re-use temporal and spatial information in the history. We design lightweight warping modules to unify the different inputs and provide good extrapolated initialization for our joint framework. Then, a neural network is used to generate high resolution images and refine the extrapolated frame, which is more efficient than a simple combination.

4 EXTRAPOLATION AND SUPER SAMPLING

4.1 System Overview

We propose a comprehensive framework aimed at concurrently augmenting spatial resolution and frame rate while minimizing latency. The overview is shown in Figure 2. Our method consists of three main components: G-buffer guided warping (Sec. 4.2), decoupled shading refinement (Sec. 4.3) and joint Extrapolation and Super Sampling (ExtraSS) network (Sec. 4.4). We will discuss the details of each component in the following sections.

4.2 G-buffer Guided Warping

As discussed the challenges of disoccluded areas and out-of screen areas in Sec. 3.2, the traditional motion vectors and occlusion motion vectors [Zeng et al. 2021] generate ghosting artifacts and blur the warped images as shown in Figure 4. We propose a new warping method, G-buffer guided warping, to efficiently warp previous frames to the current with better sharpness and less ghosting. To begin with, the rendered images are first demodulated by the base color to remove the high frequency shading from textures and then modulated back after the warping. Sec. 3.1 addressed the importance of G-buffers and they can be used for calculating the similarity of pixels that have similar radiance, which provides a good clue to re-use the spatial information. Thus we propose using joint bilateral filters [Kopf et al. 2007] with G-buffers' similarity to re-use the spatial shading information, as illustrated in Figure 3. Specifically, it will consider a large area of pixels near the reprojected pixel and use G-buffers' similarity with screen space distance to blend them together as the warped pixel. To handle the large disoccluded and out-of-screen areas, we use an *À*-trous like hierarchy structure with four levels for kernels and different stride size [Dammertz et al. 2010]. Then only 4 small kernels are needed for a large receptive field in order to re-use distant pixels for disoccluded areas.

To calculate the warped pixels' value, first we define the area of candidate pixels in the previous frame. Let $\bar{x}$ denotes the position of current pixel in the previous frame by re-projection and $N(\bar{x})$ be the pixels set in the *À*-trous kernel around $\bar{x}$. It is defined as

$$N(\bar{x}) = \{\bar{x} + (k * s, l * s) \mid k \in \{-r, \cdots, r\}, l \in \{-r, \cdots, r\}, \\ r = 1, s \in \{1, 3, 5, 9\}\}. \tag{1}$$

Then, similar to the joint bilateral filter, we use two Gaussian function $G(\cdot|\mu, \sigma)$ to calculate the blending weight for every element y in the kernel $N(\bar{x})$. The blending weight is defined as

$$w_x(y) = G(g_t(x) - g_{t-1}(y) \mid \mu_0, \sigma_0) \times G(\bar{x} - y \mid \mu_1, \sigma_1). \tag{2}$$

where $g_t(\cdot)$ refers to the G-buffers in the t frame and we use *base-color* and *normal* as the feature for weight calculation. Once we calculate the blending weight for all pixels in $N(\bar{x})$, we select 4 pixels (N^*) with the largest blending weights and then normalize the blending weights for these 4 pixels, denoted as $\bar{w}_x(y)$. Let $f_g^t(i_{t-1})$ be the G-buffer guided warped frame from frame $t-1$ to frame t, then the warped pixel values $f_g^t(i_{t-1})[x]$ are calculated by

$$f_g^t(i_{t-1})[x] = \sum_{y \in N^*(\bar{x})} \bar{w}_x(y) * i_{t-1}[y] \tag{3}$$

In Figure 4, our method not only keeps sharper boundaries and details, but also fills disoccluded areas by using G-buffers' information to guide the spatial shading blending. Note that we only use this for low resolution images since only low resolution G-buffers are available.

4.3 Decoupled Shading Refinement with FRNet

As discussed in Sec. 3.2, our G-buffer guided warping can only handle shading that directly corresponds to the geometries, leaving some other shading motions unchanged such as shadows and glossy reflections. Instead of using optical flow to directly predict all flows from the previous frame to the current frame, we first use our warping method to handle most shadings and *decouple* them from the latter shading refinement part. Then, the remaining incorrect shadings will be fixed by a lightweight flow based neural network called Flow-based Refinement network (FRNet).

The network's input contains two consecutive down-sampled frames $[i_{t-1}^-, i_{t-3}^-]$ and the down-sampled roughness r_t^- which is a hint to tell whether an area is easy to have glossy or specular reflections. The two consecutive frames are first warped to the current frame by using the G-buffer guided warping. The warped frames have almost correct shading and more importantly fill the disoccluded and some out-of-screen areas. The FRNet only predicts the flow to warp the $f_g^t(i_{t-1}^-)$ with correct shading.

$$\bar{i}_t^- = \text{FRNet}(f_g^t(i_{t-1}^-), f_g^t(i_{t-3}^-), r_t^-) \tag{4}$$

The use of down-sampled frames for the FRNet is to accelerate our shading refinement process. Figure 5 shows the lower resolution output of FRNet can be upscaled back to the original resolution with correct shadings and details. It can be done by blending with G-buffer guided warping results $\hat{i}_t = f_g^t(i_{t-1})$. Let $U(\cdot)$ be the

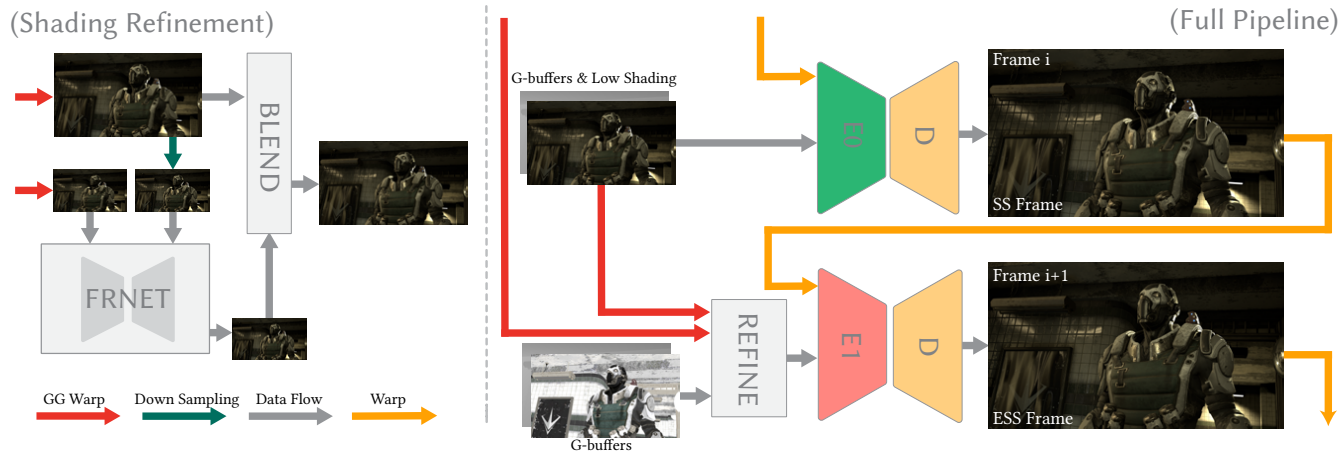

Figure 2: An overview of our pipeline. The left part shows the workflow of our shading refinement part and the right part shows the workflow of the whole pipeline. REFINE module in the right refers to the shading refinement process (the left part). GG Warp refers to G-buffer guided warping and Warp refers to regular warping. For two consecutive frames, ESS frames will use G-buffer guided warp and shading refinement module to get low resolution rendered images and SS frames directly take ground truth low resolution rendered images from rendering engine.

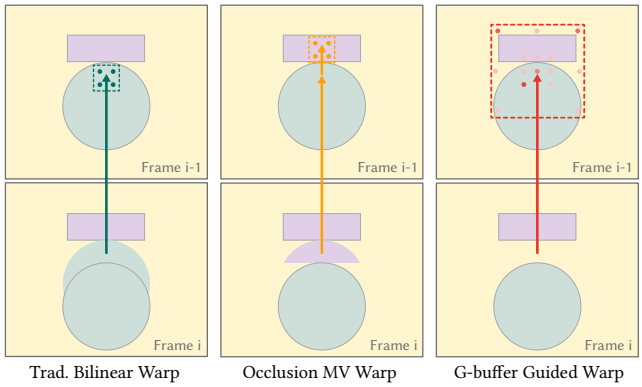

Figure 3: Different Warping methods, either using traditional or occlusion motion vectors (using objects' motion to refine the traditional motion vector (shorter arrow)), perform point-to-point matches and can result in wrong shading re-use. We propose an *À*-trous like kernel to explore a larger area to better estimate the shading in the disoccluded area.

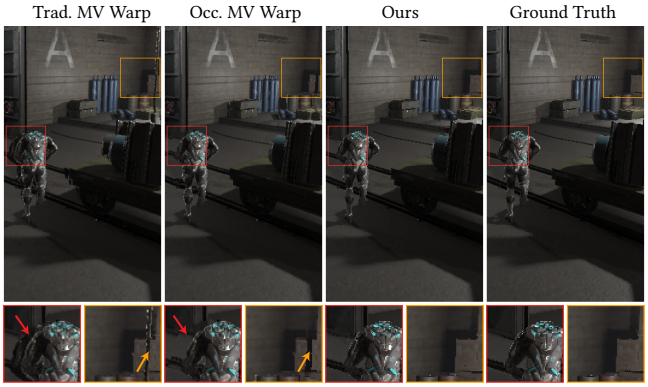

Figure 4: Comparison of different warping methods. Our method generates better warping results without ghosting artifact while both traditional warping and occlusion motion vectors generate incorrect shading in the disoccluded areas.

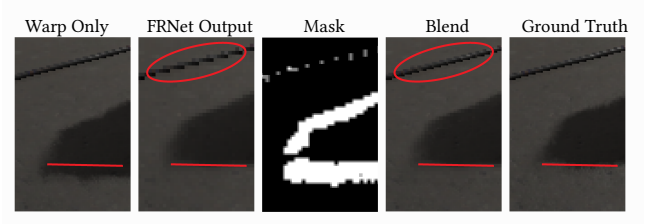

Figure 5: Output of FRNet and the blended result. The blended result contains shading details as well as correct shadow position.

nearest upsampling operation, refine results with correct shading is calculated by

$$\bar{i}_t = U(\bar{i}_t^-) \cdot \hat{m}(U(\bar{i}_t^-), \hat{i}_t) + \hat{i}_t \cdot (1 - \hat{m}(U(\bar{i}_t^-), \hat{i}_t)) \quad (5)$$

where $\hat{m}$ is the blending mask which will be introduced later. Since most shading of the scene has already been done by G-buffer guided warping, we only need to design the mask $\hat{m}$ to blend refined low resolution shadings with warped results to keep both refined shading and other details. Thus, we propose following blending strategy.

To generate high quality blending results, it requires two aspects: keeping details while resolving incorrect shading, and they lead to two criterion when we design the blending mask. The first one is low resolution shading is used only when the G-buffers are similar, which will keep the shading details in the original resolution.

We use *basecolor* and *normal* to calculate the G-buffers similarity. Second, the low resolution shading is only used for correcting the incorrect shadings in the warped results. Thus, only when difference is larger than a threshold, the low resolution shading will be used. Specifically, for the mask $\hat{m}$, we define n as *normal* and α as *basecolor*. Then the mask is calculated by

$$\hat{m}(U(\bar{i}_t^-), \hat{i}_t) = \left((1 - \frac{1}{\sqrt{3}}\|U(\alpha_t^-) - \alpha_t\|_2) \odot \mathrm{dot}(U(n_t^-), n_t) > \delta_1\right) \cap (\|U(\bar{i}_t^-) - \hat{i}_t\| / \|U(\bar{i}_t^-)\| > \delta_2), \tag{6}$$

where $\odot$ refers to the element-wise product, and δ_1, δ_2 are set to 0.1. As shown in Figure 5, the blended image not only has correct shadow position but also preserves more details of the thin wire on the top comparing to the FRNet's output.

As analyzed in the beginning of this section, since the most shadings are decoupled by G-buffer guided warping, the FRNet only handles a small part of incorrect shading. Therefore, only a lightweight flow based neural network is needed to refine the shadings. Please refer the details of FRNet in the supplementary.

Loss functions. We use image reconstruction loss $\mathcal{L}_r$ and feature space loss $\mathcal{L}_f$ to train our FRNet. Image reconstruction loss consists of Charbonnier loss $\rho(x)$ [Charbonnier et al. 1994] and census loss $\mathcal{L}_{\mathrm{cen}}(x, y)$ [Meister et al. 2018].

$$\mathcal{L}_r = \rho(\hat{i}_t - i_t^{\mathrm{gt}}) + \mathcal{L}_{\mathrm{cen}}(\hat{i}_t, i_t^{\mathrm{gt}}) \tag{7}$$

Feature space loss calculates the census distance between extracted features of ground truth frame and our predicted frame's features:

$$\mathcal{L}_f = \sum_{k=1}^{3} \mathcal{L}_{\mathrm{cen}}(\bar{\phi}_t^k, \phi_t^k) \tag{8}$$

where $\bar{\phi}_t^k$ refers to the k-th feature of predicted frame from the FRNet and ϕ_t^k refers to k-th feature of ground truth frame. Feature space loss encourages the FRNet to learn high level features to generate more accurate shadings. Combining previous two losses, the final loss to train the FRNet is formulated as

$$\mathcal{L}_{\mathrm{FR}} = \mathcal{L}_r + \lambda \mathcal{L}_f, \tag{9}$$

where λ is set to 0.001.

4.4 Joint Extrapolation and Super Sampling Network

Previous two components in our framework already provide good low resolution extrapolated images. Using them as inputs, we propose the Extrapolation Super Sampling (ExtraSS) network for joint spatial super sampling and extrapolation.

4.4.1 Network Structure. The inputs for t-th frame (spatial super sampling only (SS) frame) and $(t+1)$-th frame (spatial super sampling and extrapolation (ESS) frame) are different. We use separated encoders E_0 and E_1 to extract high level features for spatial super sampling only (SS) frames and joint extrapolation and spatial super sampling (ESS) frames. Let $f_r(\cdot)$ denotes the traditional warping. For t-th frame, the encoder E_0 takes current low resolution rendered frame i_t, scene depth d_t and warped high resolution image $f_r(\bar{I}_{t-1})$. For $(t+1)$-th frame, the encoder E_1 takes the G-buffer guided warped results $f_g^{t+1}(i_t)$, predicted results $\bar{i}_{t+1}$ from FRNet, warped high resolution frame $f_r(\bar{I}_t)$, the G-buffers including base color α_{t+1} and scene depth d_{t+1}. Then, a shared decoder D is used for generating the final high resolution results $\bar{I}_t$ or $\bar{I}_{t+1}$.

$$\begin{aligned} \bar{I}_t &= D(E_0(i_t, f_r(\bar{I}_{t-1}), d_t)) \\ \bar{I}_{t+1} &= D(E_1(f_g^{t+1}(i_t), \bar{i}_{t+1}, f_r(\bar{I}_t), \alpha_{t+1}, d_{t+1})) \end{aligned} \tag{10}$$

With the design of the G-buffer guided warping and shading refinement, the structure and the number of the parameters of E_0 and E_1 could be similar, so we don't need too much additional computation for the extrapolation frame. The networks follow the structure of Unet [Ronneberger et al. 2015] and the features extracted from the encoders will be added or concatenated to the features in the decoder by skip connections. Please refer to the supplementary for the details of ExtraSS network.

4.4.2 Losses Design. To preserve details and temporally stable joint spatial and temporal super sampling, we separate our losses into two parts to train our ExtraSS network.

The first part spatial loss $\mathcal{L}_s$ aims to increase the quality of spatial super sampling results. It contains occlusion loss $\mathcal{L}_{occ}$, $\mathcal{L}_1$ loss and VGG loss [Johnson et al. 2016] $\mathcal{L}_{\mathrm{vgg}}$. We use occlusion loss $\mathcal{L}_{\mathrm{occ}}$ to emphasize the ghosting areas, which is defined as

$$\mathcal{L}_{\mathrm{occ}} = \frac{\|m(I_t) \cdot \bar{I}_t - m(I_t) \cdot I_t^{\mathrm{gt}}\|_1}{\mathrm{sum}(m(i_t))}. \tag{11}$$

where $m(i_t)$ is the disoccluded areas' mask. Then, the full spatial loss is defined as

$$\mathcal{L}_s = \|\bar{I}_t - I_t^{\mathrm{gt}}\|_1 + \lambda_{\mathrm{occ}}\mathcal{L}_{\mathrm{occ}} + \lambda_{\mathrm{vgg}}\mathcal{L}_{\mathrm{vgg}}. \tag{12}$$

We set $\lambda_{\mathrm{occ}} = 1$ and $\lambda_{\mathrm{vgg}} = 0.01$.

The second part temporal loss $\mathcal{L}_t$ is used for generating temporally stable results. Instead of setting the losses on the final generated results, we use two encoder E_0 and E_1 to extract the features for the same frame by taking two types of input. And then we use l_1 loss to set constraints on these high level features. Specifically, the temporal loss is defined as

$$\mathcal{L}_t = \|\bar{I}_t - f_r(\bar{I}_{t-1})\|_1 + \sum_{k=1}^{4} l_1(\bar{\Phi}_t^k, \Phi_t^k) \tag{13}$$

where $\bar{\Phi}_t^k$ is the feature extracted from spatial super sampling only input by E_0, while Φ_t^k is the feature extracted from extrapolation and spatial super sampling input by E_1. Note that these two groups of inputs are from the same frame. We predict both ESS and SS results in the training and use them in turn during inference. Our final losses to train ExtraSS network are defined as

$$\mathcal{L} = \mathcal{L}_s + \lambda_t \mathcal{L}_t \tag{14}$$

where λ_t is set to 1.

4.5 Training Configuration

Our system contains two networks: FRNet and ExtraSSNet, which are trained separately with PyTorch framework [Paszke et al. 2019]. The Adam [Kingma and Ba 2014] optimizer with learning rate 0.0001, batch size 8 is used for training FRNet for 100 epochs. Then the parameters of FRNet is frozen and just used for providing the

Table 1: We compare with different spatial super sampling quality in PSNR 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.

Scenes	TAAU	NSR	Ours-SS	Ours-ESS	TAA
BUNKER (BK)	26.83	28.34	28.25	27.81	28.23
FOREST (FR)	19.21	18.76	19.89	19.86	20.16
SEQUENCER (SQ)	37.52	38.43	37.75	37.57	39.77
INFILTRATOR (IF)	30.03	30.26	30.04	29.91	32.05

input to the ExtraSSNet. ExtraSSNet also uses Adam optimizer with learning rate 0.0004 and batch size 12 for 120 epochs.

5 EXPERIMENTS AND COMPARISON

Our data generation and experiments are conducted with an AMD Ryzen9 5950X CPU and NVIDIA RTX 3090 GPU. Please refer to the supplementary for the details of data generation.

5.1 Comparison with Super Sampling Methods

Our system contains both spatial super sampling only (SS) frames and extrapolation with super sampled high resolution (ESS) frames, so we will evaluate the quality of these two parts separately. For fair comparison, we generate our results on the same test sequence twice with one starting with spatial super sampling frame and the other starting with extrapolated high resolution frame. In that case we have both ESS and SS results for every frame.

We compare our method with the Temporal Anti-Alising Upsampling (TAAU) [Yang et al. 2020], neural network based super sampling NSR [Xiao et al. 2020], commercial super sampling method NVIDIA DLSS 2 [Liu 2020]. The NSR baseline is re-implemented and trained on our dataset with the same configuration.

The reference images are generated by 64 SPP of super sampling anti-alising (SSAA) which are usually difficult to achieve in real-time rendering. We use temporal anti-aliasing (TAA) [Karis et al. 2014] running in the native resolution as an additional reference. For NVIDIA DLSS, we use the Unreal engine plugin of DLSS 2.3 with performance mode to generate results in Unreal engine 4.26 since that's the lowest version of Unreal engine supporting DLSS plugin. There are some differences in shading behaviors between Unreal 4.25 and Unreal 4.26 such as the shading at the back of the character.

In Figure 6, we show the visual comparison among TAAU [Yang et al. 2020], DLSS [Liu 2020], NSR [Xiao et al. 2020], ours, and two reference TAA [Karis et al. 2014] and SSAA. The full frame of our extrapolation with spatial super sampling (ESS) results are shown in the left and we crop 3 zoom-in views for the comparison among baselines. The quantitative results are reported in Table 1.

Note that our method only uses half of all the frames over time as input unlike other SS only methods. This means a larger stride/difference between the contents in consecutive input frames. Even in this case, our method still shows comparable results due to the good initialization provided by G-buffer guided warping and shading refinement modules. Even using the target resolution as inputs, TAA generates lagging glossy reflections while our ESS has correct shadings. Besides TAAU tends to generate blurry results and NSR has bad temporal consistency. Please refer to the supplementary materials for more results.

Table 2: PSNR (dB) values on different 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	30.10	21.50	35.85	30.63	29.52
IFRNet	26.14	17.43	35.33	26.83	26.43
Ours-W	30.74	21.26	40.02	33.10	31.28
Ours-E	31.80	22.00	40.06	33.29	31.79

5.2 Comparison with Frame Extrapolation/Interpolation Methods

To demonstrate our ability in both super sampling and extrapolation, we compare our methods with recent temporal SS methods, including frame interpolation and extrapolation. Note that interpolation methods will increase latency between user's inputs and rendering while extrapolation has less latency. Recently released Nvidia DLSS 3 utilizes optical flow to increase the frame rate in the game by interpolating frames, but the details of it are not available and it requires specific hardware support, so we compare with another optical flow based method IFRNet [Kong et al. 2022] as the interpolation-based frame generation method. And we also compare with ExtraNet [Guo et al. 2021] to show our better quality and performance on the extrapolation.

Our G-buffer guided warping and shading refinement modules provide a good extrapolated frame, which is before being SS-ed and in low resolution with aliasing. Therefore, we compare frame generation methods with two settings. The first setting compares our extrapolated frames from G-buffer guided warping and shading refinement modules with frame generation methods ExtraNet and IFRNet. The other setting compares our full framework with ExtraNet and IFRNet with the same output resolution, which means lower resolution of inputs is needed for our framework. We apply temporal anti-aliasing on the output of ExtraNet and IFRNet in the second setting for fair comparison.

In the visual comparison shown in Figure 7, all outputs in the lower resolutions are upscaled to the same resolution as others by nearest interpolation. Table 2 shows the quantitative results of our warping method and shading refinement modules comparing to the baselines without anti-aliasing, and Table 3 shows the quantitative results of our full models with frame generation methods in the same output resolution. In the first setting, our warping method and shading refinement modules show better results in both quantitative results and visual comparison. IFRNet will miss geometries when motion is large, and ExtraNet generates incorrect shading in the disoccluded areas and worse shadows. Our methods not only provide stable extrapolated frames, but also use much less time in

Table 3: PSNR (dB) values on different 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. Our input resolution is 2X smaller than other baselines.

	BK	FR	SQ	IF	Mean
ExtraNet	27.82	21.87	38.47	30.75	29.73
IFRNet	25.06	19.86	38.59	31.23	28.69
Ours-ESS	27.81	19.86	37.58	29.91	28.79

Table 4: We report the running time (milliseconds) for all methods under different input resolutions. The time doesn't include G-buffers generation and frame rendering. Note that for the same output resolution 1080p, Ours-ESS (4.1) is faster than ExtraNet (12.0).

	ExtraNet	IFRNet	NSR	Ours-W	Ours-E	Ours-SS	Ours-ESS
540p	3.3	5.8	17.1	0.4	1.6	2.5	4.1
1080p	12.0	19.5	67.6	1.7	4.6	9.1	13.7

the inference. For the second setting, we evaluate our model with frame generation methods for the same output resolutions. Our method shows comparable quantitative results, except in *FOREST* scene, given the complex geometries. Although the PSNR values are similar in other scenes, our method contains less artifacts in disoccluded areas and more accurate shadows in the visual comparison. Since the input resolution of our method is 2X smaller, our results are only slightly more blurry than baseline methods with native resolution. For the temporal stability evaluation and other results under these two settings, please refer to the supplementary materials.

5.3 Performance

Table 4 reports the performance of our method and other baseline methods. All neural networks are converted into TensorRT and evaluated in FP16. We show much better performance than baseline methods in the same output resolution while keeping comparable or better quality as analyzed before. Note that the time reported in Table 4 does not include the time of rendering and the baselines usually take more time on rendering (higher resolution or more frames). Please refer to the supplementary for more details about our framework's performance.

6 CONCLUSION AND FUTURE WORK

We have presented a framework for joint spatial super sampling and frame extrapolation. With G-buffer guided warping and shading refinement module, our framework is able to provide accurate extrapolated low resolution frames by utilizing G-buffers' information as well as spatial-temporal shading information. Furthermore, ExtraSSNet generates temporally consistent high resolution results by unifying the super sampling and extrapolation processes. Our framework generates spatially super sampled extrapolated frames effectively and achieves comparable quality and better performance than previous individual methods.

Our method has some limitations. Scenes with extremely complex geometries, such as FOREST, are challenging to almost all existing frameworks, including ours, with even less spatial and temporal information. Our method sometimes generates flickering around object boundaries and artifacts caused by error accumulation. Such artifacts are not observed in the low resolution extrapolated results, so it could be a limitation of our network design on the spatial super sampling task. This could be resolved by incorporating insights from other successful SS frameworks to our ExtraSS pipeline.

In the future, it would be interesting to explore the generalization ability to our framework by training and testing on multiple scenes, though the four scenes we currently select have already covered a variety of different types of scenes. Another direct extension is to verify the ability of multiple consecutive frame extrapolation and different scales of spatial super sampling (1.5X, 4X, etc.) rather than fixed.

ACKNOWLEDGMENTS

We thank Anton Sochenov for insightful discussion and German Ros for the help of the GPU cluster. This project is solely sponsored by Intel. And Ling-Qi Yan is also supported by gift funds from Adobe, Intel, Meta and XVerse.

REFERENCES

Kurt Akeley. 1993. Reality engine graphics. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 109–116.

AMD. 2021. AMD FidelityFX™ Super Resolution. https://www.amd.com/en/technologies/fidelityfx-super-resolution Accessed: 2023-05-23.

Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. 2011. A database and evaluation methodology for optical flow. *International journal of computer vision* 92 (2011), 1–31.

Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. 2019. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3703–3712.

Karlis Martins Briedis, Abdelaziz Djelouah, Mark Meyer, Ian McGonigal, Markus Gross, and Christopher Schroers. 2021. Neural frame interpolation for rendered content. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, Vol. 2. IEEE, 168–172.

Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 2020. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10663–10671.

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik PA Lensch. 2010. Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*. 67–75.

Epic Games. 2022. *Unreal Engine*. https://www.unrealengine.com

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.

Yu-Xiao Guo, Guojun Chen, Yue Dong, and Xin Tong. 2022. Classifier Guided Temporal Supersampling for Real-time Rendering. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 237–246.

Intel. 2022. Intel® Arc™- Xe Super Sampling. https://www.intel.com/content/www/us/en/products/docs/discrete-gpus/arc/technology/xess.html Accessed: 2023-05-23.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9000–9008.

Jorge Jimenez, Diego Gutierrez, Jason Yang, Alexander Reshetov, Pete Demoreuille, Tobias Berghoff, Cedric Perthuis, Henry Yu, Morgan McGuire, Timothy Lottes, et al. 2011. Filtering approaches for real-time anti-aliasing. *SIGGRAPH Courses* 2, 3 (2011), 4.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European*

Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer, 694–711.

Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. 2023. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2071–2082.

Brian Karis, Natasha Tatarchuk, Michal Drobot, Nicolas Schulz, Jerome Charles, and Theodor Mader. 2014. Advances in real-time rendering in games, part I. In *ACM SIGGRAPH 2014 Courses*. ACM, 10.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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.

Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. 2007. Joint bilateral upsampling. *ACM Transactions on Graphics (ToG)* 26, 3 (2007), 96–es.

Zhan Li, Carl S Marshall, Deepak S Vembar, and Feng Liu. 2022. Future Frame Synthesis for Fast Monte Carlo Rendering. In *Graphics Interface 2022*.

Edward Liu. 2020. DLSS 2.0-Image reconstruction for real-time rendering with deep learning. In *GPU Technology Conference (GTC)*.

Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. 2016. Learning image matching by simply watching video. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer, 434–450.

Simon Meister, Junhwa Hur, and Stefan Roth. 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. 2015. Phase-based frame interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1410–1418.

Fernando Navarro and Diego Gutierrez. 2011. Practical Morphological Antialiasing. *GPU Pro* 2 (2011), 95–113.

Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*. 261–270.

Simon Niklaus, Long Mai, and Oliver Wang. 2021. Revisiting adaptive convolutions for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1099–1109.

NVIDIA. 2020. NVIDIA Reflex. https://www.nvidia.com/en-us/geforce/technologies/reflex/ Accessed: 2023-05-23.

Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. 2020. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 109–125.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. 2022. FILM: Frame Interpolation for Large Motion. In *European Conference on Computer Vision (ECCV)*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

Wang Shen, Wenbo Bao, Guangtao Zhai, Charlie L Wang, Jerry W Hu, and Zhiyong Gao. 2021. Prediction-assistant frame super-resolution for video streaming. *arXiv preprint arXiv:2103.09455* (2021).

Tiago Sousa, Nick Kasyan, and Nicolas Schulz. 2011. Secrets of CryENGINE 3 graphics technology. In *ACM SIGGRAPH*, Vol. 1.

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–15.

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.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision* 127 (2019), 1106–1125.

Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A survey of temporal antialiasing techniques. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 607–621.

Zheng Zeng, Shiqiu Liu, Jinglei Yang, Lu Wang, and Ling-Qi Yan. 2021. Temporally Reliable Motion Vectors for Real-time Ray Tracing. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 79–90.

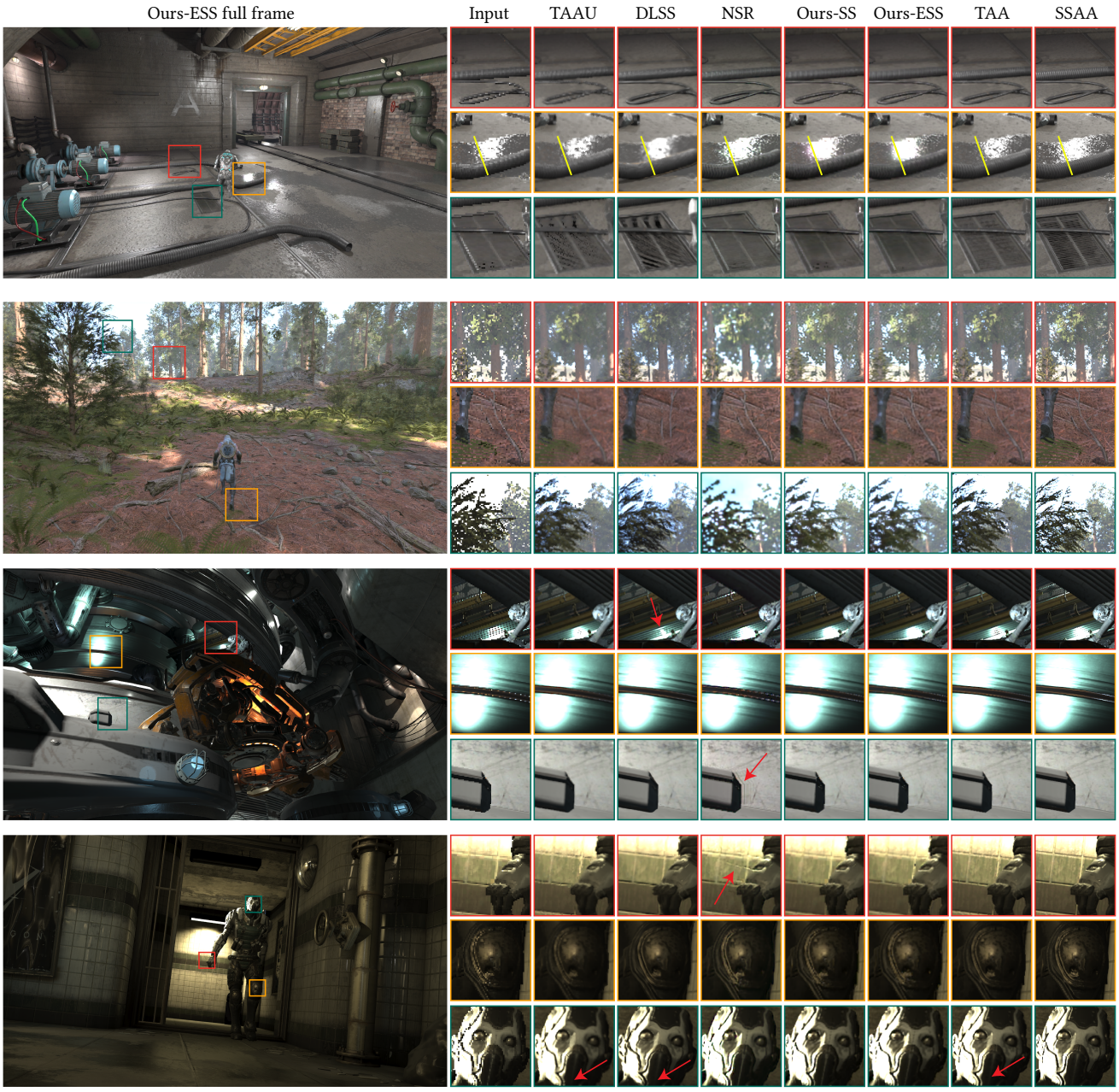

Figure 6: We compare our method with other super sampling baselines including TAAU, NSR and DLSS. We use TAA and SSAA results as our reference. To demonstrate the quality of our SS and ESS results, we evaluate on the same sequence with 1 frame offset of the start frame so we can generate SS and ESS results for the same frame. Our method achieves comparable results and even better in some situations such as less ghosting artifacts and sharper results. Note that although the results of NSR look good here, it has bad temporal stability. Please refer to the supplementary video for more details.

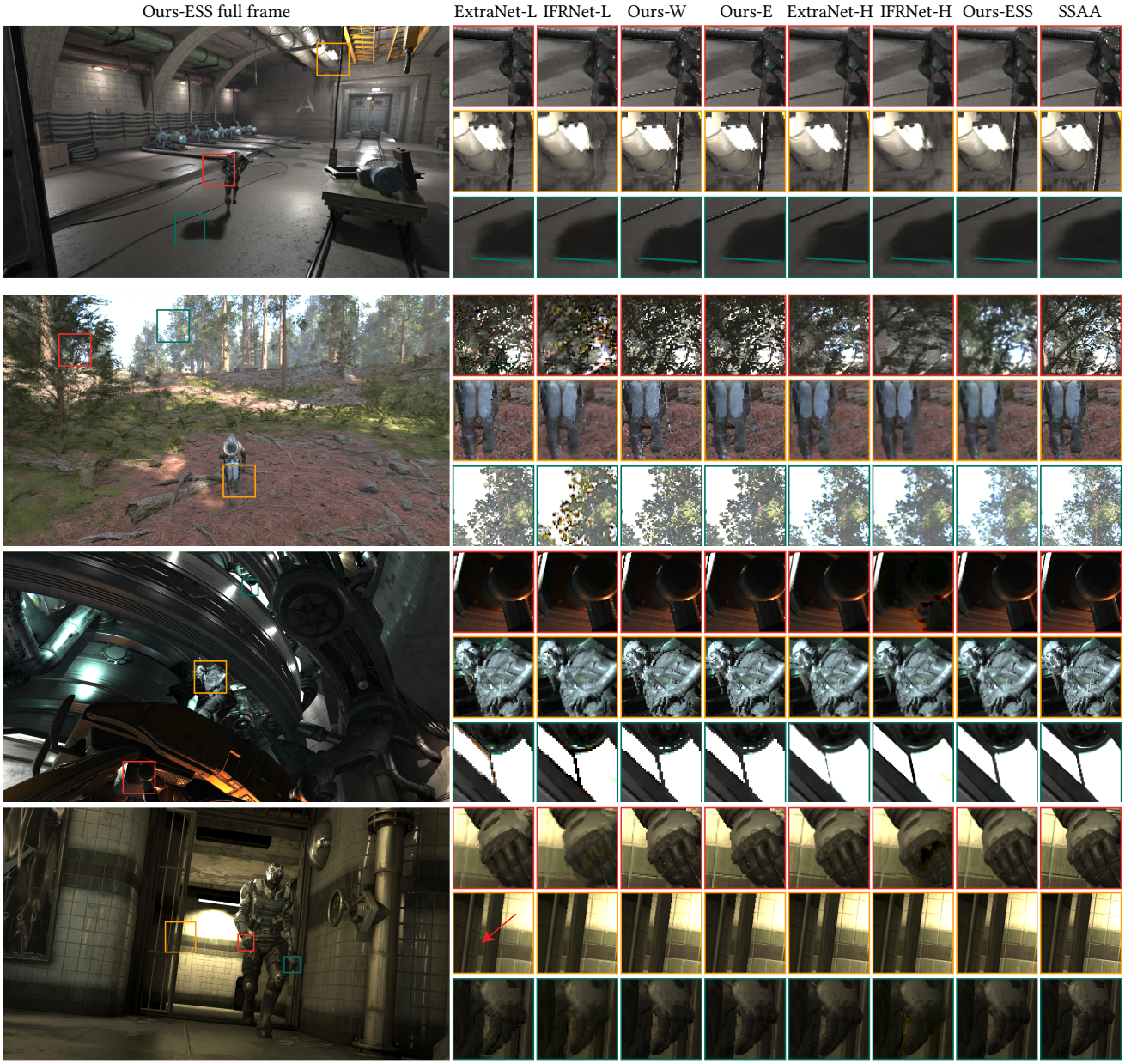

Figure 7: We also compare our framework and our intermediate outputs with temporal SS methods including ExtraNet and IFRNet. The comparison is in two different settings, as mentioned in Sec. 5.2, to evaluate our intermediate output and our joint full framework. ExtraNet-L and IFRNet-L refers to the low resolution input and output without any anti-aliasing. ExtraNet-H and IFRNet-H refers to the high resolution input and output with temporal anti-aliasing. Ours-W refers to our G-buffer guided warping only results, Ours-E refers to Ours-W with shading refinement, and Ours-ESS refers to our full framework with joint spatial SS and frame extrapolation. Our extrapolation methods show better quality than other frame generation methods and our joint framework shows comparable results with native resolution as the input for baselines.