Neural Supersampling for Real-time Rendering

LEI XIAO, SALAH NOURI, MATT CHAPMAN, ALEXANDER FIX, DOUGLAS LANMAN, and ANTON KAPLANYAN, Facebook Reality Labs

Review by Mark Baushenko, NSU

Introduction

Advantage

- Easy to integrate with modern game engines, requires no special hardware or software.
- Method takes common inputs from modern game engines: color, depth and motion vectors at a lower resolution.
- Method allows for compelling 4×4 upsampling from highly aliased input and produces high fidelity and temporally stable results in real-time.
- Created a new dataset with realistic camera movement for temporal stability (with large rotation and movement).

Summarizing the technical contribution

- This a temporal neural network tailored for image supersampling of rendered content that employs rich rendering attributes (i.e., color, depth, and motion vectors) and that is optimized for real-time applications.
- Demonstrating the first learned supersampling method that achieves significant 4×4 supersampling with high spatial and temporal fidelity.
- This method significantly outperforms prior work, including real-time temporal anti aliasing upscaling and state-of-the-art image and video super resolution methods, both in terms of visual fidelity and quantitative metrics of image quality.


COLOR
DEPTH
MOTION VECTORS

Related works

Spatial Anti Aliasing

- **M**ulti**S**ampling **A**nti **A**liasing (MSAA), where the color of a polygon covered by a pixels is only calculated once, avoiding computing multiple subpixels samples for the same polygon.
- Texture filtering, where high-frequency details coming from surface textures are prefiltered using image pyramids. A proper prefiltered region is then selected in runtime based on the pixel's footprint projected to the textured surface.
- **M**orpho**L**ogical **A**nti **A**liasing (MLAA) try to estimate the pixel coverage of the original geometry based on the color discontinuities found in the proximity of the pixels in the final image.

Spatial Anti Aliasing

- **F**ast Appro**x**imate **A**nti **A**liasing (FXAA) approaches the undersampling problem by attenuating subpixel features, which enhances the perceived temporal stability.
- **S**ubpixel **M**orphological **A**nti **A**liasing (SMAA) combines **MLAA** with **MSAA**.

Temporal Anti Aliasing and Reconstruction

- **T**emporal **A**nti **A**liasing (TAA) uses an edge detection filter as a proxy to suppress flicker by heavier temporal accumulation. Recently, TAA has been also employed to perform temporal upsampling (TAAU).
- **D**eep-**L**earned **S**uper**S**ampling (DLSS) is the closest to our method, and uses temporal history and neural networks to enhance edges and perform upscaling.
- Another recent trend in reducing the rendering cost is to apply reconstruction methods to sparsely ray-traced and foveated images. There is a recent body of work on applying machine learning methods to real-time lowsample-count reconstruction and foveated reconstruction. These methods train temporally stable U-Net architectures to achieve a stable reconstructed video out of very noisy and/or sparse input frames, which is related to our task of interpolation for upsampling.

Single Image Super Resolution

- Instead of learning the direct mapping between the high-resolution target image and the low-resolution input image, **V**ery **D**eep **S**uper **R**esolution (VDSR) learns the residual between the two.
- SRResNet applies residual network architecture to the superresolution problem, and **E**nhanced **D**eep **S**uper **R**esolution (EDSR) further improves the performance by utilizing more, modified residual blocks.
- **E**fficient **S**ub**P**ixel **C**onvolutional Neural **N**etwork (ESPCN) introduces a subpixel CNN that operates at low resolution and achieves real-time performance.
- And **Lap**lacian Pyramid **S**uper **R**esolution **N**etwork (LapSRN), **R**esidual **D**ense **N**etwork (RDN), **R**esidual **C**hannel **A**ttention **N**etworks (RCAN)

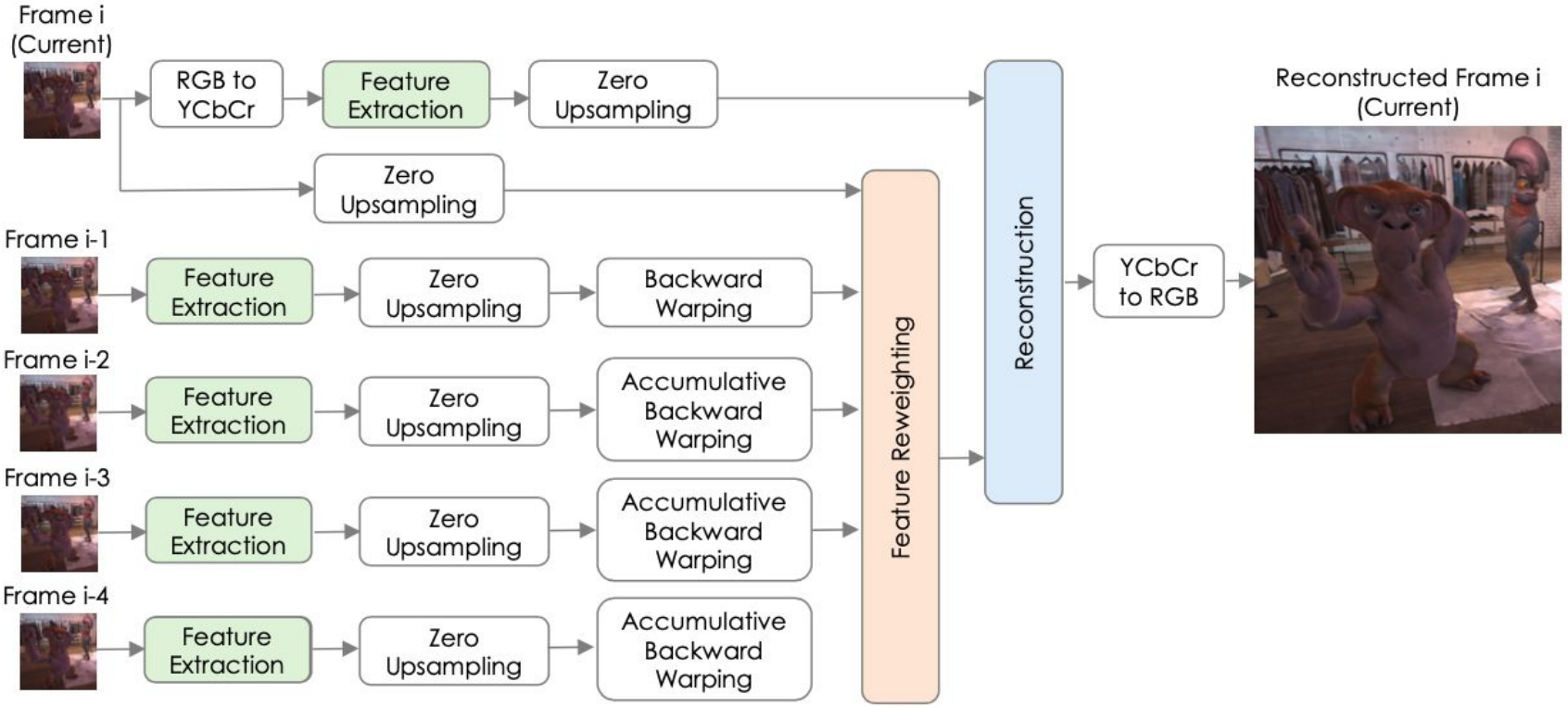
Video Super Resolution

- VESPCN introduces a multi-resolution spatial transformer module for joint motion compensation and video super resolution.
- SPMCVSR introduces a subpixel motion compensation layer to fuse multiple frames for revealing image details.
- EDVR applies a pyramid, cascading and deformable alignment module and a temporal and spatial attention module.
- FRVSR proposes a RNN that warps the previously estimated frame to facilitate the subsequent one.
- RBPN develops a recurrent encoder-decoder architecture for incorporating features extracted from single-image and multi-frame modules.
- TecoGAN

Method

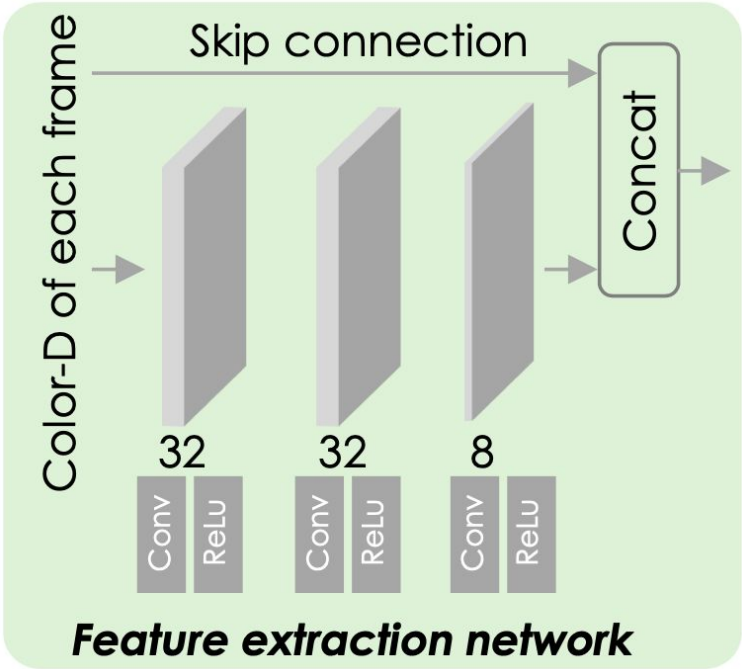
Method first warps previous frames to align with the current frame, in order to reduce the required receptive field and complexity of the reconstruction network. In contrast to existing work, however, to better exploit the specifics of rendered data, i.e., point-sampled colors and subpixel-precise motion vectors, method applies the frame warping at the target (high) resolution space rather than at the input (low) resolution. Specifically, the method projects the input pixels to the high resolution space, prior to the warping, by zero-upsampling. As the rendered motion vectors do not reflect disocclusion or shading changes between frames, the warped previous frames would contain invalid pixels mismatching with the current frame, which would mislead the post-reconstruction. To address this problem, we include a reweighting mechanism before the reconstruction network to de-select those invalid pixels. The reweighting mechanism is related to the confidence map approaches used for multi-frame blending in various applications. In contrast to these approaches, however, method utilizes a neural network to learn the reweighting weights. Lastly, the preprocessed previous frames (after zero-upsampling, warping and reweighting) are stacked together with the current frame (after zero-upsampling), and fed into a reconstruction network for generating the desired high-resolution image.

Network Architecture



Feature Extraction

The feature extraction module contains a 3-layer convolutional neural network. This subnetwork processes each input frame individually, and shares weights across all frames except for the current frame. For each frame, the subnetwork takes color and depth as input, and generates 8-channel learned features, which are then concatenated with the input color and depth, resulting in 12-channel features in total.

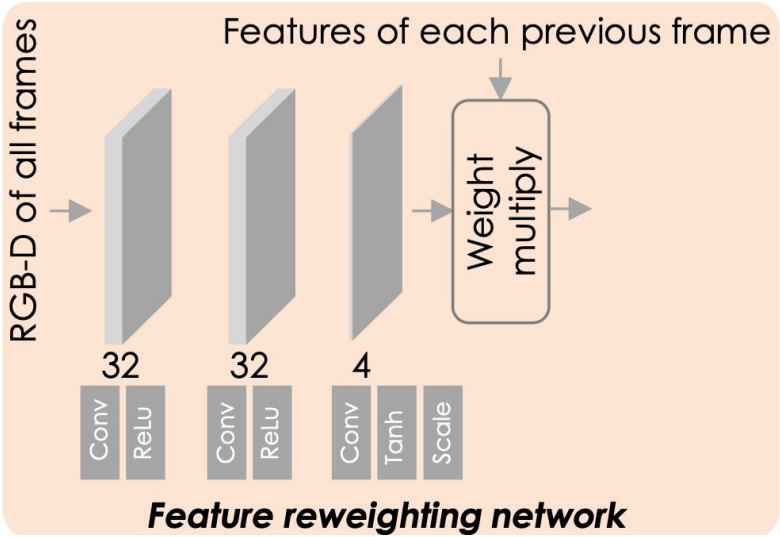


Temporal Reprojection

To reduce the required receptive field and thus complexity of the reconstruction network, we apply temporal reprojection to project pixel samples and learned features of each previous frame to the current, by using the rendered motion vectors. In order to fully exploit the subpixel backward motion vectors, we conduct the temporal reprojection at the target (high) resolution space. First, we project the pixel samples from input (low) resolution space to the high resolution space, by zero upsampling, i.e. assigning each input pixel to its corresponding pixel at high resolution and leaving all the missing pixels around it as zeros. Then, we resize the rendered low resolution map of motion vectors to high resolution simply by bilinear upsampling, taking advantage of the fact that the motion vectors are piece-wise smooth. Next, we apply backward warping of the zero-upsampled previous frames using the upsampled motion vectors, while bilinear interpolation is adopted during warping.

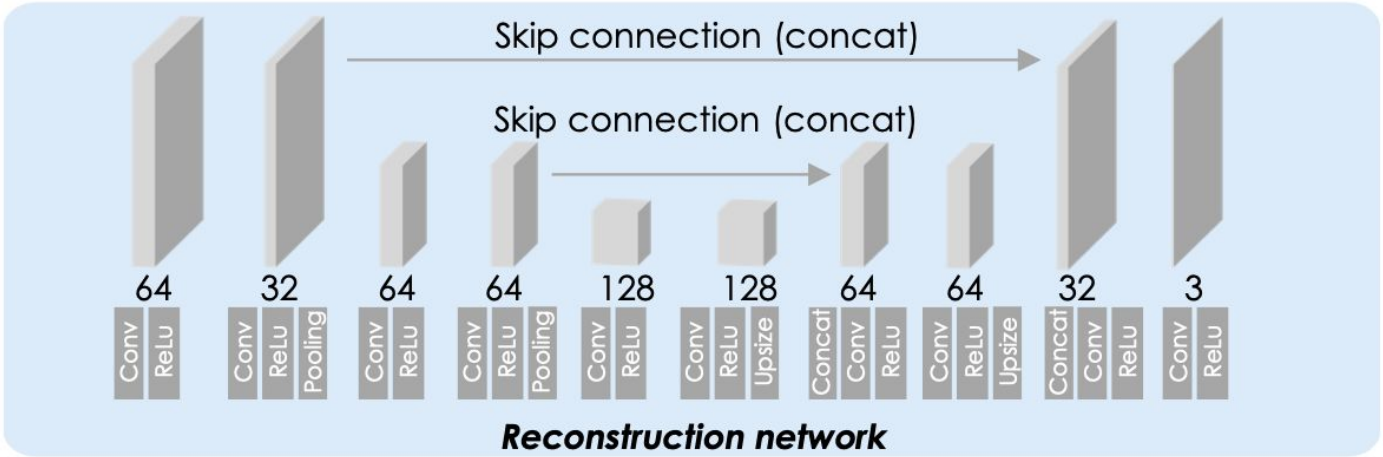
Feature Reweighting

The feature reweighting module is a 3-layer convolutional neural network, which takes the RGB-D of the zero-upsampled current frame as well as the zero-upsampled, warped previous frames as input, and generates a pixel-wise weighting map for each previous frame, with values between 0 and 10, where 10 is a hyperparameter. The hyperparameter is set to allow the learned map to not just attenuate, but also amplify the features per pixel, and empirically we found the dynamic range of 10 was enough. Then each weighting map is multiplied to all features of the corresponding previous frame.



Reconstruction

Finally, the features of the current frame and the reweighted features of previous frames are concatenated and fed into a reconstruction network, which outputs the recovered high resolution image of the current frame. We adopt a 3-scale, 10-layer U-Net with skip connections for the reconstruction subnetwork.



Losses

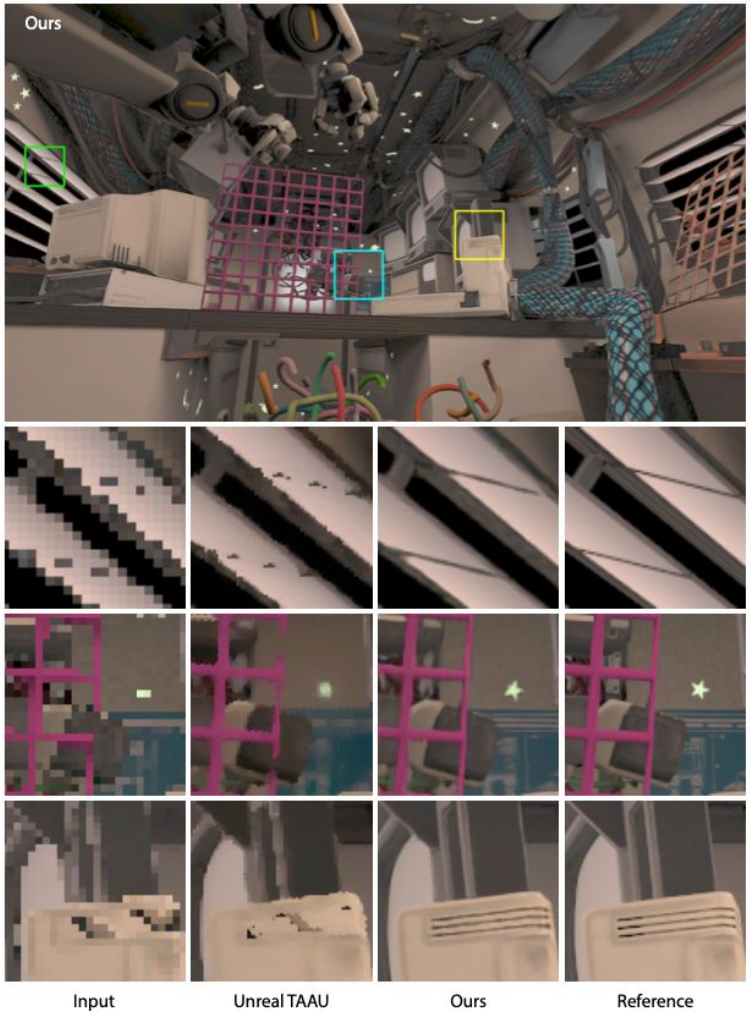
The training loss of our method, as given in formula, is a weighted combination of the perceptual loss computed from a pretrained VGG-16 network as introduced in Johnson et al., and the structural similarity index (SSIM).

$$loss(\mathbf{x}, \bar{\mathbf{x}}) = 1 - \mathrm{SSIM}(\mathbf{x}, \bar{\mathbf{x}}) + w \cdot \sum_{i=1}^{5} ||\mathrm{conv}_i(\mathbf{x}) - \mathrm{conv}_i(\bar{\mathbf{x}})||_2^2$$

where $\mathbf{x}$ and $\bar{\mathbf{x}}$ are the network output and reference high-resolution image respectively, and the relative weight is w = 0.1. We refer to Johnson et al. for the full details of selected VGG-16 layers.

Results

		ESPCN	VESPCN	DUF	EDSR	RCAN	Ours
PSNR (dB)	Robots	31.62	31.72	32.30	33.72	33.40	36.08
	Village	27.26	27.39	27.62	27.74	27.77	30.70
	DanceStudio	30.24	30.41	30.96	31.64	31.62	34.07
	Spaceship	32.73	32.80	33.65	34.29	34.39	36.69
SSIM	Robots	0.9134	0.9142	0.9335	0.9476	0.9440	0.9696
	Village	0.7908	0.7950	0.8270	0.8296	0.8294	0.9019
	DanceStudio	0.8375	0.8418	0.8640	0.8794	0.8777	0.9224
	Spaceship	0.9119	0.9123	0.9286	0.9427	0.9418	0.9712
STRRED	Robots	109.7	103.5	73.2	56.5	63.6	19.3
	Village	192.4	186.6	131.8	169.8	168.6	42.5
	DanceStudio	213.0	194.8	118.8	117.8	121.6	40.6
	Spaceship	98.8	96.6	66.6	58.1	58.4	22.1



Ours
PSNR = 31.74dB, SSIM = 0.9430
TAAU
PSNR = 30.06dB, SSIM = 0.9070

Thanks for attention!