Upscaling with Deep Learning
Definition

- Upscaling or upsampling is a procedure of increasing the image resolution while saving as many details as possible from the original image.
Methods

- Without machine learning (ML) or deep learning (DL): nearest neighbours, linear, bilinear, bicubic, AMD FidelityFX Super Resolution (FSR) 1.0, AMD FSR 2.0

- With deep learning: Nvidia Deep Learning Super Sampling (DLSS), convolutional neural networks (CNNs), generative adversarial NNs (GANs), transformers
Why?

- Usually rendering a frame in lower resolution and upscaling it to a higher one is faster than rendering a frame directly in high resolution.

- A trade-off between the quality and performance.
Non ML Upscaling

smol

Original  Nearest Neighbour  Bilinear  Bicubic  Edgedirected
AMD FidelityFX Super Resolution (FSR) 1.0

Two main passes

• An upscaling pass called **EASU** (Edge-Adaptive Spatial Upsampling) that also performs edge reconstruction. In this pass the input frame is analyzed and the main part of the algorithm detects gradient reversals – essentially looking at how neighboring gradients differ – from a set of input pixels. The intensity of the gradient reversals defines the weights to apply to the reconstructed pixels at display resolution.

• A sharpening pass called **RCAS** (Robust Contrast-Adaptive Sharpening) that extracts pixel detail in the upscaled image.

Source: https://gpuopen.com/fidelityfx-superresolution/#howitworks
AMD FidelityFX Super Resolution (FSR) 1.0
Where to integrate
AMD FidelityFX Super Resolution (FSR) 1.0

Examples

- https://gpuopen.com/fidelityfx-superresolution/#comparison
## AMD FidelityFX Super Resolution (FSR) 1.0

### Performance

<table>
<thead>
<tr>
<th>FSR target resolution</th>
<th>ENTHUSIAST GPUs</th>
<th>PERFORMANCE GPUs</th>
<th>MAINSTREAM GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4K</strong></td>
<td><strong>RADEON™ RX 6800XT, NVIDIA RTX 3080</strong></td>
<td><strong>RADEON™ RX 6700XT, NVIDIA RTX 3060 Ti</strong></td>
<td><strong>RADEON™ RX 5700XT, NVIDIA RTX 2060 SUPER</strong></td>
</tr>
<tr>
<td>Ultra Quality Quality Balanced Performance</td>
<td>0.40 ms or less</td>
<td>0.60 ms or less</td>
<td>1.0 ms or less</td>
</tr>
<tr>
<td><strong>1440p</strong></td>
<td><strong>RADEON™ RX 6800XT, NVIDIA RTX 3080</strong></td>
<td><strong>RADEON™ RX 6700XT, NVIDIA RTX 3060 Ti</strong></td>
<td><strong>RADEON™ RX 5700XT, NVIDIA RTX 2060 SUPER</strong></td>
</tr>
<tr>
<td>Ultra Quality Quality Balanced Performance</td>
<td>0.20 ms or less</td>
<td>0.30 ms or less</td>
<td>0.50 ms or less</td>
</tr>
</tbody>
</table>
AMD FidelityFX Super Resolution (FSR) 2.0
AMD FidelityFX Super Resolution (FSR) 2.0
Where to integrate
AMD FidelityFX Super Resolution (FSR) 2.0

Examples

- https://gpuopen.com/fidelityfx-superresolution-2/#comparison
AMD FidelityFX Super Resolution (FSR) 2.0

Performance

<table>
<thead>
<tr>
<th>FSR 2.0 Target resolution</th>
<th>Enthusiast GPUs (AMD RADEON RX 6800XT)</th>
<th>Performance GPUs (AMD RADEON RX 6700XT)</th>
<th>Mainstream GPUs (AMD RADEON RX 5700XT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K</td>
<td>&lt; 1.1ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1440p</td>
<td></td>
<td>&lt; 0.8ms</td>
<td></td>
</tr>
<tr>
<td>1080p</td>
<td></td>
<td></td>
<td>&lt; 0.6ms</td>
</tr>
</tbody>
</table>
Deep Learning Upscaling

- Nvidia Deep Learning Super Sampling (DLSS)
- Efficient Sub-Pixel Convolutional Neural Network (ESPCN)
- Super-Resolution Generative Adversarial Network (SRGAN)
- and many more…
Nvidia Deep Learning Super Sampling (DLSS)

Training

HOW DLSS IS TRAINED

Ray Tracing → 1080P Rendered Image → Motion Vectors → Temporal Feedback → 4K Super Resolution Output → ΔW

Supercomputer Rendered 16K Ground Truth
Nvidia Deep Learning Super Sampling (DLSS) Architecture

NVIDIA DLSS — HOW IT WORKS

TEMPORAL SUPERSAMPLING
Each pixel is super-sampled across multiple frames

TEMPORAL FEEDBACK
Motion vectors & the prior 4K frame enable the AI to track pixels & objects for improved stability & better reconstruction

CONVOLUTIONAL AUTOENCODER
Generalized AI network trained to handle any game content makes optimal decisions on final pixel colors

NATIVE-QUALITY SUPER RESOLUTION OUTPUT
DLSS offers image quality comparable to native resolution while rendering only one quarter to one half of the pixels
Nvidia Deep Learning Super Sampling (DLSS)
Where to integrate
Nvidia Deep Learning Super Sampling (DLSS)

Examples

DLSS vs AMD FSR

- DLSS uses deep learning while AMD FSR relies on handcrafted algorithms
- DLSS produces better picture
- DLSS requires Nvidia RTX graphics card while AMD FSR can run on most of modern graphics cards (both Nvidia and AMD)
- AMD FSR is fully open-source and easy to integrate, DLSS is not open-source
- More games support AMD FSR
Tensor Cores

- A special GPU core that can multiply two 4x4 matrices and add the result to another 4x4 matrix in one operation

- A newer version of Pascal Core that could only multiply two scalars, i.e. 1x1 matrices

- Tensor Core is 8x faster for FP16 multiplication, 16x faster for INT8, and 32x faster for INT4

CUDA

- API that allows to use the GPU for general purpose processing
- Very roughly, Tensor Cores are buffed shader cores that we can use independently from the rendering pipeline
- Since neural networks are just a bunch of matrix operations, CUDA is indispensable for deep learning
Convolutional Neural Networks (CNNs)

Source layer

Convolutional kernel

Destination layer

\[
\begin{align*}
(-1 \times 5) + (0 \times 2) + (1 \times 6) + \\
(2 \times 4) + (1 \times 3) + (2 \times 4) + \\
(1 \times 3) + (-2 \times 9) + (0 \times 2) &= 5
\end{align*}
\]
Convolutional Neural Networks (CNNs)

- CNNs can be parallelized very well, so they can use the full potential of GPUs
- Widely applied for image processing
- Easy to understand
CNNs for Upscaling

• From the definition, convolutions usually make the input smaller. How do we use them for upscaling?

• Add more filters and then use pixel shuffle!
Efficient Sub-Pixel Convolutional Neural Network (ESPCN)
Efficient Sub-Pixel Convolutional Neural Network (ESPCN)

Examples

(a) Baboon Original
(b) Bicubic / 23.21db
(c) SRCNN [7] / 23.67db
(d) TNRD [3] / 23.62db
(e) ESPCN / 23.72db

(f) Comic Original
(g) Bicubic / 23.12db
(h) SRCNN [7] / 24.56db
(i) TNRD [3] / 24.68db
(j) ESPCN / 24.82db
CNN in a Shader

• If Tensor Cores are just fancy Shader Cores can we actually run a neural network in a shader?

• Turns out we can!

• https://medium.com/analytics-vidhya/inferring-a-super-resolution-neural-network-on-raspberry-pi-gpu-89b5456d21ef
CNN in a Shader

• Inference is just a bunch of matrix multiplications

• We can hard code the network weights values into a fragment shader and transform the input with it

• This way, we can run it on any OpenGL compatible GPU
CNN in a Shader (Even with Training!)

SimpNet RenderTexture Layout
CNN in a Shader

Challenges

- Operations limited to 0-1 range
- Need to account for an upscaled image on later steps (decals, HUD, LOD)
- CUDA cores are more optimised and give more flow control
Conclusion

• Non-ML and non-DL approaches are easier to integrate and they can still produce quite good results

• Deep Learning allows to train the models to produce great quality upscaling but is more challenging to integrate and requires at least RTX-family GPU

• Non-ML approaches are easily controllable and predictable, DL models can sometimes produce unexpected artefacts

• We can run and even train (with limits) neural networks directly on the shaders
References

- https://www.nvidia.com/en-us/on-demand/session/gdc21-gdc2103/?playlistId=playList-f95a44a8-59c2-4d0f-8bca-9cf153bae6e9
- https://gpuopen.com/fidelityfx-superresolution/#howitworks
- https://gpuopen.com/fidelityfx-superresolution-2/#howitworks
References

