DLSS & XeSS

Quick Overview

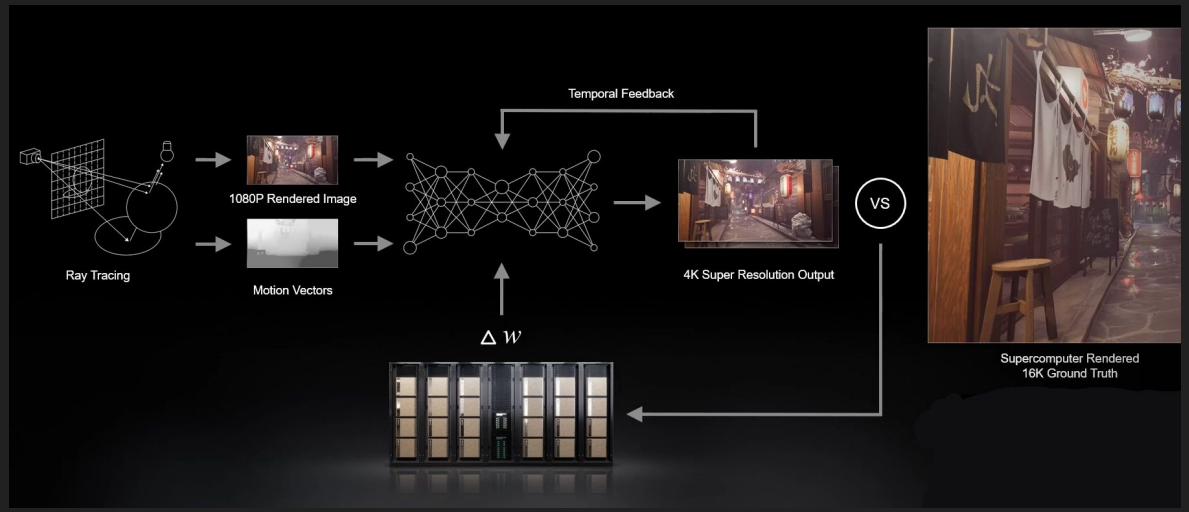
DLSS 2.x - Spatial SS

DLSS first version in 2018 needed anti-aliased LR image and was trained on a per game basis

In 2020 was replaced with DLSS 2.x and uses motion vectors (calculated by previous frames)

DLSS 2.x uses different network presets for scaling options

Works on 2000+ series



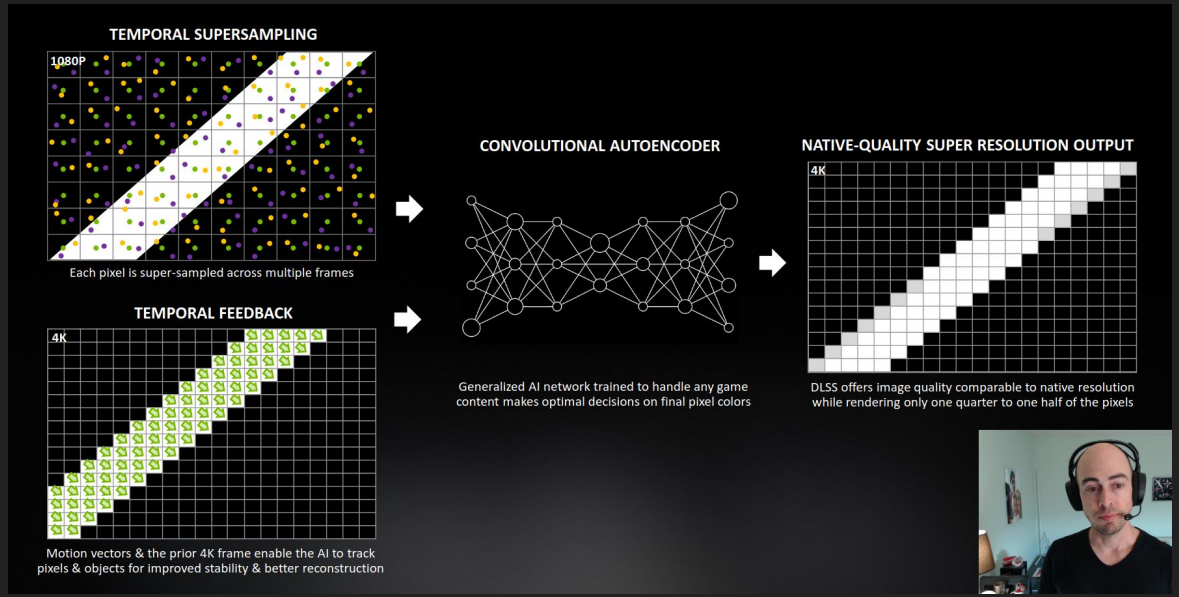
DLSS 2.x - Spatial SS

Uses a Convolutional AutoEncoder network

Uses 2 Losses:

- Content Loss for geometry
- Style Loss for textures

Unfortunately no details about network is known!



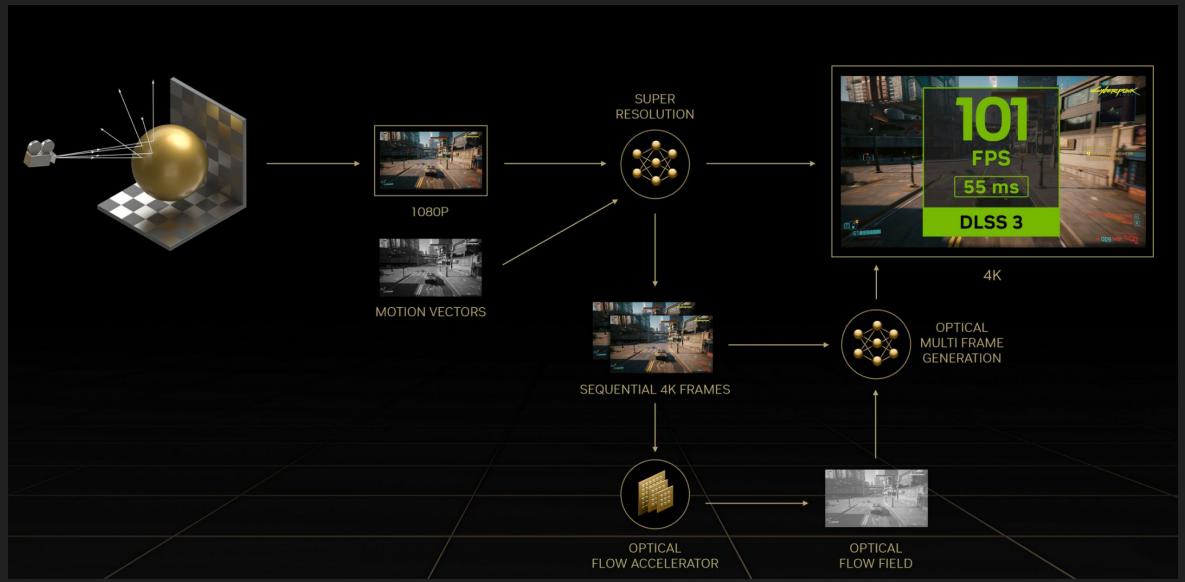
DLSS 3 - Temporal SS - Frame Interpolation

In 2022 Nvidia added Frame Interpolation in form of DLSS 3

Uses an optical flow field (calced on dedicated HW) to track motion of more than game objects (motion vectors!) like particles or shadows

most likely 2 different networks (1 for spatial and 1 for temporal)

Interpolates frames btw. 2 rendered frames, introduces latency!
-> works on 4000+ series



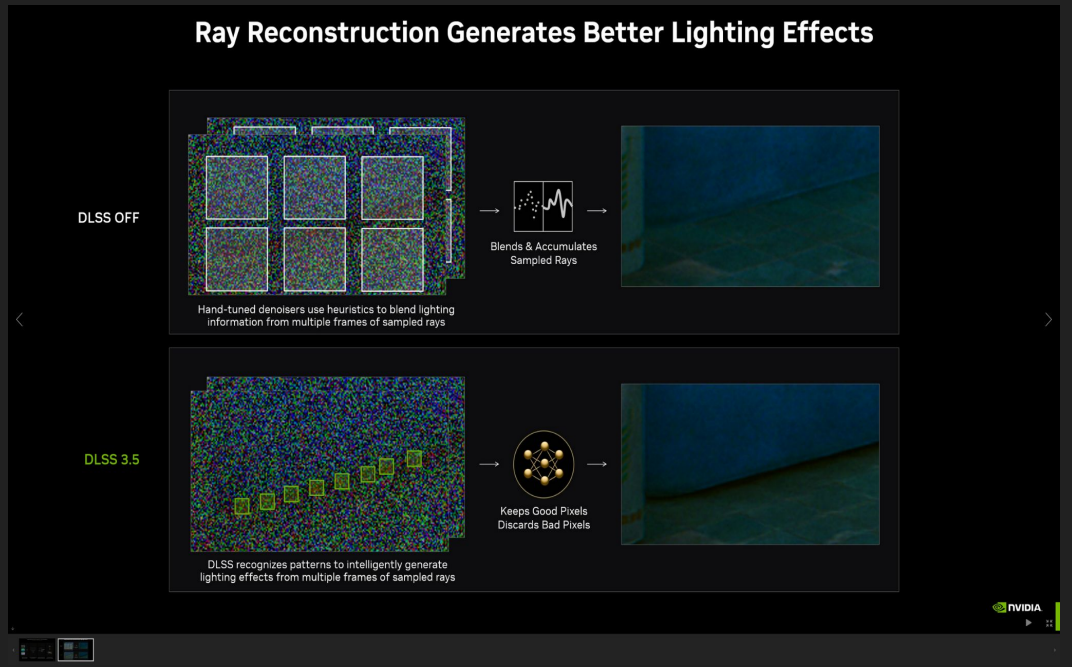
DLSS 3.5 - Ray Reconstruction

In 2023 DLSS updated to 3.5

Focuses on “ray reconstruction” replaces hand-tuned denoisers to generate more accurate ray tracing fidelity like better reflections, sharper shadows etc.

Trained with 5x amount of image data then DLSS 3

Spatial SS usable with 3000+ series
Temporal SS usable with 4000+ series



XeSS - Spatial SS

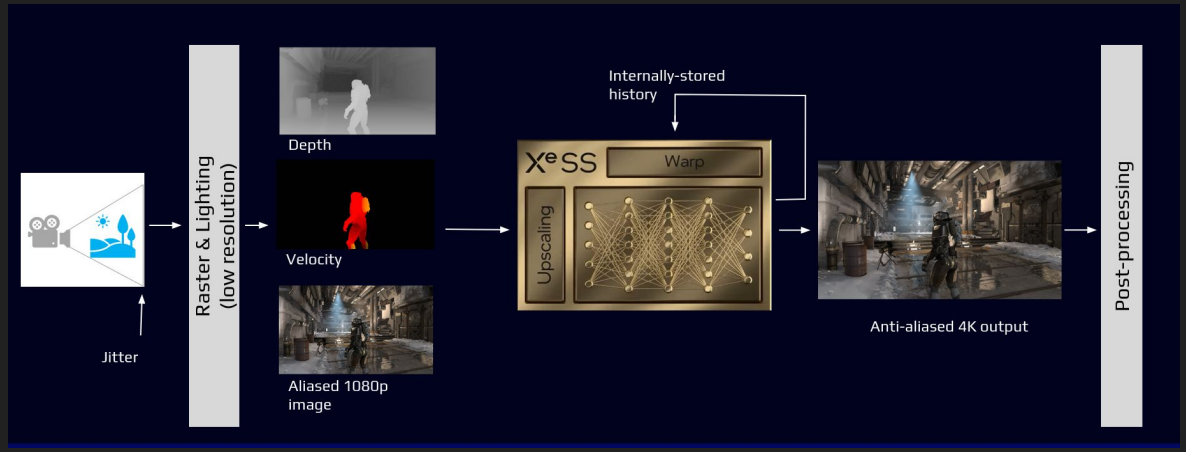
In 2022 Intel developed XeSS

Input:
- Jitter offset (Halton)
- LR image
- motion vectors
- depth buffer

No information about network model or details available!

Works on all GPUs

Best on Intel GPU because of Xe cores, but can run with dp4a shader on different gpus



Conclusion

DLSS network architecture:
Convolutional AutoEncoder

XeSS network architecture:
Unknown

No information about network details (optimizer, layers, training data) available!

Nvidia and Intel recommend using upscaling from 1080p (not lower..)
Common scaling factors: 1.3, 1.5, 1.7, 2.0, 3.0

Extra SS - Frame Extrapolation

Frame Interpolation adds latency and overhead for game engines as 2 frames need to be rendered for generated frame and special scheduling needs to be implemented

Frame Extrapolation tries to solve this by “guessing” the generated frame based on previous real frames

Intel sponsored paper worked on this, for accurate lighting they calculated only geometry buffer for generated frame