Beyond Polygons, Voxels & Rasterization

Nejc Lešek  CTO, Lightmass Dynamics
Minimum Requirements

- Variable rate rendering
- Always grouping similar work items
- No rasterization
- Real-time rates (50 ms or less per frame)
DNR (Deconstruction Rendering)

• Groups similar work items

• Enables efficient implementation of:
  • Variable rate rendering
  • Foveated rendering
  • Checkerboard rendering
  • Any analytic or random pattern
Smart Geometry (SMG)

• Attaching neural networks to geometric primitives

• Main concepts:
  • Simple, small and shallow networks
  • Millions of NN working together
  • Real-time training and inference

TIP: as NN input, find scene properties that can be mostly represented with a continuous function.
Performance impact

BVH vs. SMG

BVH – Linear
SMG – Sublinear
SMG: DENoise

- Spatial denoise
- NN approximate energy at surface
GOOD FOR:

• Static scenes

• Can compliment lightmaps; by vectorizing soft shadow regions.

BAD FOR:

• Dynamic scenes

• Very small primitives
A viable high performance substitute for:

- Bidirectional PT
- Metropolis light transport

Finds up to 70% more paths than unidirectional path tracing.
PGNR (Path Guided Neural Rendering)
PGNR (Path Guided Neural Rendering)

- DNN autoencoder denoise, but...
- Very *sparse* secondary ray sampling
- **Full** resolution primary rays
- Variable DNN depth
- Scene data & NN organized into voxels
PGNR: Voxelized autoencoders

- Offline voxel data interpolation
- A bit of overfitting is welcome
- Each voxel can be processed by a different GPU, training scales linearly!
Autoencoder inputs

- Normals 3D
- Roughness 1D
- Adaptive temporal reprojection
- Albedo 3D
- Depth 1D
- Direct Light 1D
DSRVT (Deep Super Resolution Virtual Texturing)
DSRVT (Deep Super Resolution Virtual Texturing)

- Provides Virtual Texturing benefits:
  - Memory management (use only what you need / can see)
  - Effectively unlimited texture resolution

- Super Resolution:
  - Adds extra details when we run out of higher res. textures

  OR

- Reduces shipped texture size by x4
QNM (Quantum Neural Models)

- Defines volumetric object with properties
- Uses neural primitives
- Uses Tensor Pipeline
- Potential to unify physics, animation, geometry and materials
Animation guided by NN inputs

QNM model size: \(\sim5\) KB
QNM primitives: 9

Polygonal model size: \(\sim1\) MB
(vertices, normals, texture coordinates)

Polygonal primitives (triangles): 31415
BVH
acceleration structure

QNBVH
acceleration structure

20-40% reduction in ray-box intersection tests
Questions
Thank you!