Towards Neural Path Tracing in SRAM

Overview

- Neural rendering techniques are on the rise.
- We implemented a simple on-chip rendering application for Graphcore intelligence processing units (IPUs).
- A path tracer queries HDR environment lighting via a small neural network.
- The implementation can act as a proxy for other neural rendering technology (e.g. NeRFs, neural radiance caches, neural materials).

IPU Description

- IPU: massively parallel processor with 1472 homogeneous MIMD cores, more general purpose than other dedicated AI processors.
- Each core (tile) has the following features:
  - Private SRAM: 624KiB per tile, 897MiB aggregate per chip.
  - 6 hardware worker threads (8832 independent instruction streams per chip).
  - Dual issue of (4 wide) half or (2 wide) float instructions with memory/integer operations.
  - An accumulating matrix product (AMP) unit, 349 TFLOP/sec per chip (half precision).
  - A hardware random number generator (RNG).
  - All to all data exchange with other tiles.

IPU Path Tracing

- Simple path tracing with no light sampling.
  - Almost entirely plain C++ kernels.
  - The BVH is replicated on every tile, compressed nodes use float16 extents.
  - All Monte Carlo sampling uses IPU's hardware RNG.
  - Results shown here use a maximum path length 10 and roulette termination starting at depth 3.

Neural HDRI

- Escaped rays receive HDR environment light contribution from a neural network.
- HDRIs are approximated by training NeRF like co-ordinate networks (using Keras on IPU).
  - Network is trained to regress (equirectangular) pixel co-ords to colour: $(r, g, b) = f(u, v)$

Figure: Images path traced using neural-HDRIs trained in different colour-spaces (Left to right: RGB, YCoCg, YUV).

Neural Image Field Network

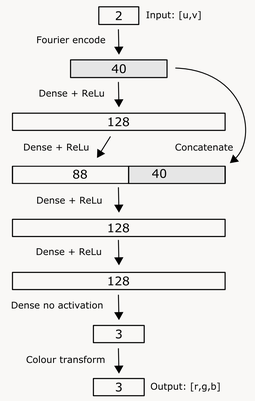

Figure: Activation sizes and operations between them.

Example Scenes

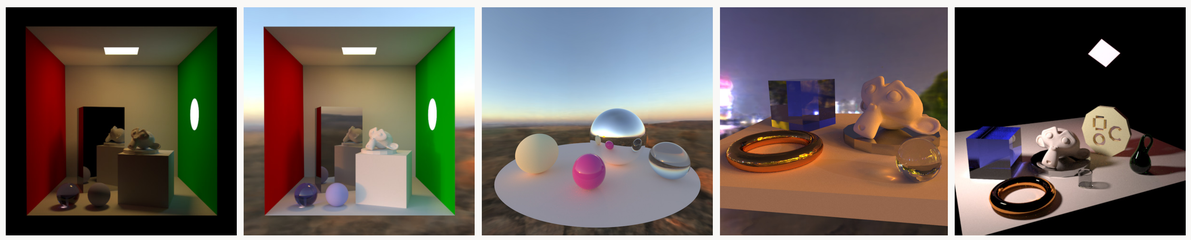

Figure: Images path traced on a Graphcore Bow-Pod. The HDR environment lighting is compressed into 97 KiB of neural network weights. The weights, activations, and scene BVH reside entirely in on-chip SRAM. (Rays are streamed from/to external DRAM).

Results

Figure: Comparison of neural path tracing results with different network sizes (left to right: H64/L2, H256/L4, H1024/L8, Blender 4K reference).

Model Config	PSNR RGB	PSNR Luminance	PSNR Chrominance	Sample-Rate (Paths/sec)
H64/L2	29.0	31.1	**14.5**	311.8M
H256/L4	29.3	35.6	14.1	149.1M
H1024/L8	**29.4**	**37.3**	14.0	9.3M

Table: PSNRs and sample rate for Urban Alley HDR-NIF approximations.

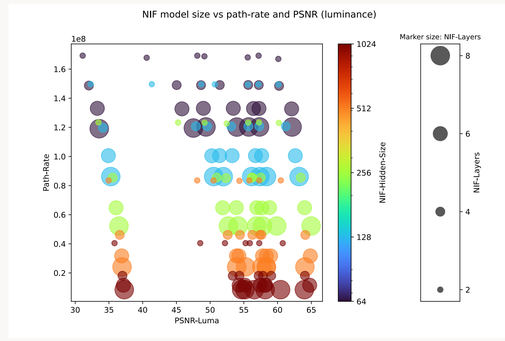

Figure: Influence of model size on sample rate and luminance PSNR.

Scene	MSE (vs Embree)		MSE (vs CPU)	
	Normal	Hit	Normal	Hit
Box	1.1×10^{-13}	7.6×10^{-9}	2.6×10^{-16}	0
Spheres	2.1×10^{-14}	2.5×10^{-14}	1.6×10^{-16}	0
Small BVH	4.5×10^{-7}	2.3×10^{-14}	1.8×10^{-16}	0
Large BVH	1.2×10^{-7}	7.1×10^{-14}	2.2×10^{-16}	0

Table: Precision of IPU ray tracing calculations.

Conclusions

- IPUs are suitable for elementary ray/path tracing calculations in combination with a neural rendering component.
- Potential future enhancements:
  - Movement of treelets or rays between tiles to better utilise on-chip communication bandwidth.
  - Use float8 instead of float16 for faster neural HDRI inference.
  - Compress BVH further using float8.
  - Neural material/geometry.

Mark Pupilli
markp@graphcore.ai
Graphcore

Paper on Arxiv

Code on GitHub