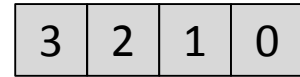# Improving BVH Ray Tracing Speed Using the AVX Instruction Set
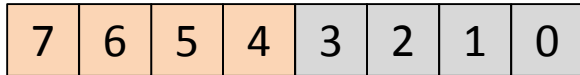
Attila T. Áfra

Budapest University of Technology and Economics, Hungary
Babeș-Bolyai University, Cluj-Napoca, Romania

- **SSE**: very popular SIMD instruction set, 128-bit, 4 floats *(1999)*

| 3 | 2 | 1 | 0 | SSE |

- **AVX**: introduced with **Intel Sandy Bridge**, 256-bit, 8 floats *(2011)*
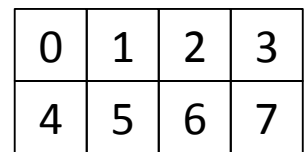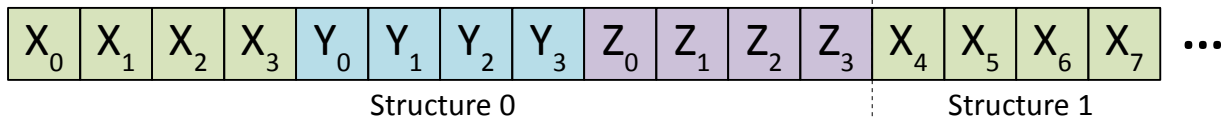
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AVX |

- BVH ray packet traversal algorithms: **ranged**, **partition**, etc.
- Smallest ray primitive: **SIMD ray**
- SSE: 2×2 SIMD rays
- **AVX: 4×2 SIMD rays**
- Frustum culling: interval arithmetic (no SIMD) and corner rays (4-wide SIMD)
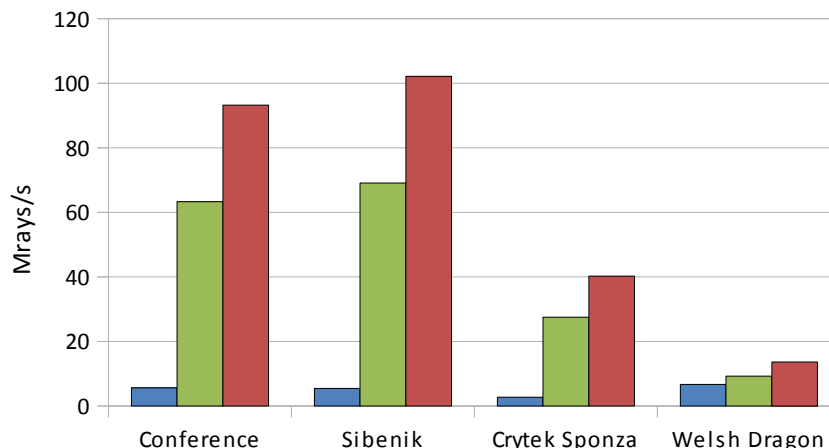
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |

AVX 4×2 SIMD ray

- Rays are stored in **AoSoA** (array of structures of arrays) layout
- AoSoA combines the SIMD-friendliness of SoA with the locality of AoS
- *Example (3D vector, 4-wide SIMD):*

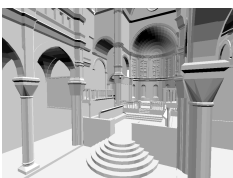| X_0 | X_1 | X_2 | X_3 | Y_0 | Y_1 | Y_2 | Y_3 | Z_0 | Z_1 | Z_2 | Z_3 | X_4 | X_5 | X_6 | X_7 | ... |

Structure 0   Structure 1

- **Performance: AVX ~50% faster than SSE4.1**
- Ranged traversal with frustum culling
- The sublinear speedup is due to larger SIMD rays with lower utilization and non-SIMD parts of the algorithm
- **Intel Core i5-2400** (4 cores, 4 threads, 3.1 GHz), 64-bit, Visual C++ 2010

Conference (283K tris)

Sibenik (80K tris)

Crytek Sponza (279K tris)

Welsh Dragon (2.2M tris)

Legend: Single, Ranged SSE, Ranged AVX

Mrays/s (y-axis: 0, 20, 40, 60, 80, 100, 120)

x-axis categories: Conference, Sibenik, Crytek Sponza, Welsh Dragon