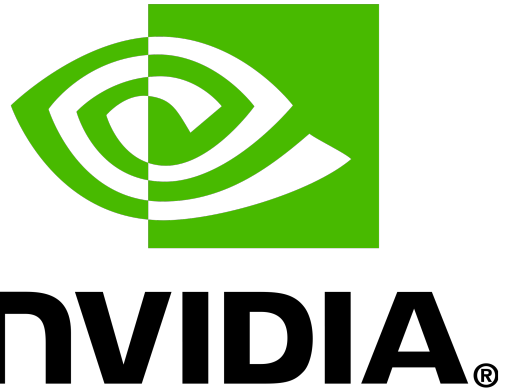
ON RAY REORDERING TECHNIQUES FOR FASTER GPU RAY TRACING

DANIEL MEISTER, JAKUB BOKŠANSKÝ, MICHAEL GUTHE, JIŘÍ BITTNER

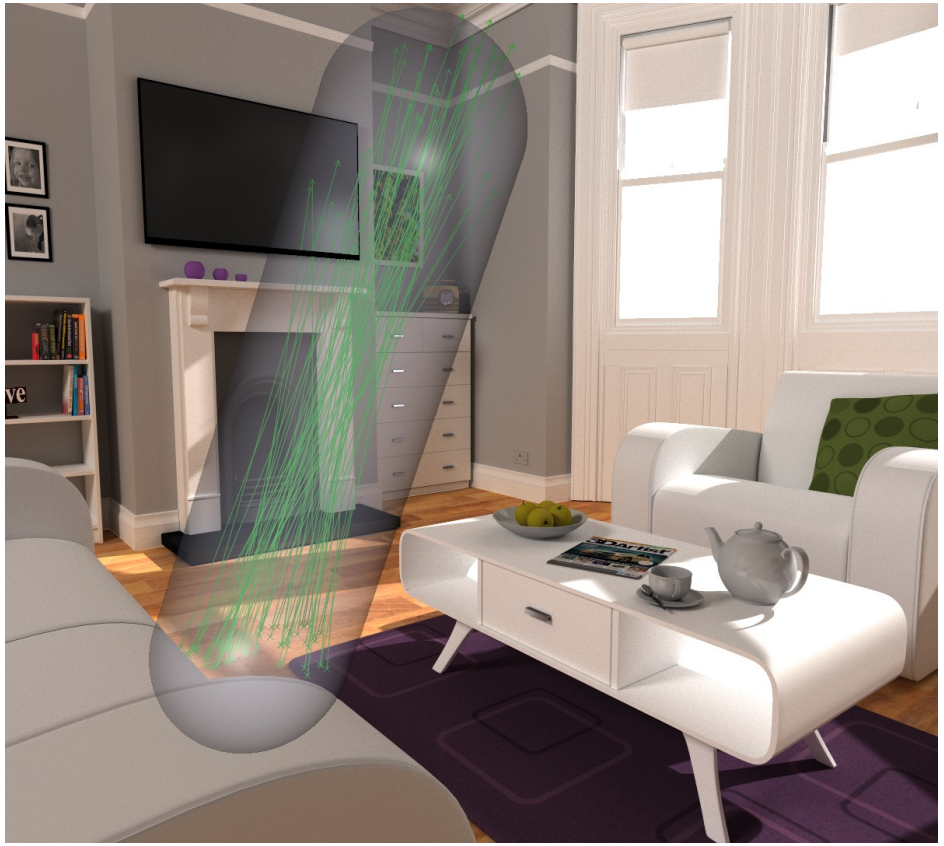


RAY TRACING – INCOHERENT RAYS

Not sorted

2355 Mrays/s

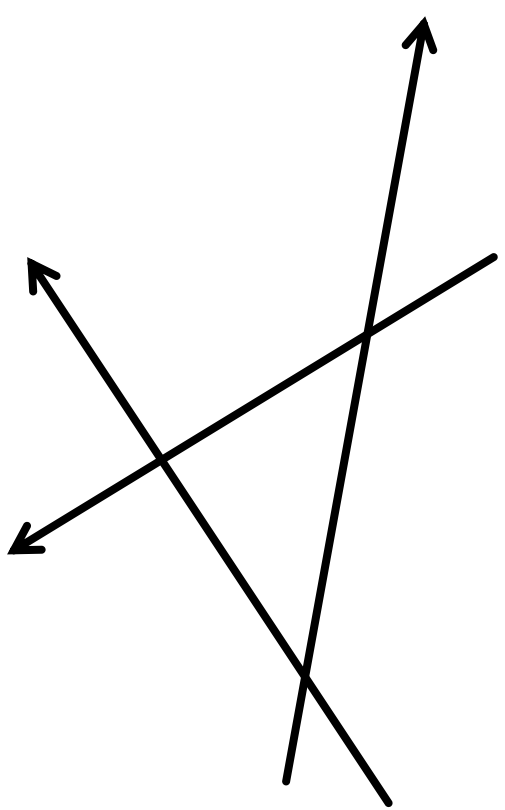
Sorted (our method)

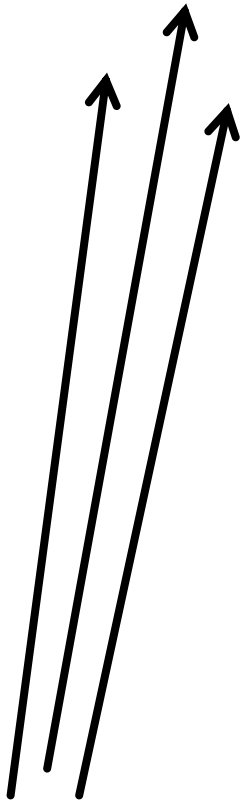


3914 Mrays/s
(1.7x speedup)

RAY REORDERING

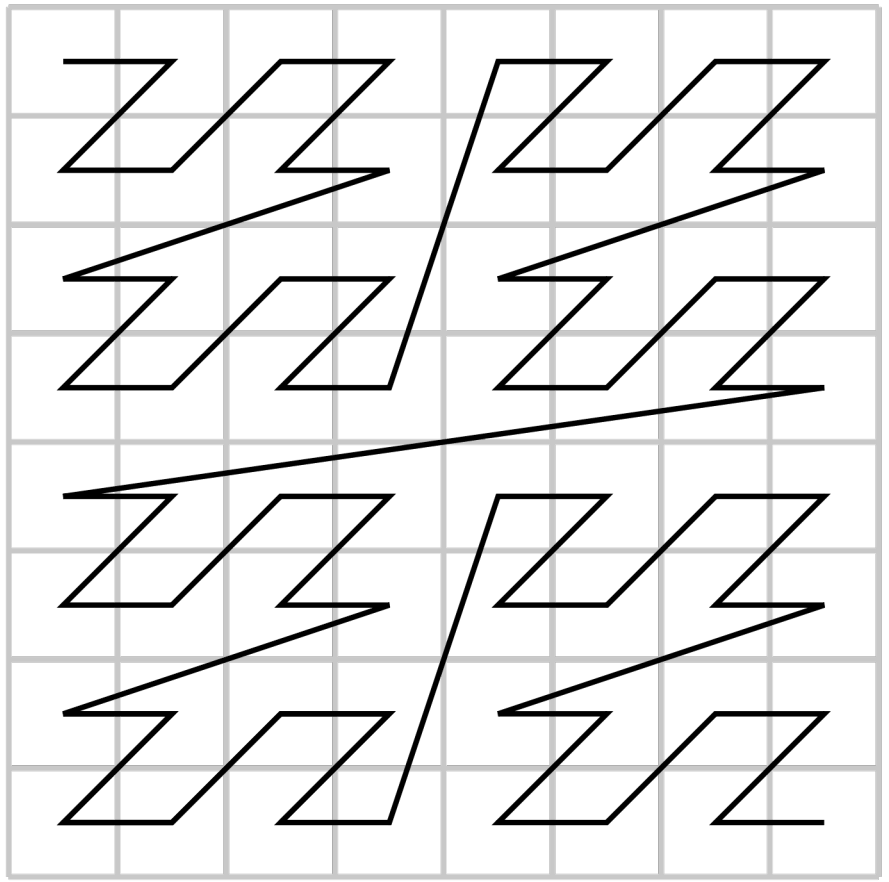
- Increasing ray coherence by grouping similar rays
  - Control flow
  - Cache hit ratio
  - Warp divergence





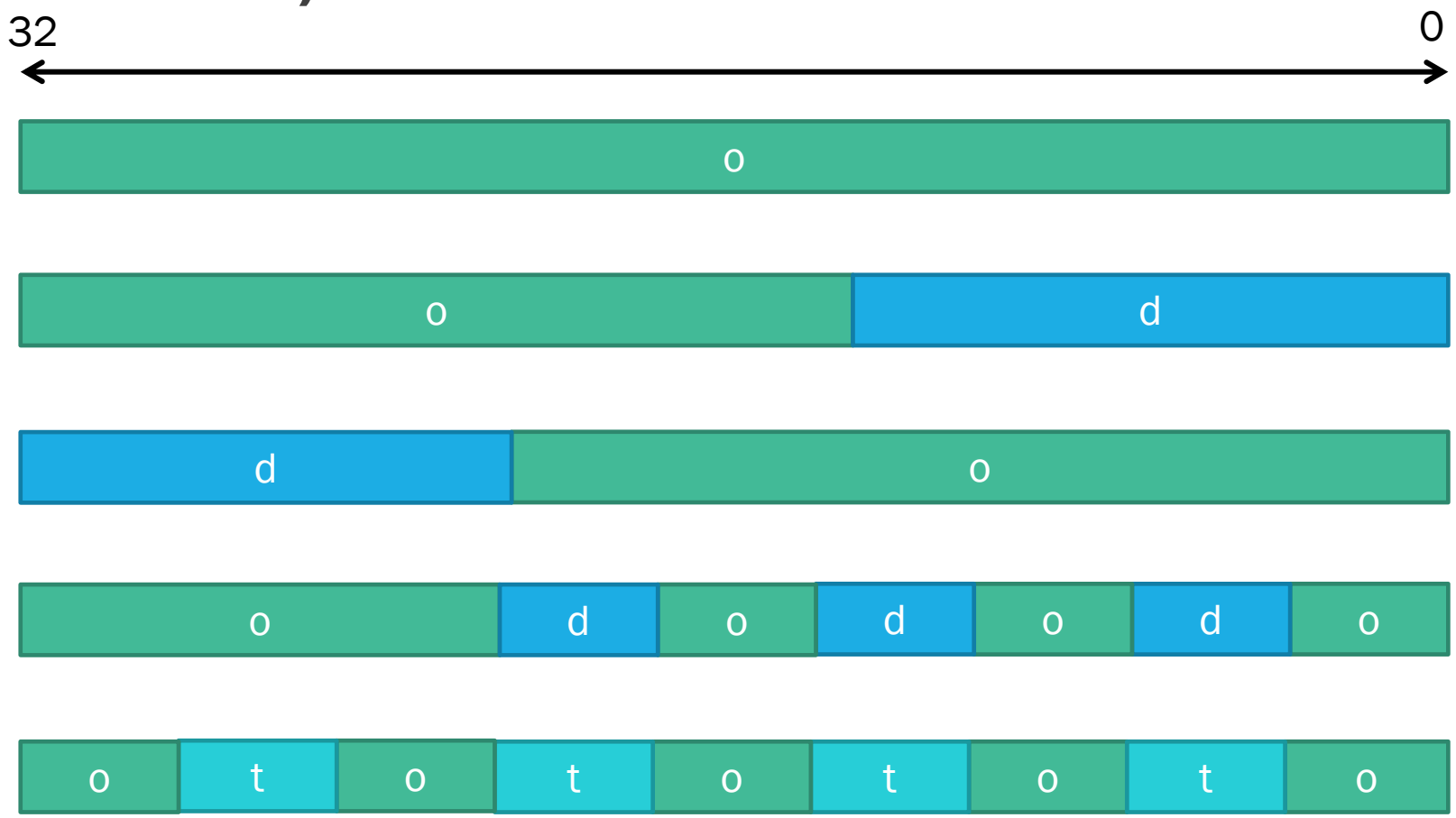
MORTON CURVE

- A space-filling curve subdividing space into a regular grid
- Order along the curve given by Morton codes
  - Interleaving successive bits of cell coordinates
- Sort rays along Morton curve
  - How to compute Morton codes (origins and directions)?
  - Agnostic to the trace kernel (OptiX, DirectX)



Morton curve in 2D

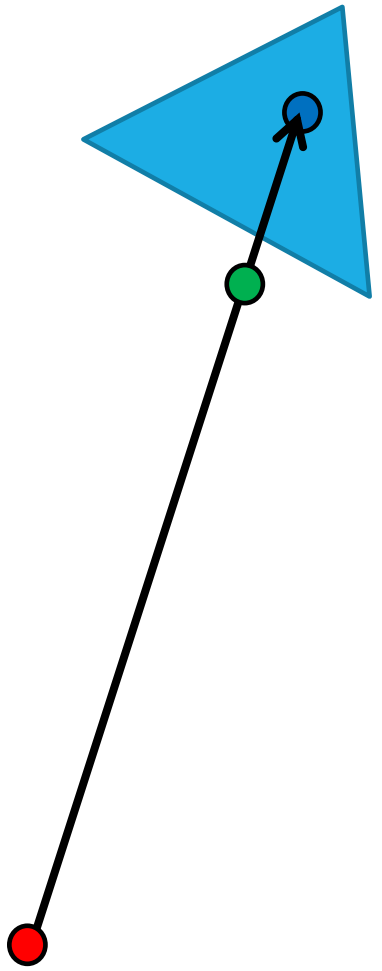
MORTON CODES (32 OR 64 BITS)



- Sufficient for shadow rays
- How many bits use for origin?
- How many bits use for direction?

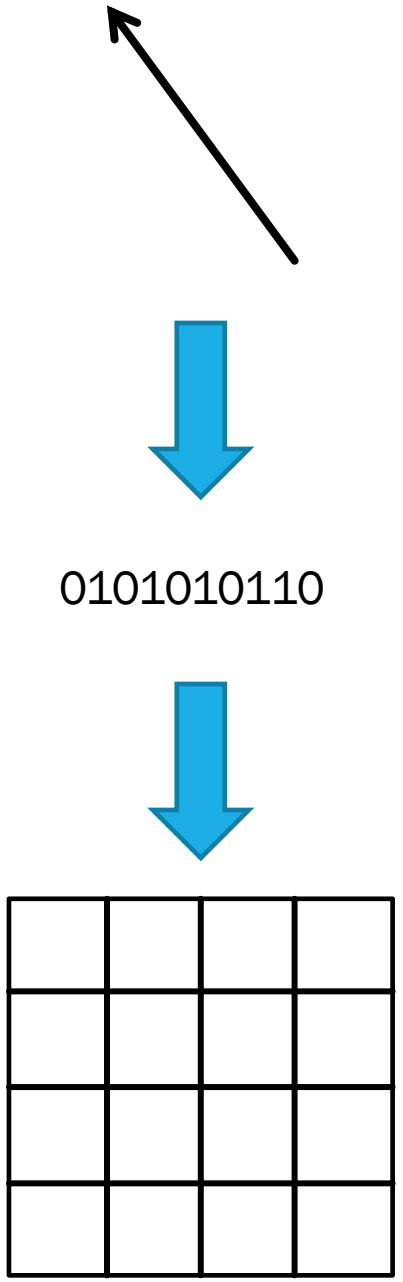
TWO-POINT SORTING KEY

- Interleaving bits of origin and termination point
  - Compact space footprint
  - Surface area of ray bounding volume
- Estimated termination points
  - Constant ray length (0.25 of the largest scene extent)
  - Spatial hashing (caching values from previous passes)
- Actual termination points
  - Not practical as we have to trace a ray twice
  - Theoretical performance upper bound



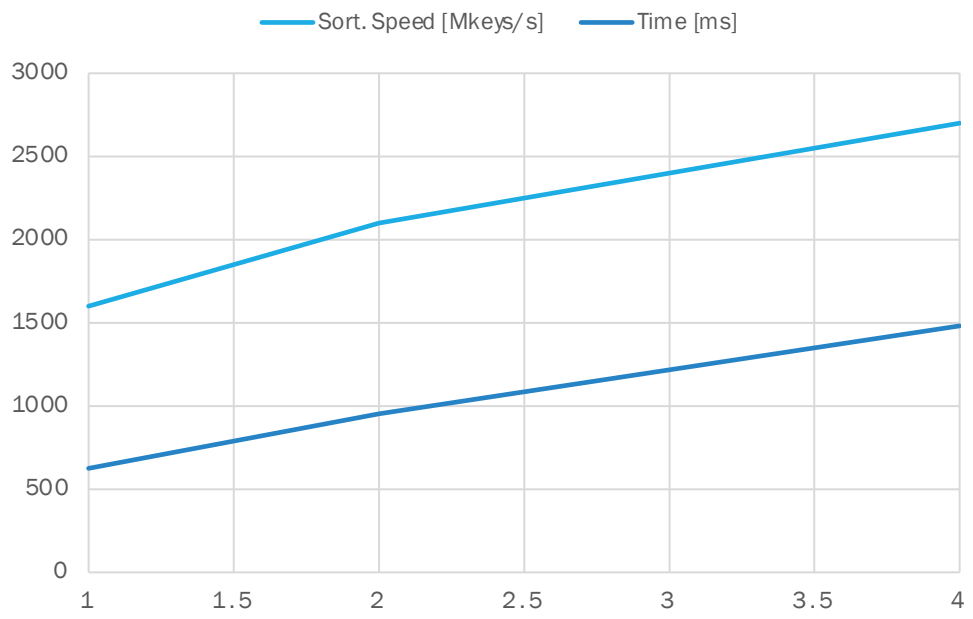
RAY LENGTH ESTIMATION

- Cashing ray lengths using spatial hashing
- Short Morton codes as keys (20 bits)
  - Hash table consists of 2^{20} cells
  - Computed by another method
- Accumulating ray lengths and ray counts
  - Returning average on query
  - Cells initialized with one dummy ray with length 0.25



SORTING ALGORITHM

- Parallel radix sort on GPU
  - CUB library [Merill and Grimshaw 2011]
- Sorting speed up to 6190 Mkeys/s on Tesla P100
- Benchmark – sorting 32-bit keys on RTX 2080 Ti
  - 1600 MKeys/s for 1M keys
  - 2100 MKeys/s for 2M keys
  - 2700 MKeys/s for 4M keys
- Sorting 64-bit keys is ~2.5x slower



EXPERIMENTAL SETUP

- 7 scenes (262k - 2833k tris)
- GPU ray tracing with RTX
  - OptiX 7 and DirectX 12
  - GPU RTX 2080 Ti
- Wavefront path tracer
  - Next event estimation
  - Point lights with radius 0.05
- Image resolution 1920x1080
- 32-bit Morton codes
- Not considering sorting overhead



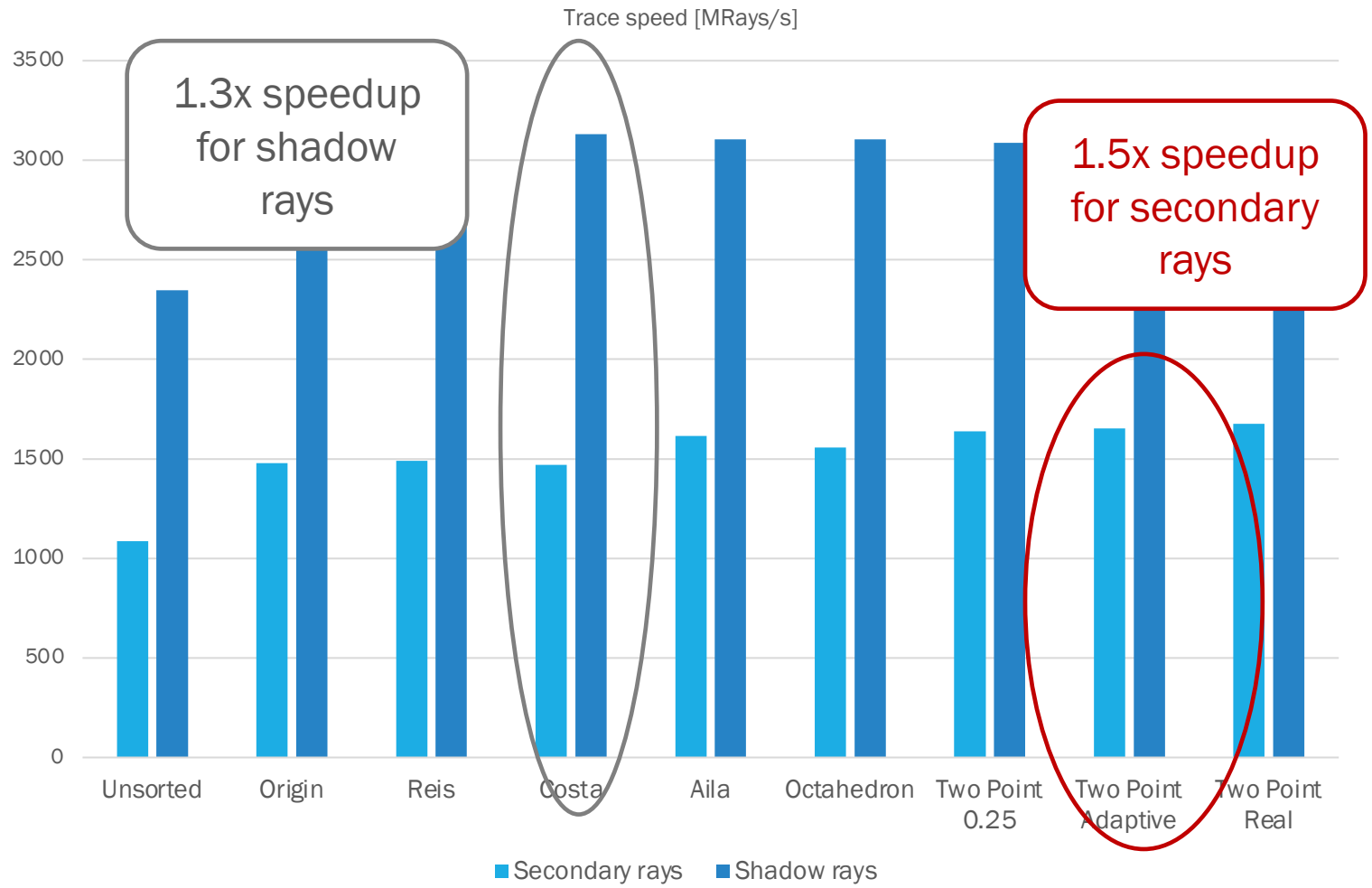




TESTED METHODS

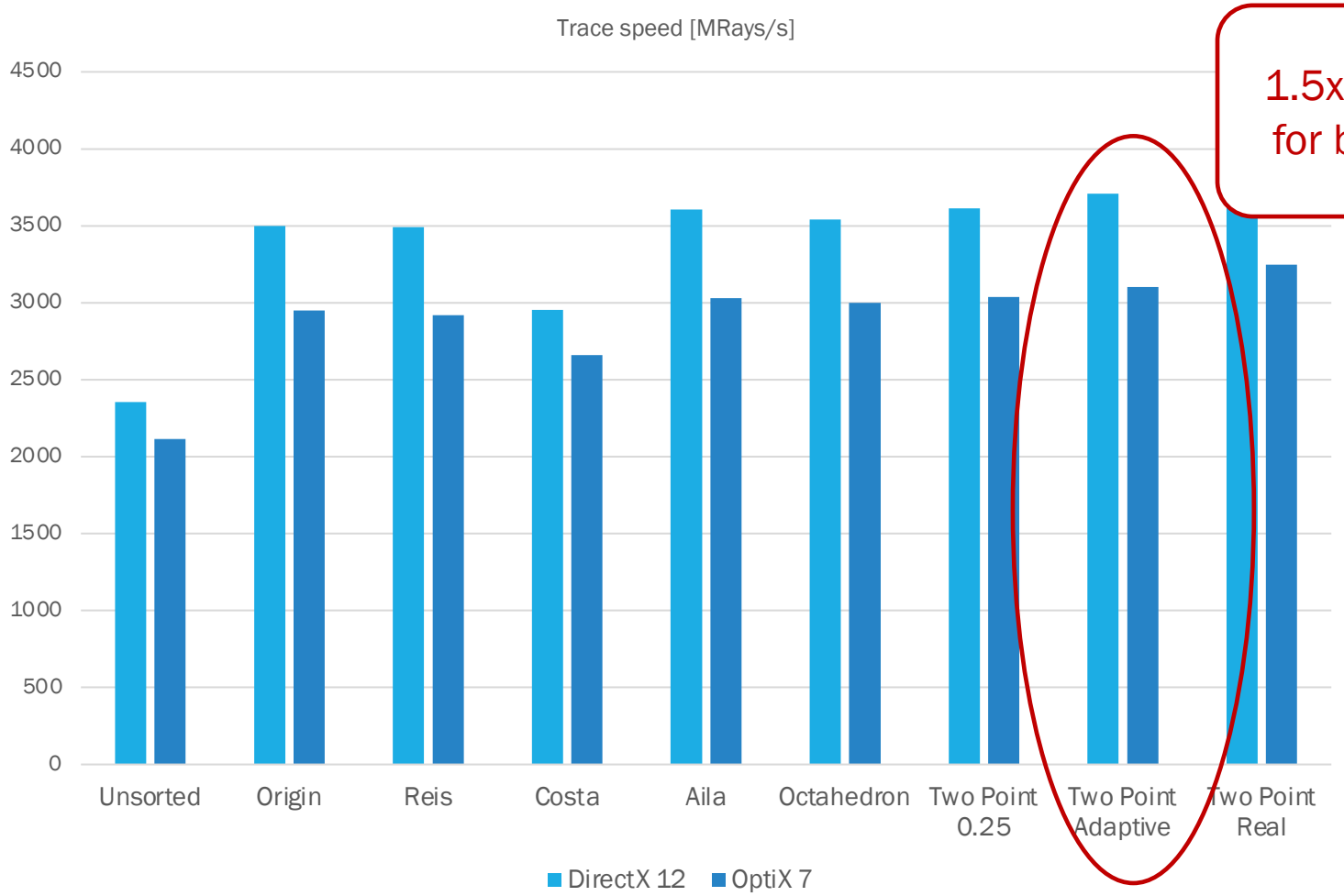
- Unsorted
  - Baseline
- Origin
  - Using only origins
- Reis
  - Origin-direction
- Costa
  - Direction-origin
- Aila
  - Origin-direction interleaved
- Octahedron
  - Octahedron direction parametrization
- **Two-Point 0.25**
  - Constant ray length 0.25
- **Two-Point Adaptive**
  - Adaptive ray length estimation
- **Two-Point Real**
  - Actual ray length

BREAKFAST – SECONDARY AND SHADOW RAYS (DIRECTX 12)



1347k tris

LIVING ROOM – DIRECTX 12 AND OPTIX 7 (SECONDARY RAYS)

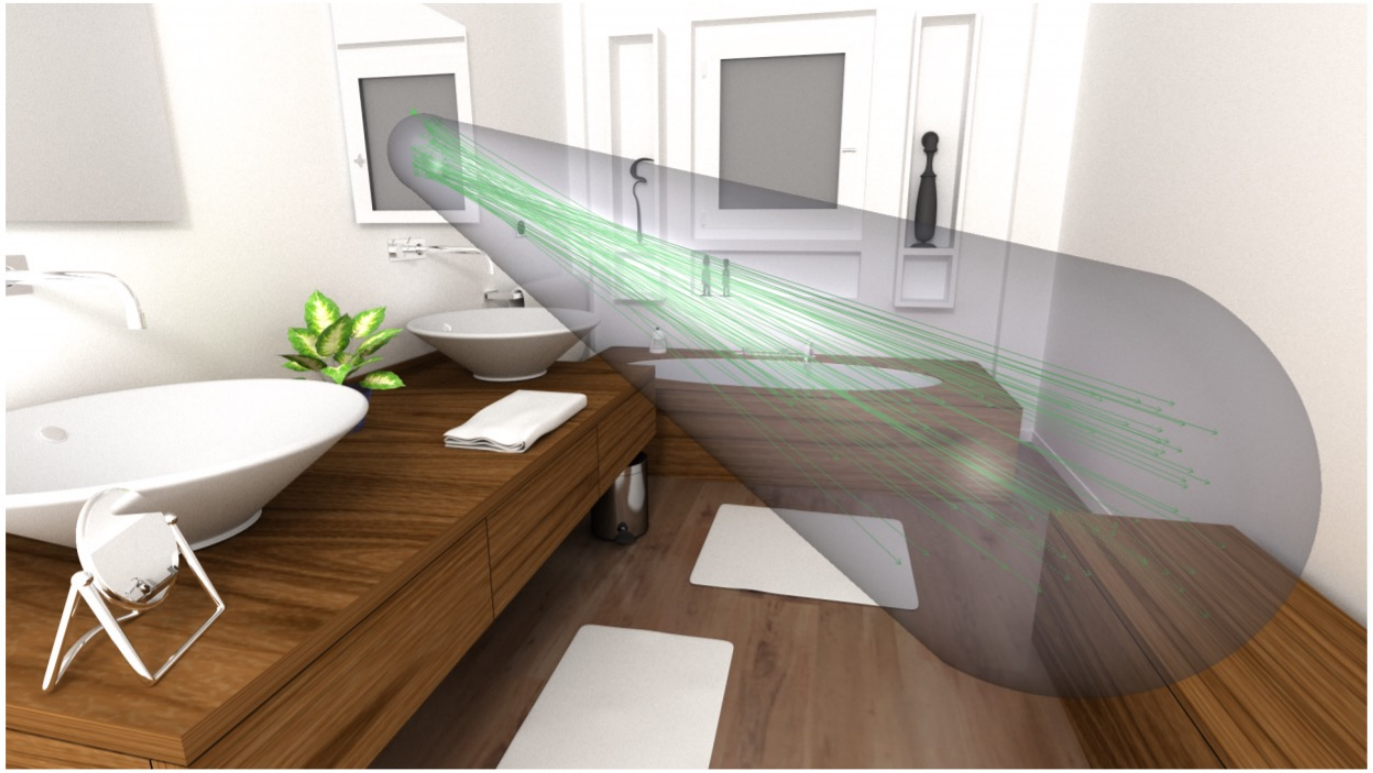


1.5x speedup for both APIs

580k tris

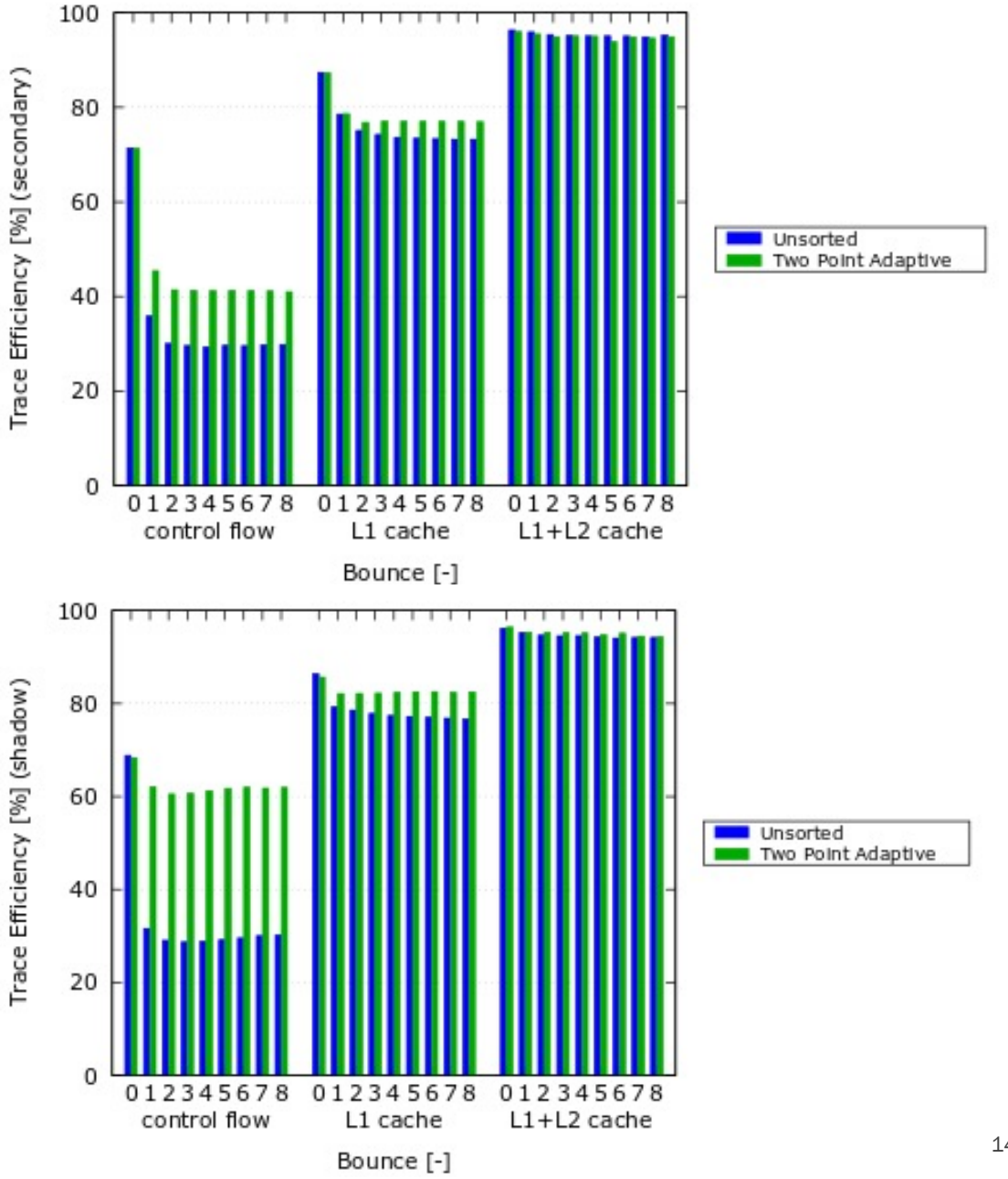
MEASURING RAY COHERENCE

- Surface area of ray bounding volume
- *Capsule* - conical section with two hemispheres
- Enclosing a groups of consecutive 64 rays (RTX scheduling)
- Weak to moderate correlation



PROFILING

- CUDA ray tracing kernel [Aila09]
- Control flow
  - 1.41x speedup for secondary rays
  - 2.21x speedup shadow rays
- Higher memory bandwidth
- L1 and L2 caches
  - Only marginal improvement



BISTRO – REORDERING OVERHEAD

- Two-Point Adaptive in OptiX
  - Morton codes gen. 0.33 ms
  - Index sorting 0.66 ms
  - **Ray reordering** **2.26 ms**
  - Ray length accum. 0.45 ms
- Total overhead 3.66 ms
- Trace time 1.17 ms
- Trace speedup 1.85x

2833k tris

REORDERING OVERHEAD

- Ray reordering does not pay off overall for RTX (hardware accelerated)
  - Extremely fast RTX trace kernel
  - Relatively slower sorting (~18%)
  - **Very slow actual reordering (~60%)**
- Ray reordering pays off overall in CUDA (software)
  - Similar speed of trace and sorting
  - Indirect access through sorted indices

CONCLUSION

- Surveyed ray reordering techniques for GPU ray tracing (RTX)
- Two-point method using estimated termination points
  - Good results for secondary rays (1.63x speedup on average)
  - Other methods more suitable for shadow rays
- Up to 2x speedup but difficult to recover reordering overhead

- Future work
  - A specialized method for shadow rays
  - Hardware sorting units

THANK YOU FOR YOUR ATTENTION!