Accelerated Single Ray Tracing for Wide Vector Units

Valentin Fuetterling°*, Carsten Lojewski°, Franz-Josef Pfreundt°, Bernd Hamann^ and Achim Ebert*

° Technische Universität Kaiserslautern

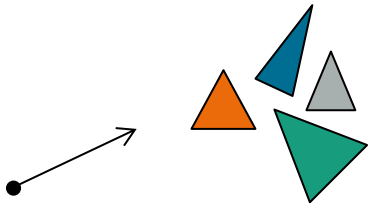
* Fraunhofer ITWM

^ UC Davis University of California

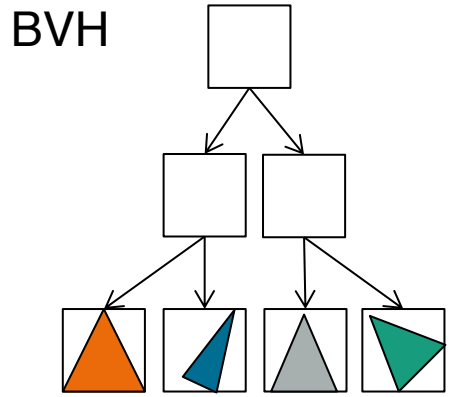
Fraunhofer

Introduction – Ray Tracing

- Ray
- Set of triangles
- Find first intersection



- Acceleration
  - Bounding Volume Hierarchy (BVH)
  - Requires traversal



Motivation

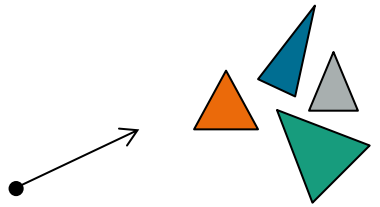
- Accelerating single ray traversal on data-parallel hardware

- Why single rays?
  - Ease of use
  - Consistent performance
  - Immediate results

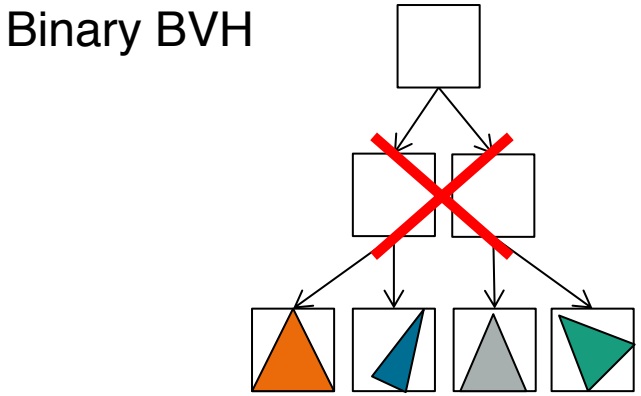
Dissection

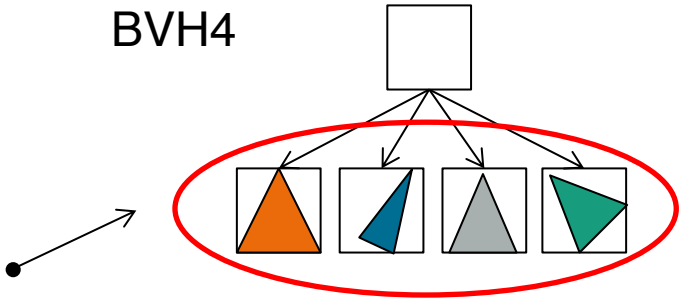
CURRENT SINGLE RAY TRAVERSAL

State-Of-The-Art



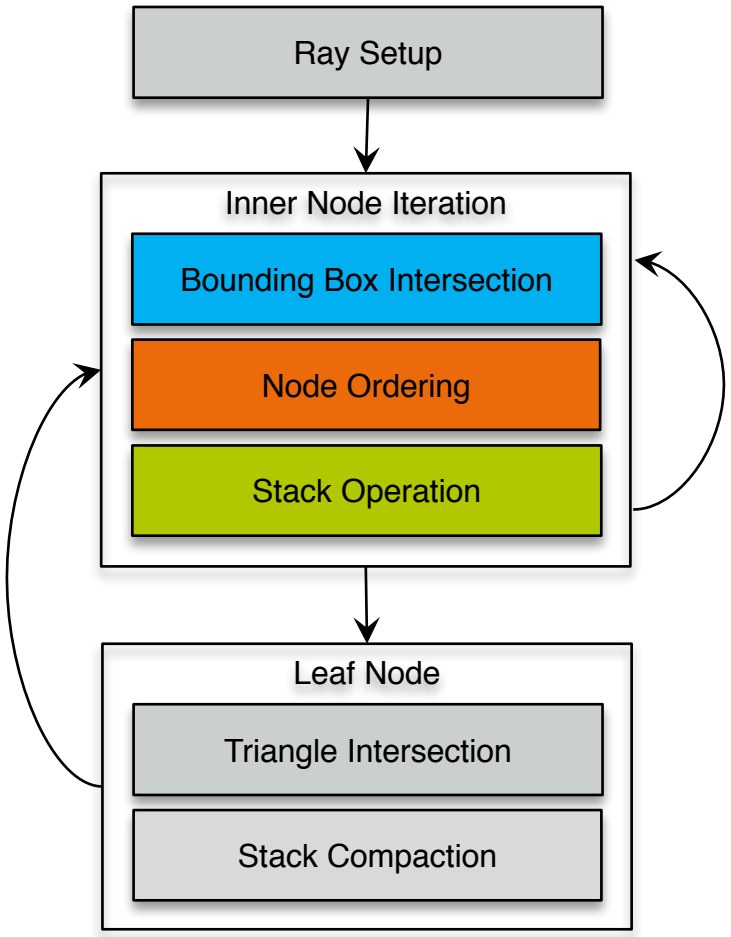
- Multi-branching BVH
  - Data parallelism
  - Vector instructions (SIMD)
  - Less traversal iteration
  - More coherent memory access

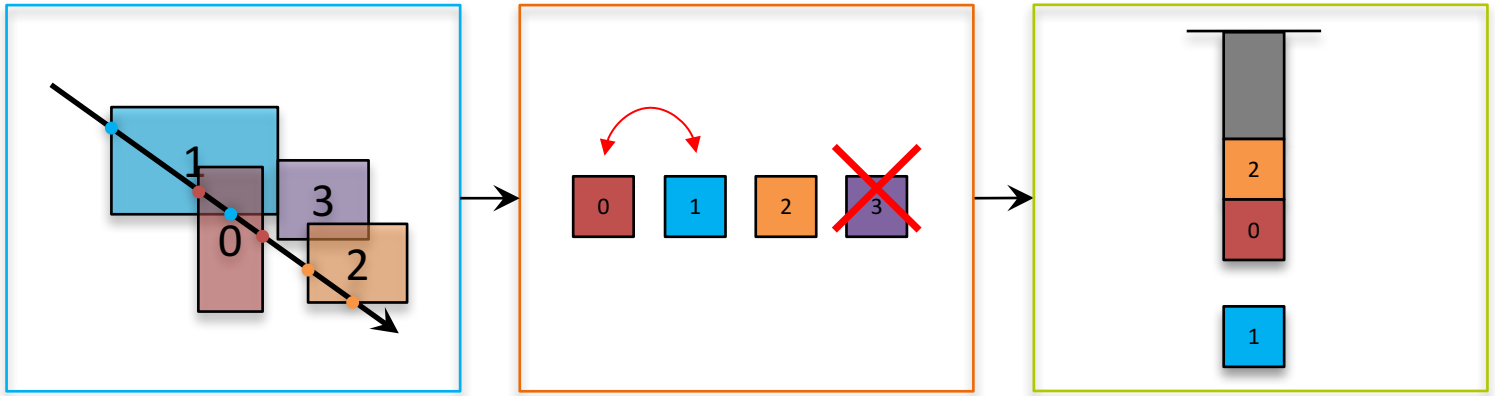




[Dammertz et al. 2008; Ernst and Greiner 2008; Wald et al. 2008]

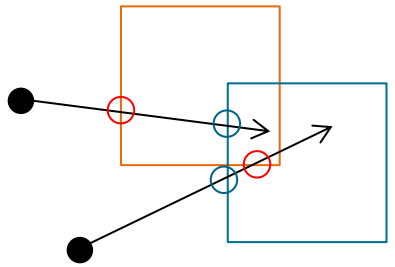
Building Blocks

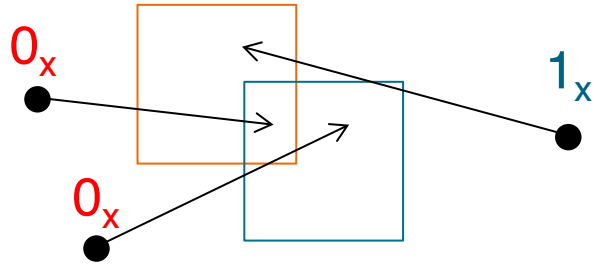




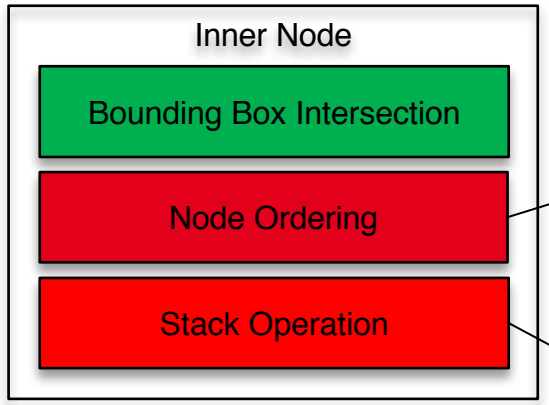
Node Ordering: Heuristics

- Ideal traversal order
  - Minimal traversal iterations
  - Unknown
- Distance heuristic
  - Intersection distance
- Sign heuristic
  - Pre-compute order





Data-parallel Limitations?




```
int numActiveNodes = popcnt(activeNodeMask);

if(numActiveNodes == 0)
{
    //pop stack
    ...
}
else if(numActiveNodes == 1)
{
    //use only active child as new node
    ...
}
else if(numActiveNodes == 2)
{
    //use closer child node as new node, push other to stack
    ...
}
else
{
    //sort child nodes
    ...

    for(int i=0; i<numActiveNodes-1; ++i)
    {
        //push to stack
        ...
    }

    //use closest child node as new node
    ...
}
```

- Variable number of active nodes N
  - Treat N < 3 as special cases for efficiency
  - Branchy code

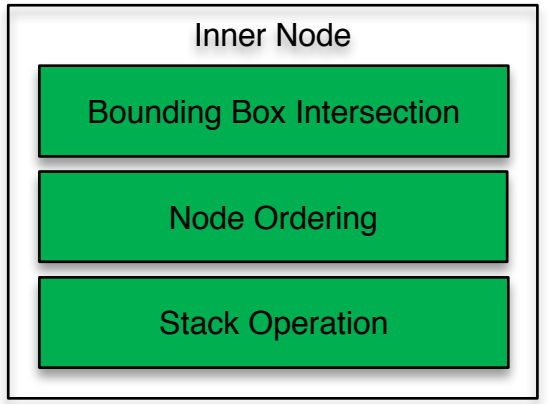
Accelerated Single Ray Tracing for **Wi**de **Ve**ctor Units

WIVE

Fraunhofer

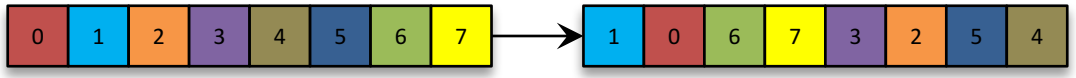
Contributions

- WiVe: Single Ray Traversal Algorithm
  - Encoding for sign-based ordering heuristic
  - Full data-parallel traversal for multi-branch BVHs
  - Constant-time (i.e. branch-less) inner node iteration

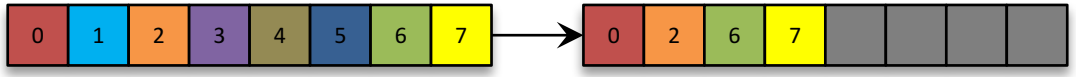


WiVe Key Ideas

- Encode fixed traversal orders into nodes
  - One order per ray signs combination (8)
  - Order defined by permute vector
- Map node ordering to a single permute vector operation

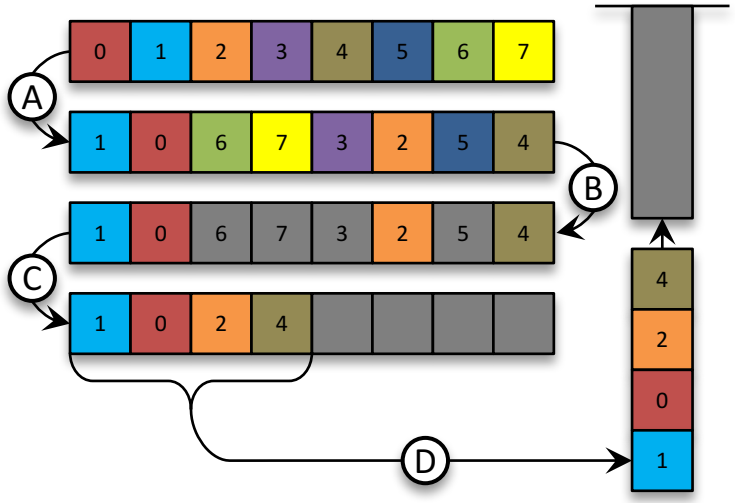


- Map stack operation to a single vector compress operation



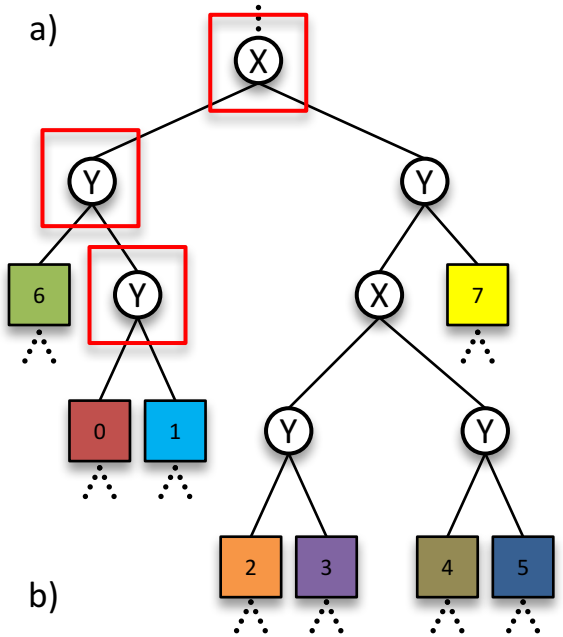
WiVe Algorithm

- Initially nodes are in memory order
- Permute nodes (A)
- Intersection test (B)
- Compress (C)
- Store to stack (D)



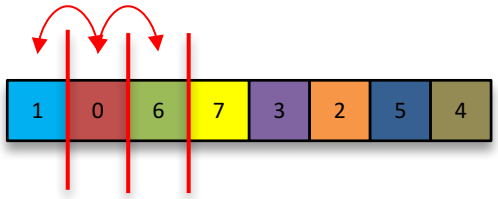
WiVe Node Order

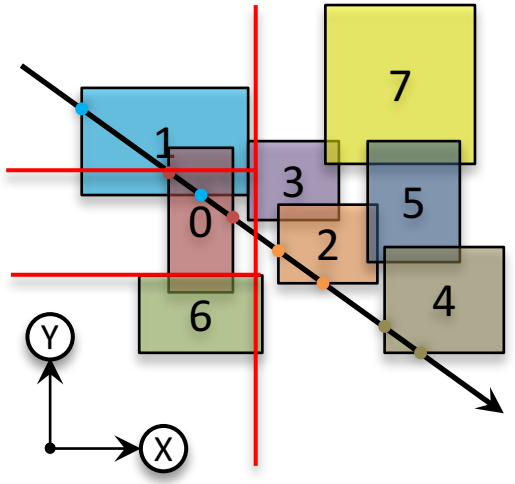
- Binary treelet with split axis labels (a)
  - Leaves form BVH8 node cluster
  - Split hierarchy determines permute vectors
- BVH8 node cluster in spatial domain (b)



- Ray with +x and -y signs (octant is 10b = 2)
- If signs[split axis] negative: Swap default order

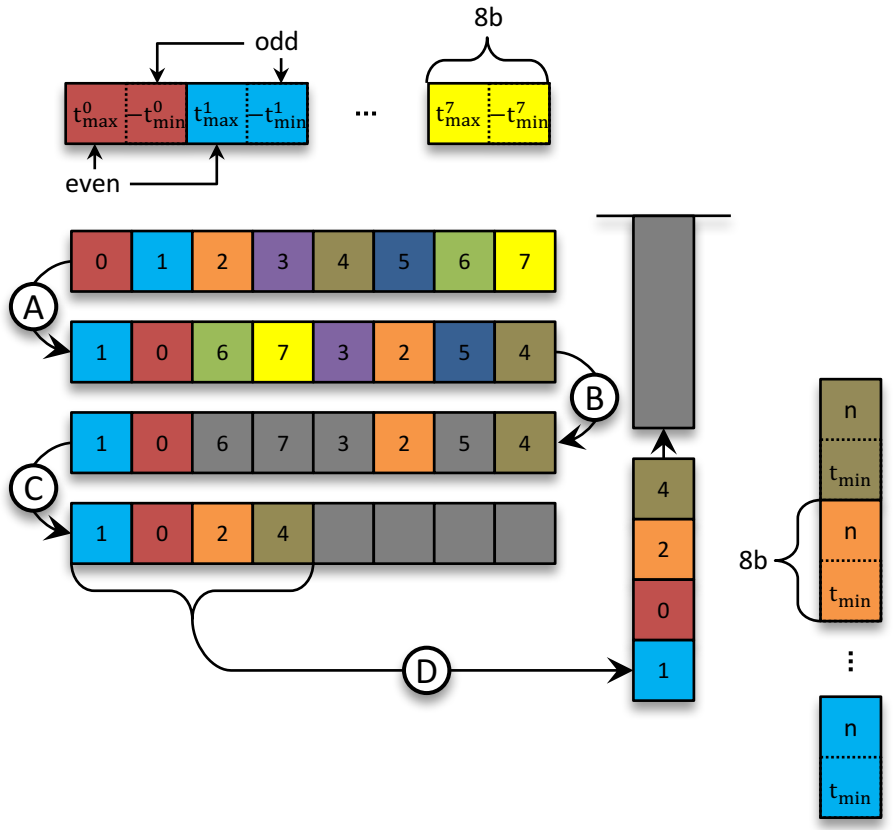
Permute_vector[2] =





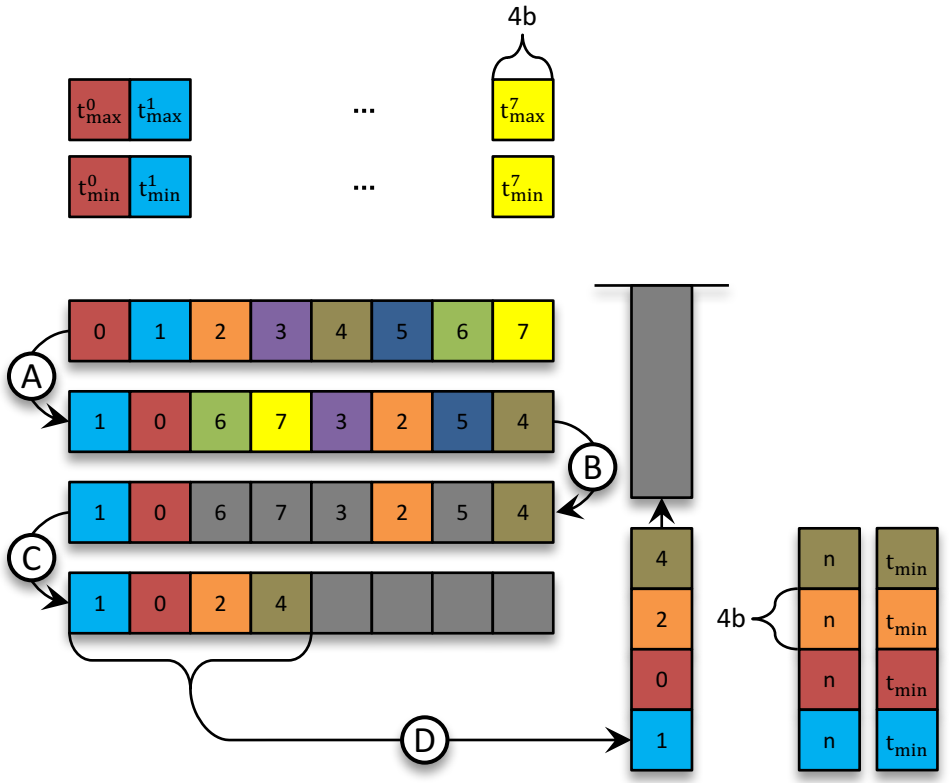
WiVe Configurations (BVH8-half)

- BVH branching-factor equals half the vector width
  - E.g BVH8 with 16-wide vector instructions
  - t_{min} and t_{max} values interleaved in register
  - Child offset and t_{min} interleaved on stack



WiVe Configurations (BVH8-full)

- BVH branching-factor equals full vector width
  - E.g. BVH8 with 8-wide vector instructions
  - t_{min} and t_{max} values in separate registers
  - Child offset and t_{min} on separate stacks



WiVe

EVALUATION

Experimental Setup

- Compare WiVe to Embree 2.15.0 single ray
  - Integrated into Embree Protoray benchmark
  - WiVe BVH derived from Embree BVH, same topology
  - BVH uses SAH-based binning (no spatial splits)
  - Path tracing with 8 bounces
- Two implementations of WiVe
  - BVH8-half for AVX-512 (16-wide)
  - BVH8-full for AVX2 (8-wide)

Input / Output

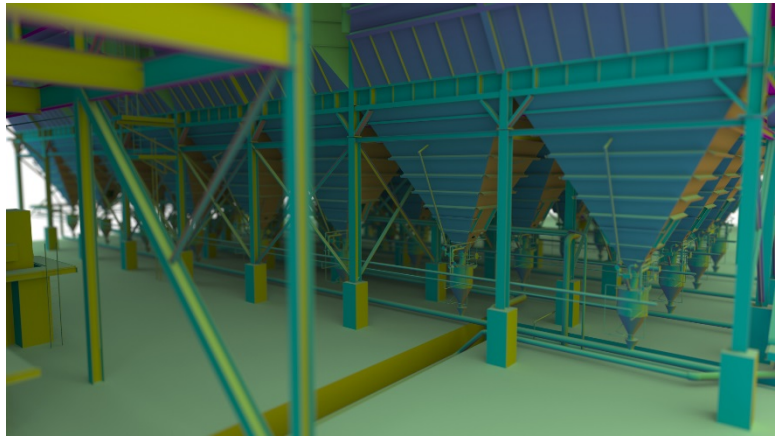
- White environment light
- Diffuse shading
- Normal colors
- 3840 x 2160 resolution

Mazda (5.7M)

Art Deco (10.5M)

San Miguel (10.7M)

Villa (37.5M)



Powerplant (12.8M)

Results – Order Heuristic

- Sign-based vs. distance-based heuristic
- Per-ray average indicators
- Very close, slight bias towards distance
- Worst case is San Miguel with 6-7% discrepancy

	Sign			Distance		
	Inner Nodes	Leaves	Triangles	Inner Nodes	Leaves	Triangles
Mazda	14.1	3.6	4.1	14.1	3.6	4.1
San Miguel	21.2	4.5	5.4	21.2	4.2	5.1
Art Deco	11.1	2.3	2.9	11.0	2.2	2.8
Powerplant	20.5	5.6	9.3	20.3	5.6	9.2
Villa	17.4	4.6	5.5	17.4	4.5	5.4

Results – Performance (BVH8-half)

- AVX-512 implementation
  - Native permute and compress instructions
- Measured on Intel® Xeon Phi™ 7250 @ 1.4GHz (Knight's Landing)
- Timings include full (off-screen) rendering

	Mazda	San Miguel	Art Deco	Powerplant	Villa
# triangles[M]	5.7	10.5	10.7	12.8	37.5
AVX-512					
WiVe	**126.7**	**73.1**	**165.0**	**85.4**	**87.4**
Embree	110.0	63.2	143.4	68.4	76.3
WiVe[+%]	**15**	**16**	**15**	**25**	**15**

Results – Performance (BVH8-full)

- AVX2 implementation
  - Native permute instruction, compress emulated with permute + shift + look-up table (1kb)
- Measured on dual-socket Intel® Xeon™ E5 2680v3 @ 2.5GH (Haswell)

	Mazda	San Miguel	Art Deco	Powerplant	Villa
# triangles[M]	5.7	10.5	10.7	12.8	37.5
AVX2					
WiVe	**74.0**	**46.0**	**97.2**	**57.4**	**48.8**
Embree	70.9	43.0	93.7	51.9	46.2
WiVe[+%]	**4**	**7**	**4**	**11**	**6**

Conlusion & Outlook

- Introduced the WiVe algorithm
  - Fully vectorized SIMD single ray traversal
  - Branch-free traversal iteration for multi-branching BVHs
  - Fastest on CPUs

- Combine WiVe with compression
- Investigate WiVe on GPUs
  - Trade throughput for latency

Thank you!

Email: valentin.fuetterling@itwm.fraunhofer.de
Blog: http://rapt.technology