TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

# Half-Precision Floating-Point Ray Traversal

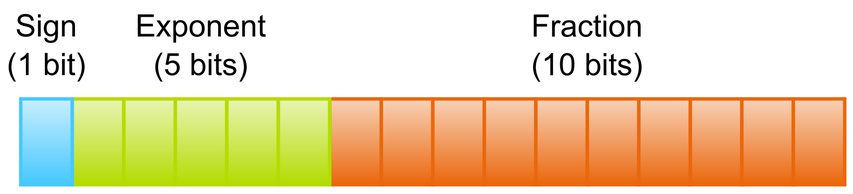
*Matias Koskela, Timo Viitanen, Pekka Jääskeläinen & Jarmo Takala*
Tampere University of Technology, Finland

## Half-Precision Floating-Point

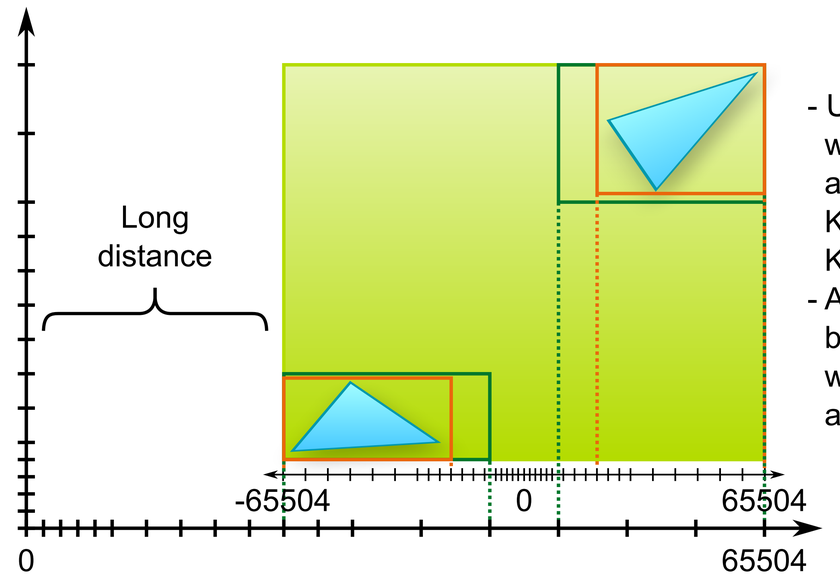
- 16-bit floating-point format defined in IEEE 754-2008 standard
- Storage support on most of the modern CPUs and GPUs
- Native computation support especially on mobile platforms
  (Up coming nVidia Pascal desktop GPUs are announced to have native computation support)

Pros:
- Smaller cache footprint (compared to "regular" 32-bit floats)
- More energy efficient
- Often doubles throughput

Cons:
- Less accurate
- Already 65520 is rounded up to infinity




## Hierarchical Bounding Volume Hierarchies




- Used in previous work (Mahovsky and Wyvill 2006, Keely 2014, Koskela & al 2015)
- Allows tighter bounding boxes with less accurate datatypes

= Tightest bounding box in world coordinates

= Tightest bounding box in hierarchical coordinates

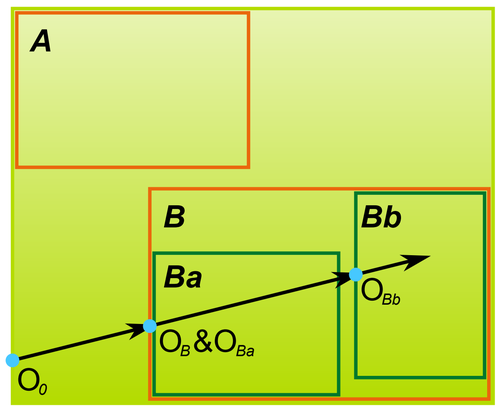
## The Proposed Ray Traversal Algorithm

Similar to (Keely, 2014), but can be done in existing half-precision hardware.

Ray traversal inner loop:
  if node is an inner node
    Compute ray-AABB intersection tests in the child's hierarchical coordinates in **half-precision floating-point** numbers

    For every intersected child
      Move ray origin to edges of intersected child bounding boxes
      Convert ray origin to hierarchical child coordinates
    end for
  else (node is a leaf node)
    Do triangle tests
  end if
end loop

## Ray Origin Movement



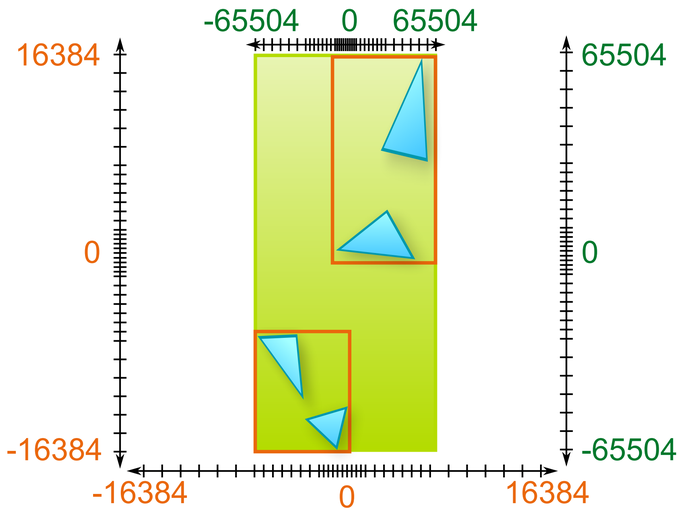

= Intermediate origin

= Child bounding box

= Grandchild bounding box

Origin is moved to the edge of every intersected child node's AABB
- This way origin stays within the half-precision non-infinite range even though it is converted to the child node's coordinates
- Requires extra computations & extra memory space for the intermediate values

## Conversion to Hierarchical Coordinates




0 = Coordinates in proposed method

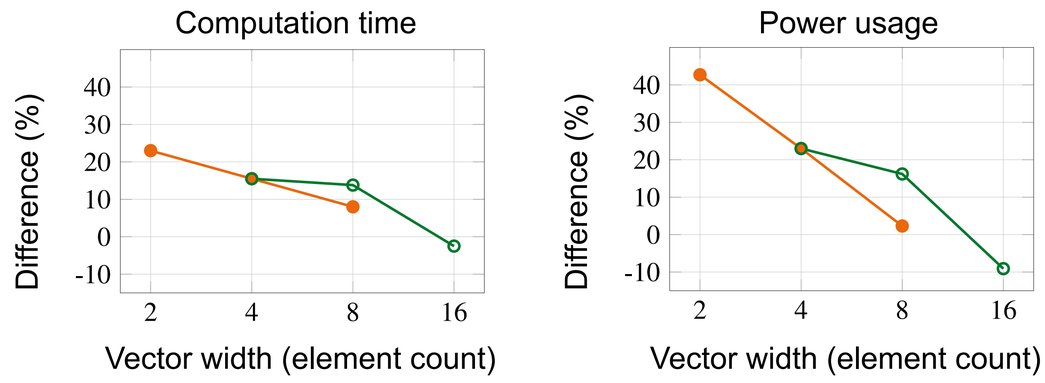
0 = Coordinates in previous work

The proposed method scales all hierarchical coordinates according to largest bounding box side dimension
- This way rays are not bending when they go deeper into hierarchy
- But precision is lost

Only the interval of -16384 ... 16384 is used to keep ray bounding box tests from rounding up to infinity (fourth of the maximum half-precision non-infinite interval -65504 ... 65504)

## Results

Average differences to regular single-precision traversal show that proposed method gets better when wider vectors are used




= MBVHS with different brancing factors

= Different ray packet sizes with MBVH4

- With narrow vectors the benefits are hidden by reduced precision bounding volumes, extra computations and extra stacks for intermediate values
- If a ray tracer is fastest with wide vector instructions and the targeted hardware has native half-precision computation support the proposed ray traversal should be worth testing.

## References:

Mahovsky, J. and Wyvill, B. (2006). *Memory-conserving bounding volume hierarchies with coherent ray tracing*. Computer Graphics Forum, 25(2):173–182.
Keely, S. (2014). *Reduced precision for hardware ray tracing in GPUs*. In Proceedings of the High Performance Graphics Conference, pages 29–40.
Koskela, M., Viitanen, T., Jääskeläinen, P., Takala, J., and Cameron, K. (2015). *Using half-precision floating-point numbers for storing bounding volume hierarchies*. In Proceedings of the 32nd Computer Graphics International Conference.