Hash-Based Ray Path Prediction: Skipping BVH Traversal Computation by Exploiting Ray Locality

Francois M. Demoullin
fdemoullin@ece.ubc.ca
The University of British Columbia

Ayub A. Gubran
ayoubg@ece.ubc.ca
The University of British Columbia

Tor M. Aamodt
aamodt@ece.ubc.ca
The University of British Columbia

Background & Motivation

Insight:

- The cost of traversing deep BVH trees can be reduced by exploiting ray locality
- Rays from close-by origins and similar directions follow a similar path through the tree

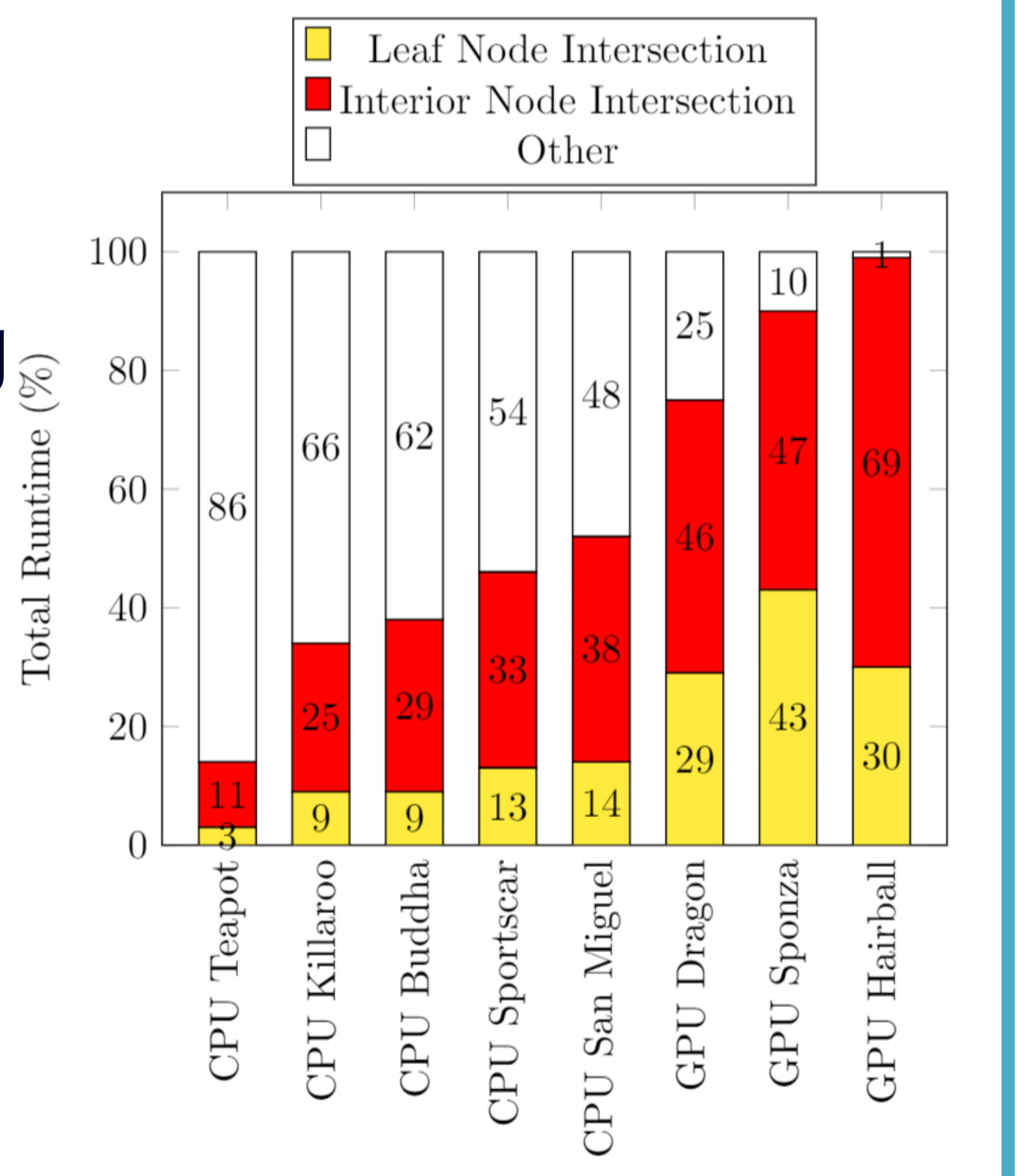

Figure 1: Runtime proportions of increasingly complex scenes – CPU + GPU

This work:

- Highlights the potential of ray path prediction for accelerating hierarchical tree traversal
- Proposes a hash function to predict the path of similar rays
- Presents a limit study that quantifies ray locality and evaluates its potential for performance improvements

Hash Function

- Map ray properties of origin and direction to unique index into predictor table
- Create hash efficiently in hardware by extracting important bits from IEEE 754 floating point representation
- Map rays with similar properties to the same predictor table entry

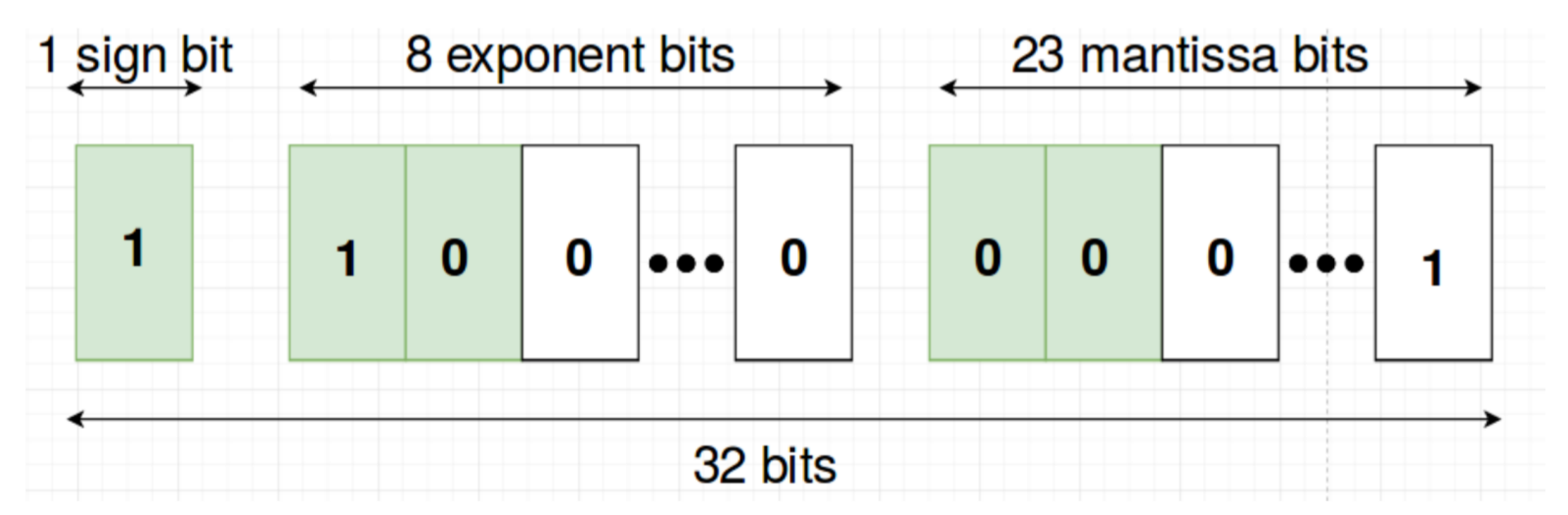

Figure 2: Illustration of our hash function's extraction of bits from IEEE 754 floating point with an example precision of 2 bits. Bits marked in green are included in the hash representation

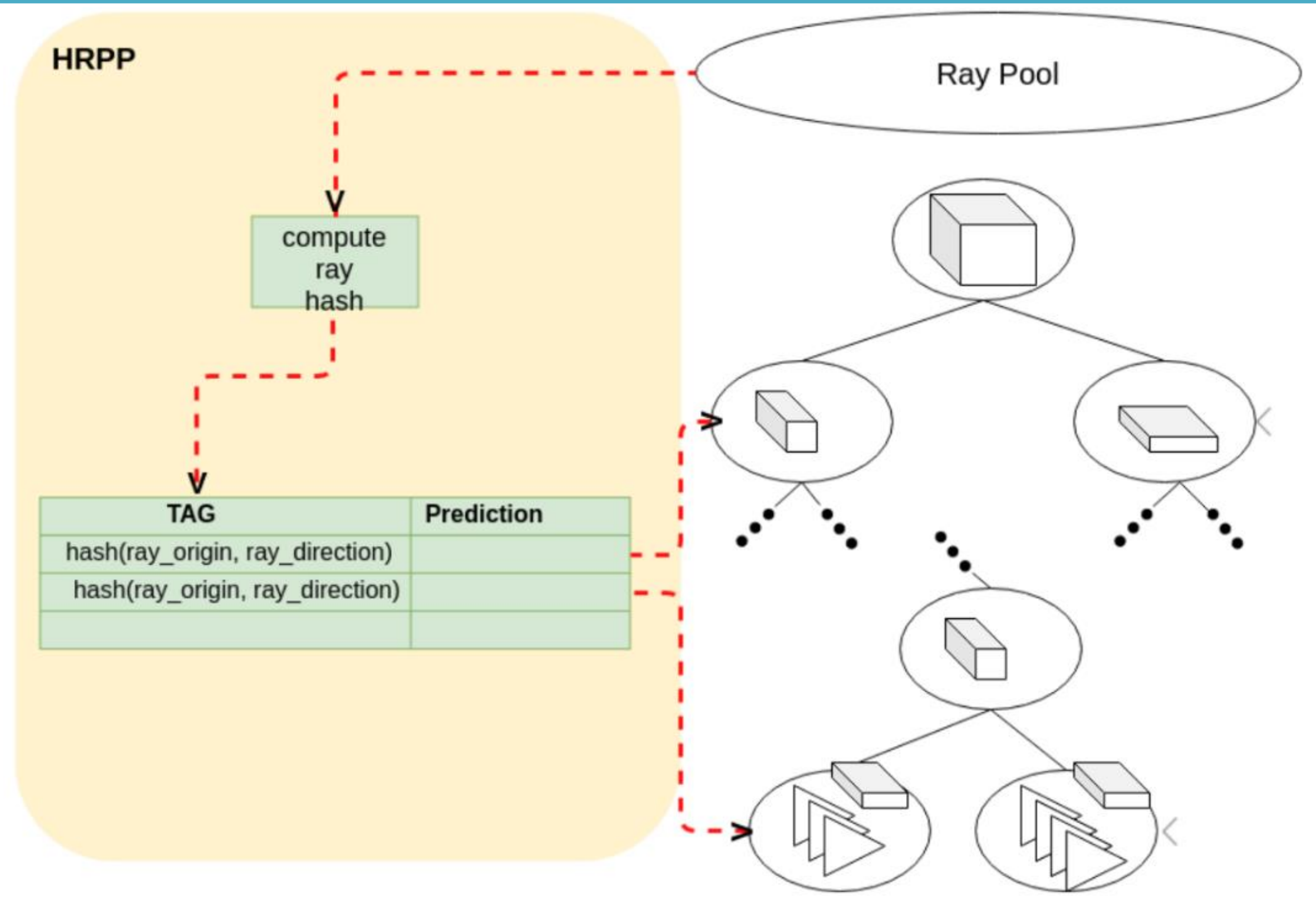
HRPP
Ray Pool
compute ray hash
TAG
Prediction
hash(ray_origin, ray_direction)
hash(ray_origin, ray_direction)

Figure 3: Binary BVH Acceleration Structure augmented by HRPP. The interior nodes are represented by their bounding boxes. Leaf nodes have both bounding boxes and triangular scene geometry. HRPP additions are marked in green. The red arrows represent the control flow from root to leaves through HRPP

Hash-Based Ray Path Prediction (HRPP)

- HRPP exploits ray locality [PKGH97] that is naturally present in a frame but goes unexploited during the BVH traversal
- Skip interior node computation by predicting which leaf nodes a ray will intersect based on information collected from previously traced similar rays
- Use HRPP's Hardware Predictor Table to make a prediction for certain rays
- Bypass interior node computation when predicting correctly
- The penalty for incorrect prediction is additional ray-leaf intersections in addition to a full BVH prediction

Limit Study

- Simulate HRPP in software by extending PBRT [PH10]
- Establish an upper bound on ray coherence in test scenes assuming infinite amount of memory
- Quantify memory cost of exploiting ray locality and the skipped interior node computation

Scene	hit-any savings(%)	size (MB)	closest-hit savings(%)	size (MB)
Teapot	4	16	69	2
Killeroo	48	10.8	52	84
Buddha	23	18	41	18
Sportscar	<1	21	32	18

Table1: Evaluation of test scenes – resolution 1024 x 1024 – 8spp – hash precision 6

Discussion

- We use Precision and Recall to evaluate HRPP predictions (true/false positives, true/false negatives)
- Hash function precision is highly correlated with number of predictions, prediction accuracy, and predictor table size

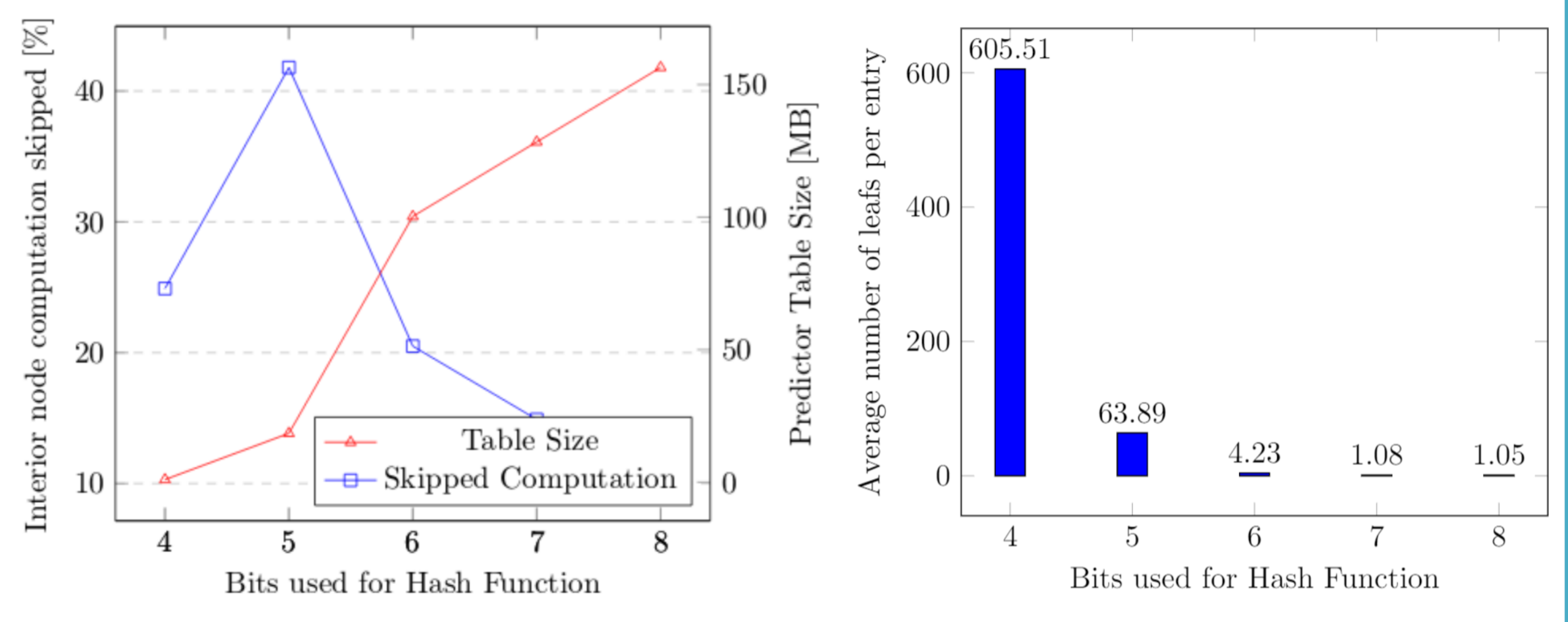

Figure 4: Effect of hash function precision on skipped computations, table size and number of table entries – resolution 1024 x 1024 – 8 spp

- Chance of false positive can lead to incorrect visual output for closes intersection rays
- Go-Up-Level, how many layers above the leafs should HRPP predict
- Samples per pixel is inversely correlated with HRPP's efficiency because of hash conflicts

Future Work

- Address memory footprint: implement replacement policy for predictor table
- Evaluate HRPP on more scenes with increased geometric and illumination complexity
- Use HRPP to package coherent rays for increased SIMD efficiency
- Evaluate precision and recall and quantify the amount of false positives
- Implement conservative hit-any predictor

THE UNIVERSITY OF BRITISH COLUMBIA

References
[PH10] PHARR M., HUMPHREYS G.: Physically Based Rendering, Second Edition: From Theory To Implementation, 2nd ed. Morgan Kaufmann Publishers Inc., 2010
[PKGH97] PHARR M., KOLB C., GERSHBEIN R., HANRAHAN P.: Rendering complex scenes with memory-coherent ray tracing. SIGGRAPH '97. 1