An Architecture for Hierarchical Collision Detection

Gabriel Zachmann & Günter Knittel
Uni Bonn Uni Tübingen

Motivation

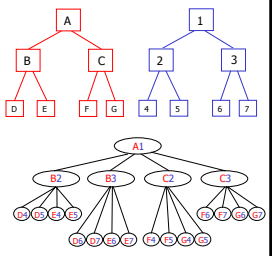
- Fundamental operation:
  - Virtual prototyping
  - Rigid bodies
  - Interaction in VR
  - Haptic rendering

- The bottleneck:
  - Few years ago: graphics
  - Today: simulation / coll.det.

Bounding Volume Tree Traversal

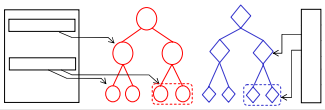
Traditional scheme:

traverse(X, Y)
if X,Y do not overlap **then**
  return
if X,Y are leaves **then**
  check polygons
else
  for all children pairs **do**
    traverse(X_i, Y_j)



New Traversal Scheme

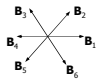
- Problem with old scheme:
  1. Nodes are visited multiple times
  2. Node-specific part of BV overlap test performed multiple times
- New scheme solves problem 2 and alleviates 1
- Idea:
  - Traverse *one* tree ("tumbled" nodes)
  - Stack of lists of nodes of other tree ("aligned" nodes)
  - Hybrid between depth-first and breadth-first

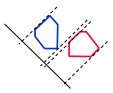


Discrete Orientation Polytopes (DOPs)

- Generalization of bounding boxes
- Overlap test:

D, E do not intersect $\Leftrightarrow$
$\exists i:\ d_i > e_{i+\frac{k}{2}} \vee d_{i+\frac{k}{2}} < e_i$



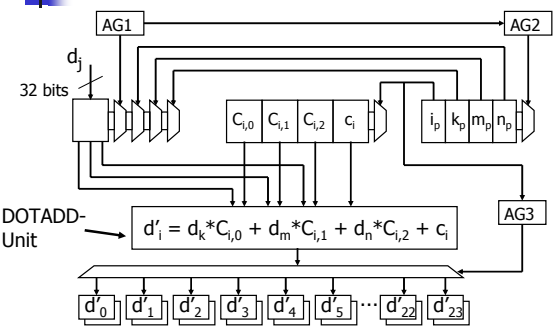


- Transformation:

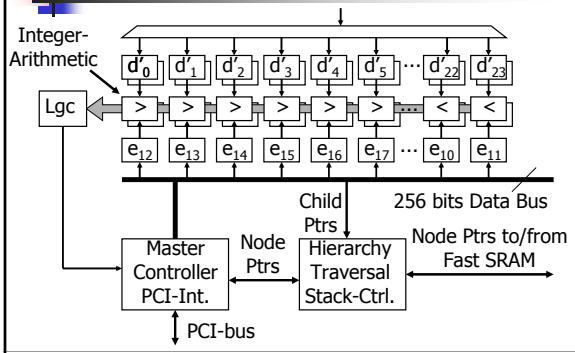
$$d_i' = \mathbf{B}_i \begin{pmatrix} \mathbf{b}_{j_{i,1}} \\ \mathbf{b}_{j_{i,2}} \\ \mathbf{b}_{j_{i,3}} \end{pmatrix}^{-1} \begin{pmatrix} d'_{j_{i,1}} \\ d'_{j_{i,2}} \\ d'_{j_{i,3}} \end{pmatrix} + \boxed{\mathbf{B}_i \mathbf{o}} \quad , \quad \mathbf{b}_j = \mathbf{B}_j \mathbf{M}^{-1}$$

- Cost: O(k), previously $O(k^2)$

Hardware Architecture – DOP Transformation



Hardware Architecture – Overlap Test



Hardware Architecture – Additional Features

- Stack Engine:
  - Facilitates optimized Hierarchy Traversal
  - Maintains "Stack of Stacks"
  - Node Pointers from Fast SRAM
- Triangle Intersection Test Unit:
  - Re-Use of DOTADD, Register Files
  - Small, low-performance Unit
- Memory Interface:
  - High-Speed 256-bit Interface to local SDRAM

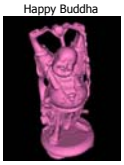
Hardware Architecture – Performance Estimate

- Assumptions:
  - Chip Clock 266MHz, DDR-SDRAM 133MHz, 2-2-2 Access Characteristic
  - Burst Length = eight 256-bit Words, One Node =128 Bytes in Memory
  - Random Access to a Node Pair: 16 Cycles
  - Results Register ready after 88 Cycles
  - The first two Aligned Node Pairs processed after 100 Cycles
  - All further Node Pairs are processed after additional 10 Cycles
- Asymptotic Performance:
  - T = (100 + (a-2) * 10) * t;
  - a: Number of Node Pairs in a List, t: Number of Lists

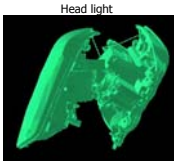
Evaluation of the New Scheme

- Platform: PentiumIII, 1GHz
- Procedure: two objs, one rotating
- Suite:

Happy Buddha

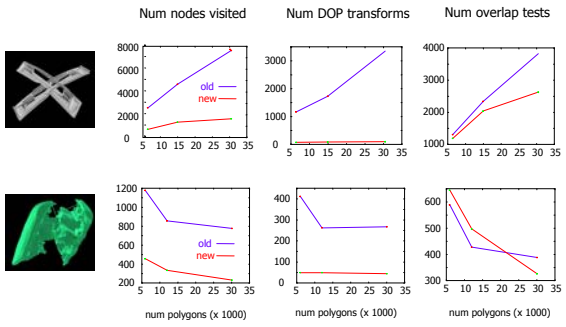


Head light

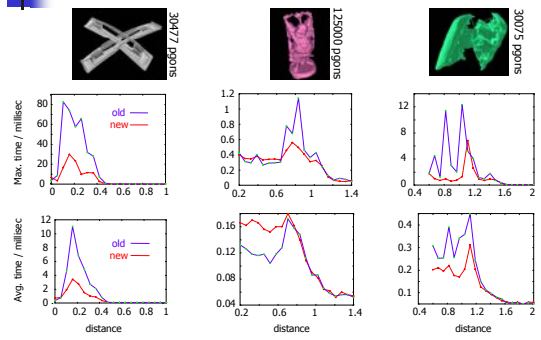


Car (courtesy VW)

Reduction of Memory Transfer



Speedup



Speedup HW vs SW (average case)

Obj	Num pgons	Aligned nodes visited	Tumbled nodes visited	Pgon checks	Time in HW	Time in SW	Speedup
Filter	19 326	12 474	240	1 660	153	14 936	98
Headlight	30 075	389	73	68	15	524	36
Door lock	62 023	279	81	9	15	401	27
Car body	60 755	259	66	55	12	383	31
Buddha	125 000	159	50	7	9	240	26

Speedup HW vs SW (worst case)

Obj	Num pgons	Aligned nodes visited	Tumbled nodes visited	Time in HW	Time in SW	Speedup
Filter	19 326	542 888	3 553	5 638	717 164	127
Headlight	30 075	7 065	937	207	9 226	44
Door lock	62 023	4 854	877	178	6 804	38
Car body	60 755	3 076	538	110	5 390	49
Buddha	125 000	3 345	301	77	4 074	53

Conclusions and Future Work

- Efficient Algorithms for Hierarchy Traversal & Triangle Intersection Test
- Compact and fast Hardware Architecture:
  - All Registers together need less than 100k Transistors
  - Not much Floating-Point Arithmetic
  - FPGA-Implementation?
- Future Work:
  - Improve Hardware Design
  - Reduce Latency

The End.

Thanks!