Sony Computer Entertainment US Research & Development http://www.research.scea.com

Optimizing Proximity Queries for CPU, SPU and GPU

SIGGRAPH 2010

Erwin Coumans

In a nutshell

- Intro to the Bullet physics engine
- Broadphase acceleration
  - Parallel sweep and prune broadphase, Uniform/Hierarchical Grid on GPU
  - Dynamic AABB tree general purpose acceleration structure
- Midphase acceleration
  - Dynamic AABB tree, History traversal of BVH trees on GPU
  - Stackless quantized BVH trees on SPU, GPU
- Narrowphase collision detection
  - All you need is a support map: CCD CA, GJK closest points, EPA PD, deformable objects
  - Cubemap to accelerate the support mapping function on GPU

Bullet physics engine

- Simulate Rigid Body, Cloth, Deformables
- Open source using the Zlib license
- Free for commercial use
- Written in C++
- OpenCL and Direct Compute for GPU

Primary parallel target hardware

- PlayStation 3 Cell SPUs
- Multi-core CPUs, Xbox 360 etc
- GPGPU CPU-GPU with shared memory
  - AMD Fusion, Intel Sandy Bridge

Some games using Bullet Physics

Destroying LA for “2012”

Tuesday @ 2pm. Room 515 AB

DIGITALDOMAIN®

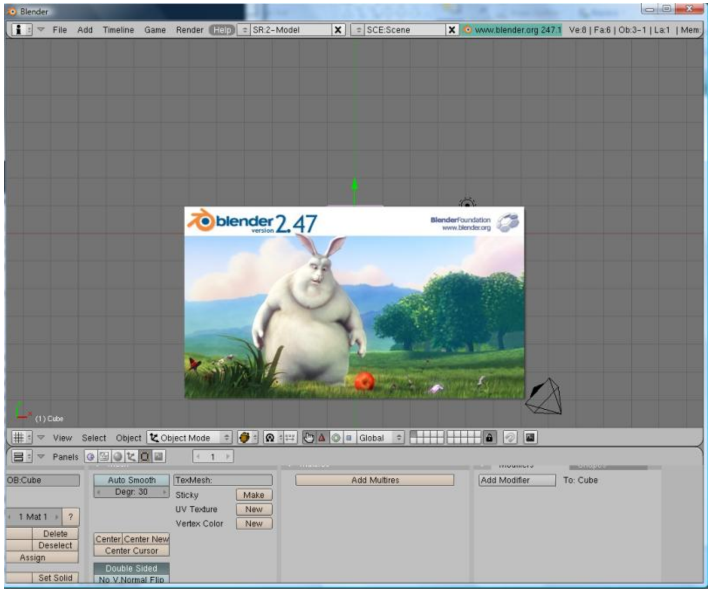
Movie studios using Bullet

- Sony Imageworks, Weta Digital, Disney Animation, Framestore, PDI Dreamworks, Digital Domain etc.

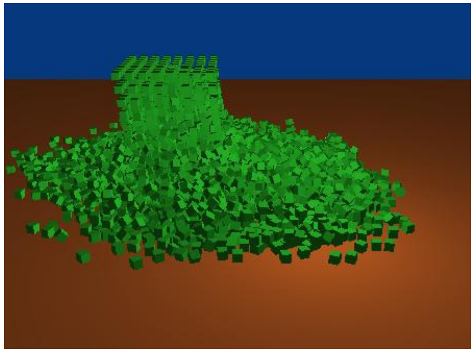
IT AIN'T OGRE... TIL IT'S OGRE
DreamWorks
SHREK
THE FINAL CHAPTER
3D
In Theatres, real D 3D and IMAX 3D
MAY 21

Bullet Authoring tools

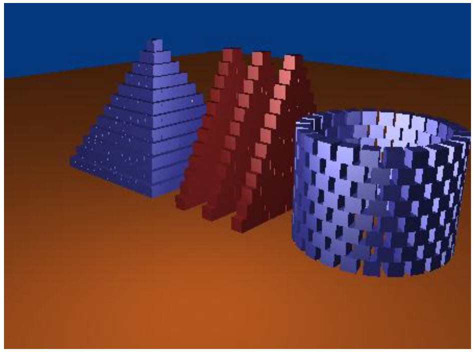
- Cinema 4D 11.5
- Lightwave
- Blender
- Maya Dynamica Plugin
- Houdini Plugin



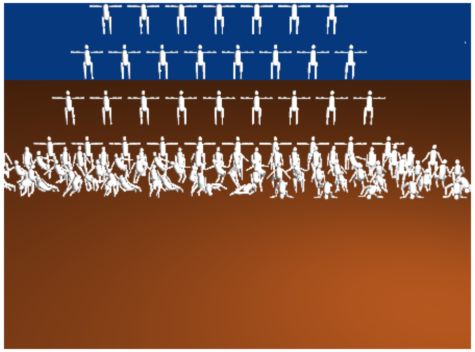
Collision Detection Benchmarks



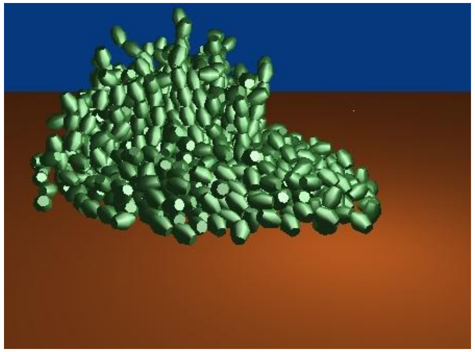
3000 falling boxes



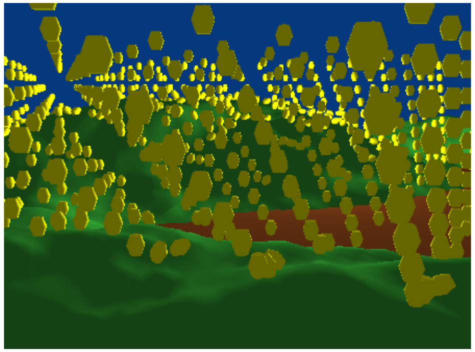
1000 stacked boxes



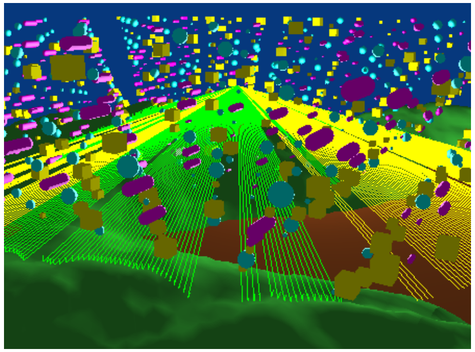
136 ragdolls



1000 convex hulls

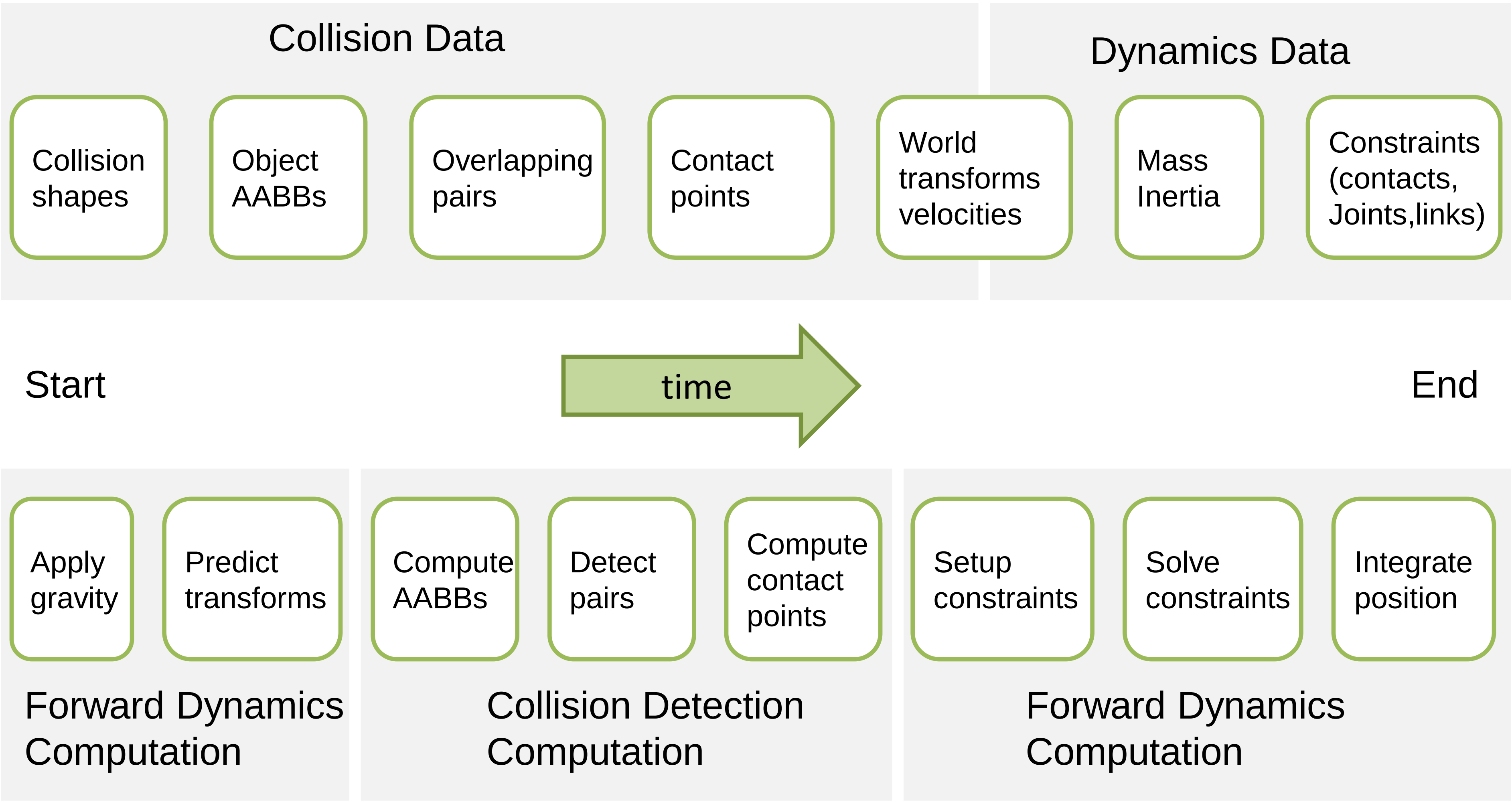


1000 convex
against trimesh



ray casts against 1000
primitives and
trimesh

Multi Physics Pipeline



Collision Data

Dynamics Data

- Collision shapes
- Object AABBs
- Overlapping pairs
- Contact points
- World transforms velocities
- Mass Inertia
- Constraints (contacts, Joints,links)

Start

time

End

Forward Dynamics Computation

- Apply gravity
- Predict transforms

Collision Detection Computation

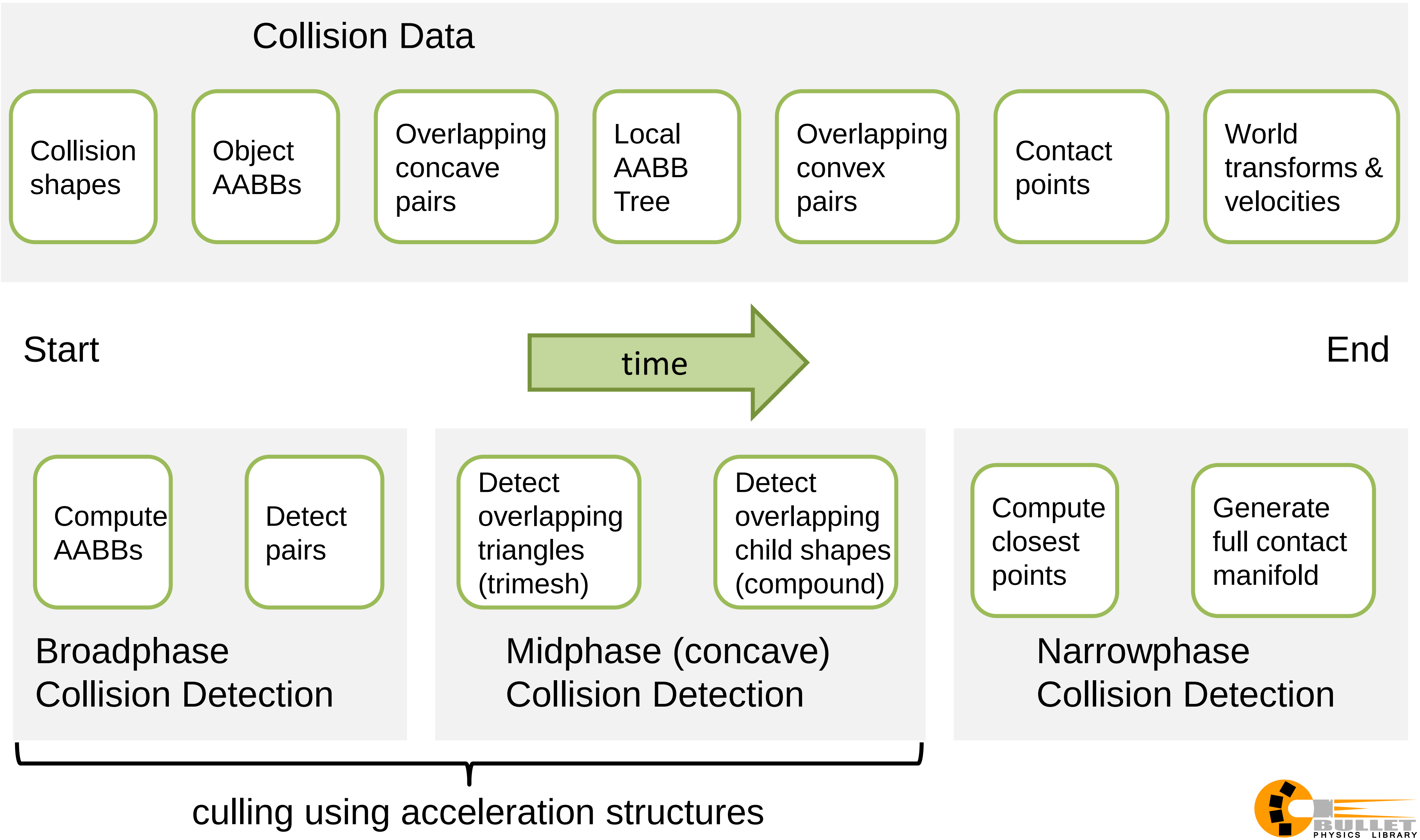
- Compute AABBs
- Detect pairs
- Compute contact points

Forward Dynamics Computation

- Setup constraints
- Solve constraints
- Integrate position

AABB = axis aligned bounding box

Collision Detection Pipeline



Collision Data

Collision shapes

Object AABBs

Overlapping concave pairs

Local AABB Tree

Overlapping convex pairs

Contact points

World transforms & velocities

Start

time

End

Compute AABBs

Detect pairs

Broadphase Collision Detection

Detect overlapping triangles (trimesh)

Detect overlapping child shapes (compound)

Midphase (concave) Collision Detection

Compute closest points

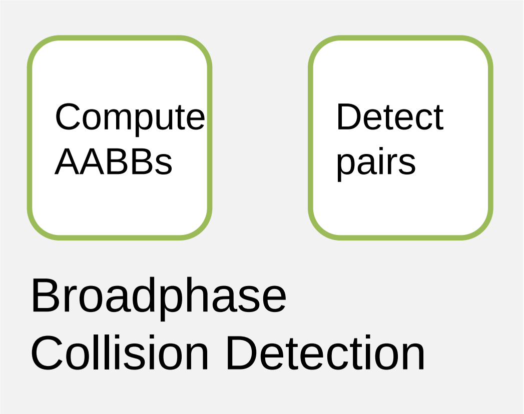
Generate full contact manifold

Narrowphase Collision Detection

culling using acceleration structures

Broadphase N-body problem

- Avoid brute-force N*N tests



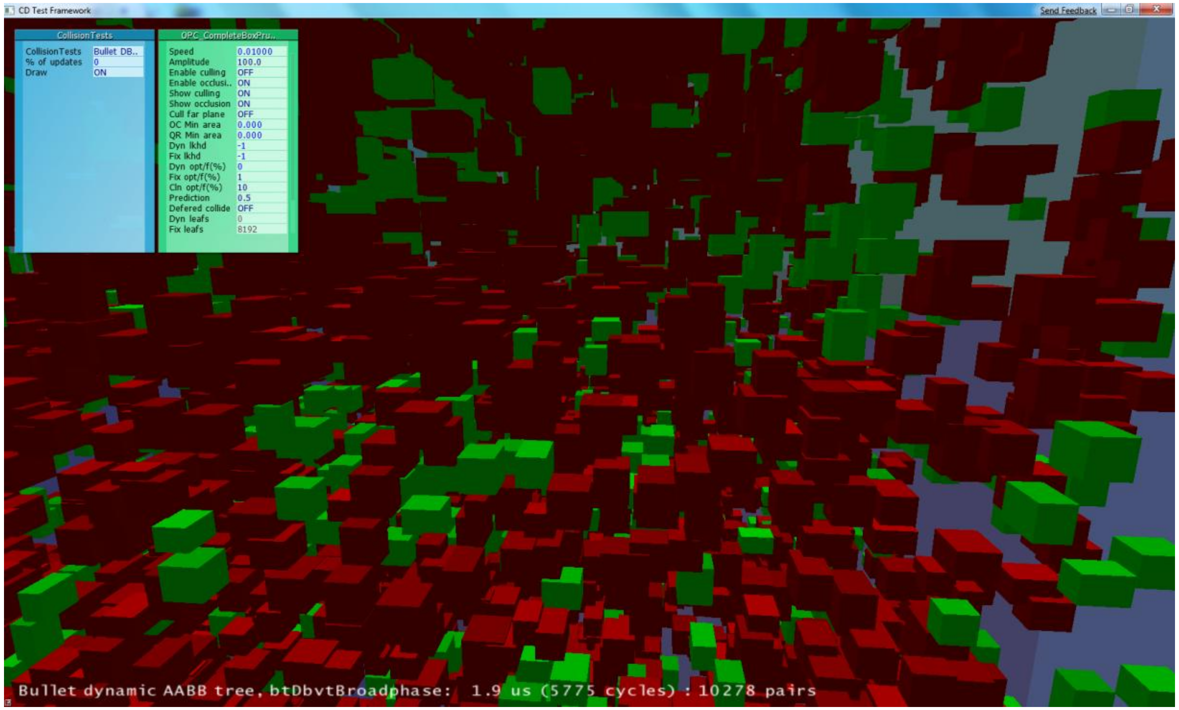
- Input: world space BVs and unique IDs
- Output: array of potential overlapping pairs

Decompose pairs

- Need to store previous applied impulse per contact point: match new and existing points

- Output:
  - Newly added pairs
  - Old existing pairs
  - Removed pairs

Broadphase Benchmark



```
CollisionTests    Bullet DB..
% of updates      0
Draw              ON
```

```
Speed             0.01000
Amplitude         100.0
Enable culling    OFF
Enable occlusi..  ON
Show culling      ON
Show occlusion    ON
Cull far plane    OFF
OC Min area       0.000
QR Min area       0.000
Dyn lkhd          -1
Fix lkhd          -1
Dyn opt/f(%)      0
Fix opt/f(%)      1
Cln opt/f(%)      10
Prediction        0.5
Defered collide   OFF
Dyn leafs         0
Fix leafs         8192
```

```
Bullet dynamic AABB tree, btDbvtBroadphase:  1.9 us (5775 cycles) : 10278 pairs
```

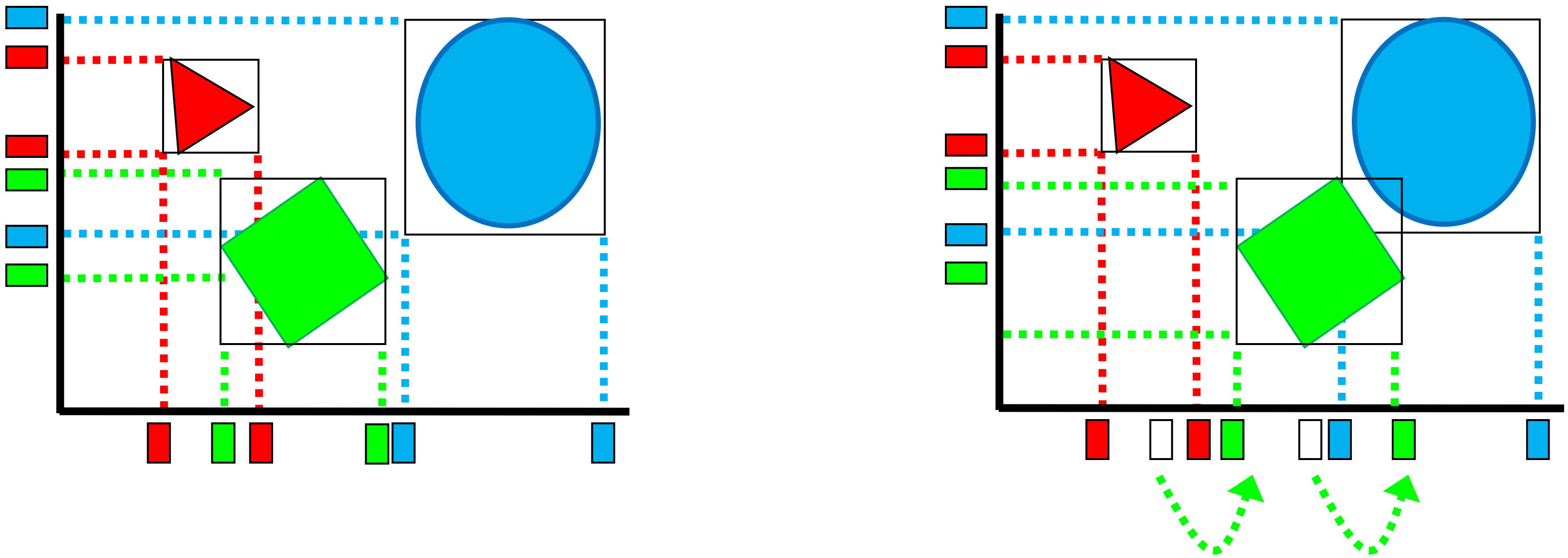
- See Bullet/Extras/CDTestFramework

Broadphase N-body solutions

- Sweep and prune (SAP)
- Uniform Grid, Hierarchical Grid
- Dynamic BVH tree

Incremental sweep and prune

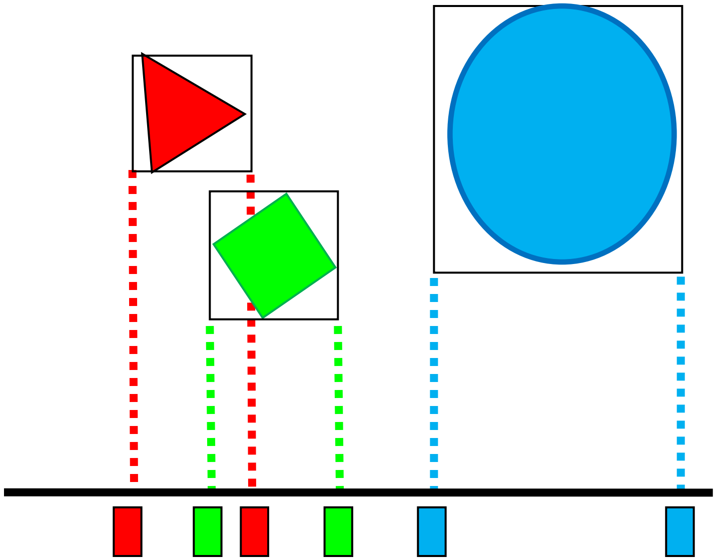
- Update 3 sorted axis and overlapping pairs



- Performs best if most objects hardly move
- Difficult to parallelize

Parallel 1 axis sweep and prune

- From scratch sort 1 axis sweep to find all pairs

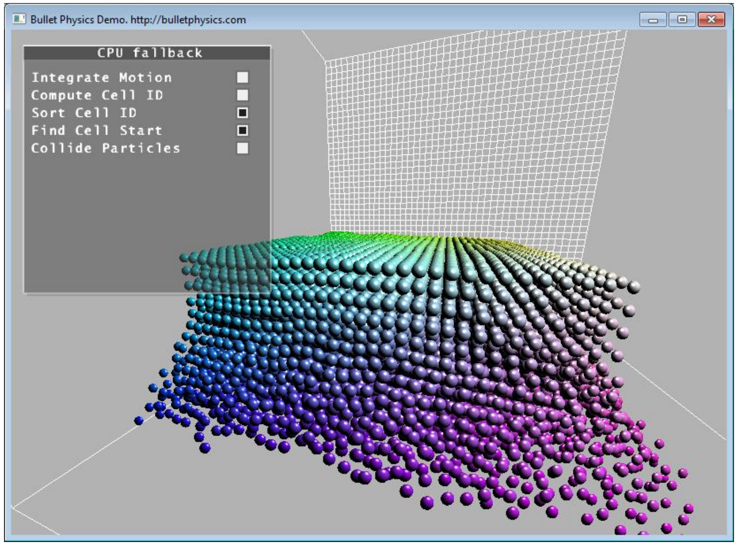


- Parallel bitonic sort and sweep in parallel

[Game Physics Pearls, 2010, AK Peters]

Uniform and hierarchical grid

- Very GPU friendly, parallel radix or bitonic sort
- Use modulo to make grid unbounded
- Use a hierarchy of grids to allow varying sizes



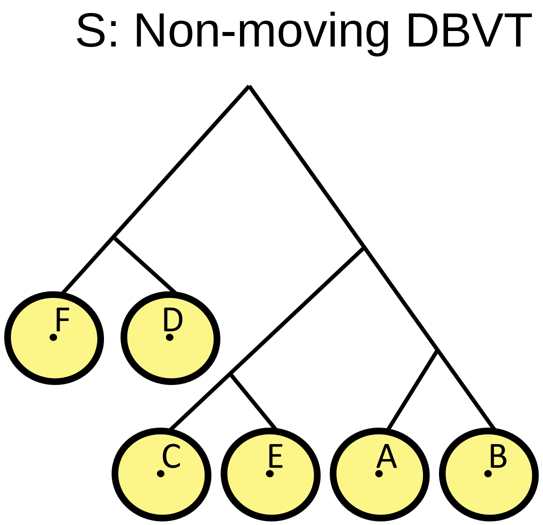
Bullet Physics Demo. http://bulletphysics.com

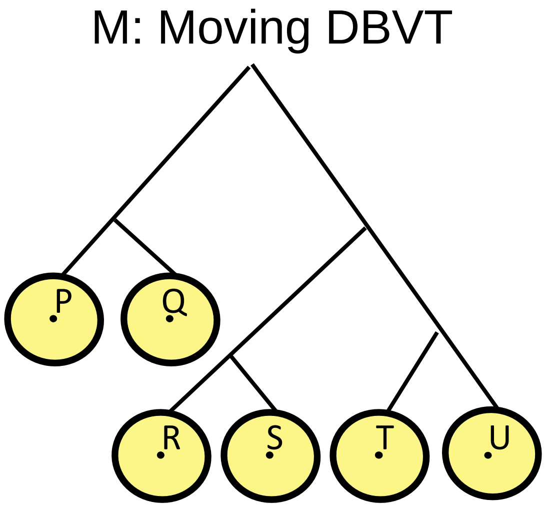
CPU fallback

Integrate Motion □
Compute Cell ID □
Sort Cell ID ▣
Find Cell Start ▣
Collide Particles □

Dynamic BVH tree broadphase

- Keep two dynamic trees, one for moving objects, other for objects (sleeping/static)
- Find neighbor pairs:
  - Overlap M versus M and Overlap M versus S



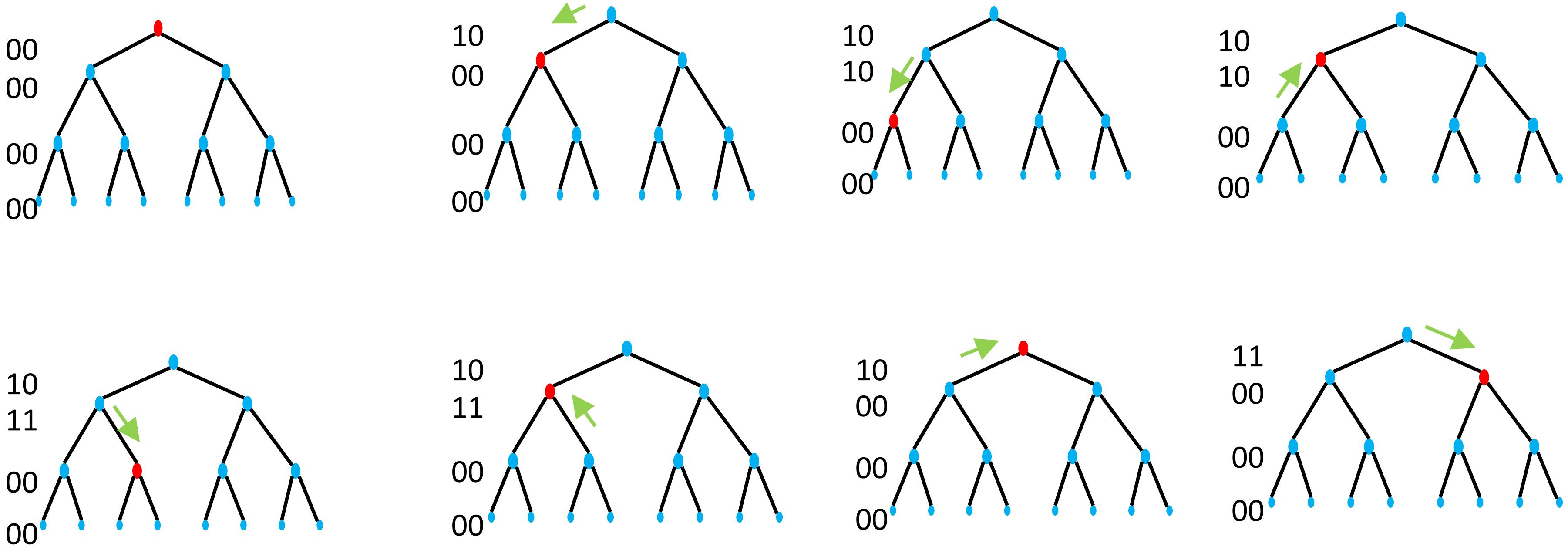


Update/move a leaf node

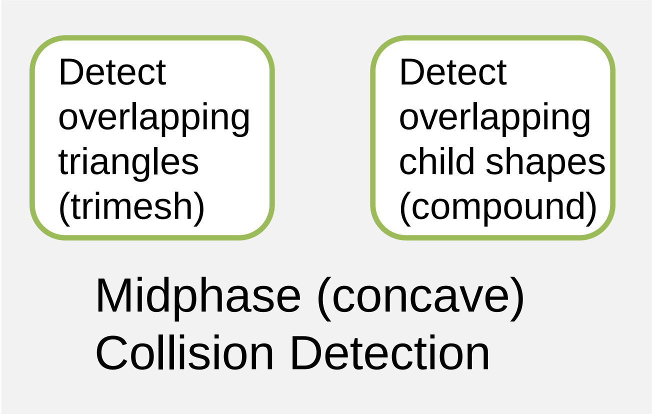
- If new AABB is contained by old do nothing
- Otherwise remove and re-insert leaf
  - Re-insert at closest ancestor that was not resized during remove
- Expand AABB with margin
  - Avoid updates due to jitter or small random motion
- Expand AABB with velocity
  - Handle the case of linear motion over n frames

Parallel BVH tree traversal

- Incremental update on CPU (shared memory)
- Use parallel history traversal on GPU



Midphase culling



Detect overlapping triangles (trimesh)

Detect overlapping child shapes (compound)

Midphase (concave) Collision Detection

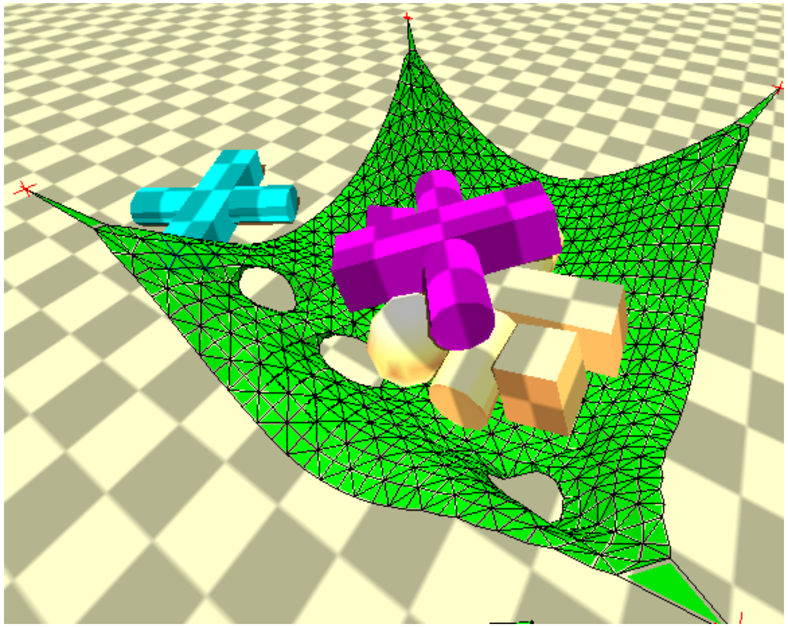
BVH with fixed topology

- Quantized nodes, 16 bytes
- Cache-friendly memory layout suitable for SPU
- Supports tree refit, stackless traversal
- Implemented in btOptimizedBVH

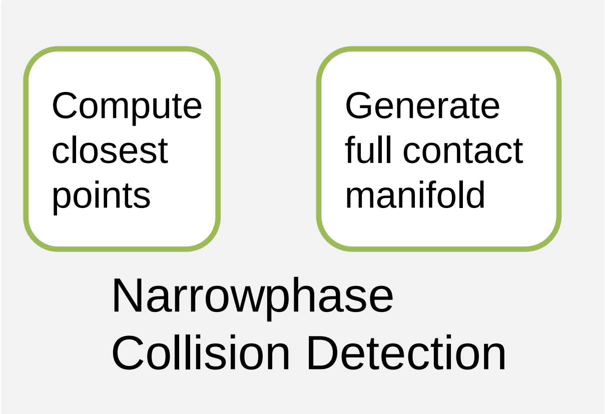
[PGSS07] Stackless KD-Tree Traversal for High Performance GPU Ray Tracing, POPOV S., GÜNTHER J., SEIDEL H.-P., SLUSALLEK P., Eurographics 2007

Dynamic BVH, topology change

- Same acceleration structure as broadphase
- Accelerate fracture, tearing, deformation



Narrowphase elementary queries

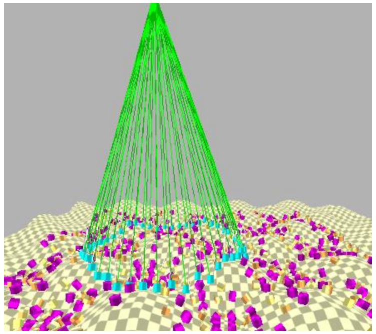


Compute closest points

Generate full contact manifold

Narrowphase Collision Detection

Continuous Queries



- Input:
  - Shapes, in-between motion
- Output:
  - Time of impact fraction, hit normal, point

Bullet uses Conservative Advancement for CCD

Examples:

- Ray test, swept convex query

Discrete Collision Queries

- Input:
  - Collision shapes, world transforms
- Output:
  - Shortest distance, witness points, normal

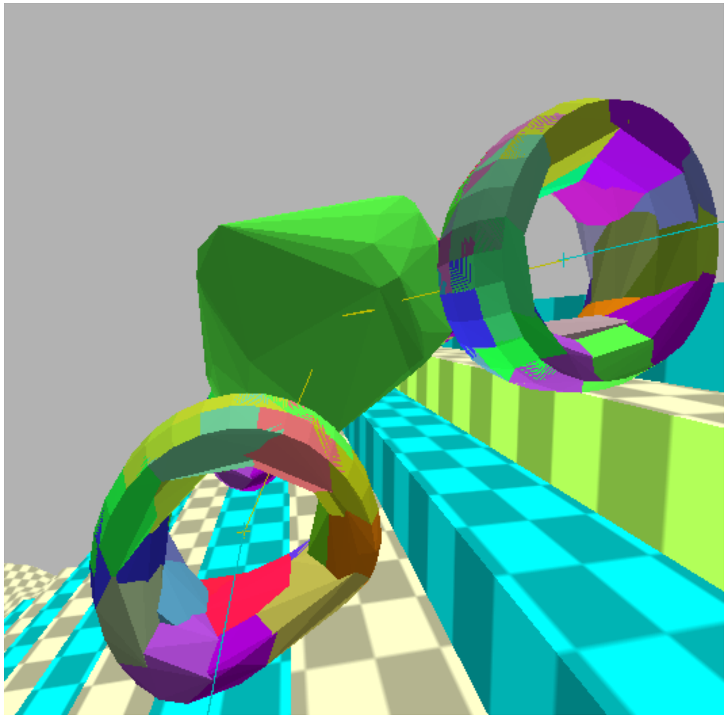
GJK, SAT, EPA or special algorithms

- Expanding polytope algorithm for PD
- Pairwise tests can be trivially dispatched in parallel

[Collision Detection in Interactive Environments, 2003, Gino van den Bergen, Morgan Kaufmann]

Using GJK for deformable objects

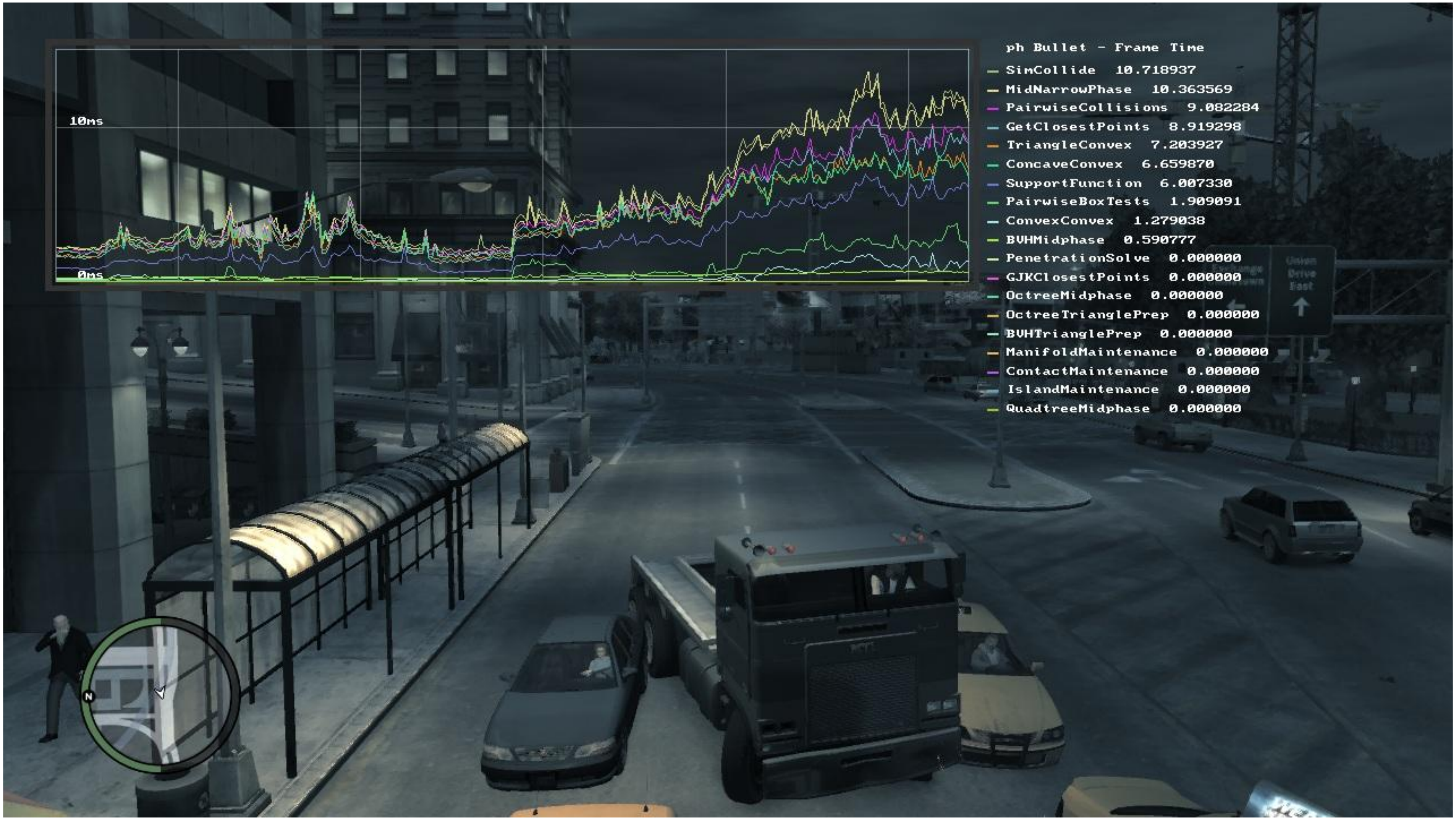
- Triangle-triangle test (Discrete or CCD-CA)
- Convex clusters instead of triangles



GPU GJK collision detection

- GJK convex collision detection fits current GPU
- EPA penetration depth harder to port to GPU
  - Larger code size, dynamic data structures
- Instead of EPA, sample penetration depth
  - Using support mapping
- Sample support map using GPU cube mapping

Cubemap GJK test in GTA IV



```
ph Bullet - Frame Time
— SimCollide  10.718937
— MidNarrowPhase  10.363569
— PairwiseCollisions  9.082284
— GetClosestPoints  8.919298
— TriangleConvex  7.203927
— ConcaveConvex  6.659870
— SupportFunction  6.007330
— PairwiseBoxTests  1.909091
— ConvexConvex  1.279038
— BVHMidphase  0.590777
— PenetrationSolve  0.000000
— GJKClosestPoints  0.000000
— OctreeMidphase  0.000000
— OctreeTrianglePrep  0.000000
— BVHTrianglePrep  0.000000
— ManifoldMaintenance  0.000000
— ContactMaintenance  0.000000
— IslandMaintenance  0.000000
— QuadtreeMidphase  0.000000
```

Summary

- SPU and GPU are great to accelerate CD queries
- Dynamic AABB trees are fast and versatile acceleration structure for
  - broadphase pair search, ray test, CCD
  - midphase for triangle meshes, cloth, deformables
  - fracturing or tearing
  - occlusion and view frustum culling
- In combination with GJK and EPA it provides a complete collision detection pipeline

Thank You

Questions

erwin_coumans@playstation.sony.com

http://bulletphysics.org

http://bullet.googlecode.com