GPU rigid body simulation using OpenCL

Introduction

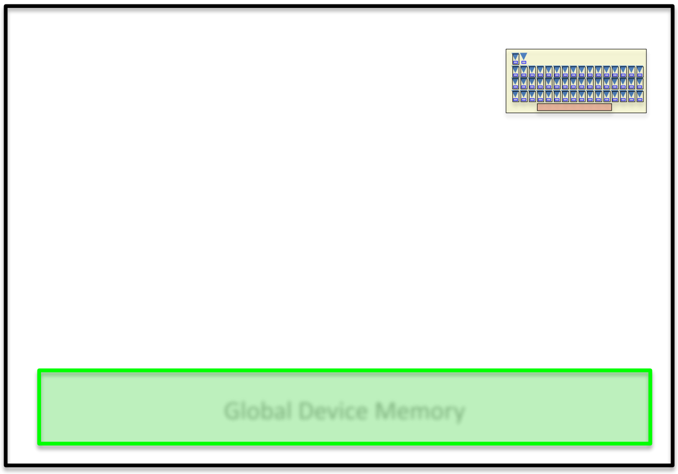
Bullet 2.x Refactoring

-
-
-
-
-

Bullet 3.x Full Rewrite

Getting started with OpenCL

OpenCL terminology



Our first OpenCL kernel

```
typedef struct
{
        float4 m_pos;
        float4 m_linVel;
} Body;

void   integrateTransforms (Body* bodies, int nodeID, float timeStep)
{
        for (int nodeID = 0;nodeID<numBodies;nodeID++)
        {
                if( bodies[nodeID].m_invMass != 0.f)
                {
                        bodies[nodeID].m_pos +=  bodies[nodeID].m_linVel * timeStep;
                }
        }
}
```

```
__kernel void   integrateTransformsKernel( __global Body* bodies,const int numNodes, float
timeStep)
{
        int nodeID = get_global_id(0);
        if( nodeID < numNodes && (bodies[nodeID].m_invMass != 0.f))
        {
                bodies[nodeID].m_pos +=  bodies[nodeID].m_linVel * timeStep;
        }
}
```

Porting existing code to OpenCL

Replace C++ by C

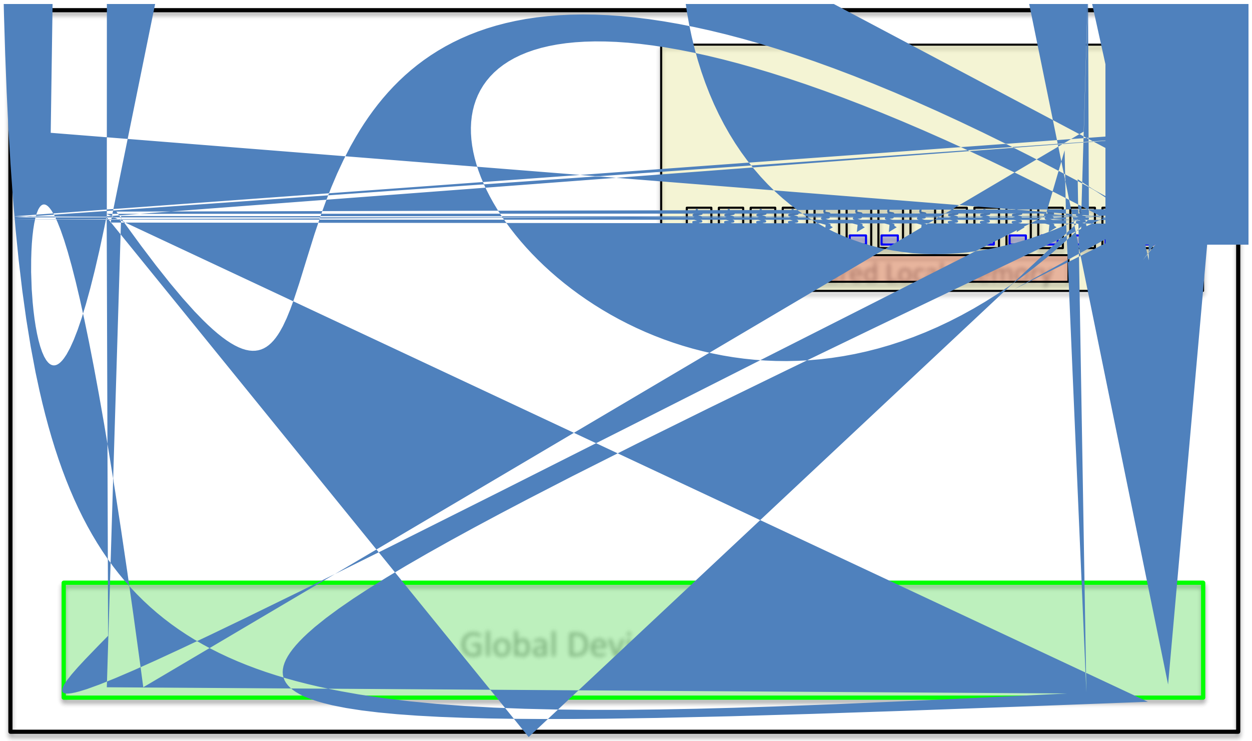
Move data to contiguous memory

Replace pointers by indices

```
struct btTransform
{
       btMatrix3x3    m_basis;
       btVector3      m_position;
};
class btRigidBody : public btCollisionObject
{
       btMatrix3x     m_inverseInertiaWorld;
       btVector3      m_linearVelocity;
       btVector3      m_angularVelocity;
       btScalar       m_mass;
       . . .
};
class btCollisionObject
{
       btTransform       m_worldTransform;
       btCollisionShape* m_collisionShape;
       . . .
};
```

```
struct b3RigidBody
{
       b3Vector3      m_position;
       b3Quaternion   m_orientation;
       int            m_collidableIndex;
       . . .
};
struct b3Collidable
{
       int m_shapeType;
       int m_shapeIndex;
};
```

Exploiting GPU hardware



Dealing with branchy code/thread divergence

```
__kernel void   branchyKernel (. . .)
{
        if (conditionA)
        {
                        someCodeA(. . .);
        } else
        {
                        someCodeNotA(. . .);
        }
}
```

Use Parallel Primitives

-
-
-

Use Local Memory

-
-
-

Barrier synchronization

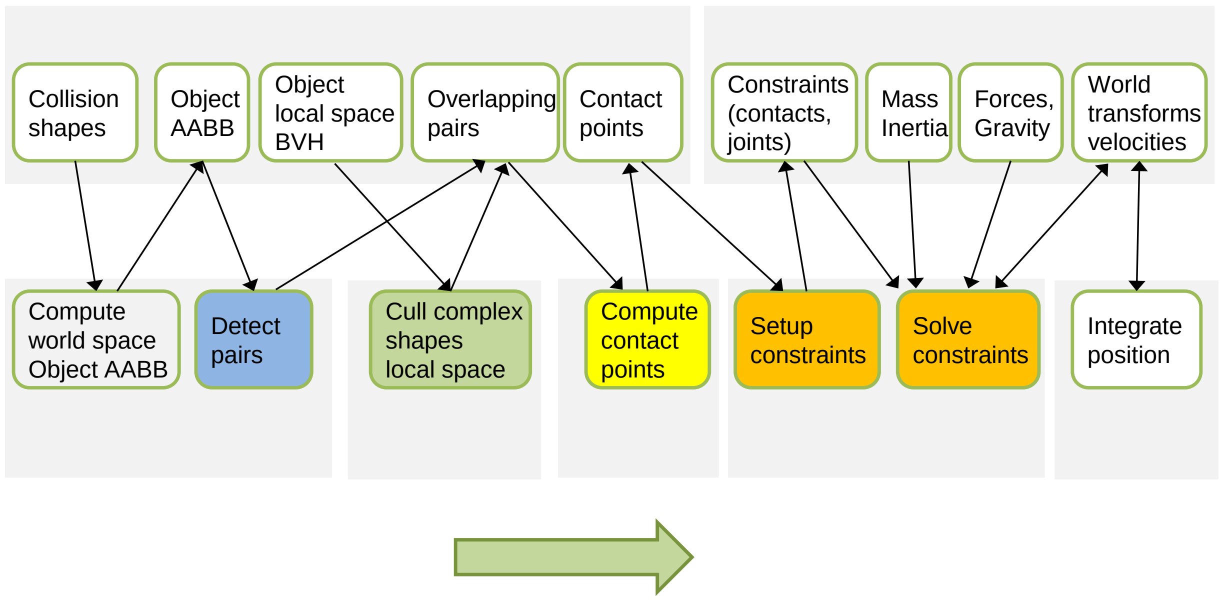
-
-
-

Atomics

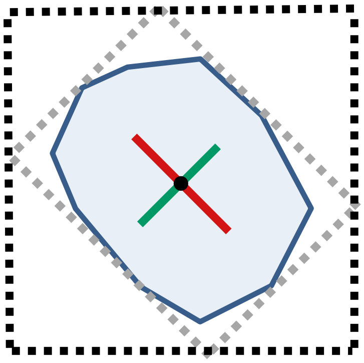
GPU rigid body simulation

Rigid body introduction

The rigid body pipeline



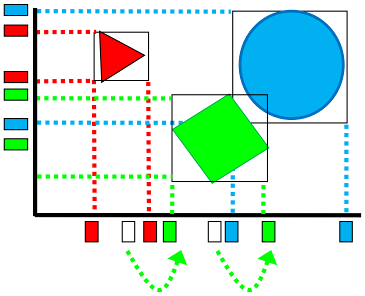
Computing the object AABBs

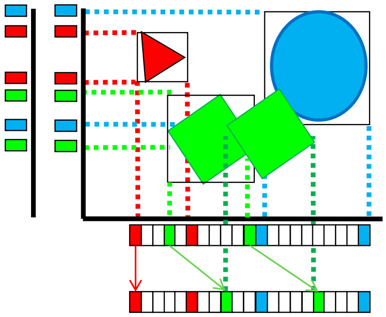


GPU overlapping pair detection

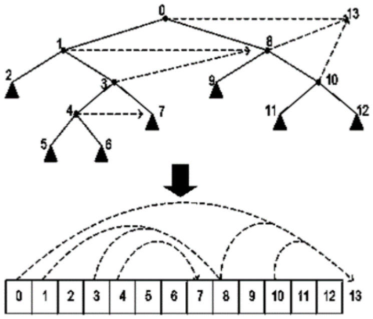
```
void   computePairsKernelBruteForce (const btAabbCL* aabbs, volatile __global int2*
pairsOut,volatile  __global int* pairCount, int numObjects, int maxPairs)
{
   for (int i=0;i<numObjects;i++)
   {
        for (int j=i+1;j<numObjects;j++)
        {
               if (TestAabbAgainstAabb2GlobalGlobal(&aabbs[i],&aabbs[j]))
               {
                       int2 myPair;
                       myPair.x = aabbs[i].m_objectIndex;
                       myPair.y = aabbs[j].m_objectIndex;
                       int curPair = *pairCount;
                       if (curPair<maxPairs)
                       {
                              pairsOut[curPair] = myPair; //flush to main memory
                              pairCount++;
                       }
               }
        }
   }
}
```

```
__kernel void   computePairsKernelBruteForceKernel ( __global const btAabbCL* aabbs, volatile
__global int2* pairsOut,volatile  __global int* pairCount, int numObjects, int maxPairs)
{
        int i = get_global_id(0);
        if (i>=numObjects)
               return;
        for (int j=i+1;j<numObjects;j++)
        {
               if (TestAabbAgainstAabb2GlobalGlobal(&aabbs[i],&aabbs[j]))
               {
                       int2 myPair;
                       myPair.x = aabbs[i].m_objectIndex;
                       myPair.y = aabbs[j].m_objectIndex;
                       int curPair = atomic_inc (pairCount);
                       if (curPair<maxPairs)
                       {
                              pairsOut[curPair] = myPair; //flush to main memory
                       }
               }
        }
}
```

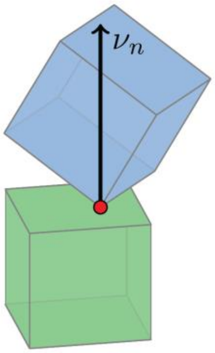




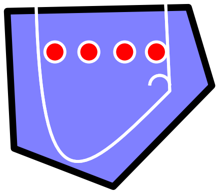
GPU local space BVH culling for complex shapes



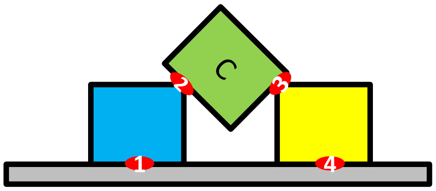
GPU contact computation



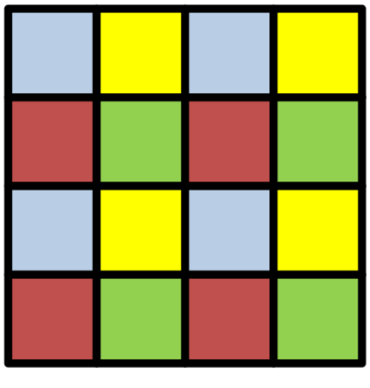
```
void   computeContactsSAT(convexHullA,convexHullB,transformA,transformB)
{
        b3Vector3 satAxis;
        if (findSeparatingAxis(convexHullA,convexHullB,transformA,transformB))
        {
                if (clipHullHull(satAxis, convexHullA,convexHullB,transformA,transformB))
                {
                        findClippingFacesKernel(. . .)
                        clipFacesAndContact(. . .)
                        contactReduction(...)
                }
        }
}
```



GPU parallel contact solving



```
void   batchConstraints(constraints, int numConstraints)
{
        int batchIdx=0;
        while( numValidConstraints < numConstraints)
        {
                clear(bodiesUsed);
                for(int i=numValidConstraints; i<numConstraints; i++)
                {
                        int bodyA = constraints[i].m_bodyA;
                        int bodyB = constraints [i].m_bodyB;
                        if (isAvailable(bodyA,bodyB,bodiesUsed))
                        {
                                markUsed(bodyA,bodyB,bodiesUsed);
                                constraint[i].m_batchId = batchIdx;//assign the batch index
                        }
                }
                batchIdx ++;
        }
}
```

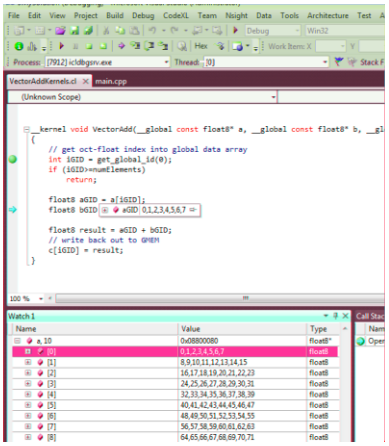


GPU parallel joint solving

Debugging and Performance Profiling

Debug on the CPU

Intel OpenCL debugger



File Edit View Project Build Debug CodeXL Team Nsight Data Tools Architecture Test
Debug
Win32
Hex
Work Item: X
Y
Process: [7912] icldbgsrv.exe
Thread: [0]
Stack F
VectorAddKernels.cl
main.cpp
(Unknown Scope)
100 %
Call Stac
Nam
Oper

```
__kernel void VectorAdd(__global const float8* a, __global const float8* b, __gl
{
    // get oct-float index into global data array
    int iGID = get_global_id(0);
    if (iGID>=numElements)
        return;

    float8 aGID = a[iGID];
    float8 bGID ⊞ aGID 0,1,2,3,4,5,6,7

    float8 result = aGID + bGID;
    // write back out to GMEM
    c[iGID] = result;
}
```

Watch 1

Name	Value	Type
⊟ a, 10	0x08800080	float8*
⊞ [0]	0,1,2,3,4,5,6,7	float8
⊞ [1]	8,9,10,11,12,13,14,15	float8
⊞ [2]	16,17,18,19,20,21,22,23	float8
⊞ [3]	24,25,26,27,28,29,30,31	float8
⊞ [4]	32,33,34,35,36,37,38,39	float8
⊞ [5]	40,41,42,43,44,45,46,47	float8
⊞ [6]	48,49,50,51,52,53,54,55	float8
⊞ [7]	56,57,58,59,60,61,62,63	float8
⊞ [8]	64,65,66,67,68,69,70,71	float8

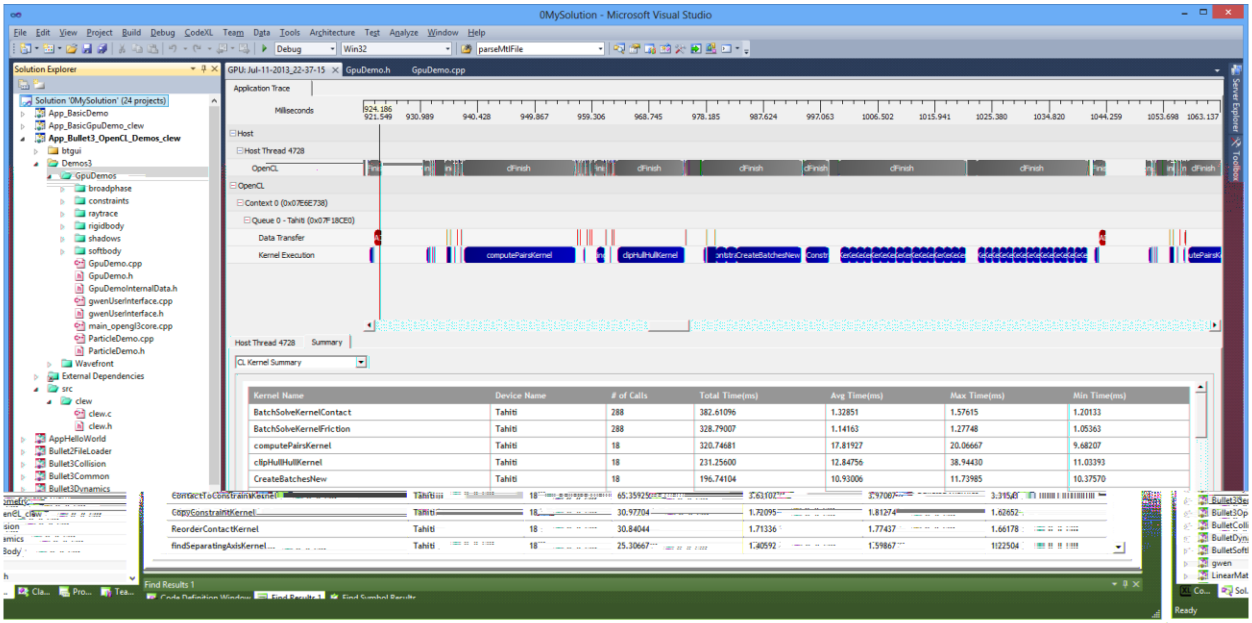
printf debugging

Debug buffers

Debugging a frozen system

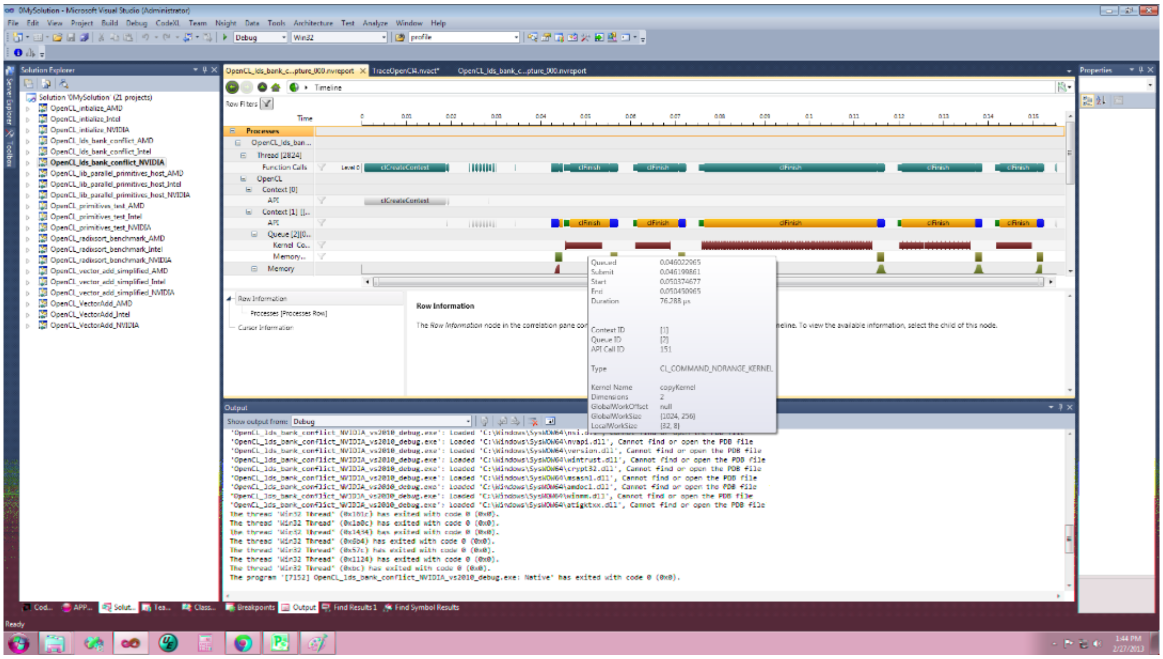
Profile Zones

CodeXL Performance Profiler



Kernel Name	Device Name	# of Calls	Total Time(ms)	Avg Time(ms)	Max Time(ms)	Min Time(ms)
BatchSolveKernelContact	Tahiti	288	382.61096	1.32851	1.57615	1.20133
BatchSolveKernelFriction	Tahiti	288	328.79007	1.14163	1.27748	1.05363
computePairsKernel	Tahiti	18	320.74681	17.81927	20.06667	9.68207
clipHullHullKernel	Tahiti	18	231.25600	12.84756	38.94430	11.03393
CreateBatchesNew	Tahiti	18	196.74104	10.93006	11.73985	10.37570
ContactToConstraintKernel	Tahiti	18	65.35925	3.63102	3.97007	3.31548
CopyConstraintKernel	Tahiti	18	30.97704	1.72095	1.81274	1.62652
ReorderContactKernel	Tahiti	18	30.84044	1.71336	1.77437	1.66178
findSeparatingAxisKernel...	Tahiti	18	25.30667	1.40592	1.59867	1.22504

NVIDIA NSIGHT Profiler



Row Information

The *Row Information* node in the correlation pane con... ...meline. To view the available information, select the child of this node.

Queued	0.046022965
Submit	0.046199861
Start	0.050374677
End	0.050450965
Duration	76.288 µs
Context ID	[1]
Queue ID	[2]
API Call ID	151
Type	CL_COMMAND_NDRANGE_KERNEL
Kernel Name	copyKernel
Dimensions	2
GlobalWorkOffset	null
GlobalWorkSize	[1024, 256]
LocalWorkSize	[32, 8]

Show output from: Debug

```
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\nsi.d
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\nvapi.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\version.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\wintrust.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\crypt32.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\msasn1.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\amdocl.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\winmm.dll', Cannot find or open the PDB file
'OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe': Loaded 'C:\Windows\SysWOW64\atigktxx.dll', Cannot find or open the PDB file
The thread 'Win32 Thread' (0x101c) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0x1a0c) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0x1434) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0x6b4) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0x57c) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0x1124) has exited with code 0 (0x0).
The thread 'Win32 Thread' (0xbc) has exited with code 0 (0x0).
The program '[7152] OpenCL_lds_bank_conflict_NVIDIA_vs2010_debug.exe: Native' has exited with code 0 (0x0).
```

OpenCL Tips and Tricks

Create your own OpenCL wrapper

Dynamically load OpenCL

Cache the precompiled OpenCL kernel binaries

Keep a host implementation of your kernel

Unit test an OpenCL kernel

References

Appendix A: Bullet 3.x Source code

Requirements

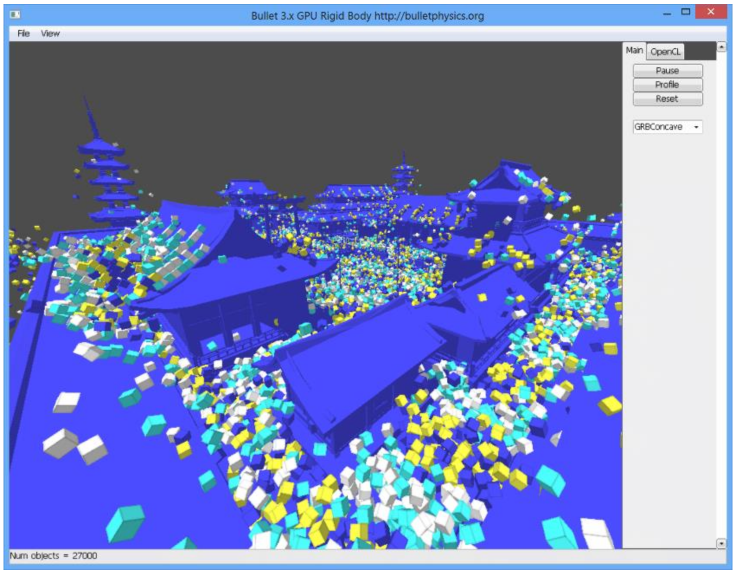
Building on Windows using Visual Studio

Building on Linux (or Mac OSX) using gcc

```
cd build3
./premake_linux64 gmake
cd gmake
make
```

Building on Max OSX using XCode

Usage



Num objects = 27000

Benchmark

Feedback