GPU TECHNOLOGY CONFERENCE

Scalable Multi Agent Simulation on the GPU

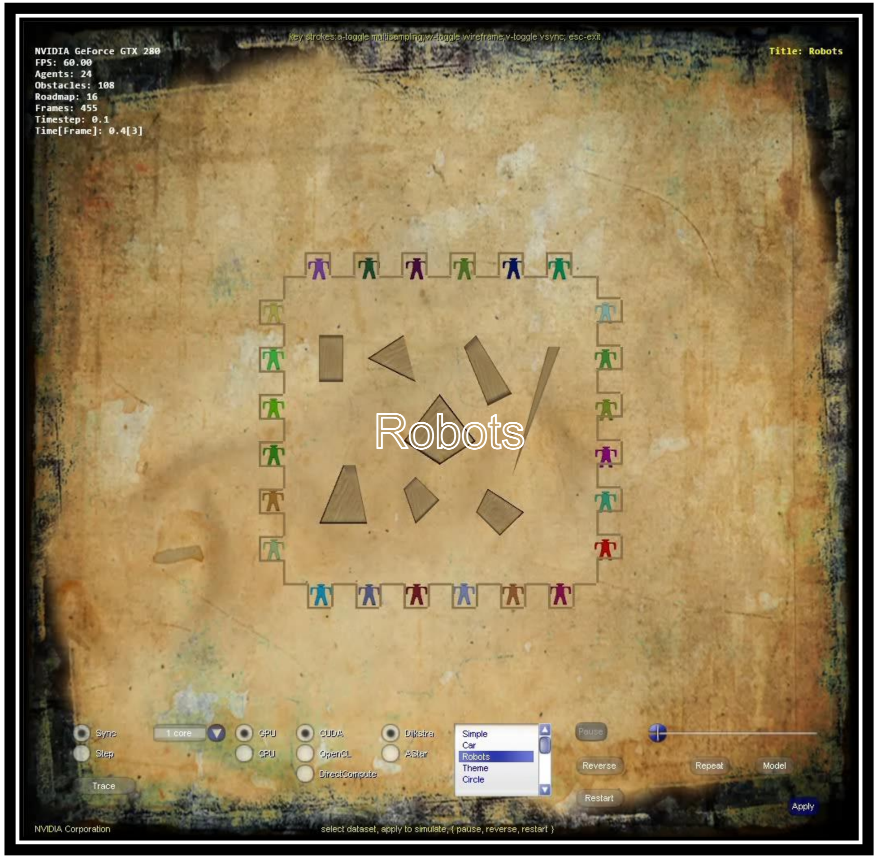
Avi Bleiweiss
NVIDIA Corporation

San Jose, 2009

NVIDIA.

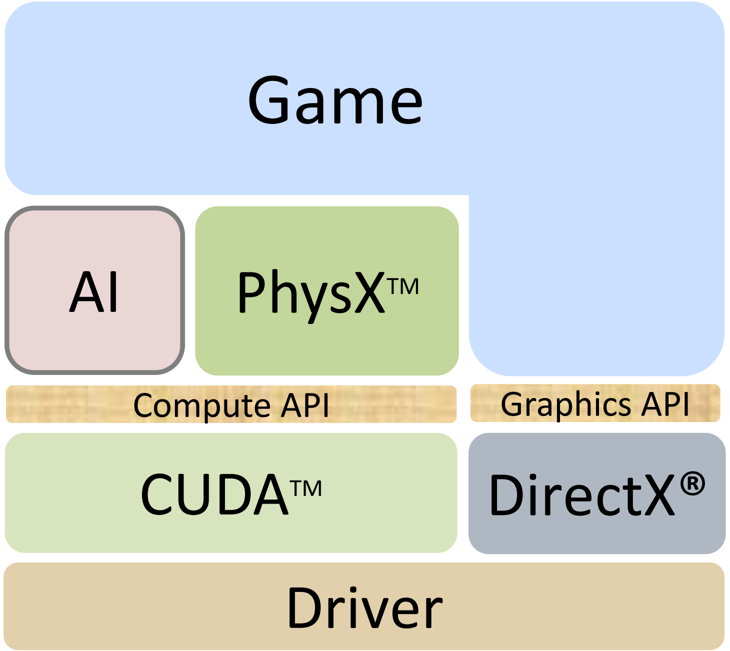
Reasoning

- Explicit
  - State machine, serial
- Implicit
  - Compute intensive
  - Fits SIMT well
- Collision avoidance



Motivation

- Games re-designed
  - Casual, bursty
- Scalable, real time
  - Dense environments
- Joint framework
  - AI, physics simulation



Problem

Planner

- Efficient mesh to roadmap conversion.
- Searches a global, optimal path
  - From start to goal
- Locally, avoids collisions with
  - Static, dynamic objects

Simulator

- Visually compelling motion
- Economical memory footprint
- A subset of compute units
- Linear scale with # characters

Solution

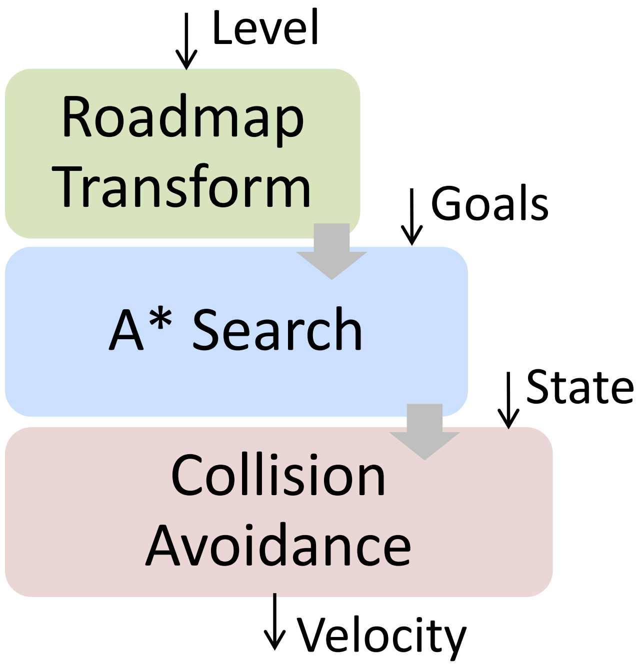
- Small, quality roadmap
- Heterogeneous agents
- Velocity Obstacles
- GPU specific optimizations
  - Spatial hash
  - Nested parallel

Outline

- Algorithm
- Implementation
- Results
- Integration
- Takeaways

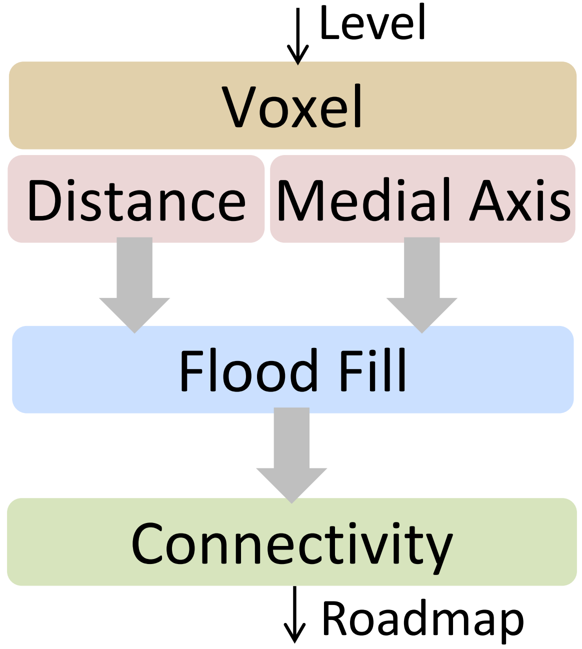
Pipeline

- Game level mesh input
- Inline computed roadmap
- Goals, roadmap decoupled
- Discrete time simulation
- Intuitive multi threading



Roadmap Transform

- Reachability roadmap[1]
- An existed C_{free} path
  - Guaranteed in roadmap
- Predictable termination
- Highly parallelizable
  - 3D grid operators

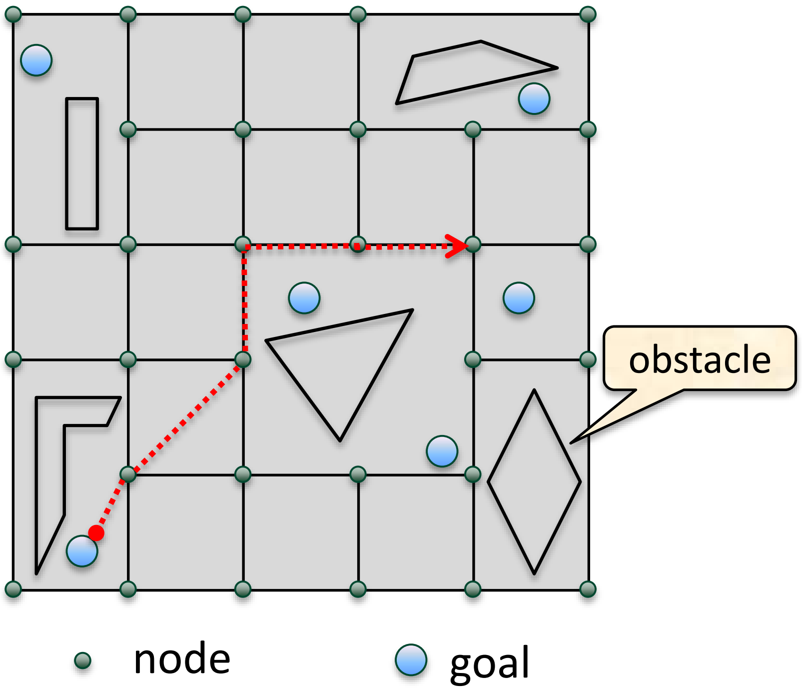


1 [Geraerts and Overmars 2005]

GPU TECHNOLOGY CONFERENCE

Visibility

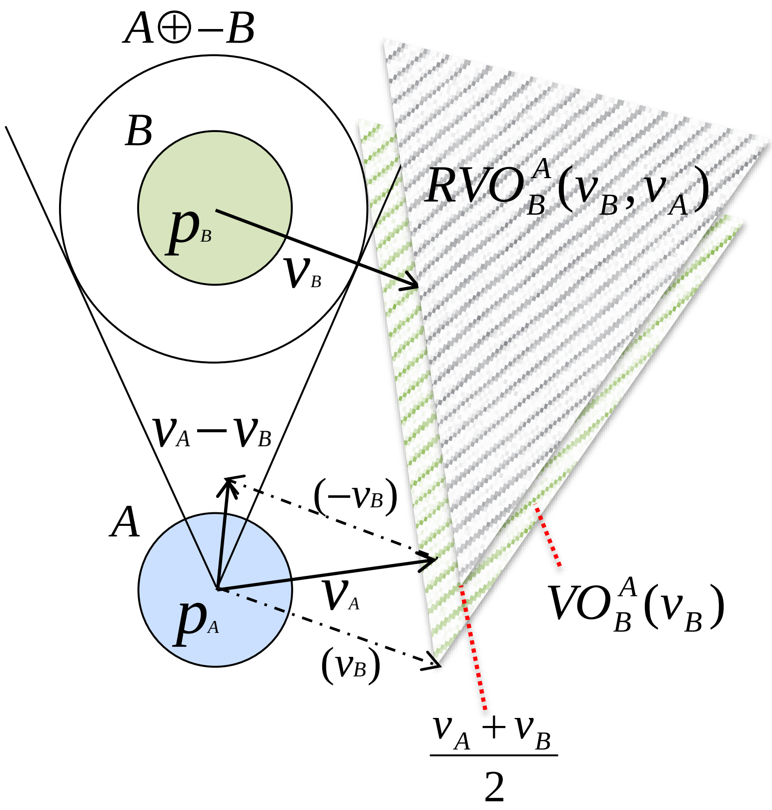
- Two sets of edges
  - Visible roadmap node pairs
  - Goals to unblocked nodes
- Static obstacles outline
- A* search, shortest path
  - From goal to any node



NVIDIA.

Velocity Obstacles

- Well defined, widely used
- Avoidance velocity set[1]
- Reciprocal Velocity Obstacles[2]
  - Oscillation free motion
- Agents moving in 2D plane



1 [Fiorini and Shiller 1998]
2 [Van Den Berg et al. 2008]

GPU TECHNOLOGY CONFERENCE

Simulation

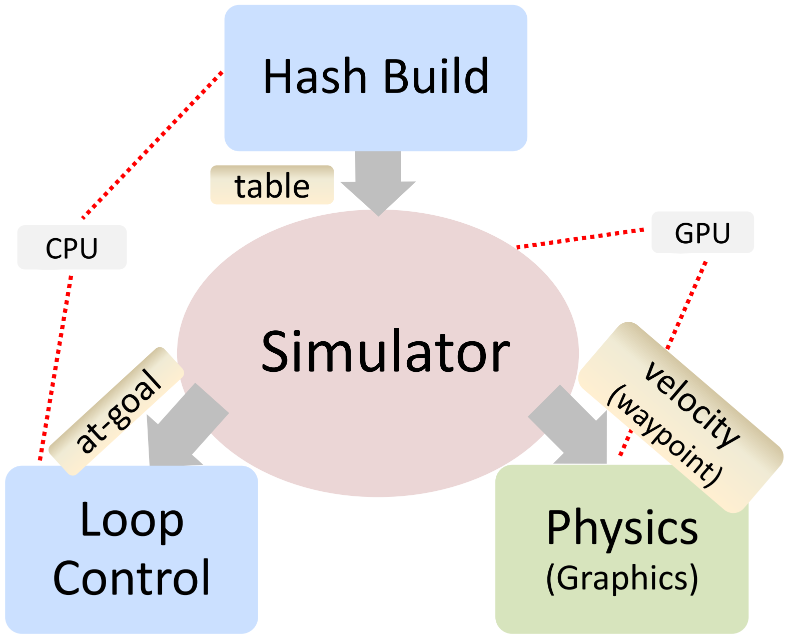
- Simulator advances until
  - All agents reached goal
- Path realigned towards
  - Roadmap node or goal
- Agent, velocity parallel

```
do
  hash
    construct hash table
  simulate
    compute preferred velocity
    compute proximity scope
    foreach velocity sample do
      foreach neighbor do
nested    if OBSTACLE then VO
          elseif AGENT then RVO
      resolve new velocity
  update
    update position, velocity
    resolve at-goal
while not all-at-goal    flat
```


NVIDIA.

Workflow

- Deterministic resources
  - Linear, pitched 3D
- Roadmap static for
  - Multiple simulation steps
- Dozen compute kernels
- Split frame, multi GPU



Collision Avoidance Frame Process

Challenges

Hiding memory latency

Divergent, irregular threads

Small agent count (≤32)

Hash construction cost

Distance Transform

- Input, binary grid of n^3 cells
- Squared Euclidian distance
- Serial running time $O(n^3)$
- Parallel linear time $O(n)$
  - Slice, column, row passes
  - n^2 GPU threads, per pass

[Felzenszwalb and Huttenlocher 1996]

0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	1	1	1	0	0
0	1	1	1	1	1	0
0	1	1	0	1	1	0
0	0	1	1	1	1	0
0	0	0	0	0	0	0

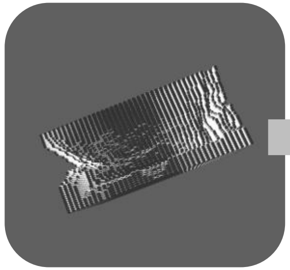
0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	4	4	4	0	0
0	4	9	1	9	1	0
0	1	4	0	4	4	0
0	0	1	1	1	1	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	4	4	1	0	0
0	1	4	1	4	1	0
0	1	1	0	1	1	0
0	0	1	1	1	1	0
0	0	0	0	0	0	0

$$DT_f(p) = \min((p-q)^2 + f(q))$$

Medial Axis Transform

- Input, binary grid of n^3 cells
- Serial running time $O(kn^3)$
- n^3 GPU threads, per pass
  - – $O(k)$ time for CDT
  - – $O(1)$ for qualifier T
  - – $O(1)$ for resolve



Chess Distance Transform → Qualify → Resolve

$$MAT(i,j,k) = \min\{\max(|i-x|,|j-y|,|k-z|)\}$$
$$i \le x \le N, j \le y \le N, k \le z \le N$$

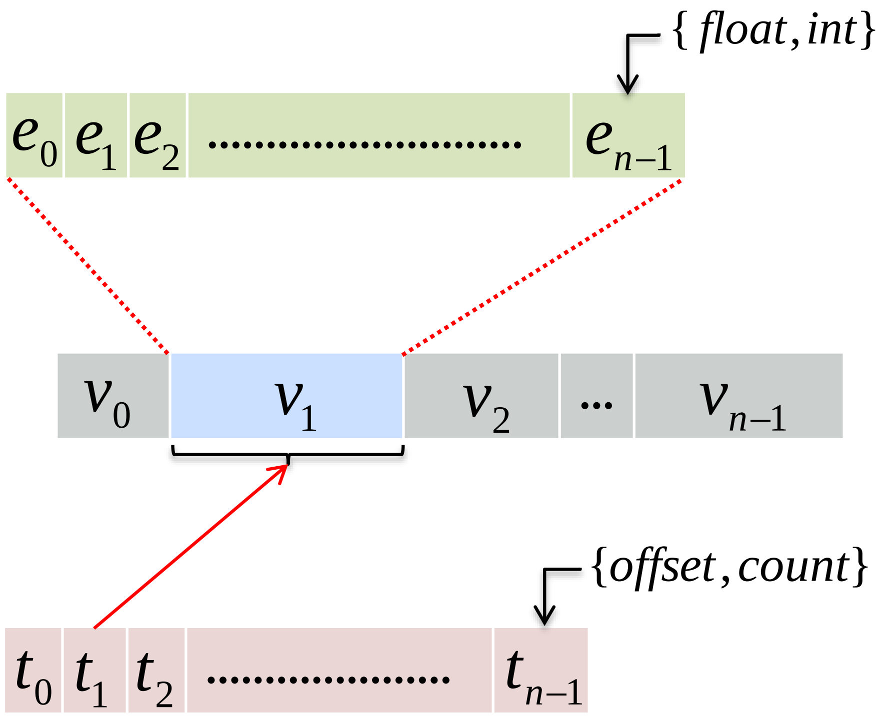
$$T[i,j,k] = \max\{MAT(x,y,z)\} \le MAT(i,j,k)$$
$$i-1 \le x \le i, j-1 \le y \le j, k-1 \le z \le k$$
$$!(x == i \;\&\&\; y == j \;\&\&\; z == k)$$

[Lee and Horng 1996]

GPU TECHNOLOGY CONFERENCE

Data Layout

- Persistent resources
  - Reside in global memory
- Thread aligned data
  - Better coalescing
- Consistent access pattern
  - Improves bandwidth

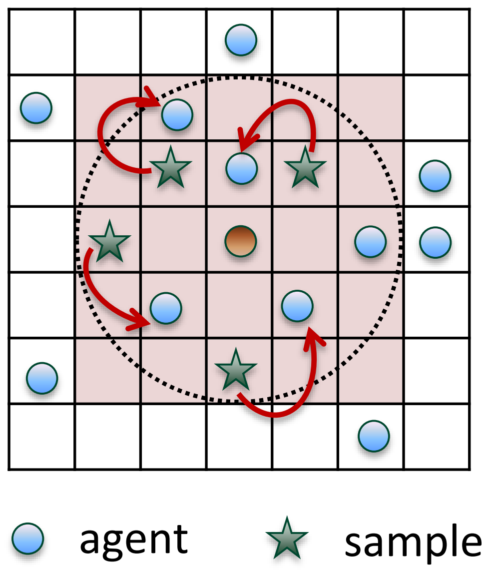


Variable Length Vector Access

NVIDIA.

K-Nearest Neighbor

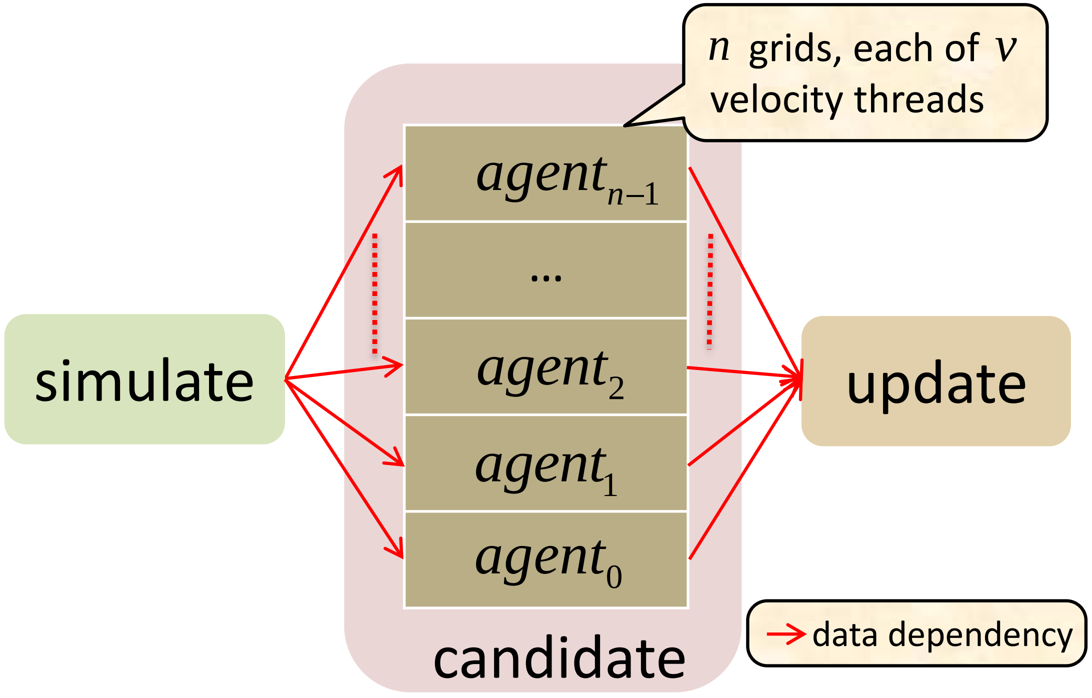
- Naïve, exhaustive search
  - $O(n^2)$ system running time
- Spatial hash
  - 3D point to a 1D index
- Per frame table build
  - Current agents' position



$$h(p) = \mathrm{determinant}(p, p_{ref})$$

Nested Parallel

- Flat parallel limiting
- Thread grid DAG
  - Independent grids
  - Same kernel per level
- Thread amplification
  - Improved occupancy



Thread Grid DAG

GPU TECHNOLOGY CONFERENCE

Velocity Threads

- Hundreds of threads
- Graceful grid sync
- Fine reduce-min
  - Into Shared memory
- Global atomic CAS
  - Inter thread block

```
__global__ void
candidate(CUAgent* agents,
          int index,
          CUNeighbor* neighbors)
{
   float3 v, float t;
   CUAgent a = agents[index];
   if(!getThreadId()) v = a.prefvelocity;
   else v = velocitySample(a);
   t = neighbor(a, agents, neighbors, v);
   float p = penalty(a, v, t);
   reduceMinAtomicCAS(a, p);
   if(p == a.minpenalty)  a.candidate = v;
}
```

sync

GPU TECHNOLOGY CONFERENCE

Methodology

Environment

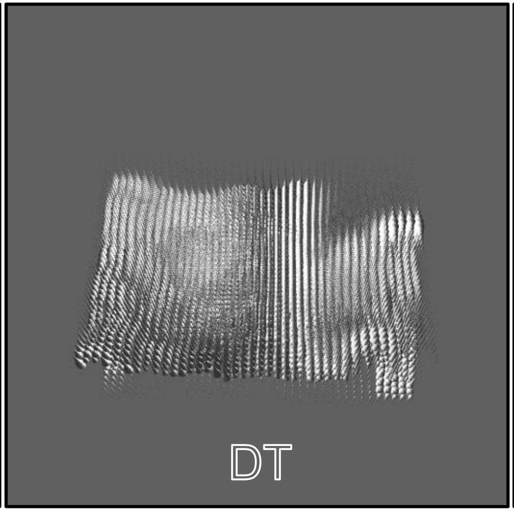
- Vista 32
- CUDA 2.3, OpenMP
- Simulation-only
- Flat and nested parallel
- Copy to/from device included

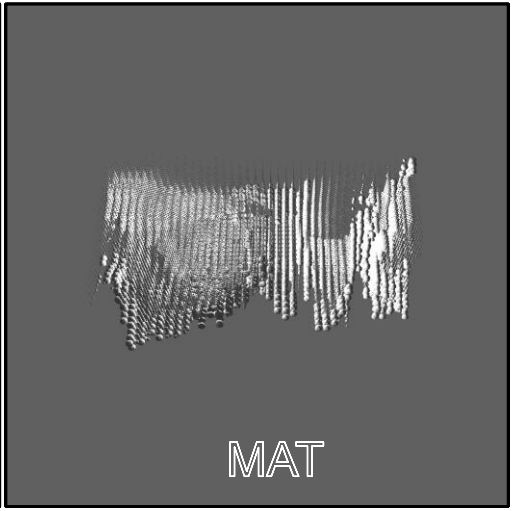
Property	NVIDIA GTX280	Intel	
		i7-940	X7350
Core Clock (MHz)	601	2942	2930
Memory Clock (MHz)	1107	3*1066	1066
Global Memory (MB)	1024	3072	8192
Compute Units	30	8	4
Threads/Compute Unit	512	1	4

NVIDIA.

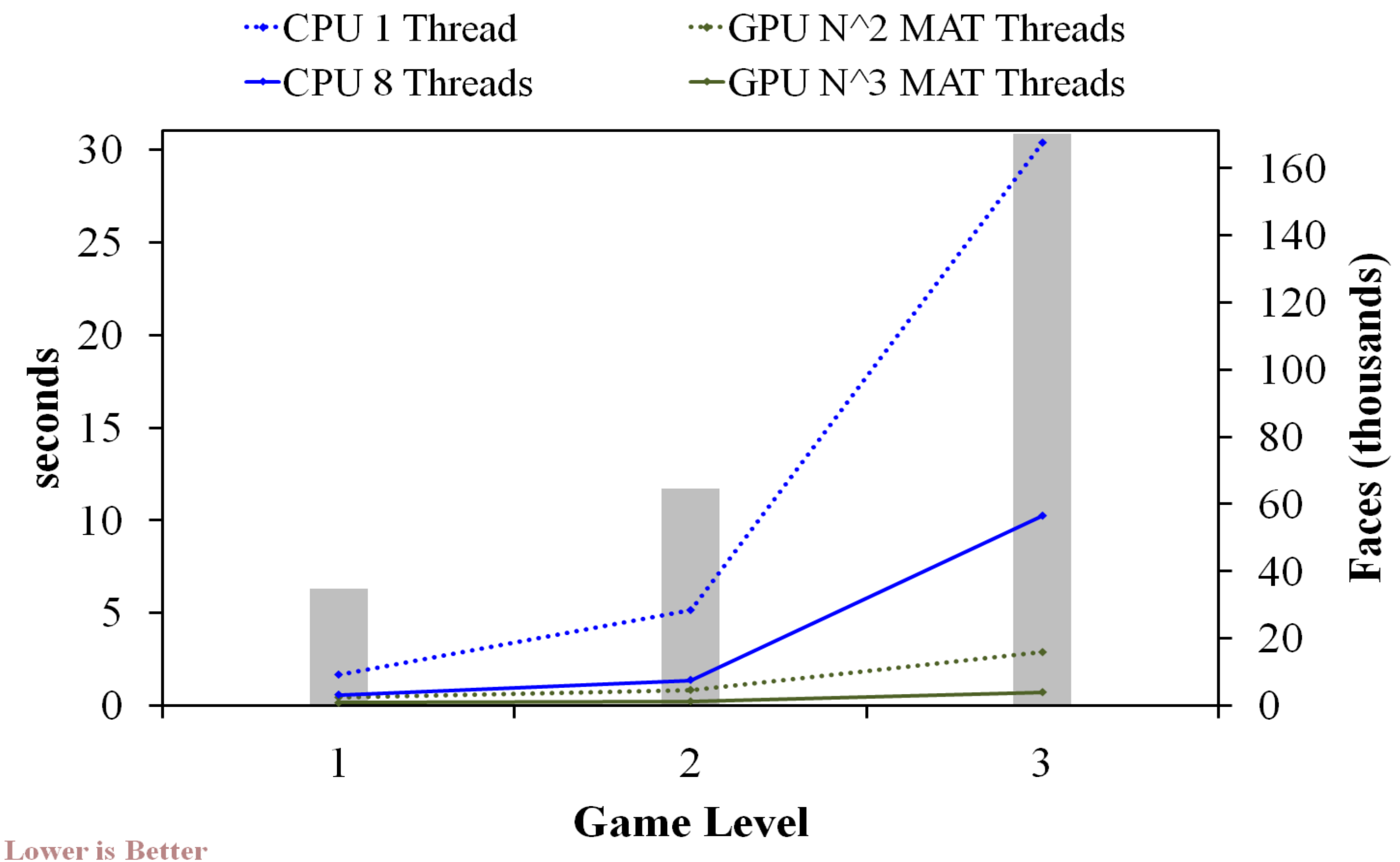
Experiments I

Game Level		Grid Resolution	GPU Threads	
Vertices	Faces		Distance	Medial Axis
82800	34750	33	1089	35937
161463	64451	40	1600	64000
347223	170173	55	3025	166375





Roadmap Transform



Lower is Better

GPU TECHNOLOGY CONFERENCE

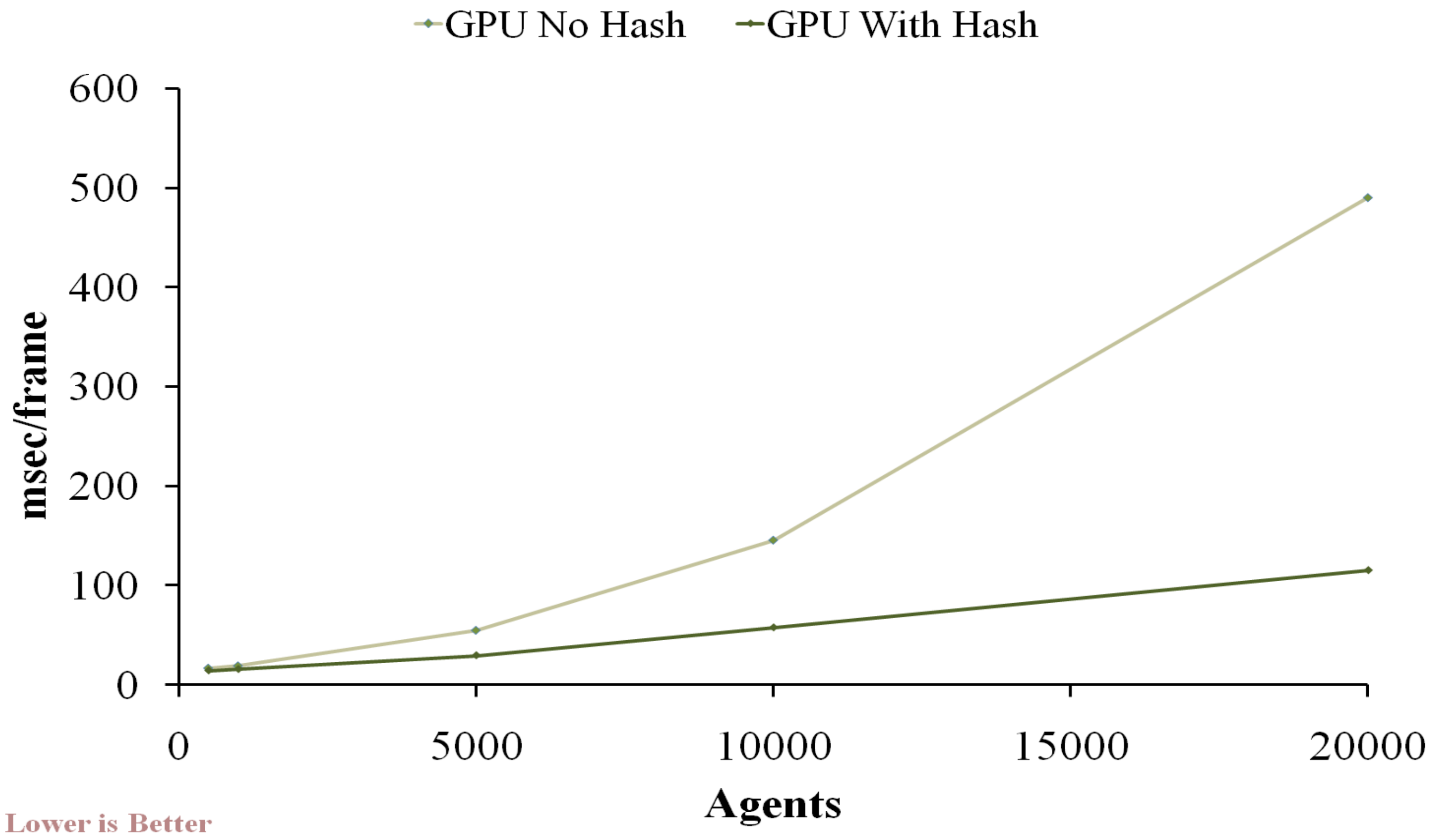
Experiments II

Timestep	Proximity		Velocity Samples	Frames
	Neighbors	Distance		
0.1	10	15	250	1200

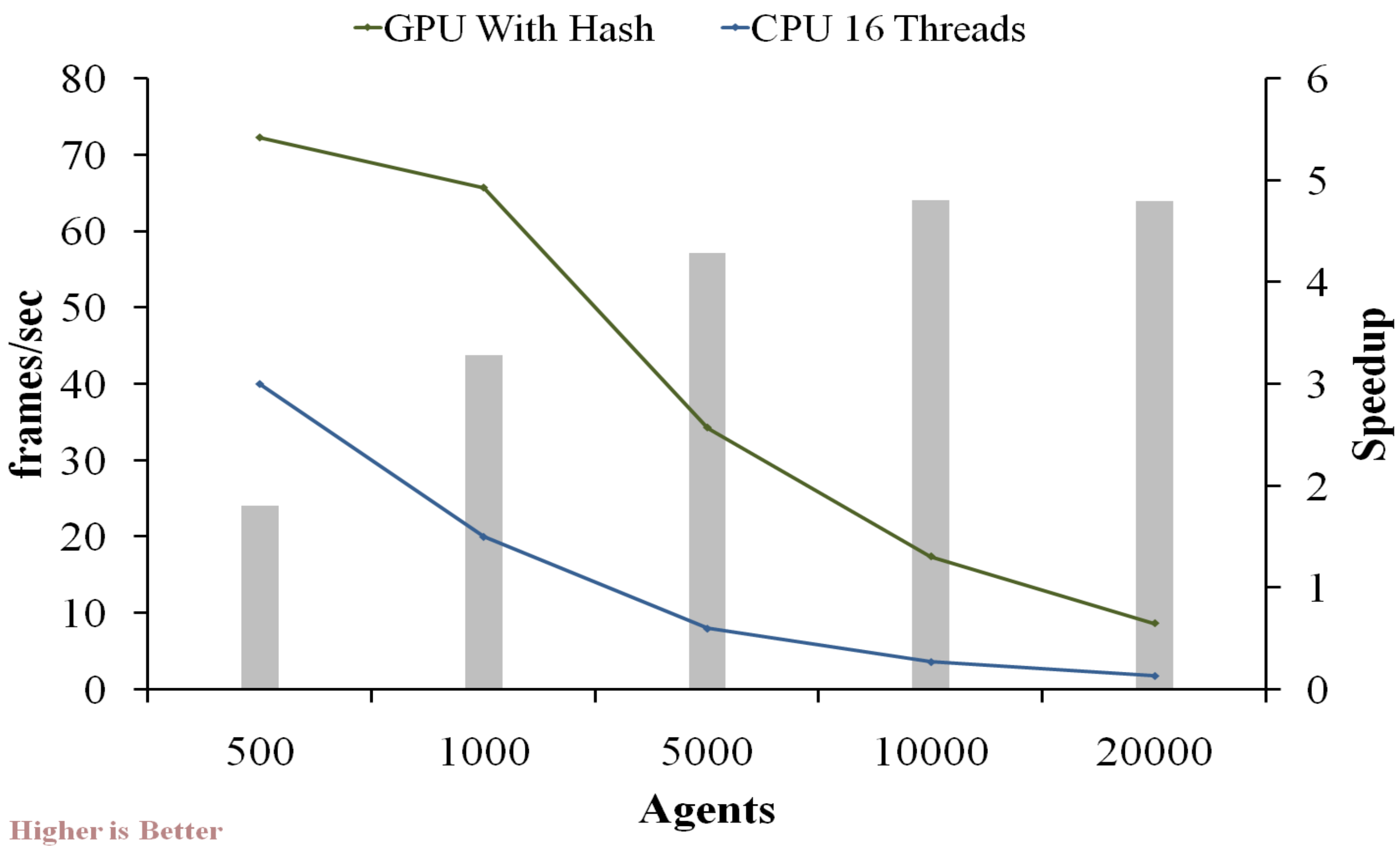
Dataset	Segments	Roadmap Nodes	Agents	Compute Units
Evacuation	211	429	500—20000	1—40

NVIDIA.

Spatial Hash



Frame Rate



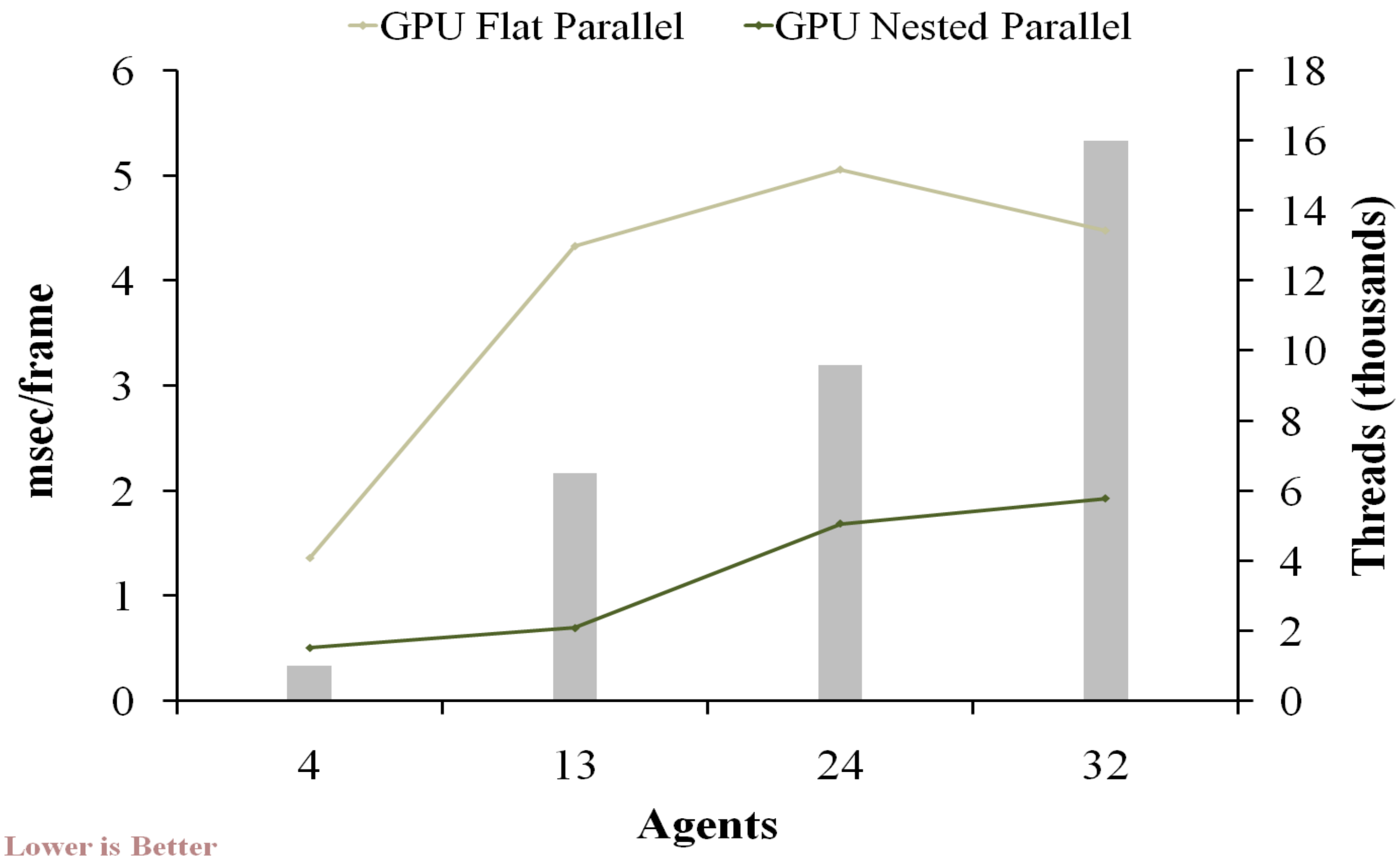
Higher is Better

Experiments III

Dataset	Agents	Velocity Samples	Velocity Threads	Frames
Simple	4	250	1000	607
Car	13	500	6500	228
Robots	24	400	9600	455
Circle	32	500	16000[1]	596

1 Exceeds max GPU threads: 15360

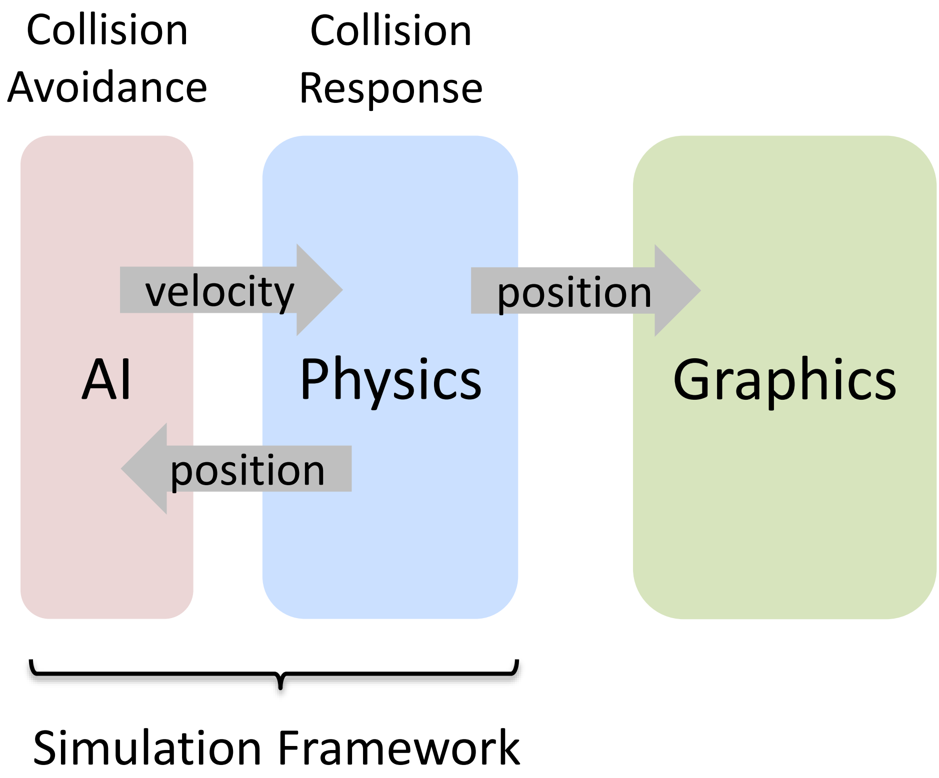
Nested Parallel



Lower is Better

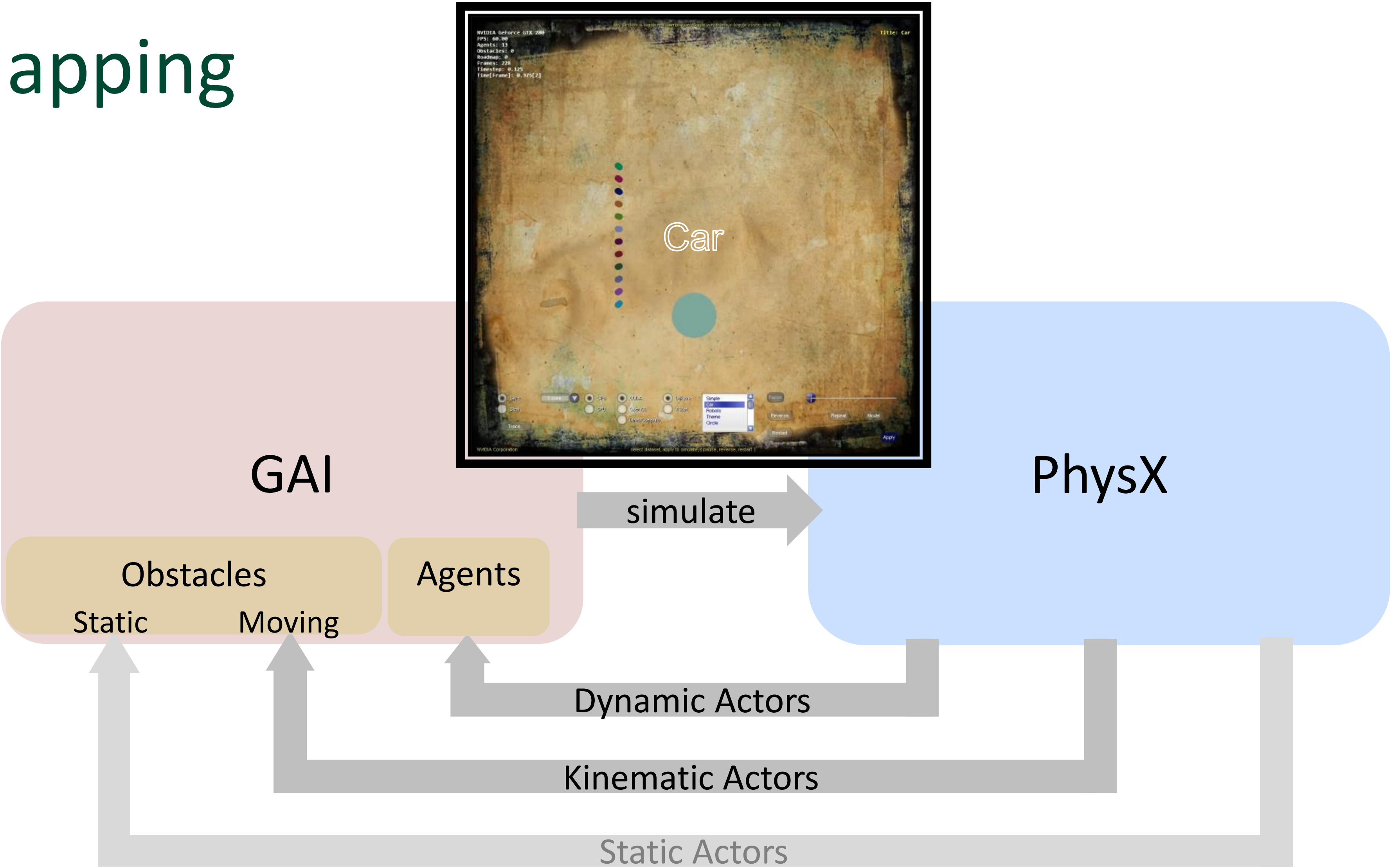
Interaction

- AI, physics interop
  - Compute context pair
  - One active at a time
  - Single GPU formation
- Compute, graphics
  - Single interop port



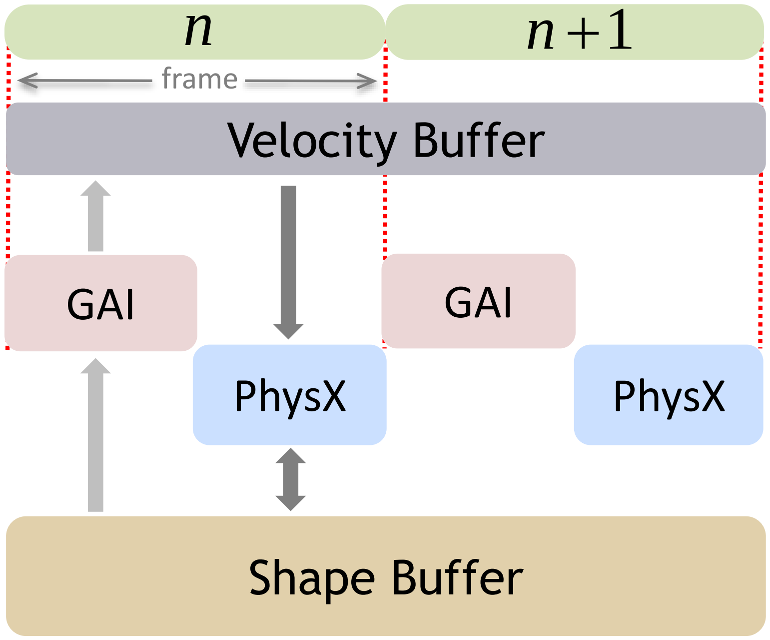
GPU TECHNOLOGY CONFERENCE

Mapping



Compute Interop

- Shared Actor buffers
  - Shape, velocity
  - In global memory
- AI controls simulation
- Interleaved frame
  - Implicit synchronization



Limitations

- Small, SW thread cache
- Not fully 3D aware
- Hash table build
  - Single threaded
- Thread load imbalance
  - Non, at-goal agent mix

Performance

Parameter		NVIDIA GTX280	Intel CPU
Roadmap Transform	Speedup	Up to 14.5X	i7-940 8 Threads
	Running Time (sec)	0.7 @170K face level	
Collision Avoidance	Hash vs. Naive	Up to 4X	Little to no effect[1]
	Nested vs. Flat	Up to 6.2X	Not easy to program
	Simulation Speedup	Up to 4.8X	X7350 16 Threads[1]
	Simulation FPS	18 @ 10K agents	

1 Van Den Berg et al., RVOLib 2008

Future Work

- Fermi architecture scale
- Full 3D collision avoidance
- Parallel hash build
- Up hash sampling quality
- Evaluate UNC's FVO

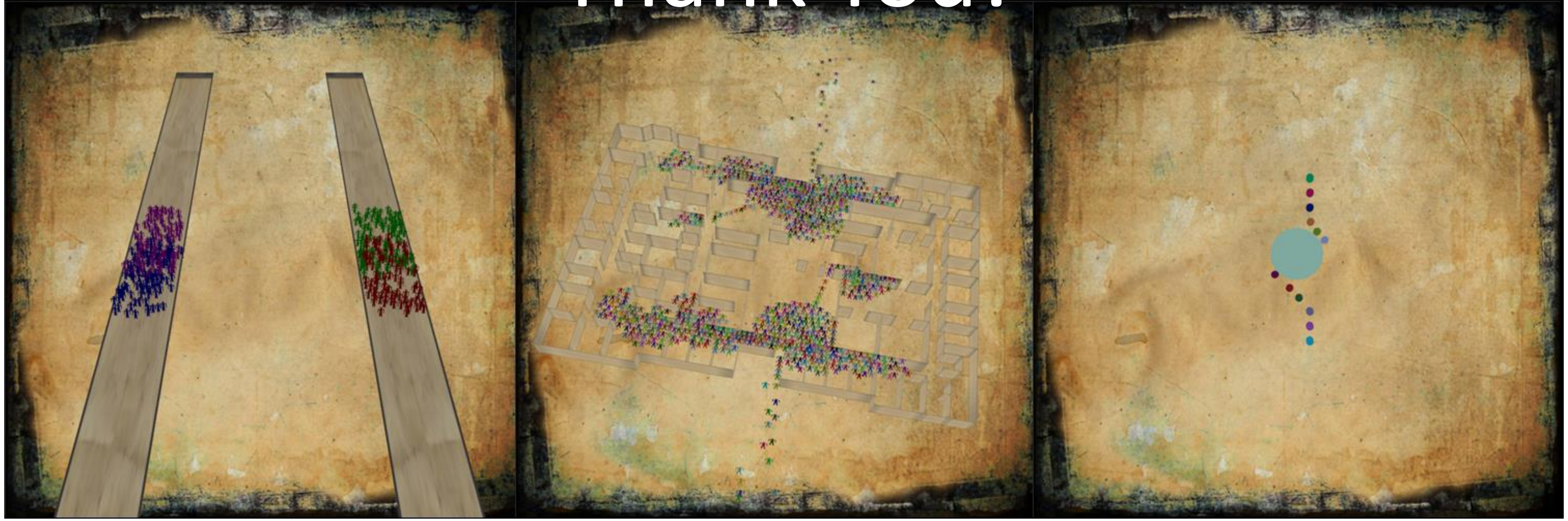
Property	GT200	Fermi
Threads / SM (32 regs)	512	1024
L1 Cache	None	48K
L2 Cache	None	768KB

Summary

- Dynamic roadmap, goals
- Multi agent solution
  - Compact, scalable
  - Fermi optimizations
- Nested parallel potential
- AI, physics integration

GPU TECHNOLOGY CONFERENCE

Thank You!



NVIDIA.

Info

- White Paper: http://tinyurl.com/MultiAgentGPU-paper-2009
- Videos: http://developer.nvidia.com/object/gdc-2009.html

- CUDA: http://www.nvidia.com/cuda
- OpenCL: http://www.nvidia.com/object/cuda_opencl.html
- DirectCompute:http://www.nvidia.com/object/directcompute.html
- Nexus: http://developer.nvidia.com/object/nexus.html