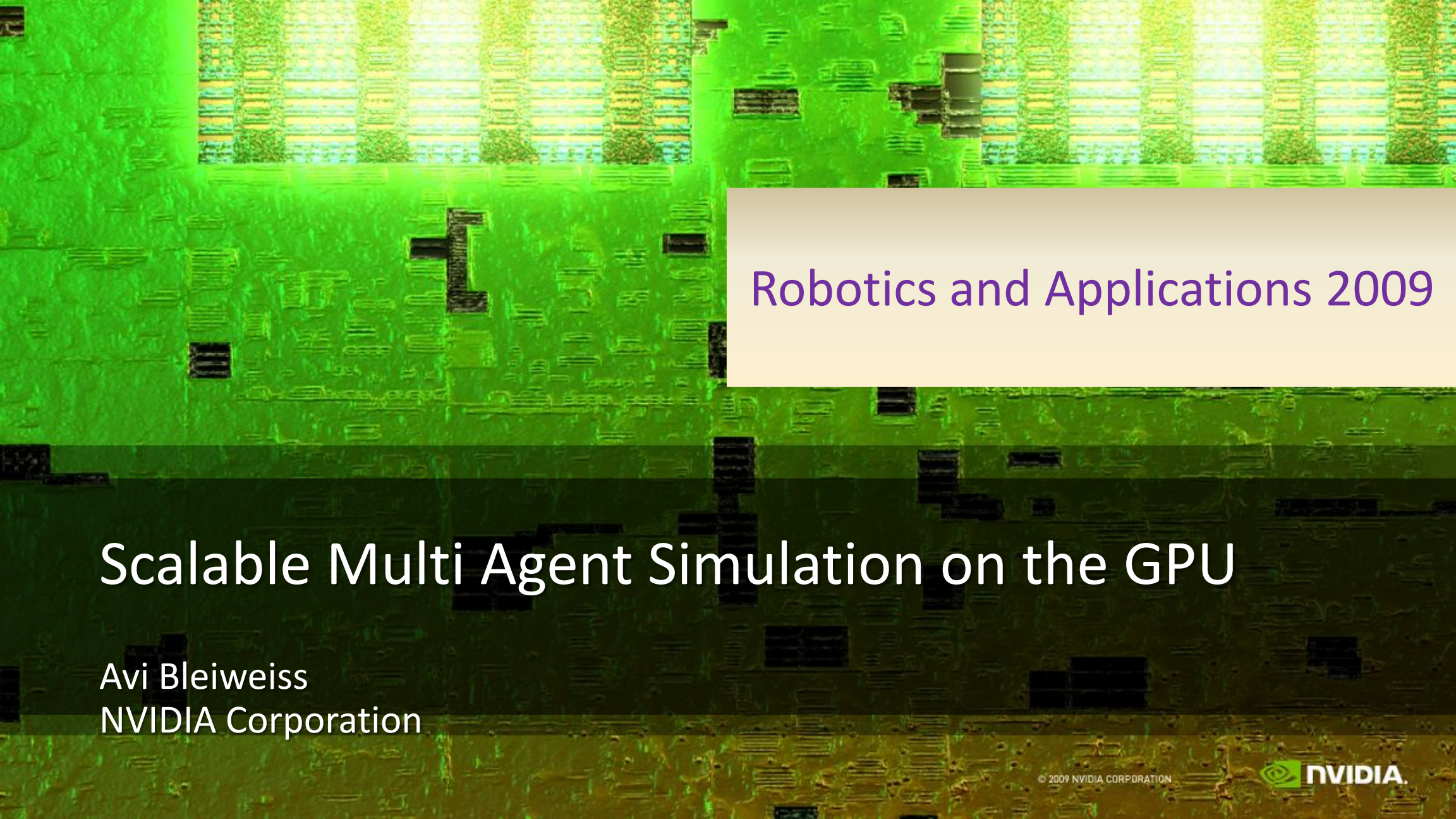


Robotics and Applications 2009

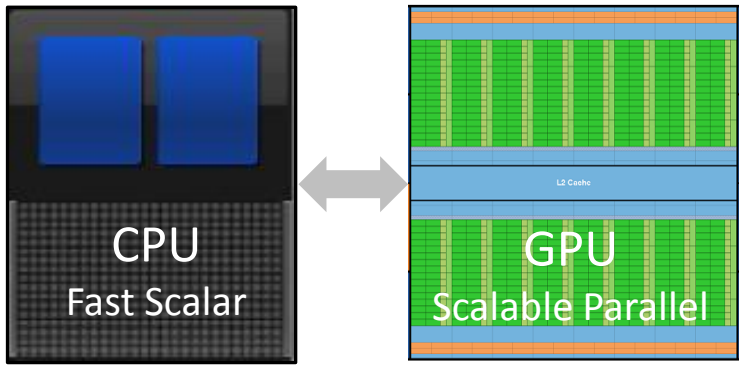
Scalable Multi Agent Simulation on the GPU

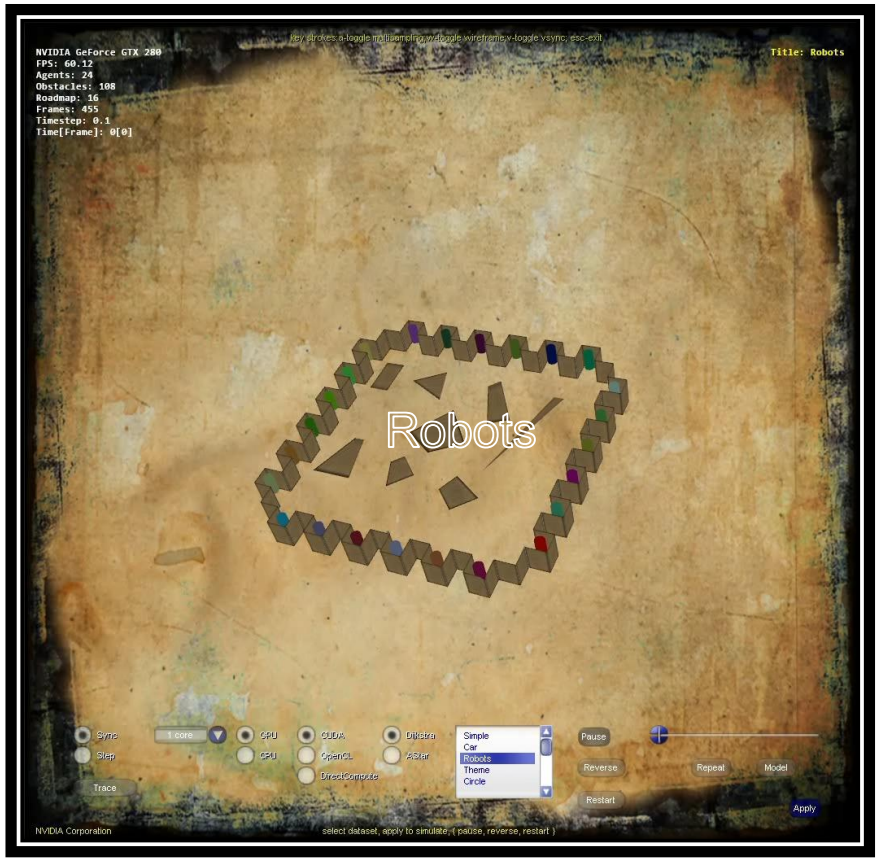
Avi Bleiweiss
NVIDIA Corporation

NVIDIA.

GPU Computing

- CPU, GPU co-processing
- C with few keywords
- Parallel kernels, SIMT
- Thread memory model

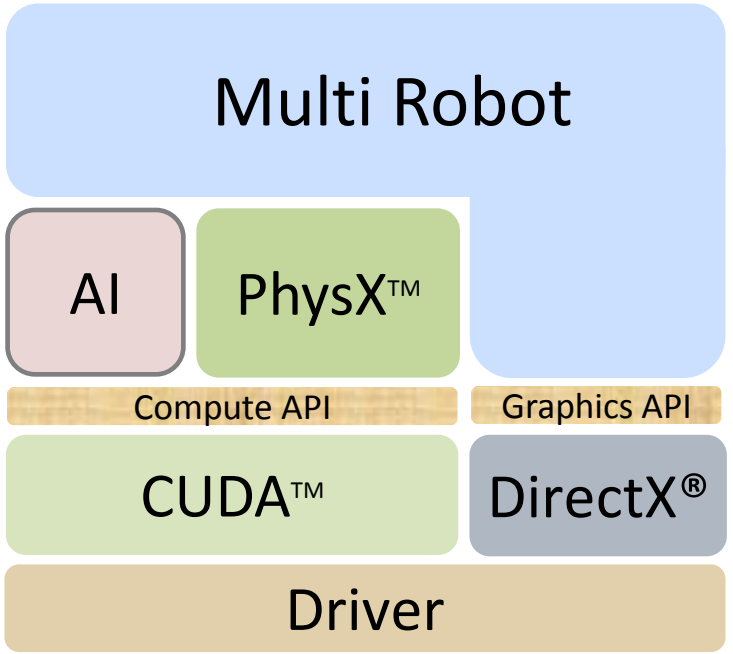




NVIDIA.

Motivation

- Effective team tasks
  - Virtual, physical
- Scalable, real time
  - Dense environments
- Joint framework
  - AI, physics simulation



Problem

Planner

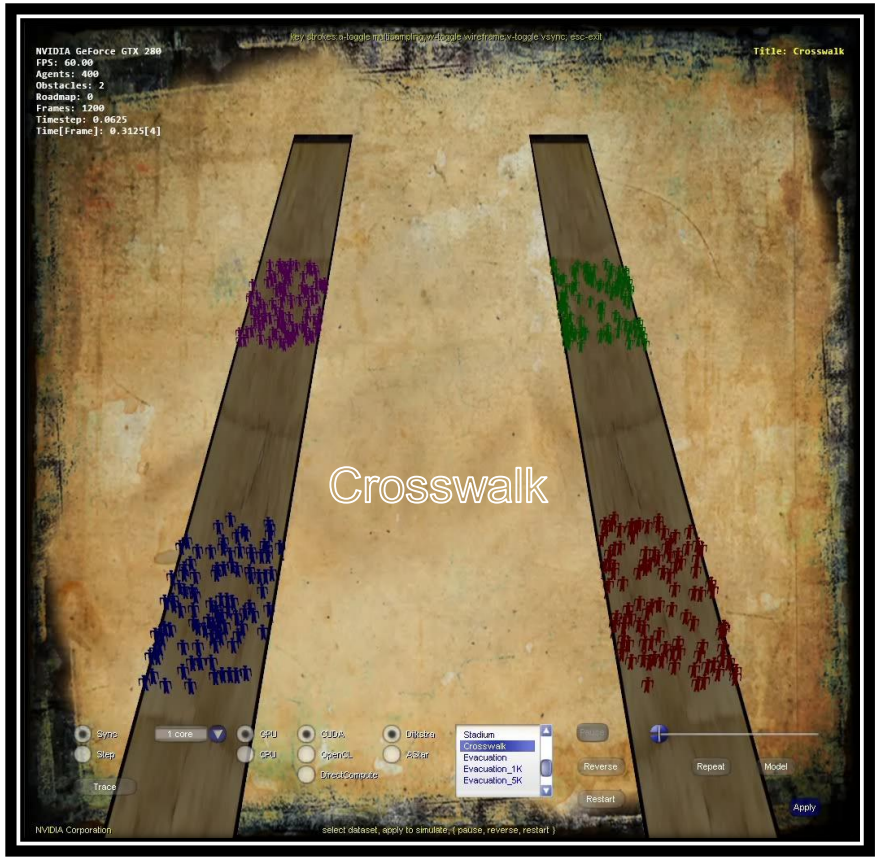
- Efficient mesh to roadmap conversion.
- Searches a global, optimal path
  - From start to goal
- Locally, avoids collisions with
  - Static, dynamic objects

Simulator

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

Solution

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



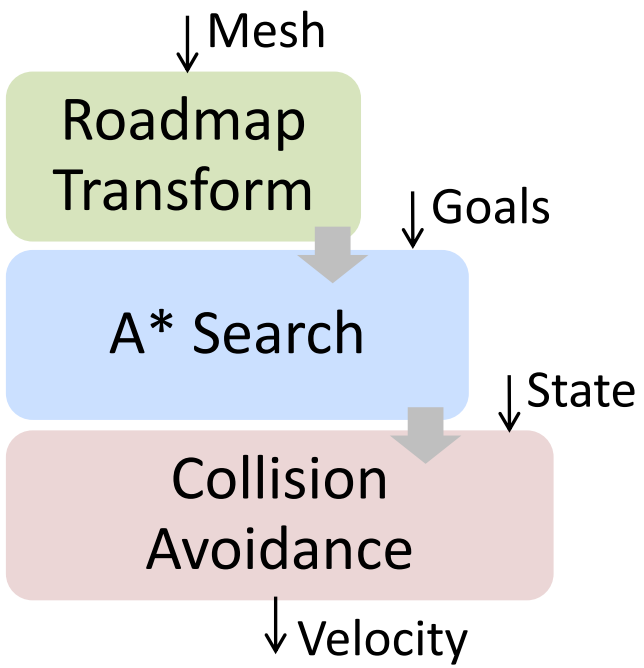
NVIDIA.

Outline

- Algorithm
- Implementation
- Results
- Takeaways

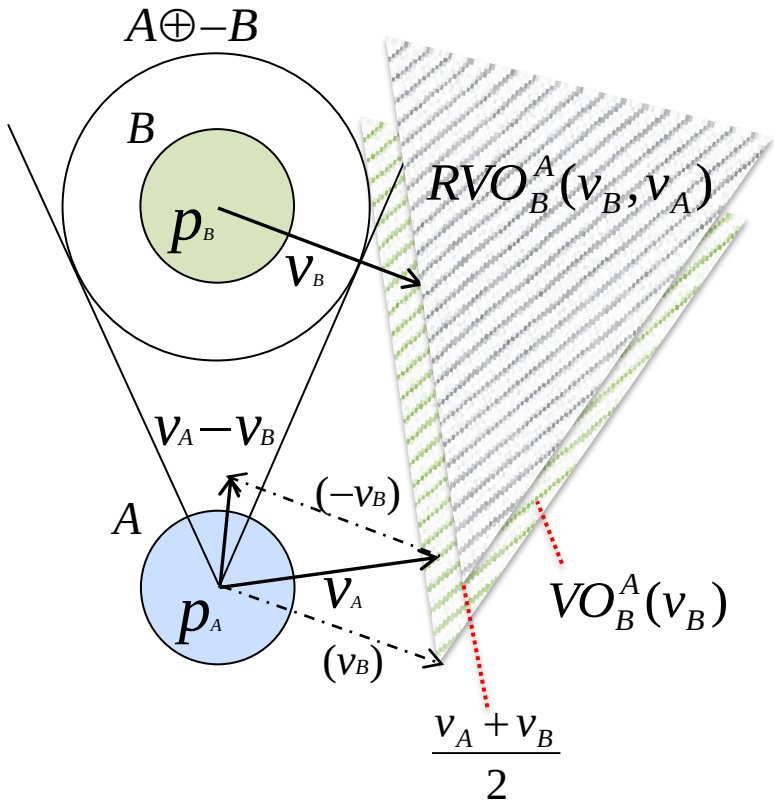
Pipeline

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



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]

Simulation

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

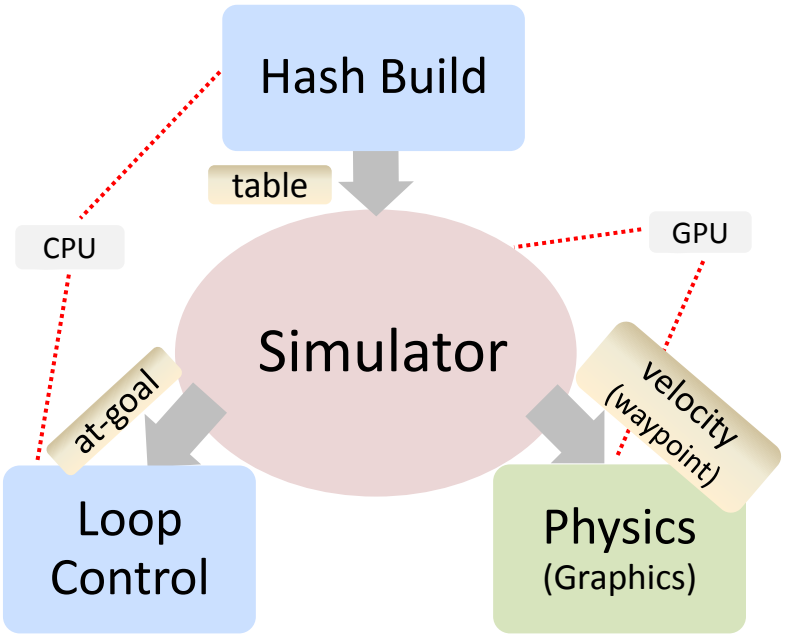
```
do
   hash
      construct hash table
   simulate
      compute preferred velocity
      compute proximity scope
      foreach velocity sample do
         foreach neighbor do
            if OBSTACLE then VO
            elseif AGENT then RVO
         resolve new velocity
   update
      update position,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

nested

flat

Workflow

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



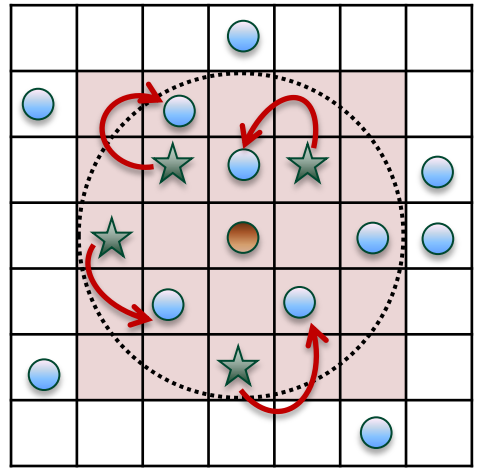
Collision Avoidance Frame Process

Challenges

- Hiding memory latency
- Divergent, irregular threads
- Small agent count (≤32)
- Hash construction cost

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

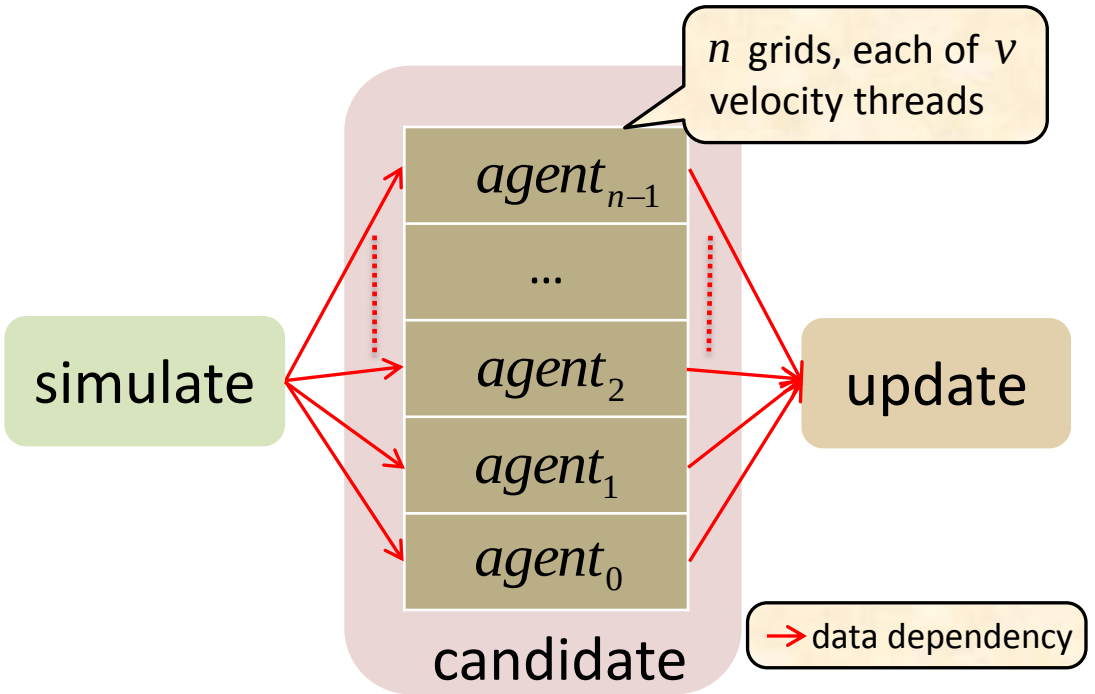


agent sample

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

Nested Parallel

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



Thread Grid DAG

Velocity Threads

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

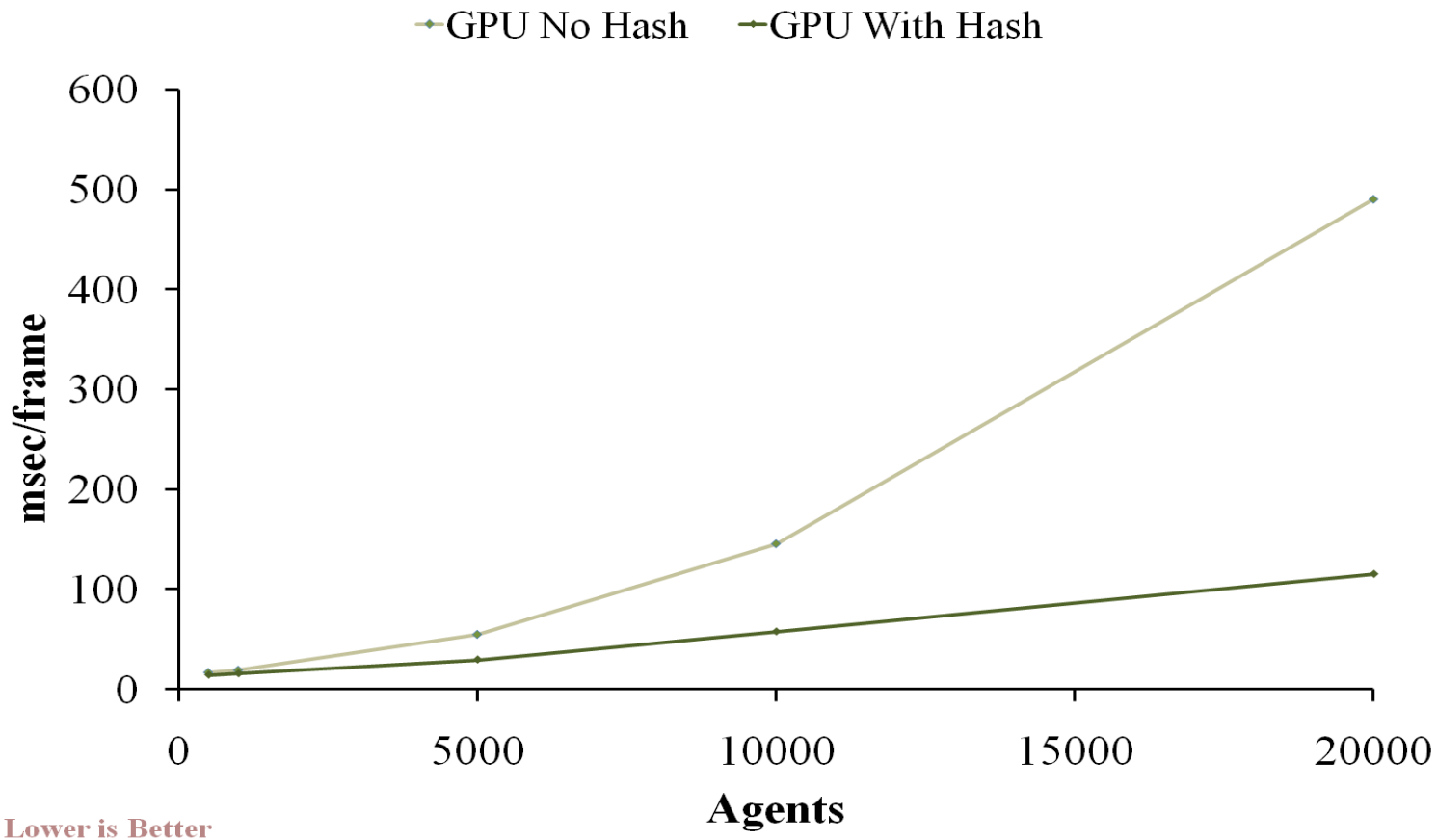
```
__global__ void
candidate(CUAgent* agents,
          int index,
          CUNeighbor* neighbors)
{
  float3 v, float t;
  CUAgent a = agents[index];
  if(!getThreadId()) v = a.prefvelocity;
  else v = velocitySample(a);
  t = neighbor(a, agents, neighbors, v);
  float p = penalty(a, v, t);
  reduceMinAtomicCAS(a,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

Experiments I

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

Spatial Hash



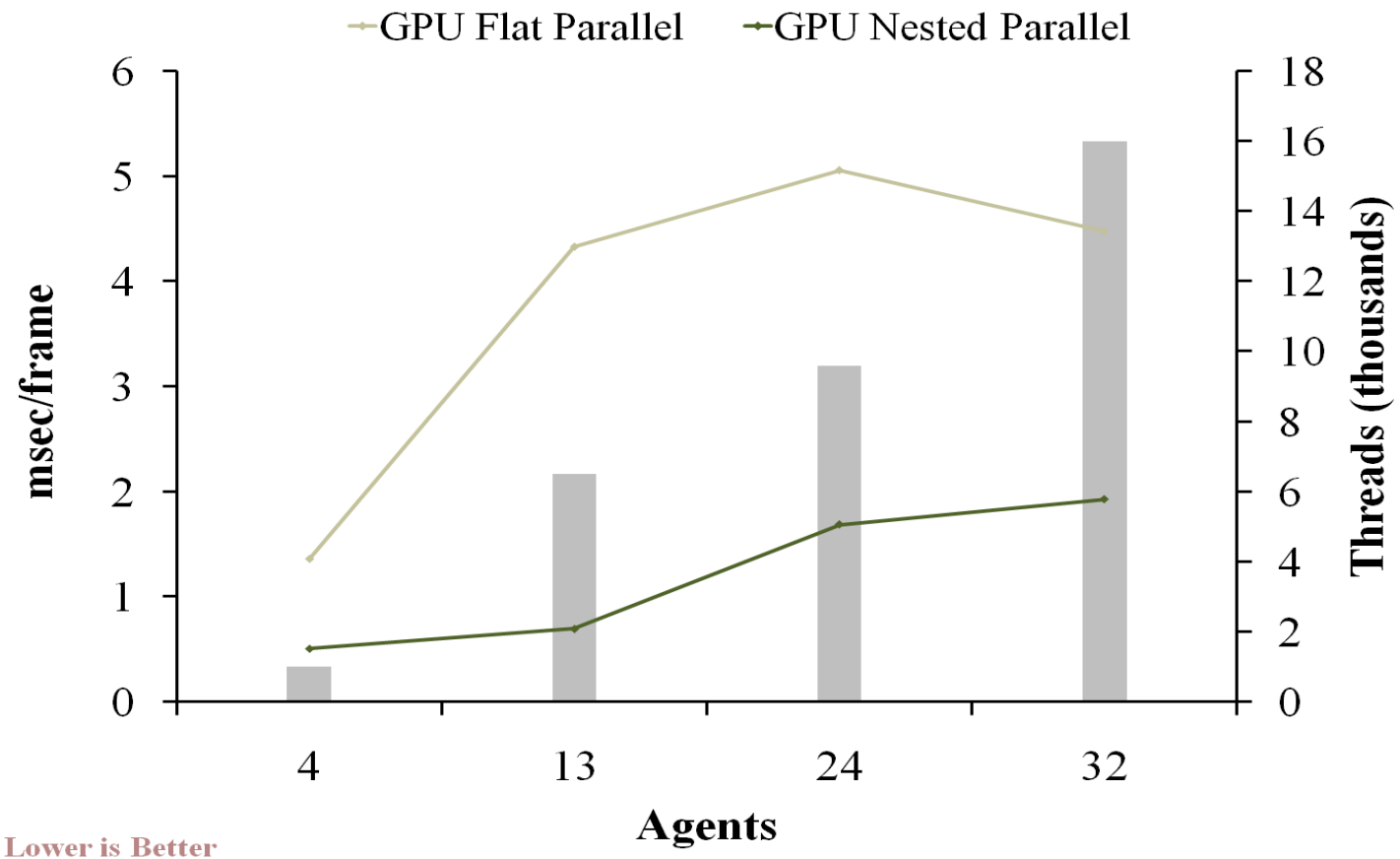
Lower is Better

Experiments II

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

Limitations

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

Future Work

- Fermi architecture scale
- Full 3D collision avoidance
- Parallel hash build
- Learning induced goals
- Evaluate UNC's FVO

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

Summary

- Multi agent solution
  - Compact, scalable
  - Fermi optimizations
- Cooperating robots
- Nested parallel potential
- AI, physics integration

Info

- Paper: GPU Accelerated Pathfinding, Graphics Hardware 2008
- Video: http://vimeo.com/7047078
- 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