Motion Planning and Proximity Computations for Industrial Robots

Dinesh Manocha

Department of Computer Science

UNC Chapel Hill

dm@cs.unc.edu

http://gamma.cs.unc.edu

Collaborators

- Jia Pan (UNC/Berkeley/Hong Kong Univ)
- Chonhyon Park (UNC)
- Ming Lin (UNC)
- Stephen Guy (UNC/Univ. of MN)
- Jamie Snape (UNC/Kitware)
- Jur Van Den Berg (UNC/Utah/Google)
- Sachin Chitta (Willow Garage/SRI)
- Ioan Sucan (Willow Garage/Google)

Motion Planning

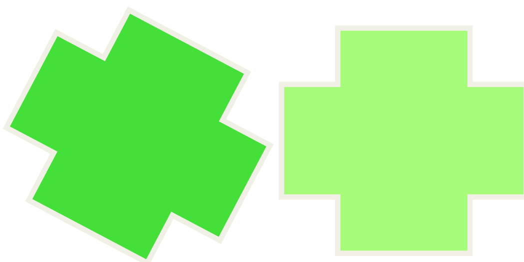
- Widely studied in academic for 40+ years
- Good algorithms and software tools
- Technology transfer (CAD/CAM, games, simulation)
- Limited use for industrial robots

Planning tools for Industrial Robotics: Challenges

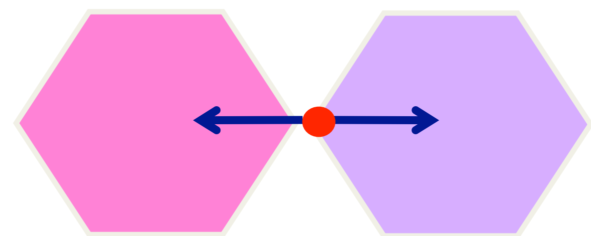
- Limited capabilities
- Limited development tools
- Lack of portability and flexibility
- Slow technology adoption

Collision & Proximity Queries

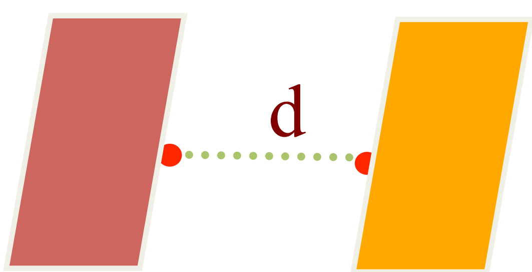
*Geometric reasoning of spatial relationships among objects (*in a dynamic environment)



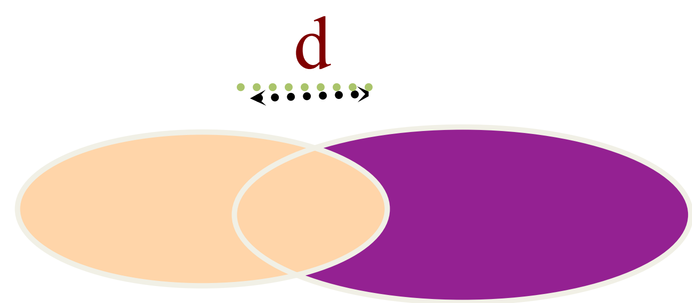
Collision Detection



Contact Points & Normals



Closest Points & Separation Distance



Penetration Depth

Collision & Proximity Computations

- A key component of motion planning algorithms (90% of total time)
- Widely used in CAD/CAM, simulation and virtual prototyping
- Studied in academia for 30+ years
- Supported in robot simulation and CAD systems

Our work on Proximity Computations

- Fast algorithms for convex polytopes (1991 onwards)
- Bounding volume hierarchies for general polygonal models (1995 onwards)
- Deformable models & self-collisions (2000 onwards)
- Use of GPUs and multi-core hardware (2005 onwards)
- Multi-robot planning and coordination (2008 onwards)

Prior work on Proximity Computations

Multiple software systems

- I-Collide, RAPID, PQP, DEEP, SWIFT, SWIFT++, DeformCD, PIVOT, Self-CCD, RVO, HRVO

http://gamma.cs.unc.edu/software/#collision

- More than 120,000 downloads from 1995 onwards

- Issued more than 55 commercial licenses (Kawasaki, MSC Software, Ford, Honda, Sensable, Siemens, BMW, Phillips, Intel, Boeing, etc.)

Prior work on Proximity Computations

Multiple software systems

- I-Collide, RAPID, PQP, DEEP, SWIFT, SWIFT++, DeformCD, PIVOT, Self-CCD, RVO, HRVO

  http://gamma.cs.unc.edu/software/#collision

- More than 120,000 downloads from 1995 onwards

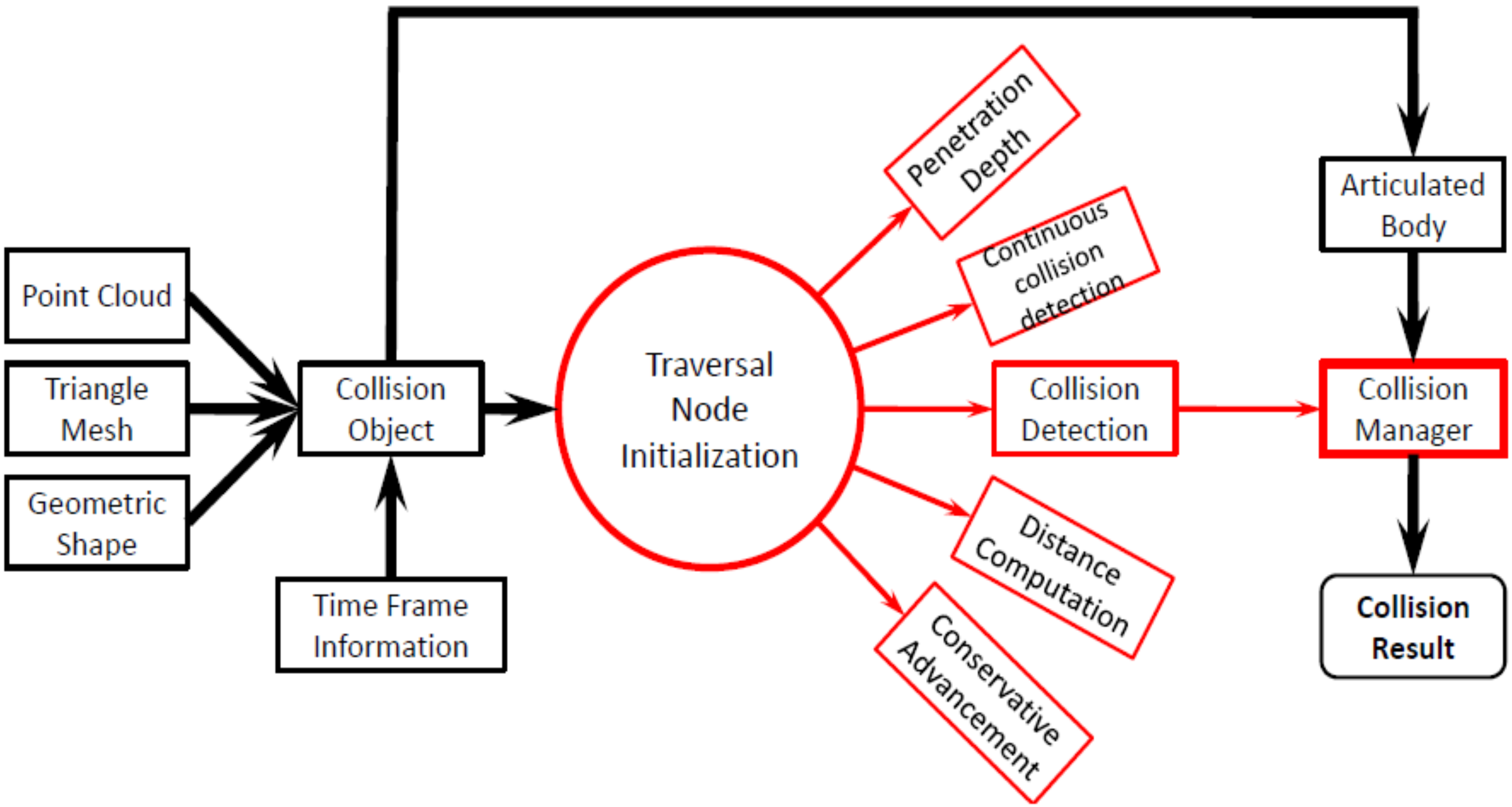
- Issued more than 55 commercial licenses (Kawasaki, MSC Software, Ford, Honda, Sensable, Siemens, BMW, Phillips, Intel, Boeing, etc.)

Widely used, but not on industrial robots

Recent Work: FCL

- A new collision and proximity computation library
  - Flexible: different object types/ queries
  - Extensible: adding new algorithms is easy
  - Efficiency: similar performance with the best libraries
- Provide many functions from state-of-the-art research, more in future

FCL Overview



http://gamma.cs.unc.edu/FCL/

FCL: Supported Functions (Spring 2014)

	Rigid Objects	Point Clouds	Deformable Objects	Articulated Objects
(Discrete) Collision Detection	Y	Y	Y	Y
Continuous Collision Detection	Y	Y	Y	Y
Self Collision Detection	Y	Y	Y	Y
Penetration Estimation	Y	N	N	N
Distance Computation	Y	N	Y	Y
Broad-phase Collision	Y	Y	Y	Y

FCL: Usage

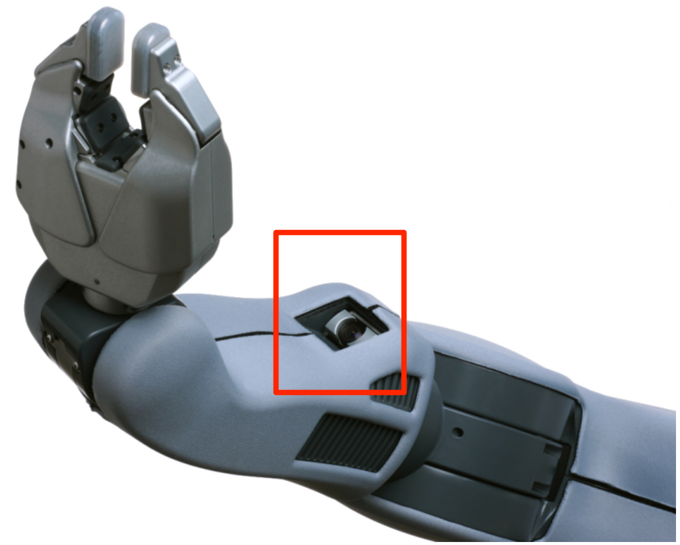
- Independent code, but ROS interface is provided
- Available at http://gamma.cs.unc.edu/FCL
- Part of MoveIt:

  http://moveit.ros.org/wiki/MoveIt!

FCL Application: Optimal Inverse Kinematics for Complex Path Planning

http://rosindustrial.org/

Robot Sensors: Data Collection

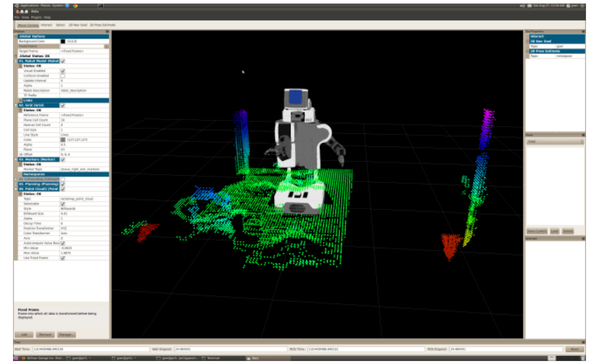


Cameras

Robot Sensors: Data Collection

Laser Scanners

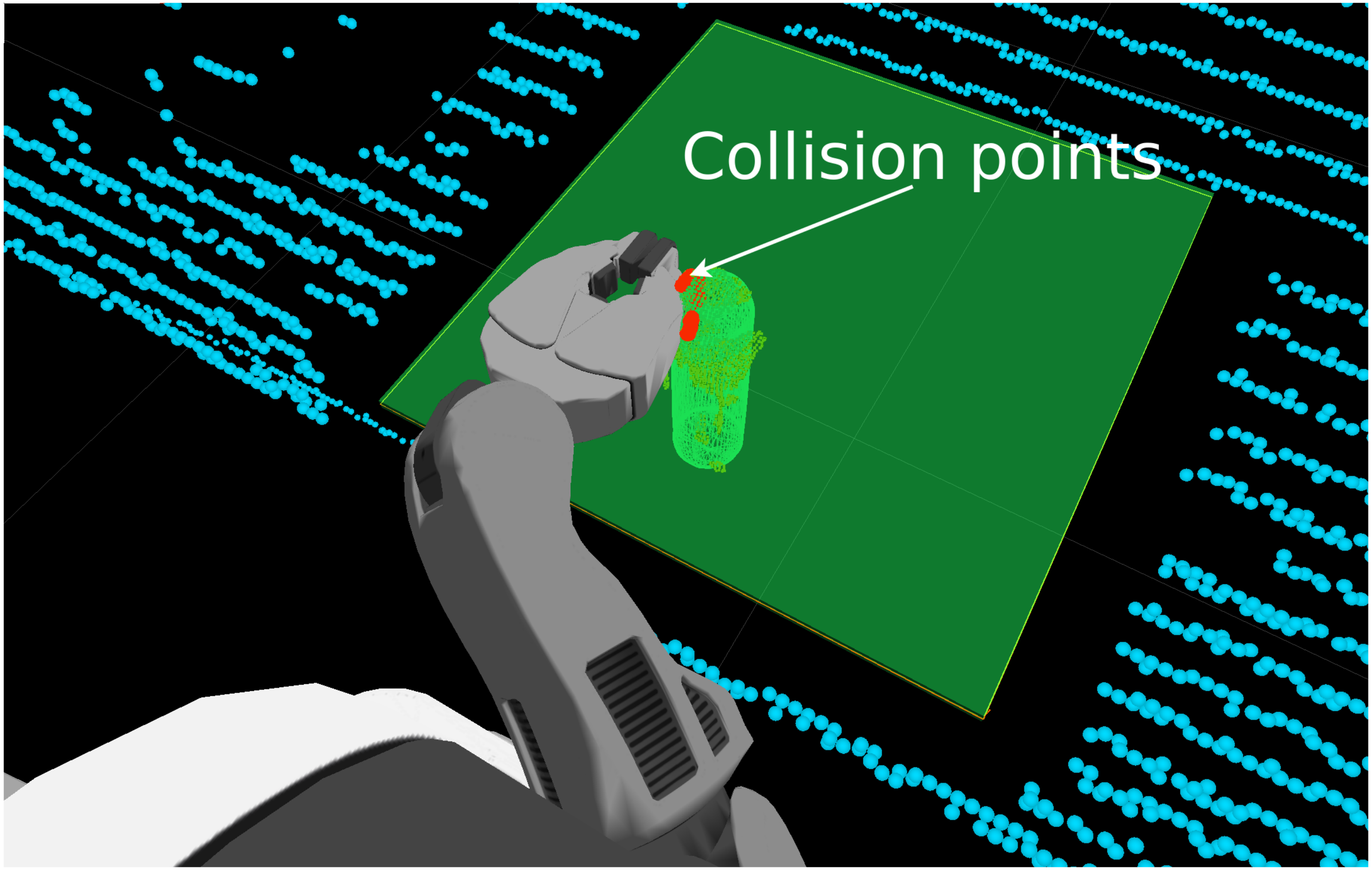
Handling Sensor Data



- Human environments
  - Clutter, dynamic obstacles
- Data from 3D sensors
  - Large number of points (~10k for laser scans, ~20k for stereo)
- Real-time computation important for fast online reactive grasping, motion planning
- Proximity computation important for many useful heuristics in robotics

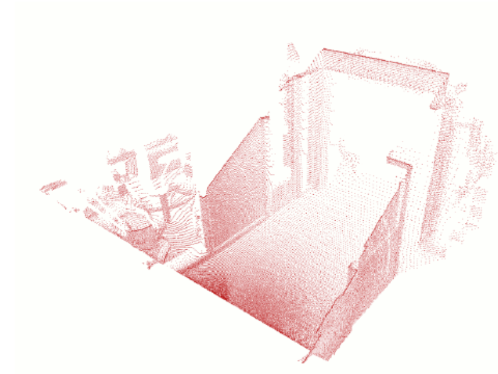
Efficient collision and proximity computation is essential for any online robot operations in human environments

Reconstructed Point Clouds

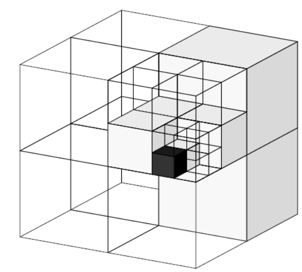


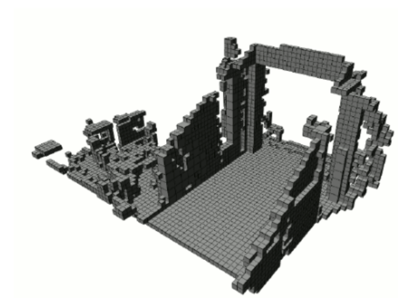
http://gamma.cs.unc.edu/POINTC/

Sensor Data



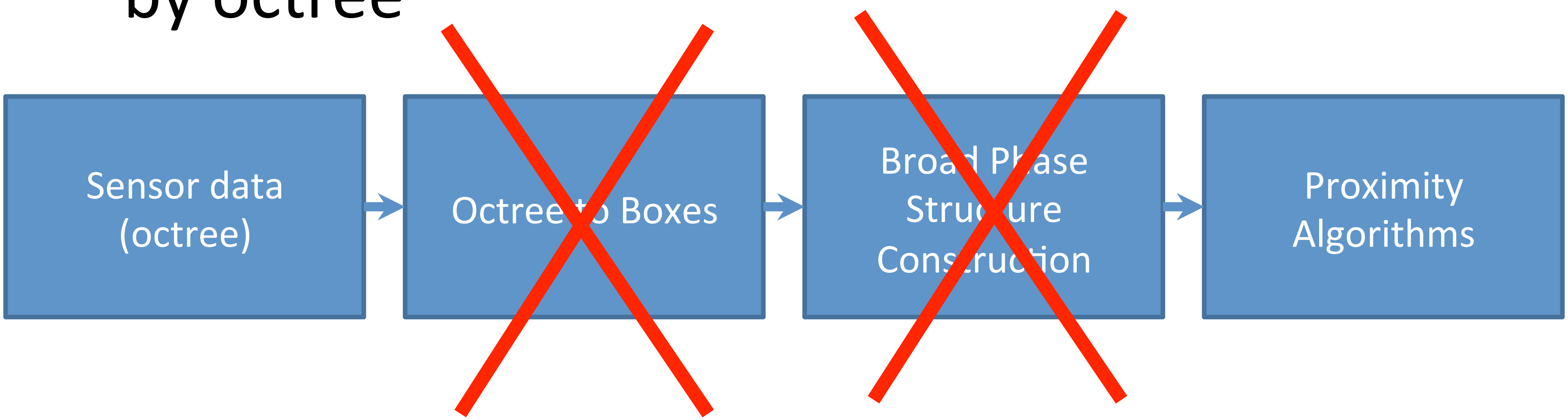
- Point cloud
  - Output from laser/Kinect, etc.
  - Cannot encode unknown regions
  - Very large
- Octree (octomap)
  - Store point cloud in a compact manner
  - Support multi-resolution
  - Encode occupied/free/unknown regions





Accelerated Pipeline: Point-Cloud Data

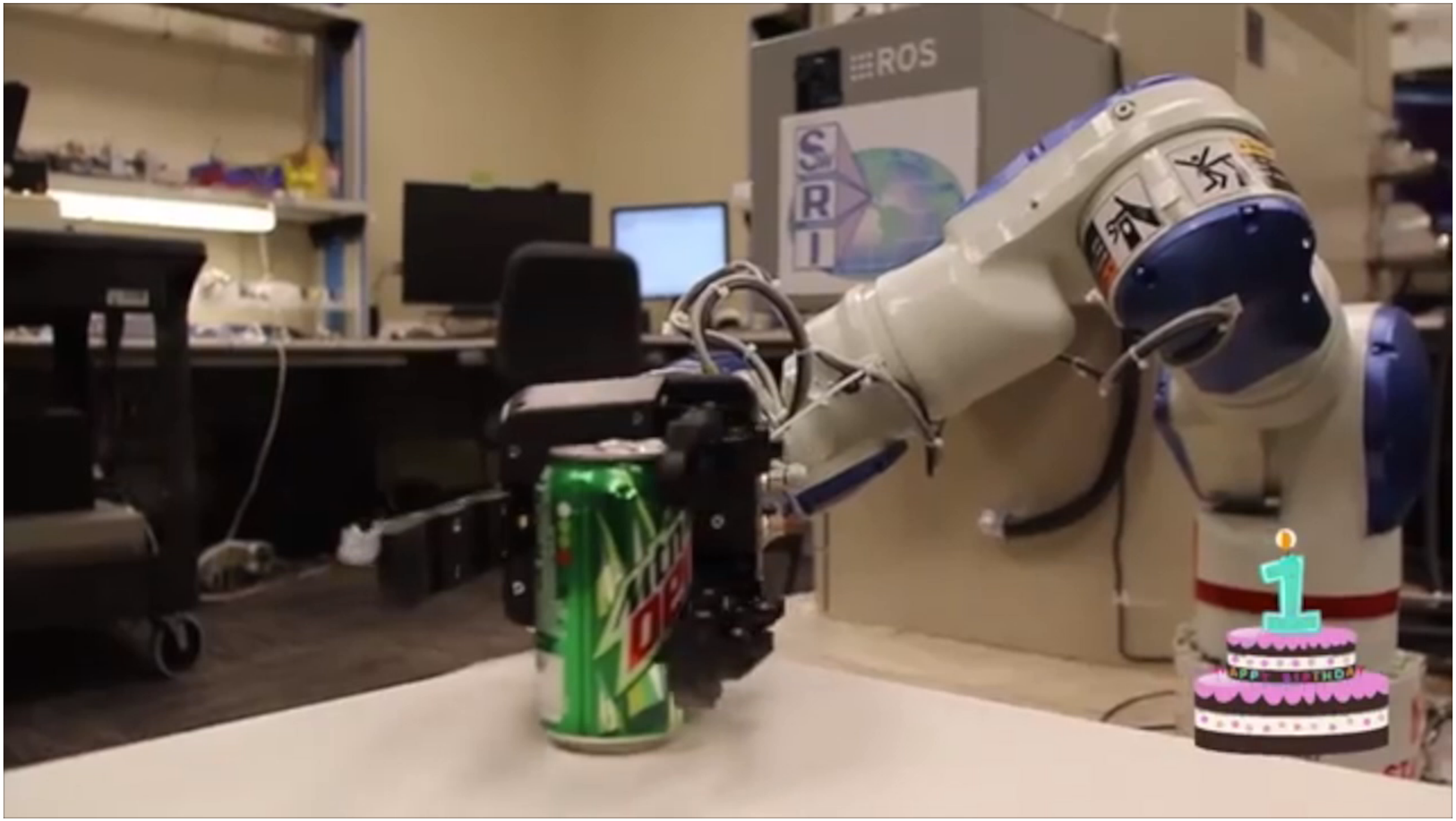
- Directly collide with sensor data represented by octree



- We eliminate construction time in this pipeline

http://gamma.cs.unc.edu/POINTC/

Proximity & Planning with Industrial Robots

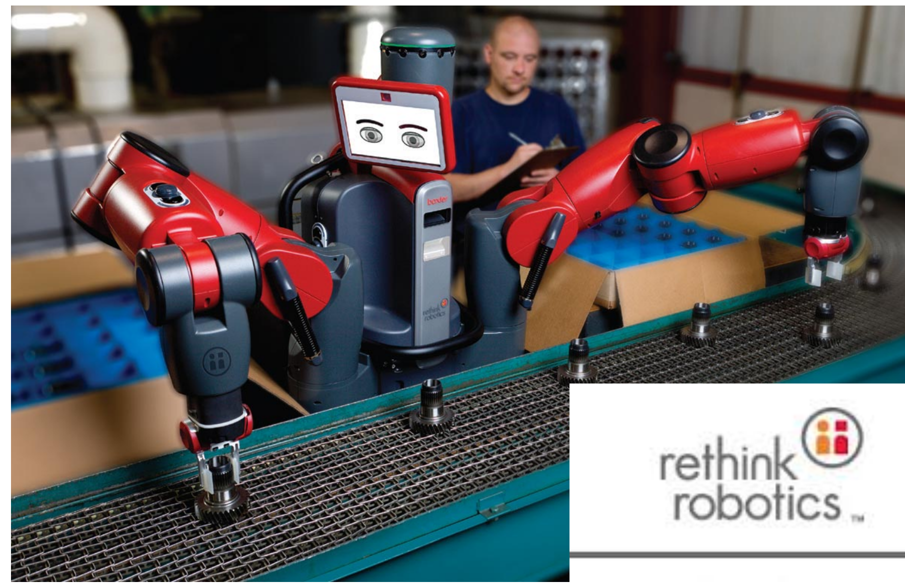


http://rosindustrial.org/

Task Executions of Robots

- Advances in technology allow robots to perform complex tasks

<PR2: fetching a beer from the fridge>



<Baxter: $22k robot needs no programming>

Task Execution with Multiple Components

- A task is decomposed into many primitive subtasks

<PR2: taking out a beer from the fridge>
(From Willow Garage)

1. Move the body to the fridge.
2. Move the left arm to the handle.
3. Move the body to open the fridge door.
4. Move the body to in front of the fridge.
5. Move the left arm to hold the door.
6. Move the right arm to the beer.
7. Grasp the bottle.
8. Move the right arm to the basket.
9. Release the bottle.

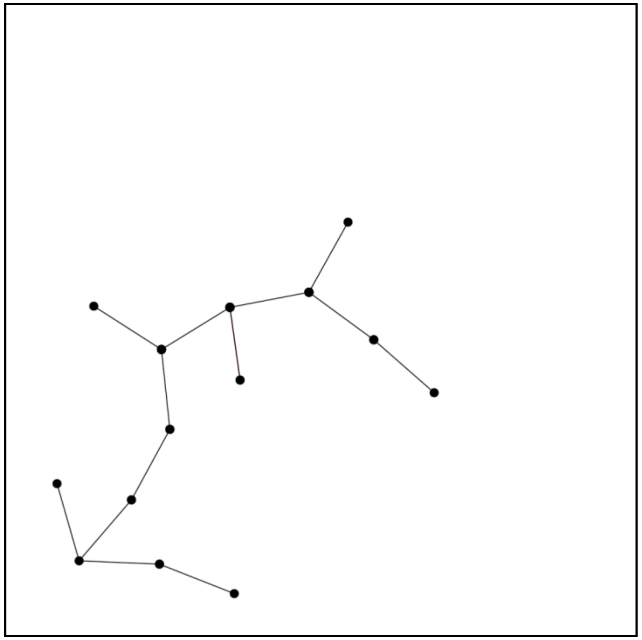
Most of the subtasks are moving the robot to the next desired pose

Subtask Execution

- 'Move the body to the fridge' subtask
  - Use sensors to recognize the objects and obstacles in the environment
  - Compute a collision-free path to the pose close to the fridge
  - Control motors to execute the computed motion

Motion Planning: RRT Algorithm

- Serial RRT
  tree expansion

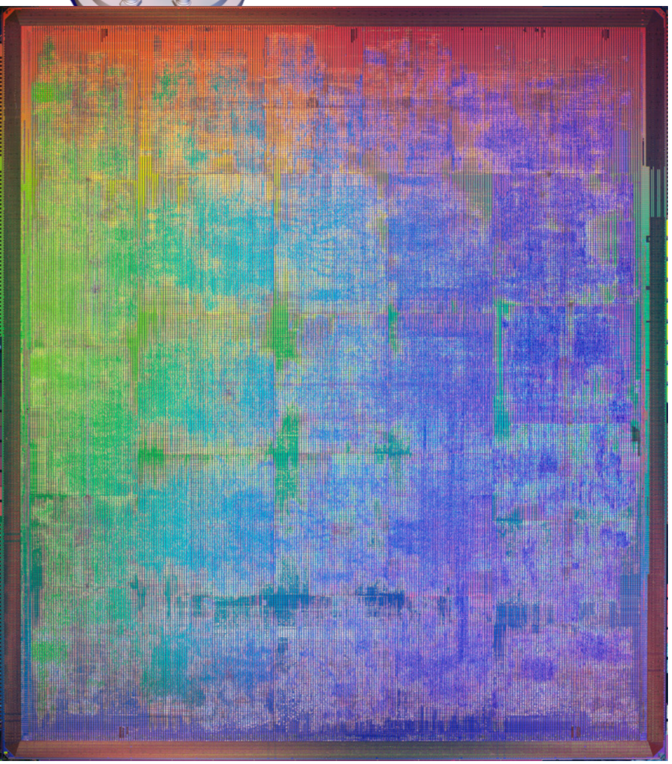


Motion Planning: Challenges

- How to perform motion planning computation in realtime?

- How to handle high DOF robots

NVIDIA & AMD GPU Compute Accelerators



AMD Radeon 7970

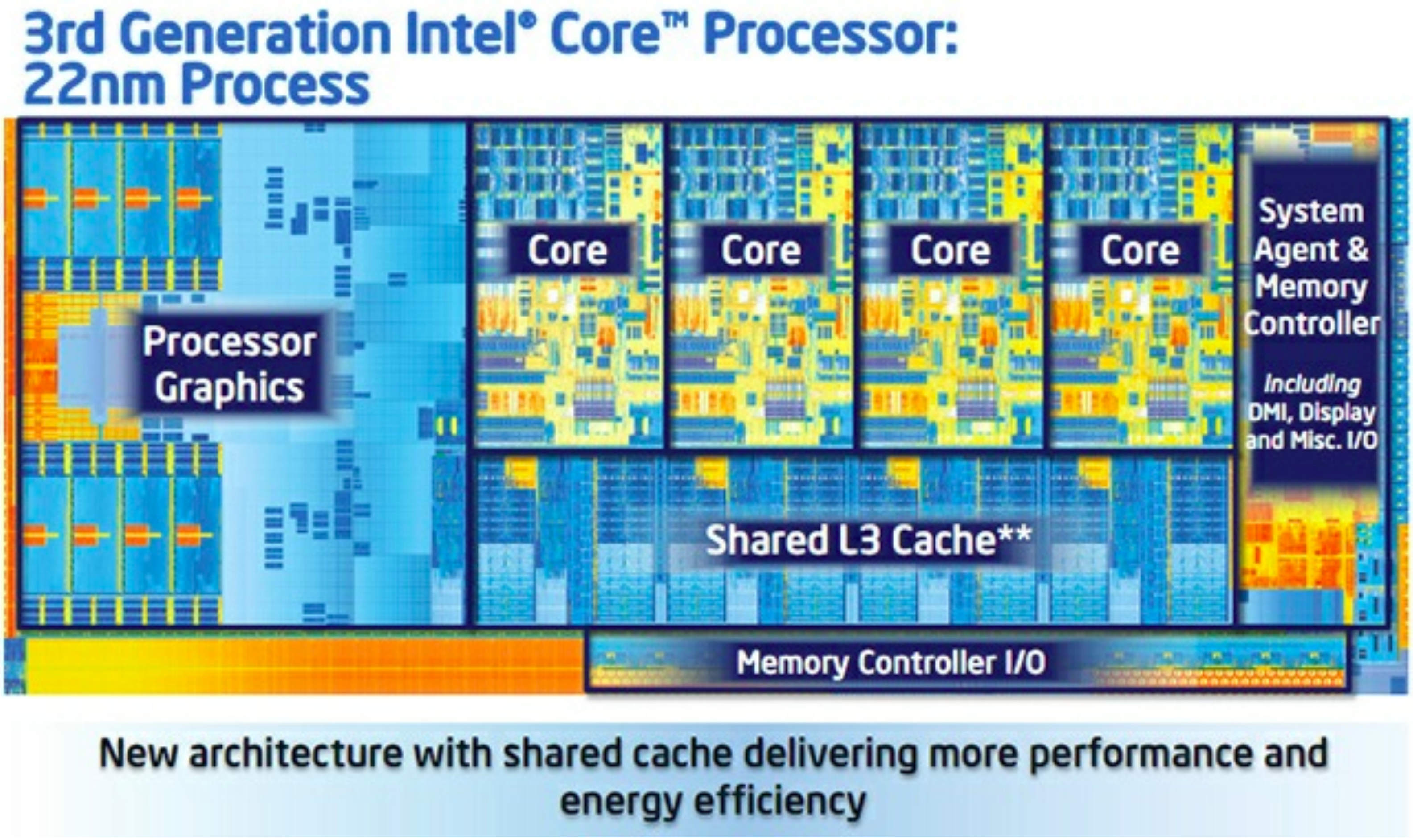
3.79 Single Tflops
947 Double Gflops
2048 Stream Cores

NVIDIA GTX 680

3.09 Single Tflops
1.1 Double Tflops
1536 CUDA Cores

Commodity Tera-Flop Processor (peak performance)

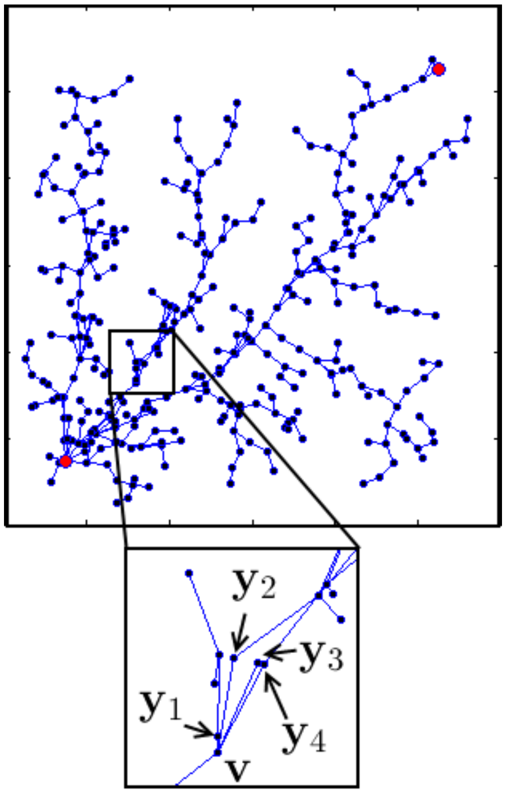
Heterogeneous Processing

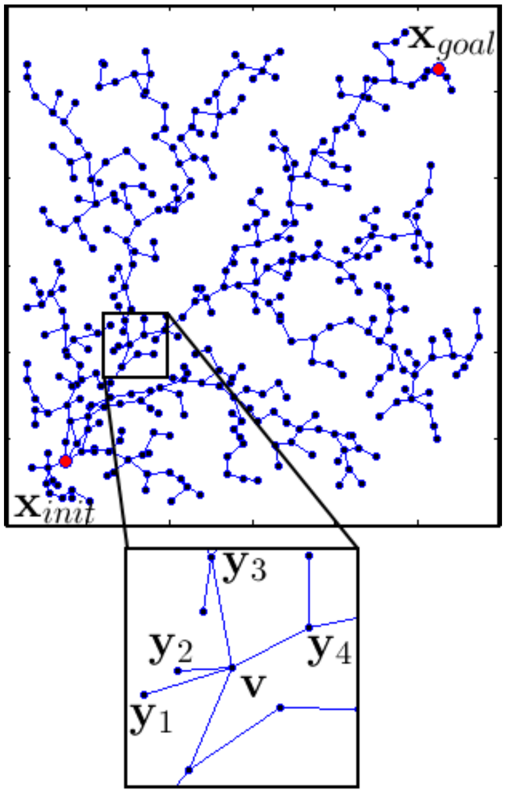


CPU + GPU on the same chip

Parallel Poisson-RRT Algorithm

- AND Parallel RRT Tree
- Poisson-RRT Tree

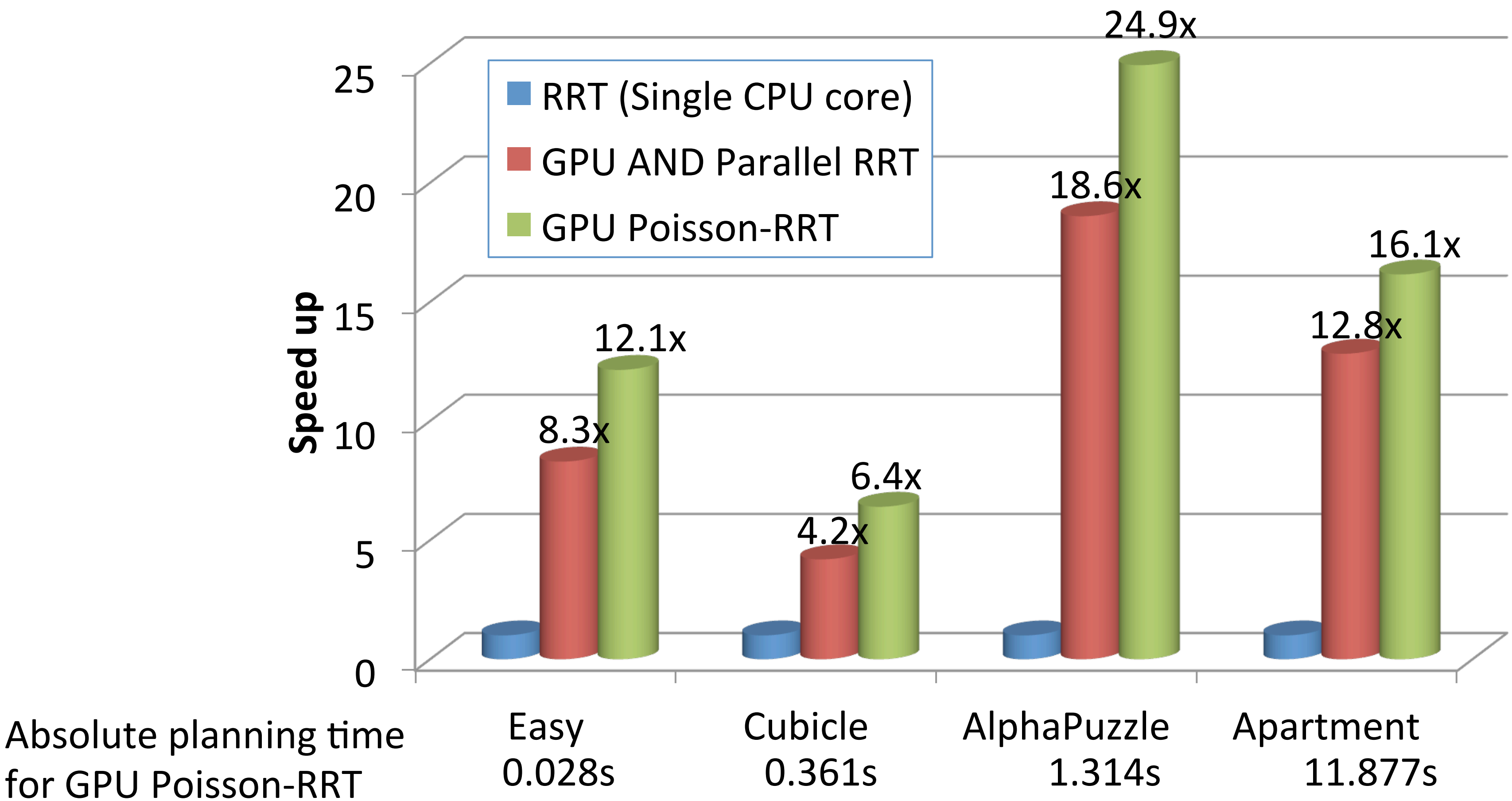




No nodes which are too close to each other

GPU-based RRT Planner

- OMPL Benchmarks



High-DOF Realtime RRT Planning

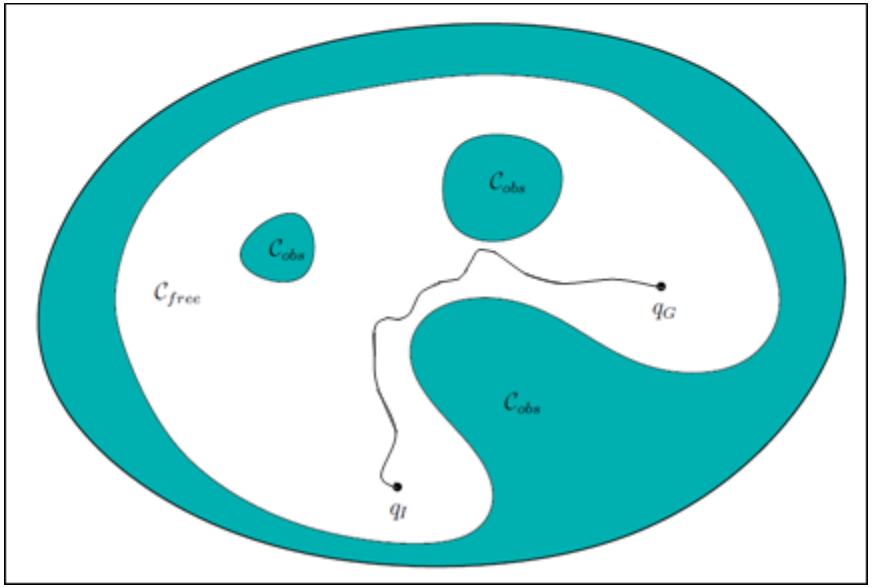
http://gamma.cs.unc.edu/PoissonRRT/

Real-Time Planning: High DOF Robots

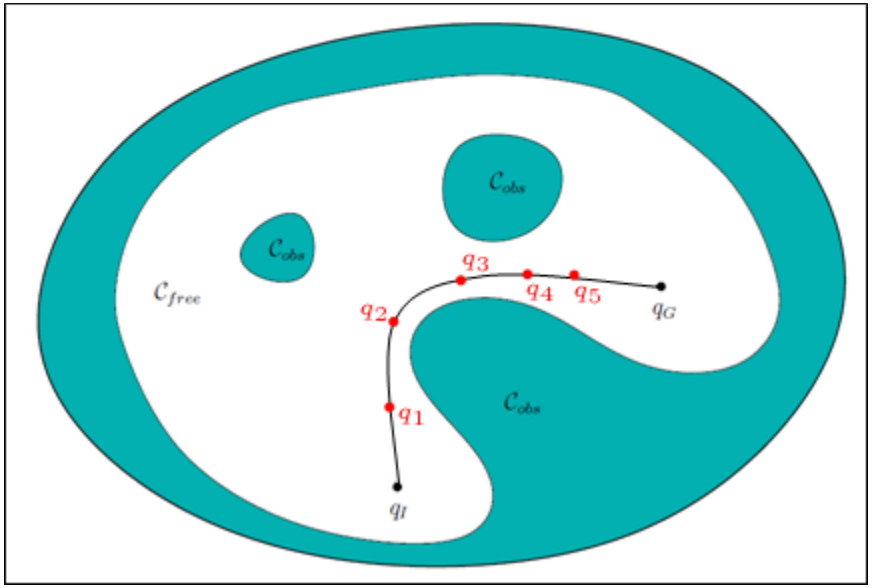
- High-DOF robots (40 DOF for humanoids)
- Generate collision-free and smooth paths
- Dynamics constraints (e.g. dynamic stability)
- Moving obstacles (e.g. humans)

Motion Planning Algorithms

- Random sampling-based algorithms



- Optimization-based algorithms



Real-Time Planning

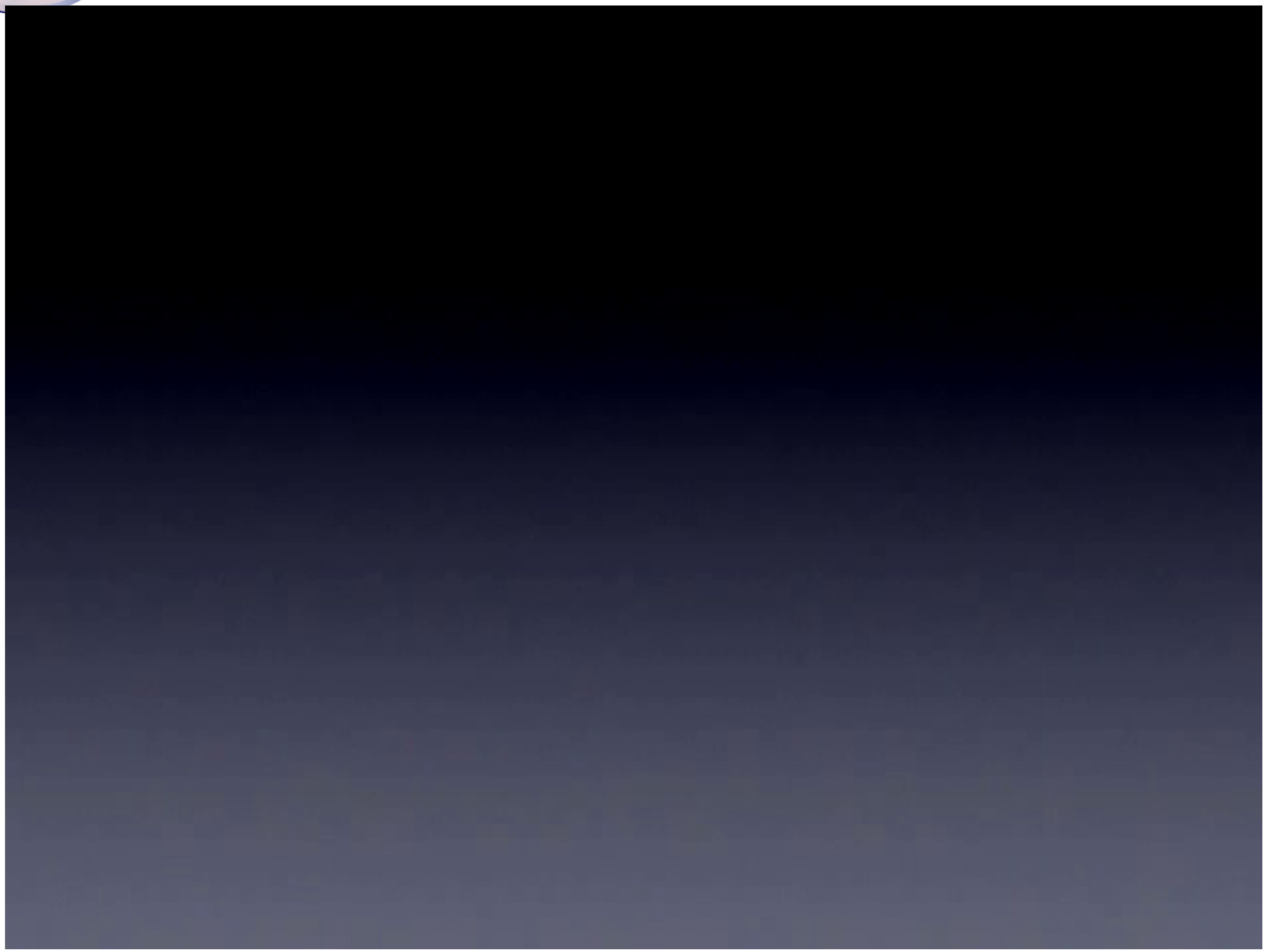
- Use optimization-based techniques
- Formulate constraints
- Parallel computation on multi-core CPUs and many-core GPUs

Parallel Trajectory Optimization

- Parallel optimization of multiple trajectories
  - Use Multiple threads
    - Start from different initial trajectories
    - Trajectories are generated by quasi-random sampling
  - Exploits the multiple CPU cores (multi-cores) or GPU-based cores (many-cores)

http://gamma.cs.unc.edu/ITOMP/ITOMP_ROS/

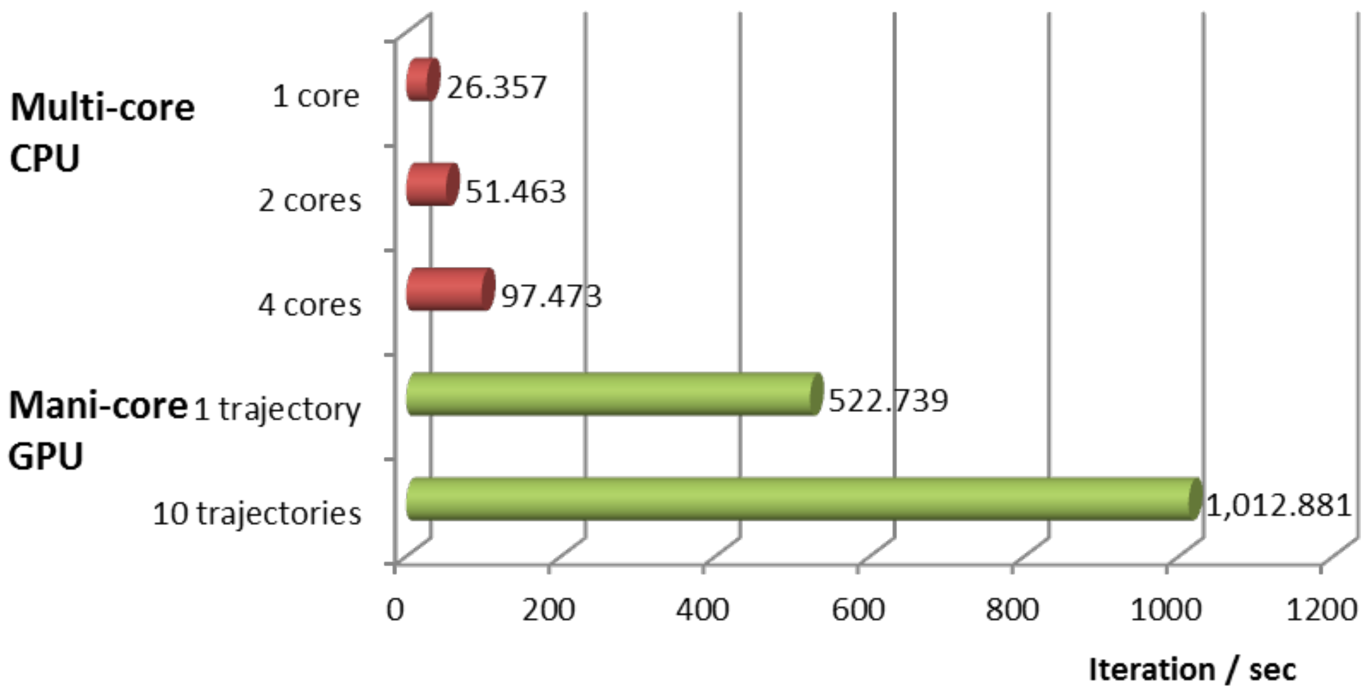
Real-Time High DOF Planning



http://gamma.cs.unc.edu/ITOMP

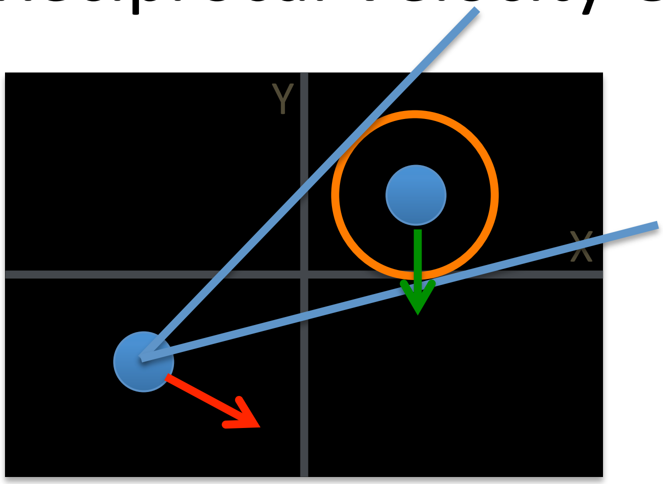
Parallel Trajectory Optimization

Performance improvement with number of cores

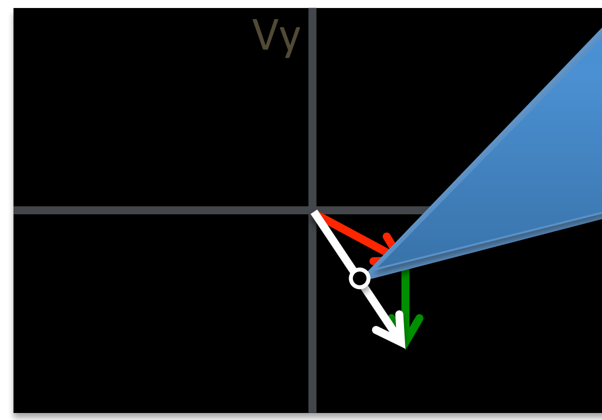


Multi-Robot Planning

- Reciprocal Velocity Obstacles (RVOs)



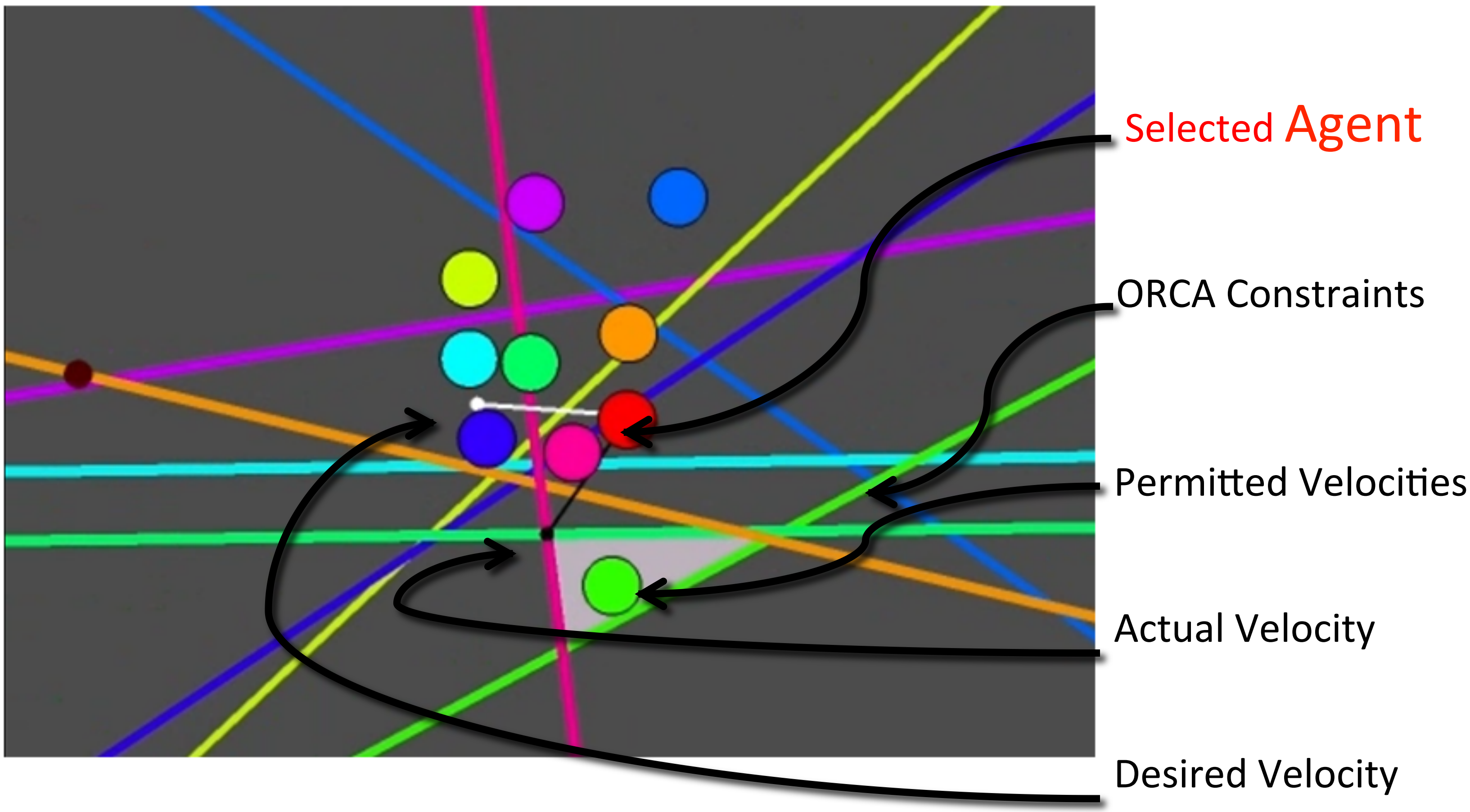
Workspace



Velocity Space

- $RVO^A_B(v_B, v_A) = \{v'_A \mid 2v'_A - v_A \in VO^A_B(v_B)\}$ [Berg et al. 2008]

MULTI-AGENT COLLISION AVOIDANCE (ORCA)

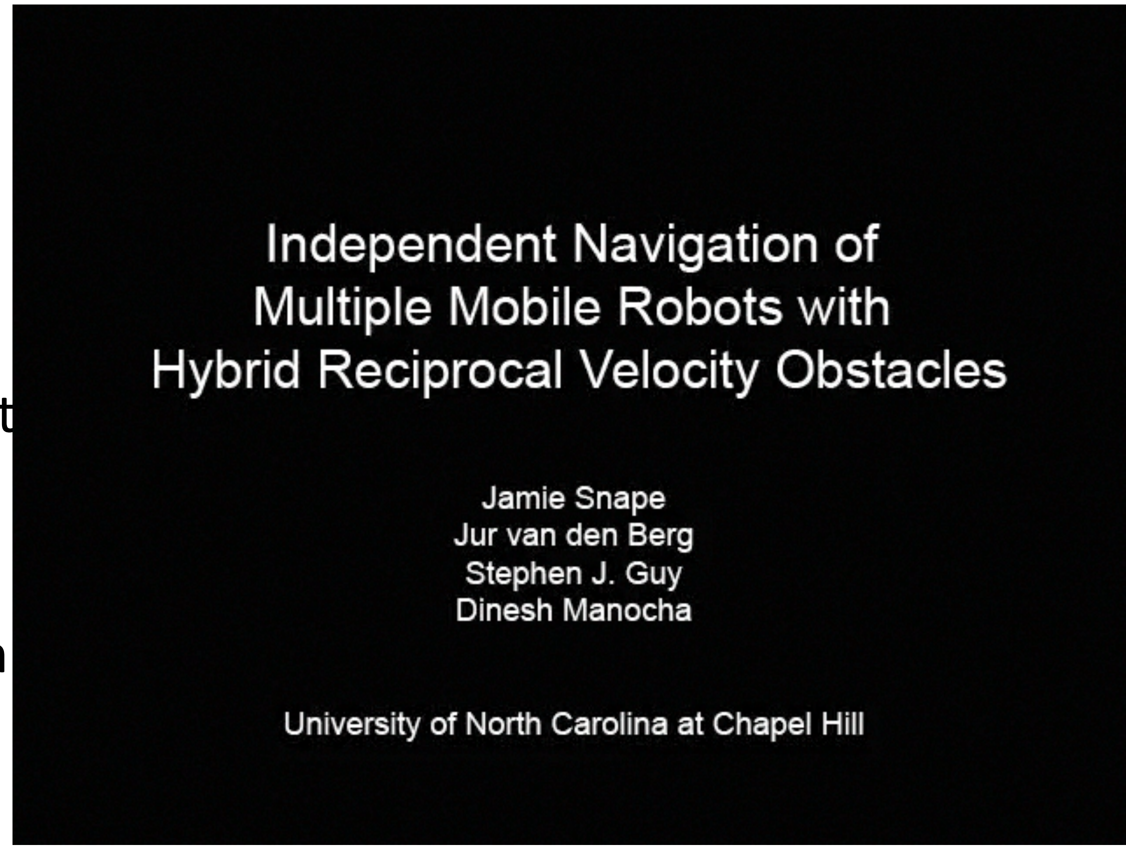


[Berg. Guy et al. 2010]

Multi-Robot Navigation

[Snape et al. 2009, 2011]

- Reciprocal velocity-space planning on robots
- Challenges for robots
  - Sensor Uncertainty
  - Motion Uncertainty
  - Kineodynamic Constraint
- Implementation on iRobot Create
- ROS Library implementation available



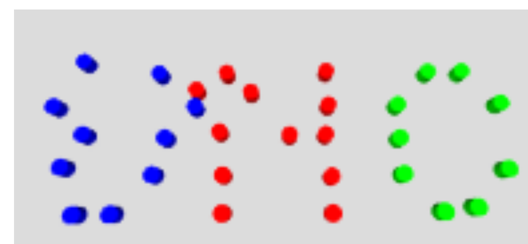
http://gamma.cs.unc.edu/HRVO

Multi-Robot Navigation: ROS Integration

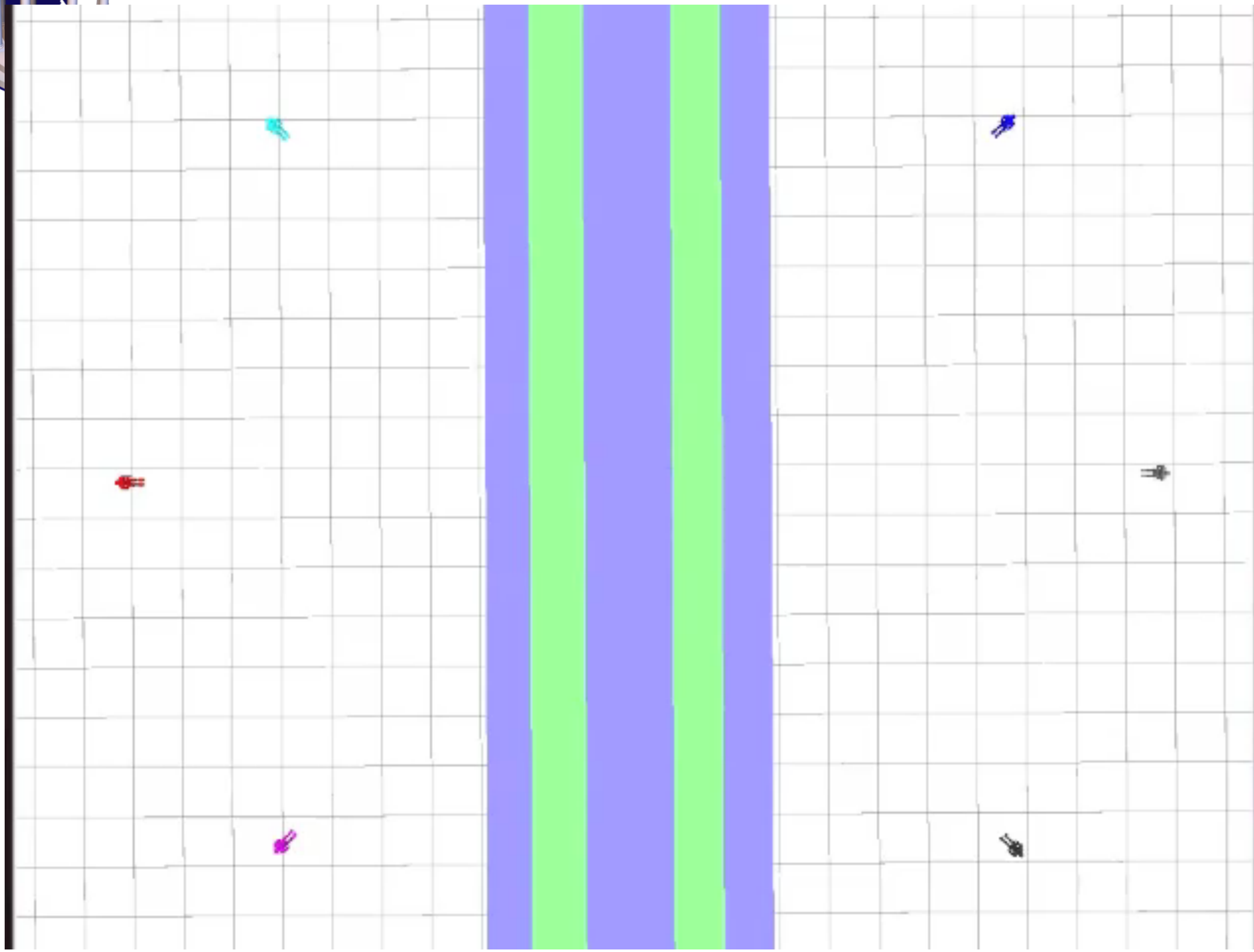
[Claes et al., 2011, Willow Garage]

RVO2 Library

- Publically available Library: http://gamma.cs.unc.edu/RVO2
- 6,500+ Downloads
- Licensed to Relic Entertainment, EA,GameLoft …
- Widely used in game engines (Unity, UDK)
- Integrated into ROS
- Ports
  - ORCA – C++
  - ORCA – C#
  - ORCA3D – C++



Multi Human-Like Robots



http://gamma.cs.unc.edu/ITOMP

Going Forward

- Significant recent progress in algorithmic technology for motion planning and proximity computations
- Good software tools: FCL, ROS, MoveIT, OpenRAVE, etc.
- Applications
  - Advanced manipulation
  - Advanced perception
  - Flexible automation
- Major Challenge: Interface with robot devices and industrial use

Ackowledgements

- Army Research Office
- National Science Foundation
- Intel
- Sandia Labs
- Willow Garage