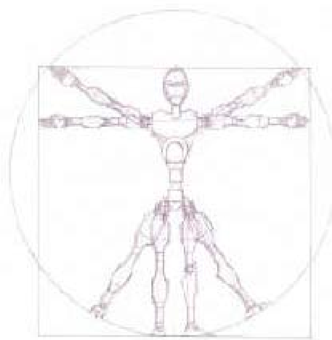
RoboCup Rescue Simulation
from the Agent Competition towards the DARPA Challenge



Arnoud Visser

Universiteit van Amsterdam
Informatics Institute

14th TAROS Conference,
Towards Autonomous Robots and Systems,
St. Anne's College, Oxford, August 29, 2013

Intelligent Robotics Lab

The RoboCup Challenge for the AI

By the year 2050,

develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.

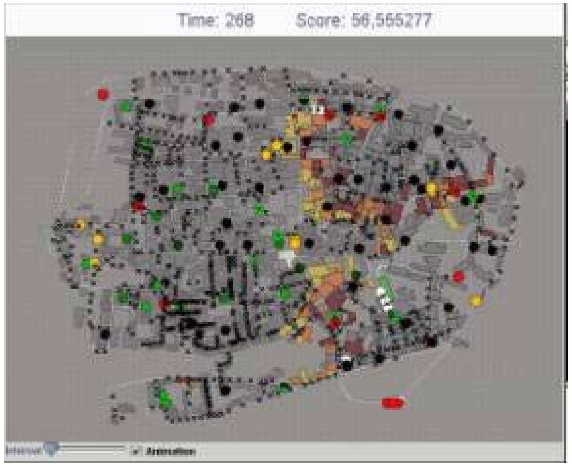
RoboCup

Robocup World Championships

RoboCup

Yearly competition & symposium, with ± 400 participating teams

RoboCup Rescue



Agent Simulation competition

Robot competition

Virtual robot competition

The Common Knowledge Model of a Team of Rescue Agents

M.L. Fassaert, S.B.M Post, and A. Visser

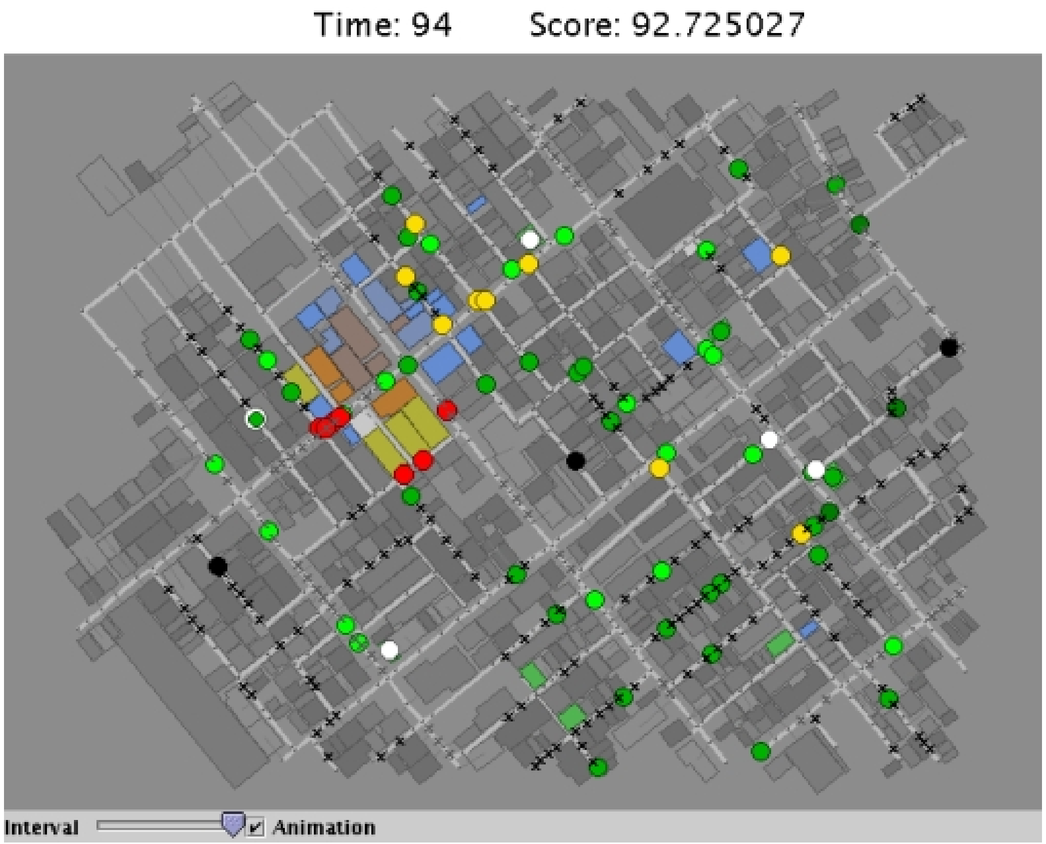
UvA Rescue C2003

1th International Workshop on Synthetic Simulation and
Robotics to Mitigate Earthquake Disaster, Padova, Italy, 6 July 2003

Universiteit van Amsterdam
Informatics Institute

Intelligent Systems Laboratory
Intelligent Autonomous Systems group

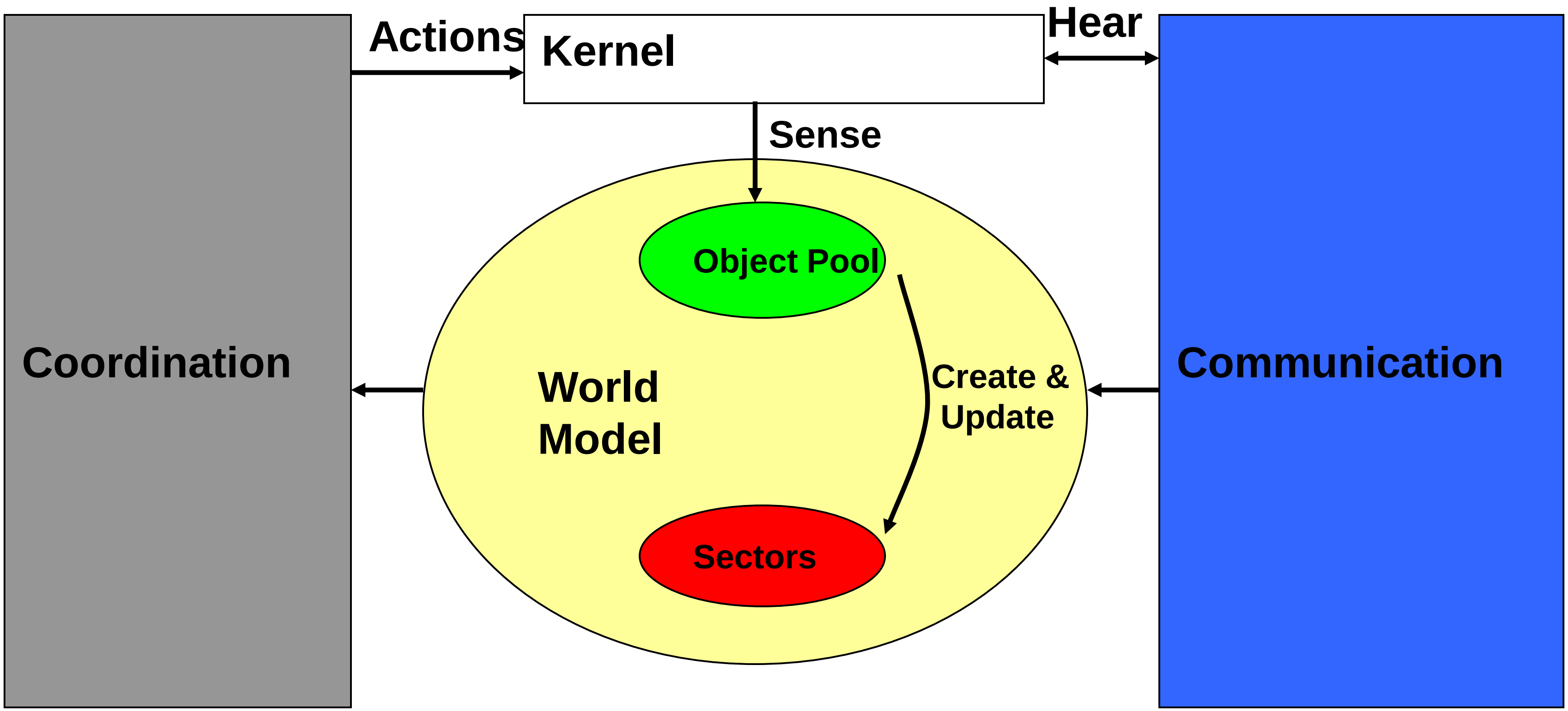
Challenge



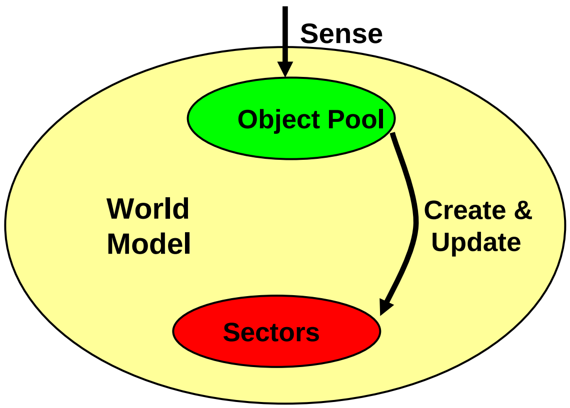
Disaster in a city (Kobe):

- **Buildings on fire**
- **Roads blocked**
- **Civilians buried**
- **Communication limited**

Overview – World Model



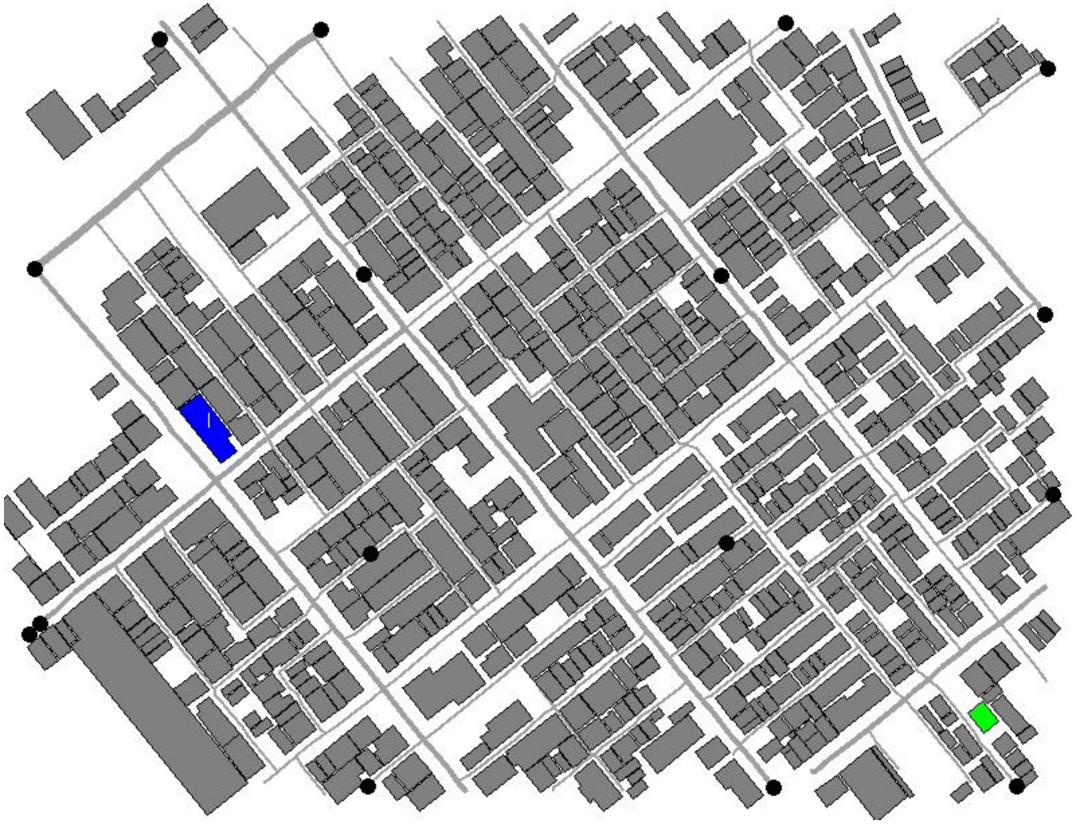
Sectors



- The sector map consist of
  - Highways
  - Pre computed paths
  - Object assignment
- Sector advantages
  - Communication language for summaries
  - Pre computed paths possible
  - Reduction in the number of possible actions

Creating Sectors - Kobe

Creating Sectors - Corner points

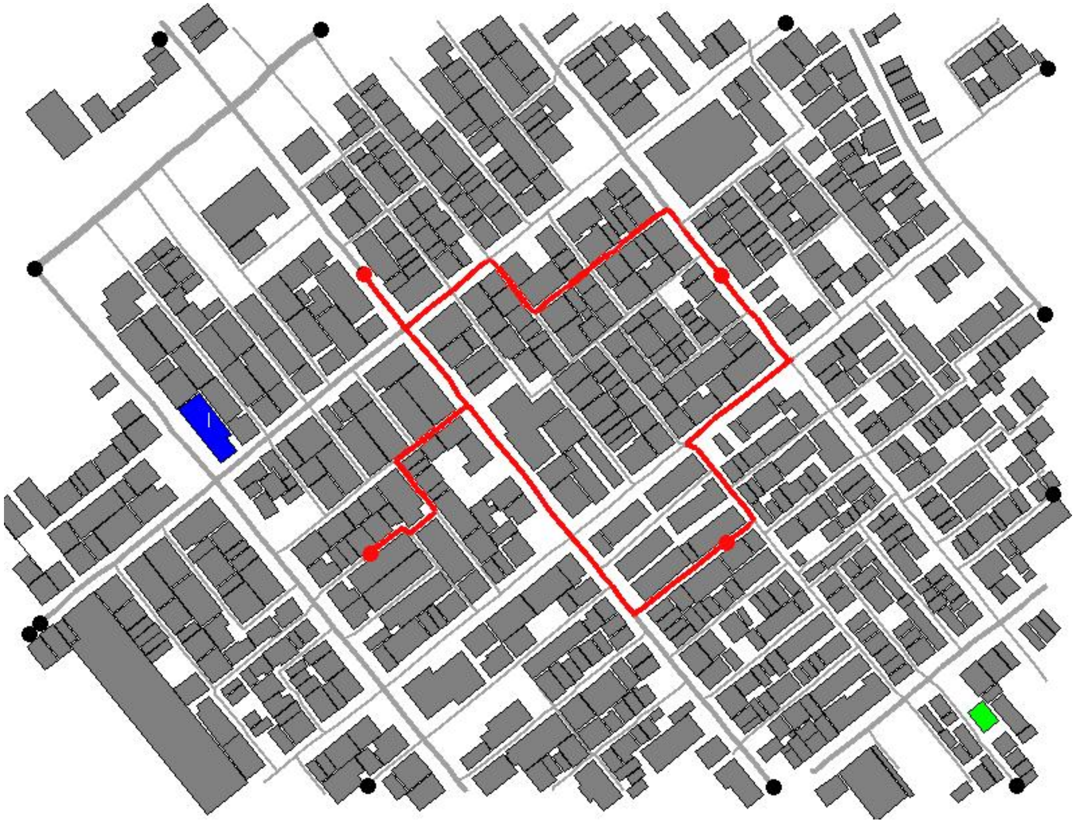


- Corner points are placed

Creating Sectors - Borders

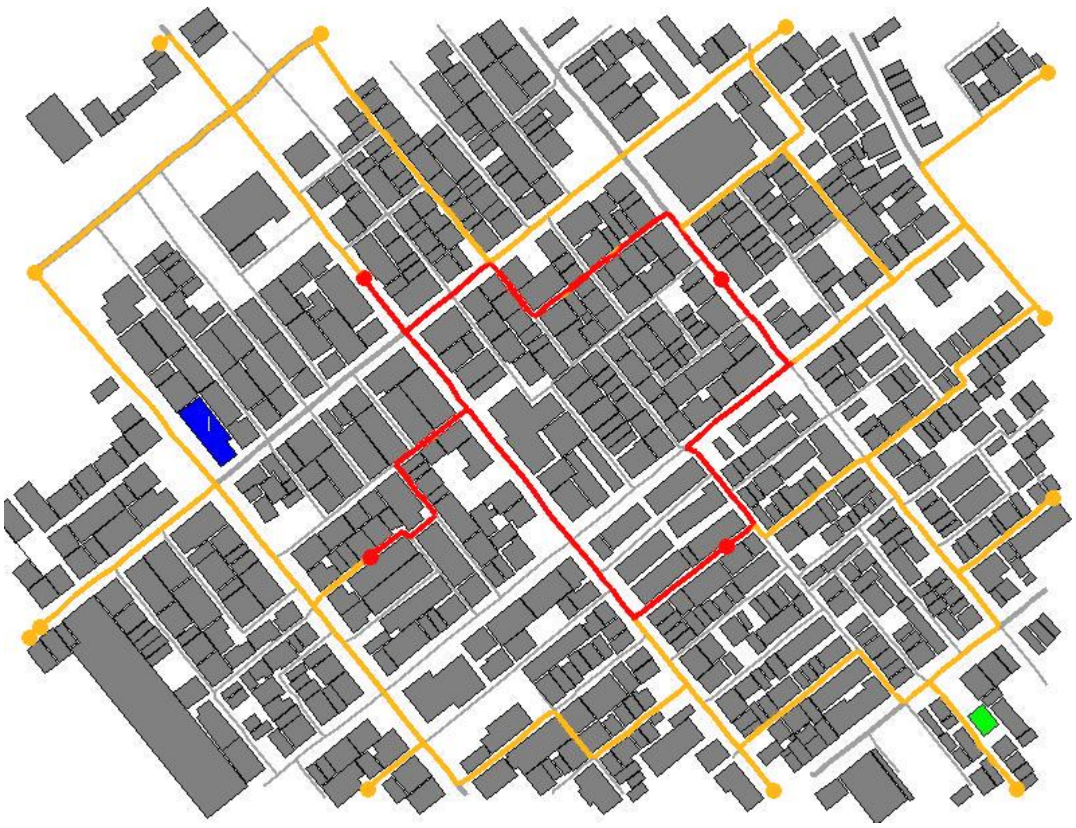
- Corner points are placed
- Paths between corner points

Creating Sectors - Polygons



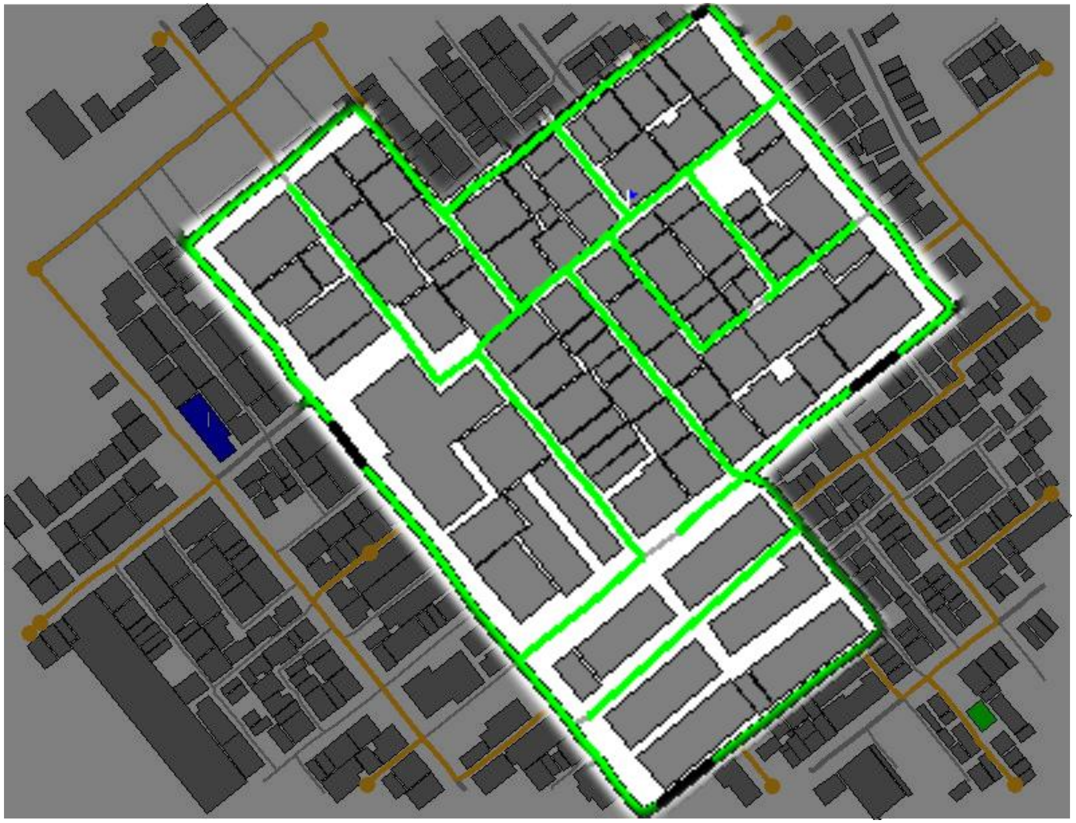
- **Corner points are placed**
- **Paths between corner points**
- **Paths form a concave polygon**

Creating Sectors - Sectors



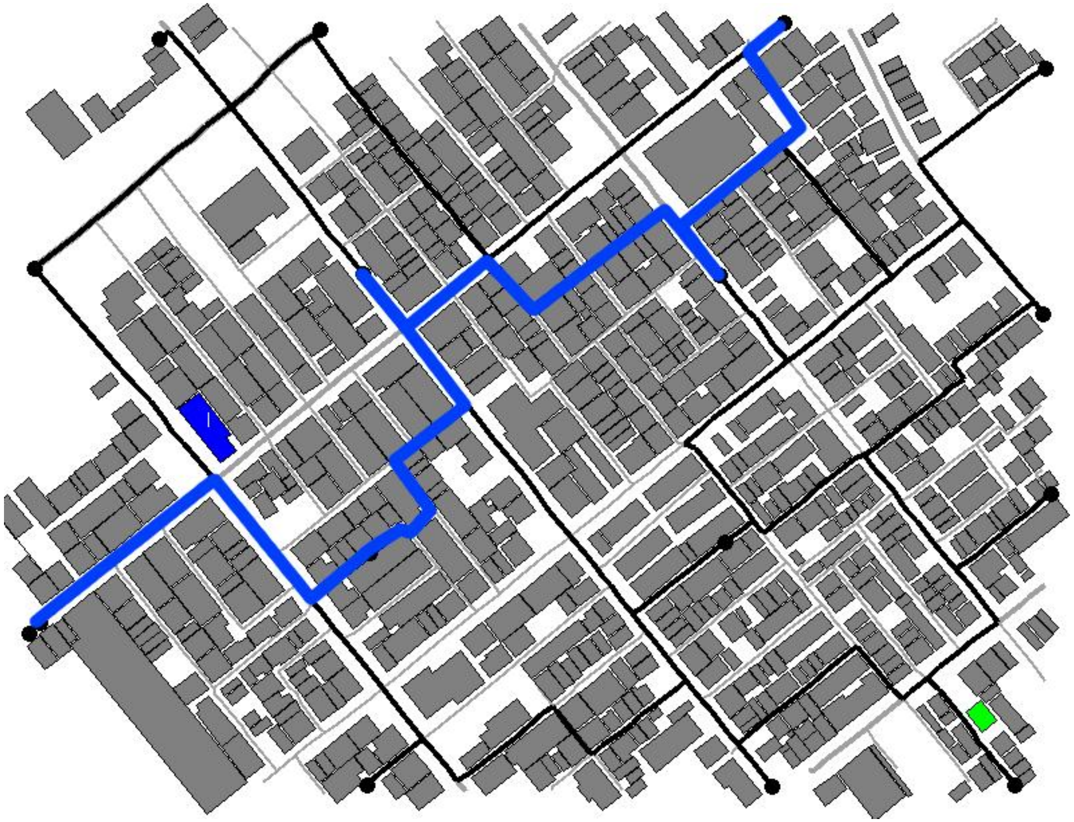
- Corner points are placed
- Paths between corner points
- Paths form a concave polygon
- Map is divided in polygons

Creating Sectors - Paths



- **Corner points are placed**
- **Paths between corner points**
- **Paths between form a concave polygon**
- **Map is divided in polygons**
- **Paths in each sector computed**

Creating Sectors - Highways



- Corner points are placed
- Paths between corner points
- Paths form a concave polygon
- Map is divided in polygons
- Paths in each sector computed
- Paths are concatenated into highways

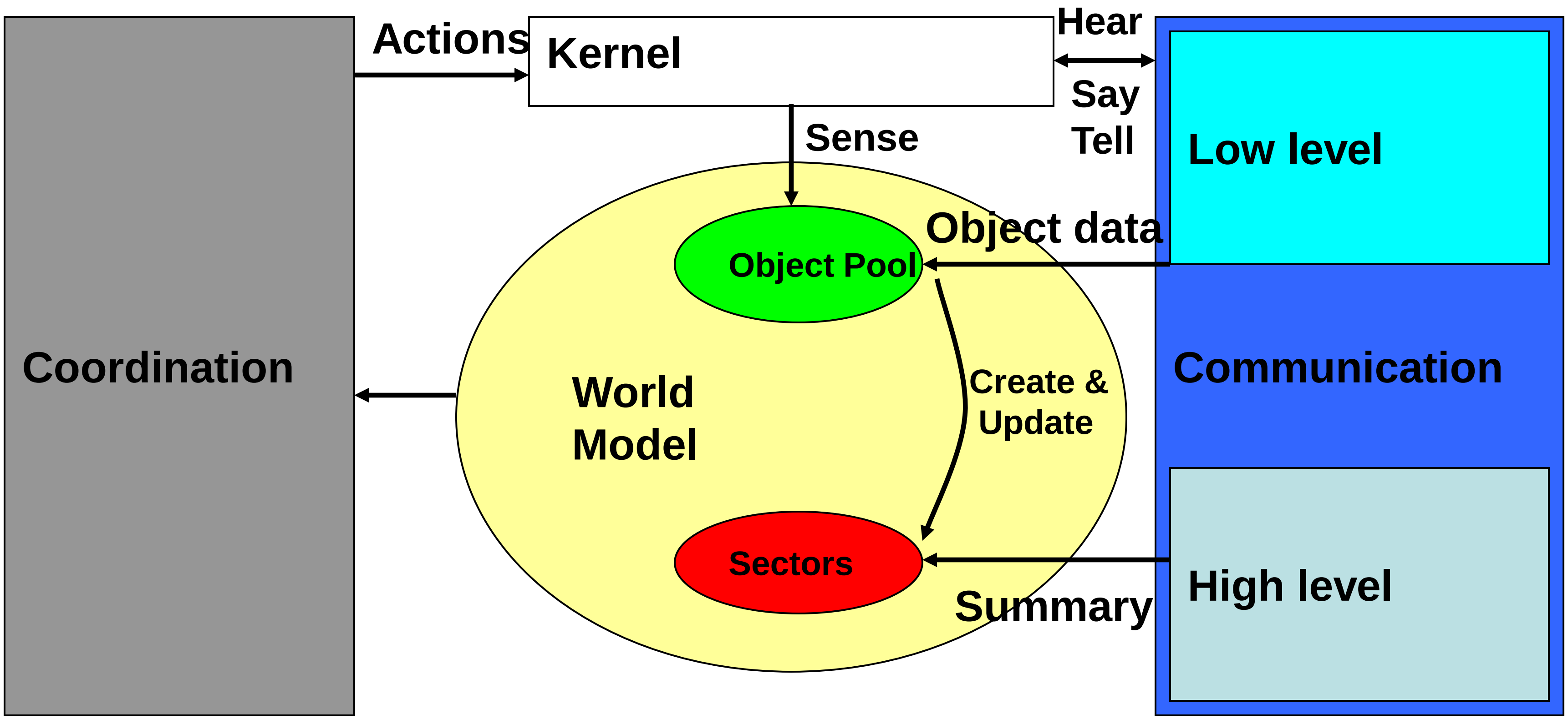
Summaries

- are sent using high level communication.
- are used for creating teams and tasks.
- are synchronized among agents.

Summaries

- consist of:
  - The amount of fire in a sector
  - Total road blockage in a sector
  - Structural damages in a sector
  - How much an agent knows about a sector
  - Whether highways are blocked or clear
  - Positions of all platoon agents within the map

Overview - Communication



Communication

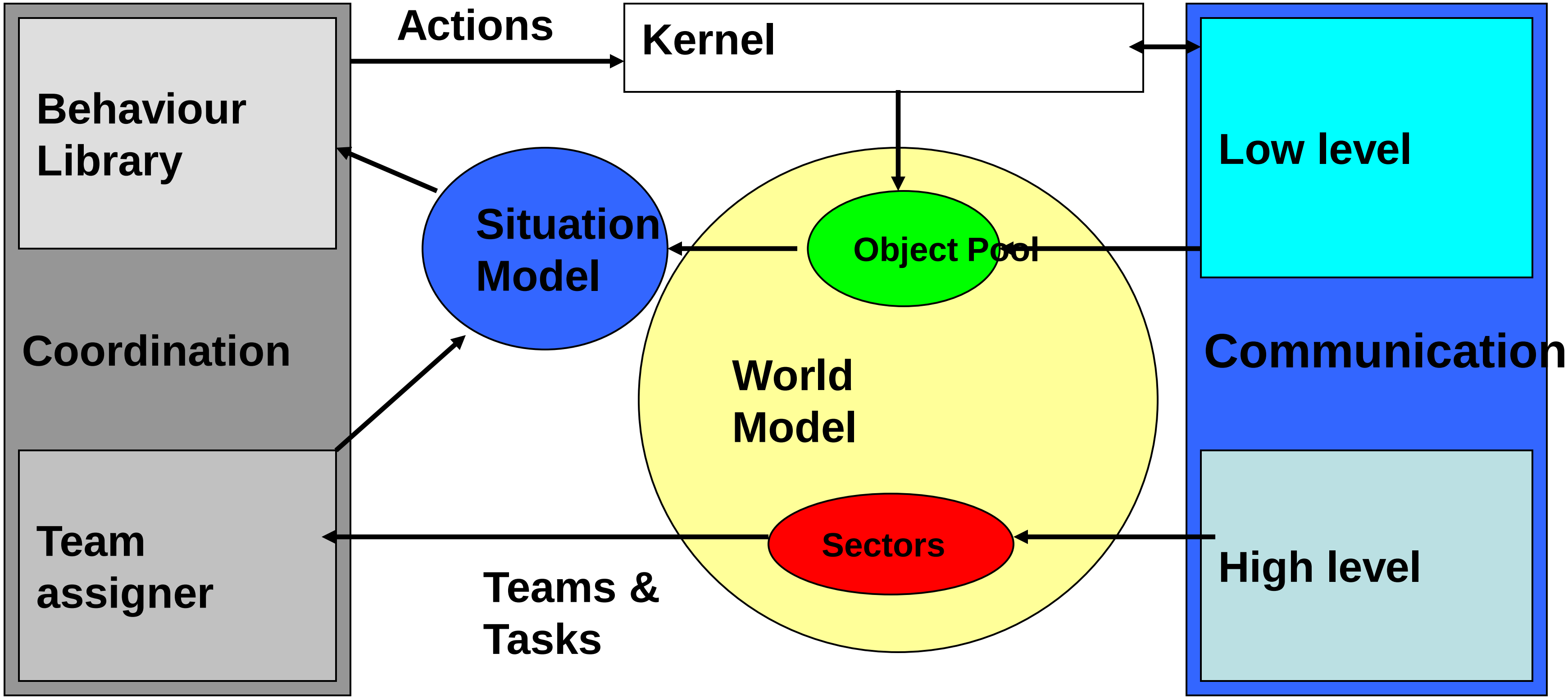
High level

- **Summaries**
- **Agents to centers and back**
- **Homogeneous**
- **Tell messages**
- **Synchronous, nothing lost**

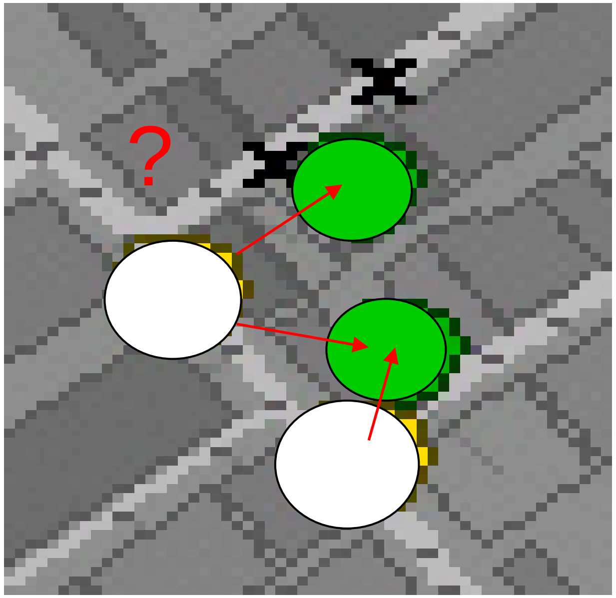
Low level

- **Object data**
- **Between agents**
- **Heterogeneous**
- **Say messages**
- **Say and pray**

Overview - Coordination

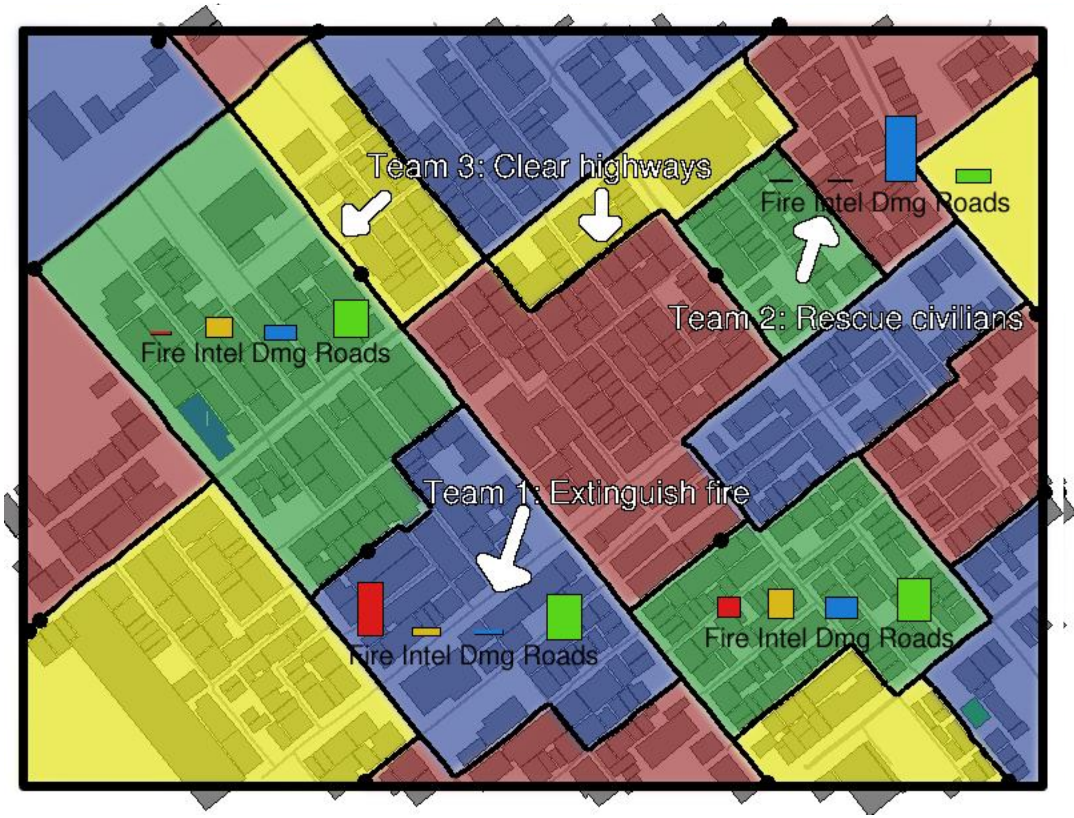


Coordination



- Coordination improves performance:
  - Fire brigades have to extinguish the same target
  - Ambulance agents have to select their own civilian
  - Police agents have to help others do their job

Tasks and Teams

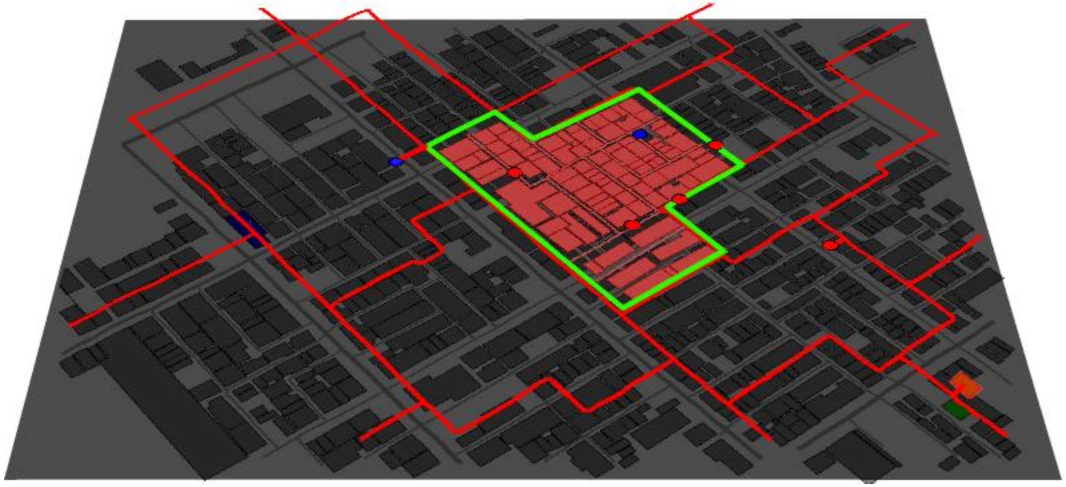


- Each agent and center divides the agents into teams.

- Each team is assigned to a task.
  - Common knowledge is used.

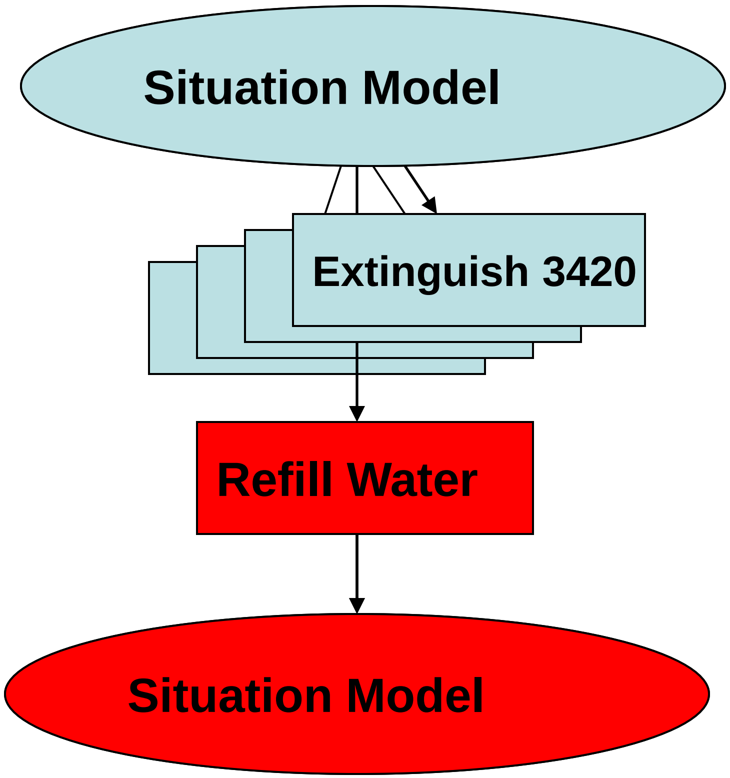
Situation Model

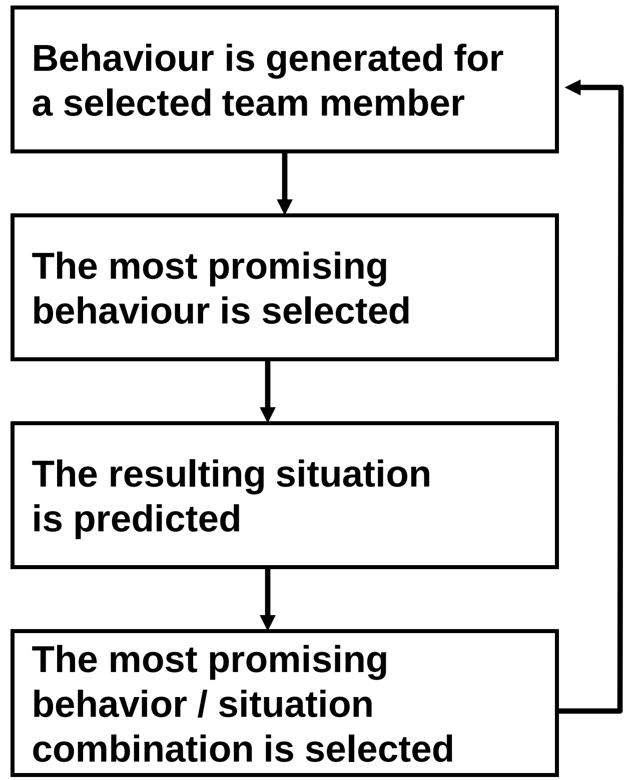
The common knowledge is a situation is extracted from the world model



- The sector is selected
- Sector contents
- Highways to sector
- Nearest refuge and path
- Team members and paths

Cooperative Game Tree





Game Tree Algorithm

$$
\begin{aligned}
&\mathbb{Y} = \mathbb{F}_{think}(\Omega, \tau) : \\
&\qquad \mathbb{G} = \mathbb{F}_{expand}(\Omega, \tau) \\
&\qquad \mathbb{S} = \{\mathbb{Y}_0 \ldots \mathbb{Y}_\epsilon\} = \mathbb{F}_{select}(\mathbb{G}) \\
&\qquad \mathbb{Y} = \mathbb{Y}_0
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{G} = \mathbb{F}_{expand}(\Omega, \tau) : \\
&\qquad foreach\ \mathbb{Y}_i \in \tau \\
&\qquad\qquad \Omega'_i = \mathbb{F}_{simulate}(\Omega, \mathbb{Y}x) \\
&\qquad\qquad c \in \Omega'_i = \epsilon? \\
&\qquad\qquad\qquad \mathbb{G}'_i = \{\mathbb{Y}_i, \Omega'_i\} \\
&\qquad\qquad c \in \Omega' \neq \epsilon? \\
&\qquad\qquad\qquad \tau'_i = \mathbb{F}_{responses}(\Omega'_i) \\
&\qquad\qquad \mathbb{G}'_i = \mathbb{F}_{expand}(\Omega'_i, \{\mathbb{Y}_i, \tau'_i\}) \\
&\qquad \mathbb{G} = \mathbb{G}'1 \cap \ldots \cap \mathbb{G}'_i
\end{aligned}
$$

game	length	branch factor	strategies
tic tac toe	5 to 9	9 to 1	26830 [2]
chess	38.58 [3]	35 [17]	$3.71 * 10^{59}$
RCR UvAC2003	300	1000^{25}	$1000^{25 \cdot 300} = 10^{22500}$

Computation Time

- Computation problems

  Each agent

  – has to select from a lot of actions.

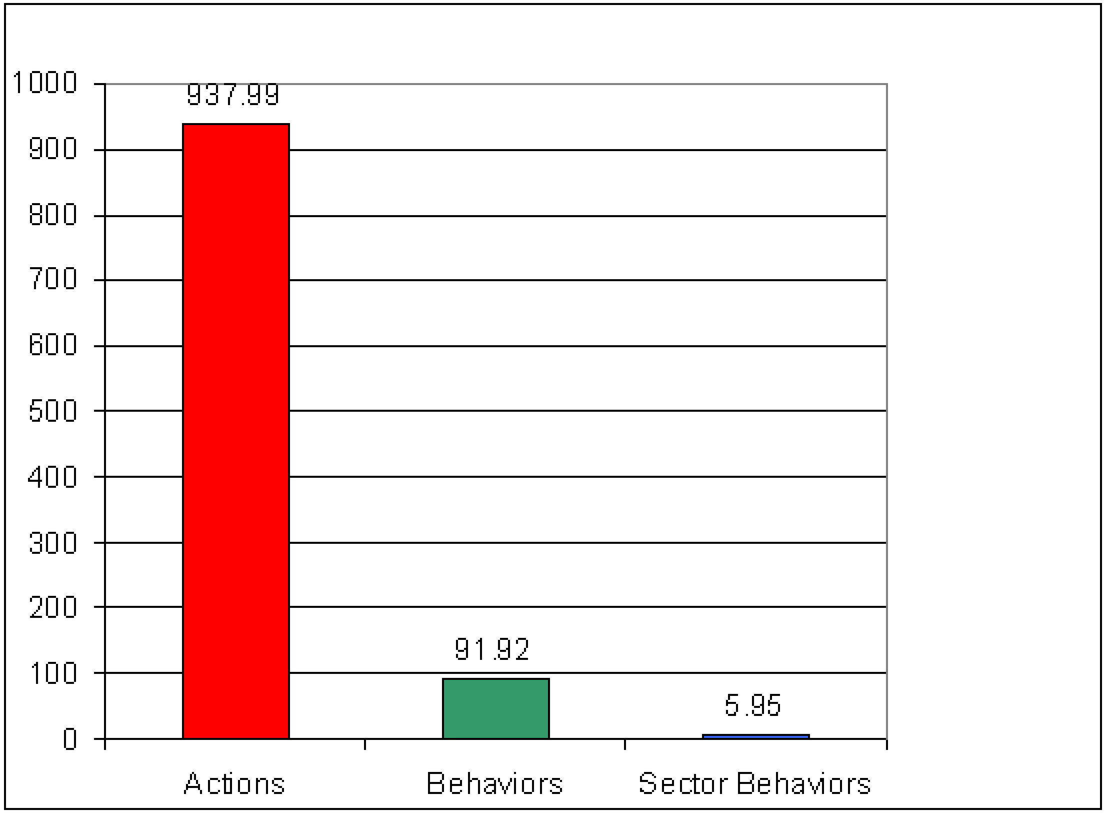
  – has to think for other team members.

  – has to think into the future.

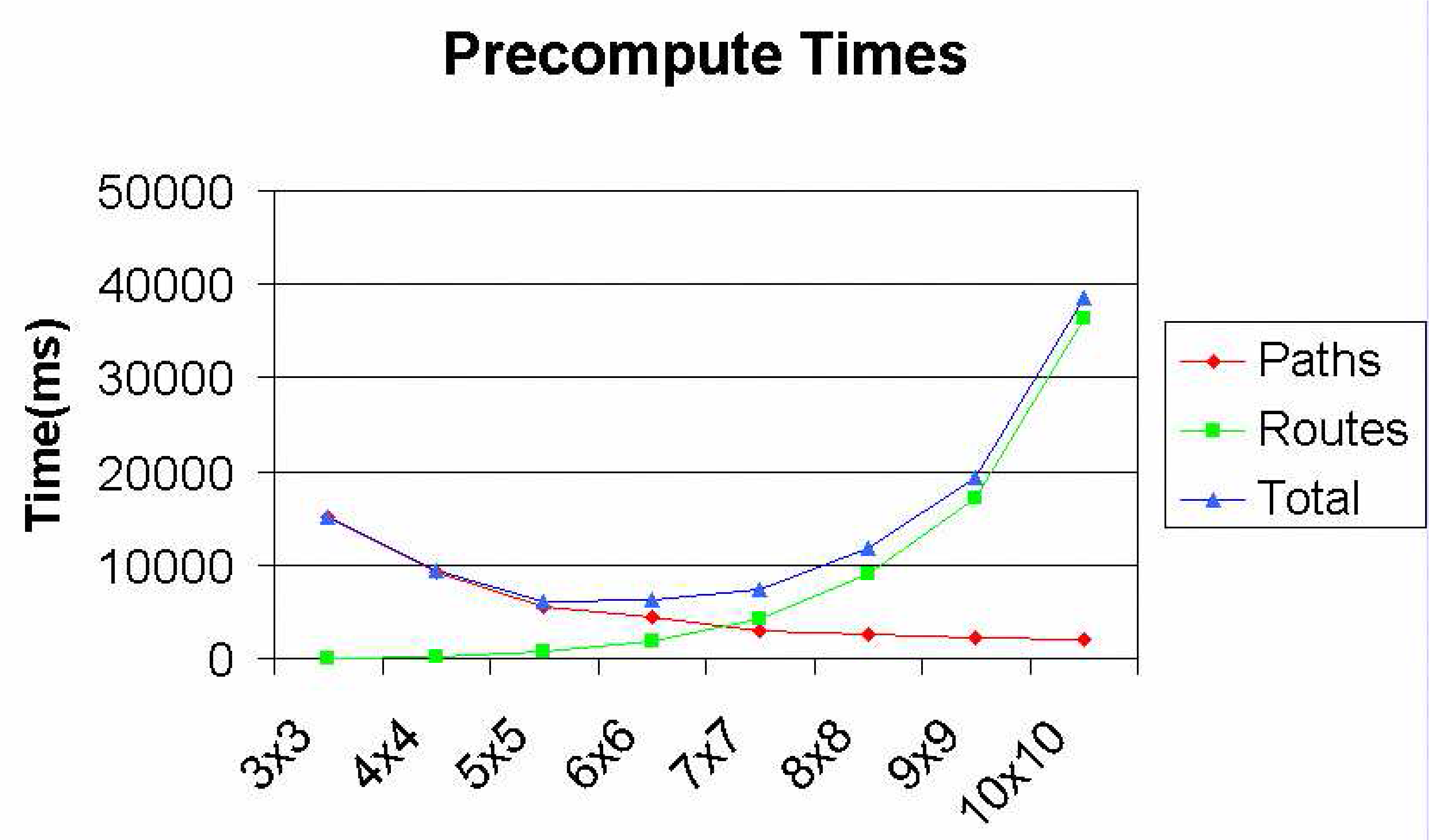
Computation Time

- Computation solutions
  - Actions grouped in behaviours.
  - Prediction uses precomputed paths.
  - Targets reduced by sectors.
  - Behaviours explored using heuristics.
  - Predictions stored using only the differences.

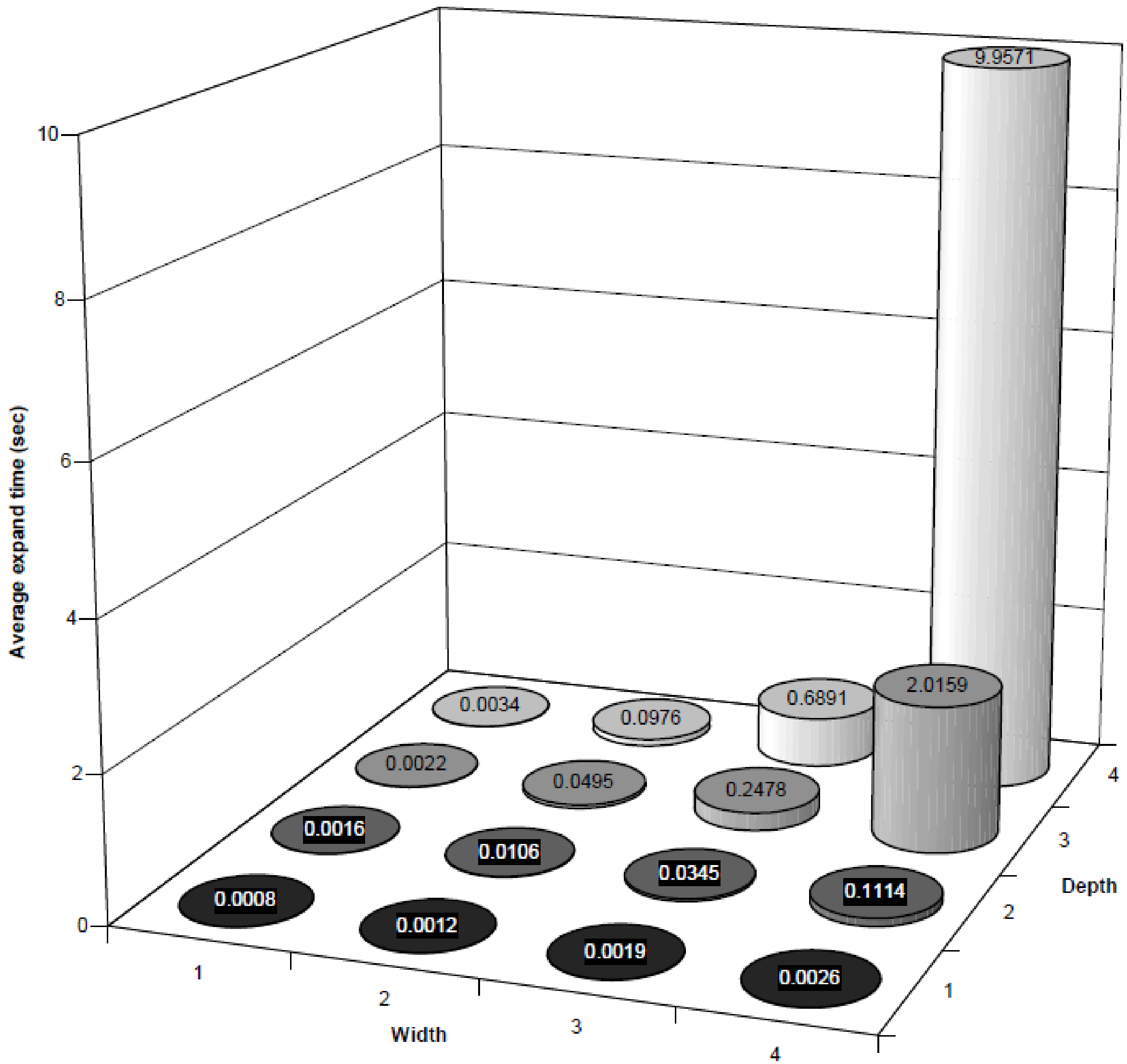
Reduction of Actions



Reduction of Prediction times



Results


Average expand time (sec)
0
2
4
6
8
10
9.9571
2.0159
0.6891
0.0976
0.0034
0.0022
0.0495
0.2478
0.0016
0.0106
0.0345
0.1114
0.0008
0.0012
0.0019
0.0026
Width
1
2
3
4
Depth
1
2
3
4

Lessons learnt

- Provide all agents with the same information, which allows them to work as a team
- Use common knowledge to come to a decision, instead of negotiation over the (multi-hop) communication channels
- Use game trees to reason about strategies and future situations (after reducing the problem)

RoboCup Rescue Virtual Robot Competition

Simultaneously solve:

- Where am I? (Localization)
- Where have I been? (Mapping)
- Find the victims (Exploration)
- Share this information and coordinate with others (Distributed Decision making)

The Challenge

Role-Based Multi-Robot Exploration: Using Mobile Relays in the Exploration of Communication-Limited Environments

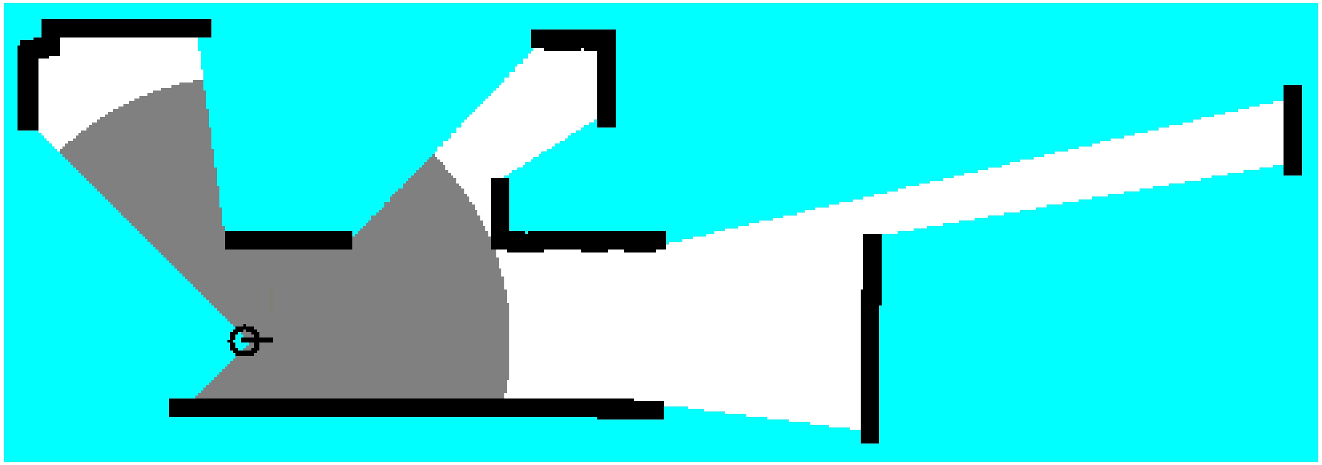
Julian de Hoog, Stephen Cameron, Arnoud Visser, A. Jimenez-Gonzalez, J. R. Martnez de-Dios and A. Ollero

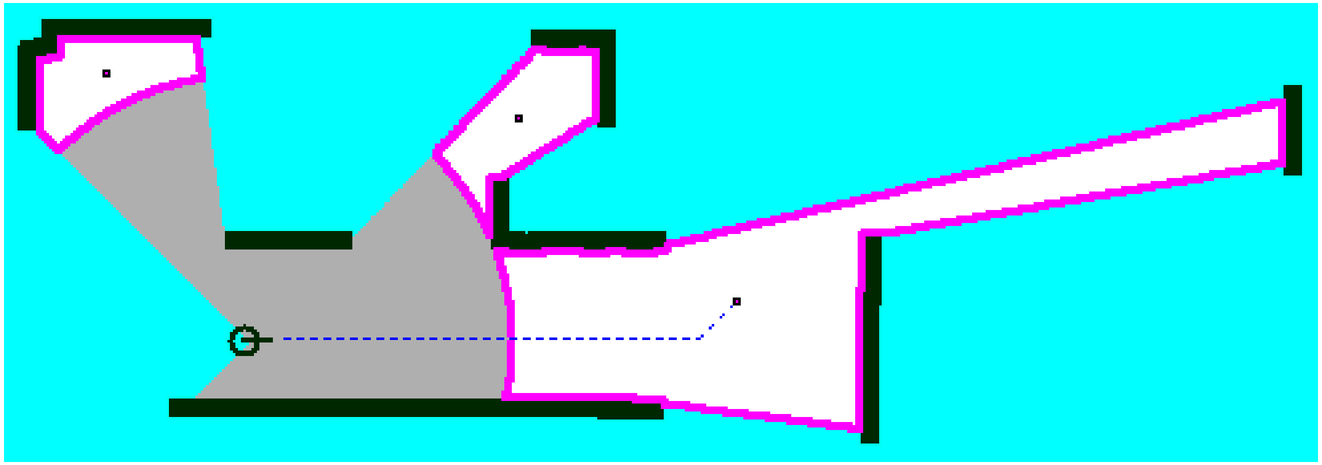
University of Oxford
Computing Laboratory

Universiteit van Amsterdam
Intelligent Systems Laboratory

University of Sevilla
Robotics, Vision and Control Group

Beyond Frontier Exploration





Exploration based on Utility

$$U(f) = Area(f)/dist(f)$$

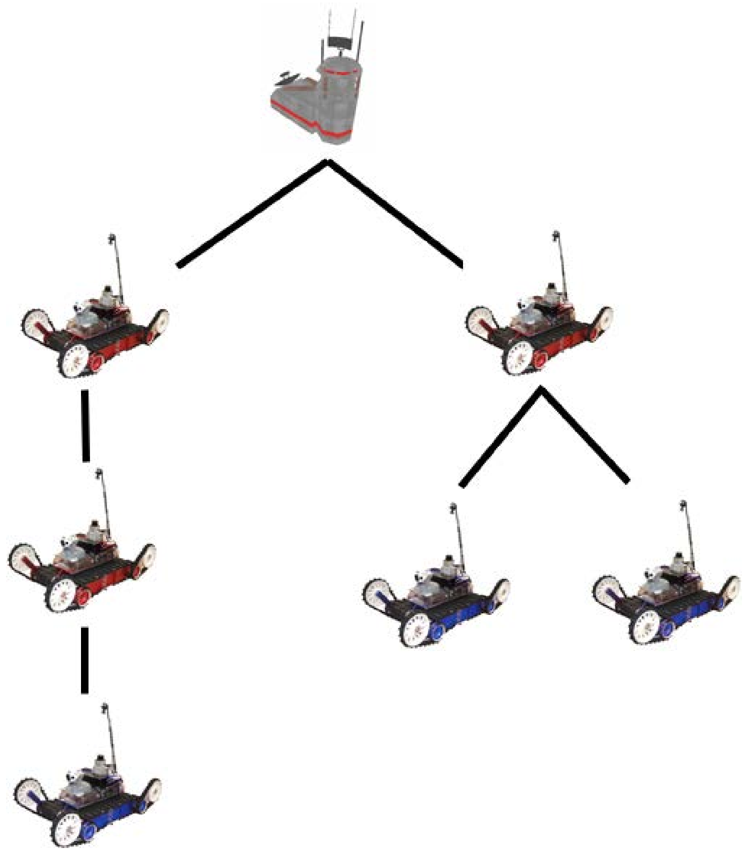
Input: Set R of explorers within range; Set F of frontier polygons
Output: List L of $\{r_i, f_j\}$ robot to frontier polygon assignments for $\forall r_i \in R$
Data: Q is a priority queue of all $\{r_i, f_j\}$ pairings, ordered by utility $U_{i,j}$

```
foreach r_i ∈ R do
    foreach f_j ∈ F do
        U_{i,j} =
        Area(f_j)/StraightLineDistance(r_i, f_j)^n
        Q.add({r_i, f_j})
    end
end
while not Q.isEmpty() do
    {r_a, f_b} = Q.pop()
    U_{a,b} = Area(f_b)/PathCost(r_a, f_b)^n
    if U_{a,b} > U(Q.peek()) then
        L.add({r_a, f_b})
        foreach {r_i, f_j} ∈ Q where i == a or j == b
        do
            Q.remove({r_i, f_j})
        end
    else
        Q.add({r_a, f_b}
    end
end
```

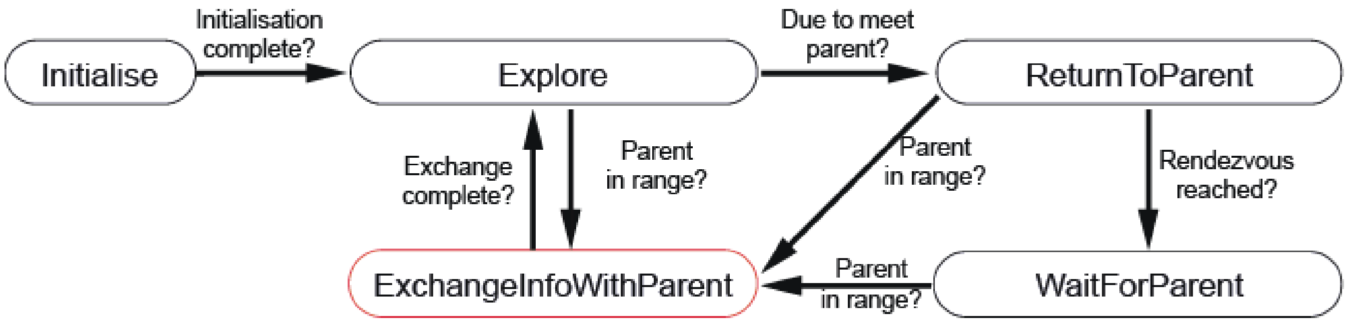
Arnoud Visser and Bayu A. Slamet, ‘Balancing the information gain against the movement cost for multi-robot frontier exploration’, 2nd European Robotics Symposium, Prague, March 26, 2008

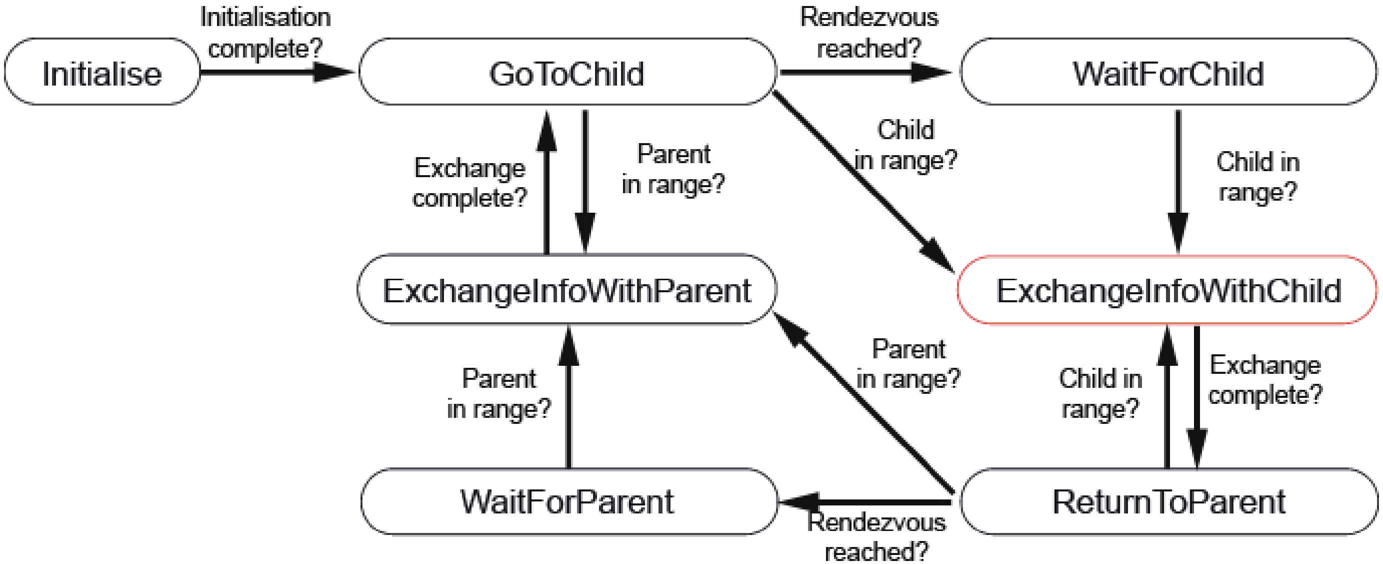
Planned Cooperation

Assistant communication relay roles have to be planned:

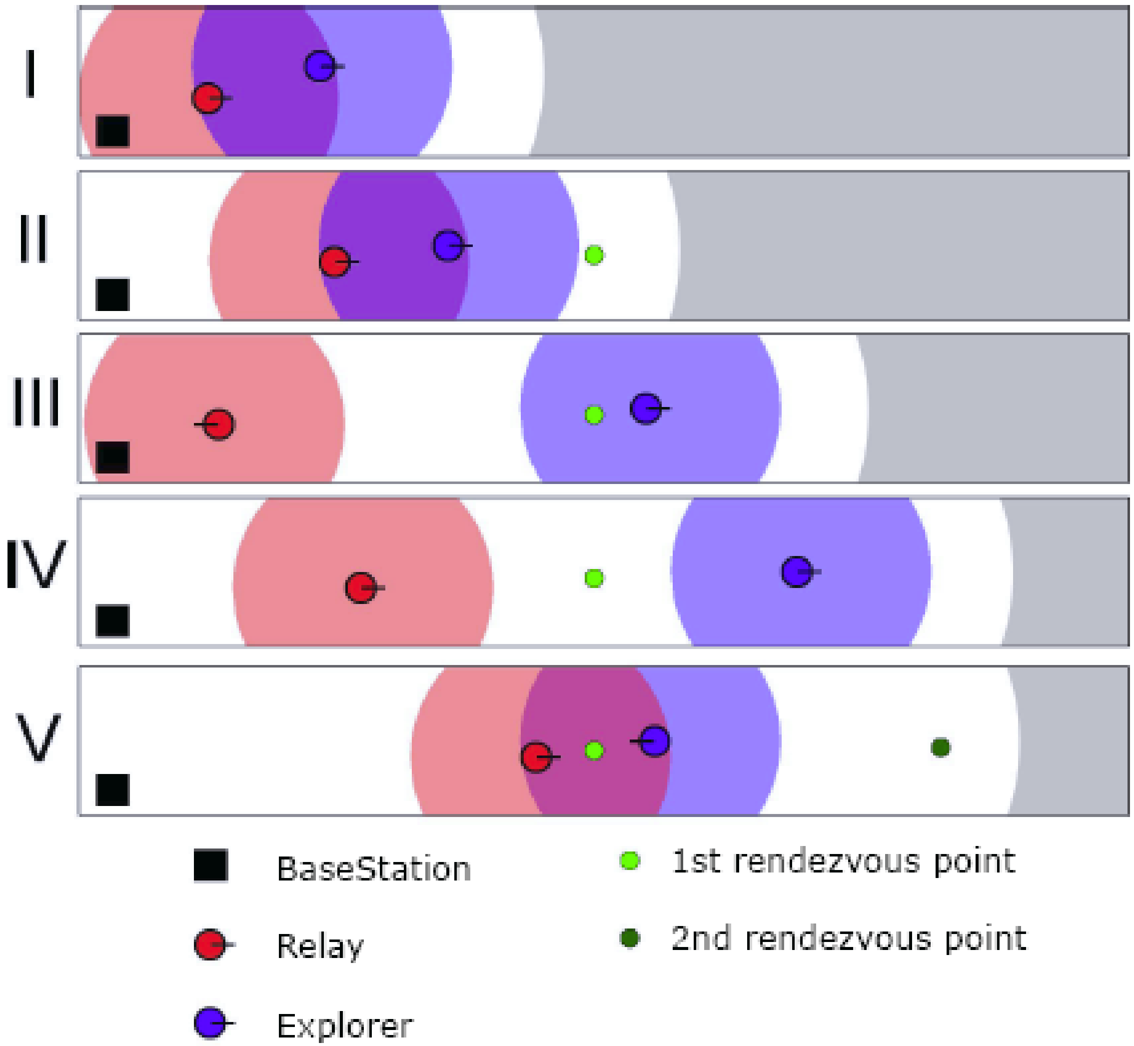


State Diagram of both roles

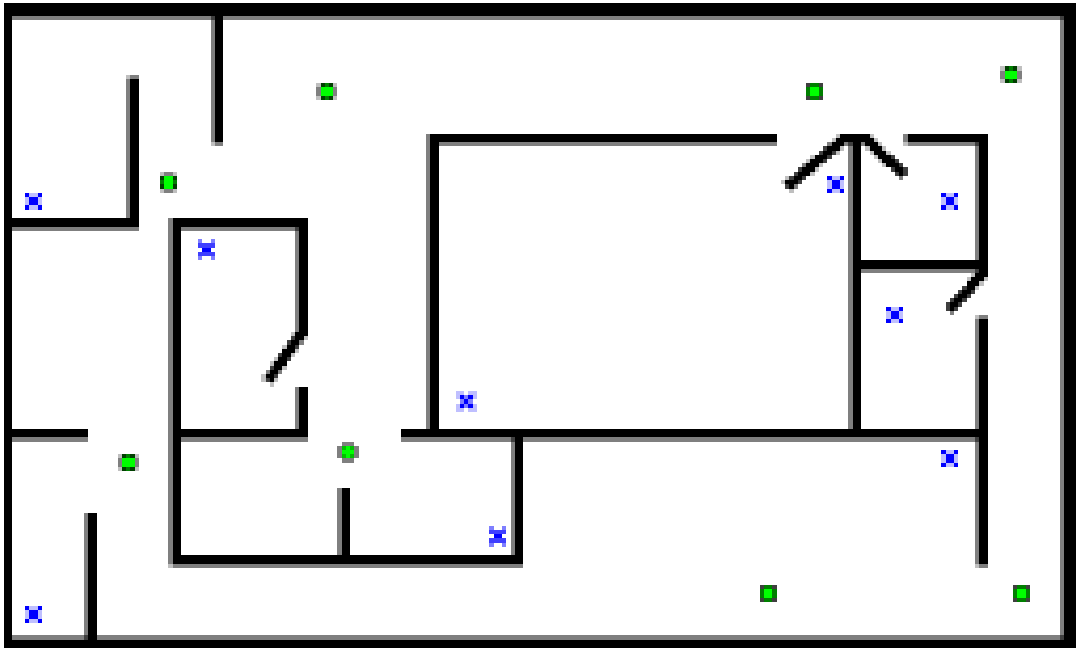




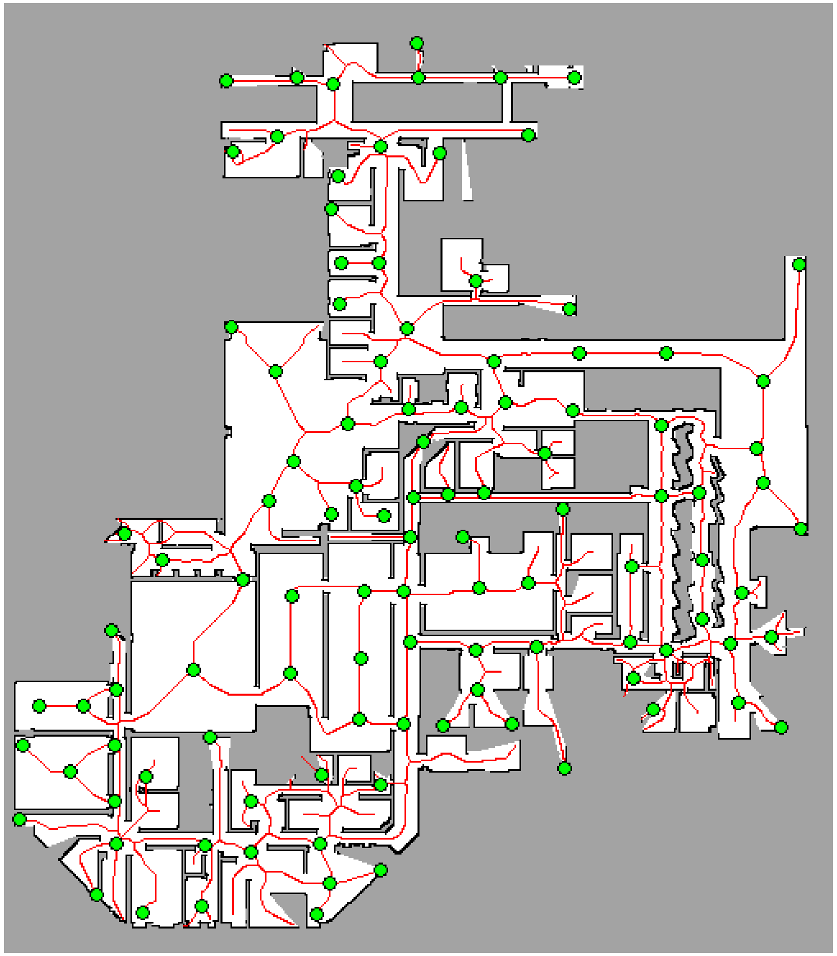
Demonstration of roles



Selection of rendezvous points

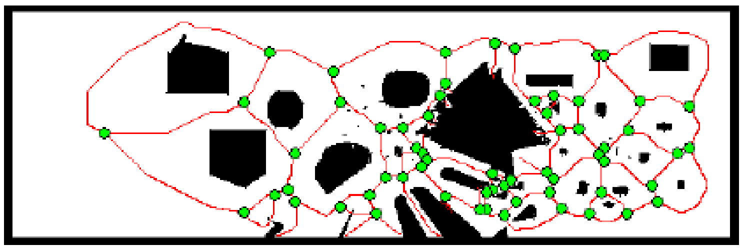


Selection based on a medial axis transform

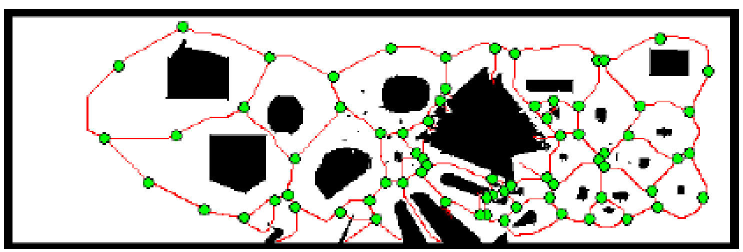


Also known as thinning, it gives better results than Voronoi diagrams.

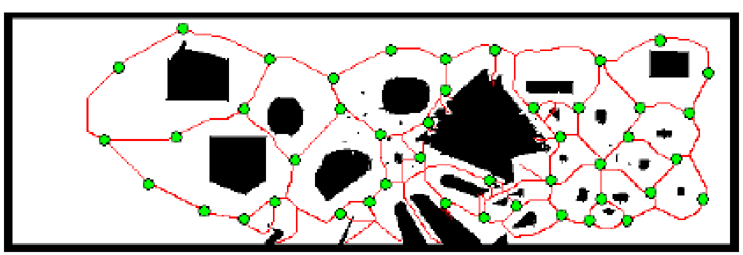
Optimize rendezvous set



(a) Step 1: Finding junctions



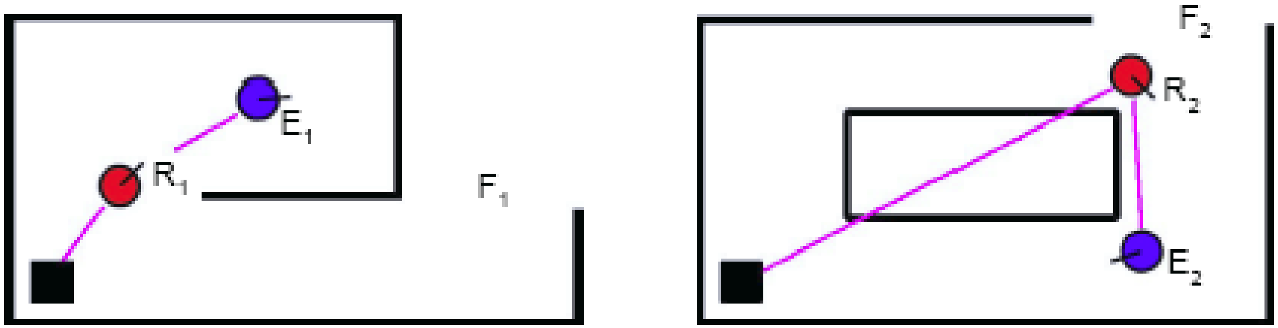
(b) Step 2: Filling



(c) Step 3: Pruning

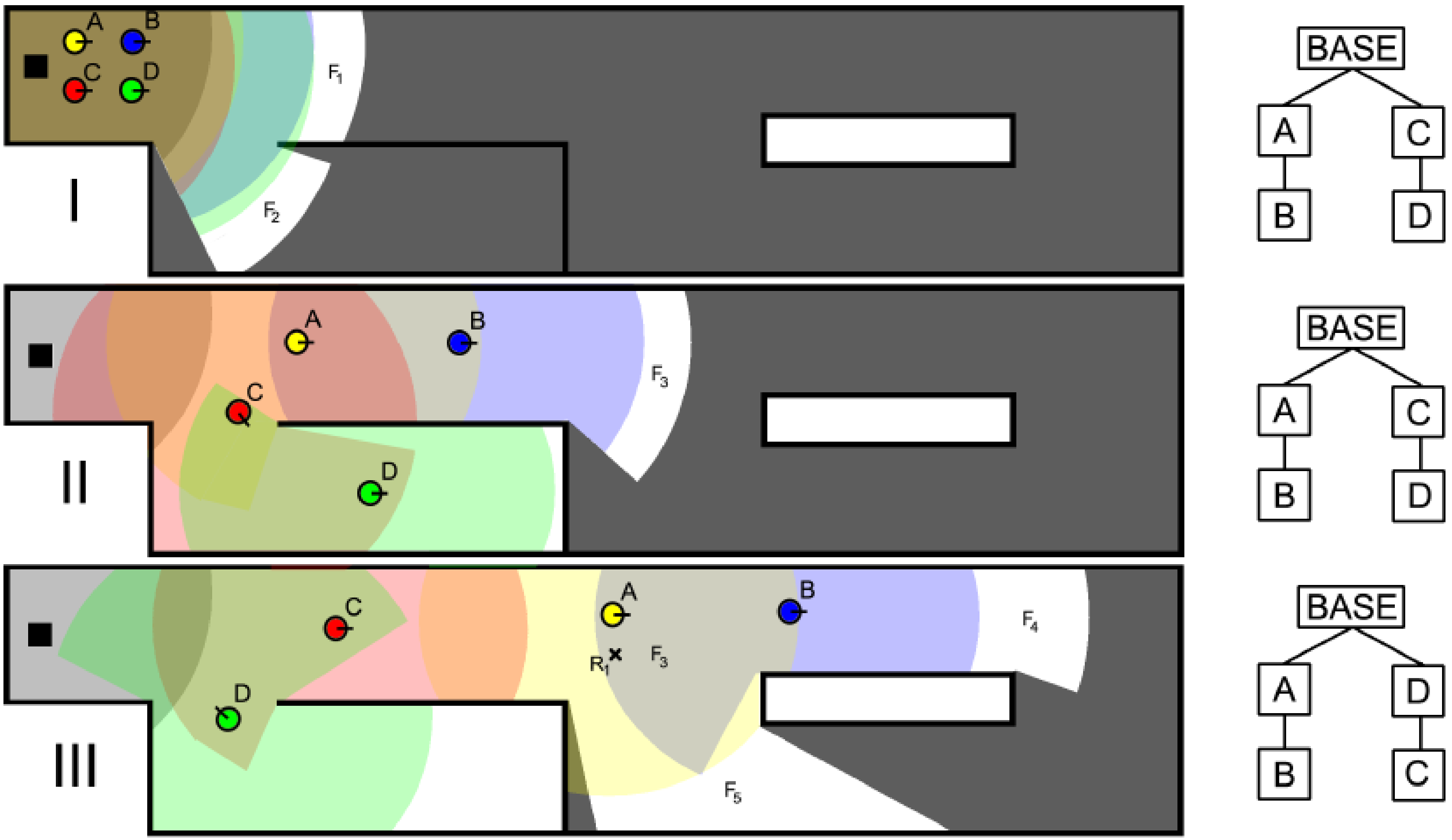
Input: The Explorer's map M; the Explorer's next frontier F
Output: The next rendezvous point r_{next}
Data: List of points S = hilditchThinning(M)
Data: List of points R (the list of candidate rendezvous points)
`// Step 1: Add junction points`
foreach $s_i \in S$ **do**
  if $neighbourTraversal(s_i) \geq 3$ **then**
    $R.add(s_i)$
  end
end
`// Step 2: Fill in extra points in open space`
foreach $s_i \in S$ **do**
  boolean $addToList = true$
  foreach $r_j \in R$ **do**
    if $distance(s_i, r_j) < threshold\ T_1$ **then**
      $addToList = false$
      $break$
    end
  end
  if $addToList$ **then**
    $R.add(s_i)$
  end
end
`// Step 3: Prune points that are too close`
foreach $r_i \in R$ **do**
  foreach $r_j \in R, i \neq j$ **do**
    if $distance(r_i, r_j) < threshold\ T_2$ **then**
      $R.remove(r_i)$
      $break$
    end
  end
end
`// Step 4: Choose the point closest to the Explorer's next frontier`
$r_{next} = R.pop()$
$d_{min} = distance(r_{next}, F)$
while not $R.isEmpty()$ **do**
  $r_{curr} = R.pop()$
  if $distance(r_{curr}, F) < d_{min}$ **then**
    $r_{next} = r_{curr}$
    $d_{min} = distance(r_{curr}, F)$
  end
end
$return(r_{next})$

Role Switching

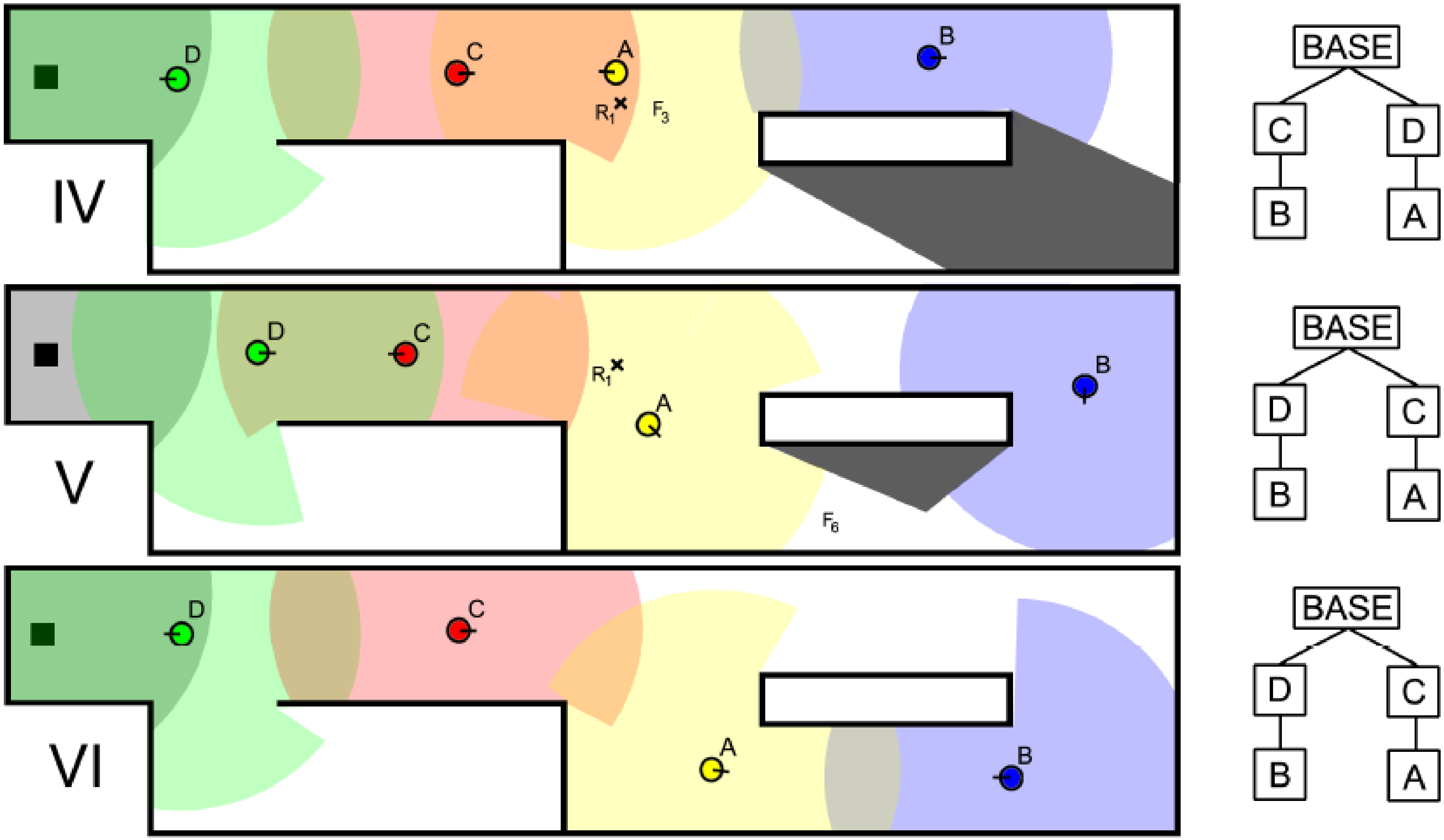


Two examples where it is beneficial to switch roles.

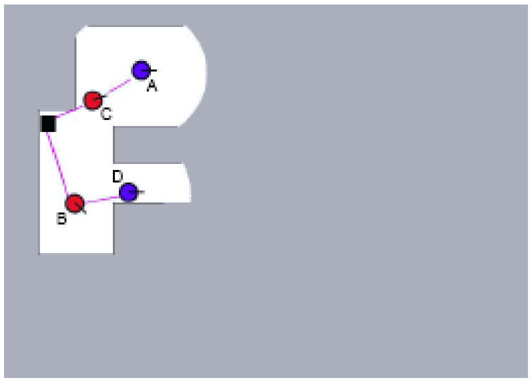
Role switching



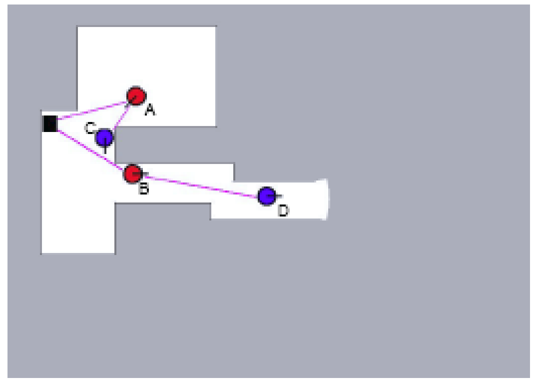
Role switching



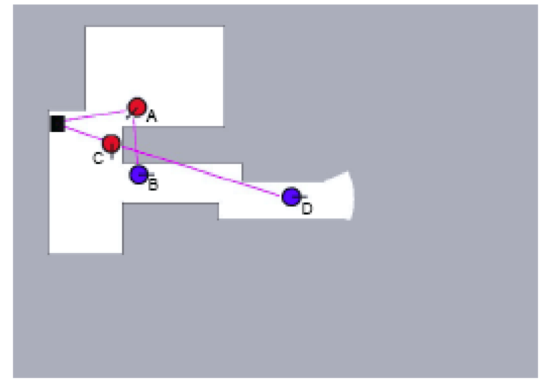
Role switching in practice



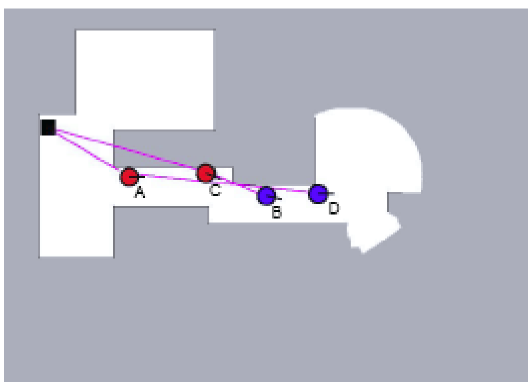
(a) After 41 time steps



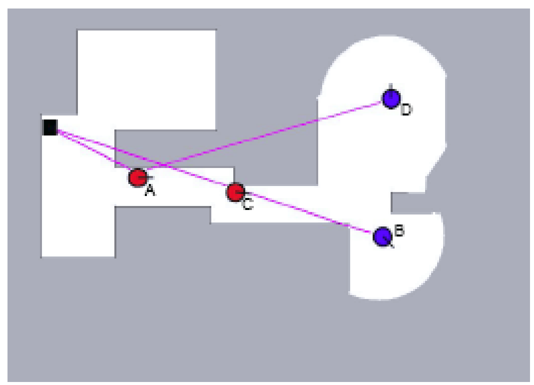
(b) After 96 time steps



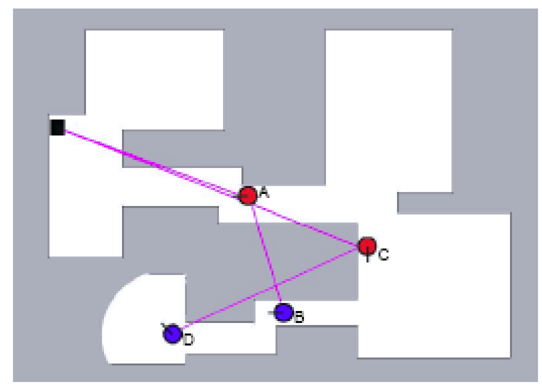
(c) After 101 time steps



(d) After 143 time steps

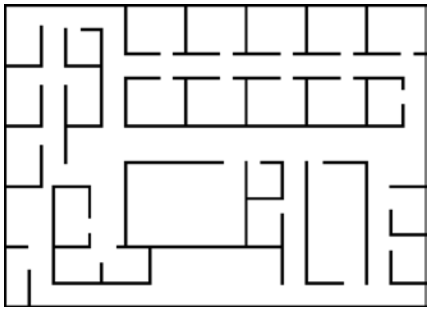


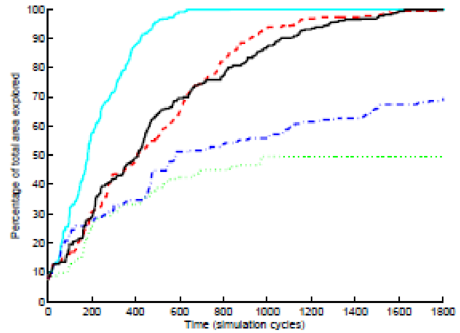
(e) After 178 time steps



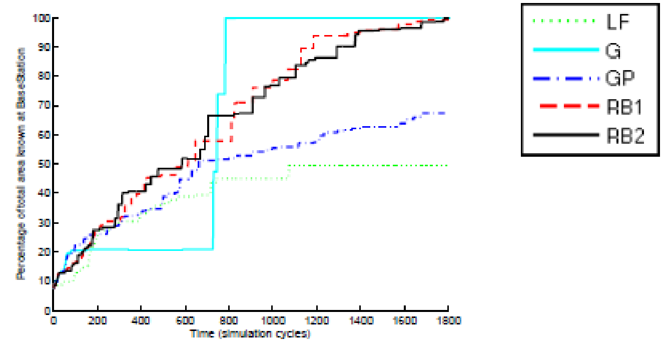
(f) After 392 time steps

Experiments





(a) Metric 1: Percentage of total area explored by full team

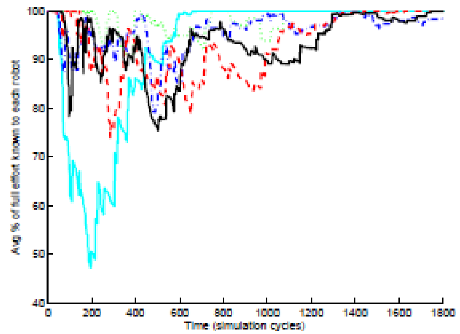


(b) Metric 2: Percentage of total area known at BaseStation

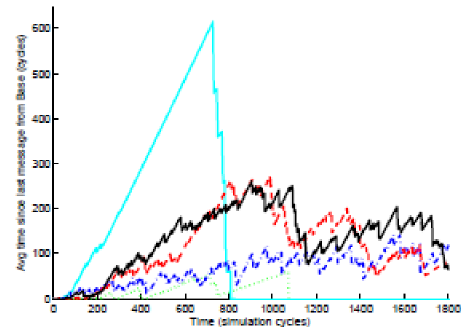
Leader-Follower Exploration (LF) performs best regarding metrics 3 (information sharing) and 4 (responsiveness).

Greedy Exploration (G) is by far the fastest method regarding metric 1 (exploring the full environment).

Role-Based Exploration (RB) provides the best performance for metric 2 (return of information to the BaseStation)

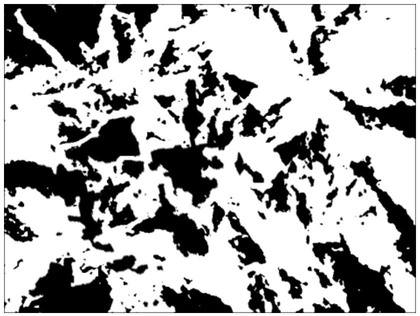


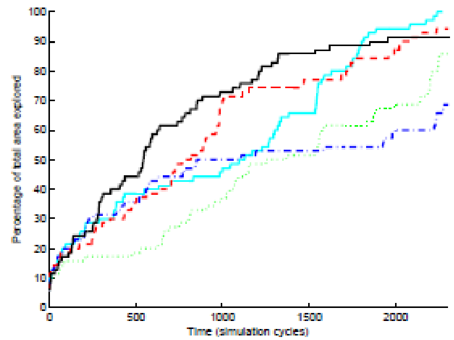
(c) Metric 3: Percentage of current team knowledge known to each robot, average over all robots (information sharing)



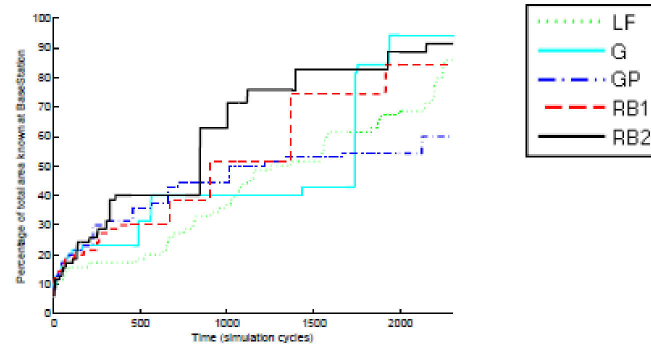
(d) Metric 4: Time since last message received from BaseStation, average over all robots (responsiveness)

Experiments





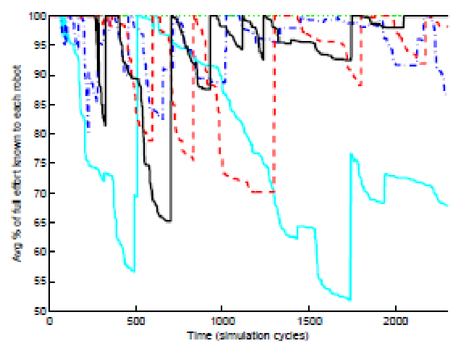
(a) Metric 1: Percentage of total area explored by full team



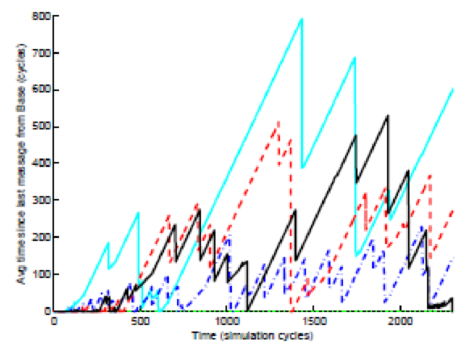
(b) Metric 2: Percentage of total area known at BaseStation

Role-Based Exploration (RB) outperformes Greedy Exploration (G), even for metri 1 (exploring the full environment).

Role Switching (RB2) is now effective (compared to RB1).

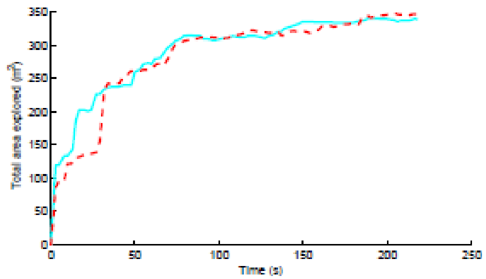


(c) Metric 3: Percentage of current team knowledge known to each robot, average over all robots (information sharing)

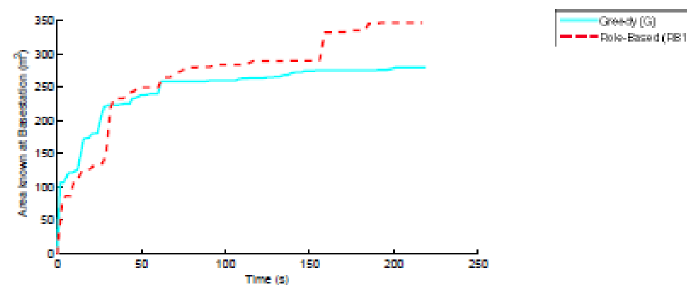


(d) Metric 4: Time since last message received from BaseStation, average over all robots (responsiveness)

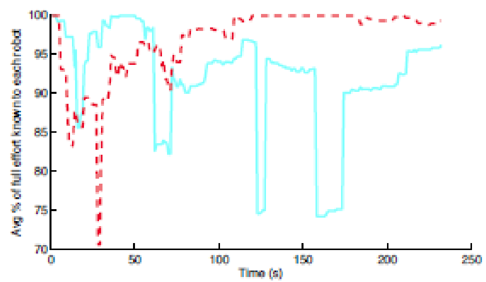
Real Experiments



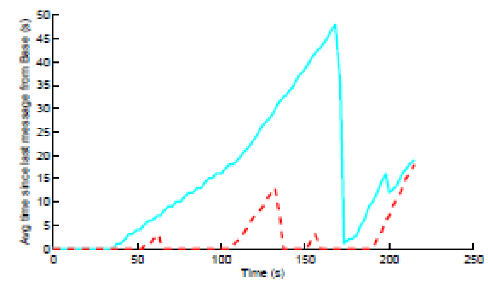
(a) Metric 1: Total area explored by full team



(b) Metric 2: Total area known at BaseStation



(c) Metric 3: Percentage of current team knowledge known to each robot, average over all robots (information sharing)



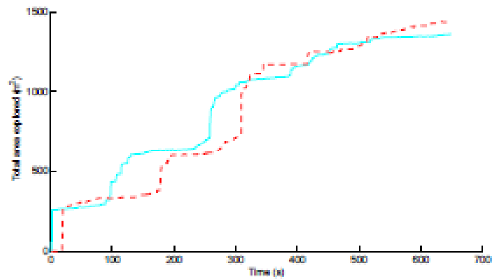
(d) Metric 4: Time since last message received from BaseStation, average over all robots (responsiveness)

Role-Based Exploration (RB) catches up fast on Greedy Exploration (G), even for metric 1 (exploring the full environment).

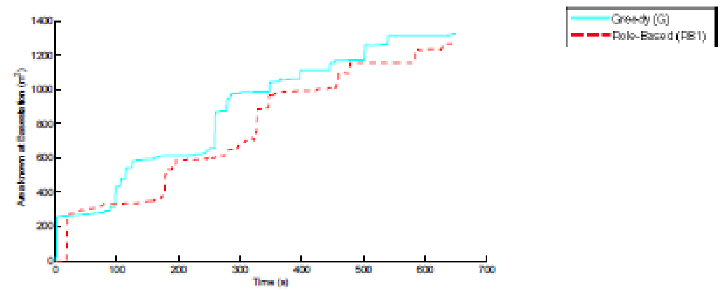
On metric 4 Greedy Exploration is completely dominated by RB.

Here the communication range was artificial reduced to 8m.

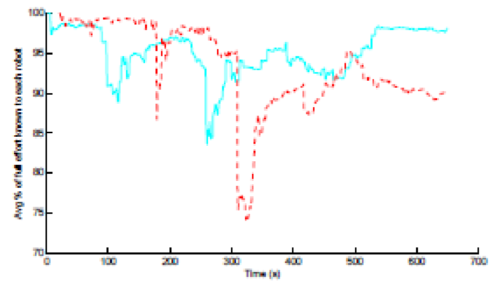
Real Experiments



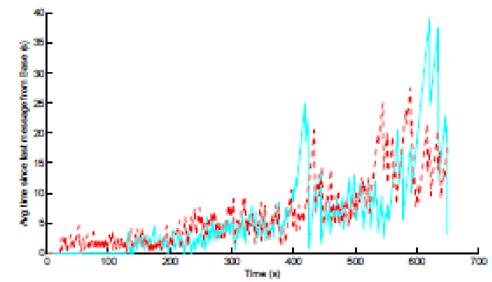
(a) Metric 1: Total area explored by full team



(b) Metric 2: Total area known at BaseStation



(c) Metric 3: Percentage of current team knowledge known to each robot, average over all robots (information sharing)



(d) Metric 4: Time since last message received from BaseStation, average over all robots (responsiveness)

It is more difficult for Role-Based Exploration (RB) to catches up on Greedy Exploration (G).

Even for metric 4 Greedy Exploration can keep up with RB.

This result was due because the communication range was not artificial reduced, so relatively the environment was smaller.

Overall Results

Lessons learnt

Cooperation & coordination is essential in USAR scenarios.

Role-Based Exploration should be used when:

- the Base-Station needs to receive quick information updates
- the communication range of the robots is small compared with the size or complexity of the environment.
- there are a large numbers of robots in the team (four or more). This is particularly true in environments with fewer frontiers.

RoboCup 2006 Bremen-Germany
3rd place

RoboCup Atlanta 2007
4th place

RoboCup German Open
2nd place

Brazil Open
1st place

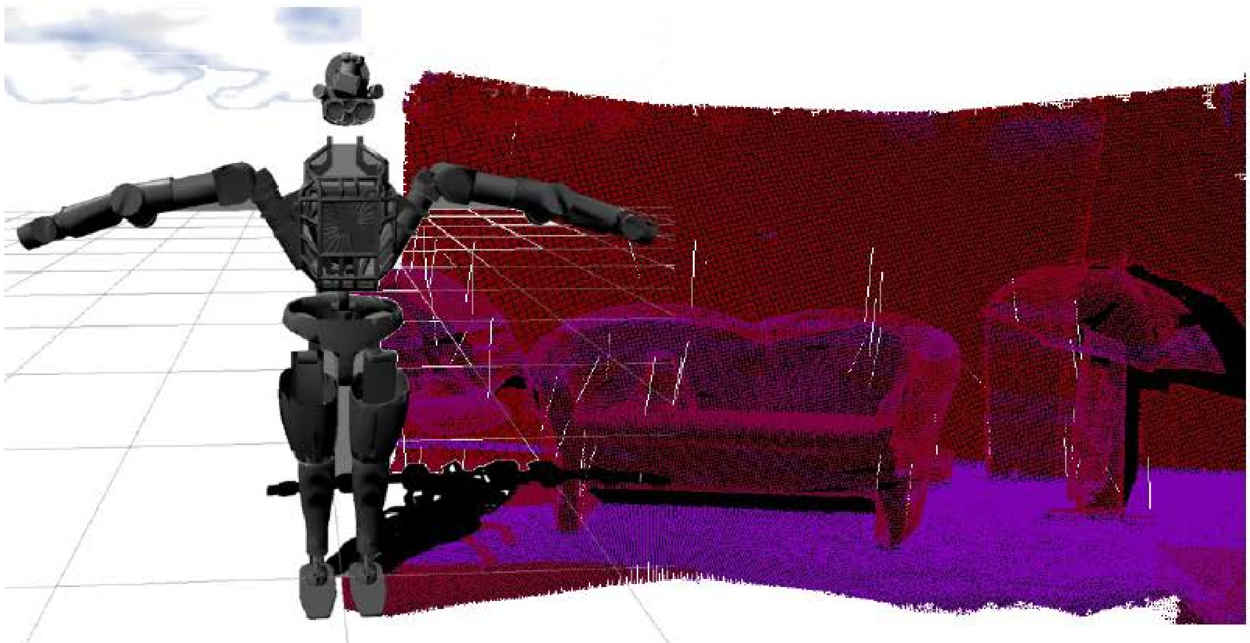
RoboCup 2009 Graz • Austria
3rd place

Iran Open 2010
Development price

Iran Open 2011
3rd place

RoboCup 2012 18 - 24 June Mexico City
Infrastructure price

Analysis of flat terrain for the Atlas robot



UvA Rescue

RoboCup IranOpen 2013 competition
Tehran, April , 2013

Universiteit van Amsterdam

Intelligent Systems Laboratory

DARPA Challenge 2005

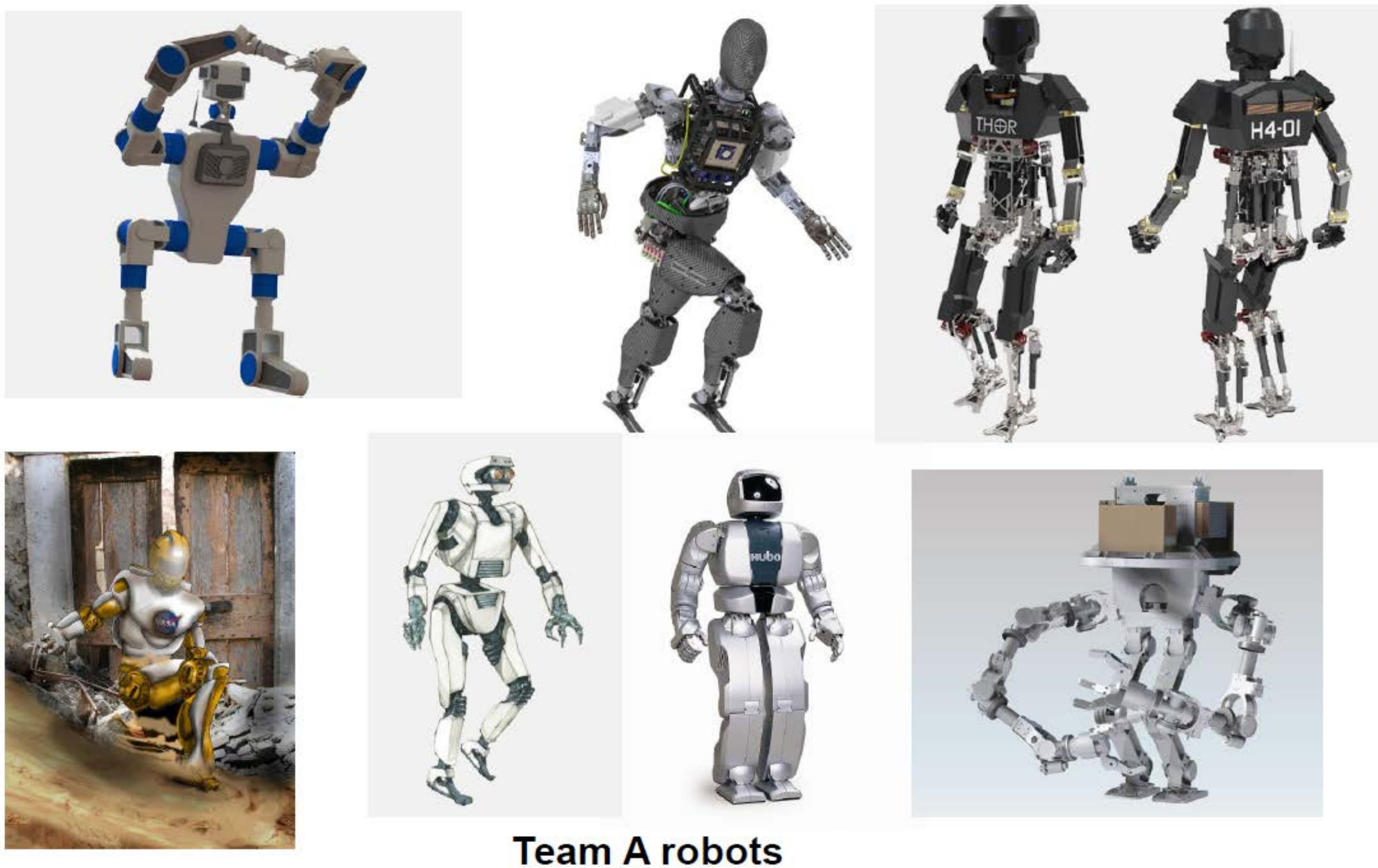
DARPA Urban Challenge

Urmson, C.; Whittaker, W., "Self-Driving Cars and the Urban Challenge," *Intelligent Systems, IEEE* , vol.23, no.2, pp.66,68, March-April 2008

DARPA Challenge 2013

http://www.theroboticschallenge.org/

DARPA Challenge 2013



Team A robots

[Meet the teams]

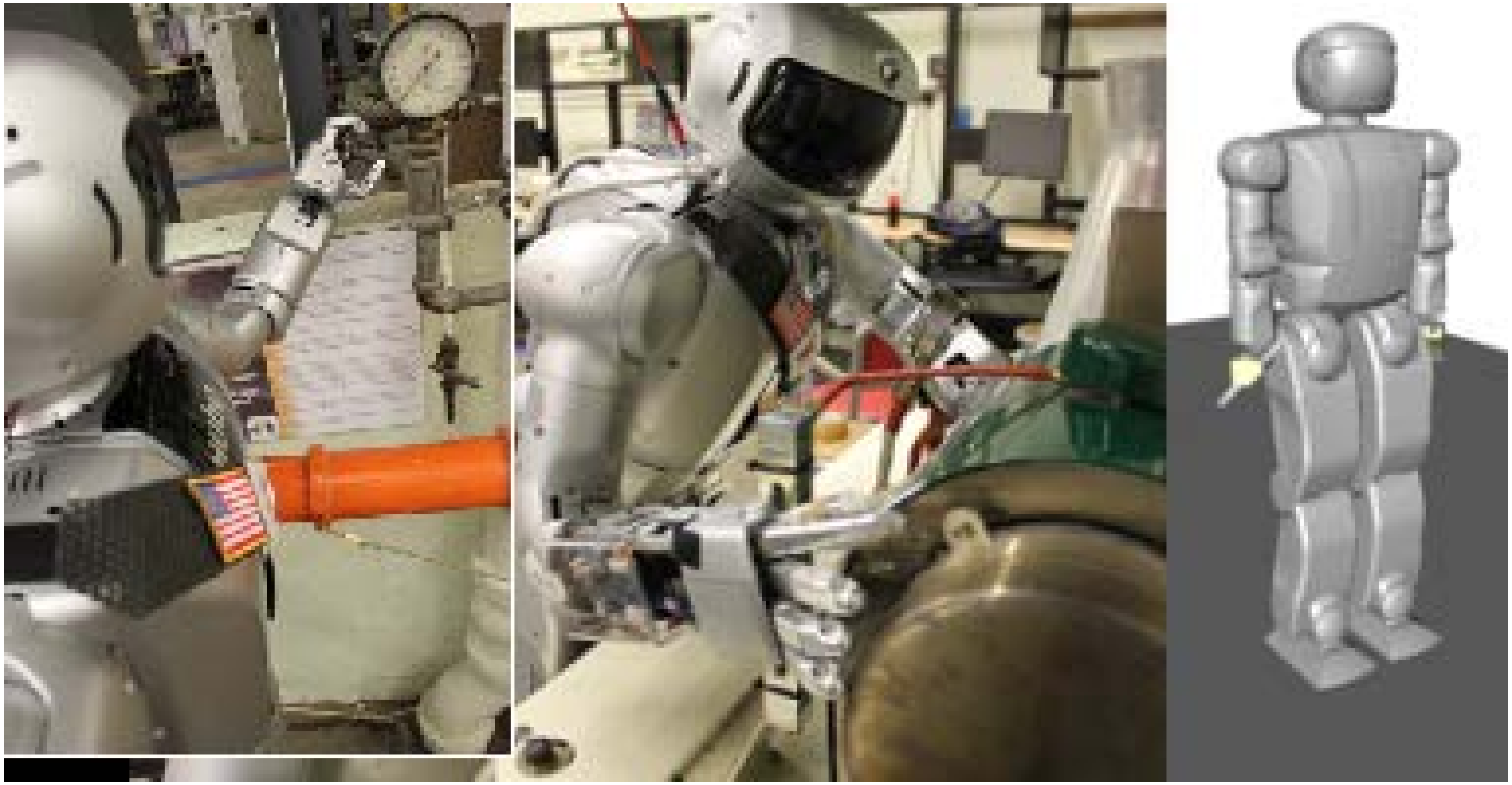
DARPA Challenge 2013

- [Hubo Vision on competition]

DARPA Challenge 2013

- [Hubo Vision on competition]

DARPA Challenge 2013



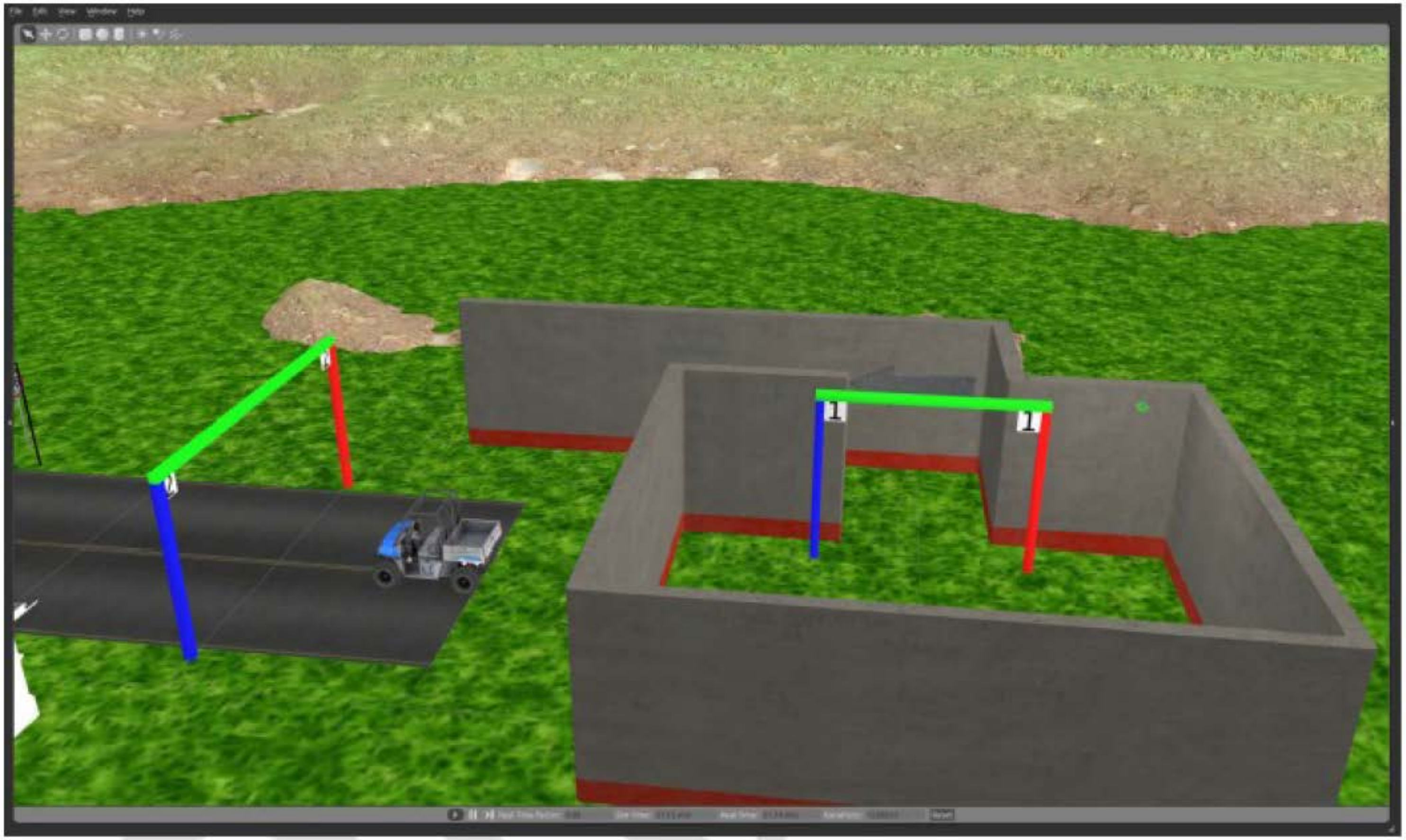
- Hubo Vision on competition

DARPA Robotics Challenge
Task 1

http://www.theroboticschallenge.org/

Challenge Overview

DARPA Robotics Challenge Task 1

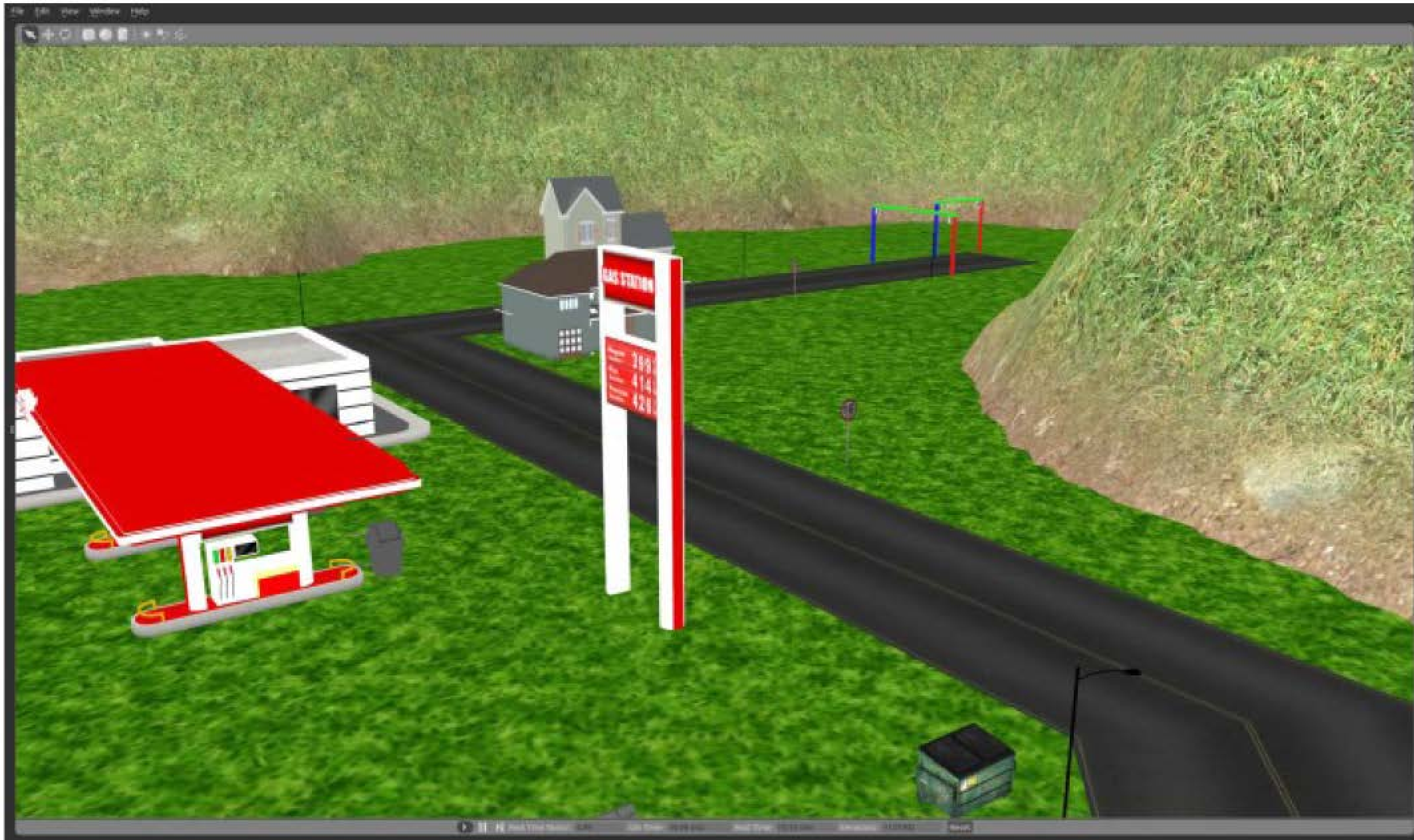


http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 1

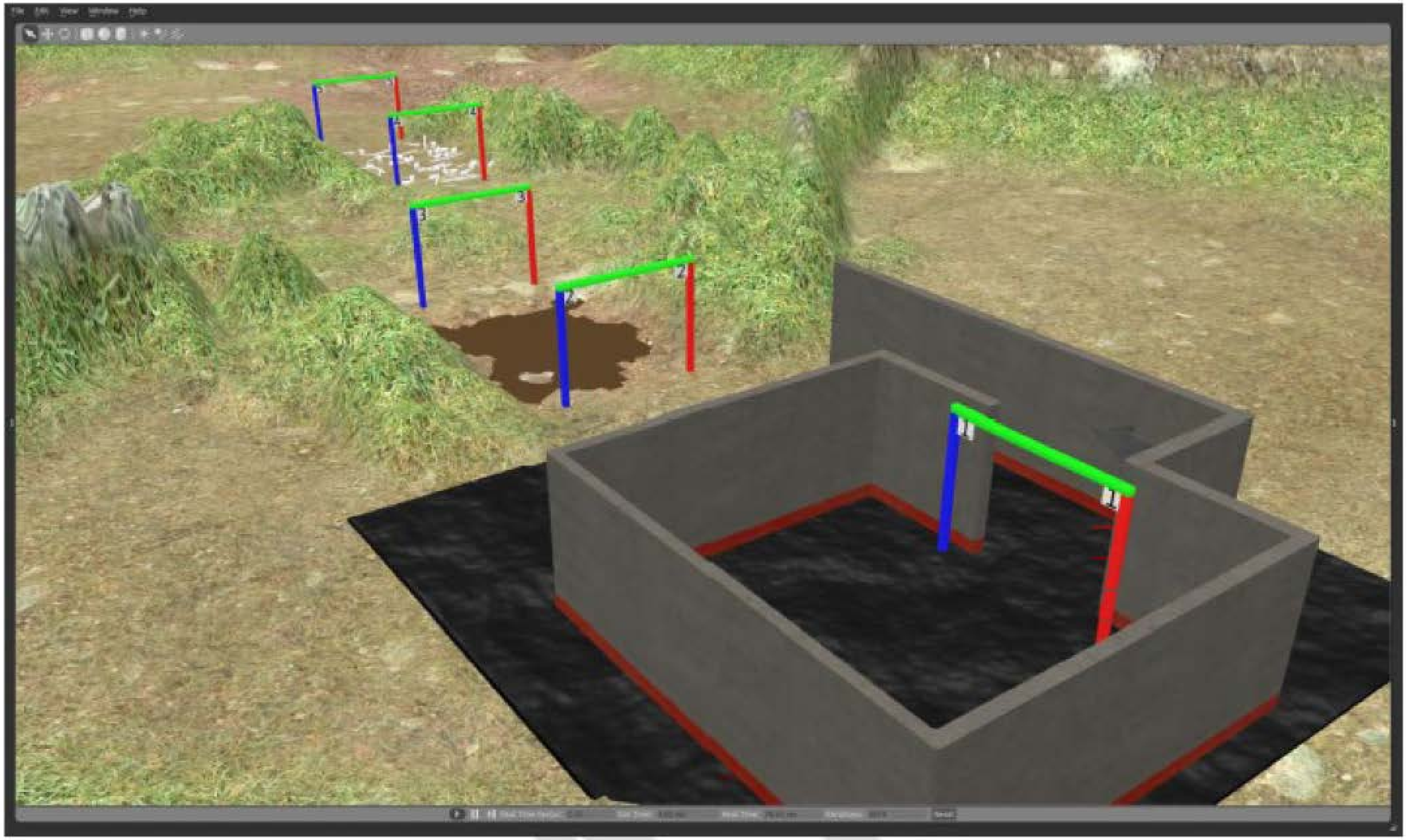
http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 1



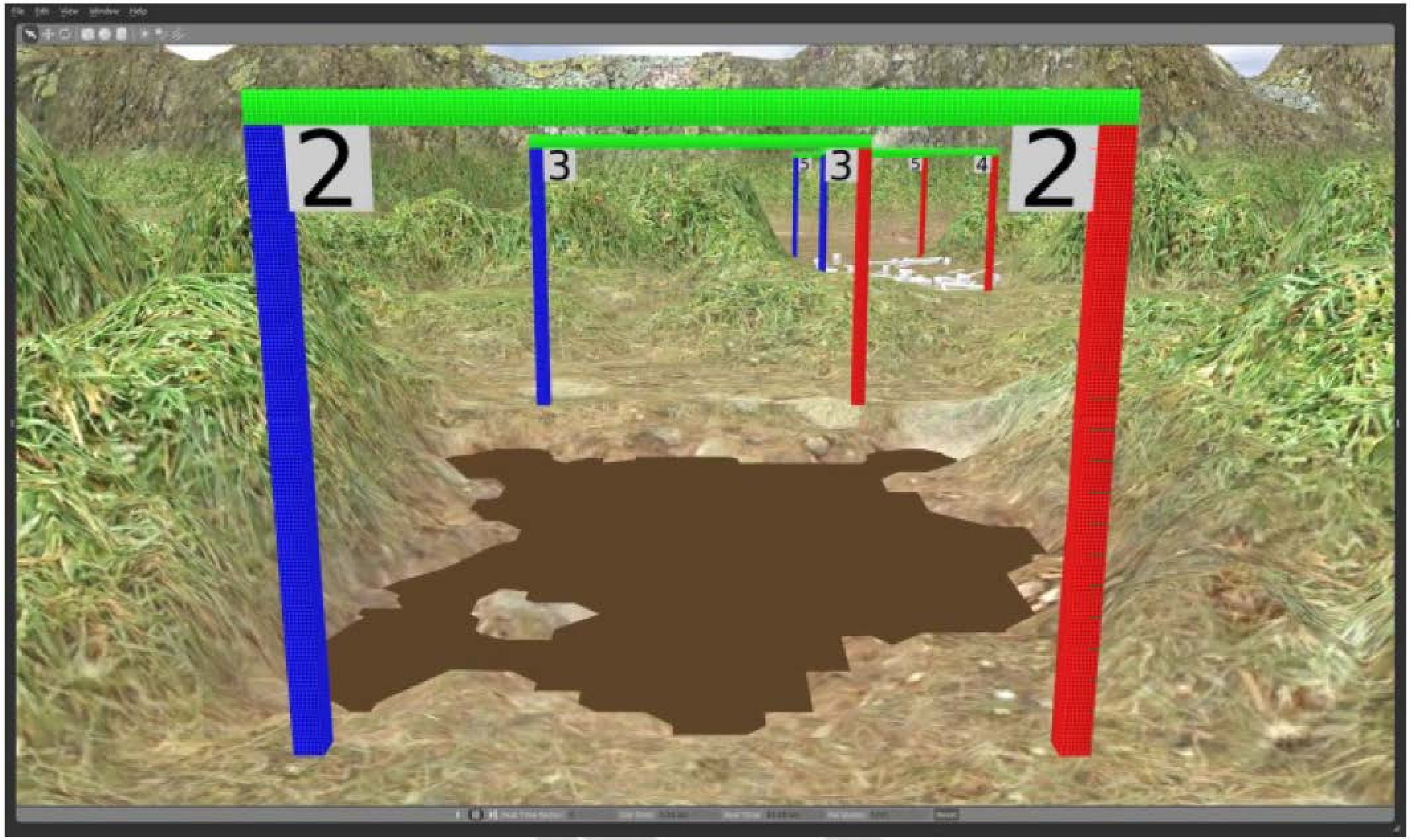
http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 2



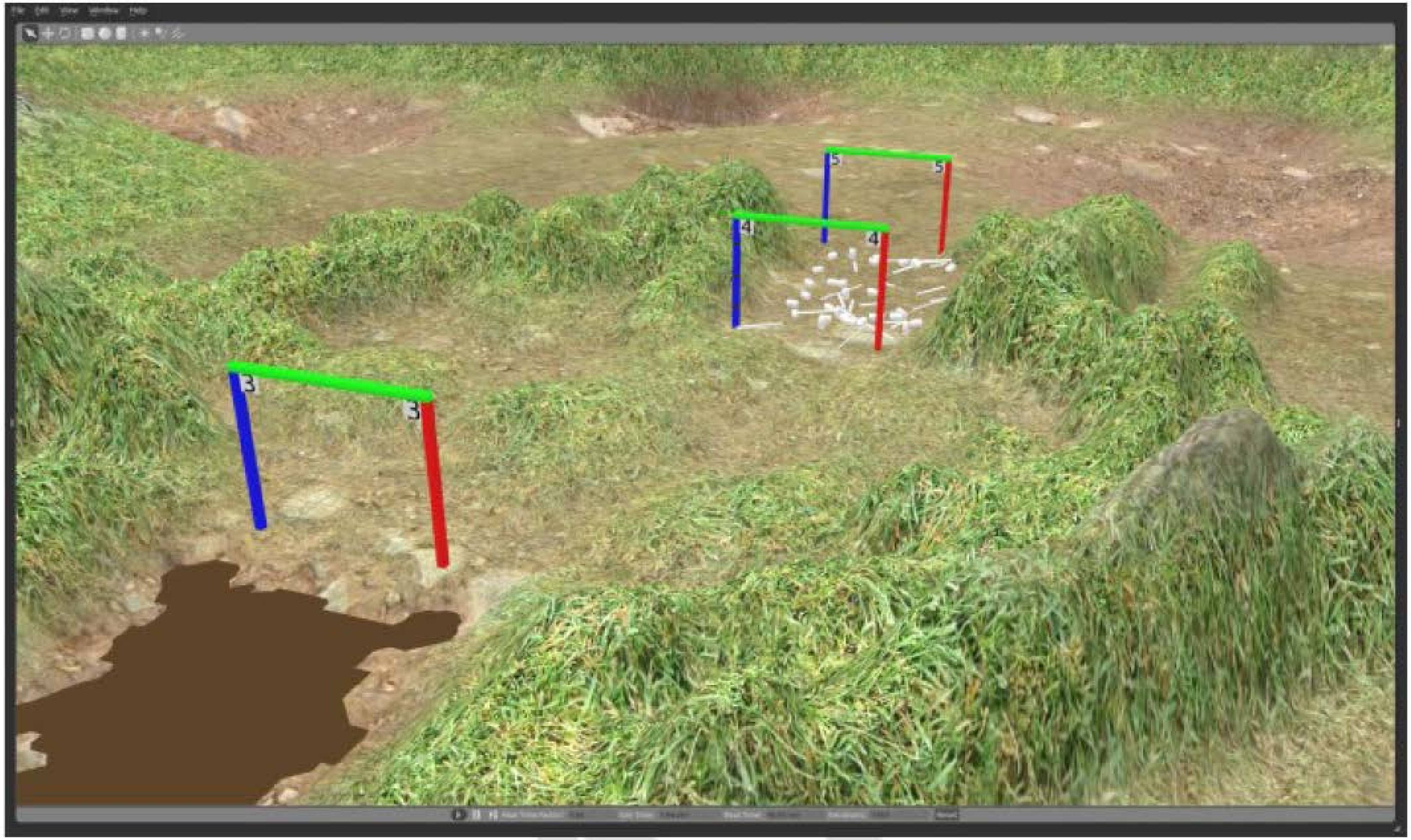
http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 2



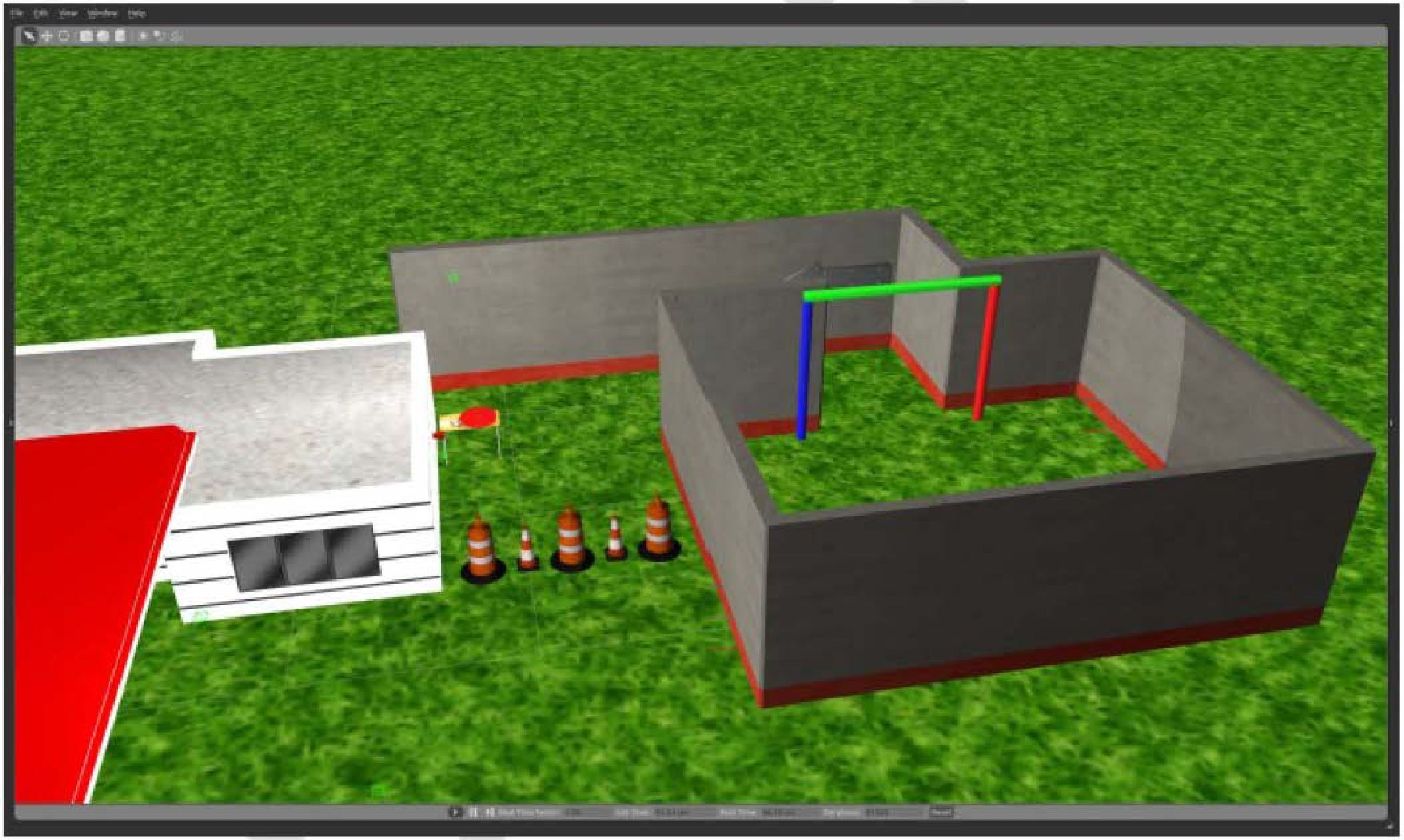
http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 2



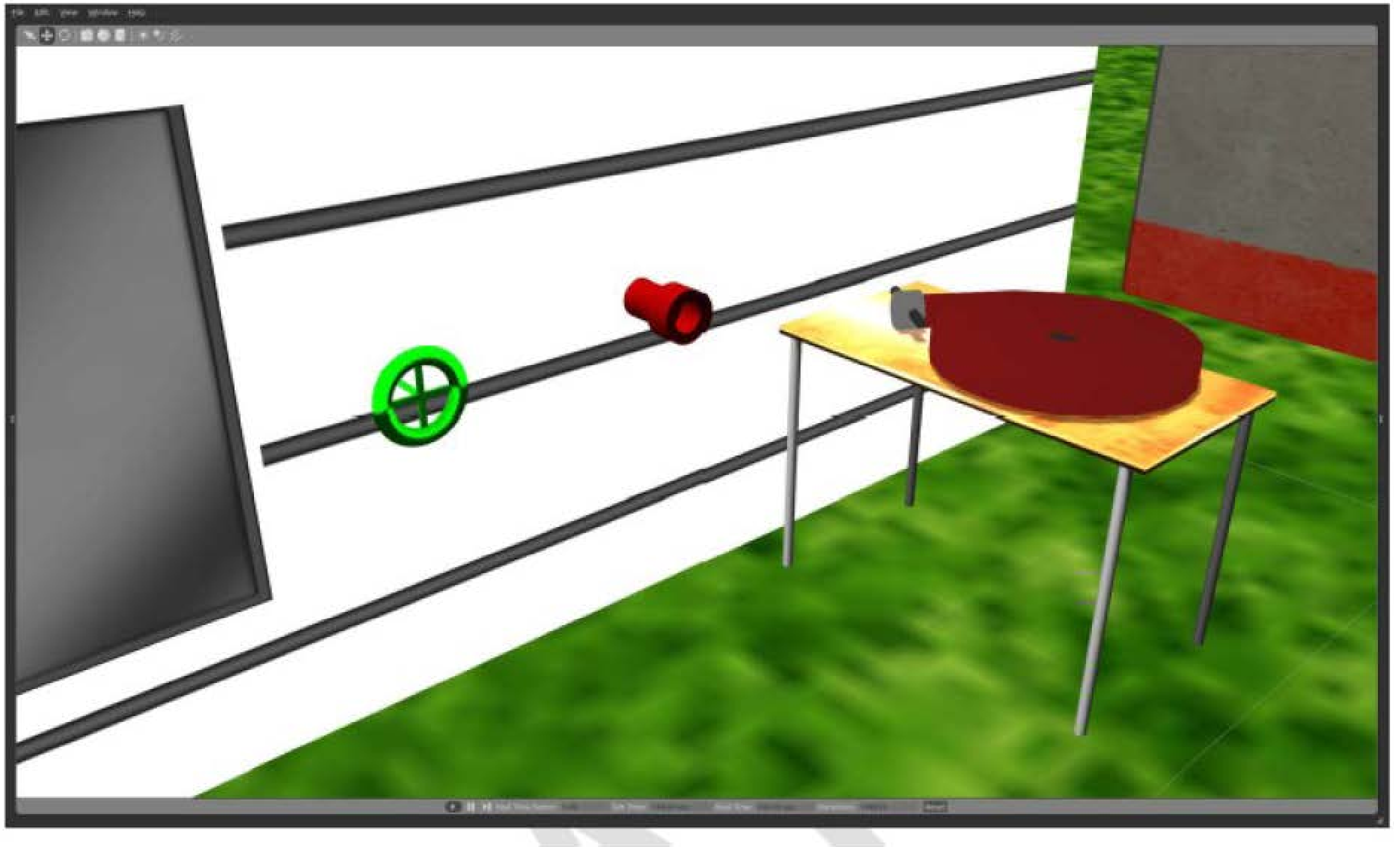
http://www.theroboticschallenge.org/

DARPA Robotics Challenge
Task 3



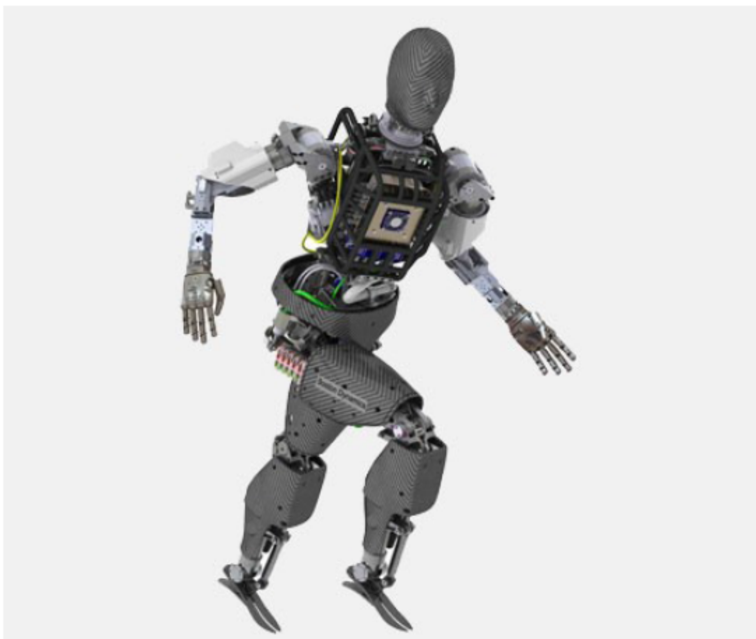
http://www.theroboticschallenge.org/

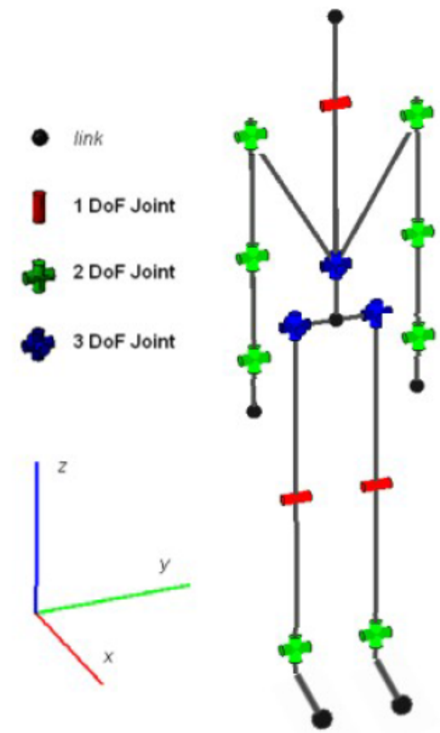
DARPA Robotics Challenge
Task 3



http://www.theroboticschallenge.org/ Challenge Overview

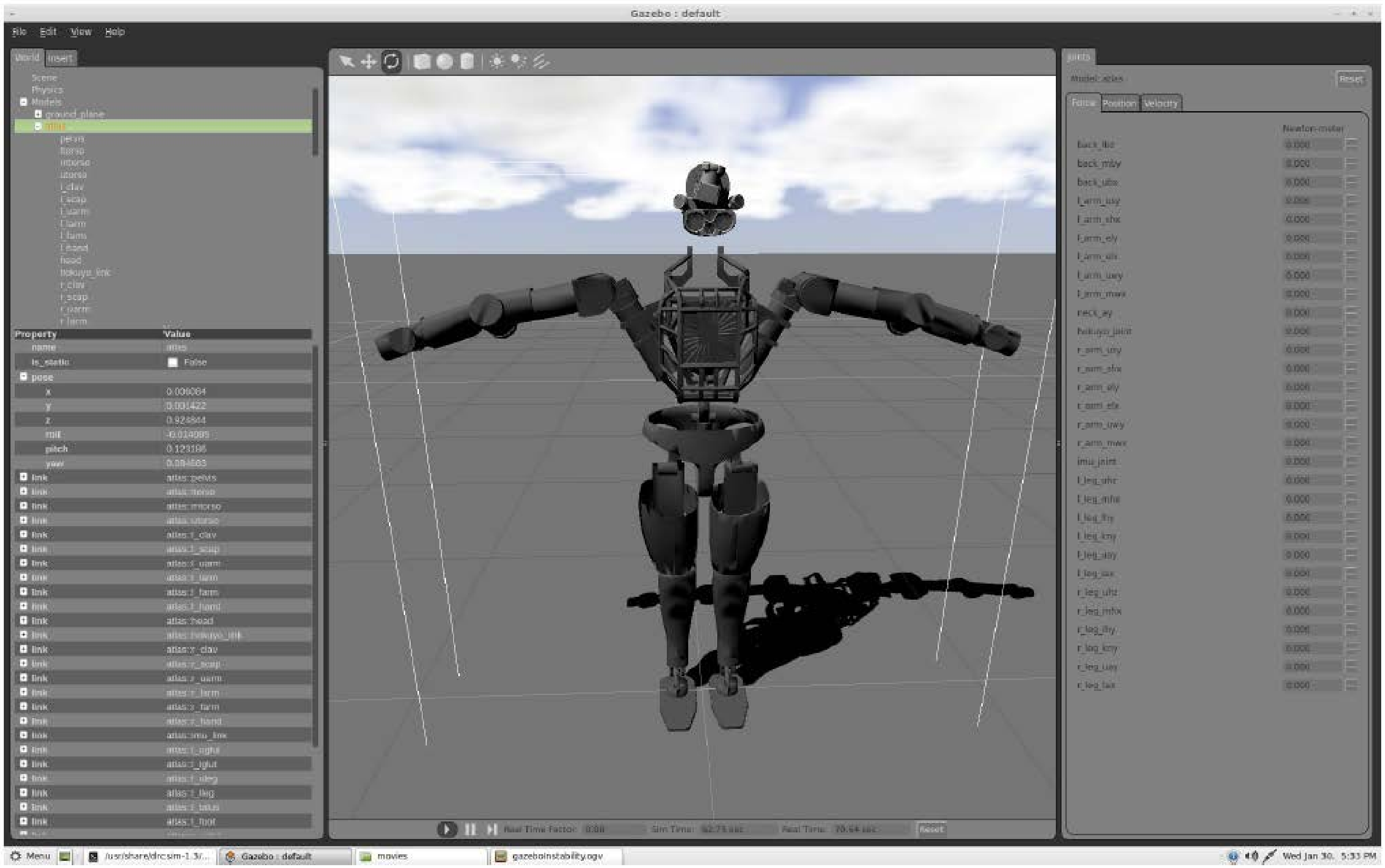
Walking over uneven terrain





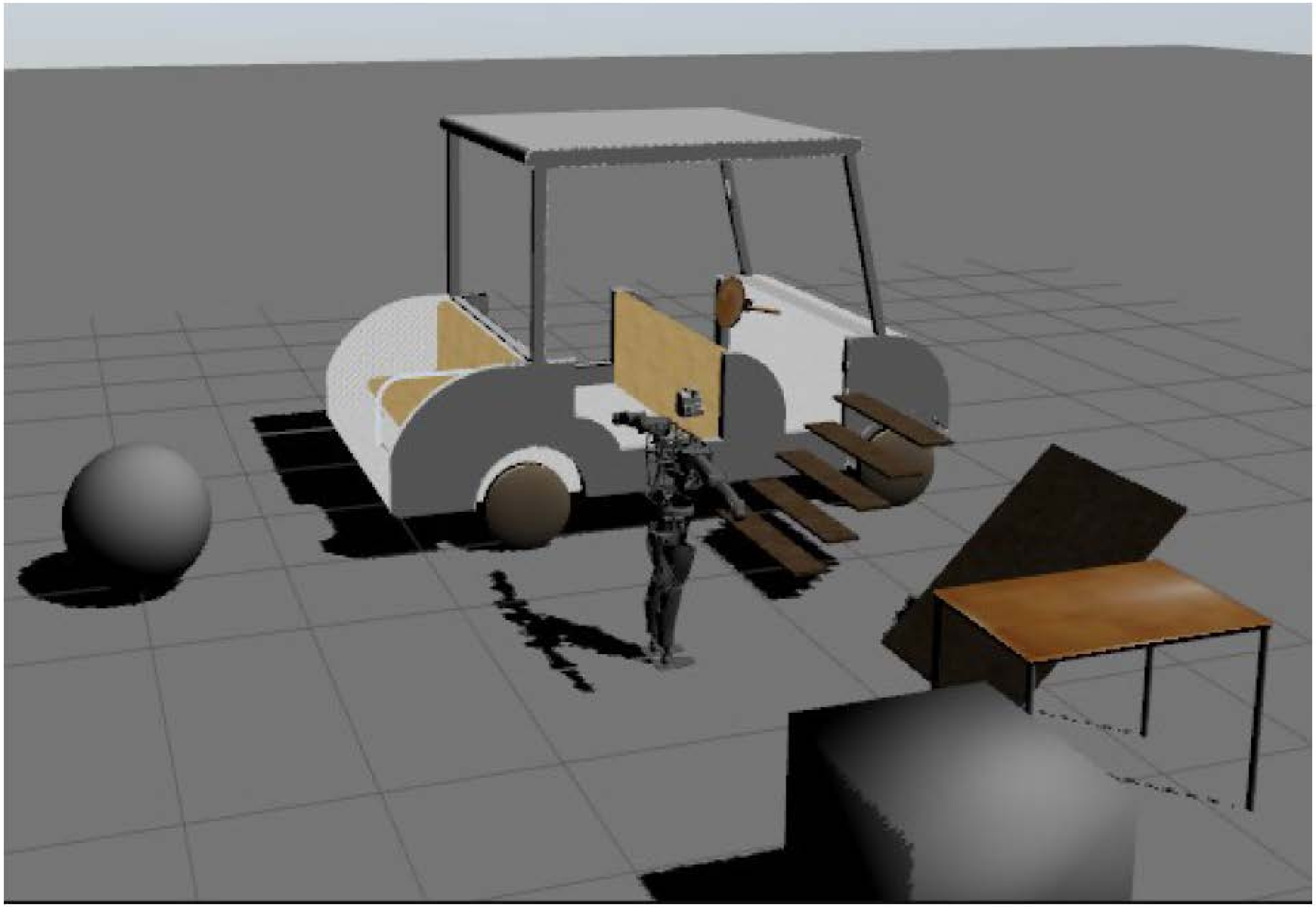
Atlas robot from Boston Dynamics: 28 Degrees of Freedom

DrcSim in Gazebo



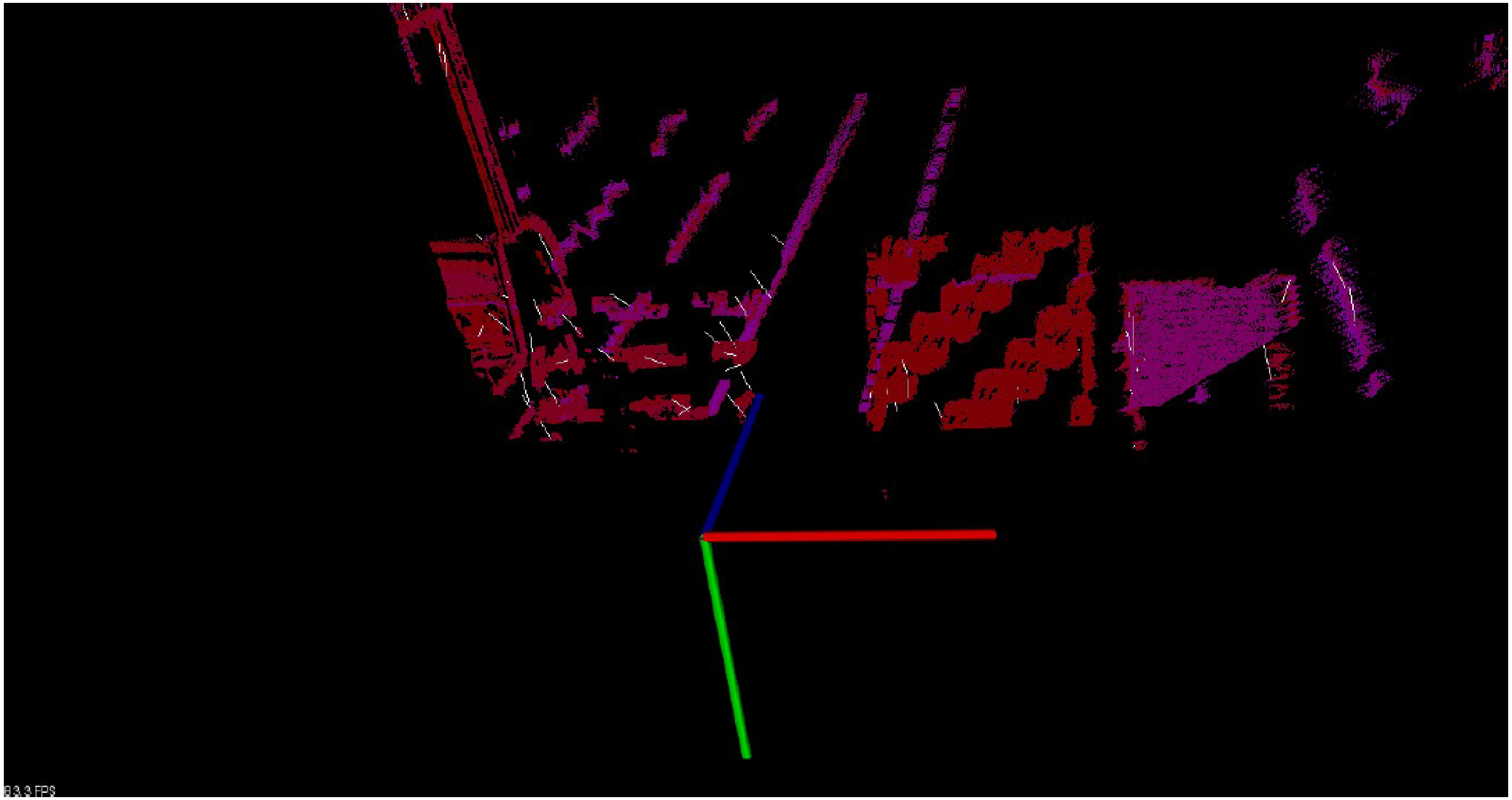
Build by Open Software Foundation

Test world used in this study



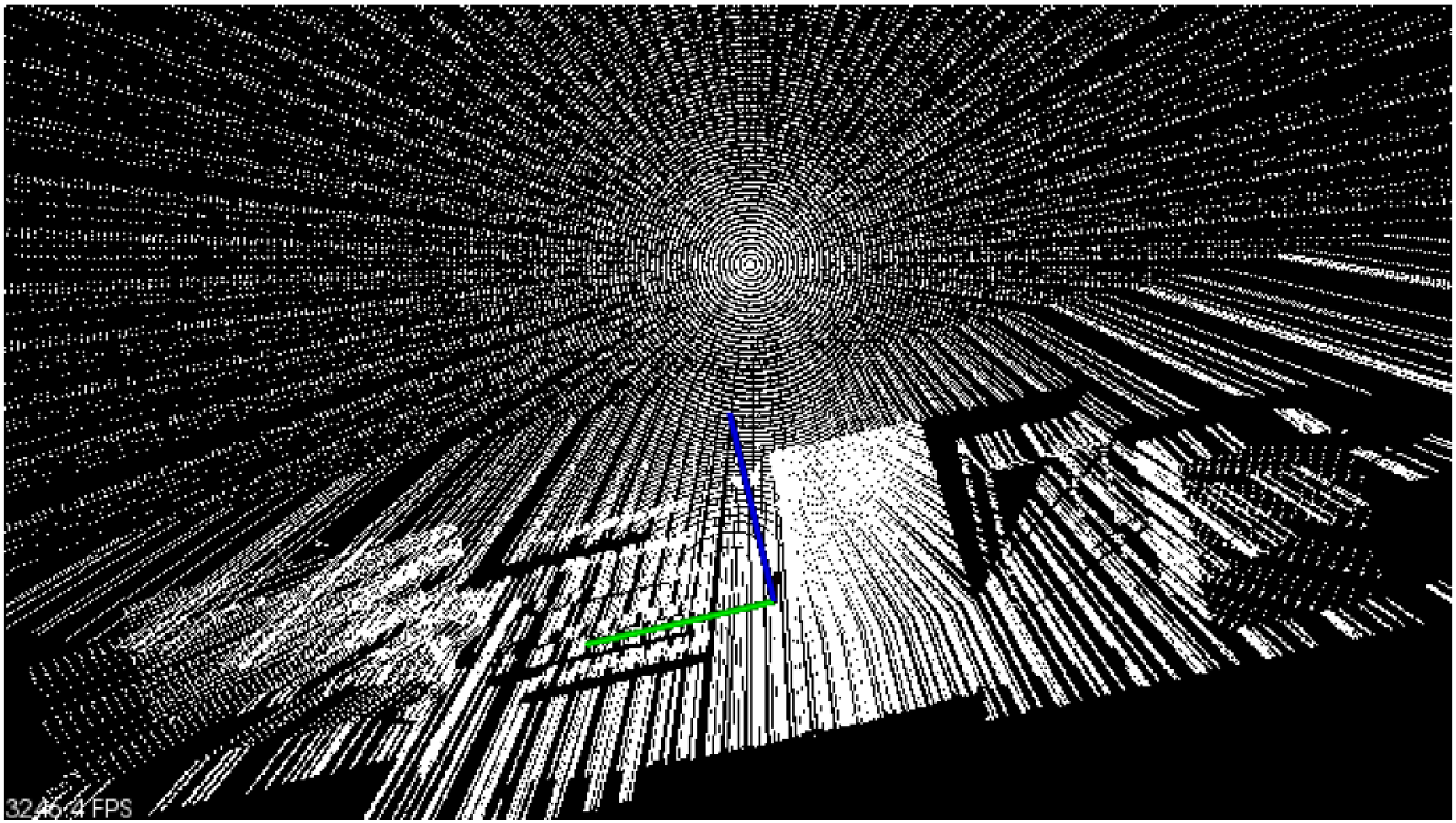
Number of flat surfaces, sloped, curved, stairs and a car.

Point Cloud sensor



The result: a sparse grid of points

Rotary laser sensor



The result: dense lines of points

The algorithm

Find Planes:

1) Plane Segmentation:

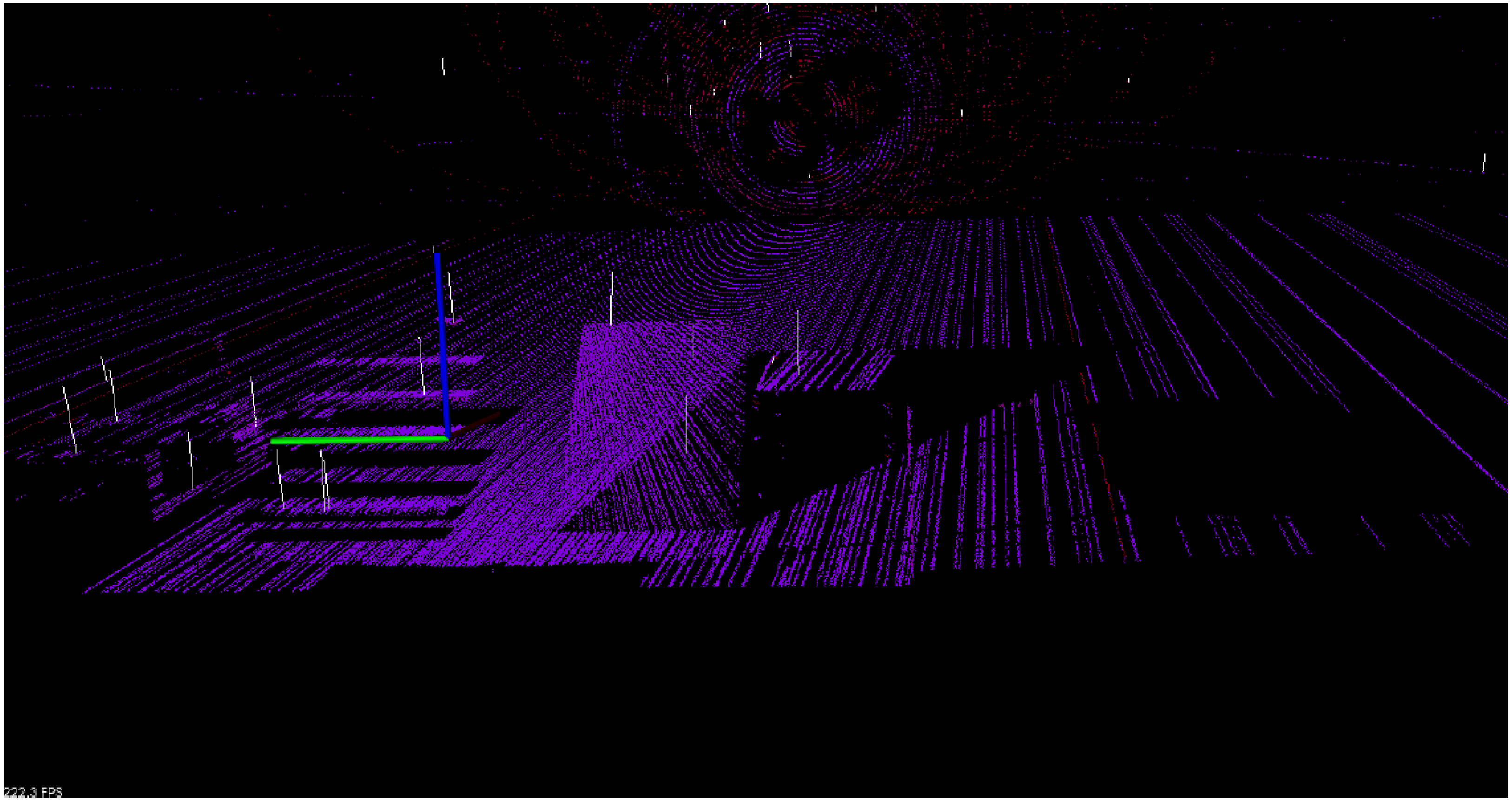
   The RANSAC algorithm informally goes as followed:

   - Randomly select a subset of the point cloud and estimate the free model parameters
   - Other data is considered, if a point fits the model a point is added (considered an inlier)
   - The model is re-estimated considering all the inliers
   - The model is evaluated by estimating the error relative to the model

2) Calculate normal for plane

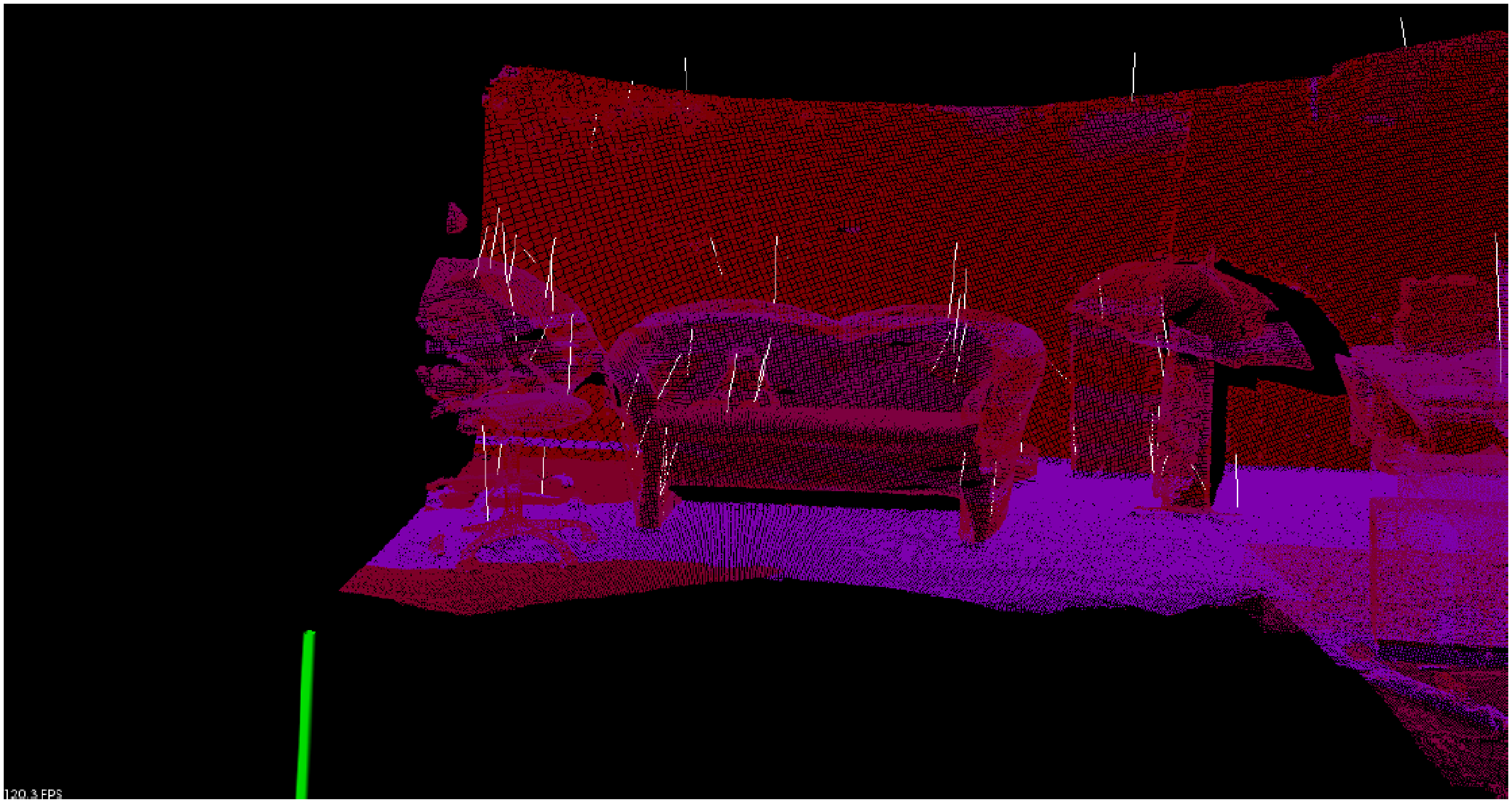
3) Plane Evaluation: distance to ideal normal $[0, 0, 1]$

Results



Artificial color: indication for equality: *equality^5 * 255 = color*

Results



Artificial color: indication for equality: *equality^5 * 255 = color*

http://www.pointclouds.org/documentation/tutorials/using kinfu largescale.php

Discussion

- One would want to find planes that are uninterrupted.
- Also planes with a small curve in them should be allowed.
- Initially this could be solved by (re)implementing region growing
- Ideally other factors for stability of foot placements could be learned by machine learning techniques

Discussion

- Retrieve the size of the plane and comparing that with the size of the robot's foot.
- Footstep locations should be found at locations that are reachable by a single step.
- Footstep locations should be found in the direction that one likes to go.

Conclusion

It is possible to find planes in the environment that can be stepped on by the Atlas robot.

RoboCup 2006 Bremen Germany

3rd place

RoboCup Atlanta 2007

4th place

RoboCup German Open

2nd place

Brazil Open

1st place

RoboCup 2009 Graz • Austria

3rd place

Iran Open 2010

Development price

Iran Open 2011

3rd place

RoboCup 2012 18 - 24 June Mexico City

Infrastructure price

Quite some robots were hurt by this research

DARPA Robotics Challenge
Task 1

DARPA Robotics Challenge
Task 2

DARPA Robotics Challenge
Task 3

www.jointrescueforces.eu

Amsterdam Oxford Joint Rescue Forces
RoboCup Rescue Simulation - Virtual Robots Competition

Publications

Publications listed below are relevant to research conducted by UvARescue and Amsterdam Oxford Joint Rescue Forces in the USARSim simulator. For a more extensive list of publications related to this competition see the RoboCup Rescue wiki and the Success Stories on Sourgeforge.

2013

- Francesco Amigoni, Arnoud Visser and Masotoshi Tsushima, "RoboCup 2012 Rescue Simulation Winners", To be published in the Springer Lecture Notes on Artificial Intelligence series, volume 7500, pp. 20-35, 2013 (PDF).
- Sander van Noort and Arnoud Visser, "Extending Virtual Robots towards RoboCup Soccer Simulation and @Home", To be published in the Springer Lecture Notes on Artificial Intelligence series, volume 7500, pp. 332-343. (PDF).
- Maarten de Waard, Maarten Inja and Arnoud Visser, "Analysis of flat terrain for the Atlas robot", Proceedings of the RoboCup IranOpen 2013 Symposium (RIOS13), April 2013. (PDF).
- H.L. Akin, N. Ito, A. Kleiner, J. Pellenz and A. Visser, "RoboCup Rescue Robot and Simulation Leagues", AI Magazine, Vol 34, 2013.
- Maarten Inja, Norbert Heijne, Sander Nugteren and Maarten de Waard, "Project AI - The Darpa Robotics Challange - F.O.O.T.L.O.O.S.E.", Project Report, Universiteit van Amsterdam (February 2013) (PDF).

2012

- Arnoud Visser, "UvA Rescue Technical Report: A description of the methods and algorithms implemented in the UvA Rescue code release", Technical Report IAS-UVA-12-02, Universiteit van Amsterdam (December 2012) (PDF).
- Sander van Noort and Arnoud Visser, "Validation of the dynamics of an humanoid robot in USARSim", in Proceedings of Performance Metrics for Intelligent Systems Workshop (PerMIS'12), (Edited by Rajmohan Madhavan, Elena R. Messina and Brian A. Weiss), NIST Special Publication 1136, pp. 190-197, National Institute of Standards and Technology, (November 2012) (PDF).

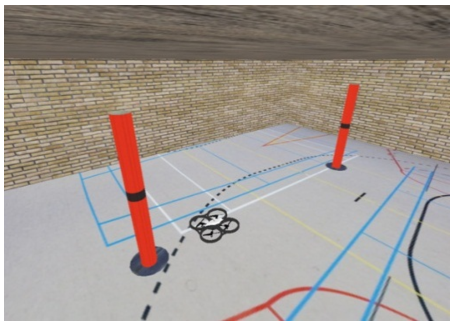
Universiteit van Amsterdam

UvA Rescue

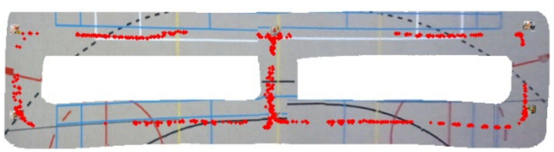
Intelligent Systems Laboratory

Innovations for Iran Open 2012 (i.e.)

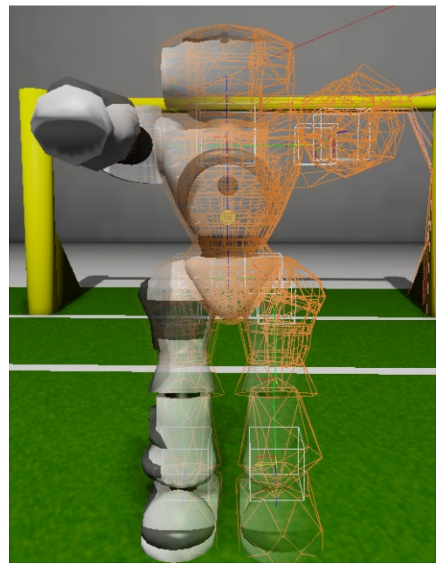
- Visual Localization And Mapping



AR.Drone localizing on visual map



- Nao humanoid robot



collision frame Nao

- Automatic map generator

map generated with high difficulty

Other assets:

- Can read many logfile formats (Radish, Carmen, etc.)
- Graph based map, which can be easily shared and corrected
- Smooth transition from teleoperated to fully autonomous behavior

www.jointrescueforces.eu

In close cooperation with

University of Oxford
Computing Laboratory

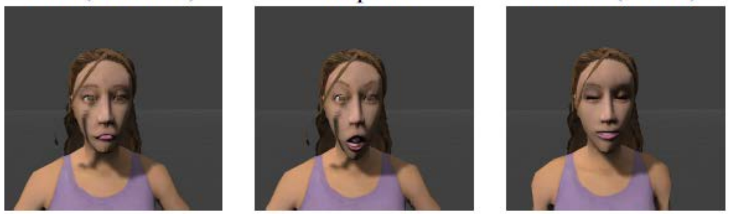
Amsterdam Oxford Joint Rescue Forces

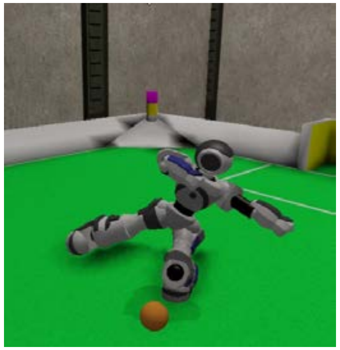
AR.Drone (including camera and sonar)

2011 INFRASTRUCTURE INNOVATIONS:

- **Realistic Victim behaviors**
- **Nao kinemetics model**
- **AR Drone model**
- **Kenaf model**

Sad (Sadness) Surprise bLink (Blink)



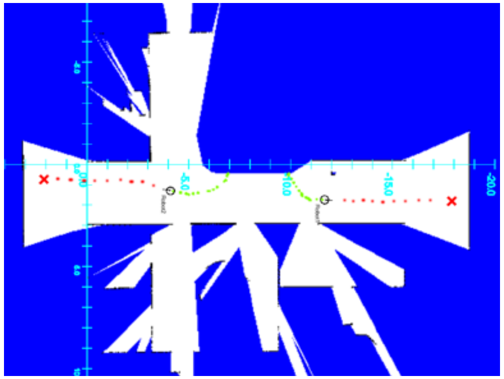


Nao (balancing on one foot)

Kenaf robot with flippers

2011 TEAM BEHAVIOUR INNOVATIONS:

- **Graph based map, which can be easily shared and corrected**
- **Smooth transition from teleoperation to full autonomy**
- **Waypoint following behaviour**



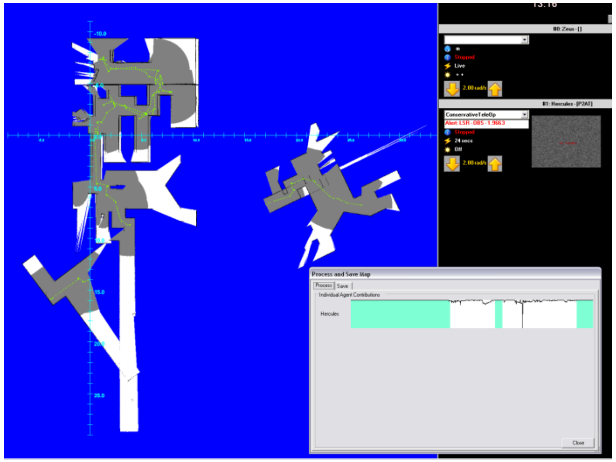
Using waypoints for improved exploration

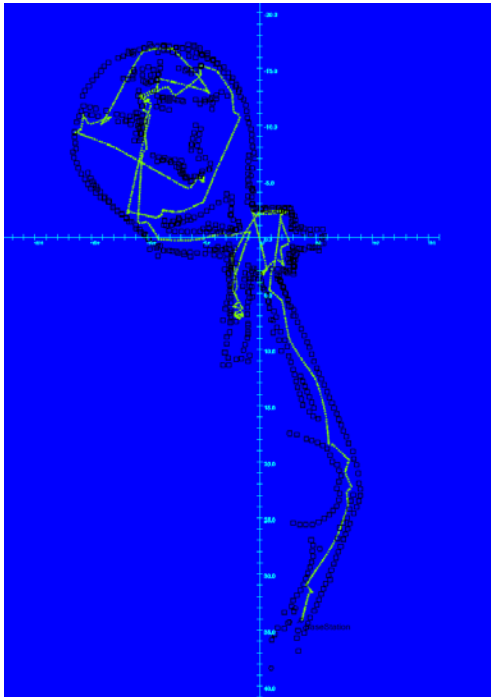
www.jointrescueforces.eu

Amsterdam Oxford Joint Rescue Forces

Innovations for Iran Open 2010 (i.e.)

- Realistic Smoke
- Confidence selection in maps
- Autonomous AirRobots
- Local sonar maps





Other assets:

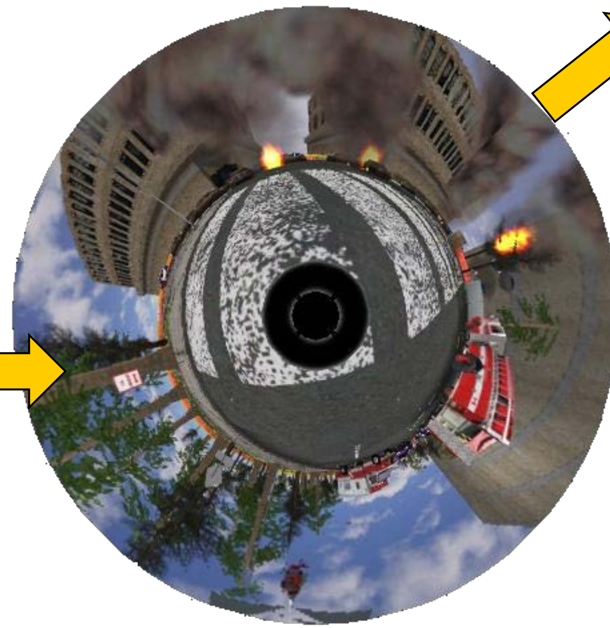
- Can control many robots (Matilda, Element, Talon, AirRobot, ATRVJr, Zerg. etc.)
- Graph based map, which can be easily shared and corrected
- Smooth transition from teleoperated to fully autonomous behavior

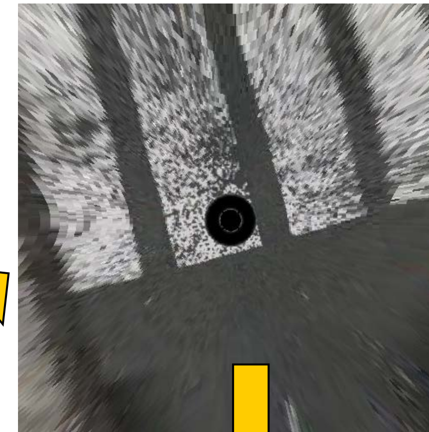
www.jointrescueforces.eu

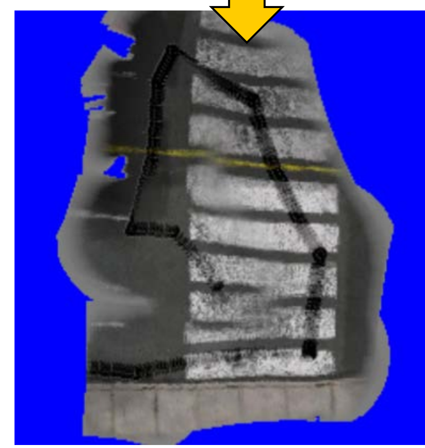
Amsterdam Oxford Joint Rescue Forces

2008 innovations (i.e.)

- Created omnicamera
- Omniview can be easily transformed in other views
- Landmarks stay in view
- Bird-eye views can be combined into visual maps







Other assets:

- Graph based map, which can be easily shared and corrected
- Smooth transition from teleoperated to fully autonomous behavior

www.jointrescueforces.eu