Mobile Robotics

Prof. Dr. Ir. Moh. Khairudin, IPU.

Email: moh_khairudin@uny.ac.id

Expectation

Before

After

SESUDAH

Our discussion

01 **The Indonesian Contest Robot (KRI)**

02 **Mobile robot.**

03 **Path Planning**

04 **Simultaneous Localization & Mapping - SLAM**

05 **Case Study.**

The Indonesian Contest Robot (KRI)

- ✓ The Indonesian Robot Contest (KRI) is an annual student competition in the field of robotics design and engineering
- ✓ Organized by the National Achievement Center, Ministry of Education, Culture, Research and Technology of the Republic of Indonesia.

The Indonesian Contest Robot (KRI)

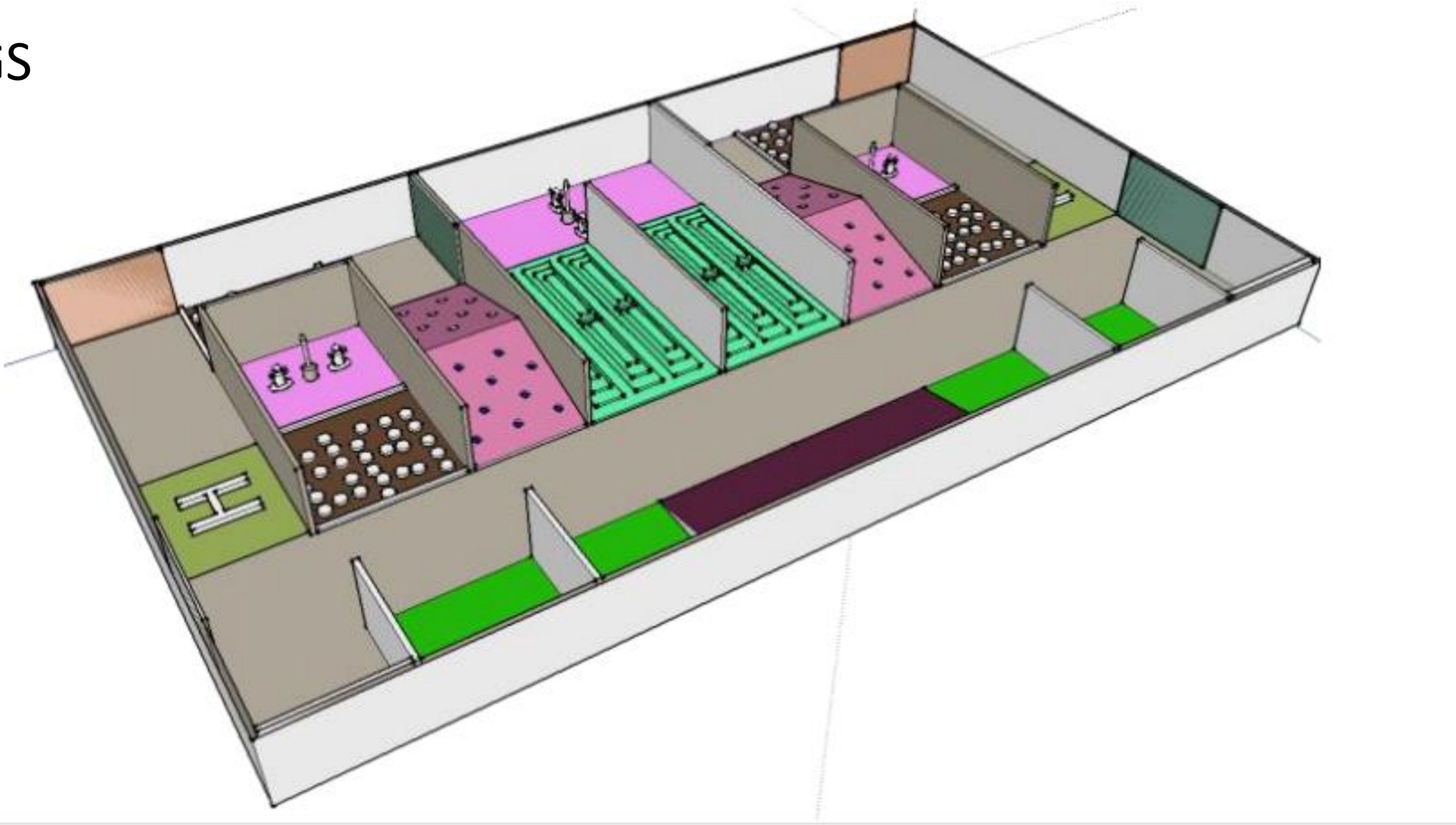
- KRI was first organized in 2003 under the Directorate General of Higher Education, Ministry of Education and Culture at that time.
- KRI can be attended by all students at tertiary institutions in the territory of the Republic of Indonesia, from various Ministries/Institutions or official offices, which are recorded in the Higher Education Database.

The Indonesian Robot Contest competes in 6 (six) divisions, namely:

1. the Indonesian ABU Robot Contest (KRAI);
2. Indonesian SAR Robot Contest (KRSRI);
3. Wheeled Indonesian Football Robot Contest (KRSBI);
4. Humanoid Indonesian Football Robot Contest (KRSBI);
5. The Indonesian Dance Robot Contest (KRSTI); and
6. The Indonesian Thematic Robot Contest (KRTMI).

The Indonesian SAR Robot Contest (KRSRI);

AUTONOMOUS ROBOT ON LEGS
FIRE VICTIMS SAVIOR



The Indonesian SAR Robot Contest (KRSRI);

Mission on this contest

- ✓ The robot tries to put out the fire and save potential victims in two indoor rooms a building represented by an arena.
- ✓ On the way to the second room these, the robot must be able to overcome obstacles on the wall and on the floor.
- ✓ Where the robot should becan move independently and make their own decisions without being controlled by the operator man.
- ✓ The rescue of potential victims is carried out by elevating them to a safety zone without sliding it on the floor. Robot may return to Home after success perform fire-and-rescue in both rooms.

The Indonesian SAR Robot Contest (KRSRI);

- ✓ The robot at the 2022 KRSRI is assigned not only to blow out candles,
- ✓ but also to save potential victims from potential fires to a safe place.

The Indonesian SAR Robot Contest (KRSRI);

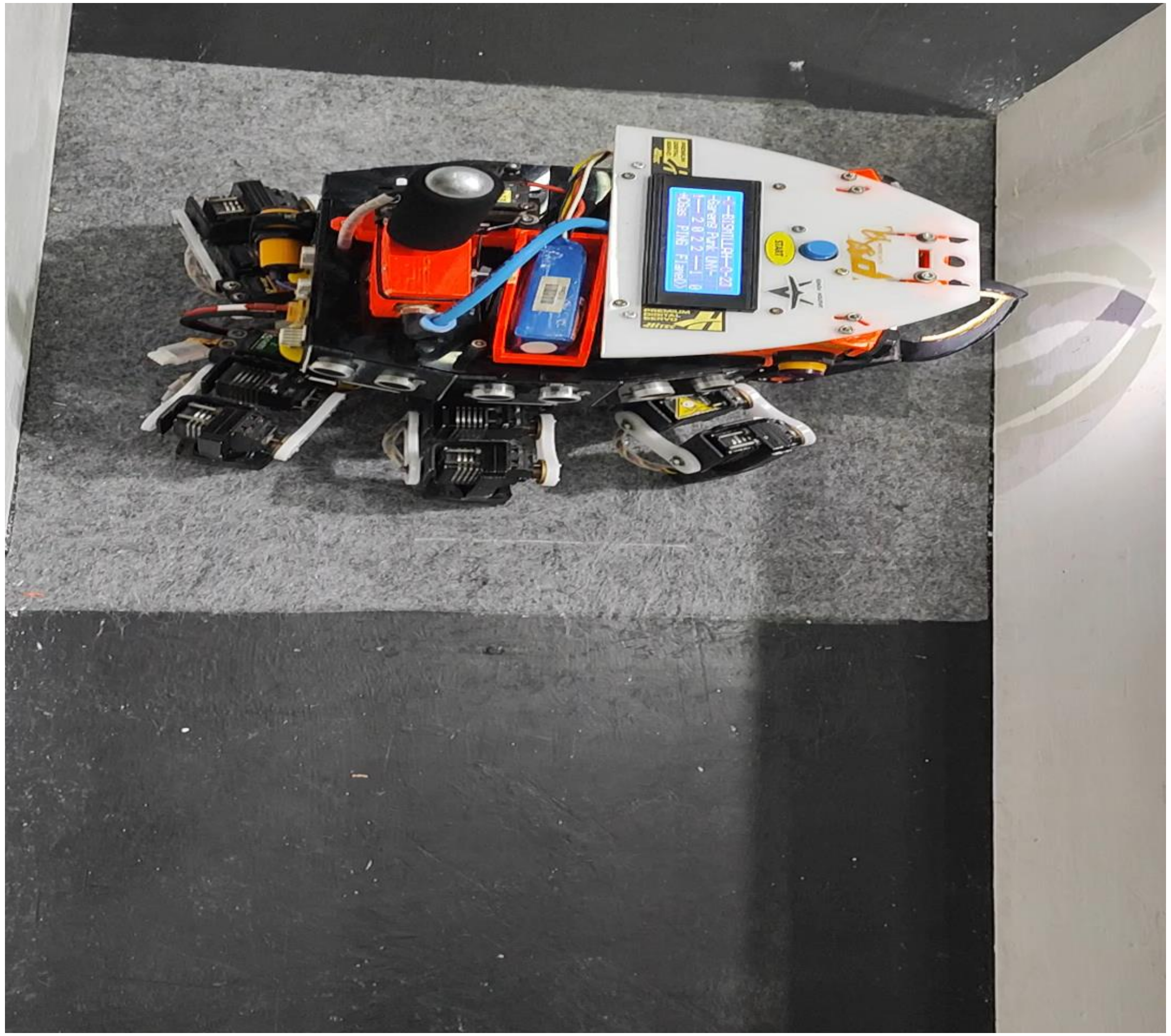
- Along with the name of the robot contest to prioritize the SAR function with legged robots,
- each trajectory is givenmore challenging obstacles as the irregular environment forms as challenges commonly faced by SAR robots in disaster areas.

The Indonesian SAR Robot Contest (KRSRI);

The robots that follow this KRSRI must be able to meet the KRSRI robot criteria and save potential victims as following;

1. Can move independently without operator and without guiding line
2. Can make their own decisions based on the variety of challenges given
3. Can pass through obstacles on the way
4. Not affected by room disturbance parameters such as color arenas, beams of light, uneven and sound-absorbing wall surfacesand walls that reflect light

The Indonesian SAR Robot Contest (KRSRI);



The Indonesian SAR Robot Contest (KRSRI);

The robots that follow this KRSRImust be able to meet the KRSRI robot criteria and save potential victims as following;

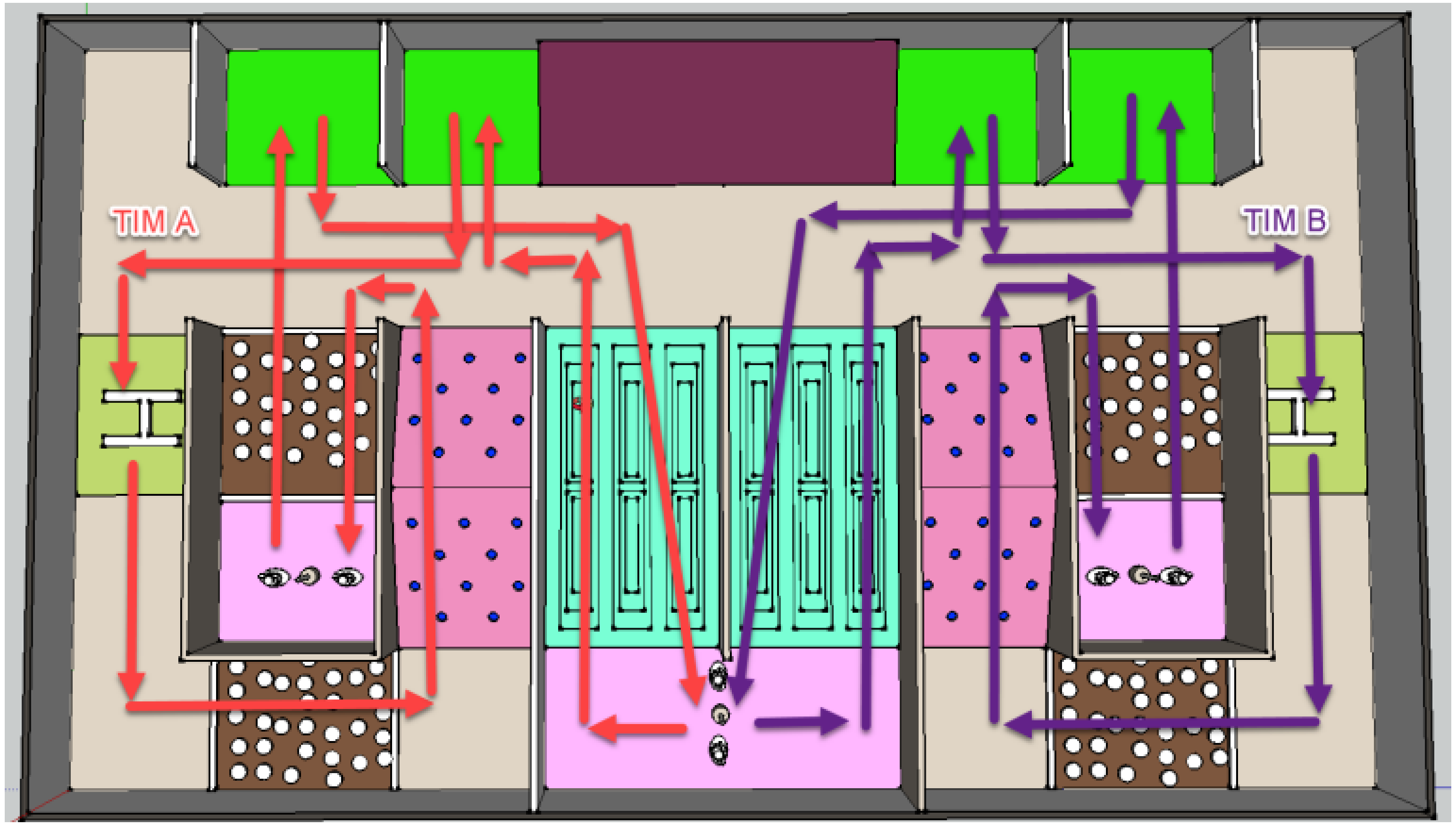
5. Can move on uneven floor surfaces in the form of debris, holes or steps
6. Can find the presence of heat represented by candles and put it out
7. It can prevent the candle or its support from falling towards the potential victim represented by dolls
8. Can detect and pick up potential victims without dropping them
9. Can save potential victims to a safe area / safety zone nearby though with an uneven floor surface without dropping it

The Indonesian SAR Robot Contest (KRSRI);

The dimensions of the arena for regional/national contests use the old / previous year's arena base measuring 2.4 x 2.4m.

- Aisle Width 45cm.
- The width of the fire / casualty room is 56cm
- Wall thickness 2cm
- The distance between the candles/dolls from the wall is 15cm
- 2cm thick debris barrier barrier
- The Mirror/Damper Hurdle is missing for the turf contest

The Indonesian SAR Robot Contest (KRSRI);



The Wheeled Indonesian Football Robot Contest (KRSBI);

- ✓ Basically at this stage it still applies the rules of the game in Robocup.
- ✓ Because That is, the basic rules of Robocup, such as robot size, ball size, method herding, making goals and so on as far as possible is still carried out accordingly Robocup rules.

The Wheeled Indonesian Football Robot Contest (KRSBI);

At this stage, the Wheeled KRSBI robot used is the same robot as the robot used for National matches, with differences, namely:

2.1 Number of robots: two and only two are allowed, namely Robot 1 (R1) and Robot 2 (R2)

2.2 Type of robot: attack robot

2.3 Size, weight and shape of the robot:

The projection of the robot onto the floor is a minimum of 30 cm x 30 cm, and a maximum of 52 cm x 52 cm.

The minimum robot height is 40 cm, and the maximum is 80 cm.

If the height of the robot is more than 60 cm, then the part of the robot above 60 cm from the ground must be
into a 25 cm diameter cylinder.

Robot Weight : maximum 40 kg.

Robot form: free.

Robots Color: black.

The Wheeled Indonesian Football Robot Contest (KRSBI);

The Wheeled Indonesian Football Robot Contest (KRSBI);

2.4 Robot identity

The number of robot 1 (robot R1) or robot 2 (robot R2) is attached to the robot's body. Color number 1's background is magenta, and number 2's background is cyan.

This background should be in the form of a shawl that covers the entire body of the robot so that it is easy to see from a far.

2.5 Robot control

Robot MUST be started remotely. The remote method used must be using wifi in a network. So, in one network, there are computers controller (base station), to control R1 and R2.

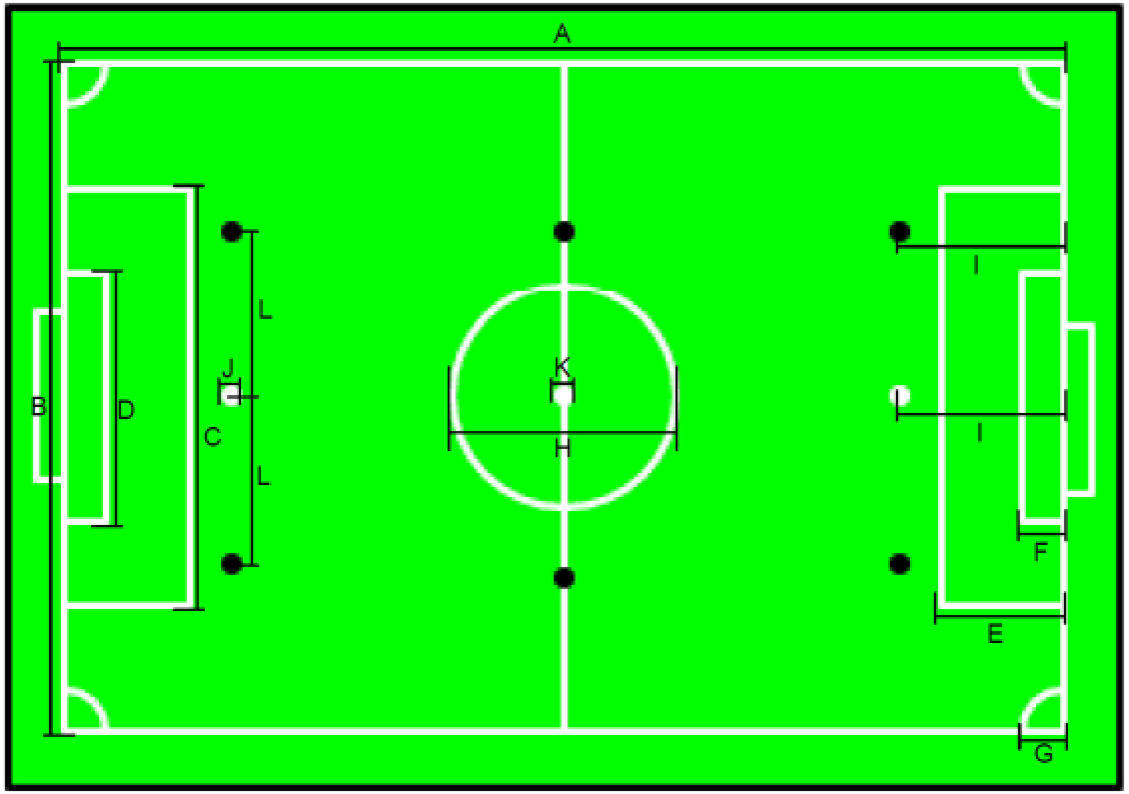
After starting, the robot cannot be controlled, but must move autonomously, either when finding the ball, dribbling or kicking the ball.

The Wheeled Indonesian Football Robot Contest (KRSBI);

- ✓ The robot must be equipped with tools to catch and dribble the ball so that when
- ✓ dribbling: the ball is not lifted, the ball must rotate naturally, and meet
- ✓ condition that only one-third of the ball enters the robot
- ✓ catching the ball from the pass: half of the ball can enter the robot.

The Wheeled Indonesian Football Robot Contest (KRSBI);

The Wheeled Indonesian Football Robot Contest (KRSBI);



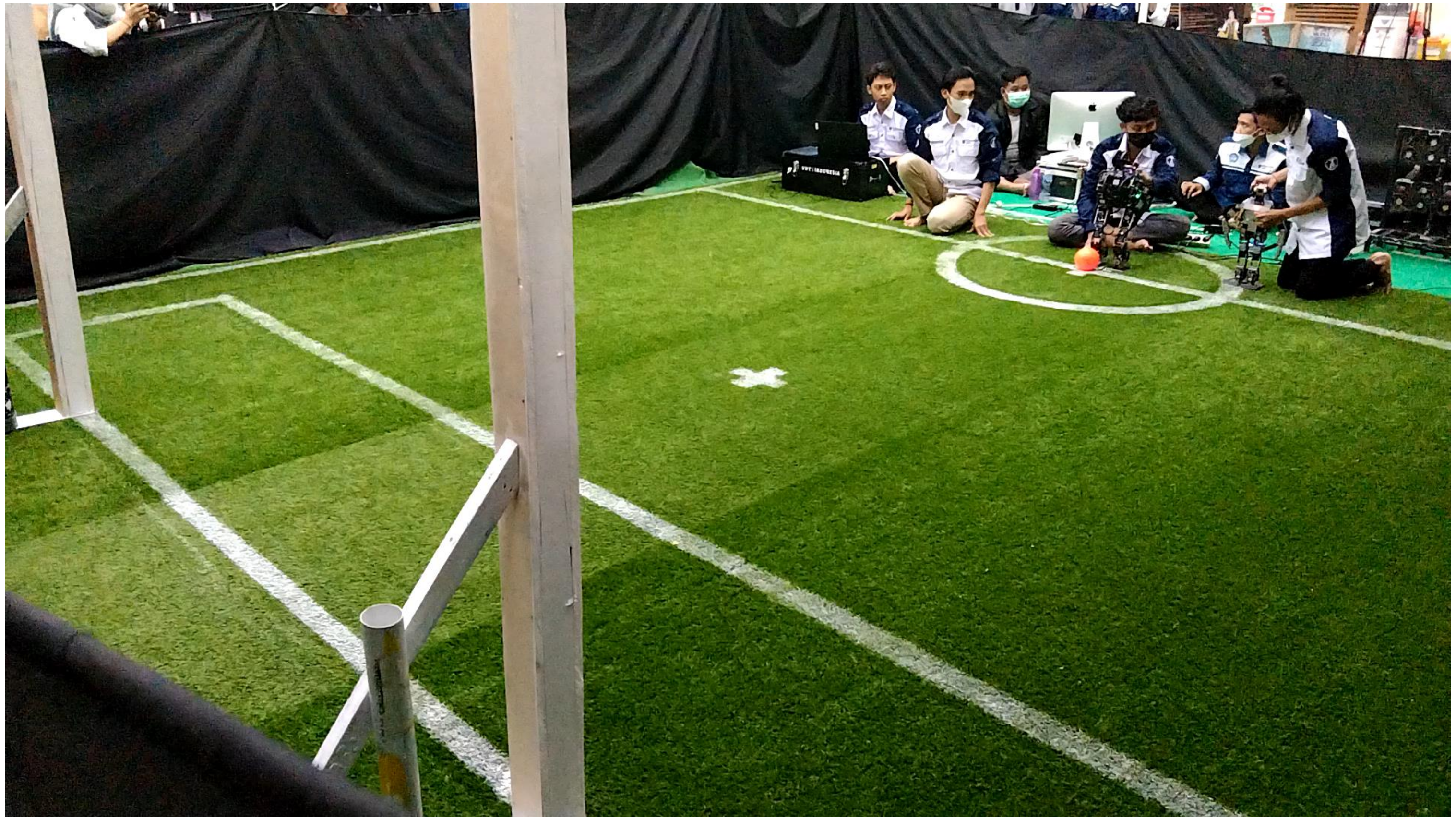
The size of a football field

A	12	G	0,5
B	8	H	2,6
C	Lebar gawang + 3	I	2
D	Lebar gawang + 1	J	0,06
E	1,8	K	0,1
F	0,5	L	2

The Wheeled Indonesian Football Robot Contest (KRSBI);

- ✓ As long as the robot carries the ball, the robot may move freely.
- ✓ But at the moment pass the ball and receive the ball, the robot must stay in place.
- ✓ Slight left and right turning motion is allowed.
- ✓ This condition applies to all modes, both at kick-off as well as during corner kicks.
- ✓ There are no special rules for corner kicks.

Humanoid Indonesian Football Robot Contest (KRSBI);

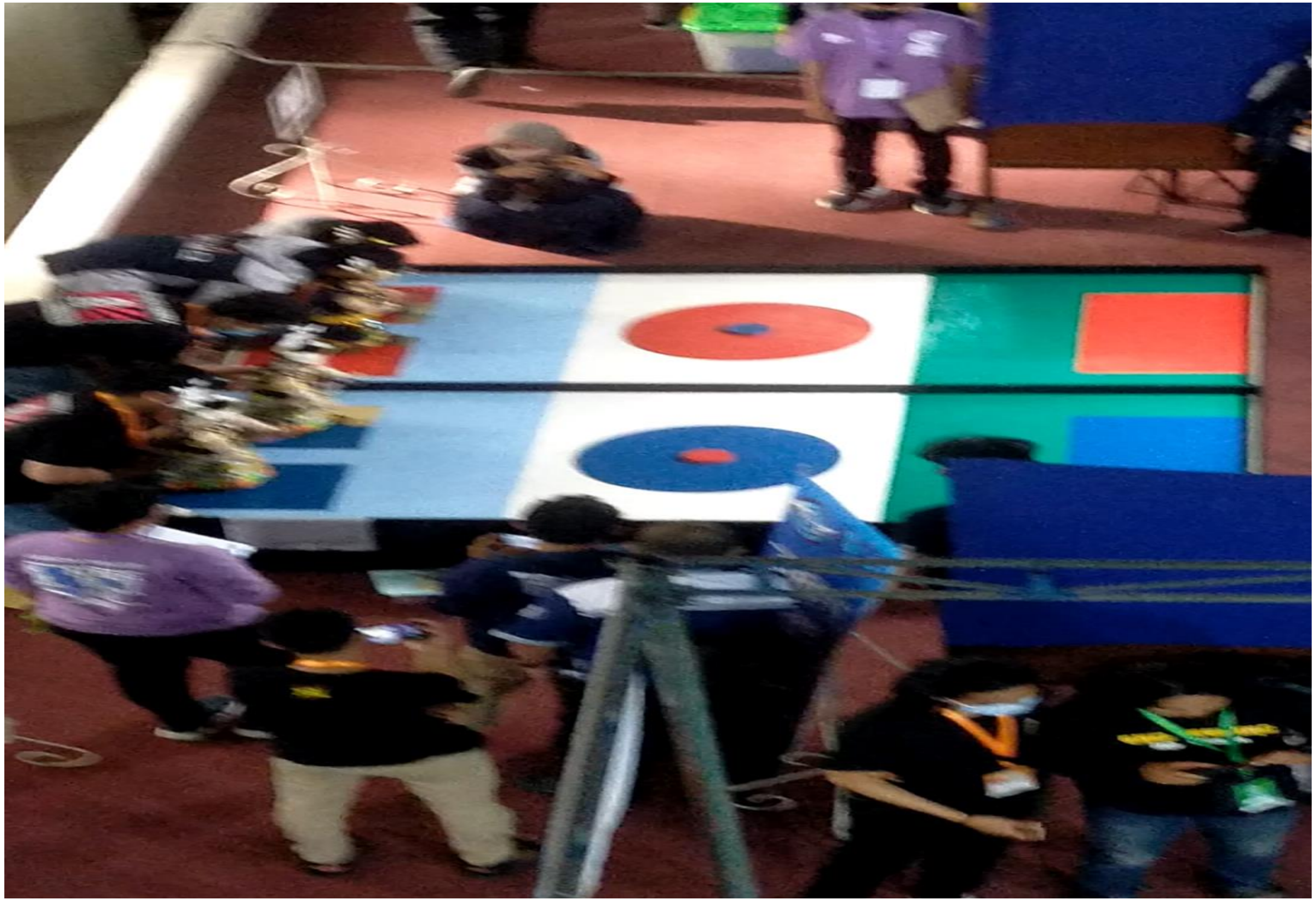


Indonesian Dance Robot Contest (KRSBI);

- ✓ The Indonesian Dance Robot Contest (KRSTI) is a competition for designing,
- ✓ making and programming robots accompanied by elements of art and culture the Indonesian nation,
- ✓ especially the art of dance which is well known in the motherland.

KRSI for the first time was held in 2010 with the theme "Jaipong Dancing Robot", in 2019 with the theme "Jaipong Dancer Robot", in 2020 with the theme "Enggang Dancer Robot" and in 2021 with the theme "Robot Dancer Gambyong Pareanom".
Each participating team consists of 3 (three) active student and an active supervising lecturer from the same institution.
Every team participants are required to create two coordinated robots to display the art dance that reflects Indonesian culture according to the theme of the contest.

Indonesian Dance Robot Contest (KRSTI);



The Indonesian Thematic Robot Contest (KRTMI)

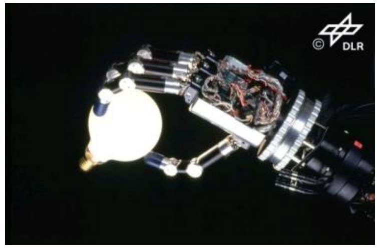
Robotics Yesterday

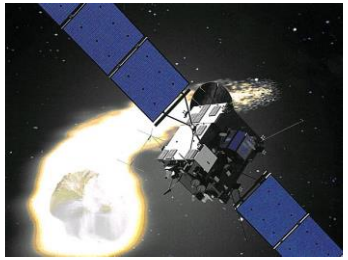
Current Trends in Robotics

Robots are moving away from factory floors to

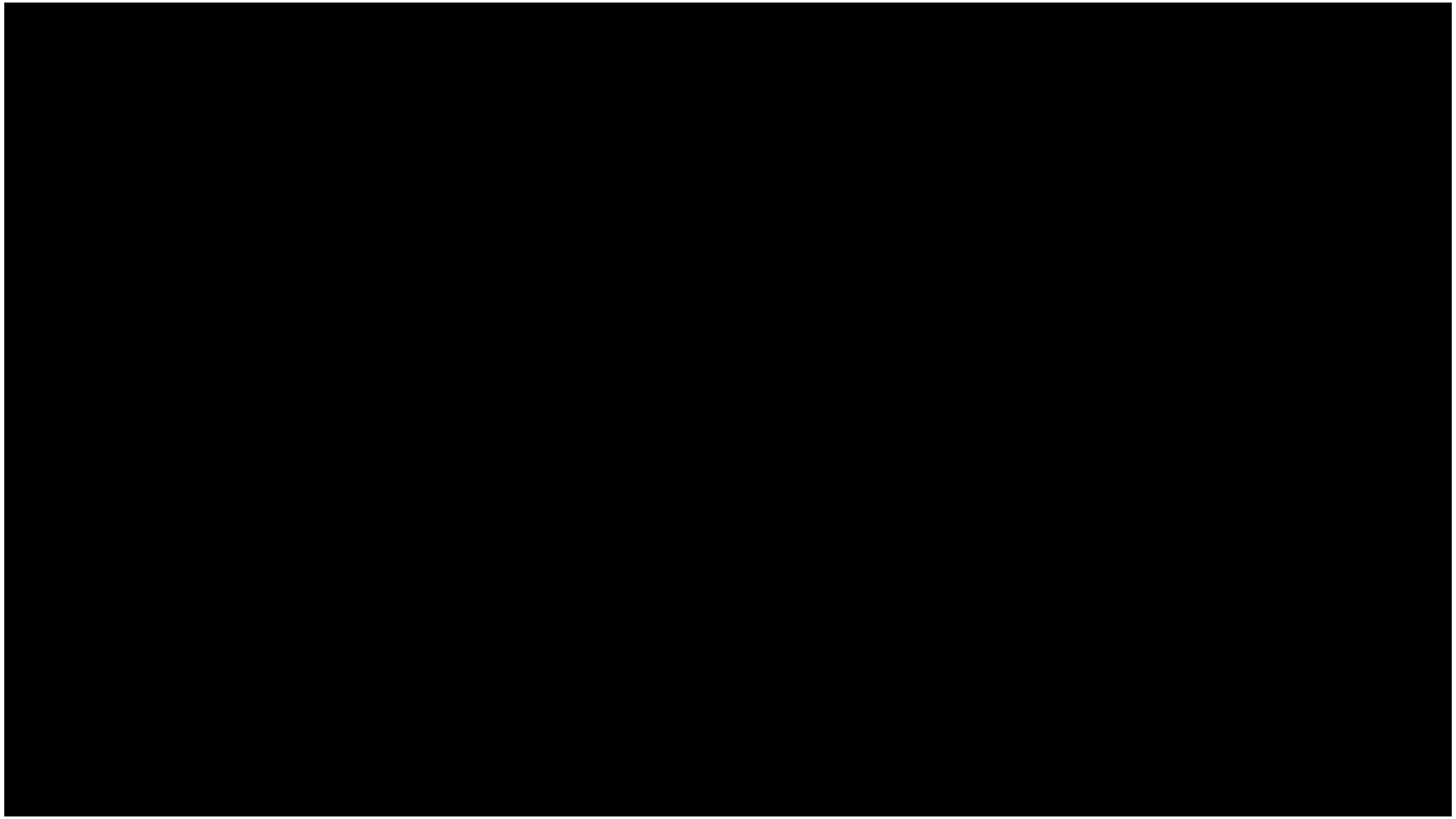
- Entertainment, toys
- Personal services
- Medical, surgery
- Industrial automation (mining, harvesting, …)
- Hazardous environments (space, underwater)

Robotics Today





Mobo Evo – Tim KRSBI UNY @ FIRA Korea



Mobile Manipulation

Asimov's Three Laws of Robotics

1. Robot tidak boleh melukai manusia, atau, karena kelambanan, membiarkan manusia datang untuk menyakiti.
2. Robot harus mematuhi perintah yang diberikan oleh manusia kecuali jika perintah tersebut akan bertentangan dengan hukum pertama.
3. Robot harus melindungi keberadaannya sendiri selama perlindungan tersebut tidak bertentangan dengan hukum pertama atau kedua.

[Runaround, 1942]

Trends in Robotics Research

- Classical Robotics (mid-70's)
  - exact models
  - no sensing necessary
- Reactive Paradigm (mid-80's)
  - no models
  - relies heavily on good sensing
- Hybrids (since 90's)
  - model-based at higher levels
  - reactive at lower levels
- Probabilistic Robotics (since mid-90's)
  - seamless integration of models and sensing
  - inaccurate models, inaccurate sensors

Components of Typical Robots

cameras

sonars

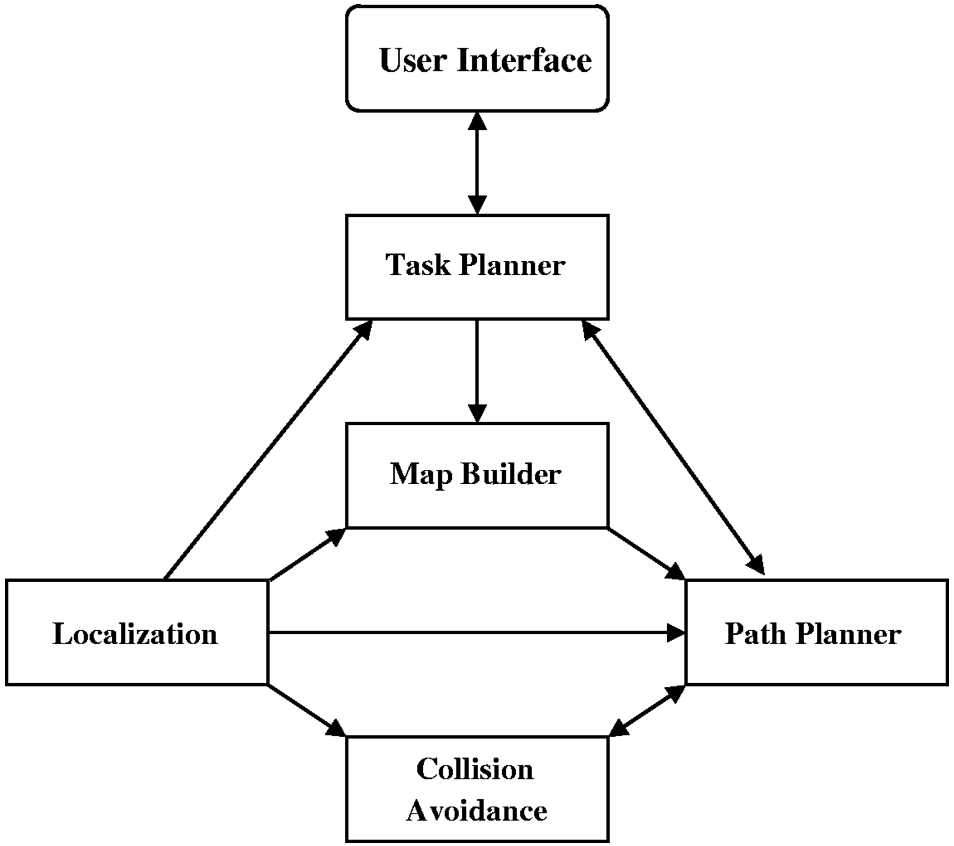
laser

sensors

base

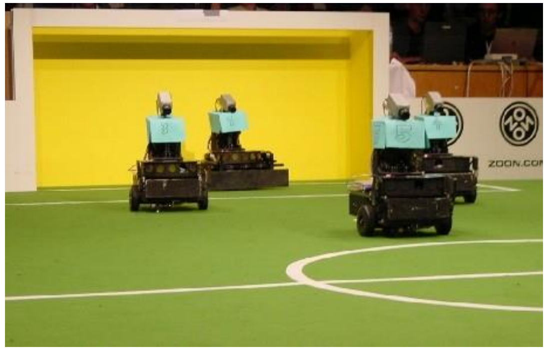
actuators

Architecture of a Typical Control System



Foundations of Artificial Intelligence

- Action Planning: Theory and Practice
  - Fast planning systems (intern. competitions!)
  - Applications at airports and for lift systems
  - Theoretical results (see new Russell/Norvig)
  - SFB AVACS
- Qualitative Temporal-Spatial Reasoning
  - Theory and reasoning algorithms
  - Application in qualitative layout description
  - SFB “Spatial Cognition”
- RoboCup
  - World champion three times
  - Autonomous table soccer
  - RoboCup Rescue
    (Multi-Agent-System for disaster relief)



Autonomous Intelligent Systems

- Mobile robots
- State estimation and models
- Adaptive techniques and learning
- Multi-robot systems
- Applications of mobile robots
- Robots and embedded systems
- Interaction and Web interfaces
- Probabilistic robotics









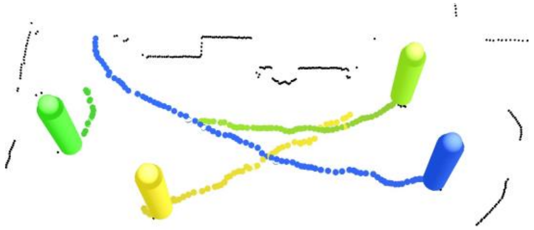

Machine Learning

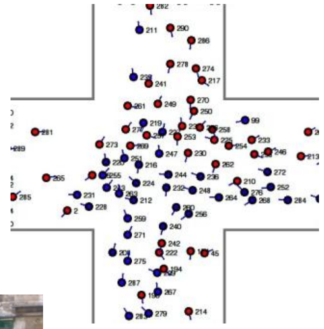
- Reinforcement Learning
- Supervised Learning
- Efficient Learning Algorithms
- Learning in Multi-Agent systems
- Self-learning robots
- Neural Forecasting Systems
- Neural Controllers
- Learning soccer robots in RoboCup
- Industrial Applications

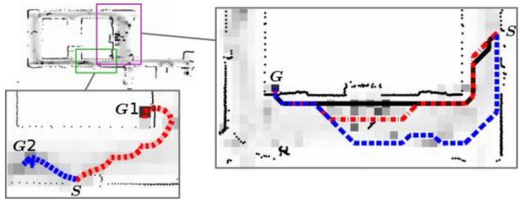
Social Robotics

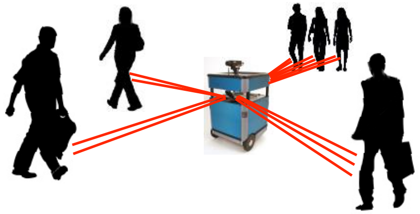
- Towards socially compatible robots
- Social learning, learning by observation
- People detection and tracking
- Motion planning
- Robot navigation
- Spatio-temporal models of human social behavior
- Human–robot interaction

"Free robots from their social isolation"







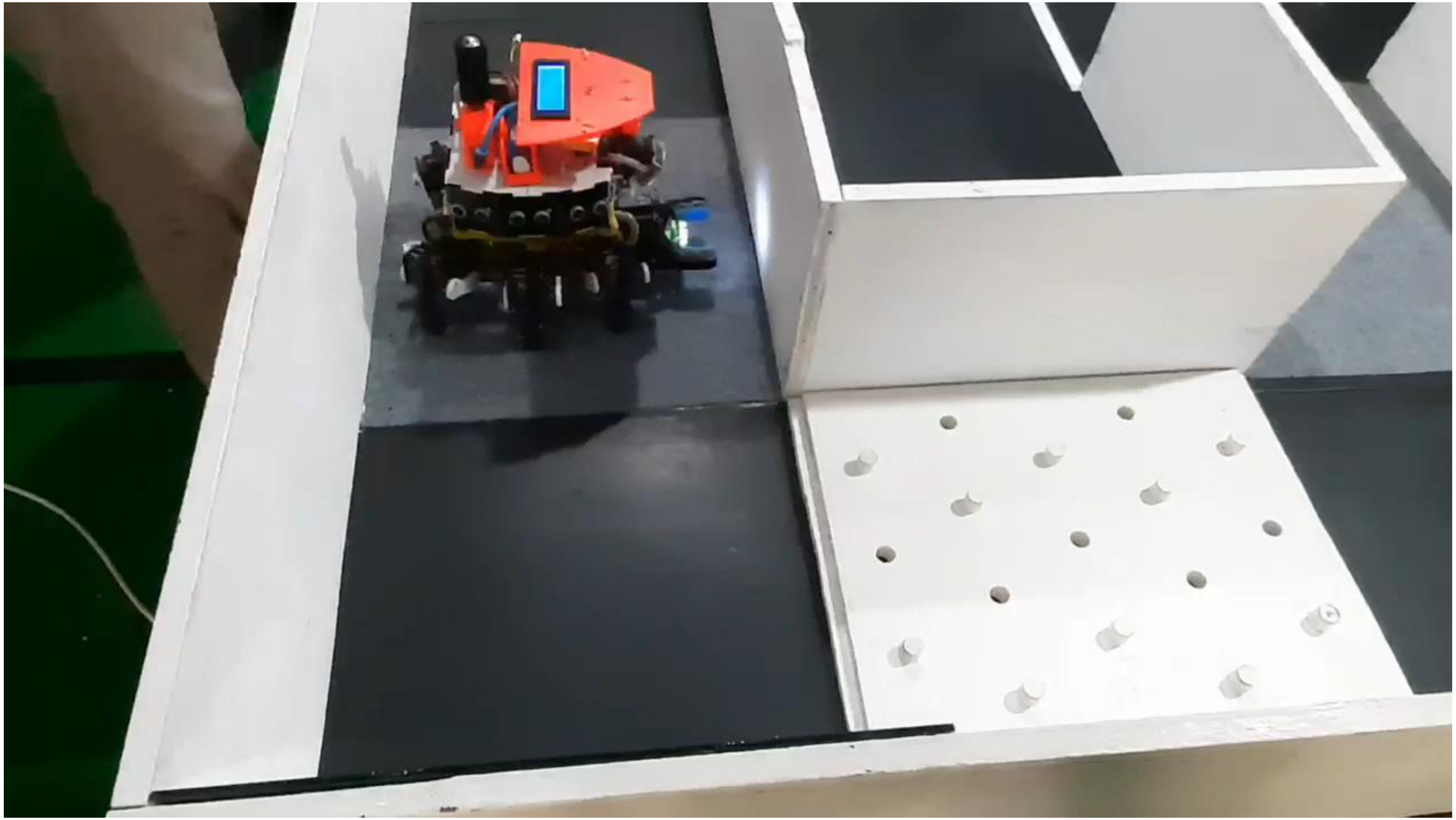


Mobile Robot

Product demo

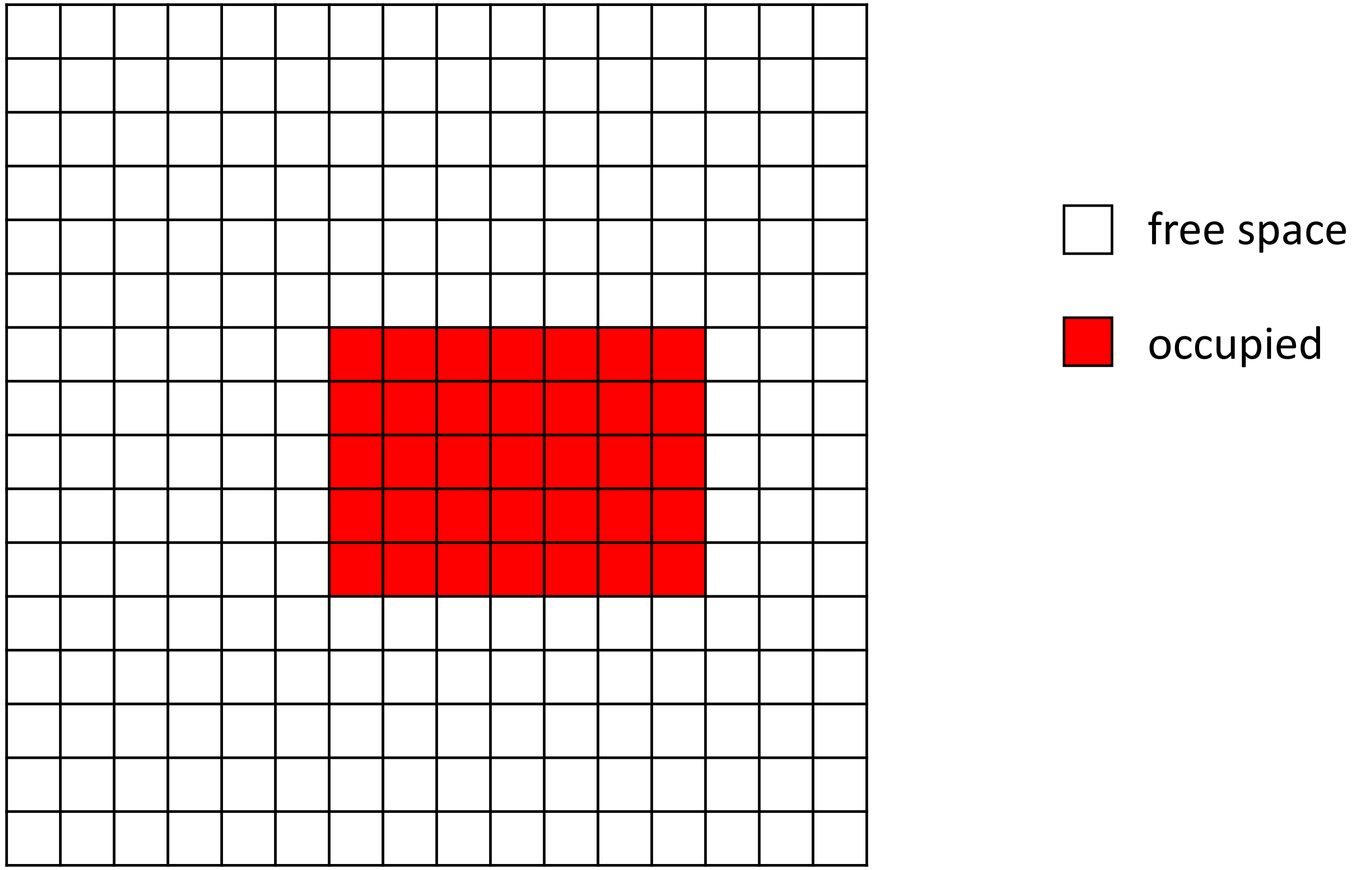
Path Planning

Example SAR Robot



Spatial Decomposition

- represent space itself, rather than the objects in it, using discrete samples
- many ways to perform sampling, but the simplest is to use a grid



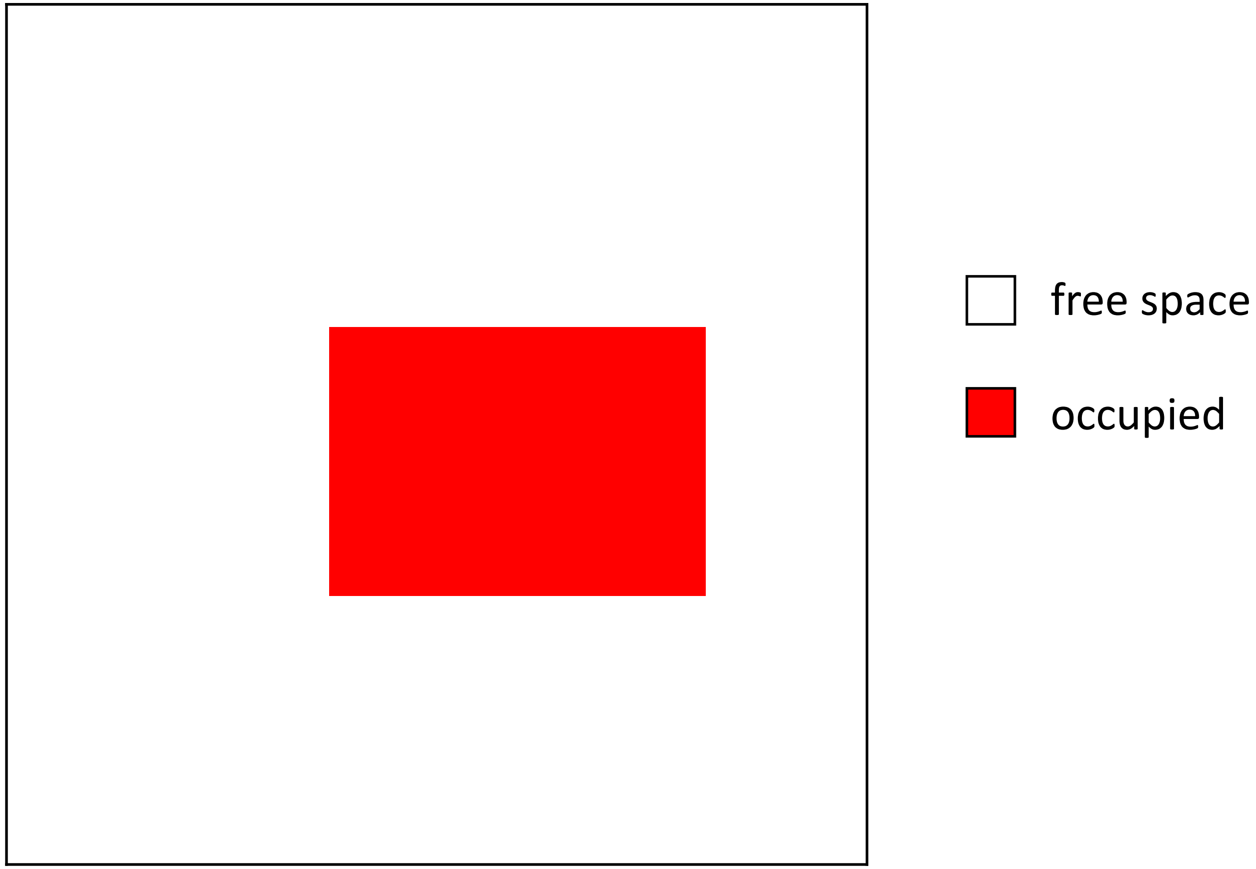
Uniform Spatial Sampling

- very general representation
  - grid locations can represent anything
- if something moves then the representation does not change dramatically
- limited by grid resolution
  - large cell size gives a coarse representation
  - small cell size is storage intensive
    - football pitch at $1cm^2$ resolution
      - 105m x 68m x 100 x 100 = 71,400, 000 cells
    - 3D is much worse

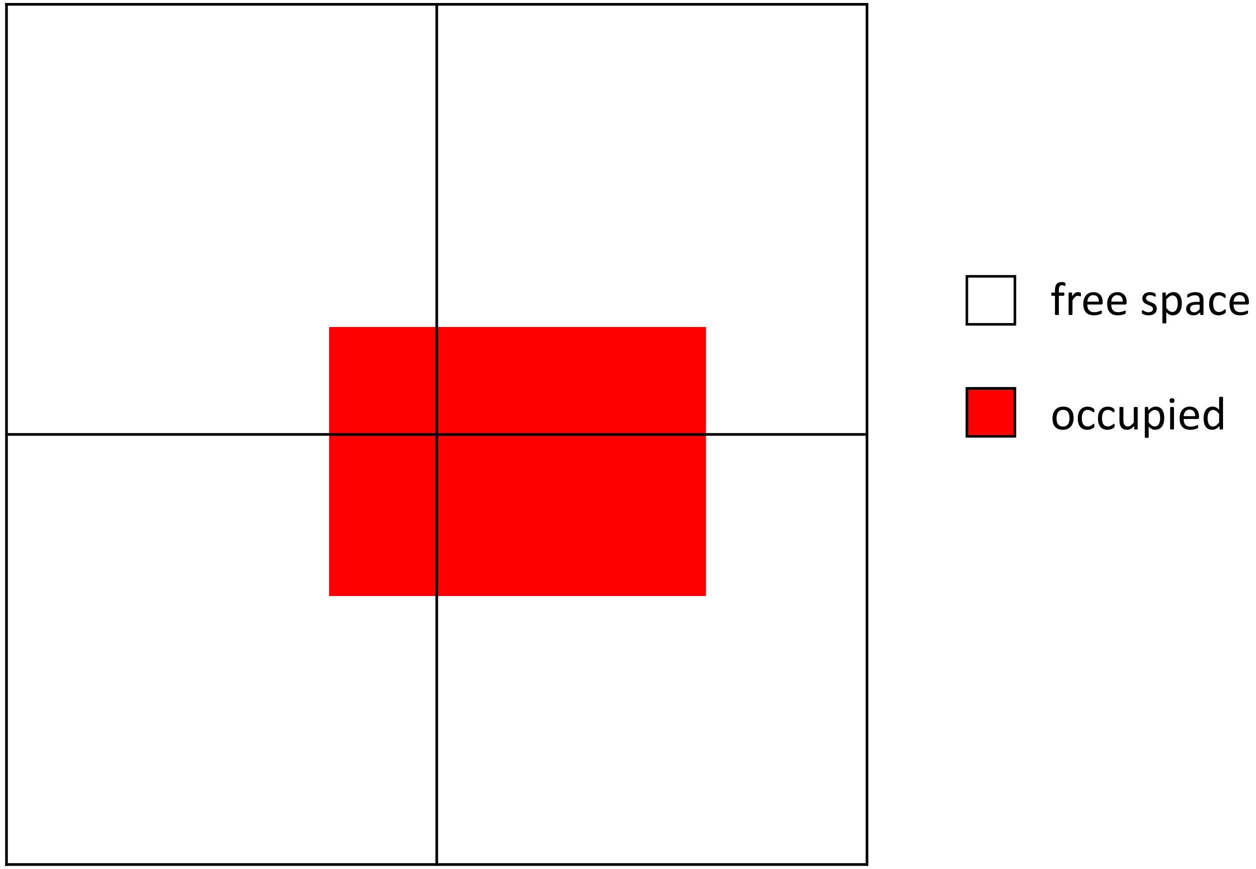
Recursive Hierarchical Representations

- storage space can be conserved by observing that free space cells and occupied cells tend to cluster
  - group the clusters into larger cells
- quadtree
  - recursively subdivide space into 4 equal-sized cells until every cell is either uniformly free or uniformly occupied
    - or some threshold resolution is reached

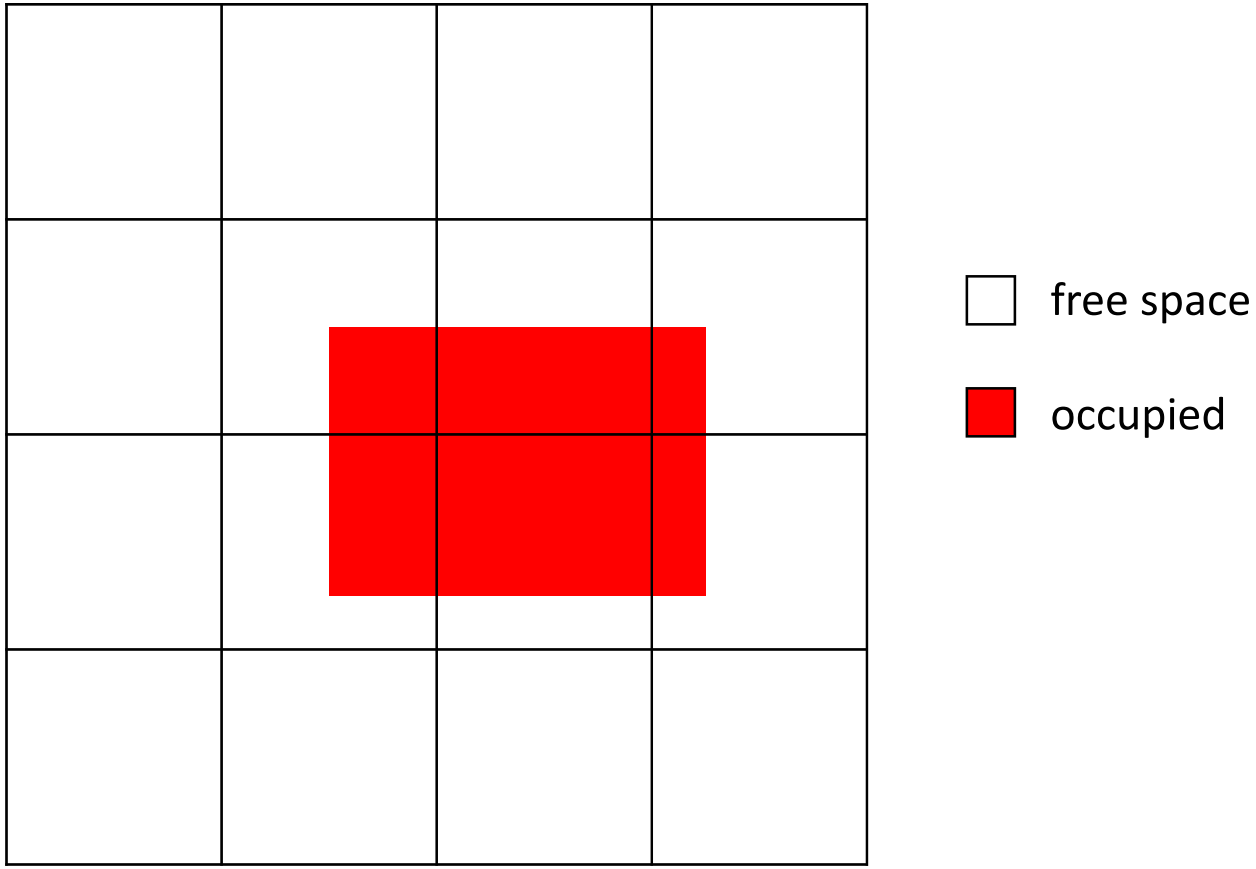
Quadtree Decomposition



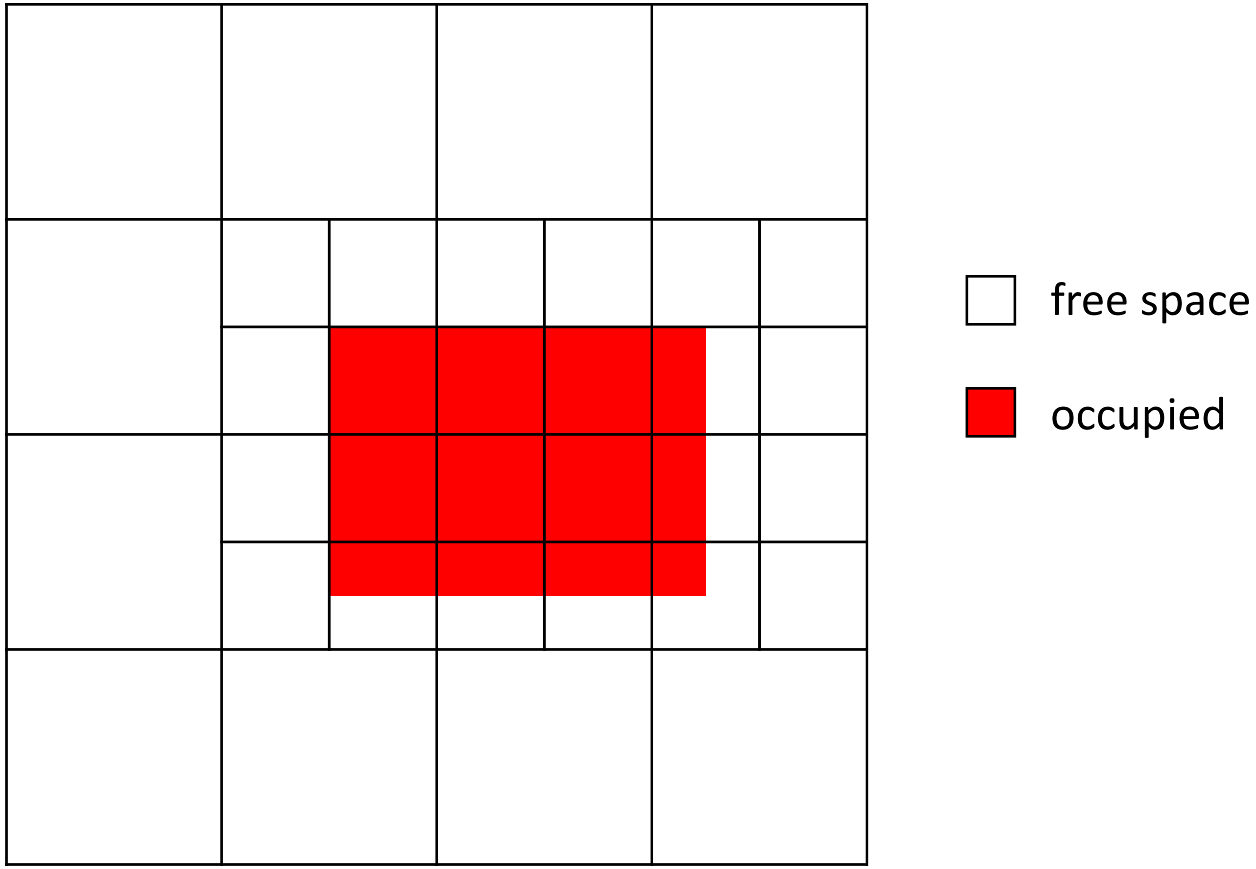
Quadtree Decomposition



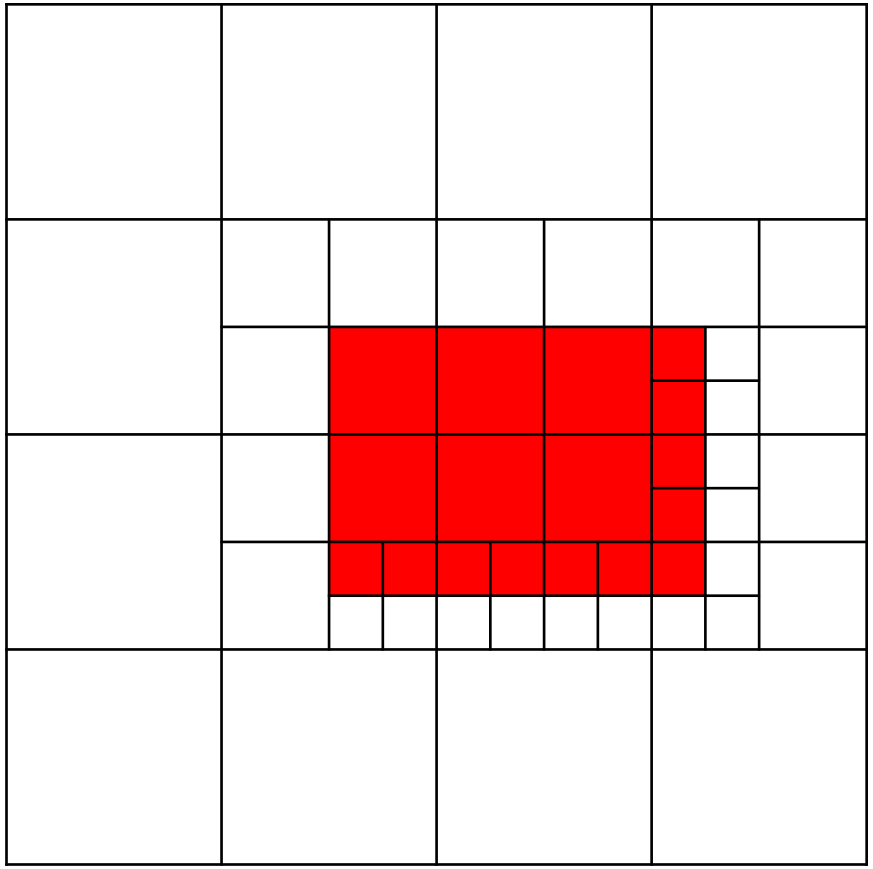
Quadtree Decomposition



Quadtree Decomposition



Quadtree Decomposition



free space

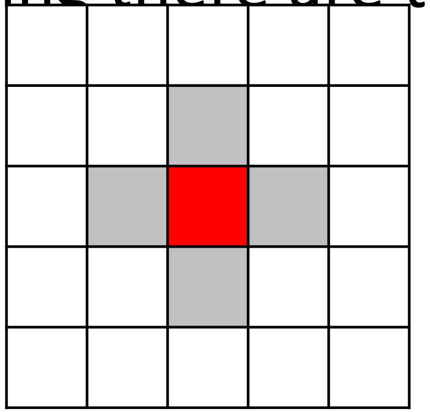
occupied

Quadtree Decomposition

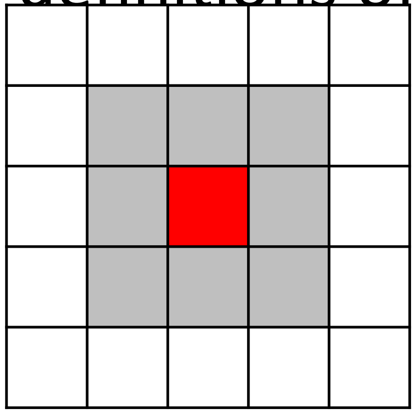
- worst case performance
  - same as uniform subdivision
- if most of the space is occupied or freespace then the representation is compact
- generalizes to N dimensions
- representation changes dramatically if objects move even a small amount

Connectivity in Discrete Sampled Space

- a path on a discrete grid is a sequence of moves between connected cells
- for a square tiling there are two possible definitions of connectivity



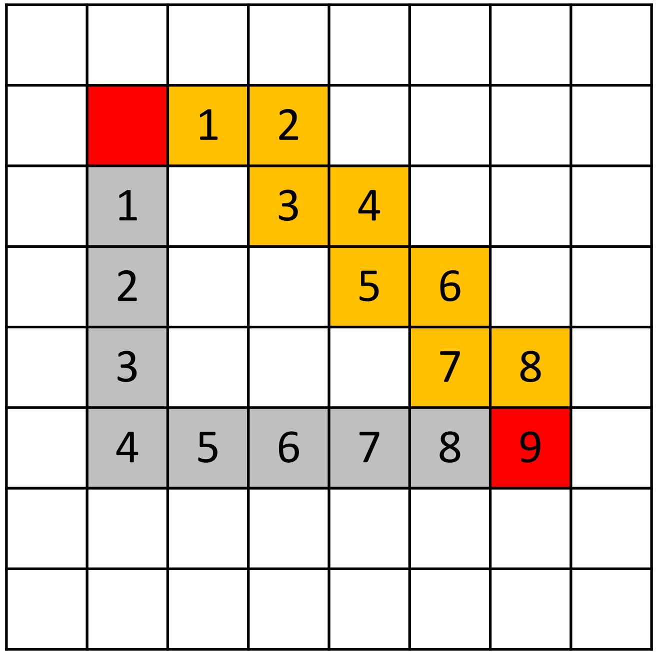
4-connectivity



8-connectivity

4-Connectivity

- on a 4-connected tiling the distance between two cells is called the taxicab distance, rectilinear distance, L_1 distance, L_1 norm, city block distance, or Manhattan distance



Wave-Front Planner

- the wave-front planner finds a path between a start and goal point in spaces represented as a grid where
  - free space is labeled with a 0
  - obstacles are labeled with a 1
  - the goal is labeled with a 2
  - the start is known

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

goal

Wave-Front Planner

- starting at the goal cell

```
L := 2
while start cell is unlabelled
    for each cell C with label L
        for each cell Z connected to C with label 0
            label Z with L+1
    L := L + 1
```

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	14	13	12	11	10	9	8	7	6	5	4	3	2
	0	0	0	0	14	13	12	11	10	9	8	7	6	5	4	3
	0	0	1	1	0	14	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	0	20	1	1	17	16	17	18	19	20	0	0	1	1	0	0
	1	1	1	1	18	17	18	19	20	0	0	0	1	1	0	0
	1	1	1	1	19	18	19	20	0	0	0	0	1	1	0	0
	0	0	1	1	20	19	20	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
	1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
	1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
	0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
	0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
	0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
	0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
	0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	33	34
	0	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
	50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
goal	*	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37

Wave-Front Planner

- to generate a path starting from the start point

L := start point label

while not at the goal

  move to any connected cell with label L-1

  L := L-1

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
	1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
	1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
	0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
	0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
	0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
	0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
	0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
	0	51	1	1	1	1	1	1	1	1	1	1	1	1	33	34
	51	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
	50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
goal	51	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37

Another Example

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	2	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	3	4	0	0	0	0	0	0
0	0	0	0	0	0	4	3	2	3	4	0	0	0	0	0
0	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

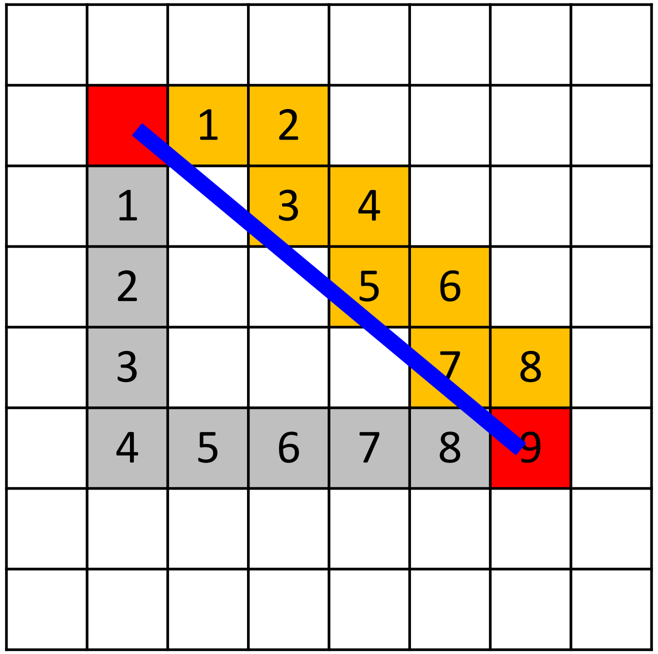
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	7	6	7	0	0	0	0	0	0
0	0	0	0	0	0	7	6	5	6	7	0	0	0	0	0
0	0	0	0	0	7	6	5	4	5	6	7	0	0	0	0
0	0	0	0	7	6	5	4	3	4	5	6	7	0	0	0
0	0	0	7	6	5	4	3	2	3	4	5	6	7	0	0
0	0	0	0	7	6	5	4	3	4	5	6	7	0	0	0
0	0	0	0	0	7	6	5	4	5	6	7	0	0	0	0
0	0	0	0	0	0	7	6	5	6	7	0	0	0	0	0
0	0	0	0	0	0	0	7	6	7	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Wave-Front Planner

- advantage:
  - will find a shortest path (in terms of connectivity) between start and goal if a path exists
  - generalizes to higher dimensions
- disadvantages:
  - path often runs adjacent to obstacles
  - planner searches the entire space with radius R around the goal (where R is the distance between the start and goal)
  - paths restricted by grid connectivity are longer than necessary

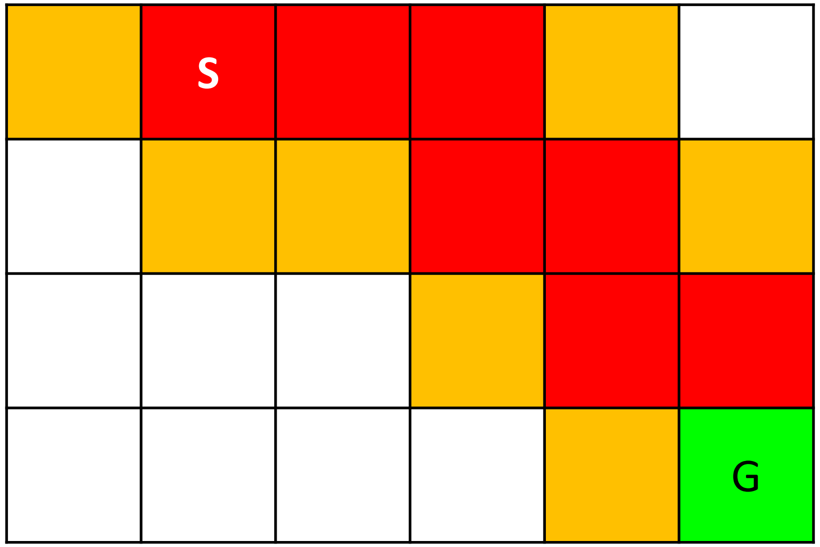
Wave-Front Planner

- paths restricted by grid connectivity are longer than necessary
  - Manhattan distance = 9
  - straight line distance = sqrt(16 + 25) = 6.403…



Greedy Best-First Search

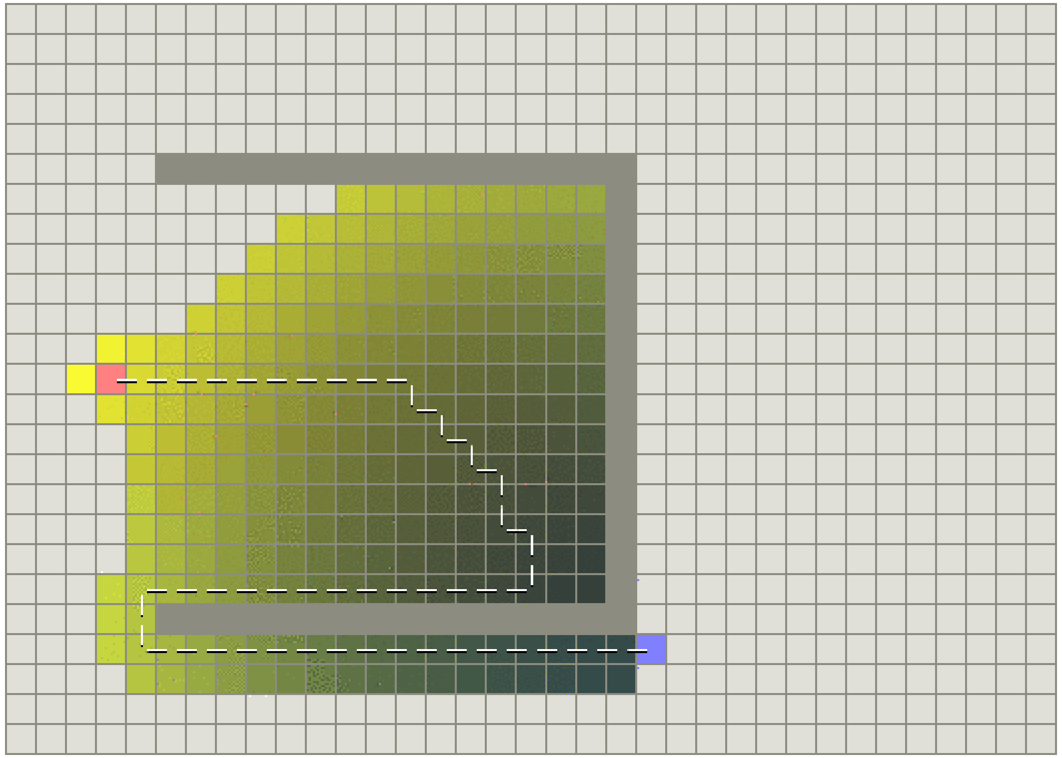
- to avoid searching in all possible directions we might consider searching first in a direction towards the goal
- idea
  - use an estimate (called the heuristic) of how far a cell is from the goal
  - consider the cell whose heuristic distance is the smallest first



possible path with heuristic distance = Euclidean distance

Greedy Best-First Search

- produces expensive paths when there are concave obstacles



http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

A*

- A* is a common algorithm in game AI programming and robotics
  - first described in 1968
  - http://theory.stanford.edu/~amitp/GameProgramming/
- A* is the foundation for Theta*
  - Daniel, Nash, Koenig. Theta*: Any-Angle Planning on Grids, Journal of Artificial Intelligence Research, 39, 2010.
  - path planning on a grid where paths are allowed to pass through cells at any angle (not just using 4- or 8-connectivity)

A*

- A* combines two pieces of information
  - g(n) : the cost of the path from the starting point to n
  - h(n) : the heuristic cost of the path from n to the goal
  - considers the cell n with the lowest cost

    $$f(n) = g(n) + h(n)$$

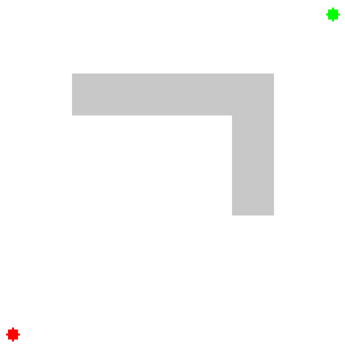
    first
- compromise between Dijkstra’s algorithm and greedy best-first search

A*

- the heuristic distance function h(n) affects how the algorithm performs the search
  - h(n) = 0
    - equivalent to Dijkstra's algorithm
  - h(n) <= true cost of moving from n to the goal
    - guaranteed to find a shortest path
    - the smaller h(n) the more it expands the search to cells closer to the start
  - h(n) = true cost of moving from n to the goal
    - will find a best path with the minimal amount of searching
  - h(n) > true cost of moving from n to the goal some of the time
    - not guaranteed to find a shortest path but might find a path in a shorter amount of time
  - h(n) >> true cost of moving from n to the goal
    - behaves like greedy best-first search

A*

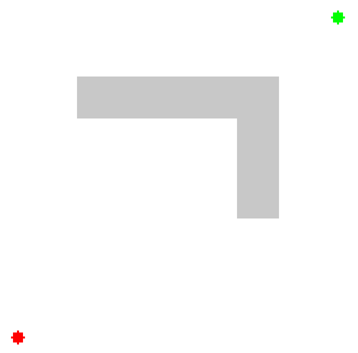
- h(n) <= true cost of moving from n to the goal



http://en.wikipedia.org/wiki/File:Astar_progress_animation.gif

A*

- h(n) > true cost of moving from n to the goal



http://en.wikipedia.org/wiki/File:Weighted_A_star_with_eps_5.gif

Potential Functions

- in continuous space potential functions can be used for path planning
- a potential function is a differentiable real-valued function U

$$U : \mathrm{R}^m \to \mathrm{R}$$

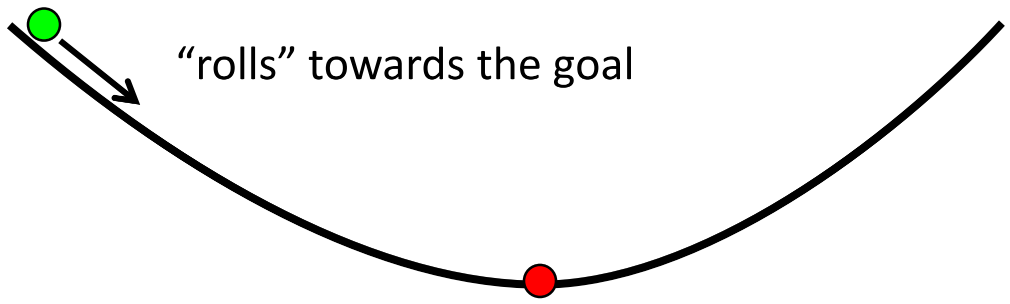
  - i.e., U assigns a scalar real value to every point in space
- potential functions you might know
  - gravitational potential
  - electrostatic potential

Goal Potential

- the goal potential should be an attractive potential
  - small near the goal
  - large far from the goal
  - monotonically increasing
    - nice too if it is continuously differentiable

Goal Potential

- consider the quadratic potential $U_{\text{attract}} = \alpha \| q - q_{\text{goal}} \|^2$



Goal Potential

- "rolling towards the goal" can be accomplished using gradient descent

$$
\begin{aligned}
F &= \nabla U_{\text{attract}} \\
&= \begin{bmatrix} \partial U / \partial x \\ \partial U / \partial y \end{bmatrix} \\
&= \alpha\left(q - q_{\text{goal}}\right)
\end{aligned}
$$

- gradient descent
  - starting at initial configuration, take a small step in the direction opposite to the gradient F until $|F| = 0$

Goal Potential

- notice that the wave-front planner basically works this way
  - it defines a potential where there is only one minimum
    - the minimum is located at the goal
  - it then uses gradient descent to move towards the goal

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
0	51	1	1	1	1	1	1	1	1	1	1	1	1	33	34
51	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
51	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37

start

Simultaneous Localization & Mapping – SLAM

Difficulties of Autonomy for Large Size Petrochemical Complexes Fire Disaster

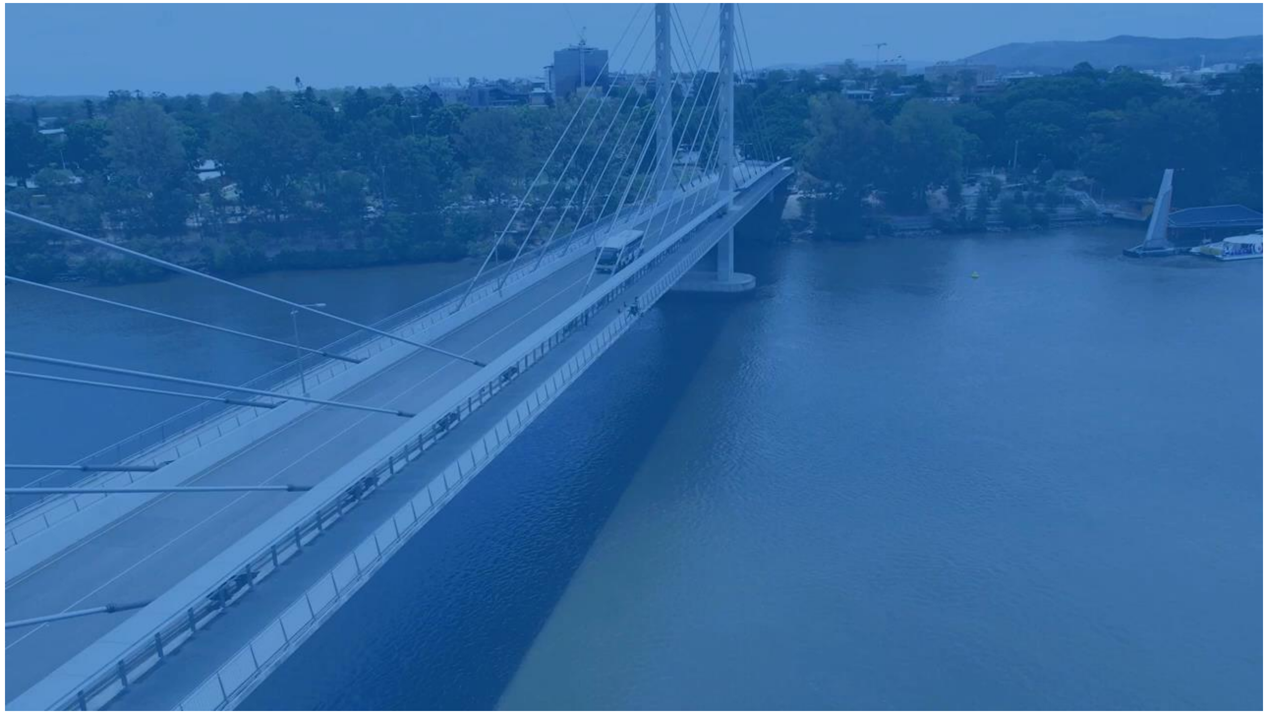
A. Petrochemical complexes:
   - Large (1000 m x 600 m).
   - Dynamic.

B. Fire disaster:
   - Fire.
   - Water mist.
   - Smoke.

C. Human Operator:
   - Map consistency btw. robot map with aerial map.

D. Firefighter robot constrains:
   - Water-hose-layout
   - Multi-robot.

Result of SLAM based on GPS and LIDAR



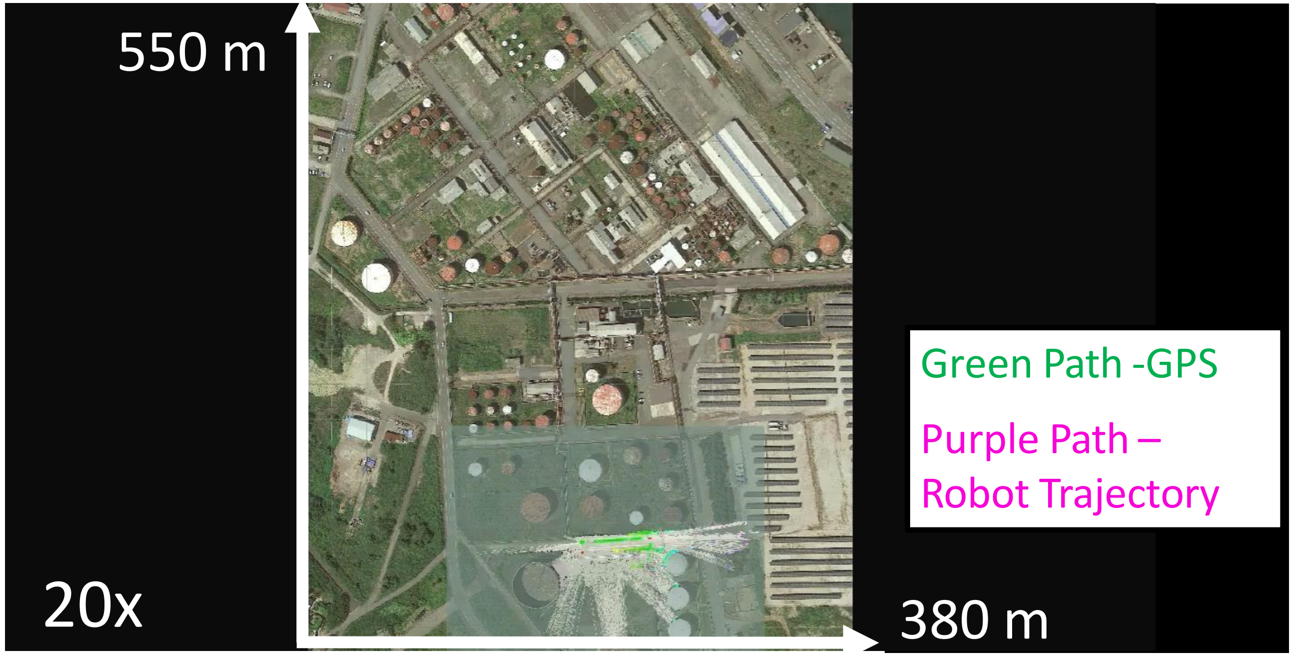
- Parameters for RBPFs SLAM :
  - 50 particles.
  - 0.2 m grid map.

Green-GPS

Purple-Path

Process of SLAM based on GPS and LIDAR in the bridge and river area.

Result: Real-time consistent mapping using FS 2.0 based on GPS and LIDAR



Real-time consistent mapping : (Using 50 particles)

1. **Localization (Real-time)** : 0.01 - 0.04 s
2. **Parallel process**
   - Localization correction & mapping : 0.25 - 0.42 s
   - Whole map update time : 0.02 - 3.08 s

Result satisfy real-time SLAM requirement

NUT SHELL

- SLAM is a technique used to build up a map within an unknown environment or a known environment while at the same time keeping track of the current location.

To a human

- Assume you are blindfolded in a room

What is SLAM?

- o The problem has 2 stages
  - Mapping
  - Localization
- o The paradox:
  - In order to build a map, we must know our position
  - To determine our position, we need a map!
- o SLAM is like the chicken-egg problem
- o Solution is to alternate between the two steps.

SLAM – Multiple parts

- Landmark extraction
- data association
- State estimation
- state update
- landmark update

There are many ways to solve each of the smaller parts

Hardware

- Mobile Robot
- Range Measurement Device
  - Laser scanner – CANNOT be used underwater
  - Sonar – NOT accurate
  - Vision – Cannot be used in a room with NO light

The goal of the process

The SLAM process consists of number of steps.

- Use environment to update the position of the robot. Since the odometry of the robot is often erroneous we cannot rely directly on the odometry.
- We can use laser scans of the environment to correct the position of the robot.
- This is accomplished by extracting features from the environment and re observing when the robot moves around.

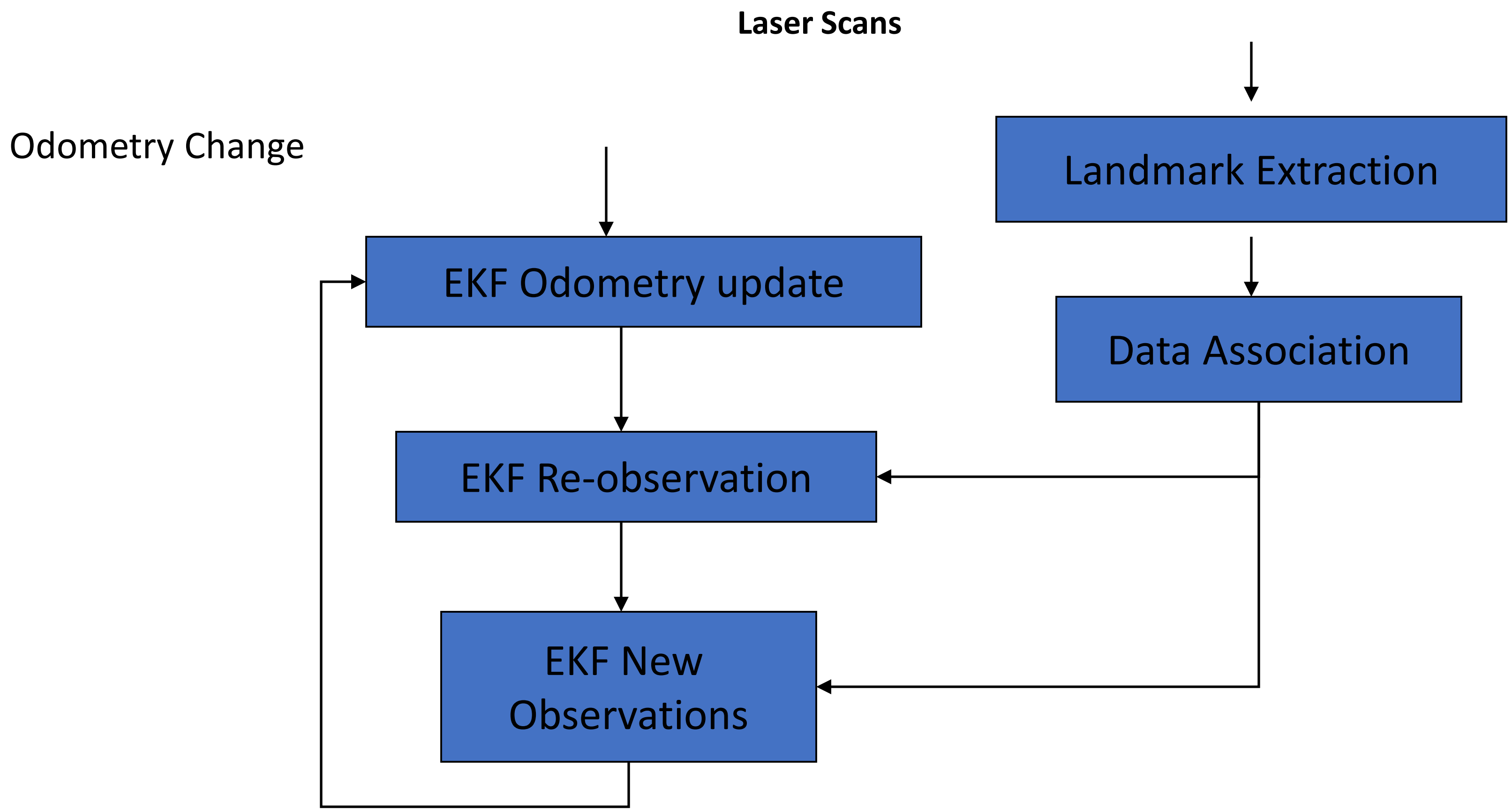
Extended Kalman Filter

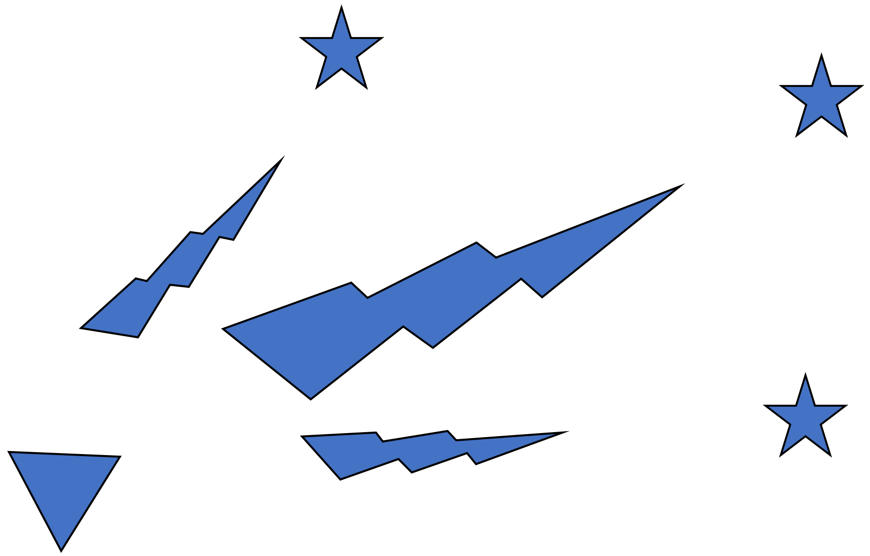
An EKF (Extended Kalman Filter) is the heart of the SLAM process.

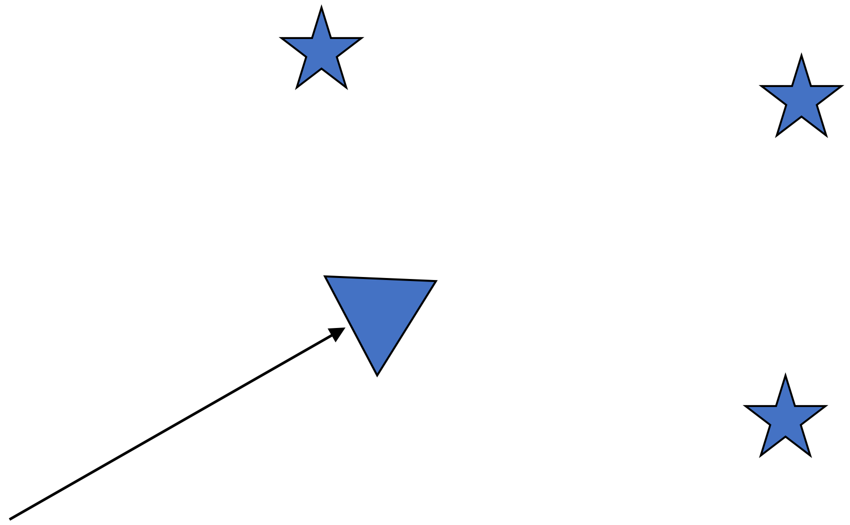
- It is responsible for updating where the robot thinks it is based on the Landmarks (features).

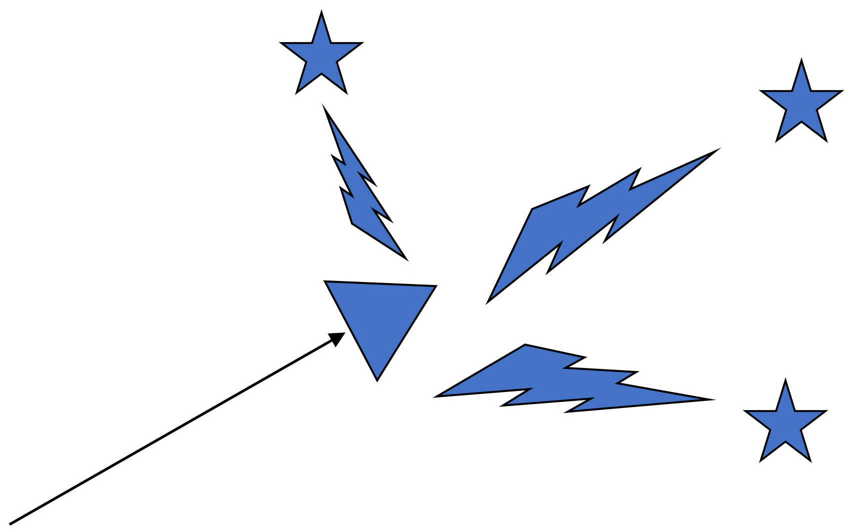
- The EKF keeps track of an estimate of the uncertainty in the robots position and also the uncertainty in these landmarks it has seen in the environment.

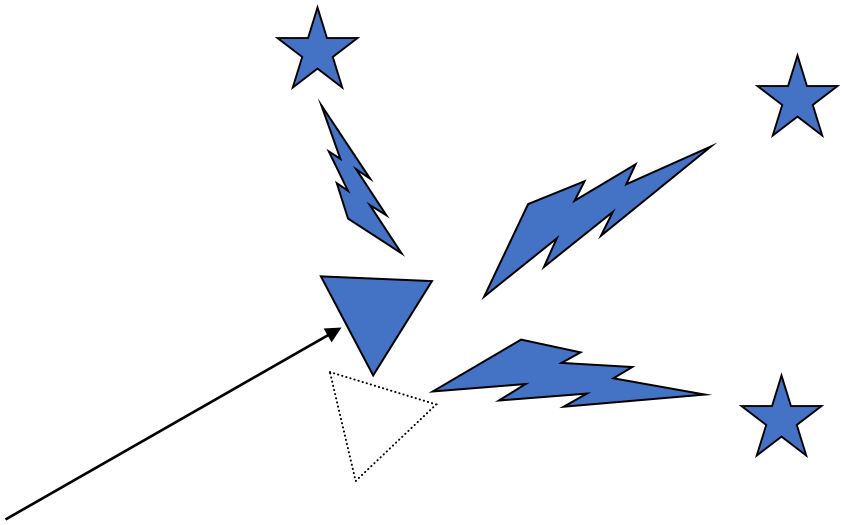
Overview

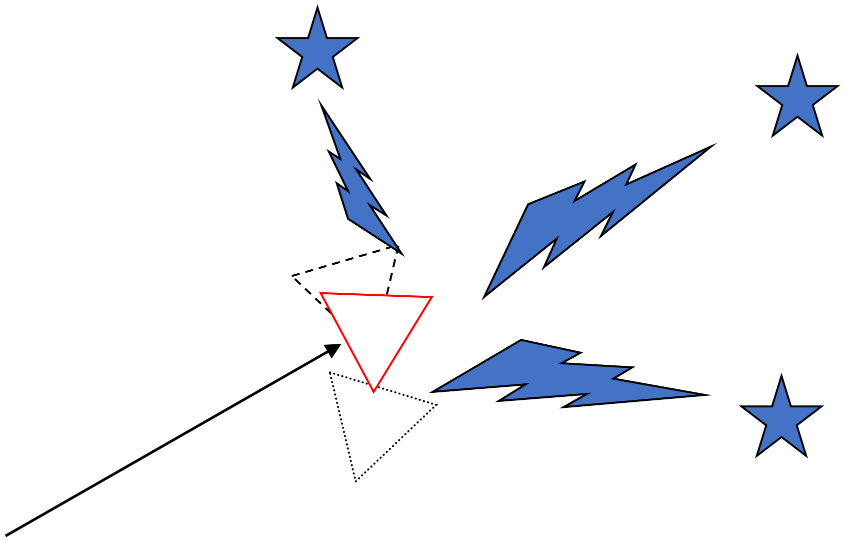

Laser Scans
Odometry Change
Landmark Extraction
EKF Odometry update
Data Association
EKF Re-observation
EKF New Observations











Laser & Odometry data

- Laser data is the reading obtained from the scan
- The goal of the odometry data is to provide an approximate position of the robot
- The difficult part about the odometry data and the laser data is to get the timing right.

Landmarks

- Landmarks are features which can easily be re-observed and distinguished from the environment.
- These are used by the robot to find out where it is (to localize itself).

The key points about suitable Landmarks

- Landmarks should be easily re-observable.
- Individual landmarks should be distinguishable from each other.
- Landmarks should be plentiful in the environment.
- Landmarks should be stationary.

In an indoor environment such as that used by our robot there are many straight lines and well defined corners. These could all be used as landmarks.

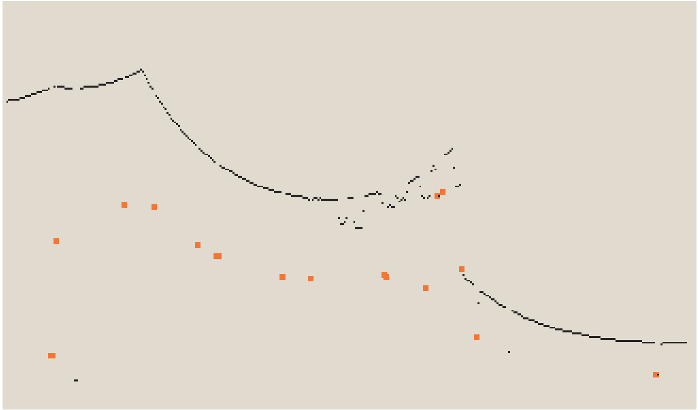
Landmark Extraction

- Once we have decided on what landmarks a robot should utilize we need to be able to somehow reliably extract them from the robots sensory inputs.
- The 2 basic Landmark Extraction Algorithms used are Spikes and RANSAC

Spike

- The spike landmark extraction uses extrema to find landmarks.

- when some of the laser scanner beams reflect from a wall and some of the laser scanner beams do not hit this wall, but are reflected from some things further behind the wall.

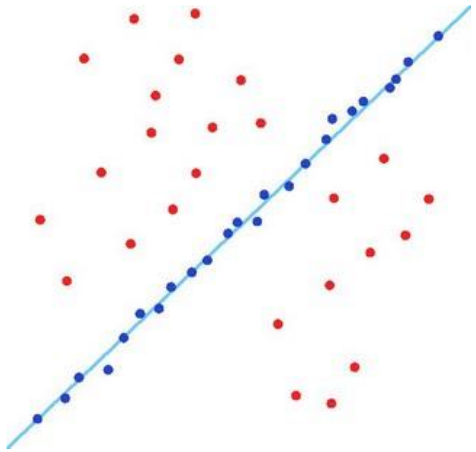


Spike landmarks. The red dots are table legs extracted as landmarks.

Spike landmarks rely on the landscape changing a lot between two laser beams. This means that the algorithm will fail in smooth environments.

RANSAC (Random Sampling Consensus)

- This method can be used to extract lines from a laser scan that can in turn be used as landmarks.
- RANSAC finds these line landmarks by randomly taking a sample of the laser readings and then using a least squares approximation to find the best fit line that runs through these readings.



Consensus

Data Association

- The problem of data association is that of matching observed landmarks from different (laser) scans with each other.

- Problems in Data Association
  - You might not re-observe landmarks every time.
  - You might observe something as being a landmark but fail to ever see it again.
  - You might wrongly associate a landmark to a previously seen landmark.

Algorithm – Nearest Neighbour Approach

- When you get a new laser scan use landmark extraction to extract all visible landmarks.
- Associate each extracted landmark to the closest landmark we have seen more than N times in the database.
- Pass each of these pairs of associations (extracted landmark, landmark in database) through a validation gate.
  - If the pair passes the validation gate it must be the same landmark we have re-observed so increment the number of times we have seen it in the database.
  - If the pair fails the validation gate add this landmark as a new landmark in the database and set the number of times we have seen it to 1.

Overview of the process

- Update the current state estimate using the odometry data
- Update the estimated state from re-observing landmarks.
- Add new landmarks to the current state.

Final Review – Open Areas

- There is the problem of closing the loop. This problem is concerned with the robot returning to a place it has seen before. The robot should recognize this and use the new found information to update the position.
- Furthermore the robot should update the landmarks found before the robot returned to a known place, propagating the correction back along the path.

References

- Slam for dummies, by Soren Riisgaard & Morten Rufus Blas
- Wikipedia - Slam
- Minimal Slam for Efficient Floor-Planning, by Stephen Pfetsch

http://farm1.static.flickr.com/34/101152162_a59da9b562.jpg

http://3.bp.blogspot.com/_beboLKBKnDc/Ryai4RoJavI/AAAAAAAAACI/oq5h56z9ZzY/s320/Fitted_line.jpg

Case Study_1

Telepresence robot.

Spesifikasi produk

Dimensi	: 58cm x 60cm x 100cm
Bahan Body	: Akrilik 3mm
Bahan Rangka	: Aluminium Plat dan Aluminium Hollow
Mikrokontroler	: Arduino Mega 2560
Mini PC	: Intel NUC 8I3BEH2
Software	: Visual Studio 2015 dan Arduino IDE
Sensor	: Kamera Kinect XBOX 360
Komponen Output	: Mini LED Monitor 10 Inch

Spesifikasi produk

Sumber Tegangan : Baterai Lippo 24V 2 Buah dan Baterai Lippo 12 V

Jenis Kamera : Kamera Kinect XBOX 360

Jenis Aktuator : Motor DC PG45, 24 V

Jenis Roda : *Omni-Directional Wheel*

Jumlah Roda : Tiga Buah

Kecepatan Respon : $\pm$ 30 detik

Jangkauan Deteksi : Dua Meter (Maksimal)

Jangkauan *Object Tracking* : Dua Meter (Maksimal)

Jangkauan *Object Following* : Dua Meter (Maksimal)

Tampilan GUI

Object detection

No.	Jarak	Hasil
1.	0 cm	Objek Tidak Terdeteksi
2.	30 cm	Objek Tidak Terdeteksi
3.	60 cm	Objek Terdeteksi
4.	90 cm	Objek Terdeteksi
5.	120 cm	Objek Terdeteksi
6.	150 cm	Objek Terdeteksi
7.	180 cm	Objek Terdeteksi
8.	210 cm	Objek Terdeteksi
9.	240 cm	Objek Terdeteksi
10.	270 cm	Objek Terdeteksi
11.	300 cm	Objek Terdeteksi
12.	330 cm	Objek Terdeteksi
13.	360 cm	Objek Tidak Terdeteksi

Case Study_2

Autonomous Car.

- Sebastian Thrun’s TED Talk describing Google’s driverless car
    - www.ted.com/talks/sebastian_thrun_google_s_driverless_car.html

- Junior: The Stanford Entry in the Urban Challenge
  - http://robots.stanford.edu/papers/junior08.pdf

DARPA Urban Challenge

- multiple robotic vehicles carrying out missions on the same course at the same time
- basic rules:
  - stock vehicle
  - obey California driving laws
  - entirely autonomous
  - avoid collisions with objects typical to urban environment
  - must also be able to operate in parking lots
  - DARPA supplied environment map with information on lanes, lane markings, stop signs, parking lots, and special checkpoints

Junio

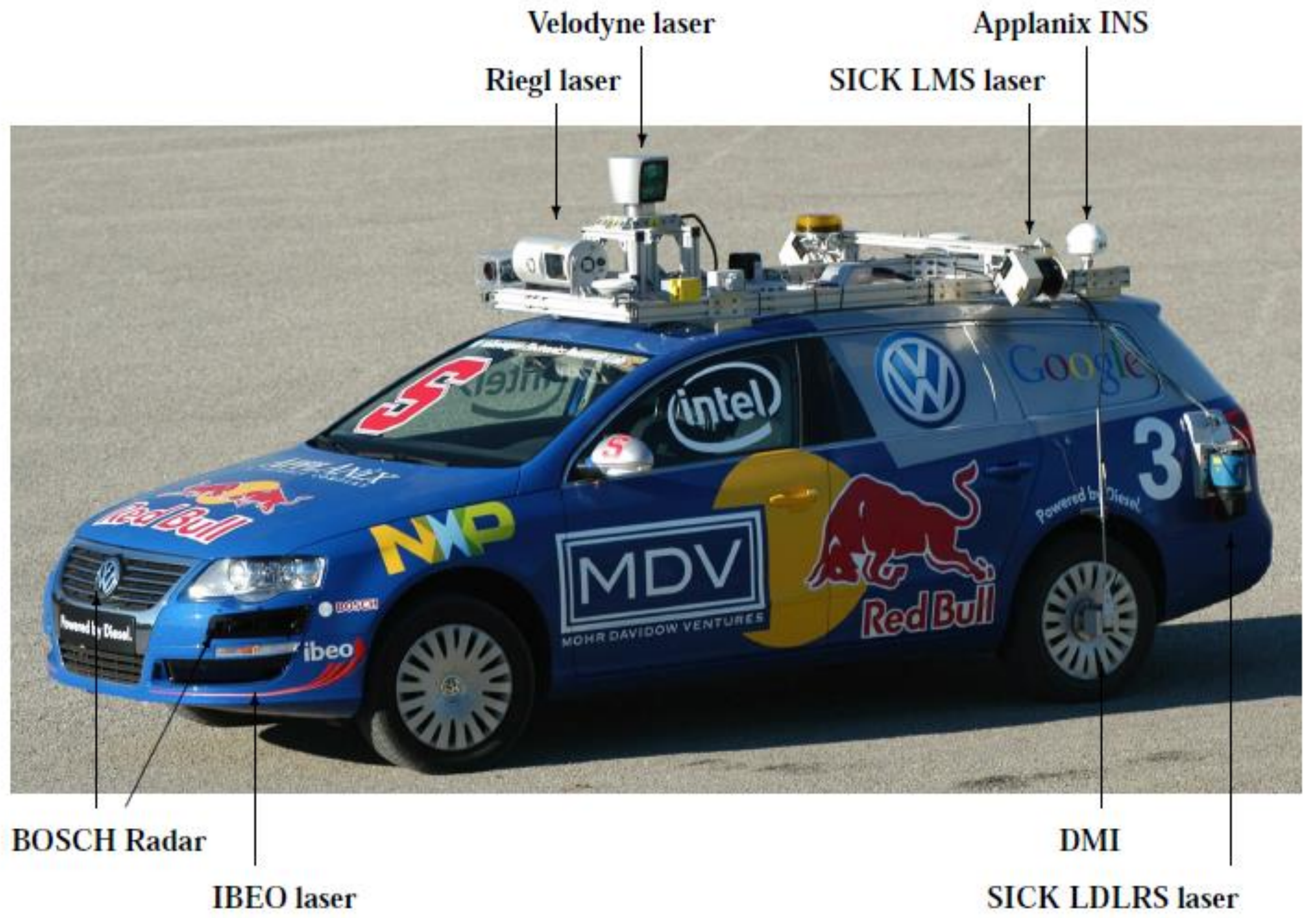


Figure 1: Junior, our entry in the DARPA Urban Challenge. Junior is equipped with five different laser measurement systems, a multi-radar assembly, and a multi-signal inertial navigation system, as shown in this figure.

Junior: Computation

Figure 2: All computing and power equipment is placed in the trunk of the vehicle. Two Intel quad core computers (bottom right) run the bulk of all vehicle software. Other modules in the trunk rack include a power server for selectively powering individual vehicle components, and various modules concerned with drive-by-wire and GPS navigation. A 6 DOF inertial measurement unit is also mounted in the trunk of the vehicle, near the rear axle.

Junior: Sensors

- Applanix POS LV 420 position and orientation system
  - multiple GPS receivers, GPS heading measurement, inertial measurement, distance measurement, Omnistar Virtual Base Station
    - position and orientation errors less than 100 cm and 0.1 degrees
- 2 side facing SICK LMS laser range finders
- 1 forward facing RIEGL LMS-Q120 laser range finder
- 1 roof mounted Velodyne HDL-64E laser range finder
- 2 rear mounted SICK LDLRS laser range finders
- 2 front mounted IBEO ALASCA XT laser range finders
- 5 BOSCH radars mounted in front grill

Laser Obstacle Detection

- challenging!
  - curbs
  - moving and static obstacles
  - ov… iven
    un…

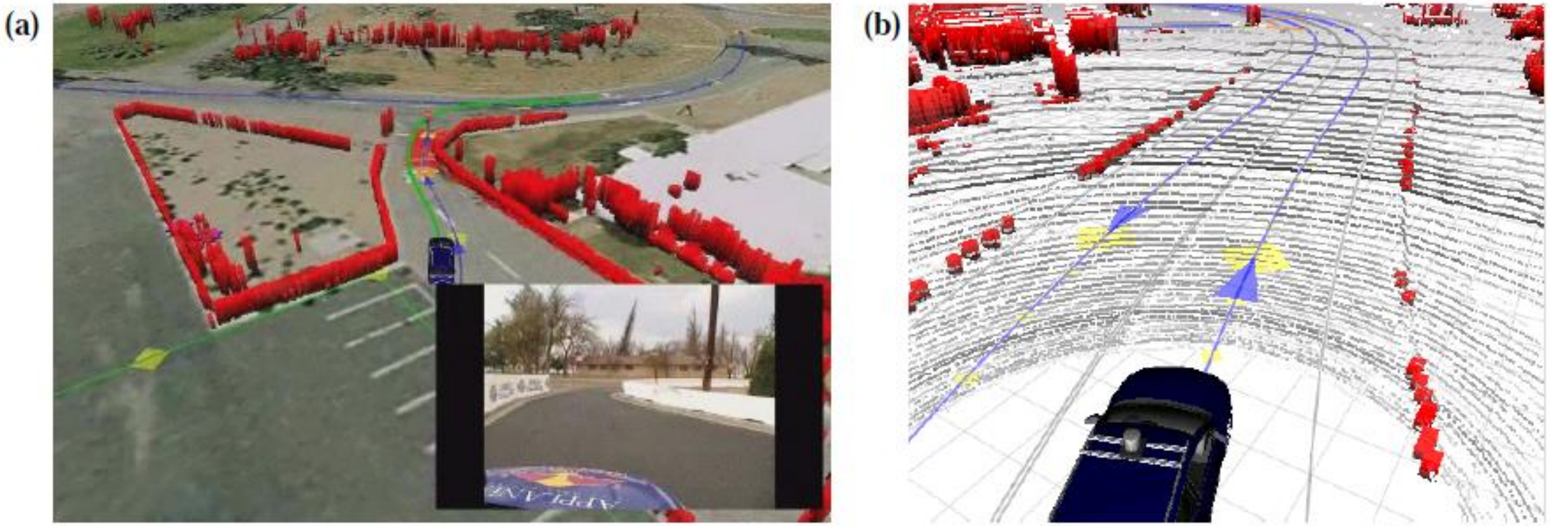


Figure 5: Obstacles detected by the vehicle are overlayed over aerial imagery (left) and Velodyne data (right). In the example on the right, the curbs along both sides of the road are detected.

Laser Obstacle Detection

- primary sensor is the Velodyne laser range finder
  - elf-

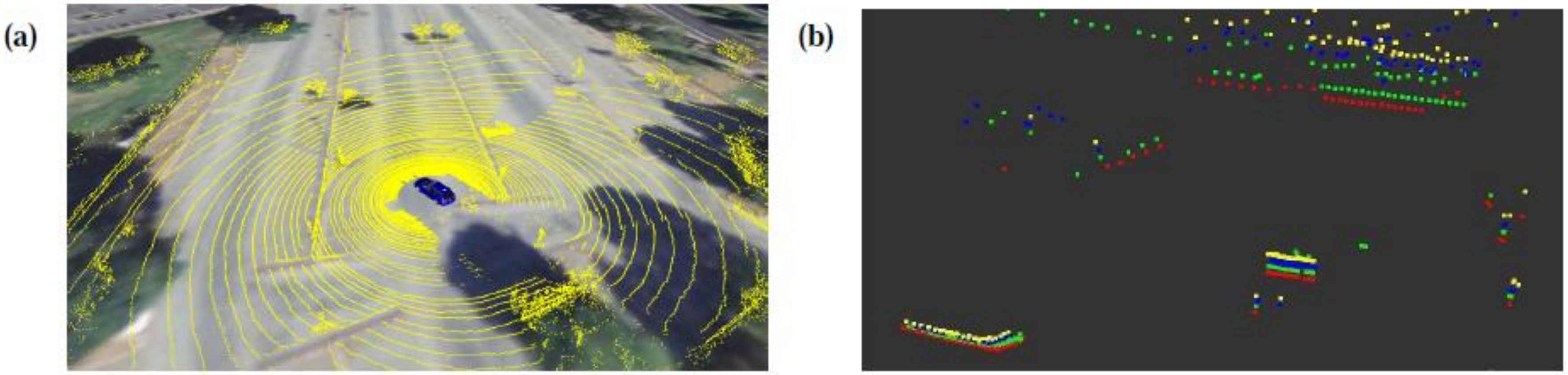


Figure 4: (a) The Velodyne contains 64 laser sensors and rotates at 10 Hz. It is able to see objects and terrain out to 60 meters in every direction. (b) The IBEO sensor possesses four scan lines which are primarily parallel to the ground. The IBEO is capable of detecting large vertical obstacles, such as cars and signposts.

Laser Obstacle Detection

- curbs near the vehicle detected using the 2 front facing IBEO laser range finders and 2 rear facing SICK laser range finders
- only obstacles close to the vehicle (5m in front and 15m in back) considered

Static Mapping

- discrete gri

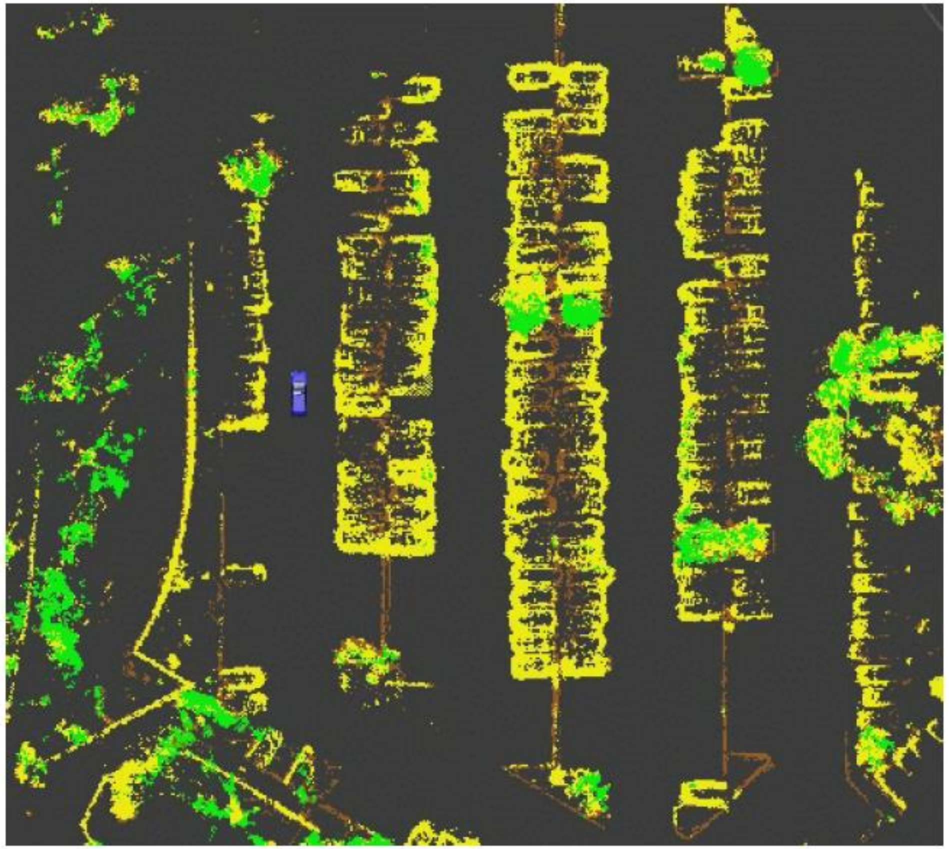


Figure 6: A map of a parking lot. Obstacles colored in yellow are tall obstacles, brown obstacles are curbs, and green obstacles are overhanging objects (e.g. tree branches) that are of no relevance to ground navigation.

Static Mapping

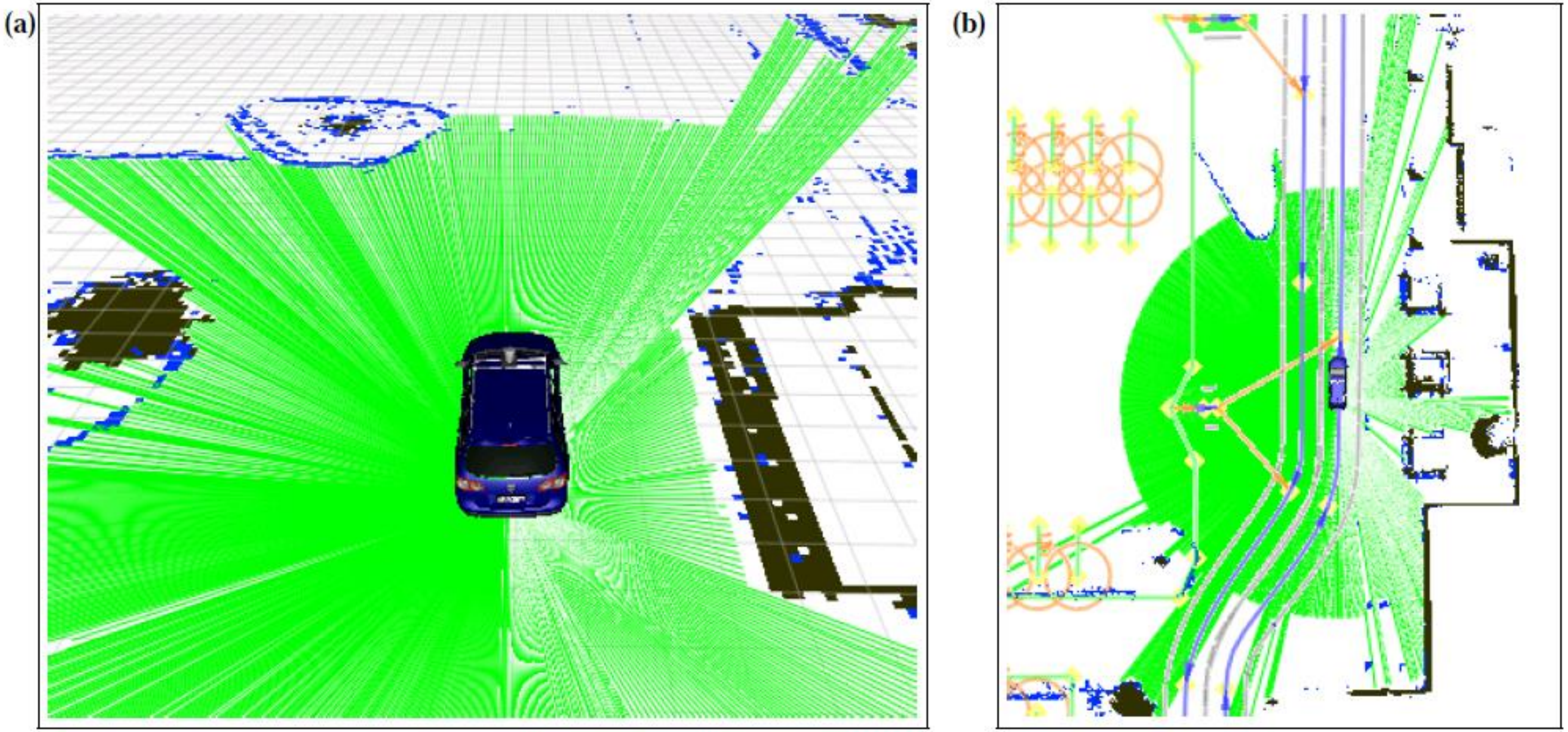
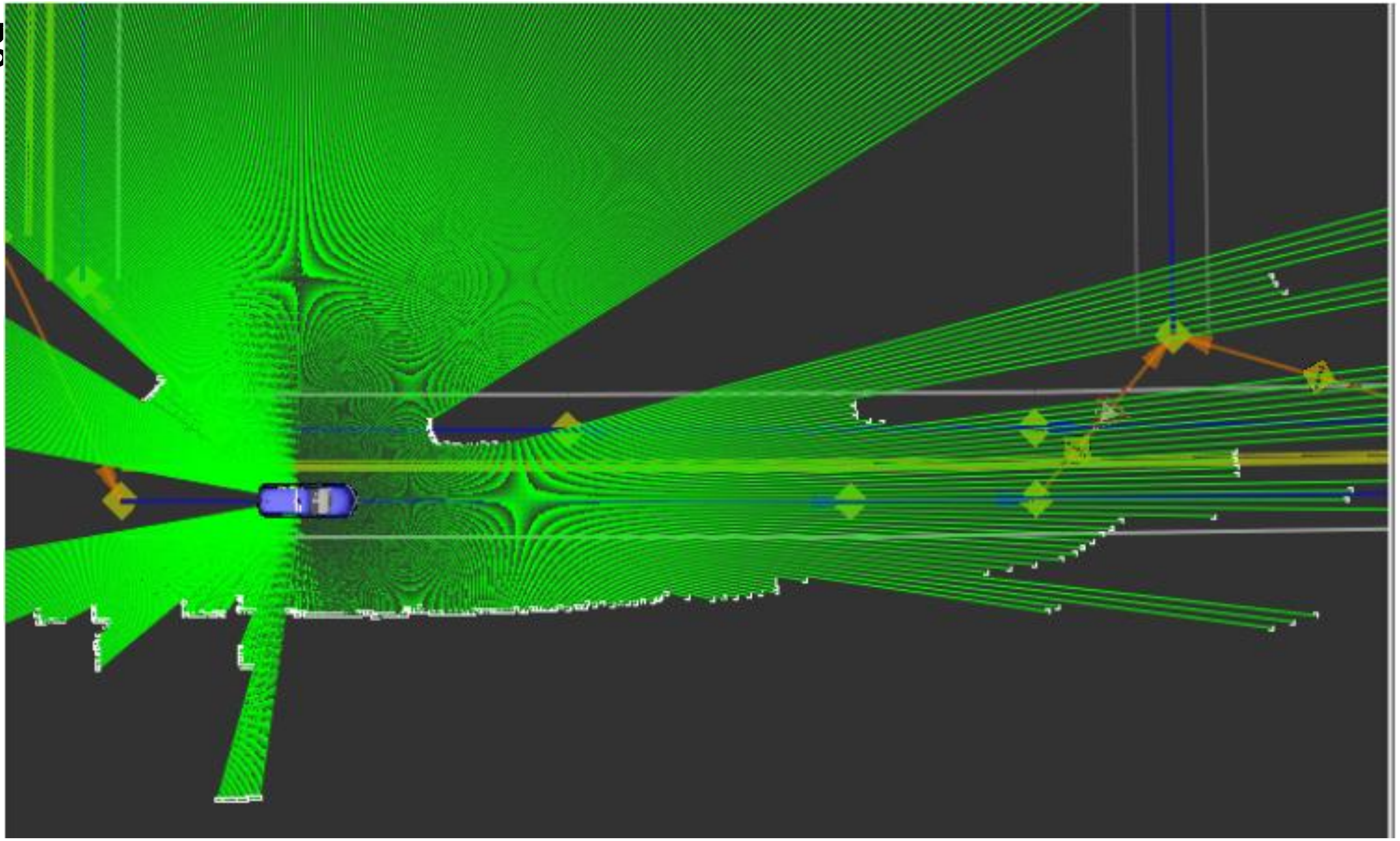


Figure 7: Examples of free space analysis for Velodyne scans. The green lines represent the area surrounding the robot that is observed to be empty. This evidence is incorporated into the static map, shown in black and blue.

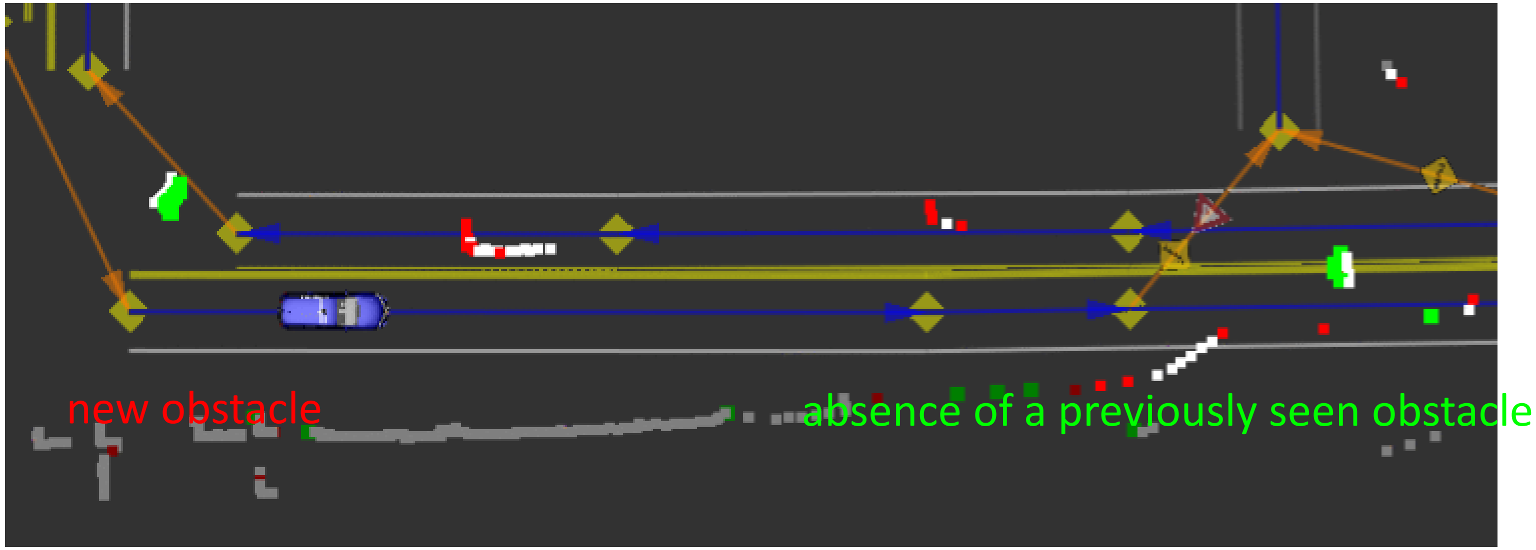
Dynamic Object Detection and Tracking

- laser rang the environm



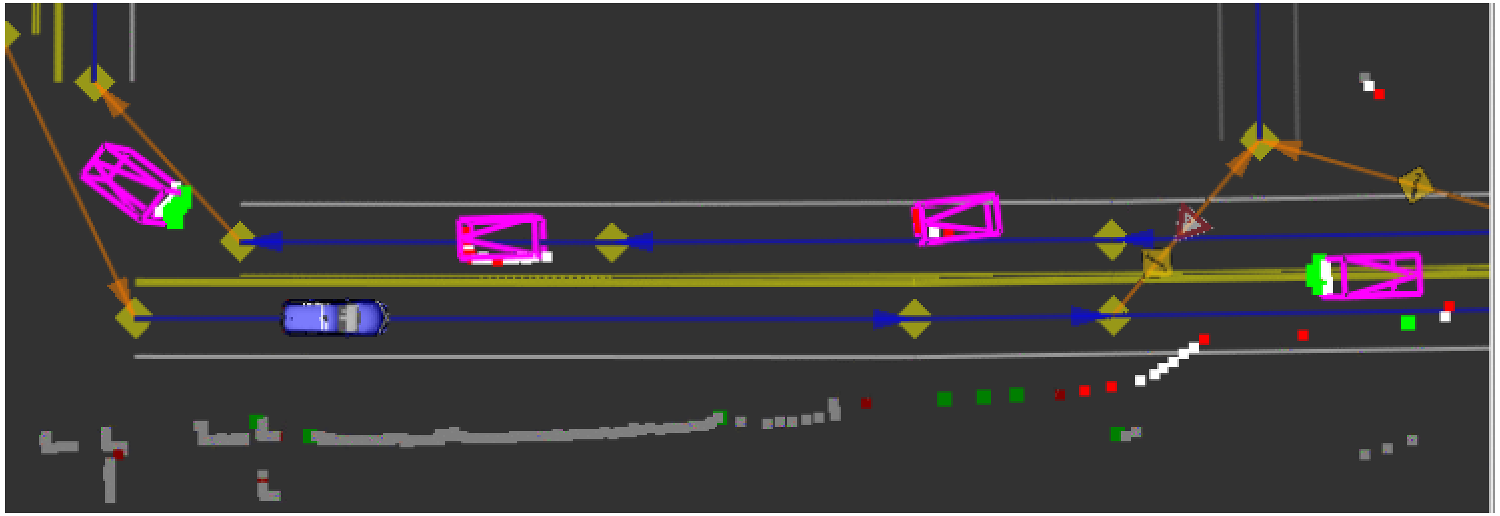
Dynamic Object Detection and Tracking

- areas of change are detected by comparing two synthetic scans taken over a short time interval
  - if an obstacle in one scan falls in the freespace of the second scan then this is evidence of motion



Dynamic Object Detection and Tracking

- particle filters are instantiated at each detected moving obstacle to track the object over time



Dynamic Object Detection and Tracking

- camera view

Precision Localization

- high precision GPS and inertial measurements are not sufficiently precise to achieve single lane localization in the DARPA supplied map
- DARPA map is also inaccurate!
- high precision localization within the lane is achieved using curb measurements and road reflectivity
  - 2 side facing SICK LMS and the forward facing RIEGL LMS-Q120 laser range finders pointed downward to measure infrared reflectivity change between road surface and painted lane markers
- offsets greater than 1 m relative to GPS were common observed

Pre

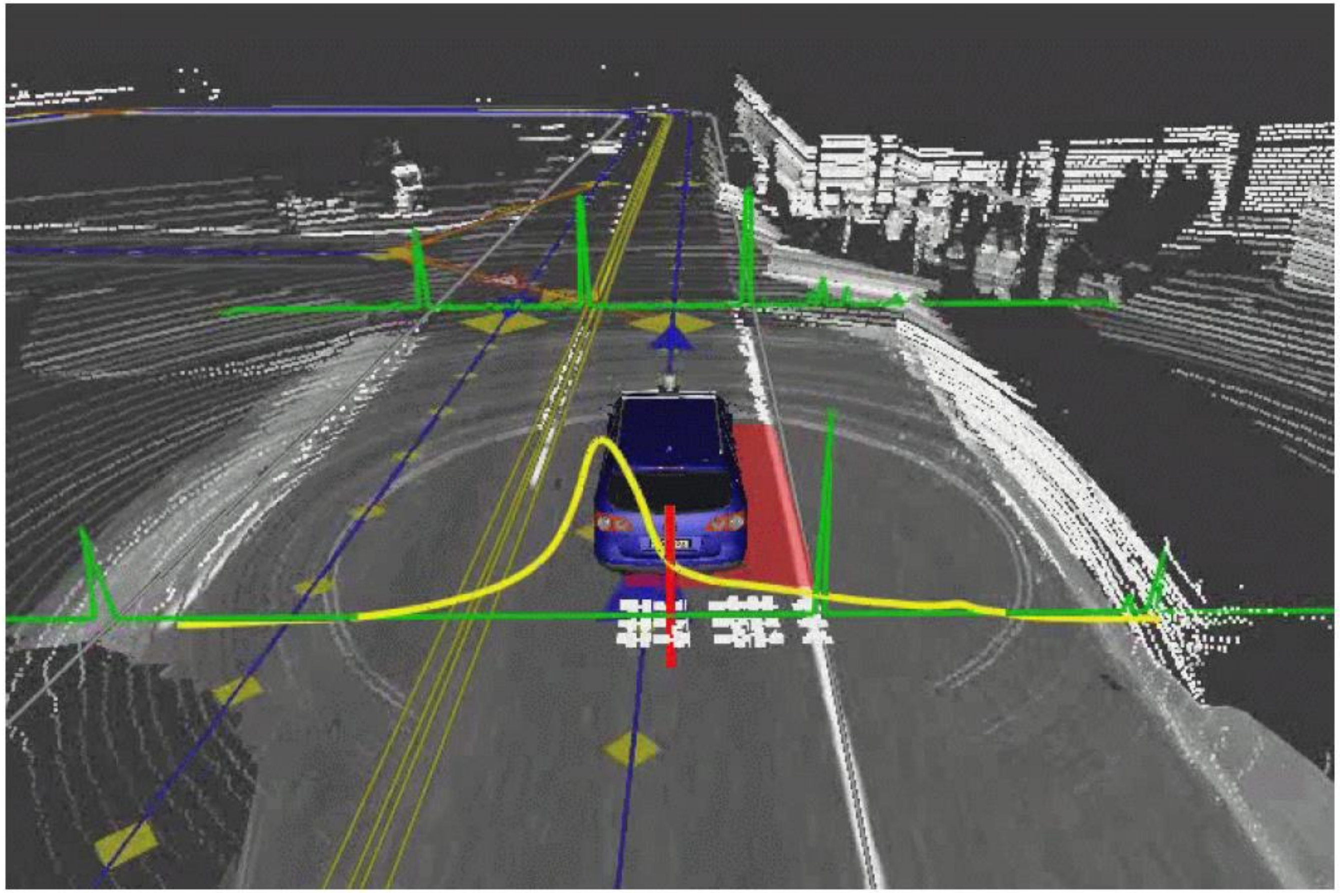


Figure 10: Typical localization result: The red bar illustrates the Applanix localization, whereas the yellow curve measures the posterior over the lateral position of the vehicle. The green line depicts the response from the lane line detector. In this case, the error is approximately 80 cm.

Global Path Planning

- performed d blockage is detected

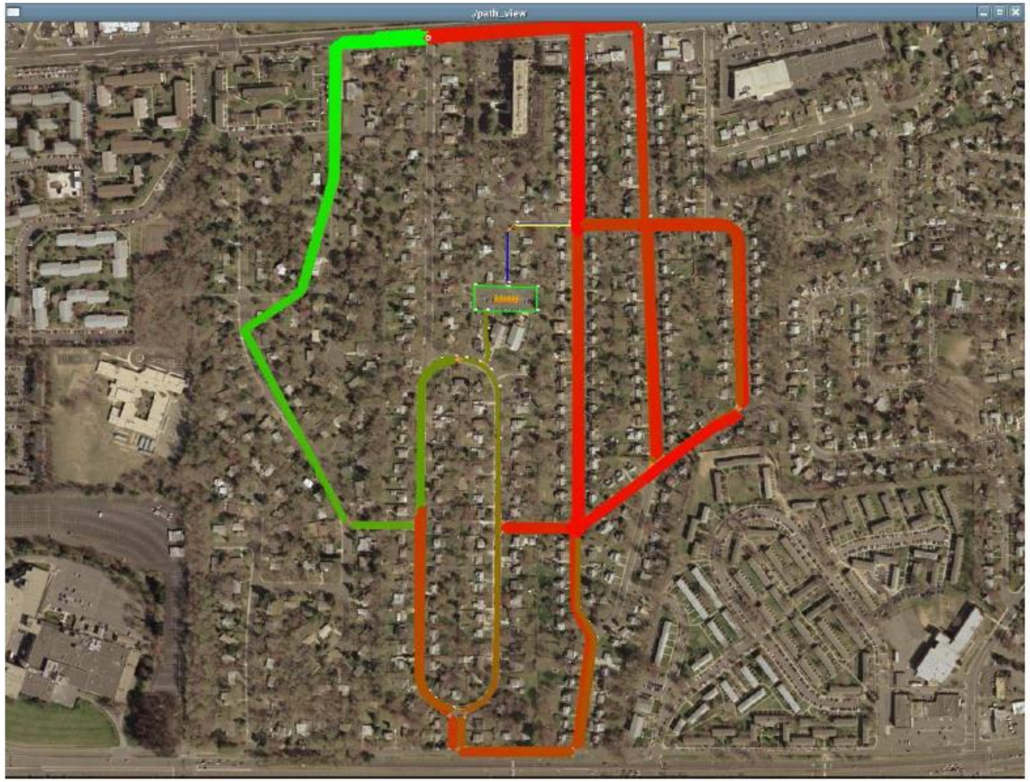


Figure 12: Global planning: Dynamic programming propagates values through a crude discrete version of the environment map. The color of the RNDF is representative of the cost to move to the goal from each position in the graph. Low costs are green and high costs are red.

Global Path Planning

- performed using dynamic programming on a discrete version of the map
- cost function weights choices (which way to turn at an intersection, lane changes, etc.) probabilistically
  - tends to perform lane changes earlier rather than later
  - will avoid passing a slow moving car in the right lane if a right hand turn is up coming
  - may decide against left hand turns at intersections if an alternate route is not much longer

Road Navigation

- global planner outputs a central path and also paths that have slight lateral shift
  - allows the

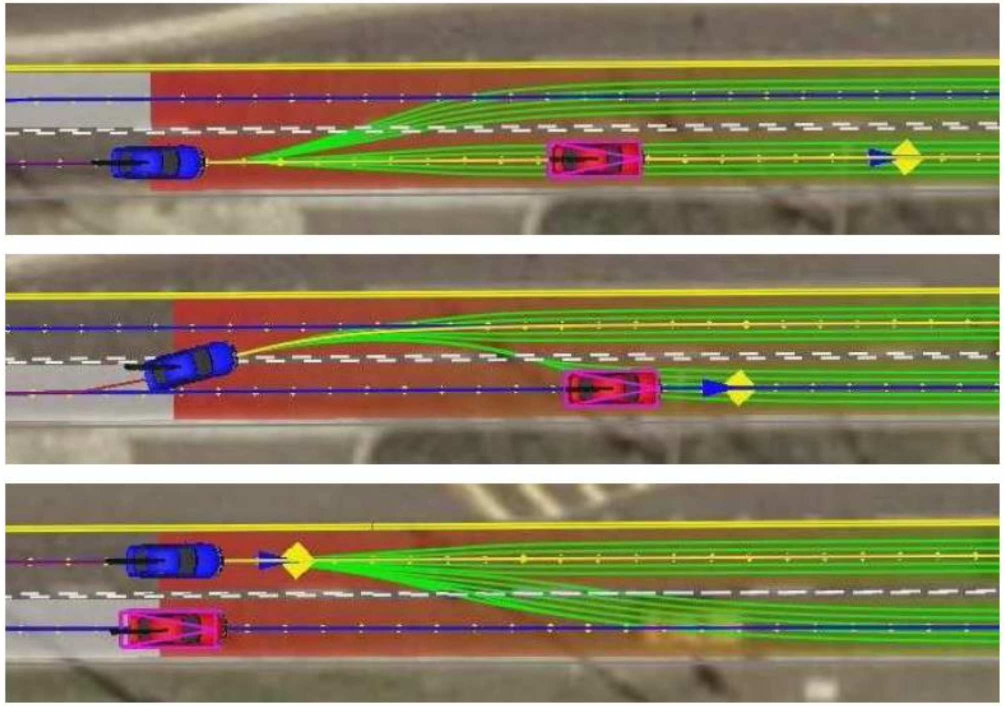
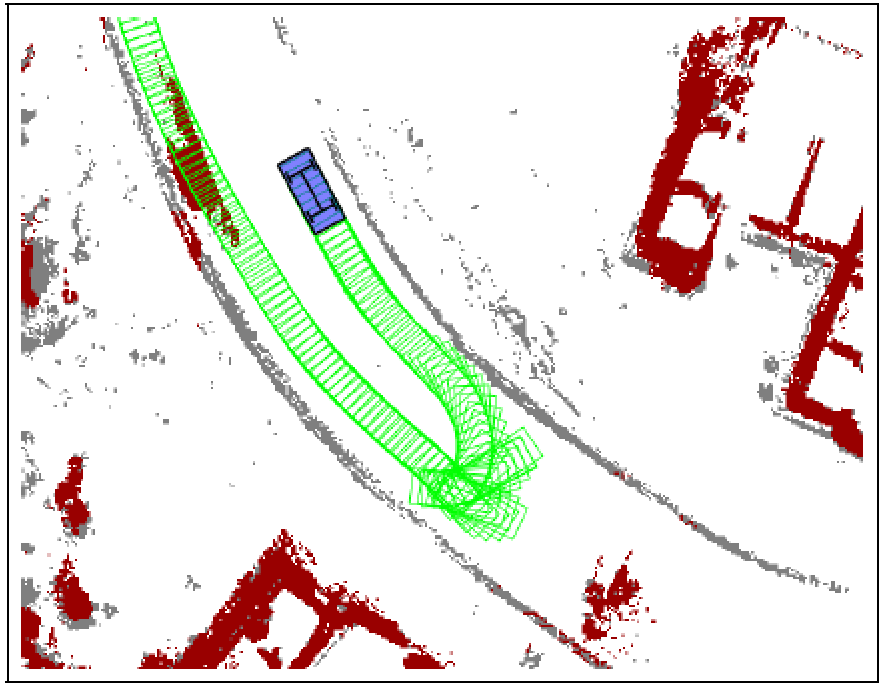
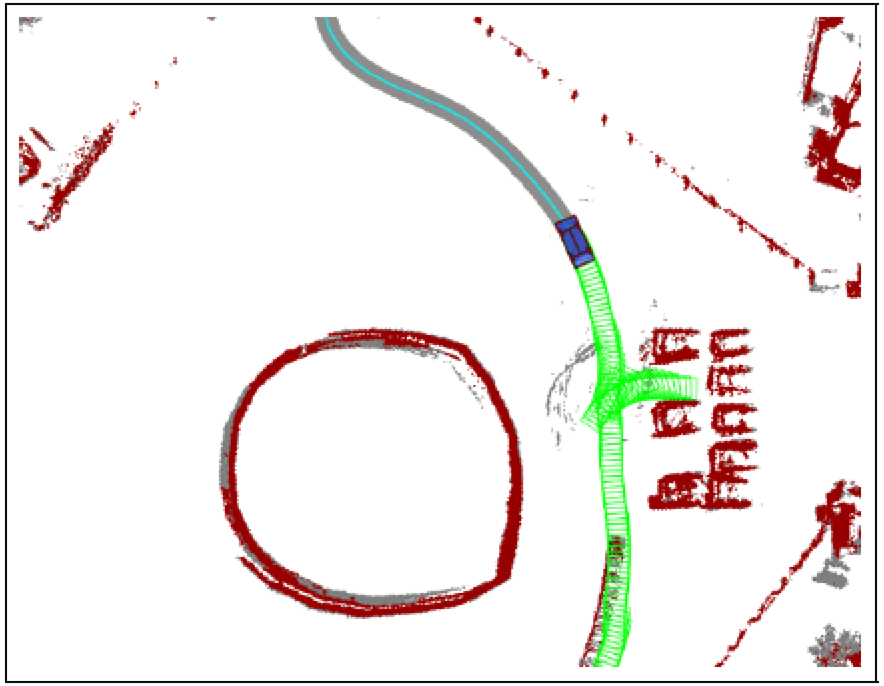


Figure 14: A passing maneuver. The additional cost of being in a slightly sub-optimal lane is overwhelmed by the cost of driving behind a slow driver, causing Junior to change lanes and pass.

Freeform Navigation

- used in parking lots and maneuvers such as U-turns
- planning performed using a variation of A* called hybrid-A*





Behavior Hierarchy

- behavior governed by a finite state machine

not shown
*TRAFFIC_JAM
*ESCAPE

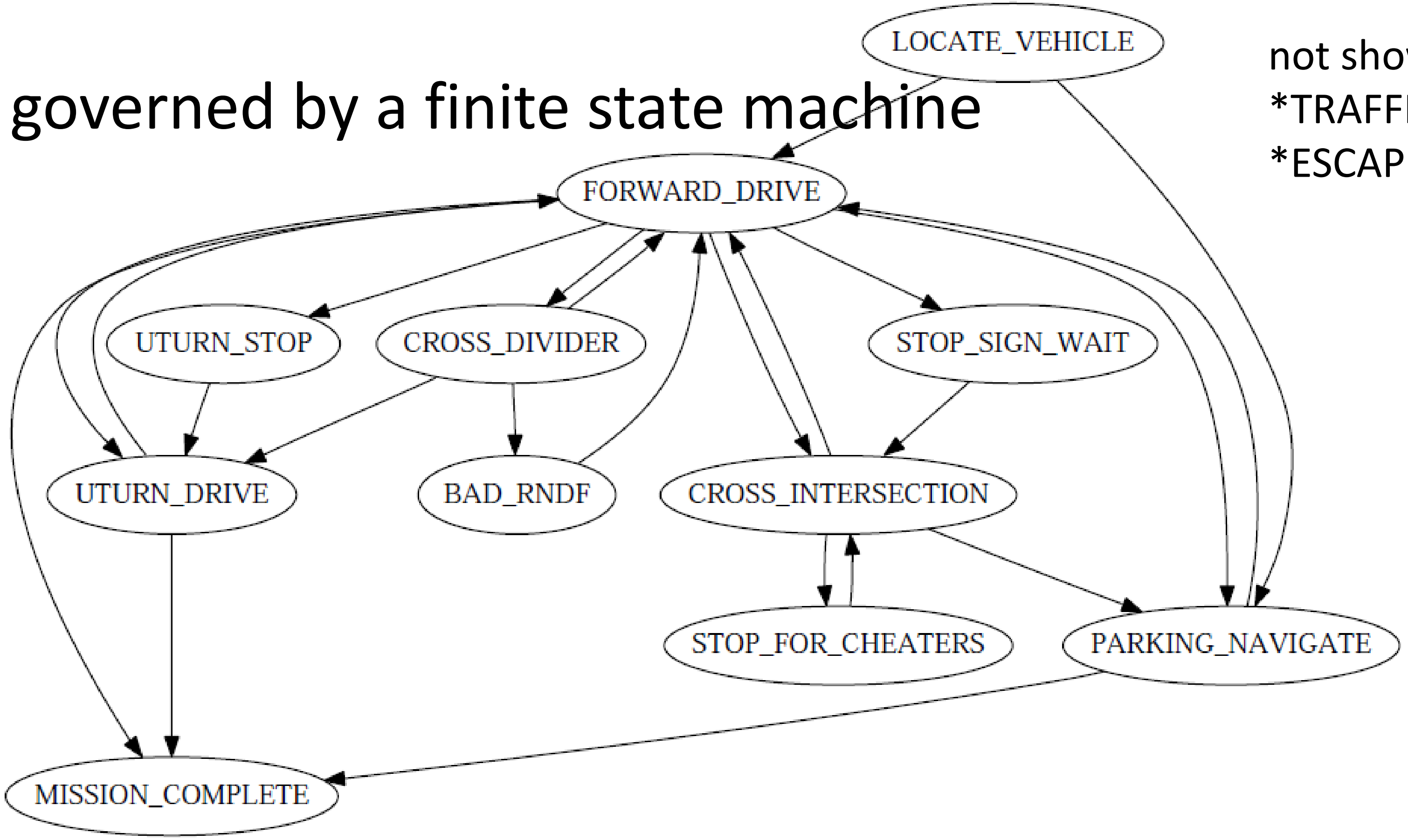


Figure 21: Finite State Machine that governs the robot's behavior.

Escaping Blockages

- TRAFFIC JAM and ESCAPE use hybrid-A* planner to find a path arc

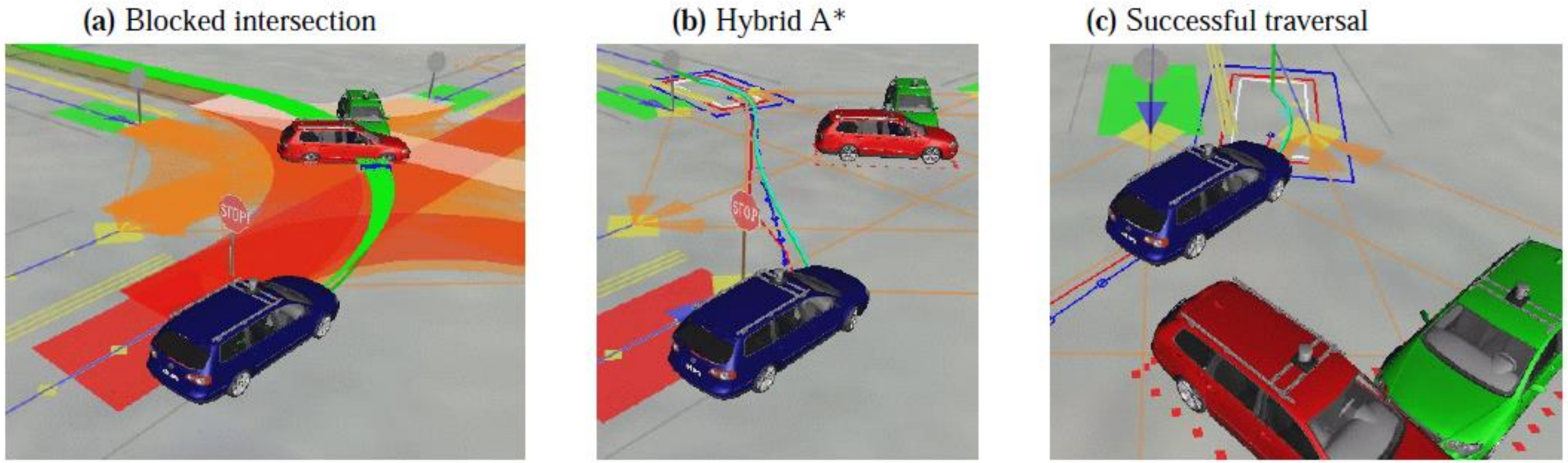


Figure 22: Navigating a simulated traffic jam: After a timeout period, the robot resorts to hybrid A* to find a feasible path across the intersection.

Results

- all 3 missions accomplished
- 55.96 miles in 4 hrs 5 mins 6 secs (22 km per hour)
  - good for 2nd place
- notable race events
  - travel over a dirt road
  - passing a disabled competitor
  - avoiding a head on collision with a competitor driving partially in the wrong lane
  - braking in reaction to an aggressive merge
  - guilty of an aggressive merge
  - guilty of pulling alongside a competitor stopped at a stop sign (mistaken for a parked car)

Thank you!

Write a closing statement or call-to-action here.

moh_khairudin@uny.ac.id

WA : 085878754037