CMP3103M/CMP9050M
AUTONOMOUS MOBILE ROBOTS

INTRODUCTION

Marc Hanheide

UNIVERSITY OF LINCOLN

L-CAS

CMP3103M AUTONOMOUS MOBILE ROBOTS

- **Semester B: AMR**
  - Marc Hanheide, Paul Baxter, Ayse Kucukyilmaz
  - Demonstrators: Sergi Molina + XXX
- Assessment items:

1. Robotics Practical Assignment (30%)
2. Exam (70%)

UNIVERSITY OF LINCOLN

L-CAS

SEMESTER B: AMR

- Lectures
- Workshops
  - Group A, B, C, D
  - Robot programming in ROS, partially in simulation, partially on real robots
- Attendance of all scheduled sessions is mandatory!
- Please check your timetable & blackboard for any timing & updates

WHAT'S ON?

B	Week	Lecture	Lecturer
1	22/01/2019	Intro	Marc Hanheide
2	29/01/2019	Robot Programming ROS	Marc Hanheide
3	05/02/2019	Robot Sensing	Marc Hanheide
4	12/02/2019	Motion and Control	Marc Hanheide
5	19/02/2019	Robot Behaviour	Ayse Kucukyilmaz
6	26/02/2019	Navigation	Ayse Kucukyilmaz
7	05/03/2019	Navigation	Ayse Kucukyilmaz
8	12/03/2019	Robot mapping - SLAM	Ayse Kucukyilmaz
9	19/03/2019	Control Architecture	Paul Baxter
10	26/03/2019	HRI1	Paul Baxter
11	02/04/2019	HRI2	Paul Baxter
12	09/04/2019	Application	Paul Baxter
13	30/04/2019	Mock Exam	Marc Hanheide

COURSEWORK

UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document	
Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	**Indicative Weighting CMP3103M: 30%** **Indicative Weighting CMP9050M: 20%**

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

COURSEWORK

UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document	
Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	Indicative Weighting CMP3103M: 30% Indicative Weighting CMP9050M: 20%

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

Requirements

This assessment item is split into two main tasks: "Group Robot Tasks" and "Visual Object Search", both detailed below

Note: The "one-stop shop" for resources relating to this assessment component and the workshops in CMP3103M is https://github.com/LCAS/teaching/wiki/CMP3103M.

1. "Group Robot Tasks" (Criterion 1)

Your first task (relating to Criterion 1 "Group Robot Tasks" in the CRG, 30% of the assessment item one mark) consists of continuous engagement with a total of four workshop tasks you work on as a group of 3-4 students, demonstrated successfully on a real Turtlebot robot and in simulation.
The tasks will be made available to you at the beginning of each workshop session at the latest, and are to be demonstrated to a member of the delivery team by the week after the latest (this gives each team 2 workshop sessions to program, test and demonstrate each task). Demonstrating later than the defined deadline results in a reduction of the merit mark by 50%. for this individual task. This is *group work*, so it is fine to work together on the task, but only students who are present at the demonstration of the group work, and can who answer questions about it, will be awarded the marks. Absent group members will have to demonstrate the work individually, under the same rules as outlined above, ie, will only obtain half the marks if demonstrated later than the week after the workshop task has been officially released. The individual tasks are available from https://github.com/LCAS/teaching/wiki/CMP3103M.

COURSEWORK

UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document	
Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	**Indicative Weighting CMP3103M: 30%** **Indicative Weighting CMP9050M: 20%**

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

Requirements

This assessment item is split into two main tasks: "Group Robot Tasks" and "Visual Object Search", both detailed below

Note: The "one-stop shop" for resources relating to this assessment component and the workshops in CMP3103M is https://github.com/LCAS/teaching/wiki/CMP3103M.

1. "Group Robot Tasks" (Criterion 1)

Your first task (relating to Criterion 1 "Group Robot Tasks" in the CRG, 30% of the assessment item one mark) consists of continuous engagement with a total of four workshop tasks you work on as a group of 3-4 students, demonstrated successfully on a real Turtlebot robot and in simulation.
The tasks will be made available to you at the beginning of each workshop session at the latest, and are to be demonstrated to a member of the delivery team by the week after the latest (this gives each team 2 workshop sessions to program, test and demonstrate each task). Demonstrating later than the defined deadline results in a reduction of the merit mark by 50%. for this individual task. This is *group work*, so it is fine to work together on the task, but only students who are present at the demonstration of the group work, and can who answer questions about it, will be awarded the marks. Absent group members will have to demonstrate the work individually, under the same rules as outlined above, ie, will only obtain half the marks if demonstrated later than the week after the workshop task has been officially released. The individual tasks are available from https://github.com/LCAS/teaching/wiki/CMP3103M.

2. "Visual Object Search" (Criterion 2 & 3)

Your second task (relating to Criterion 2 and 3 in the CRG, total of 70% of the mark for assessment item one) is to develop an object search behaviour, programmed in Python using ROS, that enables a robot to search for coloured objects visible in the robot's camera. This assessment is purely done in simulation, and *not* on the real robot. As part of this task, you must submit an *implementation* (criterion 2) and a *presentation* (criterion 3).

Implementation (Criterion 2)

Your task is to implement a behaviour that enables the robot in simulation to find a total of 4 objects distributed in a simulated environment. You need to utilise the robot's sensory input and its actuators to guide the robot to each item. Success in locating an item is defined as: (a) being less than 1m from the item, and (b) indication from the robot that it has found an object.

For the development and demonstration of your software component, you will be provided with a simulation environment (called "Gazebo"). The required software is installed on all machines in the Labs. The simulated environment includes four brightly coloured objects hidden in the environment at increasing difficulty. Your robot starts from a predefined position. You will be provided with a "training arena" in simulation (a simulation of an indoor environment in which 4 objects will be "hidden"). This "training arena" will resemble the "test arena" in terms of structure and complexity (same floor plan of the environment), but the positions of the objects will slightly vary to assess the generality of your approach.
You may choose any sensors available on the robot to drive your search behaviour. However, your system design should include the following elements:

1. Perception of the robot's environment using the Kinect sensor, either in RGB or Depth space, or using a combination of both RGB and Depth data in order find the object;
2. An implementation of an appropriate control law implementing a search behaviour on the robot. You may choose to realise this as a simple reactive behaviour or a more complex one, eg, utilising a previously acquired map of the environment;
3. Motor control of the (simulated) Turtlebot robot using the implemented control law.

COURSEWORK

UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document

Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	**Indicative Weighting CMP3103M: 30%** **Indicative Weighting CMP9050M: 20%**

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

Requirements

This assessment item is split into two main tasks: “Group Robot Tasks” and “Visual Object Search”, both detailed below

Note: The “one-stop shop” for resources relating to this assessment component and the workshops in CMP3103M is https://github.com/LCAS/teaching/wiki/CMP3103M.

1. “Group Robot Tasks” (Criterion 1)

Your first task (relating to Criterion 1 “Group Robot Tasks” in the CRG, 30% of the assessment item one mark) consists of continuous engagement with a total of four workshop tasks you work on as a group of 3-4 students, demonstrated successfully on a real Turtlebot robot and in simulation.
The tasks will be made available to you at the beginning of each workshop session at the latest, and are to be demonstrated to a member of the delivery team by the week after the latest (this gives each team 2 workshop sessions to program, test and demonstrate each task). Demonstrating later than the defined deadline results in a reduction of the merit mark by 50%. for this individual task. This is *group work*, so it is fine to work together on the task, but only students who are present at the demonstration of the group work, and can who answer questions about it, will be awarded the marks. Absent group members will have to demonstrate the work individually, under the same rules as outlined above, ie, will only obtain half the marks if demonstrated later than the week after the workshop task has been officially released. The individual tasks are available from https://github.com/LCAS/teaching/wiki/CMP3103M.

2. “Visual Object Search” (Criterion 2 & 3)

Your second task (relating to Criterion 2 and 3 in the CRG, total of 70% of the mark for assessment item one) is to develop an object search behaviour, programmed in Python using ROS, that enables a robot to search for coloured objects visible in the robot’s camera. This assessment is purely done in simulation, and *not* on the real robot. As part of this task, you must submit an *implementation* (criterion 2) and a *presentation* (criterion 3).

Implementation (Criterion 2)

Your task is to implement a behaviour that enables the robot in simulation to find a total of 4 objects distributed in a simulated environment. You need to utilise the robot’s sensory input and its actuators to guide the robot to each item. Success in locating an item is defined as: (a) being less than 1m from the item, and (b) indication from the robot that it has found an object.

For the development and demonstration of your software component, you will be provided with a simulation environment (called “Gazebo”). The required software is installed on all machines in the Labs. The simulated environment includes four brightly coloured objects hidden in the environment at increasing difficulty. Your robot starts from a predefined position. You will be provided with a “training arena” in simulation (a simulation of an indoor environment in which 4 objects will be “hidden”). This “training arena” will resemble the “test arena” in terms of structure and complexity (same floor plan of the environment), but the positions of the objects will slightly vary to assess the generality of your approach.
You may choose any sensors available on the robot to drive your search behaviour. However, your system design should include the following elements:

1. Perception of the robot’s environment using the Kinect sensor, either in RGB or Depth space, or using a combination of both RGB and Depth data in order find the object;
2. An implementation of an appropriate control law implementing a search behaviour on the robot. You may choose to realise this as a simple reactive behaviour or a more complex one, eg, utilising a previously acquired map of the environment;
3. Motor control of the (simulated) Turtlebot robot using the implemented control law.

The minimum required functionality consists of a simple reactive behaviour, allowing in principle to find objects. For an average mark the behaviour should be able to successfully find some objects at unknown locations. Further extensions are possible to improve your mark in this assessment, also to enable you to find all objects. Possible extensions to the system may include (but are not limited to):

- An enhanced perception system – in-built colour appearance learning, use of additional visual cues (e.g. edges), combination of RGB and Depth features, etc.;
- Exploiting maps and other structural features in the environment or more clever search strategies.
- Utilising other existing ROS components that are available (like localisation, mapping, etc)

The software component must be implemented in Python and be supported by use of ROS to communicate with the robot. The code should be well-commented and clearly structured into functional blocks. The program must run on computers in Labs B and C. To obtain credit for this assignment you will need to demonstrate the various components of your software to the module instructors and be ready to answer questions related to the development of the solution – please follow carefully the instructions given in the lectures on the requirements for the demonstration and see below for presentation requirements.

Your implementation needs to be submitted via blackboard, with the source code containing good documentation (functionality and code quality account for 40% of assessment item one).

COURSEWORK

UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document	
Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	**Indicative Weighting CMP3103M: 30%** **Indicative Weighting CMP9050M: 20%**

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

Requirements

This assessment item is split into two main tasks: “Group Robot Tasks” and “Visual Object Search”, both detailed below

Note: The “one-stop shop” for resources relating to this assessment component and the workshops in CMP3103M is https://github.com/LCAS/teaching/wiki/CMP3103M.

1. “Group Robot Tasks” (Criterion 1)

Your first task (relating to Criterion 1 “Group Robot Tasks” in the CRG, 30% of the assessment item one mark) consists of continuous engagement with a total of four workshop tasks you work on as a group of 3-4 students, demonstrated successfully on a real Turtlebot robot and in simulation.
The tasks will be made available to you at the beginning of each workshop session at the latest, and are to be demonstrated to a member of the delivery team by the week after the latest (this gives each team 2 workshop sessions to program, test and demonstrate each task). Demonstrating later than the defined deadline results in a reduction of the merit mark by 50%. for this individual task. This is *group work*, so it is fine to work together on the task, but only students who are present at the demonstration of the group work, and can who answer questions about it, will be awarded the marks. Absent group members will have to demonstrate the work individually, under the same rules as outlined above, ie, will only obtain half the marks if demonstrated later than the week after the workshop task has been officially released. The individual tasks are available from https://github.com/LCAS/teaching/wiki/CMP3103M.

2. “Visual Object Search” (Criterion 2 & 3)

Your second task (relating to Criterion 2 and 3 in the CRG, total of 70% of the mark for assessment item one) is to develop an object search behaviour, programmed in Python using ROS, that enables a robot to search for coloured objects visible in the robot’s camera. This assessment is purely done in simulation, and *not* on the real robot. As part of this task, you must submit an *implementation* (criterion 2) and a *presentation* (criterion 3).

Implementation (Criterion 2)

Your task is to implement a behaviour that enables the robot in simulation to find a total of 4 objects distributed in a simulated environment. You need to utilise the robot’s sensory input and its actuators to guide the robot to each item. Success in locating an item is defined as: (a) being less than 1m from the item, and (b) indication from the robot that it has found an object.

For the development and demonstration of your software component, you will be provided with a simulation environment (called “Gazebo”). The required software is installed on all machines in the Labs. The simulated environment includes four brightly coloured objects hidden in the environment at increasing difficulty. Your robot starts from a predefined position. You will be provided with a “training arena” in simulation (a simulation of an indoor environment in which 4 objects will be “hidden”). This “training arena” will resemble the “test arena” in terms of structure and complexity (same floor plan of the environment), but the positions of the objects will slightly vary to assess the generality of your approach.
You may choose any sensors available on the robot to drive your search behaviour. However, your system design should include the following elements:

1. Perception of the robot’s environment using the Kinect sensor, either in RGB or Depth space, or using a combination of both RGB and Depth data in order find the object;
2. An implementation of an appropriate control law implementing a search behaviour on the robot. You may choose to realise this as a simple reactive behaviour or a more complex one, eg, utilising a previously acquired map of the environment;
3. Motor control of the (simulated) Turtlebot robot using the implemented control law.

The minimum required functionality consists of a simple reactive behaviour, allowing in principle to find objects. For an average mark the behaviour should be able to successfully find some objects at unknown locations. Further extensions are possible to improve your mark in this assessment, also to enable you to find all objects. Possible extensions to the system may include (but are not limited to):

- An enhanced perception system – in-built colour appearance learning, use of additional visual cues (e.g. edges), combination of RGB and Depth features, etc.;
- Exploiting maps and other structural features in the environment or more clever search strategies.
- Utilising other existing ROS components that are available (like localisation, mapping, etc)

The software component must be implemented in Python and be supported by use of ROS to communicate with the robot. The code should be well-commented and clearly structured into functional blocks. The program must run on computers in Labs B and C. To obtain credit for this assignment you will need to demonstrate the various components of your software to the module instructors and be ready to answer questions related to the development of the solution – please follow carefully the instructions given in the lectures on the requirements for the demonstration and see below for presentation requirements.

Your implementation needs to be submitted via blackboard, with the source code containing good documentation (functionality and code quality account for 40% of assessment item one).

Presentation (Criterion 3)

You are also required to submit and present a Powerpoint presentation with a maximum of *3 slides*, *including the title slide*. One slide should cover your system design, including details of the distinctive features of your perception and control algorithms. A diagram showing the system architecture is recommended for this part. The other slide should cover your results, including your observations and reflections on the system performance.

You will be given up to 5 minutes to present your slides to the module coordinators at a special workshop session (scheduled in weeks B10 and B11) and show your implementation working the labs. You must furthermore be prepared to answer any further questions about your work. Please follow carefully the instructions given in the lectures and workshops on the requirements for the presentation. For assessment, you will be assigned to a “marking group” to briefly present your working implementation and explain it to members of the delivery team to have it marked. You will be marked on both the functionality and the quality of your presentation as detailed in the CRG.

COURSEWORK

Learning Outcome	Criterion	Pass	2:2	2:1	1st
[LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments	Criterion 1: Group Robot Tasks (30%)	You and your group have demonstrated basic functionality in simulation only.	You and your group have demonstrated the full functionality in simulation only, or the basic functionality in simulation and on the real robot.	You and your group have demonstrated the full functionality in simulation and basic functionality on the real robot.	You and your group have demonstrated the full functionality in both, simulation and on the real robot.
	Criterion 2: Individual Visual Search Task(in Simulation only, 40%), Implementation	A working software component with basic functionality. Fair program structure and some code comments. The working implementation is demonstrated, accomplishing the search task partially.	A working software component with good functionality. Clear program structure and appropriate comments. The implementation is demonstrated successfully, accomplishing most of the search task with a good performance.	A good implementation with some extra functionality or originality. The program code is well structured and commented. Good demonstration of basic and additional features, accomplishing the search task with a very good performance.	An excellent implementation featuring original functionality and elements beyond the original specification. The program code is efficient, well structured and commented. The solution is demonstrated very well, highlighting the additional functionalities, accomplishing the full search task with an excellent performance.
	Criterion 3: Individual Visual Search Task (in Sim, 30%), Presentation	A basic presentation of the system design and its performance.	A good presentation of the system design and reflections on its performance.	A very good presentation of the system design and reflections on its performance, including evidence of testing and evaluation of the important system features.	An excellent presentation of the system design and reflections on its performance, including evidence of thorough testing and evaluation of the important system features.
Weighting	The criteria for this assessment are weighted as indicated.				

WHAT'S ON?

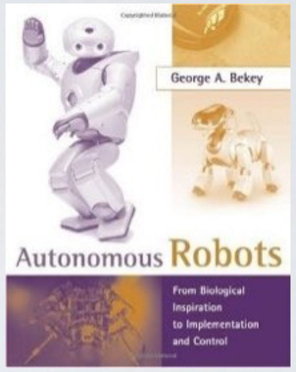
B week	Date	Workshop
1	**25/01/2019**	**Minitask 1: setup, running simulation**
2	**01/02/2019**	**Minitask 2: ROS intro, topics, services, example to make robot move cmd_vel**
3	**08/02/2019**	**Minitask 3: Rviz, point clouds, tf, more on event-driven programming, open_cv, etc.**
4	**15/02/2019**	**Minitask 4: cmd_vel, Braitenberg**
5	**22/02/2019**	**assignment support**
6	**01/03/2019**	**assignment support**
7	**08/03/2019**	**assignment support**
8	**15/03/2019**	**assignment support**
9	**22/03/2019**	**assignment support**
10	**29/03/2019**	**eval**
11	**05/04/2019**	**eval**
12	**12/04/2019**	**Q&A**
13		

UNIVERSITY OF LINCOLN

L-CAS

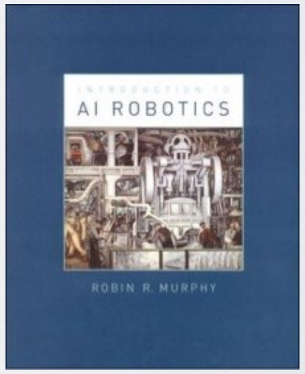
READING MATERIAL

- Bekey, G.A.: ***Autonomous robots***. MIT Press, 2005



- Siegwart, R. and Nourbakhsh, I.R.: ***Introduction to autonomous mobile robots***. MIT Press, 2004

- Murphy, R.R.: ***An Introduction to AI Robotics.*** MIT Press, 2000



LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

TURTLEBOT 2 – OUR WORKSHOP BUDDY

This year's assignment: design and implement a robotic visual search (more soon...)

TurtleBot™

ROS Enabled

UNIVERSITY OF LINCOLN

L-CAS

ROBOT

- What is it?
  - “I can't define a robot, but I know one when I see one.”, J. Engelberger
- What is it for?
  - help/replace humans in monotonous, boring, repetitive, dangerous, difficult tasks
  - companion? helper? servant?

[HTTPS://POLLEV.COM/MHANHEIDE](https://pollev.com/mhanheide)

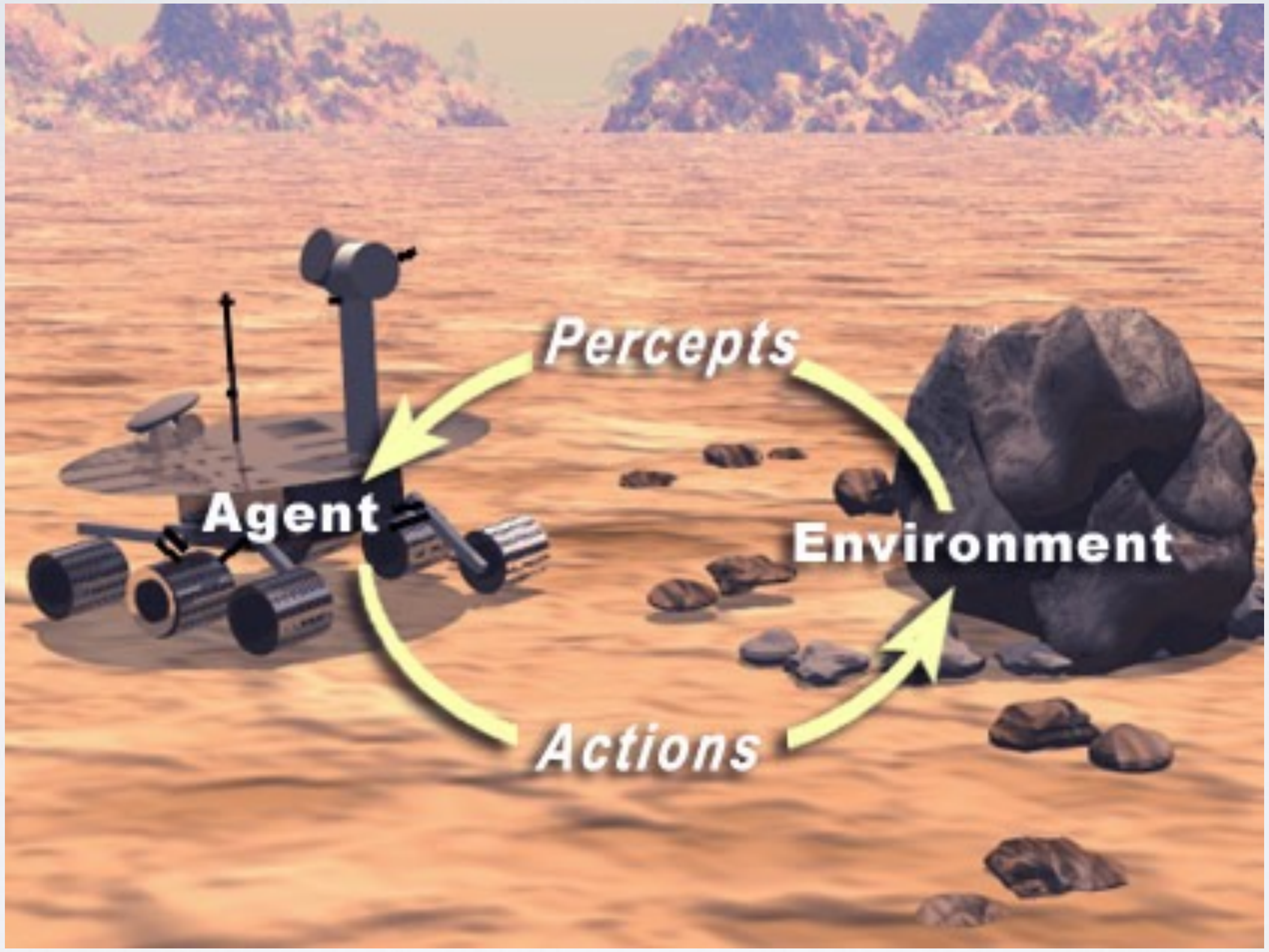
UNIVERSITY OF LINCOLN

L-CAS

What do you associate with the word robot?

fury
future :-p
perception
doom

ROBOTICS



- Robotics = ”the intelligent connection of perception and action” (Brady 1985)

Which of the following features are essential for a robot

- Hardware Controllers
- Actuators
- Locomotion (e.g. wheels or legs)
- Embodiment (physical interaction with the world)
- Internal Sensors
- External Sensors

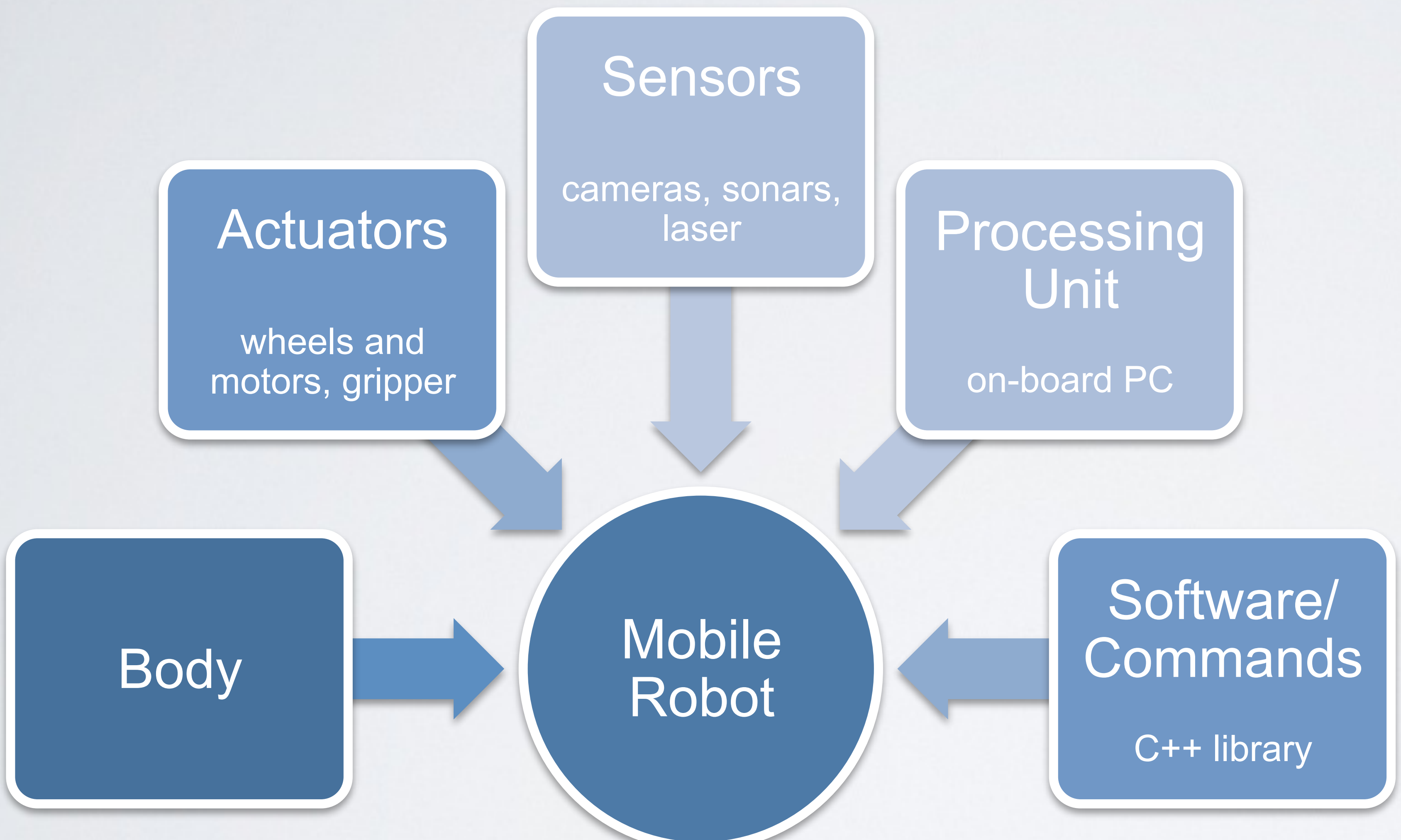
MOBILE ROBOTS

- Mobility
  - opens possibilities for new tasks: transportation, surveillance, cleaning etc.
  - unstructured environments
  - main challenge: navigation
- Autonomy
  - reasoning: making decisions, plan
  - learning from experience
  - building representation of the (dynamic) environment

UNIVERSITY OF LINCOLN

L-CAS

ANATOMY OF A MOBILE ROBOT

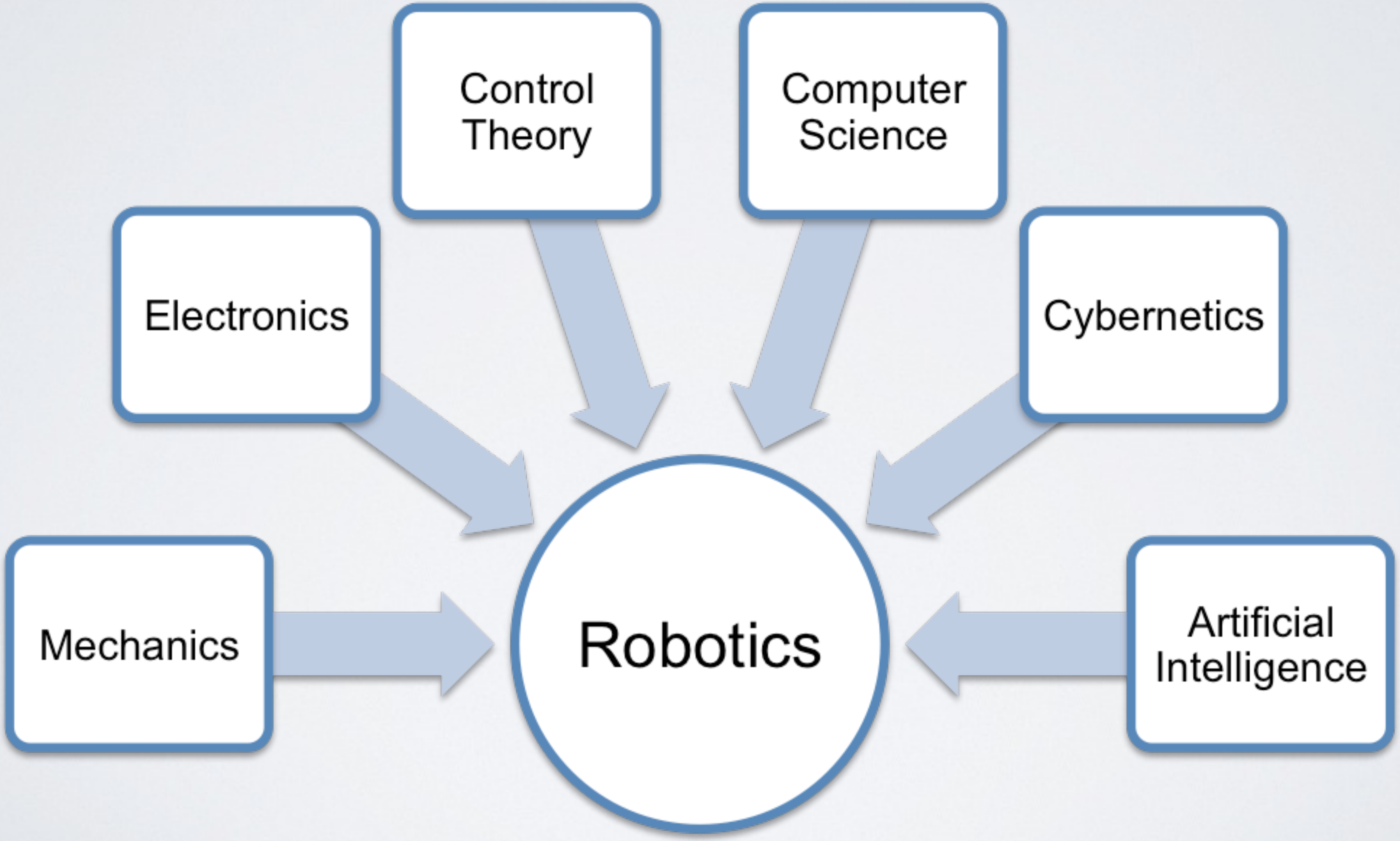


UNIVERSITY OF LINCOLN

L-CAS

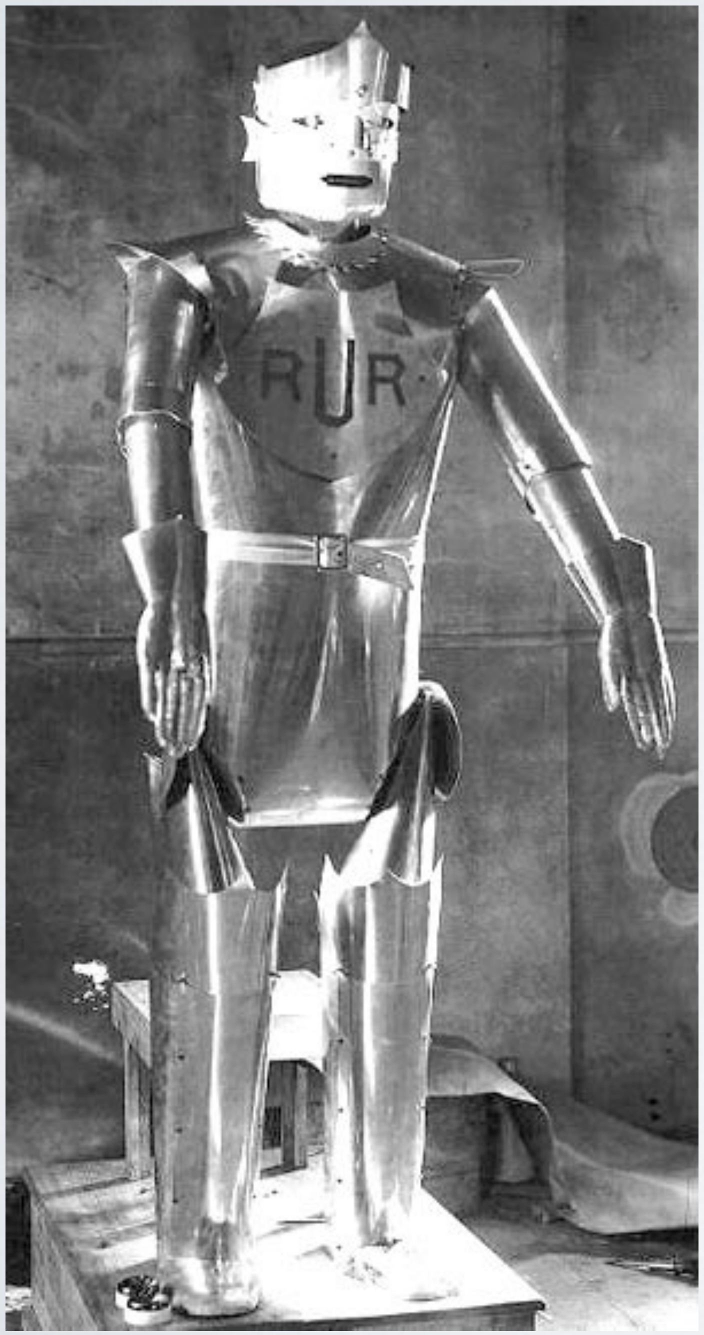
ROBOTICS

Science and technology of robots



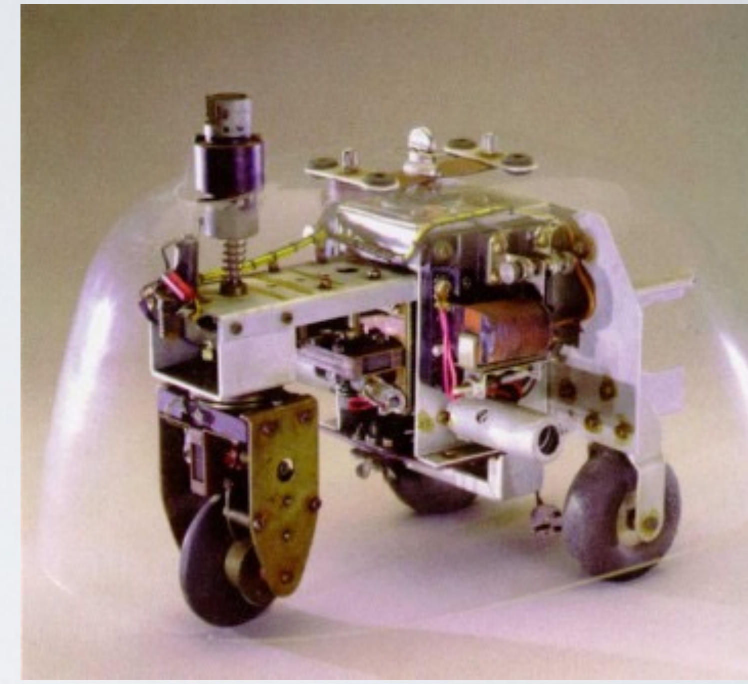
PAST AND PRESENT

R.U.R.

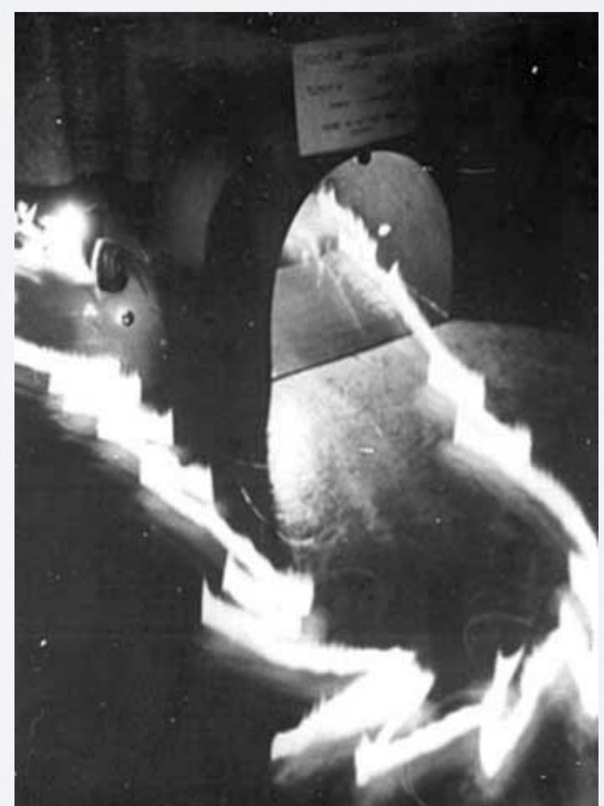


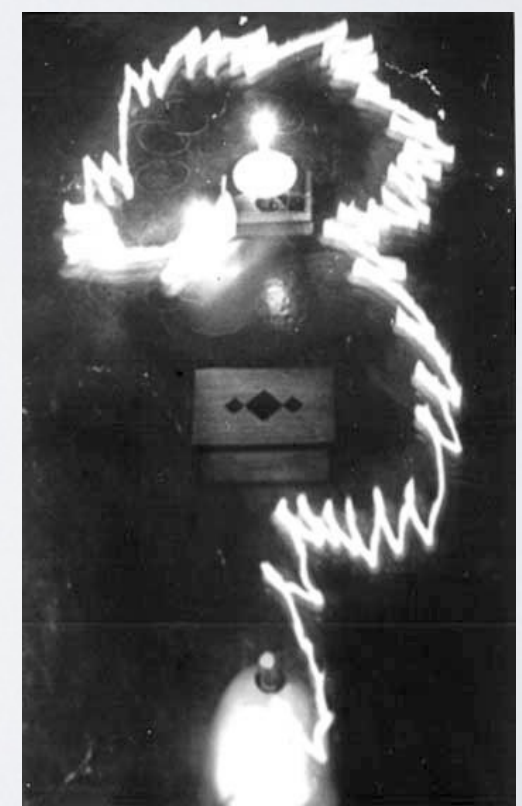
- Karel Čapek, 1921
  - R.U.R. - Rossum's Universal Robots, a stage play
  - the word 'robot' appears for the first time
  - 'robota' – forced/hard labour in Czech
  - robots – artificial men that can think but seem to be happy to serve
  - exploited (?) by humans
  - finally rebel against their creators

GREY WALTER'S TURTLES

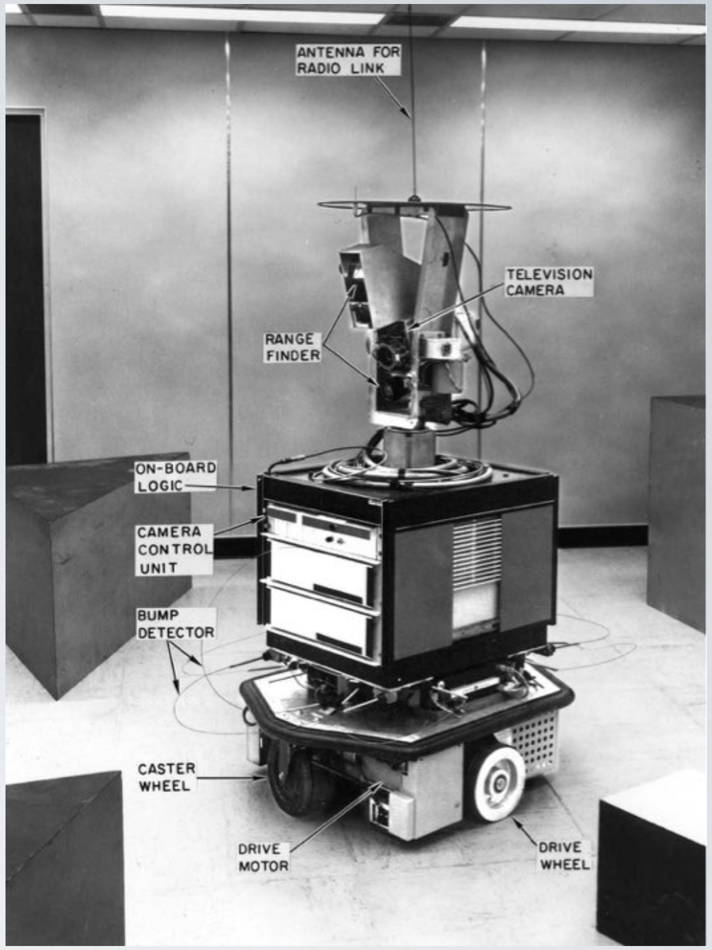


- "Machina Speculatrix", 1948
  - experiments in reflex behaviour
  - built of electronic valves and photo-cells
  - approach or escape a light source

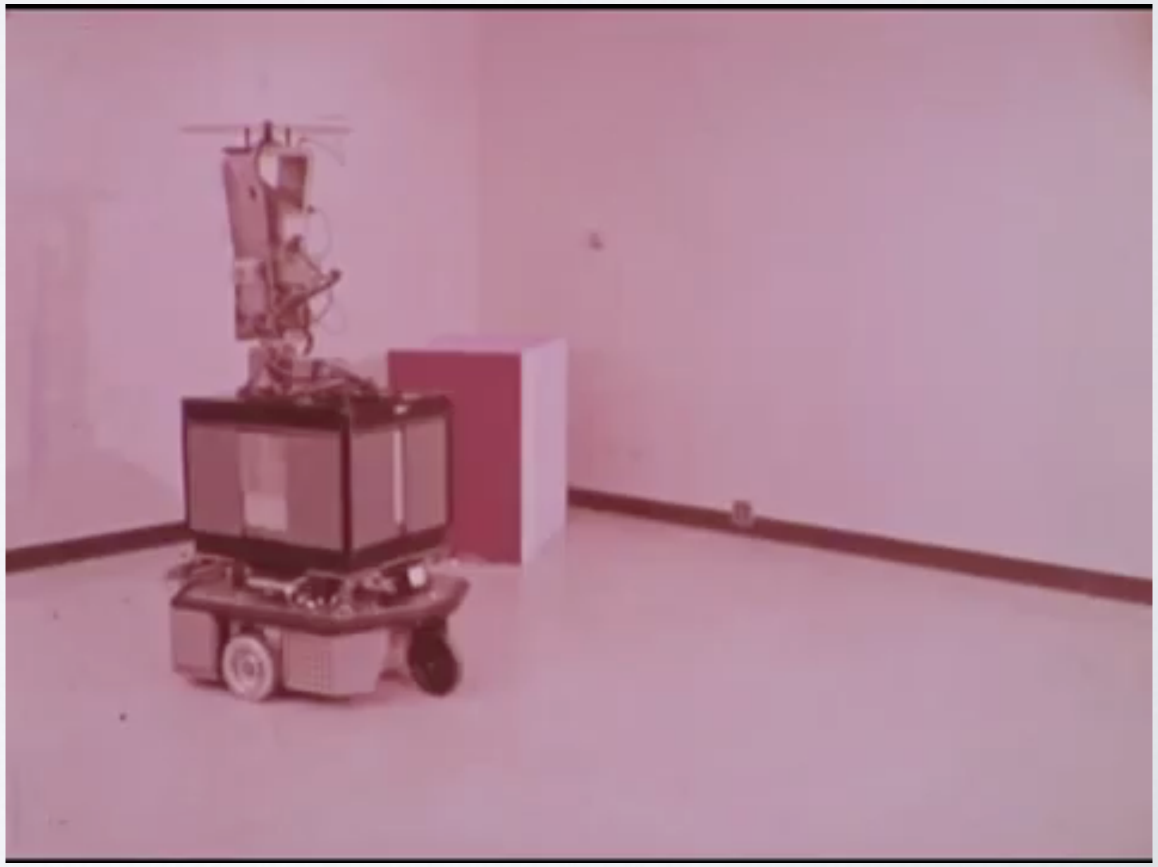




SHAKEY



- Stanford Artificial Intelligence Centre, 1966
  - first mobile robot that could be programmed for various tasks
  - on-board I/O logic
  - radio-link
  - external computer (PDP-10)



INDUSTRIAL ROBOTS

- Manipulators
  - “big arms”
  - precise, strong and fast
  - well studied
  - many manufactured
  - operate in controlled environments
  - limited sensory abilities
  - pre-programmed

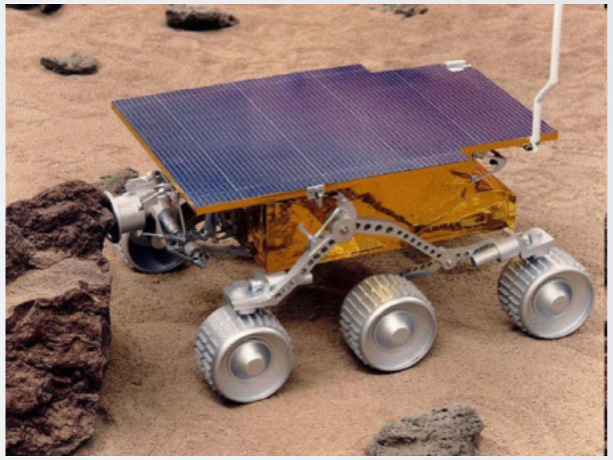
INDUSTRIAL ROBOTS

"Big arms"

AGVs

APPLICATIONS

- Exploration
- Surveillance
- Security
- Care assistants
- Agricultural robots
- Intelligent vehicles
- many others...



Which jobs are most likely to be replaced by Robots/AI? Put the most likely one to the top.

- Bin man
- Investment Banker
- Law Associate
- Dentist
- Programmer
- Brick layer
- Fruit picker
- Lecturer
- Accountant

Which jobs are at risk?

Researchers at Oxford University published a widely referenced study in 2013 on the likelihood of computerisation for different occupations.

Out of around 700 occupations, 12 were found to have a 99 per cent chance of being automated in the future:

- Data Entry Keyers
- Library Technicians
- New Accounts Clerks
- Photographic Process Workers and Processing Machine Operators
- Tax Preparers
- Cargo and Freight Agents
- Watch Repairers
- Insurance Underwriters
- Mathematical Technicians
- Sewers, Hand
- Title Examiners, Abstractors, and Searchers
- Telemarketers

- http://www.telegraph.co.uk/news/2017/09/27/jobs-risk-automation-according-oxford-university-one/

UNIVERSITY OF LINCOLN

L-CAS

BREAK?!

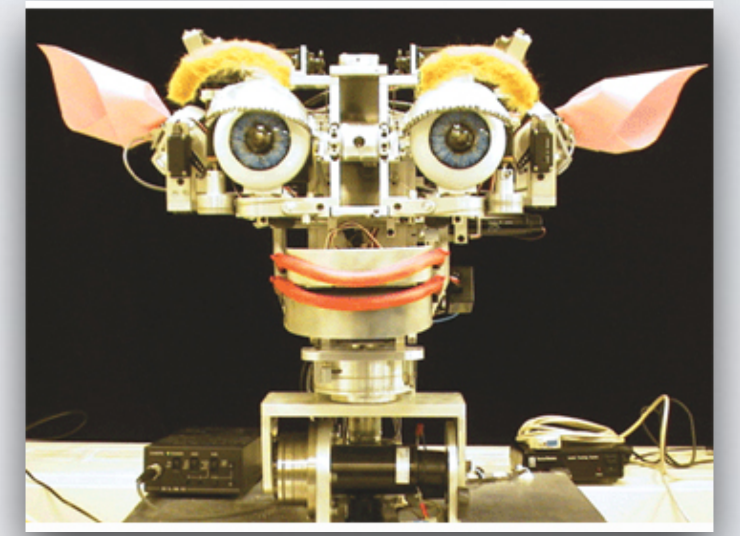
SENSORS AND ACTUATORS

EFFECTORS AND ACTUATORS

- Effectors:
  - hand, arm, gripper – manipulators
  - wheels, legs, tracks, rotors – mobile robots
- Actuators:
  - electric motors, pneumatic and hydraulic systems

EFFECTORS – EXAMPLES

pan-tilt unit



robotic head

parallel arm

gripper

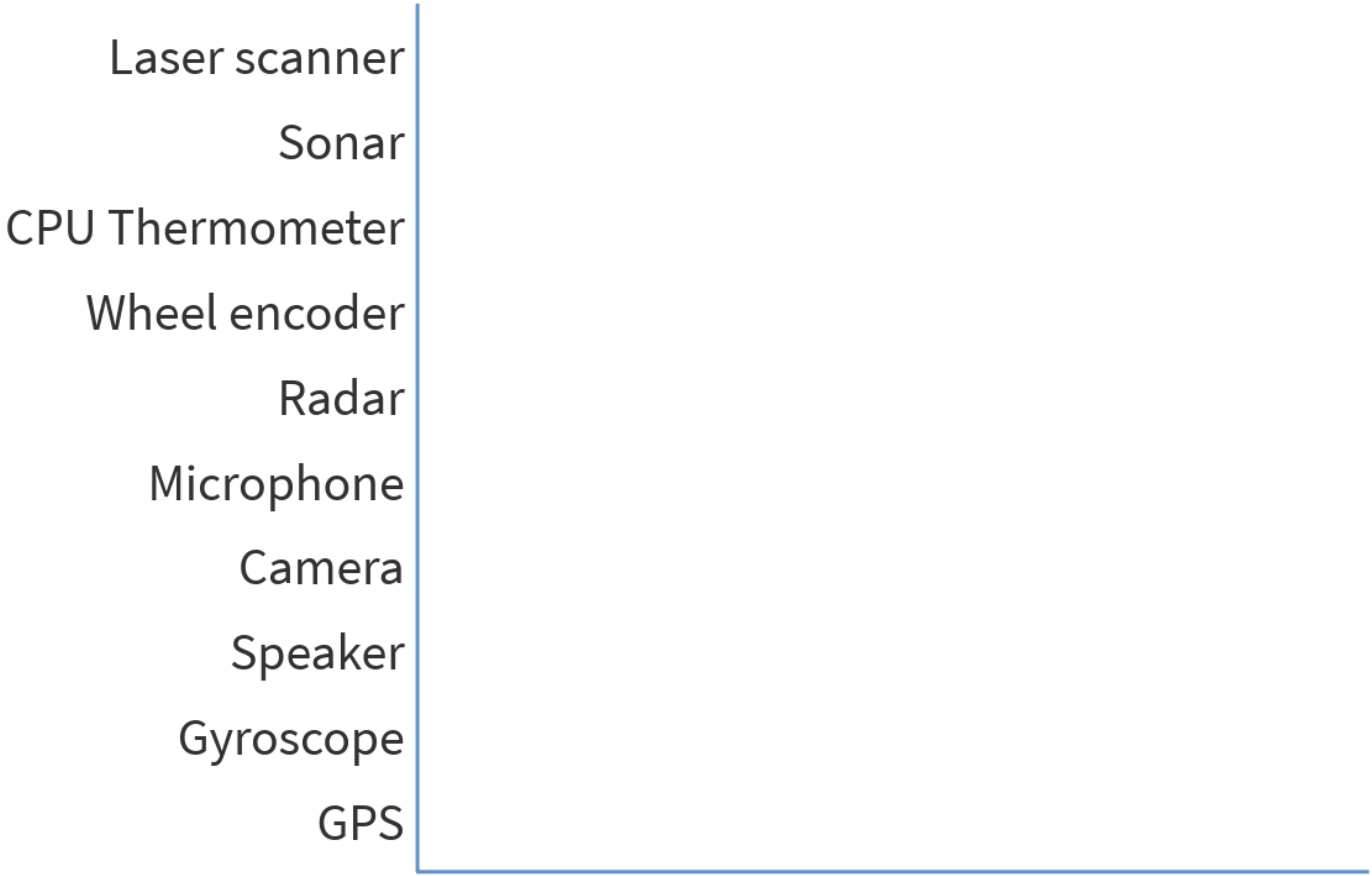
SENSORS: CLASSIFICATION

- **Proprioceptive**
  - internal state of the robot, self-awareness
- **Exteroceptive**
  - state of the environment

UNIVERSITY OF LINCOLN

L-CAS

Which of the following are exterioceptive sensors?

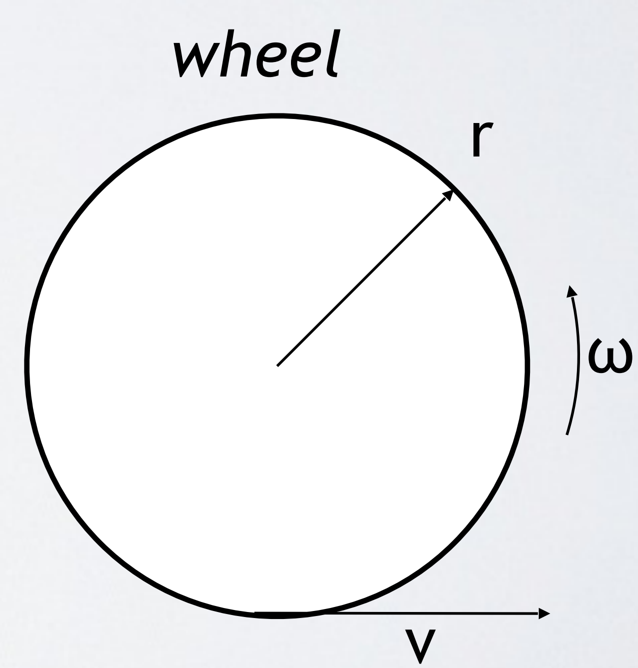


SENSORS: CLASSIFICATION

- **Proprioceptive**
  - internal state of the robot, self-awareness
  - e.g. odometry, battery level, temperature
- **Exteroceptive**
  - state of the environment, light intensity, distance measurements
  - e.g. sonars, video cameras
- Passive vs. Active
  - measuring phenomena directly or indirectly (e.g. reflected light/sound)

ODOMETRY – WHERE AM I?

- Dead reckoning
  - "deduced reckoning" – marine navigation
  - position estimation based on the previous known position
  - used by animals: pigeons, ants
- Mobile robots – odometry sensor
  - measure the speed of each wheel
  - use wheel geometry to calculate the velocity of a robot



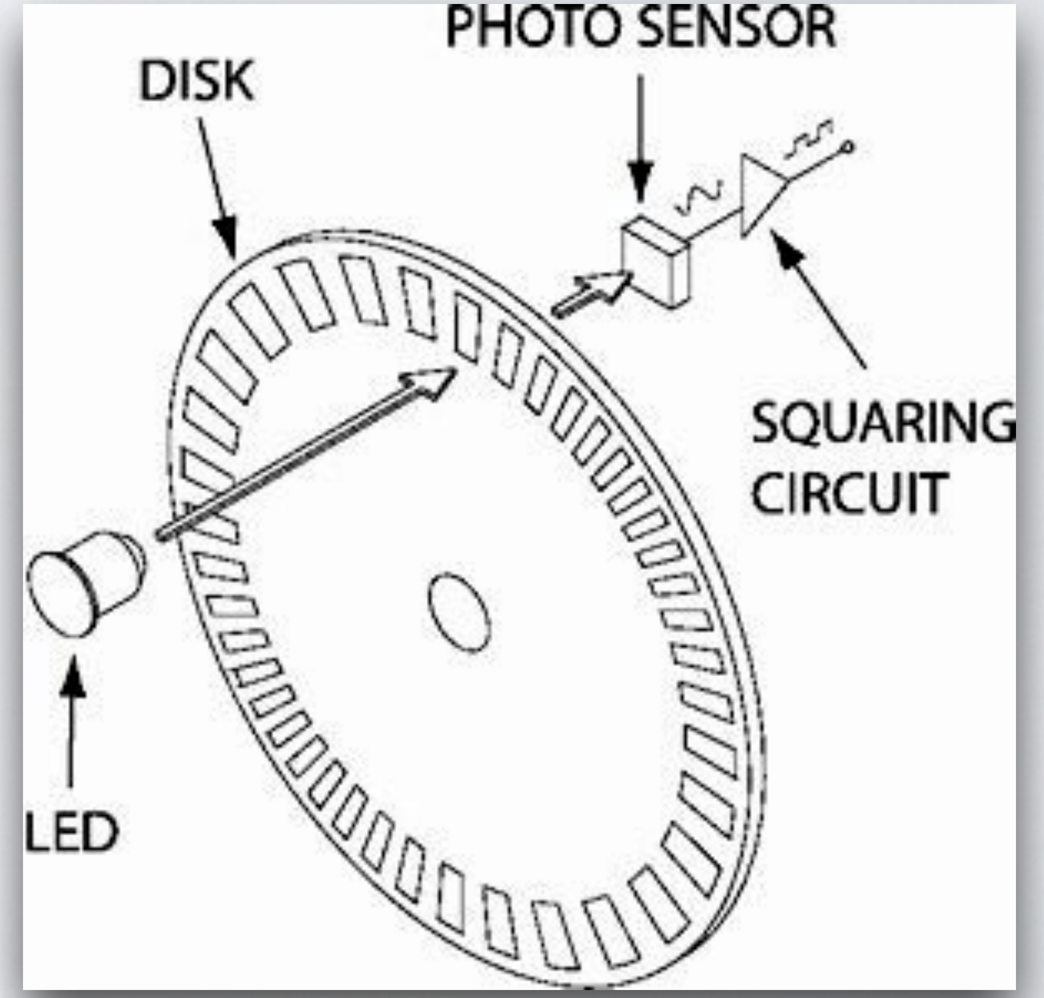
$v = r\omega$

WHEEL SPEED ESTIMATION

- Nominal motor speed + gear ratio
  - provided by the motor manufacturer
  - may change under different loads

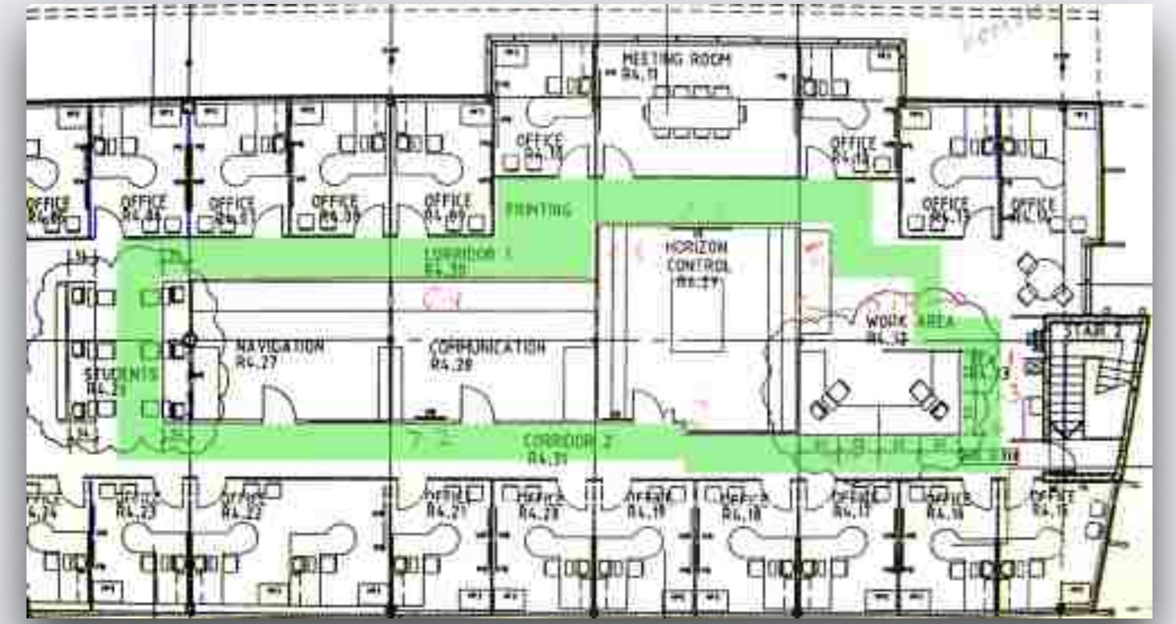
- Motor Encoder
  - sensor mounted on the wheel shaft
  - counts motor revolutions
  - similar to a computer mouse

$\omega = ?$

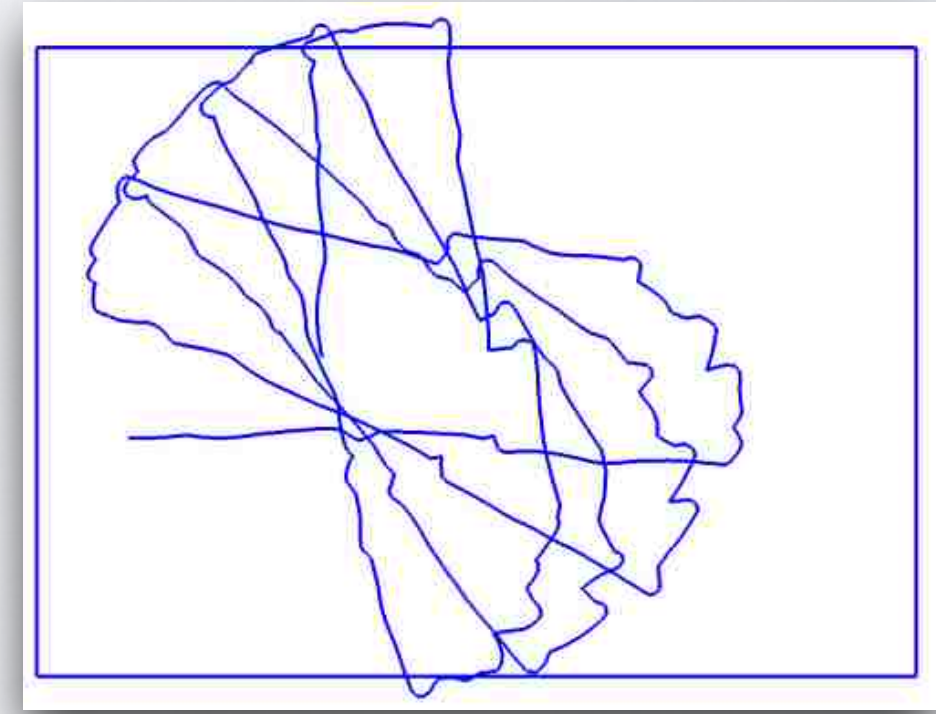


ODOMETRY – LIMITATIONS

- Wheel slippage, uneven friction and wheel size, etc. can cause errors that accumulate with time



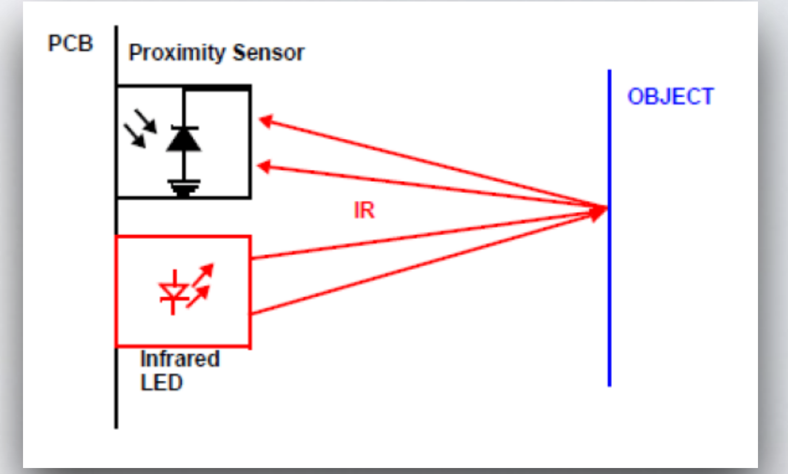
Floor plan and robot route



Robot's perceived trajectory using odometry

LIGHT SENSOR

- Passive – measuring light intensity directly (e.g. photo-resistor)
  - light can be used to mark important places e.g. recharging station, exit from the room, etc.
- Active – measuring reflected light
  - e.g. IR sensor
  - inexpensive but short range



UNIVERSITY OF LINCOLN

L-CAS

TIME OF FLIGHT SENSORS

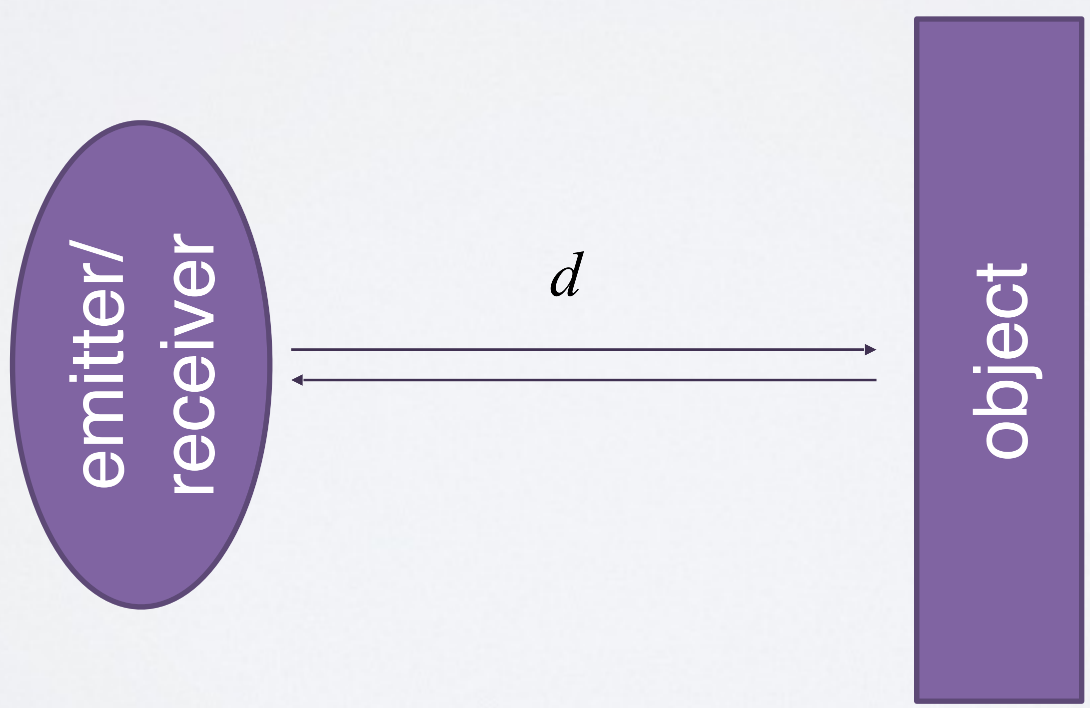
- Active distance measurements – reflected signal

signal speed

$$d = \frac{v \cdot t}{2}$$

distance

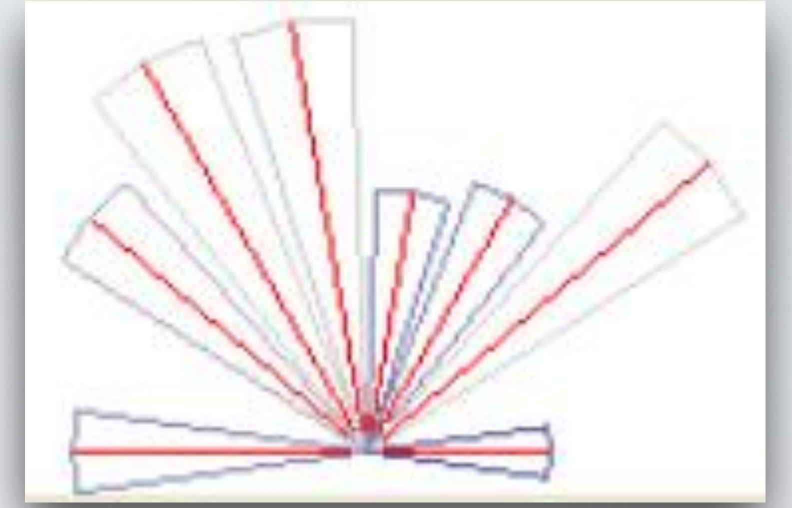
time elapsed between the start of emission and reception

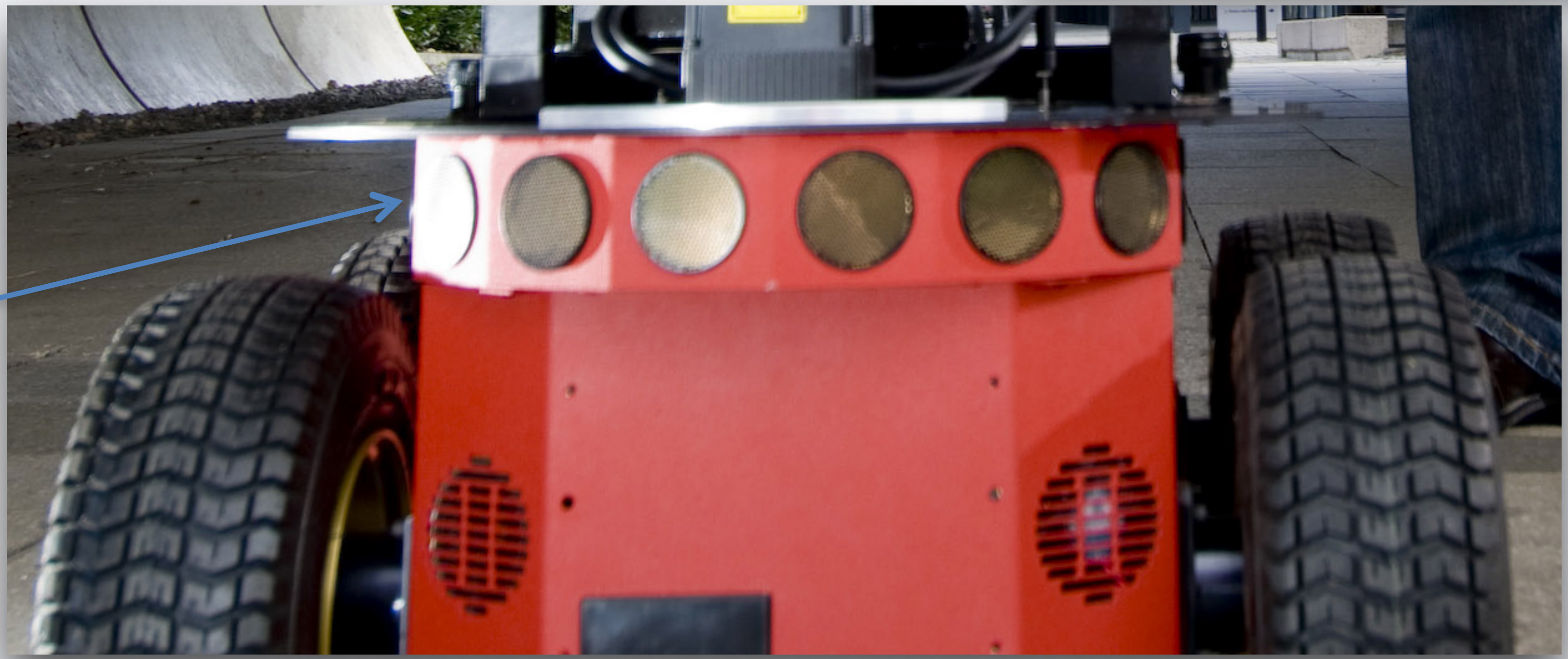


SONAR SENSOR

- Sonar
  - ultrasonic signal
  - v is speed of sound = 343 m/s
  - processing is 'slow'

sonar reading

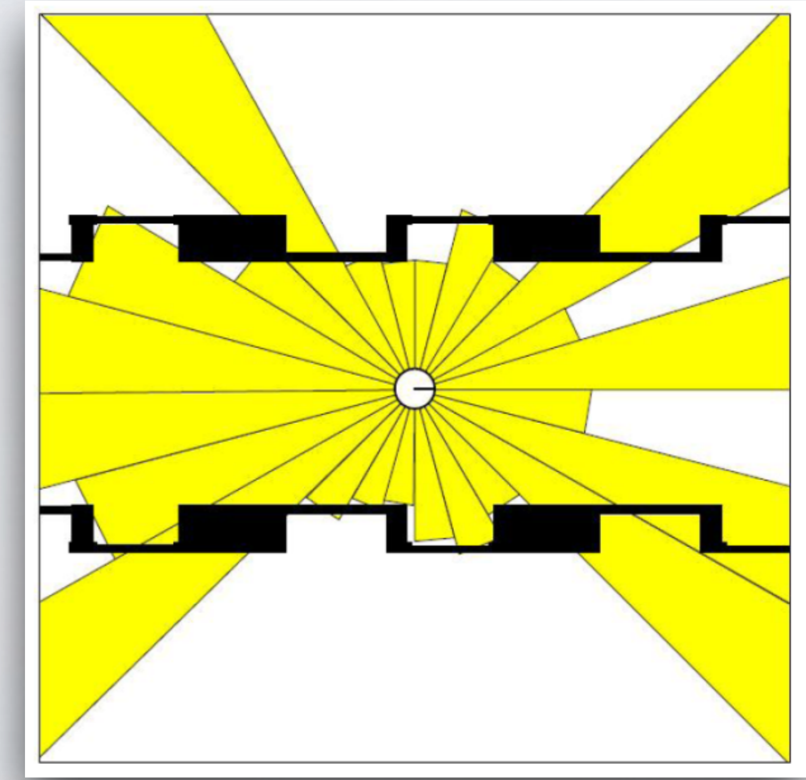




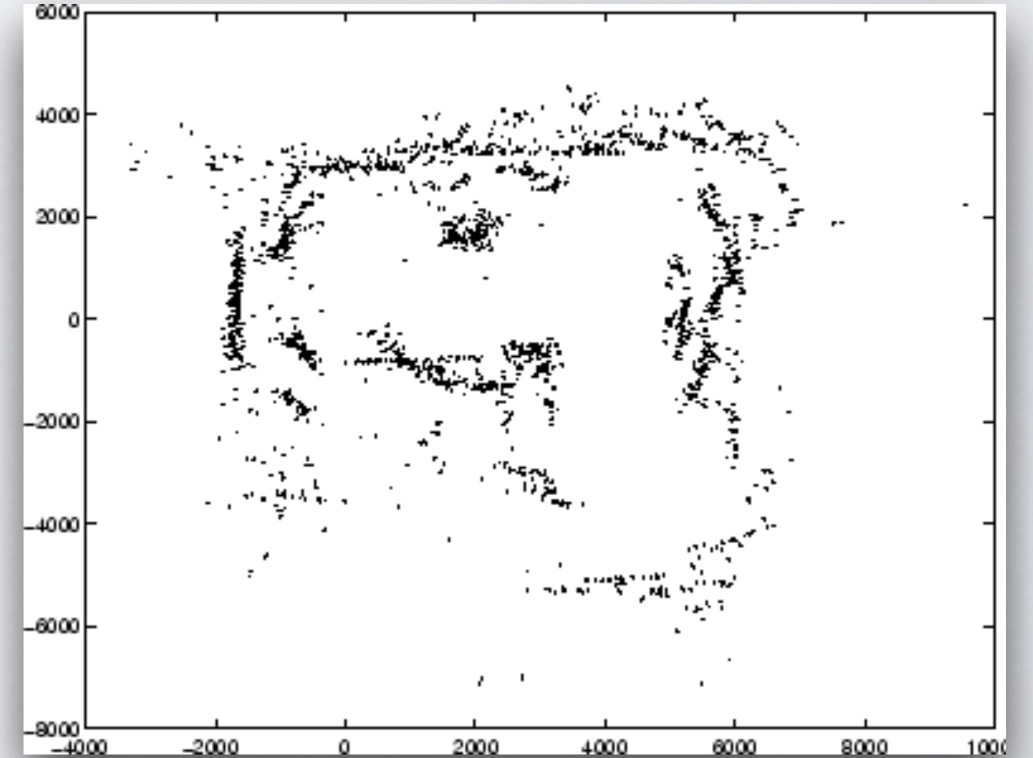
an array of sonar sensors

SONAR – APPLICATIONS

- Pros: cheap and good for obstacle avoidance
- Cons: slow and noisy



sonar reading in a corridor

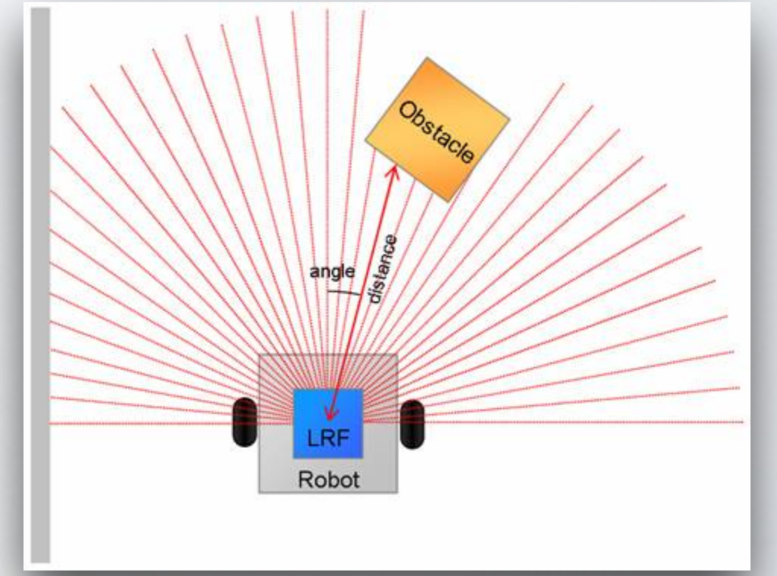


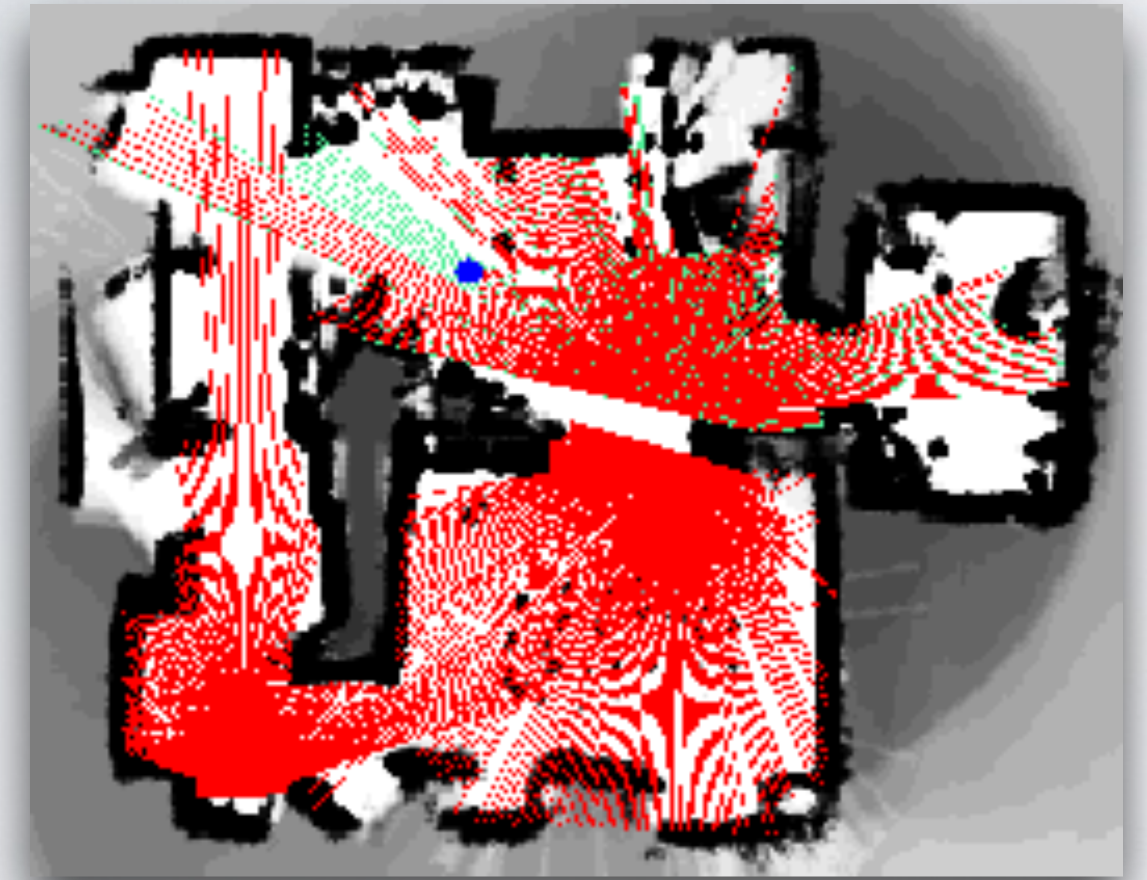
sonar map

LASER SENSORS

- Principle
  - time of flight sensor (light)
  - pulsed laser and rotating mirror

- Characteristics
  - high precision (mm)
  - long range (tens of meters)
  - wide field of view
  - fast (~30 scans/s)



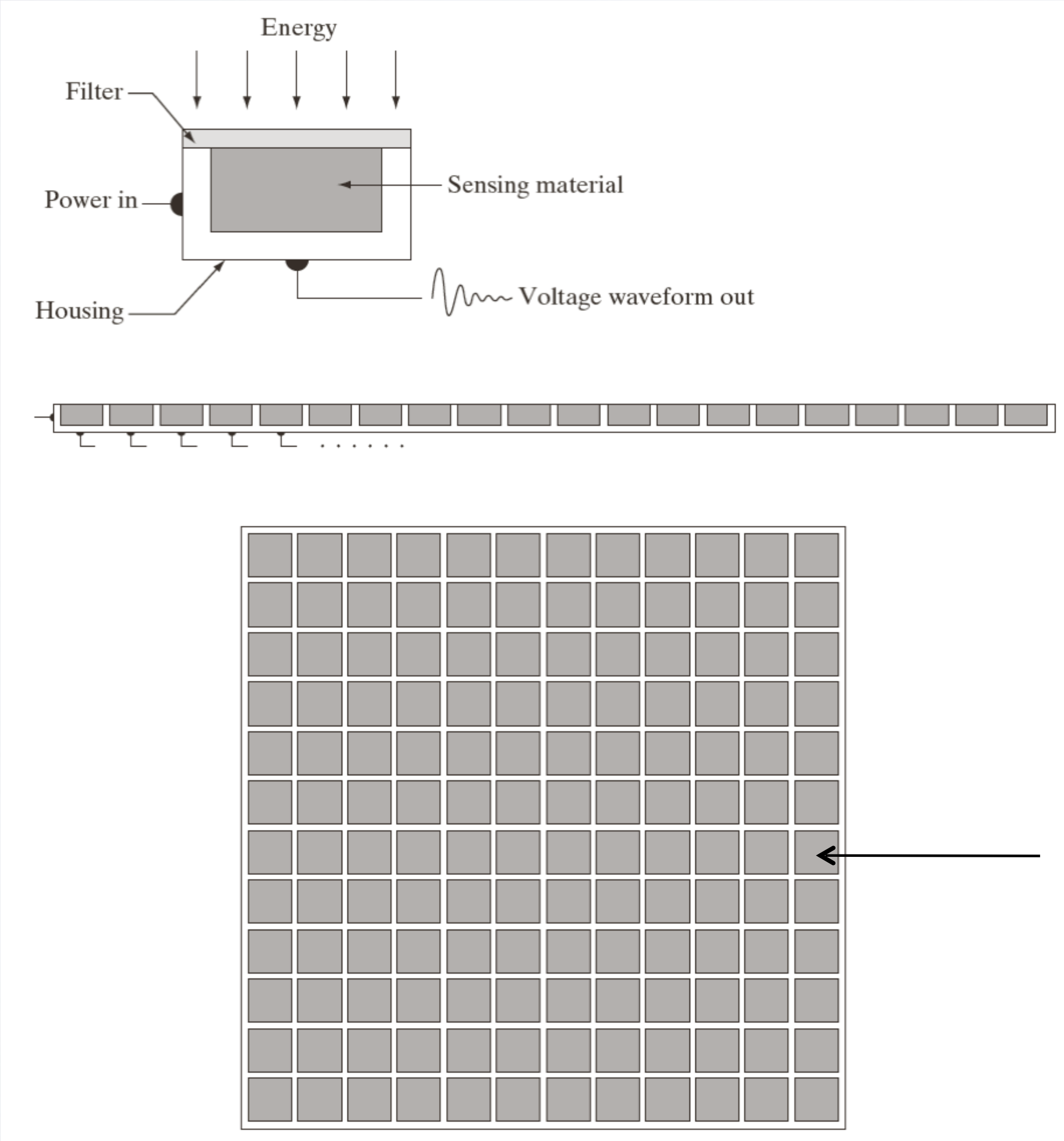


2d maps and people detection

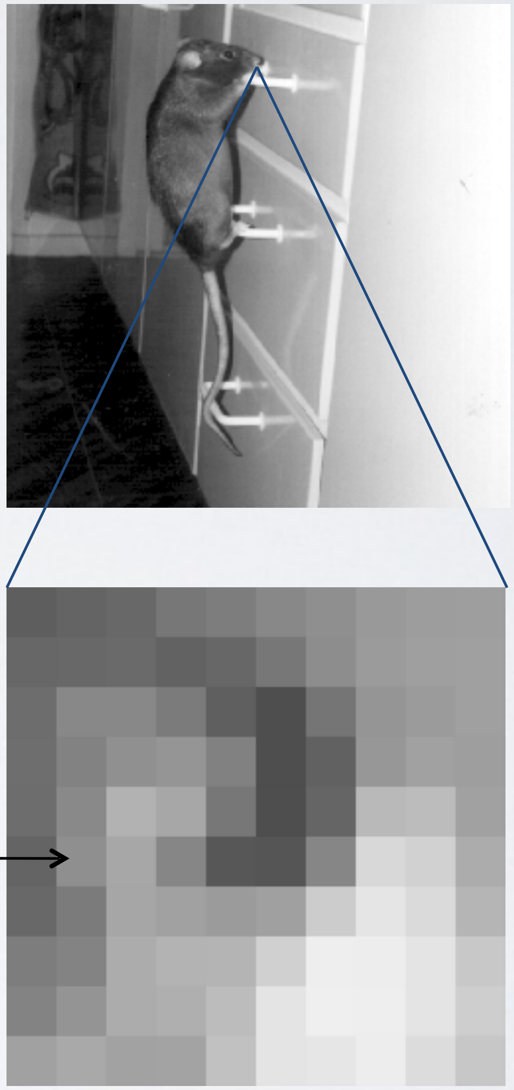
3d maps

VISION SENSOR

Vision Sensor



Digital Image



a rat

pixel

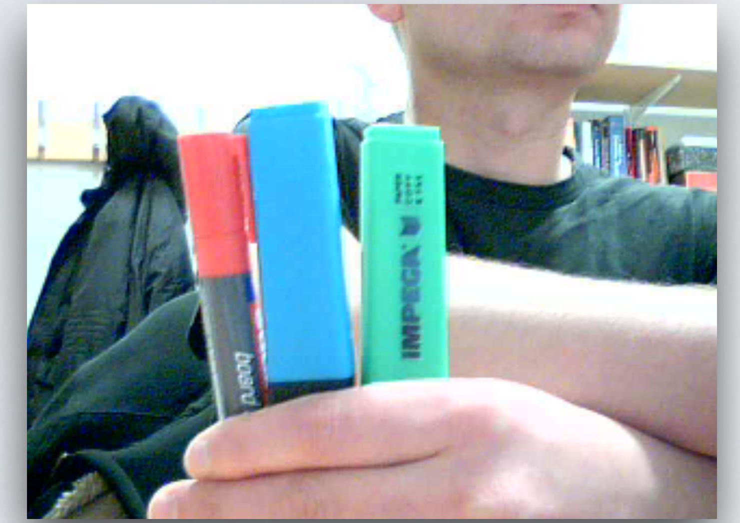
the rat's nose

UNIVERSITY OF LINCOLN

L-CAS

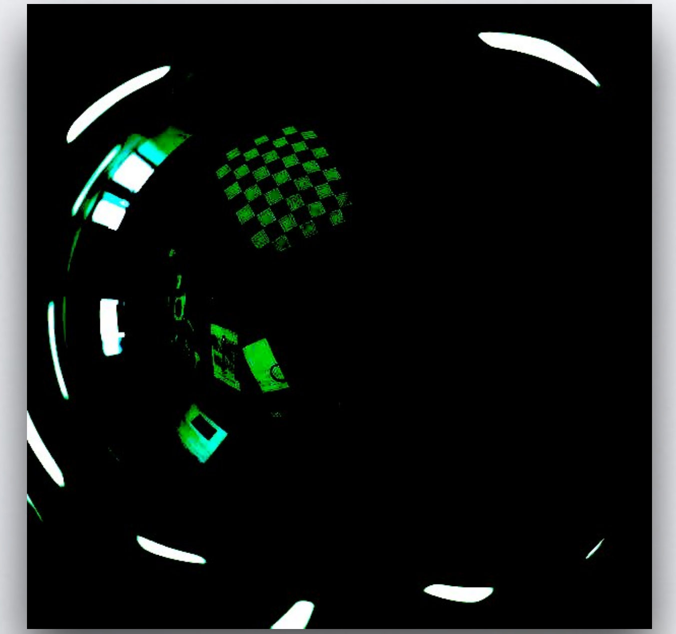
DIFFERENT TYPE OF VISION SENSORS

colour



thermal

omni-directional

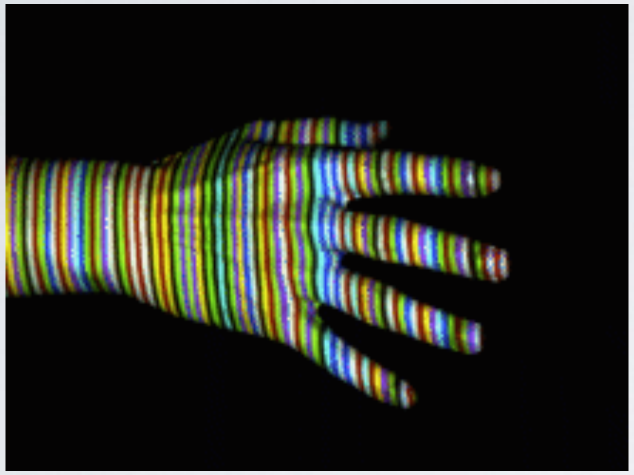
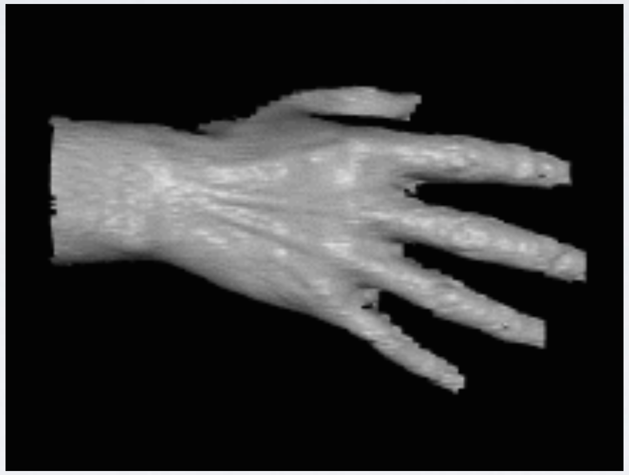
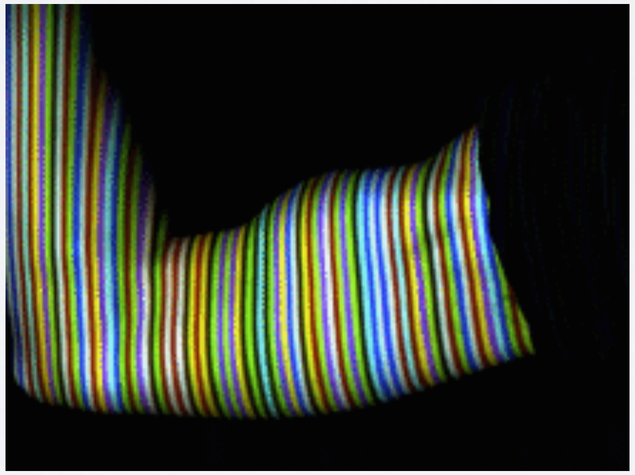
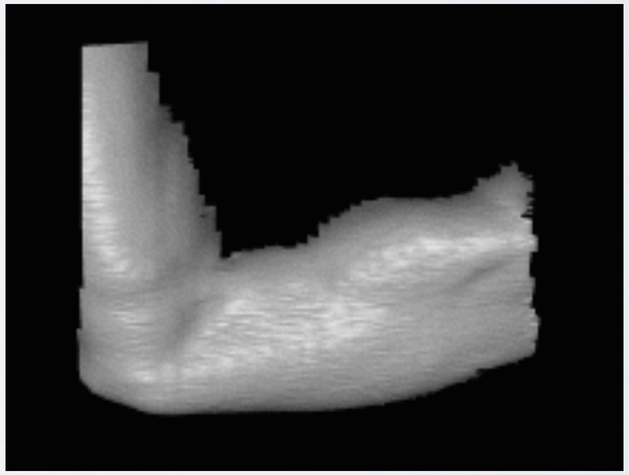


stereo

Kinect

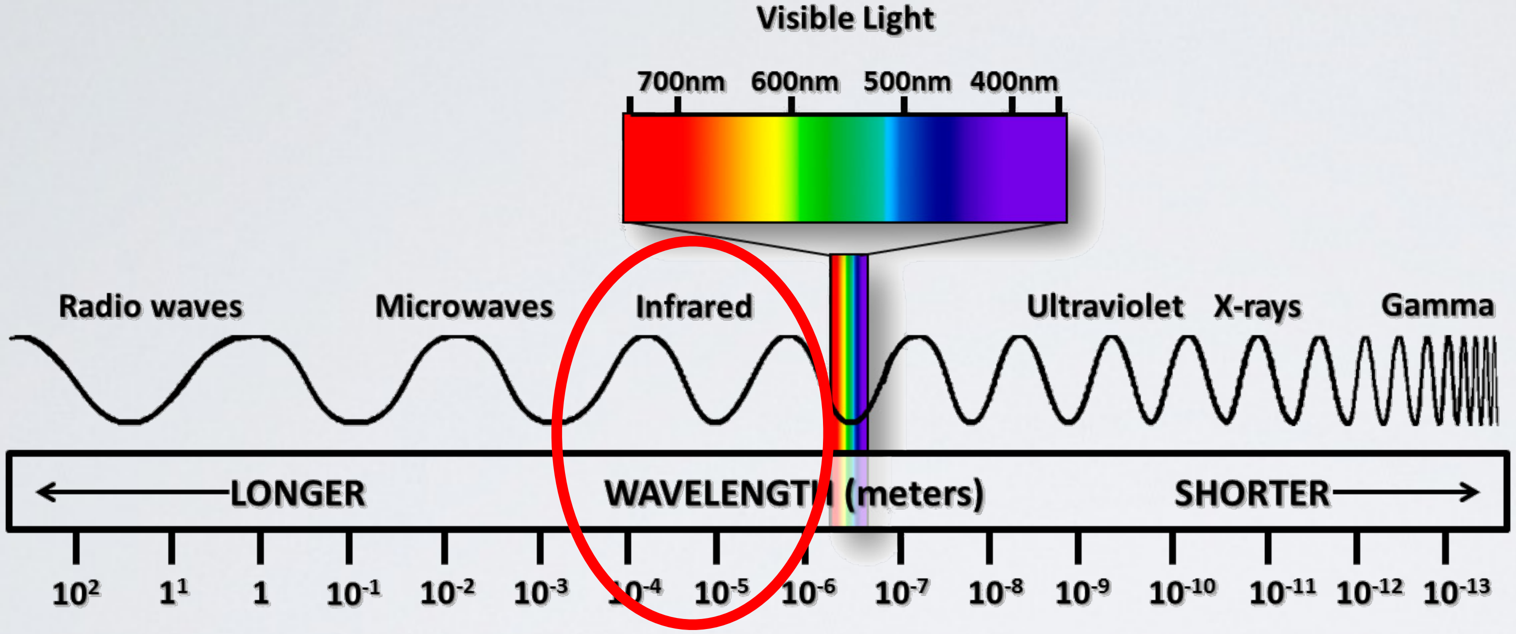
PrimeSense™
Natural Interaction

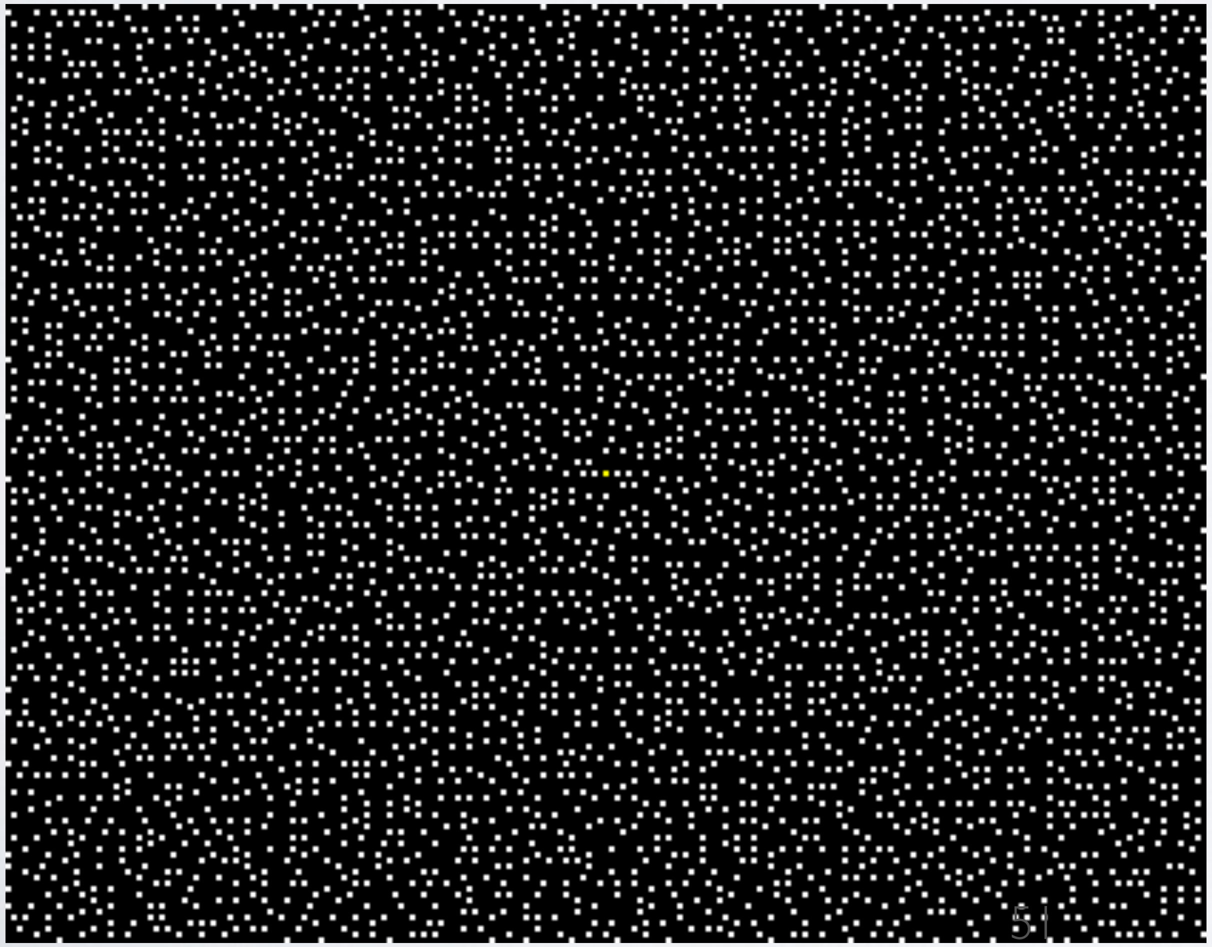
STRUCTURED LIGHT

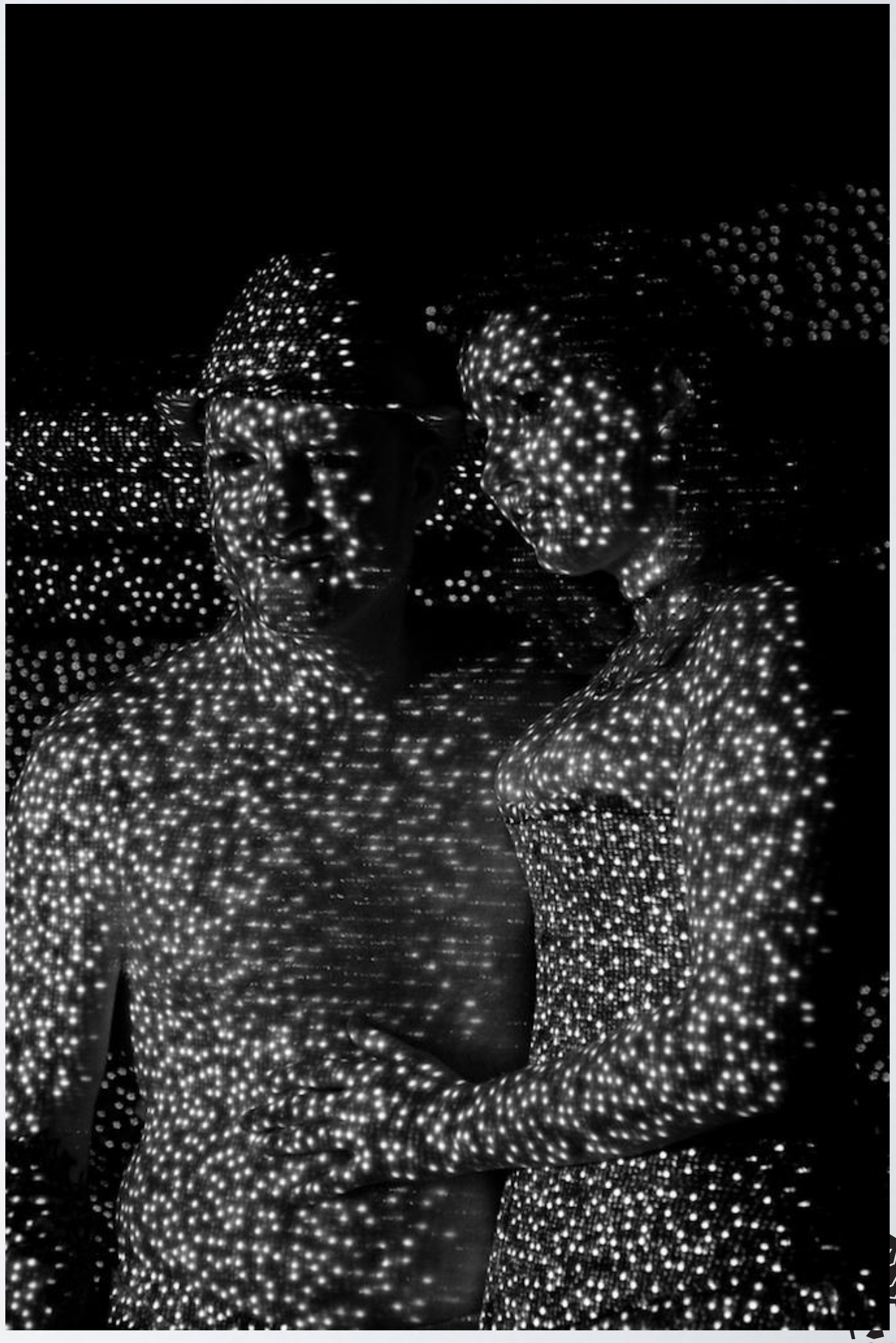





Zahng et al., 2002
http://grail.cs.washington.edu/projects/moscan/

STRUCTURED LIGHT - KINECT



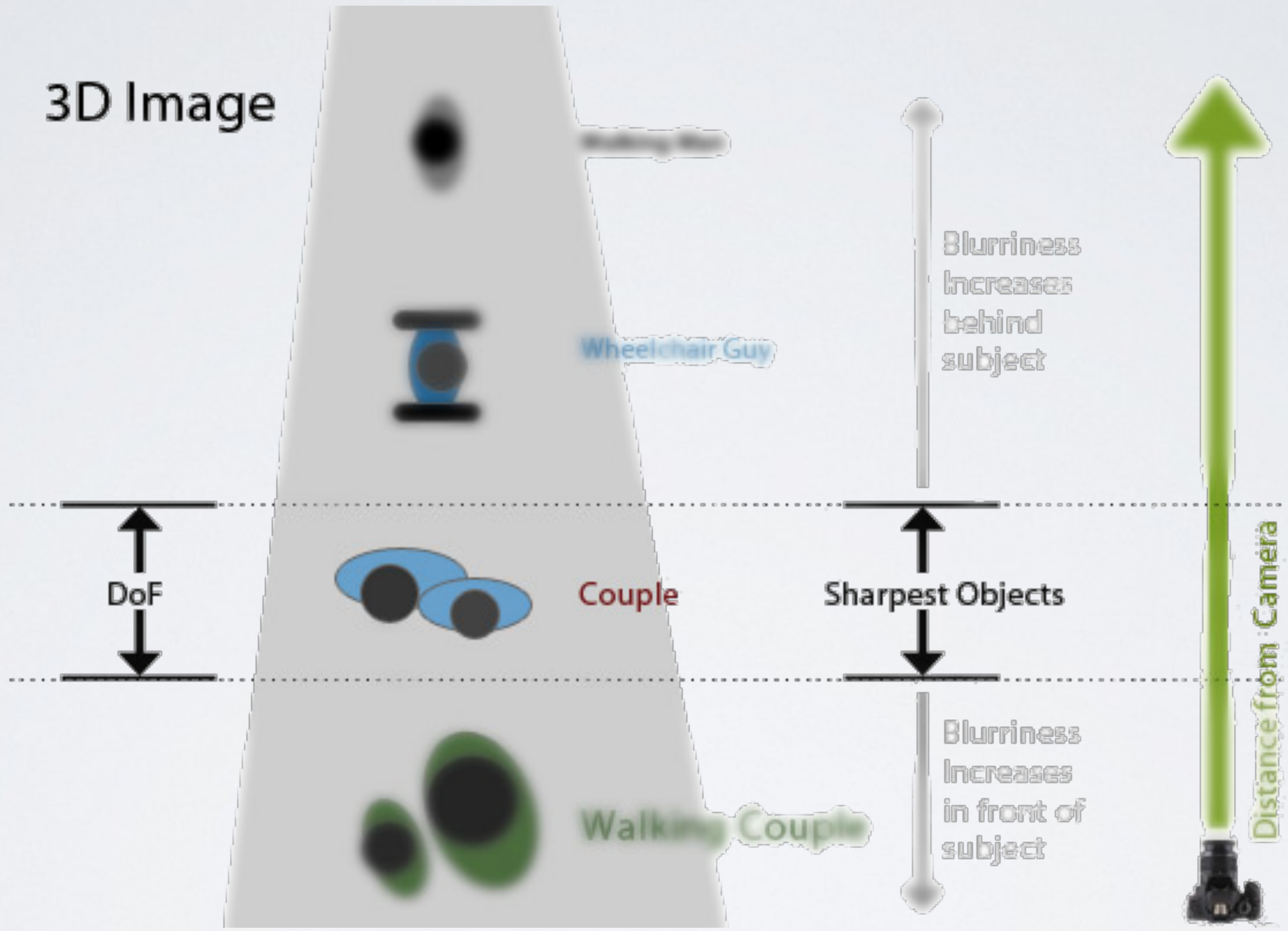




UNIVERSITY OF LINCOLN

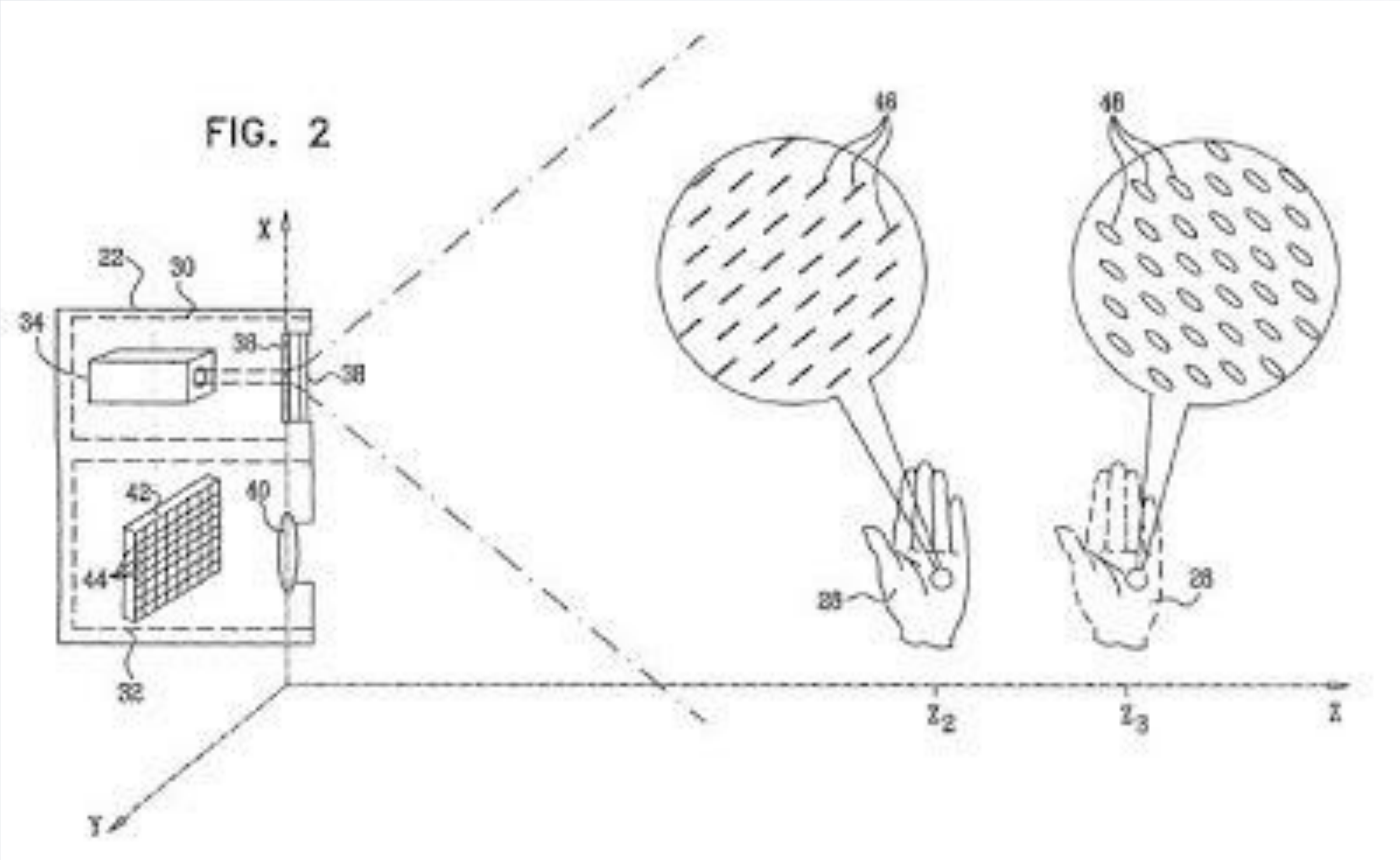
L-CAS

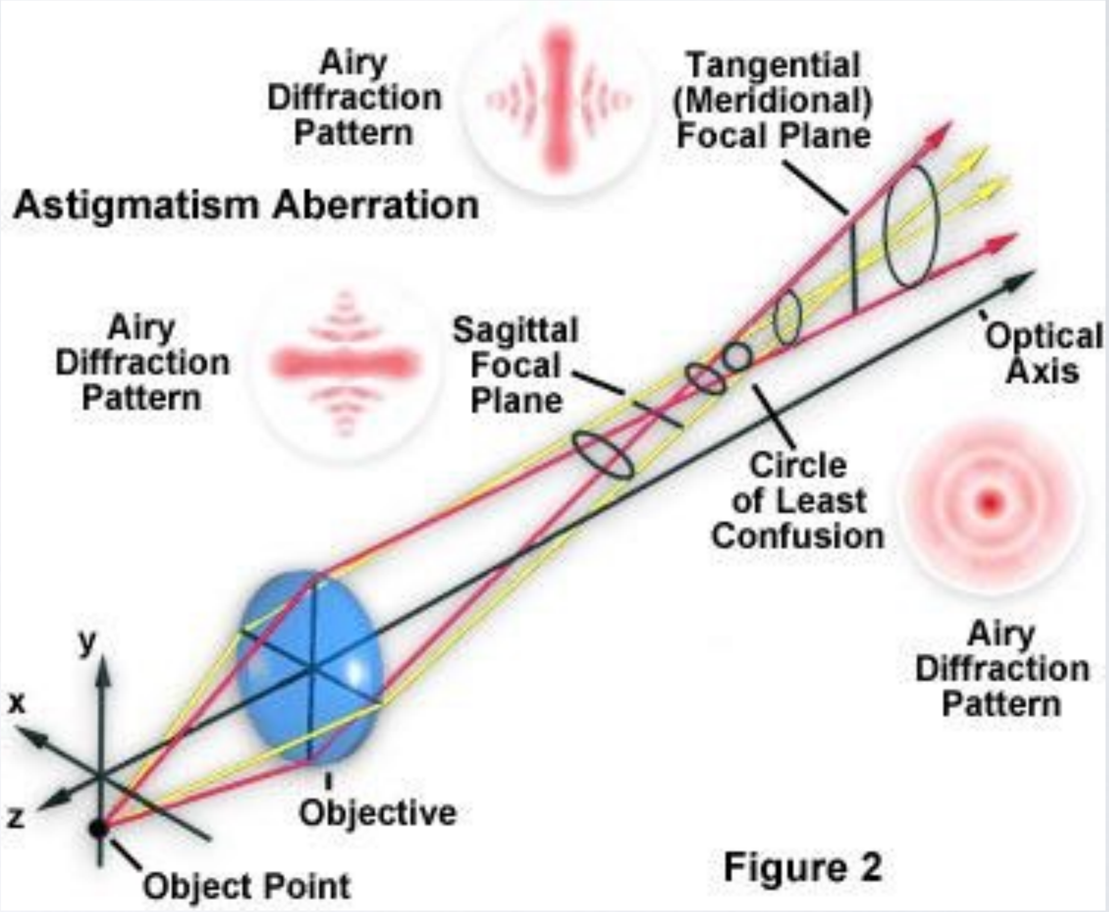
DEPTH FROM FOCUS



http://www1.cs.columbia.edu/CAVE/projects/depth_defocus/

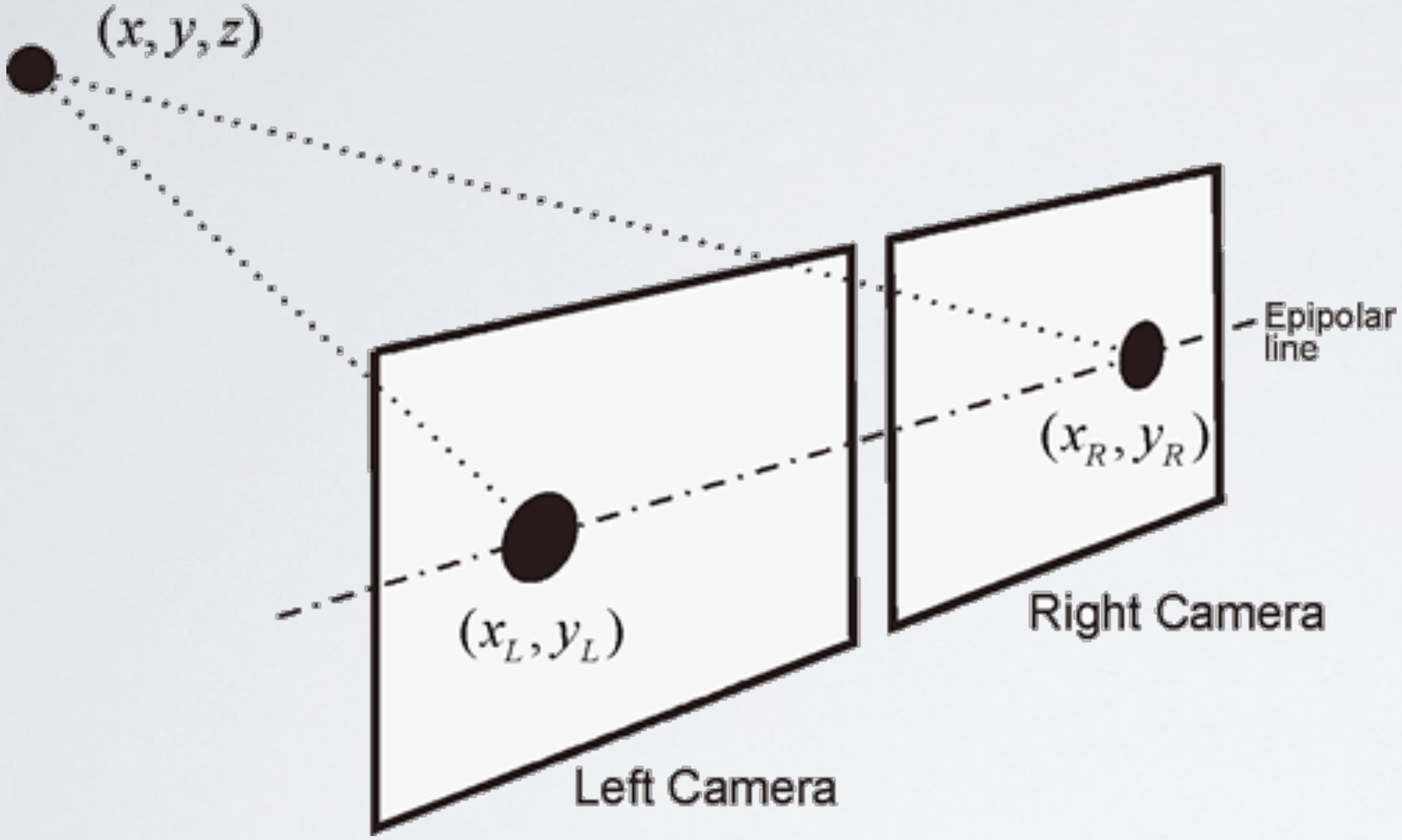
DEPTH FROM FOCUS - KINECT

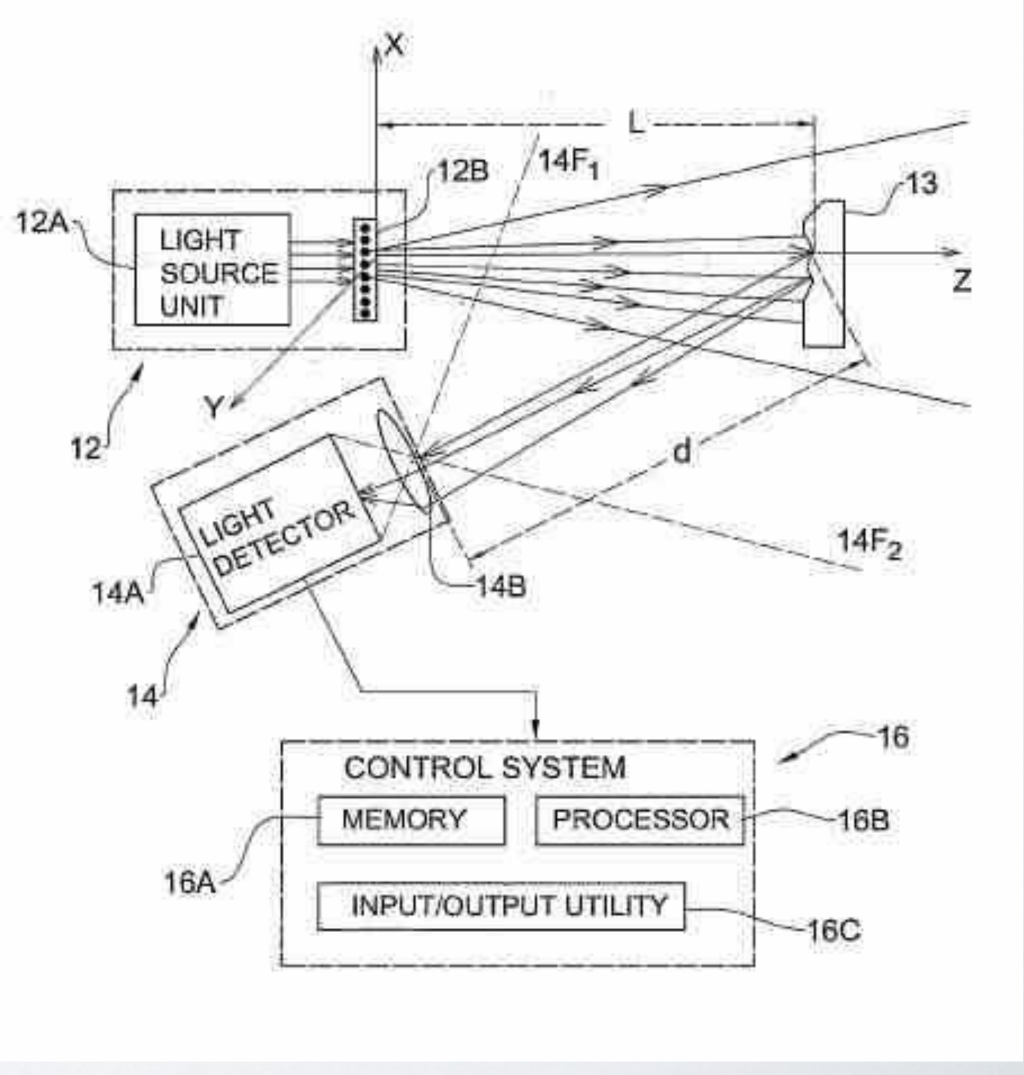




"astigmatic" lens with different focal length in x and y direction

DEPTH FROM STEREO



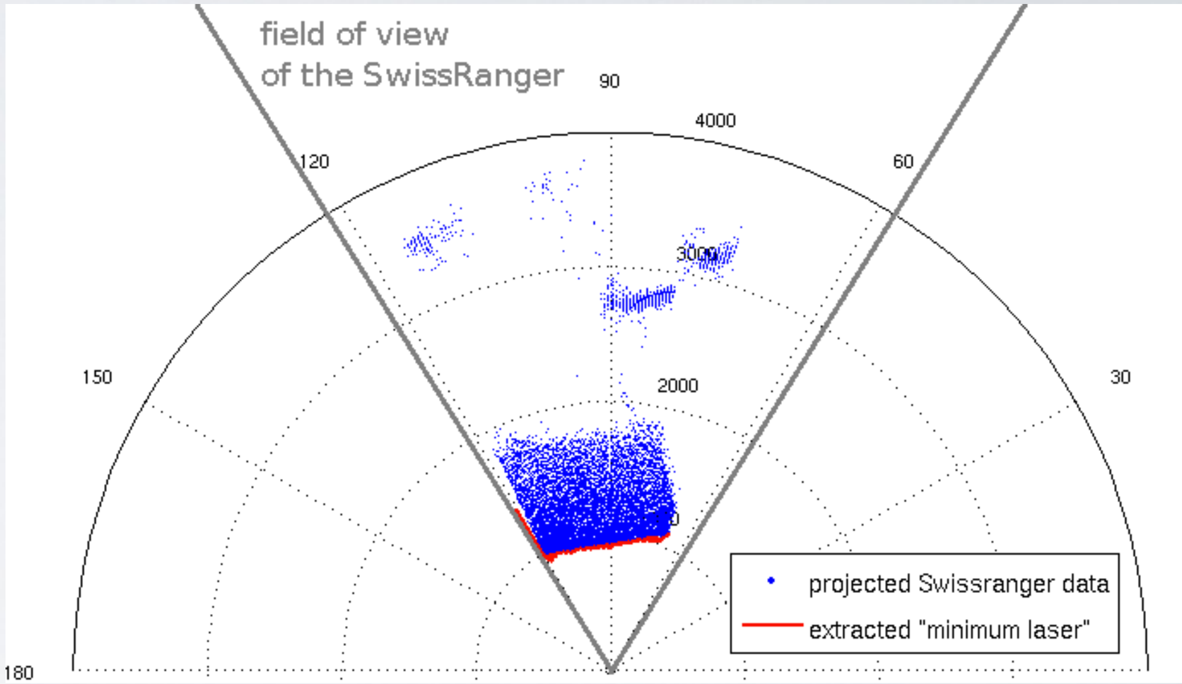


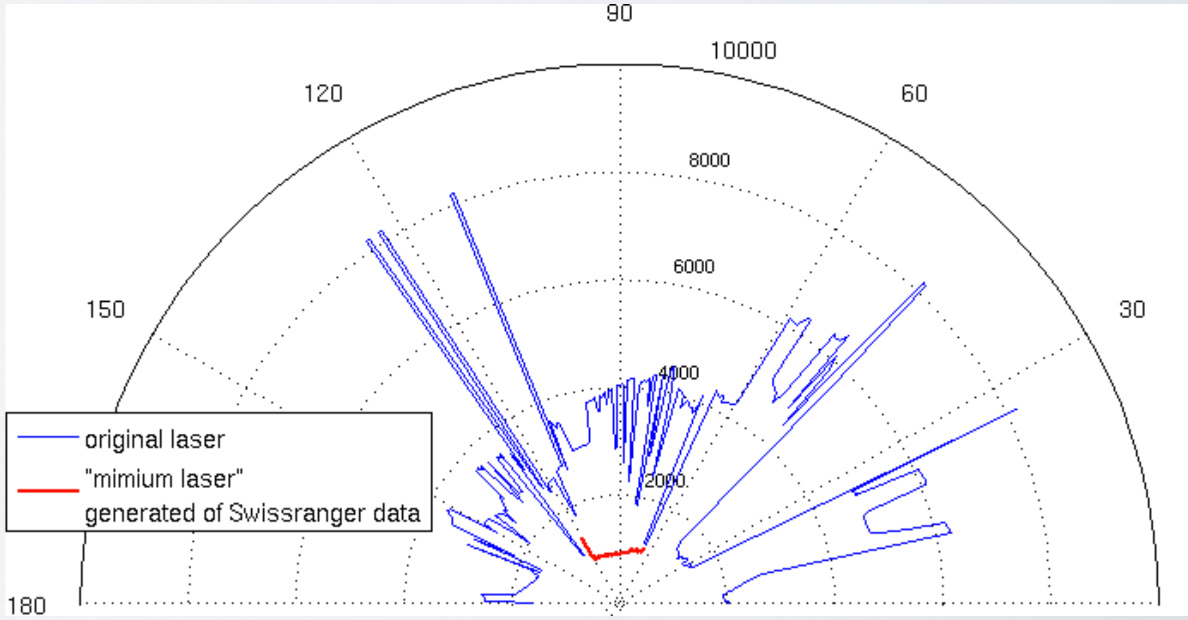
KINECT 2

- a "real" 512x424 pixel time-of-flight camera
  - higher resolution
  - higher frame rate
  - increased field of view

EMULATING A LASER SCANNER AND MERGING READINGS

- requires "calibration" of the two sensors (measuring where one is in relation to the other) => Rotation and Translation
    - in 3D given by 6 degrees of freedom
    - often represented as a 3x3 rotation matrix and a 3x1 translation vector

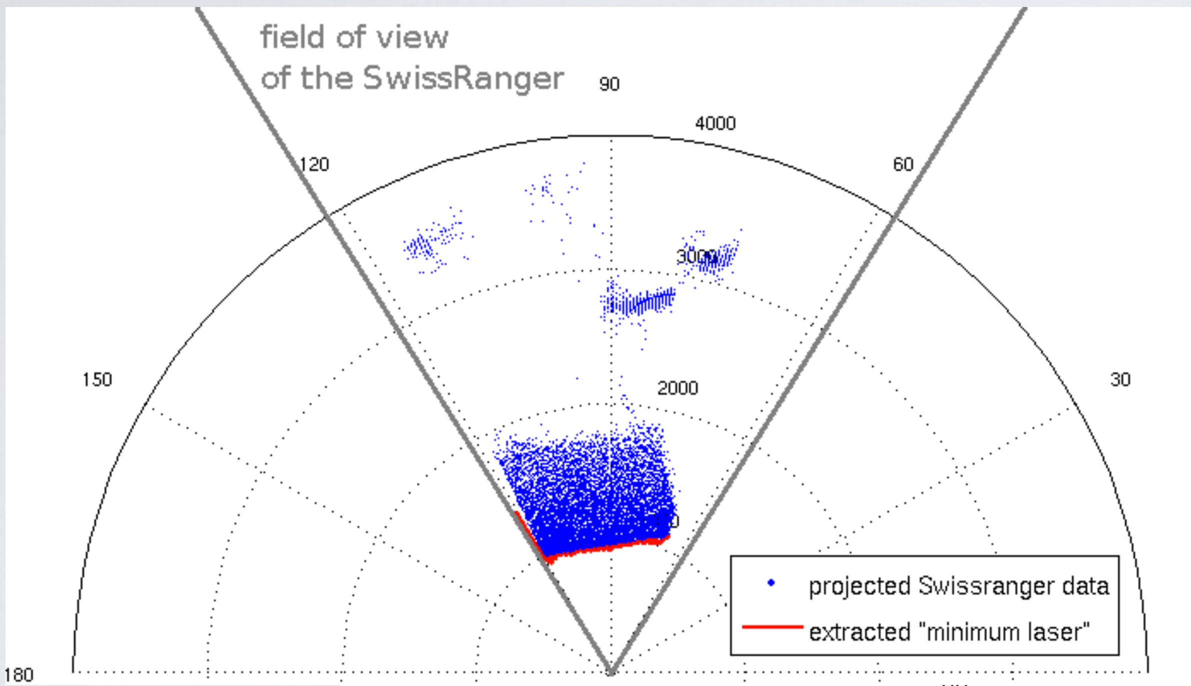


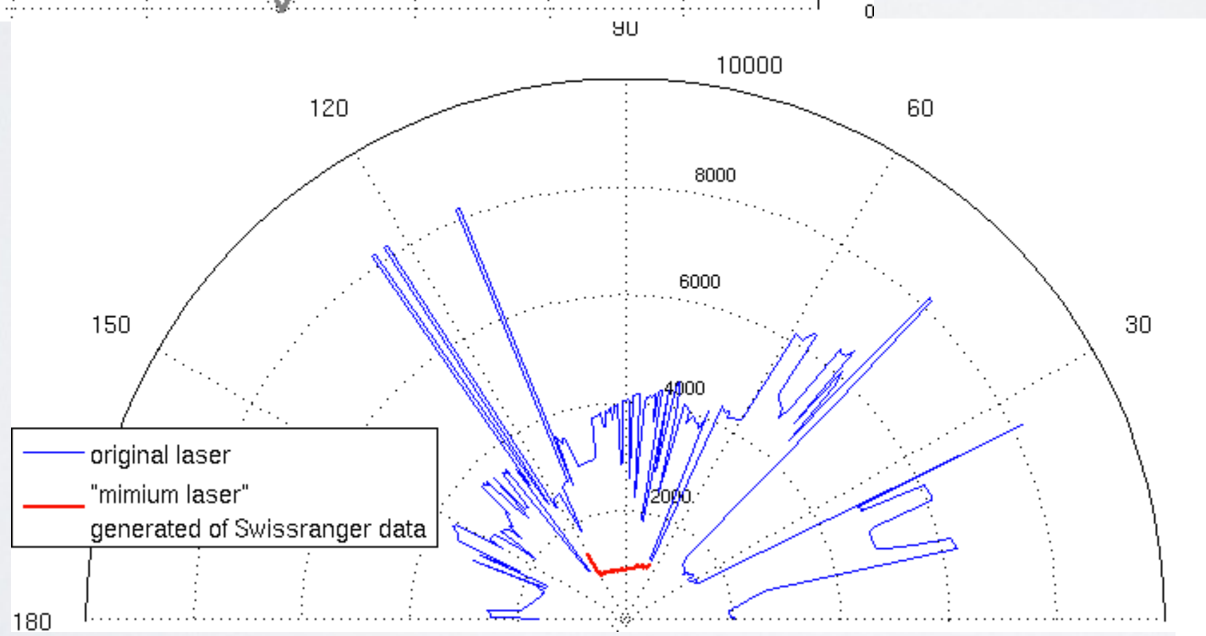


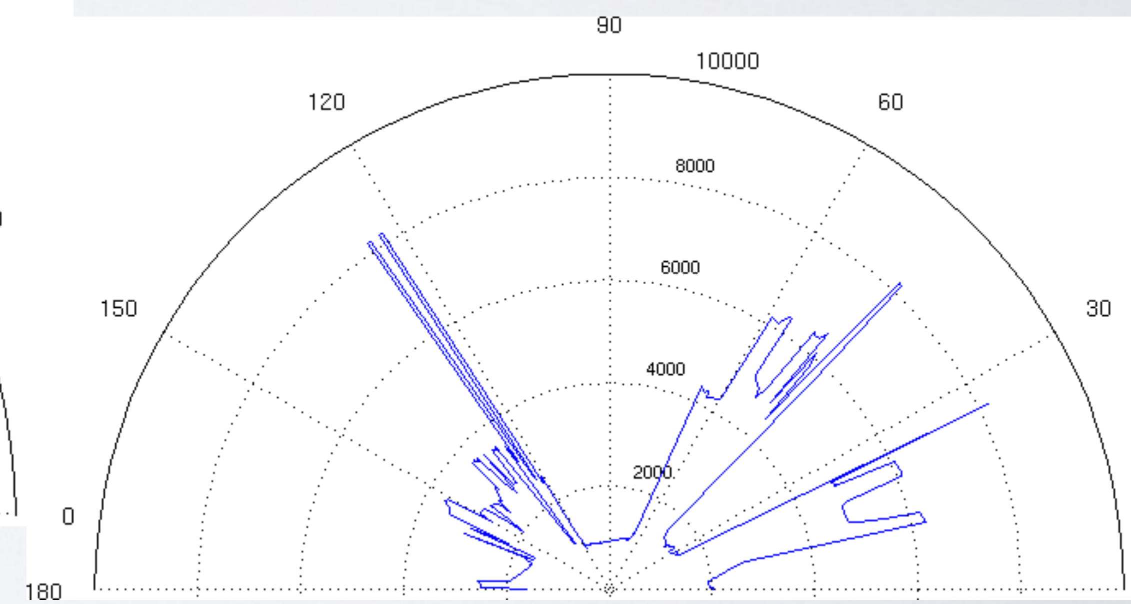
UNIVERSITY OF LINCOLN

L-CAS

AVOIDING THE TABLE

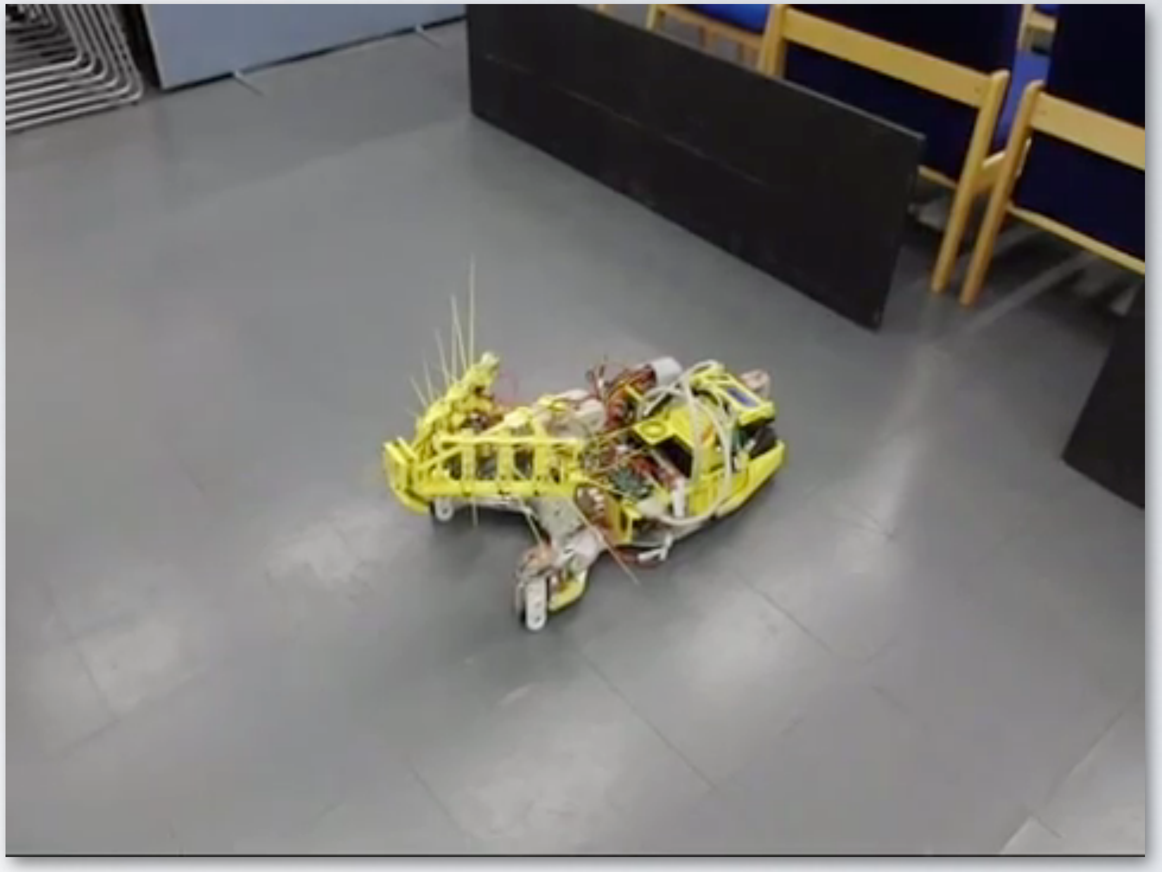


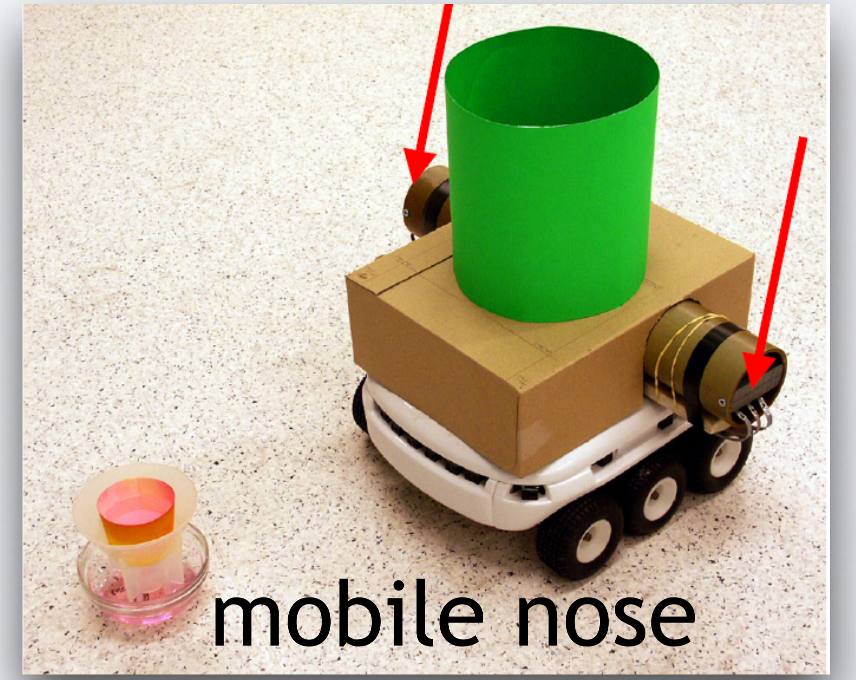




OTHER SENSORS

- Tactile
  - bumpers, whiskers, buttons
- Direction and orientation
  - compass, gyroscope, accelerometers
- Global position
  - GPS
- Motion
  - Gyroscope
- Temperature, sound, even smell!

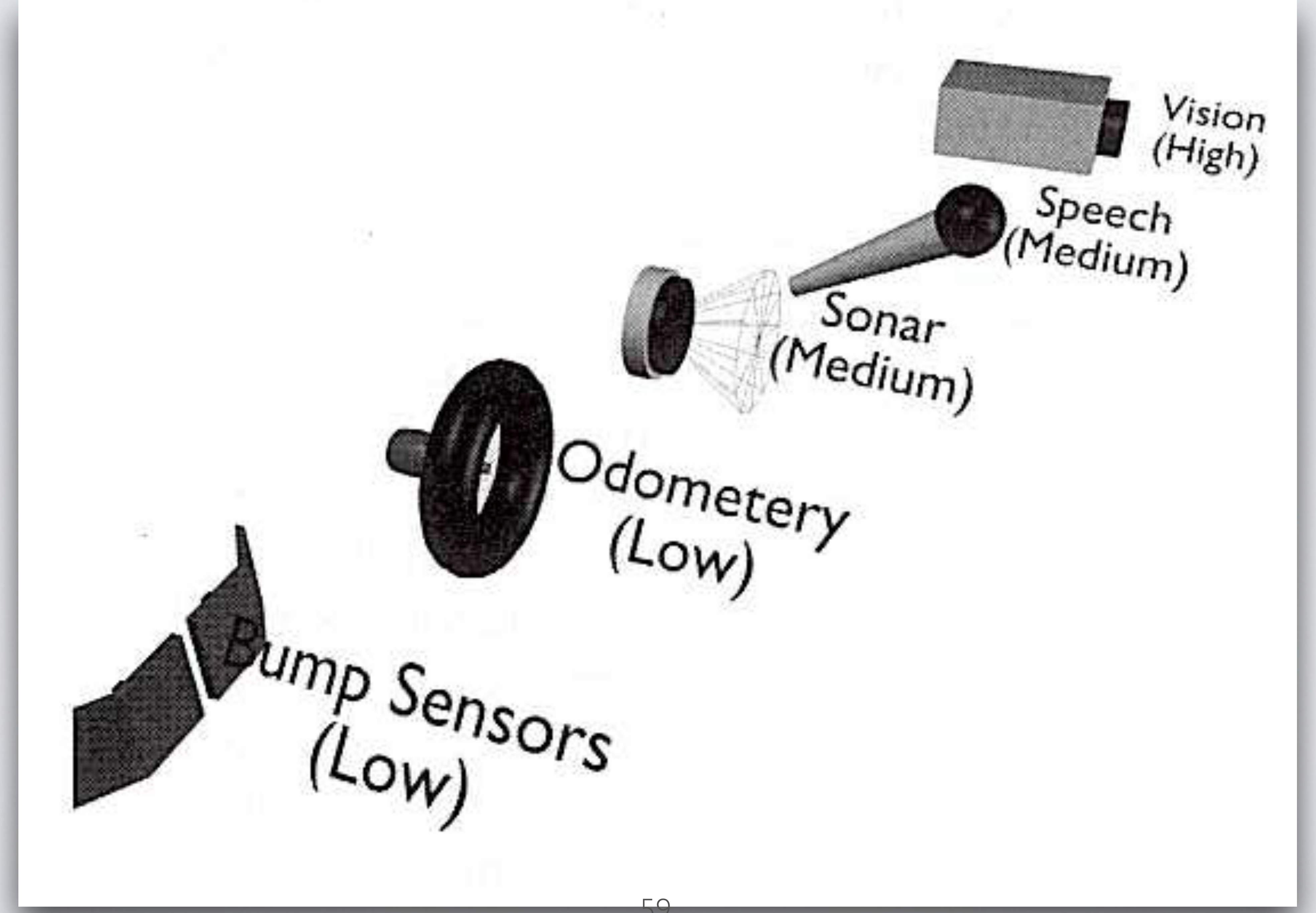




UNIVERSITY OF LINCOLN

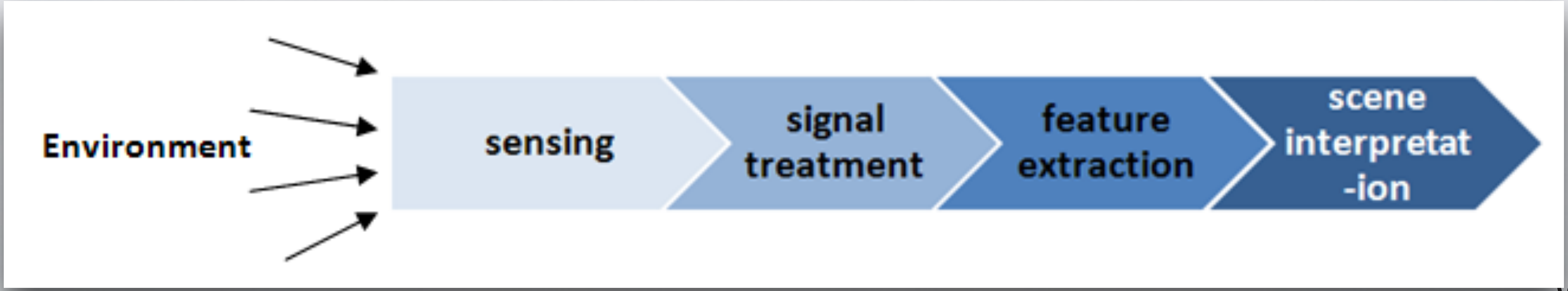
L-CAS

HIERARCHY OF PROCESSING REQUIREMENTS

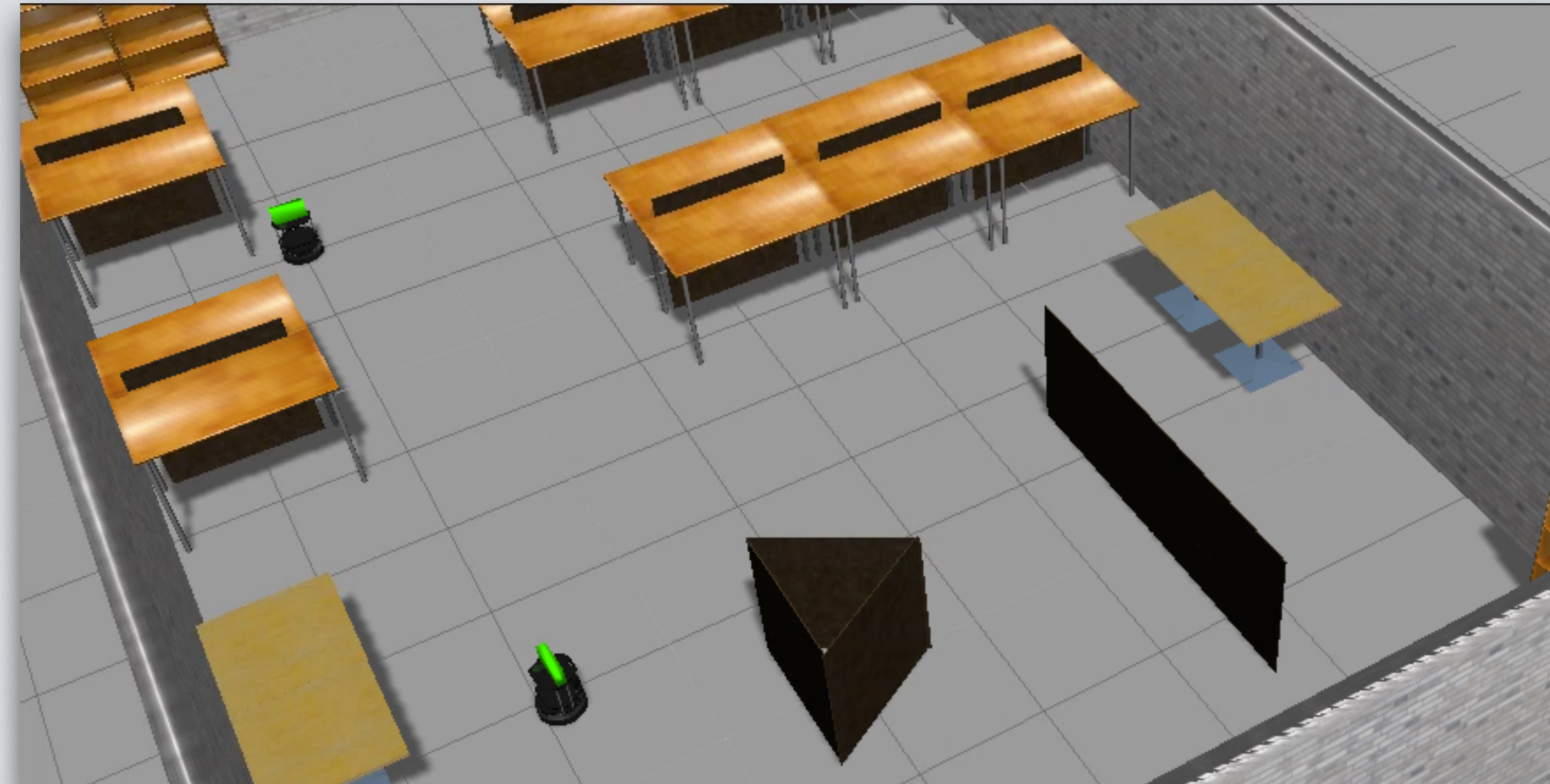


PERCEPTION

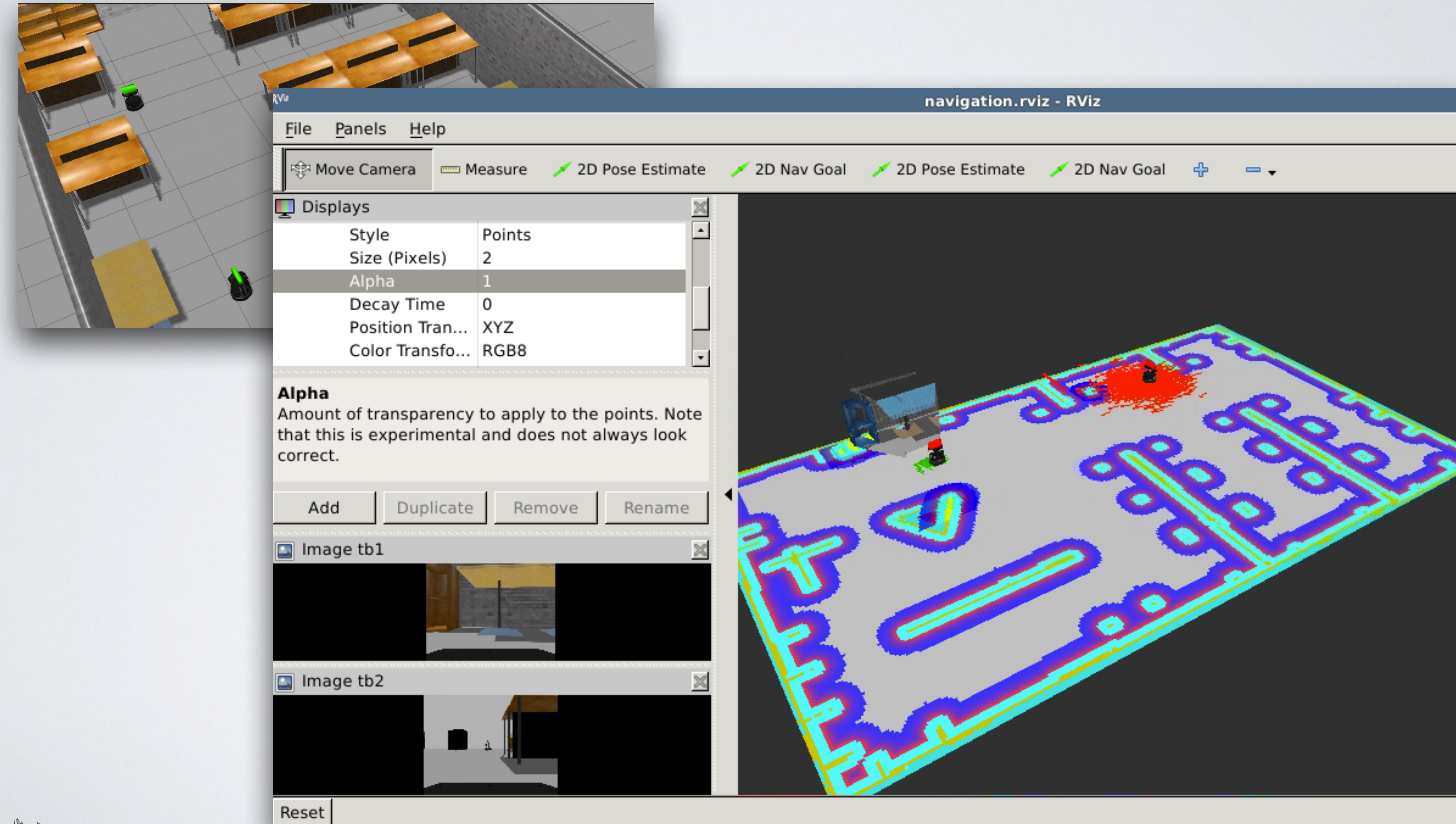
- Data Interpretation and Processing
  - sensors often do not provide direct measurements nor provide the exact values
  - some sensors provide very rich information (e.g. vision) that needs to be reduced
  - some sensors provide very sparse information that needs to be interpolated
  - The **data bandwidth** of sensors differs significantly!



SIMULATED TURTLEBOTS



SIMULATED TURTLEBOTS



UNIVERSITY OF LINCOLN

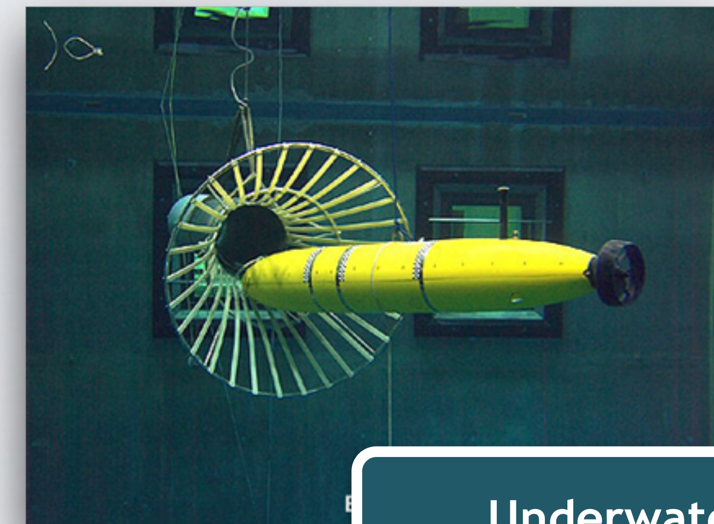
L-CAS

APPLICATIONS

Which robot is more autonomous?

Which one of these robots is more autonomous?

DIFFERENT ROBOT – DIFFERENT APPLICATION



Underwater

autonomous exploration

Tracked

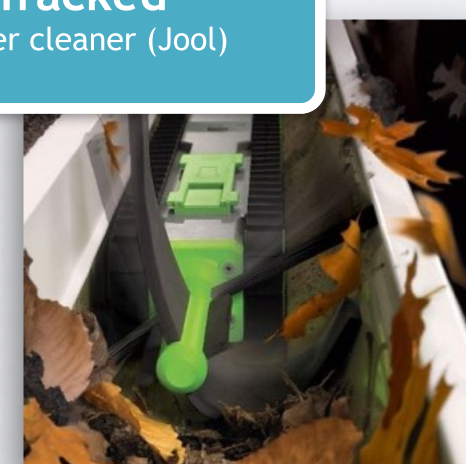
inspection of radiated areas (Pioneer)

Airborne

surveillance (MQ-1 Predator)

Tracked

gutter cleaner (Jool)

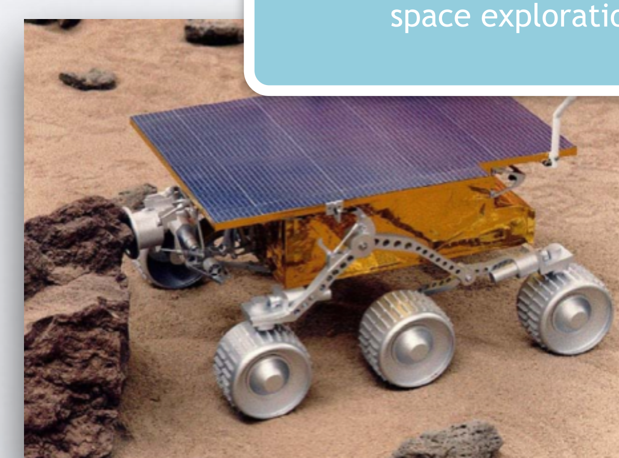


DIFFERENT ROBOT – DIFFERENT APPLICATION

Legged
exploration of volcanoes (Sojouner)

Legged
transportation (BigDog)

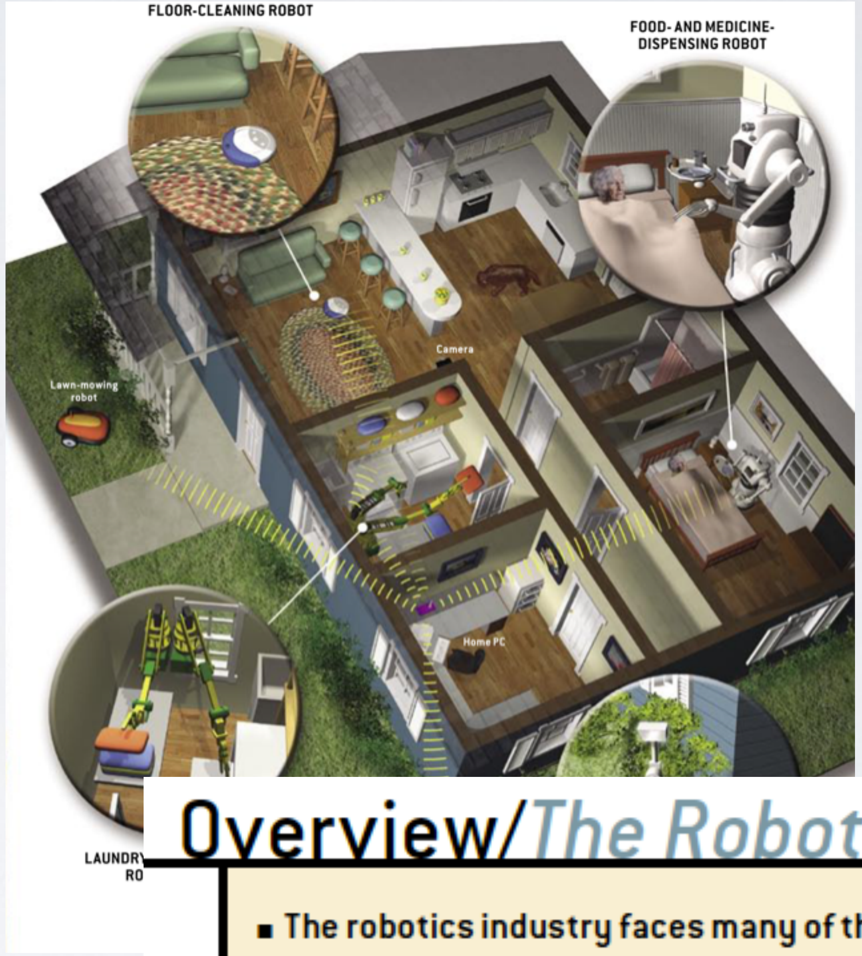
Wheeled
space exploration



Wheeled
autonomous car (DARPA Grand Challenge)

A ROBOT IN EVERY HOME

- Service Robots
  - cohabitant with humans in their everyday environments
- Bill Gates predicts (in 2006):
  - “in 20 years (2026) there will be a robot in every home”
  - read his article in Scientific American

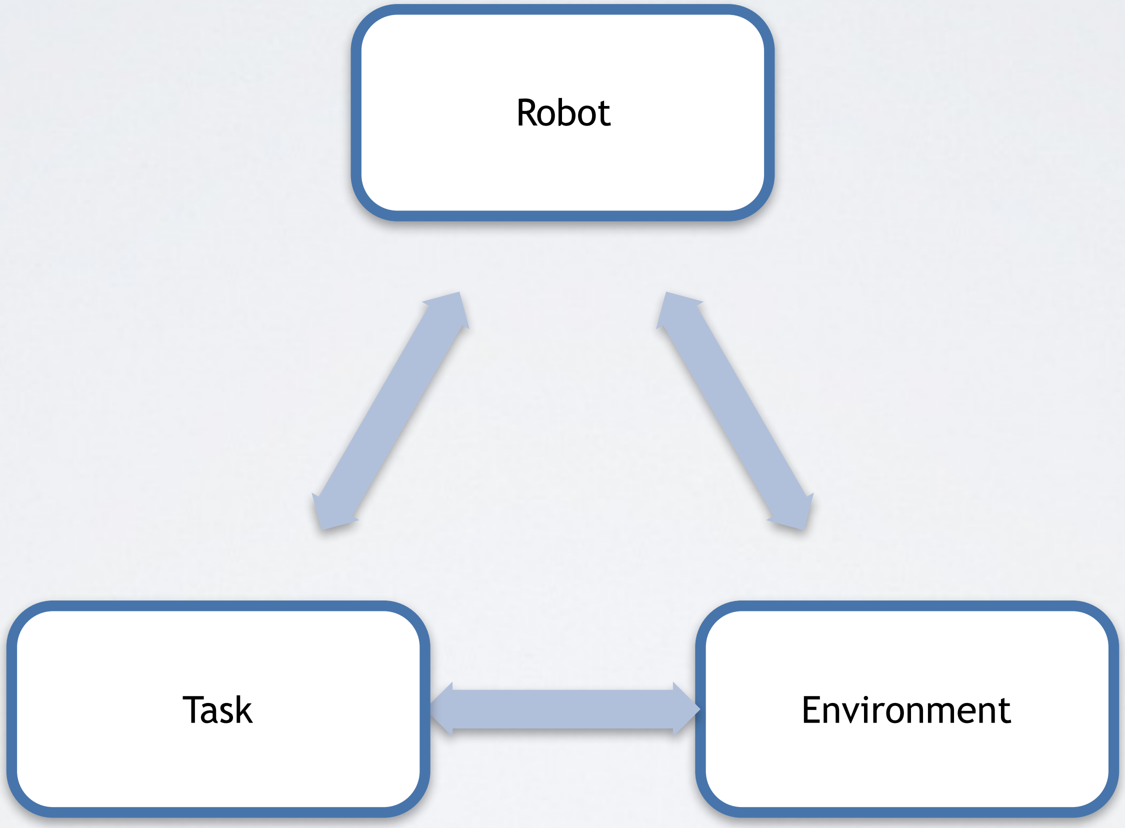


Overview/*The Robotic Future*

- The robotics industry faces many of the same challenges that the personal computer business faced 30 years ago. Because of a lack of common standards and platforms, designers usually have to start from scratch when building their machines.
- Another challenge is enabling robots to quickly sense and react to their environments. Recent decreases in the cost of processing power and sensors are allowing researchers to tackle these problems.
- Robot builders can also take advantage of new software tools that make it easier to write programs that work with different kinds of hardware. Networks of wireless robots can tap into the power of desktop PCs to handle tasks such as visual recognition and navigation.

ROBOT, TASK, ENVIRONMENT

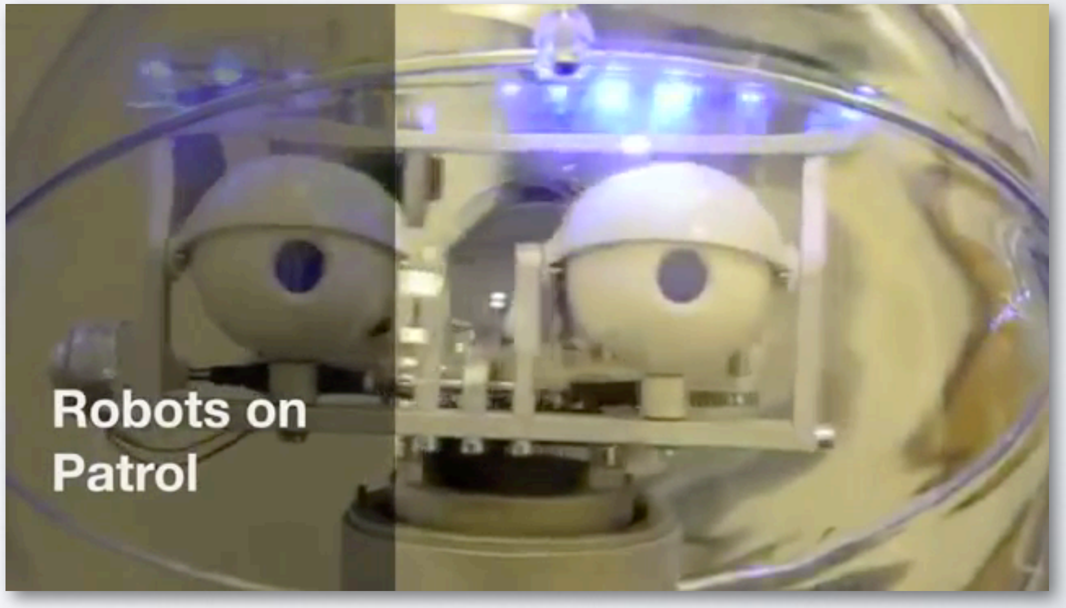
- There is no general purpose robot



- These three aspects are dependent upon and influence each other

ROBOTICS RESEARCH AT LINCOLN

- Our webpage: lcas.lincoln.ac.uk
  - Human-Centered Robotics
  - Agri-Food Technology
  - Bio-inspired Embedded Systems
  - Learning for Autonomous Systems
- Applications
  - Security
  - Assistive Care
  - Agricultural Robotics
  - Intelligent Transportation
  - Nuclear Robotics



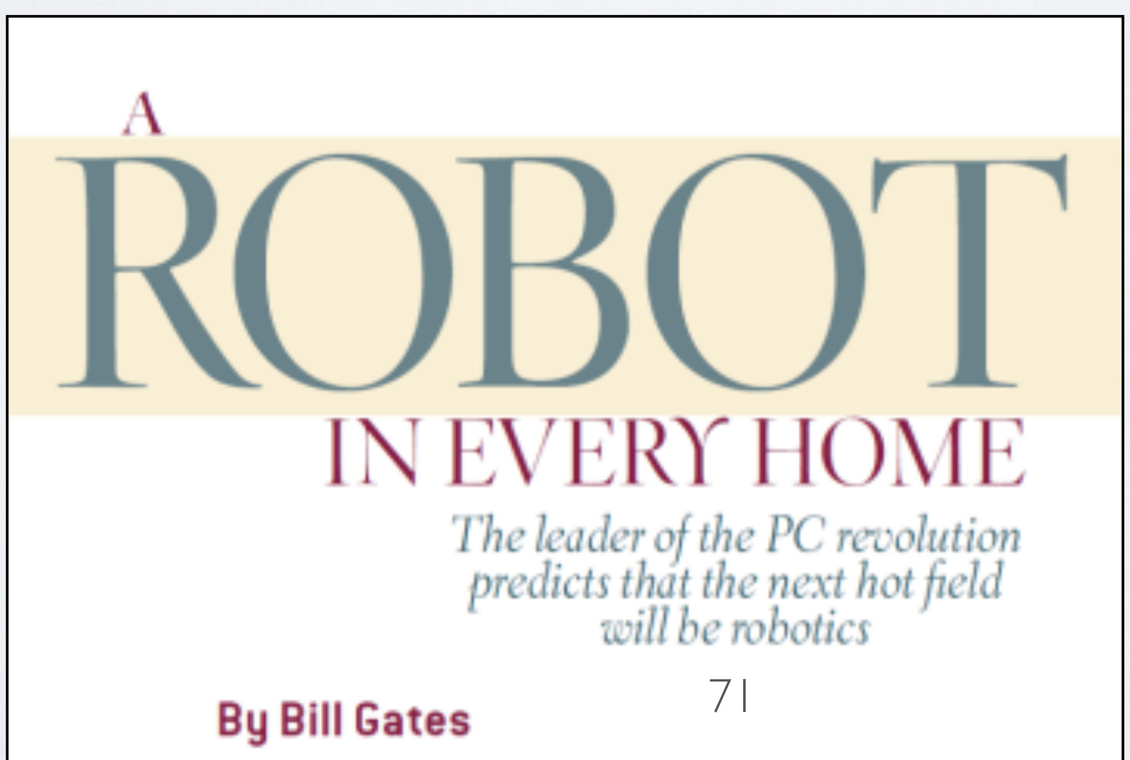
FP7 Project STRANDS

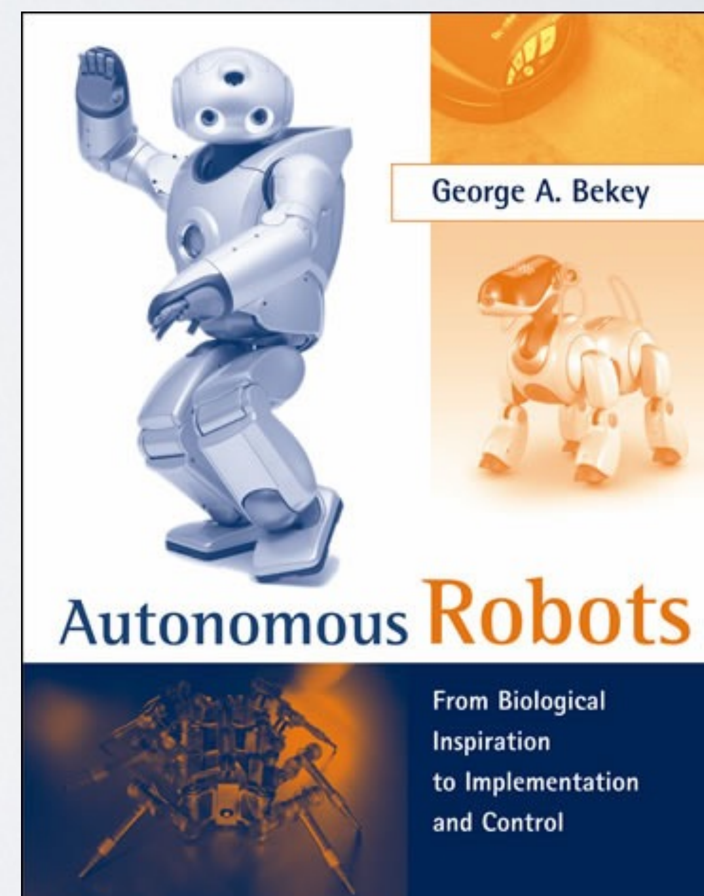
ROBOTICS RESEARCH AT LINCOLN

- Our webpage: lcas.lincoln.ac.uk
  - Human-Centered Robotics
  - Agri-Food Technology
  - Bio-inspired Embedded Systems
  - Learning for Autonomous Systems
- Applications
  - Security
  - Assistive Care
  - Agricultural Robotics
  - Intelligent Transportation
  - Nuclear Robotics

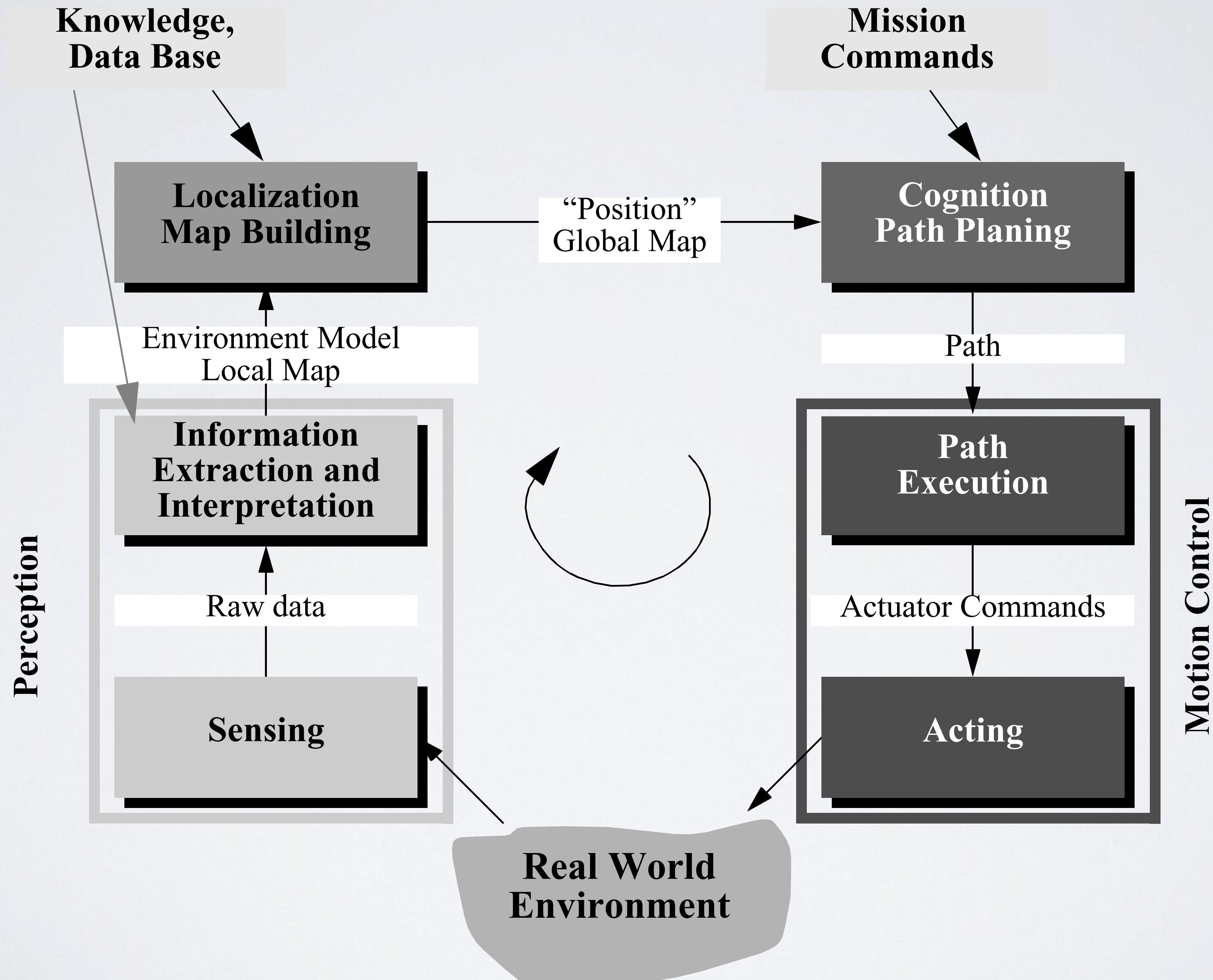
RECOMMENDED READING

- Gates, B. *A Robot in Every Home*, Scientific American, 2006
- Siegwart et al. Autonomous Mobile Robots, 2004 (chapter 1, also on blackboard)
- Bekey, G.A. Autonomous robots – Chapter 1, also Section 4.1





SUMMARY



Knowledge, Data Base
Mission Commands
Localization Map Building
"Position" Global Map
Cognition Path Planing
Environment Model Local Map
Path
Information Extraction and Interpretation
Path Execution
Raw data
Actuator Commands
Sensing
Acting
Perception
Motion Control
Real World Environment

Thank you for listening!
Any questions ?

End of Lecture Feedback

When survey is active, respond at **PollEv.com/mhanheide**

0 surveys done

1 survey underway