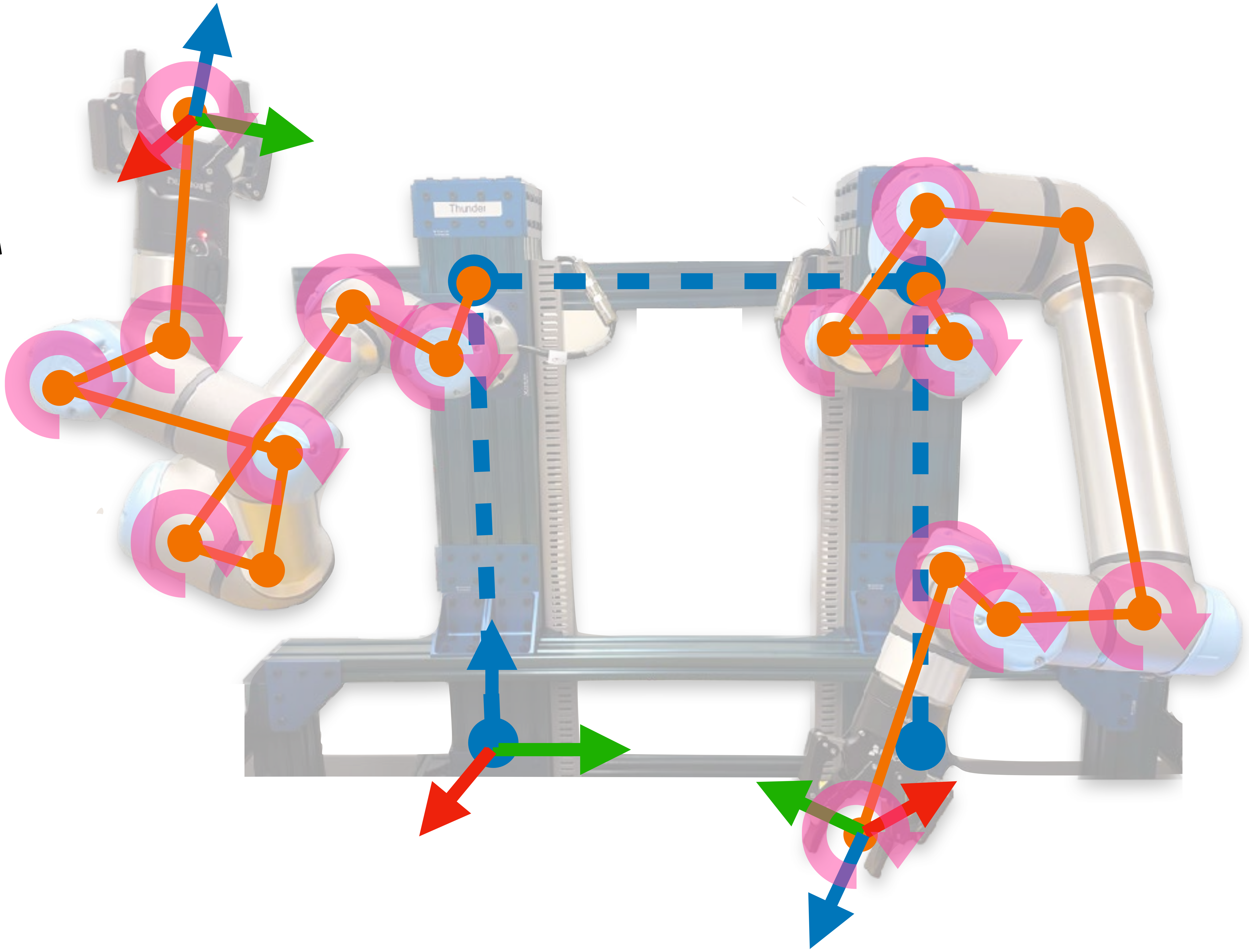
Intro to Intelligent Robotic Systems



CSCI 5551

Spring 2024

University of Minnesota

Welcome to 5551!

Section - 001, 881, 883

Course Staff

- **Instructor: Karthik Desingh *(he/him)***
  - Assistant Professor, CS&E | MnRI
  - kdesingh@umn.edu
  - OH: Mondays 8:30-10:00 am CT Shepherd 2-234

- **TA: Chahyon Ku *(he/him)***
  - MS Student in Robotics
  - ku000045@umn.edu
  - OH:
    - Thursdays 3:00-4:00 PM CT
    - Fridays 9:00-10:00 AM CT at Keller 2-209

- **TA: Xun Tu *(he/him)***
  - PhD Student in CS
  - tu000080@umn.edu
  - OH: Tuesdays and Thursdays 3:00-4:00 PM CT at Keller 2-209

Acknowledgement

- This course builds on and is indebted to materials from:
    - Prof. Chad Jenkins (University of Michigan) and the staff of autorob.org
    - Prof. Dieter Fox (Univ of Washington),
    - Prof. Cyrill Stachniss (Univ of Bonn),
    - Prof. Nikolaos Papanikolopoulos (University of Minnesota),
    - Prof. Junaed Sattar (University of Minnesota)

What are intelligent robotic systems?

What are intelligent robotic systems?

"systems that provide intelligent services and information by
interacting with their environment, including human beings,
via the use of various sensors, actuators and human interfaces"

What are intelligent robotic systems?

"systems that provide intelligent services and information by interacting with their environment, including human beings, via the use of various sensors, actuators and human interfaces"

What are intelligent robotic systems?

It is getting very hard to define this term.
For the sake of this course,
let us call this **“ability to operate with some autonomy”**

What are intelligent robotic systems?

Fully Teleoperational

Fully Autonomous

Spectrum of Shared autonomy

What are intelligent robotic systems?

Fully Teleoperational

Fully Autonomous

Spectrum of Shared autonomy

HaptX: https://www.youtube.com/watch?v=uwYtwQtoOh0

What are intelligent robotic systems?

Fully Teleoperational ⟵ ⟶ **Fully Autonomous**

Spectrum of Shared autonomy

HaptX: https://www.youtube.com/watch?v=uwYtwQtoOh0

TUM/IAS group: https://www.youtube.com/watch?v=cTCJSNjTdo0

What are intelligent robotic systems?

…systems that can perform Sense-Plan-Act….

What are intelligent robotic systems?

…systems that can perform Sense-Plan-Act….

Zhiqiang Sui et al. 2017

What are intelligent robotic systems?

…systems that can perform Sense-Plan-Act….

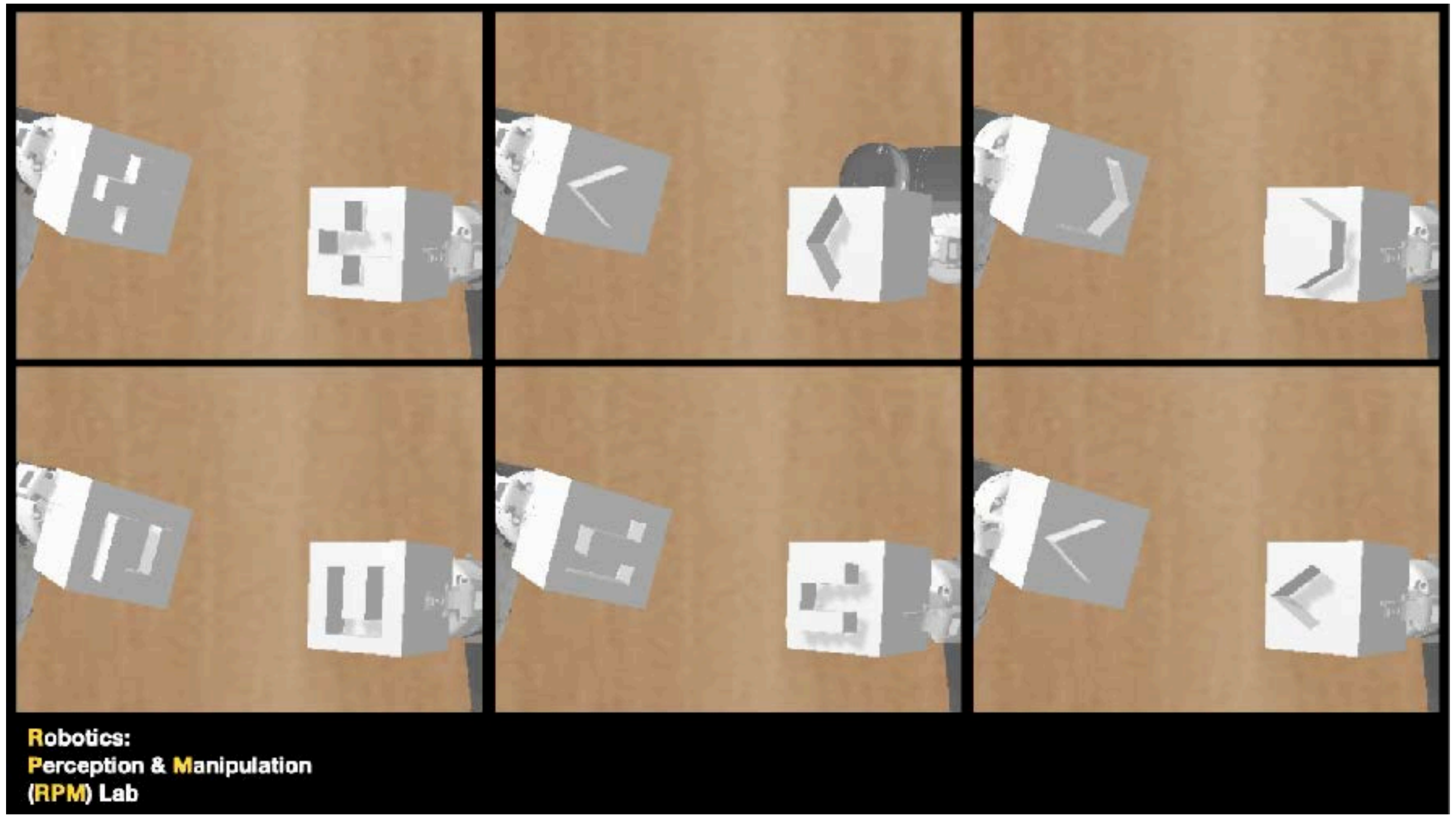
Zhiqiang Sui et al. 2017

What are intelligent robotic systems?

…systems that can perform Sense-Plan-Act….

… can also learn skills … transfer these skills … adapt to new environments …

Carl Winge et al. 2022



Chahyon Ku et al. 2023

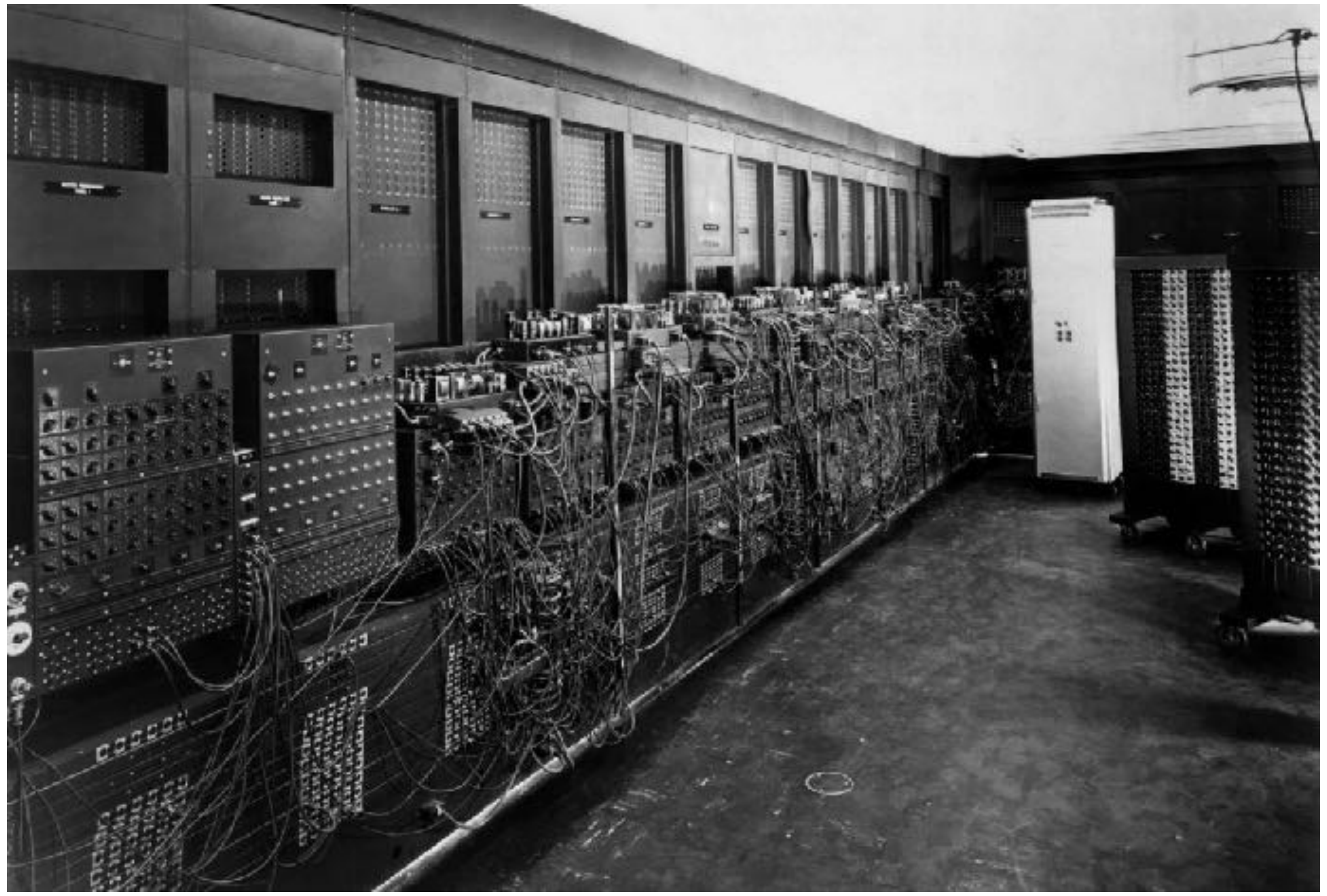
What are intelligent robotic systems?

…systems that can perform Sense-Plan-Act….

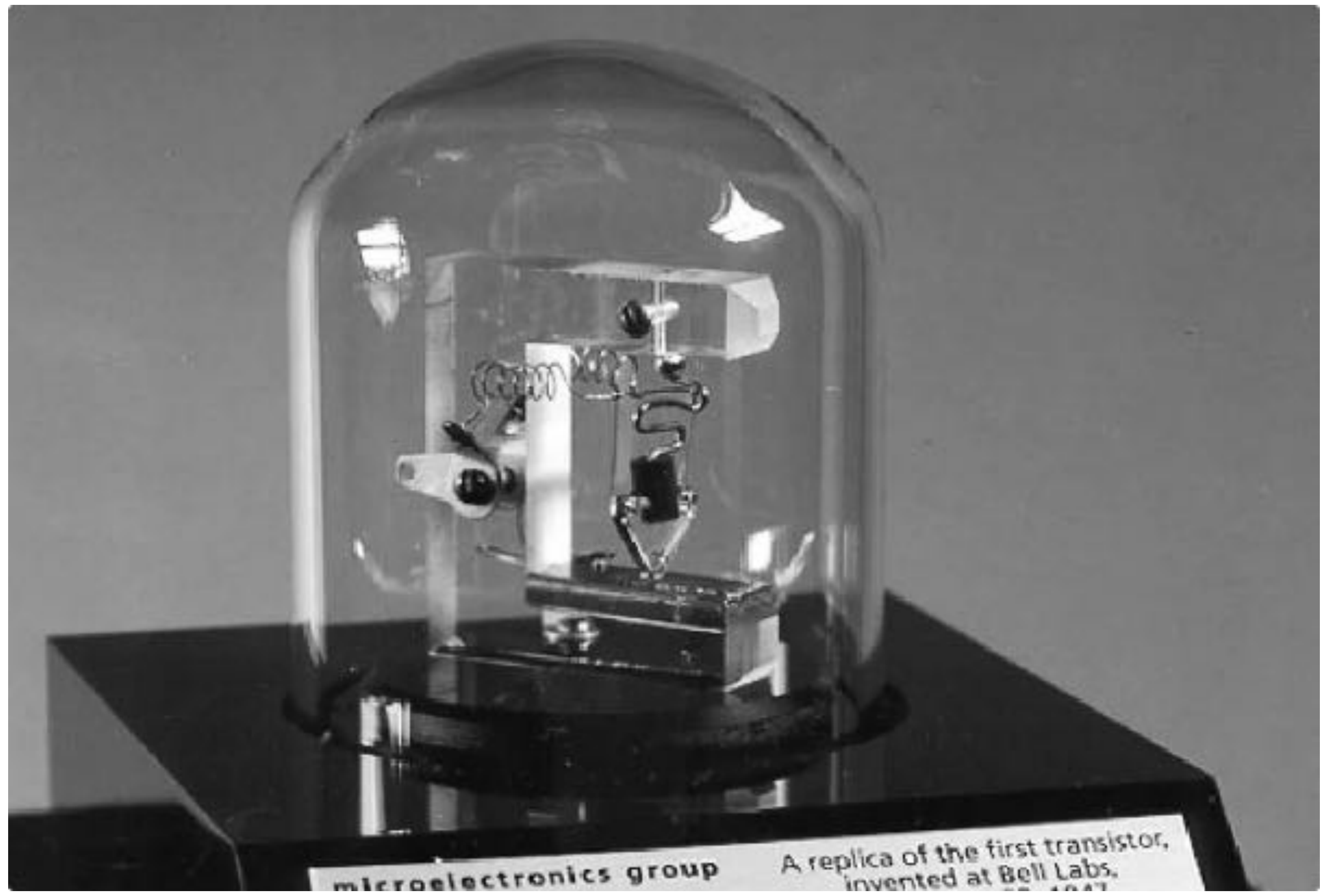
… can also learn skills … transfer these skills … adapt to new environments …

Bahaa Aldeeb et al. 2023

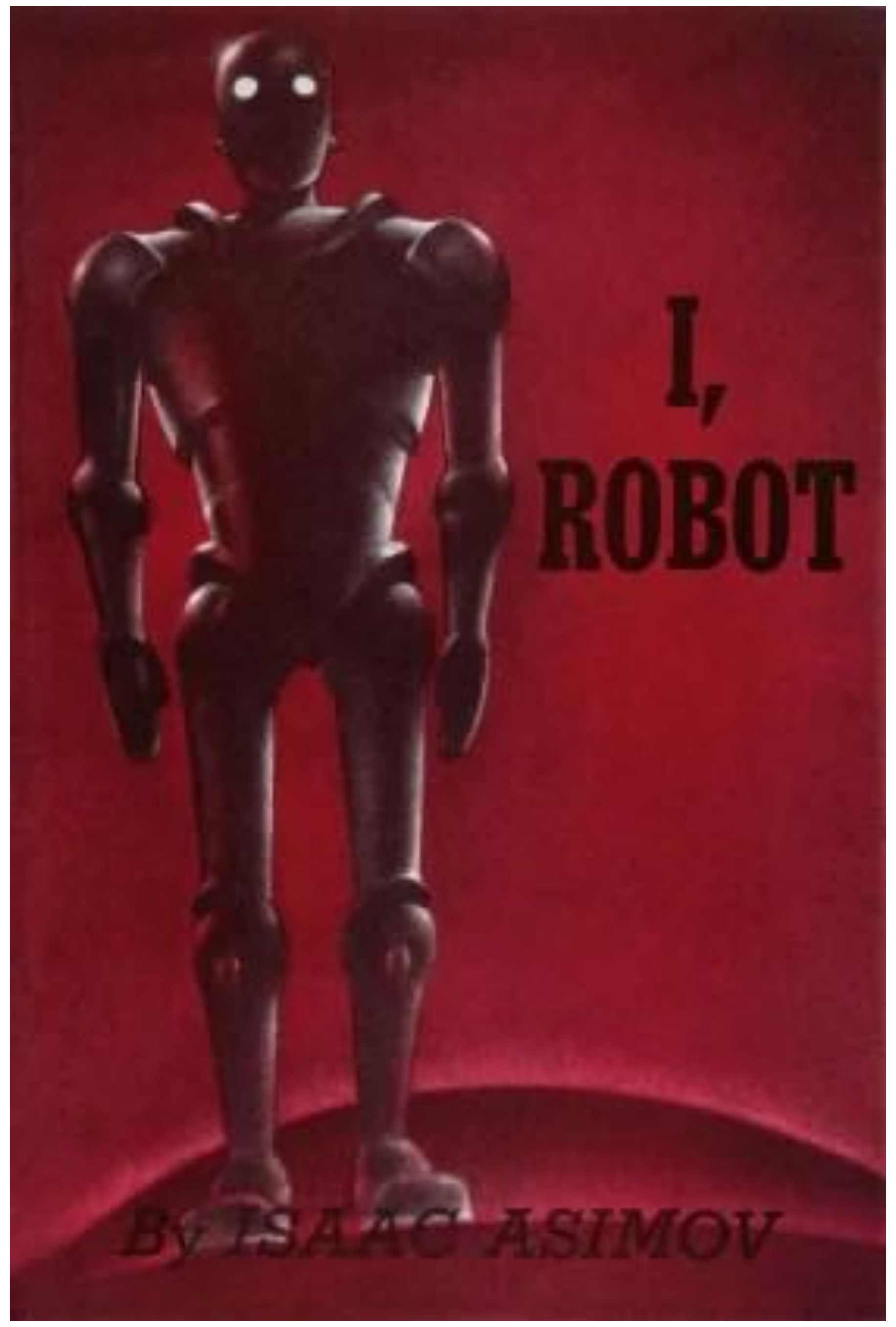
History of Computers, AI and Robotics



Pennsylvania University professors
John Mauchly and J. Presper Eckert build
the ‘grandfather’ of digital computers, the
Electronic Numerical Integrator and Calculator (ENIAC)



Researchers William Shockley, John Bardeen
and Walter Brattain
at Bell Laboratories invent the transistor.



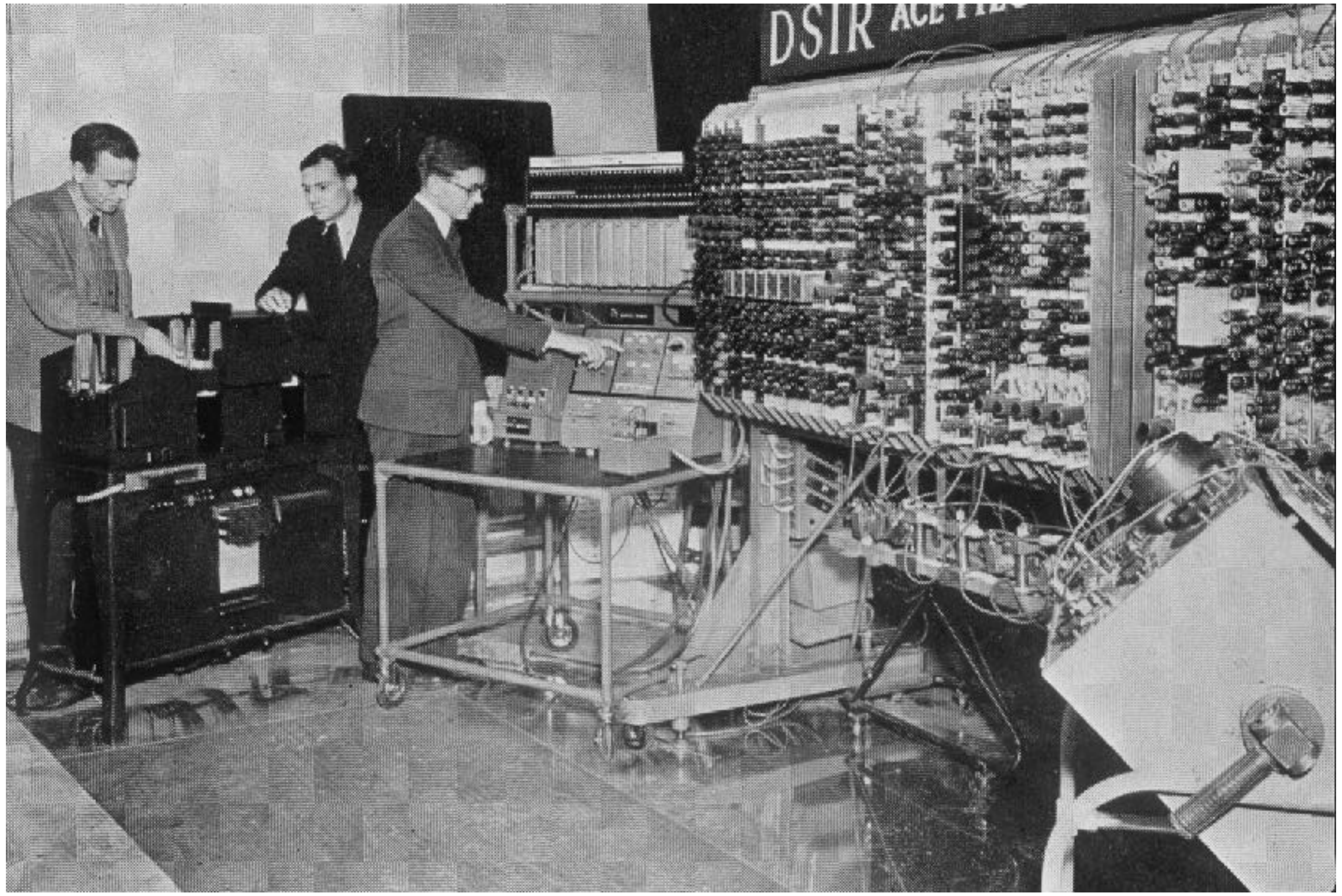
‘I,Robot’ by Issac Asimov is
published, laying the foundations
for the idea of robots in culture.

1944

1948

1950

https://everydayrobots.com/



Alan Turing introduces ‘The Turing Test’ —
a test of a machine's ability to exhibit intelligent behavior
equivalent to, or indistinguishable from, that of a human.

Grace Hopper develops COBOL, the first computer language.
The second, FORTRAN, is developed by a team of
IBM programmers a year later.

1950

1953

https://hotcorn-cdn.s3.amazonaws.com/wp-content/uploads/sites/2/2020/09/22125112/bletchleypark-pilotace-scaled.jpg
https://everydayrobots.com/

Dartmouth conference coins the term 'artificial intelligence' and launches the field of AI.
IBM mainframes are used in early experiments

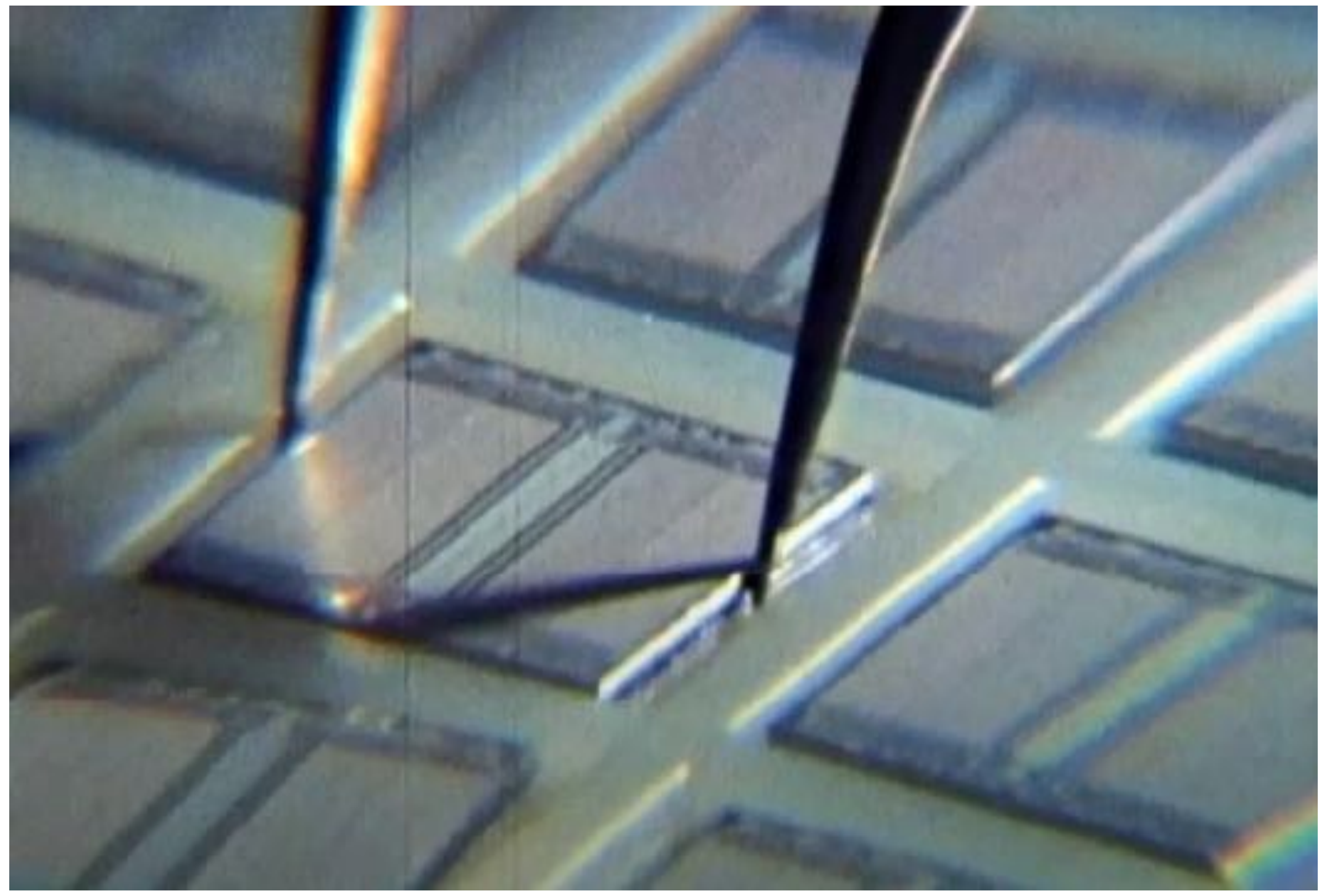
Five of the attendees of the 1956 Dartmouth Summer Research Project on Artificial Intelligence reunited at the July AI@50 conference. From left: Trenchard More, John McCarthy, Marvin Minsky, Oliver Selfridge, and Ray Solomonoff. (Photo by Joseph Mehling '69)

IBM's chairman and CEO, Thomas J. Watson Jr., bets the company's future on the IBM Series/360 — the largest privately-financed commercial project in history. The risk pays off, changing the computer industry forever. Work is revolutionized, productivity is enhanced and countless new tasks become possible.

1956

1964

https://everydayrobots.com/



Intel and Ted Hoff introduce the first microprocessor, the Intel 4004.

Intel co-founder, Gordon Moore, theorizes that computing would dramatically increase in power, and decrease in relative cost, at an exponential pace. The insight, known as Moore's Law, becomes the golden rule for the electronics industry, and a springboard for innovation.

Steve Wozniak and Steve Jobs release the Apple 1

A year later, Apple releases the Apple II

1971

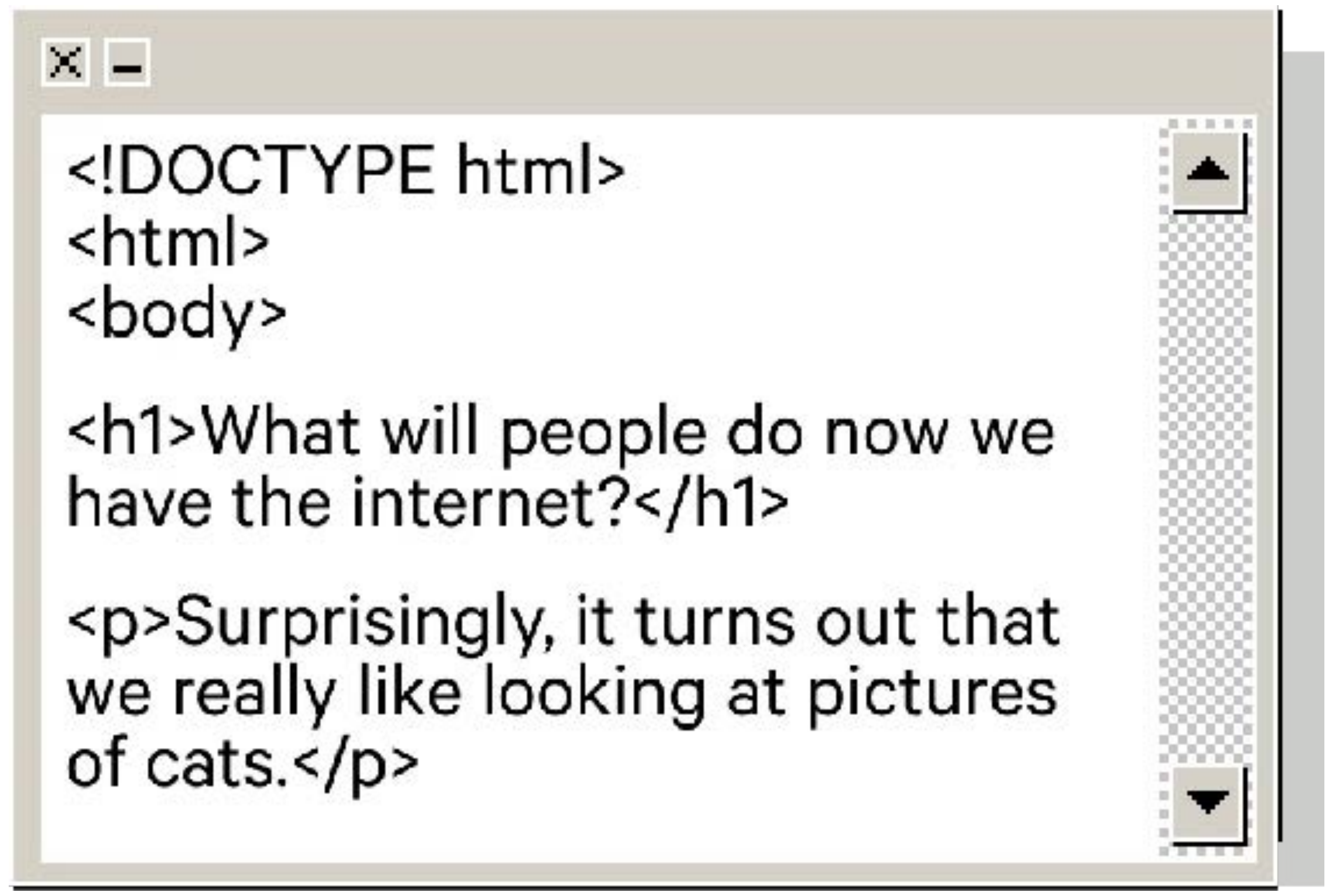
1976

1977

https://image.cnbcfm.com/api/v1/image/100932798-128279719-1.jpg?v=1583960525

https://everydayrobots.com/

The U.S. Defense Department funds the first experimental computer network— ARPANET. It connects computers everywhere, and is a forerunner to the internet.



Tim Berners-Lee and his colleagues at CERN develop hypertext markup language (HTML) and the uniform resource locator (URL), giving birth to the first incarnation of the World Wide Web.

IBM's Deep Blue Computer, a form of AI, beats reigning world chess champion Gary Kasparov.

1966

1995

1997

https://i.insider.com/55947fbf2acae7b7188b5388?width=750&format=jpeg&auto=webp
https://everydayrobots.com/

Larry Page & Sergey Brin, two computer science graduate students from Stanford University, pioneer a new way to search for and find information on the web. They call their invention ‘Google’.

Four founders start a company called ‘Android

1998

2003

https://i.insider.com/55947fbf2acae7b7188b5388?width=750&format=jpeg&auto=webp
https://indonesiamendesain.com/wp-content/uploads/2020/06/original-google-logo-font.jpg
https://everydayrobots.com/

Steve Jobs unveils the iPhone at Macworld

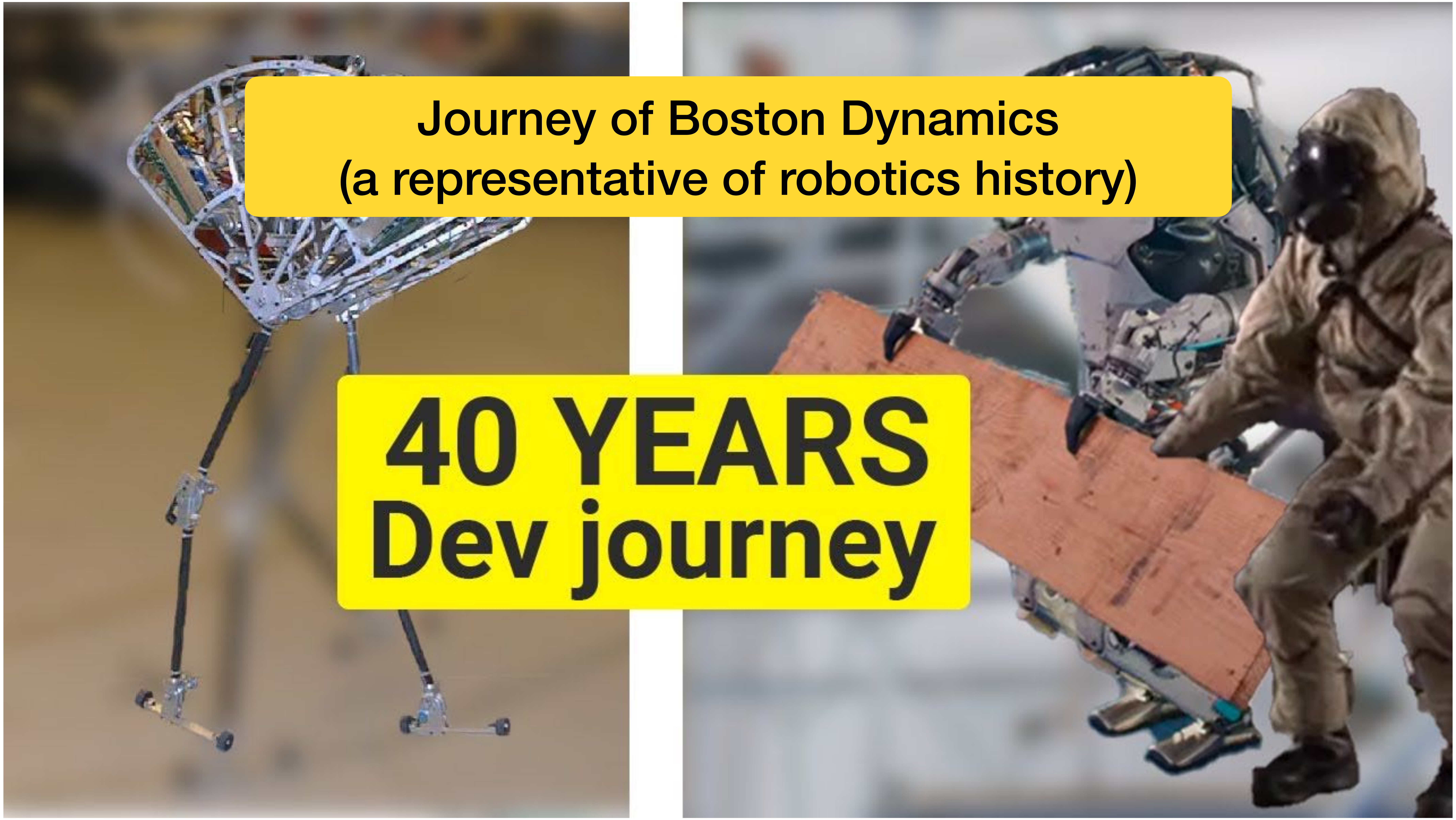
Search becomes intuitive.
Maps are intelligent.
Work is more productive than ever

2007

https://i.insider.com/587374fadd0895e1148b47e7?width=1136&format=jpeg
https://everydayrobots.com/

So, this is computing and AI…
What was happening in robotics?

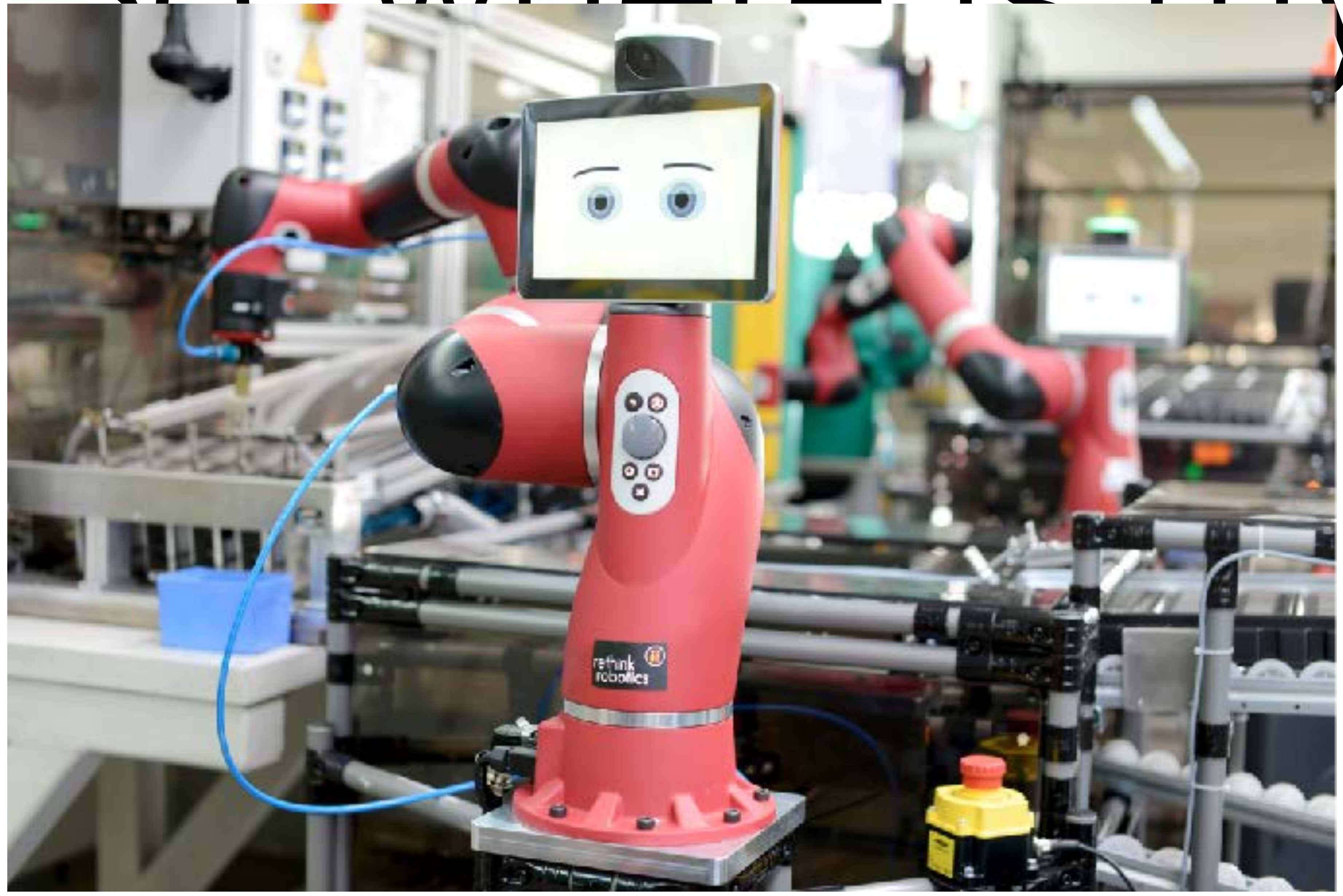
Journey of Boston Dynamics
(a representative of robotics history)



https://youtu.be/qtos3ezujn4?feature=shared

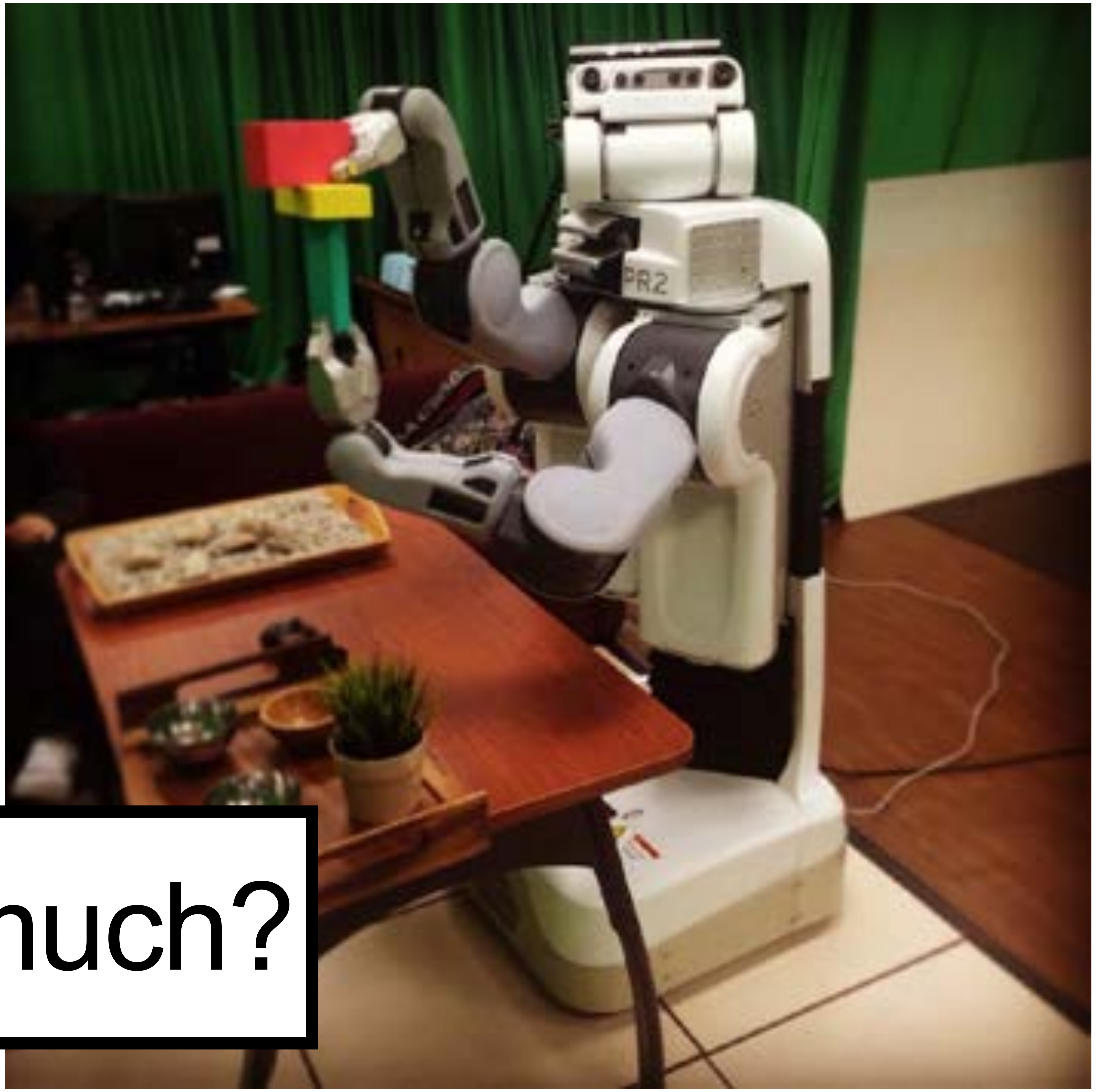
So, where is my robot?

So, where is my robot?



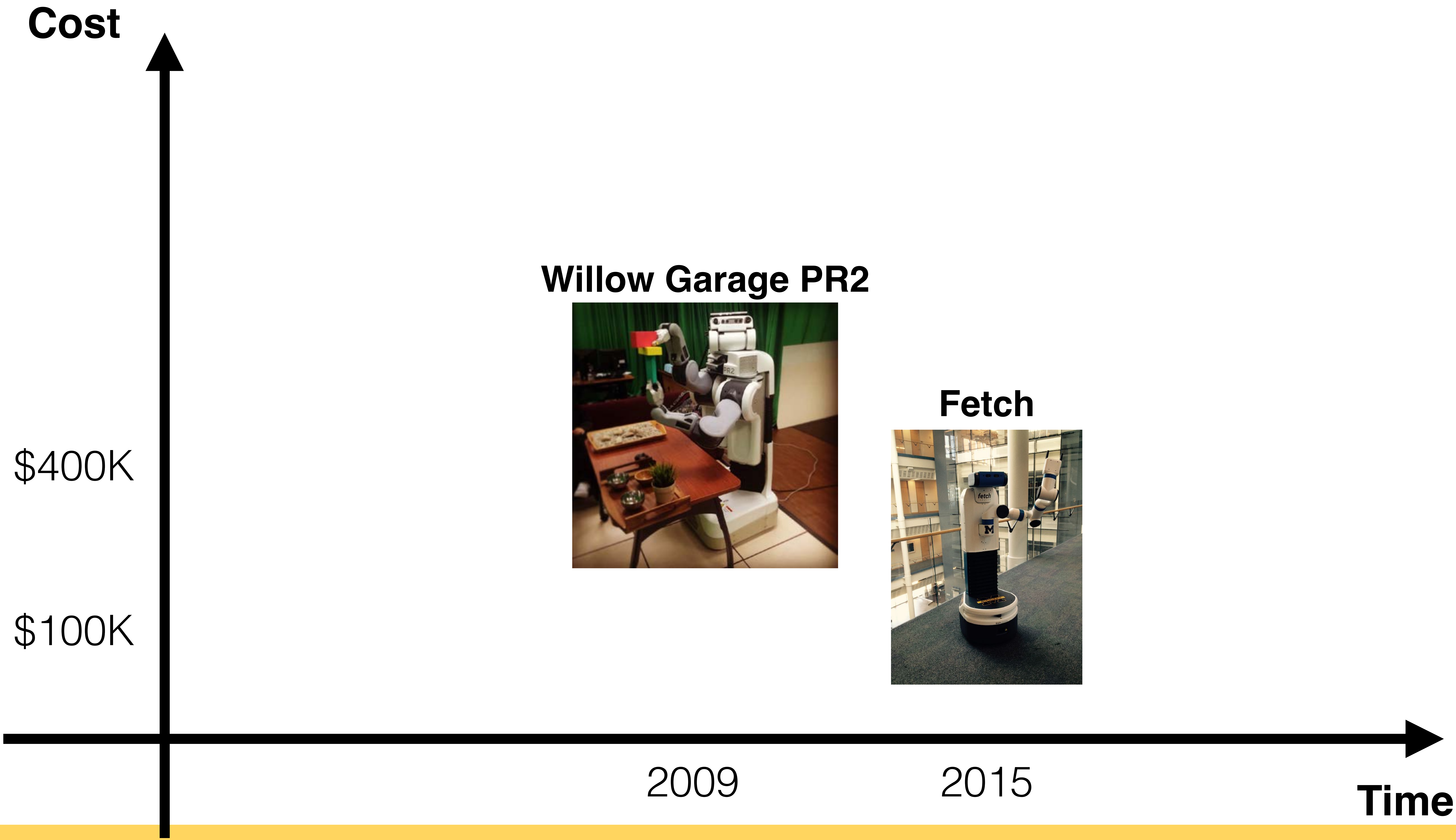
Slide borrowed from Michigan Robotics autorob.org

Mobile Manipulation Robots



How much?

Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

2002

Willow Garage PR2

Fetch

$400K

$100K

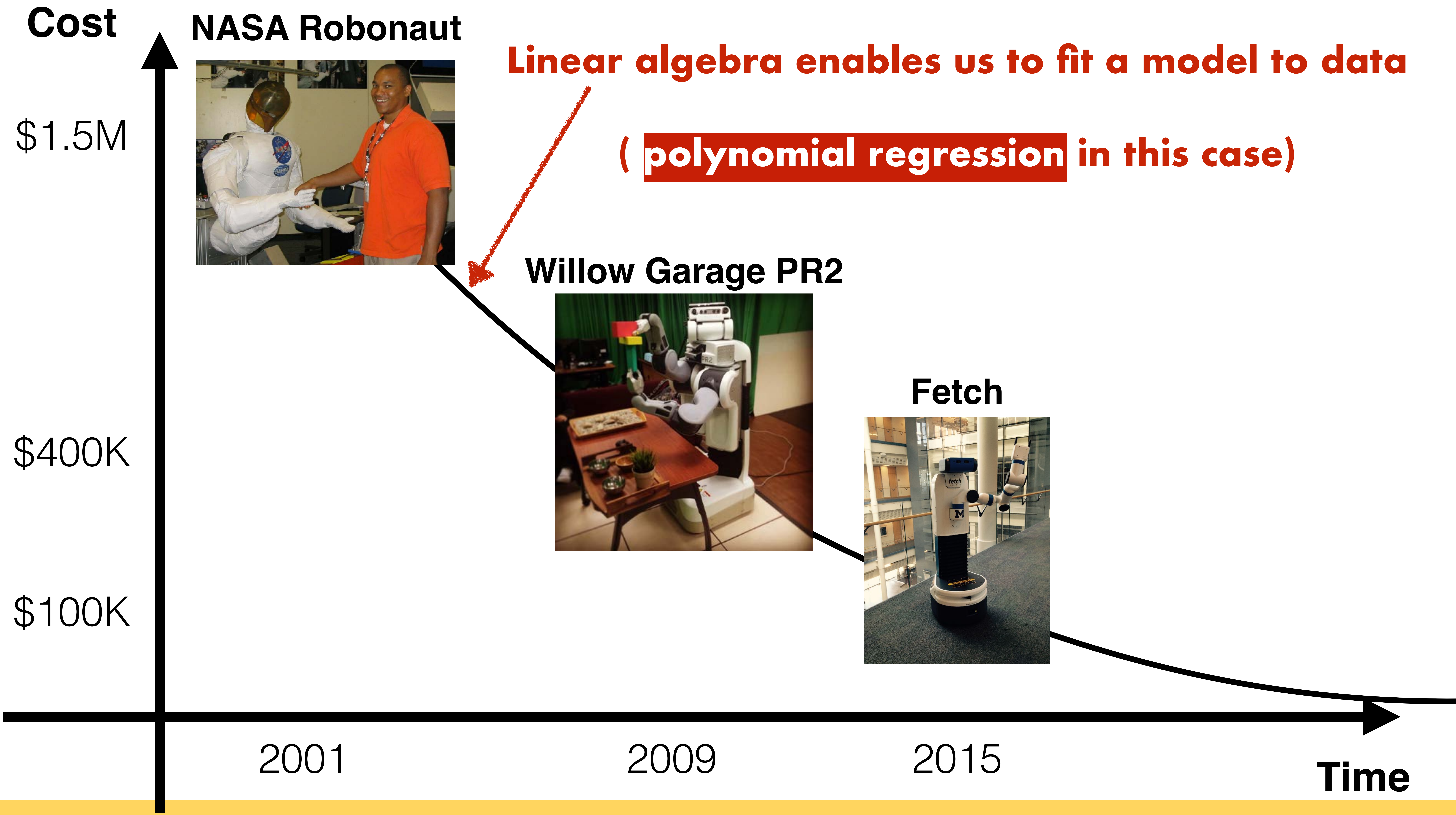
2009

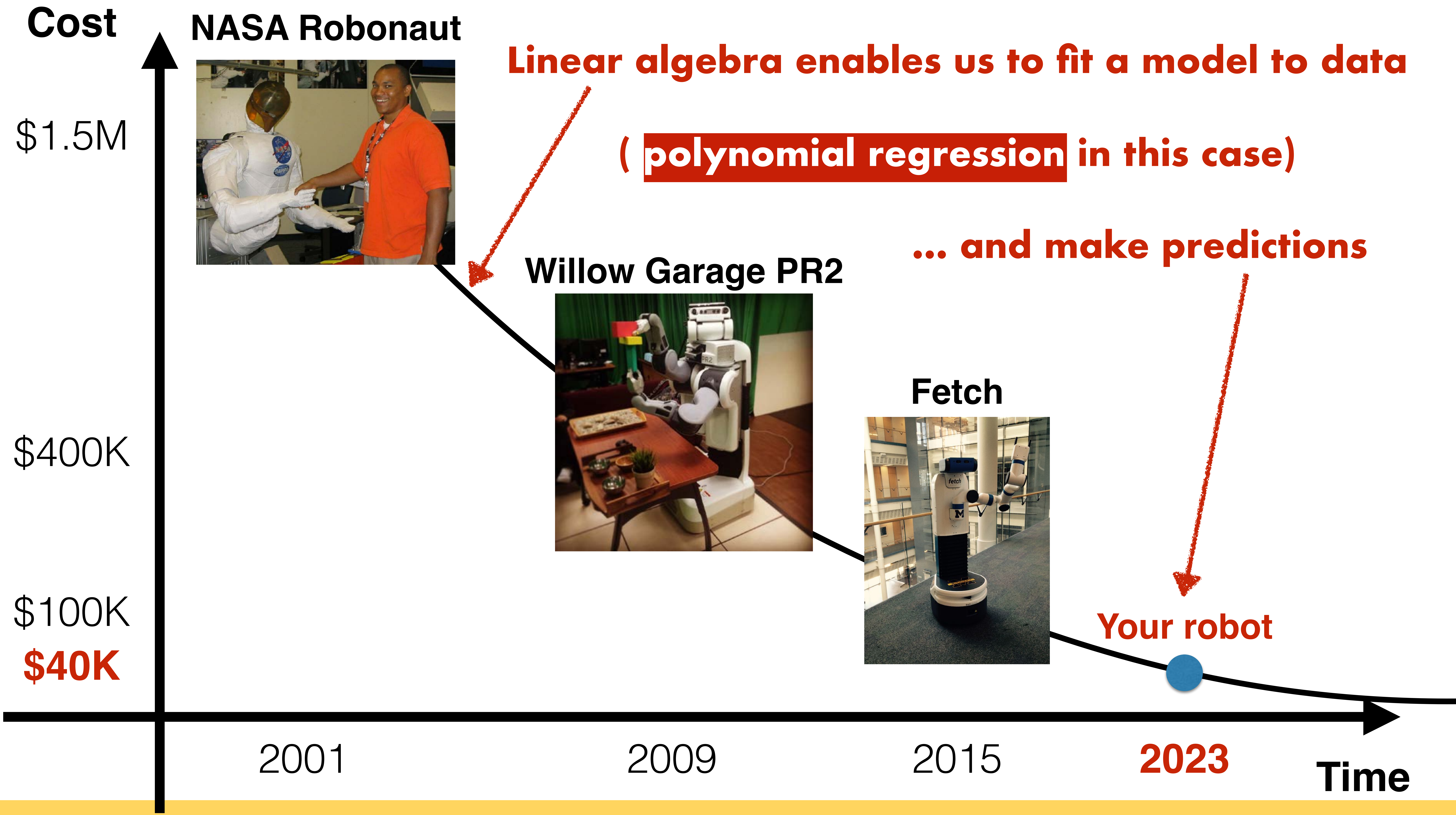
2015

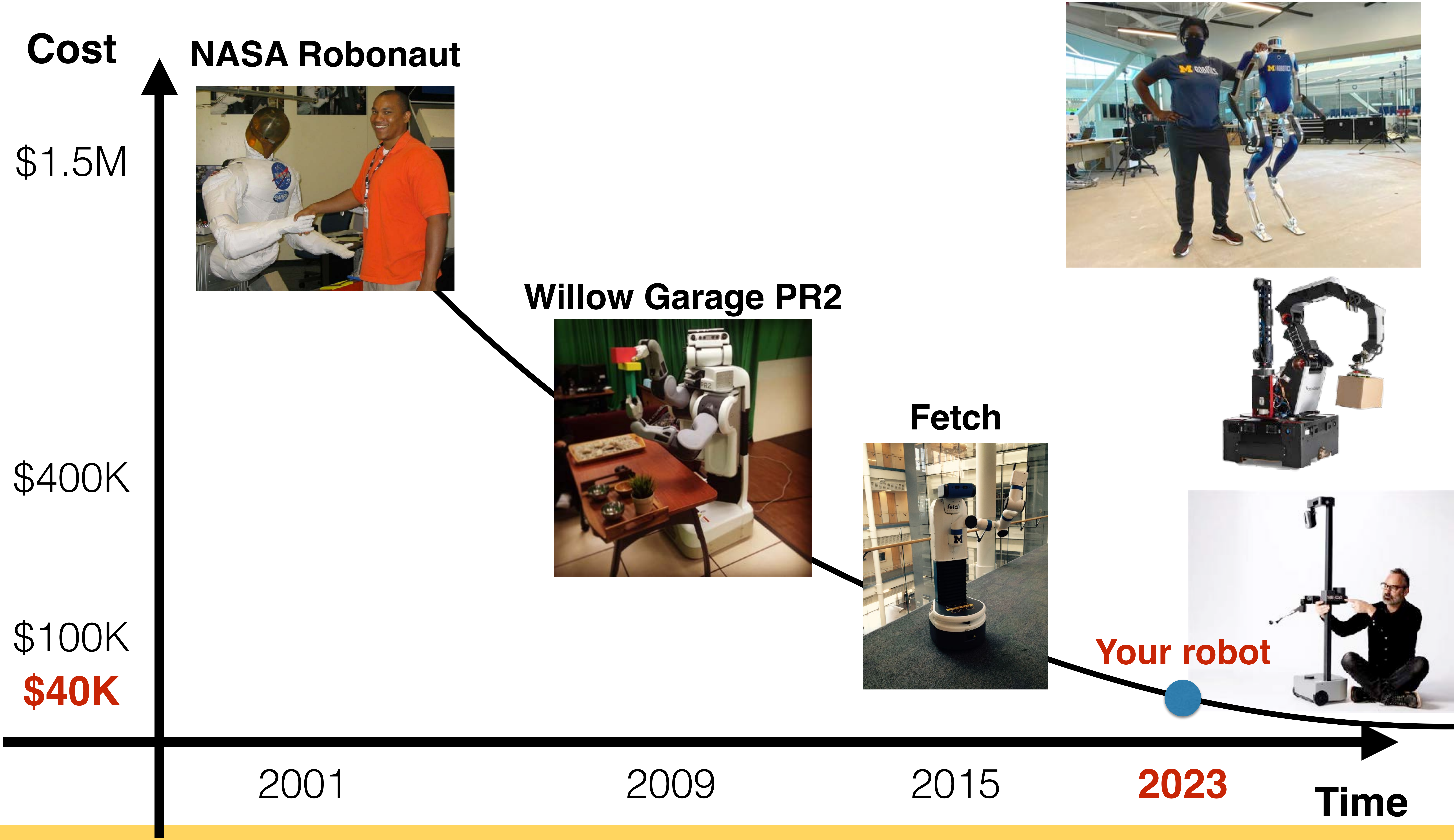
Time

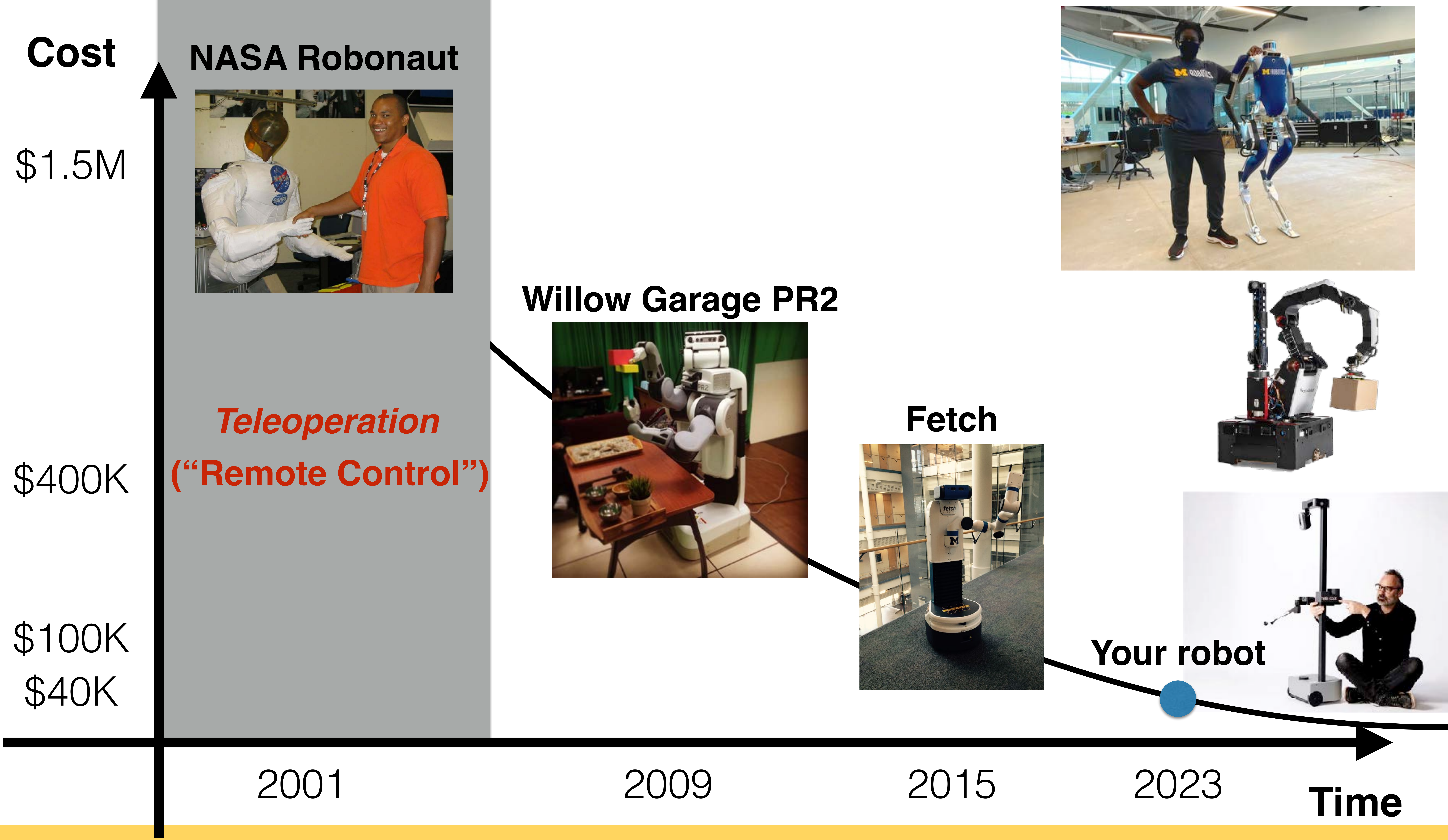
Linear algebra enables us to fit a model to data

(polynomial regression in this case)




Cost
NASA Robonaut
Linear algebra enables us to fit a model to data
(polynomial regression in this case)
... and make predictions
$1.5M
Willow Garage PR2
Fetch
$400K
$100K
$40K
Your robot
2001
2009
2015
2023
Time


Cost
NASA Robonaut
$1.5M
Willow Garage PR2
Fetch
$400K
$100K
$40K
Your robot
2001
2009
2015
2023
Time


Cost
NASA Robonaut
$1.5M
Teleoperation
(“Remote Control”)
$400K
$100K
$40K
Willow Garage PR2
Fetch
Your robot
2001
2009
2015
2023
Time

Cost

$1.5M

$400K

$100K

$40K

NASA Robonaut

Teleoperation

("Remote Control")

2001 2009 2015 2023

Time

Cost

$1.5M

$400K

$100K

$40K

NASA Robonaut

Teleoperation

(“Remote Control”)

Pick-and-Place

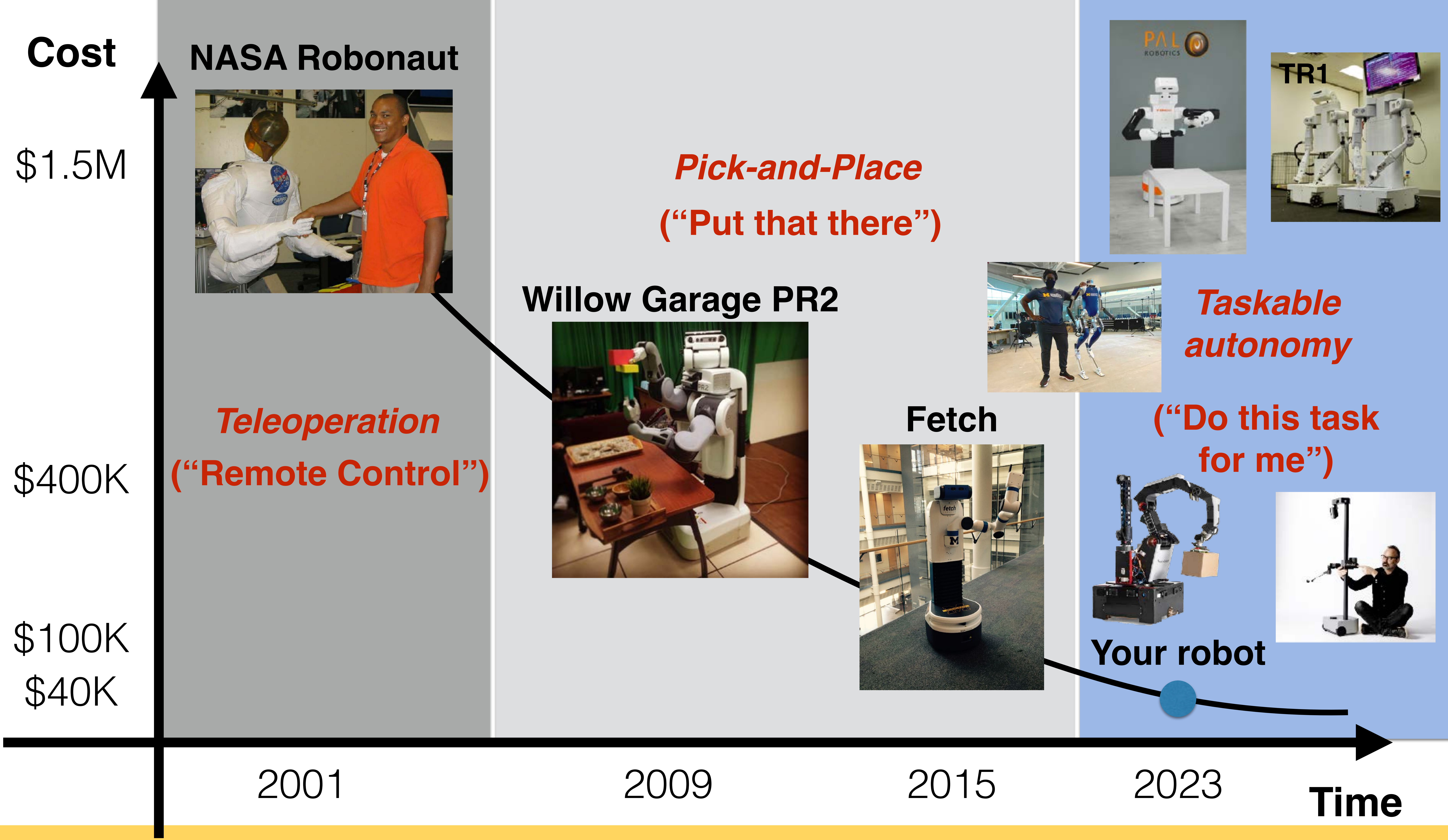
(“Put that there”)

Willow Garage PR2

Fetch

2001 2009 2015 2023

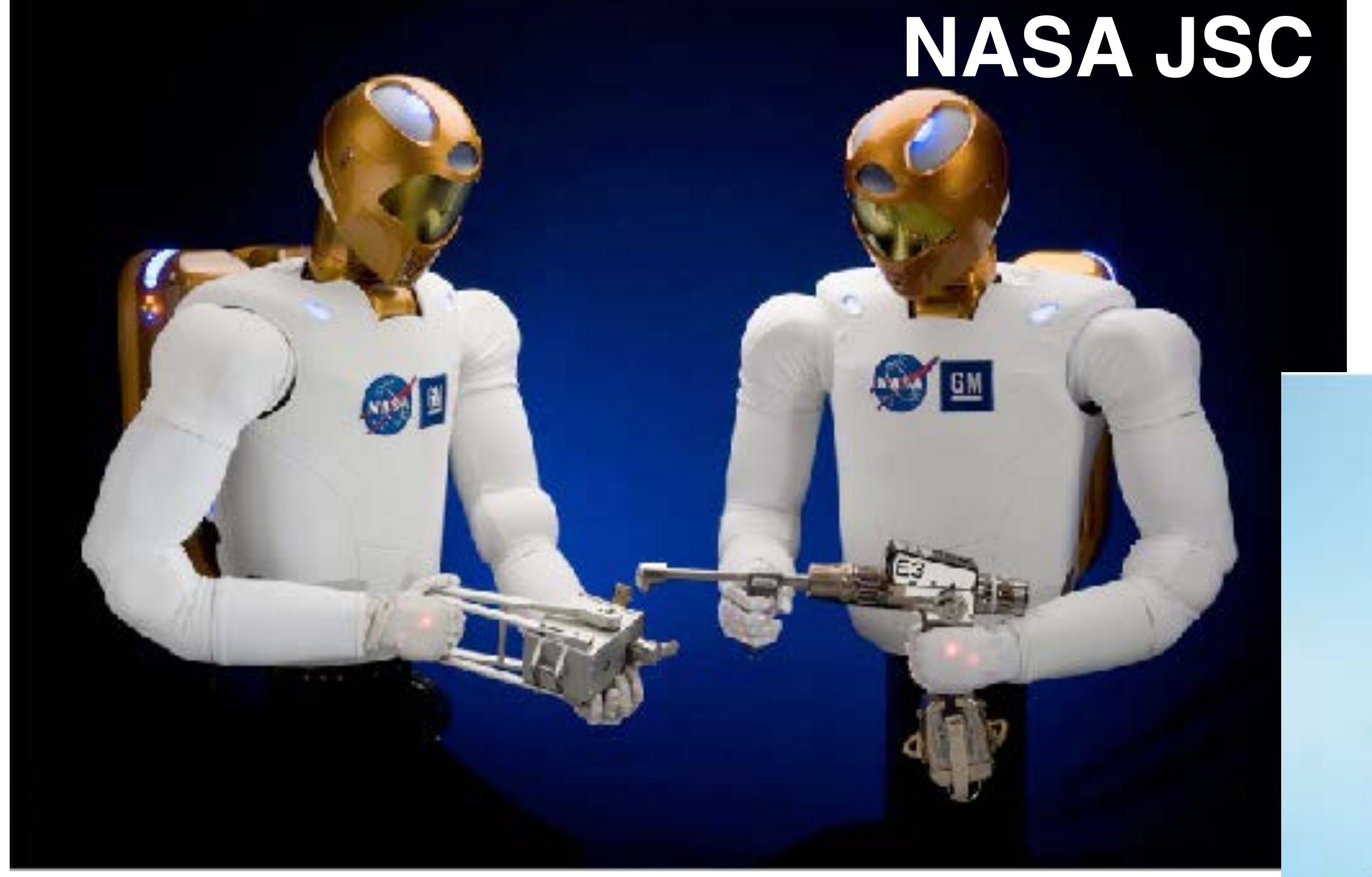
Time


Cost
$1.5M
$400K
$100K
$40K
NASA Robonaut
Teleoperation
("Remote Control")
Willow Garage PR2
Pick-and-Place
("Put that there")
Fetch
PAL ROBOTICS
TR1
Taskable
autonomy
("Do this task
for me")
Your robot
2001
2009
2015
2023
Time

Taskable autonomy

Pick-and-Place

Teleoperation



Dexterous Manipulation

Harvard/Wyss

Taskable autonomy

Dexterous Manipulation

Teleoperation

Taskable autonomy

Dexterous Manipulation

Teleoperation

Operating system

From Wikipedia, the free encyclopedia

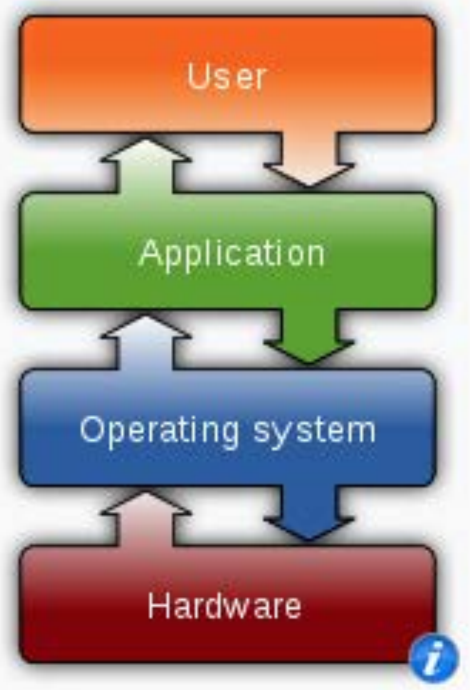
An **operating system** (**OS**) is system software that manages computer hardware, software resources, and provides common services for computer programs.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

The dominant desktop operating system is Microsoft Windows with a market share of around 82.74%. macOS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%).[3] In the mobile sector (including smartphones and tablets), Android's share is up to 70% in the year 2017.[4] According to third quarter 2016 data, Android's share on smartphones is dominant with 87.5 percent with also a growth rate of 10.3 percent per year, followed by Apple's iOS with 12.1 percent with per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent.[5] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.

Operating systems



Common features

Process management · Interrupts · Memory management · File system · Device drivers · Networking · Security · I/O

V · T · E

Contents [hide]

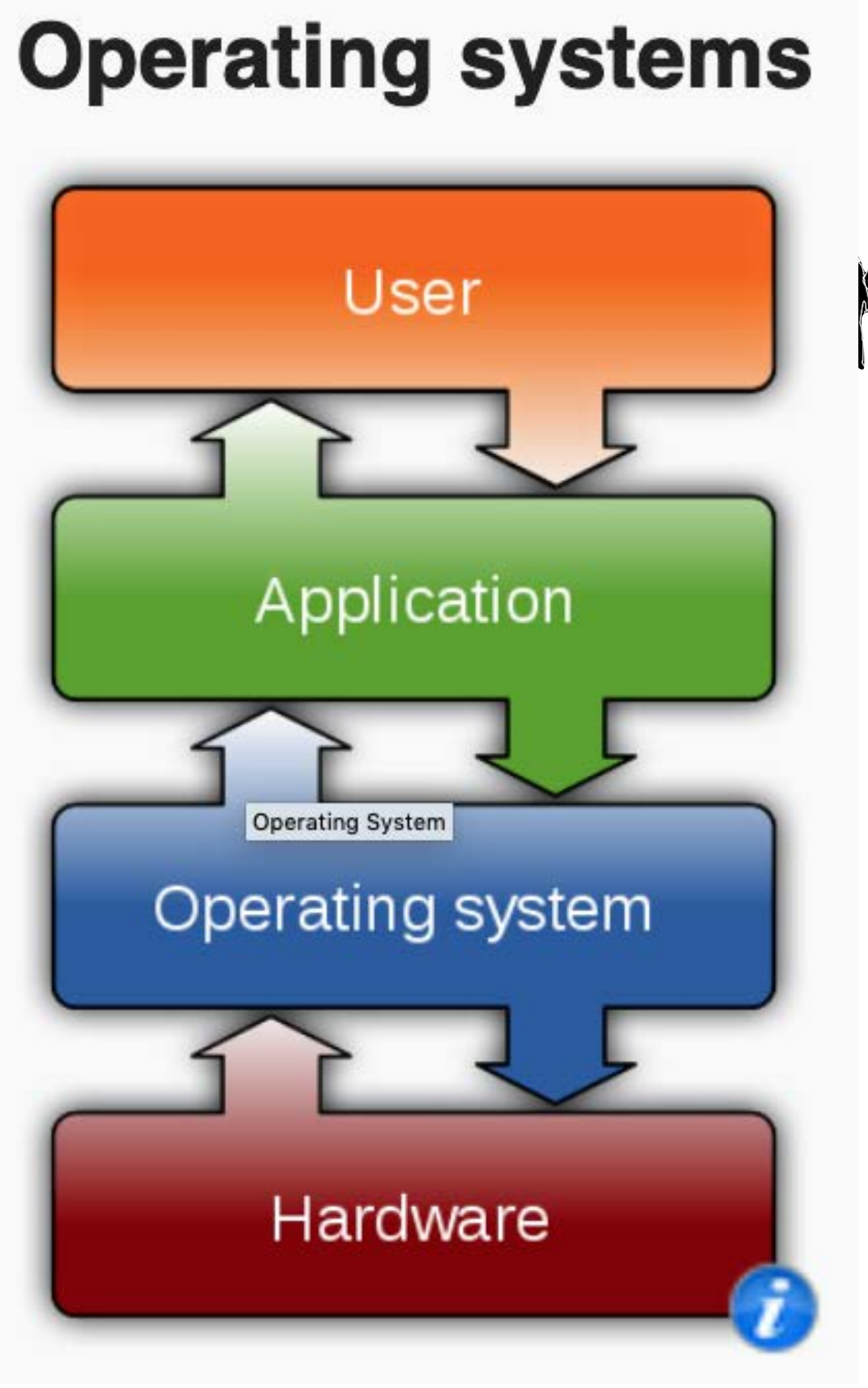
Taskable autonomy

Dexterous Manipulation

Teleoperation

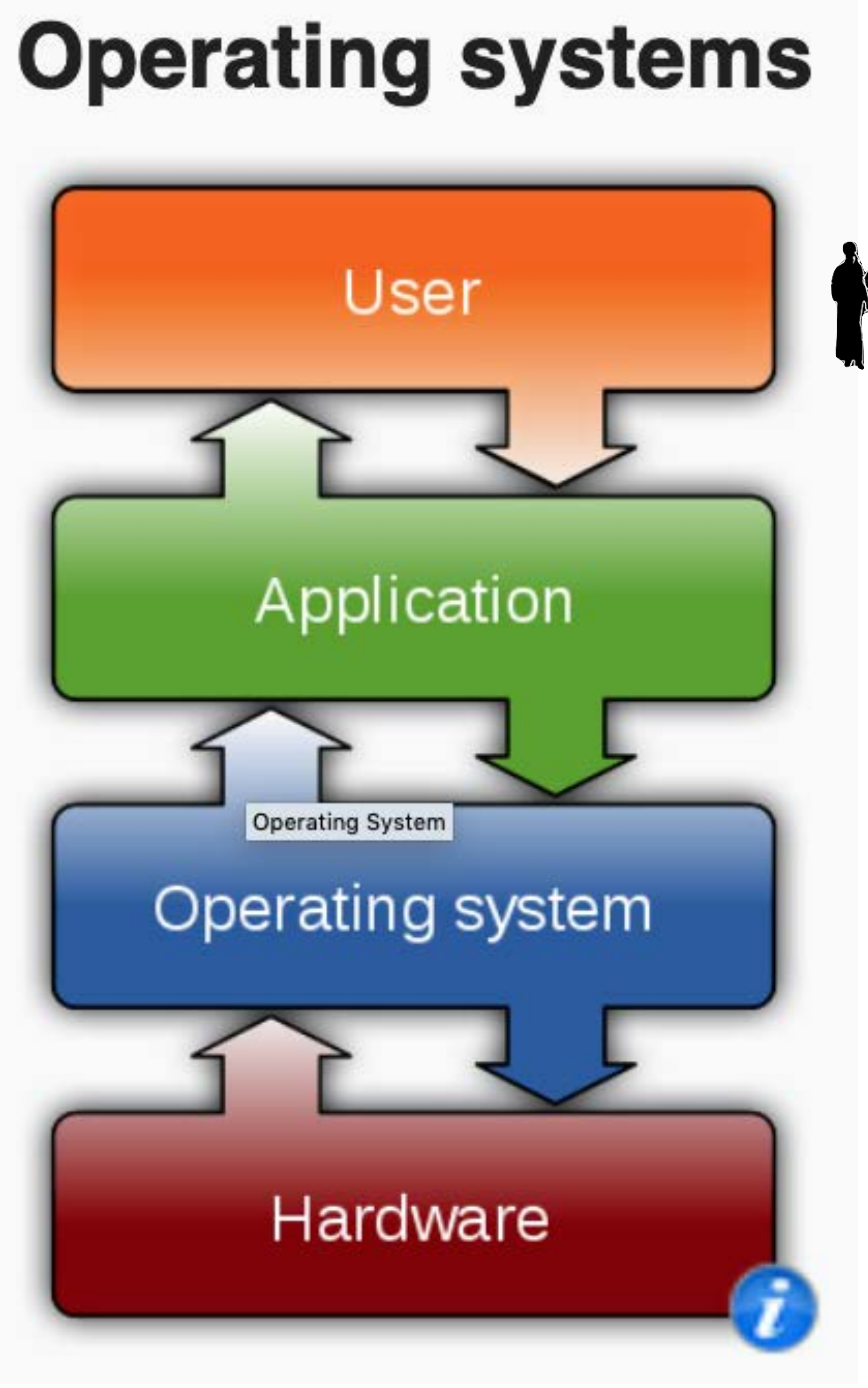
An **operating system (OS)** is
a special program that runs on the bare machine and hides the gory details of managing processes and devices.

- https://perldoc.perl.org/perlglossary.html#operating-system



An **operating system (OS)** is
a special program that runs on the bare machine and hides the gory details of managing processes and devices.

- https://perldoc.perl.org/perlglossary.html#operating-system



Users

Robot Applications

Dexterous Manipulation ← Then, what is this?

Operating System

Hardware

Users

Robot Applications

Robot Operating System

Operating System

Hardware

A **robot operating system (robot OS)** is a special program that runs on the operating system and hides the gory details of controlling robot devices, autonomy processes, and sensorimotor routines.

This abstraction provides a platform for robot applications to run seamlessly across a wide variety of robots capable of mobility and/or dexterous manipulation.

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Slide borrowed from Michigan Robotics autorob.org

Users

Robot Applications

Robot Operating System

LCM ROS Player YARP MOOS 10

Operating System

Hardware

Users

Robot Applications

Robot Operating System

LCM ROS Player YARP MOOS

Operating System

Hardware

Users

Robot Applications

Then, what is this?

Robot Operating System

LCM ROS Player YARP MOOS

Operating System

Hardware

Slide borrowed from Michigan Robotics autorob.org

Users

Robot Applications

Apps of the Future…

"Do this task for me"

Robot Operating System

LCM

ROS

Player

YARP

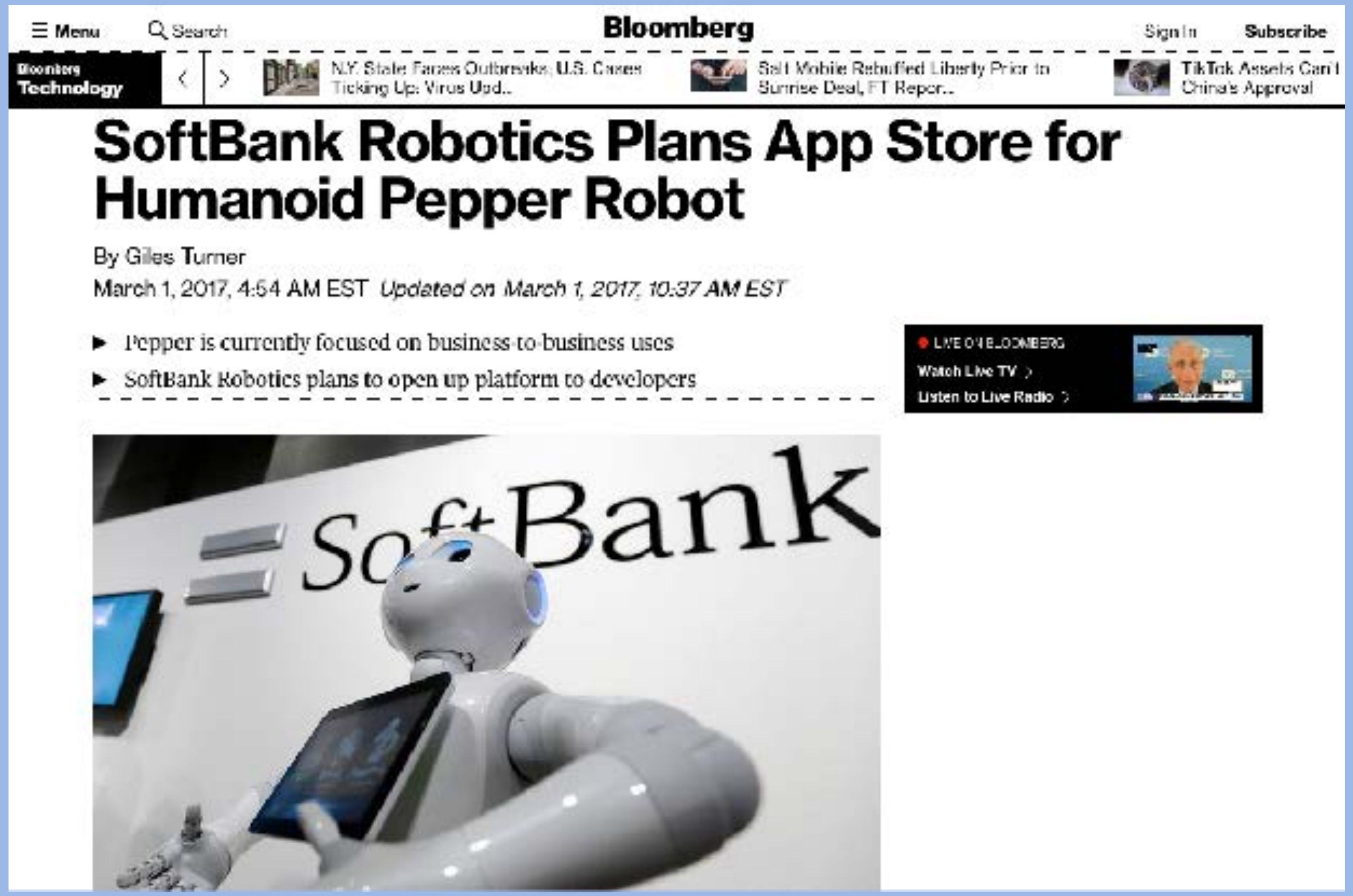
MOOS

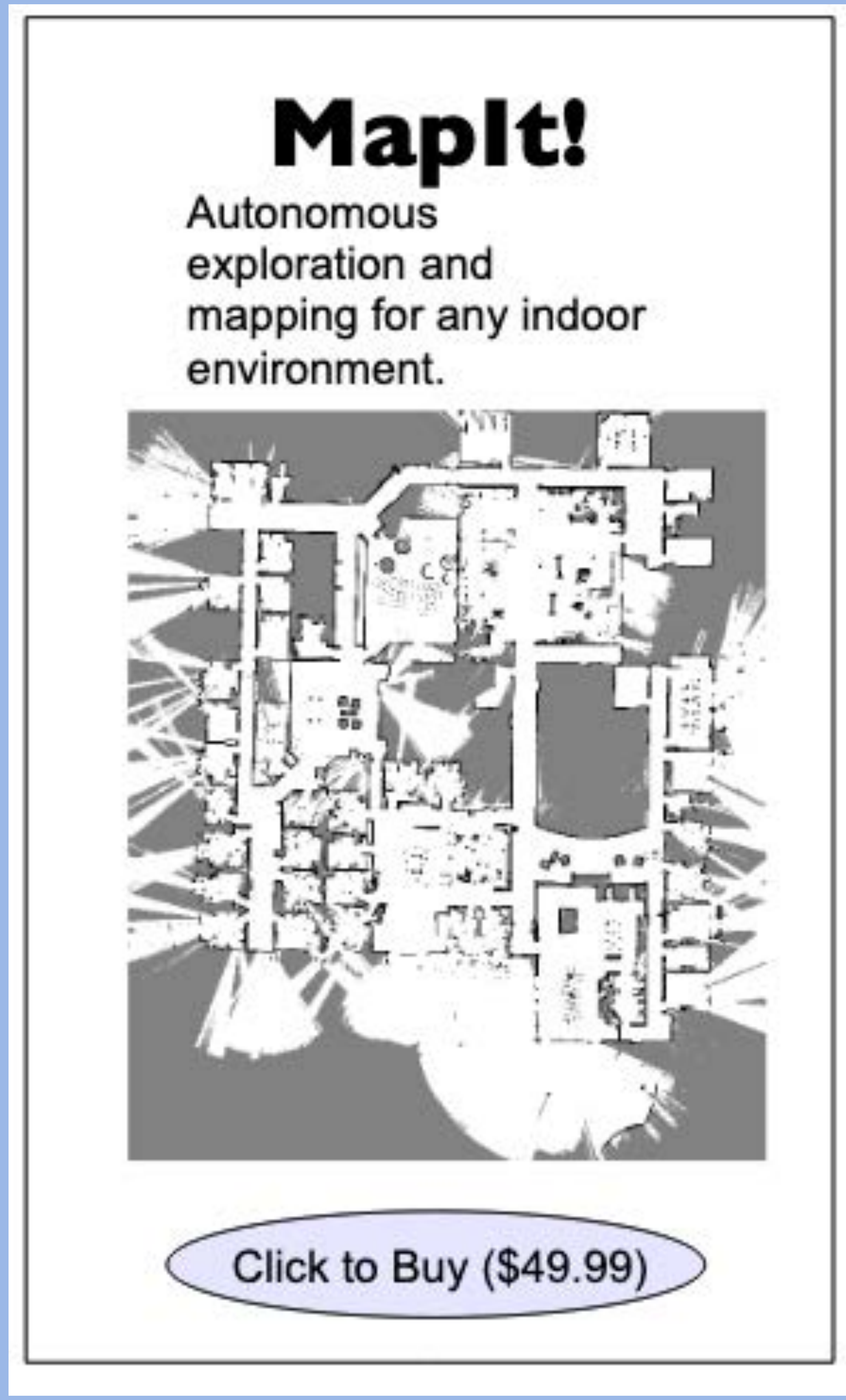
Operating System

Hardware

“Do this task for me”

Can we make your
world programmable ?





...e make a ...app store ?

What's a robot app?

- In the near future →
- Eventually:
  - CleanTheHouse
  - PatrolTheBuilding
  - …
- For now:
  - demonstrations
  - experiments
  - challenge entries (!)

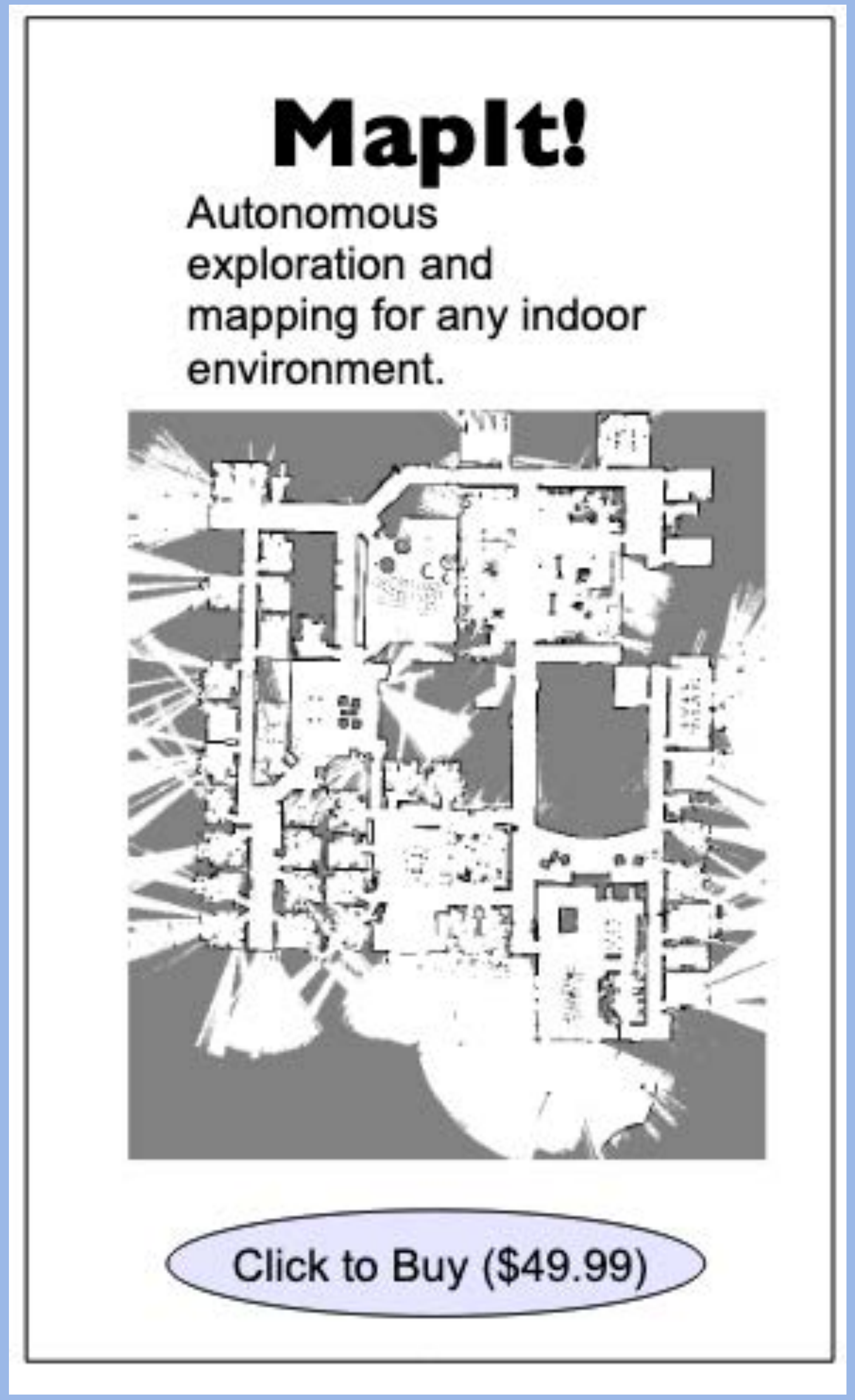
MapIt!

Autonomous exploration and mapping for any indoor environment.

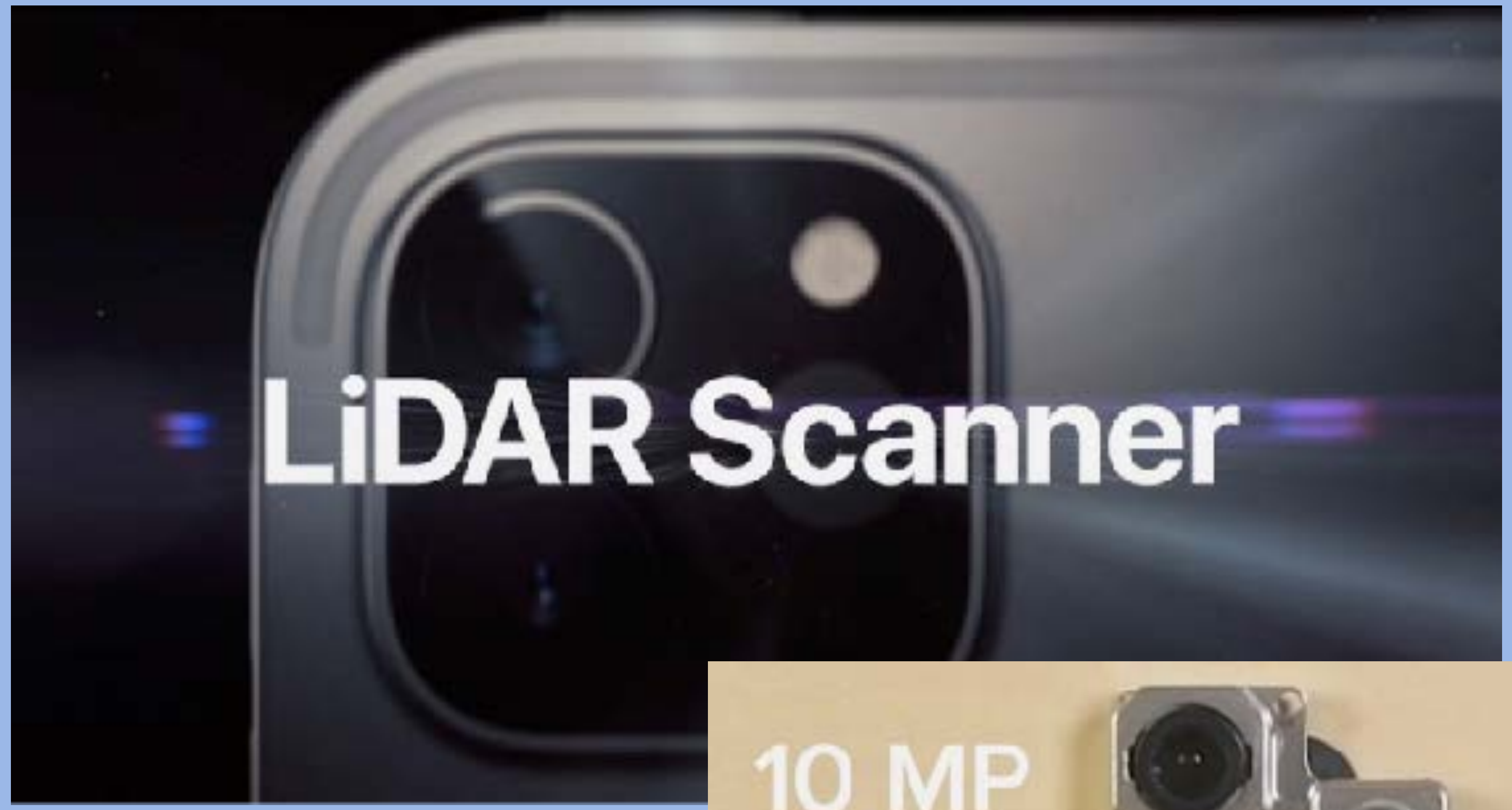
Click to Buy ($49.99)

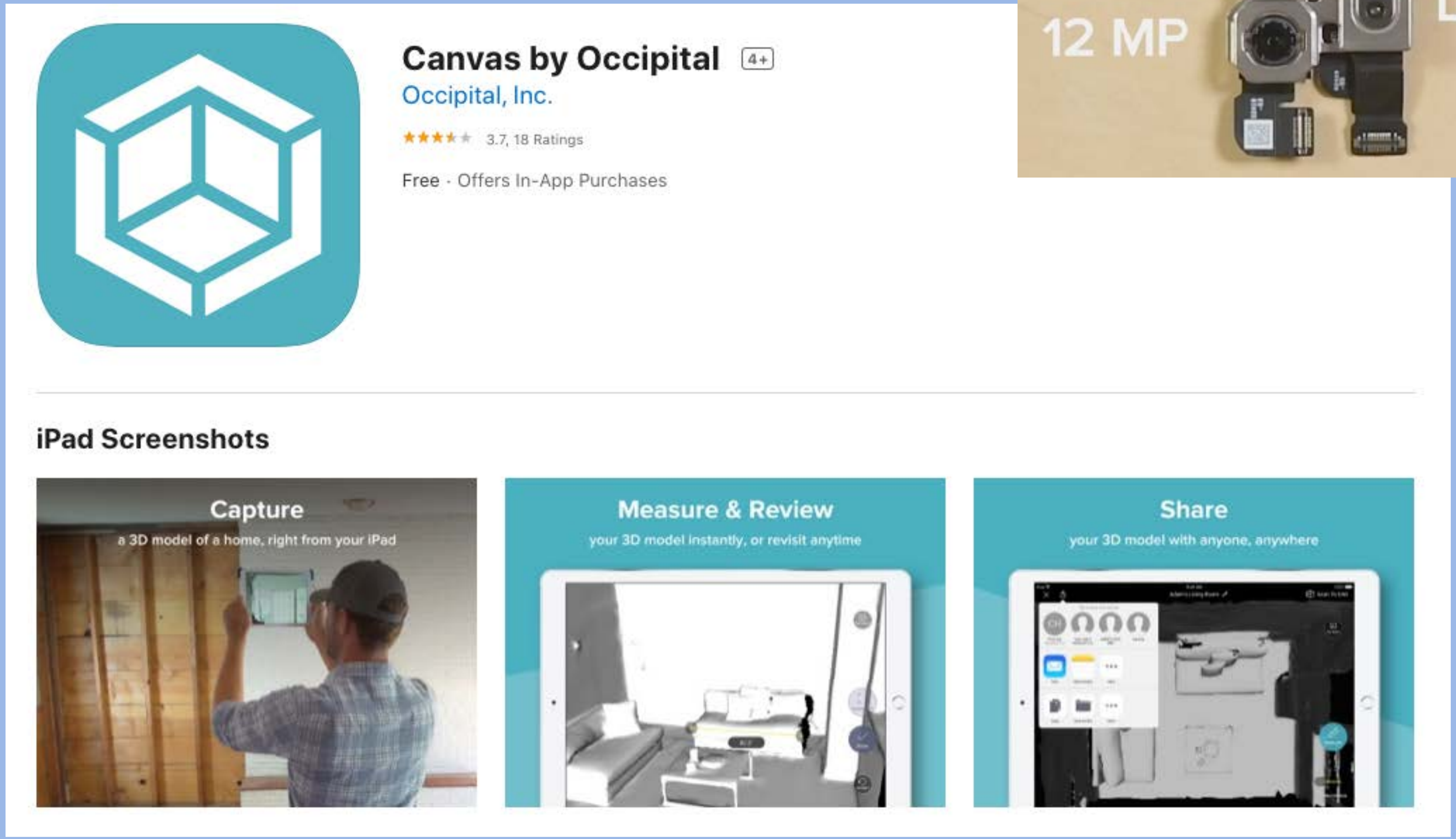
https://www.ros.org/presentations/2009-07_Gerkey_IJCAI-robotics-workshop.pdf

2009

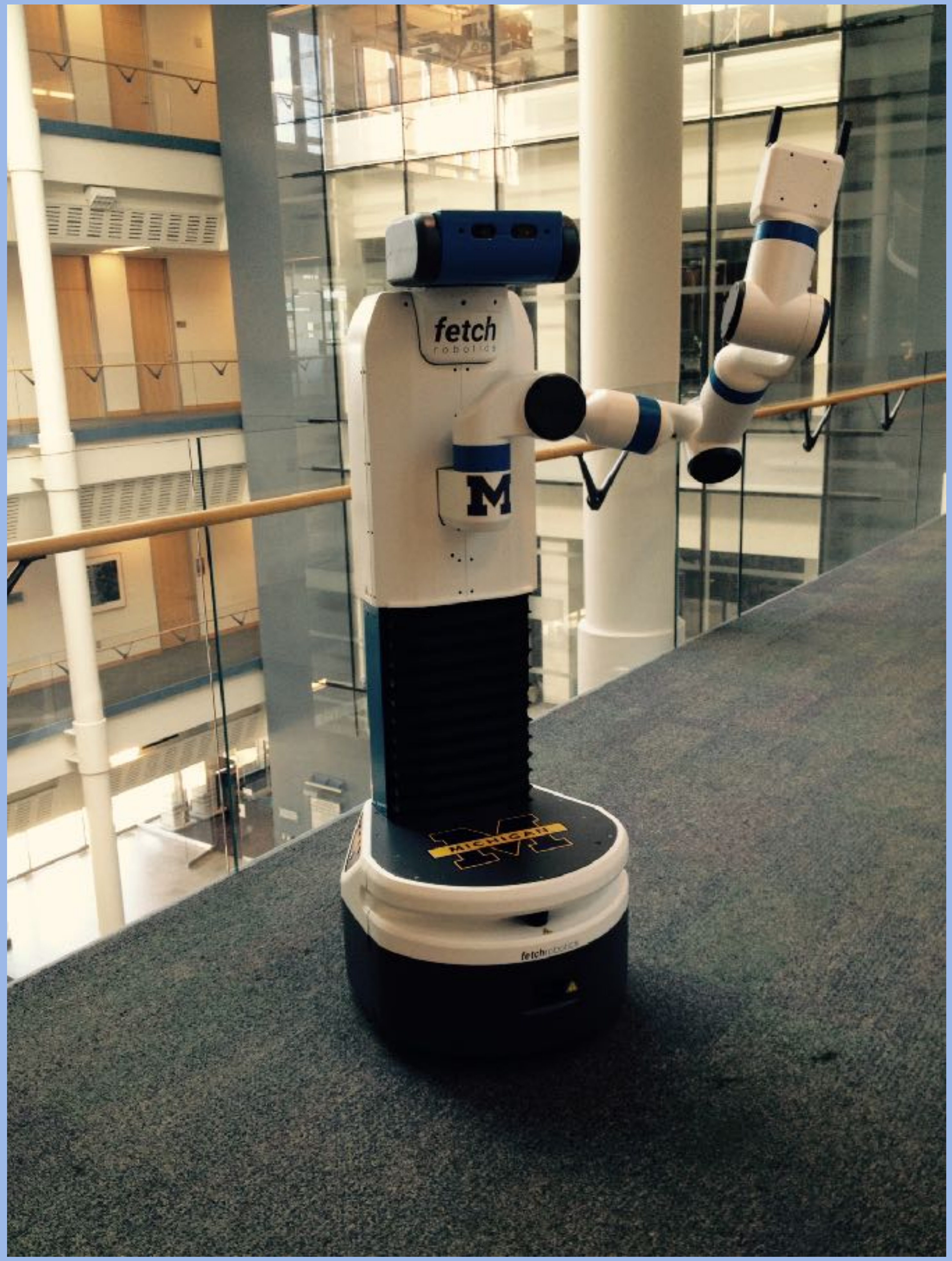


2020

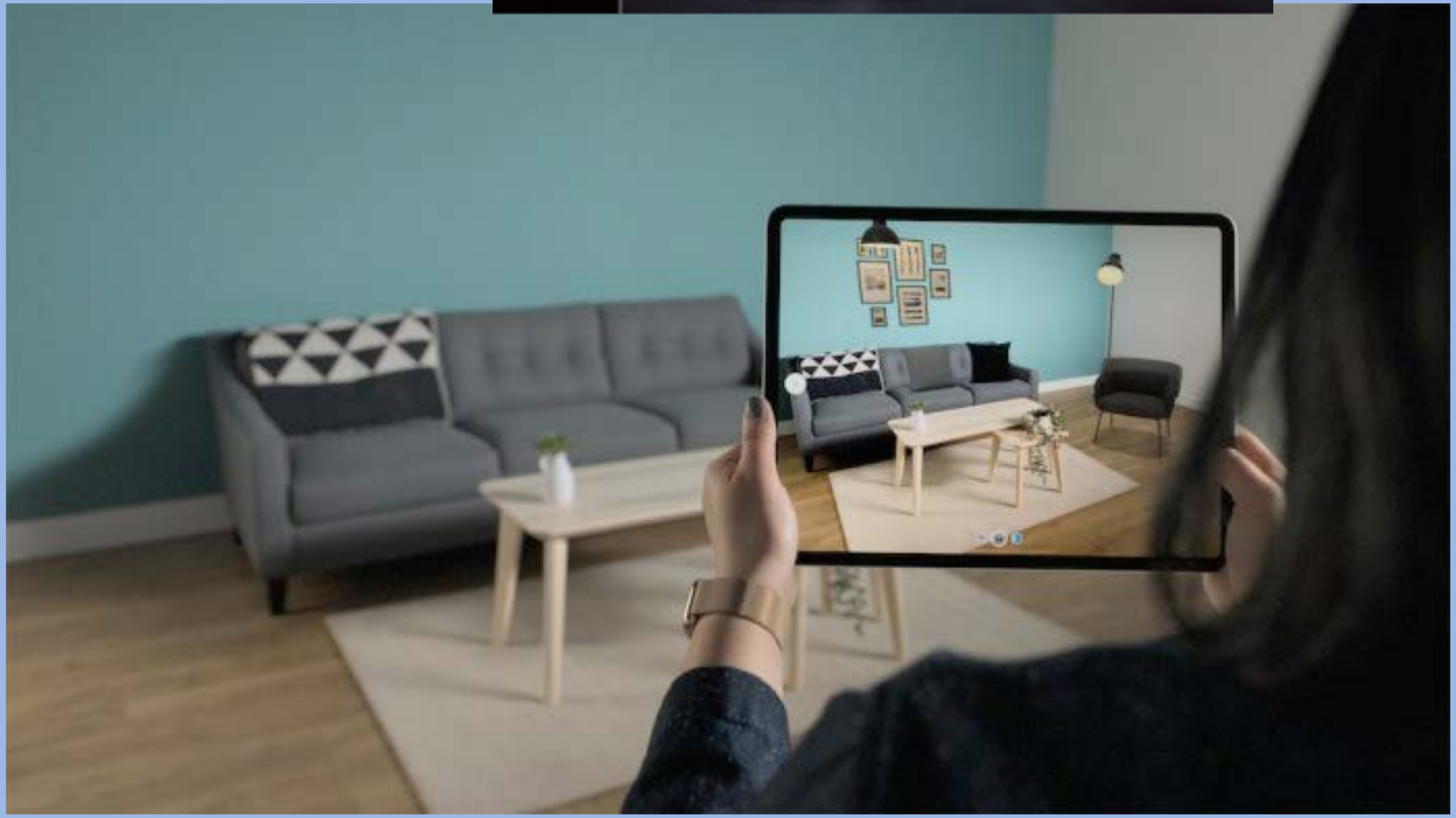




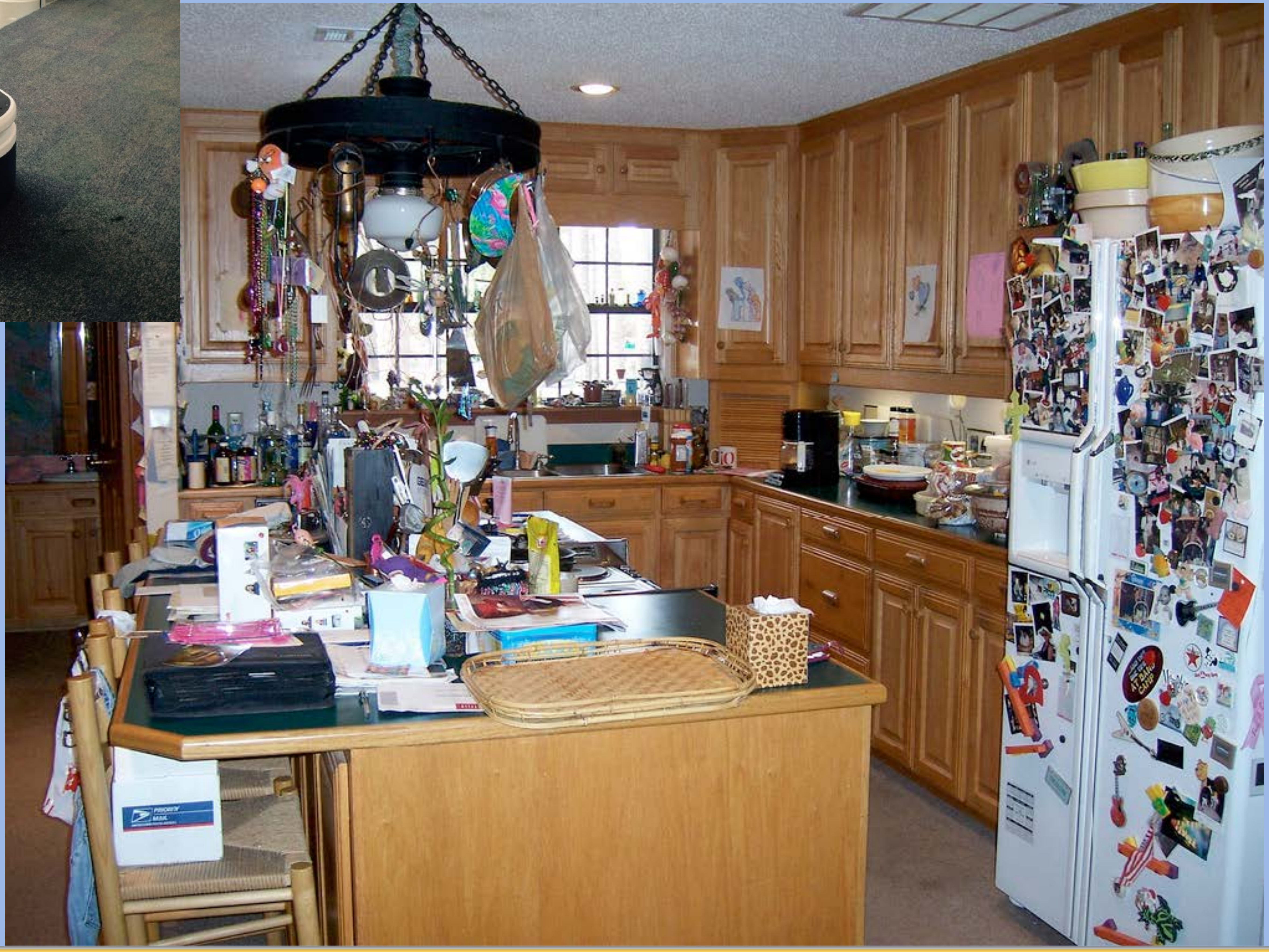
Slide borrowed from Michigan Robotics autorob.org


fetch
M

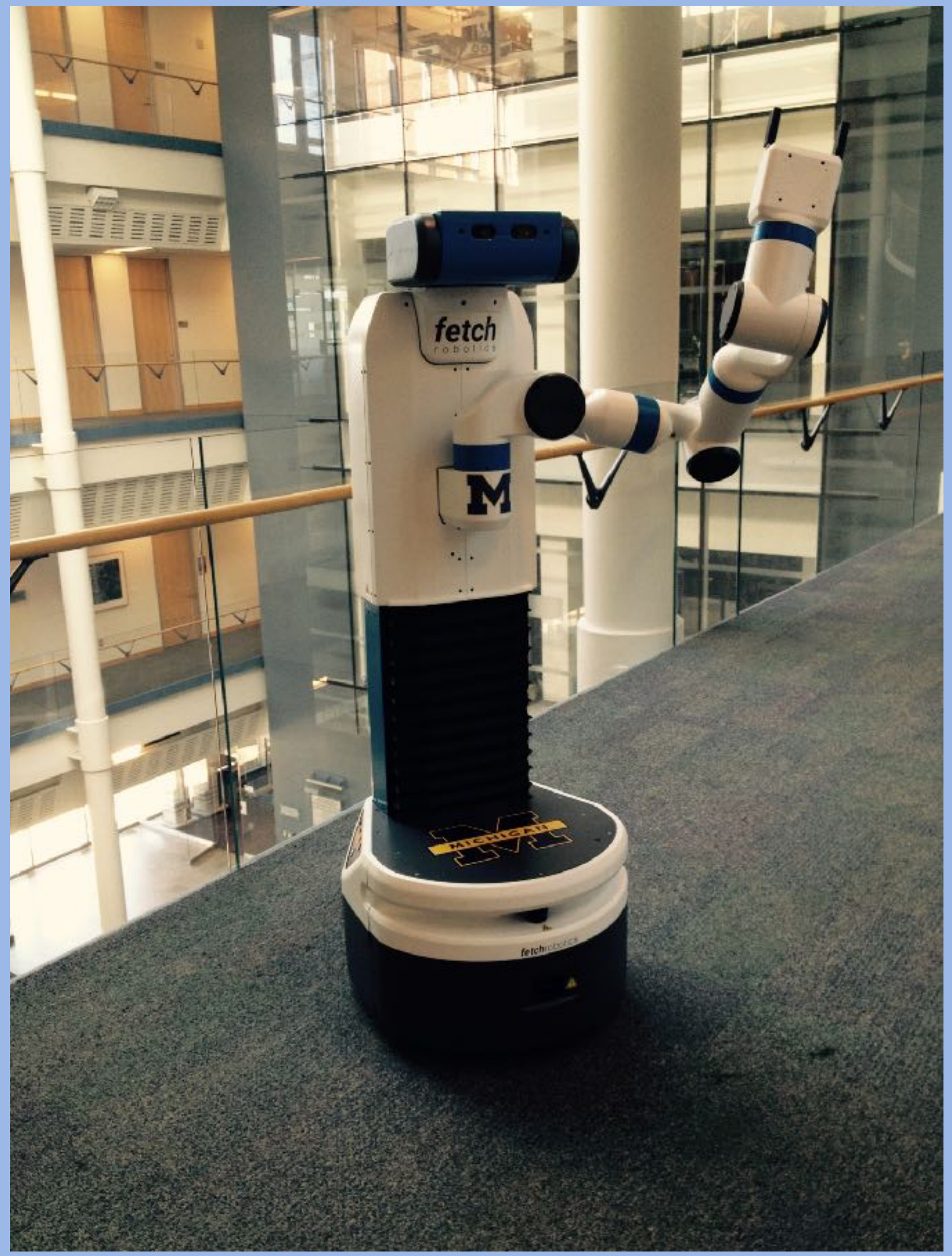

LiDAR Scanner




fetch

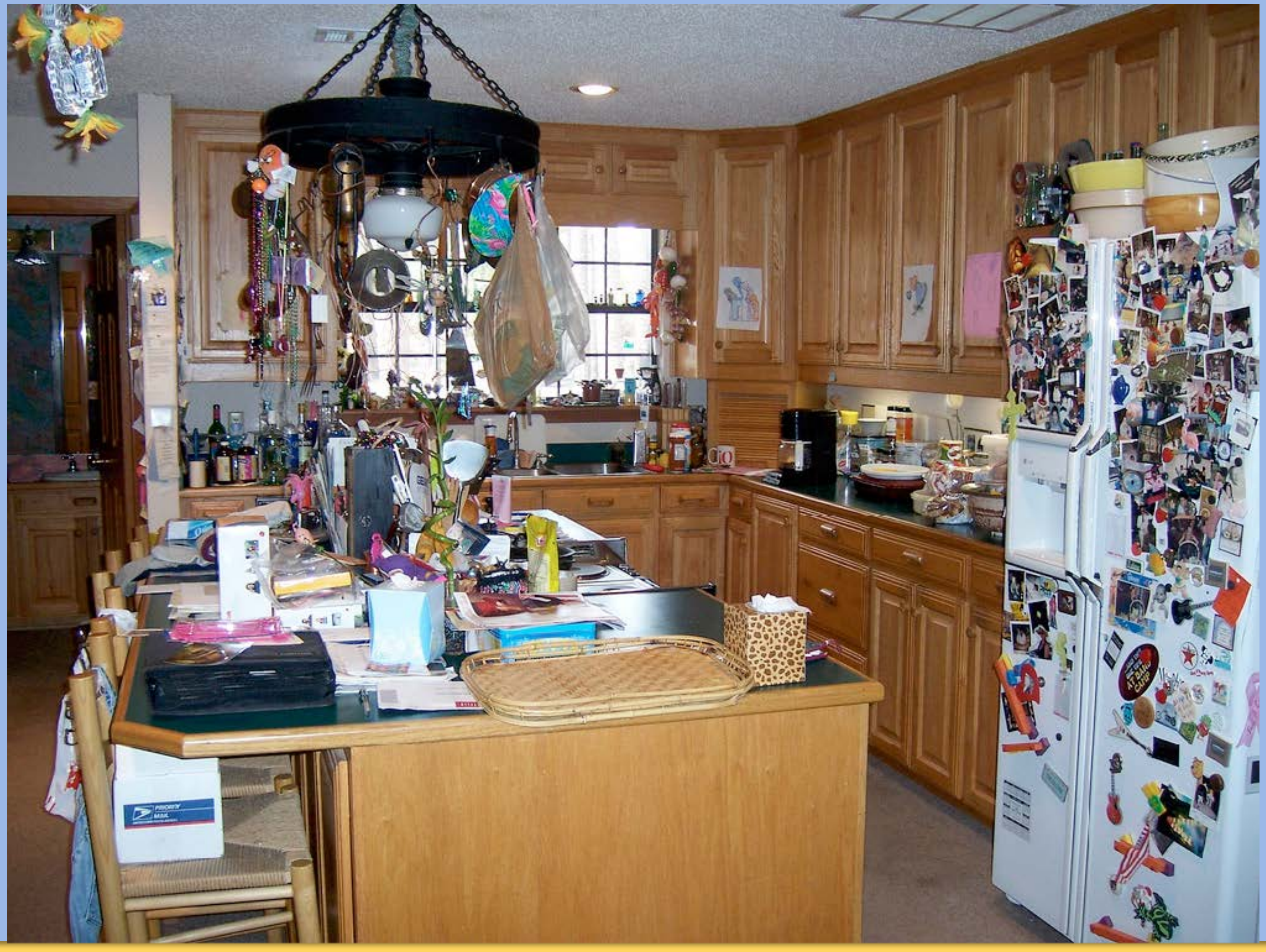


Use any robot x



to perform any task y

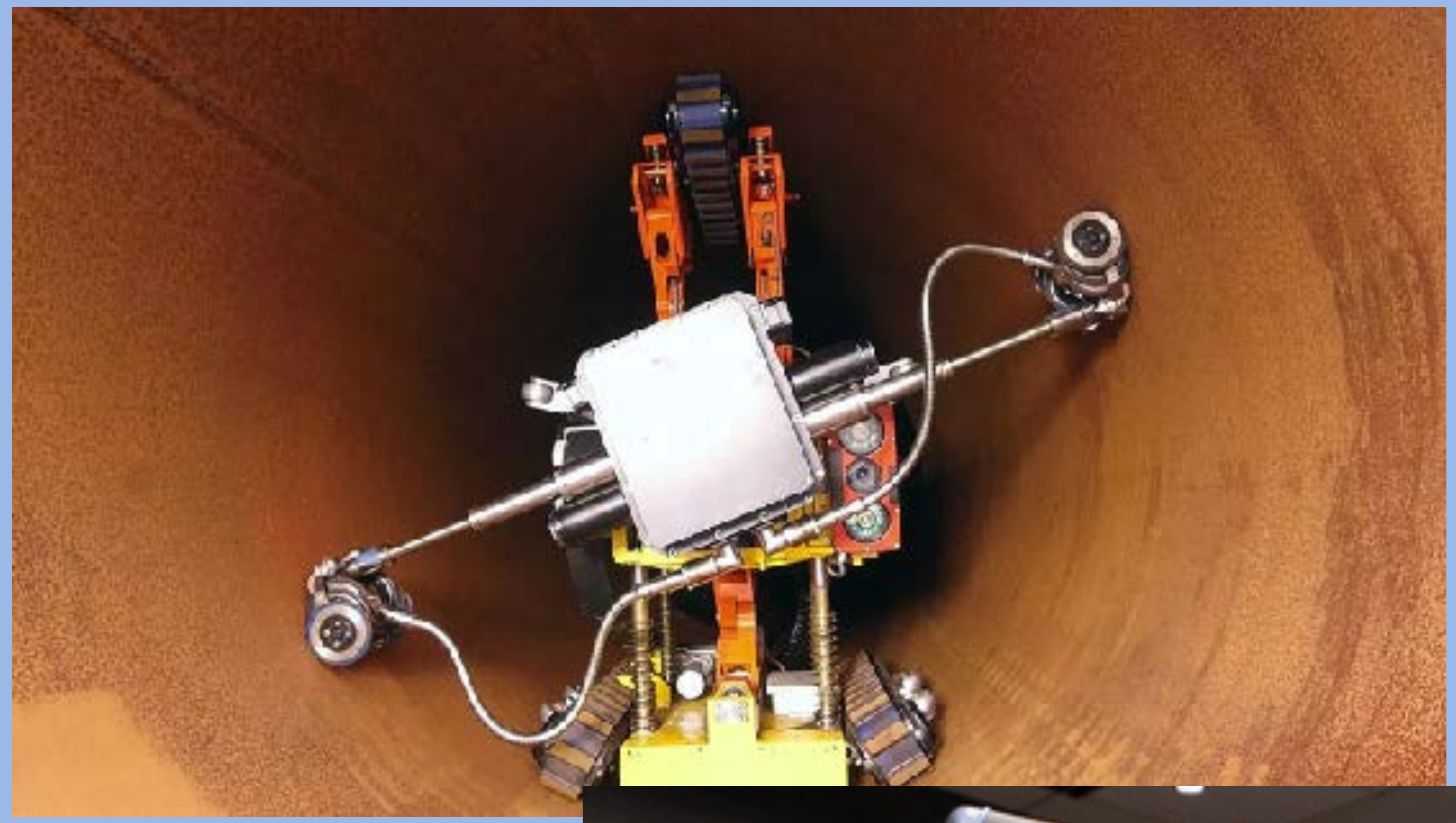
in any environment z



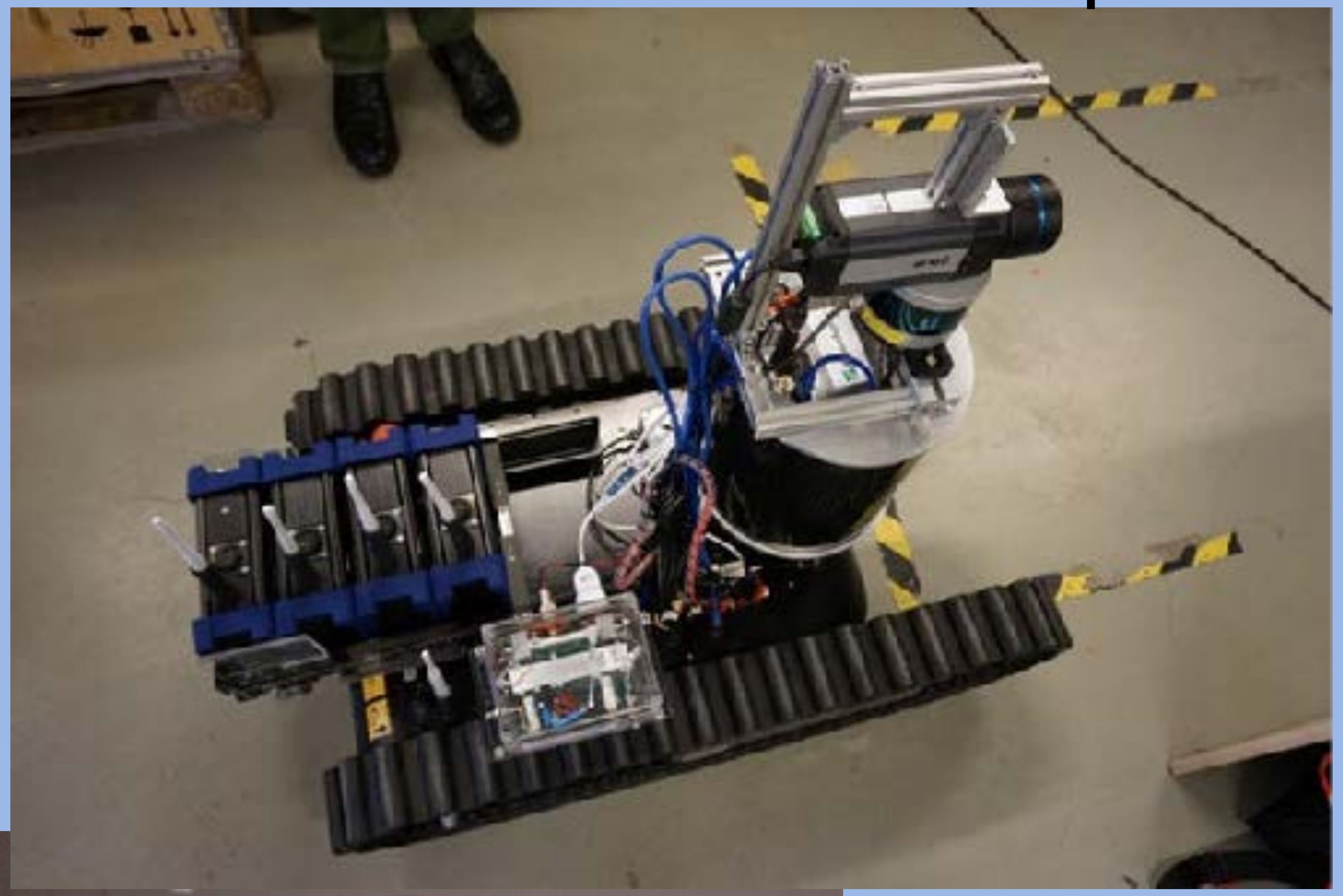
The 3Ds: Dirty, Dull, and Dangerous

"Autonomous" Driving

Infrastructure inspection



Nuclear cleanup



https://www.shadowrobot.com/blog/robots-saving-humans-from-dangerous-jobs/

https://techcrunch.com/2018/06/05/remote-control-driverless-car-startup-partners-with-vehicle-manufacturers/

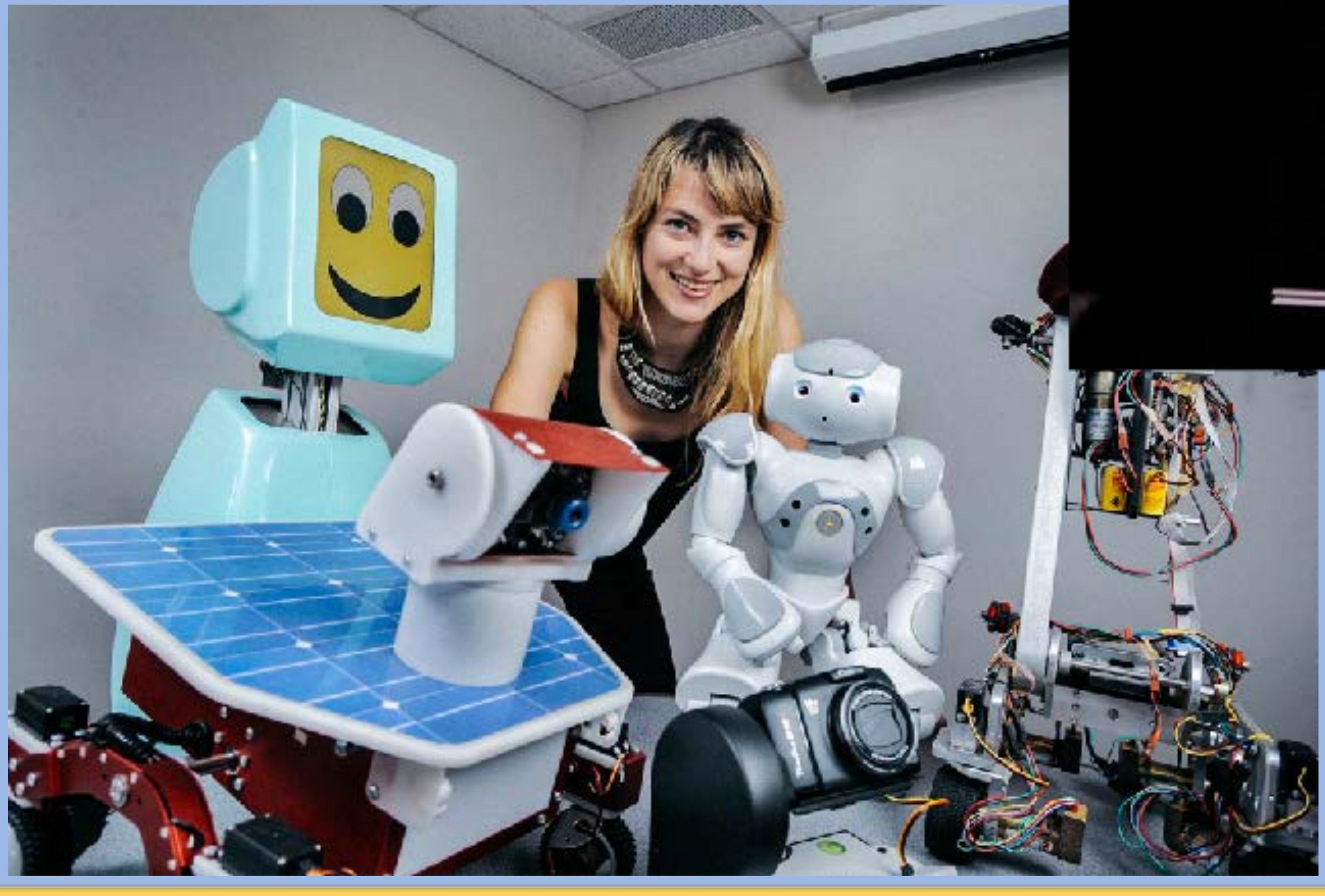
Social Robotics

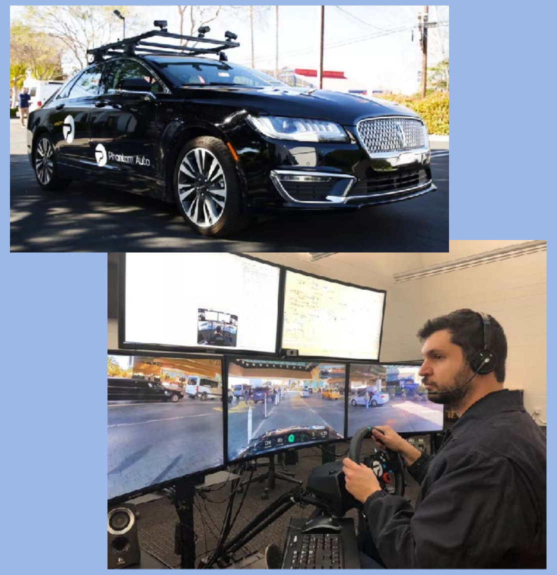
Autism treatment

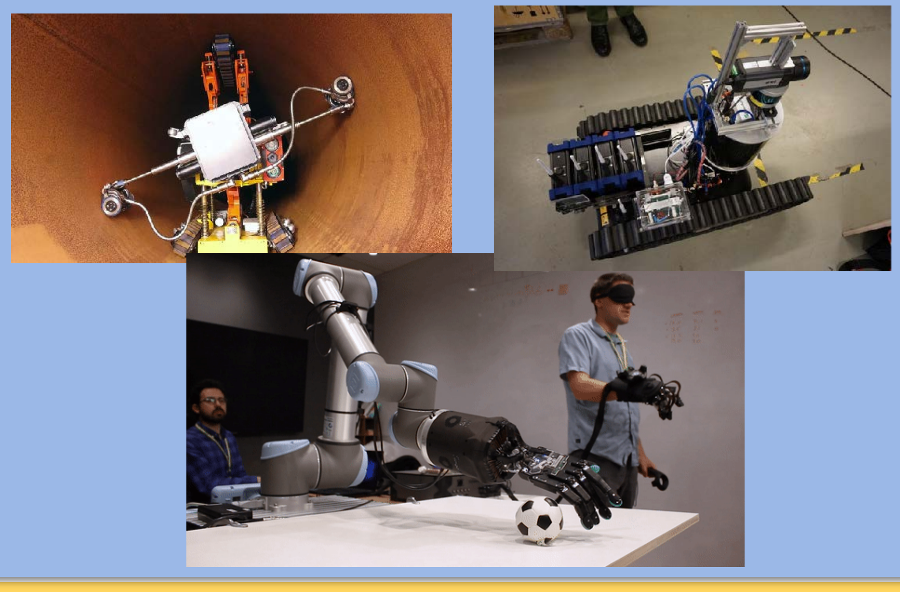
Education

Rehabilitation

Entertainment







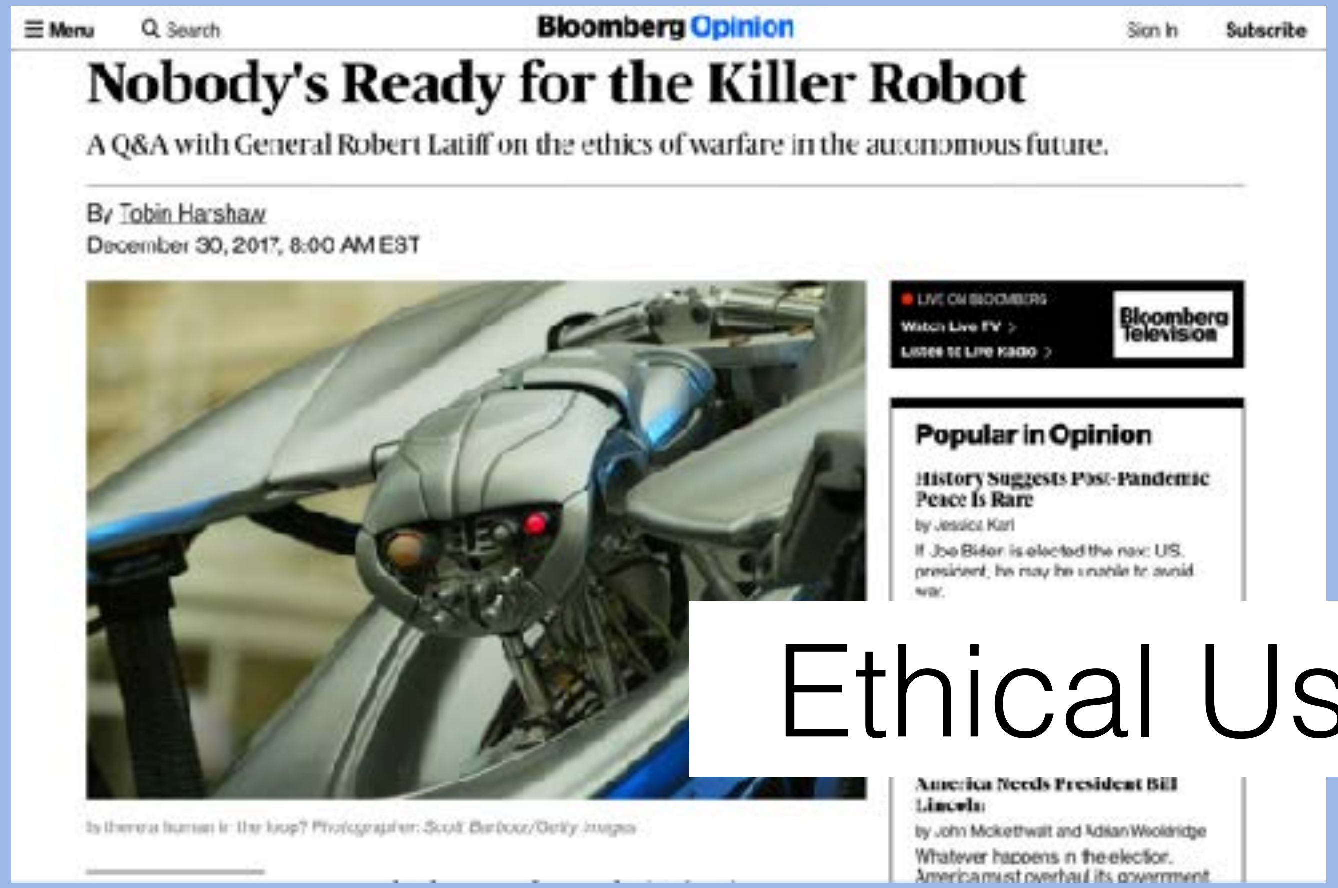
Elder care

Agriculture

Exploration

Manufacturing

Security


Menu Search Bloomberg Opinion Sign In Subscribe

Nobody's Ready for the Killer Robot

A Q&A with General Robert Latiff on the ethics of warfare in the autonomous future.

By Tobin Harshaw
December 30, 2017, 8:00 AM EST

Popular in Opinion

History Suggests Post-Pandemic Peace Is Rare

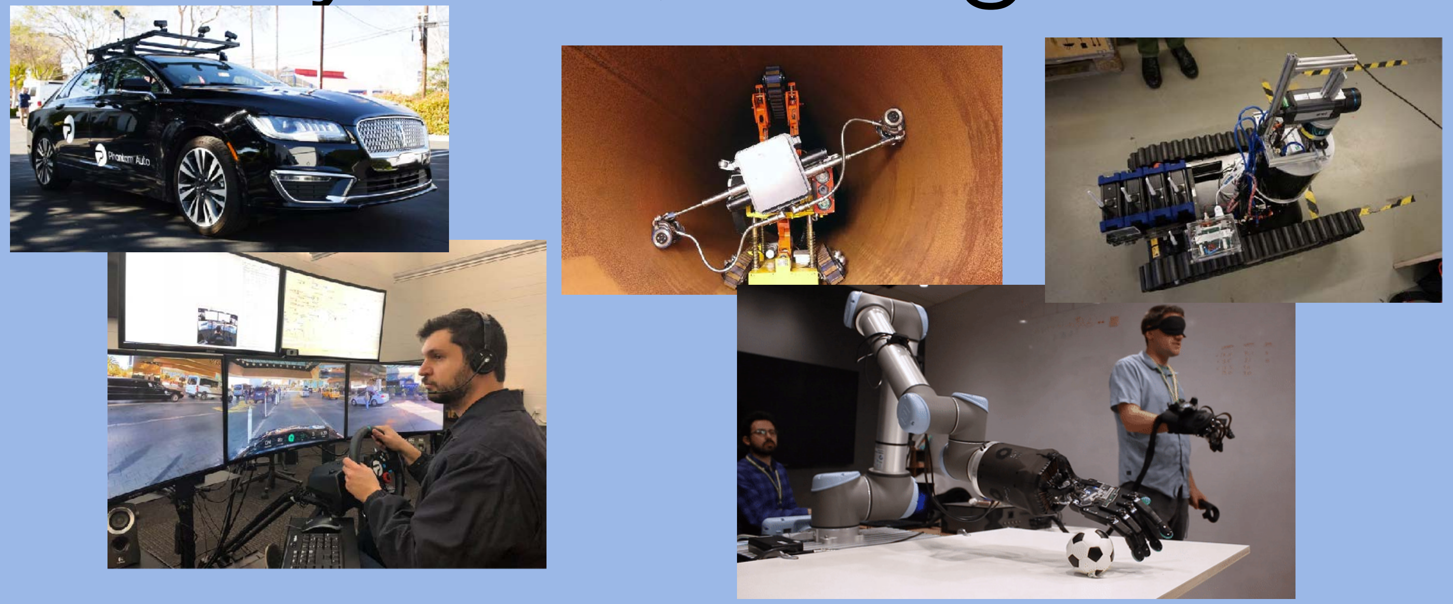
America Needs President Bill Lincoln

Ethical Use

Lethal Force

Social Robotics

Dirty, Dull, Dangerous



Medicine

Slide borrowed from Michigan Robotics autorob.org

Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

LCM
ROS
Player
YARP
MOOS

Operating System

Hardware

Slide borrowed from Michigan Robotics autorob.org

Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

LCM ROS Player YARP MOOS10

Operating System

Hardware

Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

Build your own Robot OS

Operating System

Hardware

Slide borrowed from Michigan Robotics autorob.org

Robot Operating System

Build your own Robot OS

Localization and Mapping

Path Planning

Feedback Control

Robot Vision

Motion Planning

Dynamical Simulation

Collision Detection

Decision Making Systems

Forward Kinematics

Multi-robot Coordination

Task planning

Inverse Kinematics

Robot Operating System

Build your own Robot OS

Robot Operating System

Covered at **breadth** in CSCI5551

Slide borrowed from Michigan Robotics autorob.org

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Work with a real robot once this semester

Turtlebot3

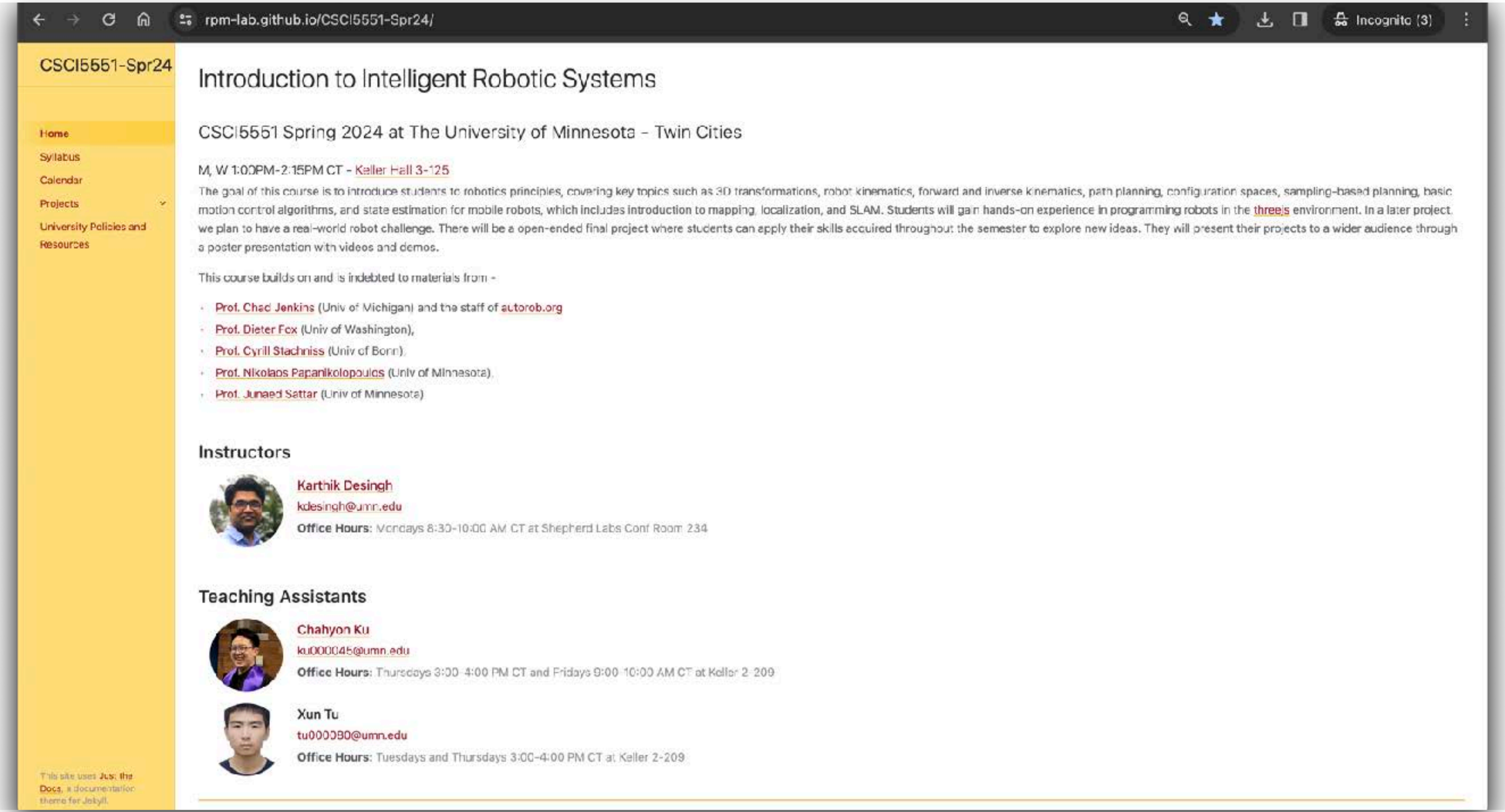
To be determined

Turtlebot4

Course Resources

Course Website

https://rpm-lab.github.io/CSCI5551-Spr24/


rpm-lab.github.io/CSCI5551-Spr24/

Incognito (3)

CSCI5551-Spr24

Home

Syllabus

Calendar

Projects

University Policies and Resources

Introduction to Intelligent Robotic Systems

CSCI5551 Spring 2024 at The University of Minnesota - Twin Cities

M, W 1:00PM-2:15PM CT - Keller Hall 3-125

The goal of this course is to introduce students to robotics principles, covering key topics such as 3D transformations, robot kinematics, forward and inverse kinematics, path planning, configuration spaces, sampling-based planning, basic motion control algorithms, and state estimation for mobile robots, which includes introduction to mapping, localization, and SLAM. Students will gain hands-on experience in programming robots in the threejs environment. In a later project, we plan to have a real-world robot challenge. There will be a open-ended final project where students can apply their skills acquired throughout the semester to explore new ideas. They will present their projects to a wider audience through a poster presentation with videos and demos.

This course builds on and is indebted to materials from -

- Prof. Chad Jenkins (Univ of Michigan) and the staff of autorob.org
- Prof. Dieter Fox (Univ of Washington),
- Prof. Cyrill Stachniss (Univ of Bonn),
- Prof. Nikolaos Papanikolopoulos (Univ of Minnesota),
- Prof. Junaed Sattar (Univ of Minnesota)

Instructors

Karthik Desingh

kdesingh@umn.edu

Office Hours: Mondays 8:30-10:00 AM CT at Shepherd Labs Conf Room 234

Teaching Assistants

Chahyon Ku

ku000045@umn.edu

Office Hours: Thursdays 3:00-4:00 PM CT and Fridays 9:00-10:00 AM CT at Keller 2-209

Xun Tu

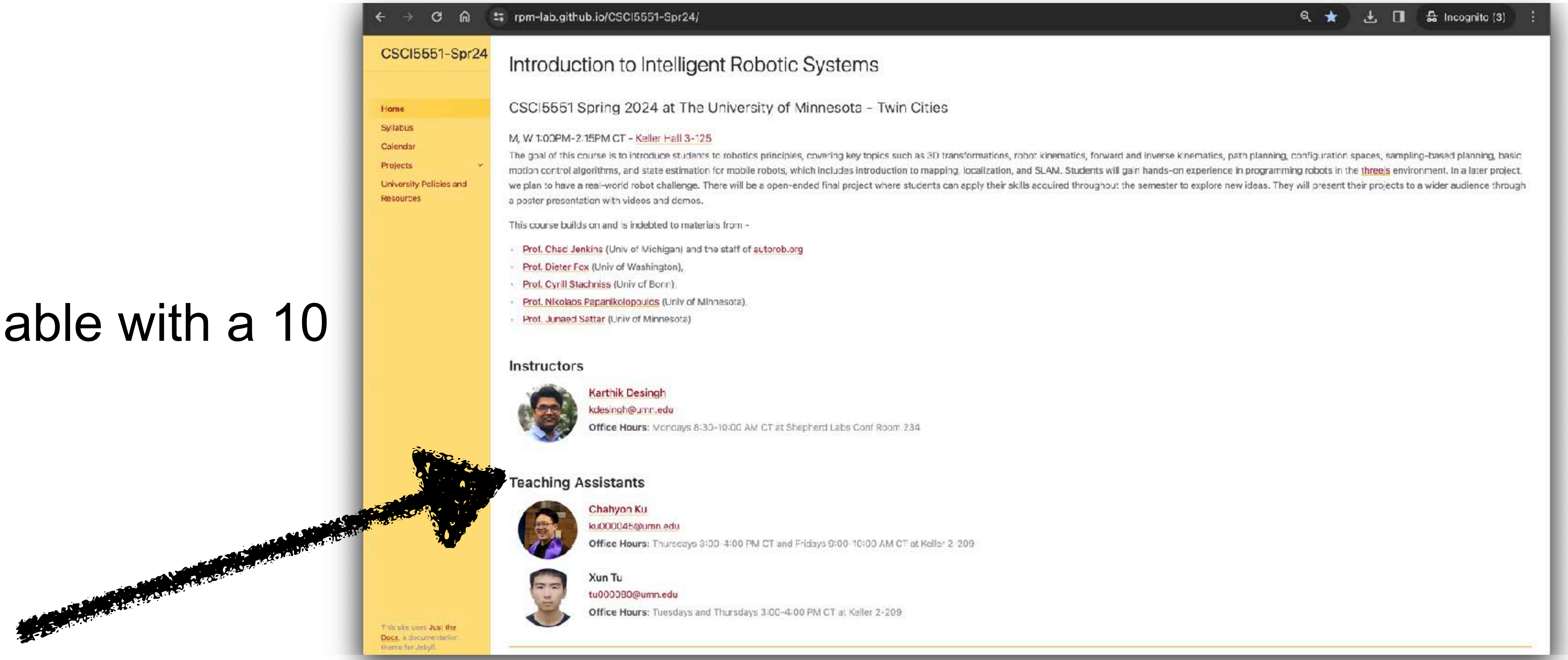
tu000080@umn.edu

Office Hours: Tuesdays and Thursdays 3:00-4:00 PM CT at Keller 2-209

This site uses Just the Docs, a documentation theme for Jekyll.

Meeting Logistics

- **In-person Lectures**
  - Mon & Wed 1:00-2:15 PM CT
  - Keller Hall 3-125
  - UNITE recordings will be available with a 10 day delay
- **Office Hours**
  - Times posted on the website
  - Or by appointment



Course Structure

- **Objective:** Give you the computational skills to understand the nuts and bolts of developing a robotic system using kinematics and dynamics. Give you a broader idea of topics in robotics to further pursue advanced courses and research on these topics.

- **Project focused class:**

  - 7 total projects: building in complexity from basic transformations-rotations to motion planning and mobile manipulation

Course Schedule

https://rpm-lab.github.io/CSCI5551-Spr24/calendar/

Snapshot of Planned Schedule

Spring-24-Calendar : Sheet1

Lec #	Date	Topic	Project Announcement	Project Due	In-class Quiz
1	01/17	Introduction			
2	01/22	Planning I - Path Planning			
3	01/24	Linear Algebra Refresher	P1: JS, BFS, DFS		Q1
4	01/29	Representations I - Transformations			
5	01/31	Representations II - Rotations - Quaternions	P2: Forward Kinematics	P1: Due	Q2
6	02/05	Manipulation I - Forward Kinematics			
7	02/07	Manipulation II - Inverse Kinematics	P3: Robot Dance	P2: Due	Q3
8	02/12	Manipulation III - Inverse Kinematics			
9	02/14	Planning II - Bug Algorithms	P4: Inverse Kinematics	P3: Due	Q4
10	02/19	Planning III - Configuration Space			
11	02/21	Planning IV - Sampling-based Planning			Q5
12	02/26	Planning V - Potential Fields			
13	02/28	Planning VI - Collision Detection	P5: Planning	P4: Due	Q6
14	03/04	Spring Break			
15	03/06	Spring Break			
16	03/11	Planning - New Frontiers	*Forming groups for P7 and FP*		
17	03/13	Motion Control	P6: Mobile Manipulation	P5: Due	Q7
18	03/18	Mobile Robotics I - Probability			
19	03/20	Mobile Robotics II - Sensor and Motion Models	P7: Real Robot Challenge	P6: Due	Q8
20	03/25	Mobile Robotics III - Kalman			
21	03/27	Mobile Robotics IV - Localization			Q9
22	04/01	Mobile Robotics V - Localization			
23	04/03	Mobile Robotics VI - Mapping	FP Proposals Request		Q10
24	04/08	Mobile Robotics VII - SLAM			
25	04/10	Open Ended Final Project Pitches		FP Proposals Due	Q11
26	04/15	Open Ended Final Project Pitches			
27	04/17	Open Ended Final Project Pitches		P7: Due	Q12
28	04/22	P7 Challenge Day - Drone Lab			
29	04/24	P7 Challenge Day - Drone Lab			
30	04/29	Guest Lectures - Extra office hours			
31	05/01	Guest Lectures - Extra office hours		FP Posters Due	
32	05/06	Poster Day		FP Videos Due	

Guided Projects P1-P6 (Individual)

- Project 1
  - JS, BFS, DFS (Search and Planning)
- Project 2
  - Forward Kinematics
- Project 3
  - Robot dance
- Project 4
  - Inverse Kinematics
- Project 5
  - Planning
- Project 6
  - Mobile Manipulation


0.0.0.0:8000/home.html
Not secure 0.0.0.0:8000/home.html
Incognito
Welcome to KinEval. I want to see some text. Can you place a message here?
Autograder
localhost:8000/home.html?robot=robots/sawyer/sawyer.urdf.js
localhost:8000/home.html?robot=robots/baxter/baxter.urdf.js
0.0.0.0:8000/home.html?robot=robots/robot_urdf_example.js
0.0.0.0:8000/home.html?robot=robots/robot_crawler.js
0.0.0.0:8000/home.html?robot=robots/fetch/fetch.urdf.js
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Geometries and Axes
Colors
Close Controls


localhost:8000/home.html
Welcome to KinEval. I want to see some text. Can you place a message here?
kineval
just_starting
User Parameters
Alt+A
0.0.0.0:8000/home.html?world=worlds/world_mobile_manipulate.js
Not secure
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Mobile Manipulate
Display
Close Controls
0.0.0.0:8000/home.html?robot=robots/fetch/fetch.urdf.js
Not secure
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Close Controls

Guided Projects (Group)

- Project 7
  - Real Robot Challenge (**TBD**)

Turtlebot3

To be determined

Turtlebot4

Open-ended Final Project (Group)

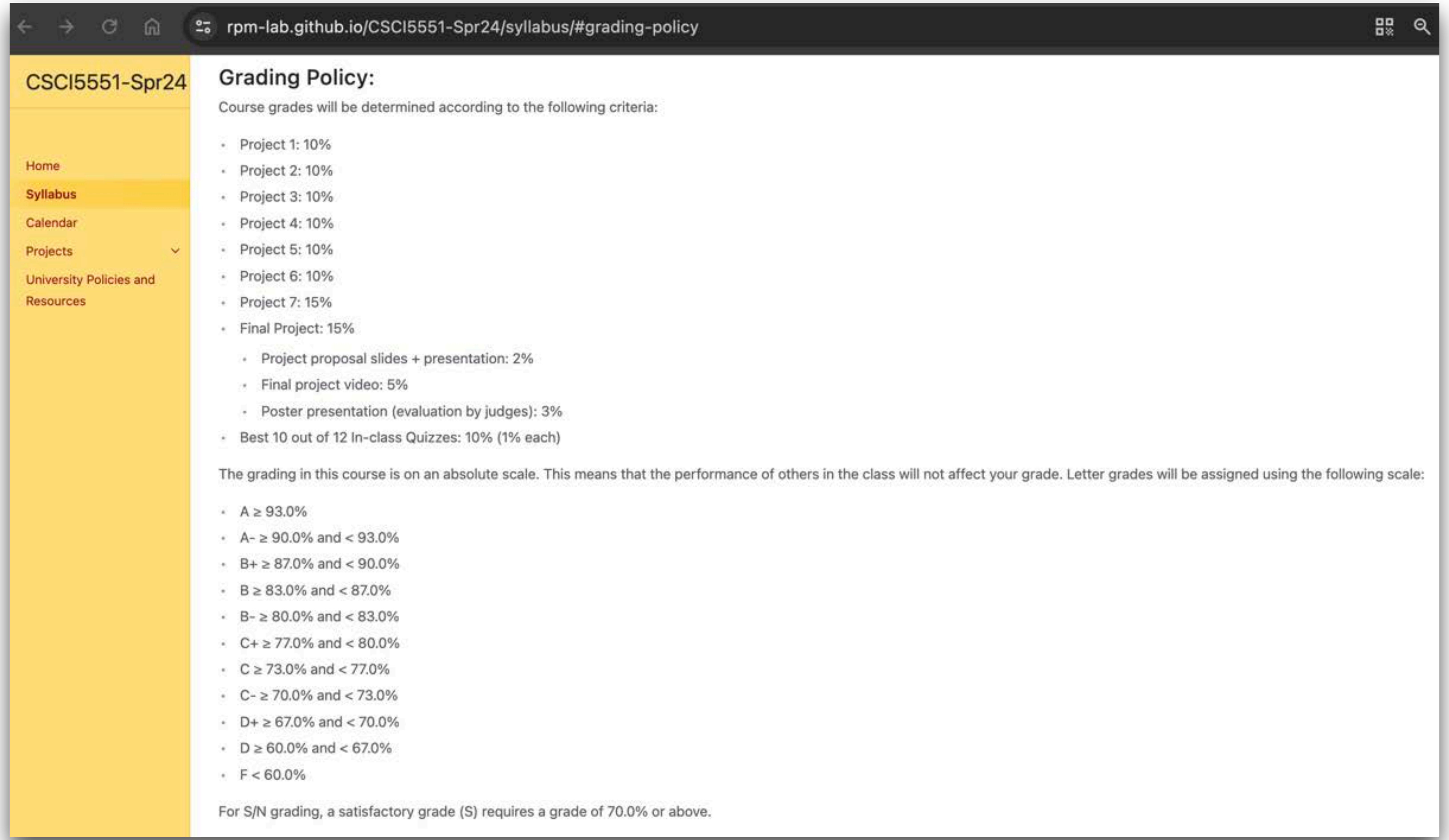
- Open-ended and will let student groups explore ideas with their learnings from the course.

Project Grading

- Guided Projects 1-6
  - 3 total late day tokens are available
  - 25% daily penalty after deadline, if you run out of late tokens.
- Guided Project 7
  - No late days
- Open-ended Final Project
  - No late days

Overall Grading Policy

https://rpm-lab.github.io/CSCI5551-Spr24/syllabus/#grading-policy



rpm-lab.github.io/CSCI5551-Spr24/syllabus/#grading-policy

CSCI5551-Spr24

- Home
- Syllabus
- Calendar
- Projects
- University Policies and Resources

Grading Policy:

Course grades will be determined according to the following criteria:

- Project 1: 10%
- Project 2: 10%
- Project 3: 10%
- Project 4: 10%
- Project 5: 10%
- Project 6: 10%
- Project 7: 15%
- Final Project: 15%
  - Project proposal slides + presentation: 2%
  - Final project video: 5%
  - Poster presentation (evaluation by judges): 3%
- Best 10 out of 12 In-class Quizzes: 10% (1% each)

The grading in this course is on an absolute scale. This means that the performance of others in the class will not affect your grade. Letter grades will be assigned using the following scale:

- A ≥ 93.0%
- A- ≥ 90.0% and < 93.0%
- B+ ≥ 87.0% and < 90.0%
- B ≥ 83.0% and < 87.0%
- B- ≥ 80.0% and < 83.0%
- C+ ≥ 77.0% and < 80.0%
- C ≥ 73.0% and < 77.0%
- C- ≥ 70.0% and < 73.0%
- D+ ≥ 67.0% and < 70.0%
- D ≥ 60.0% and < 67.0%
- F < 60.0%

For S/N grading, a satisfactory grade (S) requires a grade of 70.0% or above.

Collaboration Policy

- All work submitted must be your own.
  - All code submitted must comply with College of Engineering Honor Code.
- Cheating will not be tolerated and can lead to termination from the program.
- No code can be communicated, including verbally.
  - Explicit use of external sources must be clearly cited.
- Free flow of discussion and ideas is encouraged.

University Policy

https://rpm-lab.github.io/CSCI5551-Spr24/policies_resources/

rpm-lab.github.io/CSCI5551-Spr24/policies_resources/ Incognito (3)

CSCI5551-Spr24

Search CSCI5551-Spr24

Ed Forum Autograder Gradescope RPM Lab

- Home
- Syllabus
- Calendar
- Projects
- **University Policies and Resources**

Standard University Policies

This class follows standard university policies. It's your responsibility to be familiar with:

- Student conduct code
- Academic dishonesty
- Makeup work for legitimate absences
- Student responsibilities
- Grading and transcripts
- Sexual harassment
- Equity, diversity, equal opportunity, and affirmative action
- Safety in classroom and campus

Mental Health Information

The Department of Computer Science & Engineering strives to ensure all students have access to resources that will help them feel safe and supported. We are deeply committed to the wellbeing of all students, staff, and faculty. Your mental health is part of who you are and if you are unsure where to turn, we are here to help you. If you are encountering challenges, I encourage you to visit our department's Mental Health Resources website and connect with one of our Mental Health Advocates, who are available to answer questions about campus mental health resources and services.

Disability Information

University policy is to provide, on a flexible and individualized basis, reasonable accommodations to students who have documented disability conditions (e.g., physical, learning, psychiatric, vision, hearing, or systemic) that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities are encouraged to contact the Disability Resources Center (DRC) to discuss individual needs for accommodations.

If you have already been working with the Disability Resource Center and have a letter from them to help direct your instructors on to set you up for success, then please send that to me at the beginning of the semester. In many cases, you will find that this course is already designed to be accessible for students that would benefit from additional flexibility (e.g., non-timed quizzes instead of large exams, flexible late policy on programming assignments, etc.). However, if additional accommodations are necessary, I will make every reasonable effort to make sure this class is a good experience for you.

Acknowledgments

Aspects of this syllabus (including this statement) were adapted from Evan Suma Rosenberg, Nathan Taylor, Daniel Keefe, Blair MacIntyre, Shana Watters, Lana Yarosh, and the American Association of University Professors Joint Statement on Rights and Freedoms of Students, because writing a good syllabus is hard.

Discussion Forum

- EdStem is the discussion forum used in this course.
- Discussion of quizzes and verbatim code must be private.
- You will be added to it this week.

Next lecture:
Search Algorithms - Path Planning