Robotics

Institut
de Robòtica
i Informàtica
Industrial

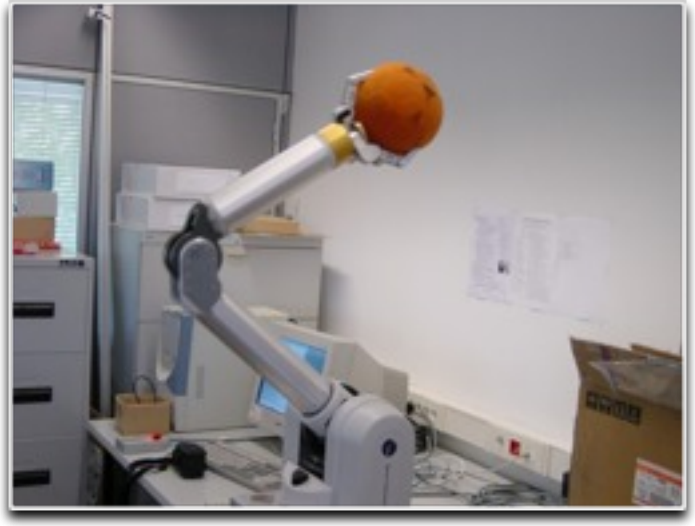
Guillem Alenyà

Fronteras de la Computación, 2011

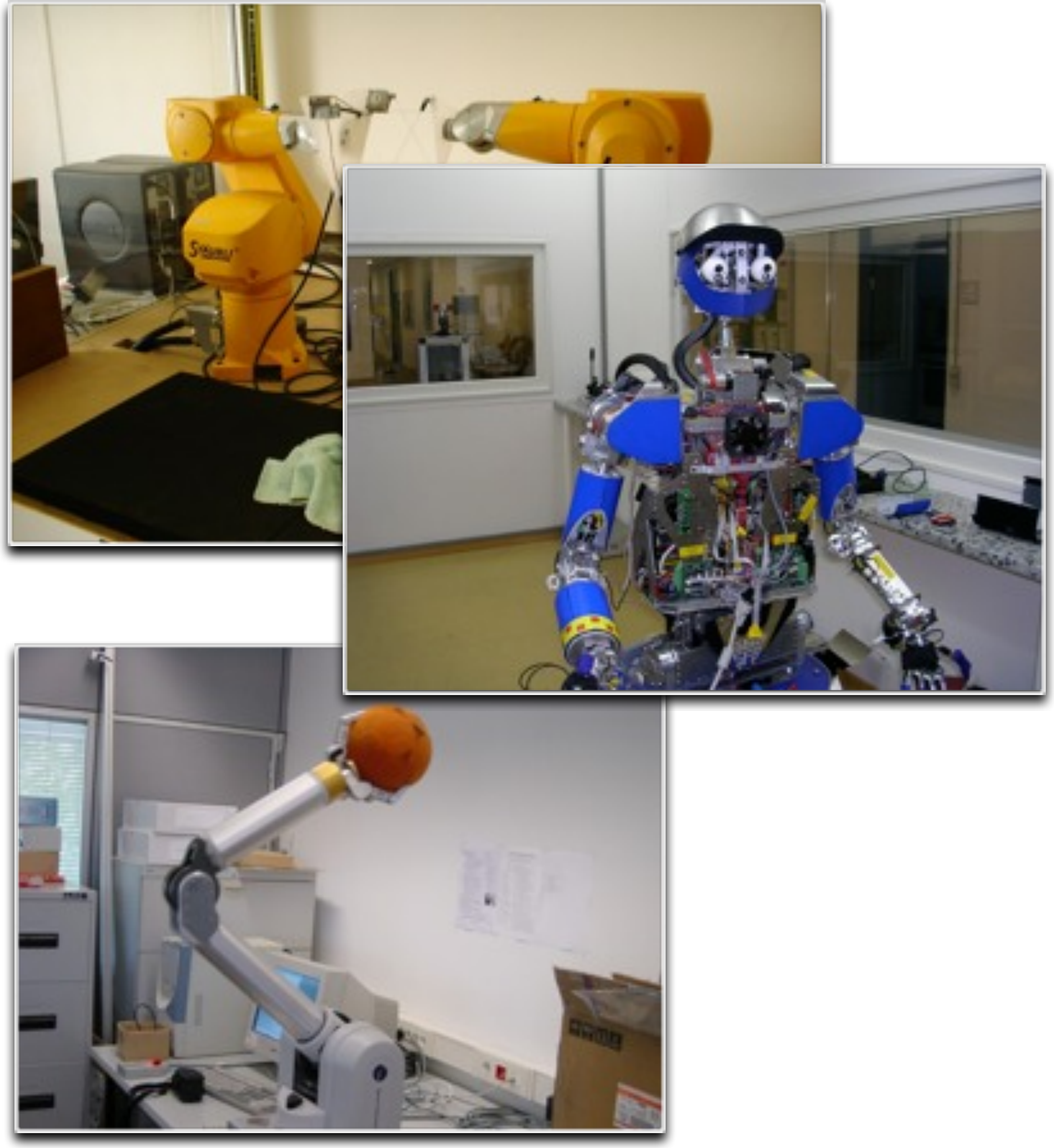
Robots

Robots

Robots



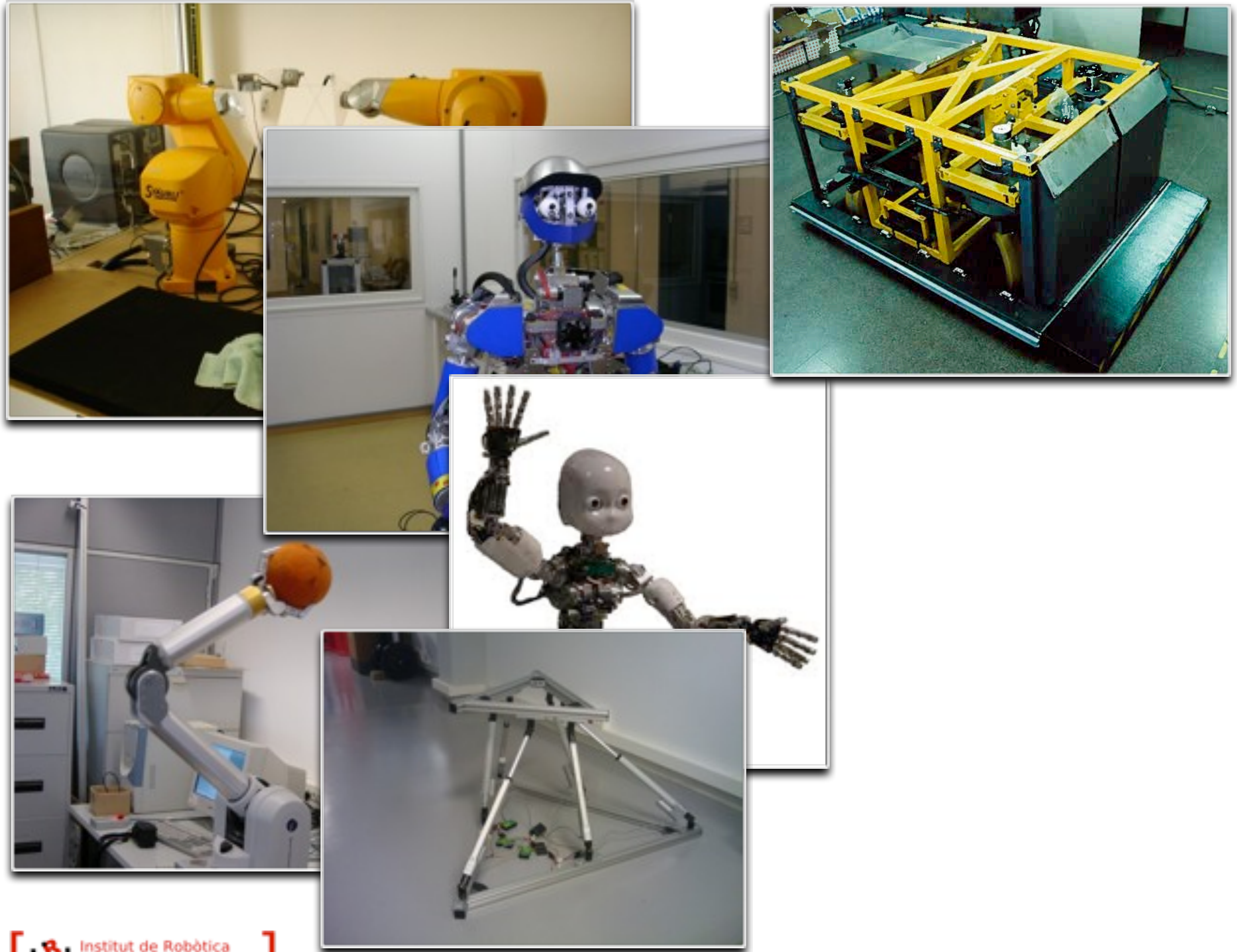
Robots



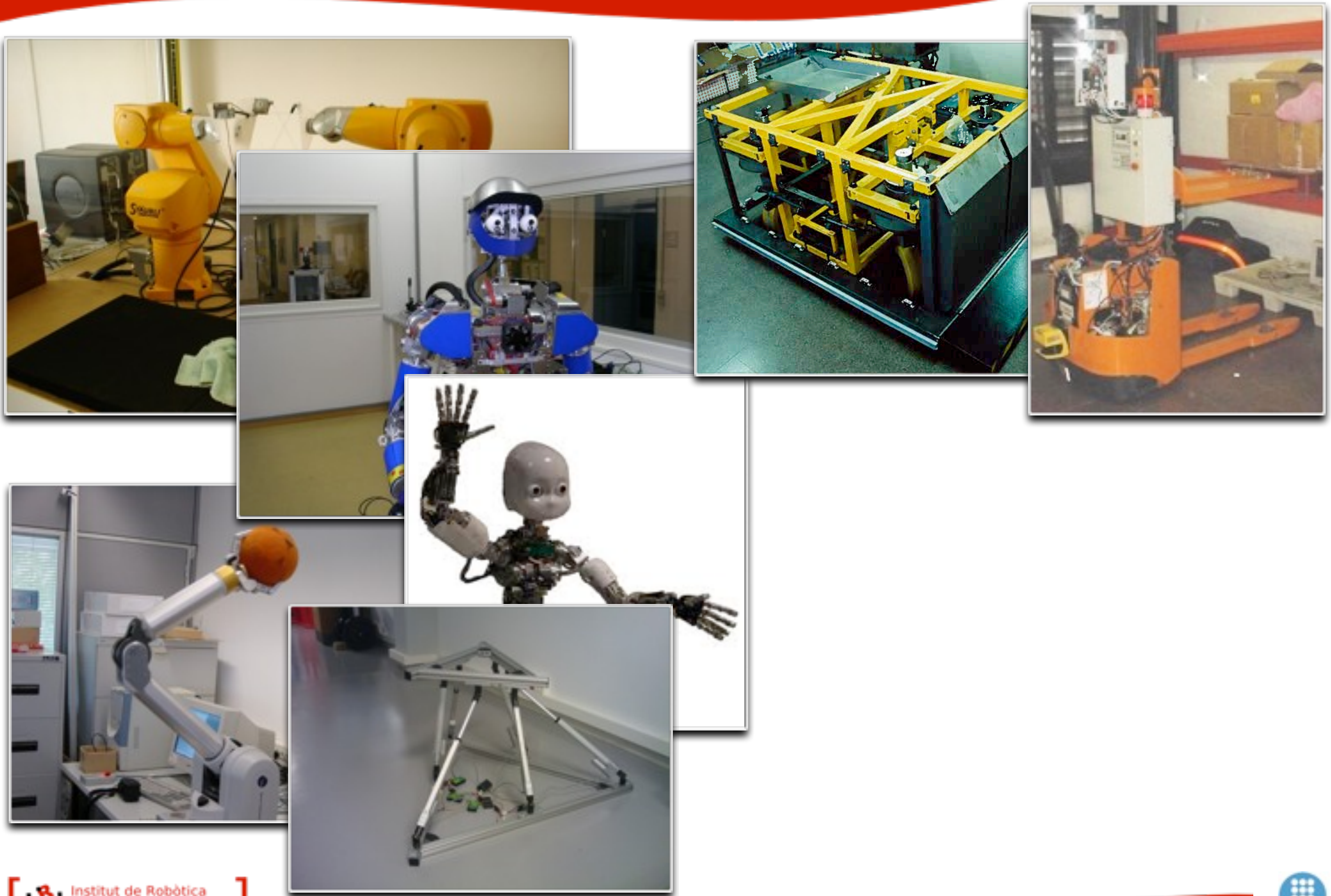
Robots

Robots

Robots



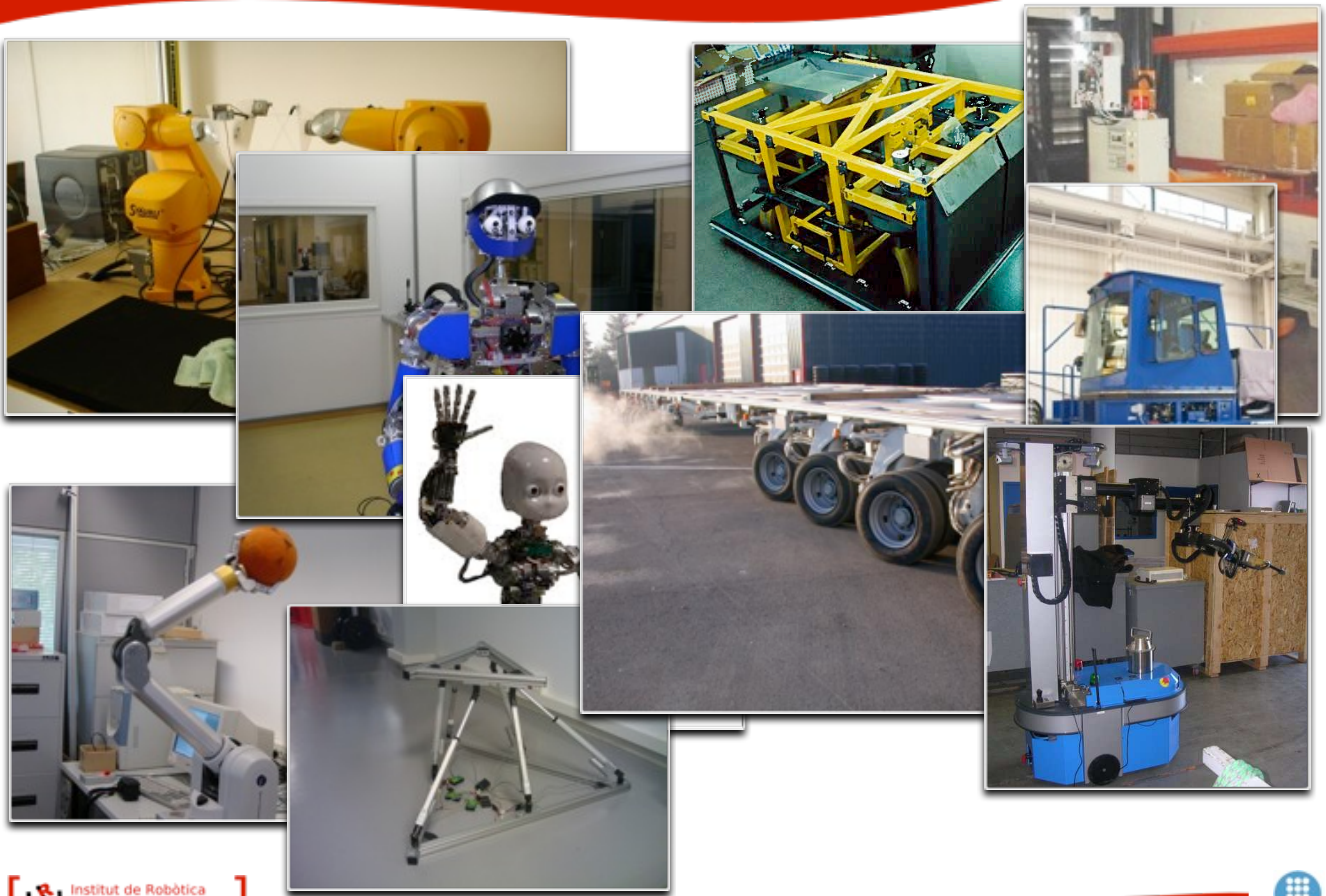
Robots



Robots

Robots

Robots

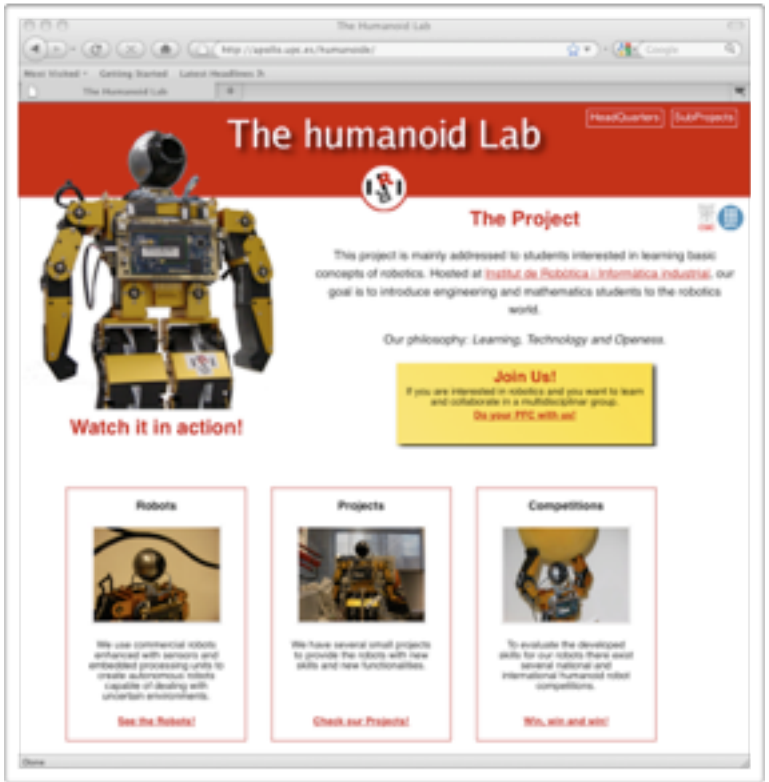


Robots

Robots

Multidisciplinarity

Multidisciplinarity



The humanoid Lab

The Project

This project is mainly addressed to students interested in learning basic concepts of robotics. Hosted at Institut de Robòtica i Informàtica industrial, our goal is to introduce engineering and mathematics students to the robotics world.

Our philosophy: *Learning, Technology and Openess.*

Join Us!
If you are interested in robotics and you want to learn and collaborate in a multidisciplinar group.
Do your PFC with us!

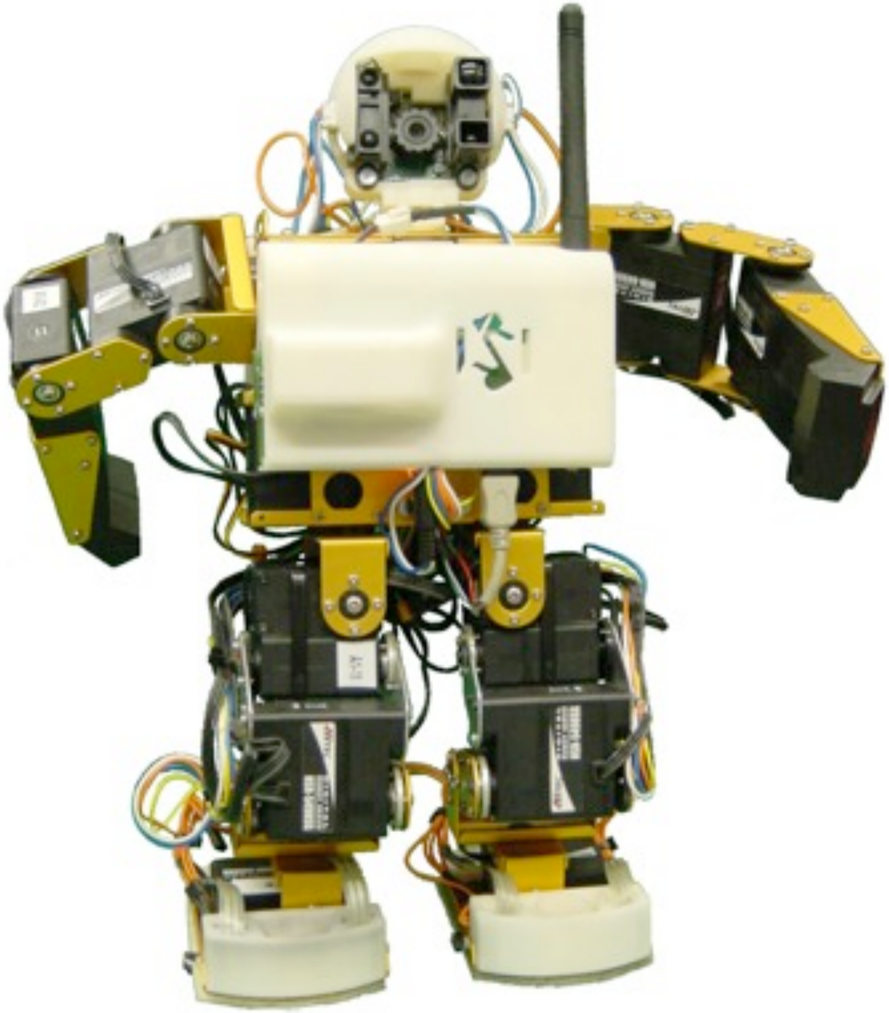
Watch it in action!

Robots
We use commercial robots enhanced with sensors and embedded processing units to create autonomous robots capable of dealing with uncertain environments.
See the Robots!

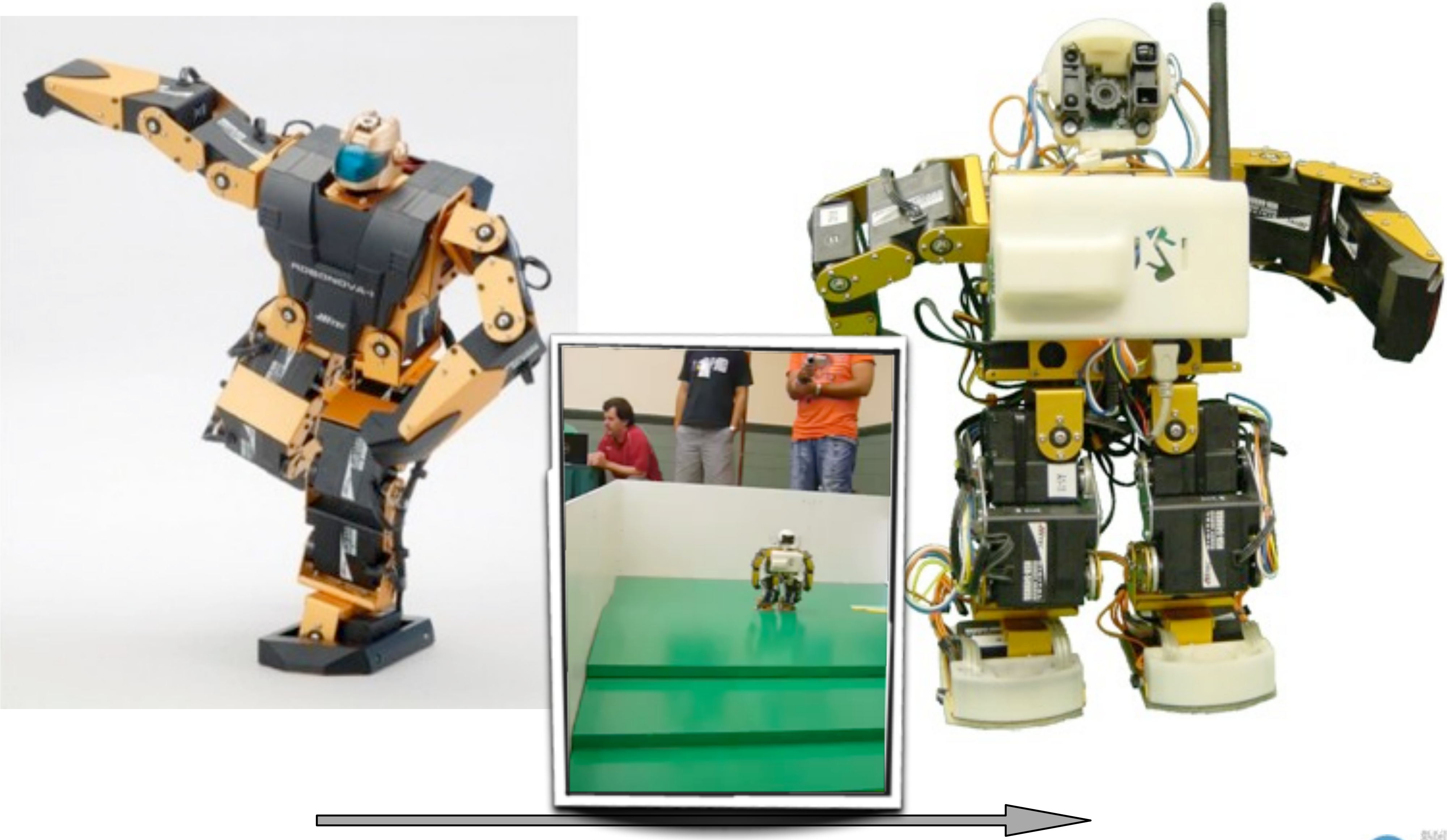
Projects
We have several small projects to provide the robots with new skills and new functionalities.
Check our Projects!

Competitions
To evaluate the developed skills for our robots there exist several national and international humanoid robot competitions.
Win, win and win!

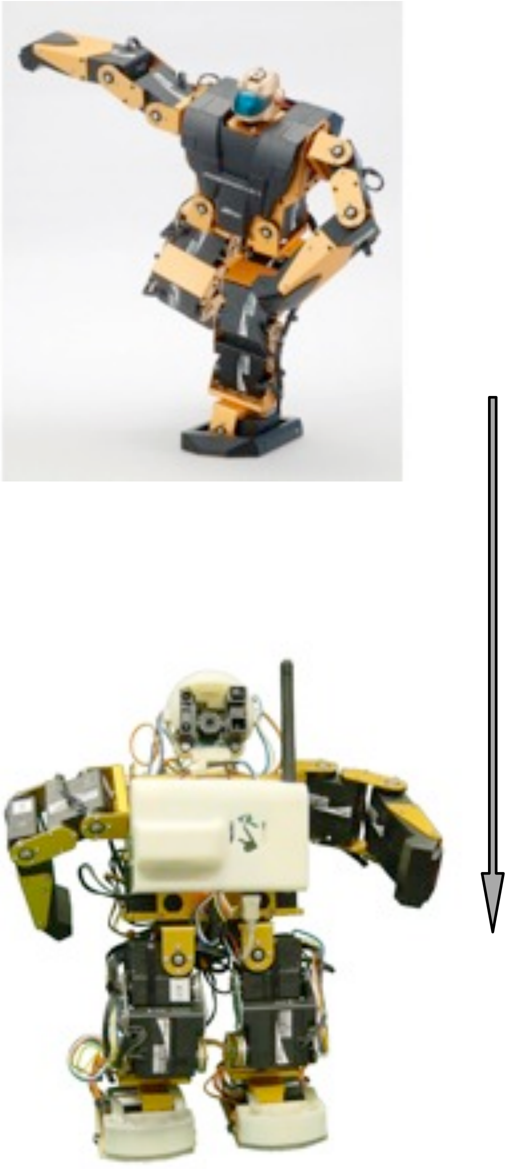
Multidisciplinarity



Multidisciplinarity



Multidisciplinarity



- Engineering:
  - Mechanical
  - Electronic
  - Computer Science
- Mathematics
- Psychology
- Designers
- ...

Definitions

ROBOTICS: *the branch of technology that deals with the design, construction, operation, and application of robots.*

Definitions

ROBOTICS: *the branch of technology that deals with the design, construction, operation, and application of robots.*

ROBOT: *machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer.*

Definitions

ROBOTICS: *the branch of technology that deals with the design, construction, operation, and application of robots.*

ROBOT: *machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer*

Computation in robotics

- Low level

- High level

Computation in robotics

- Low level
  - Middleware/frameworks
    - ICE
    - YARP
    - ROS
- High level

Computation in robotics

- Low level
  - Middleware/frameworks
    - ICE
    - YARP
    - ROS
- High level
  - Libraries
    - Perception: OpenCV, PCL
    - Motion and Planning: KDL, OpenRave
  - Visualization and Simulation platforms

Why middleware?

Why middleware?

- Use others' work

Why middleware?

- Use others' work
- We produce SW (that dies with the robot)

Why middleware?

- Use others' work
- We produce SW (that dies with the robot)
- Hardware/Software diversity

Why middleware?

- Use others' work
- We produce SW (that dies with the robot)
- Hardware/Software diversity
  - Differences in sensors, actuators, bodies

Why middleware?

- Use others' work
- We produce SW (that dies with the robot)
- Hardware/Software diversity
  - Differences in sensors, actuators, bodies

  - Processors (intel, ARM..), OS/ embedded OS, libraries, languages...

Why middleware?

- Use others' work
- We produce SW (that dies with the robot)
- Hardware/Software diversity
  - Differences in sensors, actuators, bodies
  - Processors (intel, ARM..), OS/ embedded OS, libraries, languages...
- Very quick evolution

Modular systems

- Coupled systems: changes in one part trigger changes in another
  - Leads to complexity
  - Systems hard to maintain/evolve

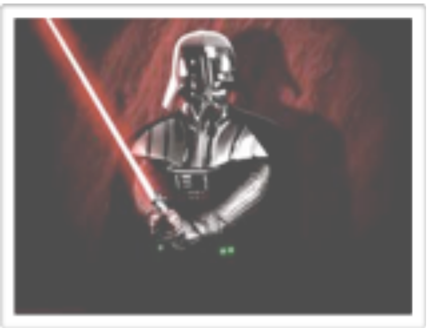
Modular systems

- Coupled systems: changes in one part trigger changes in another
  - Leads to complexity
  - Systems hard to maintain/evolve

This is the path to the Dark Side

Modular systems

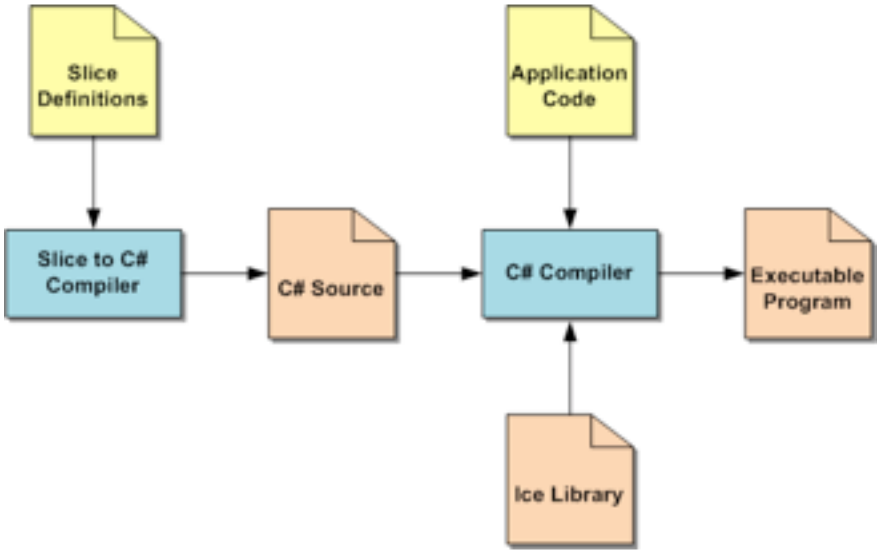
- Coupled systems: changes in one part trigger changes in another
  - Leads to complexity
  - Systems hard to maintain/evolve



This is the path to the Dark Side

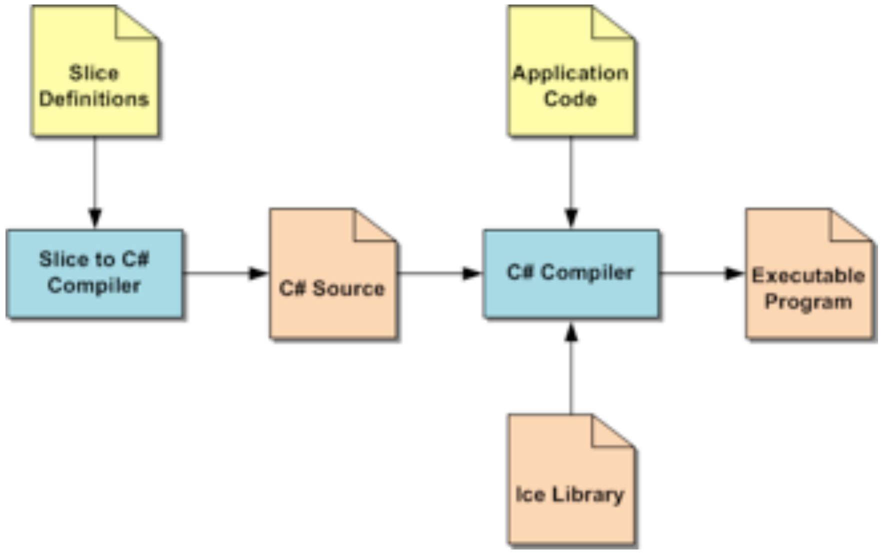
- Modular robots
  - Robot software is hardware-specific and task-specific
  - Hardware and task change quickly
  - Robot complexity is high, so teams of developers are needed

ICE



- GPL License
- C++, Java, .NET, Python, PHP, Ruby, and Objective-C
- Linux, Mac, W$
- Slice (Specification Language for Ice)
- Synchronous and Asynchronous

ICE

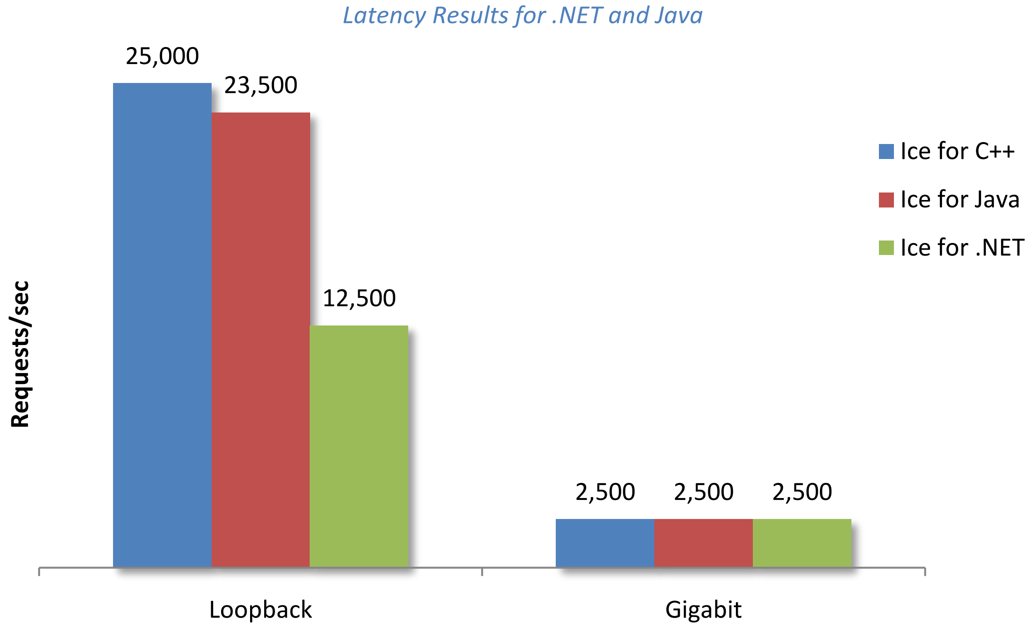


- GPL License
- C++, Java, .NET, Python, PHP, Ruby, and Objective-C
- Linux, Mac, W$
- Slice (Specification Language for Ice)
- Synchronous and Asynchronous

1. Define types and interfaces with Slice
2. Compile the Slice definitions into source code for your chosen programming language
3. Write client-side application code and compile it—together with the code generated by the Slice compiler—into a client program.
4. Write server-side application code and compile it—together with the code generated by the Slice compiler—into a server program.

Tuesday, July 19, 2011

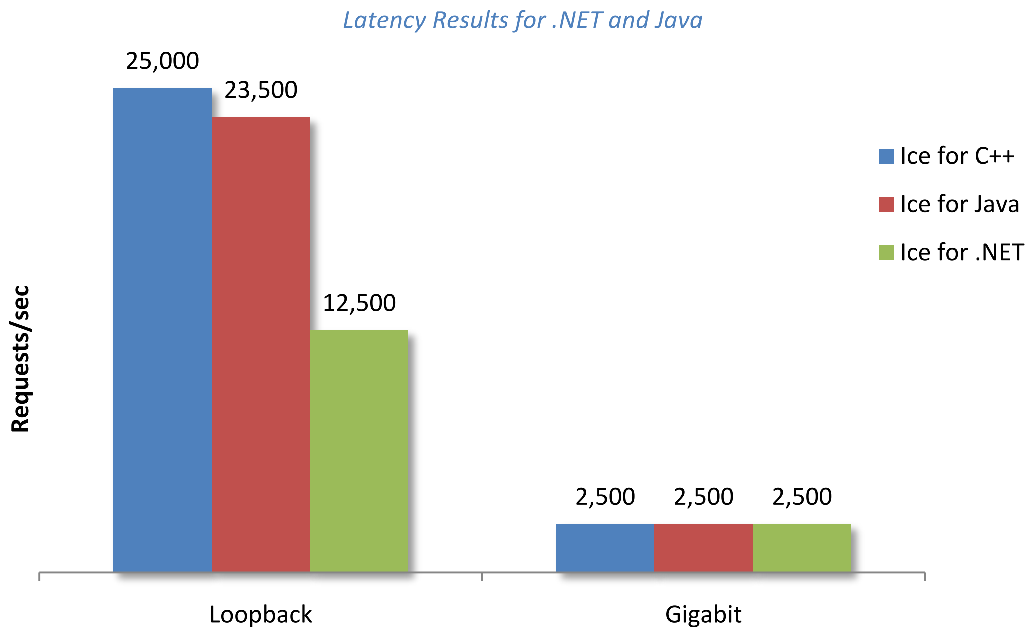
ICE Performance



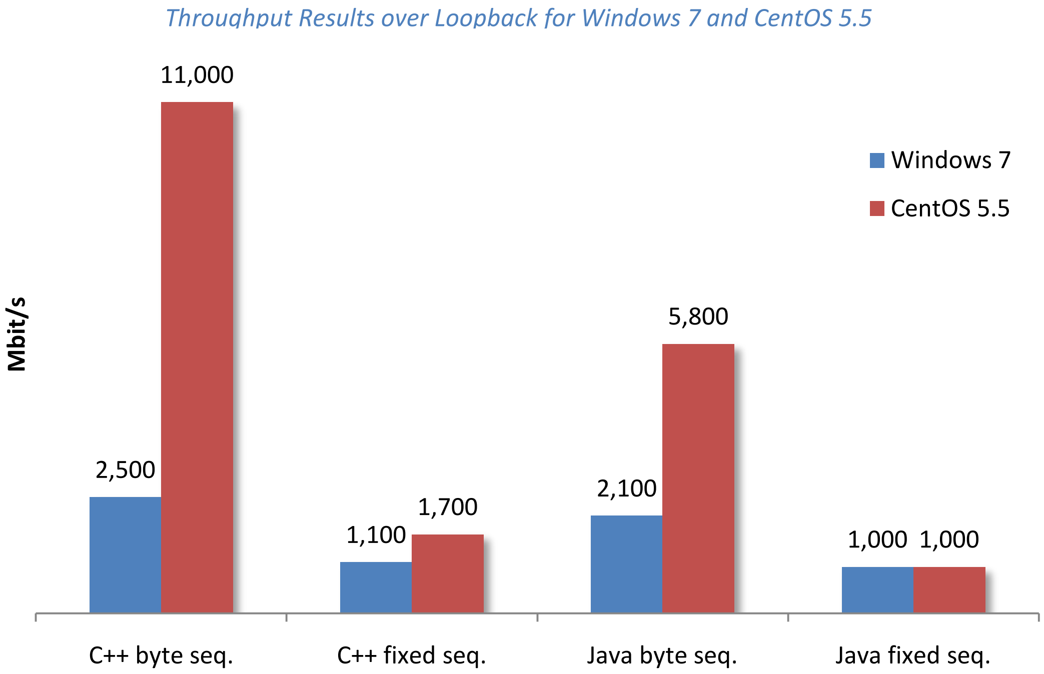
If processes are in the same machine don't use ICE!

M. Henning and M. Spruiell, Choosing Middleware: Why Performance and Scalability do (and do not) Matter, ZeroC Inc.

ICE Performance



If processes are in the same machine don't use ICE!



M. Henning and M. Spruiell, Choosing Middleware: Why Performance and Scalability do (and do not) Matter, ZeroC Inc.

YARP

YARP

- LGPL license
- Portability:

CMake

SWIG

YARP

YARP

- Abstract details of data flow (keep algorithm and “plumbing” separate)
  - Observer design pattern
  - “port” objects deliver data to any number of observers
  - in any number of processes
  - distributed across any number of computers/OSes
  - using several communications protocols

- LGPL license
- Portability:

CMake

SWIG

YARP

- Abstract details of data flow (keep algorithm and "plumbing" separate)
  - Observer design pattern
  - "port" objects deliver data to any number of observers
  - in any number of processes
  - distributed across any number of computers/OSes
  - using several communications protocols
- Abstract details of used devices from program source code (easy to replace)
  - Implement specific drivers
  - Define device families
  - Implement network wrappers

- LGPL license
- Portability:

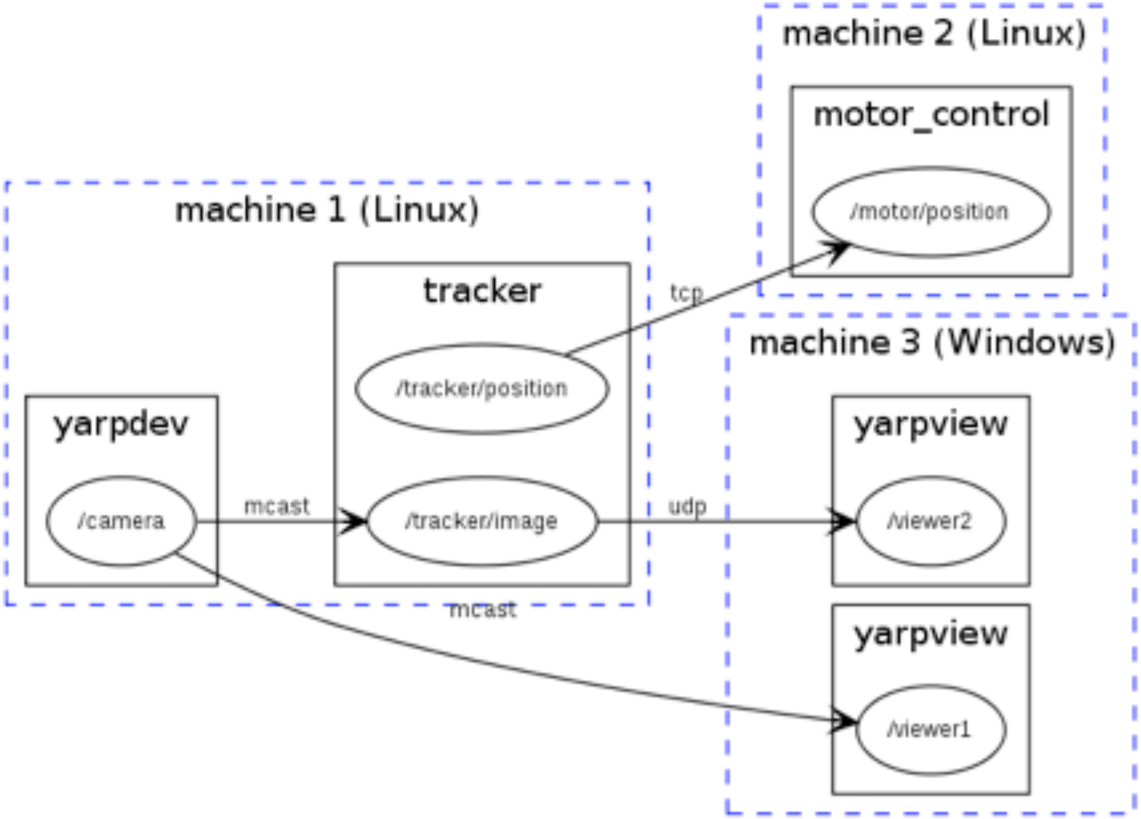
CMake

SWIG

YARP: Interfacing libraries and devices

- Organization:
  - libYARP_OS - interfacing with the operating system(s) to support easy streaming of data across many threads across many machines. YARP uses the open-source ACE (ADAPTIVE Communication Environment) library, which is portable across a very broad range of environments, and YARP inherits that portability. YARP is written almost entirely in C++.
  - libYARP_sig - performing common signal processing tasks (visual, auditory) in an open manner easily interfaced with other commonly used libraries, for example OpenCV.
  - libYARP_dev - interfacing with common devices used in robotics: framegrabbers, digital cameras, motor control boards, etc.

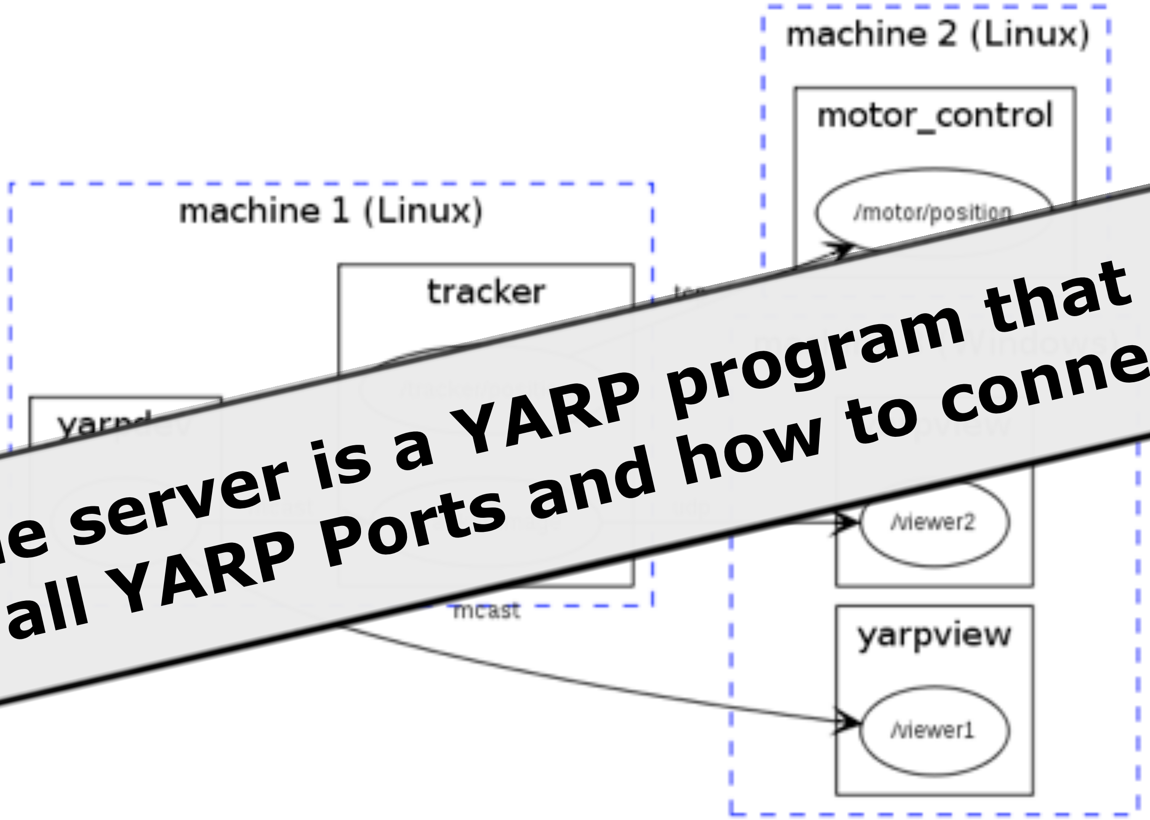
YARP - ports



Ports can be on different machines and OSes

- Connections use different protocols
- Ports belong to processes
- Processes can be on different machines/OS

YARP - ports



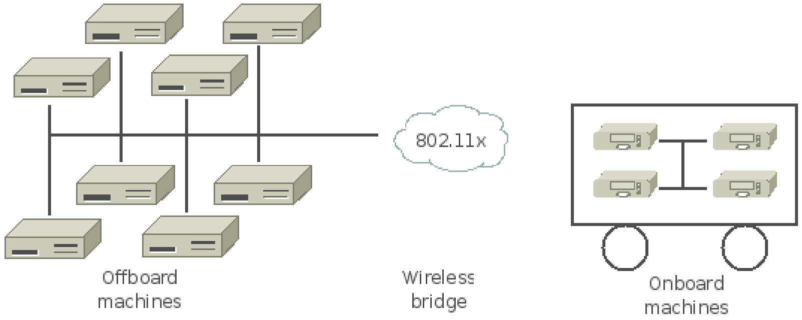
The name server is a YARP program that maintains a list of all YARP Ports and how to connect to them.

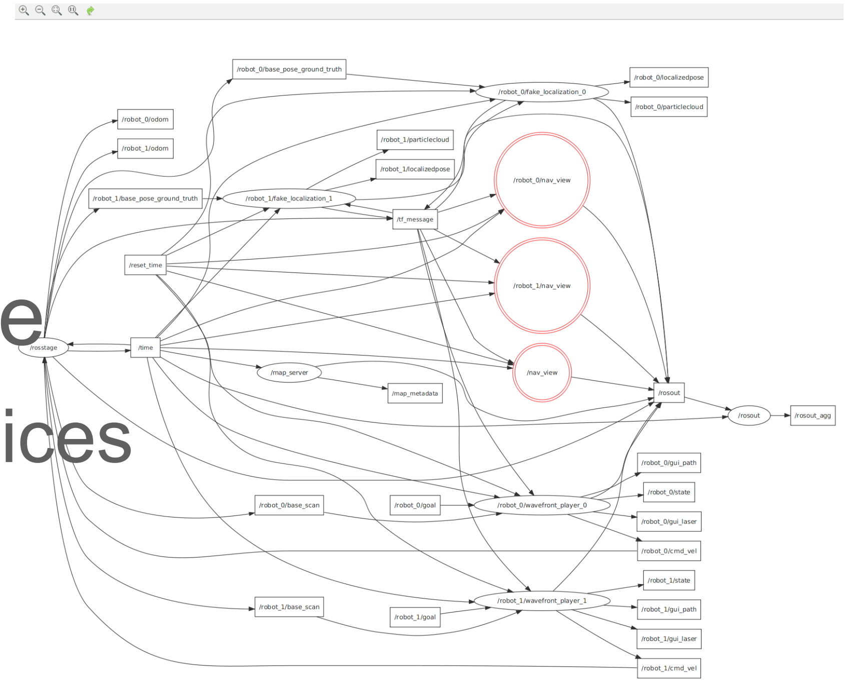
Ports can be on different machines and OSes

- Connections use different protocols
- Ports belong to processes
- Processes can be on different machines/OS

ROS

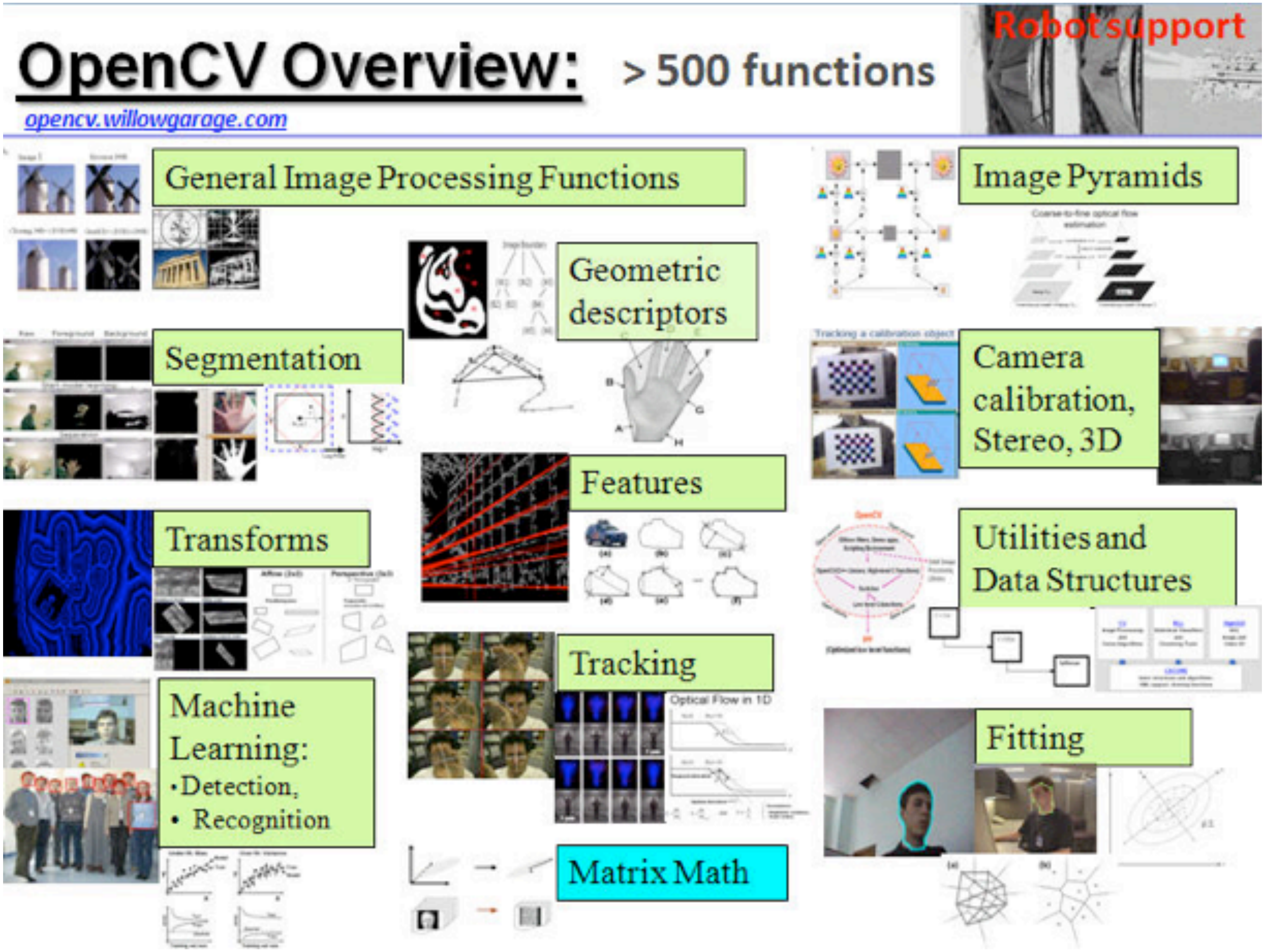
- Middleware -- robotics suite
  - Peer-to-peer: Parameter server
  - Tools-based: Logging, bags, graphs ...
  - Multi-lingual
  - Thin
  - Free and Open-Source
- Communications infrastructure
  - Nodes, messages, topics, services
- Applications packages
  - Interface with other libraries





Institut de Robòtica i Informàtica Industrial

OpenCV



OpenCV Overview: > 500 functions

opencv.willowgarage.com

Robot support

General Image Processing Functions

Image Pyramids

Geometric descriptors

Segmentation

Camera calibration, Stereo, 3D

Features

Transforms

Utilities and Data Structures

Tracking

Machine Learning:

- Detection,
- Recognition

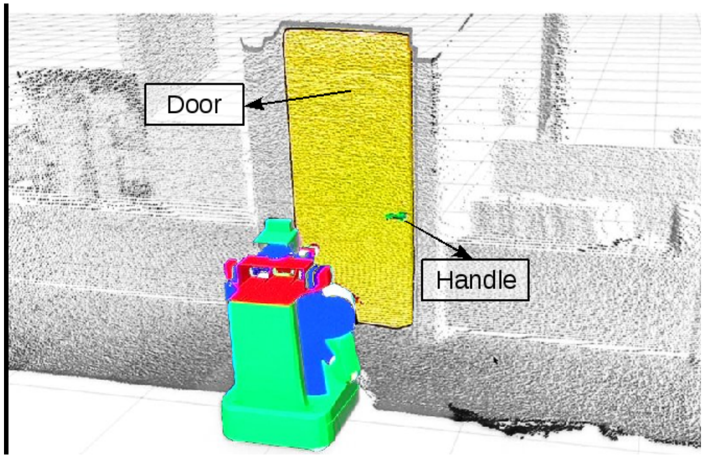
Fitting

Matrix Math

Coarse-to-fine optical flow estimation
Tracking a calibration object
Optical Flow in 1D
Affine (2x3)
Perspective (3x3)
OpenCV

PCL

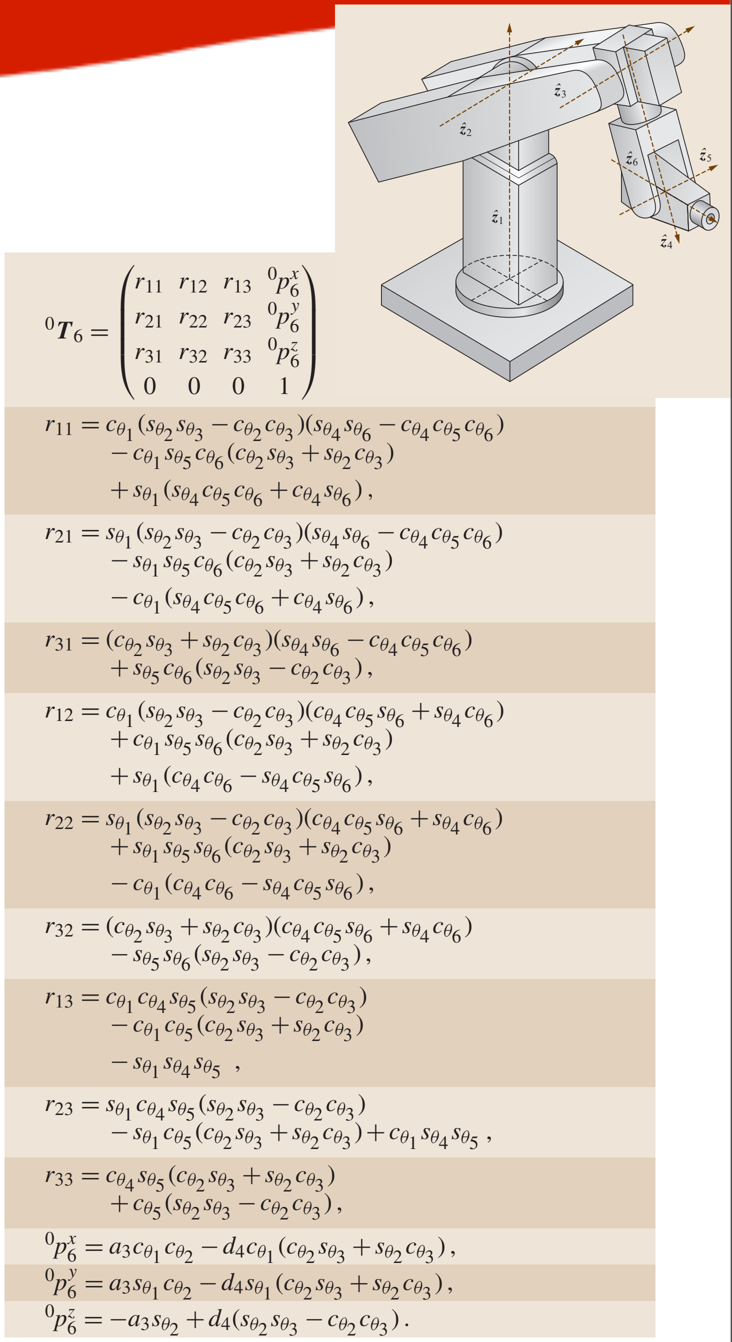
- ...split into a collection of smaller, modular C++ libraries:
  - libpcl_keypoints: nD interest points
  - libpcl_features: nD feature descriptors
  - libpcl_surface: surface meshing/reconstruction techniques
  - libpcl_filters: point cloud data filters and smoothing
  - libpcl_io:I/O operations,3D camera drivers(e.g.,Kinect)
  - libpcl_kdtree: fast nearest neighbor operations
  - libpcl_octree: downsampling,compression,change detection
  - libpcl_range_image: efficient 3D operations
  - libpcl_sample_consensus: RANSAC,MSAC,MLESAC, planes, spheres, etc
  - libpcl_segmentation: model segmentation operations
  - libpcl_registration: point cloud registration methods
  - libpcl_visualization: 2D/3D visualization library



KDL

Extensive support for :

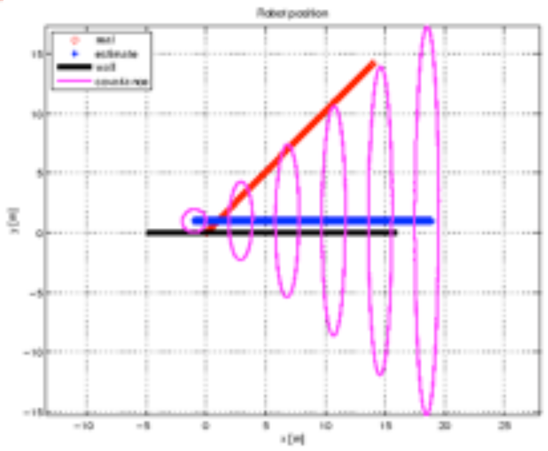
- Geometric primitives: point, frame, twist ...
- Kinematic Chains: serial and tree structures (D-H parameters)
- Kinematic Solvers: various generic forward and inverse kinematic algorithms, redundancy resolution, ...
- Motion Trajectories: Cartesian paths, velocity profiles, Cartesian trajectories



$$
{}^0\boldsymbol{T}_6 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & {}^0p_6^x \\ r_{21} & r_{22} & r_{23} & {}^0p_6^y \\ r_{31} & r_{32} & r_{33} & {}^0p_6^z \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

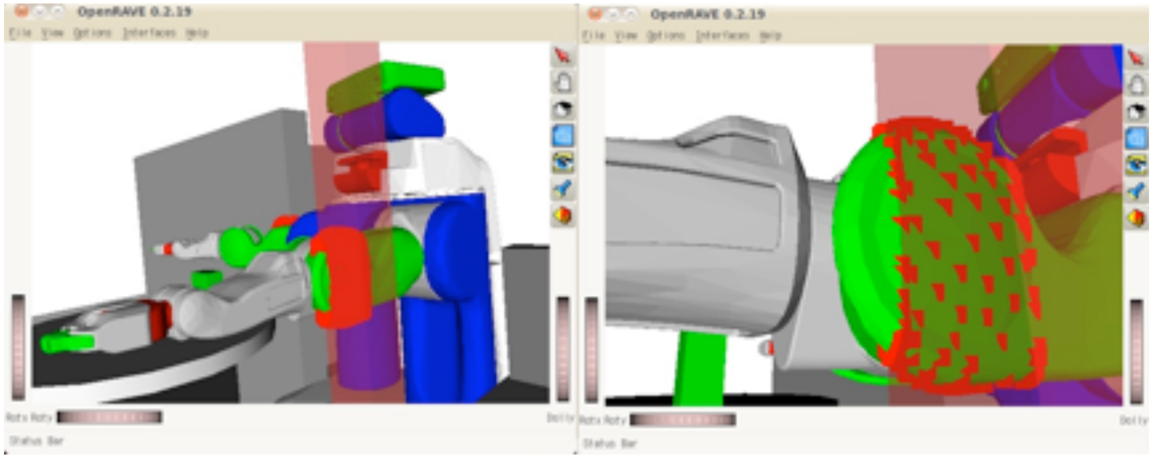
$$
\begin{aligned}
r_{11} = {} & c_{\theta_1}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})(s_{\theta_4}s_{\theta_6} - c_{\theta_4}c_{\theta_5}c_{\theta_6}) \\
& - c_{\theta_1}s_{\theta_5}c_{\theta_6}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& + s_{\theta_1}(s_{\theta_4}c_{\theta_5}c_{\theta_6} + c_{\theta_4}s_{\theta_6})\,, \\
r_{21} = {} & s_{\theta_1}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})(s_{\theta_4}s_{\theta_6} - c_{\theta_4}c_{\theta_5}c_{\theta_6}) \\
& - s_{\theta_1}s_{\theta_5}c_{\theta_6}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& - c_{\theta_1}(s_{\theta_4}c_{\theta_5}c_{\theta_6} + c_{\theta_4}s_{\theta_6})\,, \\
r_{31} = {} & (c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3})(s_{\theta_4}s_{\theta_6} - c_{\theta_4}c_{\theta_5}c_{\theta_6}) \\
& + s_{\theta_5}c_{\theta_6}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})\,, \\
r_{12} = {} & c_{\theta_1}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})(c_{\theta_4}c_{\theta_5}s_{\theta_6} + s_{\theta_4}c_{\theta_6}) \\
& + c_{\theta_1}s_{\theta_5}s_{\theta_6}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& + s_{\theta_1}(c_{\theta_4}c_{\theta_6} - s_{\theta_4}c_{\theta_5}s_{\theta_6})\,, \\
r_{22} = {} & s_{\theta_1}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})(c_{\theta_4}c_{\theta_5}s_{\theta_6} + s_{\theta_4}c_{\theta_6}) \\
& + s_{\theta_1}s_{\theta_5}s_{\theta_6}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& - c_{\theta_1}(c_{\theta_4}c_{\theta_6} - s_{\theta_4}c_{\theta_5}s_{\theta_6})\,, \\
r_{32} = {} & (c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3})(c_{\theta_4}c_{\theta_5}s_{\theta_6} + s_{\theta_4}c_{\theta_6}) \\
& - s_{\theta_5}s_{\theta_6}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})\,, \\
r_{13} = {} & c_{\theta_1}c_{\theta_4}s_{\theta_5}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3}) \\
& - c_{\theta_1}c_{\theta_5}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& - s_{\theta_1}s_{\theta_4}s_{\theta_5}\ , \\
r_{23} = {} & s_{\theta_1}c_{\theta_4}s_{\theta_5}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3}) \\
& - s_{\theta_1}c_{\theta_5}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) + c_{\theta_1}s_{\theta_4}s_{\theta_5}\,, \\
r_{33} = {} & c_{\theta_4}s_{\theta_5}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3}) \\
& + c_{\theta_5}(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})\,, \\
{}^0p_6^x = {} & a_3c_{\theta_1}c_{\theta_2} - d_4c_{\theta_1}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3})\,, \\
{}^0p_6^y = {} & a_3s_{\theta_1}c_{\theta_2} - d_4s_{\theta_1}(c_{\theta_2}s_{\theta_3} + s_{\theta_2}c_{\theta_3})\,, \\
{}^0p_6^z = {} & -a_3s_{\theta_2} + d_4(s_{\theta_2}s_{\theta_3} - c_{\theta_2}c_{\theta_3})\,.
\end{aligned}
$$

BFL - OpenRave



- The Bayesian Filtering Library (BFL):
  - (Extended) Kalman Filters,
  - Particle Filters (or Sequential Monte Carlo methods), etc.

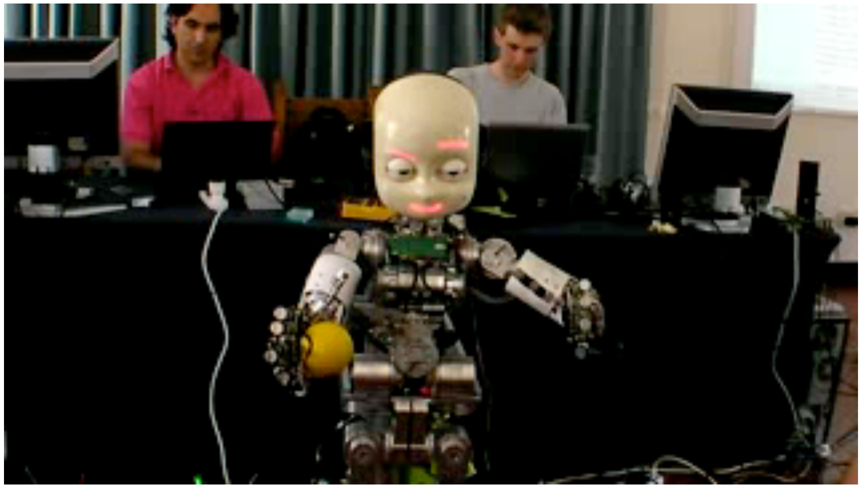
- OpenRAVE
  - IK
  - Motion planners
  - Real time - industrial



Visualization - replay

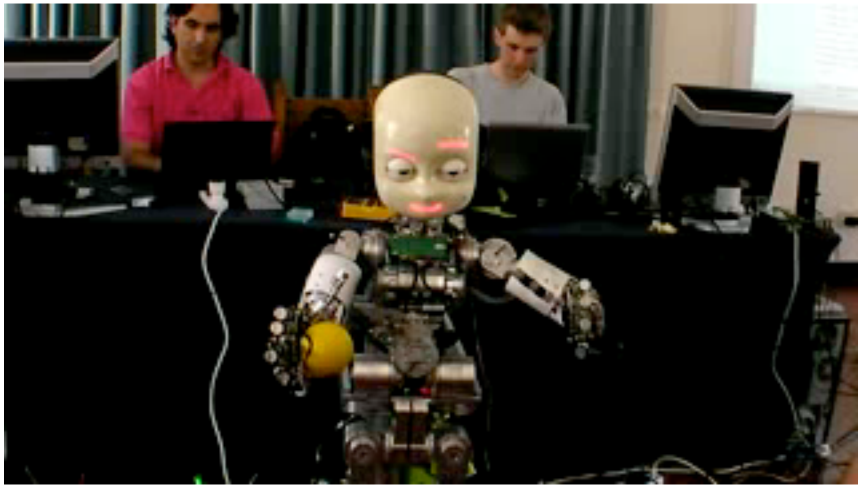
Simulation

- Realism
  - Kinematics of motion
  - Dynamics of motion
  - Environment, i.e, gravity
  - Sensor simulation
  - Transparent to the user



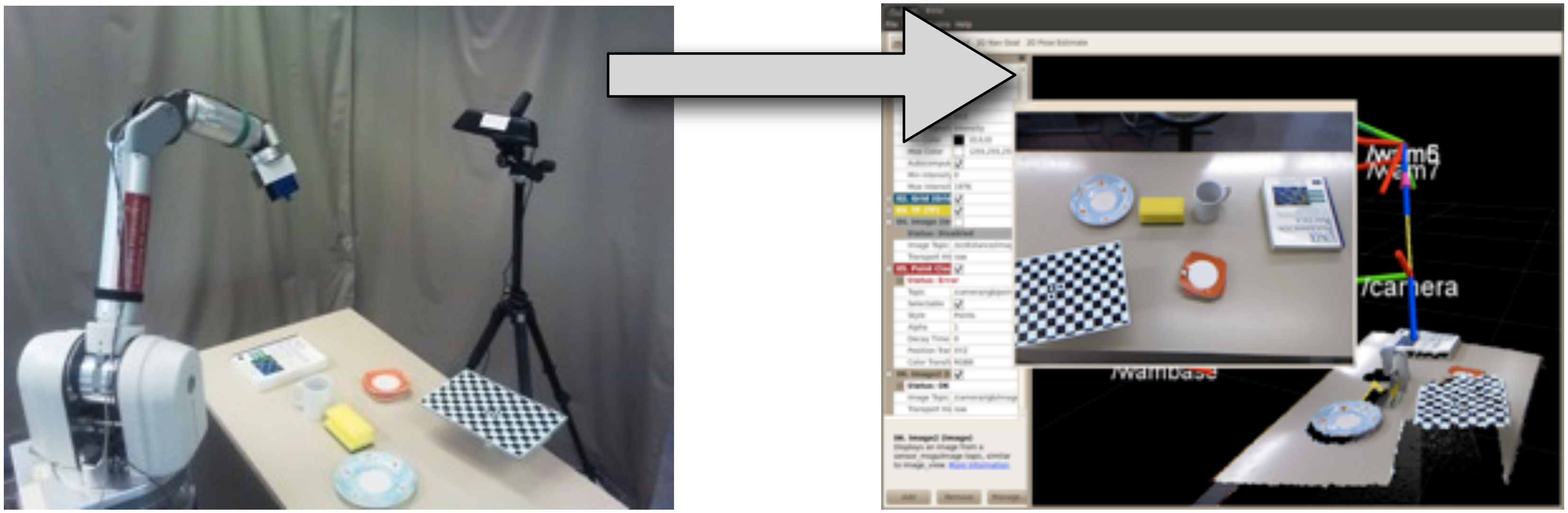
Simulation

- Realism
  - Kinematics of motion
  - Dynamics of motion
  - Environment, i.e, gravity
  - Sensor simulation
  - Transparent to the user

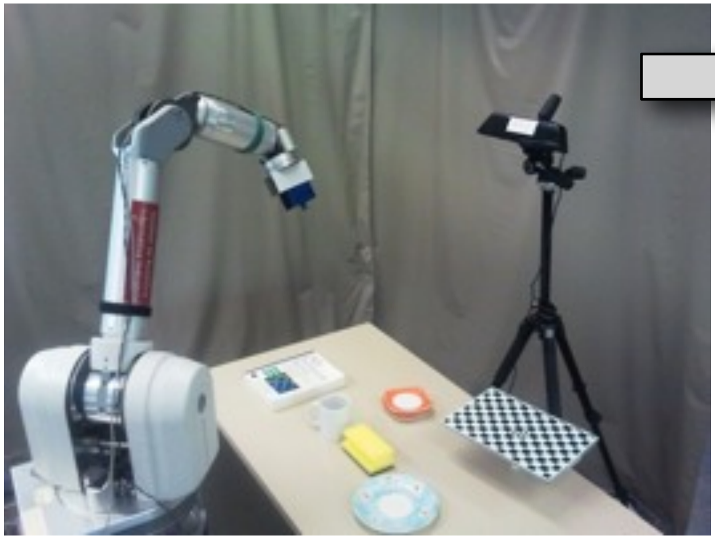


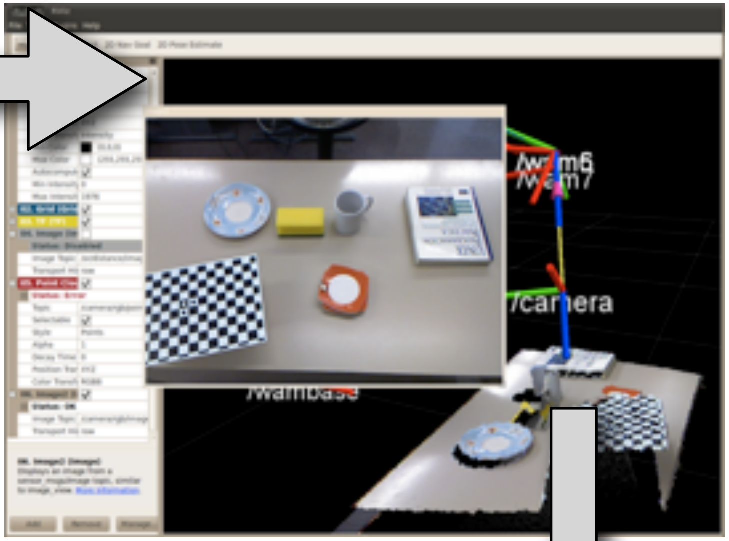
The perception-action loop

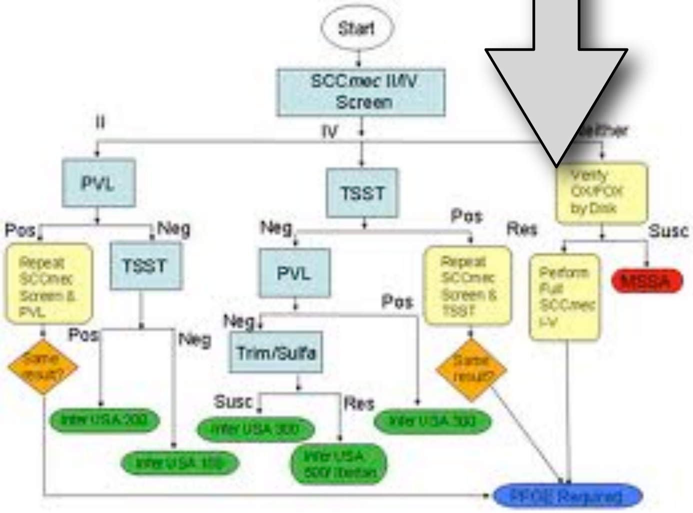
The perception-action loop



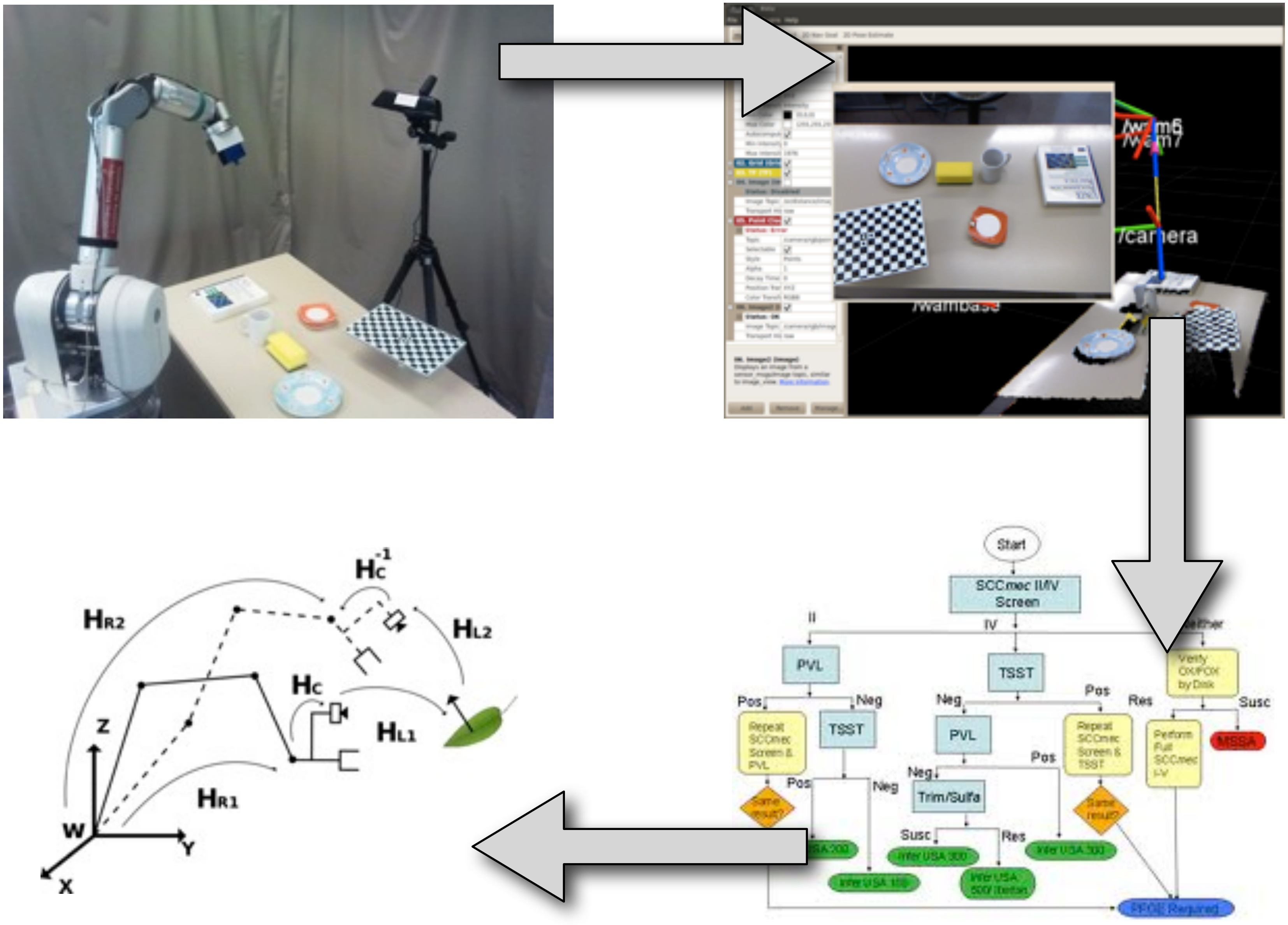
The perception-action loop



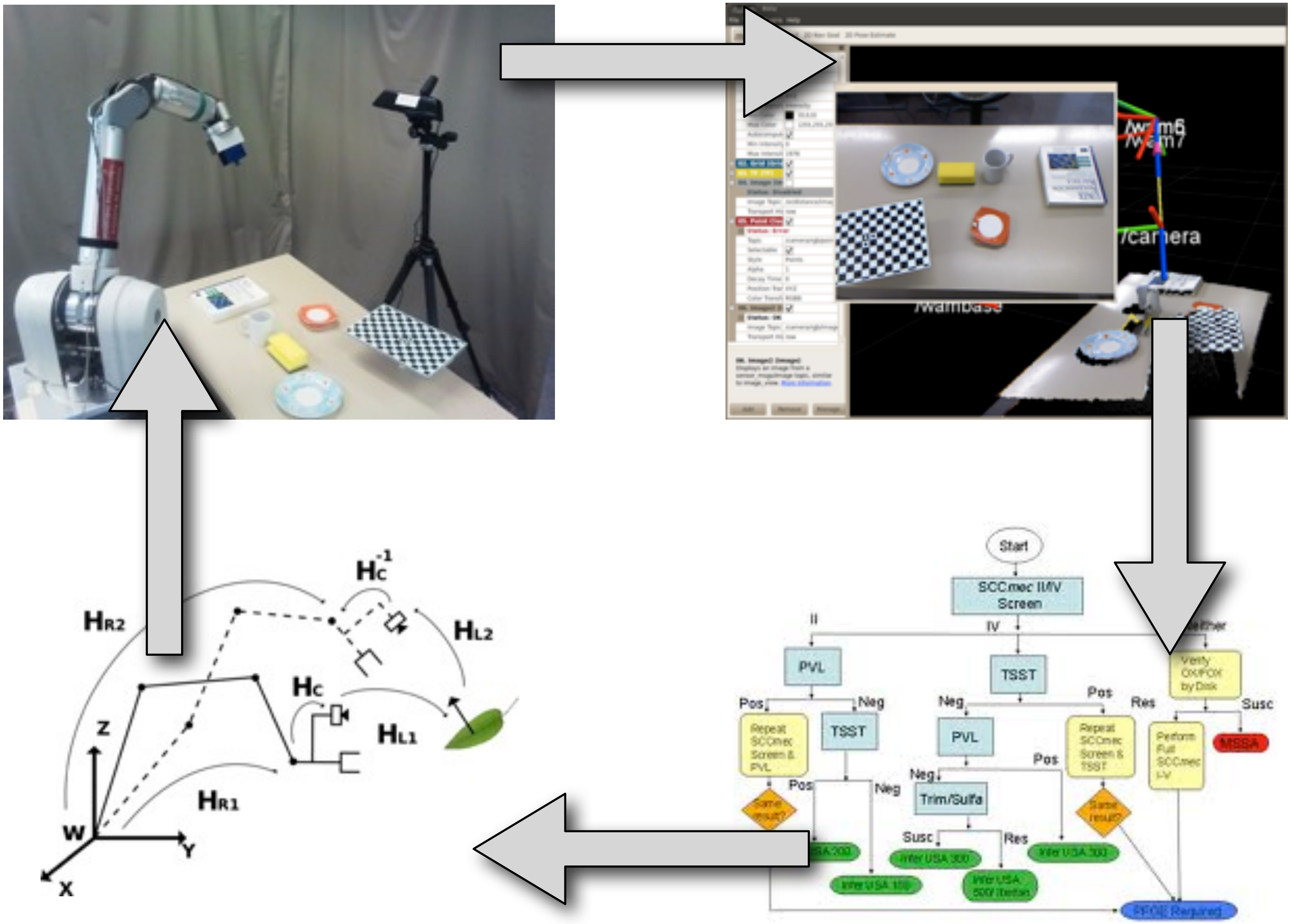




The perception-action loop



The perception-action loop



Perception

- Force/tactile
- Inertial/GPS
- Sonar
- Range
- Vision/3d
- ...

Perception

- Force/tactile
- Inertial/GPS
- Sonar
- Range
- Vision/3d
- ...

Perception

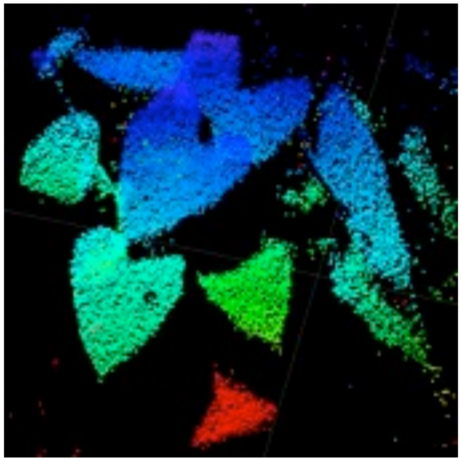
- Force/tactile
- Inertial/GPS
- Sonar
- Range
- Vision/3d
- ...

Perception

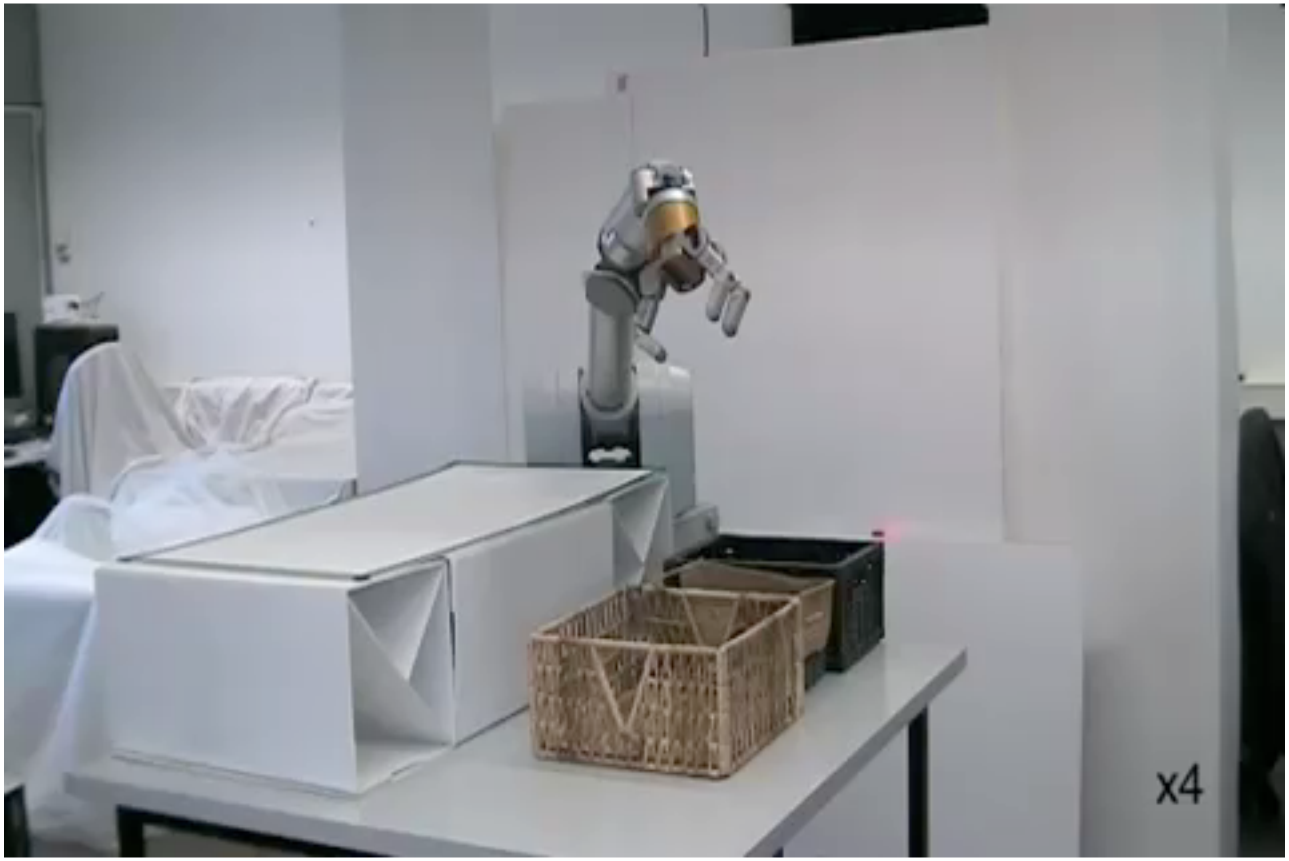
- Force/tactile
- Inertial/GPS
- Sonar
- Range
- Vision/3d
- ...

Perception

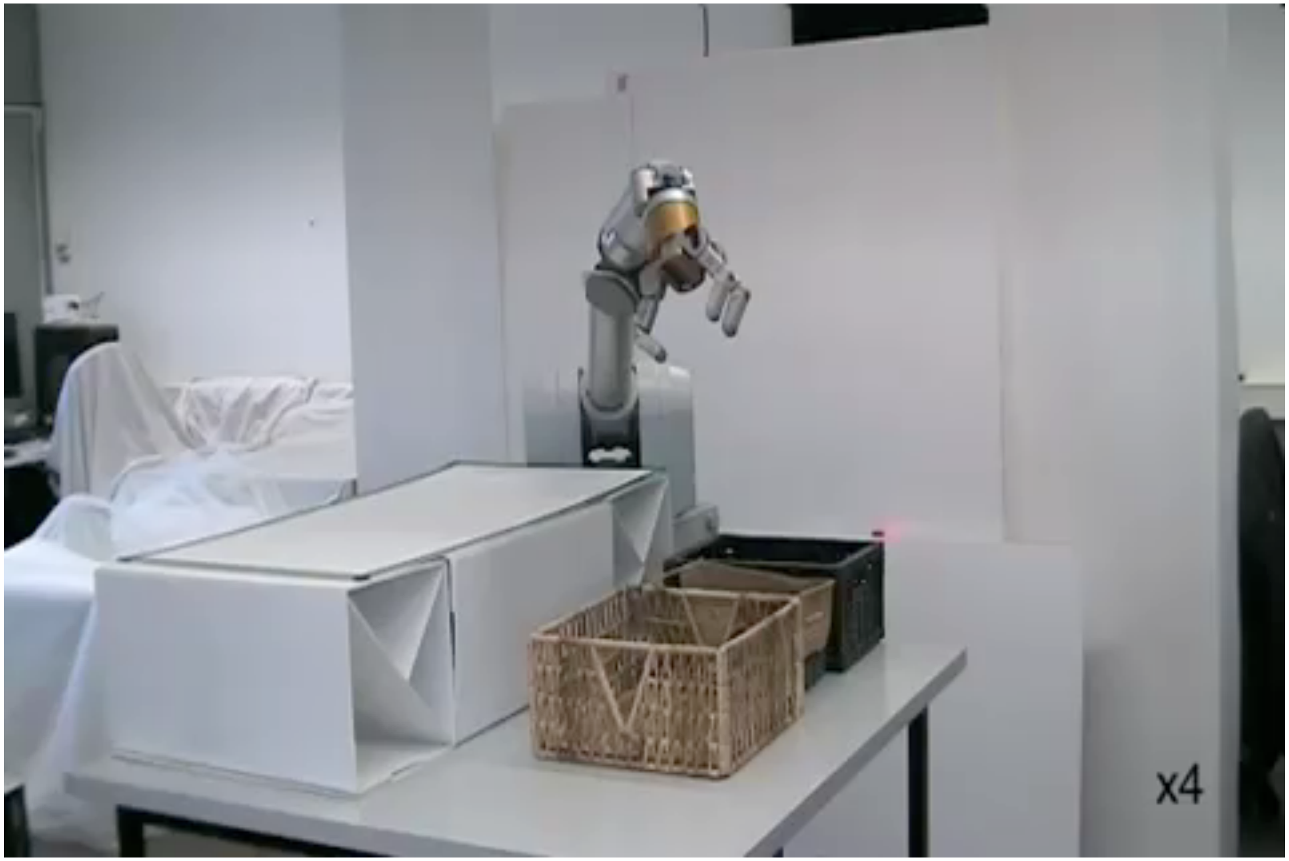
- Force/tactile
- Inertial/GPS
- Sonar
- Range
- Vision/3d
- ...



Rigid object manipulation



Rigid object manipulation



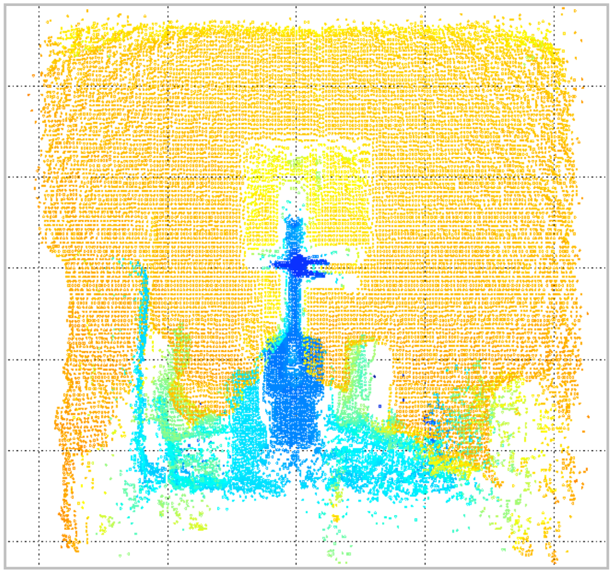
- Calibration: models
- Previous knowledge about objets: models

Calibration

- Hand-eye calibration

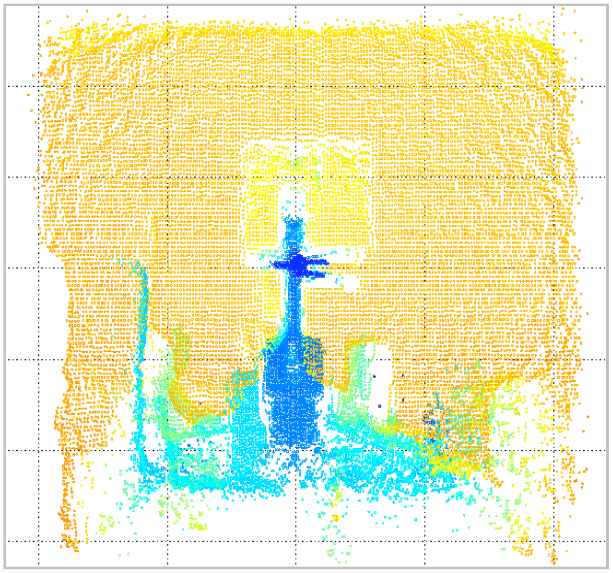
Avoid calibration

- POMDP based navigation



Avoid calibration

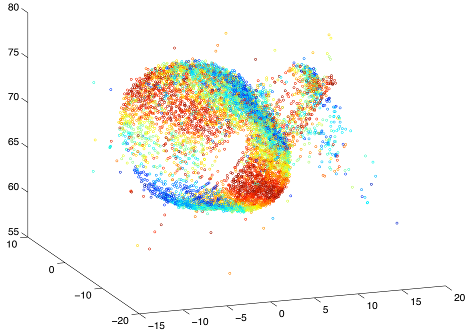
- POMDP based navigation



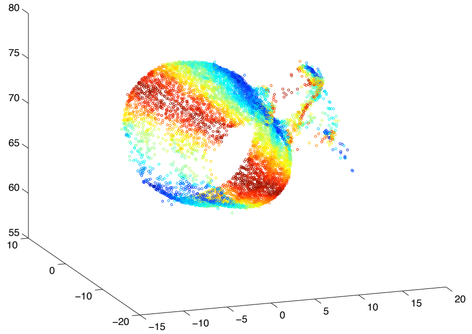
In can be always used?

Avoid previous knowledge

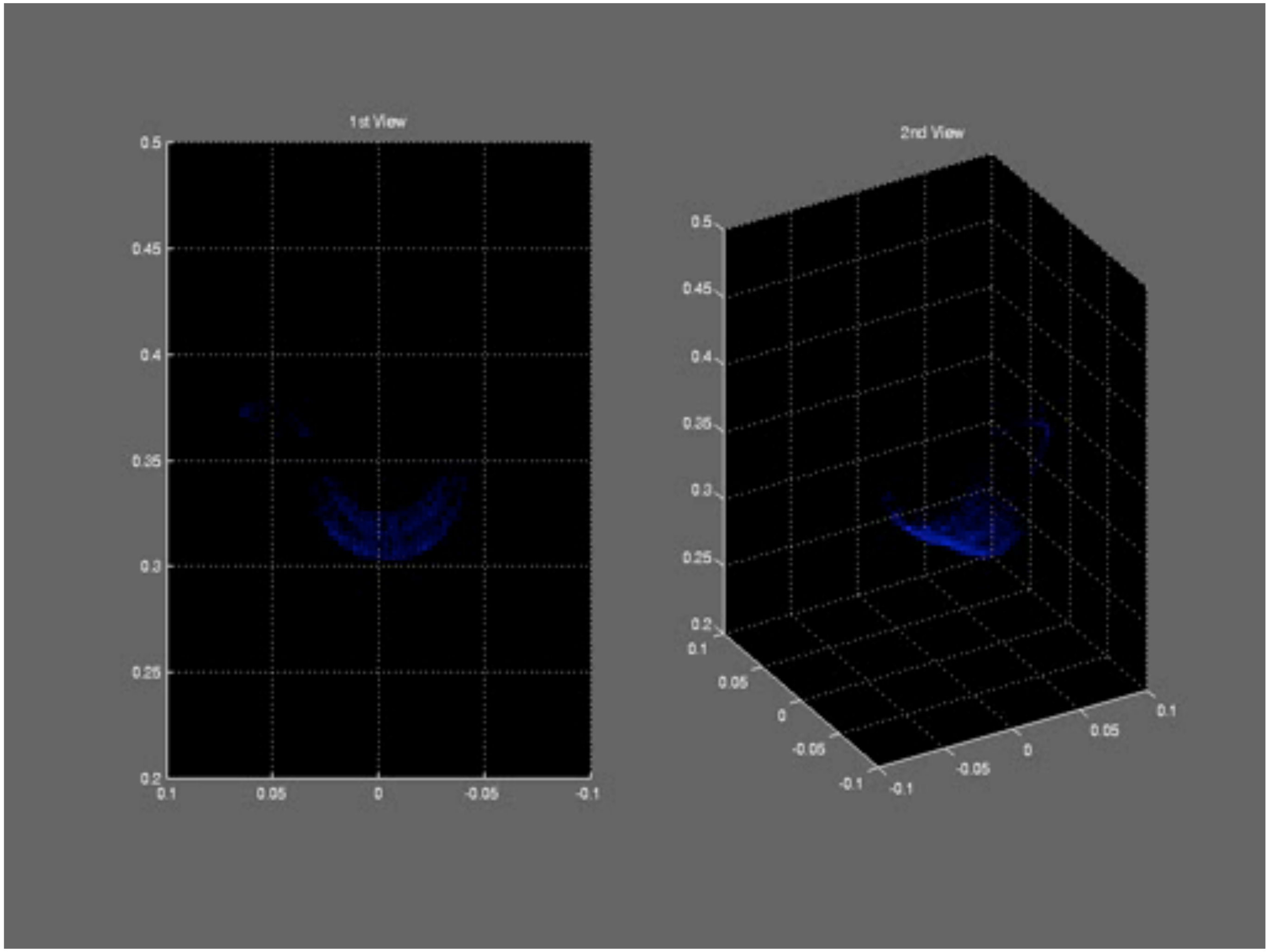
- Active modelling



(a) Fine registration



(b) Result model B



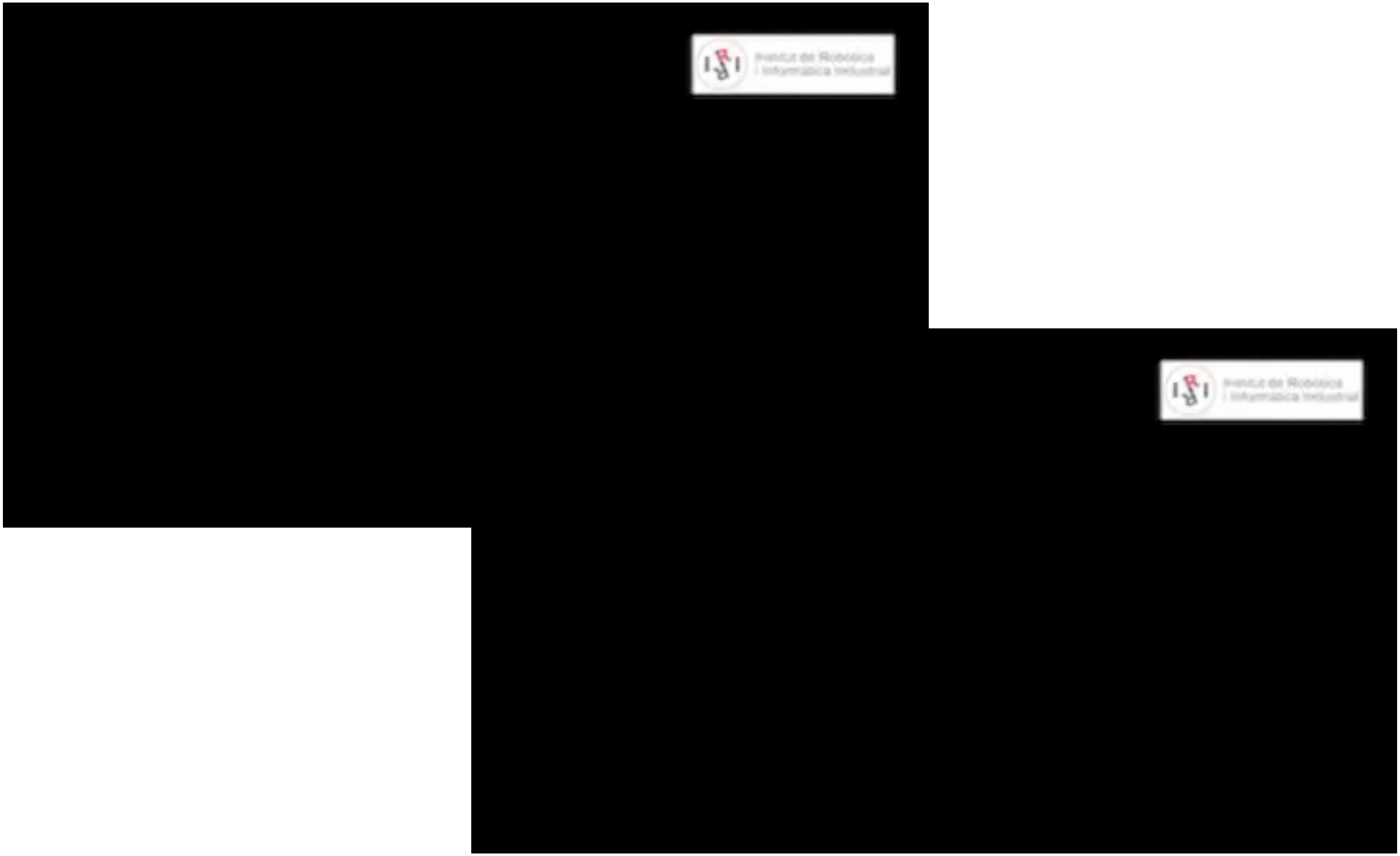
Planning on deformable objects

Plant modelling

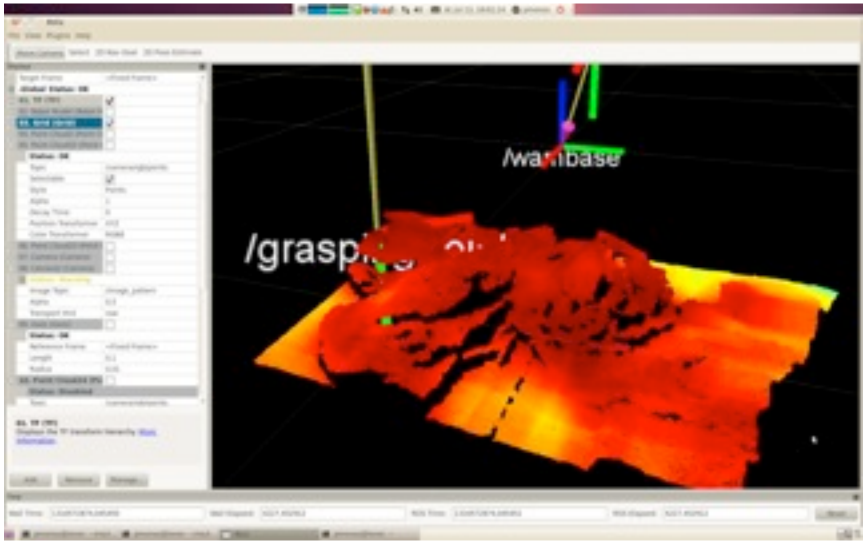
- Next best view - camera motion planning

Leaf probing

Leaf probing



Manipulation of textiles



Opportunities

Opportunities

- USA: New prog. 500M$ (NASA, NSF, DARPA)

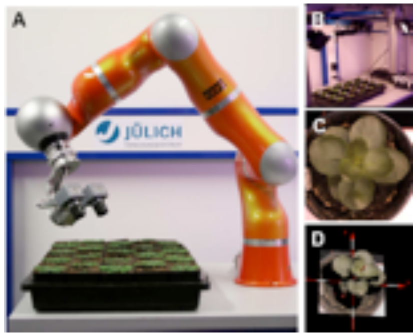
Opportunities

- USA: New prog. 500M$ (NASA, NSF, DARPA)
- Korea: robotics at school (45M$)

Opportunities

- USA: New prog. 500M$ (NASA, NSF, DARPA)
- Korea: robotics at school (45M$)
- European Projects (9050 M€)
  - Cooperation with Eu labs

SEVENTH FRAMEWORK PROGRAMME



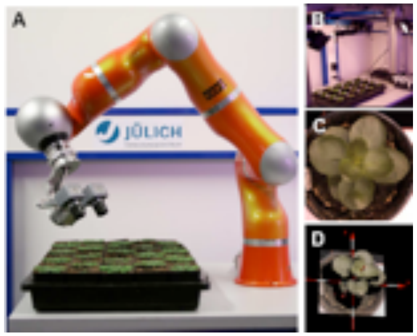
Garnics

IntellAct

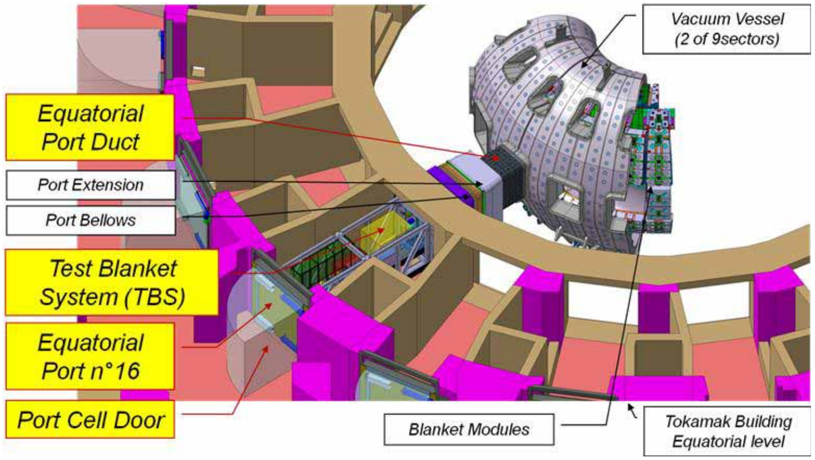
Opportunities

- USA: New prog. 500M$ (NASA, NSF, DARPA)
- Korea: robotics at school (45M$)
- European Projects (9050 M€)
  - Cooperation with Eu labs
- ITER - Barcelona

SEVENTH FRAMEWORK PROGRAMME



Garnics



IntellAct

Robotics

Institut
de Robòtica
i Informàtica
Industrial

Guillem Alenyà

Fronteras de la Computación, 2011

Questions?

- SO
- Real time
- Computer architectures
- Languages
- Communication buses
- Real robots - Fukushima, Roomba...
- Seguridad: robots entre humanos