the iCub project
an open source platform for research in embodied cognition

MIT

May 9th, 2012

Giorgio Metta

Cognitive Humanoids Laboratory
Robotics, Brain and Cognitive Sciences dept.
the Italian Institute of Technology

iit ISTITUTO ITALIANO DI TECNOLOGIA

Cogsys Cognitive Systems

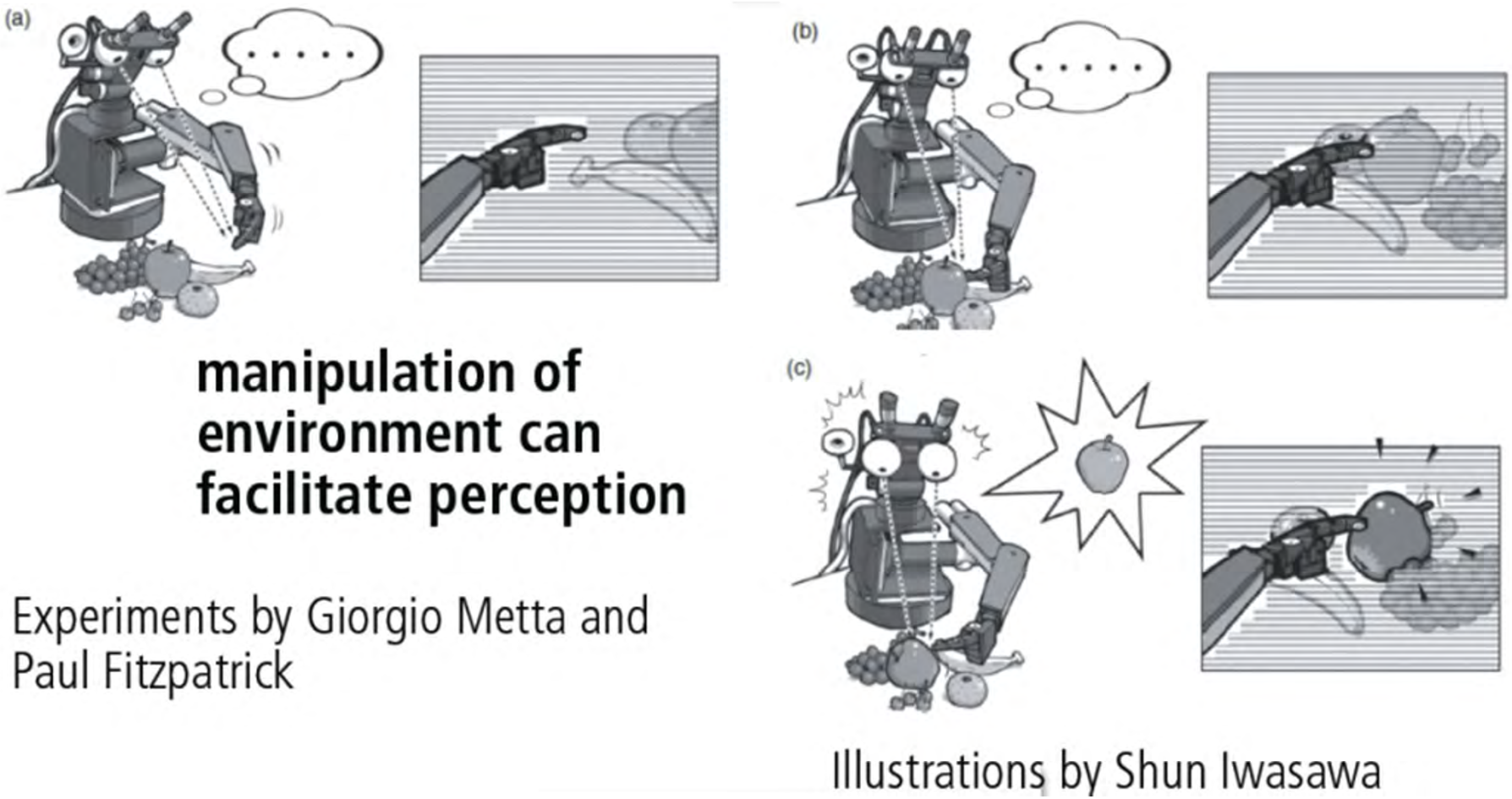
RobotCub

- rbcs (in short)
  - neuroscience
  - robotics
- robotics
  - iCub intelligence
  - iCub hardware
  - iCub software
  - iCub production

what are we looking for?

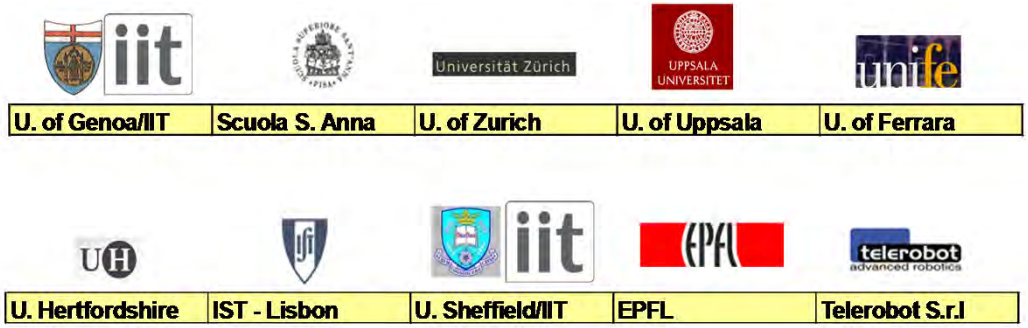
- the focus of our research is in the implementation of biologically sound models of cognition in robots of humanoid shape

- this has the two-fold aim of:
  - furthering our understanding of brain functions
  - realizing robot controllers that can learn and adapt from their mistakes



R. Pfeifer, J. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence.* 2006, MIT press

…a bit of (additional) history



- RobotCub (FP6): started 2004 – *finished 2010*, initial design
- ITALK: started 2008, extensions to language
- Poeticon: started 2008, supported the development of fingertips – *finished 2011*
- CHRIS: started 2008, supported the development of force control (for safety)
- RoboSKIN: started 2009, took over the development of a skin system
- Viactors: started 2009, study on intrinsic compliance and variable stiffness
- AMARSi: started 2010, compliance and learning, motor richness
- ImClever: learning and intrinsic motivations
- ROSSI: sensorimotor and social interaction
- Xperience: started 2011, cognitive architecture & affordances
- EFAA: started 2011, social interaction and learning from interaction
- Darwin: started 2011, manipulation and assembly
- Poeticon++: starting 2012, language and action

RobotCub goals

- ✓ design a humanoid robot platform, namely the iCub
- ✓ make it the platform of choice for researchers in artificial cognitive systems
- ✓ study cognition from a developmental perspective (neuroscience)

iCub community goals now

- ✓ maintain and improve the iCub to keep it alive
- ✓ make it the platform of choice for researchers in artificial cognitive systems
- ✓ study cognition from a multitude of points of view

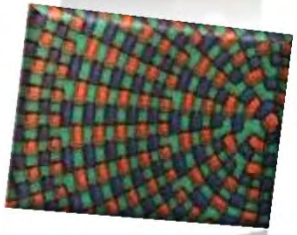
iCub is an **open source** international endeavour initially funded by the EU project **RobotCub**

- a full humanoid robot
- is 104cm, weighs 22 kg
- has 53 degrees of freedom
- can crawl, sit and manipulate
- open design as LGPL/GPL

open hardware

why is the iCub so special?

- hands: we started the design from the hands
  - 5 fingers, 9 degrees of freedom, 19 joints



- sensors: human-like, e.g. no lasers
  - cameras, microphones, gyros, encoders, force, tactile…

- electronics: flexibility for research
  - custom electronics, small, programmable (DSP)

- reproducible platform: community designed
  - reproducible & maintainable yet evolvable platform

why humanoids?

- scientific reasons
  - e.g. elephants don't play chess

- natural human-robot interaction

- challenging mechatronics

- fun!

why open platforms?

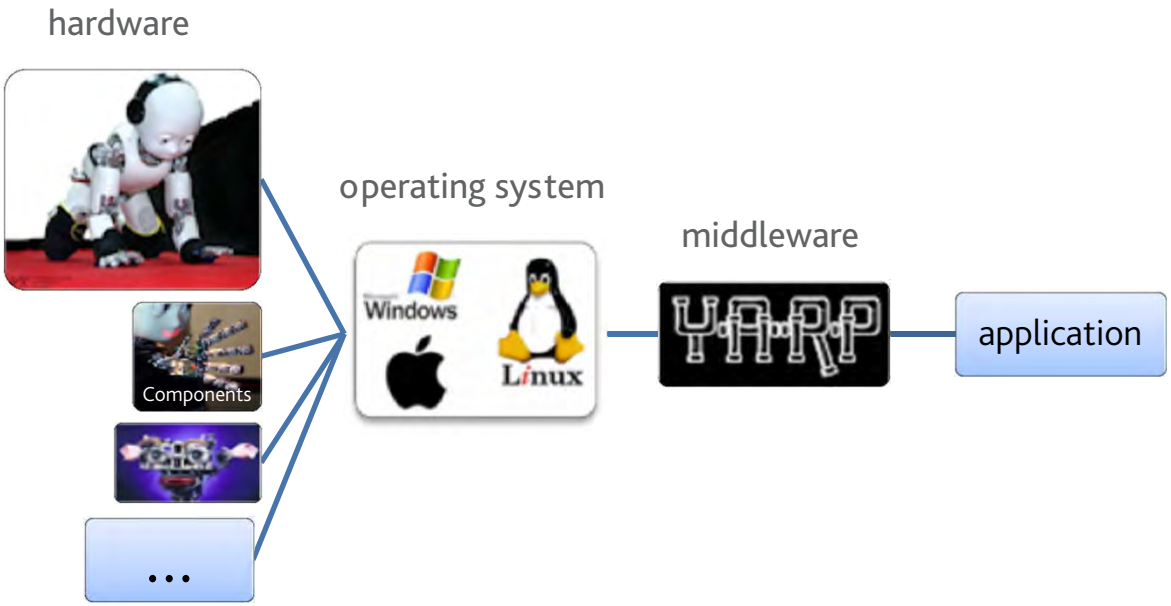
- repeatable experiments

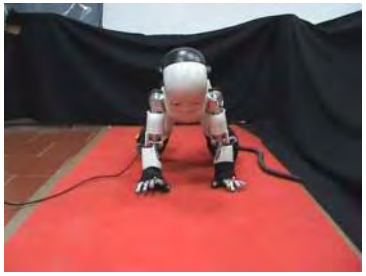
- benchmarking

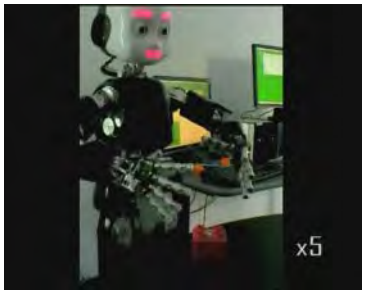
- quality

this resonates with industry-grade R&D in robotics

development tools



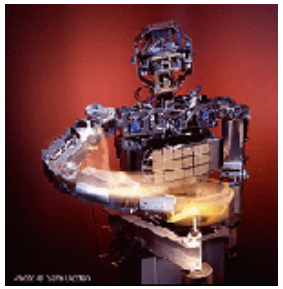




Yet Another Robot Platform

- YARP is an open-source (LGPL) middleware for humanoid robotics
- history
  - an MIT / Univ. of Genoa collaboration
  - born on Kismet, grew on COG, under QNX
  - with a major overhaul, now used by the iCub project
- C++ source code (some 400K lines)
- IPC & hardware interface
- portable across OSs and development platforms

2000-2001



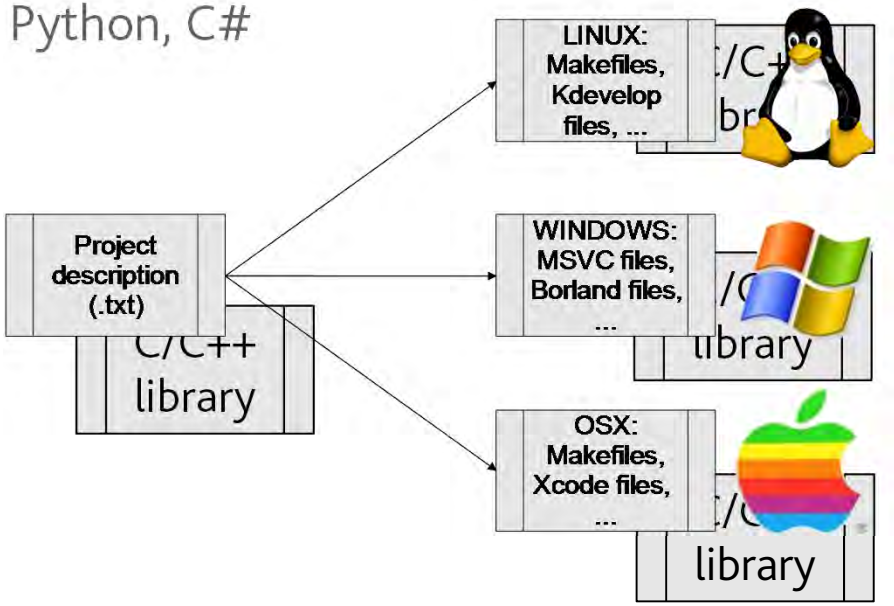
2001-2002

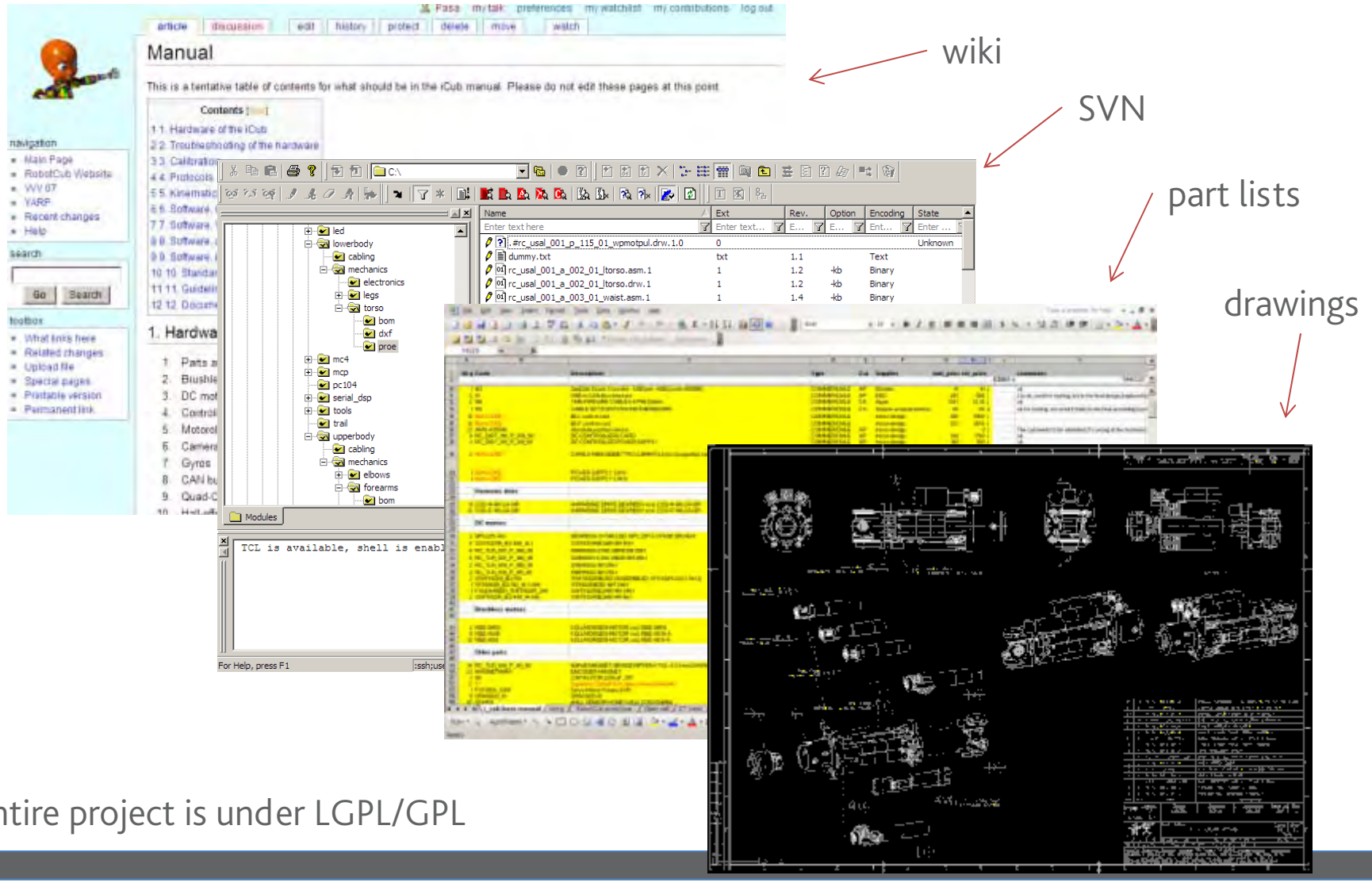
2003

2004-Today

exploit diversity: portability

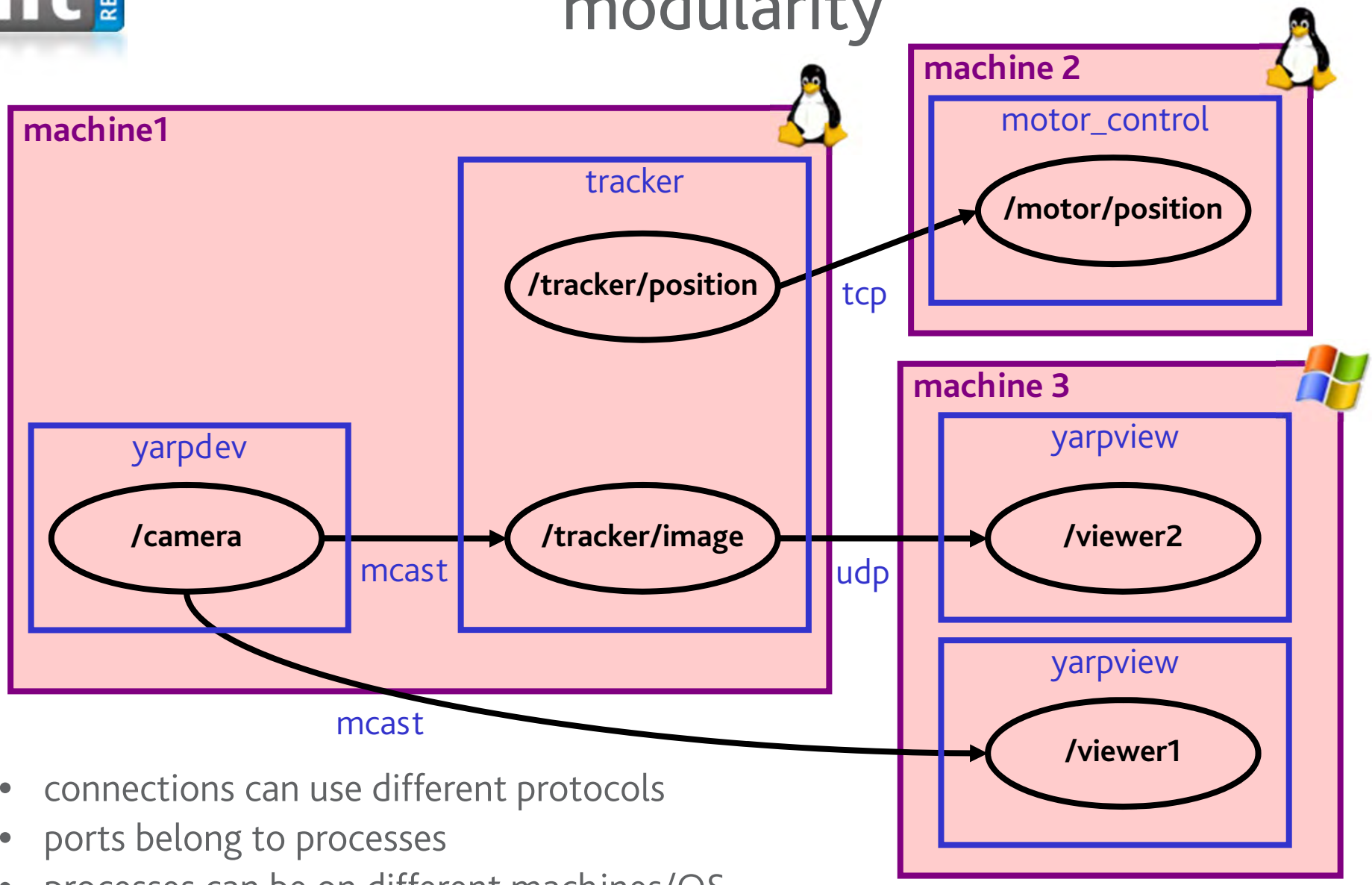
- operating system portability:
  - Adaptive Communication Environment , C++ OS wrapper: e.g. threads, semaphores, sockets
- development environment portability:
  - CMake
- language portability:
  - via Swig: Java (Matlab), Perl, Python, C#



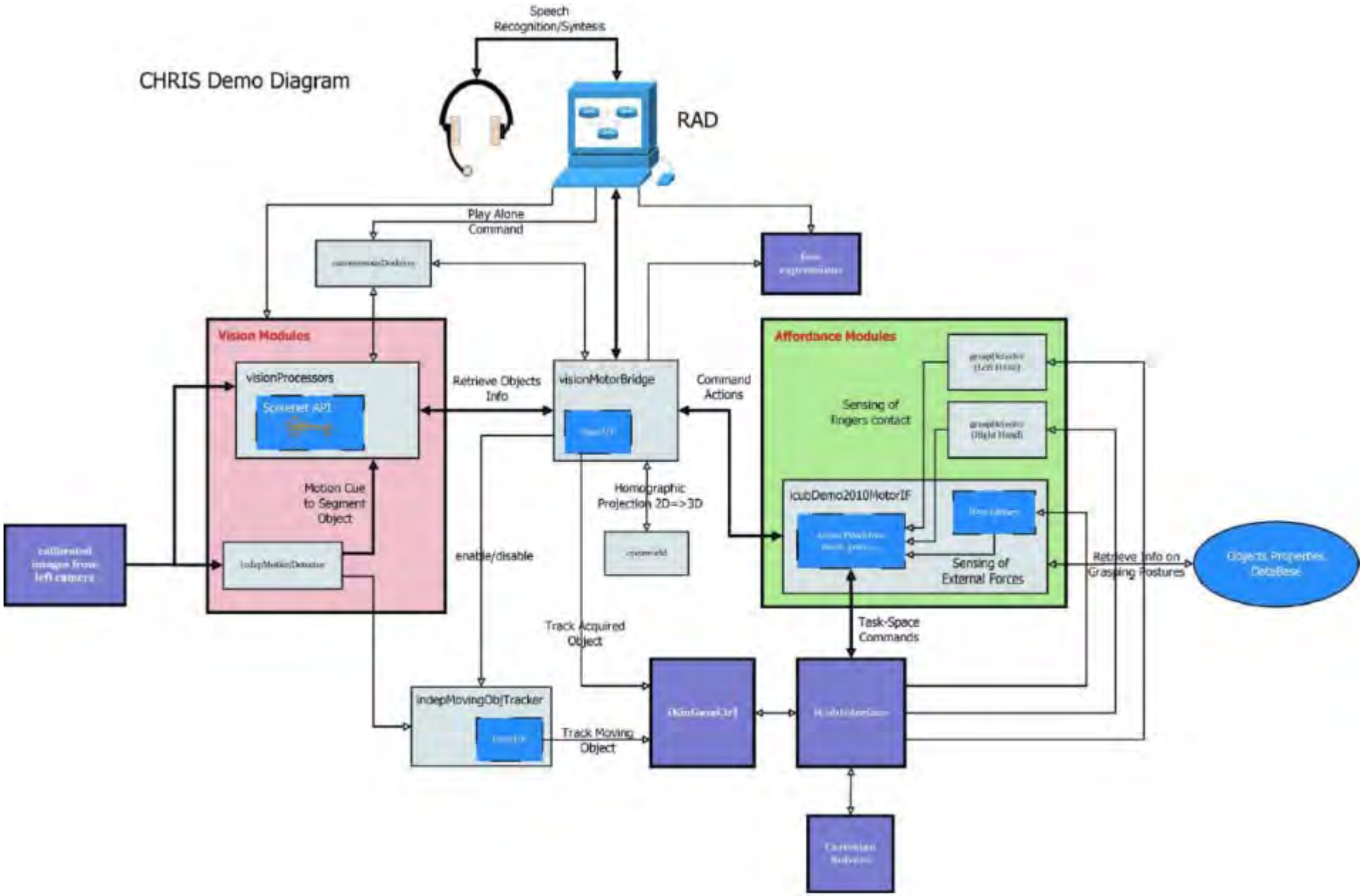


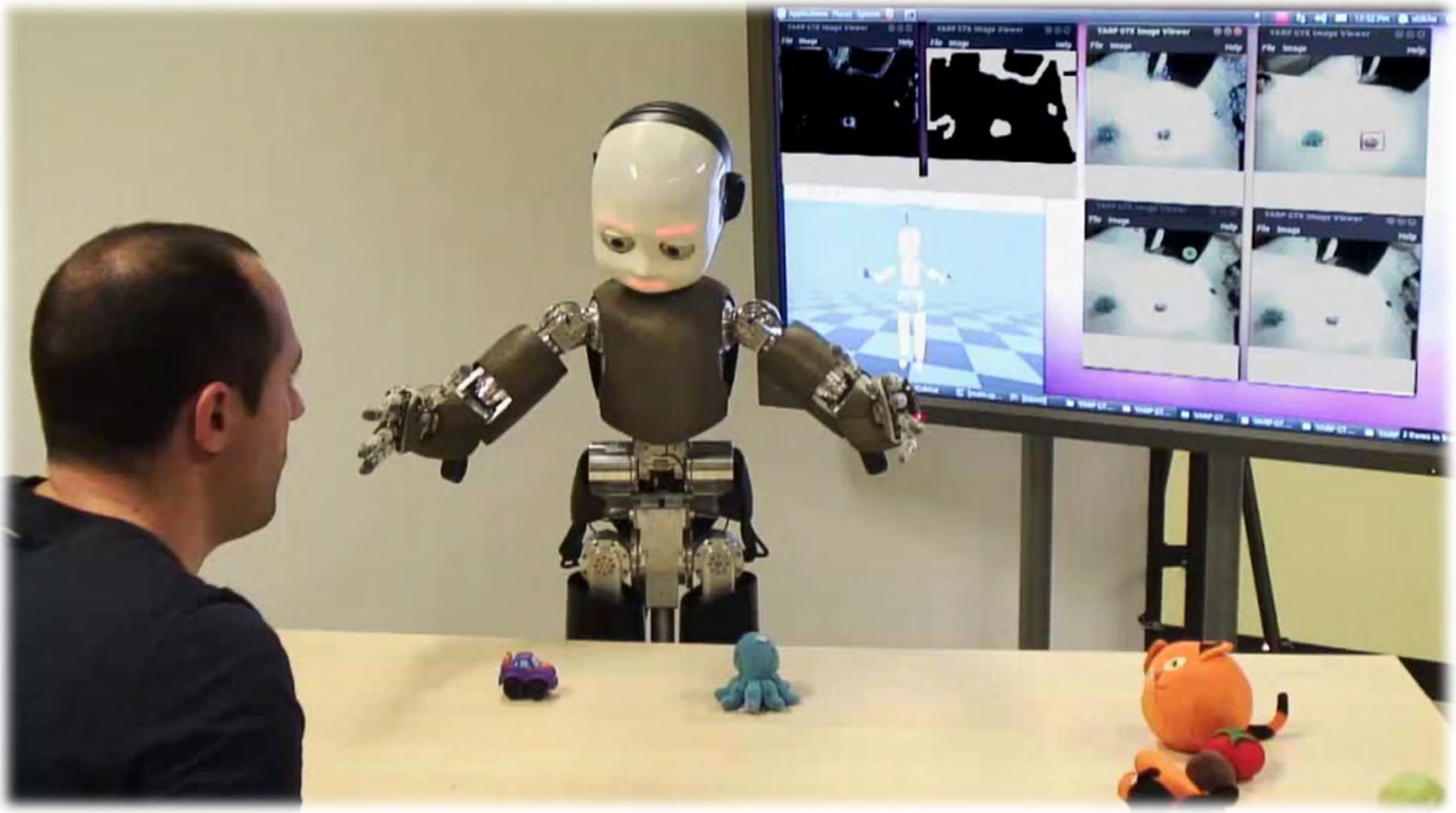
the entire project is under LGPL/GPL

modularity

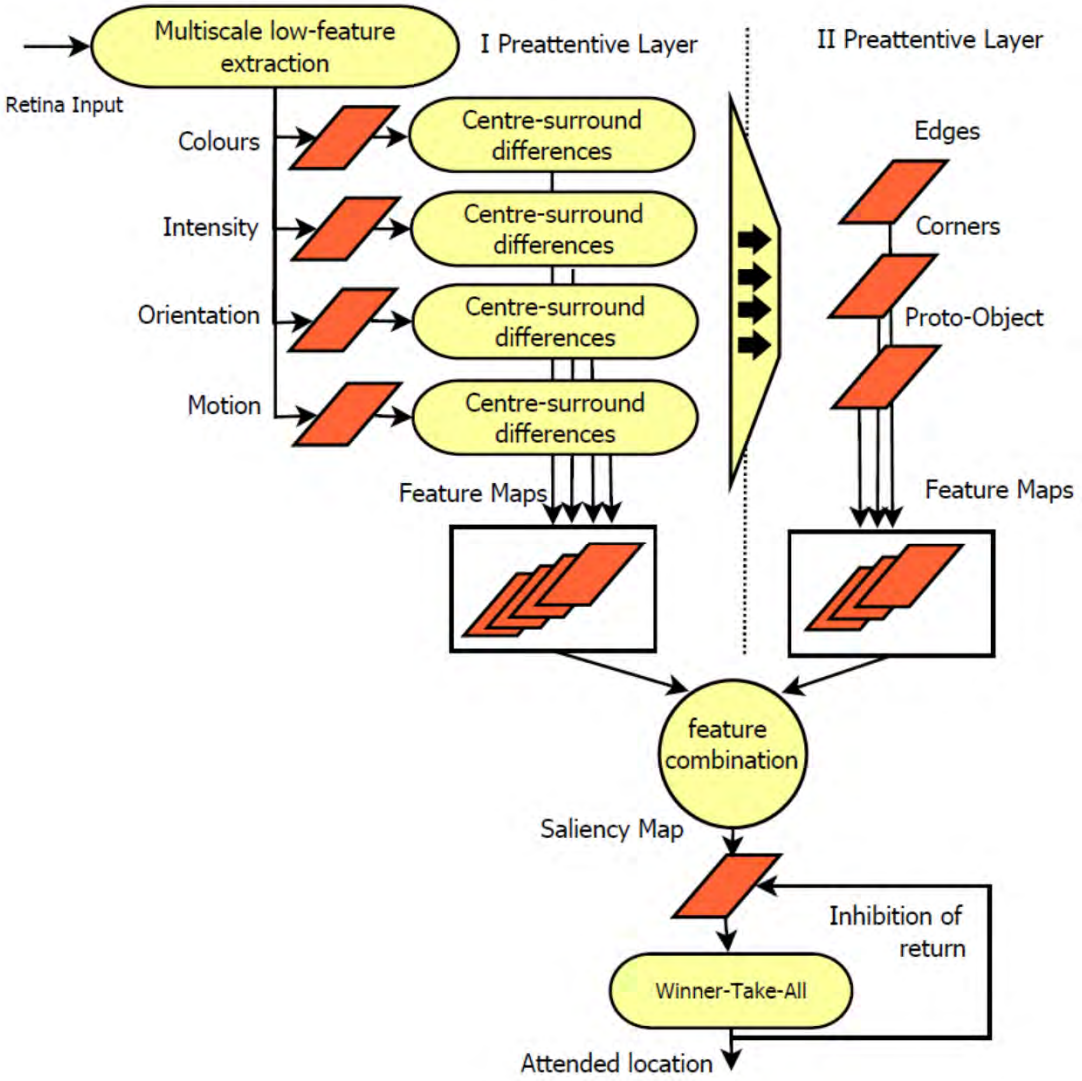


- connections can use different protocols
- ports belong to processes
- processes can be on different machines/OS

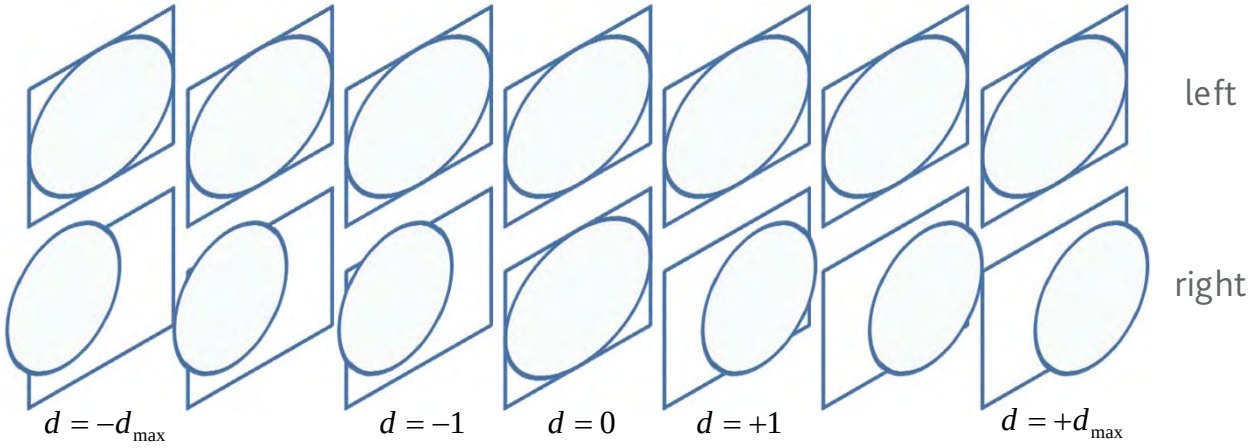

CHRIS Demo Diagram
Speech Recognition/Syntesis
RAD
Play Alone Command
Vision Modules
visionProcessors
Retrieve Objects Info
visionMotorBridge
Command Actions
Affordance Modules
Sensing of Fingers contact
Motion Cue to Segment Object
Homographic Projection 2D=>3D
icubDemo2010MotorIF
enable/disable
Sensing of External Forces
Retrieve Info on Grasping Postures
Track Acquired Object
Task-Space Commands
indepMovingObjTracker
Track Moving Object

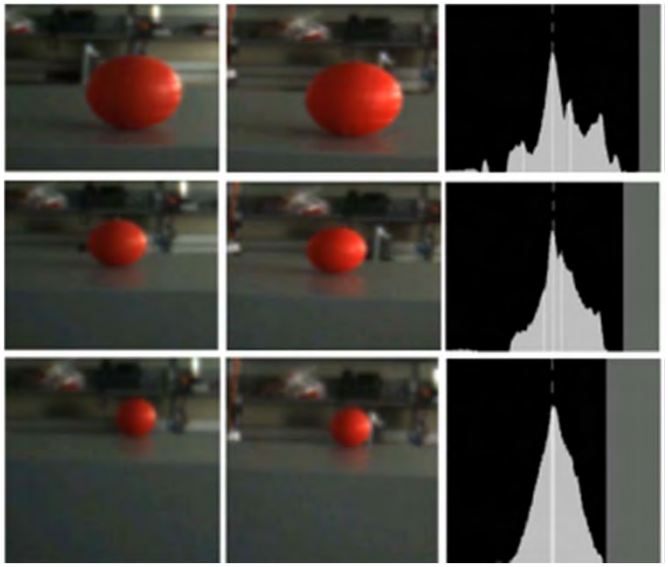


attention system



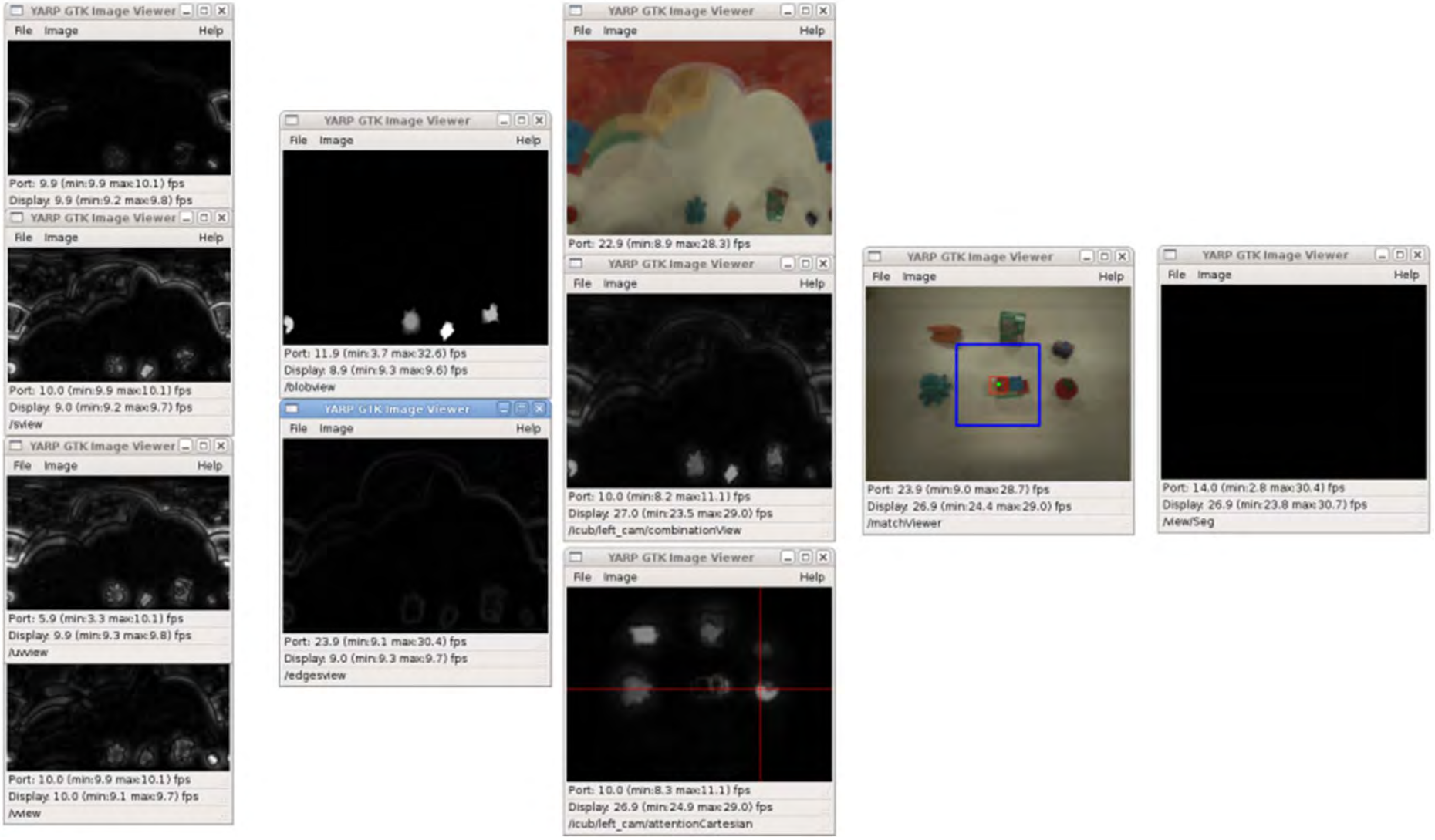
attention system



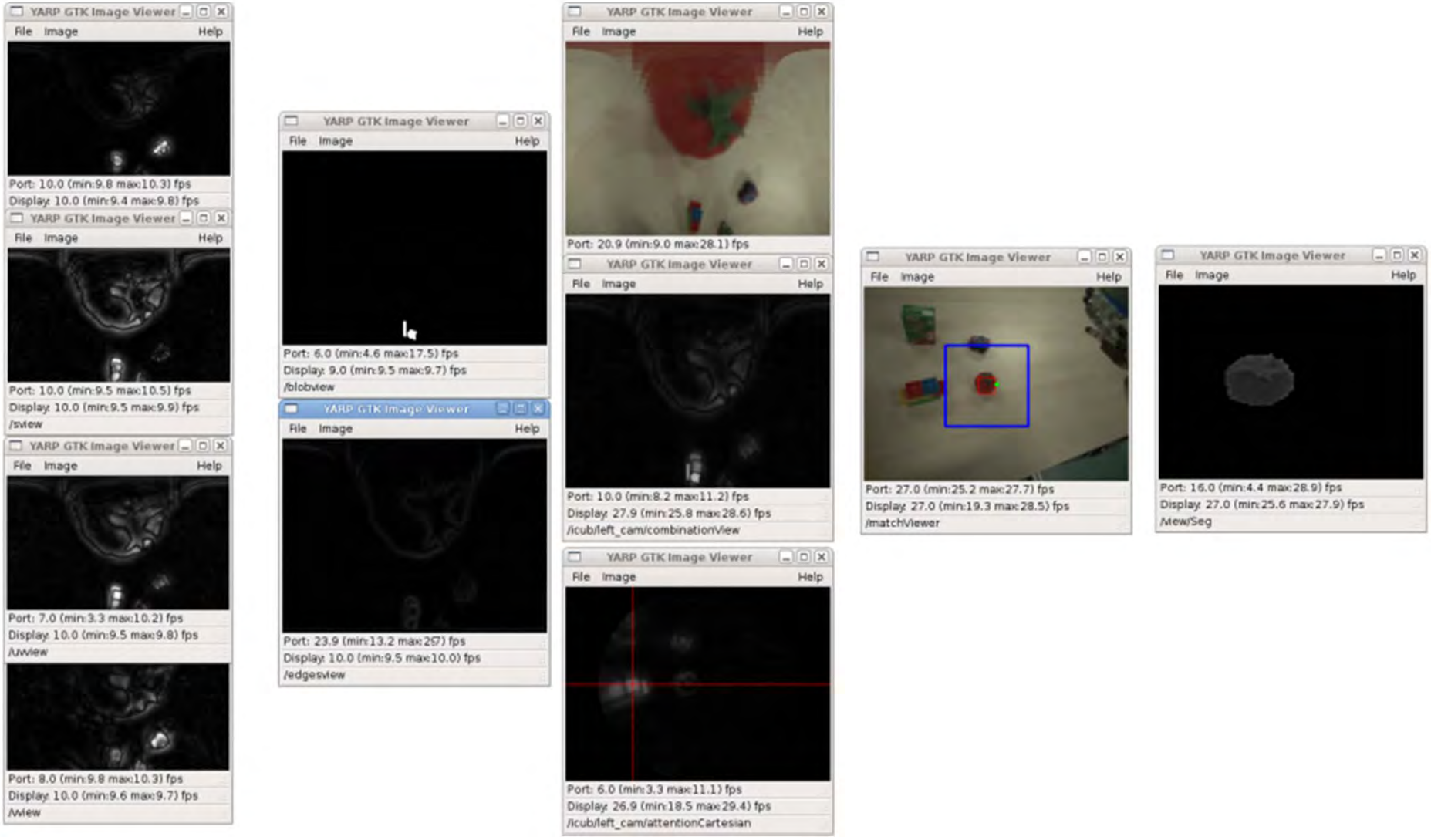


d is used as control signal (zero = foveation)

attention system

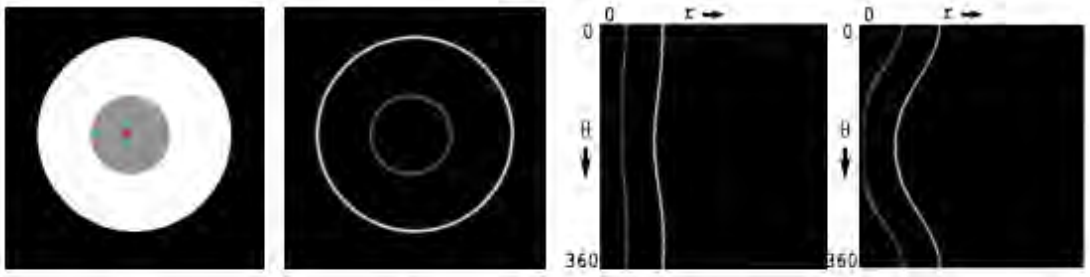


attention system



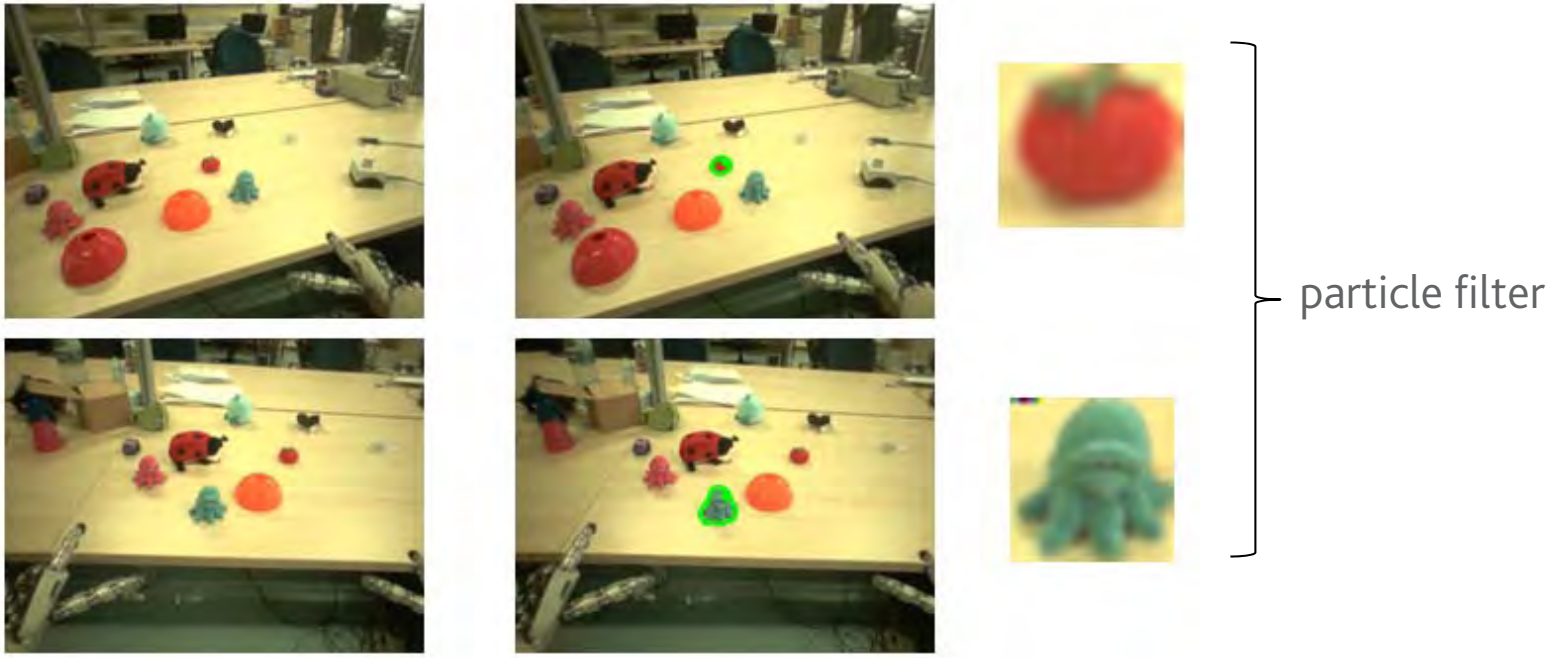
active segmentation with fixation

- the fixation point lies inside a particular region of arbitrary shape and size in the scene which can either be an object or just a part of it
- a (probabilistic) boundary edge map of the image is generated using all available low level cues
- this edge map is transformed into the polar space with the center at the fixation point and the path through this edge map that optimally splits the map into two parts is determined
- graph cut is used to find globally optimal solution to this binary segmentation problem



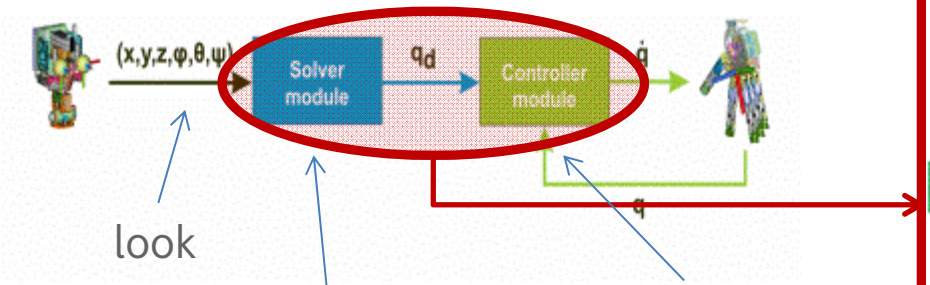
Active Segmentation With Fixation Ajay K. Mishra, Yiannis Aloimonos and Cheong Loong Fah International Conference on Computer Vision, 2009.

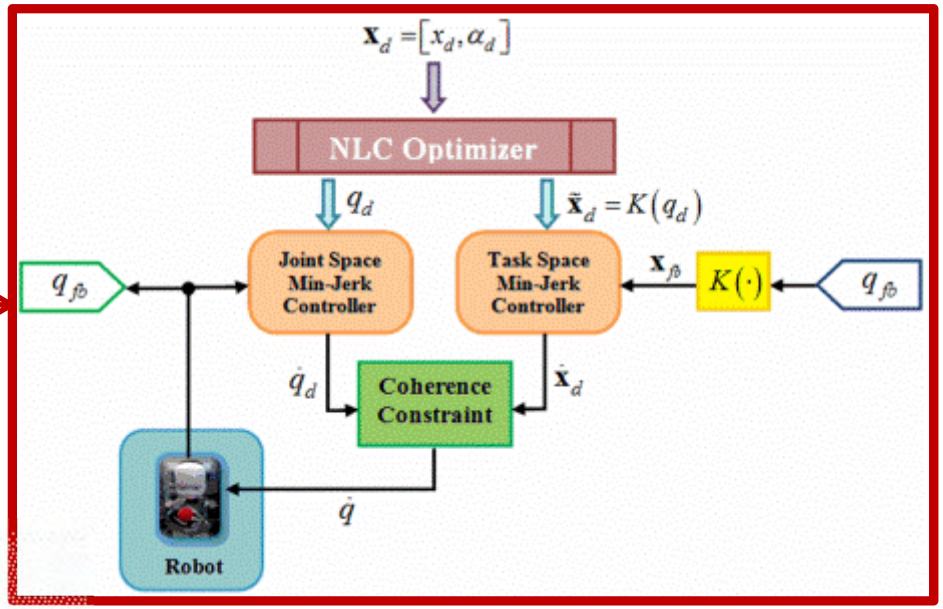
active segmentation with fixation

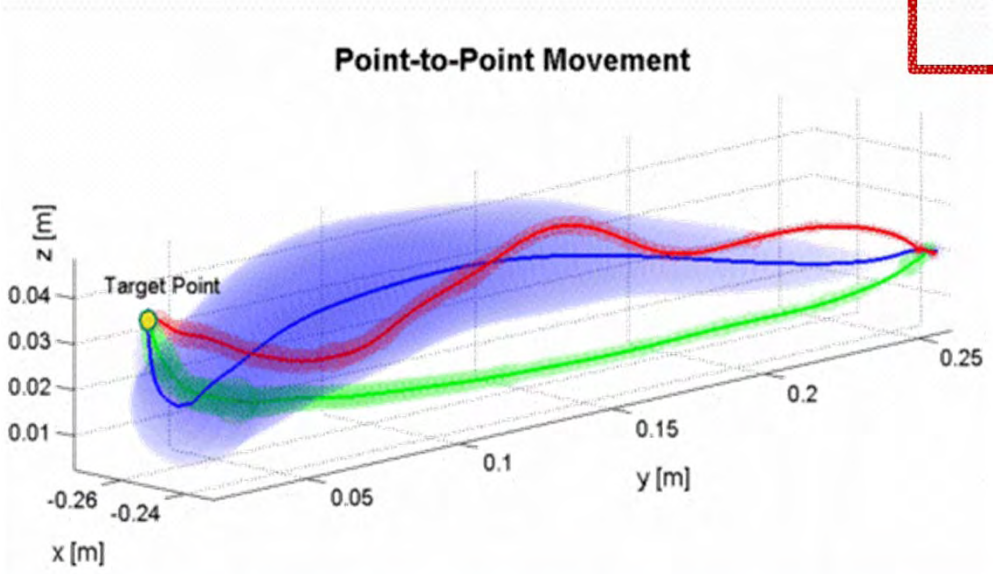


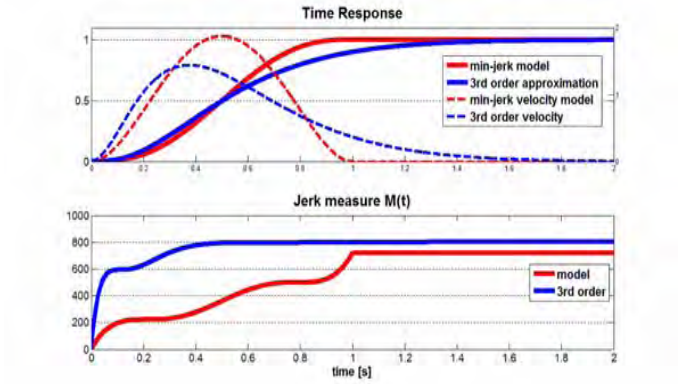
Active Segmentation With Fixation Ajay K. Mishra, Yiannis Aloimonos and Cheong Loong Fah International Conference on Computer Vision, 2009.

reaching


(x,y,z,φ,θ,ψ)
Solver module
qd
Controller module
q̇
q
look
inv kinematics
controller

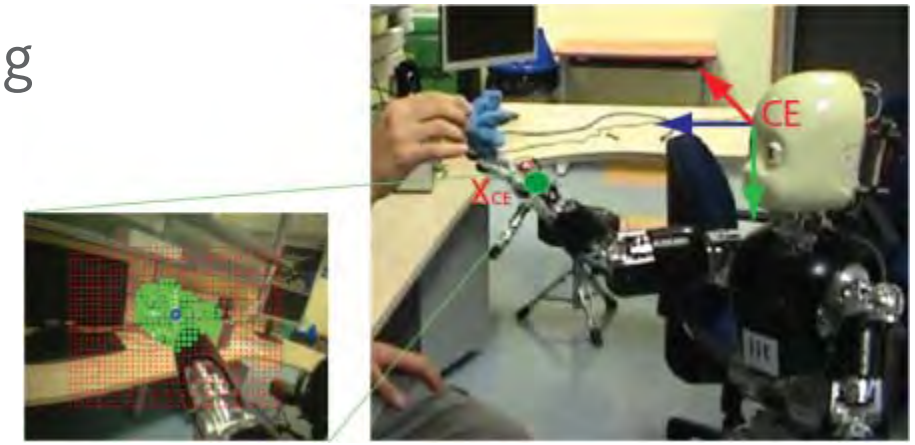

$\mathbf{x}_d = [x_d, \alpha_d]$
NLC Optimizer
q_d
$\tilde{\mathbf{x}}_d = K(q_d)$
q_{fb}
Joint Space Min-Jerk Controller
Task Space Min-Jerk Controller
$\mathbf{x}_{fb}$
$K(\cdot)$
q_{fb}
$\dot{q}_d$
Coherence Constraint
$\dot{\mathbf{x}}_d$
$\dot{q}$
Robot

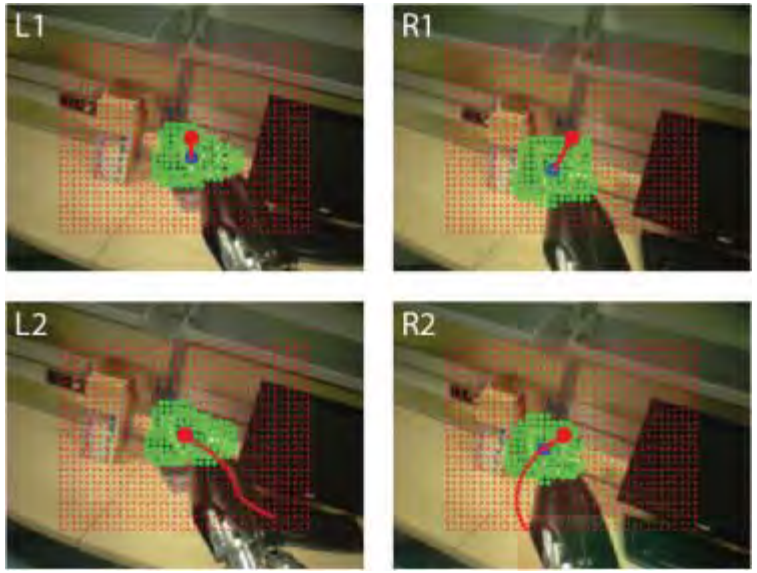

Point-to-Point Movement
Target Point
z [m]
0.04
0.03
0.02
0.01
-0.26
-0.24
x [m]
0.05
0.1
0.15
y [m]
0.2
0.25


Time Response
min-jerk model
3rd order approximation
min-jerk velocity model
3rd order velocity
Jerk measure M(t)
model
3rd order
time [s]

reaching

refine reaching through learning





$$(x, y, z)_{CE} = M(u_l, v_l, u_r, v_r, T, V_s, V_g)$$

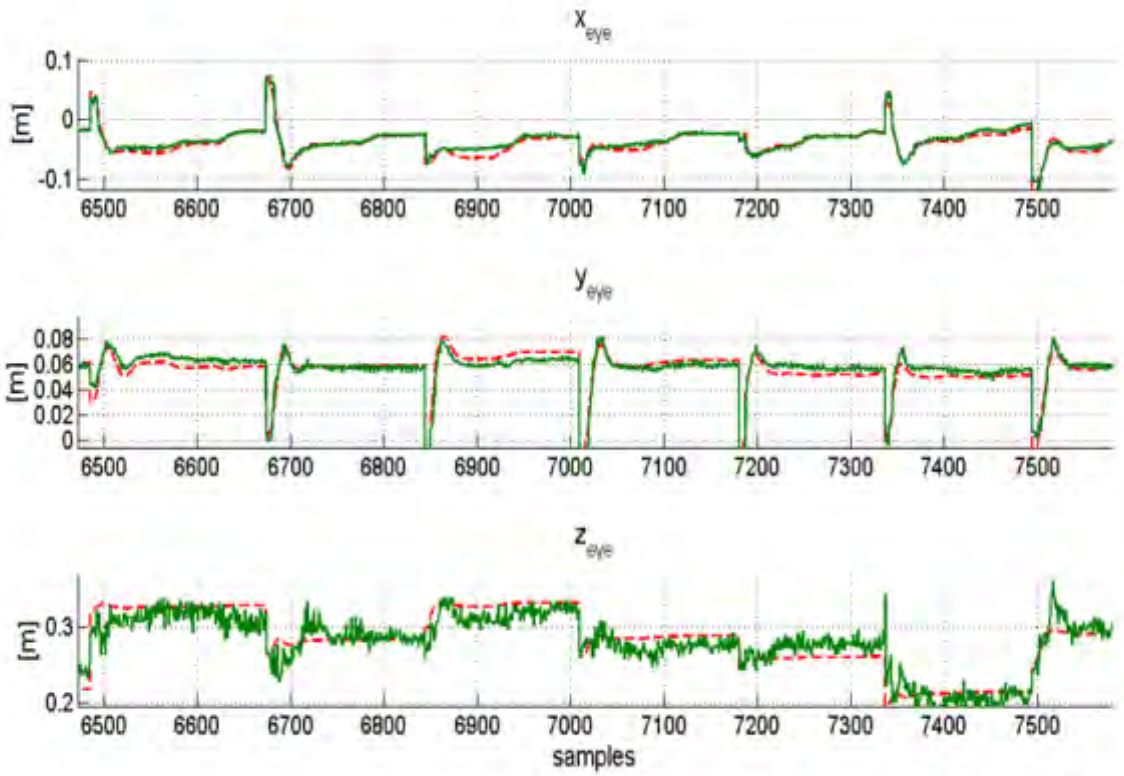
fixation point

to learn

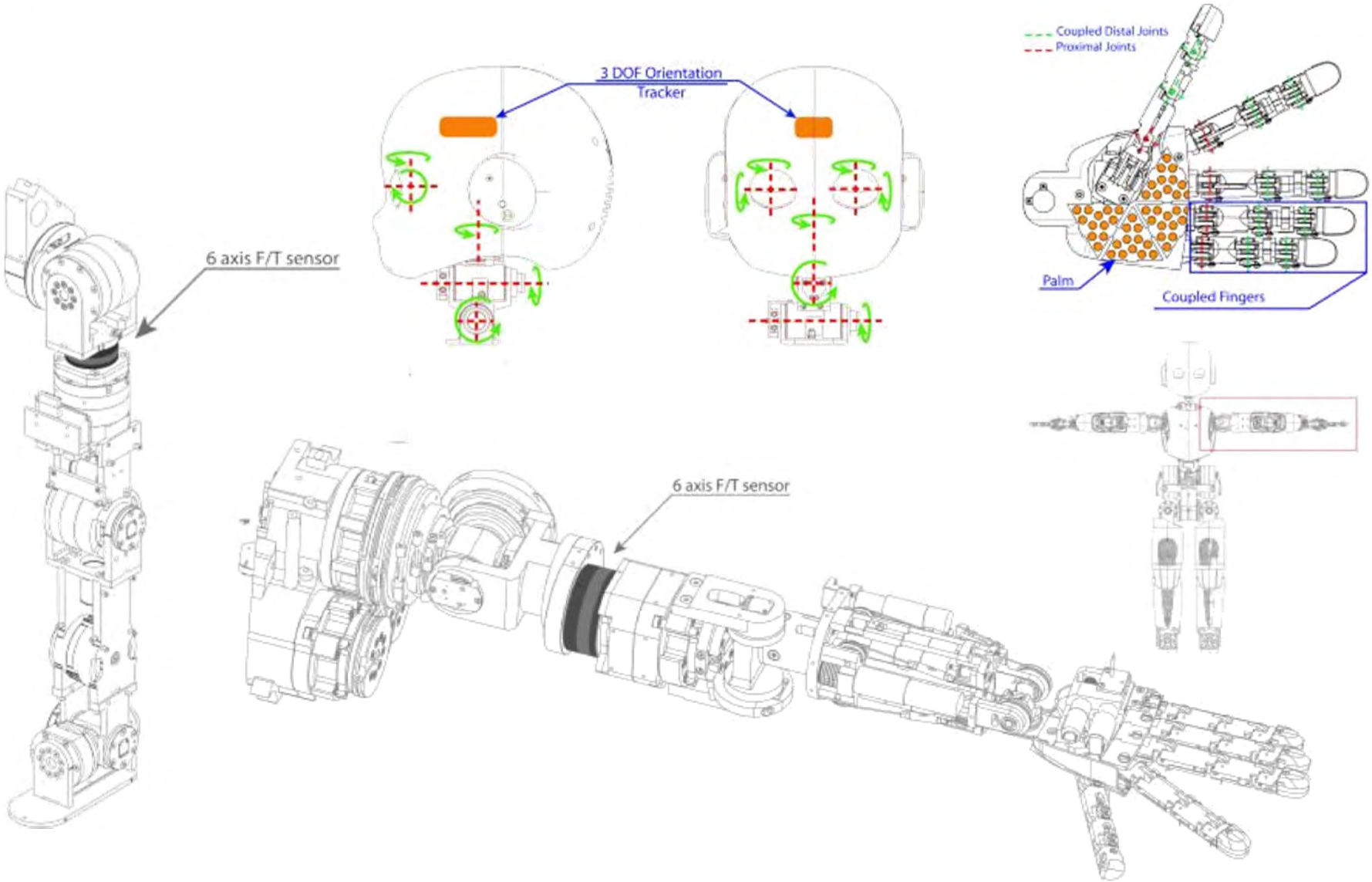
image

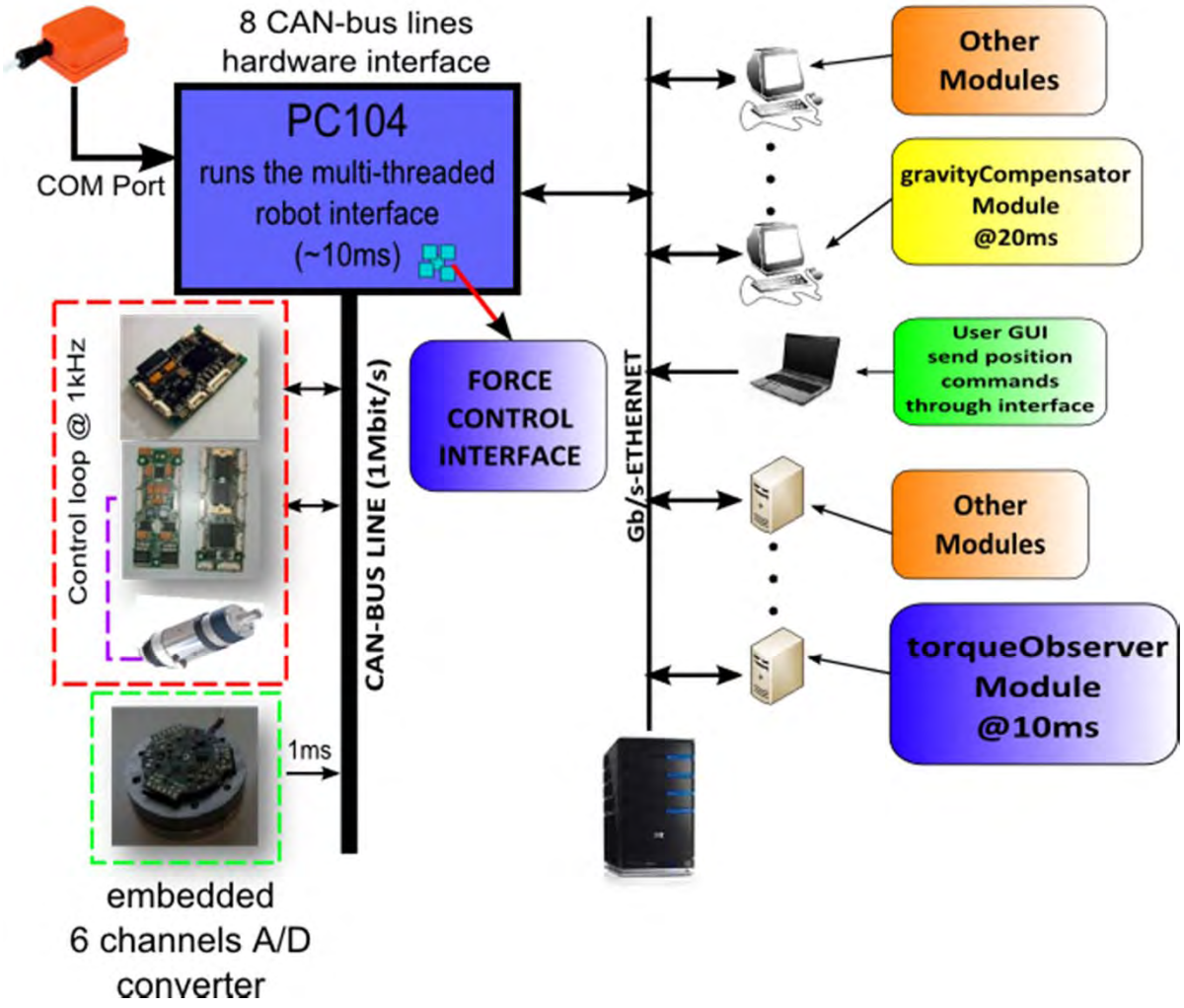
eye configuration

reaching: control vs. desired



iCub sensorization





$$e = \tau - \tau_d$$

$$\widehat{w}_e = \begin{bmatrix} I & 0 \\ -[r_{se}]_\times & I \end{bmatrix} \cdot (w_s - w_i)$$

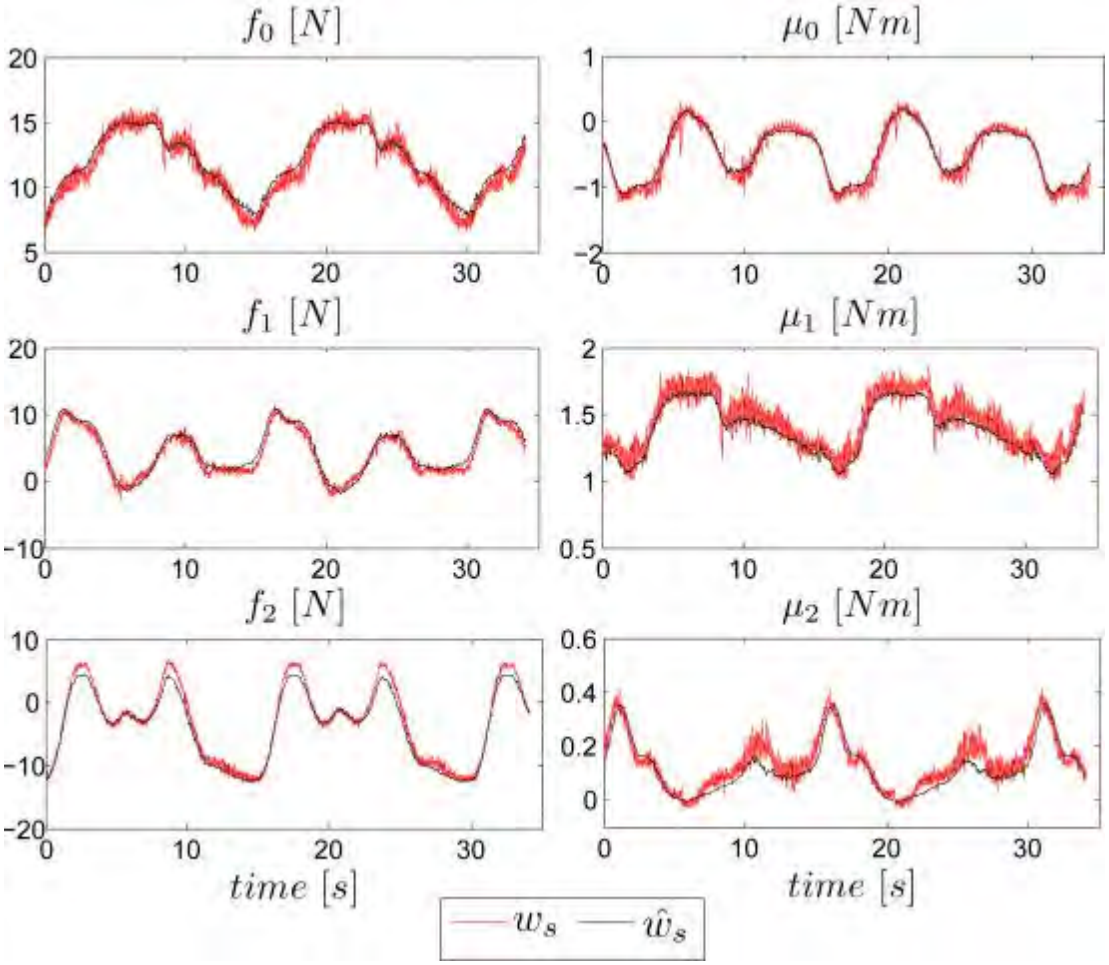
$$\hat{\tau}_e = J^T(q) \cdot \widehat{w}_e$$

$$e = \hat{\tau}_e - \tau_d$$

$$u = k_p \cdot e + k_d \cdot \dot{e} + k_i \cdot \int e$$

$$\tau_d = K \cdot (q - q_d) + D \cdot (\dot{q} - \dot{q}_d)$$

Test of the dynamical model



main idea: build a soft capacitor

capacitor

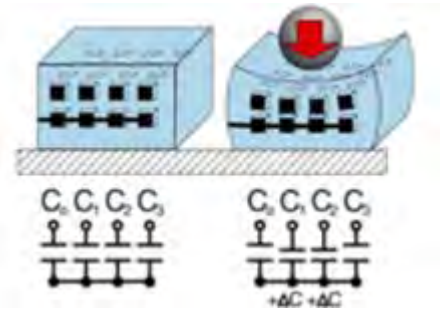
ground plane: e.g. conductive fabric
parameters: mechanical properties, impedance, etc.

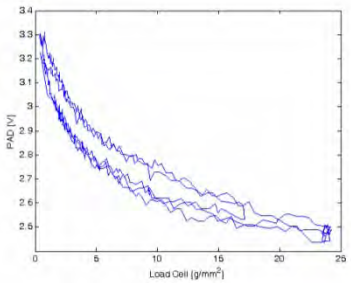
soft material: e.g. silicone
parameters: dielectric constant, mechanical stiffness, etc.

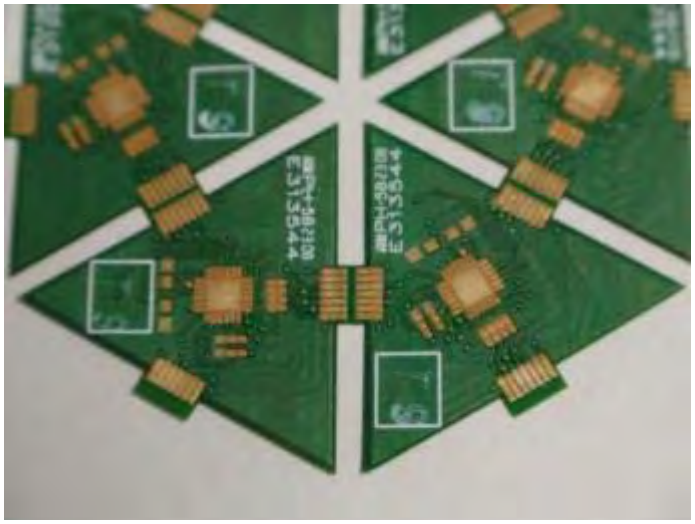
electrodes: etched on a flexible PCB
parameters: shape, folding, etc.

skin

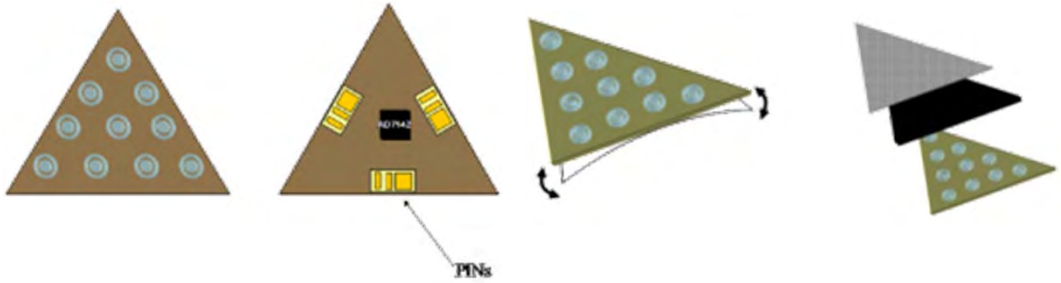
principle

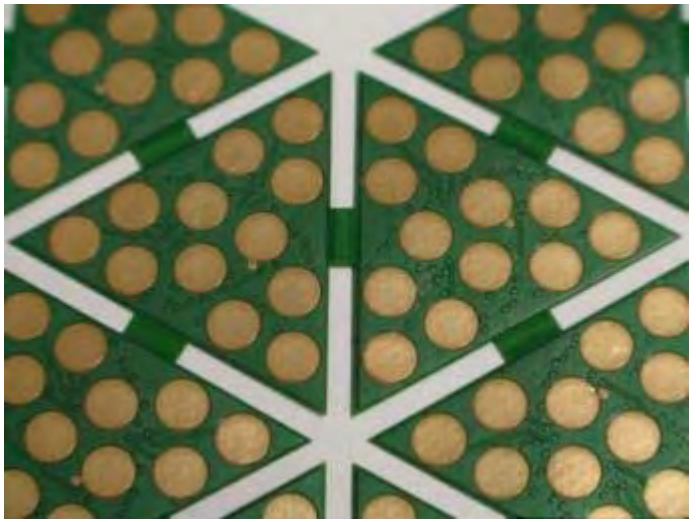




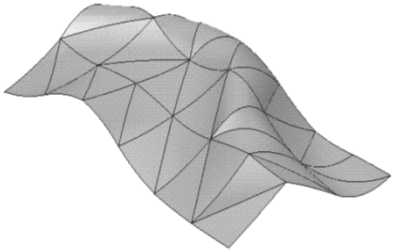


lots of sensing points

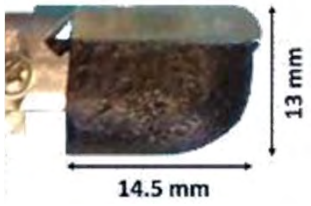




structure of the skin



skin evolution (fingertips)



skin evolution: large areas