Control, 3-D reconstruction, and recognition on a humanoid head

Aleš Ude

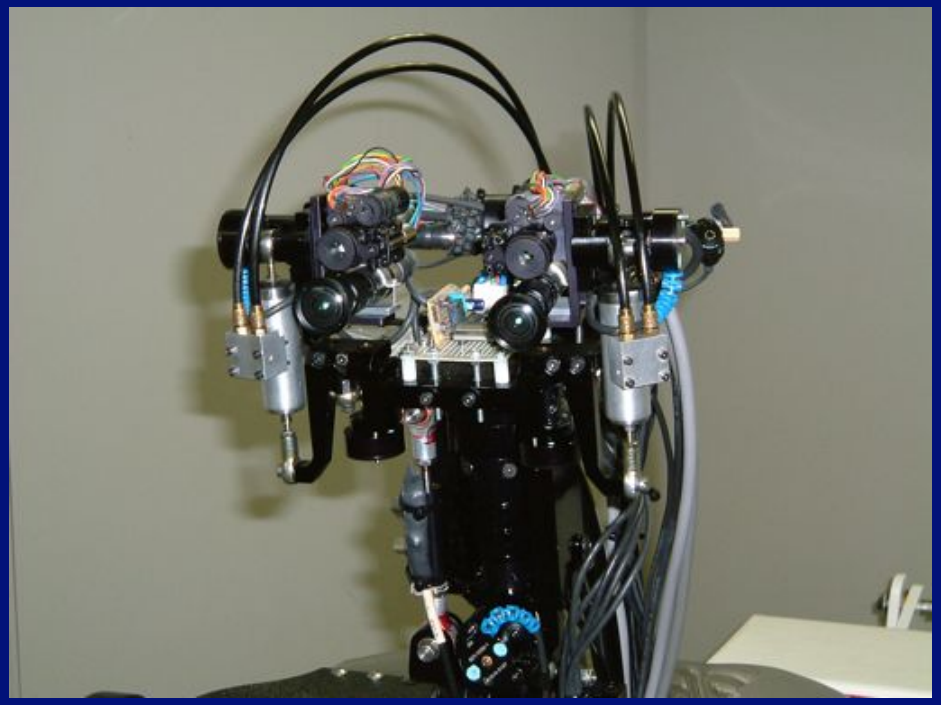
Jožef Stefan Institute

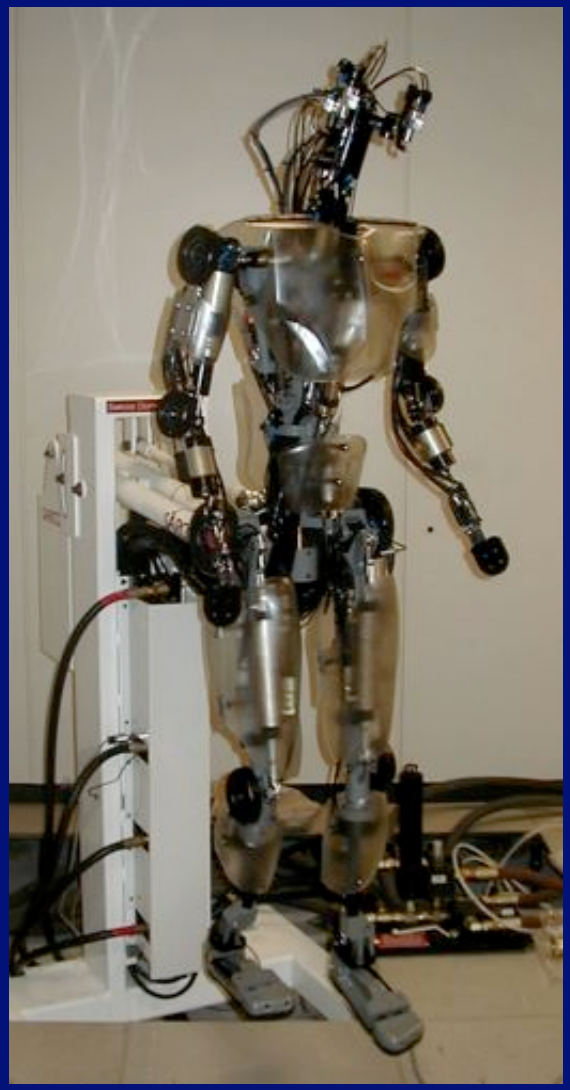
Ljubljana, Slovenia

ATR Computational Neuroscience Laboratories

Kyoto, Japan

Humanoid vision systems





Humanoids with foveated vision

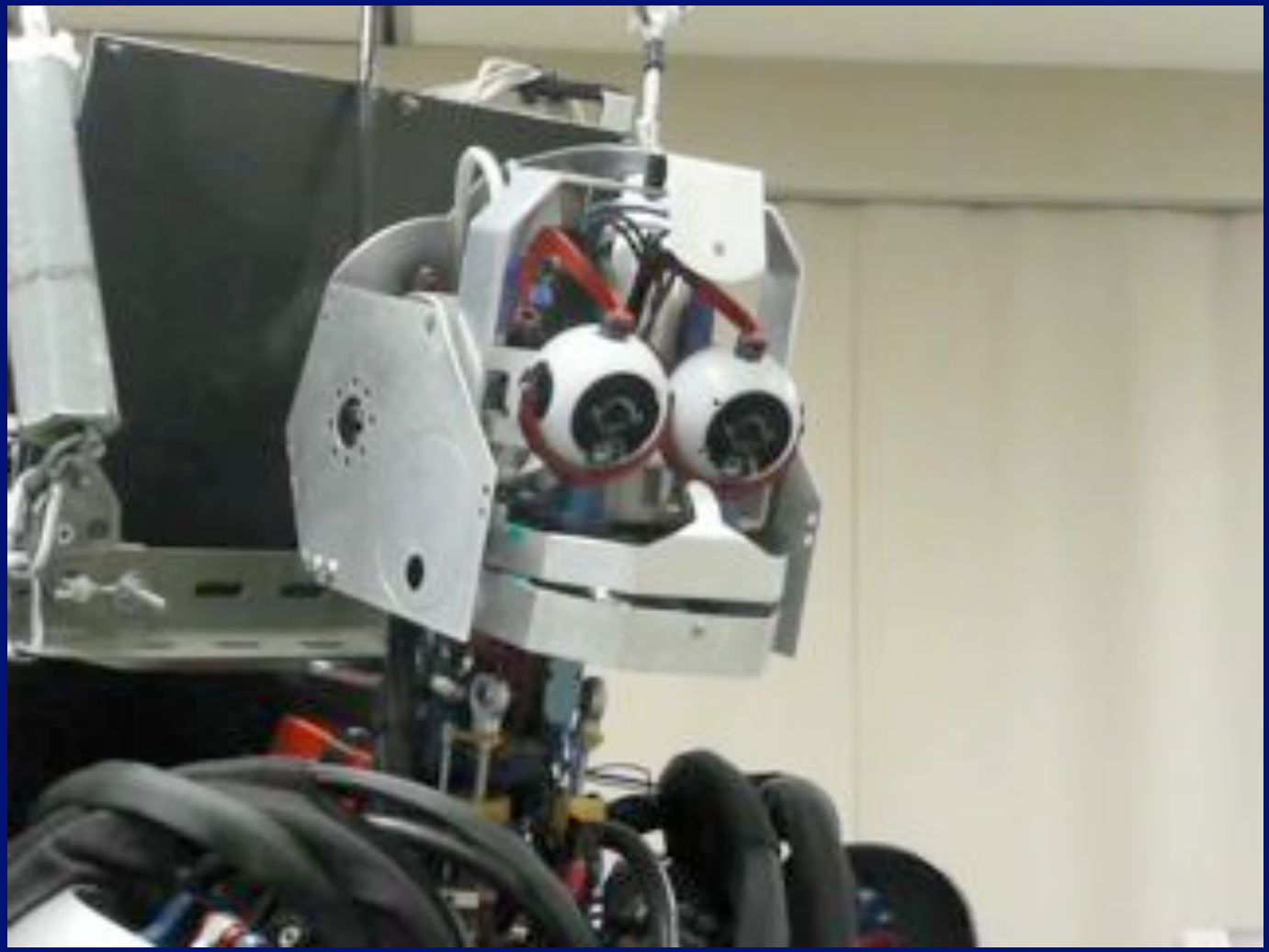
- Humanoid head with special lenses having a variable viewing angle (Kuniyoshi et al.)
- Babybot with log-polar cameras at the University of Genova (Sandini et al.)
- Humanoids with two pairs of cameras in each eye for foveal and peripheral vision:
  - DB, CB-i at ATR
  - ARMAR (Dillmann and Asfour)
  - Cog and Kismet at MIT (Brooks et al.)
  - Infantoid at CRL (Kozima)

Foveated vision

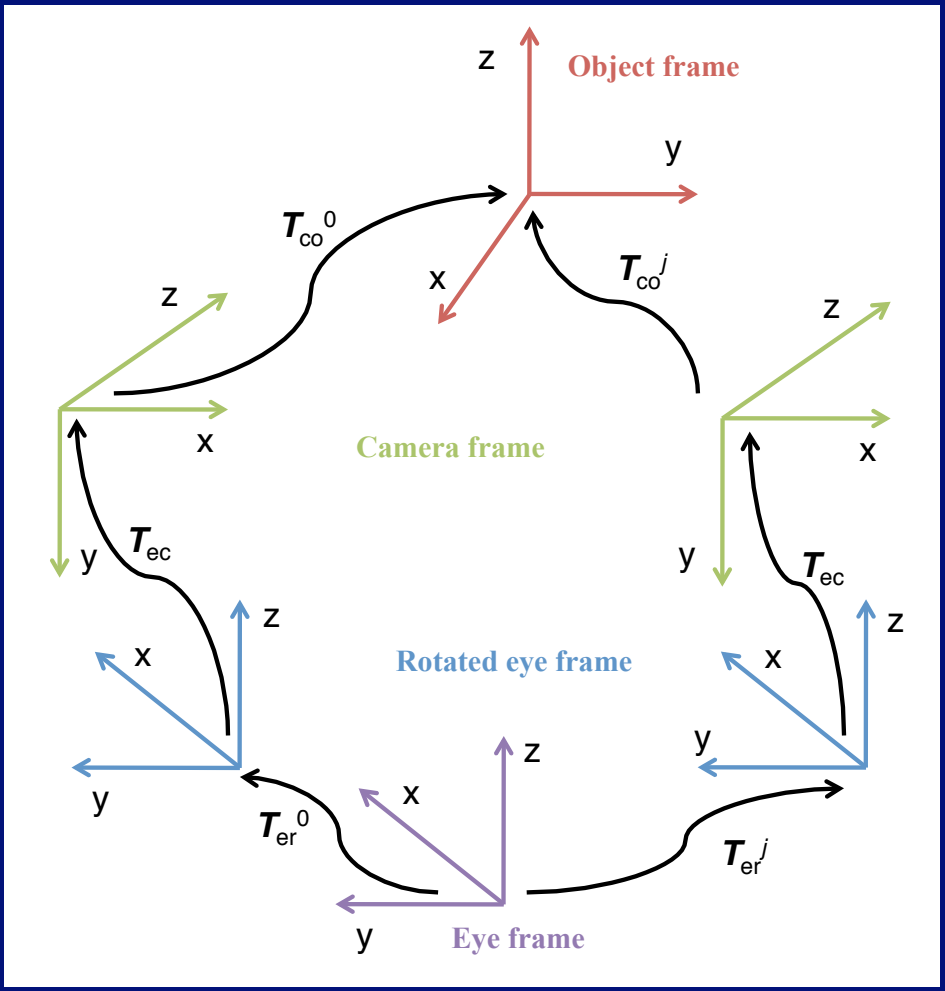
Integration with motor control

- Makes sense only on active systems.
- Vision should be able to deal with fast robot movements and occlusions.
- Motor control should be able to deal with vision failures.

3-D vision on an active head



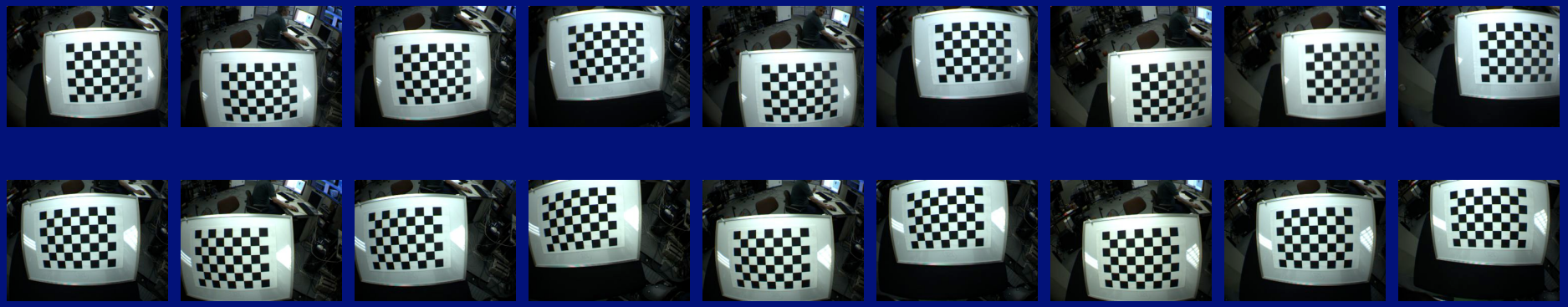
Eye coordinate systems

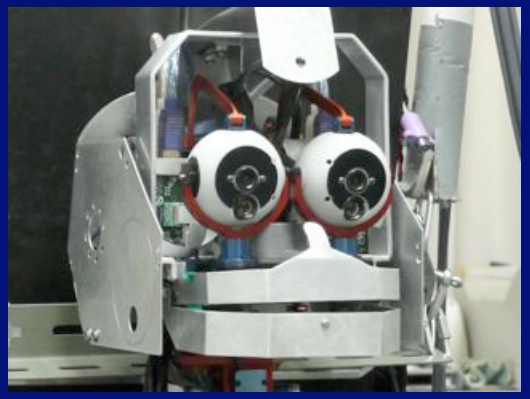


- T_{co}^j, $j = 0, 1, \ldots, n$: object positions in camera coordinates
- T_{ec}: the unknown and constant transformation between the eye and camera coordinate systems
- T_{er}^j: the rotated eye coordinate frames

Model identification

We estimate a number of postures of the fixed calibration pattern observed from different eye orientations:





Transformations

- The following relation can be derived::

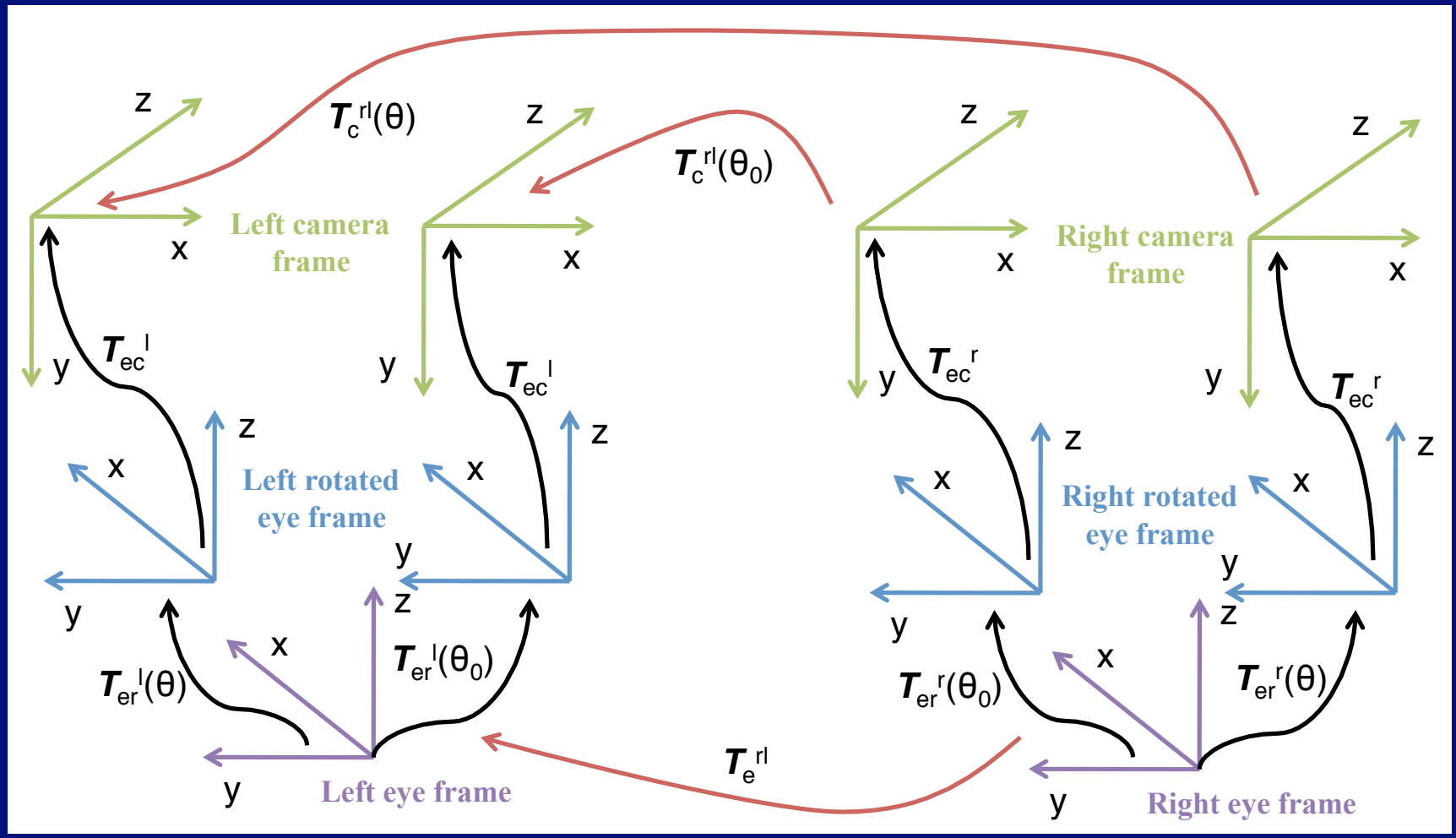
$$\mathbf{T}_{\text{co}}^{0\ -1} * \mathbf{T}_{\text{co}}^{j} = \mathbf{T}_{\text{ec}} * \mathbf{T}_{\text{er}}^{0} * \mathbf{T}_{\text{er}}^{j\ -1} * \mathbf{T}_{\text{ec}}^{-1}$$

- The following equation system needs to be solved on the special Euclidean group of rigid body transformations SE(3):

$$\mathbf{A}_j\mathbf{X} = \mathbf{X}\mathbf{B}_j,\ j = 1,\ldots,n$$

where $\mathbf{A}_j = \mathbf{T}_{\text{co}}^{0\ -1} * \mathbf{T}_{\text{co}}^{j}$, $\mathbf{B}_j = \mathbf{T}_{\text{er}}^{0} * \mathbf{T}_{\text{er}}^{j\ -1}$, $\mathbf{X} = \mathbf{T}_{\text{ec}}$

Stereo coordinate systems



3-D reconstruction

- We need to express 3-D coordinates in a common coordinate frame (we use left eye camera frame)

$$\mathbf{y}_r = \mathbf{T}_c^{rl}(\theta)\mathbf{y}_l = \mathbf{T}_{ec}^{r\,-1}\mathbf{T}_{er}^{r}(\theta)^{-1}\,\mathbf{T}_e^{rl}\mathbf{T}_{er}^{l}(\theta)\mathbf{T}_{ec}^{l}\mathbf{y}_l$$

- This allows us to calculate $\mathbf{y}_l$ by solving ($\mathbf{A}_l$, $\mathbf{A}_r$ are the internal camera parameters)

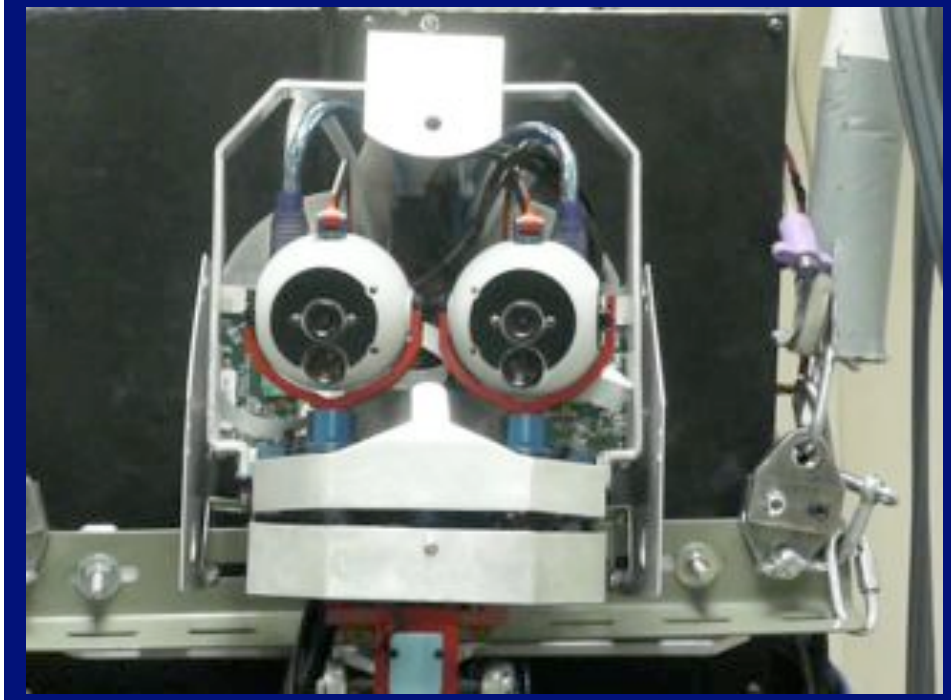
$$\tilde{\mathbf{u}}_l = \mathbf{A}_l\mathbf{y}_l$$

$$\tilde{\mathbf{u}}_r = \mathbf{A}_r\mathbf{y}_r = \mathbf{A}_r\mathbf{T}_c^{rl}(\theta)\mathbf{y}_l$$

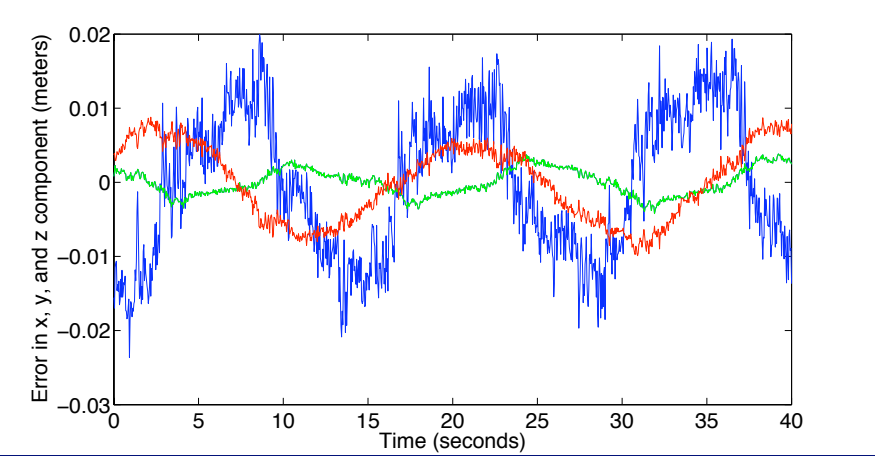
- The point in body coordinate frame is given by

$$\mathbf{y}_b = \mathbf{T}_{be}^{l}(\theta)\mathbf{T}_{er}^{l}(\theta)\mathbf{T}_{ec}^{l}\mathbf{y}_l$$

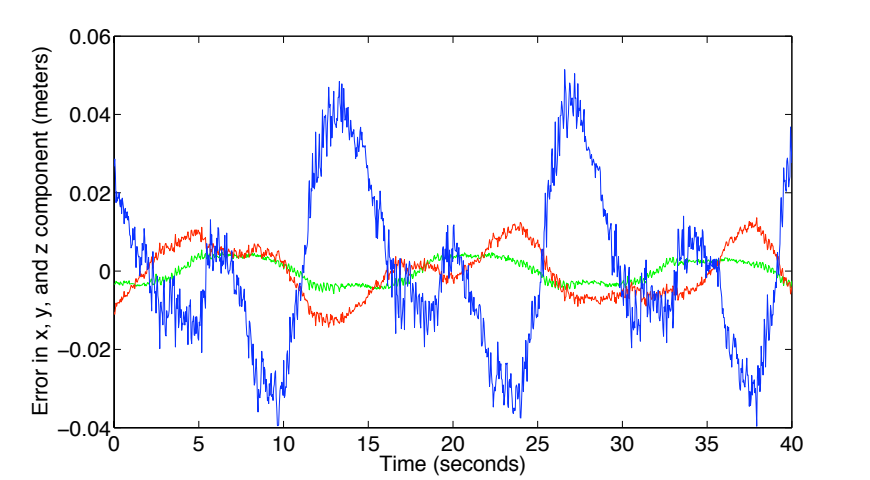
Experiments



Experimental results

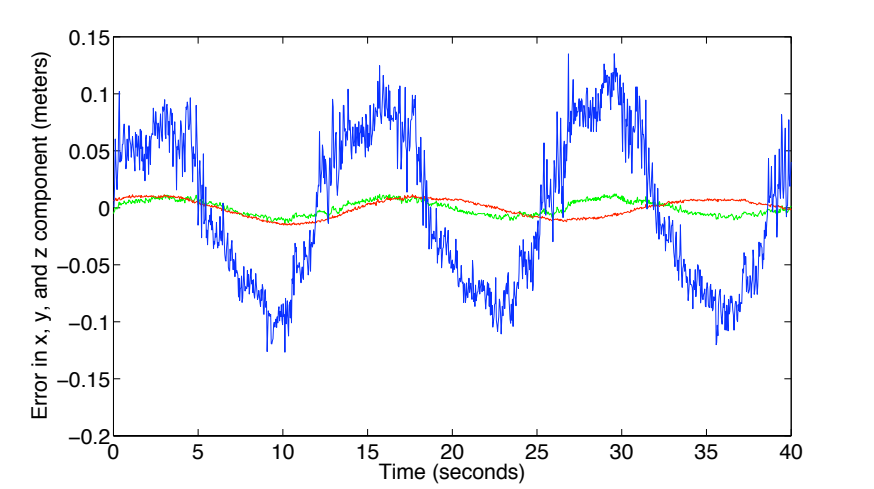


Convergent eye movement, wide lense, distance about 0.45 m. The standard deviation was (0.010; 0.002; 0.005) m.

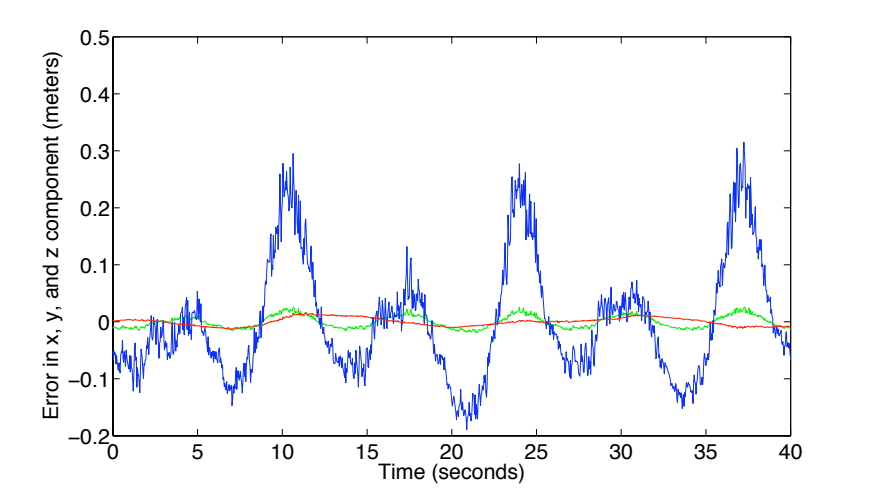


Divergent eye movement, wide lense, distance about 0.45 m. The standard deviation was (0.020; 0.003; 0.007) m

Experimental Results

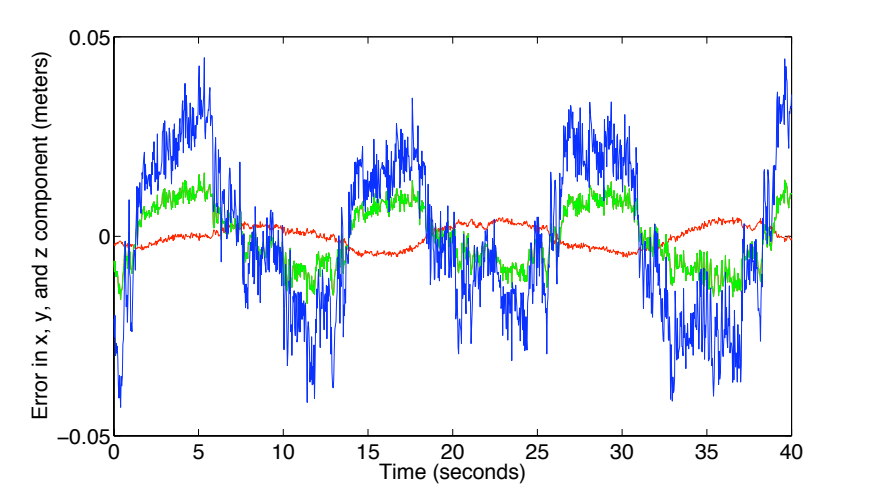


Convergent eye movement, wide lense, distance about 1.1 meter. The standard deviation was (0.066; 0.006; 0.008) m.

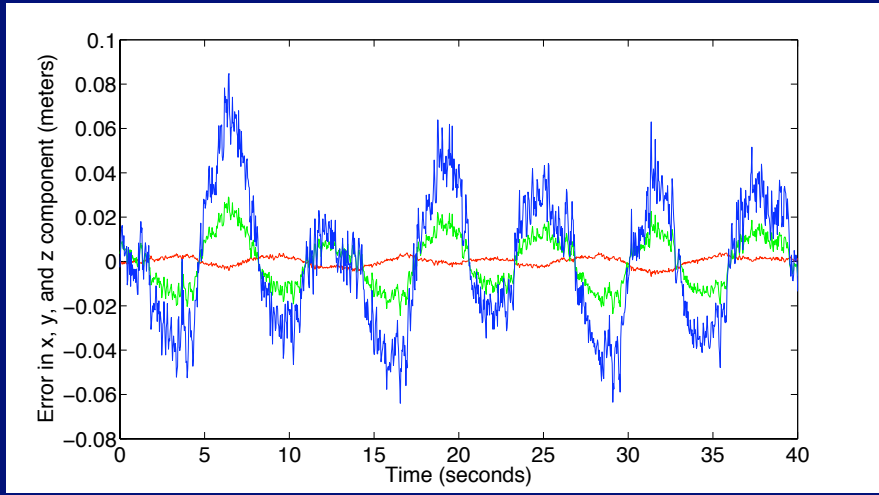


Divergent eye movement, wide lense, distance about 1.1 meter. The standard deviation was (0.102; 0.011; 0.007) m.

Experimental results



Convergent eye movement, narrow lense, distance about 0.65 m. The standard deviation was (0:019; 0:007; 0:003) m.



Divergent eye movement, narrow lense, distance about 0.65 m. The standard deviation was (0:029; 0:011; 0:002) m.

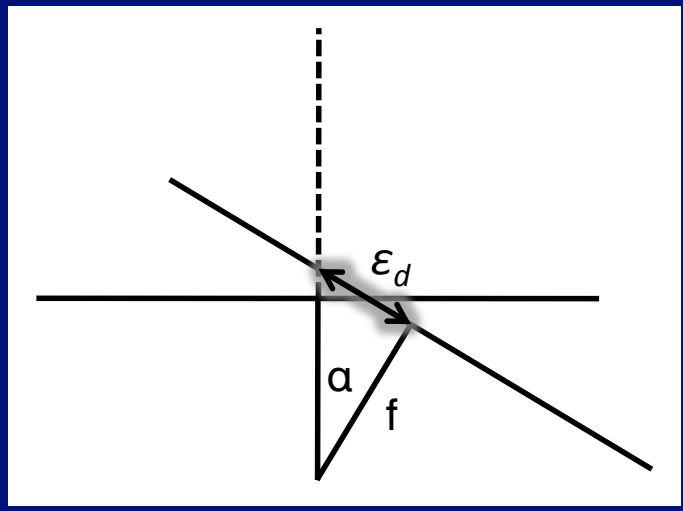
Depth errors in foveal and peripheral views

- The expected error in stereo reconstruction increases with distance and decreases with focal length:

$$\varepsilon_z = \frac{z^2}{bf}\varepsilon_d$$

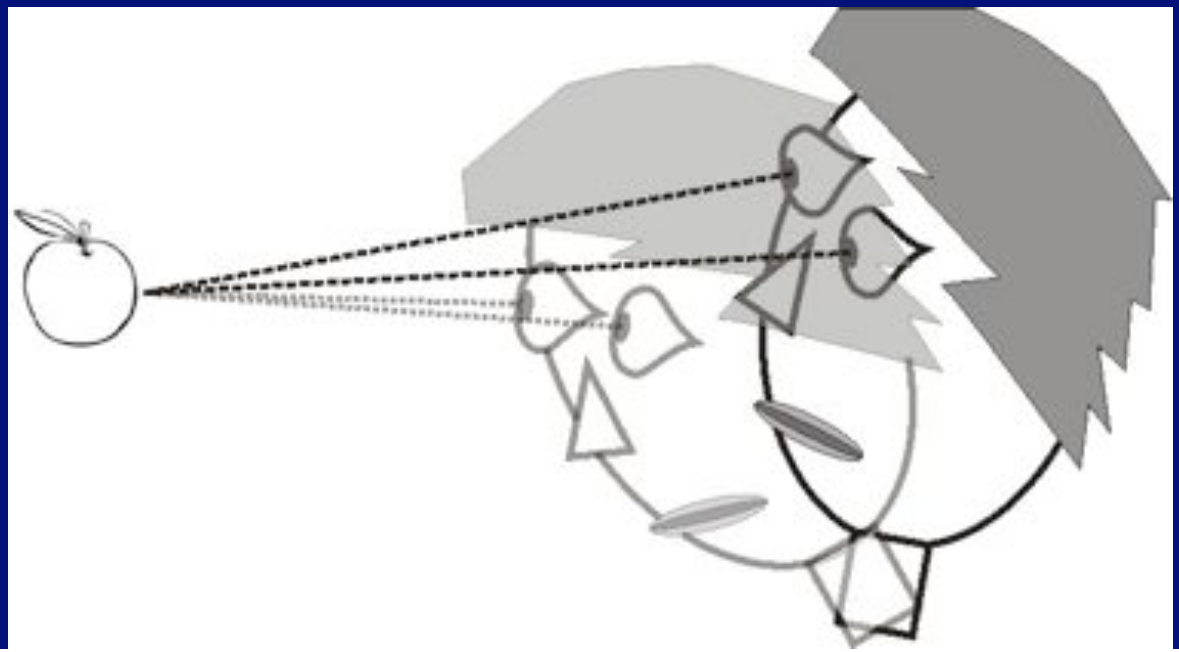
- On an active system, the disparity error increases with the focal length:

$$\varepsilon_d = f\tan(\alpha)$$



3-D object tracking: A virtual mechanism approach

- Angle of view is a function of robot configuration (not very standard description of the task)
- We can not use “off the shelf” control algorithms



Humanoid head object tracking
A virtual mechanism approach

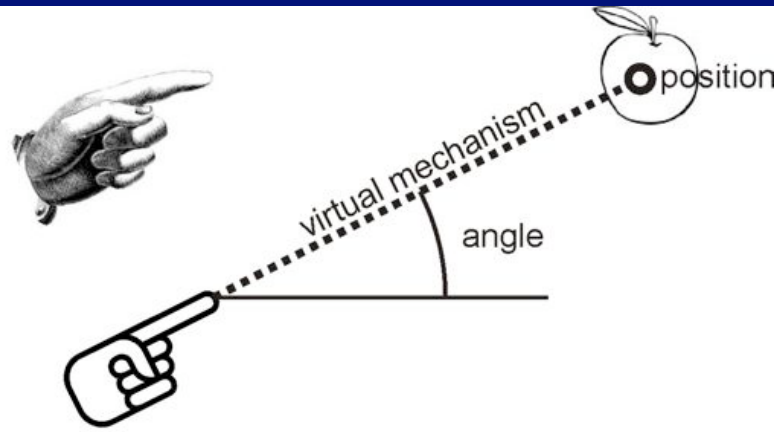
- "Virtual mechanism"
- Example:

  Task: 1 DOF (angle)

  (depends on obj. and hand position)

  Modified task using a virtual mechanism:

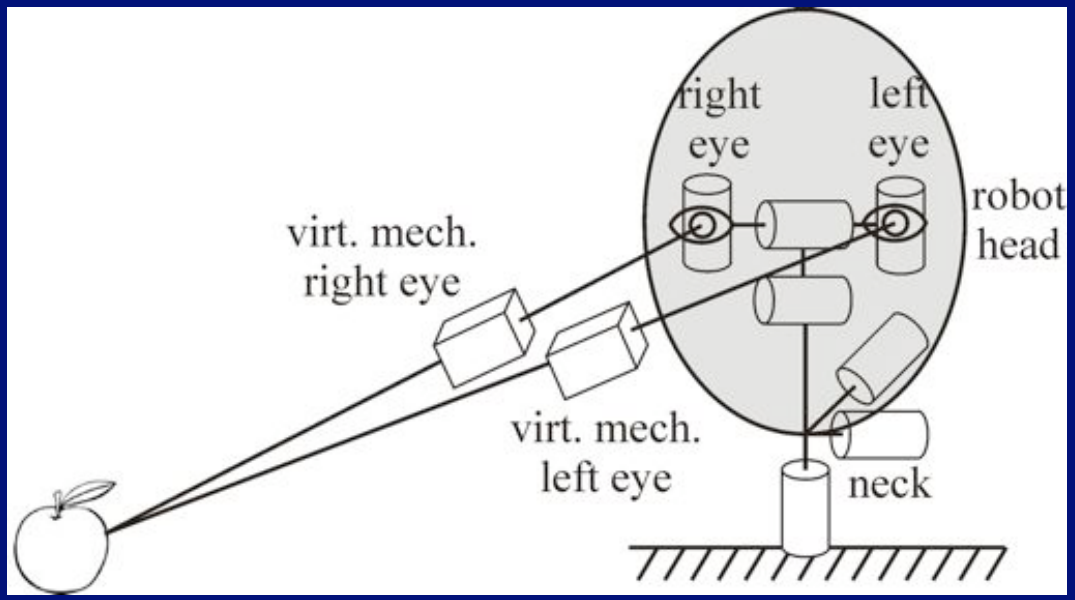
  2 DOF (object position)



Humanoid head object tracking
A virtual mechanism approach

- Two virtual mechanisms – one in each robot eye
- Task space (object position): 2 × 3 DOF = 6 DOF

  Head 7 DOF + 2 DOF = 9 DOF



Control algorithm

- Head kinematics (with virtual mechanism)

$$\boldsymbol{x}_{vm} = f(\boldsymbol{q}_{head}, l_{virt.m.})$$

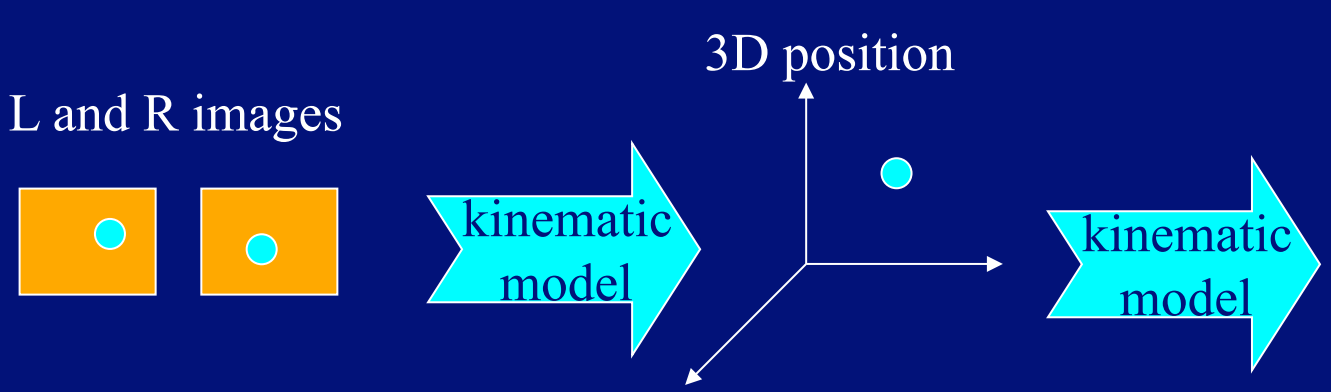
- standard velocity based controller

$$\dot{\boldsymbol{q}}_c = \mathbf{J}^{\#}\dot{\boldsymbol{x}}_{vm_c} + \mathbf{N}\dot{\boldsymbol{q}}_n$$

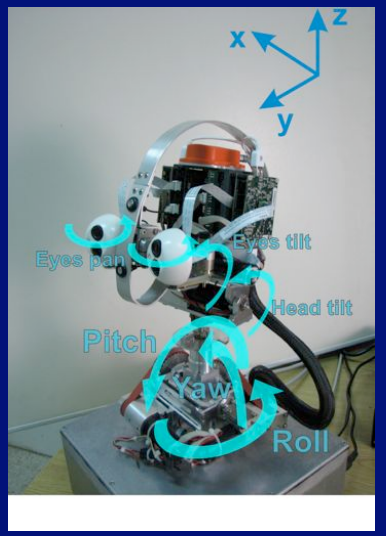
- branching mechanism

Tracking accuracy

- accuracy???
- 3D object position
- the **same** kinematic model is used twice

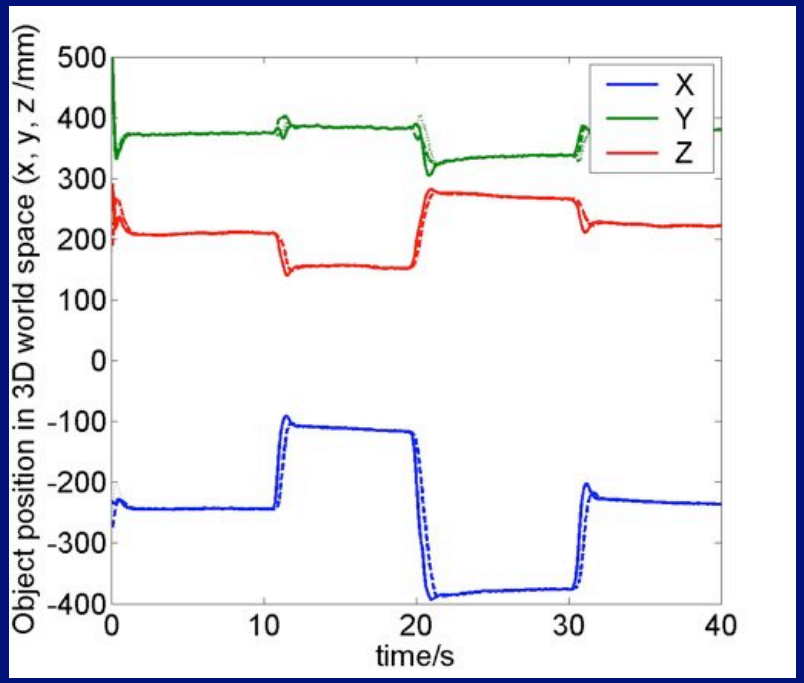


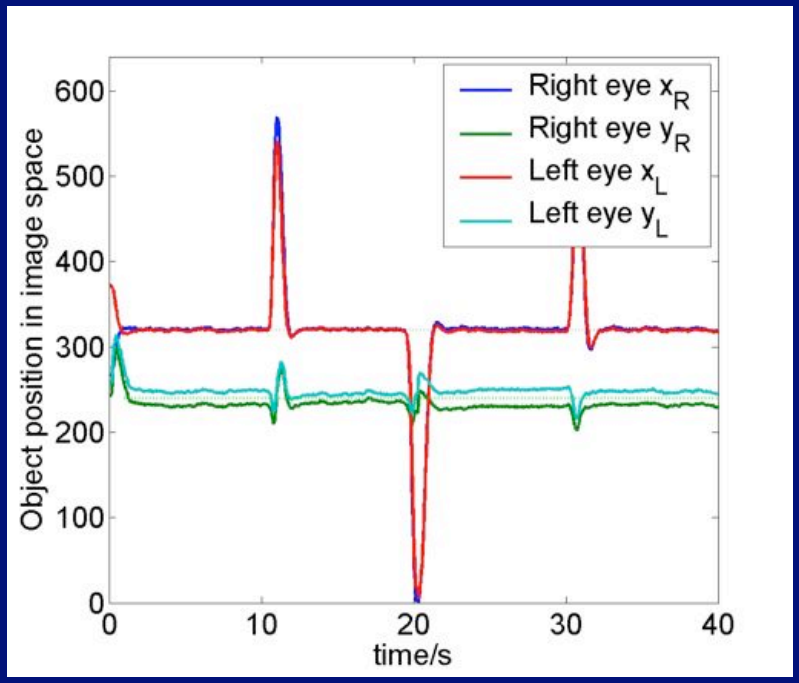
head configuration



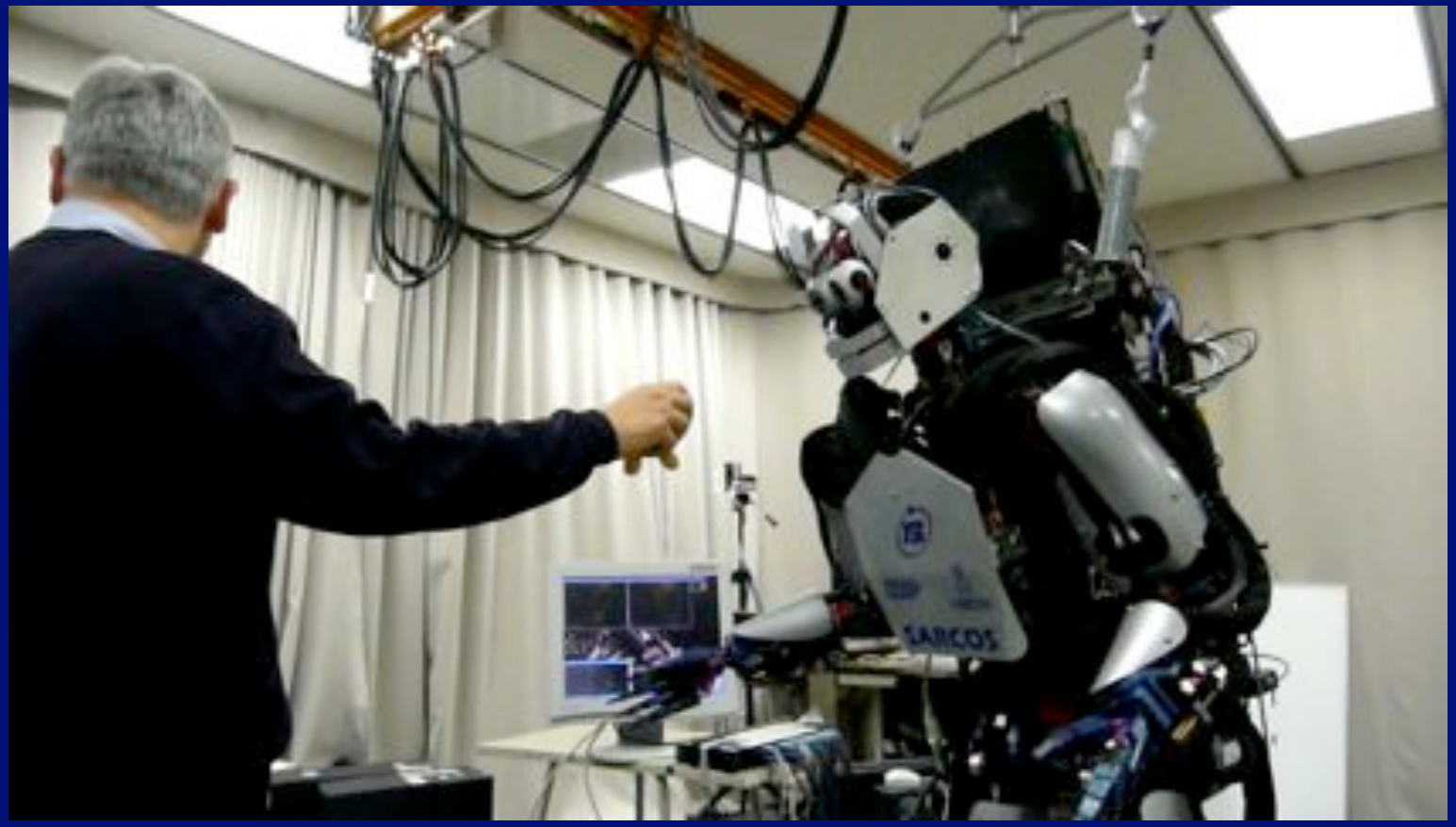
Results – tracking

- Accurate enough even when the kinematic model is not precise.





Grasping Using Active 3-D Vision



Dynamic Systems for Movement Generation

- Equations for point-to-point (discrete) movements:

$$\tau \dot{z} = \alpha_z \left(\beta_z (g - y) - z \right) + \frac{\sum_i w_i \psi_i(x)}{\sum_i \psi_i(x)} x, \quad \psi_i(x) = \exp\left(\frac{1}{2\sigma_i^2} (x - c_i)^2 \right)$$

$$\tau \dot{y} = z,$$

- Canonical system:

$$\tau \dot{x} = -\alpha_x x$$

Can we avoid using 3-D vision?

- Several problems:
  - Object must be detected in both views
  - System must be calibrated

Peripheral and foveal views

- Position in peripheral view that results in central position in foveal view

$$\hat{x}_p = \frac{\alpha_p \mathbf{r}_1 \bullet \mathbf{t} + \gamma_p \mathbf{r}_2 \bullet \mathbf{t} - \left(\alpha_p r_{13} + \gamma_p r_{23}\right) Z}{\mathbf{r}_3 \bullet \mathbf{t} - r_{33} Z}$$

$$\hat{y}_p = \frac{\beta_p \mathbf{r}_2 \bullet \mathbf{t} - \beta_p r_{23} Z}{\mathbf{r}_3 \bullet \mathbf{t} - r_{33} Z}$$

- Note:
  - Transformation between foveal and peripheral camera: $\mathbf{R} = [\mathbf{r}_1^{\mathrm{T}}, \mathbf{r}_2^{\mathrm{T}}, \mathbf{r}_3^{\mathrm{T}}]$, $\mathbf{t}$
  - Position of the point in space in foveal camera c. s.: X, Y, Z
  - Internal camera parameters: $\alpha_p, \beta_p, \gamma_p, \alpha_f, \beta_f, \gamma_f$

Standard configurations

Given that the point is in the center of the fovea, where is the point in the peripheral image?

Cameras with parallel optical axes

$$\hat{x}_p \approx -\frac{\alpha_p \mathbf{r}_1 \bullet \mathbf{t}}{Z}$$

$$\hat{y}_p \approx -\frac{\beta_p \mathbf{r}_2 \bullet \mathbf{t}}{Z}$$

Vertically displaced cameras

$$\hat{x}_p \approx 0$$

$$\hat{y}_p \approx -\frac{\beta_p t_Y}{Z}$$

Displacement in the fovea (D_x, D_y) due to the error in the periphery (d_x, d_y)

$$D_x \approx \frac{\alpha_f}{\alpha_p} d_x, \quad D_y \approx \frac{\beta_f}{\beta_p} d_y$$

Example (humanoid robot DB)

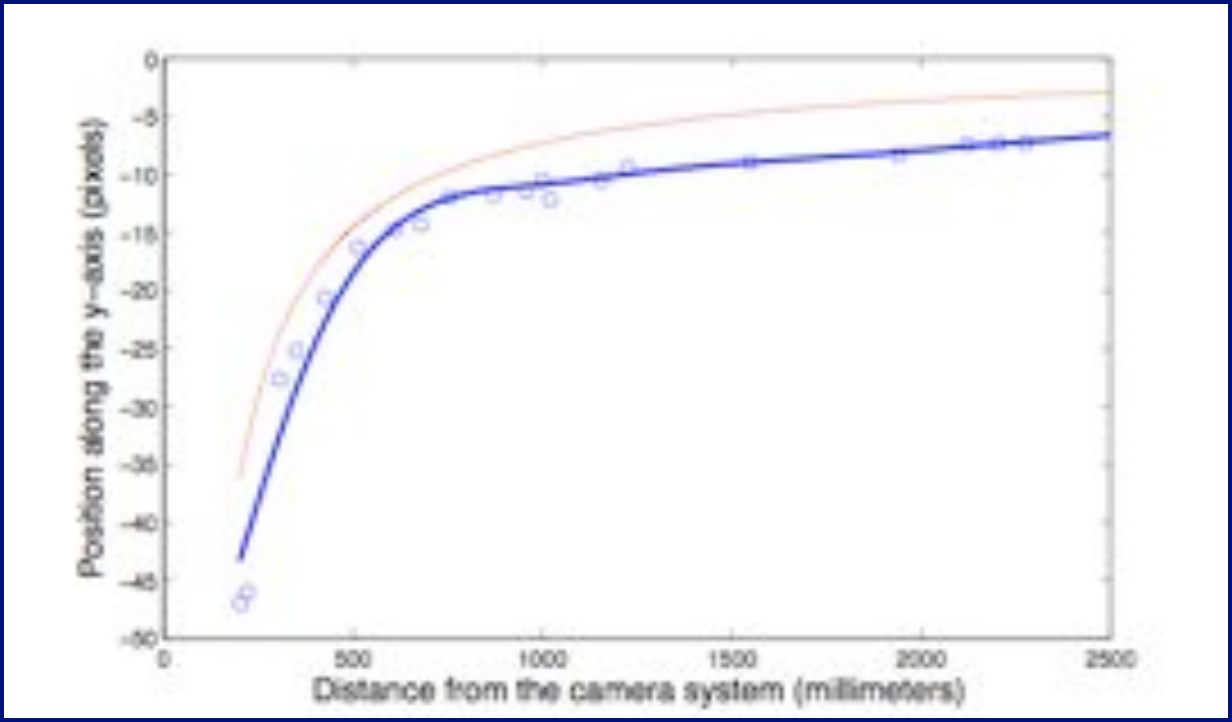
$$\mathbf{R} \approx \mathbf{I}, \quad \mathbf{t} \approx \begin{bmatrix} 0 \\ t_y \\ 0 \end{bmatrix}$$

$$\hat{x}_p \approx 0$$

$$\hat{y}_p \approx -\frac{\beta_p t_y}{Z}$$

$$\beta_p \approx 290.9$$

$$t_y \approx 25\text{mm}$$



Closed-loop image-based control

- Network of PD-controllers to exploit the redundancy of our humanoid.
- The controller network attempts to:
  - position the object in the fovea,
  - introduce cross-coupling between the eyes to help the eye movements if the object is lost in one view,
  - assist preceding joints to maintain natural posture away from the joint limits

Example controller

$$D_{\text{joint}} = \left(\theta^{*}_{\text{joint}} - \theta_{\text{joint}}\right) - K_{d}\dot{\theta}_{\text{joint}}$$

$$D_{\text{blob}} = \left(x^{*}_{\text{blob}} - x_{\text{blob}}\right) - K_{dv}\dot{x}_{\text{blob}}$$

- Left eye pan:

$$\dot{\theta}_{LEP} = K_{p}\left[K_{\text{relaxation}}D_{LEP} - K_{\text{target}\rightarrow EP}K_{v}C_{\text{LX blob}}D_{\text{LX blob}} + K_{\text{cross-target}\rightarrow EP}K_{v}C_{\text{RX blob}}D_{\text{RX blob}}\right]$$

Example Controller

$$D_{\text{joint}} = \left(\theta^{*}_{\text{joint}} - \theta_{\text{joint}}\right) - K_d \dot{\theta}_{\text{joint}}$$

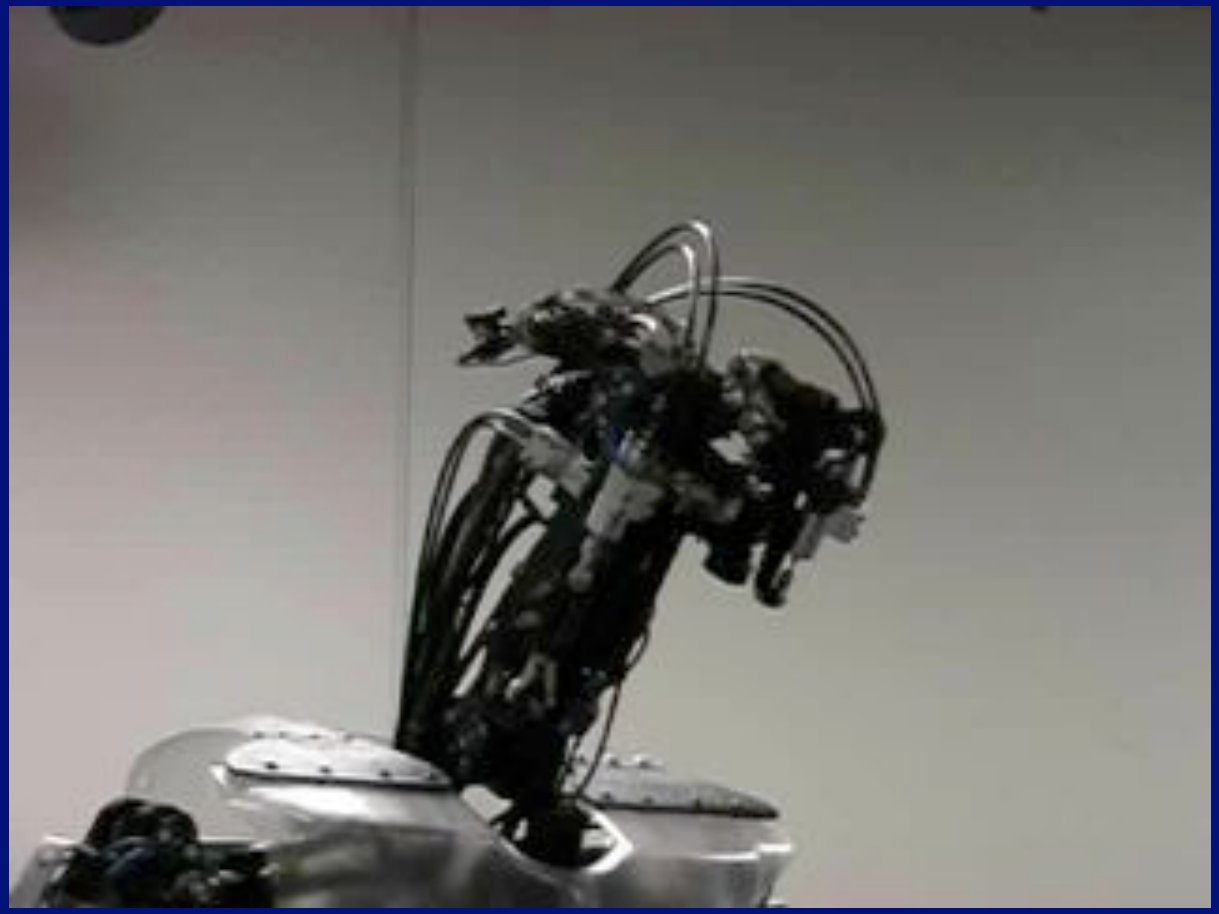
- Head nod:

$$\dot{\theta}_{HN} = K_p \left[K_{\text{relaxation}} D_{HN} - K_{ET->HN}\left(D_{\text{LET}} + D_{\text{RET}}\right)\right]$$

Control System Properties

- Accurate forward kinematics is not needed.
- The system can automatically compensate for failures in joint movements.
- When the target is not visible, the system brings the robot back to the preferred posture.

Head motion



Head motion

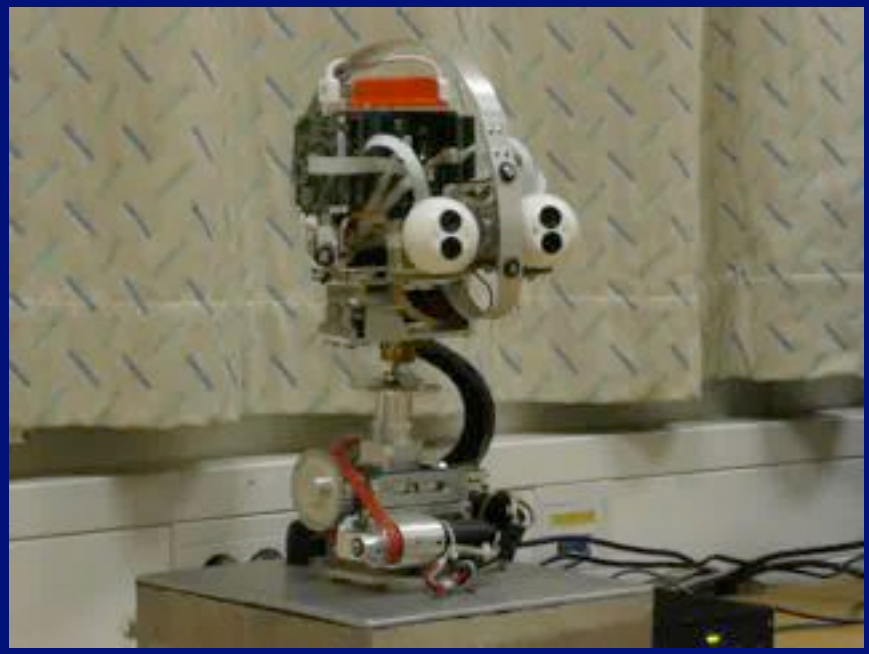
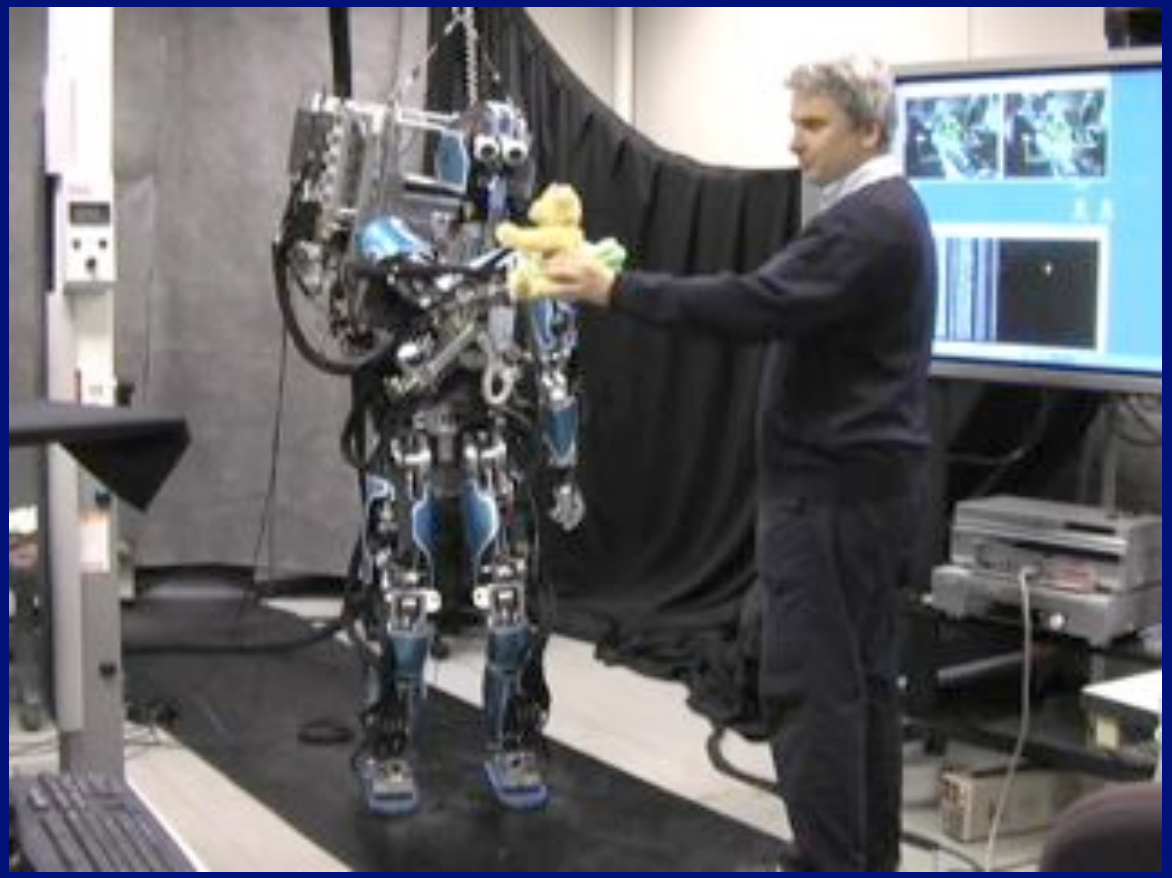


Image-based reaching



Object recognition and foveation

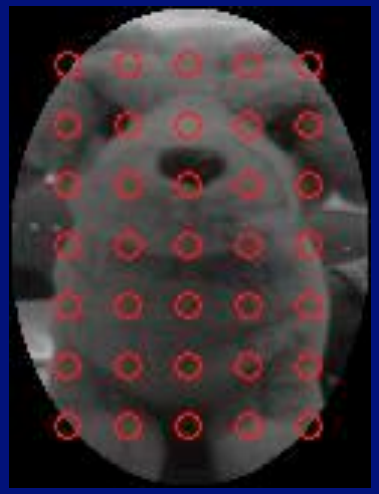
- Experiments with view-based approaches: train the system by showing the object from many viewpoints.
- Preprocess the images to achieve robustness against change in position, orientation, scale and brightness (Gabor filters).
- Classification using support vector machines.

Training

Collected images used for training

Affine warping and Gabor jets

Gabor jets consists of magnitudes of several complex values calculated by convolution with Gabor kernels at each node. Such collections are used to represent object views.



Affine warping (using results of visual tracking) makes it possible to compare Gabor jets (Wiskott et al., 1997) across views.

Nonlinear multi-class SVMs

$$\arg\max_{r\in\Omega}\left\{\sum_{i=1}^{n}\tau_{ir}\,\mathrm{K}\left(\mathbf{x}_i,\mathbf{x}\right)+b_r\right\}$$

- Scalar products arising in standard linear SVMs are replaced by symmetric kernel functions.
- The convergence of the optimization algorithm can be guaranteed for all kernels that fulfill Mercer condition.

Kernel construction

$$\left.\begin{aligned}\mathbf{I}_X &= \{\mathbf{a}_x\}_{x\in X}\\ \mathbf{J}_X &= \{\mathbf{b}_x\}_{x\in X}\end{aligned}\right\}:$$ A collection of Gabor jets calculated on grid of points **X** in images **I** and **J**, respectively.

Proposed kernel function:

$$K(\mathbf{I}_X,\mathbf{J}_X) = \exp\left(\rho\frac{1}{M}\sum_{x\in X}\left(1-\frac{\mathbf{a}_x^T * \mathbf{b}_x}{\|\mathbf{a}_x\|\|\mathbf{b}_x\|}\right)\right)$$

M : number of grid points

Results for multi-class SVMs

Tr. views per object	SVM	NNC
208	97.6%	95.9%
104	96.7%	93.7%
52	95.1%	91.5%
26	91.9%	86.7%

Fovea resolution: 160x120

Tr. views per object	SVM	NNC
208	94.2%	89.3%
104	92.4%	87.3%
52	90.7%	84.4%
26	86.7%	79.2%

Fovea resolution: 80x60

Tr. views per object	SVM	NNC
208	91.0%	84.7%
104	87.2%	81.5%
52	82.4%	77.8%
26	77.1%	72.1%

Fovea resolution: 40x30

- 14 objects
- At most 2912 views were used for SVM training.
- Models invariant against 3-D depth rotations were learned.

Learning object representations: Constraining vision by manipulation

- By taking control of the object, the robot can focus on the relevant part of the image, thus bypassing potential pitfalls of pure bottom-up attention and segmentation.
- We are looking at the following task: what kind of sensorimotor processes are needed to learn a full 3-D sensory representation of an object.

Visuomotor processes

- Motor
  - Grasp the object
  - Get the object into and away from the fovea
  - Manipulate the object to collect snapshots from various viewpoints
- Visual
  - Segment the object from the background
  - Build a model suitable for recognition

Figure-ground segmentation for snapshot acquisition

Bayesian approach (closed-world assumption):

- the object, which we model by process θ_o
- the background (θ_b),
- the hand (θ_h), and
- everything we do not know about the environment - outliers (θ_t)

Estimating object area (EM-algorithm)

- Closed-world assumption allows us to to express the probability that object o is observed in image I

$$P\left(\mathbf{I}\middle|\Theta_o\right)=\prod_{\mathbf{u}}P\left(\mathbf{I_u},\mathbf{u}\middle|\Theta_o\right)$$

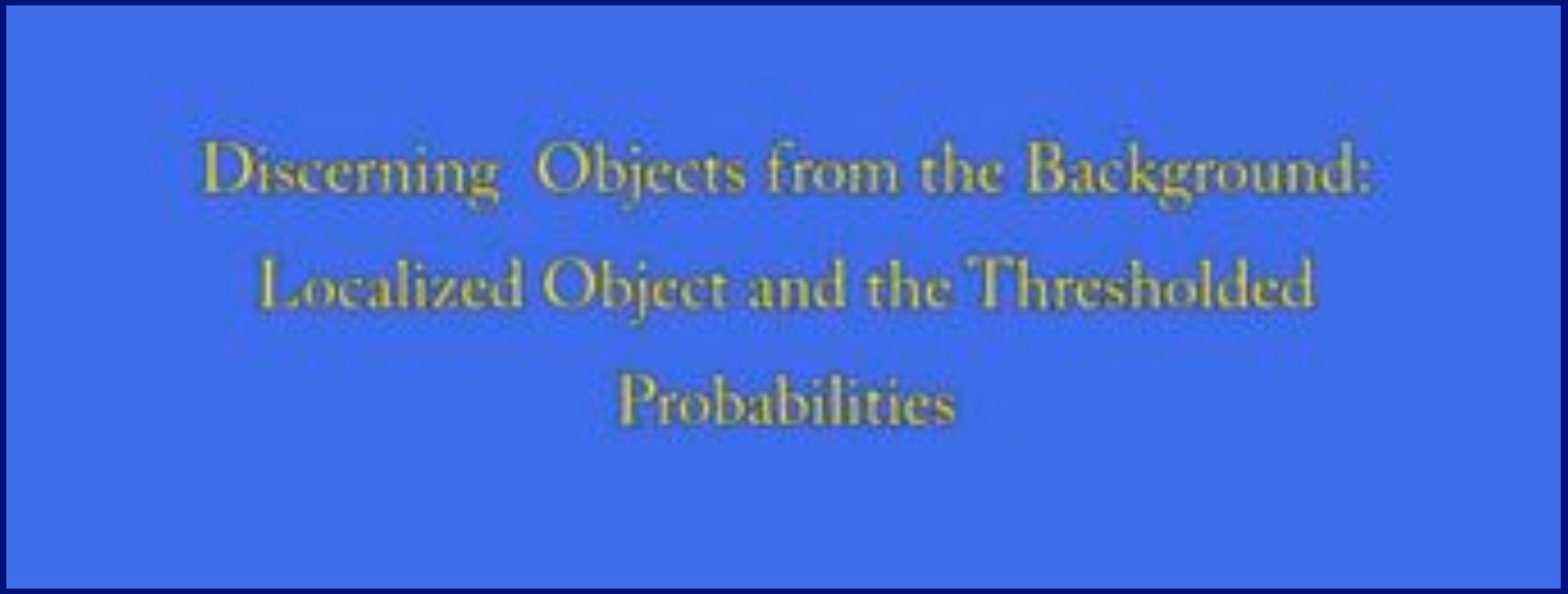
$$P\left(\mathbf{I_u},\mathbf{u}\middle|\Theta_o\right)=\frac{\omega_o P\left(\mathbf{u}\middle|\Theta_o\right)}{\omega_b P\left(\mathbf{I_u},\mathbf{u}\middle|\Theta_b\right)+\omega_h P\left(\mathbf{I_u}\middle|\Theta_h\right)+\omega_o P\left(\mathbf{u}\middle|\Theta_o\right)+\omega_t P\left(\Theta_t\right)}$$

- The position and extent of the object can be estimated by minimizing the log-likelihood

$$L\left(\Theta,\omega\right)=-\sum_{\mathbf{u}}\log\left(P\left(\mathbf{I_u},\mathbf{u}\middle|\Theta\right)\right)$$

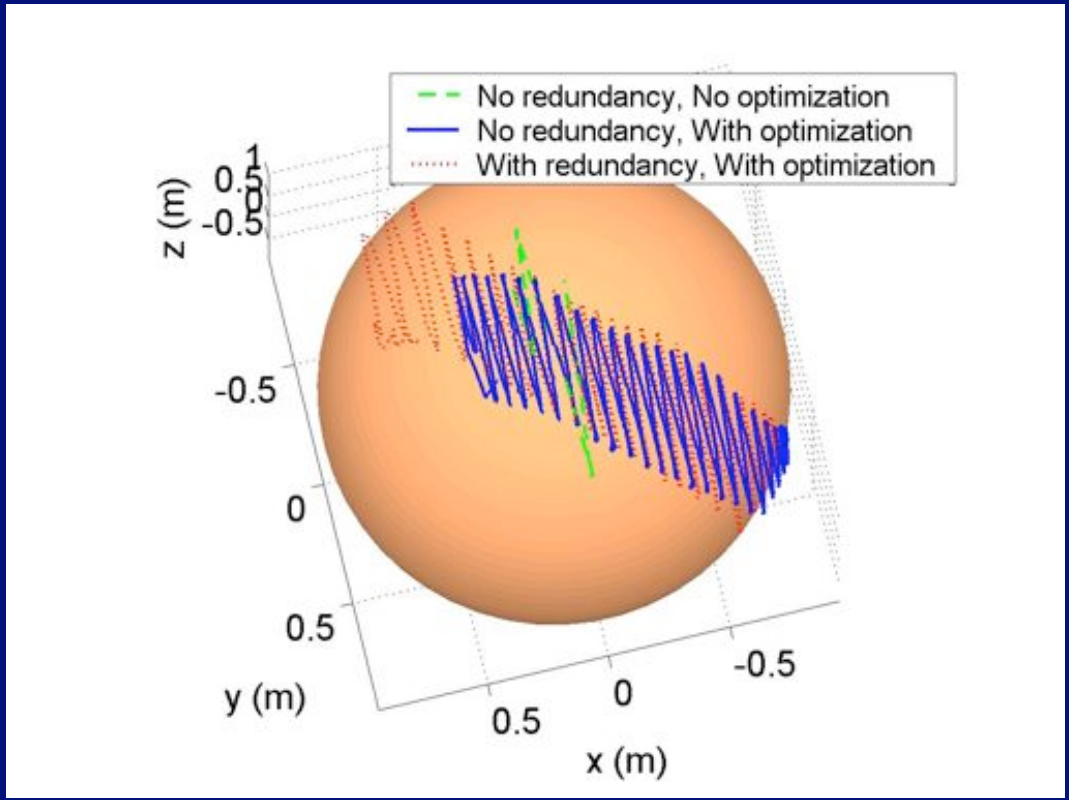
Figure-ground segmentation

- The position and extent of the object can be estimated by minimizing the log-likelihood (using EM-algorithm)



Exploiting the redundancy

- Maximize manipulability to increase the range of rotations in depth.



Implementation on humanoid robot ARMAR

Summary

- Evaluation of 3-D vision on a humanoid robot
- Grasping using active 3-D vision
- Head control and foveated vision
- Object recognition and foveated vision
- Acquiring models of objects without having any prior knowledge about them

Collaborators

- Mitsuo Kawato
- Tomohiro Shibata
- Gordon Cheng
- Tamim Asfour
- Damir Omrčen
- Chris Gaskett
- …