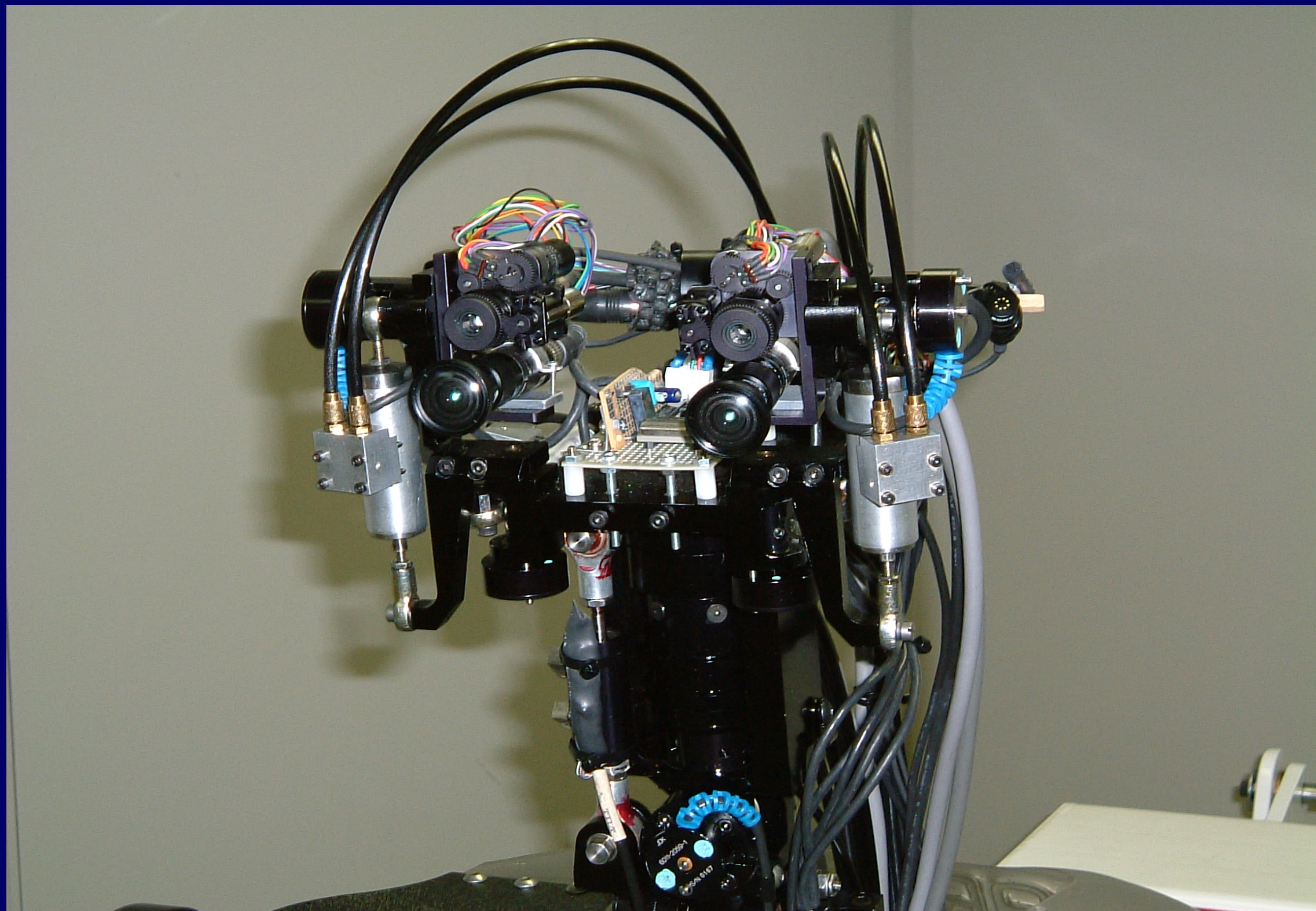
Foveated Vision and Object Recognition on Humanoid Robots

Aleš Ude

Japan Science and Technology Agency,
ICORP Computational Brain Project

Jožef Stefan Institute, Dept. of Automatics,
Biocybernetics and Robotics

Foveated Humanoid Vision System



Foveated Vision

- Vision should be able to deal with fast robot movements and occlusions.
- Motor control should be able to deal with vision failures.

Tight integration between perception and motor control is necessary.

Objective

- Our goal is to integrate peripheral and foveal vision with the motor control and to use foveal vision for the task for which it is suited best:

Object recognition in dynamic scenes

Motor Control for Foveated Vision

- Two goals:
  - position the object in the fovea of each eye
  - move smoothly to enable processing of foveal images

- 3-D stereo vision using actuated, head-mounted cameras is difficult due to inaccurate joint angle readings, delays, and vibrations.

⇓

Uncalibrated stereo results in smoother movements.

Closed-Loop Control System

- 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:

$$\dot{\theta}_{\text{L}EP} = K_p \Big[K_{\text{relaxation}} D_{\text{L}EP} - K_{\text{target}\to EP} K_v C_{\text{LX target}} D_{\text{LX target}} + K_{\text{cross-target}\to EP} K_v C_{\text{RX target}} D_{\text{RX target}} \Big]$$

Example Controller

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

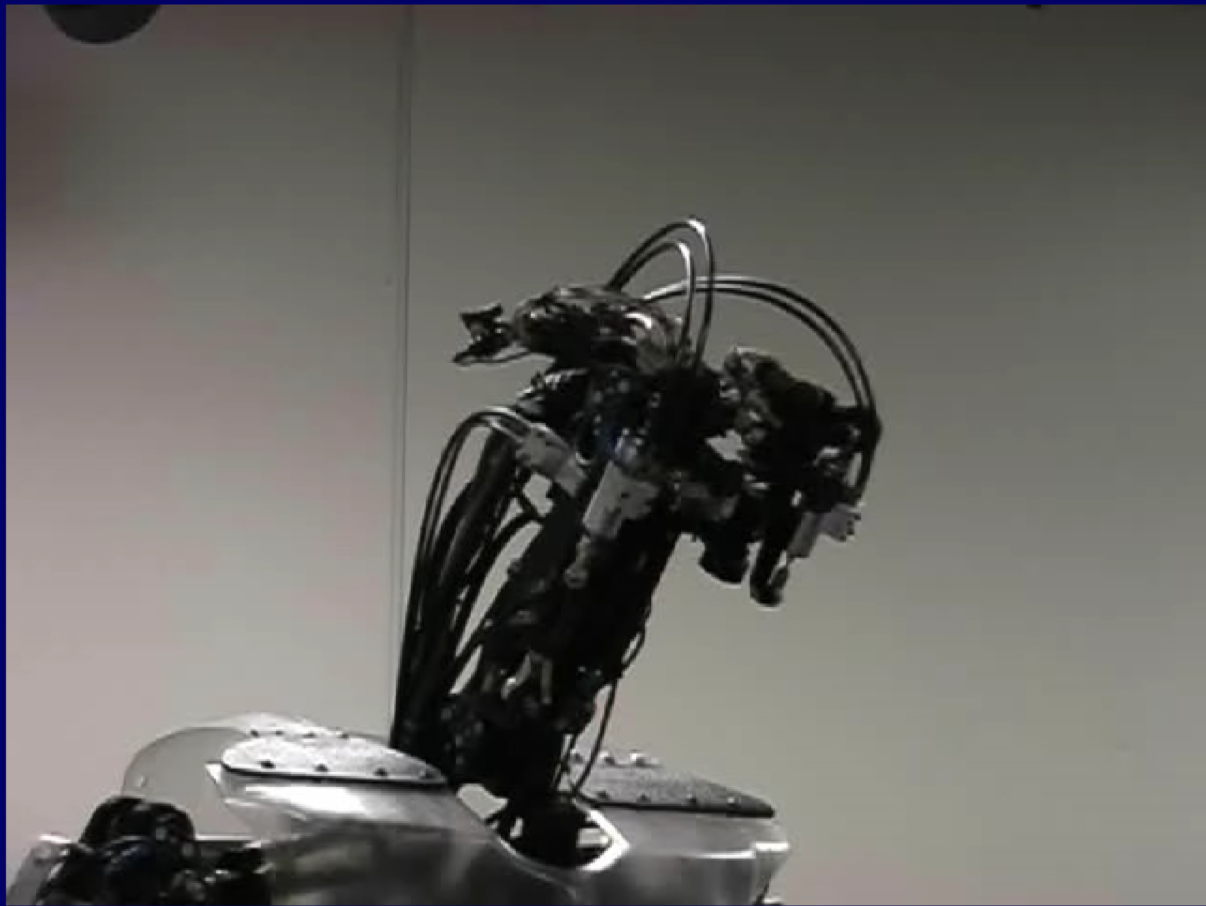
- Head nod:

$$\dot{\theta}_{HN} = K_{p}\left[K_{\text{relaxation}} D_{HN} - K_{ET->HN}\left(D_{\text{L}ET} + D_{\text{R}ET}\right)\right]$$

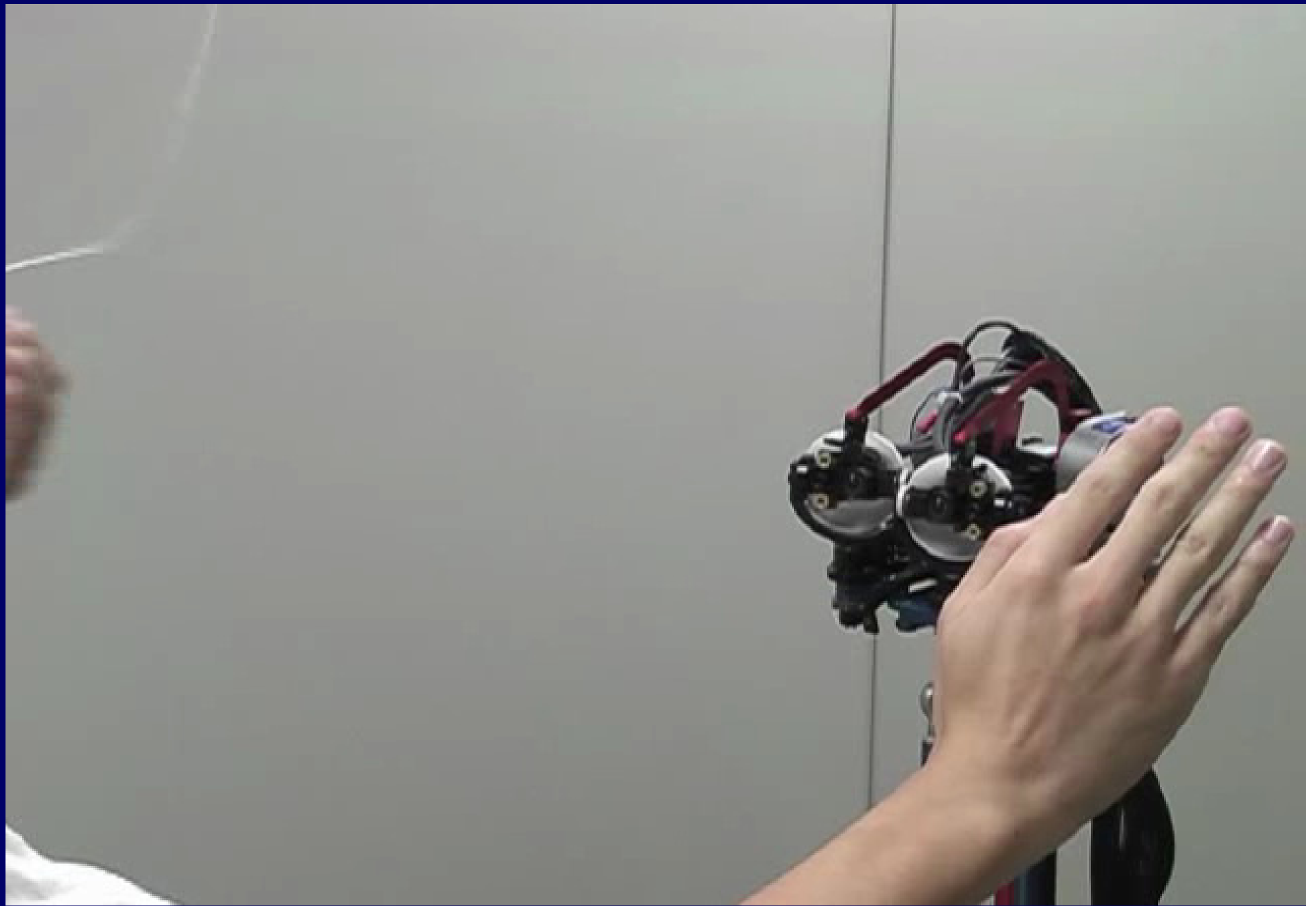
Control System Properties

- 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.

DB head motion



DB2 head motion



Positioning in the fovea

- Foveal cameras are vertically displaced from peripheral cameras.
- Vertical displacement from the central position in peripheral views to get the object in the center of foveal views.
- Constant displacement is sufficient because peripheral cameras are equipped with very wide angles lenses.

Is it useful for foveal vision?

Vision System Overview

- Visual search in peripheral images for object detection that initializes tracking.
- Results of tracking used to control the robot and normalize the images.
- Early processing for feature extraction.
- Learning and classification.

Probabilistic Approach

- Gaussian color mixture models

$$p(\mathbf{I_u} \mid \Theta_l) = \sum_{k=1}^{K_l} \frac{\omega_{l,k} \exp\left(-\frac{1}{2} (\mathbf{I_u} - \mathbf{I}_{l,k})^{\mathrm{T}} \Sigma_{l,k}^{-1} (\mathbf{I_u} - \mathbf{I}_{l,k}) \right)}{\sqrt{(2\pi)^{2 \text{ or } 3} \det(\Sigma_{l,k})}}$$

- Random search to trigger the system to attend a high probability region.
- Automatic threshold selection.

Tracking

- Probabilistic approach:
  - color distributions
  - pixel (shape) distributions

$$p(\mathbf{u} \mid \Theta_l) = \frac{1}{2\pi\sqrt{\det(\Gamma_l)}} \exp\left(-\frac{1}{2} (\mathbf{u} - \mathbf{u}_l)^{\mathrm{T}} \Gamma_l^{-1} (\mathbf{u} - \mathbf{u}_l) \right)$$

  - EM algorithm to minimize log-likelihood with respect to shape $\{\mathbf{u}_l, \Gamma_l\}$ and mixture parameters $\{\omega_l\}$.
- Real-time implementation (60 Hz).

Tracking and Pursuit

Normalization

- To compare the images, we must account for changes in object position and scale.
- The tracker calculates an approximation for the object's position, orientation and scale.
- This limits our recognition system to the object we can track, but this is a necessary prerequisite to get the object into fovea anyway.

Normalization

To compare the images, we must account for changes in object position and scale

Affine warping

- In our system, affine warping accounts for translations, scale changes and planar rotations

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- As in all viewpoint-dependent models, we must account for rotations in depth by collecting sufficient amount of training data.

Object Representations

- Geon structural description models: non-accidental properties (Marr, Biederman, …)
  - difficult to build such descriptions
- View-based approaches: collections of viewpoint dependent surfaces and contours (Tarr, Poggio, Bülthoff, …)
  - problems with generalization over different instances of a perceptually defined class

Object Recognition System

- View-based approach: train the system by showing the object from many viewpoints.
- Preprocess the images to achieve robustness against change in position, orientation, scale and brightness and to extract relevant features.
- Classification using support vector machines.

Training Data Collection

Training Images



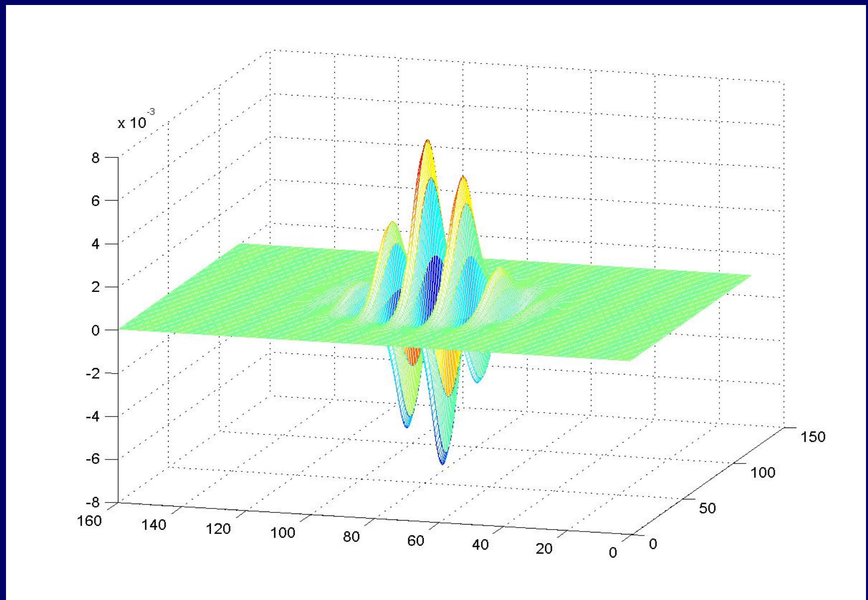

- No accurate turntables to systematically capture all possible viewpoints.
- Is such training data good enough for recognition?

Feature Extraction

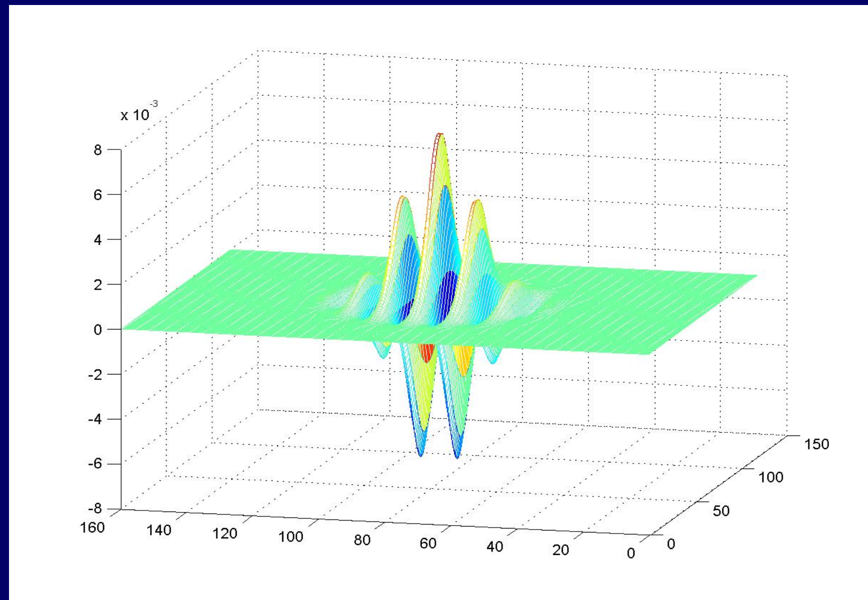
- Gabor filters are defined as a convolution of the image with a family of Gabor kernels:

$$\Theta_{\mathbf{k}}(\mathbf{x}) = \frac{\|\mathbf{k}\|^2}{\sigma^2} \exp\left(-\frac{\|\mathbf{k}\|^2 \|\mathbf{x}\|^2}{2\sigma^2} \right) \cdot \left\lfloor \exp(i\mathbf{k} * \mathbf{x}) - \exp\left(-\frac{\sigma^2}{2} \right) \right\rfloor$$

Gabor Kernels



Odd Gabor wavelets (Gaussian modulated by sine wave)



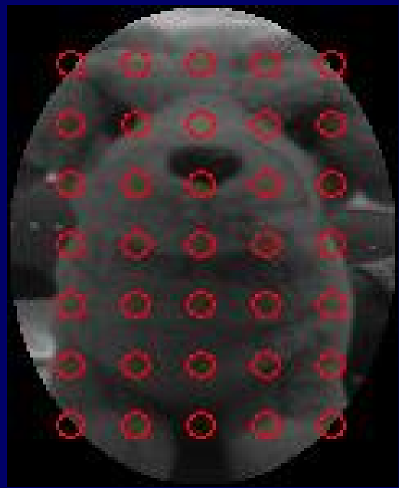
Even Gabor wavelets (Gaussian modulated by cosine wave)

Properties of Gabor Kernels

- Biological relevance: they have similar shape as the receptive fields of simple cells in the visual cortex.
- Machine vision: Best tradeoff for localization in image and frequency space, which yields robustness against small distortions, rotation, and scaling.

Gabor Jets

- Gabor jets (Wiskott et al.) are calculated at each node. It has been suggested to compute jets with:



$$\mathbf{k}_{m,n} = 2^{-\frac{m+2}{2}}\pi\begin{bmatrix}\cos(n\pi/8)\\ \sin(n\pi/8)\end{bmatrix},\quad \begin{matrix} \mathrm{m}=0,\ldots,4 \\ \mathrm{n}=0,\ldots,7\end{matrix}$$

- New representation consists of magnitudes of 40 complex values calculated by convolution with image patch at each node.

Improving Performance

- Compute convolutions with Gabor filters using Fourier transform.
- This makes it possible to calculate Gabor jets at
  - 15 Hz for 40 different orientations and scales
  - 30 Hz for 16 different orientations and scales.

Model Representation with Gabor Jets

- Images of all objects, which need to be included in the database, from many different viewpoints are acquired to account for rotations in depth.
- Conversion into Gabor jets at each lattice node.
- Initially, we used PCA to extract a reduced number of features.

Learning Machines for Classification

- Data: observatio ns $\mathbf{x}_i \in \Re^n$

  and the asocciated class $y_i \in \{-1,1\}$
- Assumption: data is drawn from an unknown distribution P($\boldsymbol{x}$,y).
- The task of a learning machine is to learn mapping $\boldsymbol{x}_i \to y_i$.
- The machine is defined by a set of posssible mappings $\{f(\boldsymbol{x},\alpha)\}$.

Support Vector Machines for Classification

- SVMs are based on class of hyperplanes:

$$(\mathbf{w} * \mathbf{x}) + b = 0, \ \ \mathbf{w} \in \Re^n, \ \ b \in \Re$$

corresponding to decision functions

$$\operatorname{sgn}((\mathbf{w} * \mathbf{x}) + b)$$

- Optimal linear SVMs can be calculated by solving a quadratic program.

Nonlinear SVMs

- Decision functions

$$\mathrm{f}(\mathbf{x}) = \mathrm{sgn}\left(\sum_i \alpha_i y_i \, \mathrm{K}(\mathbf{x}_i, \mathbf{x}) + b \right),$$

- Polynomial kernels: $\mathrm{K}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} * \mathbf{y} + 1)^p$
- RBF kernels: $\mathrm{K}(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / (2\sigma^2)\right)$
- Sigmoid kernels: $\mathrm{K}(\mathbf{x}, \mathbf{y}) = \tanh\left(\kappa(\mathbf{x} * \mathbf{y}) - \delta\right)$

Recognition with SVMs

We considered two different problems:

- Given one object, decide whether this object is in the image or not.

  $\rightarrow$ can be solved by one SVM
- Decide which of the objects from the database is in the image.

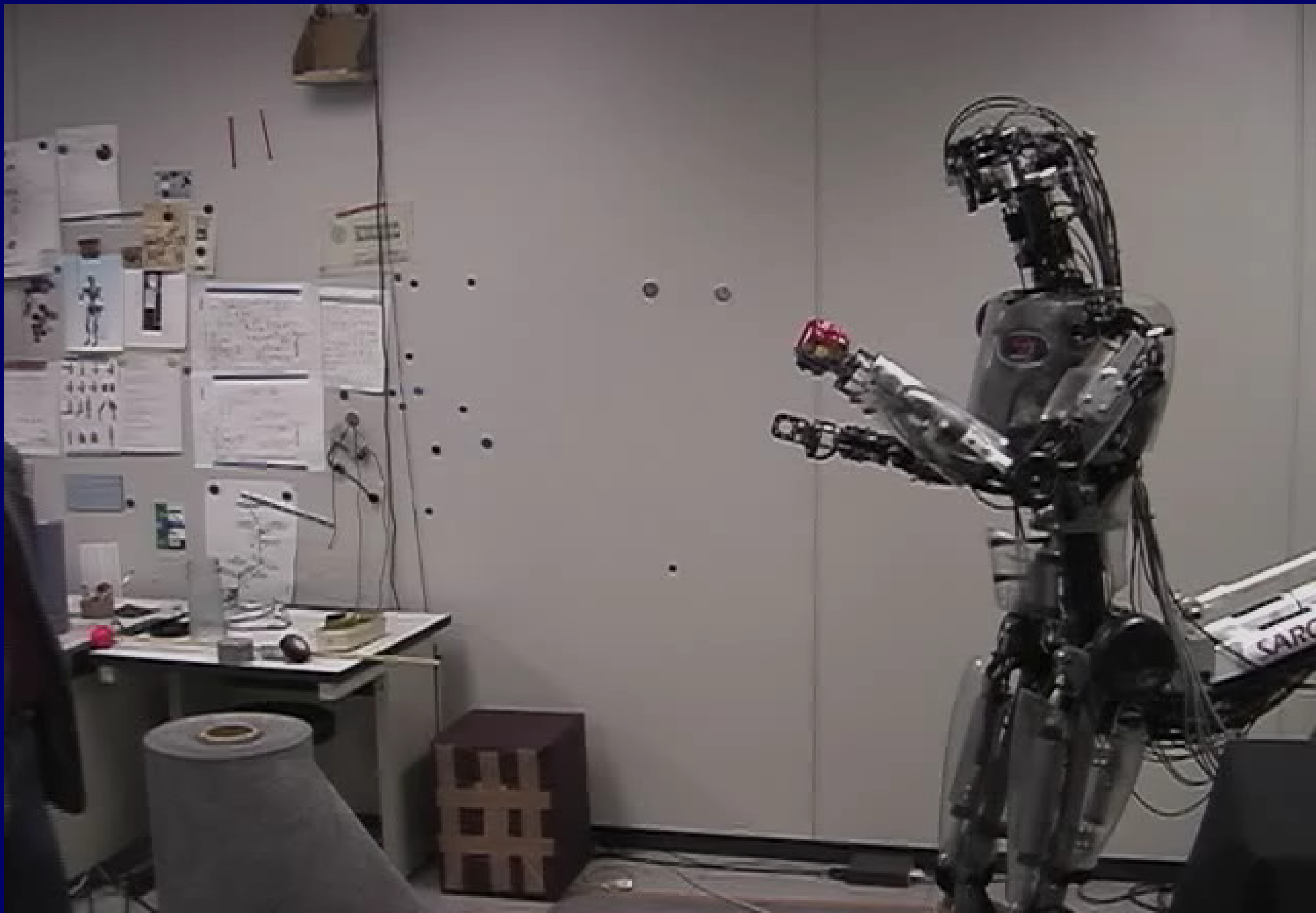
  $\rightarrow$ many SVMs organized in a tree structure

Implementation Details

- System implemented on 3 PCs that communicate over Ethernet

  First PC: Detection & Tracking
  Second PC: Classification
  Third PC: Closed-loop kinematic control and communication with DB.

- Time needed for classification (in case of Gabor jets + RBF SVMs): between 15 and 30 Hz (at 160x120 resolution)



Statistical tests (less training images)

	False positives	False negatives
Teddy Bear 1	4.5 %	0.5 %
Teddy Bear 2	7.8 %	0.3 %
Teddy Bear 3	1.4 %	0.9 %
Toy Dog	3.9 %	2.7 %
Coffee Mug	2.1 %	0.7 %

200 images / object, 120 x 160 pixels, linear SVMs

	False positives	False negatives
Teddy Bear 1	9.9 %	0.3 %
Teddy Bear 2	13.8 %	0.3 %
Teddy Bear 3	13.6 %	0.1 %
Toy Dog	11.1 %	2.3 %
Coffee Mug	2.1 %	0.1 %

100 images / object, 120 x 160 pixels, linear SVMs

Statistical tests (resolution reduction)

	False positives	False negatives
Teddy Bear 1	5.8 %	0 %
Teddy Bear 2	6.8 %	0.3 %
Teddy Bear 3	3.1 %	0 %
Toy Dog	2.7 %	0.3 %
Coffee Mug	0.8 %	0 %

200 images / object, 120 x 160 pixels, RBF SVMs

	False positives	False negatives
Teddy Bear 1	9.1 %	0.5 %
Teddy Bear 2	9.3 %	1.0 %
Teddy Bear 3	10.5 %	0.3 %
Toy Dog	8.9 %	3.0 %
Coffee Mug	9.5 %	0 %

200 images / object, 45 x 60 pixels, RBF SVMs

Statistical tests (linear and nonlinear SVMs)

	False positives	False negatives
Teddy Bear 1	5.8 %	0 %
Teddy Bear 2	6.8 %	0.3 %
Teddy Bear 3	3.1 %	0 %
Toy Dog	2.7 %	0.3 %
Coffee Mug	0.8 %	0 %

200 images / object, 120 x 160 pixels, RBF SVMs

	False positives	False negatives
Teddy Bear 1	4.5 %	0.5 %
Teddy Bear 2	7.8 %	0.3 %
Teddy Bear 3	1.4 %	0.9 %
Toy Dog	3.9 %	2.7 %
Coffee Mug	2.1 %	0.7 %

200 images / object, 120 x 160 pixels, linear SVMs

Conclusion

Gabor jets + RBF-based support vector machines offer a very robust and reliable way to identify and recognize objects in the humanoid's foveal views.

Biologically oriented system: foveated vision, Gabor filtering, view-based recognition.