International Journal of Humanoid Robotics
Vol. 5, No. 2 (2008) 267–286
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

MAKING OBJECT LEARNING AND RECOGNITION AN ACTIVE PROCESS

ALEŠ UDE

Department of Automatics, Biocybernetics, and Robotics,
Jožef Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia

Department of Humanoid Robotics and Computational Neuroscience,
ATR Computational Neuroscience Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288, Japan
aude@atr.jp

DAMIR OMRČEN

Department of Automatics, Biocybernetics, and Robotics,
Jožef Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
damir.omrcen@ijs.si

GORDON CHENG

Knowledge Creating Communication Research Center,
National Institute of Information and Communication Technology,
2-2-2 Hikaridai Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

ICORP Computational Brain Project,
Japan Science and Technology Agency,
4-1-8 Honcho, Kawaguchi, Saitama, Japan

Department of Humanoid Robotics and Computational Neuroscience,
ATR Computational Neuroscience Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
gordon@atr.jp

Received 12 July 2007
Accepted 11 January 2008

The exploration and learning of new objects is an essential capability of a cognitive robot. In this paper we focus on making use of the robot's manipulation abilities to learn complete object representations suitable for 3D object recognition. Taking control of the object allows the robot to focus on relevant parts of the images, thus bypassing potential pitfalls of purely bottom-up attention and segmentation. The main contribution of the paper consists in integrated visuomotor processes that allow the robot to learn object representations by manipulation without having any prior knowledge about the

objects. Our experimental results show that the acquired data is of sufficient quality to train a classifier that can recognize 3D objects independently of the viewpoint.

Keywords: Cognitive behavior; active vision; humanoid vision; object recognition; object learning.

1. Introduction

Human vision is very effective at segmenting images into their meaningful constituents and focusing on the perceptually relevant parts, but this property has proven to be difficult to replicate with machine vision. Finding objects in images without any prior knowledge is a hard problem and is very difficult if not impossible to achieve in a purely bottom-up manner. Passive computer vision systems — see e.g. Ref. 14 — often attempt to solve it by introducing top-down processes, which convey information about the objects that assists the linking of early features into larger groupings. It is hoped that this process will eventually result in a meaningful scene decomposition, which can then be used to learn object representations suitable for recognition and other tasks.

Unfortunately, it is not easy to formulate the top-down processes guiding the search for objects in a completely general way. We take the view that statistical approaches involve a hard time learning how to generate such image decompositions from example images because the decomposition of images as done by people depends on the experience we gain when we interact with the environment. This information is not readily available in the images but rather comes from the experience of how our actions affect the external world. It is not clear how such knowledge could be brought into the learning process on a passive system.

A humanoid robot, however, has the potential to explore its world using causality, by performing probing actions and learning from the response.[4] It has been shown that poking an object can extract visual evidence for the boundary of the object, which is useful for segmentation.[3] These early approaches demonstrated that by actively exploring the environment, the robot can gain some knowledge about the objects in its surroundings. Here we demonstrate that by making use of its manipulation capabilities, the robot can provide enough grounding information to solve difficult early segmentation and feature grouping problems, which allows it to learn complete 3D object representations suitable for recognition.

Besides being able to segment an unknown object from the background, the robot must be able to observe it from all relevant viewing directions. We use visual servoing techniques to realize the observation of an object manipulated by the robot. Although visual servoing has been studied extensively in the past, effective object observation can be quite difficult to realize in practice because of the physical limitations of the robot. These limitations include kinematic singularities throughout the workspace and the robot joint limits, which can be exceeded during the manipulator motion. However, the visual servoing task does not constrain all DOFs of a humanoid robot. By exploiting its redundancy the robot can achieve a wider range

of motion.[9,11] In this paper we propose a method that exploits the redundancies of a humanoid robot to achieve a wider range of motions with respect to the rotation of the object in depth.

Object recognition is a prerequisite for an autonomous robot. Many of the successful recognition systems are view-based and build suitable representations from snapshots of objects.[8,12,13,15,17] While early approaches used the collected patterns of objects without much preprocessing, most of the current works use local image features, e. g. scale-invariant feature transform (SIFT keys)[8] or Gabor jets.[18] The methods proposed in this paper enable the robot to obtain data required by this kind of techniques.

Humanoid robots (see Fig. 1) are our target platform because they are most suitable for cognitive tasks such as autonomously learning representations of new objects. However, to facilitate the validation process we also utilized a 7 DOF Mitsubishi PA-10 manipulator arm combined with an external camera system in some of our experiments. This is useful because many of the problems arising in learning object representations are equivalent for the two systems, but it is easier to carry out experimental validation on a simpler system.

1.1. *Outline of the approach and discussion*

We designed an interactive system for learning object representations. The user starts the learning procedure by placing a new object into the robot's hand, thus enabling the robot to grasp and manipulate the object. The main idea is that by having control of the object, the robot can bring enough knowledge into the system to ensure that it can segment the object from the background. This allows the robot to capture object snapshots needed for learning appearance-based representations.

The initial robot action is to move the object away from the camera view and to learn probabilistic background distributions based on features such as color and

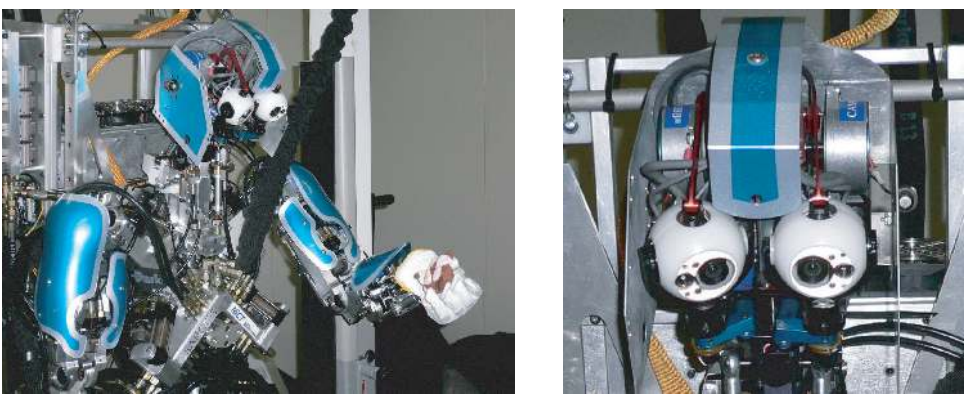


Fig. 1. Humanoid robot *i*–1, which was used in some of our experiments.

disparity. After learning the background model, the robot moves its hand to the starting position for the acquisition of object views. At this configuration, the object is placed in such a way that its projection falls onto the center of the camera image and the size of the object's image is suitable for learning (big enough to ensure proper resolution of snapshots but not too close to the image boundary). From this configuration, the robot begins to rotate the object about the axes, which causes the object to rotate in depth.[a] At the same time, the object is kept in the center of the image and its image size should remain constant. Once all objects to be learnt are grasped and snapshots are acquired from all relevant viewpoints, a suitable object representation can be generated, such as a classifier for object recognition.

The main contribution of this paper is the integrated sensorimotor processes that enable the robot to segment an object from the background and acquire snapshots of the object from all relevant viewpoints without having any prior information about it. The proposed techniques are based on the assumption that the robot can find and grasp the objects. In the current version of the system, the humanoid robot grasps the objects in an interactive way. Alternatively, the robot could attempt to generate initial hypotheses about the existence of an object automatically, such as by extracting visual features hinting at the presence of the object. Such hypotheses could be tested by attempting to grasp the hypothesized objects. While this is a difficult and very relevant problem, it is outside the scope of this paper.

2. Movement Behaviors for Observation

To gain a complete viewpoint-independent representation, the robot should observe the object across the whole 3D view sphere. For this purpose we developed control algorithms that achieve an optimal scanning behavior by actively controlling the arm in the null space, continuously optimizing the manipulability[19] of the robot for depth rotations. In this way the configuration of the robot is much more appropriate for observation and we can achieve a wider range of viewing directions without regrasping the object. The goal of the manipulation process is to first bring the object into the view of the robot cameras at an optimal size for observation, which is followed by rotating the object so that it can be observed over a continuous portion of a 3D view sphere. This process needs information about the object's position and size, which is provided by the algorithms described in Sec. 3.

In this section we focus on the following control processes:

- A movement that allows the robot to estimate the transformation from the camera to the world coordinate system at a given eye configuration.
- An explorative behavior that can be used to determine the optimal 3D position of a new object with respect to the robot's eyes. The goal is to place the object at a 3D location such that its 2D image projects onto the image center at the

[a]Depth rotations are rotations that cause a different part of a 3D object to be visible in an image.

appropriate size for learning, i.e. it covers a significant portion of the image while being away from the image boundary.

- An explorative behavior that can be used to observe the grasped object from various viewpoints. Due to occlusions and limited manipulation capabilities of humanoid robots, it is unavoidable to regrasp the observed object to ensure that the robot looks at it from all relevant viewpoints. However, the number of necessary grasps can be reduced by performing the exploratory movements in an optimal way so that the redundancy of the humanoid is exploited and the manipulability of its arm is maximized.

2.1. *Hand–eye transformation*

Initially the humanoid head is driven to an arbitrary location in space suitable for observation. Since we do not assume a fully calibrated system, the position and orientation (extrinsic parameters) of the eyes in the robot coordinate system are not known after motion. Thus once the initial configuration of the eyes is fixed, we need to learn the relationship between the robot and the eye coordinate system.

Assuming that we can acquire the position of the hand by vision, we can estimate the transformation between the two coordinate systems by performing a random arm motion observed by the robot's camera. Every point that differs enough from the previous points in the image or world coordinate system is saved. Once enough points have been acquired, the camera transformation matrix can be calculated. Since these are standard computational processes, we omit the details here.

2.2. *Centering the object at optimal distance*

With the known hand–eye transformation we can place the object grasped by the robot into the center of the camera image (see Fig. 3). We realized this behavior using an analytical expression for the image Jacobian, which defines the relationship between velocities of the point in the 3D world ($[\dot{x}, \dot{y}, \dot{z}]^T$) and the 2D image velocities ($[\dot{u}, \dot{v}]^T$):

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \mathbf{J}_{\text{im}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \tag{1}$$

The parameters of $\mathbf{J}_{\text{im}}$ are calculated analytically using the results of the estimation process of Subsec. 2.1. Since Eq. (1) is underdetermined, one redundant DOF exists, i.e. we can find a vector $\mathbf{N}_{\text{im}}$ in the space of world velocities, which does not produce any movement of the point in the image (see Fig. 2). This vector is directed along the ray from the projection center to the observed 3D point.

Figure 2 also shows the two vectors ($\mathbf{j}^u_{\text{im}}$ and $\mathbf{j}^v_{\text{im}}$), which represent the vectors in the world coordinate system that produce only movements along the u- and v-directions in the image, respectively, and do not produce any motion along the

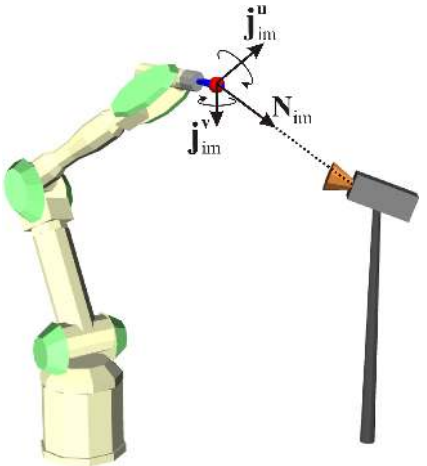


Fig. 2. Vectors in the world coordinate system $\mathbf{j}^u_{\mathrm{im}}$ and $\mathbf{j}^v_{\mathrm{im}}$ correspond to the u and v axes in the image coordinate system and $\mathbf{N}_{\mathrm{im}}$ is the camera null space vector, which is parallel to the camera ray and orthogonal to both $\mathbf{j}_{im}$ vectors.

Fig. 3. Object brought into the center of the camera view field.

camera ray. These two vectors are given by the rows of the Jacobian. We can thus compute $\mathbf{j}^u_{\mathrm{im}}$ and $\mathbf{j}^v_{\mathrm{im}}$ by normalizing the two rows of the Jacobian:

$$\mathbf{j}^u_{\mathrm{im}} = \frac{\begin{bmatrix} j_{11} & j_{12} & j_{13} \end{bmatrix}^T}{\left\| \begin{bmatrix} j_{11} & j_{12} & j_{13} \end{bmatrix} \right\|}, \quad \mathbf{j}^v_{\mathrm{im}} = \frac{\begin{bmatrix} j_{21} & j_{22} & j_{23} \end{bmatrix}^T}{\left\| \begin{bmatrix} j_{21} & j_{22} & j_{23} \end{bmatrix} \right\|}.$$

The vector $\mathbf{N}_{\mathrm{im}}$, which does not produce any movement in the image, is in the null space of the image Jacobian. It can be calculated using the vector product of $\mathbf{j}^u_{\mathrm{im}}$

and $\mathbf{j}_{\text{im}}^{v}$:

$$\mathbf{N}_{\text{im}} = \frac{\mathbf{j}_{\text{im}}^{u} \times \mathbf{j}_{\text{im}}^{v}}{\|\mathbf{j}_{\text{im}}^{u} \times \mathbf{j}_{\text{im}}^{u}\|}. \tag{2}$$

2.2.1. *The control algorithm*

The controller is composed of two parts. The first part deals with the position control of the object and the second part with the size control of the object. Our method belongs to the class of image-based control algorithms.[6] The task of the position controller is to bring the object to the center of the image (see Fig. 3). The size controller should only control the object size and should not affect the position control, and hence it should act in the null space of the image Jacobian. Once the object is in the center of the image, it is only moved directly toward or away from the camera, changing the size, while the position is kept constant. Hence the following controller can be used:

$$\dot{\boldsymbol{x}}_c = \mathbf{J}_{\text{im}}{}^{\#} \dot{\boldsymbol{i}}_c + \mathbf{N}_{\text{im}} \dot{d}_c, \tag{3}$$

where $\dot{\boldsymbol{x}}_c$ is the control velocity in the world coordinate system; $\mathbf{J}_{\text{im}}{}^{\#}$ is the weighted generalized inverse of the image Jacobian, $\mathbf{J}_{\text{im}}{}^{\#} = \mathbf{W}^{-1}\mathbf{J}_{\text{im}}{}^{T}(\mathbf{J}_{\text{im}}\mathbf{W}^{-1}\mathbf{J}_{\text{im}}{}^{T})^{-1}$, with weight $\mathbf{W}$; $\dot{\boldsymbol{i}}_c$ and $\dot{d}_c$ are the control velocity vectors, which correspond to the point positioning and the size setting, respectively. These two vectors can be defined by the following P controllers:

$$\dot{\boldsymbol{i}}_c = K_p^i \begin{bmatrix} u_d - u \\ v_d - v \end{bmatrix}, \quad \dot{d}_c = K_p^d(\text{size}_d - \text{size}), \tag{4}$$

where $[u_d, v_d]^T$ and $[u, v]^T$ are the desired and the actual position of the point (or object) in the image coordinate system, respectively, whereas size_d and size are respectively the desired and the estimated size of the object in the image. K_p^i and K_p^d are the controller gains.

To control a robot, we have to define the control velocities in the joint space. Since the task control vector $\dot{\boldsymbol{x}}_c$ has three DOFs (position of all three coordinates in space) and the robot arms used in the experiments have seven DOFs, the degree of redundancy is four. The following controller can be used:

$$\dot{\boldsymbol{q}}_c = \mathbf{J}_r{}^{\text{pos}\#} \dot{\boldsymbol{x}}_c + \mathbf{N}_r{}^{\text{pos}} \dot{\boldsymbol{q}}_n. \tag{5}$$

Here $\mathbf{J}_r{}^{\text{pos}\#}$ is the generalized inverse of the positional part of the robot Jacobian and $\mathbf{N}_r{}^{\text{pos}}$ is the projection in the null space of $\mathbf{J}_r{}^{\text{pos}}$. Due to the robot's redundancy, we can apply additional subtasks to the robot in the null space of $\mathbf{J}_r{}^{\text{pos}}$. Our choice for the null space motion is to optimize the robot's manipulability.

To show the object from different viewpoints, the robot needs to rotate it about axes which are given by $\mathbf{j}_{\text{im}}^{u}$ and $\mathbf{j}_{\text{im}}^{v}$ in the world coordinate system. It is therefore advantageous to optimize the manipulability for the rotations about both image axes because high manipulability in a certain direction usually corresponds to a

higher ability of motion in the selected direction. Hence we define the null space term as follows:

$$\dot{\boldsymbol{q}}_n = K_m \nabla \sqrt{\det(\mathbf{J}_r{}^{\mathrm{dr}} \mathbf{W}_m \mathbf{J}_r{}^{\mathrm{dr}T})}, \tag{6}$$

where K_m is the controller gain, $\mathbf{W}_m$ is the weight and $\mathbf{J}_r{}^{\mathrm{dr}}$ is the robot Jacobian that corresponds to both depth rotations. We can define $\mathbf{J}_r{}^{\mathrm{dr}}$ by modifying the rotational part of the robot Jacobian ($\mathbf{J}_r{}^{\mathrm{rot}}$) as follows:

$$\mathbf{J}_r{}^{\mathrm{dr}} = \begin{bmatrix} \mathbf{j}_{\mathrm{im}}^u & \mathbf{j}_{\mathrm{im}}^v \end{bmatrix}^{\#} \mathbf{J}_r{}^{\mathrm{rot}}. \tag{7}$$

Here $\mathbf{J}_r{}^{\mathrm{dr}}$ is the robot Jacobian, where the first row corresponds to the rotation about the vector $\mathbf{j}_{\mathrm{im}}^u$ and the second row to the rotation about $\mathbf{j}_{\mathrm{im}}^v$. These are the rotations that correspond to the coordinate axes of the image plane.

In this phase the robot's task is to place the object (center) onto the optical axis of the camera so that its projection has a desired size. In addition to this task, the robot optimizes the manipulability of the system for depth rotations. The goal is to position the robot in an appropriate configuration that enables the best showing of the object from different viewpoints. Note that additional conditions can be applied in the null space, e.g. joint limits and/or self-collision avoidance.

2.3. *Object scanning*

To acquire data about the object from different viewpoints, the robot needs to rotate it in depth with respect to the image system. Rotation in depth is defined as any rotation with the rotation axis not parallel to the camera ray. The largest rotations in depth will therefore be caused by rotations about $\mathbf{j}_{\mathrm{im}}^u$ and $\mathbf{j}_{\mathrm{im}}^v$, since these two vectors are orthogonal to the camera ray (see Fig. 2). Note that the rotation about the vector in the direction of the camera ray ($\mathbf{N}_{\mathrm{im}}$) is not important and can be considered as redundant. Due to the additional two DOFs for rotation, the task now has five DOFs and the degree of redundancy is two. The task space control velocity in this case also includes the angular velocity:

$$\dot{\boldsymbol{q}}_{c_2} = \begin{bmatrix} \mathbf{J}_r{}^{\mathrm{pos}} \\ \mathbf{J}_r{}^{\mathrm{dr}} \end{bmatrix}^{\#} \begin{bmatrix} \dot{\boldsymbol{x}}_c \\ \dot{\boldsymbol{s}}_c \end{bmatrix} + \mathbf{N}_r{}^{\mathrm{pos,dr}} \dot{\boldsymbol{q}}_n, \tag{8}$$

where $\dot{\boldsymbol{s}}_c$ is the vector specifying the rotation in depth about both image axes and $\mathbf{N}_r{}^{\mathrm{pos,dr}}$ is the projection in the null space of $[\mathbf{J}_r{}^{\mathrm{pos}T}, \mathbf{J}_r{}^{\mathrm{dr}T}]^T$. In this way we ensure that the arm retains high manipulability while the robot rotates the object.

3. Snapshot Acquisition

Now we turn to the problem of how to acquire snapshots of objects manipulated by the robot. It is evident from Eq. (4) that to properly manipulate the object, the robot needs to be able to determine its position and size in each image. We shall see that this same information can be used to extract the object's views.

Once the robot gets hold of an object, it can take advantage of the fact that it knows how the object moves. This allows the robot to first learn the background model by removing the object from the view field of its eyes. After the background model is learned, the manipulation procedure of Sec. 2 can be started. Although background models are subject to frequent changes, this is not a problem here because they are needed only for short periods of time and can be learned anew if necessary. This allows us to avoid dealing with varying backgrounds, e.g. in Ref. 5. Moreover, since the robot controls the data acquisition process, there exists additional knowledge about the environment, which can be used to improve the quality of the acquired snapshots. To discern the manipulated object from the rest of the image, we model the following image processes:

- the unknown object (denoted by process Θ_{o}),
- the background (Θ_{b}),
- the hand (Θ_{h}),
- the outlier process (Θ_{t}), modeling any unexpected event in the scene.

Stationary background can be modeled using various features, such as color distribution, disparity, and motion. Currently we use the first two. The color distribution at each pixel in the stationary background is modeled by a Gaussian process, $\Theta_{\mathrm{b1}} = \{\overline{\boldsymbol{I}}_{\boldsymbol{u}}, \overline{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\}$, which is characterized by mean $\overline{\boldsymbol{I}}_{\boldsymbol{u}}$ and covariance matrix $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$ at each pixel $\boldsymbol{u}$, with the associated probability distribution

$$\mathrm{p}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b1}}) = \frac{1}{2\pi\sqrt{\det(\overline{\boldsymbol{\Sigma}}_{\boldsymbol{u}})}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{I}_{\boldsymbol{u}} - \overline{\boldsymbol{I}}_{\boldsymbol{u}})^T \overline{\boldsymbol{\Sigma}}_{\boldsymbol{u}}^{-1} (\boldsymbol{I}_{\boldsymbol{u}} - \overline{\boldsymbol{I}}_{\boldsymbol{u}})\right). \tag{9}$$

To obtain a certain degree of robustness against the brightness changes, we characterize the color intensities by hue and saturation, which makes color space two-dimensional. The means and the covariances are learned by gathering statistics of the background images $\boldsymbol{I}$ for about 10 s, just before the robot brings the object into the camera view. It is essential to smooth the images significantly before applying this calculation (see Fig. 4), otherwise even small disturbances can cause failure.

Disparity as shown in Fig. 5 is another strong cue and has the advantageous property of being robust against changes in lighting conditions. Let D be the estimated disparity image. We model the disparity distribution as a Gaussian process, $\Theta_{\mathrm{b2}} = \{\overline{D}_{\boldsymbol{u}}, \sigma_{\mathrm{D}}^2\}_{\boldsymbol{u}}$. Similarly to color, we estimate the means $\overline{D}_{\boldsymbol{u}}$ at each pixel by collecting disparity images of a stationary background for 10 s. The standard deviation σ_{D} is not estimated but is set to a constant value. This results in the following estimate for the background distribution:

$$\mathrm{p}(\boldsymbol{I}_{\boldsymbol{u}}, D_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b}}) = \mathrm{p}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b1}})\mathrm{p}(D_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b2}}). \tag{10}$$

Even though the hand position in the image could be calculated using proprioceptive information, this information is not sufficient because we cannot know in advance which part of the hand is visible and which part is covered by the manipulated object. We thus need to model the appearance of the hand in the image. For

Fig. 4. Example for the extraction of object appearance. From left to right we have images showing one of the images used for background learning, the smoothed version of this image that we use for the collection of background statistics, the estimated extent of the object in the image while being manipulated by the robot, and the binary image containing the largest connected component of object pixels after thresholding probabilities $\mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}})$ and applying the morphological operation close.

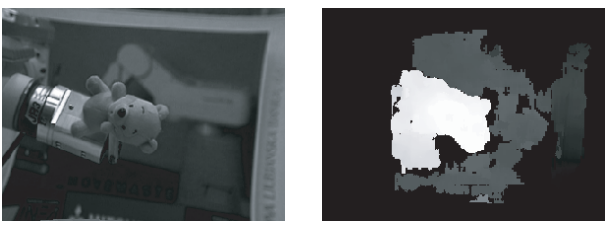


Fig. 5. One of the collected images after rectification and the corresponding disparity map.

the modeling of the hand appearance, we experimented with standard approaches from the object tracking theory, such as color histograms[1] and Gaussian (mixture) models.[10] Unlike in tracking, we are not really interested in computing the hand position, but only in estimating the probability that a particular pixel belongs to the hand. Both color histograms and Gaussian mixture models offer this ability. Gaussian mixture models are defined as follows:

$$\mathrm{p}(\boldsymbol{I_u}|\Theta_{\mathrm{h}}) = \sum_{k=1}^{K} \frac{\omega_k}{2\pi\sqrt{\det(\boldsymbol{\Sigma}_k)}} \exp\left(-\frac{1}{2}(\boldsymbol{I_u} - \overline{\boldsymbol{I}_k})^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{I_u} - \overline{\boldsymbol{I}_k})\right). \tag{11}$$

While motion cues could certainly help to extract the object from the background, such cues alone are not sufficient for the extraction of the object appearance. When the robot holds the object, the object motion is the same as the motion of the robot hand. We thus cannot distinguish between the object and the hand using motion cues only.

Since no prior knowledge about the object is available, we obviously cannot model its appearance, which is what we want to learn. The open-loop trajectory that we use to manipulate the object is, however, well defined and we know approximately where the object is in the image. We can thus model the probability that an image pixel falls within the extent of the object by using the mean value $\overline{\boldsymbol{u}}$ and the covariance $\overline{\boldsymbol{\Sigma}}$ of pixels belonging to the object in the previous step. This results

in the following distribution:

$$\mathrm{p}(\boldsymbol{u}|\Theta_\mathrm{o}) = \frac{1}{2\pi\sqrt{\det(\overline{\boldsymbol{\Sigma}})}} \exp\left(-\frac{1}{2}(\boldsymbol{u}-\overline{\boldsymbol{u}})^T\overline{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{u}-\overline{\boldsymbol{u}})\right). \tag{12}$$

Since the robot attempts to bring the object to the center of the image and to keep it there, the object's position is normally close to the image center and we can initialize the appearance extraction by assuming that the object is centered in the image with an initially large extent.

Having no prior information about the appearance of the arm and other unexpected objects that might appear in the scene, we model such events in the image by an outlier process, which is assigned a small, constant probability $\mathrm{P}(\Theta_\mathrm{t})$ regardless of the position of the pixel in the image or the color intensity value at this pixel. The interaction between this process and the object process Θ_o occurs in such a way that an area with a texture different from the background and the hand will be classified as an object of interest if it is close to the expected object position and outlier otherwise [see Eq. (16)].

As for the arm, the part of the image containing it can be excluded from calculations using proprioceptive information. On a dynamic humanoid robot like *i*–1 of Fig. 1, proprioceptive information provides only a rough estimate for the location of the arm in the image. It is nevertheless sufficient to exclude from the calculations most of the image containing the arm. Our experiments showed that, combined with the outlier process, this is sufficient to filter out the arm when estimating the extent of the object of interest in the image.

Assuming that every pixel in the image stems from one of the mutually independent processes $\boldsymbol{\Theta} = \{\Theta_\mathrm{b}, \Theta_\mathrm{h}, \Theta_\mathrm{o}, \Theta_\mathrm{t}\}$ (closed-world assumption), we can write the probability that color $\boldsymbol{I_u}$ was observed at location $\boldsymbol{u}$ using the total probability law:

$$\mathrm{P}(\boldsymbol{I_u}, \boldsymbol{u}|\boldsymbol{\Theta}) = \omega_\mathrm{b}\mathrm{P}(\boldsymbol{I_u}, D_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_\mathrm{b}) + \omega_\mathrm{h}\mathrm{P}(\boldsymbol{I_u}|\Theta_\mathrm{h}) + \omega_\mathrm{o}\mathrm{P}(\boldsymbol{u}|\Theta_\mathrm{o}) + \omega_\mathrm{t}\mathrm{P}(\Theta_\mathrm{t}), \tag{13}$$

where ω_x are the prior (mixture) probabilities of observing the processes Θ_x and $\omega_\mathrm{b} + \omega_\mathrm{h} + \omega_\mathrm{o} + \omega_\mathrm{t} = 1$.

We need to estimate the current position of the unknown object and its extent, which enables us to extract an appearance image for learning. This can be achieved by maximizing the probability of observing image $\boldsymbol{I}$ given the processes $\boldsymbol{\Theta} = \{\Theta_\mathrm{b}, \Theta_\mathrm{h}, \Theta_\mathrm{o}, \Theta_\mathrm{t}\}$. Neglecting the correlation of assigning neighboring pixels to processes, we can evaluate the overall probability of observing image $\boldsymbol{I}$ as follows:

$$\mathrm{P}(\boldsymbol{I}) = \mathrm{P}(\boldsymbol{I}|\boldsymbol{\Theta}) = \prod_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{I_u}, \boldsymbol{u}|\boldsymbol{\Theta}). \tag{14}$$

Since the background and the color distribution of the hand are assumed to be stationary, we can maximize (14) with respect to the position $\overline{\boldsymbol{u}}$ of the object, the covariance $\overline{\boldsymbol{\Sigma}}$ of pixels belonging to the object, and the mixture probabilities ω_b, ω_h, ω_o, and ω_t. Instead of maximizing (14), it is easier to minimize the negative log

likelihood

$$\mathrm{L}(\boldsymbol{\Theta}, \boldsymbol{\omega}) = -\log(\mathrm{P}(\boldsymbol{I}|\boldsymbol{\Theta})) = -\sum_{\boldsymbol{u}} \log\left(\mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\boldsymbol{\Theta})\right), \tag{15}$$

where $\boldsymbol{\omega} = (\omega_{\mathrm{b}}, \omega_{\mathrm{h}}, \omega_{\mathrm{o}}, \omega_{\mathrm{t}})$. Using the theory of Lagrange multipliers, it is possible to show that the above log likelihood can be minimized with an EM algorithm. Writing

$$\mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_x) = \frac{\omega_x \mathrm{p}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_x)}{\sum_{y\in\{\mathrm{o,h,b,t}\}} \omega_y \mathrm{p}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_y)}, \tag{16}$$

where $x = \mathrm{o},\ \mathrm{h},\ \mathrm{b},\ \mathrm{t}$, the EM algorithm consists of the expectation step, in which the pixel probabilities (16) are estimated, and the maximization step, in which the probabilities $\mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b}}) = \mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}})$ are used to estimate the mean and the covariance of the object pixels:

$$\overline{\boldsymbol{u}} = \frac{1}{\sum_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}})} \sum_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}})\boldsymbol{u}, \tag{17}$$

$$\overline{\boldsymbol{\Sigma}} = \frac{1}{\sum_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}})} \sum_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{u}|\Theta_{\mathrm{b}}) (\boldsymbol{u} - \overline{\boldsymbol{u}}) (\boldsymbol{u} - \overline{\boldsymbol{u}})^T. \tag{18}$$

Note that the probabilities $\mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_{\mathrm{b}})$ and $\mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}|\Theta_{\mathrm{h}})$ remain constant throughout the EM process and thus need to be estimated only once for each image. This helped us to implement the extraction of the object appearance at the video rate, i.e. 30 Hz. The mixture probabilities can either be assumed to be constant or estimated as part of the EM process:

$$\omega_x = \frac{1}{n} \sum_{\boldsymbol{u}} \mathrm{P}(\boldsymbol{I}_{\boldsymbol{u}}, \boldsymbol{u}|\Theta_x), \tag{19}$$

where n is the number of pixels and $x = \mathrm{o},\ \mathrm{h},\ \mathrm{b},\ \mathrm{t}$.

The described EM process realizes the estimation of the object position $\overline{\boldsymbol{u}}$ and of the approximate object size and orientation in the image, both encoded by the covariance matrix $\overline{\boldsymbol{\Sigma}}$. Different stages in the algorithm are illustrated in Fig. 4.

4. Learning Object Representations

Snapshots acquired using the methods of Secs. 2 and 3 can be used to learn a classifier for object recognition. The enclosing ellipse estimated by the proposed algorithms can be used to warp the snapshots onto a window of constant size. This ensures invariance against scaling and planar rotations, and also provides images of standard size, which can be compared to each other. Figure 6 shows the warped images of four objects used in our experiments.

To ensure maximum classification performance, the acquired snapshots need to be preprocessed. Most modern view-based approaches characterize the views by ensembles of local features. We use complex Gabor kernels to identify local structure

Fig. 6. Images of four objects used in the experiments after warping. Such images are used as input to Gabor jet calculations and SVM training.

in the images. The acquired snapshots are first transformed to grayscale and then filtered with Gabor kernels, which are defined by

$$\mathbf{\Phi}_{\mu,\nu}(\boldsymbol{x}) = \frac{\|\boldsymbol{k}_{\mu,\nu}\|^2}{\sigma^2} \cdot \exp\left(-\frac{\|\boldsymbol{k}_{\mu,\nu}\|^2\|\boldsymbol{x}\|^2}{2\sigma^2}\right)\left(\exp\left(i\boldsymbol{k}_{\mu,\nu}^T\boldsymbol{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right), \qquad (20)$$

where $\boldsymbol{k}_{\mu,\nu} = k_\nu[\cos(\phi_\mu),\ \sin(\phi_\mu)]^T$. A Gabor jet at pixel $\boldsymbol{x}$ is defined as a set of complex coefficients $\{J_j^{\boldsymbol{x}}\}$ obtained by convolving the image with a number of Gabor kernels at this pixel. Gabor kernels are selected so that they sample a number of different wavelengths k_ν and orientations ϕ_μ. Wiskott *et al.*[18] proposed using $k_\nu = 2^{-\frac{\nu+2}{2}},\ \nu = 0,\ldots,4$, and $\phi_\mu = \mu\frac{\pi}{8},\ \mu = 0,\ldots,7$, but this depends on both the size of incoming images and the image structure.

In our system, feature vectors are built by sampling Gabor jets on a regular grid of pixels $\boldsymbol{X}_G$. At each grid point we calculate the Gabor jet and add it to the feature vector. Naturally, the grid points need to be parsed in the same order in every image. The grid size used in our experiments was 6×6, the warped image size was 160×120 with pixels outside the enclosing ellipse excluded, and the dimension of each Gabor jet was 40, which resulted in feature vectors of dimension 16,080. To compute a classifier we employed nonlinear multiclass support vector machines proposed in Ref. 2. It was very important to normalize the jets to achieve robustness against changes in brightness conditions.

5. Experimental Results

The algorithms presented in this paper were tested in several experiments with two different robots: humanoid robot *i*–1 and a Mitsubishi PA-10 robot arm with external cameras.

5.1. *Validation of the control processes*

The task of the object manipulation process is to achieve the widest range of the directions of view. To show the efficiency of the proposed method, we compared three different approaches. In the first approach, we controlled the robot without

manipulability optimization and without exploiting the redundant DOFs about the axis along the camera ray $\mathbf{N}_{\text{im}}$. Hence, the robot does not configure in the optimal configuration for performing the observation procedure. Additionally, the orientation of the object about the camera ray axis is fixed, which significantly influences the range of motion. In the second approach the manipulability was optimized, so the robot configures in the appropriate configuration for performing depth rotations, but we still did not exploit the redundancy of the rotation about the camera ray axis. In the third approach, we optimized the manipulability and exploited the redundancy about the camera ray axis.

To objectively show the range of motion for each of the compared methods, we represented the space of all rotations by azimuth, elevation, and rotation. This representation is shown in Fig. 8, where *azimuth* and *elevation* correspond to the point on the sphere, i.e. the direction of view, while *rotation* corresponds to the rotation about the camera ray axis, which is insignificant because we are not interested in orientation of the object about the camera ray (this rotation does not change the part of the object visible to the camera system). We are interested in the rotational motion with respect to the initial orientation of the object. In the initial configuration the azimuth, elevation, and rotation angles are assumed to be zero. These three angles can be converted in the rotation matrix as follows:

$$R = \text{rot}(z, \text{rotation})\text{rot}(y, \text{azimuth})\text{rot}(x, \text{elevation}).$$

We implemented and compared the three described methods on a Mitsubishi PA-10 with an external camera system. The robot attempted to rotate the object about both depth rotations with angular velocity of 0.2 and 0.1 rd/s, respectively. It moved the object about the first depth rotation until it came to the end of its workspace (i.e. until the singularity or joint limit occurs). Then the robot changed

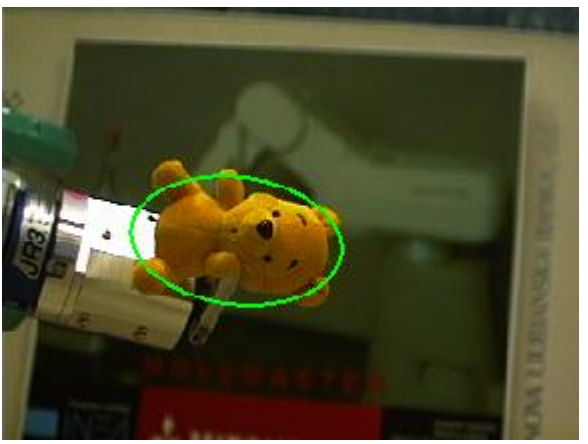


Fig. 7. The robot holding an object to be learned. The object's position and extent are estimated using the knowledge about the robot's motion and short term background models.

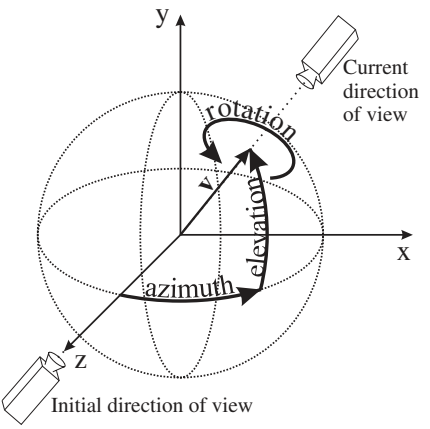


Fig. 8. The coordinate system used for validating the direction of view.

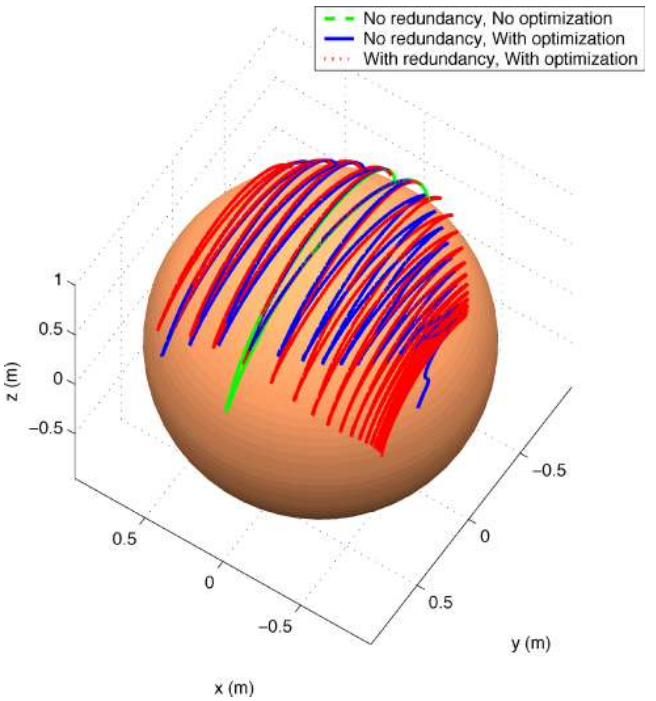
the direction of rotation. With this procedure we acquired the whole range of motion of the robot within its workspace. The configuration of the robot and the camera during the rotation process is shown in Fig. 7.

Figure 9 depicts the direction of view while rotating the object as shown in Fig. 10. Figure 9(a) depicts the direction of view on the sphere, and Fig. 9(b) the direction of view represented by the *azimuth* and *elevation* angles. It is clear from these figures that the largest range of motion is achieved when we make use of the redundancy of the rotation about the camera ray axis in addition to a suitable configuration control using manipulability optimization. Without manipulability optimization, the robot falls into a singularity very quickly. With manipulability optimization but without exploiting the redundancy, the robot successfully avoids the singularities, but its range of motion is smaller.

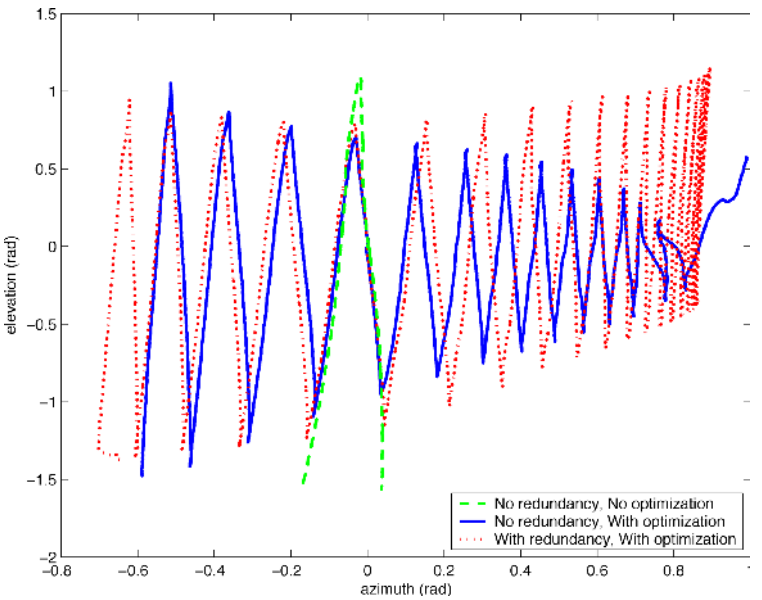
5.2. *Validation of the learned classifier*

To test the effectiveness of the object manipulation process described in Sec. 2 together with the Bayesian technique of Sec. 3, we used them to extract views for object learning and recognition. The proposed techniques enabled the robot to collect the views shown in Fig. 6 without any prior knowledge about the objects. The procedure for discerning the object from the rest of the scene has proven to be reliable as long as the assumptions made by the Bayesian approach were fulfilled.

To prove that the proposed approach can indeed be used for learning object representations, we compared it to the classification results achieved when known color textures were used to discern the object from the rest of the image and the objects were manipulated by a human, not a robot.[16] To train the SVM, we collected 104 views of 14 different objects. The appearance images of 4 of them were extracted using the proposed approach, while the images of the rest were collected by applying models of color texture for segmentation. To train a fully rotationally



(a) Shown on the sphere



(b) Shown in the azimuth/elevation space

Fig. 9. Direction of view.

Fig. 10. Object manipulation example.

Table 1. Classification results.

	Correct	Errors	Recognition Rate (%)
Full library	7307	421	94.6
Objects detected by color	4897	303	94.2
Objects detected by manipulation	2410	118	95.3

and scale-invariant classifier for a library of 14 objects, we acquired 1456-feature vectors of dimension 16,080. We used the implementation of nonlinear multiclass SVMs described in Ref. 7 to train the classifier.

For testing we collected another 7728 snapshots of the objects from the library. The results in Table 1 prove that the views collected by the proposed approach are just as usable as the views that we collected using prior color texture models. The recognition results with the proposed approach were even a bit better, although this was caused by a relatively bad classification rate for one of the objects for which we used color texture segmentation to extract the views. Excluding this object, the recognition rates were almost identical.

6. Summary

The main result of this paper is the procedure for acquiring object snapshots and the subsequent learning of complete object representations for recognition by a humanoid robot without any prior knowledge about the objects and without manual tinkering with the images. Our experiments showed that the generated models are fully scale and rotationally invariant in 3D and that we achieve comparable recognition rates on the proposed system as on the earlier system that used prior knowledge about the objects' color textures to discern their images from the rest of the scene.

Some issues still remain to be considered in the future. One of them is to replace the statistical collection of views with a more sophisticated planning process that would allow the reduction of the number of views by collecting only the most informative snapshots. Another interesting issue is the use of proprioceptive information to organize the training views by orientation. Such information can be used to estimate the orientation of an object after recognition. Also, it is clear that to acquire a complete viewpoint-independent model, the robot needs to regrasp the object and observe it from different starting orientations. We are currently working on the implementation of a systematic regrasping behavior on a humanoid robot.

The main future goal of our work is the integration of all sensorimotor processes that are needed to achieve cognitive behavior of a robot in the case of object learning. At the start the robot should find and pick up an object using visual attention. Next, the robot should try to grasp the object. Using sensorimotor processes described in this paper, the robot will finally be able to acquire object representations without any external help.

Acknowledgment

The work described in this paper was partially conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657), funded by the European Commission.

References

1. D. Comaniciu, V. Ramesh and P. Meer, Kernel-based object tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5) (2003) 564–577.
2. K. Crammer and Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, *J. Mach. Learn. Res.* **2** (2001) 265–292.
3. P. Fitzpatrick, First contact: An active vision approach to segmentation, in *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, Nevada (2003), pp. 2161–2166.
4. P. Fitzpatrick and G. Metta, Grounding vision through experimental manipulation, *Philos. Trans. Roy. Soc.: Math. Phys. Eng. Sci.* **361**(1811) (2003) 2165–2185.
5. E. Hayman and J.-O. Eklundh, Statistical background subtraction for a mobile observer, in *Proc. IEEE. Int. Conf. Computer Vision*, Nice, France (2003), pp. 67–74.
6. S. Hutchinson, G. D. Hager and P. I. Corke, A tutorial on visual servo control, *IEEE Trans. Robot. Autom.* **12**(5) (1996) 651–670.
7. T. Joachims, Making large-scale support vector machine learning practical, in B. Schölkopf, C. J. C. Burges and A. J. Smola (eds.), *Advances in Kernel Methods — Support Vector Learning* (MIT Press, Cambridge, MA, 1999).
8. D. G. Lowe, Local feature view clustering for 3D object recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Kauai, Hawaii (2001), pp. 682–688.
9. E. Marchand, F. Chaumette and A. Rizzo, Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Osaka, Japan (1996), pp. 1083–1090.
10. S. J. McKenna, Y. Raja and S. Gong, Tracking colour objects using adaptive mixture models, *Image Vision Comput.* **17** (1999) 225–231.
11. B. Nelson and P. Khosla, Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance, *Int. J. Robot. Res.* **14**(3) (1995) 255–269.
12. T. Poggio and S. Edelman, A network that learns to recognize three-dimensional objects, *Nature* **343** (1990) 263–266.
13. B. Schiele and J. L. Crowley, Recognition without correpsondence using multidimensional receptive field histograms, *Int. J. Comput. Vision* **36**(1) (2000) 31–52.
14. Z. Tu, X. Chen, A. L. Yuille and S.-C. Zhu, Image parsing: Unifying segmentation, detection, and recognition, *Int. J. Comput. Vision* **63**(2) (2005) 113–140.
15. M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cogn. Neurosci.* **3**(1) (1991) 71–86.

16. A. Ude, C. G. Atkeson and G. Cheng, Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, Nevada (2003), pp. 2173–2178.
17. C. Wallraven, A. Schwaninger and H. H. Bülthoff, Learning from humans: Computational modeling of face recognition, *Network: Comput. Neural Syst.* **16**(4) (2005) 401–418.
18. L. Wiskott, J.-M. Fellous, N. Krüger and C. von der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7) (1997) 775–779.
19. T. Yoshikawa, Basic optimization methods of redundant manipulators, *Lab. Robot. Autom.* **8**(1) (1996) 49–60.

Aleš Ude studied applied mathematics at the University of Ljubljana, Slovenia and received his doctoral degree from the Faculty of Informatics, University of Karlsruhe, Germany. He received the STA fellowship (by Science and Technology Agency of Japan) for postdoctoral studies (2 years) in ERATO Kawato Dynamic Brain Project, Japan. He has been a visiting researcher at ATR Computational Neuroscience Laboratories, Kyoto, Japan for a number of years and is still associated with this group. Currently he is a senior researcher at the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. His research focuses on humanoid robot vision, visual perception of human activity, imitation and action learning, and humanoid cognition.

Damir Omrčen received his Ph.D. degree from the University of Ljubljana, Faculty of Electrical Engineering in 2005. Currently he is a research assistant at the Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Slovenia. His research interests include machine learning and cognitive systems in robotics. Here he focuses on realization and implementation of primitive movements on humanoid robots, which should be learned by real world exploration. His other important research field is biomechanics, where he focusses on the realisation of humanoid robot jumping and other dynamic movements. His goal is to imitate human structure and human behavior to achieve higher efficiency of a dynamic movement.

Gordon Cheng received his Bachelor's and Master's degrees in computer science from the University of Wollongong, Wollongong, New South Wales, Australia, and his Ph.D. degree in systems engineering from the Department of Systems Engineering, Australian National University, Acton, Australian Capital Territory, Australia.

His current research interests include humanoid robotics, cognitive systems, biomimetic of human vision, computational neuroscience of vision, action understanding, human-robot interaction, active vision, mobile robot navigation, and object-oriented software construction. He is on the editorial board of the International Journal of Humanoid Robotics. Dr. Cheng is a Senior Member of the IEEE Robotics and Automation Society and the IEEE Computer Society.