Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Support Vector Machines and Gabor Kernels for Object Recognition on a Humanoid with Active Foveated Vision

Aleš Ude[1,2] Chris Gaskett[3] Gordon Cheng[1]

[1]ATR Computational Neuroscience Lab.
Dept. of Hum. Robot. and Comp. Neurosc.
2-2-2 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288, Japan

[2]Jožef Stefan Institute, Dept.
of Automatics, Biocybernetics
and Robotics, Jamova 39
1000 Ljubljana, Slovenia

[3]James Cook University
School of Information
Technology, PO Box 6811
Cairns, QLD 4870, Australia

Abstract— **Object recognition requires a robot to perform a number of nontrivial tasks such as finding objects of interest, directing its eyes towards the objects, pursuing them, and identifying the objects once they appear in the robot's central vision. We have recently developed a recognition system on a humanoid robot which makes use of foveated vision to accomplish these tasks [1]. In this paper we present several substantial improvements to this system. We present a biologically motivated object representation scheme based on Gabor kernel functions and show how to employ support vector machines to identify known objects in foveal images based on this representation. A mechanism for visual search is integrated into the system to find objects of interest in peripheral images. The framework also includes a control scheme for eye movements, which are directed using the results of attentive processing in peripheral images.**

I. Introduction

A robot vision system is humanoid if it firstly possesses an oculomotor system similar to human eyes, and secondly if it is capable of simultaneously acquiring and processing images of varying resolution taken from two slightly different viewing directions. Approaches proposed to mimic the foveated structure of biological vision systems include the use of two cameras per eye [2]–[5], i. e. a narrow-angle foveal camera and a wide-angle camera for peripheral vision; lenses with space-variant resolution [6], i. e. a very high definition area in the fovea and a coarse resolution in the periphery; and space-variant log-polar sensors [7]. Our work follows the first approach (see Fig. 1) and we have recently presented a system that can make use of foveated vision for object recognition [1].

We utilize foveation as follows: our humanoid robot DB relies on peripheral vision to search for interesting areas in visual scenes. The attention system reports about salient regions and triggers saccadic eye movements. After the saccade the robot starts pursuing the area of interest, thus keeping it visible in the high-resolution foveal region of the eyes, assisted by peripheral vision if foveal tracking fails. Finally, high-resolution foveal vision provides the humanoid with a more detailed description of the detected events and objects, upon which the robot can take further actions.

Our initial system employed LoG (Laplacian of the Gaussian) filters at a single, manually selected scale and principal component analysis to represent objects. The nearest neighbor approach was used to identify the modeled objects in visual scenes. This system was used successfully in interactive experiments with DB. To improve its performance, we explored some alternative object representation schemes and classification algorithms. We experimented with representations based on Gabor jets [21], which are constructed by convolving an image with a number of Gabor filters at different scales, and Gabor wavelet networks [8], [9], which can be effectively tuned to represent local object features. Gabor kernels are prominent in machine vision because they achieve the best possible joint resolution in 2-D visual space and 2-D Fourier domain [10]. Gabor filters are often used for feature detection. It is also interesting for humanoid robot vision that receptive field profiles of simple cells in the primary visual cortex of primates can be interpreted as Gabor wavelet functions [10]. Gabor wavelet networks have been employed before in the context of head tracking and face recognition [11], [12]. In this paper we show how support vector machines (SVMs) can be used to classify objects represented by Gabor jets or Gabor wavelet networks. We also present some more details of a complete system starting from visual search over the generation of eye movements to object recognition.

II. Visual Search

To classify an object in the visual scene, the robot must first identify the object's location in the image. Visual search and the role of attention in search has been much discussed in recent literature [13]. Treisman's feature integration theory is one of the most thoroughly studied approaches and resulted – somewhat modified – in several technical implementations, e. g. [14], including some implementations on humanoid robots [2], [5]. These implementations are mainly concerned with bottom-up, data-driven processing directed towards the generation of

0-7803-8463-6/04/$20.00 ©2004 IEEE

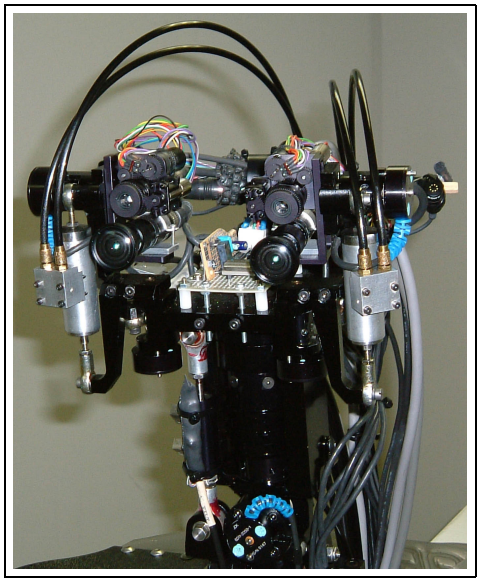

Fig. 1. DB's head. Foveal cameras are above peripheral cameras.

saliency maps. However, many theories of visual search, e. g. guided search, suggest that there are several ways for preattentive processing to guide the deployment of attention [13]. Besides the bottom-up guidance towards salient regions, there is also a top-down guidance based on the needs of the searcher. Here we briefly present our implementation of the top-down search process that bypasses the saliency maps.

A. Top-down Guidance

The theory that human visual search always relies on accumulating information about objects over time has been recently disputed in [15]. The authors showed, in a number of behavioral experiments, that search efficiency is not impaired if the scene is continuously shuffled while the observer is trying to search through it. They concluded that during a visual search episode, no memory is devoted to rejected distractors. Although they acknowledge the existence of inhibition of return (IOR), they argue that IOR has only a very short duration (last 4-6 attended items).

These findings suggest that it does not make sense to implement complex search schemes when a humanoid robot looks for a particular feature or object in an unknown, dynamic and cluttered environment. It is a daunting task to keep and update all the attended positions in memory when the robot moves and the scene changes. It seems therefore logical to implement the top-down search at least at real-time level in a purely random fashion. Such an approach does not exclude the existence of strategically planned searches such as for example limiting the search to a particular area in the image, but we assume that such searches are not planned in real-time and are based on higher-level knowledge of the scene.

We assign to each object in our object library a number of signal detectors describing object features such as for example color. It has been argued recently that many aspects of visual search can be explained by the signal detection theory [16]. The signal detectors do not need to be tuned to one object only, e. g. objects having the same color are associated with the same detector. Signal detectors can describe the properties of more than one feature and can thus deal with compounded features, but we have not implemented such detectors yet. We assume that 2-D shapes of objects from the library can be approximated by the second order statistics of pixels contained in their projected images. Since we do not have any information about the location and identity of the objects (apart from that we are looking for objects from the library), we start by randomly selecting the object size, shape and location in the image. If the number of feature detectors is not too large, all of them are evaluated at this location, otherwise we randomly select some of them for evaluation so that real-time processing is still possible. The group of objects associated with the detector is assumed detected if the signal detector exceeds a threshold that is learned in the training phase. The shape parameters are varied in a controlled way so that 2-D sizes of the generated object hypotheses remain within prespecified limits. This implements search at multiple resolutions. To ensure that the processing time is constant, which is necessary to guarantee real-time operation of the system, we warp the randomly generated object location onto a window of fixed size as described in [17]. We also implemented a short term inhibition of return by rejecting all newly generated locations that are located within any of the enclosing ellipses of the last 5 randomly selected object hypotheses. A new test location is generated in this case.

III. Generation of Eye Movements

The main task of the control system is to place a salient region in the field of view of both foveal cameras so that further analysis and eventually object recognition can be carried out. Although the focus of the task is to bring the object into the center of the fovea, the control system uses the view of the object from peripheral cameras as the basis for control. Motion based on information acquired from peripheral images is more reliable because objects can easily be lost from the view of the foveal cameras. Since the foveal cameras are rigidly connected to the peripheral cameras and placed above them with roughly aligned optical axes, the object can be placed in the foveal images by bringing it into a position slightly displaced from the center of peripheral images.

The robot's primary mechanism for maintaining the view of the object of interest is the eye movement: the control system continuously alters the pan and tilt of each eye to keep the object near the center of the corresponding view. This process of continuously updating the position of part of a robot based on visual information is known as visual servo control [18].

Independent motion of the eyes is acceptable when the object is being tracked properly in both peripheral views, but looks rather unnatural when one eye loses its view of the object while the other eye continues to roam. Our

solution is to introduce a gentle cross-coupling between a camera's view and the control of the opposite eye. Thus, when a camera's view of the target is lost, its corresponding eye continues to move, fairly slowly, under the influence of the opposite camera's view. As well as appearing natural, such eye movements improve the likelihood of re-finding the object.

Our robot DB has altogether 30 degrees of freedom and other joints can support the eyes to keep the object in the center of the fovea. We implemented supportive head and torso movements and thus use 10 degrees of freedom (4 on the eyes + 3 on the head + 3 on the torso) to maintain the view of the object. We consider that the task of the robot's head is to assist the eyes by increasing the viewable area and avoiding unnatural poses. To aid in coordinating the joints, we assign a relaxation position to each joint and vision blob [1]. The relaxation position for the blobs is near the center of the view, and the eyes' task is to bring the blobs to that position. The relaxation position for the 4 eye joints is to face forward, and the head's task is to bring the eyes to that position. Further, the 3 head joints have a relaxation position, and the torso's task is to bring the head to that position. For example, if the object of interest is up and to the left, the eyes would tilt up and pan left, causing the head would tilt up and turn left, and the torso to lean back and turn.

The complete control system is implemented as a network of PD controllers expressing the assistive relationships. The PD controllers are based on simple mappings described below rather than on a full kinematic model. The simple mappings are sufficient because the system is closed-loop. Cartesian 3-D information is not used because it is difficult to maintain the camera calibration under seamless motion.

We define the *desired change* for self-relaxation, D, for each joint,

$$D_{joint} = \left(\theta^*_{joint} - \theta_{joint}\right) - K_d \dot{\theta}_{joint}, \tag{1}$$

where K_d the derivative gain for joints; θ is the current joint angle; $\dot{\theta}$ is the current joint angular velocity, and the asterisk indicates the relaxation position. The derivative components help to compensate for the speed of the blobs and assisted joints.

The desired change for a vision blob is:

$$D_{blob} = (x^*_{blob} - x_{blob}) - K_{dv} \dot{x}_{blob}, \tag{2}$$

where K_{dv} is the derivative gain for vision blobs; and x is position in pixels.

The purpose of the *left eye pan* ($\mathsf{L}EP$) joint is to move the target into the center of the left camera's field of view:

$$\begin{aligned}\widehat{\dot{\theta}}_{\mathsf{L}EP} = K_p \times \Big[& K_{\text{relaxation}} D_{\mathsf{L}EP} \\ & - K_{\text{target}\rightarrow EP} K_v C_{\mathsf{LX}\text{target}} D_{\mathsf{LX}\text{target}} \\ & + K_{\text{cross-target}\rightarrow EP} K_v C_{\mathsf{RX}\text{target}} D_{\mathsf{RX}\text{target}} \Big], \end{aligned} \tag{3}$$

[1] Vision blobs give the hypothesized 2-D object locations in the image

where $\widehat{\dot{\theta}}_{\mathsf{L}EP}$ is the new target velocity for the joint; L and R represent left and right; X represents the x pixels axis; K_p is the proportional gain; K_v is the proportional gain for vision blobs; C_{blob} is the tracking confidence for that blob; and the gain $K_{\text{cross-target}\rightarrow EP} < K_{\text{target}\rightarrow EP}$.

The purpose of the *left eye tilt* ($\mathsf{L}ET$) joint is to move the target into the center of the left camera's field of view:

$$\begin{aligned}\widehat{\dot{\theta}}_{\mathsf{L}ET} = K_p \times \Big[& K_{\text{relaxation}} D_{\mathsf{L}ET} \\ & - K_{\text{target}\rightarrow ET} K_v C_{\mathsf{LY}\text{target}} D_{\mathsf{LY}\text{target}} \\ & - K_{\text{cross-target}\rightarrow ET} K_v C_{\mathsf{RY}\text{target}} D_{\mathsf{RY}\text{target}} \Big], \end{aligned} \tag{4}$$

The equations for the right eye pan and tilt joints are the same as for the left, except that L becomes R and vice versa.

Head nod joint (HN) assists the eye tilt joints:

$$\begin{aligned}\widehat{\dot{\theta}}_{HN} = K_p \times \Big[& K_{\text{relaxation}} D_{HN} \\ & - K_{ET\rightarrow HN} \left(D_{\mathsf{L}ET} + D_{\mathsf{R}ET}\right) \Big]. \end{aligned} \tag{5}$$

The *head tilt joint* (HT), which tilts the head from side to side, moves to assist the pan (EP) and equalize the tilt (ET) of the eyes:

$$\begin{aligned}\widehat{\dot{\theta}}_{HT} = K_p \times \Big[& K_{\text{relaxation}} D_{HT} \\ & - K_{EP\rightarrow HT} \left(D_{\mathsf{L}EP} - D_{\mathsf{R}EP}\right) \\ & - K_{ET\rightarrow HT} \left(D_{\mathsf{L}ET} - D_{\mathsf{R}ET}\right) \Big]. \end{aligned} \tag{6}$$

The *head rotate joint* (HR) assists the eye pan joints:

$$\begin{aligned}\widehat{\dot{\theta}}_{HR} = K_p \times \Big[& K_{\text{relaxation}} D_{HR} \\ & - K_{EP\rightarrow HR} \left(D_{\mathsf{L}EP} - D_{\mathsf{R}EP}\right) \Big]. \end{aligned} \tag{7}$$

The *torso rotate joint* (TR) assists the head rotation joint:

$$\widehat{\dot{\theta}}_{TR} = K_p \times \Big[K_{\text{relaxation}} D_{TR} - K_{HR\rightarrow TR} D_{HR} \Big]. \tag{8}$$

The *torso flexion-extension joint* (TFE) assists the head nod joint:

$$\widehat{\dot{\theta}}_{TFE} = K_p \times \Big[K_{\text{relaxation}} D_{TFE} - K_{HN\rightarrow TFE} D_{HN} \Big]. \tag{9}$$

The *torso abduction-adduction joint* (TAA) assists the head tilt joint:

$$\widehat{\dot{\theta}}_{TAA} = K_p \times \Big[K_{\text{relaxation}} D_{TAA} - K_{HT\rightarrow TAA} D_{HT} \Big]. \tag{10}$$

Our experiments have shown that this strategy is successful at smoothly pursuing objects of interest. Once the object is stabilized in the fovea, we can use foveal images for object recognition.

IV. Object Representation

Early approaches to object recognition in static images were implemented predominantly around the 3-D reconstruction paradigm of Marr [19], but many of the recent recognition systems make use of viewpoint-dependent models. View-based strategies are receiving an increasing attention because it has been recognized that 3-D reconstruction is difficult in practice and also because of some psychophysical evidence for such strategies [20].

A. Gabor Jets and Gabor Wavelets

The wavelet analysis provides methods of decomposing functions into a linear superposition of wavelets. As discussed in the introduction, we utilize Gabor kernel functions to represent multiple views of objects. Complex Gabor kernels are defined by

$$\Phi(\boldsymbol{x}) = \frac{\|\boldsymbol{k}_{\mu,\nu}\|^2}{\sigma^2} \cdot \exp\left(-\frac{\|\boldsymbol{k}_{\mu,\nu}\|^2\|\boldsymbol{x}\|^2}{2\sigma^2}\right) \cdot \left(\exp\left(i\boldsymbol{k}_{\mu,\nu}^T\boldsymbol{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right), \quad (11)$$

where $\boldsymbol{k}_{\mu,\nu} = k_\nu[\cos(\phi_\mu),\ sin(\phi_\mu)]^T$. 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 need to be selected so that they sample a number of different wavelengths k_ν and orientations ϕ_μ. Wiskott et al. [21] proposed to describe objects by normalized magnitudes of Gabor jets, i.e. $\{a_j^{\boldsymbol{x}}/\|\boldsymbol{a}^{\boldsymbol{x}}\|\}$, where $a_j^{\boldsymbol{x}}$ is the magnitude of the corresponding complex coefficient $J_j^{\boldsymbol{x}}$, $\boldsymbol{a}^{\boldsymbol{x}} = [a_1^{\boldsymbol{x}}, \ldots, a_n^{\boldsymbol{x}}]^T$, and n is the jet dimension. To create a general recognition system, we take pixels $\boldsymbol{x}$ from a regular grid with the spacing of 4 pixels.

Gabor jet representation is highly redundant. It has been proposed to generate a more compressed representation by tuning the parameters of Gabor kernels with respect to the local structure in the image. Instead of calculating the same set of convolutions at every grid point, Krüger and Sommer [8] proposed to automatically determine the pixels in the image, at which the convolutions should be calculated, and the corresponding wavelet parameters. They proposed to use the following family of wavelets for this purpose:

$$\Psi_{\boldsymbol{n}}(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\|\boldsymbol{D}(\boldsymbol{s})\boldsymbol{R}(\theta)(\boldsymbol{x}-\boldsymbol{c})\|^2\right) \cdot \sin\left(\left(\boldsymbol{D}(\boldsymbol{s})\boldsymbol{R}(\theta)(\boldsymbol{x}-\boldsymbol{c})\right)^T \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right), \quad (12)$$

where $\boldsymbol{n} = (c_x, c_y, \theta, s_x, s_y)$, $\boldsymbol{s} = (s_x, s_y)$, $\boldsymbol{c} = (c_x, c_y)$, $\boldsymbol{D}(\boldsymbol{s}) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$ and $\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & -sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$. It is easy to see that if we take $s_x = s_y$ and $c_x = c_y = 0$, (12) is equivalent to the imaginary part of (11). It can be shown that under certain conditions any function $f \in \mathbb{L}^2(\mathbb{R}^2)$ can be represented by a possibly infinite number of wavelets. Krüger and Sommer [8] proposed to approximate an arbitrary image $\boldsymbol{I}$ by minimizing the energy function

$$\min_{\boldsymbol{n}_i, w_i} \|\boldsymbol{I} - \sum_{i=1}^{M} w_i \boldsymbol{\Psi}_{\boldsymbol{n}_i}\|^2. \quad (13)$$

They called the linear superposition of Gabor wavelets $\sum_{i=1}^{M} w_i \boldsymbol{\Psi}_{\boldsymbol{n}_i}$ the Gabor wavelet network for image $\boldsymbol{I}$. Tuning of the parameters $w_i, \boldsymbol{n}_i$ makes Gabor wavelet networks a highly compressed representation in comparison to Gabor jets. Unlike principal component analysis, Gabor wavelet networks can encode local features.

B. Tuning of Wavelet Parameters

Obviously, finding the minimum of criterion (13) is a nonlinear optimization problem that can be solved only iteratively. There typically exist several local minima. We therefore followed the approach of gradually adding wavelets that was initially proposed in [8]. However, to make our representation useful for recognition of objects that rotate in depth, which is essential for natural interaction with a humanoid robot, we need to encode multiple views of objects with a single wavelet network. If each view was encoded with different network, we would need to project the incoming images on a large number of wavelet networks, which would make recognition prohibitively expensive.

We start with the regular grid of initial wavelets. At each step we add to the current network one wavelet from this grid. Let $\boldsymbol{I}_j,\ j = 1, \ldots, N$, be the views of the object that need to be encoded by the network. Let $\boldsymbol{V}_j = \sum_i^{m-1} w_i^j \boldsymbol{\Psi}_{\boldsymbol{n}_i}$ be the current approximates for these views. Note that w_i^j vary with views whereas $\boldsymbol{\Psi}_{\boldsymbol{n}_i}$ are kept the same for all views. The initial parameters $\boldsymbol{n}'_m = (c_{x,m}, c_{y,m}, \theta_m, s_{x,m}, s_{y,m})$ for the next wavelet are taken from the grid and we look for the view $\boldsymbol{I}_j$ which is represented the worst by the current network in the region $\mathcal{N}_m$ centered at $(c_{x,m}, c_{y,m})$. The size of $\mathcal{N}_m$ is defined by $(\theta_m, s_{x,m}, s_{y,m})$. Hence

$$j' = \arg\min_{1 \le j \le M} \|\boldsymbol{I}_j - \boldsymbol{V}_j\|_{\mathcal{N}_m} \quad (14)$$

The next wavelet parameters $\boldsymbol{n}_m$ are determined by minimizing

$$\min_{w_m^{j'}, \boldsymbol{n}_m} \|\boldsymbol{I}_{j'} - \boldsymbol{V}_{j'} - w_m^{j'} \boldsymbol{\Psi}_{\boldsymbol{n}_m}\|^2 \quad (15)$$

using the Levenberg-Marquardt method. The new approximations $\boldsymbol{V}_j$ are then calculated by projecting the views $\boldsymbol{I}_j$ onto the new wavelet network $\{\boldsymbol{\Psi}_{\boldsymbol{n}_i}\}_{i=1}^m$. This process is repeated, possibly with the addition of wavelets from finer grids, until the desired approximation accuracy is achieved.

C. Normalization

To reduce the amount of views that need to be considered in both the training and the recognition phase, we utilize the results of our probabilistic tracker to normalize the images [17]. The tracker is started once the object is detected by the previously described visual search system. The tracker can estimate 2-D locations and sizes of objects and this information is used to ensure invariance against changes in planar position and orientation as well as scale (see Fig. 2). This is accomplished by computing the mapping that transforms the ellipse approximating the object's shape into an ellipse of a fixed size that has both axes aligned with the coordinate axes of the new image window. Hence the size

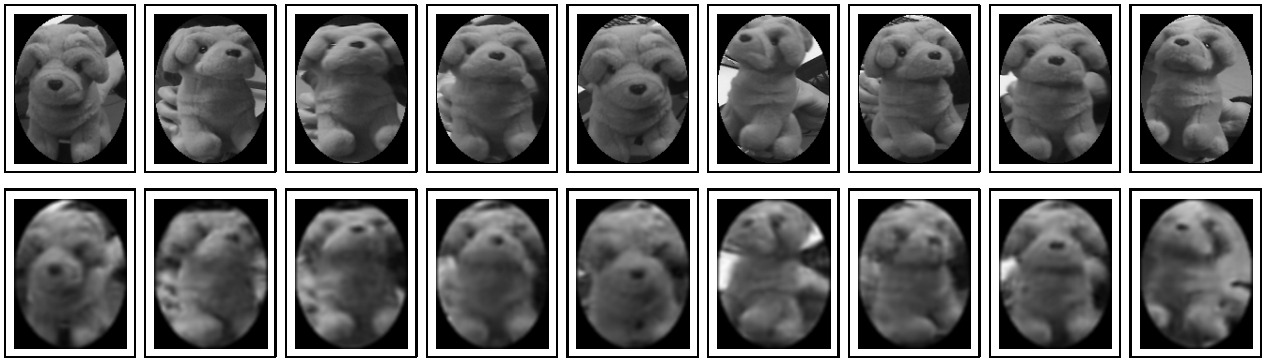

Fig. 2. The upper row shows the original object's appearance and the lower row its reconstruction from the projected wavelet coefficients, all in foveal images for higher resolution. 125 different aspects were represented by a Gabor wavelet network with 282 nodes.

of the object's image is normalized and we do not need to modify the wavelet coefficients to account for differences in scale. This transformation is applied to the collected training data and this assures that we only need to take into account the rotations in depth when sampling the data for a viewpoint-dependent model.

Fig. 2 shows the original images and the images reconstructed from the projections onto the associated wavelet network. The image resolution was fixed at 60×80 pixels. The number of wavelets in our networks was always significantly larger than the number of wavelets used to represent a single view of the object reported in [8], [12]. These authors typically used only 52 wavelets. But this can be expected because our networks must account for a number of different views of the object.

V. Recognition with Support Vector Machines

Support vector machines are a relatively new classification system rooted in the statistical learning theory. They are considered as state of the art classifiers because they deliver high performance in real-world applications. They have been applied to the problem of face recognition [22] and also to more general 3-D object recognition problems [23]. SVMs compute the optimal separating hyperplane between data points belonging to two classes. The hyperplane is optimal in the sense that it separates the largest fraction of points from each class, while maximizing the distance from either class to the hyperplane.

In the previous sections we described two approaches to object representation: Gabor jets and Gabor wavelet networks. To reduce the dimensionality of the Gabor jet representation, we first applied PCA to the training views. Normalized projections of the training jets onto the calculated principal components were used as input for SVM training. In the case of Gabor wavelet networks, the training views were projected onto the network(s) associated with the objects in the database. The projected coefficients were taken as features in this case.

SVMlight software [24] was employed to train the support vector machines. We implemented two classification schemes: one versus the rest, where the goal is to determine whether a particular object is in the scene or not, and the tree structure scheme initially proposed in [22], where the goal is to identify an object when multiple choices are allowed. In this second case each support vector machine is trained to distinguish between two objects and the final result is obtained by elimination.

We also exploit the dynamic nature of our system and run the recognition process on a time sequence of images. The object is deemed recognized only if the identity of the object does not change over a certain period of time. This is based on the assumption that correct classifications are stable whereas misclassifications are not and change as the viewpoint changes. In our experiments we typically used 3 images per second to allow for some interframe motion and waited for two seconds before accepting the recognition result.

Some experimental results are shown in Tab. I - III. The task was to determine whether an object was in the scene or not using only one support vector machine (one versus the rest scheme). For each of the 5 objects in the database, the images of the object under consideration were taken as positive examples and the images of all other objects were taken as negative examples. Training images were collected while the user moved the objects in front of the robot. Gabor jets were used as input to the SVM training. Good performance was achieved when using 200 training images per object at resolution of 120×160 pixels. The performance of the system degraded when we used less training images or lower resolution images. These results cannot be compared to the results on standard databases for benchmarking object recognition algorithms because here the training sets are much less complete. Some of the errors are caused by the lack of data in our models rather than by a deficient classification approach. Our results show that it is possible to recognize objects without using accurate turntables to systematically capture all relevant views.

We were less successful with the representation based on Gabor wavelet networks. To reduce the computing time when generating the networks, we worked at lower resolution (60×80). Fig. 2 shows that we were able to reconstruct the training views reasonably well at this resolution. However, after a more thorough analysis we had to acknowledge that recognition results were not as good as in the Gabor jet case; either the resolution was too low or the Gabor wavelet networks do not generalize well to previously unseen views.

TABLE I

CLASSIFICATION ERRORS (200 VIEWS/OBJECT, 120 × 160 PIXELS)

	false positives	false negatives
teddy bear 1	4.5 %	0.5 %
teddy bear 2	7.8 %	0.3 %
teddy bear 3	1.4 %	1.9 %
toy dog	3.9 %	2.7 %
coffee mug	2.1 %	0.7 %

TABLE II

CLASSIFICATION ERRORS (100 VIEWS/OBJECT, 120 × 160 PIXELS)

	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 %

TABLE III

CLASSIFICATION ERRORS (200 VIEWS/OBJECT, 60 × 80 PIXELS)

	false positives	false negatives
teddy bear 1	10.1 %	2.1 %
teddy bear 2	14.0 %	2.1 %
teddy bear 3	12.0 %	0.2 %
toy dog	10.7 %	2.6 %
coffee mug	2.5 %	0 %

VI. CONCLUSION AND WORK IN PROGRESS

Our experiments have shown that the proposed approach is successful at locating, pursuing and recognizing objects in motion. We have demonstrated how to integrate peripheral and foveal vision on a humanoid robot to solve these problems in real-time.

We are currently working on further statistical evaluation of the performance of the proposed recognition system. It is implemented on two dual processor PCs that concurrently process video streams coming from the peripheral and foveal cameras. However, such an architecture will become too limiting for real-time execution once the complexity of the cognitive tasks increases. Therefore a cluster of processors is being established with the ultimate aim of emulating the human visual system. The cluster can be employed to explore various cognitive architectures, such as the one in this paper. Currently 40 PCs are being used, connected together over a 1Gbit Ethernet network. The vision processing on each PC can range from the most basic (e. g. color extraction, edge filtering, etc) to higher-level (e. g. visual tracking, recognition, etc). The sophistication can increase quite rapidly simply through connecting the processing outputs of simpler elements to the inputs of more advanced processing elements in a bottom-up manner. Manipulation of the lower-level processes can also be performed in a top-down fashion. Thus, this framework will provide the flexibility to explore a greater range of cognitive architectures.

REFERENCES

[1] 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, October 2003, pp. 2173–2178.

[2] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati, "Active vision for sociable robots," *IEEE Trans. Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 443–453, September 2001.

[3] H. Kozima and H. Yano, "A robot that learns to communicate with human caregivers," in *Proc. Int. Workshop on Epigenetic Robotics*, Lund, Sweden, 2001.

[4] B. Scassellati, "Eye finding via face detection for a foveated, active vision system," in *Proc. Fifteenth Nat. Conf. Artifficial Intelligence (AAAI '98)*, Madison, Wisconsin, 1998, pp. 969–976.

[5] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal, "Biomimetic oculomotor control," *Adaptive Behavior*, vol. 9, no. 3/4, pp. 189–208, 2001.

[6] S. Rogeaux and Y. Kuniyoshi, "Robust tracking by a humanoid vision system," in *Proc. IAPR First Int. Workshop on Humanoid and Friendly Robotics*, Tsukuba, Japan, 1998.

[7] G. Metta, F. Panerai, R. Manzotti, and G. Sandini, "Babybot: an artificial developing robotic agent," in *Proc. Sixth Int. Conf. on the Simulation of Adaptive Behaviors (SAB 2000)*, Paris, France, September 2000.

[8] V. Krüger and G. Sommer, "Gabor wavelet networks for efficient head pose estimation," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 665–672, 2002.

[9] T.-S. Lee, "Image representation using 2-D Gabor wavelets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 10, pp. 1–13, 1996.

[10] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.

[11] V. Krüger and G. Sommer, "Wavelet networks for face processing," *J. Opt. Soc. Am. A*, vol. 19, no. 6, pp. 1112–1119, 2002.

[12] C. Hu, R. Feris, and M. Turk, "Active wavelet networks for face alignment," in *Proc. British Machine Vision Conference*, Norwich, UK, 2003.

[13] J. M. Wolfe, "Moving towards solutions to some enduring controversies in visual search," *Trends in Cognitive Sciences*, vol. 7, no. 2, pp. 70–76, February 2003.

[14] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1254–1259, November 1998.

[15] T. S. Horowitz and J. M. Wolfe, "Visual search has no memory," *Nature*, vol. 394, pp. 575–577, August 1998.

[16] P. Verghese, "Visual search and attention: A signal detection theory approach," *Neuron*, vol. 31, pp. 523–535, August 2001.

[17] A. Ude and C. G. Atkeson, "Probabilistic detection and tracking at high frame rates using affine warping," in *Proc. Int. Conf. Pattern Recognition, vol. II*, Québec City, Canada, August 2002, pp. 6–9.

[18] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics Automat.*, vol. 12, no. 5, pp. 651–670, 1996.

[19] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. R. Soc. of London, B*, vol. 200, pp. 269–294, 1978.

[20] M. J. Tarr and H. H. Bülthoff, "Image-based object recognition in man, monkey, and machine," *Cognition*, vol. 67, no. 1-2, pp. 1–20, 1998.

[21] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 775–779, 1997.

[22] G. Guo, S. Z. Li, and K. L. Chan, "Support vector machines for face recognition," *Image and Vision Computing*, vol. 19, pp. 631–638, 2001.

[23] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 637–646, 1998.

[24] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods – Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., Cambridge, Mass., 1999, pp. 768–776.