10

Interactive object learning and recognition with multiclass support vector machines

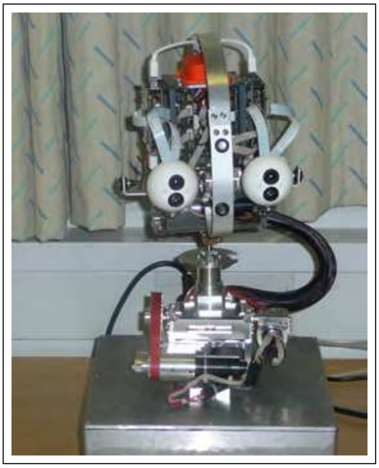
Aleš Ude
*Jožef Stefan Institute,
Slovenia*
*ATR Computational Neuroscience Laboratories,
Japan*

1. Introduction

A robot vision system can be called humanoid if it possesses an oculomotor system similar to human eyes and if it is capable to simultaneously acquire and process images of varying resolution. Designers of a number of humanoid robots attempted to mimic the foveated structure of the human eye. Foveation is useful because, firstly, it enables the robot to monitor and explore its surroundings in images of low resolution, thereby increasing the efficiency of the search process, and secondly, it makes it possible to simultaneously extract additional information – once the area of interest is determined – from higher resolution foveal images that contain more detail. There are several visual tasks that can benefit from foveated vision. One of the most prominent among them is object recognition. General object recognition on a humanoid robot is difficult because it requires the robot to detect objects in dynamic environments and to control the eye gaze to get the objects into the fovea and to keep them there. Once these tasks are accomplished, the robot can determine the identity of the object by processing foveal views.

Approaches proposed to mimic the foveated structure of biological vision systems include the use of two cameras per eye (Atkeson et al., 2000; Breazeal et al., 2001; Kozima & Yano, 2001; Scassellati, 1998) (Cog, DB, Infanoid, Kismet, respectively), i. e. a narrow-angle foveal camera and a wide-angle camera for peripheral vision; lenses with space-variant resolution (Rougeaux & Kuniyoshi, 1998) (humanoid head ESCHeR), i. e. a very high definition area in the fovea and a coarse resolution in the periphery; and space-variant log-polar sensors with retina-like distribution of photo-receptors (Sandini & Metta, 2003) (Babybot). It is also possible to implement log-polar sensors by transforming standard images into log-polar ones (Engel et al., 1994), but this approach requires the use of high definition cameras to get the benefit of varying resolution. Systems with zoom lenses have some of the advantages of foveated vision, but cannot simultaneously acquire wide angle and high resolution images.

Our work follows the first approach (see Fig. 1) and explores the advantage of foveated vision for object recognition over standard approaches, which use equal resolution across the visual field. While log-polar sensors are a closer match to biology, we note that using two cameras per eye can be advantageous because cameras with standard chips can be utilized. This makes it possible to equip a humanoid robot with miniature cameras (lipstick size and



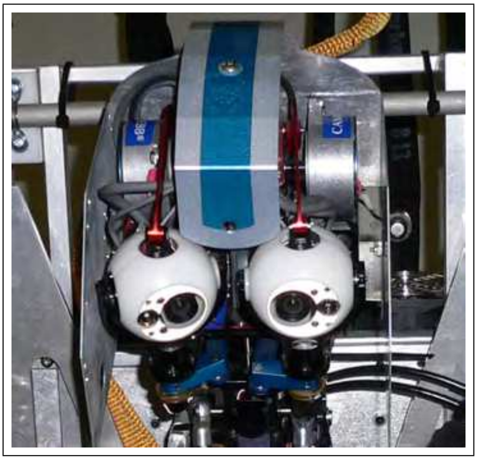


Fig. 1. Two humanoid heads with foveated vision. The left head was constructed by University of Karlsruhe for JSI (Asfour et al., 2008), while the right one is part of a humanoid robot designed by SARCOS and ATR (Cheng et al., 2007). Foveation is implemented by using two cameras in each eye. On the left head, the narrow-angle cameras, which provide foveal vision, are mounted above the wide-angle cameras, which are used for peripheral vision. The right head has foveal cameral on the outer sides of peripheral cameras.

smaller), which facilitates the mechanical design of the eye and improves its motion capabilities.

Studies on oculomotor control in humanoid robots include vestibulo-ocular and optokinetic reflex, smooth pursuit, saccades, and vergence control (Manzotti et al., 2001; Panerai et al., 2000; Rougeaux & Kuniyoshi, 1998; Shibata et al., 2001). On the image processing side, researchers studied humanoid vision for visual attention (Breazeal et al., 2001; Vijayakumar et al., 2001), segmentation (Fitzpatrick, 2003), and tracking (Metta et al., 2004; Rougeaux & Kuniyoshi, 1998). The utilization of foveation for object recognition was not of major concern in these papers. In our earlier work we demonstrated how foveation (Ude et al., 2003) can be used for object recognition. Our initial system employed LoG (Laplacian of the Gaussian) filters at a single, manually selected scale and principal component analysis to represent objects. Two other systems that utilized foveation for object recognition are described in (Arsenio, 2004), who are mainly concerned with using multi-modal cues for recognition, and (Björkman & Kragic, 2004), who present a complete system. In this chapter we focus on the analysis of benefits of foveated vision for recognition.

1.1 Summary of the Approach

On a humanoid robot, foveal vision can be utilized as follows: the robot 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 image areas, upon which it can make a decision about the identity of the object.

Since humanoids operate in dynamic environments and use active vision to monitor the external world, it is necessary that the detection and tracking algorithms are all realized in real-time. To this end, some ground knowledge is normally assumed such as for example color and shape probability distributions of the objects of interest. A suitable detection and tracking algorithm based on such assumptions is described in (Ude et al., 2001), where the details can be found. For the purpose of this chapter it is important to note that in this way we can estimate the location and extent of the object in the image. An important current research topics is how to extract image regions that contain objects without assuming prior knowledge about the objects of interest (Ude et al., 2008).

To support foveal vision we developed a control system whose primary goal is to maintain the visibility of the object based on 2-D information from peripheral views. The developed system attempts to maintain the visibility of the object in foveal views of both eyes simultaneously. The secondary goal of the system is to enhance the appearance of the humanoid through mimicking aspects of human movement: human eyes follow object movement, but without head and body movement have a limited range; thus, the robot's control system supports its eye movements through head and body movements. The details can be found in (Ude et al., 2006).

In the rest of this chapter we describe an approach to object learning and recognition that utilizes the results of these susbsytems to achieve recognition in foveal views.

2. Object Representation

Early approaches to object recognition in static images were implemented predominantly around the 3-D reconstruction paradigm of (Marr & Nishihara, 1978), but many of the more recent recognition systems make use of viewpoint-dependent models (Longuet-Higgins, 1990; Poggio & Edelman, 1990; Sinha & Poggio, 1996). View-based strategies are receiving increasing attention because it has been recognized that 3-D reconstruction is difficult in practice (mainly due to difficulties in segmentation). There is also psychophysical evidence that supports view-based techniques(Tarr & Bülthoff, 1998).

2.1 Normalization through Affine Warping

In view-based systems objects are represented by a number of images (or features extracted from 2-D images) taken from different viewpoints. These model images are compared to test images acquired by the robot. However, since both a humanoid robot and objects can move in space, objects appear in images at different positions, orientations and scales. It is obviously not feasible to learn all possible views due to time and memory limitations. The number of required views can, however, be reduced by normalizing the subimages that contain objects of interest to images of fixed size.

This reduction can be accomplished by utilizing the results of the tracker. Our tracker estimates the shape of the tracked object using second order statistics of pixels that are probabilistically classified as "blob pixels" (Ude et al., 2001). From the second order statistics we can estimate the planar object orientation and the extent of the object along its major and minor axes. In other words, we can estimate the ellipse enclosing the object pixels. As the lengths of both axes can differ significantly, each object image is normalized along the principal axis directions instead of image coordinate axes and we apply a different scaling factor along each of these directions. By aligning the object's axes with the coordinate axes, we also achieve

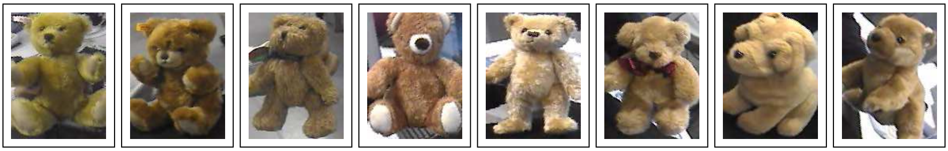


Fig. 2. Example images of eight objects. Scaling and planar rotations are accounted for by affine warping using the results of visual tracking.

invariance against planar rotations, thus reducing the number of views that need to be stored to represent an object because rotations around the optical axis result in the same example images. The results of the normalization process are shown in Fig. 2 and 3.

Normalization along the principal axes is implemented by applying the following transformations: (1) translate the blob so that its center is aligned with the origin of the image, (2) rotate the blob so that its principal directions are aligned with the coordinate axes, (3) scale the blob so that its major and minor axis are as long as the sides of a predefined window, (4) translate the blob so that its center is aligned with the center of the new window. The resulting mapping in homogeneous coordinates is given by the following affine transformation:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & \frac{w_x}{2} \\ 0 & 1 & \frac{w_y}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_x}{2a} & 0 & 0 \\ 0 & \frac{w_y}{2b} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(\theta)^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{u} = [u, v]^T$ and θ are the position and orientation of the blob, a and b are the half lengths of its major and minor axis, and w_x and w_y are the predefined width and height of the window onto which we map the window containing the blob.

The process of geometrically transforming the input image by the affine mapping given in Eq. (1) is known as *affine warping*. Since matrix $\boldsymbol{A}$ is invertible, we implemented affine warping by parsing through the pixels of the output window, which is smaller than the input window, and by applying the inverse mapping $\boldsymbol{A}^{-1}$ to each of the pixels in this window. The associated color intensities at these positions are estimated either by a nearest neighbor or cubic interpolation.

2.2 Gabor Jets

Early view-based approaches used raw grayscale images as input to the selected classifier, e. g. principal component analysis (Turk & Pentland, 1991). This kind of approaches turned out to be fairly successful as long as the amount of noise in the images is small and the illumination conditions do not change. To achieve robustness against brightness changes, it is necessary to compute an improved, illumination insensitive characterization of the local image structure. Some of the more recent recognition systems therefore apply a bank of illumination-insensitive filters to the original images before starting the recognition process. We follow the biologically motivated approach of (Wiskott et al., 1997), who proposed to apply a bank of Gabor filters to the incoming images containing objects of interest. Gabor filters are known to be good edge detectors and are therefore robust against varying brightness. They have limited support both in space and frequency domain and have a certain amount of robustness against translation, distortion, rotation, and scaling (Wiskott et al., 1997).

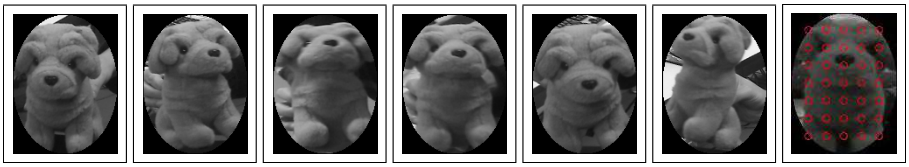


Fig. 3. Training images for one of the objects used in statistical experiments. To take care of rotations in depth, we must collect a sufficient amount of typical viewpoints. The rightmost image shows a regular pixel grid at which feature vectors are calculated. The actual grid was denser than the depicted one.

Complex Gabor kernels are defined by

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

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 are selected so that they sample a number of different wavelengths k_ν and orientations ϕ_μ. (Wiskott et al., 1997) proposed to use $k_\nu = 2^{-\frac{\nu+2}{2}}$, $\nu = 0, \dots, 4$, and $\phi_\mu = \mu\frac{\pi}{8}$, $\mu = 0, \dots, 7$, but this depends both on the size of the incoming images and the image structure. They showed that the similarity between the jets can be measured by

$$\mathrm{S}\left(\{J_i^{\boldsymbol{x}}\}, \{J_i^{\boldsymbol{y}}\}\right) = \frac{\boldsymbol{a}_{\boldsymbol{x}}^T * \boldsymbol{a}_{\boldsymbol{y}}}{\|\boldsymbol{a}_{\boldsymbol{x}}\|\|\boldsymbol{a}_{\boldsymbol{y}}\|}, \tag{3}$$

where $\boldsymbol{a}_{\boldsymbol{x}} = [|J_1^{\boldsymbol{x}}|, \dots, |J_s^{\boldsymbol{x}}|]^T$ and s is the number of complex Gabor kernels. This is based on the fact that the magnitudes of complex coefficients vary slowly with the position of the jet in the image.

We use Gabor jets to generate feature vectors for recognition. To reduce the dimensionality of these feature vectors, we did not make use of all jets. Ideally, one would calculate the jets only at important local features. We did not attempt to extract local features because it is often difficult to extract them in a stable manner. Instead, we decided to build the feature vectors from Gabor jets positioned on a regular grid of pixels (the selected grid size was 5×5). Normalized jets $\{a_j^{\boldsymbol{x}} / \|\boldsymbol{a}^{\boldsymbol{x}}\|\}_{j=1}^n$ calculated on this grid and belonging to the ellipse enclosing the object like in Fig. 3 were finally utilized to build feature vectors.

It is important to note that we first scale the object images to a fixed size and then apply Gabor filters. In this way we ensure that the size of local structure in the acquired images does not change and consequently we do not need to change the frequencies k_ν of the applied filters.

2.3 Training

Our goal is to learn a three-dimensional representation for each object of interest. To achieve this, it is necessary to show the objects to the humanoid from all relevant viewing directions. In computer vision this is normally achieved by accurate turntables that enable the collection of images from regularly distributed viewpoints. However, this solution is not practical for autonomous robots that need to seamlessly acquire new knowledge in natural environments. On the other hand, learning in human environments can effectively be supported by human-robot interaction. We therefore explored whether it is possible to reliably learn 3-D descriptions from images collected while a human teacher moves the object in front of the robot. Using the previously described attention, tracking, and smooth pursuit systems, the robot acquires foveal images of the object in motion and collects feature vectors based on Gabor jets from many different viewpoints (see Figure 3 and 2). In the next section we present our approach to object recognition using Gabor jets, followed by experimental results that show that such collections of feature vectors are sufficient for 3-D object recognition.

3. Recognition with Support Vector Machines

Support vector machines (SVMs) 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. To distinguish between two different classes, a support vector machine draws the (optimal) separating hyperplane between training data points belonging to the 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. First approaches that utilized SVMs for object recognition applied the basic method that deals with a two-class classification problem. In the context of object recognition, a binary tree strategy (Guo et al., 2001; Pontil & Verri, 1998) was proposed to solve the multi-class problem. While this approach provides a simple and powerful classification framework, it cannot capture correlations between the different classes since it breaks a multi-class problem into multiple independent binary problems (Crammer & Singer, 2001). In addition, the result is not independent of how the candidate objects are paired. There were attempts to generalize SVMs to multi-class problems, but practical implementation have started to emerge only recently. Here we follow the generalization proposed in (Crammer & Singer, 2001), which is briefly described in Section 3.1. To improve classification results we utilized nonlinear variant of support vector machines with a specially designed kernel function that exploits the properties of Gabor jets. We made use of the implementation described in (Joachims, 1999; Tsochantaridis et al., 2004).

3.1 Nonlinear Multiclass Support Vector Machines

Multi-class classification addresses the problem of finding a function defined from an input space $\Psi \subset \mathcal{R}^n$ onto a set of classes $\Omega = \{1, \ldots, m\}$. Let $\boldsymbol{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, $\boldsymbol{x}_i \in \Psi$, $y_i \in \Omega$, be a set of n training samples. We look for a function $\boldsymbol{H} : \Psi \rightarrow \Omega$ so that $\boldsymbol{H}(\boldsymbol{x}_i) = y_i$. (Crammer & Singer, 2001)[1] proposed to search for $\boldsymbol{H}$ among linear classifiers of the form

$$\boldsymbol{H}_{\boldsymbol{M},\boldsymbol{b}}(\boldsymbol{x}) = \underset{r \in \Omega}{\arg\max} \{\boldsymbol{M}_r * \boldsymbol{x} + b_r\}, \tag{4}$$

where $\boldsymbol{b} = [b_1, \ldots, b_k]^T$, $\boldsymbol{M} \in \mathcal{R}^{m \times n}$ is a matrix of size $m \times n$ and $\boldsymbol{M}_r$ is the r-th row of $\boldsymbol{M}$. Standard two-class SVMs result in classifiers $\boldsymbol{H} = (\boldsymbol{w}, b)$ that predict the label of a data point $\boldsymbol{x}$ as 1 if $\boldsymbol{w} * \boldsymbol{x} + b \geq 0$ and 2 otherwise. They can be expressed in the above form by taking a matrix $\boldsymbol{M}$ with rows $\boldsymbol{M}_1 = \boldsymbol{w}$, $\boldsymbol{M}_2 = -\boldsymbol{w}$, and $b_1 = b$, $b_2 = -b$.

[1] The bias parameters b_r were omitted in (Crammer & Singer, 2001) to simplify the optimization problem. Here we keep them in the interest of clarity of presentation.

The following error function can be used to to evaluate the performance of a multi-class predictor of form (4)

$$\frac{1}{n}\sum_{i=1}^{n}\left(\max_{r\in\Omega}\left\{\boldsymbol{M}_r * \boldsymbol{x}_i + b_r + 1 - \delta_{y_i,r}\right\} - \boldsymbol{M}_{y_i} * \boldsymbol{x}_i - b_{y_i}\right). \tag{5}$$

Here $\delta_{y_i,r} = 1$ if $y_i = r$ and 0 otherwise. The above criterion function is always greater or equal zero and has its minimum at zero if for each data point $\boldsymbol{x}_i$ the value $\boldsymbol{M}_r * \boldsymbol{x}_i + b_r$ is larger by at least 1 for the correct label than for all other labels. In this case, the sample S is called linearly separable and it satisfies the constraints

$$\boldsymbol{M}_{y_i} * \boldsymbol{x}_i + b_{y_i} + \delta_{y_i,r} - \boldsymbol{M}_r * \boldsymbol{x}_i - b_r \geq 1,\ \forall\, i, r. \tag{6}$$

Obviously, any $(\boldsymbol{M}, \boldsymbol{b})$ satisfying these conditions results in a decision function that classifies all samples correctly.
To generalize support vector learning to non-separable problems, slack variables $\xi_i \geq 0$ need to be introduced. In this case, constraints (6) become

$$\boldsymbol{M}_{y_i} * \boldsymbol{x}_i + b_{y_i} + \delta_{y_i,r} - \boldsymbol{M}_r * \boldsymbol{x}_i - b_r \geq 1 - \xi_i,\ \forall\, i, r. \tag{7}$$

(Crammer & Singer, 2001) showed that optimal multi-class support vector machine can be calculated by solving a quadratic programming optimization problem:

$$\min_{\boldsymbol{M},\boldsymbol{b},\boldsymbol{\xi}} \frac{1}{2}\beta\left(\sum_{r=1}^{m}\sum_{i=1}^{n} M_{ri}^2 + \sum_{r=1}^{m} b_r^2\right) + \sum_{i=1}^{n}\xi_i \tag{8}$$

$$\begin{aligned}\text{subject to :}\quad & \boldsymbol{M}_{y_i} * \boldsymbol{x}_i + b_{y_i} + \delta_{y_i,r} - \boldsymbol{M}_r * \boldsymbol{x}_i - b_r \geq 1 - \xi_i \\ & \xi_i \geq 0, \forall\, i, r\end{aligned}$$

Here $\boldsymbol{\xi}_i \geq 0$ are the slack variables that need to be introduced to solve non-separable problems. This constrained quadratic optimization problem is convex and can therefore be solved efficiently. Note that in optimization problem (8) the data points appear only in inner products $\boldsymbol{M}_j * \boldsymbol{x}_i$. Furthermore, the same authors proved that the rows of the optimal classifier matrix $\boldsymbol{M}$ are given by

$$\boldsymbol{M}_r^T = \sum_{i=1}^{n} \tau_{ir}\boldsymbol{x}_i,\ \tau_{i,r} \geq 0,\ r = 1, \ldots, m, \tag{9}$$

and the corresponding decision function can be written as

$$\boldsymbol{H}_{\boldsymbol{M},\boldsymbol{b}}(\boldsymbol{x}) = \arg\max_{r\in\Omega}\left\{\sum_{i=1}^{n}\tau_{i,r}\boldsymbol{x}_i^T * \boldsymbol{x} + b_r\right\}. \tag{10}$$

Here the data points appear only in inner products $\boldsymbol{x}_i^T * \boldsymbol{x}$. Lets assume now that the data points were transformed with a nonlinear mapping Φ, which maps the data into a possibly higher dimensional feature space. The optimal hyperplane can then be constructed in this space and the scalar products $\boldsymbol{x} * \boldsymbol{y}$ are replaced by $\Phi(\boldsymbol{x}) * \Phi(\boldsymbol{y})$. The main idea is to find a feature space in which it is easier to separate the classes than in the original data space.

The nonlinear mappings of interest are those that allow for an efficient calculation of high-dimensional inner products via kernel functions

$$\mathrm{K}(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{x}) * \Phi(\boldsymbol{y}). \tag{11}$$

To find the optimal multi-class support vector machine in a higher dimensional feature space, we need to solve a constrained quadratic optimization problem in which inner products $\Phi(\boldsymbol{x}_i) * \Phi(\boldsymbol{x}_j)$ are replaced with the kernel function $\mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The decision function (10) becomes

$$\boldsymbol{H}_{\boldsymbol{M},\boldsymbol{b}}(\boldsymbol{x}) = \arg\max_{r\in\Omega} \left\{ \sum_{i=1}^{n} \tau_{i,r}\mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b_r \right\}. \tag{12}$$

The convergence of the optimization algorithm can be guaranteed for all kernel functions K that allow the construction of nonlinear mapping Φ such that (11) holds. The condition for this is given by the Mercer's theorem (Burges, 1998).

3.2 Kernel Functions for Gabor Jets

The similarity measure for Gabor jets (3) provides a good starting point to define a suitable decision function. Let $\boldsymbol{X}_G$ be the set of all grid points within two normalized images at which Gabor jets are calculated and let $J_{\boldsymbol{X}_G}$ and $L_{\boldsymbol{X}_G}$ be the Gabor jets calculated in two different images, but on the same grid points. Based on (3), we define the following kernel function

$$\mathrm{K}_G(J_{\boldsymbol{X}_G}, L_{\boldsymbol{X}_G}) = \exp\left(-\rho_1 \frac{1}{M} \sum_{\boldsymbol{x}\in\boldsymbol{X}_G} \left(1 - \frac{\boldsymbol{a}_{\boldsymbol{x}}^T * \boldsymbol{b}_{\boldsymbol{x}}}{\|\boldsymbol{a}_{\boldsymbol{x}}\|\|\boldsymbol{b}_{\boldsymbol{x}}\|}\right)\right), \tag{13}$$

where M is the number of grid points in $\boldsymbol{X}_G$. This function satisfies the Mercer's condition (Burges, 1998) and can thus be used for support vector learning.

Kernel function (13) assumes that the set of grid points $\boldsymbol{X}_G$ does not change from image to image. (Wallraven et al., 2003) showed that it is possible to define kernel functions using local feature detectors computed on sets of image points that vary from image to image. They designed kernel functions defined on feature vectors of variable lengths and with different ordering of features. While feature order is not a problem for our system due to the affine warping procedure, it would be advantageous to exclude some of the grid points because due to noise some of the points in the warped images do not belong to the object, even after clipping the parts outside of the enclosing ellipse. We can, however, use the results of the tracking/segmentation to exclude such points from the calculation. For each pixel, our tracker (Ude et al., 2001) can estimate the probability whether or not this pixel belong to the tracked object. We can thus define the set $\boldsymbol{X}_G$ on each image to include only points for which these probabilities are greater than a pre-specified threshold. Let $\boldsymbol{X}_G^1$ and $\boldsymbol{X}_G^2$ be two sets of grid points with tracking probabilities greater than a pre-specified threshold. We can define a new kernel function

$$\mathrm{K}'_G(J_{\boldsymbol{X}_G^1}, L_{\boldsymbol{X}_G^2}) = \boldsymbol{K}_G(J_{\boldsymbol{X}_G^1\cap\boldsymbol{X}_G^2}, L_{\boldsymbol{X}_G^1\cap\boldsymbol{X}_G^2}) \cdot \exp\left(-\rho_1 \frac{1}{M}\left(\sum_{\boldsymbol{x}\in\boldsymbol{X}_G^1\cup\boldsymbol{X}_G^2 - \boldsymbol{X}_G^1\cap\boldsymbol{X}_G^2} 2\right)\right) \tag{14}$$

where M is the number of grid points in $\boldsymbol{X}_G^1 \cup \boldsymbol{X}_G^2$. We add the penalty of 2 for grid points that are not classified as object points only in one of both images because this is the highest possible value for one term in the criterion function (13). The reasoning for this is that if a pixel does not belong to the object, the Gabor jets calculated at this point are meaningless. We should therefore add the highest possible penalty for the function of type (3). While this kernel function assumes that the ordering of grid points is the same in both images, it is much less computationally expensive than the more general functions proposed in (Wallraven et al., 2003). This is important both for faster training and for real-time recognition.

4. Experimental results

We used a set of ten objects to test the performance of the recognition system on a humanoid robot (6 teddy bears, two toy dogs, a coffee mug, and a face). For each object we recorded two or more movies using a video stream coming from the robot's foveal cameras. In each of the recording sessions the experimenter attempted to show one of the objects to the robot from all relevant viewing directions. One movie per object was used to construct the SVM classifier, while one of the other movies served as input to test the classifiers. Thus the support vector machine was trained to distinguish between 10 classes. Each movie was one minute long and we used at most 4 images per second (out of 30) for training. Since slightly more than first ten seconds of the movies were needed to initialize the tracker, we had at most 208 training images per object. For testing we used 10 images per second, which resulted in 487 test images per object. Except for the results of Table 4, all the percentages presented here were calculated using the classification results obtained from 4870 test images. Three types of classifiers were used to test the performance of foveated recognition. The first one was a nonlinear multi-class support vector machine based on kernel functions K_G and K'_G from Eq. (13) and (14), respectively. It is denoted as SVM nonlinear in Table 1 - 6. Gabor jets were calculated at 8 different orientations and 5 different scales and the grid size was 5 pixels in both directions. The filters were scaled appropriately when using lower resolution images. The second classifier we tested was a more standard linear multi-class SVM using the same feature vectors. It is denoted by SVM linear in the tables. The third classifier was the nearest neighbor classifier (NNC) that used the similarity measure (3) – summed over all grid points – to calculate the nearest neighbor based on the same Gabor jets as input data.

Results in Tables 1 - 3 demonstrate that foveation is useful for recognition. Kernel function (14) was used here. The classification results clearly become worse with the decreasing resolution. In fact, the data of Table 3 had to be calculated differently because we could not estimate the planar orientation accurately enough for affine warping, which made the normalization procedure fail. This resulted in significantly worse classification results. To calculate the results of Table 3, we sampled the Gabor jets on the images of size 160 × 120 with a 20 sampling grid, which resulted in the same number of grid points as when image resolution is reduced from

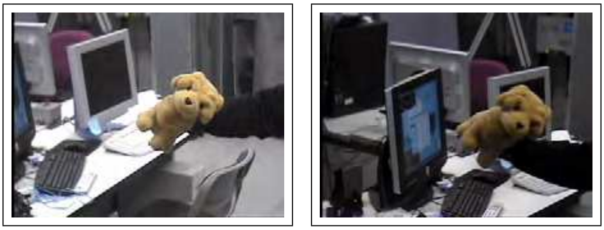


Fig. 4. Images taken under different lighting conditions

Training views per object	SVM nonlinear	SVM linear	NNC
208	97.6 %	96.8 %	95.9 %
104	96.7 %	95.3 %	93.7 %
52	95.1 %	94.0 %	91.5 %
26	91.9 %	89.9 %	86.7 %

Table 1. Correct classification rate (image resolution 120×160 pixels)

Training views per object	SVM nonlinear	SVM linear	NNC
208	94.2 %	94.4 %	89.3 %
104	92.4 %	91.1 %	87.3 %
52	90.7 %	89.7 %	84.4 %
26	86.7 %	84.5 %	79.2 %

Table 2. Correct classification rate (image resolution 60×80 pixels)

Training views per object	SVM nonlinear	SVM linear	NNC
208	91.0 %	89.6 %	84.7 %
104	87.2 %	85.8 %	81.5 %
52	82.4 %	81.1 %	77.8 %
26	77.1 %	75.5 %	72.1 %

Table 3. Correct classification rate (image resolution 30×40 pixels)

160×120 to 40×30 and the grid size is kept the same. The recognition rate dropped even with such data. Our results also show that we can collect enough training data even without using accurate turntables to systematically collect the images. While based on these results it is not possible to say what is the maximum number of objects that can be recognized using the proposed approach, we note that the method produced similar recognition rates when only subsets of objects were used for training and classification.

We also tested the performance of the system on data captured under changed lighting condition (see Fig. 4) and on noise corrupted data (see Fig. 5, two objects – a teddy bear and a toy dog – were used in this experiment). The classification performance for these two objects on original images was a bit higher than the combined performance, but this was purely coincidental and we did not intentionally select these two object to test the varying brightness condition. For classification we used the same SVMs as in Tables 1-3. While the performance decreased slightly on darker images, the results show that the method still performs well in such conditions. This is due to the properties of Gabor jets and due to the normalization of jets given by the similarity function (3). Our experiments showed that the classification rate drops significantly if one of the standard kernel functions, e. g. a linear kernel, is used for the support vector learning.

Unlike in other tables, the results of Table 5 and 6 were calculated using SVMs based on kernel function K_G from Eq. (13), thus not taking into account the segmentation results. The

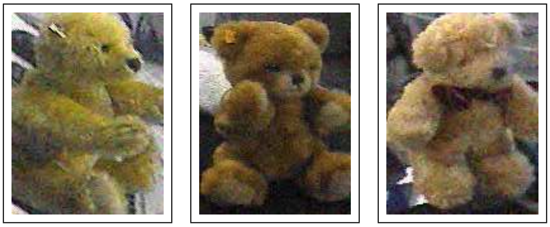


Fig. 5. Images degraded with white Gaussian noise (std. dev. = 10)

segmentation results were not used for the nearest neighbor classification either. Comparing Tables 5 and 6 we can say that SVMs are robust against noise as well. The results of Table 6 can be directly compared to Table 2, the only difference being in the use of segmentation results. While both types of SVMs performed well in this case, the performance of the nearest neighbor classification dropped significantly when all the data from the enclosing ellipse was used. This shows that unlike nearest neighbor classification, SVMs can cope well with outliers. Nevertheless, it is still advantageous to use the kernel function that can include the segmentation results because such an approach reduces the amount of data that needs to be considered to calculate SVMs, hence resulting in faster computation times. We expect that differences between the two types of kernel functions would become more significant for objects that cannot be accurately enclosed within an ellipse.

The presented results cannot be directly compared to the results on standard databases for benchmarking object recognition algorithms because here the training sets are much less complete. Some of the classification errors are caused by the lack of training data rather than by a deficient classification approach. Unlike many approaches from the computer vision literature that avoid the problem of finding objects, we tested the system on images obtained through a realistic object tracking and segmentation procedure. Only such data is relevant for foveated object recognition because without some kind of object detection procedure it is not possible to direct the fovea towards the objects of interest.

5. Conclusions

Using foveation control, our system can learn truly three-dimensional object representations just by collecting the data while the demonstrator attempts to show the objects from all relevant viewing directions. Our experimental results demonstrate that this statistical approach

Image resolution	normal	dark	very dark
120 × 160	99.5 %	97.7 %	97.9 %
60 × 80	96.7 %	93.5 %	95.0 %
30 × 40	93.6 %	89.3 %	88.2 %

Table 4. Correct classification rate for images with varying lighting conditions (see Fig. 4). Only two object were tested in this case (the database still contained ten objects) and nonlinear SVMs calculated based on 208 views per training objects were used.

Training views per object	SVM	NNC
208	91.5 %	79.8 %
104	90.7 %	74.5 %
52	90.5 %	68.0 %
26	87.1 %	60.3 %

Table 5. Correct classification rate for noise degraded images (see Fig. 5). The image resolution was 60 × 80 and segmentation results were not used. Nonlinear SVMs were used in this experiment.

Training views per object	SVM	NNC
208	94.4 %	75.8 %
104	93.1 %	69.2 %
52	91.4 %	60.3 %
26	88.1 %	53.6 %

Table 6. Correct classification rate without noise degradation. The image resolution was 60 × 80 and segmentation results were not used. Nonlinear SVMs were used in this experiment.

is sufficient for object learning and that it is not necessary to use specially designed turntables to accurately collect the views from all relevant viewing directions. Our experimental results prove (see Tab. 1 - 3) that higher resolution images provided by foveation control significantly improve the classification rates of object recognition. In addition, previous approaches that employed support vector machines for object recognition used binary SVMs combined with decision trees (Guo et al., 2001; Pontil & Verri, 1998; Wallraven et al., 2003) to solve the multi-class recognition problem. We proposed a new recognition system that makes use of multi-class nonlinear SVMs to solve the multi-class recognition problem. We also developed a new kernel function based on the Gabor jet similarity measure that can utilize the results of bottom-up segmentation. Experimental results show high recognition rates in realistic test environments.

6. References

Arsenio, A. M. (2004). Object recognition from multiple percepts, *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robots (Humanoids 2004)*, Los Angeles, California, USA.

Asfour, T., Azad, P., Welke, K., Ude, A. & Dillmann, R. (2008). The Karlsruhe humanoid head, *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robots (Humanoids 2008)*, Daejeon, Korea. To appear.

Atkeson, C. G., Hale, J., Pollick, F., Riley, M., Kotosaka, S. Schaal, S., Shibata, T., Tevatia, G., Ude, A., Vijayakumar, S. & Kawato, M. (2000). Using humanoid robots to study human behavior, *IEEE Intelligent Systems* **15**(4): 46–56.

Björkman, M. & Kragic, D. (2004). Combination of foveal and peripheral vision for object recognition and pose estimation, *Proc. 2004 IEEE Int. Conf. Robotics and Automation*, New Orleans, Louisiana, pp. 5135–5140.

Breazeal, C., Edsinger, A., Fitzpatrick, P. & Scassellati, B. (2001). Social constraints on animate vision, *IEEE Trans. Systems, Man, and Cybernetics* **31**(5): 443–452.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Min. Knowl. Discov.* **2**(2): 121–167.

Cheng, G., Hyon, S.-H., Morimoto, J., Ude, A., Hale, J. G., Colvin, G., Scroggin, W. & Jacobsen, S. C. (2007). CB: a humanoid research platform for exploring neuroscience, *Advanced Robotics* **21**(10): 1097–1114.

Crammer, K. & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of Machine Learning Research* **2**: 265–292.

Engel, G., Greve, D. N., Lubin, J. M. & Schwartz, E. L. (1994). Space-variant active vision and visually guided robotics: Design and construction of a high-peformance miniature vehicle, *Proc. 12th IAPR Int. Conf. Pattern Recognition. Vol. 2 - Conf. B: Computer Vision & Image Processing*, Jerusalem, Israel, pp. 487 – 490.

Fitzpatrick, P. (2003). First contact: an active vision approach to segmentation, *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, Nevada, pp. 2161–2166.

Guo, G., Li, S. Z. & Chan, K. L. (2001). Support vector machines for face recognition, *Image and Vision Computing* **19**(9-10): 631–638.

Joachims, T. (1999). Making large-scale support vector machine learning practical, *in* B. Schölkopf, C. J. C. Burges & A. J. Smola (eds), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, Cambridge, MA.

Kozima, H. & Yano, H. (2001). A robot that learns to communicate with human caregivers, *Proc. Int. Workshop on Epigenetic Robotics*, Lund, Sweden.

Longuet-Higgins, H. C. (1990). Recognizing three dimensions, *Nature* **343**: 214–215.

Manzotti, R., Gasteratos, A., Metta, G. & Sandini, G. (2001). Disparity estimation on log-polar images and vergence control, *Computer Vision and Image Understanding* **83**(2): 97–117.

Marr, D. & Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes, *Proc. R. Soc. of London, B* **200**: 269–294.

Metta, G., Gasteratos, A. & Sandini, G. (2004). Learning to track colored objects with log-polar vision, *Mechatronics* **14**: 989–1006.

Panerai, F., Metta, G. & Sandini, G. (2000). Visuo-inertial stabilization in space-variant binocular systems, *Robotics and Autonomous Systems* **30**(1-2): 195–214.

Poggio, T. & Edelman, S. (1990). A network that learns to recognize three-dimensional objects, *Nature* **343**: 263–266.

Pontil, M. & Verri, A. (1998). Support vector machines for 3D object recognition, *IEEE Trans. Pattern Anal. Machine Intell.* **20**(6): 637–646.

Rougeaux, S. & Kuniyoshi, Y. (1998). Robust tracking by a humanoid vision system, *Proc. IAPR First Int. Workshop on Humanoid and Friendly Robotics*, Tsukuba, Japan.

Sandini, G. & Metta, G. (2003). Retina-like sensors: motivations, technology and applications, *in* F. G. Barth, J. A. C. Humphrey & T. W. Secomb (eds), *Sensors and Sensing in Biology and Engineering*, Springer-Verlag, Wien-New York.

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

Shibata, T., Vijayakumar, S., Jörg Conradt, J. & Schaal, S. (2001). Biomimetic oculomotor control, *Adaptive Behavior* **9**(3/4): 189–208.

Sinha, P. & Poggio, T. (1996). Role of learning in three-dimensional form perception, *Nature* **384**: 460–463.

Tarr, M. J. & Bülthoff, H. H. (1998). Image-based object recognition in man, monkey, and machine, *Cognition* **67**(1-2): 1–20.

Tsochantaridis, I., Hofmann, T., Joachims, T. & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces, *Proc. Twenty-first Int. Conf. Machine Learning*, Banff, Alberta, Canada. Article No. 104.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition, *Journal of Cognitive Neuroscience* **3**(1): 71–86.

Ude, A., Atkeson, C. G. & Cheng, G. (2003). Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act, *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, Nevada, pp. 2173–2178.

Ude, A., Gaskett, C. & Cheng, G. (2006). Foveated vision systems with two cameras per eye, *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, Florida, pp. 3457–3462.

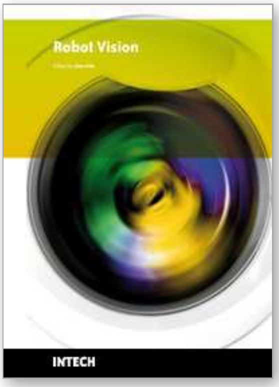
Ude, A., Omrčen, D. & Cheng, G. (2008). Making object learning and recognition an active process, *International Journal of Humanoid Robotics* **5**(2).

Ude, A., Shibata, T. & Atkeson, C. G. (2001). Real-time visual system for interaction with a humanoid robot, *Robotics and Autonomous Systems* **37**(2-3): 115–125.

Vijayakumar, S., Conradt, J., Shibata, T. & Schaal, S. (2001). Overt visual attention for a humanoid robot, *Proc. 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Maui, Hawaii, USA, pp. 2332–2337.

Wallraven, C., Caputo, B. & Graf, A. (2003). Recognition with local feature: the kernel recipe, *Proc. Ninth IEEE Int. Conf. Computer Vision*, Nice, France, pp. 257–264.

Wiskott, L., Fellous, J.-M., Krüger, N. & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching, *IEEE Trans. Pattern Anal. Machine Intell.* **19**(7): 775–779.



Robot Vision
Edited by Ales Ude

ISBN 978-953-307-077-3
Hard cover, 614 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ales Ude (2010). Interactive Object Learning and Recognition with Multiclass Support Vector Machines, Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, Available from: http://www.intechopen.com/books/robot-vision/interactive-object-learning-and-recognition-with-multiclass-support-vector-machines

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821