This work has been published in Computer Vision and Image Understanding, 77, 2000

ORASSYLL: Object Recognition with Autonomously Learned and Sparse Symbolic Representations Based on Metrically Organized Local Line Detectors

(Object Recognition with ORASSYLL))

Norbert Krüger
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Preußerstrasse 1-9
24105 Kiel
Germany
Tel:++49 431 / 560485
Fax:++49 431 / 560481
email:nkr@.ks.informatik.uni-kiel.de

Gabriele Peters
Institut für Neuroinformatik,
Ruhr-Universität Bochum,
44780 Bochum
Germany
gpeters@neuroinformatik.ruhr-uni-bochum.de
Tel: ++49 234 700 7971
Fax: ++49 234 7094 210

Corresponding author: Norbert Krüger

Abstract

We introduce an object recognition and localization system in which objects are represented as a sparse and spatially organized set of local (bent) line segments. The line segments correspond to binarized Gabor wavelets or banana wavelets, which are bent and stretched Gabor wavelets. These features can be metrically organized, the metric enables an efficient learning of object representations. It is essential for learning that only corresponding local areas are compared with each other, i.e., the correspondance problem has to be solved. We achieve correpondence (and in this way autonomous learning) by utilizing motor–controlled feedback, i.e., by interaction of arm movement and camera tracking. The learned representations are used for fast and efficient localization and discrimination of objects in complex scenes.

1 Introduction

Extracting meaningful structures from data is a difficult problem which is for a broad class of applications not satisfactorily solved. On the one hand, there exists a large variety of artificial object recognition systems in which manually generated representations of objects are used to locate and discriminate objects, e.g. [22, 49, 45]. Just as an example, in [22] faces are located successfully by matching a manually defined face model with a certain number of free parameters enabling the adaptation to a specific face in a specific pose. Because the model of the face is defined manually, each time the algorithm is applied to a new object class, a new representation has to be designed manually again. In this way in [4] resistors are localized within the framework of the object representation in [22]. On the other hand, the perspective of the neural network community to use artificial neural nets with little manual intervention as a "black box" has shown its limited success having its roots in the bias/variance dilemma [11]: If the starting configuration of the system is very general it will have to pay for this advantage by having many internal degrees of freedom resulting in bad generalization abilities —the "variance" problem. On the other hand, if the initial system has few degrees of freedom it may be able to learn efficiently, but there is great danger that the structural domain spanned by those degrees of freedom does not cover the given domain of application —the "bias" problem.

In this paper we describe a novel object recognition system called ORASSYLL (**O**bject **R**ecognition with **A**utonomously learned and **S**parse **SY**mbolic representations based on Metrically Organized **L**ocal **L**ine detectors). In ORASSYLL meaningful structure can be learned from training data with no or only little manual intervention. Extraction of meaningful structure becomes possible by using appropriately structured *a priori* knowledge.

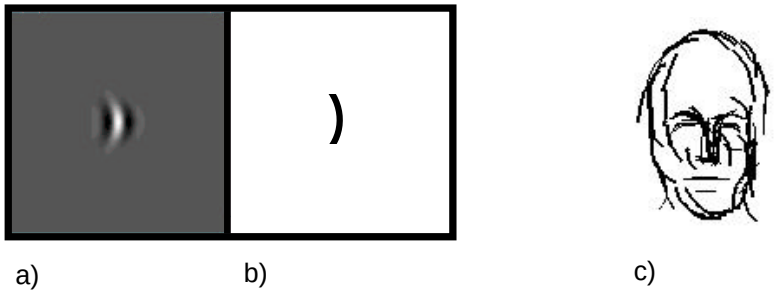

Figure 1: Path corresponding to a banana wavelet. a: Arbitrary wavelet. b: Corresponding path. c: Visualization of a representation of an object class. Gabor or Banana wavelets with lower frequencies are represented by line segments with larger width.

We introduce a number of *a priori* principles to reduce the dimension of the search space and to guide learning (i.e., to handle the variance–problem). We expect to avoid the bias–problem because of the general applicability of those principles. Important constraints are:

PF1 Restriction of object representations to features of a parametrized space corresponding to localized (bent) lines.

PF2 Metric organization of this feature space indicating differences in the feature's properties orientation, curvature and position.

PF3 Hierarchical processing of features.

PF4 Sparse coding.

Other contraints are concerned with the division of the feature space into independent subspaces (PL1: Independence), its temporal organization (PL2: Correspondence) and statistical criteria for the evaluation of significant features for an object class (Invariance Maximization (PE1) and Redundancy Reduction (PE2)). The necessity and biological plausibility of the constraints are discussed in detail in [18, 15].

In section 2 we formalize PF1 by assigning a local line segment to Gabor wavelets or banana wavelets respectively (see figure 1a,b). In addition to the parameters frequency and orientation banana wavelets possess the properties curvature and elongation. The space of banana wavelet responses is much larger than the original image: For each quality (e.g. orientation or curvature) an image, each representing the likelihood of occurence

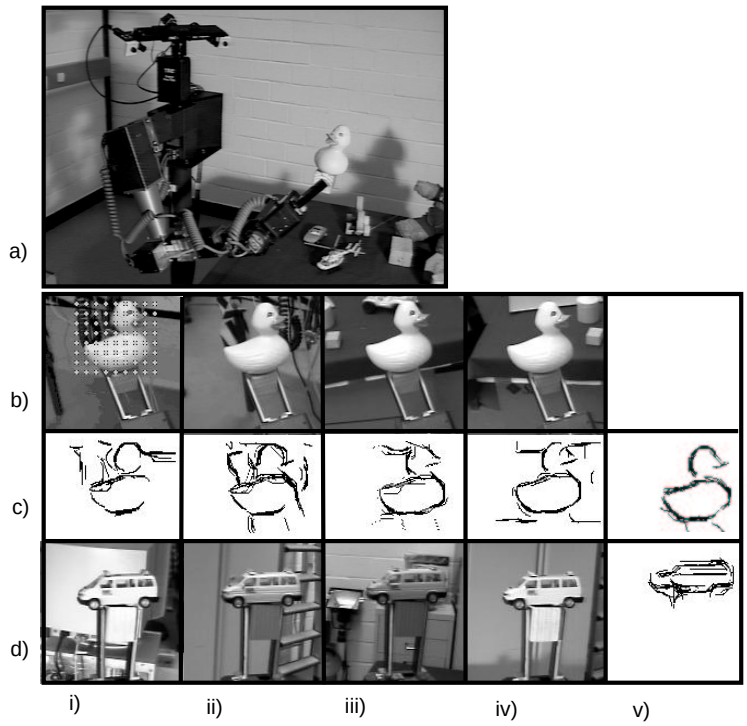

Figure 2: a) The robot arm with the camera. b) The "retinal" images produced by a camera following the robot arm holding a toy–duck. c,i–iv) Significant Features per Instance extracted in a rectangular region (shown in b,i). c,v) Learned representation. d) Training data and learned representation for a toy car.

---

at all pixel positions, is evaluated. In this way we create a feature space up to 240 times larger than the original image. An object can be represented as a configuration of a few of these features, therefore it can be coded sparsely (PF4). The feature space can be understood as a metric space (PF2), its metric representing the similarity of features. This metric is essential for feature extraction and the learning algorithm (section 3.2). The banana wavelet responses can be derived from Gabor wavelet responses by hierarchical processing (PF3) to gain speed and reduce memory requirements. The sparse representation combined with the hierarchical feature processing allows a fast and effective locating.

In order to avoid the necessity of manual intervention for the generation of ground

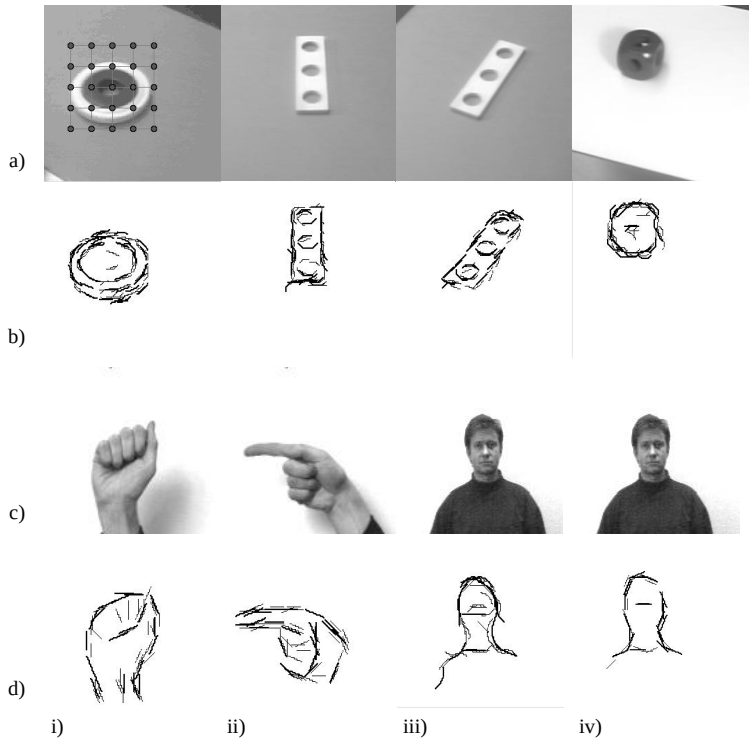

Figure 3: One–shot learning: Row a) and c) show the objects to be learned in front of homogeneous backgound. Row b) and d) show the extracted representations. For all objects a rectangular grid was roughly positioned on the object as in the first image a,i).

truth we equip the system with a mechanism which can produce controlled training data by moving an object with a robot arm and following the object by fixating the robot hand. The robot produces training data on which a certain view of an object is shown with varying background and illumination but with corresponding landmarks having the same pixel position in the image (see figure 2). We apply a learning algorithm to these data to extract object representations comprising only the important features (see figure 2v). Another way to avoid manual intervention is one–shot learning (see figure 3), which already allows for the extraction of representations successfully applicable to difficult discrimination tasks.

This paper is organized in the following way: In section 2 we describe our feature processing and the organization of the feature space. The learning of object representations

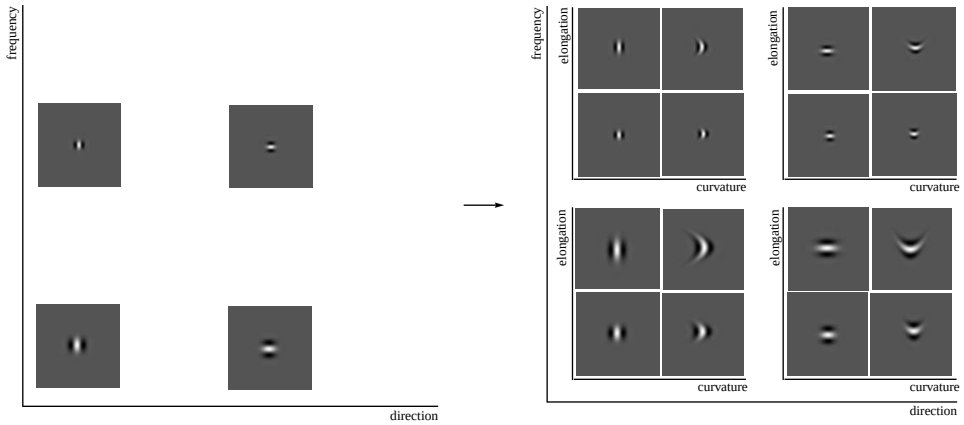

Figure 4: Relation between Gabor wavelets (left) and banana wavelets (right).

is described in section 3. In section 4 we apply these representations to object finding and discrimination. Simulations are presented in section 5. ORASSYLL is influenced — both, in terms of analogy and in terms of criticism — by another well known system [21, 46]. In section 6 ORASSYLL is compared with [21, 46]. We discuss differences to other object recognition systems in section 7. In the outlook we discuss further perspectives of our work. This work is based on the PhD thesis [18], in which (in addition to the object recognition system) the biological motivation, a detailed discussion of the *a priori* constraints and some results about the statistics of natural images in connection with feature transformations within ORASSYLL are discussed.

In order to give the reader the opportunity to understand the algorithm without going through all the formalisms in most of the subsections first a short non-formal description is given. Then, introduced by phrases such as "formally speaking" or "more formally" a precise definition follows.

2 The Feature Space

In this section we describe the realization of the constraints PF1, PF2 and PF3:

- feature generation based on banana wavelets, which are generalized Gabor wavelets (section 2.1),
- their metric organization in the feature space (section 2.2),

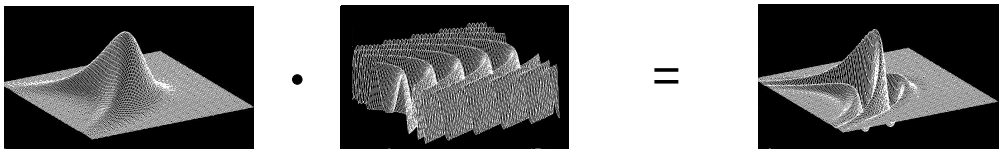

Figure 5: The real part of a banana wavelet is the product of a curved Gaussian $G^{\vec{b}}(x, y)$ and a curved wave function $F^{\vec{b}}(x, y)$.

- their usage as local line detectors (section 2.3), and
- their computation by hierarchical processing (section 2.4).

2.1 Gabor and Banana Wavelets

The basic features of the object recognition system are Gabor wavelets or a generalization of Gabor wavelets, called *banana wavelets*. A banana wavelet $B^{\vec{b}}$ is a complex valued function defined on $I\!R \times I\!R$. It is parameterized by a vector $\vec{b}$ of four variables $\vec{b} = (f, \alpha, c, s)$ expressing the attributes frequency (f), orientation (α), curvature (c) and elongation (s) (see figure 4 (right)). It can be understood as a product of a constant $\gamma^{\vec{b}}$ with a curved and rotated complex harmonic wave function $F^{\vec{b}}(x, y)$ and a stretched two–dimensional Gaussian $G^{\vec{b}}(x, y)$ bent and rotated according to $F^{\vec{b}}$ (see figure 5):

$$B^{\vec{b}}(x, y) = \gamma^{\vec{b}} \cdot G^{\vec{b}}(x, y) \cdot \left(F^{\vec{b}}(x, y) - DC^{\vec{b}} \right)$$

with

$$G^{\vec{b}}(x, y) = \exp\left(-\frac{f^2}{2} \left(\sigma_x^{-2} \left(x\cos\alpha + y\sin\alpha + c\,(-x\sin\alpha + y\cos\alpha)^2 \right)^2 + \right.\right.$$
$$\left.\left. \sigma_y^{-2} s^{-2} (-x\sin\alpha + y\cos\alpha)^2 \right) \right)$$

and

$$F^{\vec{b}}(x, y) = \exp\left(if \left(x\cos\alpha + y\sin\alpha + c\,(-x\sin\alpha + y\cos\alpha)^2 \right) \right).$$

To ensure that the kernels are DC–free, i.e., that the filter responses are independent from the mean grey value intensity, we set

$$DC^{\vec{b}} = \frac{\int G^{\vec{b}}(\vec{x}) F^{\vec{b}}(\vec{x}) d\vec{x}}{\int G^{\vec{b}}(\vec{x}) d\vec{x}} = e^{-\frac{\sigma_x}{2}}. \quad (1)$$

To compensate differences of filter responses of banana wavelets of different elongation it is set

$$\gamma^{\vec{b}} = \frac{\left(1 + \xi_s \frac{s_{max} - s}{s_{max}}\right)}{||B^{\vec{b}}||_2}$$

where $||\ ||_2$ represents the L^2 norm. $\gamma^{\vec{b}}$ ensures a more even distribution of the responses of the banana wavelets by intensifying responses for small elongation, ξ_s represents the factor by which the amplitude of a banana wavelet with certain elongation is modified. In Table 1 the value of ξ_s is shown as well as other parameter settings which are used for most of the simulations, in the following referred to as "standard settings".

2.1.1 Curve Corresponding to a Banana Wavelet

To each banana wavelet $B^{\vec{b}}$ there can be defined a corresponding curve. This curve allows the visualization of the learned representation of an object (see figure 1 or 6v). The curve corresponding to a banana wavelet represents a transition of continuous grey level feature (represented by a Gabor wavelet or banana wavelet response) to a discrete symbolic representation based on local line segments. Furthermore, the curve corresponding to a banana wavelet is used in section 2.4 to speed up feature processing by hierarchical processing.

More formally the corresponding curve $\vec{p}^{\,\vec{b}}(t)$ is defined as

$$\vec{p}^{\,\vec{b}}(t) = \begin{pmatrix} \cos(2\pi - \alpha)(-\frac{c}{f}(s\sigma_y t)^2) & + & \sin(2\pi - \alpha)(\frac{1}{f} s\sigma_y t) \\ -\sin(2\pi - \alpha)(-\frac{c}{f}(s\sigma_y t)^2) & + & \cos(2\pi - \alpha)(\frac{1}{f} s\sigma_y t) \end{pmatrix} \qquad t \in [-1, 1].$$

2.1.2 Gabor and Banana Wavelets used as Local Line Detectors

Banana wavelets are generalized Gabor wavelets (for Gabor wavelets see, e.g., [6]), they possess additionally to frequency and orientation the parameters curvature and elongation (see figure 4). The approach introduced here does not necessitate on the usage of banana wavelets but is also applicable with Gabor Wavelets. In section 6 we show that the usage of curvature is only one among a set of other important differences to the older system [21, 45]. A probably even more important distinctive feature is the usage of kernels as local

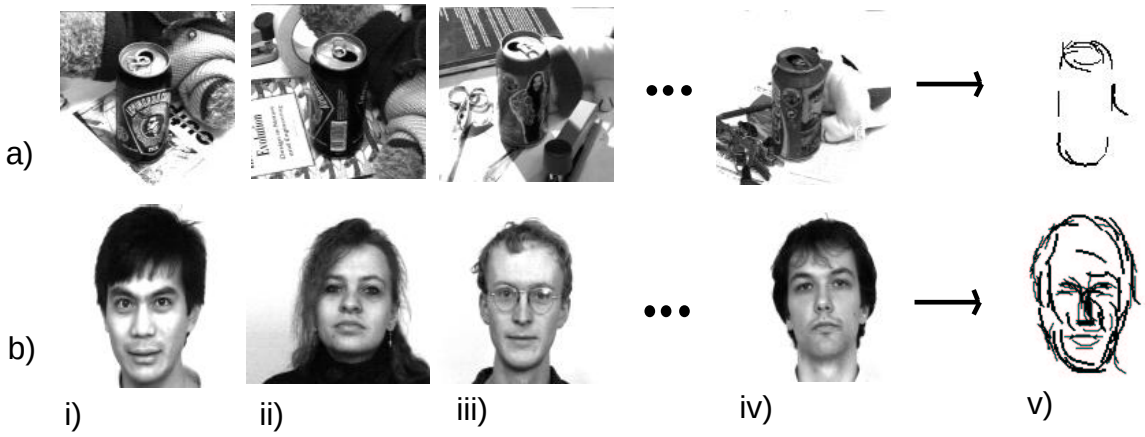

Figure 6: i–iv) Different examples of cans and faces used for learning. v) The learned representations.

line detectors within the object representation: In ORASSYLL objects are represented as a sparse and spatially ordered arrangement of local (curved) line segments as symbolic features. In this sense, Gabor wavelets and banana wavelets can be applied as local line detectors representing local oriented or curved local oriented lines, respectively.

The elongation parameter allows for representing smaller or larger line segments. Our colleague Michael Pötzsch showed that a higher elongation value s decreases the angle of intersection of lines which can be distinguished from the filter responses. The introduction of curvature allows for a smoother and sparser representation of objects.

With Gabors banana wavelets share important properties of wavelets, such as locality and reconstructability as well as the possibility to derive all filters from a mother wavelet by transformations such as translation, dilatation and rotation.

2.2 Neighborhood and Metric in the Feature Space

In this subsection we define two additional structures or relations between features, a neighborhood relation and a metric (PF2). The neighborhood relation is utilized for the feature extraction described in section 3.1 and the metric in the learning algorithm described in section 3.2.

Let I be a given picture and $I^{(x,y)}$ its value at pixel position (x, y). The discrete six–dimensional space of vectors $\vec{c} = (x, y, l, o, b, m)$ is called the *coordinate space* (referred

Standard Parameter Settings												
Transformation						Banana space			Learning			Matching		
n_l	=	3	$f_{\max}$	=	2π	e_x	=	4	τ	=	0.5	θ_1	=	-1.7
n_o	=	8	f_s	=	0.8	e_y	=	4	λ	=	2.5	θ_2	=	0.8
n_b	=	5	s_{min}	=	0.5	e_f	=	0.01	p_1	=	0.1			
n_m	=	2	s_{max}	=	1.0	e_α	=	0.3	p_2	=	0.7			
σ_x	=	1.0	$c_{\max}$	=	1.3	e_c	=	0.4	r_1	=	1.0			
σ_y	=	2.0				e_s	=	3.0	r_2	=	1.5			
ξ_s	=	0.45							R	=	9			

Table 1: Standard Settings. Columns 1,2: Parameters of transformation. Column 3: Metric of the banana space. Column 4: Parameters of learning. Column 5: Parameters for matching.

to as $\mathcal{C}$), where $\vec{c}$ represents the filter $B^{(f(l),\alpha(o),c(b),s(m))}$ at pixel position (x, y). The coordinate space has $n_l \cdot n_o \cdot n_b \cdot n_m \cdot x_{res} \cdot y_{res}$ elements, x_{res} and y_{res} representing the resolution of the image I. In the following a neighborhood relation $N(\vec{c}_1, \vec{c}_2)$ and a metric $d(\vec{c}_1, \vec{c}_2)$ is defined on $\mathcal{C}$. Two coordinates $\vec{c}_1, \vec{c}_2$ are expected to be neighbors (or have a small distance d) when their corresponding kernels are similar. For the coordinates pixel position (x, y), level l and size m it can be assumed that the similarity of corresponding kernels changes according to the distance of these parameters, i.e., the corresponding kernels can be thought to be arranged in a four–dimensional cube. For the coordinates orientation o and curvature b it is more convenient to arrange the corresponding kernels in a Moebius topology (see figure 7)[1].

More formally, two elements of the coordinate space $\vec{c}_1, \vec{c}_2$ are called 'neighboring' ($N(\vec{c}_1, \vec{c}_2)$=TRUE) when they are neighbors in the (x, y, l, o, b, m)–grid. Now a distance measure on $\mathcal{C}$ harmonizing with this neighborhood relation is defined. The mapping

$$\begin{aligned} E(\vec{c}) &= (x, y, f(l), \alpha(o), c(b), s(m)) \\ &= \left(x, y, f_{\max} f_s^{-l}, \frac{2\pi o}{n_o}, c_{\max} - \frac{2c_{\max} b}{n_b}, s_{min} + \frac{m(s_{max} - s_{min})}{n_s}\right) \end{aligned} \tag{2}$$

embeddes the discrete (x, y, l, o, b, m)–space $\mathcal{C}$ in the continuous (x, y, f, α, c, s)–space of all possible banana wavelets.

A distance measure is defined for the orientation–curvature subspace (α, c) expressing the Moebius topology thereof. Let $(e_x, e_y, e_f, e_\alpha, e_c, e_s)$ be a cube of volume 1 (the choice

[1]Note that a banana wavelet with orientation $\alpha(o)$ and curvature $c(b)$ rotated by π represents the same curve than a banana wavelet with orientation $\alpha(o)$ and curvature $-c(b)$

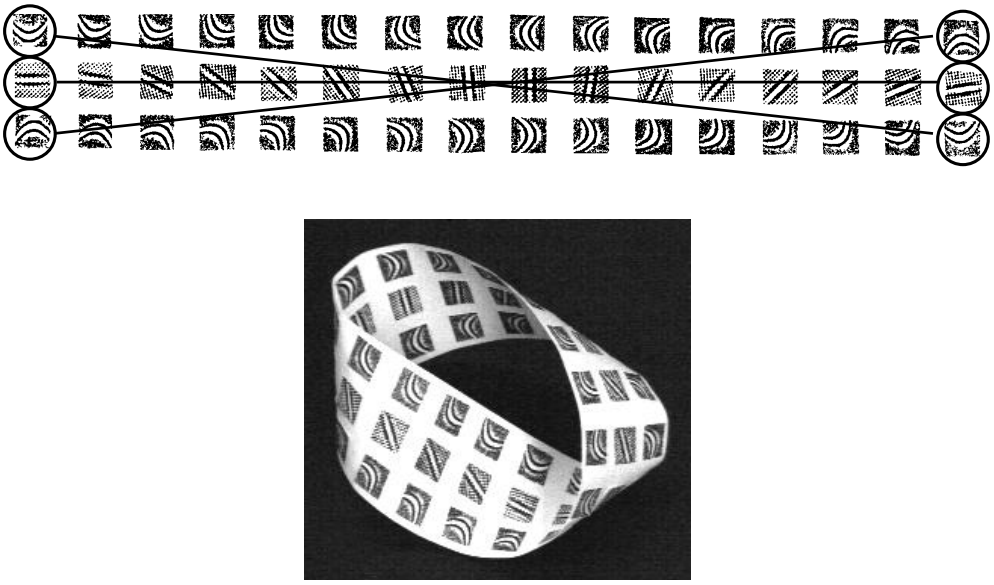

Figure 7: Moebius topology. The subspace of orientations and curvatures (o, b) with $n_o = 16$ orientations and $n_b = 3$ curvatures. Top: The banana wavelets on the left are connected by lines to the wavelets with neighboring indices (o, b) on the right. Connecting the right edge with the left edge according to these neighborhoods leads to the Moebius topology shown at the bottom.

---

of parameters are shown in table 1, column 3) in the feature space. Setting

$$= \min \left\{ \sqrt{\frac{d((\alpha_1, c_1), (\alpha_2, c_2))}{\frac{(\alpha_1-\alpha_2)^2}{e_\alpha^2} + \frac{(c_1-c_2)^2}{e_c^2}}}, \sqrt{\frac{((\alpha_1-\pi)-\alpha_2)^2}{e_\alpha^2} + \frac{(c_1+c_2)^2}{e_c^2}}, \sqrt{\frac{((\alpha_1+\pi)-\alpha_2)^2}{e_\alpha^2} + \frac{(c_1+c_2)^2}{e_c^2}} \right\} \quad (3)$$

on the subspace (α, c) a distance measure on the complete coordinate space is defined by

$$d(\vec{c}_1, \vec{c}_2) = \sqrt{\frac{(x_1 - x_2)^2}{e_x^2} + \frac{(y_1 - y_2)^2}{e_y^2} + \frac{(f_1 - f_2)^2}{e_f^2} + d((\alpha_1, c_1), (\alpha_2, c_2))^2 + \frac{(s_1 - s_2)^2}{e_s^2}}. \quad (4)$$

The parameters $(e_x, e_y, e_f, e_\alpha, e_c, e_s)$ determine the distances in each one-dimensional subspace. A smaller value indicates a stretching of this space.

2.3 Non–Linear Transformations of the Filter Responses

The feature processing of ORASSYLL consists of a two–step non–linear transformation of the complex filter responses. In a first step the magnitude of the filter response $B^{\vec{b}}$ is extracted after the convolution of $B^{\vec{b}}$ with the image I. Let

$$r(\vec{c}) = (\mathcal{A}I)\ \left(\vec{x}_0, \vec{b}\right) = \left| \int B^{\vec{b}}\left(\vec{x}_0 - \vec{x}\right) I\left(\vec{x}\right)\ d\vec{x} \right| = \left| \left(B^{\vec{b}} * I\right)\left(\vec{x}_0\right) \right|$$

be the magnitude of the filter response $B^{\vec{b}}$ at pixel position $\vec{x_0}$ in image I (or, in other words, the filter response corresponding to $\vec{c} = (\vec{x}_0, \vec{b})$). A filter $B^{\vec{b}}$ causes a strong response at pixel position $\vec{x_0}$ when the local structure of the image at that pixel position is similar to $B^{\vec{b}}$. In contrast to the complex filter response oscillating with phase, the magnitude of the response is more stable under slight variation of position [36].

The magnitude of the filter response depends significantly on the strength of edges in the image. However, here we are only interested in the presence and not in the strength of edges. Thus, in a second step a function $N(\ \ )$ normalizes the real valued filter responses $r(\vec{c})$ into the interval $[0, 1]$. The value $N\left(\vec{c}\right)$ represents the systems confidence of the presence or absence of a local line segment corresponding to $\vec{c} = (\vec{x_0}, \vec{b})$. This normalization is based on the "Above Average Criterion":

AAC a line segment corresponding to the wavelet $\vec{c}$ is present if the corresponding banana wavelet response is distinctly above the average response.

More formally, we define an average response by considering the average response in the complete feature space and also in a local area of the feature space. Therefore, a global and a local normalization are performed.

A mean total response is defined as $E^{local}(\vec{x}_0, f_o, I)$ for the f_o-th level at pixel position $\vec{x}_0$ and the mean total response for the f_o-th level $E^{total}(f_o)$ of the banana space by

$$E^{local}(\vec{x}_0, f_o, I) :=< r(\vec{x}, \vec{b}) >_{\left\{\vec{x} \in A(\vec{x}_0, r_E),\ \vec{b} \in \mathcal{B},\ l = f_o\right\}}$$

and

$$E^{total}(f_o) :=< r(\vec{x}, \vec{b}) >_{\left\{\vec{x} \in \mathcal{I},\ \vec{b} \in \mathcal{B},\ l = f_o\right\}}$$

where $A(\vec{x}_0, r_E)$ represents the cuboid square with center $\vec{x}_0$ and edge length r_E in the (x, y) space. $\mathcal{I}$ represents a set of arbitrary natural images and $\mathcal{B}$ is the full set of discrete banana wavelets at one pixel position. The average response $E(I, \vec{x}_0)$ is defined as

$$E(\vec{x}_0, f_o, I) := \frac{E^{total}(f_o) + E^{local}(\vec{x}_0, f_o, I)}{2}.$$

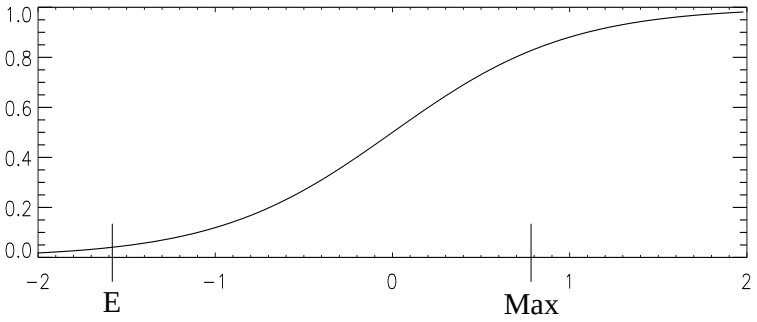

Figure 8: The normalization function starts to increase for values larger than the mean response $E(I, f_o, \vec{x}_0)$ and becomes almost flat for values higher than $\mathrm{Max}(I, f_o, \vec{x}_0)$.

The function $E(\vec{x}_0, f_o, I)$ has high values when there is structure in the local area around $\vec{x}_0$.[2]

Now the maximum response in an area of the feature space is defined by

$$\mathrm{Max}(\vec{x}_0, f_o, I) = \max_{\substack{\vec{b}\in\mathcal{B} \\ l=f_o}} \left(r(\vec{x_0}, \vec{b})\right).$$

The sigmoid function (see figure 8)

$$N(t; \vec{x}_0, f_o, I) = \frac{1}{2}\left(tanh\left(\frac{(\theta_2-\theta_1)}{\mathrm{Max}(\vec{x}_0, f_o, I) - E(\vec{x}_0, f_o, I)} \cdot t - \frac{\theta_1 - E(\vec{x}_0, f_o, I)(\theta_2-\theta_1)}{\mathrm{Max}(\vec{x}_0, f_o, I) - E(\vec{x}_0, f_o, I)}\right) + 1\right) \tag{5}$$

is the final normalization function. $E(\vec{x}_0, f_o, I)$ is mapped to θ_1 and $\mathrm{Max}(\vec{x}_0, f_o, I)$ is mapped to θ_2 before applying *tanh* (see also figure 8, here $\theta_1 = -1.7$ and $\theta_2 = 0.8$). The value $N(r(\vec{c}))$ represents the system's confidence of the presence of the feature $\vec{b}$ at position $\vec{x_0}$. According to the above average criterion, this confidence is high when the response exceeds the average activity significantly. The exact value of the response is not of interest. However, a range of indecision of the system when the response is only slightly above the average activity is still allowed to avoid a very strict decision at this stage.

[2]To reduce the time for calculating the average activities $E(\vec{x}_0, f_o, I)$, only the banana responses for the smallest size and with zero curvature are used for computation. The responses corresponding to banana wavelets with same orientation but different curvature or size are highly correlated because they represent similar features. For the computation of $E(I, f_o, \vec{x}_0)$, which just represents some kind of average activity, only one of these similar features has to be taken into account.

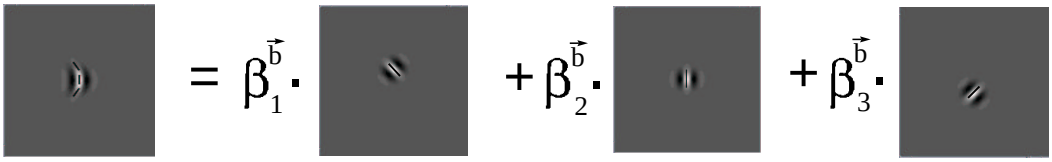

Figure 9: Approximation. The banana wavelet on the left is approximated by the weighted sum of Gabor wavelets on the right.

2.4 Approximation of Banana Wavelets by Gabor Wavelets

The banana response space contains a large number of features, their generation takes a long time on a sequential computer and requires large memory capacity. For instance, a transformation of a 128×128 image with the standard settings (as defined in table 1) takes approximately 21 seconds on a Sparc Ultra and requires 80 megabytes of main memory. In this subsection an algorithm is defined to approximate banana wavelets from a small set of Gabor wavelets and banana wavelet responses from Gabor wavelet responses. Thus banana wavelets are processed by hierarchical processing (PF3), choosing Gabor Wavelets as a first stage of processing. Figure 9 gives the idea of the approximation algorithm. The approximation can be performed before the matching (as described in section 4) or in a *virtual mode* in which only those features are evaluated "on the fly" which are actually requested for the matching. Because of the sparseness of the representations of objects only a small subset of the banana space is actually used during matching and can therefore be evaluated very quickly. In case that all banana wavelets are evaluated before matching we achieve a speed up of a factor 5 by the hierarchical processing. In the virtual mode memory requirements can be reduced by a factor 20. In [18] a precise definition of the approximation algorithm is given. The current approximation algorithm is based on the heuristic of local similarity of Gabors and banana wavelets, or, in other words, it is based on the fact that a curve can be approximated by a set of smaller line segments. Very good quality of approximation can be achieved with a small number of coefficients (for details see the appendix of [18]). An approximation approach based on steerable filters (see, e.g.,[9, 33]) may lead to even better approximation (and is probably more satisfactory from a mathematical point of view) and could be an interesting task for future research.

3 Learning

In this section we describe the representation of objects and its autonomous learning based on the a priori constraints PL2, PF2, PF4, PE1 and PE2. In subsection 3.1 a sparsification (PF4) of the image is defined. This sparsification reduces the transformed image (with the standard settings consisting of more than 5000000 real–valued features) to a small set of (less than 500) discrete features. In a second step (described in subsection 3.2) we will describe a learning algorithm (utilizing the constraints PE1 and PE2). The learning algorithm extracts an efficient representation of a certain view of an object class from a set of sparsified images making use of the metric in the feature space. Learning becomes autonomous by solving the correspondence problem (PL2) as described in subsection 3.3.

3.1 Extracting the Important Banana Responses per Instance

A further stage of preprocessing reduces the number of vectors $\vec{c}$ in the coordinate space $\mathcal{C}$ to represent a certain picture I or a local area of I. The aim is to extract the local structure in I in terms of local (curved) line segments corresponding to Gabor or banana wavelets. Some of these lines may be important to represent the specific object, but there will be also line segments representing features which are caused by accidental conditions, e.g., shadows caused by specific illumination, background or object surface texture (see figure 10bi-iv).

An *important feature* in one image (or "per instance") is defined by two properties C1 and C2. An *important feature per instance*

C1 causes a strong response,

C2 represents a local maximum within a local area of the feature space.

More formally, a banana wavelet $\vec{b}_0$ is said to have a "strong response" at a certain pixel position $\vec{x_0}$ when the response $r(\vec{x}_0, \vec{b}_0)$ exceeds a certain threshold. C1 and C2 can now be formalized as follows: A banana wavelet $\vec{c_0} = (\vec{x}_0, \vec{b}_0)$ represents a significant feature per instance if

C1': $r(\vec{x}_0, \vec{b}_0) > \lambda \cdot E(\vec{x}_0, f_o, I)$,

C2': $r(\vec{x}_0, \vec{b}_0) \geq r(\vec{x}_i, \vec{b}_i)$ within a neighborhood of $(\vec{x}_0, \vec{b}_0)$.

The parameter λ controls the distinctness a feature must exceed the average activity to be a candidate for a significant feature per instance. A larger value for λ reduces the number of significant features.

One–shot learning: By positioning a rectangular grid on a roughly segmented object (see figure 3a,i) in front of homogeneous background and extracting significant features per instance as described above suitable representations of objects can already be extracted. These representations are successfully applied to difficult discrimination tasks. Figure 10bi–iv) and 3b,d) show the significant features per instance represented by their corresponding line segments.

3.2 Learning of Object Representations in complex scenes

Now we describe an algorithm to extract invariant local features representing landmarks for a given class of objects. Here we assume the correspondence problem to be solved, i.e., assume the position of certain landmarks of an object, such as the center of left eye or the midpoint of the right edge of a can, to be known on pictures of different examples of this objects. In some of the simulations corresponding landmarks are determined by manual construction, for the rest manual intervention is replaced by motor controlled feedback (3.3). According to PL2, it is indispensable for learning to ensure that comparable entities are used as training data, otherwise the effect of learning will decrease because of the noise of the training data. Furthermore, it is advantageous to split a large learning problem (such as the learning of a representation of a face) into smaller subproblems (such as learning the representation of the eye region or the top of the head). This learning with comparable and smaller entities is the meaning of the constraints PL1 and PL2.

Briefly, the learning algorithm works as follows: The significant features per instance are extracted (as described in section 3.1) for different images of an object taken at a certain pose within an rectangular region surrounding the landmarks[3]. For each landmark all these features are collected into one bin. A certain feature is defined as significant when this feature or a similar feature (according to the metric (4)) occurs often in the bin, i.e., it occurs often in the different images of the training set. The result is a graph with its nodes labeled with elements of the banana coordinate space (or corresponding line segments) expressing the learned significant features (see, e.g., figure 10v) and its edges labeled by the spatial relations of the landmarks. It is referred to such a representation of an object class $\mathcal{O}$ as $\mathcal{S}^{\mathcal{O}}$ and to the set of elements of the coordinate space representing the k–th landmark as $\mathcal{S}_k^{\mathcal{O}}$.

A significant feature should be independent of background, illumination or accidental qualities of a certain example of the object class, i.e., it should be invariant under these transformations of an object class (PE1). This is realized by measuring the probability

[3]In the standard settings the width R of the rectangular region is 9.

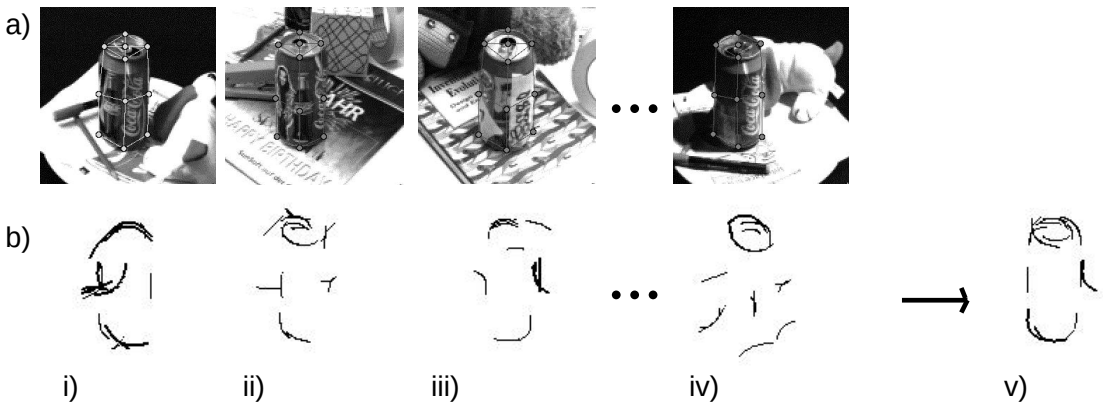

Figure 10: a: Pictures for training. bi–iv: Extracted significant features per instance. bv: the learned representation.

---

of occurrence of features in a local area of the banana space for different examples. The remaining features of the learning algorithm are those features which occur often in the training set. The metric allows the grouping of similar features into one bin, but it also allows the reduction of redundancy of information (PE2) by avoiding multiple similar features in the learned representation.

Formally speaking, let $\mathcal{I}$ be a set of pictures of different examples of a class of objects of certain orientation and approximately equal size. $I^{(j,k)}$ represents a local area in the j-th image in $\mathcal{I}$ with the k-th landmark as its center. Let $\vec{s}_{ij}^{\,k}$ be the i-th important feature per instance extracted in the area $I^{(j,k)}$ (see figure 11a, each data point represents one element $\vec{s}_{ij}^{\,k}$). All $\vec{s}_{ij}^{\,k}$ for a specific k are collected in one set S^k. Then the LBG–vector quantization algorithm [23] is applied to S^k. After vector quantization a codebook C^1 expresses the vectors $\vec{s}_{ij}^{\,k}$ with a constant number n_{C^1} of code book vectors $\vec{c}_i^{\,1} \in C^1 \subset \mathcal{C}, \vec{c}_i^{\,1} : 1, \ldots, n_{C^1}$ (figure 11b). n_{C^1} depends on the number of entries in S^k: $n_{C^1} = p_1|S^k|$, $0 < p_1 \leq 1$. In case of a large p_1 the initial code book has a higher density in the training set.

The LBG–algorithm reduces the distortion error, i.e., the average error occurring when all elements of S^k are substituted by the nearest codebook vector in C^1. In case of high densities of elements $\vec{s}_{ij}^{\,k}$ in S^k it may be advantageous in terms of the distortion error to have code book vectors $\vec{c}$ and $\vec{c}\,'$ with small distance $d(\vec{c}, \vec{c}\,')$. But the significant features for a certain class of objects are expected to express independent qualities (P2), i.e., they are expected to have large distances in the banana space. A smaller codebook

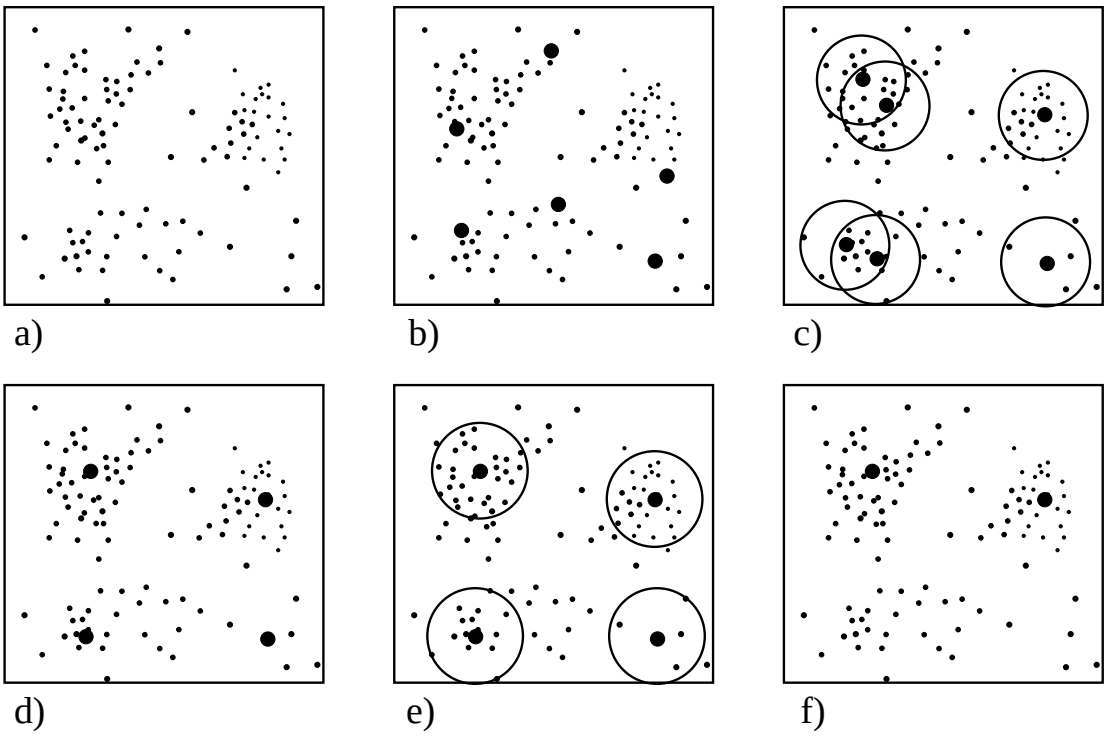

Figure 11: Clustering (detailed description see text): a) Distribution of data. b) Codebook Initialization. c) Codebook vectors after learning. d) Substituting sets of codebook vectors with small distance ($< r_1$) by their center of gravity. e) Counting number of elements within radius r_2. f) Deleting codebook vectors representing insignificant features.

C^2 is constructed in which the $\vec{c}, \vec{c}' \in C^1$ with close distances are replaced by their center of gravity: Let $r_1 \in I\!R^+$ be fixed. For all $\vec{c} \in C^1$ the number of $\vec{c}' \in C^1$ with distance $d(\vec{c}, \vec{c}') < r_1$ (figure 11c) is computed. If there exists at least one such $\vec{c}' \neq \vec{c}$ all the codebook vectors in C^1 with $d(\vec{c}, \vec{c}') < r_1$ are substituted by their center of gravity (figure 11d). C^2 now represents a code book with a lower or equal number of elements than C^1, with redundant codebook vectors being eliminated. Now the important features for the k-th landmark of a certain object can be defined as those codebook vectors $\vec{c} \in C^2$ for which a certain percentage p_2 of $\vec{s}_{ij}^{\,k}$ exists with $d(\vec{c}, \vec{s}_{ij}^{\,k}) < r_2$ (figure 11e,f). These important features are collected in a set $\mathcal{S}_k^{\mathcal{O}}$ which is our learned representation of the

k-th landmark of the given class of objects.

Compared to one–shot learning, learning over different examples leads to better representations, because different manifestations of 2D–views of objects are taken into account. This can be demonstrated for instance in the matching results for hand posture recognition, in which the representations extracted by one–shot learning achieve already good results (see table 3, row 3) on the easier test set (Set 1 without varying background and illumination), but significantly lower recognition rates on the more difficult Set 2 and Set 3. The matching with learned hand posture representations (row 1 and 2) achieves high performance on all sets. The same holds true for matching with face representations (see table 2).

3.3 Autonomous Learning

In figure 10 we defined the position of landmarks and their arrangement in a flexible grid are manually. To avoid the manual generation of ground truth we can either apply one–shot learning (see section 3.2) or make use of motor controlled feedback: By moving an object with a robot arm and following the object by keeping fixation relative to the robot hand using its known 3D position, we produce training data in which a certain view of an object is shown with varying background and illumination but with corresponding landmarks in the same pixel position within the image (see fig 2b,d). Now the flexible grid can be substituted by a rectangular grid roughly positioned on the object and the interaction of the camera and the motor controlled feedback ensures that landmarks are positioned at corresponding pixel position on the object (see figure 2) and the very same learning algorithm as described in section 3.2 for manually defined landmarks can be applied (see figure 2v for autonomously learned representations). In this way reliable representations can be learned even in complex scenes with varying background and illumination (see, e.g, figure 2). The positioning of the grid may be very rough and the grid can have large overlap with the background (see, e.g., figure 2b,i). Manual intervention is reduced to the determination of a rough rectangular area which covers the object at the beginning of the sequence. Compared to learning within the older system [46, 16, 20], in which for each view of an object a object–adapted topology for the graph has to be defined (and varying background could not be handled), this manual intervention is minimal.

4 Elastic Graph Matching with Sparse Object Representations

To apply the sparse representations for location and classification of objects a similarity between the extracted representation $\mathcal{S}^{\mathcal{O}}$ and a certain area in the image has to be defined. We would like to point out that, as in [21, 45]), for matching the complete feature space is computed. Therefore, sparseness is only a property of the stored object representation (i.e., a higher stage in the hierarchy of visual processing) and not of the feature space corresponding to the current image, i.e., the transformation of the image.

A key issue of the approach introduced here is the definition of a comparison of the large continuous–valued feature space with the discrete and binary object representation. This problem is almost solved by the normalization described in section 2.3 which mediates between these two different kinds of image respectively object representations.

In this section, the similarity function of a graph labeled with banana wavelets with certain size and position in an image is defined. For the comparison of the sparse object representation with a local area of the image it is made use of the robustness of the filter responses, see [36]. As in [45, 20] the object representation is stable up to a certain size variation, if this variation becomes too large more than one representation has to be used to cover different scales.

The robustness of the Gabor magnitude according to scale variation, translation and rotation in plane and depth is extensively discussed in [21, 36]. Roughly speaking, robustness to scale variation and variation is about 20% [36]. Additional robustness is achieved by the elasticity of the graph. In our simulations 3 graphs were sufficient to cover a size variation of up to 2 octaves (see results for face finding in section 5), which is also the case within the older system (see [45, 20]). A further possibility to improve robustness (which is not applied here) is the utilization of explicit transformations within the space of Gabor wavelet responses (for details see [36]). We also want to stress that high invariance is not always wanted but that for certain problems (e.g., for the control of a robot arm) exact positions of objects and the arm are important.

A *total similarity* expresses the system's confidence whether there is a certain object in an image I at a certain position and size. As in the former system *local similarities* (expressing the system's confidence whether a node of the graph represents a local feature) are averaged.

The complete graph matching process used in this paper proceeds in three steps. The matching procedure is performed for all graphs within the representation (e.g., graphs covering different sizes in face detection or different object classes as for the hand posture

recognition problem (see section 5)). The graph achieving the highest similarity determines the size and position of the objects within the image, while the positions of its nodes identify the landmarks.

In the first step the graph is shifted across the image while keeping its form rigid. We use steps of about 3–5 pixel in either direction for this rigid shift. For each position of the graph we calculate the total similarity of the new positioned graph to the original graph. The total similarity is just the average similarity over all local similarities. This global move procedure is able to position the graph on the object. The position which provides the highest similarity is the starting position for the second step which permits variation of the scale of the graph distortions. In the third step the nodes are shifted locally and independently in a small surroundings of their starting position. After this local move procedure the optimal position of the graph is found at the position which provides the highest total similarity.

The local similarity is defined as follows: For each learned feature in $\mathcal{S}_k^{\mathcal{O}}$ and pixel position in the image it is simply checked whether the corresponding normalized filter response in the image is high or low, i.e., the corresponding feature is present or absent. Because of the sparseness of the representation only a few of these checks have to be made, therefore the matching is fast. Because only the important features are used, the matching is efficient.

More formally, the local similarity $Sim(\mathcal{S}_k^{\mathcal{O}}, I^{(x,y)})$ between a node labeled with banana wavelet responses $\mathcal{S}_k^{\mathcal{O}}$ and a pixel position (x, y) in an image I is the average of the normalized filter responses corresponding to the k-th landmark (i.e., $\vec{c}_i = (x_i, y_i, f_i, \alpha_i, c_i, s_i) \in \mathcal{S}_k^{\mathcal{O}}$) in the image at the pixel position (x, y):

$$Sim(\mathcal{S}_k^{\mathcal{O}}, I^{\vec{x}_0}) = \frac{1}{|\mathcal{S}_k^{\mathcal{O}}|} \sum_{\vec{c}_i \in \mathcal{S}_k^{\mathcal{O}}} N\left(r(x_0 - x_i, y_0 - y_i, f_i, \alpha_i, c_i, s_i)\right), \tag{6}$$

where $|\mathcal{S}_k^{\mathcal{O}}|$ is the number of local line segments the k–th node is labeled with.

As in [21, 45, 20] the total similarity $Sim(\mathcal{S}^{\mathcal{O}}, I)$ between a graph $\mathcal{S}^{\mathcal{O}}$ at position (x, y) with size s and the image I is simply defined as the average of the local similarities defined above:

$$Sim(\mathcal{S}^{\mathcal{O}}, I) = \frac{1}{n} \sum_{k=1}^{n} Sim(\mathcal{S}_k^{\mathcal{O}}, I^{\vec{x}}),$$

with n represents the number of nodes of the graph.

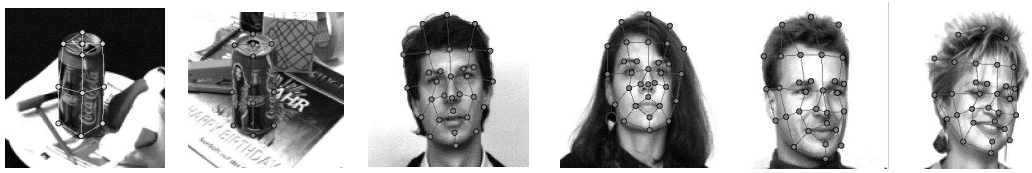

Figure 12: Manual defined graphs for a) cans, b) frontal faces and c) half profiles.

5 Simulations

In this section we demonstrate the applicability of ORASSYLL for a wide range of problems. Firstly, we learn representations of cans, faces of different poses, hand postures and different toys (section 5.1). Then we apply some of these representations to the problem of localizing these objects in complex scenes using the matching algorithm described in section 4. Additional simulations are performed in [18] and [24].

5.1 Learning of Representation

The learning algorithm described in section 3.2 will be applied to data consisting of manually provided and automatically generated landmarks.

5.1.1 Learning with Manually Provided Ground Truth

If not stated differently the training sets consist of a set of approximately 60 examples of an object viewed in a certain pose. Here, corresponding landmarks are defined manually on the different representatives of a class of objects (see figure 12).

Figure 10bi-iv) shows the significant features per instance for some of the can examples in the training set. Note the high amount of local line segments caused by texture or background (in the following called *structured noise*). In the learned representations (figure 10v) the amount of structured noise is reduced significantly. Figure 13 shows the learned representations for faces using manual defined graphs as shown in figure 12. Note that even differences between males and females can be represented and learned within ORASSYLL (see figure 13, second and third row). Figure 14 shows learned representation for ten hand postures.

With the standard settings of table 1 the transformation (without the approximation described in section 2.4) of a 128x128 picture needs 21 seconds, the extraction of significant

Figure 13: Training Set and Learned Representation. Top: half profile faces. Middle: female faces. Bottom: male faces. Note that even the fine differences between male and female faces can be expressed by curved line segments corresponding to banana wavelets.

features per instance takes approximately 0.7 seconds per node and picture and the final learning as described in section 3.2 takes 0.5 seconds for each landmark for a training set of 60 examples. All simulations were performed on a Sun UltraSparc (167MHz).

5.1.2 Learning with Automatically Generated Ground Truth

To avoid the manual generation of ground truth we make use of different strategies. The aim is the construction of training data in which a certain object is shown under changing conditions such as different background and different illumination but with only slight variation of the position of the landmarks. In these cases the learning algorithm can be applied to these pictures using a rectangular grid placed on the object (see figure 15b).

By moving an object, e.g., a toy car, by a robot and following the object with a camera

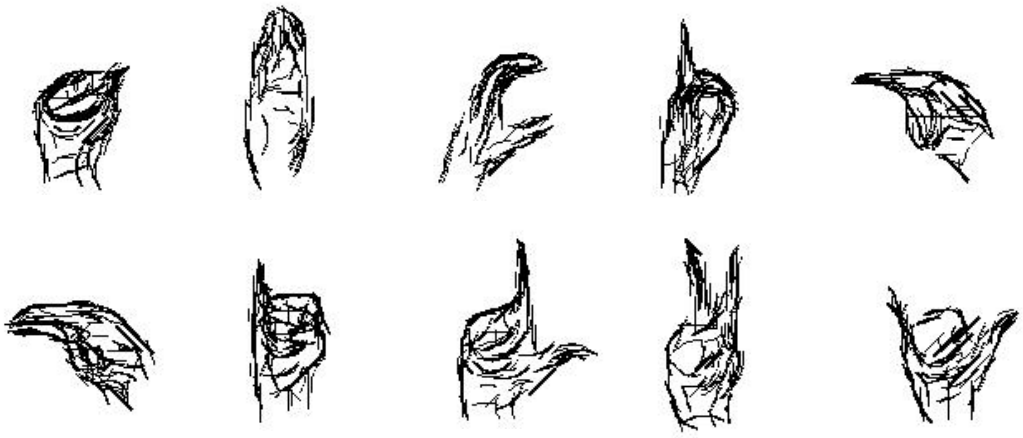

Figure 14: Learned representations of 10 different hand postures. The manually provided ground truth consists of 6 pictures per hand posture with a grid consisting of approximately 40 landmarks is placed.

---

utilizing the knowledge of the 3D position of the robot's hand a huge amount of ground truth can be generated for each object which can be moved by a robot arm (see figure 15a and figure 2). For the learning of a representation of cans the can is put on a rotating plate and background and lighting conditions are changed (see figure 15b). Figure 16 shows learned representations for ten hand postures. For each posture the ground truth consists of a sequence of 20 pictures of the hand posture created by one person in a surrounding with varying illumination. Small variation of the posture is produced by small movements of the hand of the person.

For the generation of ground truth for frontal faces a sequence of pictures were produced in which six persons are sitting fixed on a chair such that the position of eyes and nose of the different persons is approximately identical. Illumination and background are changed as for cans. Furthermore the people change their expression. To extract representations for different scales the learning algorithm is applied to the very same pictures scaled accordingly (figure 17 shows examples of face representations of different scale matched to different images).

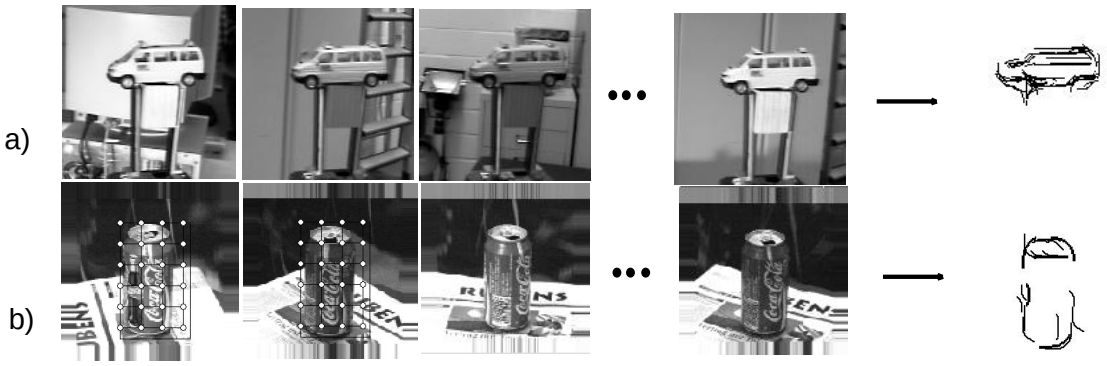

Figure 15: Automatic generation of ground truth for cars and cans and learned representation.

Matching Results for Face Finding						
	Repres.		Transformation		Performance	
	nb. reps	rep	approx	sec.	sec. match	Recog.
1)	3	standard	no approx.	10.7	2.2	77 %
2)	3	standard	approx.	3.3	2.2	77 %
3)	3	standard	virtual	1.5	7.1	77 %
4)	3	no curvature		1.5	2.1	73 %
5)	3	one instance	approx.	3.3	2.9	63 %
6)	3	bunch graph		1.1 – 5.4	3.5 – 98	35 – 54 %

Table 2: Matching results for face finding (for interpretation see text).

5.2 Matching

Table 2 and 3 show the results for two matching tasks, the localization of faces and hand postures. For both tasks matching within the approach described in this chapter is compared to the matching with bunch graphs as described in [46, 20, 43].

The extremely difficult face test set contains 120 frontal faces with uncontrolled illumination and mostly inhomogeneous background. Size variation of the faces is between 15 and 100 pixel (Figure 17 shows some examples of matches and mismatches on this data set). The first row gives the results for a matching with 3 representation of different scale. The transformation is not approximated and computation requires 10.7 seconds.

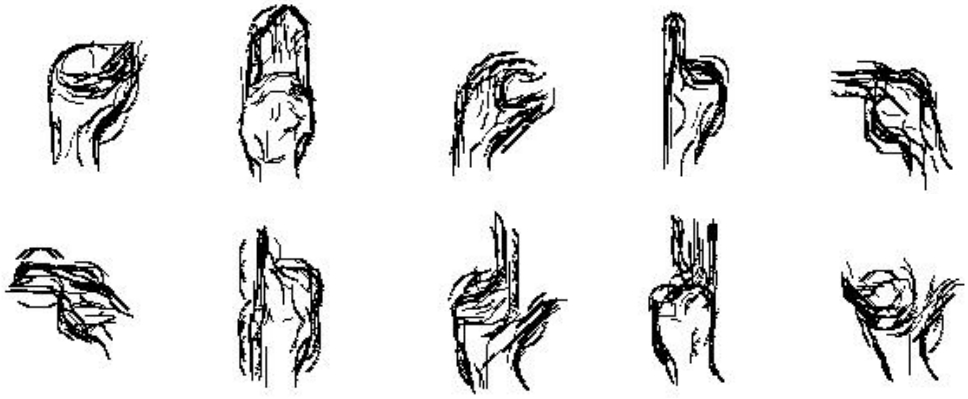

Figure 16: Representations of hand postures learned with automatic generated ground truth

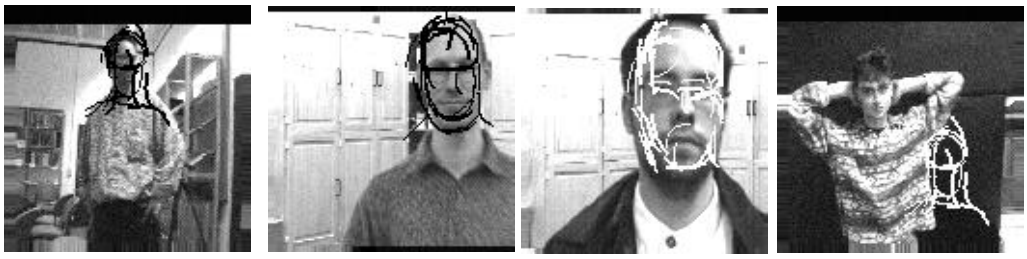

Figure 17: Face finding with autonomously learned representations for three scales. The mismatch (right) is caused by the person's unusual arm position.

Matching with three representations takes 2.2 seconds and faces were found correctly for 77% of the images. The second row gives the results when the transformation is approximated as described in section 2.4. Recognition rate is unchanged but feature generation requires only 3.3 seconds instead of 10.7 seconds[4]. The third row shows the results in case of approximation in the virtual mode. The transformation only requires 1.1 seconds (only the Gabor transformation has to be performed) but matching time increases significantly to 7.1 seconds because the the banana wavelet responses have to be computed "on the

[4]Note that for hand posture recognition a slight decrease of performance in case of approximation occurs.

Matching Results for Hand Posture Recognition							
	Repres.	Trafo		Performance			
	rep	approx	sec.	sec. match	Set 1	Set 2	Set 3
1)		no approx	17.0	9.5	93	73	90
2)		approx	4.9	9.5	93	72	80
3)	one instance	approx	4.9	12.4	80	52	52
4)	bunch graph		0.9	18.0	93	76	65

Table 3: Matching results for hand gesture recognition (for interpretation see text).

fly". In row 4 only non–curved kernels are used: only a slight decrease of performance can be achieved[5]. The simulations corresponding to the fifth row were performed with representations extracted from only one image. Performance decreases to 63%. The performance with the bunch graph approach as described in [46, 20] is given in the sixth row. We have tried different settings for the number of frequencies and orientation. For the best setting recognition rate was 54%.

The test sets of hand postures contain images of 10 different gestures (as shown in figure 14) in front of homogeneous background with controlled illumination (Set 1), inhomogeneous background with controlled illumination (Set 2), and inhomogeneous background with varying illumination (Set 3)[6]. There was only slight size variation, therefore one representation for each hand posture was sufficient to cover the size variation. The first row gives the results for a matching with the standard settings. The transformation is not approximated and computed in 17.0 seconds. Matching with ten representations takes 9.5 seconds and recognition rate was 93% (set 1), 73% (set 2) and 90% (set 3). The second row gives the results when the transformation is approximated as described in section 2.4. Recognition rate is slightly changing, in case of set 3 even significantly. Feature generation requires only 4.9 seconds instead of 17 seconds. The simulations corresponding to the third row were performed with representations extracted from only one image. Performance decreases to 80% (set 1), 52% (set 2) and 52% (set 3). The performance with the bunch graph approach as described in [43] is given in the fourth row. For test set 1 and 2 performance is comparable to ORASSYLL (in case of set 2 even slightly better). For set 3 performance is significantly worse compared to ORASSYLL.

[5]The role of curvature is discussed in more detail in section 6.

[6]In set 1 and set 2 the pictures were taken from different individuals. In set 3 a sequence of 20 pictures of each pose of one individual were recorded. This person slightly changed position and appearance of the hand posture while background and illumination was varying.

In [18] simulations with other objects are performed to investigate the influence of variation of background and illumination within the bunch graph approach and ORASSYLL. In [24] face recognition with binarized banana wavelets was performed on a very large data set (more than 700 pictures) with size variation of faces between 40 and 60 pixel, inhomogeneous background and uncontrolled illumination. For this set performance was 95%.

6 Comparison with Jet–based Systems

ORASSYLL has been heavily influenced by an older and well known vision system [21, 46, 20, 43], and has been equally influenced by Biederman's criticism of this older system [3]. The system [21, 46] was successfully applied to face recognition. High correlation between the system's and human's face recognition performance has been shown [3, 12]. However, Biederman and his associates [8, 3] also have shown that the system [21, 46] has only low correlation to human object recognition, indicating significant differences between object and face recognition.

We present a short description of the system [21, 46] — in the following called former or older system — in section 6.1. In section 6.2 differences between the older object recognition system and ORASSYLL are discussed and problems with the application of some of the basic entities of the older system (i.e., jets and bunches of jets) for object recognition are stressed. We argue that binarized Gabor or banana wavelets are a more suitable feature for this purpose. In this sense, a supplement to Biederman's arguments (which is merely based on psychophysical experiments) in terms of functional or algorithmic reasons is given.

6.1 Jets and Bunch Graphs

As models for objects the older system also employs labeled graphs. The edges of graphs are labeled with distance vectors between node positions. Nodes are labeled with jets [21] or bunches of jets [46], respectively. In a bunch of jets each jet is derived from the image of a different example of the view of an object. A bunch is thus covering a variety of forms a single landmark may take. This structure is called *bunch graph* [46].

Jets are derived from a set of linear filter operations in the form of convolutions of the image with a set of Gabor wavelets, see e.g. [6], of different wavelength and orientation (see figure 18a). A jet is formed by the set of complex values rendered by all wavelets centered at a given position of the image (see figure 18b). Due to the spatial extent of the

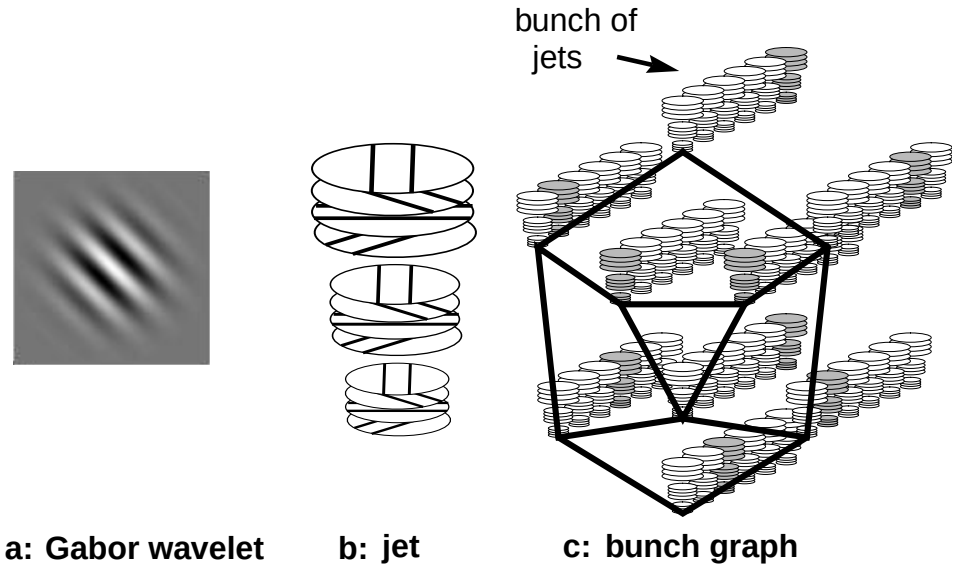

Figure 18: Representation of objects: a) a Gabor wavelet (real part). b) a jet calculated as a set of Gabor wavelets (the discs symbolize the different frequencies and directions of $\vec{k}$). c) a bunch graph.

wavelets, jets describe a local area around their position. A bunch $\mathcal{B}$ of jets taken at the same landmark (that is, at corresponding positions) of different examples of a certain view of an object class forms a generalized representation of this landmark (see figure 18c).

A bunch graph for a given view of an object class is created by placing an appropriate graph over a certain number of example images, adjusting the position of each node manually to the correct position of its landmark and letting the system extract the jet at that position[7]. All jets for a given landmark are attached as a bunch to that node. For each landmark, node positions (measured relatively to the center of gravity of their graph) are averaged, and the distance vectors between these average positions are stored as edge labels. One such bunch graph represents objects at a certain pose and size (see figure 18c).

Jet components a_j (the index j standing for length and orientation of the components'

[7]The time consuming procedure of manually positioning of landmarks can be facilitated by a semi–automatical procedure: A smaller, manually generated representation is used to place the graphs automatically and these automatically positioned graphs are then checked and corrected manually.

wave vectors) are the magnitude (which is slowly varying with position) of Gabor wavelet reponses. The similarity between two jets $\mathcal{J}$ and $\mathcal{J}'$ is defined as the normalized scalar product of the two jets:

$$S(\mathcal{J},\mathcal{J}') = \frac{1}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}} \cdot \sum_j a_j a_j' \tag{7}$$

As a node is actually labeled with a bunch of jets, a bunch similarity $S(\mathcal{B},\mathcal{J})$ to an image jet $\mathcal{J}$ is defined by the maximum similarity of the image jet to all jets of the bunch:

$$S(\mathcal{B},\mathcal{J}) = \max_i \{S(\mathcal{B}_i,\mathcal{J})\}\,.$$

As in the ORASSYLL the average over all node–similarities (as a global similarity) is optimized by shifting, scaling and distorting the graph during matching.

6.2 Conceptional Differences of Object Representations in the Former System and ORASSYLL

The object representation on ORASSYLL shows six conceptional (D1–D6) differences to the representation based on jets and bunches of jets. Here, in addition to the quantitative comparisons in section 5, we discuss how these differences influence recognition and learning. We would like to remark that the differences (D2–D6) are also valid for object representations within ORASSYLL based only on binarized Gabor wavelets. In ORASSYLL

D1 curvature can be explicitly used as a feature, allowing for a sparser and smoother representation of objects, a better recognition performance and an easier coding of Gestalt relations.

D2 a restriction to a specific set of symbolic features (local (curved) line segments) is imposed for object representation enabling learning and coding of objects by their essential features.

D3 object representations are sparse, allowing for fast and efficient matching.

D4 the metric (4) is utilized as an additional structure of the feature space which enables grouping similar features together. In this way autonomous learning of object representations in complex scenes becomes possible.

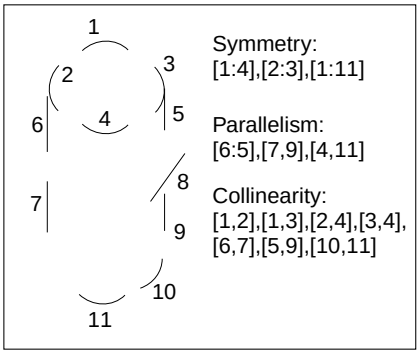

Figure 19: Sparse representation of a can with local curved lines corresponding to banana wavelets and lists of second order Gestalt relations between the local line segments (schematic).

D5 the local similarity (6) expresses the presence of a local symbolic feature but in a jet significant and insignificant features are lumped together. Therefore the similarity (6) is more robust against variation of background and illumination compared to the jet–similarity (7).

D6 only potentially interersting features are coded allowing for efficient one–shot learning.

D7 manual intervention is substituted by almost autonomous learning.

D1: Banana wavelets are generalized Gabor wavelets; curvature and elongation are added to the parameters frequency and orientation. The distinction curvature vs. straightness is a non–accidental feature in Biederman's sense, i.e., it describes a non–accidental property of the visual world: A straight, respectively curved, line in an image will usually result from a straight, respectively curved, edge in the world, therefore it is an important feature for the coding of objects and their discrimination. Furthermore, a line drawing with elongated curved local lines is smoother and also requires fewer line segments compared to a line drawing with shorter straight lines.

To emphasize the impact of utilizing curvature (D1) we would like to remark that a curved and elongated local line represents a feature of higher complexity compared to a short straight local line. The Gestalt principles collinearity ([1,2],[1,3],[2,4],[4,3],[10,11]), symmetry ([1,4],[2,3],[1,11]) or parallelism ([4,11]) in figure 19 could not be coded as second order relations of non curved lines.

D2: A jet represents the whole local image patch transformed to the Gabor space. As a consequence the image is *reconstructable* from jets [48, 37]. A complete reconstruction of the original image *can not* be regained from a representation with binarized Gabor or banana wavelets. Our reproduction of an object by local (curved) lines gives a restricted representation of the object by neglecting specifics of local patches such as the strength of edges or texture. However, this restricted representation is recognizable for humans, therefore seems to contain — despite the enormous reduction of data — relevant features used in the human visual system.

We argue that the serious restriction to local (curved) line segments, despite the indisputable loss of information (revealed in the unreconstractability) is advantageous and necessary for learning: The restricted receptiveness of the object recognition system facilitates the perception of important features and feature relations (see figure 19). As an additional evidence for the restriction to local (curved) lines, we argue that humans are easily able to give a description of a scene or an object as a simplified line drawing.

D3: The object representation described in ORASSYLL is essentially sparse, only few binary features taken from a large feature space are used to represent objects. In the bunch graph approach a large collection of continuous–valued vectors (jets), each representing an example of a local image patch, are used for object coding. In both approaches, matching time and memory requirements scale linearly with the amount of data stored in the representation of objects. The bunch graph approach, in which a whole bunch of manifestations is stored, requires much more memory capacity and matching time. For the representation of faces it is shown in [18] that the required memory can be reduced by a factor on the order of thousand.

D4: The metric (4) reflects the similarity or dissimilarity of kernels, measuring differences in the properties location, orientation and curvature, and allows to group similar features together while keeping different features separately. Learning allows to distinguish between significant features (e.g., the curved horizontal line of the top of the head) and insignificant features (e.g., corresponding to the background) and to keep only the significant features within the object representation (see figure 20). In this way the manual intervention necessary within the older system is substituted by autonomous learning.

D5: In a jet, significant and insignificant features are lumped together. Even when a single Gabor wavelet response gives information about the occurrence of a local line with a certain orientation, a jet always represents the whole local image patch. The jet similarity (7) reflects the relative strengths of a complete set of Gabor wavelet responses at the actual pixel position, and therefore reflects the fit to a whole local region. For example, a local area of an object may have an edge with a certain orientation resulting

Figure 20: Learning of the top of the head from varying background with binarized banana wavelets (schematic).

in a strong response of the corresponding Gabor wavelet. The occurrence of an edge with different orientation in the background causes a strong response for the Gabor wavelet with different orientation. Because the denominator in equation (7) increases by the "background–response", the relative strength of Gabor wavelet responses, and therefore the similarity (7) changes. However, for face discrimination the relative strength of filter responses probably is a useful feature capturing important aspects of face surfaces.

The similarity of a binarized banana wavelet to a local image patch indicates the presence or absence of the learned feature fairly independently of background and illumination and allows for a comparison of only the learned and significant features to the image. In [18] it is demonstrated that jet similarity (7) is less robust under variation of illumination and background compared to the local similarity (6) within ORASSYLL.

D6: The criterion C1 (section 3) ensures that a node of a graph is labeled with a feature only when there occurs relevant structure within the training image. Within the older jet–based systems features are extracted at each node without checking its relevance. One–shot learning (as demonstrated in subsection 3.2) in which a rectangular grid is placed on a roughly presegmented object in front of homogeneous background is more difficult within the jet–approach because each node of a jet–grid is *always* labeled by a jet, even when the node corresponds to the homogenous background or untextured surface of the object.

D7: Within the older system an object adapted graph has to be defined for each size and object class. Furthermore, to create a bunch graph this graph had to be positioned by manual control on a set of approximately 50 individuals. This procedure took approximately 4 to 8 hours for each object class. Within ORASSYLL all we have to do is to cover the object with a rectangular grid. By one–shot learning an object representation can be extracted with which we already achieve high recognition rates (see section 5). By utilizing the interaction of camera and robot (see figure 2) and also presupposing only a very rough covering of the object with a rectangular grid, we are able to extract effi-

cient representations even in complex scenes. This would be impossible within the older representation as pointed out in item D4.

7 Comparison with other Object Recognition Systems

Object recognition systems utilize different amount of *a priori* knowledge. At one extreme, there exist systems which apply learning algorithms directly to grey–level pictures. The algorithms can be called "neural" such as back-propagation or RBF–Networks, e.g., [41], or strategies of classical pattern recognition like Bayesian estimation methods, e.g., [10]. These systems apply a very small amount of constraints. The lack of *a priori* knowledge makes them applicable to any kind of problem, let it be the prediction of time series, speech recognition or vision, but they pay for this generality with bad generalization properties and unrealistic learning time. In other words, those systems fall into the trap of the variance problem [11].

As an extreme on the other side of the bias/variance dilemma there exist a large variety of systems putting a huge amount of structural knowledge into their system. As only one example in [14] football players are tracked. As *a priori* knowledge the structure of the background, i.e., the football field with its strict regulated lines and signs, is explicitly used. It is unthinkable to use such systems in other surroundings.

We assume these extremes are non–realistic attempts to build an efficient object recognition system because they are either caught in the bias or, in the variance trap. In the following we will compare ORASSYLL with some other attempts.

Cootes et. al. [4] introduce an object recognition system which is also based on line segments. The line segments are not as local as in our approach but they describe larger regions, e.g., the contour of the face from the left ear down to the chin up to the right ear. These representation of objects have to be defined manually. A similarity between this and our system is the restriction to local lines to describe objects. As an advantage of ORASSYLL, we regard the locality and metric organization of features which enable *autonomous* learning of representations of objects.

Zerroug and Nevatia [50] designed a system concerned with shape from shading. As *a priori* knowledge they assume a very complex model of 3D object representation. For certain well defined object classes (straight homogeneous generalized cylinders (SHGCs)) they are able to extract a 3D representation from 2D images. In its current state, ORASSYLL is only concerned with 2D views of objects which makes the comparison of both systems difficult. However, as a fundamental design difference I would like to point to the *openness* of ORASSYLL. It can deal with any kind of object which is representable as an

arrangement of local (curved) line segments and is not restricted to specific subclasses. In future research, we intend to *learn* higher object regularities within ORASSYLL which are presupposed within [50].

Hummel and Biederman [13] introduced an object recognition system based on Biederman's geon theory. In this system geons are inherently part of the system. In [19] the perspective of *learning* structures of geon–like complexity within ORASSYLL is discussed.

In [44, 27, 28] object recognition systems are introduced which are based on principle component analysis (PCA) methods applied to the grey level picture. PCA leads to a fast *reduction* of data by a *linear* transformation. We would like to remark, that from a biological point of view, in the human visual system there are no hints for data compression but a lot of hints for a data spreading in the first stages of visual processing [31]. A problem of PCA–methods is the restriction to linearity of transformations (for a discussion of this problem and some attempts to deal with it see [7].) Within ORASSYLL non–linear transformations (e.g., equation (5) and the criteria C1 and C2 in section 3) play an important role. In [17] it has been shown that these non–linearities lead to significant differences.

The lack of locality of features in PCA methods leads to sensitivity to varying background, partial occlusion and clutter. ORASSYLL shows high robustness to background variation as demonstrated in the simulations in section 6.2 and [18]. The features used within ORASSYLL are local in space (PL1) which makes the representation robust against at least partial occlusion (see also, [47]). Because the optimized total similarity is an average over local similarities local changes do not influence the total similarity very much. Furthermore, the similarity is rather independent in the quality orientation at a fixed position: For instance, a background edge with different orientation than the edge within the learned representation does not influence the simlilarity much.

We see our system not as a final stage, but as a basis for an more elaborated system which possesses an additional level of representation in which local features are grouped to more complex features. For this grouping process we find it important to organize the object representation of lower levels in a structured form, or more specifically, to equip the representation with meaningful features such as the qualities position, curvature and orientation. This enables the definition of relations such as collinearity, parallelism or symmetry (see figure 19). We see this as an important difference to the above–mentioned PCA–based methods, neural network based systems (such as, e.g. [38]) or Bayesian methods (such as [34, 27]). In this kind of systems the interpretation of lower and intermediate stages of representation becomes difficult.

Histogram methods such as [26, 39, 42] can take advantage of the power of multiple

cues and have the ability of fast image processing and recognition. For instance [26] applies, in addition to Gabor wavelets, cues such as color, vertices, blobs and contours.

In contrast to the histogram approaches [26, 39, 42] ORASSYLL deals with two tasks at the same time: localization and discrimination. For some tasks (e.g., when grasping is involved) localization of objects is important. Furthermore, ORASSYLL showed high robustness against changes of background, which is difficult within the histogram approaches. Even learning within these difficult situations is possible with our system.

In its current state ORASSYLL uses local (curved) line detectors for its object representation only. Current and future research addresses the integration of additional cues such as color and texture. In contrast to histogram methods ORASSYLL has a highly structured internal representation on the expense of slower processing. We intend even to increase this internal structure, e.g. by utilizing relations such as collinearity or parallelism (see [25, 17]) in which we see a great potential for improvement.

In this context a more efficient organization of the object data base (e.g. by indexing (see, e.g., [2]) or hierarchical organisation (see, e.g., [30])) can lead to improvement of matching speed. Within ORASSYLL the comparison with the data base is still a linear search, therefore in case of the ten class discrimination and localization problem of hand posture recognition the graph matching time is ten times higher than in case of localizing one posture only. Sparse coding of objects and the approximation algorithm already lead to a fast matching which could be further improved by more efficient search strategies.

As mentioned above, our symbolic representation of objects allows for application of relations such as collinearity and parallelism. Another example, which is essential for our learning algorithm, is the metric (4). There exist a variety of other systems making use of iconic representations (see, e.g., [5, 40, 29, 1, 34, 32]). In contrast to most of these icon–based systems our icons (or symbols) have a parametrized description and symbolic meaning which allows for the definition of such relations and also allows for the reconstruction of objects in an extremly sparse way.

In the object recognition system [1] an object is coded by icons consisting of high–dimensional vectors obtained from the reponses of Gaussian derivative spatial filters. This representation has some similarity to the Jet–based system [46] discussed extensively in section 6.

In [40] saliency map graphs are applied. Their icons are peaks of saliency maps which are stored on a graph the nodes which are arranged due to the scale level in a hierarchical fashion. A multiresolution representation resembling in some aspects to [40] was introduced in [5] in which objects also are represented by graphs with its nodes labeled by peaks of an energy surface. In contrast to [40] in [5] a scheme for grouping features is

given.

In [29] a representation is applied which is based on spatially loosely connected boundary fragments resembling cubist drawings. Their reconstruction of objects has some similarity to the one introduced here, although no learning is applied.

In detail very different from our approach, in [34] an object learning method is discussed which addresses some aspects of the learning problem in a line of thinking similar to our work. Learning, as it is formulated in [34] in a quite abstract way merely based on probabilty distributions, is thought to realize similar aspects such as the *a priori* constraints E1 and E2.

Beside sparseness and autonomous learning we see the meaningfulness of our features and the ability to reconstruct objects with those features as important differences to the above–mentioned icon–based systems.

A very interesting work about the visualization of faces is presented in [32]. Pearson describes an algorithm to reduce face representation to black/white images for fast data transfer. Although ORASSYLL is not primarily thought for the data compressing task and we do not claim that our representation is able to represent individual faces but rather the object class 'faces', we remark, that our symbolic representation of faces may allow for an even greater reduction (only about 50 symbols or icons are necessary to represent a face) and the application to efficient data transfer might be an interesting application of our representation.

8 Outlook

The introduced object recognition system is founded on reflections about the structure and the necessary amount of *a priori* knowledge such a system might require. By applying this knowledge representations of objects can be learned autonomously in difficult learning situations. These representations can be successfully applied to difficult discrimination tasks. ORASSYLL has shown its superiority as an object recognition system to the well established system [21, 46].

Autonomous learning became possible by interaction of action and perception: the correspondence problem was solved by shifting attention depending on the shifted object. An important extension of this idea is the treatment of general rigid body motion, i.e. consideration of rotation in addition to translation. Including the knowledge about the change of features under this general motion potentially allows for learning in more general situations. A further important extension (on which my colleague Michael Pötzsch [35] is working on) is the combination of matching and learning. He intends to combine a

successfull match with an already extracted representation (e.g., by one–shot learning) with a following learning iteration.

Two important issues of the object recognition problem are not addressed in the current work: Firstly, a full 3D–representation of objects (as done e.g., in [28, 34, 13]). In its current state ORASSYLL applies view–based representations which are robust up to a certain degree to scale variation and rotation because of the robustness of Gabors and the elasticity of the graph. One possible way of dealing with the full 3D–problem is a representation of objects by a set of views (see, e.g., [20]).

A second important issue is a full utilization of the potential of active vision (as done e.g., in [1]). The successful interaction of attention and arm movement in our learning algorithm already gives an example for this potential. Beside the introduction of an additional layer of abstraction by grouping our features to more complex entities and the integration of additional cues (such as color and texture) we see these tasks as important challenges for a system which comes closer to human performance. We argued that ORASSYLL can be a suitable intermediate stage for such a system.

We believe that the system presented here is not a dead end, but will also be a good basis for further improvement. Beside the issues addressed above the integration of other cues such as disparity information, movement and colour to support form processing at a higher stage of object representation will be a important tasks for future research.

Acknowledgement: We would like to thank Christoph von der Malsburg, Laurenz Wiskott, and Michael Pötzsch for fruitful discussions and two reviewers for valuable suggestions.

References

[1] R.P.N. Bao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence Journal*, 78:461–505, 1995.

[2] J. Beis and D. Lowe. Learning indexing functions for 3–d model based object recognition. *CVPR'94*, pages 275–280, 1994.

[3] I. Biederman and P. Kalocsai. Neurocomputational bases of object and face recognition. *Philosophical Transactions of the Royal Society: Biological Sciences*, 352:1203–1219, 1997.

[4] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models — their training and application. *Computer Vision and Image Understanding*, January:33–59, 1995.

[5] J.L. Crowley and A.C. Parker. A representation for shape based on peaks and ridges in the difference of low–pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–170, 1984.

[6] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by 2d visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1169, 1985.

[7] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks.* Wiley, New York, 1996.

[8] J. Fiser, I. Biederman, and E. Cooper. To what extent can matching algorithms based on direct outputs of spatial filters account for human shape recognition. *Spatial Vision*, 10:237–271, 1997.

[9] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE-PAMI*, 13(9):891–906, 1991.

[10] K. Fukunaga, editor. *Introduction to statistical pattern recognition (2nd ed).* Academic Press, 1990.

[11] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1995.

[12] P.J.B. Hancock, A.M. Burton, and V. Bruce. A comparison of two computer-based face identification systems with human perceptions of faces. *To appear in Vision Research*, 1997.

[13] J. Hummel and I. Biederman. Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99:480–517, 1992.

[14] S.S. Intille and A.F. Bobick. Close world tracking. *Proceedings of the Int. Conf. on Computer Vision, June*, 1995.

[15] N. Krüger. Visual learning using a priori constraints. *submitted to Neural Computation.*

[16] N. Krüger. An algorithm for the learning of weights in discrimination functions using a priori constraints. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, July:764–768, 1997.

[17] N. Krüger. Collinearity and parallism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2), 1998.

[18] N. Krüger. *Visual Learning with a priori Constraints (Phd Thesis)*. Shaker Verlag, Germany, 1998.

[19] N. Krüger, M. Pötzsch, and G. Peters. Principles of cortical processing applied to and motivated by artificial object recognition. In R. Baddeley, P. Hancock, and P. Foldiak, editors, *accepted for "Information Theory and the Brain"*. Cambridge University Press, 1998.

[20] N. Krüger, M. Pötzsch, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. *Image and Vision Computing*, August:665–673, 1997.

[21] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamik link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1992.

[22] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, July:743–756, 1997.

[23] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on communication*, vol. COM-28:84–95, 1980.

[24] H. Loos. Positionsvorhersage von bewegten Objekten in großformatigen Bildsequenzen (Diplomarbeit). Technical report, Institut für Neuroinformatik, Bochum, 1997.

[25] N. Lüdtke. Integrating of Gestalt principles by non–linear feature linking (Studiennarbeit). Technical report, Institut für Neuroinformatik, Bochum, 1998.

[26] B. Mel. Seemore: A view–based approach to 3–d object recognition using multiple visual cues. *Advances in Neural Information Processing Systems*, 8:865–871, 1996.

[27] B. Moghaddam, C. Nastar, and A. Pentland. Bayesian face recognition using deformable intensity surfaces. *CVPR*, pages 638–645, 1997.

[28] H. Murase and S.K. Nayar. Visual learning and recognition of 3–d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[29] R.C. Nelson and A. Selinger. A cubist approach to object recognition. *CVPR'95*, 1995.

[30] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.

[31] M.W. Oram and D.I. Perrett. Modeling visual recognition from neurobiological constraints. *Neural Networks*, 7:945–972, 1994.

[32] D.E. Pearson. The extraction and use of facial features in low–bit rate visual communication. *Philosphical Transactions of the Royal Society London B.*, 335:79–85, 1992.

[33] P. Perona. Deformable kernels for early vision. *IEEE-PAMI*, 17(5):488–499, 1995.

[34] A.P. Pope and D.G. Lowe. Learning object recognition models from images. In T. Poggio and S. Nayar, editors, *Early Visual Learning*. 1995.

[35] M. Pötzsch. *Context specific Statistics of Real Image Sequences leads to Corners (PhD Thesis)*. (in preparation).

[36] M. Pötzsch, N. Krüger, and C. von der Malsburg. Improving object recognition by transforming gabor filter responses. *Network: Computation in Neural Systems*, 7:341–347, 1996.

[37] Michael Pötzsch, Thomas Maurer, Laurenz Wiskott, and Christoph von der Malsburg. Reconstruction from graphs labeled with responses of gabor filters. In C. v.d. Malsburg, W. v. Seelen, J.C. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of the ICANN 1996*, Springer Verlag, Berlin, Heidelberg, New York, Bochum, July 1996.

[38] H.A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. *IEEE–PAMI*, 1998.

[39] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. *Advances in Neural Information Processing Systems*, 8:865–871, 1996.

[40] A. Shokoufandeh, I. Marsic, and S.J. Dickinson. View–based object recognition using saliency maps. *Sixth International Conference on Computer Vision*, 1998.

[41] P.K. Simpson. *Artificial Neural Systems.* Pergamon Press, 1990.

[42] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[43] J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex background. *Proceedings of the Second International Workshop on Automatic Face- and Gesture recognition, Vermont*, pages 170–175, 1996.

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

[45] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition and gender determination. *Proceedings of the International Workshop on Automatic Face- and Gesture recognition, Zürich*, 1995.

[46] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 775–780, 1997.

[47] L. Wiskott and C. von der Malsburg. A neural system for the recognition of partially occluded objects in cluttered scenes. *In I. Guyon and P.S.P. Wang, editors, Advances in Pattern Recognition Systems using Neural Networks Technologies, volume 7 of Series in Machine Perception and Artificial Intelligence. World scientific*, 1994.

[48] R.P Würtz. *Multilayer Dynamic Link Network for Establishing Image Point Correspondances and Visual Onject Recognition.* Verlag Harry Deutsch, 1995.

[49] Alan L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, 1991.

[50] M. Zerroug and R. Nevatia. Volumetric descriptions from single intensity image. *International Journal of Computer Vision*, 20(1/2):11–42, 1996.