Institut für Neuroinformatik

Ruhr-Universität Bochum

Internal Report 96–11

Object Recognition with a Sparse and Autonomously Learned Representation Based on Banana Wavelets

by

Norbert Krüger, Gabriele Peters, Christoph von der Malsburg

Ruhr-Universität Bochum
Institut für Neuroinformatik
44780 Bochum

IR-INI 96–11
Dezember 1996
ISSN 0943-2752

Object Recognition with a Sparse and Autonomously Learned Representation Based on Banana Wavelets*

Norbert Krüger§, Gabriele Peters§, Christoph von der Malsburg§‡

§ Ruhr-Universität Bochum,
Institut für Neuroinformatik,
D–44780 Bochum, Germany

‡ University of Southern California,
Dept. of Computer Science and Section for Neurobiology,
Los Angeles, CA 90089-2520, USA

Abstract

We introduce an object recognition system, based on the well known Elastic Graph Matching (EGM), but includes significant improvements compared to earlier versions. Our basic features are banana wavelets, which are generalized Gabor wavelets. In addition to the qualities frequency and orientation, banana wavelets have the attributes curvature and size. Banana wavelets can be metrically organized. A *sparse* and efficient representation of object classes is *learned* utilizing this metric organization. Learning is guided by a sensible amount of a priori knowledge in form of basic principles. The learned representation is used for a fast matching. Significant speed up can be achieved by hierarchical processing of features. Furthermore manual construction of ground truth is replaced by an automatic generation of suitable training examples using motor controlled feedback. We motivate the biological plausibility of our approach by utilizing concepts like hierarchical processing or metrical organization of features inspired by brain research and criticize a too detailed modelling of biological processing.

1 Introduction

In this paper we describe a novel object recognition system in which representations of object classes can be learned automatically. The learned representations allow a fast and effective location and identification of objects in complicated scenes. Our object recognition system is based on three pillars. Firstly, our preprocessing is based on the idea of *sparse coding* [8, 27]. Secondly, effective learning is guided by *a priori* constraints covering fundamental structure of the visual world. Thirdly, we use Elastic Graph Matching (EGM) [21, 35] for the location and identification of objects.

A sparse representation can be defined as a coding of an object by a *small number* of *binary* features taken from a *large feature space*. A certain feature is only useful for coding a small subset of objects and is not applicable for most of the other objects. Sparse coding has biologically motivated advantages like minimizing wiring length for forming associations. Baum et. al. [2] point to the increase of associative memory capacity provided by a sparse code. Ohlshausen & Field [25] argue that the retinal projection of the three–dimensional world has a sparse structure and therefore a sparse code meets the principle of redundancy reduction [1] by reducing higher–order statistical correlations of the input. As an additional advantage to the reasons mentioned above, our matching algorithm achieves a siginificant speed–up by utilizing the fact that only a small number of features is required in our sparse representation of an object. For a more detailed discussion of sparse coding we refer to [8].

Our representation of a certain view of an object class comprises only important features. These are extracted from different examples (see figure 1i-iv). The central assumption of our learning algorithm necessitates on *a priori* knowledge applied to the system in the form of general principles and mechanisms. Learning is inherently faced with the bias-variance dilemma [10]: If the starting configuration of the system is very general, it can learn from and specialize to a wide variety of domains, but it will in general have to buy this advantage by having many internal degrees of freedom. This is a serious problem since the number of examples needed to train a system scales very badly with the system's size, quickly leading to totally unrealistic learning time; or else, with a limited set of training examples the system will trivially adapt to its accidental peculiarities and the system will fail to generalize properly

*Supported by grants from the German Ministry for Science and Technology 01IN504E9 (NEUROS) and 01M3021A4 (Electronic Eye).

to new examples. This is the "variance" problem. On the other hand, if the initial system has few degrees of freedom it may be able to learn efficiently but, unless the system is designed with much specific insight into the domain at hand (the solution we criticized above), there is great danger that the structural domain spanned by those degrees of freedom does not cover the given domain of application at all —the "bias" problem.

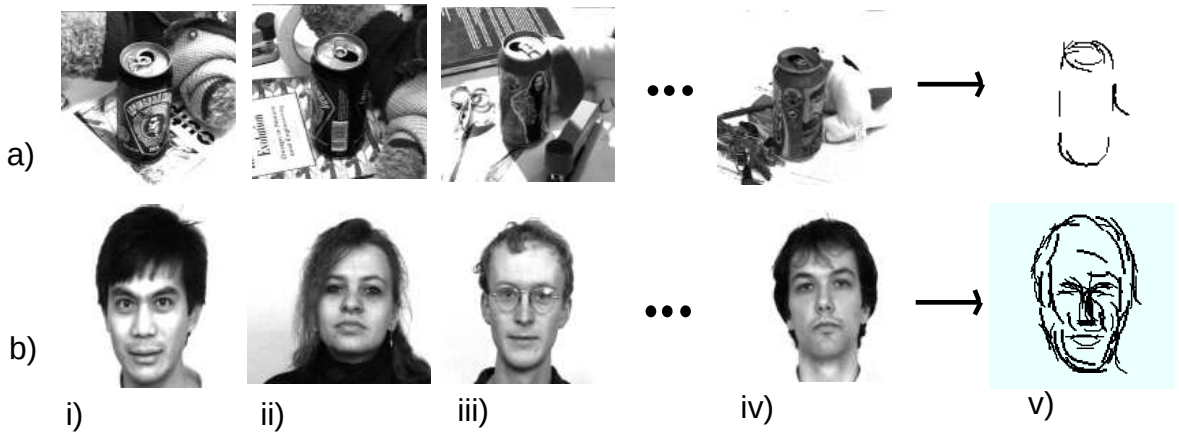

Figure 1: i–iv) Different examples of cans and faces used for learning. v) The learned representations.

We propose that *a priori* knowledge is needed to overcome the bias–variance dilemma. The challenge here is to attain generality and to avoid the extreme of equipping the system with manually constructed specific domain knowledge, such as geometry and physics in general or even the geometric and physical structure of objects themselves. We have formulated 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 assume that we can avoid the bias–problem because of the general applicability of those principles. All these principles are concerned with the selection of important features from a predefined feature space (P0, P1, P2) and the structure thereof (P3). In [18] and [17] we have already made use of the following principles: **P0** (Locality): Features refering to different locations are treated as independent; **P1** (Invariance): Features are preferred which are invariant under a wide range of object transformations; **P2** (Minimal Redundancy): Features should be selected for minimal redundancy of information.
Here we introduce a principle P3 as an important additional constraint.

P3 (Local Feature Assumption): Significant features of a local area of the two–dimensional projection of the visual world are localized curved lines.

We formalize P3 by extending the concept of Gabor wavelets (see e.g., [5]) to banana wavelets (section 2). To the parameters frequency and orientation we add curvature and size (see figure 2). [21, 35]. An object can be represented as a configuration of a few of these features (figure 1v), therefore it can be coded sparsely. The space of banana wavelet responses can be understood as a metric space, its metric representing the similarity of features. This metric is utilized for the learning of a representation of objects and for recognition of these objects during the matching procedure. The banana wavelet responses can be derived from Gabor wavelets responses by hierarchical processing to gain speed and reduce memory requests (see section 3). A set of examples of a certain view of an object class (figure 1i–iv) is used to learn a sparse representation (sections 4 and 5) which contains only the important features, i.e., features which are robust against changes of background and illumination or slight variations in scale and orientation. This sparse representation allows for quickly and effectively locating (see section 6) by using EGM.
Our system has certain analogies to the visual system of vertebrates. There is evidence for curvature sensitive features processed in a hierchical manner in early stages [6]; sparse coding is discussed as a coding scheme used in the visual system [8]; and metric organization of features seems to play an important role for information processing in the brain [14, 32]. Instead of detailed modelling of brain areas we aim to apply some basic concepts inspired by brain research (like sparse coding, hierarchical processing, metrical organisation of features, etc.) in our artificial object recognition system. We think a system does not necessarily need to contain "neurons" or "hebbian plasticity" to be called biologically motivated. Maybe we miss the important aspects of information processing in the brain by looking on a too detailed level. After all, humans did not build planes with feathers, but the observation of birds inspired the understanding of the basic principles of flying which are used by any airplane. For a more detailed discussion of the analogy to biology we refer to [19].
To enable simultaneously a rough understanding of the basic ideas of the approach and a detailed description of the algorithm this paper can be read in two modes: For every subsections we give first a short summary and then a more detailed description beginning with the phrases "Formally speaking..." or "More formally ...". The reader may skip the latter parts for a rough understanding or a first reading.

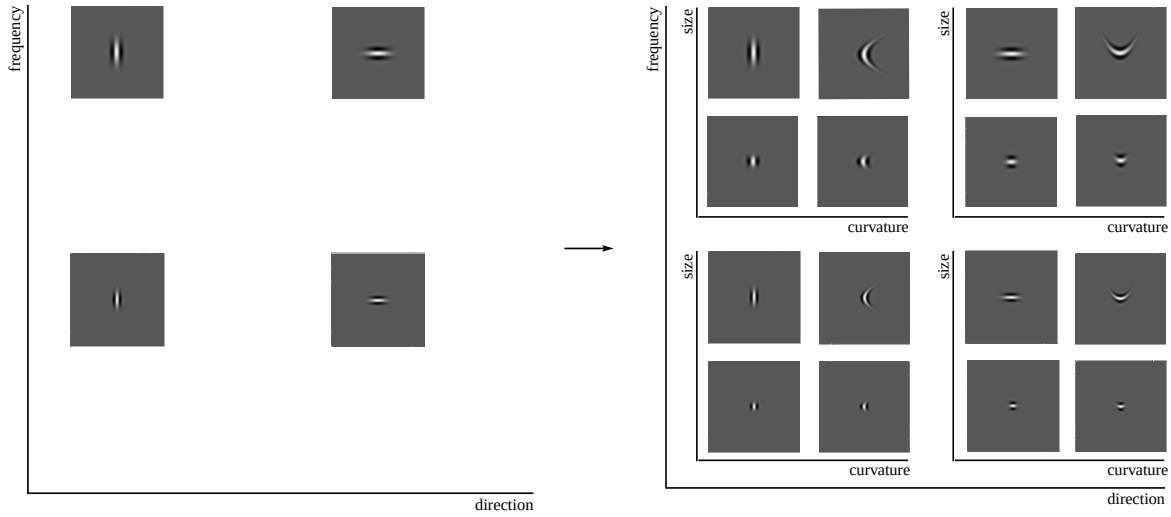

Figure 2: Relation between Gabor wavelets and banana wavelets. Left: four examples of Gabor wavelets which differ in frequency and direction only. Right: 16 examples of banana wavelets which are related to the Gabor wavelets on the left. Banana wavelets are described by two additional parameters (curvature and size).

2 The Banana Space

In this section we describe our realization of principle P3: a feature generation based on banana wavelets and its metric organization in the banana space. P3 gives us a significant reduction of the search space. Instead of allowing, e.g., all linear filters as possible features, we restrict ourself to a small subset. Considering the risk of a wrong feature selection it is necessary to give good reasons for our decision. We argue that nearly any object can be composed of localized curved lines. Furthermore, the fact that humans can easiliy handle line drawings of objects strengthens our assumption. We think that a good feature has to have a certain complexity but an extreme increase of complexity up to a spezialization to a very narrow class of objects has to be avoided. In any case, there is some arbitraryness in the assumption P3 and it therefore can only be justified by the final performance of the whole system.
Banana wavelets can be naturally organized in a metric space. Their distance expresses the similarities of qualities of the kernels sich as position, orientation or curvature. This metric organization is essential for the learning algorithm described in section 5 because it allows to summarize cluster of similar features by their center of gravity.

2.1 Banana Wavelets

Our *a priori* principle P3 states that curved lines are important features of the local visual world. A banana wavelet can be understood as a generalized Gabor Wavelet [29]. Banana wavelets, like Gabor wavelets are localized filters which can be derived from a "mother wavelet". In contrast to Gabor wavelets, which are characterized by two parameters, the set of all banana wavelets is described by four parameters (see figure 2a).
A banana wavelet $B^{\vec{b}}$ is a complex valued function defined on $\mathbb{R} \times \mathbb{R}$. It is parameterized by a vector $\vec{b}$ of four variables $\vec{b} = (f, \alpha, c, s)$ expressing the attributes frequency (f), orientation (o), curvature (c) and size (s). It can be understood as a product of a constant $\gamma^{\vec{b}}$ with a curved and rotated complex 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 3top):

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

with

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

and

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

A banana wavelet can be equivalently expressed by a combination of matrix operations $\mathcal{M}_\alpha$, $\mathcal{M}_s$ and a non–linear operation $\mathcal{M}_c$. $\mathcal{M}_\alpha$ performs a rotation by angle α, $\mathcal{M}_s$ stretches the Gaussian. $\mathcal{M}_c(\vec{x})$ is a non–linear function bending the coordinate system (see Appendix A.1).

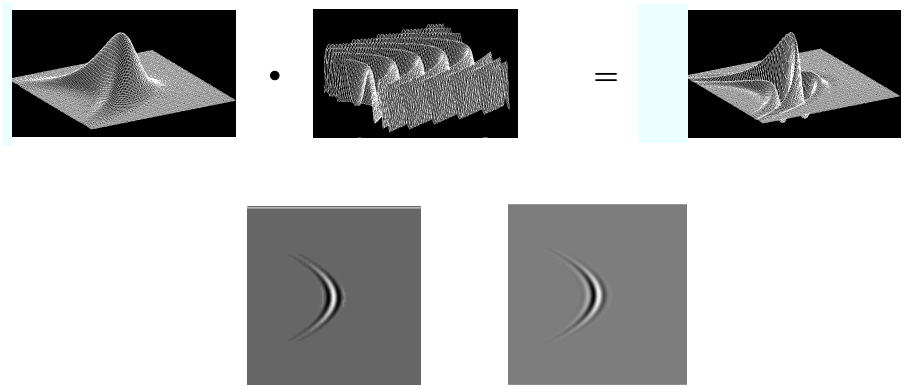

Figure 3: Top: Real part of a banana wavelet is the product of a curved Gaussian $G^{\vec{b}}(x,y)$ and a curved wave function $F^{\vec{b}}(x,y)$. Bottom: Real and imaginary part of the same banana wavelet depicted as grey level picture, with ehite encoding high values.

To ensure DC-freedom of the banana wavelets, i.e., the independence of the filter responses from the mean grey value intensity, we set

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

To compensate differences of filter responses deriving from banana wavelets of different sizes or frequencies we set

$$\gamma^{\vec{b}} = \frac{f^2 \cdot \left(1 + \xi_f \frac{f_{\max} - f}{f_{\max}}\right) \cdot \left(1 + \xi_s \frac{s_{max} - s}{s_{max}}\right)}{||B^{\vec{b}}||_2} \tag{2}$$

where $||\ ||_2$ represents the L^2 norm. The factor f^2 compensates the decrease of the power spectrum of "natural images" [7]. The factor

$$\left(1 + \xi_f \frac{f_{\max} - f}{f_{\max}}\right) \cdot \left(1 + \xi_s \frac{s_{max} - s}{s_{max}}\right)$$

ensures a more even distribution of the reponses of the banana wavelets. It intensifies responses for small size and high frequency.
We define a discrete sampling of the space of banana wavelets by a function

$$\hat{E} : (l, \hat{o}, b, m) \rightarrow (f(l), \alpha(\hat{o}), c(b), s(m))$$

embedding the discrete grid with integer coordinates $(l, \hat{o}, b, m)$ in the continuous space (f, α, c, s). In our simulations we only make use of the discrete set of banana wavelets with parameters $(f(l), \alpha(\hat{o}), c(b), s(m))$. The kernels of two banana wavelets $B^{\vec{b}_1}$ and $B^{\vec{b}_2}$ with small euclidian distance $||\vec{b}_1 - \vec{b}_2||$ have small L^2 distance by definition. Accordingly, $\hat{E}$ has to be chosen such that neighboring coordinates in the grid correspond to similar kernels. The embedding function $\hat{E}$ ensures that the features corresponding to the grid $(l, \hat{o}, b, m)$ are sufficiently seperated to avoid redundancy but also sufficiently dense to ensure a certain completeness of information.
More formally we define $\hat{E}$ by[1]

$$\begin{array}{lcll} f(l) & = & f_{\max} \cdot f_s^{-l} & l : 0, \ldots n_l - 1 \\ \alpha(\hat{o}) & = & \frac{\hat{o} \cdot 2\pi}{n_{\hat{o}}} & \hat{o} : 0, \ldots n_{\hat{o}} - 1 \\ c(b) & = & c_{\max} - \frac{2 \cdot c_{\max} \cdot b}{n_b} & b : 0, \ldots n_b - 1 \\ s(m) & = & s_{min} + m \cdot \frac{(s_{max} - s_{min})}{n_s} & m : 0, \ldots n_m - 1 \end{array} \tag{3}$$

We refer to a discrete set of banana wavelets with n_l levels, n_o orientations, n_b curvatures and n_m sizes by $\mathcal{B}$ and call it a banana plant (see figure 4). In our simulations we used the parameter settings shown in table 1, columns 1 & 2, in the following referred to as "standard settings".

[1] The parameter $\hat{o}$ runs from 0 to $2 \cdot n_o - 1$, where n_o represents the number of kernels used for the actual image processing. In case that $\hat{o}$ is larger than $n_o - 1$, i.e., $\alpha(\hat{o}) > \pi$, $B^{\vec{b}}$ with $\vec{b} = (f(l), \alpha(\hat{o}), c(b), s(m))$ represents the kernel $\mathrm{Conj}(B^{(f(l), \alpha(\hat{o}) - n_o, c(b), s(m))})$, where $\mathrm{Conj}(B)$ represents the complex conjugated kernel corresponding to B. Except for section 3 we only make use of the first n_o kernels.

Standard Parameter Settings																				
Transformation						Approximation			Banana space			Learning			Matching		
n_l	=	2	$f_{\max}$	=	2π				e_x	=	4	τ	=	0.5	θ_1	=	0.8
n_o	=	8	f_s	=	0.8				e_y	=	4	λ	=	2.5	θ_2	=	1.7
n_b	=	7	s_{min}	=	0.5	$n_b^{\mathcal{W}}$	=	1	e_f	=	10	p_1	=	0.1			
n_m	=	3	s_{max}	=	1.0	$n_m^{\mathcal{W}}$	=	1	e_α	=	0.3	p_2	=	0.7			
σ_x	=	1.0	$c_{\max}$	=	1.3	$s_{min}^{\mathcal{W}}$	=	$\upsilon_1 \cdot s_{min}$	e_c	=	0.4	r_1	=	1.0			
σ_y	=	2.0				$s_{max}^{\mathcal{W}}$	=	$s_{min}^{\mathcal{W}}$	e_s	=	3.0	r_2	=	1.5			
ξ_f	=	-0.3				υ_1	=	1.0									
ξ_s	=	0.45				υ_2	=	1.0									

Table 1: Standard Settings. Columns 1,2: Parameters of transformation. Column 3: Parameters in $\mathcal{W}$ differing from the parameters in $\mathcal{B}$. Column 4: Metric of the banana space. Column 5: Parameters of learning.

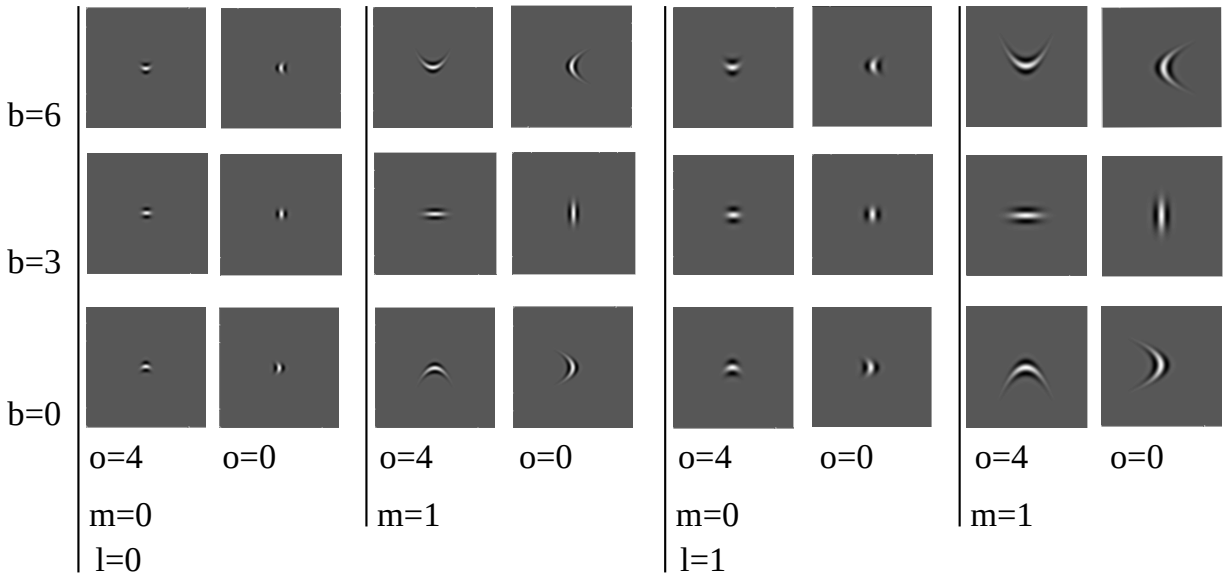

Figure 4: Banana plant. These are some examples for wavelets of a banana plant with $l = 0, \ldots, 1$ frequencies, $o = 0, \ldots, 7$ orientations, $b = 0, \ldots, 6$ curvatures and $m = 0, \ldots, 3$ magnitudes which is a standard setting.

2.2 The Banana Space

Let I be a given picture and $I(x, y)$ be its value at pixel position (x, y). The six–dimensional space of vectors $\vec{c} = (x, y, l, o, b, m)$ is called the banana coordinate space(refered to as $\mathcal{C}$), where $\vec{c}$ represents the Banana wavelet $B^{(f(l),\alpha(o),c(b),s(m))}$ at pixel position (x, y). The banana 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 we define a neighbourhood relation $N(\vec{c}_1, \vec{c}_2)$ and a metric $d(\vec{c}_1, \vec{c}_2)$ on $\mathcal{C}$. Two coordinates $\vec{c}_1, \vec{c}_2$ are expected to be neighbored (or have a small distance d) when their corresponding kernels are similar. For the coordinates pixel position (x, y), level l and size m we can assume that the similarity of corresponding kernels changes accordingly 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 convienent to arrange the corresponding kernels in a Moebius topology (see figure 5). Note that a banana wavelet with orientation $\alpha(o)$ and curvature $c(b)$ rotated by π produces the same absolute reponse as a banana wavelet with orientation $\alpha(o)$ and curvature $-c(b)$. We use the neighbourhood relation N for our feature extraction described in section 4 and the metric d in the learning algorithm described in section 5.
More formally we firstly define a Moebis topology on the subset (o, b). $(o_1, b_1), (o_2, b_2)$ are called neighbored ($N(\vec{c}_1, \vec{c}_2) := True$) if at least one of the following two conditions hold true:

$$\begin{aligned} &\text{Within Toplogy:} && \max\{|o_1 - o_2|, |b_1 - b_2|\} \leq 1 \text{ for } o_1 : 0, \ldots, n_k - 1,\ o_2 : 0, \ldots, n_k - 1 \\ &\text{Border Topology:} && (o_1 = 0 \wedge o_2 = n_o - 1 \wedge |b_1 + b_2| \leq 1) \vee (o_1 = n_o - 1 \wedge o_2 = 0 \wedge |b_1 + b_2| \leq 1)\,. \end{aligned}$$

Secondly we can extend the neighbourhood relation to $\mathcal{C}$: $\vec{c}_1$ is neighbored to $\vec{c}_2$ if (o_1, b_1) is neighbored to (o_2, b_2) and

$$\max\{|x_1 - x_2|, |y_1 - y_2|, |l_1 - l_2|, |m_1 - m_2|\} \leq 1. \tag{4}$$

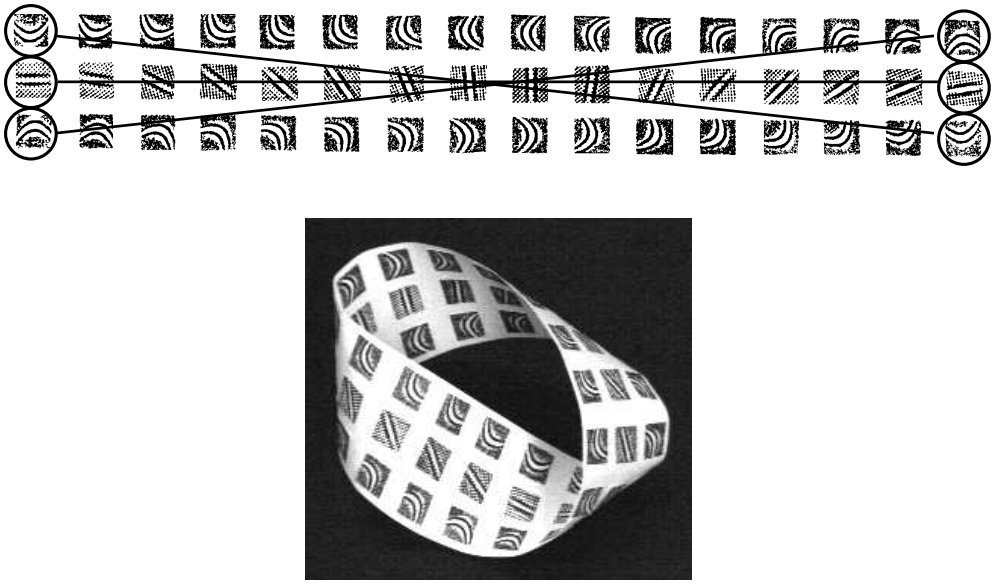

Figure 5: Moebius topology. The subspace of orientations and curvatures (o, b) with $n_o = 16$ orientations and $n_b = 3$ curvatures. Top: The banana wavelets on the left are connected by lines to the wavelets with neighbouring indices (o,b) on the right. Connecting the right edge with the left edge according to these neighbourhoods leads to the Moebius topology shown at the bottom.

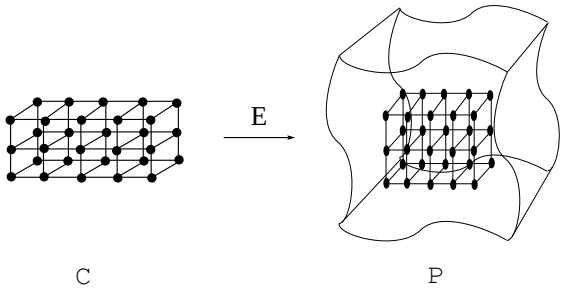

Figure 6: Embedding. The discrete banana coordinate space $\mathcal{C}$ is embedded in the continuous parameter space $\mathcal{P}$. $\mathcal{C}$ and $\mathcal{P}$ are shown in three dimensions only.

Now we define a distance measure on $\mathcal{C}$ harmonizing with its topology. The mapping

$$E : (\vec{c}) \to (x, y, f(l), \alpha(o), c(b), s(m)) \tag{5}$$

embeddes the discrete space $\mathcal{C}$ in a continuous space $\mathcal{P}$, in the following called parameter space (see figure 6). E is a simple extension of $\hat{E}$ taking also the pixel position (x, y) into account (see figure 6). After defining a metric on the parameter space $\mathcal{P}$ we use the embedding function E to translate this metric back to $\mathcal{C}$. As for the topology above we can define firstly a distance in the subspace (α, c) expressing the Moebius topology thereof. Let $(e_x, e_y, e_f, e_\alpha, e_c, e_s)$ be a cube[2] of volume one in $\mathcal{P}$. Let $d((\alpha_1, c_1), (\alpha_2, c_2))$ be defined as follows

$$\begin{aligned} &d((\alpha_1, c_1), (\alpha_2, c_2)) = \\ &\min\left\{\sqrt{\tfrac{(\alpha_1-\alpha_2)^2}{e_\alpha^2} + \tfrac{(c_1-c_2)^2}{e_c^2}},\ \sqrt{\tfrac{((\alpha_1-\pi)-\alpha_2)^2}{e_\alpha^2} + \tfrac{(c_1+c_2)^2}{e_c^2}},\ \sqrt{\tfrac{((\alpha_1+\pi)-\alpha_2)^2}{e_\alpha^2} + \tfrac{(c_1+c_2)^2}{e_c^2}}\right\} \end{aligned} \tag{6}$$

Now we can define a distance measure on $\mathcal{P}$.

$$\begin{aligned} &d(\vec{c}_1, \vec{c}_2) = \\ &\sqrt{\frac{(x_1-x_2)^2}{e_x^2} + \frac{(y_1-y_2)^2}{e_y^2} + \frac{(f_1-f_2)^2}{e_f^2} + d((\alpha_1, c_1), (\alpha_2, c_2))^2 + \frac{(s_1-s_2)^2}{e_s^2}}. \end{aligned} \tag{7}$$

Setting

$$d(\vec{c}_1, \vec{c}_2) = d(E(\vec{c}_1), E(\vec{c}_2))$$

we can finally extend $\mathcal{C}$ to a discrete metric space.

[2] Our coice of parameters are shown in table 1, column 3.

2.3 Banana Wavelet Responses

The basic feature of our object recognition system is the magnitude of the filter response of a banana wavelet $B^{\vec{b}}$ extracted by a convolution of $B^{\vec{b}}$ with the image I. In the following $(\mathcal{F}I)\left(\vec{x_0}, \vec{b}\right)$ represents the magnitude of the filter response of the banana wavelet $B^{\vec{b}}$ at pixel position $\vec{x_0}$ in image I. A banana wavelet $B^{\vec{b}}$ causes a strong response at pixel position $\vec{x_0}$ when the local structure of the image at that pixel position is similar to $B^{\vec{b}}$. We call this six–dimensional metric space $\mathcal{A}I\left(\vec{x_0}, \vec{b}\right)$ the banana response space associated with image I. The very same metric and topology as defined in (4) and (7) can be applied to this space. We call the whole construction consisting of a banana plant, the coordinate space and the response space the banana space.
More formally let the operator $\mathcal{F}$ symbolize the convolution of an image I with $B^{\vec{b}}$ for all possible $\vec{b}$ at a pixel position $\vec{x_0}$ in the image I

$$(\mathcal{F}I)\left(\vec{x_0}, \vec{b}\right) := \int B^{\vec{b}}\left(\vec{x_0} - \vec{x}\right) I\left(\vec{x}\right) d\vec{x} = \left(B^{\vec{b}} * I\right)\left(\vec{x_0}\right) \tag{8}$$

and let $\mathcal{A}I\left(\vec{x_0}, \vec{b}\right)$ be the magnitudes of $(\mathcal{F}I)(\vec{x_0}, \vec{b})$

$$\mathcal{A}I\left(\vec{x_0}, \vec{b}\right) := \left|(\mathcal{F}I)\left(\vec{x_0}, \vec{b}\right)\right| . \tag{9}$$

Figure 7 shows the complex and absolute responses for an image and a specific banana wavelet.

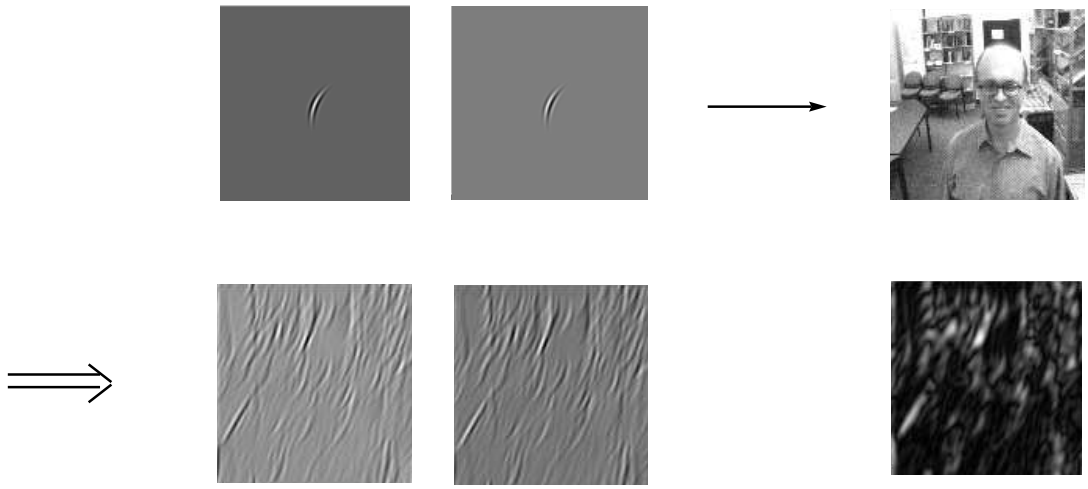

Figure 7: Results of a transformation with banana wavelets. Top: real part of a banana wavelet, imaginary part of a banana wavelet, image to be transformed. Bottom: the results of the convolution of the image with the wavelet. From left to right: real part of the convolution result, imaginary part of the convolution result, magnitude of the convolution result. White pixels code high values, so there are local maxima at those parts of the image which show lines or edges of the same orientation, curvature and size as the banana wavelet (here especially the head of the person).

2.4 Path Corresponding to a Banana Wavelet

To every banana wavelet $B^{\vec{b}}$ there can be defined a curve $\vec{p^b}$, called the path corresponding to $B^{\vec{b}}$ (see figure 8a,b)[3]. This curve is used in section 3 to speed up the transformation of an image by hierarchical processing. It also allows the visualization of the learned representation of an object (see figure 8c). Therefore the path corresponding to a banana wavelet also represents a transition of a grey level feature (represented by a banana wavelet) to a feature based on line drawings. In the approximation algorithm described in section 3 we apply two qualities connected with a curve $\vec{p}$, the derivative $\dot{\vec{p}}(t_0)$ at a certain point t_0 expressing the tangent vector at $\vec{p}(t_0)$ and the length $L(\vec{p})$ of the curve.
More formally we define

$$\vec{p^b}(t) : [-1, 1] \to \mathbb{R}^2 \tag{10}$$

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

We can equivalently express $\vec{p^b}(t)$ in our matrix notation (see appendix A.2).

[3] For the concept of curves see e.g., [23]

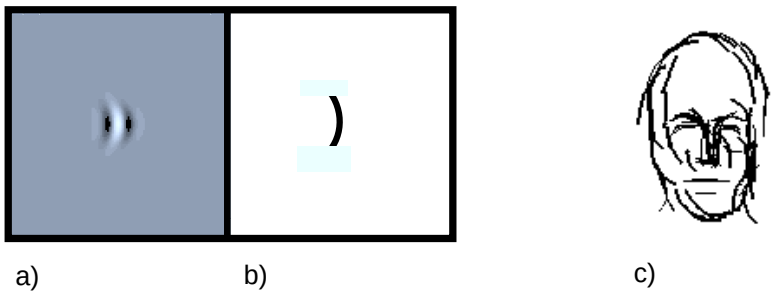

Figure 8: Path corresponding to a banana wavelet. a: Arbitrary wavelet. b: Corresponding path. c) Visualization of a representation of an object class. The width of a line segment depends on the parameter l, banana wavelets with lower frequencies are represented by line segments with larger width.

3 Approximation of Banana Wavelets by Gabor Wavelets

The banana response space contains a huge amount of features, their generation takes a long time on a sequential computer and requires large memory capacities. E.g., a transformation with our standard setting (as defined in table 1) needs approximately 21 seconds on a Sparc Ultra and requires 80 megabytes of main memory. Here we define an algorithm to approximate banana wavelets from a small set of Gabor wavelets and banana wavelet responses from Gabor wavelet responses by hierarchical processing. This approximation can be performed before the matching (as described in section 6) 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 our representations of objects only a small subset of the banana space is actually used during matching and can be evaluated therefore very fast. In case that all Banana wavelets are evaluated before matching we achieve by the hierarchical processing speed up of a factor 5. In the virtual mode we can accelerate the matching up to a factor 12 and we can reduce memory requests by a factor 20. The reader who is more interested in the learning algorithm may skip this section.

3.1 The Approximation Problem

Let $\mathcal{B}$ be a set of banana wavelets. Let $\mathcal{W}$ be a discrete set of banana wavelet $W^{\vec{w}}$ with zero curvature ($n_b = 0$), one size ($n_m = 1$) and n_f, n_o chosen as for $\mathcal{B}$. The elements of $\mathcal{W}$ can be interpreted as Gabor wavelets because they only have the variable qualities frequency and orientation. Let $W^{(\vec{x},\vec{w})}$ be $W^{\vec{w}}$ translated by the vector $\vec{x}$. Our aim is to approximate an arbitrary banana wavelet in $\mathcal{B}$ by a weighted sum of translated banana wavelets in $\mathcal{W}$ (see figure 9). Let

$$J^{\vec{b}} = \left\{ (\vec{x}_j^{\vec{b}}, \vec{w}_j^{\vec{b}}) \right\}, \; j : 0, \ldots, n^{\vec{b}} - 1$$

be a set of positions $\vec{x}_i^{\vec{b}}$ and parameter vectors $\vec{w}_i^{\vec{b}}$. We calculate the approximation $\hat{B}^{\vec{b}}$ of $B^{\vec{b}}$ by a weighted sum of Gabor wavelets in $\mathcal{W}$

$$\hat{B}^{\vec{b}} = \sum_{(\vec{x}_j^{\vec{b}}, \vec{w}_j^{\vec{b}}) \in J^{\vec{b}}} \beta_j^{\vec{b}} \cdot W^{(\vec{x}_j^{\vec{b}}, \vec{w}_j^{\vec{b}})}. \tag{11}$$

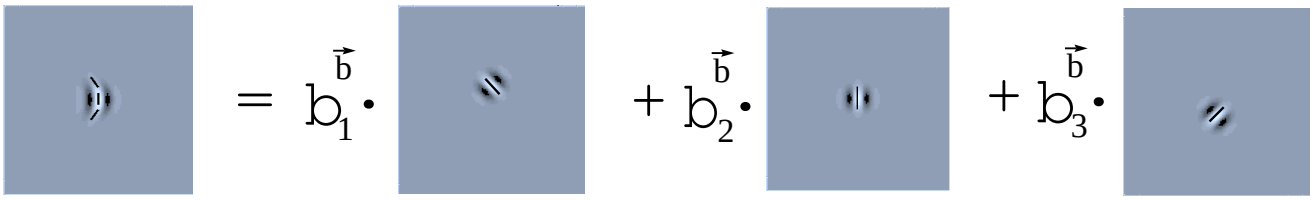

Figure 9: Approximation. The banana wavelet on the left is approximated by the weighted sum of Gabor wavelets on the right.

In this approximation problem we have to regulate two different and contradictional entities. The quality of approximation and the number of basis functions used for the approximation. In terms of the quality of approximation we like to minimize $||\hat{B} - B||_2$, where $||\ ||_2$ represents the L^2–norm. In terms of speed of approximation we like to minimize the number of additions and multiplications in (11), i.e., $|J^{\vec{b}}|$ (for a set $\mathcal{S}$ we define $|\mathcal{S}|$ as the number of elements of $\mathcal{S}$). Because of the similarity of the Gabor wavelets in $\mathcal{W}$ to a local part of a banana wavelet in $\mathcal{B}$ we expect to get a fairly accurate approximation with a small number of Gabor wavelets (see figure 10).

Let $\mathcal{F}^{\mathcal{W}}I$ be the complex reponses associated with I obtained by a convolution with the elements of $\mathcal{W}$. Given the approximation in equation (11) we can analogously calculate an approximation $\hat{\mathcal{A}}I$ of $\mathcal{A}I$ by

$$\hat{\mathcal{A}}I\left(\vec{x_0},\vec{b}\right) = \sum_{(\vec{x}_j^{\vec{b}},\vec{w}_j^{\vec{b}})\in J^{\vec{b}}} \left|\beta_j^{\vec{b}} \cdot \mathcal{F}^{\mathcal{W}}I\left((\vec{x}_0 - \vec{x}_j^{\vec{b}}),\vec{w}_j^{\vec{b}}\right)\right| \tag{12}$$

We define $s^{\mathcal{W}}_{\min} = \upsilon_1 \cdot s^{\mathcal{B}}_{\min}$ and $s^{\mathcal{W}}_{\max} = s^{\mathcal{W}}_{\min}$. The parameter υ_1 determines the width of the Gaussian of the Gabor wavelet in y–direction. The number of directions n_o is chosen independently. A large number of orientations n_α improves the accuracy of approximation but presupposes a more time consuming convolution to obtain $\mathcal{A}^{\mathcal{W}}I$ (see subsection 3.3). The approximation in (12) can be performed before the later matching stages or in a virtual mode, i.e., (12) can be calculated only if a certain banana response is requested from the matching algorithm (see section 6). In the first case we achieve a speed up by a factor of 4.7. In the virtual mode the speed up depends on the complexity of the representation used for matching. For a typical task as decribed in section 7 we achieve a speed up by a factor of 10 (wird noch mehr werden).

3.2 Approximation using a Path Corresponding to a Banana

We present a solution of the approximation problem defined above by utilizing the path corresponding $\vec{p^b}$ to a banana wavelet (as described in section 2.4). We simply choose as $\vec{x}_j, \vec{w}_j$ the closest Gabor wavelet in $\mathcal{W}$ to the tangent on $\vec{p^b}(t_i)$ for aquidistantly seperated t_i in the interval $[-1,1]$ and we choose the weight $\beta_i^{\vec{b}}$ according to the magnitude of $B^{\vec{b}}$ at the position $\vec{p^b}(t_i)$ (see figure 9).

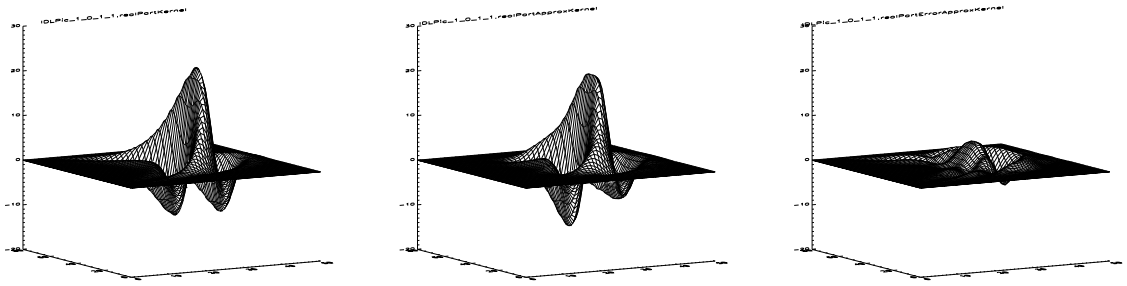

Figure 10: Left: Real part of a banana wavelet. Middle: Approximation of the banana wavelet. Note that the symmetry along the contour line of the original banana wavelet is not conserved in the approximation. Especially for stronger curvatures this effect increases. Right: error of approximation.

Formally speaking, the number $n^{\vec{b}}$ of Gabor wavelets used to approximate a certain $B^{\vec{b}}$ with is proportional to the length of the path corresponding to $B^{\vec{b}}$ devided by the length of a path corresponding to the Gabor wavelet with same frequency f and zero orientation[4], i.e., $W^{\vec{w}}$ with $\vec{w} = (f,0,0,s^{\mathcal{W}}_{\min})$

$$n^{\vec{b}} = \upsilon_2 \frac{L(B^{\vec{b}})}{L(W^{(f,0,0,s^{\mathcal{W}}_{\min})})}.$$

An increase of υ_2 leads to a narrowing of the base points of the approximation and therefore to an overlapping of the Gabor wavelets. The centre of the j–th Gabor wavelet $\vec{x}_j$ is defined as

$$\vec{x}_j^{\vec{b}} = \vec{p^b}(t_j)$$

for aquidistantly seperated

$$t_j = -1 + 2\frac{j}{(n^{\vec{b}}-1)},\ j:0,\ldots,n^{\vec{b}}-1.$$

Let $o(\vec{p^b}(t))$ be the index of the orientation[5] of the $W \in \mathcal{W}$ with associated derivative closest to the derivative $\dot{\vec{p^b}}(t)$. Then we set

$$W^{(\vec{x}_j,\vec{b}_j)} = W^{(\vec{p^b}(t_j),(f,o(\vec{p^b}(t))))},$$

[4]Note that the pathes corresponding to Gabor wavelets with a certain frequency have the same length.

[5]Here $o(\vec{p^b}(t))$ goes from 0 to $2\cdot n^{\mathcal{W}}_o - 1$. The imaginary part of a banana wavelet is not axis symmetric, therefore the conjugated Gabor wavelet is needed to cover all curvatures.

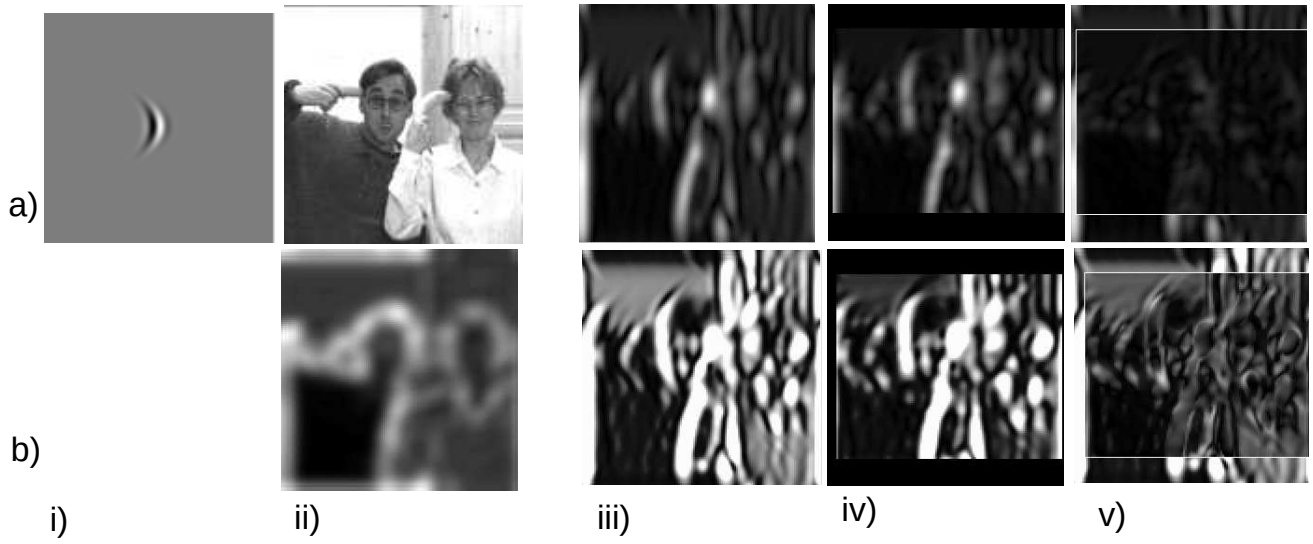

Figure 11: a,i: Banana wavelet. a,ii) Input picture. aiii) original transformation of the picture in a,ii) with the banana wavelet in a,i). a,iv) Gabor approximation of the trafo in a,iii). a,v) difference between a,iV) and a,iii). b,ii) The function $E(I,\vec{x})$. b,iii–v) The normalized trafo, its Gaborapproximation, and the difference of both for the kernel in a,i).

i.e., we have $(\vec{x}_j, \vec{b}_j) = (\vec{p^b}(t_j), (f, o(\vec{p^b}(t))))$. We calculate the weights $\beta_j^{\vec{b}}$ the following way. We define

$$\tilde{\beta}_j^{\vec{b}} = \text{real}(B(\vec{p^b}(t_j))).$$

and ensure that $\hat{B}^{\vec{b}}$ has the same norm as $B^{\vec{b}}$ by setting

$$\beta_j^{\vec{b}} = \tilde{\beta}_j \cdot \frac{||B^{\vec{b}}||_2}{||\sum \tilde{\beta}_j^{\vec{b}} \cdot W^{(\vec{x}_j, \vec{b}_j)}||_2}.$$

3.3 Quality of Approximation

We can measure the quality of the approximation in the space of filters by calculating the mean L^2 distance of the banana wavelets and its approximation[6]

$$q_1(\mathcal{B}, \hat{\mathcal{B}}) := \frac{1}{|\mathcal{B}|} \sum_{\vec{b} \in \mathcal{B}} \frac{||\hat{B}^{\vec{b}} - B^{\vec{b}}||_2}{||B^{\vec{b}}||_2}$$

or in the space of filter reponses by evaluating the differences of the transformation using the original kernels or the formula (12)

$$q_2(\mathcal{A}, \hat{\mathcal{A}}, \mathcal{I}) := \frac{1}{|\mathcal{I}||\mathcal{B}|} \sum_{I \in \mathcal{I}} \sum_{\vec{b} \in \mathcal{B}} \frac{||\hat{\mathcal{A}} I(\vec{b}) - \mathcal{A} I(\vec{b})||_2}{||\mathcal{A} I(\vec{b})||_2},$$

where $\mathcal{I}$ is a set of pictures and $\mathcal{A}I(\vec{b})$ respectively $\hat{\mathcal{A}}I(\vec{b})$ are the functions representing the whole image convoluted with $B^{\vec{b}}$ respectively $\hat{B}^{\vec{b}}$. Note that q_1 and q_2 are not completely dependent (see caption table 2). Table 2 gives the quality of approximation and the speed up for different parameter settings of $n_o^{\mathcal{W}}$, υ_1 and υ_2.

4 Extracting the important Banana Responses per Instance

Our second stage of preprocessing reduces the number of vectors $\vec{c}$ in the coordinate space $\mathcal{C}$ to represent a certain picture I or an local area of I. Our aim is to extract the local structure in I in terms of curved lines expressed by banana wavelets. Some of these lines may be important to represent the specific object, but there will be also curved lines representing features which are caused by accident conditions, e.g., shadows caused by specific illumination, background or object surface texture. An algorithm extracting the important features for a class of objects from different pictures of this object based on the preprocessing described here is presented in section 5.

[6] The division by $||B^{\vec{b}}||_2$ ensures that q_1 is independent of a simple scalar multiplication of the banana wavelets.

Quality of Approximation														
Parameter							quality		org. trafo		appr. trafo		virt. trafo	
Org. Trafo				App. Trafo					sec.		sec.		sec.	
n_k	n_o	n_b	n_m	$n_o^{\mathcal{W}}$	υ_1	υ_2	q_1	q_2	match	conv	match	conv	match	conv
2	8	7	3	8	0.5	2.0	0.27	0.21	1.1	21	1.1	10.5	0.9	1.0
2	8	7	3	8	1.0	2.0	0.27	0.19	1.1	21	1.1	6.4	0.8	1.0
2	8	7	3	8	1.5	2.0	0.39	0.22	1.1	21	1.1	5.2	0.74	1.0
2	8	7	3	8	2.0	2.0	0.49	0.26	1.1	21	1.1	4.4	0.7	1.0
2	8	7	3	8	1.0	0.75	0.34	0.17	1.1	21	1.1	4.3	0.69	1.0
2	8	7	3	8	1.0	1.0	0.34	0.16	1.1	21	1.1	4.5	0.7	1.0
2	8	7	3	8	1.0	1.5	0.31	0.19	1.1	21	1.1	5.5	0.75	1.0
2	8	7	3	8	1.0	2.0	0.27	0.19	1.1	21	1.1	6.4	0.8	1.0
2	8	7	3	8	1.0	2.5	0.28	0.2	1.1	21	1.1	7.06	0.83	1.0
2	8	7	3	4	1.0	1.0	0.5	0.3	1.1	21	1.1	4.3	0.69	0.5

Table 2: Quality of Approximation: Row 1–4: Variation of υ_1 with constant υ_2. Row 5–9: Variation of υ_2 with constant υ_2. Although q_1 is minimal for the $\upsilon_1 = 1.0,\ \upsilon_2 = 2.0$ q_2 has its minimum for $\upsilon_1 = 1.0,\ \upsilon_2 = 1.0$. We assume this effect is caused by the fact that an increase of υ_2 narrows the base points of approximation. In natural pictures lines are frequently features. This regularity decreases the necessity of many base points. Row 10: Approximation with only 4 curvatures in the first trafo. The transformation without approximation requests 80 MB main memory (the trafo and and the Fourier transformed kernel have to be stored), tha approximated trafo requests 40 MB main memory and the virtual trafo requests 4 MB main memory for the transformation of the kernel in $\mathcal{W}$

We define an *important feature* in one image (or per instance) by two qualities C1 and C2. An *important feature per instance*

C1 has a strong response,

C2 has to represent a local maximum in the banana space.

C1 represents the requirement that a certain feature or similar feature is present, whereas C2 allows a more specific characterization of this feature. Banana responses vary smoothly in the coordinate space. Therefore the six–dimensional function $\mathcal{A}I\left(\vec{x_0}, \vec{b}\right)$ is expected to have a properly defined set of local maxima. In terms of analogy to the processing in area V1 in the vertebrate visual system C1 may be interpreted as the response of a certain column which indicates the general presence of a feature coded in this column, whereas C2 represents the interculumnar competition giving a more specific coding of this feature [32]. Figure 12 shows the significant features per instance represented by their corresponding path.
We say a banana wavelet has a "strong response" at a certain pixel position $\vec{x}$ when it is larger than an average response $E(I, \vec{x}_0)$. For this average response we consider the average activity in the complete reponse space, but we take also the average activity of a local area in the response space into account. Therefore a global and local normalization is performed.
Formally speaking, we define the mean local activity $E(I, \vec{x}_0)$ at pixel position $\vec{x}_0$ and the mean total activity $E(I)$ of the banana space by

$$E(I, \vec{x}_0) = \sum_{\vec{x} \in A(\vec{x}_0, r_E)} \sum_{\vec{b} \in \mathcal{B}} \mathcal{A}I\left(\vec{x}, \vec{b}\right)$$

and

$$E(I) = \sum_{\vec{x} \in I} \sum_{\vec{b} \in \mathcal{B}} \mathcal{A}I\left(\vec{x}, \vec{b}\right)$$

where $A(\vec{x}_0, r_E)$ represents the cuboid with center $\vec{x}_0$ and length of side r_E in the (x, y) space in which the local activity is calculated (see figure 13). The function $E(I, \vec{x}_0)$ has high values, when there is a lot of structure in the local area around $\vec{x}_0$. We now define a threshold by the average of these two activities

$$T(\vec{x}_0) = \frac{\tau E(I) + (1 - \tau) E(I, \vec{x}_0)}{2}$$

and we can formalize C1 and C2 as follows: A banana reponse $\mathcal{A}I\left(\vec{x_0}, \vec{b}_0\right)$ represents a significant feature per instance if

C1: $\mathcal{A}I\left(\vec{x}_0, \vec{b}\right) > \lambda \cdot T_0$,

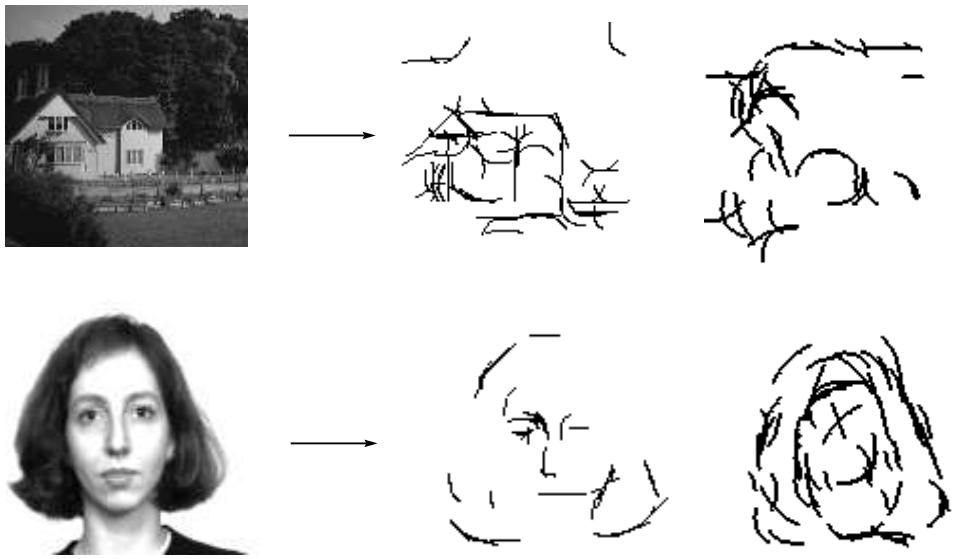

Figure 12: Result of the second stage of preprocessing. Left column: the original images. Middle column: Significant Features corresponding to banana wavelets of high frequency expressed by its corresponding path. Right column: Significant Features corresponding to low frequency. The detailed structure of the house and the inner features of the face are best described by elements of the banana space with high frequency. E.g., the eyes of the person are best described by banana wavelets with small size.

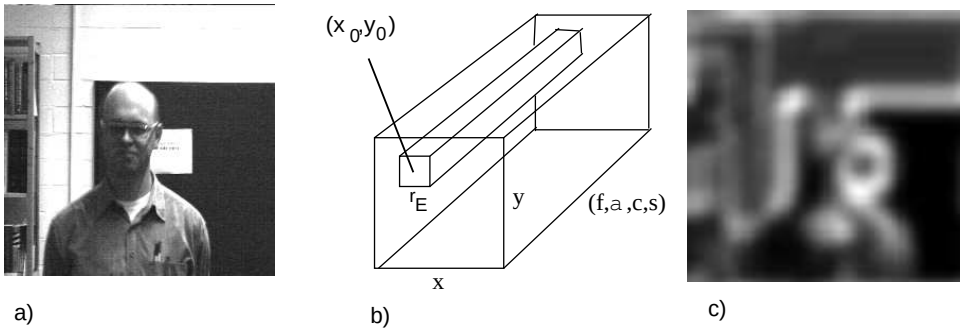

Figure 13: Normalization. a: Input Picture I. b: The local activity is calculated within the small cuboid $A(\vec{x}_0, r_E)$. c: The function $E(I, \vec{x})$.

C2: $\mathcal{A}I\left(\vec{x}_0, \vec{b}_0\right) \geq \mathcal{A}I\left(\vec{x}_i, \vec{b}_i\right)$ for all neighbours of $(\vec{x}_0, \vec{b}_0)$ as defined in (4).

The parameter λ regulates 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. The parameter τ regulates the influence of the local versus the global activity (our choice of parameters is shown in table 1). To reduce the time for calculating the average activities $E(I, \vec{x}_0)$, we approximate them by taking only the banana responses for the smallest size and with zero curvature into account. The responses corresponding to banana wavelets with same orientation but different curvature or size are highly dependent because they represent similiar features. For the calculation of $E(I, \vec{x}_0)$, which just represents some kind of average activity, only one of these similar features has taken into account.

5 Learning

Here we describe an algorithm to extract invariant local features representing landmarks for a class of objects. We assume the correspondence problem to be solved, i.e., we 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 our simulations we determine corresponding landmarks by manual construction, for the rest we replaced this manual intervention by motor controlled feedback (see section 7). For learning, it is indispensable to ensure that comparable entities are used as training data, otherwise the effect of learning will decrease because of

the noise of the trainings data. Furthermore it is advantagous to split a large learning problem (like the learning of a representation of a face) into smaller subproblems (like learning the representation of the eye region or the top of the head). This learning with comparable and smaller entities is the meaning of our a priori principle P0.
In a nutshell the learning algorithm works as follows: We extract the significant features for (as described in section 4) different images of an object taken at a certain pose for a specific landmark. For each landmark we collect these features in one bin. We define a certain feature as significant when this feature or a similar feature (according to our metric (7)) occurs often in the bin, i.e. it occurs often in the different images of our training set. We end up with a graph with its nodes labeled with elements of the banana coordinate space, expressing the learned significant features mostly representing edges of an object or invariant inner features like eyes or the nose. We refer to such a representation of an object class $\mathcal{O}$ as $\mathcal{S}^{Rep(\mathcal{O})}$ and to the set of pixels of the coordinate space representing the k–th landmark as $\mathcal{S}_k^{Rep(\mathcal{O})}$. Figure 14 illustrates the learning algorithm.
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 (P1). This is realized by measuring the probability of occurence of features in a local area of the banana space for different examples. Therefore its metric allows the grouping of similar features in one bin, but it also allows the reduction of redundancy of information (P2) by avoiding multiple features of small distance in the learned representation.

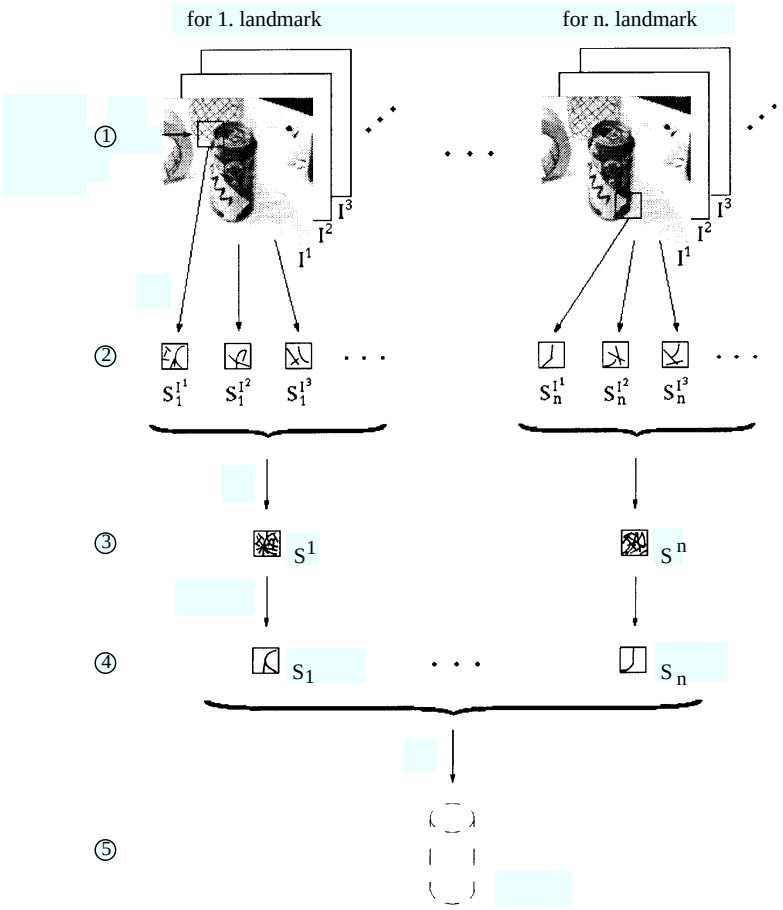

Figure 14: Schematic explanation of the learning algorithm: 1. Calculate the convolution of a banana plant with corresponding landmarks in all training images. 2. Extract the significant features per instance for a specific landmark. 3. Collect these features in one bin. 4. Learned significant features for a landmark extracted from all images. 5. Learned representation for an object of a certain view.

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 an 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 significant feature per instance extracted in the area $I^{(j,k)}$. We collect all $\vec{s}_{ij}^{\,k}$ for a specific k in one set S^k. Then we apply the LBG–vector quantization algorithm [20] to S^k (see figure 15). 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 15b). n_{C^1} dependends 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 las a higher density in the training set.
The LBG–algorithm reduces the distortion error, i.e., the average error occuring, when all elements of S^k are replaced by the nearest codebook vector in C^1. In case of high densities of elements $\vec{s}^k_{ij}$ 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. We construct a smaller codebook C^2 in which the $\vec{c},\vec{c}' \in C^1$ with close distances are combined to their centre of gravity: Let $r_1 \in I\!R^+$ be fixed. We calculate for all $\vec{c} \in C^1$ the number of $\vec{c}' \in C^1$ with distance $d(c,c') < r_1$. (figure 15c). If there exist one such $\vec{c}' \neq \vec{c}$ we substitute all the codebook vectors in C^1 with $d(\vec{c},\vec{c}') < r_1$ by their center of gravity (figure 15d). C^2 now represents a code book with less or equal elements than C^1 without redundant codebook vectors. Now we can define the important features for the k-th landmark of a certain object as those codebook vectors $\vec{c} \in C^2$ for which a certain percentage p of $\vec{s}^k_{ij}$ exists with $d(\vec{c},\vec{s}^k_{ij}) < r_2$ (figure 15e,f). We collect these important features in a set $\mathcal{S}_k^{Rep(\mathcal{O})}$ which is our learned representation of the k-th landmark of a certain class of objects.

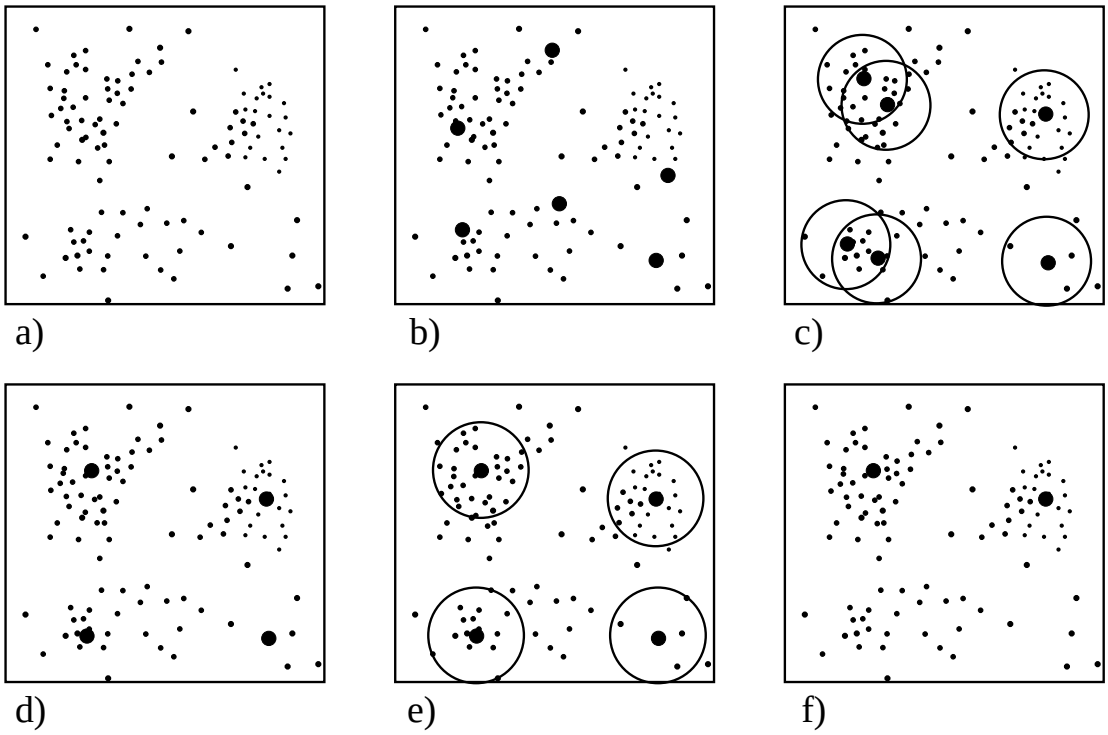

Figure 15: Clustering: a) Distrubution of data. b) Codebook Initialization. c) Codebook vectors after learning. d) Substituting sets of codebook vectors with small distance ($< r_1$) by their center of gravity. e) Counting number of elements within radius r_2. f) Deleting codebook vectors representing insignificant features.

6 Matching

To use our learned representation for location and classification of objects we have to define a similarity between the extracted representation $\mathcal{S}^{Rep(\mathcal{O})}$ and a certain position in the image. A view of an object is characterized by a small number of binary features (a certain banana is present or absent) from a large feature space (the banana space). This sparse coding will allow a fast matching, because only the presence of a few features has to be checked in the pictures. Here we define a simililarity function of a graph labeled with banana wavelets with certain size and position in an image. We define a *total similarity* expressing the system's confidence whether there is a certain object on an image I at a certain position and size. As in [35] it simply averages *local similarities*, expressing the system's confidence whether a node of the graph represents a local feature. A graph is adapted to an image by EGM [21, 35]. The total similarity is optimized in two steps: Shifting (global move) and scaling of the graph. The optimal similarity value for a graph gives the quality of its fit to the image. For each stored size of an object we perform a separate match. The graph with the highest similarity determines the size and position of the objects within the image, while the positions of its nodes identify the landmarks.
In a nutshell the local similarity is defined as follows: For each learned feature in $\mathcal{S}_k^{Rep(\mathcal{O})}$ and pixel position in the image we simply check whether the corresponding banana response is high or low, i.e., the corresponding feature is

present or absent. Because of the sparseness of our representation only *a few* of these checks have to be made, therefore the matching is very *fast*. Because we make use only of the *important* features, the matching is very *efficient*.

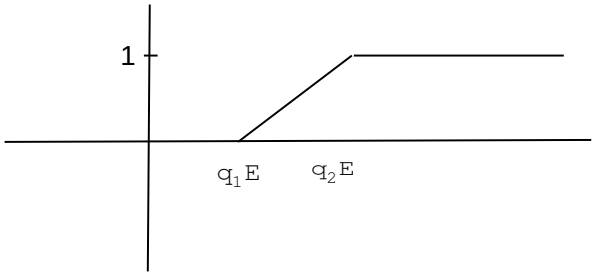

Figure 16: The normalization function $N(t, I, \vec{x})$.

More formally we introduce a normalization in the banana space to transform our real valued filter responses $\mathcal{A}I\left(\vec{x_0}, \vec{b}\right)$ into quasi binary features which are comparable to the pixels of the coordinate space in our learned representation. The normalized responses do less depend on the exact filter response but represent the presence or absence of a certain feature. Let the sigmoid function

$$N(t, I, \vec{x}_0) = \begin{cases} 0 & \text{for} \quad s < \theta_1 E(I, \vec{x}_0) \\ \frac{t - \theta_1 E(I, \vec{x}_0)}{\theta_2 E(I, \vec{x}_0)} & \text{for} \quad \theta_1 E(I, \vec{x}_0) < s < \theta_2 E(I, \vec{x}_0) \\ 1 & \text{for} \quad s > \theta_2 E(I, \vec{x}_0) \end{cases}$$

be our normalization function (see figure 16). Figure 11b) shows the normalized transformation. The value $N(\mathcal{A}I\left(\vec{x_0}, \vec{b}\right))$ represents the system's confidence of the presence of the feature $\vec{b}$ at position $\vec{x_0}$. This confidence is high when the response exceeds the average activity significantly. The exact value of the response is not of any interest. We like to avoid a very strict decision at this stage, therefore we still allow a range of indicision of the system when the response is only slightly above the average activity.
Now we can define a local similarity $Sim(\mathcal{S}_k^{Rep(\mathcal{O})}, I^{(x,y)})$ between a node labeled with banana wavelet reponses $\mathcal{S}_k^{Rep(\mathcal{O})}$ and a pixel position $I^{(x,y)}$ in an image I by simply averaging the normalized filter responses corresponding to the learned representation of the k-th landmark (i.e., $\vec{s}_i = (x_i, y_i, f_i, \alpha_i, c_i, s_i) \in \mathcal{S}_k^{Rep(\mathcal{O})}$) in the image at the pixel position (x, y):

$$Sim(\mathcal{S}_k^{Rep(\mathcal{O})}, I^{(x,y)}) = \frac{1}{|\mathcal{S}_k^{Rep(\mathcal{O})}|} \sum_{\vec{s}_i \in \mathcal{S}_k^{Rep(\mathcal{O})}} N\left(\tilde{\mathcal{A}}I\left((x - x_i, y - y_i), (f_i, \alpha_i, c_i, s_i)\right)\right). \tag{13}$$

The number of pixels in the coordinate space a node of the graph $\mathcal{S}^{Rep(\mathcal{O})}$ is labeled with is very small, therefore the evaluation of 13 is very fast. In the bunch graph representation in [35] a node is labeled by a large number of vectors (approximately 70) of Gabor Filter responses, each describing a landmark of one instance of the landmark of an object in the training set. Therefore the evaluation of the local similarity in [35] takes much longer.
As in [35, 18] the total similarity $Sim(\mathcal{S}^{Rep(\mathcal{O})}(x, y, s), I)$ between a graph $\mathcal{S}^{Rep(\mathcal{O})}(x, y, s)$ at position (x, y) with size s and the image I is simply defined as the average of the local similarities defined above:

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

with n_k represents the number of nodes of the graph.

7 Simulations

We demonstrate the applicability of our algorithm to a wide range of problems. First we learn representations of cans and faces of different poses. We apply these representations to the problem of locating these objects in complex scenes using the matching algorithm described in section 6. Finally we demonstrate a classification task, the discrimination of frontal faces and non–frontal faces. If not stated explicitly, we used in our simulations the standard settings defined in table 1. With these settings the transformation (without the approximation described in section 3) of a 128x128 picture needs 21 seconds, the extraction of significant features per instance takes approximately 0.7 seconds per node and picture and the final learning as described in section 5 takes 0.5 seconds for each landmark for a training set of 60 examples. All simulations were performed on a Sparc Ultra.

7.1 Learning of Representation

Firstly we apply the learning algorithm described in section 5 to data consisting of manually provided landmarks. In subsection 7.1.2 we replace this manual intervention by motor controlled feedback.

7.1.1 Learning with manually provided ground truth

Our training sets consist of a set of approximately 60 examples an object viewed in a certain pose. As objects we used cans, frontal faces and half profile faces. Corresponding landmarks are defined manually on the different representatives of a class of objects (see figure 17).

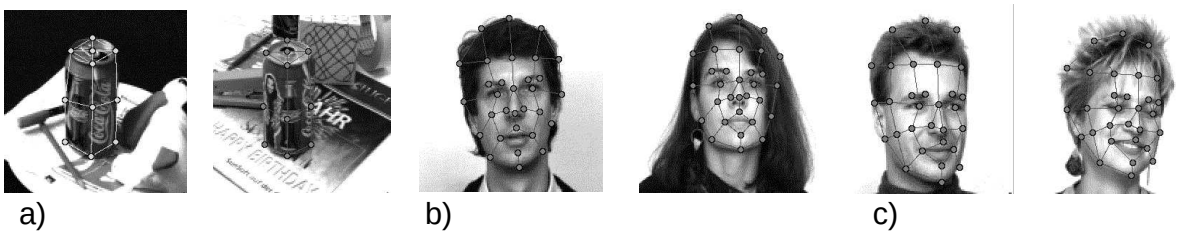

Figure 17: Manual defined graphs for a: cans, b: frontal faces and c: half profiles.

Figure shows 18 the significant features per instance for some of the can examples in the training set as well as the learned representations. Figure 19 shows the learned representations for faces using manual defined graphs as shown in figure 17.

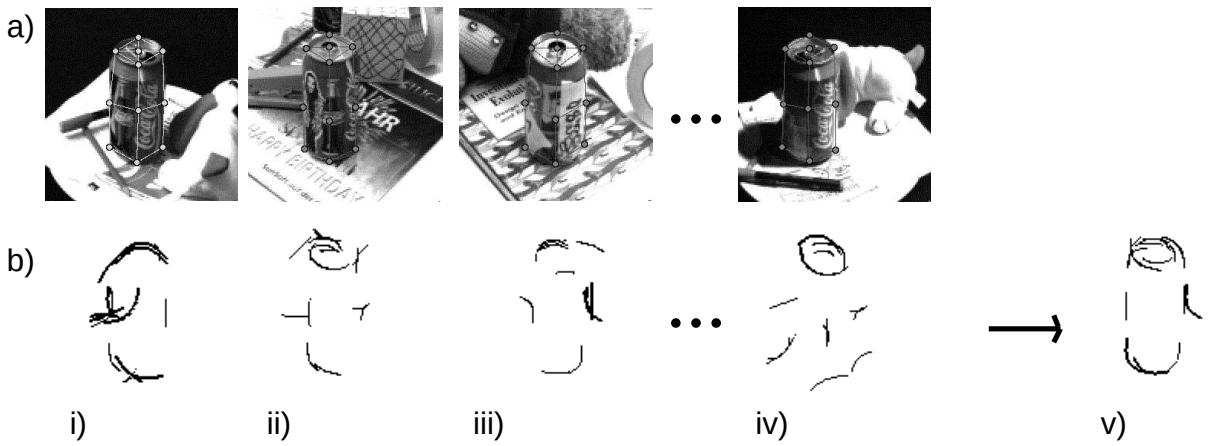

Figure 18: a: Pictures for training. bi–iv: Extracted significant features per instance. c: the learned Representation.

In figure 20a) the variability of representation for different runs of the learning algorithm is demonstrated caused by the random initialization of the LBG–algorithm. The learned representations for different p_2 is shown in figure 20b). The parameter p_2 determines the fraction of features needed to be present in the training data to define a significant feature. The change of representation for different size of the training set is demonstrated in figure 20c).

7.1.2 Learning with automatic landmark definition

To avoid the manual generation of ground truth we made use of motor controlled feedback. Our aim is the construction of training data in which a certain object is shown under changing conditions like different background and different illumination but without change of the position of the landmarks. Then we can simply apply our learning algorithm using a rectangular grid to this data.
We put a can on a rotating plate and changed background and lighting conditions in a sequence of pictures (see figure 21). The whole generation of training data just took about 30 seconds. For the generation of ground truth for frontal faces we recorded a sequence of pictures in which a person is sitting fixed on a chair. Illumination and background is changed as for cans (see figure 22). To extract representations for different scales we simply apply the learning algorithm to the very same pictures of the different sequences scaled accordingly.

Figure 19: 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 banana wavelets.

7.2 Matching

Table 3 gives the results for various matching tasks, the location of cans and faces in scenes of different complexity. In row one to four the matching with banana wavelets is compared to the matching with bunch graphs as described in [35, 18]. We tested both approaches on two data sets (row 1 and 3 gives the results with the approach described here, row 2 and 4 gives the results for the bunch graph matching). The first set (set 1) contains frontal faces with very controlled illumination in front of a homogenous background (column 7 gives information about the background, h = homogenous, n.h. = non homogenous) . The faces vary in size between 50 and 100 pixels (column 4) and there is a modest pose variation (column 5 & 6). To handle the size variation we do matching with two graphs labeled with banana wavelets resp. two bunch graphs. Both approaches have comparable performance, but the matching for the banana approach is faster. More interesting are the results for a more complex task (set 2). Figure 23 shows some examples of matches and mismatches on this data set. The size variation of the faces is between 15 and 100 pixel. The pose and illumination is much less controlled and the background is non homogenous for most of the pictures, therefore this data set represents a very hard task. Row 3 and 4 give the results for the matching with bananas and row four gives the results for the bunch graph matching. We see a big gap of performance, the bunch graph matching found 46% of the faces but the matching with banana wavelets 75%.

Match Results														
	Data						Repres.		Trafo			perf.		
	object	nb.	size	rot pl	rot dp	bg	nb. reps	rep	mode	approx	sec.	sec. match	perf.	
1)	faces	100	50-100	$\pm 15^o$	$\pm 15^o$	h.	2	a.g.	ban	v.	1.4	1.8	95%	
2)	faces	100	50-100	$\pm 15^o$	$\pm 15^o$	h.	2	m.g.	bunch		1.3	3.9	97%	
3)	faces	100	15-100	$\pm 30^o$	$\pm 30^o$	n.h.	3	a.g.	ban	v.	1.4	2.6	75%	
4)	faces	100	15-100	$\pm 30^o$	$\pm 30^o$	n.h.	3	m.g.	bunch		1.25	5.8	46%	
5)	cans	60	$\pm 10\%$	$\pm 10^o$	$\pm 10^o$	n. h.	1	m.g.	ban.	n.a.	21	0.3	95%	
6)	cans	60	$\pm 10\%$	$\pm 10^o$	$\pm 10^o$	n. h.	1	m.g.	ban.	a.	4.6	0.3	95%	
7)	cans	60	$\pm 10\%$	$\pm 10^o$	$\pm 10^o$	n. h.	1	m.g.	ban.	v.	1.0	0.5	95%	
8)	cans	60	$\pm 10\%$	$\pm 10^o$	$\pm 10^o$	n. h.	1	a.l.	ban.	v.	3	0.5	84%	

Table 3:

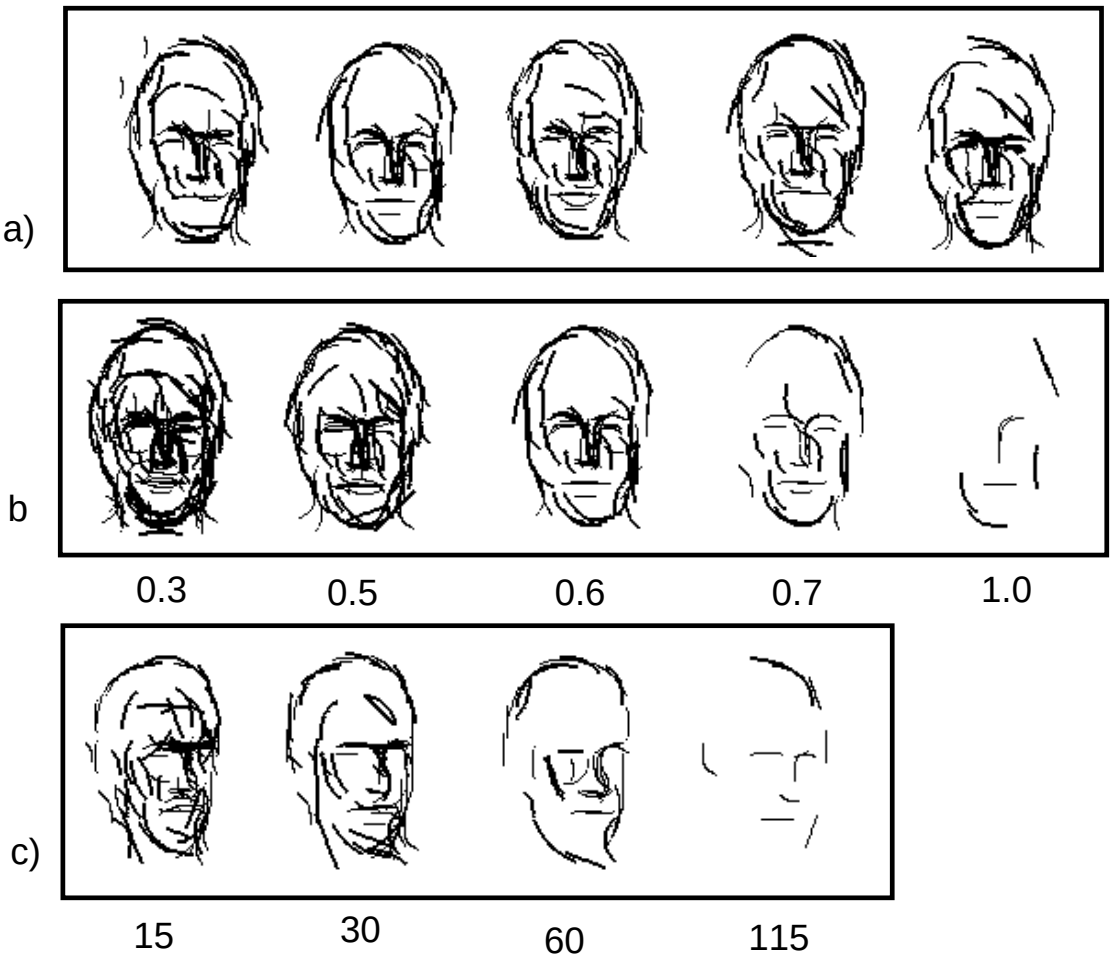

Figure 20: Representations learned with different parameters. a: Different representations caused by random initialization of the LBG-algorithm. b: Variation of p_2, the number of features which have to be in a cluster to call its centroid significant. c: Variation when the size of the training data is varied from 15 to 115 examples.

7.3 Discrimination: The False Positive Test

We applied our representation to the problem of finding a face and classifying it into the classes frontal face and non frontal. Our test set consists of 100 pictures generated from a face finder based on color and disparity information developed by Hartmut Neven [24]. It consists of 75 non–frontal faces (especially hands found by the color detector or faces looking rotated in plane or depth) and 25 frontal faces. The size of the faces varied between 30 and 80 pixel. Our system rejected 67 non frontals correctly by identifying 22 frontals correctly. 3 frontal faces were not found and 8 non frontals were characterized as frontals by the system.

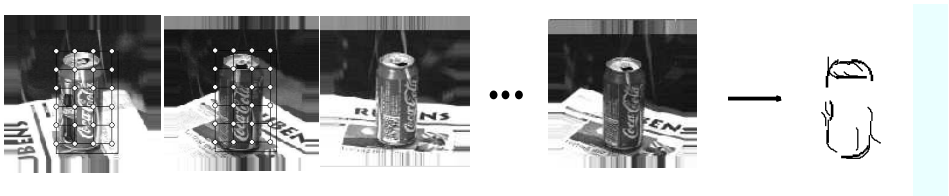

Figure 21: Automatical generation of ground truth for cans. i–iv: Rotated Cans on a rotating table with varying illumination (note the shadow of the can). i,ii: Rotated cans with rectangular grid. v) Learned Representation.

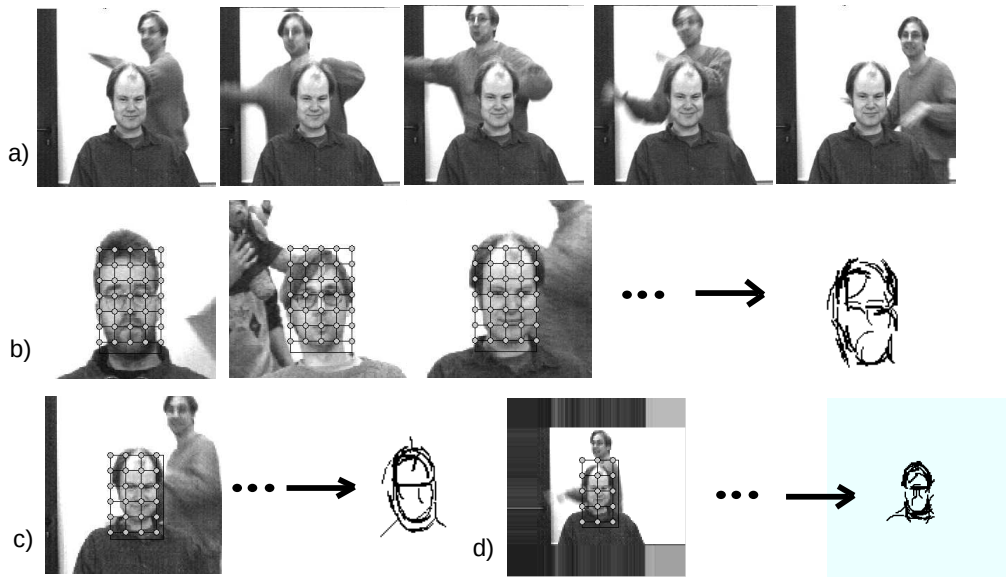

Figure 22: Learned representations for frontal faces with automatically generated ground truth. a: One of the three sequences with a persons face fixed, but with different background and illumination. b: Positioning of the grid learning a representation for largest size on one example for each sequence. The faces have approximately same position and size, therefore the nodes of the fixed grid represent comparable features. c: Representation for medium size. d: Representation for small size.

8 Comparison with other systems

Comparison with earlier versions of Elastic Graph Matching: In earlier versions [21, 35, 18] was based on the concept of "jets"[5, 21] which were used as labels of the graph. A jet gives a local description of an image at a certain pixel position. It is a vector, with its coefficients representing Gabor wavelet responses of different orientation and frequency[7] at a certain pixel position. the norm of the jets is set to one by dividing each coefficient by the norm of the array of Gabor wavelet responses. This normalization ensures the jet's independence of the average grey level intensity of an image.
The concept of jet is one possible formalization of our *a priori* principle P0 (Locality) which enables us to handle landmarks of different localities separately. In the locality of jets we see an conceptual advantage compared to systems based on non local features like, e.g., the principal components of a whole image [34][8]. As a problem of jets we criticise the *mixing of features within one jet.* E.g., if the top of the head occurs in a certain image in front of a textured

[7] Our standard settings were 5 levels of frequency and 8 orientations.
[8] Hancock et. al. [11] suggests for the problem of face recognition it is easier for our localized jet features to deal with varying background or symmetry changes than for a PCA–based approach.

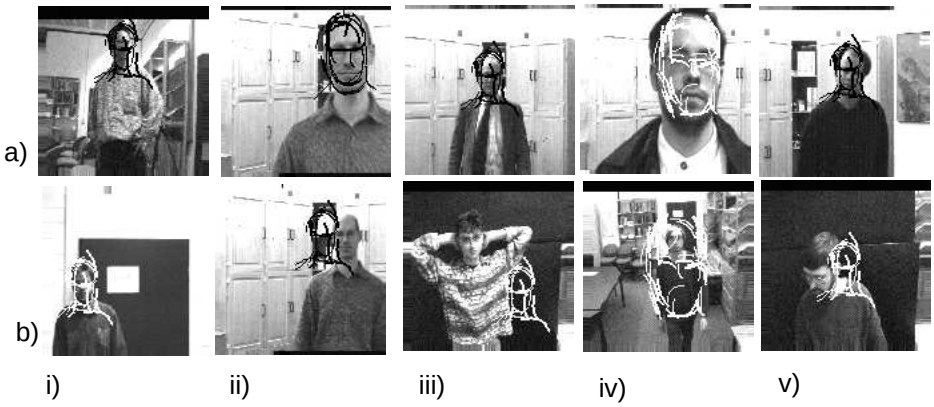

Figure 23: Good matches and mismatches for the set 2.a i–v, b,i) good matches. b,ii-iv) Mismatches. In b,ii) the algorithm found a face like form in the picture. In b,iii) the learned form with shoulders did not fit to the person's face with his hands behind the head. In b,iv) the face is too small. In b,v) the rotation of the head is too large.

background the texture of this background is inherently part of the corresponding jet coefficients and its separation is a non trivial task [30]. For learning it is advantageous to represent both qualities separately, then the convex contour of the head can be recognized as important feature compared to the varying background representing an accidental feature. Our learning algorithm is able to separate important and accidental features based on the quality of banana wavelets to represent structure of the background *separately* from the structure of the head. As an additional advantage compared to the application of Gabor wavelets in [21, 35, 18] we remark that the concept of sparseness found a more convincing and consistent realization in our banana wavelet approach. Already the representation of an image by Gabor wavelets of different orientation and frequency leads to an increase of data: instead of one grey-level image we have the same amount of data for each pair of orientations and frequency. The expansion of the feature space can be found analogously in the visual cortex of vertebrates [14, 26] and in this aspect is fundamental differing from PCA approaches [34, 31] in which the data space is reduced in early stages of processing. Our banana wavelet approach enhances the data expansion of the earlier versions based on Gabor Wavelets by adding the qualities curvature and size. Differing from the feature binarity request in our formulation of sparseness (see introduction) the similarity function in [21, 35, 18] was highly dependent on the exact value of the filter responses. In the banana approach we substituted the filter response value by binary features which are present or absent.

In [35] the idea of "bunch graphs" is introduced to represent the variation of a certain view of an object class. In a bunch graph a landmark is labeled by a "bunch of jets", each jet representing an instance of this landmark in form of a jet extracted from pictures of different persons at the corresponding landmark. E.g., in a bunch graph a left eye of a frontal face is represented as a set of jets extracted from the left eye of frontal faces of different persons. The bunch graph idea is successfully applied to other object recognition problems, e.g., the discrimination of hand gestures [33] and pose estimation [18]. In [35] each landmark was described by approximately 70 jets, each containing 40 complex values. Especially to represent contour edges hitting the background a large amount of jets would be necessary to cover all possible combinations of this edge and the different backgrounds. With our banana approach we can reduce the data needed to represent a landmark to a few banana wavelets. Furthermore in [35] and [33] the creation of an object representation is very time consuming, because for each view of an object an object dependent grid has to be defined and the landmarks has to be positioned manually for the pictures used to create the bunch graphs. First steps towards an automatic generaition of an object representation (based on the bunch graph approach) are made in [18] where important nodes and suitable jets were learned utilizing the principles P0, P1 and P2. But still a lot of manual intervention for the generaition of ground truth was necessary. In contrast to these manual interventions here we introduced methods to learn a representation autonomously. Anyway, there might be situations in which a landmark can not be sufficiently described by only one combination of banana wavelets, e.g., in case of an eye with and without glasses or a chair with and without armrests. In those cases a *bunch graph of combination of banana wavelets* might be more appropriate to represent this landmark. But still a much smaller amount of data than in the "jet–bunch" approach should be sufficient.

Comparison with other object recognition systems: There exists a large variety of object recognition systems utilizing different amount of *a priori* knowledge. As one extreme we refer to systems which apply learning algorithms directly to the grey level pictures. These algorithms can be called "neural" like backpropagation or RBF–Networks [12, 28] or strategies of classical pattern recognition like Bayesian estimation methods [9]. These systems apply a very small amount of *a priori* knowledge and theoretical statements about their general applicability can be made, presupposing that the number of free variables of those systems grows to infinity [13, 9]. Unfortuntately generalization and learning time is a fundamental restriction of those systems. 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 [10]. The variance problem can be reduced by choosing a suitable preprocessing of the data reducing the search space, but this manual intervention destroys the general applicability by leaving the choice of a suitable preprocessing to the creator of the system. As an extreme on the other side of the bias/variance dilemma there exist a large variety of systems putting a huge amount of *a priori* knowledge into their system. As only one example in [15] 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 those systems in another surrounding. Having in mind a system in which a large amount of different objects can be represented and recognized in complex scenes we see our systems in the middle of the two extremes mentioned above. Our system explicitly makes use of *a priori* knowledge, but because of the generality of our *a priori* assumptions we aim to avoid a too narrow specialization of our system.

In [34, 31] an object recognition system based on principle component analysis (PCA) applied to the grey level picture is introduced. PCA leads to a fast *reduction* of data by a *linear* transformation. Taking the human visual system as a model of the most successful vision algorithm existing so far there are no hints for a *data compression* but a lot of hints for a *data spreading* in the first stages of visual processing [14, 8, 26]. We assume that this data spreading is needed to allow a sparse coding which inherently has a lot of advantages for the processing of visual information (see section 9). Furthermore it seems that *non–linear* transformations play an important role in visual processing [26].

Cootes et. al. [4] introduce an object recognition system which is also based on line segments, they learn the variation of an object class by applying PCA to different instances of an object class. 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. The representation of objects has to be defined manually. For learning the variation of an object class this representation has to be positioned manually for the different examples. A similarity between this and our system we see in the restriction of local lines to describe objects. As an advantage of our system we regard the locality and metric organization of our features which enable an *autonomously* learning of our representations of objects.

9 Conclusion and Outlook

In section 7 we illustrated the applicability of our system to a wide range of difficult problems in vision. For the problem of face finding we demonstrated a significant improvement of performance compared to the older system based on jet bunch graphs [35]. Our system is able to learn an effective representation of a wide range of objects autonomously, we chose cans (artificial, rigid) and faces (natural, slightly deformable) as two very distinct examples. For faces we demonstrated that our representation is able to cover pose differences and even the fine differences of faces of males and females. We assume that any object locally describable by line segments can be represented with our system. The class of representable objects principally covers therefore most of the objects humans have to deal with. Nevertheless we have also shown that our system is far away from being as powerful as the human visual system, but we like to argue here that it might be seen as an intermediate step towards a system with even better performance.
Among others Biederman [3] suggests that it is not a single feature which is important in the representation of an object but the *relations* of features. At the present stage of our approach only metric relations expressed in the graph structure are represented. Banana wavelets represent features with certain complexity which describe suitable abstract properties (orientation, curvature). In future work we aim to utilize this abstract properties to define Gestaltrelations between Banana wavelets like parallelism, symmetry or connectivity. These abstract properties of our features enable the formalization of these relations. Furthermore sparse coding leads to an decrease of the number of possible relations for an object description (only the relations between the few "present" features have to be taken into account). Therrefore the reduction of the space of relations and the describable abstract properties of these features makes the space of those relations *manageable*. In the *reduction of the space of relations* we see an additional advantage of sparse coding not mentioned in the literature so far.
In our approach the correspondence problem must be solved before learning can start. In section 7 we used motor controlled feedback to reduce the amount of manual intervention for the generation of ground truth. In future work we like to apply a robotor arm to position landmarks correctly. By moving the robotor hand with an object in front of an non homogenous background in a surrounding with varying illumination and background and utilizing the knowledge of the actual position of the roboter hand to solve the correspondance problem we can easily create a large amount of training examples automatically. Another mechanism supporting the generation of ground truth can be the *continuity of movement*. Following an object which is moving continuously is a much easier task than finding an object without any *a priori* knowledge. Even a "primitive" representation of an object may solve this task and may be utilized for the generation of ground truth used as training data for the learning of a more sophisticated representation. In [22] the "jet–bunch approach" is already successfully applied to the problem of tracking a moving object.
As an important open question of the object recognition system described here remains its extension from the representation of different 2D–views to a powerfull representation of the complete three–dimensional object. In [18, 35] faces of different sizes and rotated in depth within a range of 180 degrees are represented by the jet–bunch approach, applying 15 different bunch graphs for three sizes (small, medium and large) and five poses (profile left, half profile left , frontal, half profile right and profile right). In [33] 10 different hand gestures are represented by bunch graphs. Analogously we could apply our banana wavelet representation by learning different representation for different sizes (as already done in some of our simulations) and different poses. In [35, 18] an object is simply represented as a loosely connected set of 2D-views of the object. A more structured connection of 2D–views is defined in [16]. In this approach the two dimensional views are connected by complex arrangements of line segments, called geons [3]. These geons are presupposed as *a priori* knowledge and mediate between 2D and 3D representation. We hope that by formalizing Gestaltrelations between banana wavelets (see below) we can *learn* geon–like structures by looking at statistical relevant relations or in terms of [3] by extracting non–accidental features.
As a further improvement we intend to introduce instead of the constant metric (6) task dependent metrics. In our similarity function (13) we simply look at the filter responses of the banana wavelets in our representation but we do not distinguish between the different qualities like curvature, size or orientation. E.g., to tell the top of a head (expressed by a banana wavelet with horizontal orientation bent downwards) from a horizontal door beam it is not the orientation or size which is important but only the curvature. In our actual representation also the door beam achieves high values because its horizontal orientation leads also to a strong response of the banana wavelet representing the top of the head, i.e., it shares the quality horizontal orientation with the door beam. For other tasks, e.g., pose discrimination the top of the head is not important at all and only the inner face features are important. In this case

not only certain qualities of a banana wavelet (like curvature and size) are insignificant but the importance of the whole banana wavelet has to be reduced in the similarity function applied for this task. In [17] an algorithm for the learning of metrics is introduced which is based on the principles P0, P1 and P2. This algorithm is applied within the frame of the "bunch–jet" approach but in future work we intend to adapt this algorithm to the object recognition system described in this paper.
In the long run we aim to a system equipped with a small number of mechanisms of small complexity (like following moving objects, shifting objects with its arm and coordinating the camera according to the movement) to initiate learning strategies representing more complex interrelations underlying the system's experience. We think that the system described in this paper is a very promising basis and an important step towards this challenging goal.

10 Acknowledgement

We like to thank Laurenz Wiskott, Michael Pötzsch and Jan Vorbrüggen for fruitful discussion. Furthermore we like to thank Thomas Maurer for solving the integral in equation (1).

A Appendix

A.1 A Banana Wavelet expressed by Matrix Operations

$$B^{\vec{b}}(x,y) := \gamma^{\vec{b}} e^{-\frac{f^2}{2}\left(\frac{x_G^2}{\sigma_x^2}+\frac{y_G^2}{\sigma_y^2}\right)} \cdot \left(e^{if x_F} - DC^{\vec{b}}\right) \tag{14}$$

with

$$\vec{x}_F = \mathcal{M}_c\left(\mathcal{M}_\alpha \vec{x}\right)$$
$$\vec{x}_G = \mathcal{M}_s \vec{x'}.$$

and

$$\mathcal{M}_\alpha = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix},$$
$$\mathcal{M}_c(\vec{x}) = \begin{pmatrix} x + cy^2 \\ y \end{pmatrix},$$
$$\mathcal{M}_s = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{s} \end{pmatrix},$$

A.2 A Path $\vec{p^b}$ expressed by Matrix Operations

$$\vec{t} = \begin{pmatrix} 0 \\ t \end{pmatrix}$$
$$\vec{t'} = \sigma \frac{1}{f} \mathcal{M}_{\frac{1}{s}} \sigma_y \vec{t}$$
$$\vec{p^b}(t) = \mathcal{M}_{2\pi-\alpha}\left(E\vec{t'} + \mathcal{M}_{-c}(\vec{t'})\right)$$

with

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and the other matrices as in A.1.

References

[1] H. B. Barlow, "Possible principles underlying the transformation of sensory messages," in *Sensory Communication*, W. A. Rosenblith, Ed. pp. 217–234, MIT 1961.

[2] E. B. Baum, J. Moody, F. Wilczek, "Internal Representation for associative memory," *Biological Cybernetics*, pp. 217–228, 1988.

[3] I. Biederman, "Recognition by Components: A theory of human image understanding", Psychological Review, Vol: 94, No. 2, 1987.

[4] T.F. Cootes, C.J. Taylor, J.Graham, "Active Shape Models–Their training and Application", Computer Vision and Image Understanding, Vol. 61, No.1 1995.

[5] J.D. Daugman, "Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1169–1179, 1988.

[6] A. Dobbins, S. Zucker, M. S. Cynader, "Endstopped neurons in the visual cortex as a substrate for calculating curvature," *Nature*, vol. 329, pp. 438–441, 1987.

[7] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America*, vol. 4, no. 12, pp. 2379–2394, 1987.

[8] D. Field, "What is the Goal of Sensory Coding?," *Neural Computation*, vol. 6, no. 4, pp. 561–601, 1994.

[9] K. Fukunaga, "Introduction to statistical pattern recognition (2nd ed)", Academic Press, Boston 1990.

[10] S. Geman and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, pp. 1–58, 1995.

[11] P. Hancock, V. Brucd, A.M. Burton, "A comparison of two computer–based face identification systems with human perception of faces", submitted to Vision Research.

[12] J. Hertz, A. Krogh, R.G. Palmer, "Introduction to the Theory of Neural Computation", Addison–Wesley 1991.

[13] K. Hornik, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, Vol. 2, pp. 359–366.

[14] D. H. Hubel and T. N. Wiesel, "Brain Mechanisms of Vision," *Scientific American*, vol. 241, pp. 130–144, 1979.

[15] S.S. Intille, A.F. Bobick, "Closed-World Tracking", In Proc. of the Int. Conf. Computer Vision, June 1996.

[16] E. Kefalea, O. Rehse, C. v.d. Malsburg, "Object Classification based on Contours with Elastic Graph Matching", submitted to 3rd Int. Workshop on Visual Form, 1997, Capri Italy.

[17] N. Krüger. "Learning Weights in Discrimination Functions using a priori Constraints," in *Mustererkennung*, G. Sagerer *et al.*, Ed. Springer Verlag, 1995, pp. 110–117.

[18] N. Krüger, M. Pötzsch, C. v.d. Malsburg, "Determination of Face Position and Pose with a Learned Representation based on Labeled Graphs," Technical Report IR–INI 96–03, 1996.

[19] N. Krüger, G. Peters, M. Pötzsch, "Utilizing Sparse Coding and Metrical Organization of Features for Artificial Object Recognition," in progress, 1996.

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

[21] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, W. Konen, "Distortion Invariant Object Recognition in the Dynamik Link Architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, 1992.

[22] T. Maurer, C. von der Malsburg, "Tracking and Learning Graphs and Pose on Image Sequences of Faces", Proceedings of the 2d Int. Conf. on Automatic Face- and Gesture-Recognition 1996.

[23] R. Millman and G. Porter, "Elements of Differential Geometry," Prentice–Hall, 1977.

[24] H. Neven, personal communication 1996.

[25] B. Ohlshausen and D. Field, "Sparse Coding with an overcomplete basis set: A strategy employed by V1?,"

[26] M.W. Orram and D.I Perret, "Modeling Visual recognition from neurobiological Constraints", Neural Networks (1994) Vol.7, pp:945–972.

[27] G. Palm, "On associative memory," *Biological Cybernetics*, vol. 36, pp. 19–31, 1980.

[28] J. Pauli, "Learning Operators for View dependent Object Recognition", Proceedings of the BMVC 1996.

[29] G. Peters, "Lernen lokaler Objektmerkmale mit Bananenwavelets," Technical Report IR–INI 96–09, Diploma Thesis, 1996.

[30] M. Pötzsch, N. Krüger, C. von der Malsburg, "Improving Object Recognition by transforming Gabor Filter Responses", Network: Computation in Neural Systems 7, 1996.

[31] D. Swets and J. Weng, "SHOSLIF-O:SHOSLIF for Object Recognition and Image Retrieval (Phase 2)", Technical Report. CPS 95–39, Michigan State University, Department of Computer Science, 1995.

[32] K. Tanaka, "Neuronal mechanisms of object recognition", *Science*, vol. 262, 1993.

[33] J. Triesch and C. von der Malsburg, "Robust Classification Of Hand Postures Against Complex Backgrounds", Proceddings of the second international Conference on Automatic Face and Gesture Recognition, Vermont 1996.

[34] M. Turk and A. Pentland., "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991.

[35] L. Wiskott, J.-M. Fellous, N. Krüger, C. von der Malsburg, "Face Recognition and Gender Determination", *Proceedings of the International Workshop on Automatic Face- and Gesture recognition*, Zürich 1995.