LETTER Communicated by Christoph von der Malsburg

Combining Feature- and Correspondence-Based Methods for Visual Object Recognition

Günter Westphal
guenter.westphal@mobile-vision-systems.de
Mobile Vision Systems, Blücherstrasse 19, D-46397 Bocholt, Germany

Rolf P. Würtz
rolf.wuertz@neuroinformatik.rub.de
Institut für Neuroinformatik, Ruhr-Universität Bochum, D–44780 Bochum, Germany

We present an object recognition system built on a combination of feature- and correspondence-based pattern recognizers. The feature-based part, called *preselection network*, is a single-layer feedforward network weighted with the amount of information contributed by each feature to the decision at hand. For processing arbitrary objects, we employ small, regular graphs whose nodes are attributed with Gabor amplitudes, termed *parquet graphs*. The preselection network can quickly rule out most irrelevant matches and leaves only the ambiguous cases, so-called model candidates, to be verified by a rudimentary version of elastic graph matching, a standard correspondence-based technique for face and object recognition. According to the model, graphs are constructed that describe the object in the input image well. We report the results of experiments on standard databases for object recognition. The method achieved high recognition rates on identity and pose. Unlike many other models, it can also cope with varying background, multiple objects, and partial occlusion.

1 Introduction

This letter is concerned with the task of invariant visual object recognition. The term *recognition* refers to the decision about an object's unique identity and requires discrimination between object identities and involves generalization across minor shape changes, as well as physical translation, rotation, and so forth (Palmeri & Gauthier, 2004).

Computational models for invariant visual object recognition come in two main types: feature-based and correspondence-based. Both start with the extraction of features, which are chunks of information gathered from an image that allow purposefully mapping that image into some feature space of a dimension that is lower than the original pixel grid. Popular examples of image features are color, shape, and texture elements. In feature-based

recognition systems, invariance over parameter variation is achieved for each feature separately, and different parameter values are combined with a logical disjunction. Each feature detector passes its assessment of feature presence in the image to master units that become activated if at least one of its contributors has observed its reference feature. Master units thus represent parameter-invariant feature types. Object recognition is achieved by comparing the list of activated master units to lists stored for known objects and picking the best match. The characteristic of this approach is that information on the original parameter values, such as position, scale, and especially the spatial arrangement of (local) features, is discarded. Examples of feature-based systems include the Neocognitron (Fukushima, Miyake, & Ito, 1983); the ones by Edelman (1995), Murase and Nayar (1995); SEEMORE Mel (1997); Schiele and Crowley (2000); VisNet (Elliffe, Rolls, & Stringer, 2002); and those by Wersing and Körner (2003), and Serre, Oliva, and Poggio (2007). There is little doubt that the mammalian visual system operates to some extent in a feature-based manner. This assumption is backed by psychophysical experiments in which signal propagation times and response times have been measured (see, e.g., Oram & Perret, 1994; Logothetis & Pauls, 1995). As a model for object recognition in the brain, feature-based methods can be implemented as feedforward networks, which would account for the amazing speed with which these processes can be carried out relative to the slow processing speed of the underlying neurons (Thorpe, Fize, & Marlot, 1996; Thorpe & Thorpe, 2001). This assumption is in accord with the available psychophysical data, but it can be doubted that object recognition in the brain is entirely feature-based, as these models encounter problems when confronted with more sophisticated recognition tasks, such as images with structured backgrounds, multiple objects, and occluded objects. As especially the spatial arrangement of features is discarded, these techniques are prone to the confusion of objects that agree in features but differ in their spatial arrangement, scale, or orientation. It has, however, been argued that nonambiguous representations can be achieved through introduction of combination-coding units (see, e.g., Mel, 1997; Riesenhuber & Poggio, 2000), but the unlimited introduction of such cells inevitably leads to a combinatorial explosion that would soon exhaust the number of cells available (Rosenblatt, 1962; Tsotsos, 1990; von der Malsburg, 1999).

In correspondence-based models, object views are represented as ordered arrays of local features. For instance, in elastic graph matching (von der Malsburg, 1988; Lades et al., 1993; Wiskott, 1995; Wiskott, Fellous, Krüger, & von der Malsburg, 1997), object views are represented by model graphs, two-dimensional graphs in the image plane, whose nodes are labeled with local image features, usually the complex responses of a set of Gabor filters, and whose edges express relations between two nodes. In elastic graph matching, object models, represented by their associated model graphs, are matched with an input image by solving the correspondence problem, that is, through establishing an organized set of point-to-point

correspondences between the input image and the object model. Further examples of correspondence-based systems include Shapiro and Haralick (1981), Bunke (1983), Ullman (1989), Würtz (1997), and von der Malsburg and Reiser (1995). These techniques usually encounter problems when applied to larger repertoires of general objects, as object models are required to be dynamic with respect to both shape and attributed features in order to cope with object variations like changes in pose, illumination, and so on. Graph-like structures, and model graphs in particular, inherently fulfill this requirement. In other words, they allow for compositionality, defined by Bienenstock and Geman (1995) as the ability to construct mental representations, hierarchically, in terms of parts and their relations. What needs to be specified are the elementary parts and the rules of composition. Then mental representations can be built starting from elementary features, composing them to ever more complicated ones, until the complexity required for the task is reached. The necessary depth of this composition hierarchy influences processing time, with simple features being recognized nearly instantaneously and each level of composition requiring extra time. It has been shown psychophysically that recognition tasks that explicitly require composition take distinctly longer than those for simpler objects. For instance, in Treismann and Gelade (1980), human subjects were presented combinations of green and red crosses and circles. Afterward, the subjects were asked to give statements like, "I have seen a red cross in the left half of the screen and a green circle in the right half." If the presentation was long enough, this was an easy task, but when the presentation times were reduced below 50 milliseconds, the assignment of color to the cross or circle dropped to chance level. This finding supports the original assumption that the construction of a suitable representation that correctly integrates the visual features "cross," "circle," "red," and "green," a representation in which, more generally, the binding problem (von der Malsburg, 1981, 1999) has been solved, takes more time than the mere detection of uncombined features. Another aspect in favor of correspondence-based processing is that the recognition of object identity is usually not sufficient, and in order to do anything useful to an object, like grasping or manipulating, the locations of objects and object parts matter (Becker et al., 1999).

Different time requirements for different recognition tasks can be explained by the necessity to construct mental representations. They would not occur naturally in feedforward architectures implementing a simple stimulus-response scheme as, for example, in Pitts and McCulloch (1947), Rosenblatt (1962), Fukushima et al. (1983), or Serre et al. (2007).

However, constructing object representations while recognizing objects, and finding a solution to the binding problem in general, is a laborious task that, in computer vision, has most often been disregarded in favor of employing object models tailored to specific object categories like faces in frontal pose. There is thus a jigsaw piece missing in the picture of correspondence-based processing: a purposeful initialization that restricts

it to model images really worth their while—to model candidates, as we will call them from now on. We propose that fast feature-based preprocessing is applied as far as it goes by excluding as many objects as possible and that only ambiguous cases, the model candidates, are subjected to the more accurate correspondence-based processing. The feasibility of this approach has been proven in Westphal (2006). There, the accent was laid on the emergence of model graphs, while in this letter, the benefit of combining feature- and correspondence-based techniques will be highlighted.

This letter is organized as follows. In section 2, a single-layer neural network is introduced that allows rapidly selecting a relatively small subset of model candidates from a much greater set of stored object views in a strictly feature-based fashion. Throughout, model candidates are considered to be model images that are supposed to contain the same object identity in a similar pose as the current input image. Section 3 is concerned with the correspondence-based verification of model candidates, using a rudimentary version of elastic graph matching. In section 4, the proposed combination of feature- and correspondence-based methods is applied to the task of visual object recognition and tested on publicly available standard databases. Finally, section 5 gives a summary and an outlook on further research.

2 Feature-Based Preselection of Model Candidates

In this section we present a neural network for preselection of model images, so-called model candidates, for an input image at hand prior to their correspondence-based verification. This network is called the *preselection network* (Westphal, 2006). Its design is motivated by the well-established finding that individual object-selective neurons tend to preferentially respond to particular object views (Perret et al., 1985; Logothetis & Pauls, 1995). The preselection network's output neurons take the part of these view-tuned units (Riesenhuber & Poggio, 2000).

The preselection network (see Figure 3) is a single-layer feedforward neural network with sparse connectivity. In the network's input layer, position-invariant feature detectors submit their assessments whether their reference feature is present in an image to dedicated input neurons. As (local) image features, we chose small regular graphs, so-called parquet graphs, whose nodes are labeled with Gabor features. The output layer comprises one neuron for each model image. Synaptic weights are chosen such that the network conforms to Linsker's infomax principle (Linsker, 1988). That principle implies that the synaptic weights in a multilayer network with feedforward connections between layers develop, using a Hebbian-style update rule (Hebb, 1949), such that the output of each cell preserves maximum information (Shannon, 1948) about its input. Subject to constraints, the infomax principle thus allows directly assigning synaptic weights. This network setup, in conjunction with the application of

the winner-take-most or winner-take-all nonlinearity as decision function, implements a weighted majority voting scheme (Lam & Suen, 1997) that allows the desired preselection of model candidates.

The selection of model candidates is based only on feature coincidences in the image and model domain. As their spatial arrangement is disregarded, false positives are frequent among them. To rule them out, similar spatial arrangement of features will be asserted for the model to be selected in the correspondence-based verification part (see section 3).

In this letter, we present a streamlined version of the preselection network for the task of invariant visual object recognition. A more elaborated variant is given in Westphal (2006).

2.1 Gabor Features. Gabor features are well suited for image representation because of their information theoretical properties (Linsker, 1988; Olshausen & Field, 1996) and their biological relevance (Hubel & Wiesel, 1962; Jones & Palmer, 1987). These features describe local texture in an image. They are the complex responses of a set of Gabor filters applied to an image at a position of interest. Gabor filters have the form of a plane wave restricted by a gaussian envelope:

$$\psi_{\underline{k}}(\underline{x}) = \frac{k^2}{\sigma^2} \exp\left(-\frac{k^2 x^2}{2\sigma^2}\right) \left[\exp\left(i\underline{k}^\top \underline{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right]. \tag{2.1}$$

Fourier-transformed Gabor filters take the form of gaussians in the frequency domain, that is, they are bandpass filters. A Gabor wavelet transform of an image I at a point $\underline{x}$ with respect to a wave vector $\underline{k}$ is given by the convolution with the Gabor kernel, the domain of integration being the image plane:

$$\mathcal{I}_{\underline{k}}(\underline{x}) = \int_{\mathbb{R}^2} I(\underline{x}')\psi_{\underline{k}}(\underline{x} - \underline{x}')\, d^2x'. \tag{2.2}$$

For actual calculations, a discrete and finite subset of wave vectors is necessary. By rotating and scaling the wave vector $\underline{k}$, a whole family of Gabor functions can be derived. Each of them is parameterized in terms of its orientation ϕ_l and frequency k_m:

$$\underline{k}_{m,l} = k_m \cdot \begin{pmatrix} \cos\phi_l \\ \sin\phi_l \end{pmatrix}. \tag{2.3}$$

The finite set of filters is chosen such that the direction space is sampled homogeneously, and the frequencies are sampled geometrically:

$$\phi_l = \frac{\pi \cdot l}{L} \quad \text{with} \quad l \in \{0, \ldots, L-1\}. \tag{2.4}$$

$$k_m = \frac{k_{max}}{(k_{step})^m} \quad \text{with} \quad m \in \{0, \ldots, M-1\}. \tag{2.5}$$

The remaining parameters are chosen according to Lades et al. (1993) and Wiskott (1995):

$$k_{step} = \sqrt{2}, \qquad k_{max} = \frac{\pi}{2}, \qquad L = 8, \qquad M = 5, \qquad \sigma = 2\pi.$$

The complex responses of this set of Gabor filters at a given location $\underline{x}$ in an image constitute a so-called (Gabor) jet (Lades et al., 1993). These are vectors of $M \cdot L$ complex numbers. In this letter, only their amplitudes $a_{\underline{k}_{m,l}}$ are used (see equation 2.6). They are a model for complex cells in the visual cortex and yield some local shift invariance, which is very useful for matching. Whenever possible, we omit the position $\underline{x}$ and write $\mathcal{J}$ instead of $\mathcal{J}(\underline{x})$:

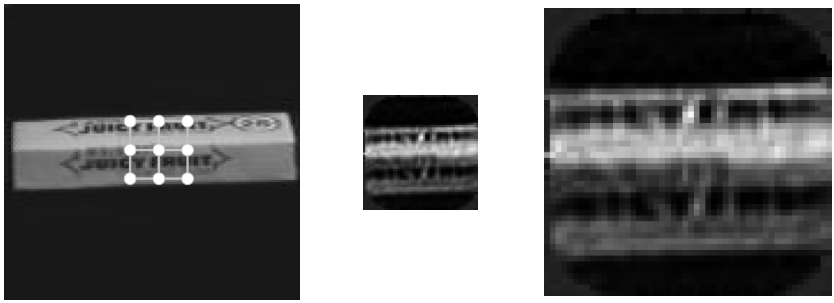
$$\mathcal{J}(\underline{x}) = \left(\left|\mathcal{I}_{\underline{k}_{m,l}}(\underline{x})\right|\right)_{0 \le m < M, 0 \le l < L} =: \left(a_{\underline{k}_{m,l}}\right)_{0 \le m < M, 0 \le l < L}. \tag{2.6}$$

Two jets may be compared with so-called similarity functions, which usually map two given jets into the interval [0, 1]. A number of these functions have been proposed (Lades et al., 1993; Würtz, 1995; Wiskott, 1995). In this letter, we exclusively use the measure of similarity based on the amplitudes of the filter responses, which is implemented as the normalized scalar product between the amplitude vectors:

$$s_{abs}(\mathcal{J}, \mathcal{J}') = \frac{\sum_{m,l} a_{\underline{k}_{m,l}} \cdot a'_{\underline{k}_{m,l}}}{\sqrt{\sum_{m,l} a_{\underline{k}_{m,l}}{}^2} \cdot \sqrt{\sum_{m,l} a'_{\underline{k}_{m,l}}{}^2}}. \tag{2.7}$$

This measure of similarity allows smooth similarity potentials with fairly wide maxima.

2.2 Parquet Graphs. The feature-based part can work with any convenient feature type. A successful application employing color and multiresolution image information is presented in Westphal and Würtz (2004). Also, higher features known to exist in the brain, such as end-stopped cells, key points, shape, and curvature-selective cells, can be incorporated. For the current combination of feature- and correspondence-based methods, we chose small, regular graphs labeled with Gabor features. We call them *parquet graphs*, inspired by the look of ready-to-lay parquet tiles. They work as simple feature detectors for preselection and can be aggregated to larger graph entities for correspondence-based processing. Throughout, parquet



(a) (b) (c)

Figure 1: Example of a parquet graph. (a) A parquet graph on the object in the given image. (b) The reconstruction from the parquet graph. (c) An enlarged version of b. All reconstructions in this letter are computed with the algorithm by Pötzsch, Maurer, Wiskott, and von der Malsburg (1996).

graphs consist of $V = 9$ nodes. A parquet graph f will be described with a finite set of node attributes:

$$f = \{(\underline{x}_v, \mathcal{J}_v, b_v) \mid 1 \leq v \leq V\}. \tag{2.8}$$

Each node v is labeled with a triple $(\underline{x}_v, \mathcal{J}_v, b_v)$, where $\mathcal{J}_v$ is a Gabor jet derived from an image at an absolute node position $\underline{x}_v$. In order to make use of segmentation information, it is convenient to mark nodes residing in the background as invalid and exclude them from further calculation. For this purpose, the node attributes comprise the validity flag b_v, which can take the values 0 and 1, meaning *invalid* and *valid*. For the given parameterization of the Gabor features, the horizontal and vertical node distances Δx and Δy are set to 10 pixels. Figure 1 shows an example of a parquet graph. Where appropriate, parquet graphs are simply referred to as features.

A parquet graph describes a patch of texture derived from an image regardless of its position in the image plane. Particularly, this means that the feature positions are irrelevant for the decision as to whether two images contain a similar patch of texture. Later, in the correspondence-based verification part (in section 3), larger graphs are constructed dynamically by assembling parquet graphs derived from earlier model images according to their spatial arrangement. Then relative node positions will become important.

The similarity between two parquet graphs f and f' is given by the average similarity of the jets associated with the nodes with the same index that are valid within both parquet graphs (Würtz, 1997; Shams, 1999):

$$s_{graph}(f, f') = \begin{cases} \left(\sum_{v=1}^{V} b_v b'_v \right)^{-1} \cdot \sum_{v=1}^{V} \left(b_v b'_v \right) \cdot s_{abs} \left(\mathcal{J}_v, \mathcal{J}'_v \right) & \text{if } \sum_{v=1}^{V} b_v b'_v > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2.9}$$

The factors $(b_v b'_v)$ are 1 if both respective jets $\mathcal{J}_v$ and $\mathcal{J}'_v$ are valid and 0 otherwise. These factors assert that only similarities between valid jets be taken into account. If all products become 0, the similarity between the two parquet graphs is 0. It is well worth noting that parquet graphs provide a means to protect from accidentally establishing point-to-point correspondences in that contiguous, topographically smooth arrays of good correspondences are favored over good but topographically isolated ones.

2.3 Feature Detectors. A local feature detector yields a binary decision whether two parquet graphs f and f' match according to a threshold ϑ:

$$\varepsilon \left(f, f', \vartheta \right) = \begin{cases} 1 & \text{if } s_{graph}(f, f') \geq \vartheta \\ 0 & \text{otherwise} \end{cases}. \tag{2.10}$$

Since many parquet graphs are taken from each model image, the total number of extracted parquet graphs is huge, and a method to compress the given data is needed. To this end, a simple variant of vector quantization (Gray, 1984) is used. While learning new objects from model images, every parquet graph gathered from the current model image is compared to all other parquet graphs in the database. The parquet graph at hand is added to the database only if all comparisons yield subthreshold similarity values (i.e., if it represents a novel piece of texture). After learning, we consider the database to contain T parquet graphs f_t.

It is important to know the number of parquet graphs required for coding a growing number and, finally, all possible images. This has two aspects: a geometrical one and one about the structure of natural images. Geometrically, the parquet graphs are feature vectors of dimension $K = L \cdot M \cdot V = 480$, with normed scalar product as similarity. Keeping the similarity between any vector in $\mathbf{R}^K$ and its closest stored parquet graph below ϑ requires covering the hypersphere of unit radius in K dimensions with K-dimensional balls of radius $\sqrt{2(1-\vartheta)}$ with centers on the hypersphere. As the hypersphere has a finite $K-1$-dimensional surface, the number $T(\vartheta, K)$ of required centers is finite for every positive ϑ. If all components are positive (as the Gabor amplitudes used in our case), only

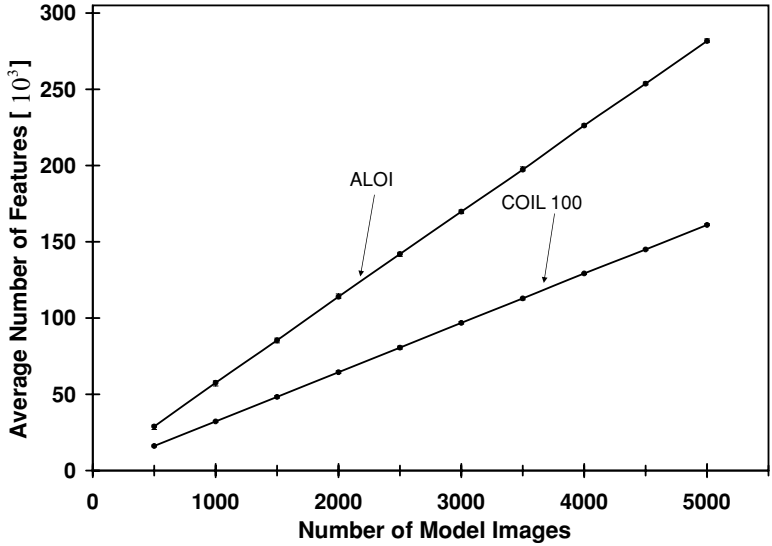


Figure 2: Number of extracted features as a function of the number of model images. The given numbers are the average numbers of extracted features calculated in five test runs. In each test run, the respective number of images was randomly picked from the original image databases. For both COIL 100 and ALOI databases (see section 4.1 for details), the number of features depends linearly on the number of model images with very small variance.

one sector of the hypersphere needs to be covered, and a slight modification of the argument by von Luxburg, Bousquet, and Schölkopf (2004) yields

$$\left\lceil \frac{1}{2\sqrt{2(1-\vartheta)}} \right\rceil^{K-1} \leq T(\vartheta, K) \leq 4 \left\lceil \frac{\pi}{4\sqrt{2(1-\vartheta)}} \right\rceil^{K-1}, \tag{2.11}$$

where $\lceil x \rceil$ denotes the smallest integer larger than x.

The upper bound in equation 2.11 again proves the finiteness, but the lower bound is certainly prohibitive for any useful feature dimensionality if $\vartheta > 0.875$ and the base exceeds 1. This leaves the hope that the number of features required for coding only natural images would be much lower. It can, however, be suspected that an arbitrary instance of a parquet graph will be part of some natural image (think of turning it into a texture by repetition and decorating an object with that texture). If this is correct, the number of stored model features will indeed grow within the calculated bounds when all possible images need to be accounted for. For relatively few model images, Figure 2 shows that the number of extracted features is

linear in the number of model images with a slope roughly proportional to the image resolution. The saturation predicted by equation 2.11 is not observed with the number of images tested in this study.

On the other hand, the total number of extracted features is problematic, because the recognition time will be proportional to it. Therefore, care must be taken to store only the most discriminative features. Following a (rather drastic) reduction from a full codebook to one model feature per model image, a typical recognition rate dropped from 99% to 76% (Westphal, 2006). We are planning to incorporate a better reduction strategy in a future study. The selection of features with high measures of information has proven to be a good approach (Ullman, 2007). In a way, it makes the value of ϑ data-dependent and allows covering relevant regions of the feature space with more examples than others.

Next, a position-invariant feature detector examines whether a (reference or model) feature f_t appears in an image regardless of its position:

$$\tau_t(I) = \begin{cases} 1 & \text{if } \sum_{f \in \mathbb{F}(I)} \varepsilon(f, f_t, \vartheta) \geq 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2.12}$$

with $\mathbb{F}(I)$ being the set of all parquet graphs extracted from the image I. From now on, we use the term *feature detector* only for the position-invariant version.

Due to vector quantization, several features, possibly from different model images, are represented by a single parquet graph. Therefore, the existence of a given parquet graph in the test image has varying relevance for the selection of possible model candidates: the more often a feature f_t appears in different model images, the less important is the fact that it also appears in the test image at hand. In order to take that into account, an entropy-based measure of information (Shannon, 1948) is assigned to each feature detector:

$$i_t = \ln D + \sum_{d=1}^{D} \mathcal{P}[I_d \mid f_t] \cdot \ln \mathcal{P}[I_d \mid f_t], \tag{2.13}$$

with I_d being the dth model image. D denotes the number of model images. These measures quantify the information contribution of the feature detectors to the decision about picking model images as model candidates. They range between 0, for (irrelevant) features that appear in all model images, and $\ln D$, for (highly significant) features that appear in exactly one model image.

The conditional probabilities that the genuine object is the one in image I_d given that feature f_t has been observed are calculated using Bayes' rule,

$$\mathcal{P}[I_d \mid f_t] = \frac{n_t(I_d)}{\sum_{d'=1}^{D} n_t(I_{d'})}, \tag{2.14}$$

with $n_t(I_d) = \sum_{f \in \mathbb{F}(I_d)} \varepsilon(f, f_t, \vartheta)$ being the total number of occurrences of feature f_t in the model image I_d.

Each time a feature detector has found its reference feature f_t in the input image, we add a pair of matching features (f, f_t) to a table $\mathcal{F}_{match}(I)$, where f belongs to the input image:

$$\mathcal{F}_{match}(I) \leftarrow \mathcal{F}_{match}(I) \cup \bigcup_{f \in \mathbb{F}(I)} \{(f, f_t) \mid \varepsilon(f, f_t, \vartheta) = 1\}. \tag{2.15}$$

That table is used for efficient construction of image and model graphs in the correspondence-based verification part (see section 3). The table is cleared before each image presentation.

2.4 Preselection Network. For the selection of model candidates, a single-layer feedforward neural network is employed. Called the *preselection network*, it consists of T neurons in the input layer, one for each feature in the database, and D neurons in the output layer, one per model image. We use generalized McCulloch and Pitts neurons (McCulloch & Pitts, 1943) with identity output function (i.e., the output of a neuron equals its input). The tth input unit receives its input from the feature detector with the same index, and the dth output neuron is assigned to model image I_d. The neurons are connected via synapses with entropy-based weights taken from a $T \times D$ weight matrix,

$$\underline{\underline{W}} = (\tau_t(I_d) \cdot i_t)_{\substack{1 \le t \le T \\ 1 \le d \le D}} =: (w_{td})_{\substack{1 \le t \le T \\ 1 \le d \le D}},$$

with w_{td} being the strengths of the synaptic weights. These weights are zero if the feature f_t does not occur in image I_d. Therefore, the matrix is sparse and also implemented as such. Neither memory nor biological synapses are required for zero weights.

The outputs of the postsynaptic neurons are given by the product of the weight matrix and the vector of feature detector responses:

$$\underline{s}(I) = \underline{\underline{W}}^{\top} \cdot (\tau_t(I))_{1 \le t \le T} = \left(\sum_{t=1}^{T} w_{td} \cdot \tau_t(I) \right)_{1 \le d \le D} =: (s_d(I))_{1 \le d \le D}. \tag{2.16}$$

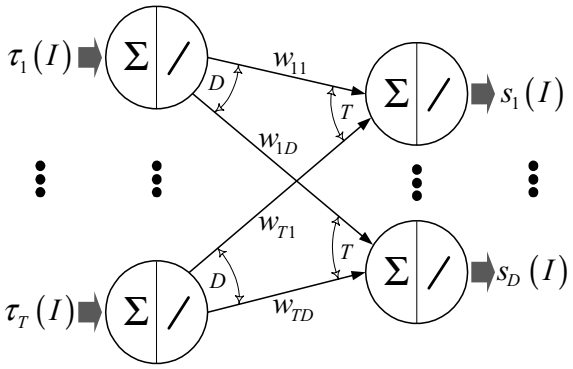


Figure 3: Preselection network. The preselection network is a single-layer feed-forward neural network. It consists of T neurons in the input layer and D neurons in the output layer. Each input neuron is associated with a feature detector and each output neuron with a model image. The synapses between pre- and postsynaptic neurons carry entropy-based weights. The outputs of the postsynaptic neurons are called activations.

These neural outputs are termed *model activations* or simply *activations*. For a given image I, the activation $s_d(I)$ of a model image I_d codes the accumulated information contributions of those feature detectors that have a coincidental model feature in the image and model domain. Thus, the activation scales proportionally with the probability that the input and the respective model image contain the same object (in a similar pose). The preselection network is given in Figure 3.

The selection of reasonable model candidates for an input image I, collected in a set $\mathbb{M}$, can be defined in different ways. For example, one could use a given number of strongly activated model images. In this letter, we select all model images whose activation exceeds a relative threshold θ:

$$\mathbb{M}(I,\theta) = \left\{ I_d \,\middle|\, s_d(I) \geq \theta \max_{1 \leq d' \leq D} \{s_{d'}(I)\};\ 1 \leq d \leq D \right\}. \tag{2.17}$$

This threshold determines the number of model candidates passed to the correspondence-based verification part (see section 3). For $\theta = 1$, only one model candidate is selected, the maximally activated model image, while for low values of θ, the set of model candidates may encompass a large number of the original model images. Thus, this parameter allows smoothly adjusting the balance between the feature- and correspondence-based parts.

The performance of the preselection network is exemplarily given in Figure 4. That figure gives the average number of model candidates in

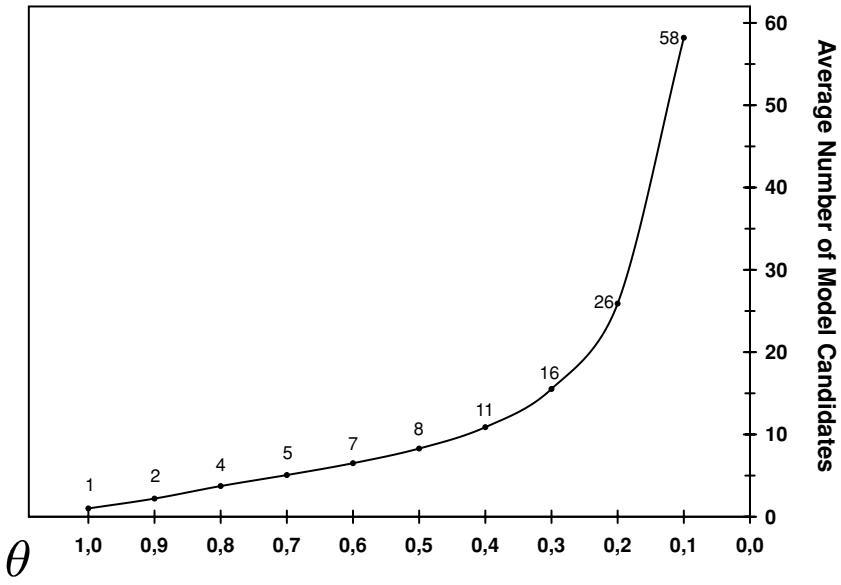


Figure 4: Performance of the preselection network. The average number of model candidates in dependence on the relative threshold θ. The learning set was composed 5600 images taken from the COIL-100 database (Nene et al., 1996). We observe that, first, the average number of model candidates is small relative to the total number of model images and, second, that this number grows rapidly with decreasing relative thresholds.

dependence on the relative threshold θ. The experiment was carried out with the object recognition application proposed in section 4. The learning set was composed of 5600 images taken from the COIL-100 database (Nene, Nayar, & Murase, 1996). The results show that on average, the preselection network favorably rules out most irrelevant matches. The average numbers of model candidates are small relative to the total number of model images, and the average number of model candidates grows rapidly with decreasing relative thresholds.

2.5 Modifications for Acceleration. In order to speed up the search of model candidates, two sets of vector-quantized features are used instead of one. The first set is created using a low threshold, $\vartheta_1 = 0.9$, which results in a low number of representatives. Due to its limited size, this set can be scanned linearly. The second set of quantized features is created using a higher-similarity threshold, $\vartheta_2 = 0.95$, which results in a large number of representatives. Once the first set has been scanned for a specific feature in an input image, only those features of the second set are investigated that

appear in the database images selected by the first set. The modifications are explained in detail in Westphal (2006).

3 Correspondence-Based Verification of Model Candidates

Thus far, the selection of model candidates has been based on the mere detection of feature coincidences in the image and model domains. The spatial arrangement of features, parquet graphs in our case, has been fully ignored, which can be particularly harmful in cases of multiple objects or structured backgrounds.

In the following, model candidates are further verified by checking that the features be in similar spatial arrangement for the model to be selected. More specifically, they are verified with a rudimentary version of elastic graph matching (von der Malsburg, 1988; Lades et al., 1993; Wiskott et al., 1997). For each model candidate, an image and a model graph are dynamically constructed through composing corresponding features into larger graphs according to their spatial arrangement. For each model candidate, the similarity between its image and model graph is computed. The model candidate whose model graph attains the best similarity is chosen as the model for the object contained in the input image. Its model graph is a good representation of that object with respect to the features in the database.

3.1 Construction of Graphs. Construction of graphs proceeds in three steps. First, from the table of matching features, equation 2.15, all feature pairs whose model feature stems from the current model candidate are transferred to a table of corresponding features. Second, templates of an image and of a model graph are instantiated with empty bunches of Gabor jets. Third, at each node position, separately for image and model graph, a bunch is assembled whose jets are taken from the respective parquet graph nodes located at that position, and the nodes are attributed with these bunches.

3.1.1 Table of Corresponding Features. During calculation of the model activations, pairs of matching features have been collected in a table of matching features $\mathcal{F}_{match}(I)$ (see equation 2.15). Given a model candidate $M \in \mathbb{M}(I, \theta)$ for the image I at hand (see equation 2.17), all feature pairs whose model feature appears in M are transferred to a table of corresponding features,

$$\mathcal{F}_{corr}(I, M) = \left\{ \left(f_n^I, f_n^M\right) \in \mathcal{F}_{match}(I) \mid 1 \le n \le N(M) \right.$$
$$\left. \wedge H\left(\sum_{f \in \mathbb{F}(M)} \varepsilon\left(f, f_n^M, 1\right) \right) = 1 \right\}, \tag{3.1}$$

of length $N(M)$, which will be used for efficient aggregation of parquet graphs into larger model and image graphs. Let f_n^I denote the image and f_n^M the model parquet graph of the nth feature pair, $n = 1, \ldots, N(M)$. Note that from now on, we speak of *corresponding* rather than of *matching* parquet graphs and assume that those graphs establish local arrays of contiguous point-to-point correspondences between the input image and the model candidate. These tables may differ considerably between model candidates: a pair of corresponding features need not necessarily occur in the tables of corresponding features of two distinct model candidates. Therefore, model and image graphs may vary between model candidates with respect to number of feature correspondences $N(M)$, graph topology, and attributed features.

Nodes of parquet graphs are attributed with a triple consisting of an absolute image position, a Gabor jet derived from an image at that position, and a validity flag (see section 2.2). To globally address node label components, the following notation is introduced: nodes of image parquet graphs are attributed with triples $(\underline{x}_{n,v}^I, \mathcal{J}_{n,v}^I, b_{n,v}^I)$, where n specifies the feature pair in the table of corresponding features and v specifies the node index. The same notation is used for model parquet graphs, with a superscript M for distinction:

$$\begin{aligned} f_n^I &= \left\{ \left(\underline{x}_{n,v}^I, \mathcal{J}_{n,v}^I, b_{n,v}^I \right) \middle| \, 1 \leq v \leq V \right\} \\ f_n^M &= \left\{ \left(\underline{x}_{n,v}^M, \mathcal{J}_{n,v}^M, b_{n,v}^M \right) \middle| \, 1 \leq v \leq V \right\}. \end{aligned} \tag{3.2}$$

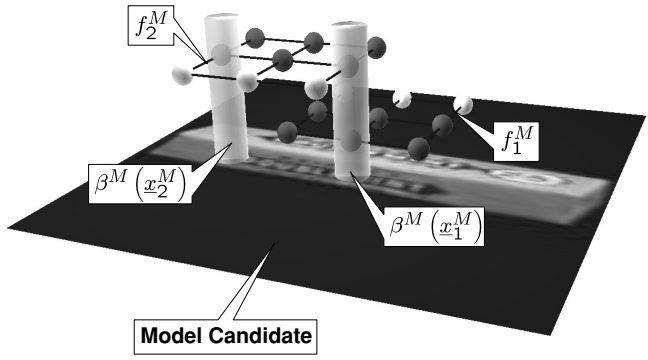
Storing absolute feature positions is just a conveniently simple implementation. For the graph assembly, only relative positions are required; more precisely, coincident invariant features must be composed with coincident ones if they are neighbors in the input image.

3.1.2 Graph Templates. First, templates of an image and of a model graph are instantiated without node labels. Number and positioning of nodes are determined by the valid-labeled nodes of image and model parquet graphs. Their positions are collected in sets $\mathbb{X}^I$ and $\mathbb{X}^M$, respectively:

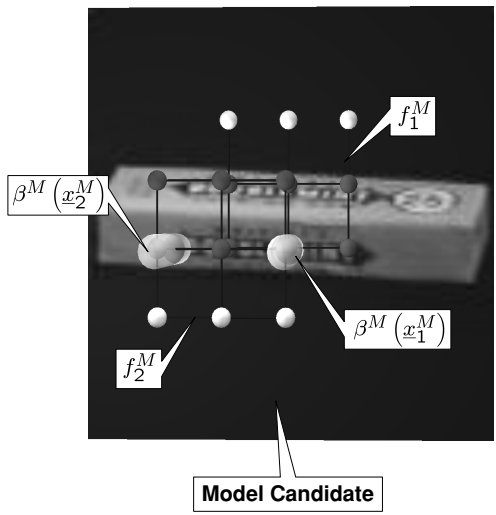
$$\begin{aligned} \mathbb{X}^I &= \bigcup_{n,v} \left\{ \underline{x}_{n,v}^I \middle| \, b_{n,v}^I = 1 \right\} \\ \mathbb{X}^M &= \bigcup_{n,v} \left\{ \underline{x}_{n,v}^M \middle| \, b_{n,v}^M = 1 \right\}. \end{aligned} \tag{3.3}$$

The creation of graph templates is illustrated in Figure 5.

3.1.3 Node Labels. The nodes of model and image graphs become attributed with bunches of Gabor jets: nodes of image graphs become labeled with bunches of Gabor jets that stem from node labels of valid-labeled nodes of image parquet graphs located at a given position $\underline{x}$ in the input image. The same applies to the nodes of model graphs, in which, of course,



(a)



(b)

Figure 5: Construction of model graphs. (a) A side view of the setup (b) A top view of the same setup. For clarity, both figures show only two overlapping model parquet graphs f_1^M and f_2^M drawn from the table of corresponding features. For illustration of the overlap, the graphs are drawn in a stacked manner. Number and position of the model graph's nodes are determined by the valid-labeled model parquet graph nodes (dark gray nodes). Nodes that reside in the background have been marked as invalid (white nodes). (b) The spatial shape of the emerging model graph can be seen. Compilation of bunches is demonstrated with two bunches only. Like stringing pearls, all valid Gabor jets at position $\underline{x}_1^M$ are collected into bunch $\beta^M(\underline{x}_1^M)$, and those at positions $\underline{x}_2^M$ become assembled into bunch $\beta^M(\underline{x}_2^M)$. From a, we learn that bunch $\beta^M(\underline{x}_1^M)$ comprises two jets, while bunch $\beta^M(\underline{x}_2^M)$ contains only one jet. Image graphs are constructed in the same fashion.

the jets are taken from the model parquet graphs. Let $\beta^I(\underline{x})$ denote a bunch assembled at an absolute position $\underline{x}$ in the input image. The same notation is used for the model graph's bunches, with a superscript M for distinction:

$$\begin{aligned}\beta^I(\underline{x}) &= \bigcup_{n,v} \left\{ \mathcal{J}^I_{n,v} \middle| \underline{x}^I_{n,v} = \underline{x} \wedge b^I_{n,v} = 1 \right\} \\ \beta^M(\underline{x}) &= \bigcup_{n,v} \left\{ \mathcal{J}^M_{n,v} \middle| \underline{x}^M_{n,v} = \underline{x} \wedge b^M_{n,v} = 1 \right\}.\end{aligned} \tag{3.4}$$

Whenever possible, we omit the position $\underline{x}$ and write β^I and β^M. The assembly of Gabor jets into bunches is also illustrated in Figure 5.

For the assessment of whether a point in the image corresponds to a point in the model candidate, a measure of similarity between two bunches is needed. It is defined as the maximal similarity between the bunches' jets, which is computed in a cross run. If one of the bunches is empty, the similarity between them yields 0. The jets are compared using the similarity function given in equation 2.7, which is based on the Gabor amplitudes:

$$s_{bunch}(\beta, \beta') = \begin{cases} \max\limits_{\mathcal{J}\in\beta, \mathcal{J}'\in\beta'} \{s_{abs}(\mathcal{J}, \mathcal{J}')\} & \text{if } \beta \neq \emptyset \wedge \beta' \neq \emptyset \\ 0 & \text{otherwise} \end{cases}. \tag{3.5}$$

3.1.4 Graphs. Like parquet graphs, image and model graphs are specified by a set of node labels:

$$\begin{aligned}\mathcal{G}^I &= \bigcup_{\underline{x}\in\mathbb{X}^I} \{(\underline{x}, \beta^I(\underline{x}))\} \\ \mathcal{G}^M &= \bigcup_{\underline{x}\in\mathbb{X}^M} \{(\underline{x}, \beta^M(\underline{x}))\}.\end{aligned} \tag{3.6}$$

Node labels comprise an absolute position in the input or model image drawn from the sets of node positions (see equation 3.3) and the bunch assembled at that position (see equation 3.4). The image graph is decorated with a superscript I, while the model graph receives a superscript M.

Model graphs of highly activated model candidates provide an approximation of the object in the input image by features present in the database. Figure 7 shows a number of model graphs (third column) that have been constructed for the input image given in the first column. The reconstructions from the model graphs of the first two model candidates in column 4 demonstrate that the emerged model graphs describe the object in the input image well.

3.2 Matching. In order to verify that a constructed model graph represents the object in the given image well, it is matched with the input image. It is moved as a template over the entire image plane in terms of maximizing the similarity between model and image graph. This action can be compared with the scan global move, which is usually performed as the

first step of elastic graph matching (Lades et al., 1993; Wiskott et al., 1997). For each translation of the model graph, the similarity between model and image graph is computed. The translation vector that yields the best similarity defines the optimal placement of the model graph in the image plane. In the process, the model graph's absolute node positions are transformed into relative ones by subtracting a displacement vector $\underline{t}_0$ from the positions of the model graph's nodes. That vector is chosen such that after subtraction, the smallest x and the smallest y coordinate become zero. However, the y coordinate of the left-most node and the x coordinate of the upper-most node are not necessarily 0:

$$\underline{t}_0 = \left(\min_{n,v}\left\{\left(\underline{x}^M_{n,v}\right)_x\right\}, \min_{n,v}\left\{\left(\underline{x}^M_{n,v}\right)_y\right\}\right)^\top. \tag{3.7}$$

The similarity between model and image graph with respect to a given translation vector $\underline{t}$ is defined as the average similarity between image and model bunches:

$$s(I, M, \underline{t}) = |\mathcal{G}^M|^{-1} \cdot \sum_{(\underline{x}^M, \beta^M) \in \mathcal{G}^M} s_{bunch}(\beta^I(\underline{x}^M - \underline{t}_0 + \underline{t}), \beta^M). \tag{3.8}$$

In order to find the object in the input image, the model graph is iteratively translated in the image plane so that the measure of similarity between model and image graph becomes maximal. In the process, model graphs of suitable model candidates move to the object's position in the input image. Let

$$s_{best}(I, M) = \max_{\underline{t} \in \mathbb{G}}\{s(I, M, \underline{t})\} \tag{3.9}$$

denote the similarity attained at that position. The displacement vectors $\underline{t}$ come from a set $\mathbb{G}$ of all grid points defined by the given distances Δx and Δy between neighbored parquet graph nodes (see section 2.2). The matching setup is given in Figure 6.

Executing the global move is important in cases where the average similarity between matching individual features is high but the similarity between the image and model graph is low. This can happen due to erroneous feature coincidences or multiple matches of the same image or model parquet graph. To obtain even better correspondences, graph matching including local optimization is certainly necessary.

3.3 Model Selection. For selection of the model, the most similar model image for the given input image, an image and a model graph are constructed for each model candidate. The model candidate that attains the best similarity between its model and image graph is chosen as the final

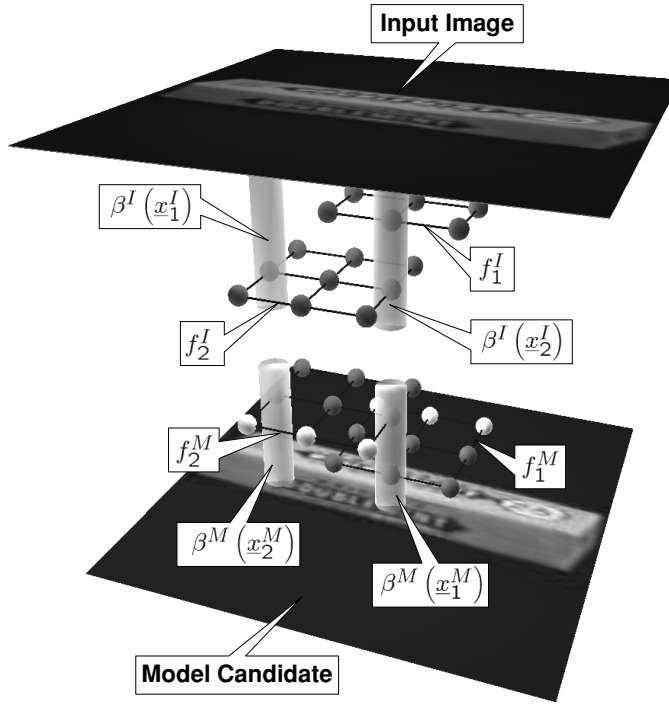


Figure 6: Matching setup. The setup consists of the input image, the model candidate, and the graphs constructed using the proposed method. For clarity, only two pairs of corresponding parquet graphs have been taken from the table of corresponding features. Parquet graph f_1^I corresponds to f_1^M and f_2^I corresponds to f_2^M. As in Figure 5, dark gray nodes represent nodes that have been marked as valid, and white nodes represent nodes that have been marked as invalid for residing in the background. Since only model images provide figure-ground information, invalid nodes appear only in the model parquet graphs. The compilation of bunches is illustrated for two sample positions $\underline{x}_1^I$ and $\underline{x}_2^I$ in the input image, and $\underline{x}_1^M$ and $\underline{x}_2^M$ in the model candidate. In order to find the object in the input image, the model graph is iteratively moved over the entire image plane and matched with the image graph.

model for the input image:

$$M_{best} = \underset{M \in \mathbb{M}(I,\theta)}{\arg\max}\{s_{best}(I, M)\}. \tag{3.10}$$

In Figure 7, four model candidates (column 2) have been computed for the given input image (column 1). The similarities attained through matching image against model graphs are given next to the reconstructions

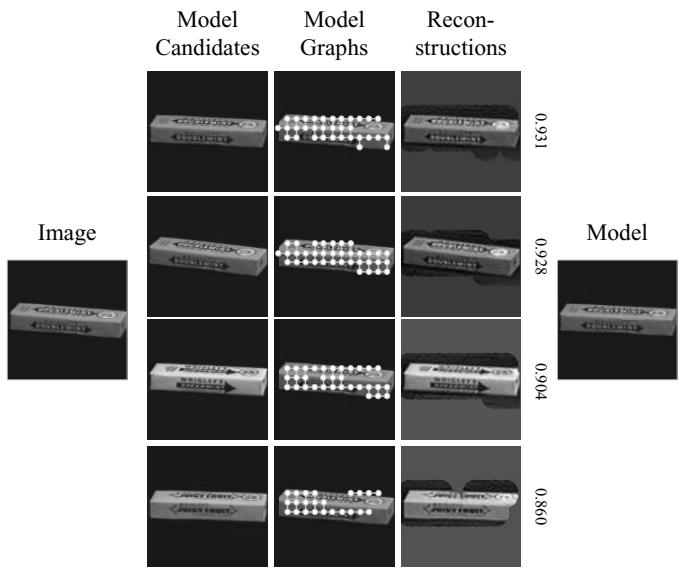


Figure 7: Selection of the model. Given the input image in the first column, the preselection network selects four model candidates (second column). For each model candidate, a model graph is dynamically constructed by assembling matching model features into larger graphs according to their spatial arrangement (third column). The fourth column shows the reconstruction from the model graph. Each model candidate is verified using a rudimentary version of elastic graph matching. Model graphs are optimally placed on the object contained in the input image in terms of maximizing a measure of similarity (third column). The attained similarities between the model candidates, represented by their model graphs, and the image graph, are indicated next to the reconstructions. The model candidate that attains the best similarity to the input image is chosen as the recognized model (fifth column).

from the model graphs (column 4). Since the first model candidate yields the highest similarity, it is chosen as the final model for the object in the input image (column 5).

4 Experiments

We present the results of three experiments. The first experiment (see section 4.2) was concerned with the recognition of single objects with respect to object identity and pose. Furthermore, the average execution times were measured. In the second experiment (see section 4.3), recognition performance was evaluated in the case of input images that contained multiple, nonoverlapping objects. Finally, the third experiment (see section 4.4) dealt with the recognition of partially occluded objects.

Figure 8: Example images of the COIL-100 and the ALOI image databases. (a, b) The images stem from the COIL-100 (Nene et al., 1996). (c, d) The images stem from the ALOI database (Geusebroek et al., 2005).

4.1 Experimental Setting. Experiments were conducted on two publicly available image databases for object recognition: the well-known Columbia Object Image Library (COIL-100) (Nene et al., 1996) and the more recent Amsterdam Library of Object Images (ALOI) (Geusebroek, Burghouts, & Smeulders, 2005). The COIL-100 database contains images of 100 objects. Images were acquired by placing the physical objects on a motorized turntable in front of a plain black background. In order to vary object pose with respect to a fixed color camera, the turntable was rotated through 360 degrees around the vertical axis, sampled in steps of 5 degrees. This corresponds to 72 poses per object identity and 7200 images for the whole collection. All images are 128×128 pixels in size. The images are normalized in size (i.e., the object always covers a maximal fraction of the image). The ALOI database contains images of 1000 objects with 72 poses per object identity. The mode of image acquisition was about the same as for the COIL-100 database. All images are 192×144 pixels in size. We selected a subset of 100 objects from the database. Since the images of the first 200 objects were considered too dark, we decided on objects numbered 200 to 299. The chosen subset consists of 7200 images. Compared to the COIL-100 database, the creators of the ALOI database invested less effort in image preprocessing. Especially, the images are much darker, the objects are not normalized in size, and the objects cover a much smaller fraction of the image, which results in larger number of parquet graphs from the background relative to the COIL-100 images in the case of unsegmented images. Therefore, experimental results attained with the ALOI database fall short compared to the COIL-100 database. Especially, they are subject to increased mean variations, as the experiments will show. Some example images of both databases are given in Figure 8.

(a) (b)

Figure 9: Input images of a single object. The figure shows an object from the COIL-100 database (Nene et al., 1996) as a (a) segmented and (b) unsegmented image.

Experimental results were obtained with fivefold cross-validation (Witten & Frank, 2000). In N-fold cross-validation, the data are split into N partitions of equal size; we decided for $N = 5$ partitions. Each is used once for testing, while the remaining $N - 1$ partitions are used for learning. This procedure is repeated N times such that every example has been used exactly once for testing. In this fashion, we created five pairs of disjoint learning and testing sets for each database, except where mentioned otherwise. Each learning set comprised 56, each testing set 14 views per object, for a total of 5600 and 1400 images, respectively. The images in both databases are perfectly segmented, that is, the objects are placed in front of a plain black background. In some experiments, we added structured backgrounds to the test images before presentation.

In the following, we present recognition results computed within the cross-validation and their dependence on the relative weighting of the feature- and correspondence-based parts. Each data point was averaged over $5 \cdot 1400 = 7000$ single measurements. Weighting of the feature- and correspondence-based parts was controlled by the relative threshold θ (equation 2.17) that ranged between 0.1 and 1, sampled in steps of 0.1.

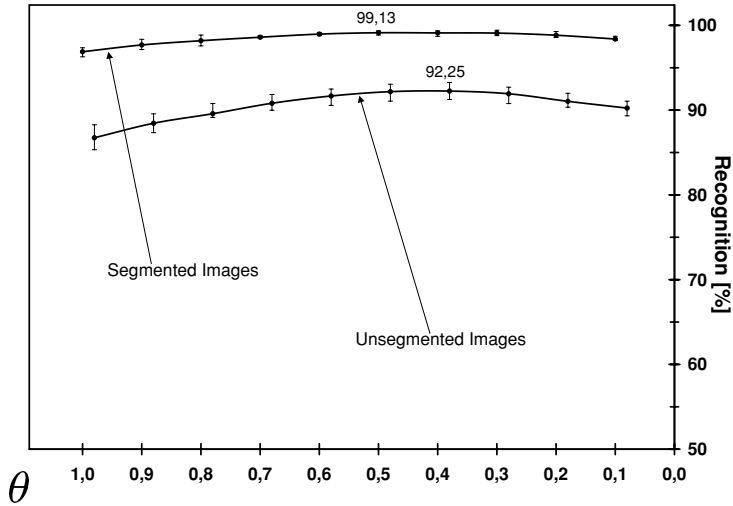
4.2 Recognition of Single Objects. In this experiment, we presented images containing a single object and evaluated the recognition performance with respect to object identity and pose for segmented and unsegmented images. Furthermore, the average execution times for one recognition attempt were measured. Since the images of both databases were perfectly segmented, unsegmented test images were manually created by pasting the object into a cluttered background before presentation. Backgrounds consisted of arbitrarily chosen image patches of random size derived from images of the current testing set. This is a challenging background for feature-based systems due to the multitude of ambiguous feature coincidences. Figure 9 shows an example of a segmented and an unsegmented test image.

4.2.1 Recognition of the Object Identity. Recognition performance with respect to object identity is shown in Figure 10. We considered the object in the test image to be correctly recognized if the test and model images showed the same object identity regardless of the object's pose. Throughout, better recognition rates were attained if segmented images were presented. Most interesting, a well-balanced combination of the feature- and correspondence-based parts led to optimal performance. Only for such well-balanced combinations was the selection of model candidates optimally carried out in the sense that neither too few nor too many model images became chosen as model candidates. If the number of model candidates was too small, the spectrum of alternatives the correspondence-based part could choose the final model from becomes too limited. This is especially harmful if false positives were frequent among model candidates. Conversely, the number of false positives among model candidates unavoidably increased with overemphasis of the correspondence-based part: for too low values of the relative threshold, even weakly activated model images became selected as model candidates. Accordingly, the mere probability of choosing a false positive as the final model increased and, consequently, the average recognition rate decreased.

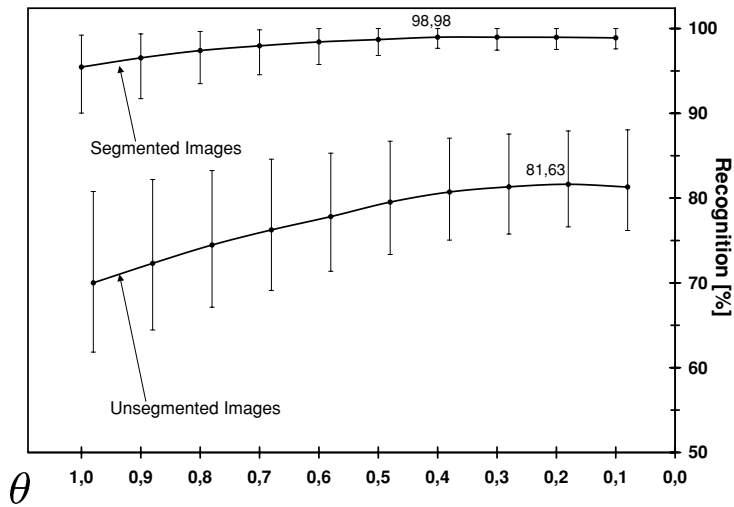
4.2.2 Recognition of the Object Pose. The same findings apply for the performance with respect to object pose, which is given in Figure 11. The average pose errors were calculated over the absolute values of angle differences of correctly recognized, nonrotation-symmetric objects. Note that two consecutive model images of the same object were at least 5 degrees apart. The same applies for the objects in the test images. The pose errors contain all errors due to pose ambiguity, which are negligible in practice. For robot grasping (see, e.g., Schmidt & Westphal, 2004), the number of misclassified poses is more relevant than the mean pose error.

4.2.3 Average Execution Time. The average execution time of a single recognition attempt as a function of relative weighting of the feature- and correspondence-based parts is given in Figure 12. It is rather remarkable that they are almost independent of the number of model candidates. Thus, correspondence-based techniques do not automatically imply slow execution relative to feature-based approaches: The difference in processing time is negligible if the correspondence-based verification part is restricted to only a few model candidates, which is achieved here by a feature-based preselection.

4.3 Recognition of Multiple Objects. This experiment was concerned with the recognition of multiple, simultaneously presented, nonoverlapping objects (i.e., input images showed simple visual scenes). Only the recognition performance with respect to object identity was evaluated. The experiment was subdivided into six test cases per database. In the first three test cases, we simultaneously, presented $N = 2$, 3, or 4 objects

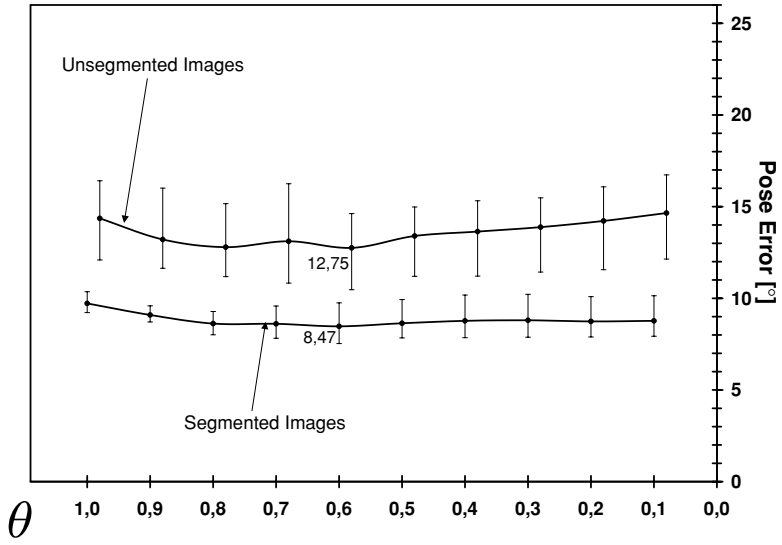


(a)

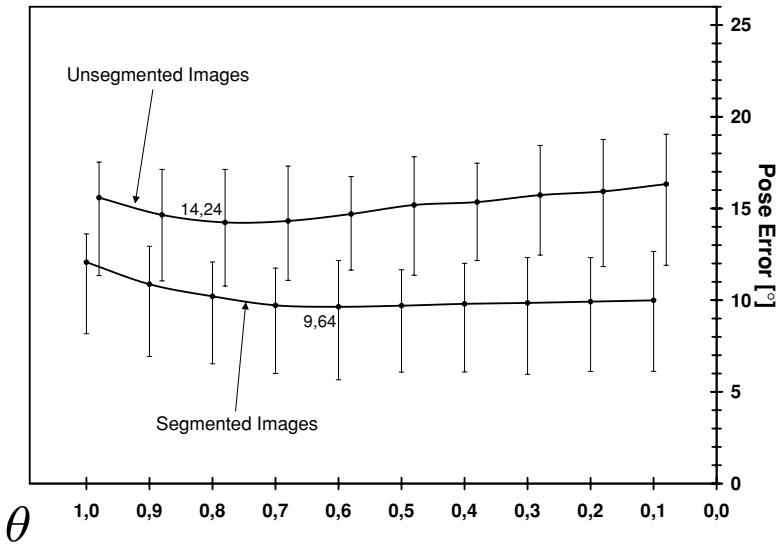


(b)

Figure 10: Recognition of single objects (identity). The figure shows the recognition performance with respect to object identity. (a) Results attained with the COIL-100. (b) Results attained with the ALOI database. The recognition performance is shown as a function of relative weighting of the feature- and correspondence-based parts controlled by θ. The best results are indicated next to the respective data points. Optimal performance was attained using a well-balanced combination of the feature- and correspondence-based parts.

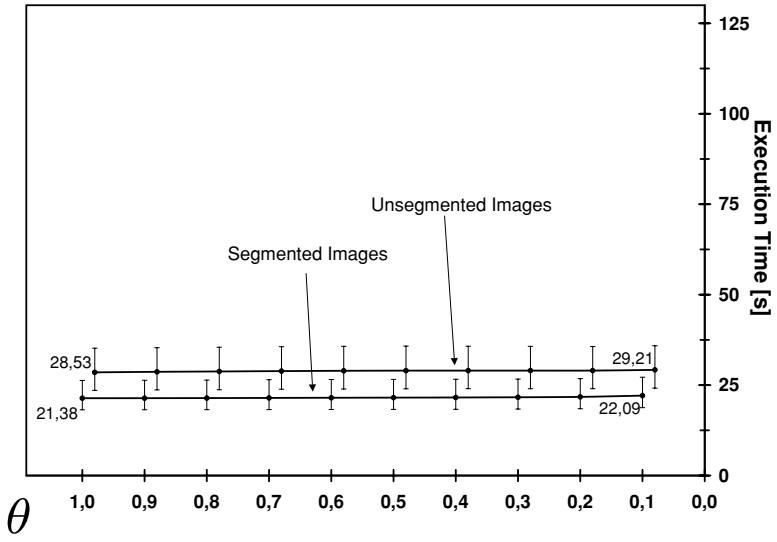


(a)

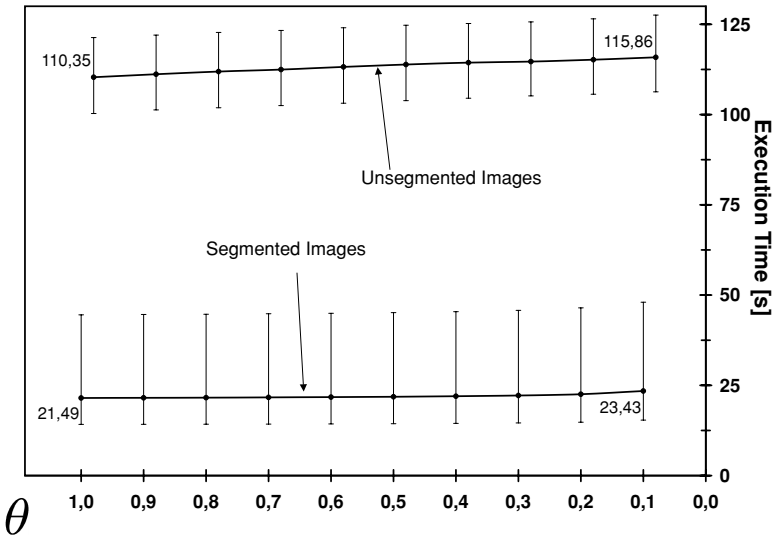


(b)

Figure 11: Recognition of single objects (pose). Recognition performance with respect to object pose: (a) COIL-100, (b) ALOI database. As in the identity case (see Figure 10), optimal performance was attained with a well-balanced combination of the feature- and correspondence-based parts.

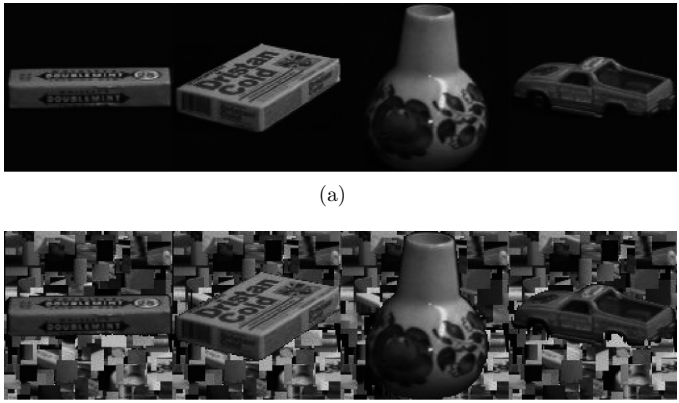


(a)



(b)

Figure 12: Recognition of single objects (execution time). The average execution times of a single recognition attempt depending on relative weighting of the feature- and correspondence-based parts is given. (a) COIL-100. (b) ALOI database. It is rather remarkable that they are almost independent of the number of model candidates.



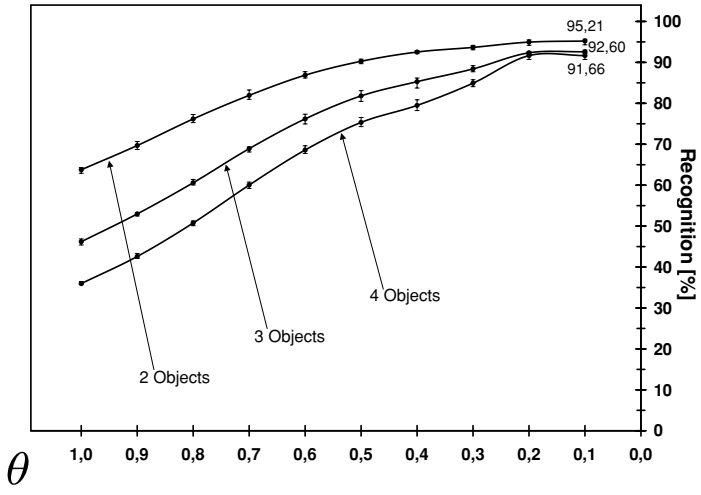
(a)

(b)

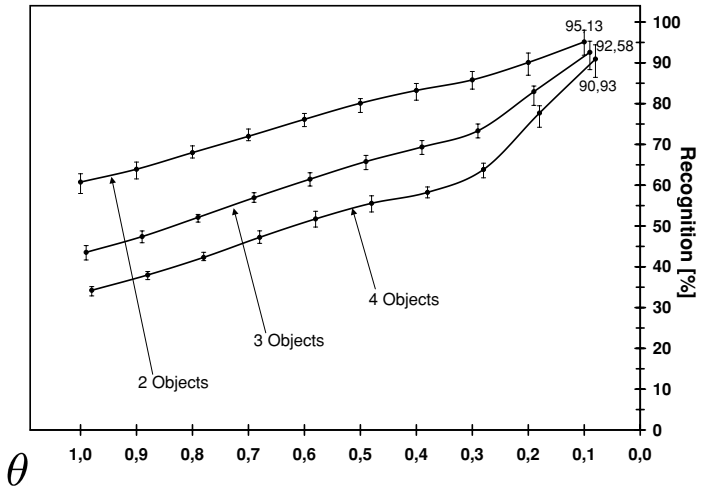
Figure 13: Input images of multiple objects. An example of (a) a segmented and (b) an unsegmented input image containing four objects drawn from the COIL-100 database. Backgrounds were constructed in the same fashion as in the first experiment.

placed in front of a plain black background, in the last three test cases, cluttered background was manually added. The procedure of background construction was the same as in the first experiment. Figure 13 shows two images containing four objects with and without background. Objects were randomly picked, a test image contained only different ones, and each object appeared at least once. The system returned the N most similar models. Each coincidence with one of the presented objects was counted as a successful recognition response; the correctness of position was not checked. Mixing up of objects appears improbable given the good recognition rates of single objects. Accounting problems would occur with input images containing multiple instances of the same object identity, which we have excluded in the construction of the test images. The average recognition rates were calculated over all responses.

The result of this experiment is given in Figures 14 and 15. It shows that compared to the single-object experiments (see section 4.2), the point of optimal recognition performance considerably moved to the right: putting more emphasis on the correspondence-based verification part improved recognition performance. This finding can be explained with the assumption that solving the binding problem (von der Malsburg, 1981, 1999) is required in the case of multiple objects. Presentation of segmented images yielded better results. For both segmented and unsegmented images, the system's performance degraded smoothly with the number of simultaneously presented objects. Especially in the test cases conducted on the ALOI database, one can expect that recognition rates could have been improved further by

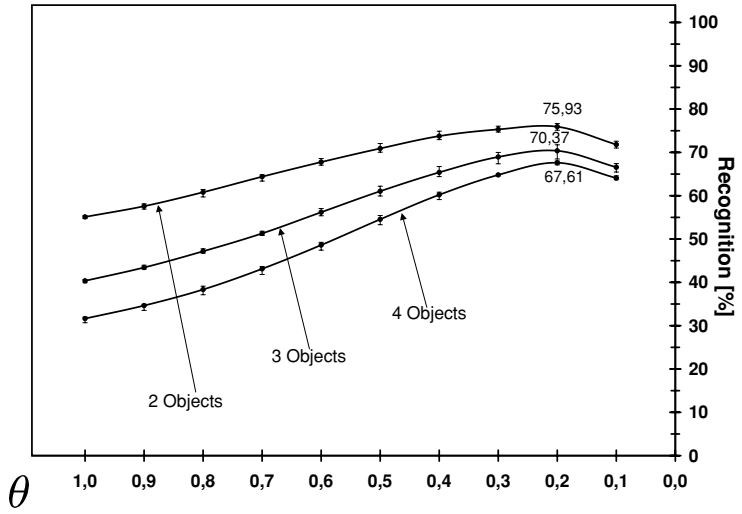


(a)

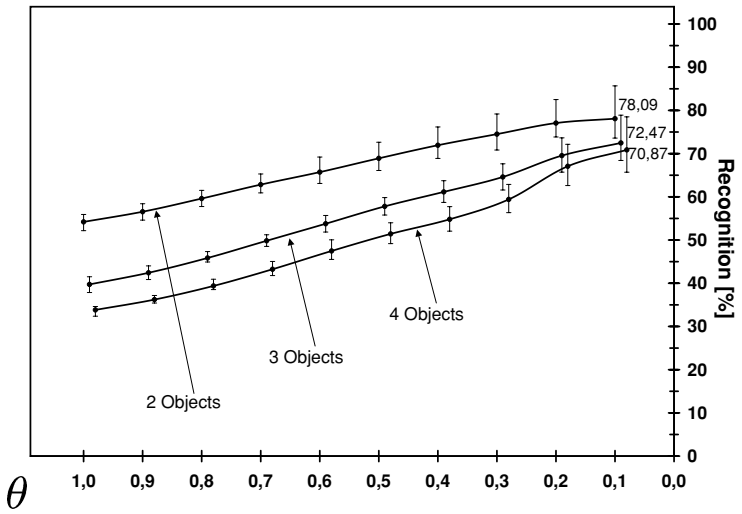


(b)

Figure 14: Recognition of multiple objects (segmented images). The recognition performance with respect to object identity in the case of multiple nonoverlapping objects where the objects in the input images were placed in front of a plain black background. (a) Results for the COIL-100 database. (b) Results for the ALOI database. Compared to the first experiment (in section 4.2), the point of optimal recognition performance moved considerably to the right: correspondence-based verification is more important in the case of multiple objects. Performance degraded smoothly with the number of simultaneously presented objects.



(a)



(b)

Figure 15: Recognition of multiple objects (unsegmented images). Recognition performance with respect to object identity in the case of multiple nonoverlapping objects where the objects in the input images were placed in front of a structured background. (a) Results attained with the COIL-100. (b) Results attained with the ALOI database. As in the case of segmented input images (see Figure 14), recognition performance degraded smoothly with the number of simultaneously presented objects.

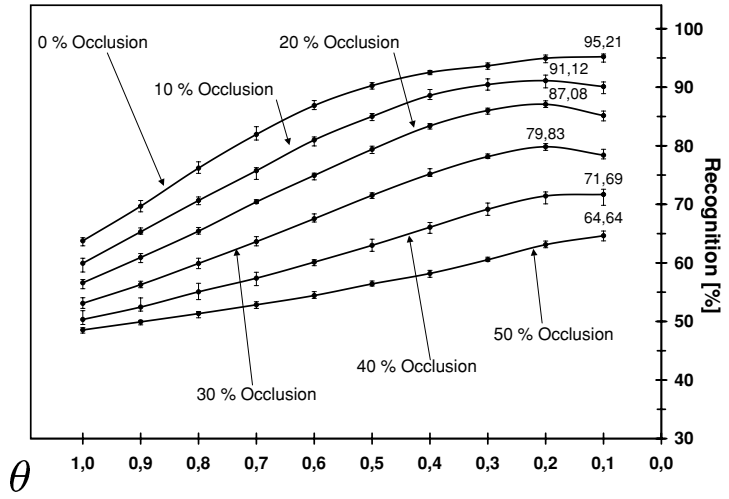
(a) (b)

Figure 16: Input images of a partially occluded object. (a) A segmented and (b) an unsegmented input image of a partially occluded object. In this example, the occluding object covers about 50% of the occluded object.

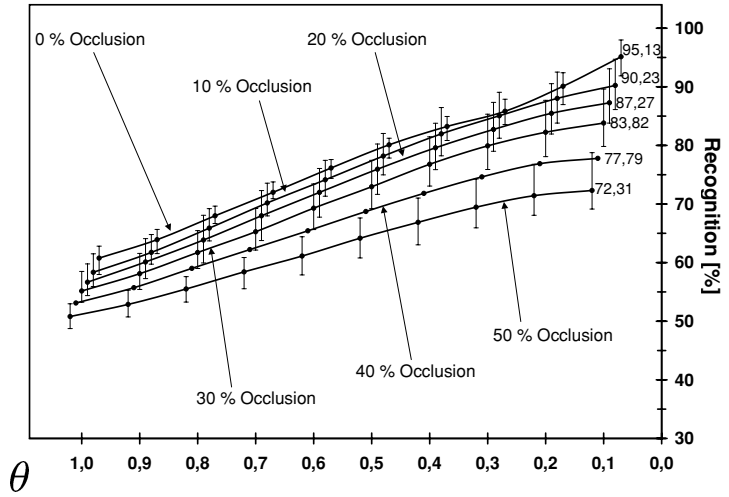
putting more emphasis on the correspondence-based part through choosing $\theta < 0.1$. For performance reasons, this was not carried out.

4.4 Recognition of Partially Occluded Objects. While in the previous experiment (in section 4.3) the objects were presented in a nonoverlapping manner, this final object recognition experiment was concerned with recognition of partially occluded objects. Only the recognition performance with respect to object identity was evaluated. The experiment was organized into 12 test cases per database. In the first 6 test cases, we simultaneously presented two objects where 0 to 50% of the object on the left was occluded by the object on the right. The amount of occlusion was sampled in 10% steps. Occluded and occluding objects were different and randomly picked, and each object appeared at least once as occluded. In the remaining 6 test cases, cluttered background was added. The procedure of background construction was the same as in the first experiment. Accounting of recognition responses was the same as in the experiments with multiple objects, again without checking the objects' positions. Figure 16 shows sample input images of a partially occluded object with and without added background.

The result of this experiment is given in Figures 17 and 18. In Figure 17 the objects in the input images were placed in front of a plain black background, while the result given in Figure 18 was attained with unsegmented images. As in the previous experiment, emphasis of the correspondence-based part improved recognition performance: a solution of the binding problem is also important in the case of partially occluded objects. Moreover, presentation of segmented images yielded better results. For both segmented and unsegmented images, the system's performance degraded smoothly with the amount of occlusion. Experimental results for the test cases with no occlusion were taken from the first test case of the previous experiment, in which the input images contained two nonoverlapping objects. As in the experiment with multiple objects, one can expect that recognition rates

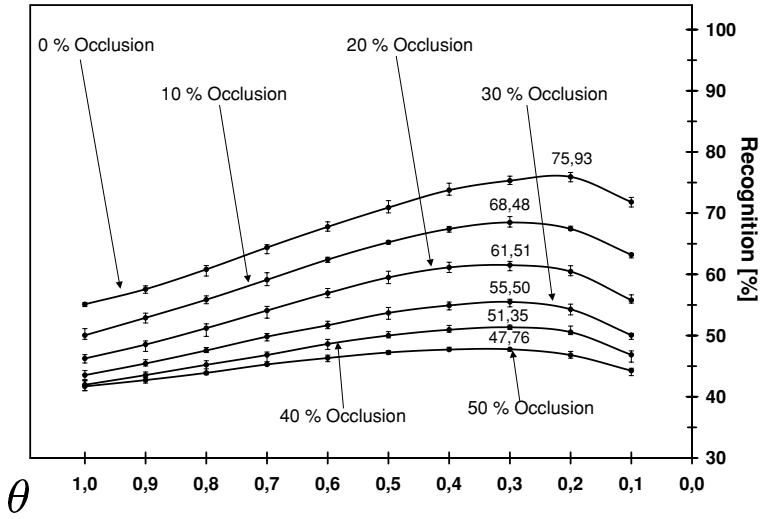


(a)

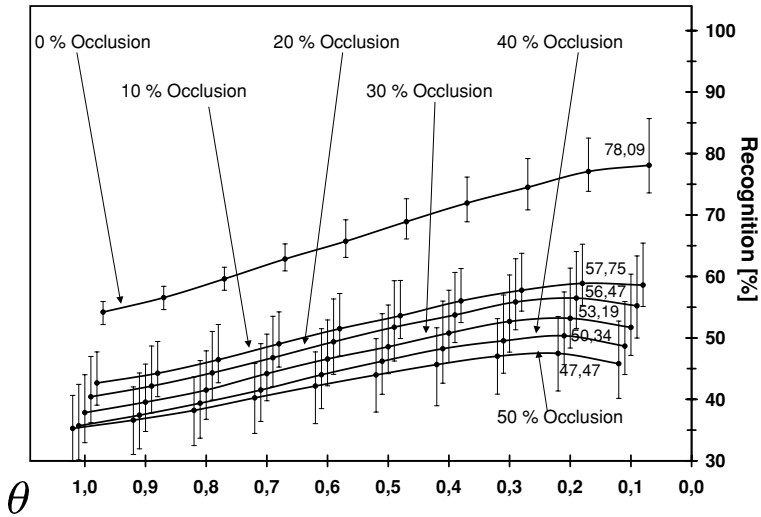


(b)

Figure 17: Recognition of partially occluded objects (segmented images). Recognition performance with respect to object identity in the case of partially occluded objects placed in front of a plain black background. (a) Results attained with the COIL-100 database. (b) Results obtained with the ALOI database. As in the case of multiple objects (see section 4.3), emphasis of the correspondence-based verification of model candidates considerably improved recognition performance. Performance degraded smoothly with the number of simultaneously presented objects.



(a)



(b)

Figure 18: Recognition of partially occluded objects (unsegmented images). Recognition performance with respect to object identity in the case of partially occluded objects placed in front of a structured background. (a) Results attained with the COIL-100 database. (b) Results attained with the ALOI database. As in the case of segmented input images (see Figure 17), recognition performance degraded smoothly with the amount of occlusion.

could in some cases have been improved further by putting more emphasis on the correspondence-based part by choosing $\theta < 0.1$.

4.5 Comparison with Other Techniques. Our system performed favorably compared with other techniques. The original system by Murase and Nayar (1995), which performs a nearest-neighbor classification to a manifold representing a collection of objects or class views, attained a recognition rate of 100% for segmented images of single objects drawn from the COIL-100 database. Our system attained a recognition rate of 99.13% in the same test case (see section 4.2). The recognition performance of the system by Murase and Nayar is, however, unclear if it would be confronted with more sophisticated recognition tasks, such as images with structured backgrounds, multiple objects, or occluded objects.

Wersing and Körner (2003) compare their method of setting up the feature extraction layers in an evolutionary fashion with the performance of the one in Murase and Nayar (1995). The authors conducted their experiments on the COIL-100 database. In the case of segmented images, their system and ours performed about equally well (see Figure 4b and Table 1 in Wersing & Körner, 2003, and Figure 10a).

In the case of unsegmented images, our system outperformed the system by Wersing and Körner (2003). (see Figure 6a in Wersing & Körner, 2003, and Figure 11a.) Our system attained a recognition rate of 92.25%, while the system from Wersing and Körner achieved a recognition rate slightly below 90%. It is, however, worth mentioning that the experimental setting differs considerably in the compared experiments. Wersing and Körner performed their experiment on the first 50 objects of the COIL-100 database and constructed structured backgrounds out of fairly big patches of the remaining 50 objects. In contrast, we conducted the experiment on all objects and pasted them into a cluttered background consisting of arbitrarily chosen image patches of random size derived from the other test images.

5 Summary and Future Work

We presented a method for invariant visual recognition of objects that employs a combination of rapid feature-based preselection with self-organized model graph creation and subsequent correspondence-based verification of model candidates. This hybrid method outperformed both purely feature-based and purely correspondence-based approaches, especially for more sophisticated recognition tasks, such as images with structured background, multiple objects, or partially occluded objects. Throughout, a well-balanced combination of the feature-based and correspondence-based parts produced optimal results in terms of recognition rate and pose error. In all test cases, the system's performance degraded smoothly with the increasing complexity of the recognition tasks. In a qualitative sense the results attained with the COIL-100 database are comparable to those attained with

the ALOI database. Because of the poorer quality of the ALOI images relative to the COIL-100 images, which was especially harmful in the case of structured backgrounds and occlusion, the results achieved with that database are subject to an increased mean variation relative to those attained with the COIL-100 database.

As an intermediate result, the system also produces model graphs, which are representations of a presented object in terms of the memorized features. A variety of further processing can build on these graphs. The simple graph matching employed here can be replaced by the more sophisticated methods from Lades et al. (1993), Wiskott et al. (1997), and Tewes (2006), which should lead to increased robustness under shape and pose variations.

In the existing state, the method can also be used for the purposeful initialization of sophisticated but slow techniques. For instance, it can produce a coarse pose estimation followed by refinement through correspondence-field evaluation. Another promising extension will be to use diagnostics from the classification process for novelty detection and subsequent autonomous learning.

The computational model we have described here can be criticized on several grounds; we discuss three of them. First, the testing data can be seen as insufficient. It has been remarked that available object databases do not address the important problems of object recognition because they do not contain enough invariances (Pinto, Cox, & DiCarlo, 2008). As the authors correctly state, real-world images contain much more variability than can be captured by reasonably sized databases. However, a mechanism that dynamically constructs object representations from correspondences can greatly alleviate problems with varying backgrounds and allows a whole range of invariances like background clutter and partial occlusion, as we have demonstrated. Furthermore, the use of standard databases is the only way to compare different algorithms beyond speculation on how they would perform on arbitrary images.

Second, the features used can be regarded as too simple for recognition under hard real-world conditions. We are using a hierarchy of features starting with complex cell responses, composing them into jets, and jets into parquet graphs, and these into model and image graphs, according to matching dynamics. A more flexible feature hierarchy like the one proposed by Ullman (2007), Bart and Ullman (2008) would certainly improve recognition in our system. We think that the invariance under local shifts exhibited by the complex cell responses while keeping essential image information (Wundrich, von der Malsburg, & Würtz, 2004) is an advantage of our feature types over gray-value patches. Further invariances can be achieved by assuming that objects with similar features in one view will also have similar features in another view. This allows learning depth-rotated versions of features from examples (Bart & Ullman, 2008; see also Müller, Heinrichs, Tewes, Schäfer, & Würtz, 2007). In the future, we will incorporate relational networks into our system, which code, for example,

that different views belong to the same object, which is the basis of using this information for invariant recognition.

Third, it may be doubted that the brain actually works this way. At the current state, experimental evidence for correspondence-computing circuits in the brain is sparse. Their necessity can, however, be argued on the basis of computational theory; this letter is an example. Recently, detailed models have been presented that allow fast correspondence estimation in a neural system (Lücke, Keck, & von der Malsburg, 2008).

Another point with weak biological motivation is the use of a maximum operator in the calculation of a relative threshold for the selection of model candidates (see equation 2.17). Stimulation of cells in the inferotemporal cortex reveals that invariant responses to the presence of several objects are well described by the average of the responses to single objects (Zoccolan, Cox, & DiCarlo, 2005), meaning that the maximum operation cannot be the final nonlinearity used for a recognition decision. However, if the computational goal is that all present objects lead to superthreshold responses in "their" neurons, this behavior is actually compatible with the application of a relative threshold. It is also what we require for preselection of local features.

To conclude, this letter proposes that a combination of feature- and correspondence-based methods for the task of invariant visual object recognition is a good computational strategy. Here, it has been modeled as a two-stage process. First, feature-based processing is applied as far as it goes by excluding as many objects as possible and, second, the ambiguous cases, the model candidates, are subjected to correspondence-based processing. Within the more biology-inspired dynamic link matching (DLM) framework (von der Malsburg, 1981, 2002; Wiskott, 1995), the result of the feature-based stage would supply a suitable initialization of the correspondence-based part, that is, an initialization of the dynamic links. The final decision of an appropriate model for the object contained in the input image is then subjected to the DLM machinery. This will be the topic of a forthcoming publication.

Acknowledgments

Partial funding from Deutsche Forschungsgemeinschaft (WU 314/2-2 and WU 314/5-2) and from the European Commission in the NOVOBRAIN project is gratefully acknowledged. We thank the reviewers for thoughtful comments, which significantly improved this letter.

References

Bart, E., & Ullman, S. (2008). Class-based feature matching across unrestricted transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(09), 1618–1631.

Becker, M., Kefalea, E., Maël, E., von der Malsburg, C., Pagel, M., Triesch, J., et al. (1999). GripSee: A gesture-controlled robot for object perception and manipulation. *Autonomous Robots*, *6*(2), 203–221.

Bienenstock, E., & Geman, S. (1995). Compositionality in neural systems. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 223–226). Cambridge, MA: MIT Press.

Bunke, H. (1983). Graph grammars as a generative tool in image understanding. In H. Ehrig, M. Nagl, & G. Rozenberg (Eds.), *Graph grammars and their application to computer science* (pp. 8–19). Berlin: Springer.

Edelman, S. (1995). Representation, similarity, and the chorus of prototypes. *Minds and Machines*, *5*(1), 45–68.

Elliffe, M., Rolls, E., & Stringer, S. (2002). Invariant recognition of feature combinations in the visual system. *Biological Cybernetics*, *86*, 59–71.

Fukushima, K., Miyake, S., & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, *13*(5), 826–834.

Geusebroek, J., Burghouts, G., & Smeulders, A. (2005). The Amsterdam library of object images. *International Journal of Computer Vision*, *61*, 103–112.

Gray, R. (1984). Vector quantization. *IEEE Signal Processing Magazine*, *1*(2), 4–29.

Hebb, D. (1949). *The organization of behavior*. New York: Wiley.

Hubel, D., & Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*, 106–154.

Jones, J., & Palmer, L. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, *58*(6), 1233–1258.

Lades, M., Vorbrüggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., et al. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, *42*(3), 300–310.

Lam, L., & Suen, S. (1997). Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, *27*(5), 553–568.

Linsker, R. (1988). Self-organization in a perceptual network. *IEEE Computer*, *21*, 105–117.

Logothetis, N., & Pauls, J. (1995). Psychophysical and physiological evidence for viewer-centered object representation in the primate. *Cerebral Cortex*, *3*, 270–288.

Lücke, J., Keck, C., & von der Malsburg, C. (2008). Rapid convergence to feature layer correspondences. *Neural Comput.*, *20*(10), 2441–2463.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

Mel, B. (1997). SEEMORE: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Comput.*, *9*, 777–804.

Müller, M. K., Heinrichs, A., Tewes, A. H., Schäfer, A., & Würtz, R. P. (2007). Similarity rank correlation for face recognition under unenrolled pose. In S.-W. Lee & S. Z. Li (Eds.), *Advances in biometrics* (pp. 67–76). Berlin: Springer.

Murase, H., & Nayar, S. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, *14*, 5–24.

Nene, S., Nayar, S., & Murase, H. (1996). *Columbia Object Image Library (COIL-100)* (Tech. Rep. CUCS-006-96). New York: Columbia University.

Olshausen, B., & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Oram, M., & Perret, D. (1994). Modelling visual recognition from neurobiological constraints. *Neural Networks*, *7*(6/7), 945–972.

Palmeri, T., & Gauthier, I. (2004). Visual object understanding. *Nature Reviews Neuroscience*, *5*, 291–304.

Perret, D., Smith, P., Potter, D., Mistlin, A., Head, A., & Milner, A. (1985). Visual cells in the temporal cortex sensitive to face view and gaze direction. *Proceedings of the Royal Society B*, *223*, 293–317.

Pinto, N., Cox, D. D., & DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, *4*(1), 151–156.

Pitts, W., & McCulloch, W. (1947). How we know universals: The perception of auditory and visual forms. *Bulletin of Mathematical Biophysics*, *9*, 127–147.

Pötzsch, M., Maurer, T., Wiskott, L., & von der Malsburg, C. (1996). Reconstruction from graphs labeled with responses of Gabor filters. In C. von der Malsburg, W. von Seelen, J. Vorbrüggen, & B. Sendhoff (Eds.), *Proceedings of the ICANN 1996* (pp. 845–850). Berlin: Springer.

Riesenhuber, M., & Poggio, T. (2000). Models of object recognition. *Nature Neuroscience*, *3*, 1199–1204.

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington DC: Spartan.

Schiele, B., & Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, *36*(1), 31–50.

Schmidt, P., & Westphal, G. (2004). Object manipulation by integration of visual and tactile representations. In U. J. Ilg, H. H. Bülthoff, & H. A. Mallot (Eds.), *Dynamic perception* (pp. 101–106). Amsterdam: IOS Press.

Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *PNAS*, *104*(15), 6424–6429.

Shams, L. (1999). *Development of visual shape primitives*. Unpublished doctoral dissertation, University of Southern California.

Shannon, C. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, *27*, 623–656.

Shapiro, L., & Haralick, R. (1981). Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *3*(5), 504–519.

Tewes, A. (2006). *A flexible object model for encoding and matching human faces*. Aachen: Shaker.

Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, *381*, 520–522.

Thorpe, S., & Thorpe, M. (2001). Seeking categories in the brain. *Neuroscience*, *291*, 260–263.

Treismann, A., & Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, *12*, 97–136.

Tsotsos, J. K. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, *13*(3), 423–445.

Ullman, S. (1989). Aligning pictorial descriptions: An approach to object recognition. *Cognition*, *32*(3), 193–254.

Ullman, S. (2007). Object recognition and segmentation by a fragment-based hierarchy. *Trends in Cognitive Sciences*, *11*(2), 58–64.

von der Malsburg, C. (1981). *The correlation theory of brain function* (Internal Rep. 81-2). Göttingen: Max-Planck-Institute for Biophysical Chemistry, Department of Neurobiology.

von der Malsburg, C. (1988). Pattern recognition by labeled graph matching. *Neural Networks*, *1*, 141–148.

von der Malsburg, C. (1999). The what and why of binding: The modeler's perspective. *Neuron*, *24*, 95–104.

von der Malsburg, C. (2002). The dynamic link architecture. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed., pp. 1002–1005). Cambridge, MA: MIT Press.

von der Malsburg, C., & Reiser, K. (1995). Pose invariant object recognition in a neural system. In F. Fogelmann-Soulié, J. C. Rault, P. Gallinari, & G. Dreyfus (Eds.), *International Conference on Artifical Neural Networks (ICANN 1995)* (pp. 127–132). Paris, France: EC2 & Cie.

von Luxburg, U., Bousquet, O., & Schölkopf, B. (2004). A compression approach to support vector model selection. *Journal of Machine Learning Research*, *5*, 293–323.

Wersing, H., & Körner, E. (2003). Learning optimized features for hierarchical models of invariant object recognition. *Neural Comput.*, *15*, 1559–1588.

Westphal, G. (2006). *Feature-driven emergence of model graphs for object recognition and categorization*. Electronic dissertation, University of Lübeck, Germany, urn:nbn:de:gbv:841-20070904428.

Westphal, G., & Würtz, R. P. (2004). Fast object and pose recognition through minimum entropy coding. In J. Kittler, M. Petrou, & M. Nixon (Eds.), *17th International Conference on Pattern Recognition (ICPR 2004)* (Vol. 3, pp. 53–56). Cambridge, UK: IEEE Press.

Wiskott, L. (1995). *Labeled graphs and dynamic link matching for face recognition and scene analysis*. Thun, Frankfurt am Main: Deutsch.

Wiskott, L., Fellous, J.-M., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775–779.

Witten, I., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with JAVA implementations*. San Francisco: Morgan Kaufmann.

Wundrich, I. J., von der Malsburg, C., & Würtz, R. P. (2004). Image representation by complex cell responses. *Neural Comput.*, *16*(12), 2563–2575.

Würtz, R. P. (1995). *Multilayer dynamic link networks for establishing image point correspondences and visual object recognition*. Thun, Frankfurt am Main: Deutsch.

Würtz, R. P. (1997). Object recognition robust under translations, deformations, and changes in background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 769–775.

Zoccolan, D., Cox, D., & DiCarlo, J. (2005). Multiple object response normalization in monkey inferotemporal cortex. *Journal of Neuroscience*, *25*(36), 8150–8164.

Received December 19, 2007; accepted November 14, 2008.