

Aus dem Institut für Neuro- und Bioinformatik  
der Universität zu Lübeck  
Direktor: Prof. Dr. Thomas Martinetz

---

Feature-Driven Emergence of Model Graphs  
for Object Recognition and Categorization

Inauguraldissertation

zur  
Erlangung der Doktorwürde  
der Universität zu Lübeck  
— Aus der **Technisch-Naturwissenschaftlichen Fakultät** —

vorgelegt von

Günter Westphal  
aus Anholt, jetzt Isselburg

Lübeck 2006

1. Berichtstatter: Prof. Dr. Thomas Martinetz
  2. Berichtstatter: Prof. Dr. Christoph von der Malsburg
  3. Berichtstatter: Prof. Dr. Peter König
  4. Berichtstatter: Prof. Dr.-Ing. Til Aach
- Tag der mündlichen Prüfung: 09.Mai 2007

# Acknowledgments

*I cannot even imagine where I would be today were it not for that handful of friends.*

Charles R. Swindoll

I have been fortunate to work with Professor von der Malsburg's group at the *Institut für Neuroinformatik* at the *Ruhr-Universität Bochum*. I had the privilege to work in his system biophysics group for almost four years and I fondly enjoyed every single second of it. I immensely enjoyed the superb working conditions, the creative atmosphere, the free flow of ideas, the freedom to develop my own ideas, and the friendship I experienced there. I am especially obliged to my advisor, *Prof. Dr. Christoph von der Malsburg*, for his help in scientific questions, his advice, and for guiding my research in many invaluable discussions.

I would like to cordially thank *Prof. Dr. Thomas Martinetz*, director of the *Institut für Neuro- und Bioinformatik* at the *Universität zu Lübeck*, for making this cooperation possible, for taking the trouble to read the dissertation and provide the first report.

Thanks go also to *Prof. Dr. Peter König* and *Prof. Dr. Til Aach* for providing the third and the fourth report.

My warmest thanks go to *Dr. Rolf Würtz* who read the whole manuscript and considerably improved it with his thoughtful comments, for uncountable passionate debates and for asking the most piercing but right questions.

I further would like to heartily thank *Uta Schwalm*, *Anke Bücher*, and *Babett Bernitt* for their kind help in removing many administrative obstacles, probably without me noticing, and *Michael Neef* for his tireless efforts to tame the institute's computing infrastructure.

A lot of thanks go to the current and former members of the system biophysics group, especially to *Alexander Heinrichs*, *Wilfried Horn*, *Andreas Tewes*, *Bram Bolder*, and *Peer Schmidt* who proofread parts of this work and improved it with their critical remarks, for innumerable discussions not only related to my work, and for their friendship.

My dearest thanks go to *Angelika* and *Reiner Elting*, *Vera* and *Jochen Elting*, and *Tanja* and *Martin Elting* for being a never-failing source of support and encouragement, for seeing me through the, fortunately, few bad days, and for their warm friendship.

I would like to thank *Josef Bielefeld*, managing director of the *Technische Informationssysteme GmbH*, for teaching me what software development is really about, for his fatherly treatment, and for letting me return to the university without issuing reproaches.

Thanks go also to *Werner Brand* who proofread the German appendix.

Finally, I would like to express my wholehearted thanks to my dear parents *Hans-Joachim* and *Anna Westphal* for their constant support and encouragement throughout all the years that led me to this thesis.

Thank you, all of you.

# Abstract

An important requirement for the expression of cognitive structures is the ability to form mental objects by rapidly binding together constituent parts. In this sense, one may conceive the brain's data structure to have the form of graphs whose nodes are labeled with elementary features. These provide a versatile data format with the ability to render the structure of any mental object. Because of the multitude of possible object variations the graphs are required to be dynamic. Upon presentation of an image a so-called model graph should rapidly emerge by binding together memorized subgraphs derived from earlier learning examples driven by the image features. In this model, the richness and flexibility of the mind is made possible by a combinatorial game of immense complexity. Consequently, emergence of model graphs is a laborious task which, in computer vision, has most often been disregarded in favor of employing model graphs tailored to specific object categories like faces in frontal pose. Invariant recognition or categorization of arbitrary objects, however, demands dynamic graphs.

In this work we propose a form of graph dynamics, which proceeds in three steps. In the first step position-invariant feature detectors, which decide whether a feature is present in an image, are set up from training images. For processing arbitrary objects, these features are small regular graphs, termed *parquet graphs*, whose nodes are attributed with Gabor amplitudes. Through combination of these classifiers into a linear discriminant that conforms to LINSKER's infomax principle a weighted majority voting scheme is implemented. The network is well suited to quickly rule out most irrelevant matches and only leaves the ambiguous cases, so-called *model candidates*, to be processed in a third step using a rudimentary version of elastic graph matching, a standard correspondence-based technique for face and object recognition. To further differentiate between model candidates with similar features it is asserted that the features be in similar spatial arrangement for the model to be selected. Model graphs are constructed dynamically

by assembling model features into larger graphs according to their spatial arrangement. The model candidate whose model graph attains the best similarity to the input image is chosen as the recognized model.

We report the results of experiments on standard databases for object recognition and categorization. The method achieved high recognition rates on identity, object category, pose, and illumination type, provided that individual object variations are sufficiently covered by learning examples. Unlike many other models the presented technique can also cope with varying background, multiple objects, and partial occlusion.

# Contents

<b>Acknowledgments</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Outline . . . . .	4
1.2 List of Symbols . . . . .	5
<b>2 Elastic Graph Matching</b>	<b>9</b>
2.1 Model Graph . . . . .	10
2.1.1 Node Labels . . . . .	11
2.1.2 Similarity Function . . . . .	14
2.2 Matching . . . . .	14
2.3 Bunch Graph . . . . .	19
2.4 Discussion . . . . .	20
<b>3 Feature-Driven Emergence of Model Graphs</b>	<b>23</b>
3.1 Learning Set, Partitionings, and Categories . . . . .	28
3.2 Parquet Graphs . . . . .	30

---

3.2.1	Similarity Function . . . . .	33
3.2.2	Local Feature Detectors . . . . .	33
3.3	Learning a Visual Dictionary . . . . .	35
3.3.1	Feature Calculators . . . . .	35
3.3.2	Feature Vectors . . . . .	37
3.4	Preselection Network . . . . .	37
3.4.1	Neural Model . . . . .	40
3.4.2	Position-Invariant Feature Detectors . . . . .	40
3.4.3	Weighting of Feature Detectors . . . . .	41
3.4.4	Neurons, Connectivity, and Synaptic Weights . . . . .	45
3.4.5	Saliencies . . . . .	45
3.4.6	Synaptic Plasticity . . . . .	51
3.4.7	Selection of Salient Categories and Model Candidates . . . . .	53
3.4.8	Accelerated Feature Search . . . . .	55
3.5	Verification of Model Candidates . . . . .	64
3.5.1	Construction of Graphs . . . . .	64
3.5.2	Matching . . . . .	67
3.5.3	Model Selection . . . . .	68
3.6	Parameterization . . . . .	71
3.6.1	Gabor Features . . . . .	71
3.6.2	Parquet Graphs . . . . .	71
3.6.3	Visual Dictionaries . . . . .	71
3.6.4	Accelerated Feature Search . . . . .	72
<b>4</b>	<b>Object Recognition</b>	<b>73</b>
4.1	Review of Literature . . . . .	73



---

4.2	Experimental Setting . . . . .	75
4.3	Experiments . . . . .	77
4.3.1	Recognition of Single Objects . . . . .	78
4.3.2	Recognition of Single Objects using Majority Vote as Verification Method . . . . .	82
4.3.3	Recognition of Scaled Single Objects . . . . .	82
4.3.4	Recognition of Single Objects with Sparse Learning Sets	84
4.3.5	Recognition of Single Objects with Sparse Visual Dic- tionaries . . . . .	86
4.3.6	Recognition of Multiple Objects . . . . .	88
4.3.7	Recognition of Partially Occluded Objects . . . . .	94
4.4	Discussion . . . . .	94
<b>5</b>	<b>Object Categorization</b>	<b>101</b>
5.1	Review of Literature . . . . .	102
5.2	Experimental Setting . . . . .	102
5.3	Experiments . . . . .	103
5.3.1	Categorization Using Hierarchically Organized Cate- gories . . . . .	103
5.3.2	Categorization Using Single-Element Categories . . . .	104
5.4	Discussion . . . . .	109
<b>6</b>	<b>Estimation of Pose and Illumination of Human Faces</b>	<b>111</b>
6.1	Experimental Setting . . . . .	112
6.2	Experiments . . . . .	113
6.2.1	Estimation of Pose and Illumination Type Using Pre- defined Categories . . . . .	113
6.2.2	Estimation of Pose and Illumination Type Using Single- Element Categories . . . . .	115

---

6.3 Discussion . . . . .	116
<b>7 Summary and Future Work</b>	<b>119</b>
<b>A Anhang in deutscher Sprache</b>	<b>123</b>
A.1 Zusammenfassung der Dissertation . . . . .	123
A.1.1 Einleitung . . . . .	124
A.1.2 Elastische Graphenanpassung . . . . .	124
A.1.3 Emergenz von Modellgraphen . . . . .	125
A.1.4 Objekterkennung . . . . .	127
A.1.5 Objektkategorisierung . . . . .	127
A.1.6 Schätzung von Pose und Beleuchtung menschlicher Gesichter . . . . .	128
A.1.7 Zusammenfassung und Ausblick . . . . .	128
A.2 Lebenslauf . . . . .	131
<b>List of Figures</b>	<b>133</b>
<b>List of Tables</b>	<b>137</b>
<b>Bibliography</b>	<b>139</b>
<b>Previously Published Contents of this Thesis</b>	<b>147</b>

# Chapter 1

## Introduction

*The nervous system is organized (or organizes itself)  
so that it computes a stable reality.*

Heinz von Foerster

We live in a world of composite structures. For instance, the computer I write this thesis on is an assembly consisting of a processor, a hard disk, a number of circuit boards and so forth, which, provided they are properly put together and are functional, constitute a working machine. The same is the case for all physical objects. Even the human body, like all other creatures, is hierarchically composed of cells, organs, subsystems of interacting organs and so on, that, configured according to the rules of anatomy, constitute a viable organism. The same applies for the non-physical. For instance, in western languages, only twenty-six characters and a half-dozen symbols are required to compose syllables, words, sentences, paragraphs, chapters, this thesis, and any story on any subject one can ever possibly imagine. At the same time, from a mere combinatorial point of view, the majority of arrangements violates grammatical rules, which makes them illegitimate. We give another example. In philosophy, schema theory (PIAGET, 1975a,b) states that the ability to perform an action, termed *skill*, for instance, to grasp an object, is implemented by a schema, which is hierarchically composed out of simpler ones. These can be innate or learned. Schema theory thus explains the ability of humans to learn ever more complex skills, for instance, to drive a car or to play a musical instrument, with the ability to purposefully compose schemas in a hierarchical fashion.

Organization by composition is so ubiquitous that it has been suggested to be fundamental to cognition as well:

*Compositionality refers to our ability to construct mental representations, hierarchically, in terms of parts and their relations. The “rules” of composition are such that (i) we have at our disposal an infinite repertoire of hierarchically constructed entities [...] and (ii) allowable constructions nevertheless respect specific constraints, whereby overwhelmingly most combinations are made meaningless.* (BIENENSTOCK and GEMAN, 1995)

Cognition may thus be understood as a process in which the brain actively constructs *mental representations*. This idea is absent in early neural network models like (PITTS and MCCULLOCH, 1947; ROSENBLATT, 1962; FUKUSHIMA et al., 1983). This hypothesis is backed by psychophysical experiments that prove that some recognition tasks take distinctly longer than others. For instance in (TREISMANN and GELADE, 1980) human subjects were presented combinations of green and red crosses. Afterwards, 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. When the presentation times were reduced below some 50 milliseconds the performance degraded in a remarkable fashion. The subjects could still decide if they had seen cross and circle or only crosses and that these had the same or different colors. However, the assignment of color to the cross or circle dropped to chance level. This result can be interpreted with the assumption that the construction of a suitable representation that correctly associates the visual features ‘cross’, ‘circle’, ‘red’, and ‘green’ according to the presented visual scene takes more time than the mere detection of uncombined features. This findings cannot be explained with conventional models of brain function: once developed to their final state as pattern recognizers, the processing time needed to classify an input pattern is practically constant, i.e., they implement a simple stimulus-response scheme. Although these models for their own part have been successful in the task of invariant pattern recognition they run into problems when confronted with more realistic problems, for instance, real images. The reason for this is that the range of invariances achieved by the brain is so large that it cannot be covered with enough examples for the network to learn all of them. This can be alleviated by introducing extra neurons every time a new invariance is needed. This is nicely demonstrated with the *neocognitron* in (FUKUSHIMA et al., 1983). However, in order to achieve a realistic system, the amount of new cells to be introduced to cover

the whole spectrum of invariances would soon exhaust the total number of cells available. In this respect, sharing constituent-encoding cells among representations can be considered as the brain's method to conserve with its rich but nonetheless limited resources. Sharing resources, however, implies the necessity of active construction.

*Binding* (VON DER MALSBERG, 1981, 1999), a neural mechanism that allows to correctly associate constituent entities with each other, is a fundamental requirement for compositionality. The classical example of binding is given by ROSENBLATT (1962). It is concerned with a visual scene containing a red triangle and a blue square. The mere coactivation of four entities representing the four visual features 'red', 'blue', 'triangle', and 'square' would lead to a *superposition catastrophe* (VON DER MALSBERG, 1999) that is, in this case, the inability to distinguish a scene containing a red triangle and a blue square from a scene containing a red square and a blue triangle. Composition is thus more than coactivation of constituents. In order to share constituents among different representations at different points in time, binding needs to be *dynamical*. In ROSENBLATT's example the features 'red' and 'triangle' are to be bound to each other if a visual scene containing a red triangle is presented, while the feature 'red' is to be bound to feature 'square' if a scene containing a red square is presented. Binding further needs to be *relational*, that is, qualified in terms of the correct arrangement of constituents in the composite structure. In ROSENBLATT's example the spatial arrangement of features in the representation has to reflect that in the presented scene, i.e., the feature 'red' should be located at the same position in the image plane as the feature 'triangle' and the feature 'blue' should reside at the same position as the feature 'square'.

How can compositionality be incorporated in theories of human object recognition? We give two examples. *Geon structural description* (BIEDERMAN, 1987) posits that objects and scenes are represented as an arrangement of a small number of simple, viewpoint-invariant volumetric primitives called geometric icons, or, as a shorthand, *geons*. In that theory human object recognition is described as a three-stage process: first, the object is decomposed into its individual components, second, each component is attempted to be recognized as a geon, and, third, the object is recognized as a memorized arrangement of geons. For instance, a cup may be recognized as an arrangement of a cylinder and a side-connected arc. In *elastic graph matching* (VON DER MALSBERG, 1988; LADES et al., 1993; WISKOTT, 1995; WISKOTT et al., 1997) the data structure of stored object views has the form of two-dimensional graphs, termed *model graphs*, whose nodes are la-

beled with elementary features. They provide a versatile data format with the capability to render the structure of any object. An object in an image is supposed to be recognized if the model graph represents the object well in terms of a super-threshold measure of similarity. Because of the multitude of possible object variations, like changes in identity, pose, or illumination, the graphs are required to be dynamic with respect both to shape and attributed features.

## 1.1 Outline

In this thesis we propose a form of graph dynamics that upon image presentation lets a model graph emerge that represents the object in the input image well. We demonstrate the dynamics' capability in extensive experiments. Throughout this thesis we use the terms *recognition* and *categorization* according to (PALMERI and GAUTHIER, 2004). The term *recognition* refers to a decision about an object's unique identity. Recognition thus requires subjects to discriminate between similar objects and involves generalization across some shape changes as well as physical translation, rotation and so forth. The term *categorization* refers to a decision about an object's kind. Categorization thus requires generalization across members of a class of objects with different shapes. Especially, generalization over identity is required.

In chapter 2 *elastic graph matching* and *bunch graph matching* are introduced, which are standard correspondence-based techniques for object recognition and categorization.

In chapter 3 we present a form of graph dynamics that upon image presentation lets a model graph rapidly emerge by binding together memorized subgraphs derived from earlier learning examples.

In the following three chapters the proposed graph dynamics is applied to the task of invariant object recognition (Chapter 4), to the task of object categorization (Chapter 5), and to the task of estimating pose and illumination type of human faces (Chapter 6).

In chapter 7 the thesis will be summarized.

Finally, appendix A gives an abstract of the thesis in German and the author's curriculum vitae.

## 1.2 List of Symbols

SYMBOL	DESCRIPTION	PAGE
$\mathbb{X}$	Set	
$\wp(\mathbb{X})$	Power set of $\mathbb{X}$	
$ \mathbb{X} $	Number of elements in $\mathbb{X}$	
$\emptyset$	Empty set	
$\mathbb{R}$	Set of real numbers	
$\mathbb{N}_0$	Set of natural numbers incl. 0	
$\underline{x}$	Vector	
$\underline{x}^\top$	Transposed Vector	
$(x_n)_{1 \leq n \leq N}$	Vector with $N$ components $x_n$	
$\underline{X}$	Matrix	
$(x_{n,m})_{\substack{1 \leq n \leq N \\ 1 \leq m \leq M}}$	Matrix with $N$ rows, $M$ columns, and components $x_{n,m}$	
$\mathcal{J}$	Jet	13
$\mathcal{G}^M$	Model graph	10, 66
$\mathcal{G}^I$	Image graph	66
$s_{abs}$	Function that returns the similarity between two jets which is based on the Gabor amplitudes only	14
$\mathbb{I}$	Set of images	29
$I$	Image	
$\mathbb{D}$	Learning set	29
$M$	Model image	
$K$	Number of partitionings of the learning set	29
$k$	Index of partitioning	29
$\Pi^k$	Partitioning of the learning set with index $k$	29
$C^k$	Number of categories in partitioning $\Pi^k$	29
$c$	Index of category	29
$\mathbb{C}_c^k$	Category with label $c$ of partitioning $\Pi^k$ , a subset of learning images that share a semantic property	29

SYMBOL	DESCRIPTION	PAGE
$V$	Number of nodes of a parquet graph	32
$s_{graph}$	Function that returns the similarity between two parquet graphs	33
$\varepsilon(f, f', \vartheta)$	Local feature detector, returns 1 if the similarity between parquet graph $f$ and $f'$ is greater or equal than $\vartheta$ and 0 otherwise	33
$R$	Number of feature calculators	35
$\mathbb{F}$	Set of all possible features	35
$r$	Index of feature calculator	35
$f^r$	Feature calculator with index $r$	35
$\underline{f}^r$	Feature vector computed using feature calculator $f^r$	35
$T^r$	Number of features in feature vector $\underline{f}^r$	37
$t$	Feature index	37
$f_t^r$	Feature with index $t$ in feature vector $\underline{f}^r$	37
$H(\cdot)$	Heaviside threshold function	40
$\tau_t^r$	(Position-Invariant) Feature detector with reference feature $f_t^r$	41
$\tau_t^r(I)$	Result of feature detector $\tau_t^r$ , 1 if feature $f_t^r$ can be observed in $I$ and 0 otherwise	41
$\mathcal{F}_{match}(I)$	Table of matching features	41
$\mathcal{H}_t^{r,k}$	Uncertainty of feature detector $\tau_t^r$ about choosing categories of partitioning $\Pi^k$ , SHANNON entropy	43
$i_t^{r,k}$	Contribution of feature detector $\tau_t^r$ to the decision about choosing categories of partitioning $\Pi^k$ , measure of information	43
$w_{t,c}^{r,k}$	Synaptic weight between the input neuron assigned to feature detector $\tau_t^r$ and the output neuron assigned to category $\mathbb{C}_c^k$	45



SYMBOL	DESCRIPTION	PAGE
$\underline{W}^{r,k}$	Matrix of synaptic weights between the input neurons assigned to feature detectors $\tau_t^r$ with $t \in \{1, \dots, T^r\}$ and the output neurons assigned to categories $\mathbb{C}_c^k$ with $c \in \{1, \dots, C^k\}$	45
$s_c^k(I)$	Saliency of category $\mathbb{C}_c^k$	46
$\Gamma^k(I)$	Set of salient categories of partitioning $\Pi^k$	55
$\mathbb{M}(I)$	Set of model candidates	55
$\mathcal{F}_{corr}(I, M)$	Table of corresponding features	65
$\mathbb{X}^I$	Positions of all valid nodes of image parquet graphs	65
$\mathbb{X}^M$	Positions of all valid nodes of model parquet graphs	65
$\beta^I(\underline{x})$	Bunch of jets at position $\underline{x}$ drawn from image parquet graphs	66
$\beta^M(\underline{x})$	Bunch of jets at position $\underline{x}$ drawn from model parquet graphs	66
$s_{bunch}$	Function that returns the similarity between two bunches of jets	66



## Chapter 2

# Elastic Graph Matching

*Not even the gods fight against necessity.*

Simonides

One of the most fundamental problems in computer vision is the correspondence problem:

*Given two images of the same object or of two objects of the same category, a pixel in one image corresponds to a pixel in the other if both pixels are projections of the same point on the physical object. The problem is to determine this correspondence between pixels of the given images.*

The correspondence problem occurs in a number of tasks related to vision. For instance, recognition of objects can be achieved through comparison of local image features at corresponding points. As much as a solution of the correspondence problem is desirable as difficult it has turned out to obtain one.

Elastic graph matching (VON DER MALSBERG, 1988; LADES et al., 1993; WISKOTT, 1995; WISKOTT et al., 1997), an algorithmic implementation of Dynamic Link Matching (VON DER MALSBERG, 1981, 2002), provides a successful method to solve the correspondence problem. In this approach object views are represented by graphs whose nodes are labeled with local image features. These graphs are called *model graphs*. Recognition of an object is

achieved through optimal placement of a model graph that represents the object to be recognized on the input image in terms of maximizing a measure of similarity based on local similarities between local image features at corresponding points. The process of similarity maximization is called *matching*. The object is supposed to be recognized if the final similarity value exceeds a predefined threshold. *Bunch graphs* provide a successful extension of model graphs (WISKOTT, 1995; WISKOTT et al., 1997). They combine model graphs of similar object views, for instance faces of approximately the same size in frontal pose, in a stack-like structure and are thus able to generalize over identity to some degree.

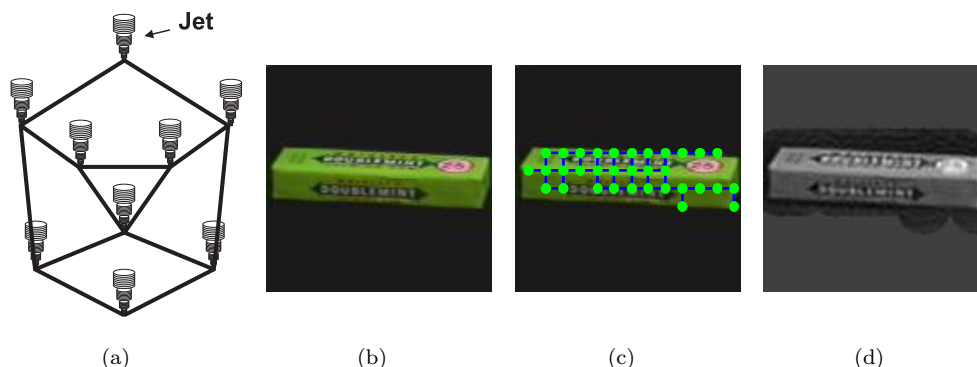
Elastic graph matching and elastic bunch graph matching have mainly been applied for the reliable recognition of human faces (PHILLIPS et al., 2000; MESSER et al., 2004). Recent research has focused on the development of object models that allow to describe object variations with few, low-dimensional parameters (WUNDRICH, 2004; TEWES, 2006).

In the following sections the elastic graph matching approach is introduced as far as the graph dynamics proposed in chapter 3 is concerned. More detailed descriptions are given in (WISKOTT, 1995; WIEGHARDT, 2001; TEWES, 2006).

## 2.1 Model Graph

In the elastic graph matching approach object views are represented by model graphs, which are two-dimensional graphs whose nodes are labeled with local image features. Usually, the nodes are labeled with the complex responses of a set of Gabor filters, that constitute a so-called *jet*. Two nodes may be connected with an edge. For face recognition, nodes are usually associated with so-called facial landmarks like the tip of the nose, the pupil of an eye, or the corners of the mouth. It is nevertheless also possible to use arbitrary landmarks as has been demonstrated in (LADES et al., 1993; LOOS, 2002). Exemplary model graphs are given in fig. 2.1. In the following, model graphs are supposed to be given in the form of eq. (2.1): a model graph  $\mathcal{G}^M$  that represents the object in an example image  $M$  is specified by a set of  $V$  tuples, i.e., a tuple specifies one node. For a node  $v$  they comprise the absolute node position  $\underline{x}_v$  and the jet  $\mathcal{J}_v$  derived at that position. Although the edges can be harnessed for shape preservation purposes they are deliberately ignored in what follows.

$$\mathcal{G}^M = \left\{ (\underline{x}_v, \mathcal{J}_v) \mid 1 \leq v \leq V \right\} \quad (2.1)$$

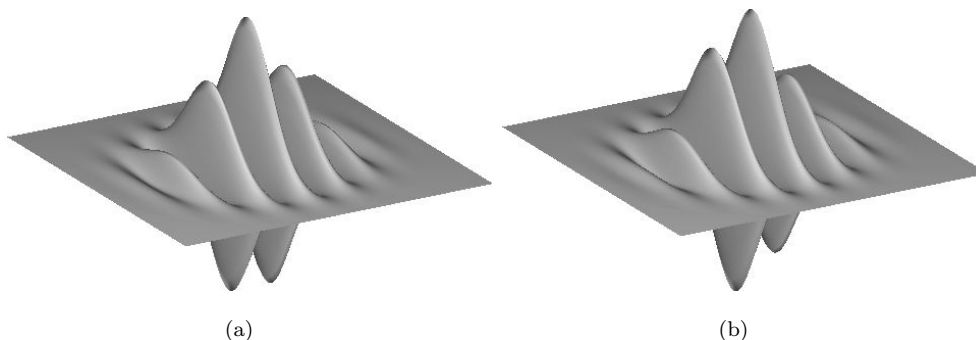


**Figure 2.1: Model Graph** — In (a) a face graph is shown schematically. For clarity, the model graph comprises only nine nodes while a typical face graph consists of up to 60 nodes. In (c) a model graph for the object in (b) is given. The reconstruction from the model graph in (d) demonstrates that the model graph represents the object well. The model graph has been computed by the graph dynamics proposed in chapter 3. Like all reconstructions in this thesis, the reconstruction in (d) has been computed with the algorithm from (PÖTZSCH *et al.*, 1996).

### 2.1.1 Node Labels

Each node of a model graph is labeled with a feature vector that describes the texture in the node's surrounding. Features are the complex responses of a set of Gabor filters. They constitute a so-called *jet* (LADES *et al.*, 1993). A Gabor function has the form of a plane wave restricted by a Gaussian envelope (Eq. (2.2)). An example of a two-dimensional Gabor function is given in fig. 2.2. Gabor functions are well-suited for image representation because of their properties regarding information theory (LINSKER, 1988; OLSHAUSEN and FIELD, 1996) and because of their biological relevance (HUBEL and WIESEL, 1962; JONES and PALMER, 1987). Fourier-transformed Gabor functions take the form of Gaussians in the frequency domain.

$$\psi_{\underline{k}}(\underline{x}) = \frac{k^2}{\sigma^2} \exp\left(-\frac{k^2 x^2}{2\sigma^2}\right) \left[ \exp(i\underline{k}^\top \underline{x}) - \exp\left(-\frac{\sigma^2}{2}\right) \right] \quad (2.2)$$



**Figure 2.2: Gabor Function** — Gabor functions have the shape of a plane wave restricted by a Gaussian envelope. (a) shows the real part, (b) the imaginary part of a two-dimensional Gabor function.

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 (Eq. (2.3)). The domain of integration is the image plane.

$$\mathcal{I}_{\underline{k}}(\underline{x}) = \int_{\mathbb{R}^2} I(\underline{x}') \psi_{\underline{k}}(\underline{x} - \underline{x}') d^2x' \quad (2.3)$$

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  $\phi_l$  and frequency  $k_m$  (Eq. (2.4)).

$$\underline{k}_{m,l} = k_m \cdot \begin{pmatrix} \cos \phi_l \\ \sin \phi_l \end{pmatrix} \quad (2.4)$$

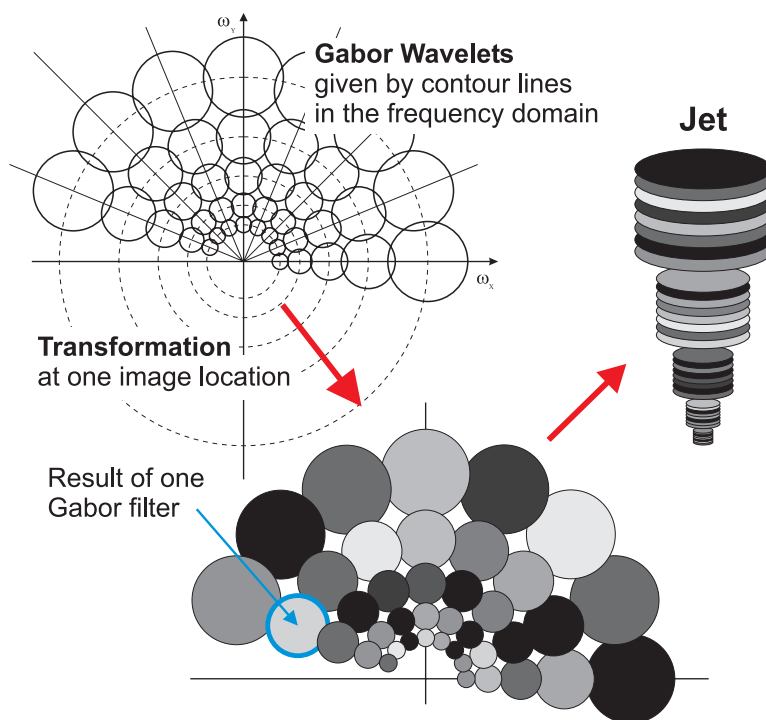
The finite set of filters is chosen such that the direction space is sampled homogeneously (Eq. (2.5)) and the frequencies are sampled geometrically (Eq. (2.6)).

$$\phi_l = \frac{\pi \cdot l}{L} \quad \text{with } l \in \{0, \dots, L-1\} \quad (2.5)$$

$$k_m = \frac{k_{max}}{(k_{step})^m} \quad \text{with } m \in \{0, \dots, M-1\} \quad (2.6)$$

The remaining parameters are chosen according to (LADES et al., 1993; WISKOTT, 1995). This parameterization is used in the entire thesis.

$$k_{step} = \sqrt{2} \quad k_{max} = \frac{\pi}{2} \quad L = 8 \quad M = 5 \quad \sigma = 2\pi \quad (2.7)$$



**Figure 2.3: Creation of Jets** — The figure shows the calculation of a jet for one position in the image with Gabor wavelets for  $L = 8$  directions and  $M = 5$  frequencies resulting in a vector of  $M \cdot L = 40$  complex values. This vector of filter responses is called jet. The wavelets are represented by contour lines in the frequency domain.

The complex responses of this set of Gabor filters at a given location in an image constitute a so-called *jet* (LADES et al., 1993). These jets are vectors of  $M \cdot L$  complex numbers which are characterized by their associated set of filters. The creation of a Gabor jet is illustrated in fig. 2.3. In the following jets are supposed to be given in the form of eq. (2.8): the complex filter responses are expressed in terms of *amplitudes*  $a_{\underline{k}_{m,l}}$  and *phases*  $\phi_{\underline{k}_{m,l}}$ . Whenever possible we omit the position  $\underline{x}$  and write  $\mathcal{J}$  instead of  $\mathcal{J}(\underline{x})$ .

$$\mathcal{J}(\underline{x}) = \left( \mathcal{I}_{\underline{k}_{m,l}}(\underline{x}) \right)_{0 \leq m < M, 0 \leq l < L} =: \left( a_{\underline{k}_{m,l}} \cdot e^{i \cdot \phi_{\underline{k}_{m,l}}} \right)_{0 \leq m < M, 0 \leq l < L} \quad (2.8)$$

## 2.1.2 Similarity Function

For the assessment whether two points from two different images actually correspond to each other, a measure of similarity between local features is needed. It is commonly introduced by so-called *similarity functions* that map two jets into the interval  $[0, 1]$ . They are required to be invariant against image changes irrelevant for solving the correspondence problem such as modification of the total brightness or image contrast. In order to ease the search of correspondences, smooth changes of the similarity values for local transformations such as translation, scaling, and rotation are required. Moreover, similarity functions are expected to be symmetrical and their result has to be 1 whenever the arguments are identical. A number of similarity functions have meanwhile been proposed (LADES et al., 1993; WÜRTZ, 1995; WISKOTT, 1995). In this thesis we exclusively use the measure of similarity that is solely based on the amplitudes of the filter responses (Eq. (2.9)). This measure of similarity allows for smooth similarity potentials with fairly wide maxima. The similarity potentials are exemplarily given in fig. 2.4.

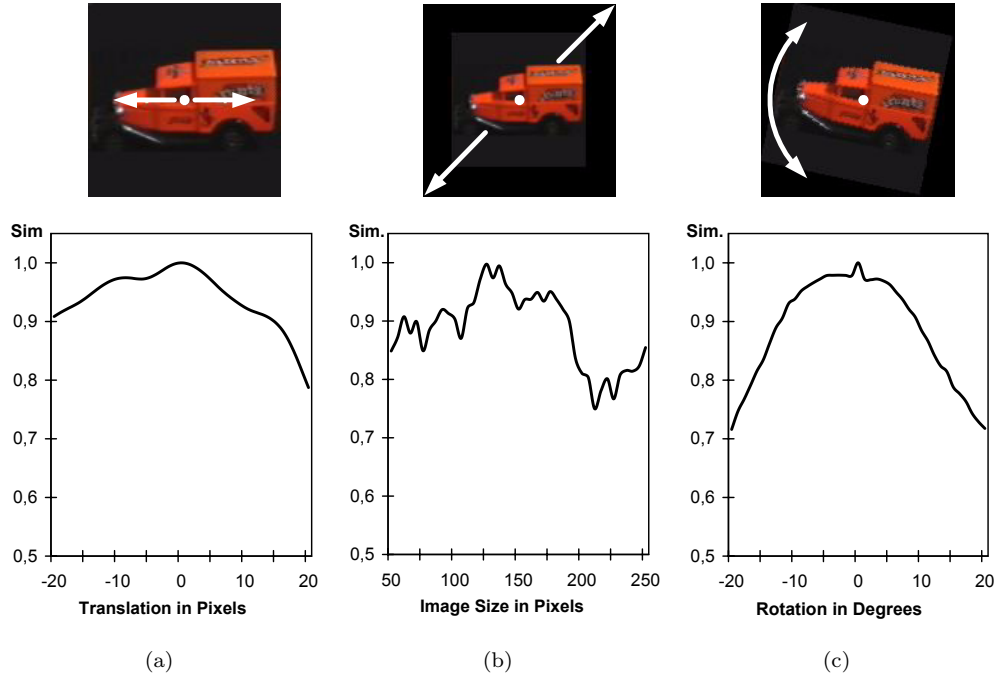
$$s_{abs}(\mathcal{J}, \mathcal{J}') = \frac{\sum_{m,l} a_{\underline{k}_{m,l}} \cdot a'_{\underline{k}_{m,l}}}{\sqrt{\sum_{m,l} a_{\underline{k}_{m,l}}^2} \cdot \sqrt{\sum_{m,l} a'_{\underline{k}_{m,l}}^2}} \quad (2.9)$$

## 2.2 Matching

In elastic graph matching recognition of an object in an input image is achieved through optimal placement of a model graph that represents the object to be recognized as a deformable template on the input image. To this end a measure of similarity that is based on local similarities between jets at corresponding points is maximized. The process of similarity maximization is called matching. It consists of several steps, so-called *moves*. Each move modifies the placement of the model graph's nodes in order to maximize the measure of similarity. The order of moves and their parameterization is specified in a *matching schedule*, which is usually built up to pursue a coarse-to-fine strategy. An overview of moves is given in fig. 2.5.

In this thesis the *scan global move* over the whole image plane on a coarse grid is of particular interest. In the graph dynamics proposed in chapter 3 it is used to assert that the features be in similar spatial arrangement for the model to be selected while in elastic graph matching it is generally applied





**Figure 2.4: Similarity Potentials of the Jet Similarity Function** — The potentials of the jet similarity function  $s_{abs}$  (LADES *et al.*, 1993) for a single jet taken from the center of the given image with respect to translation, scale, and rotation are displayed. The image has arbitrarily been chosen from the COIL-100 database (NENE *et al.*, 1996). In the translation case (a) a jet has been extracted and compared to jets derived at positions along a line of increasing distance from the original position. The sensitivity to scaling (b) has been tested by comparing the original jet to jets at corresponding positions in scaled images. The original image is  $128 \times 128$  pixels in size. For the chosen image the measure of similarity does not yield a smooth similarity potential in the case of scaled objects. Rotation (c) has been tested by rotating the image around the point at which the original jet has been extracted. The original jet has been compared to jets extracted at the same pixel position in the rotated images.

first in order to find the object in the input image. To this end the rigid model graph is iteratively moved over the entire image plane on a coarse grid. For each translation the similarity between the model graph and the Gabor wavelet transformed input image is computed. 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 (Eq. (2.10)).

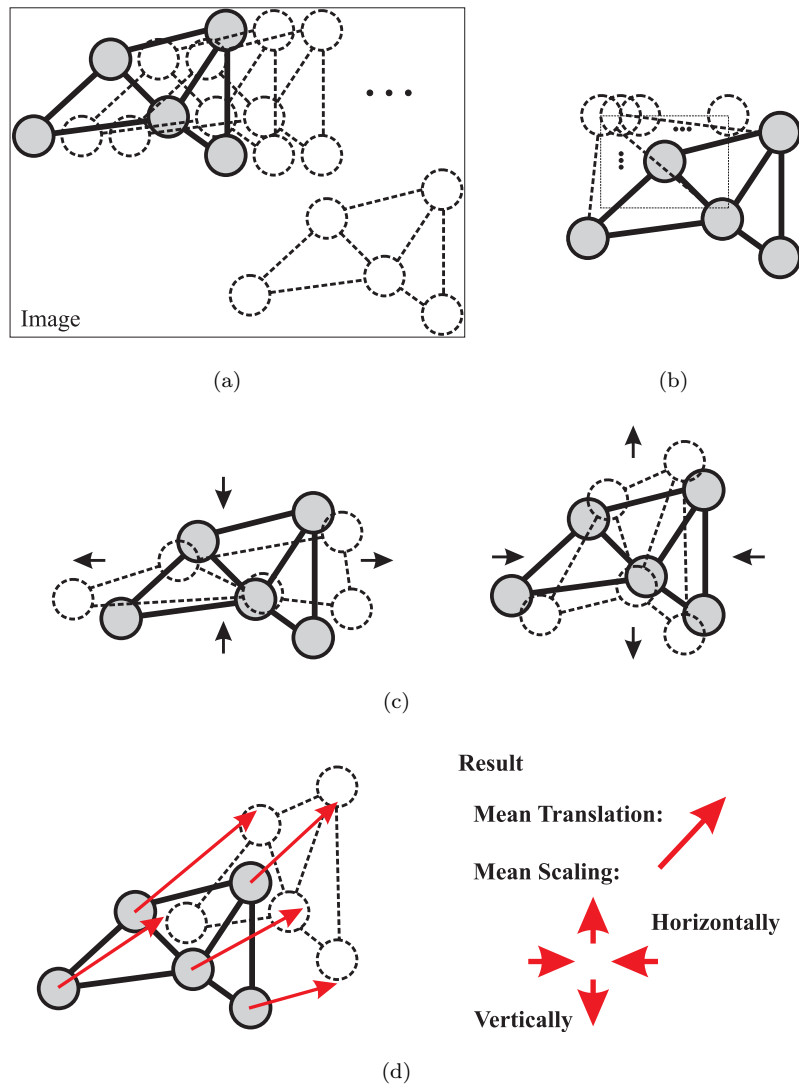
$$\underline{t}_0 = \left( \min_v \{(\underline{x}_v)_x\}, \min_v \{(\underline{x}_v)_y\} \right)^\top \quad (2.10)$$

The total similarity between the model graph and the Gabor wavelet transformed input image with respect to a given translation vector  $\underline{t}$  is defined as the average of local similarities at corresponding points (Eq. (2.11)). Let  $s(\cdot)$  denote some similarity function that compares two Gabor jets. This similarity function may, for instance, be the one given in eq. (2.9).

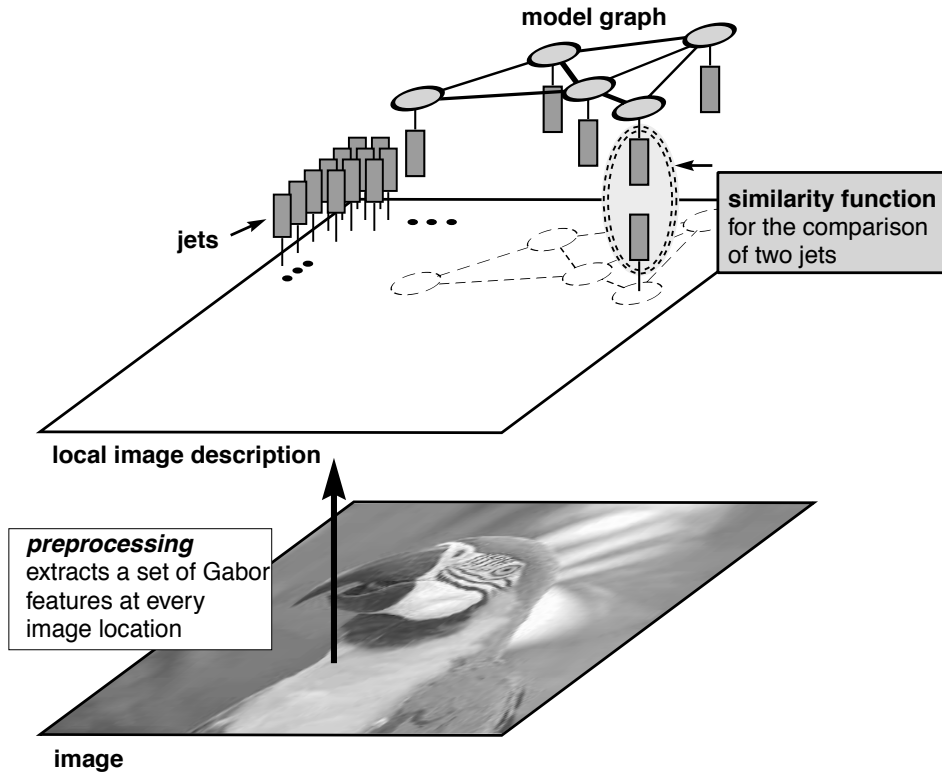
$$s(I, M, \underline{t}) = |\mathcal{G}^M|^{-1} \cdot \sum_{(\underline{x}_v, \mathcal{J}_v) \in \mathcal{G}^M} s(\mathcal{J}^I(\underline{x}_v - \underline{t}_0 + \underline{t}), \mathcal{J}_v) \quad (2.11)$$

In order to find the object in the input image, the model graph is iteratively translated about a displacement vector in the image plane so that the total measure of similarity becomes maximal (Eq. (2.12)). The model graph thus moves to the position in the input image where the object is most likely located. Let  $s_{best}(I, M)$  denote the similarity attained at that position. If that similarity exceeds a given threshold the object is supposed to be found and recognized. The displacement vectors  $\underline{t}$  stem from the set  $\mathbb{G}$  of the scan global move's grid points.

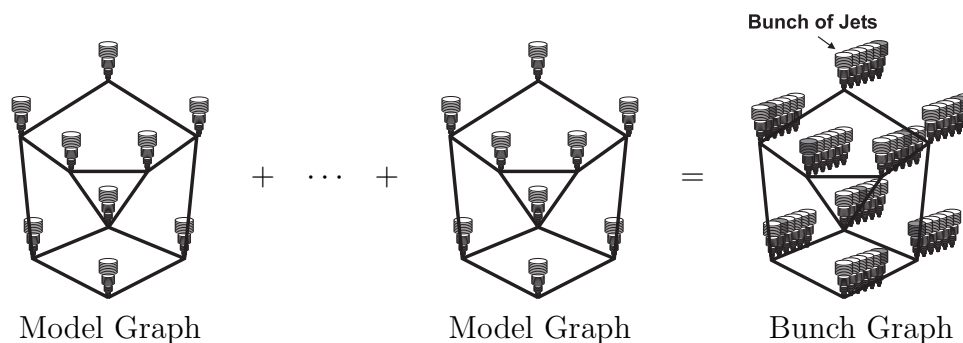
$$s_{best}(I, M) = \max_{\underline{t} \in \mathbb{G}} \{s(I, M, \underline{t})\} \quad (2.12)$$



**Figure 2.5: Overview of Moves** — The figure shows the different moves of a model or bunch graph. (a) *Scan Global Move*: the graph is moved over the entire image or over a specific region. The movement can be restricted to an adjustable raster. The scan global move is used typically for the determination of the first position of an object in the image. (b) *Scan Local Move*: for each node of the graph an optimal position is looked up. This search can be limited to a certain area around the node. Usually, the scan local move is performed as the last step of the matching schedule. (c) *Scan Scale Move*: the graph is scaled either as a whole or independent horizontally and/or vertically. (d) *Disp Scale Move*: a displacement vector for each node is calculated. These vectors are averaged to determine a displacement and scale vector for the entire graph.



**Figure 2.6: Overview of Elastic Graph Matching** — *In elastic graph matching recognition of an object in an input image is achieved through optimal placement of a model graph that represents the object to be recognized as a deformable template on the preprocessed input image. To this end a measure of similarity that is based on the local similarities between jets at corresponding points is maximized. The process of similarity maximization is called matching. It consists of several steps, so-called moves. Each move modifies the placement of the model graph's nodes in order to maximize the measure of similarity. The order of moves and their parameterization is specified in a matching schedule which is usually built up to pursue a coarse-to-fine strategy. The object is located at that position in the input image at which the similarity becomes maximal. If the final similarity value exceeds a given threshold the object is supposed to be recognized.*

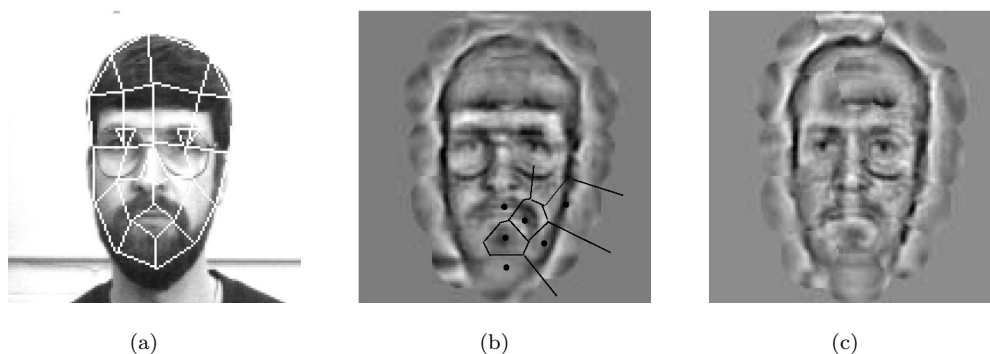


**Figure 2.7: Bunch Graph** — A bunch graph is a stack-like structure composed out of model graphs. The bunch graph in this figure is composed out of six model graphs with the topology of the schematic face graph given in fig. 2.1 (a). It is highly important that the model graphs' nodes are exactly positioned on the landmarks. While matching, the jet of a given bunch that attains the highest similarity to the corresponding image jet is selected independently from the selection of the most similar jet in the other bunches, here illustrated by gray shading. This selection depends on the object the bunch graph was matched with.

## 2.3 Bunch Graph

Model graphs have proven to perform well for *recognition* of objects, especially human faces. However, for *categorization* of objects, for instance to distinguish faces from non-faces, they suffer from the deficiency that they encode only one identity and, hence, are not able to cover intra-category variations, which can be considerable. For instance, human faces can have glasses, beards, different expressions, different age, gender, or face form. To this end a bunch graph provides a representation of a whole category of objects. Fig. 2.8 illustrates this statement for the category of human faces in frontal pose. The bunch graph combines model graphs of similar object views in a stack-like structure. It is very important that all constituting model graphs have the same topology and the nodes code the same local features, i.e., they are positioned on the same landmarks. An illustration of the bunch graph concept is given in fig. 2.7.

Elastic graph matching can easily be adapted to bunch graphs. Basically, only the local similarity function needs to be modified. Like in elastic graph



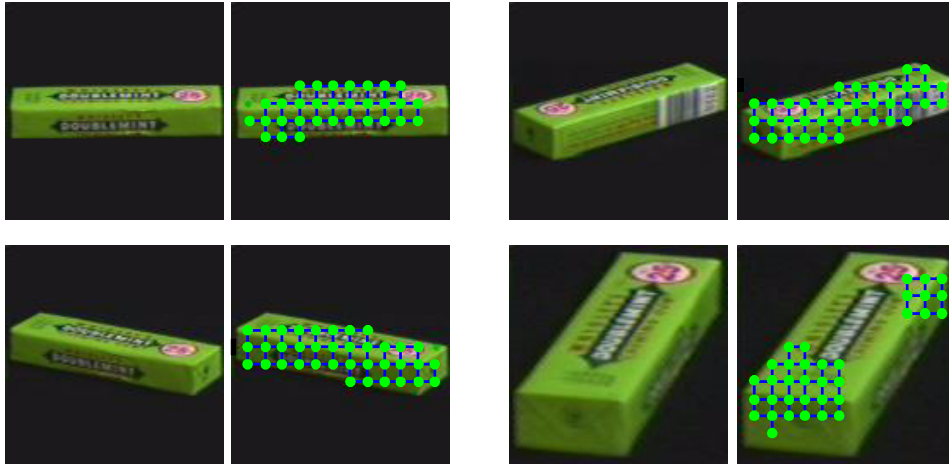
**Figure 2.8: Reconstruction from a Bunch Graph** — (a) shows a face with its associated model graph. (b) shows the reconstruction of the image from the model graph in (a). (c) shows the reconstruction of the jets of a bunch graph, which fits best on the face in (a). The used bunch graph contains approximately 100 different faces, however, not that in (a).

---

matching the similarity between the bunch graph and the Gabor wavelet transformed input image is given by the average of local similarities at corresponding points. The local similarities are evaluated as nearest neighbor similarities: the similarity between a bunch of jets and a jet at the corresponding position in the Gabor wavelet transformed input image is given by the maximum of all similarities between the image and bunch jets. The selection of the most similar jet in a bunch is independent of the selection in the other bunches.

## 2.4 Discussion

Elastic graph matching and elastic bunch graph matching have proven to perform well for object recognition and categorization under the implicit assumption that the variations of objects to be recognized or categorized can be covered with a small ensemble of suited model or bunch graphs. These are matched in succession to the input image and the graph that attains the highest similarity is supposed to represent the object in the input image best. Problems arise if that assumption is not applicable, for instance, if arbitrary objects are to be recognized with full pose invariance. This is illustrated in fig. 2.9. It is certainly not sensible, neither in a biological nor in



**Figure 2.9: Examples of Emerged Model Graphs** — *Even though the object is always the same, the model graphs differ considerably with respect both to shape and attributed features depending on the object’s pose. The model graphs were computed by the graph dynamics proposed in chapter 3.*

---

a computational sense, to store one model graph per learning image in order to cope with the multitude of possible object variations.

Through the years, several approaches have been proposed to address this problem. PETERS (2001) suggests to store only the model graphs of so-called canonical views. Others propose parameterized object models that allow to describe object variations with few, low-dimensional parameters. For instance, WUNDRICH (2004) proposes a model that allows to parameterize illumination type and head pose while the flexible object model of TEWES (2006) allows to parameterize facial gestures and head pose. These models are, however, purposely tailored to the category of human faces; in their current state they can hardly handle a variety of arbitrary objects. To this end a graph dynamics is desirable that upon image presentation lets a model graph rapidly emerge by binding together memorized subgraphs derived from earlier learning examples. Such a graph dynamics will be proposed in the following chapter.





## Chapter 3

# Feature-Driven Emergence of Model Graphs

*When you have eliminated the impossible, whatever remains, however improbable, must be the truth.*

Sir Arthur Conan Doyle

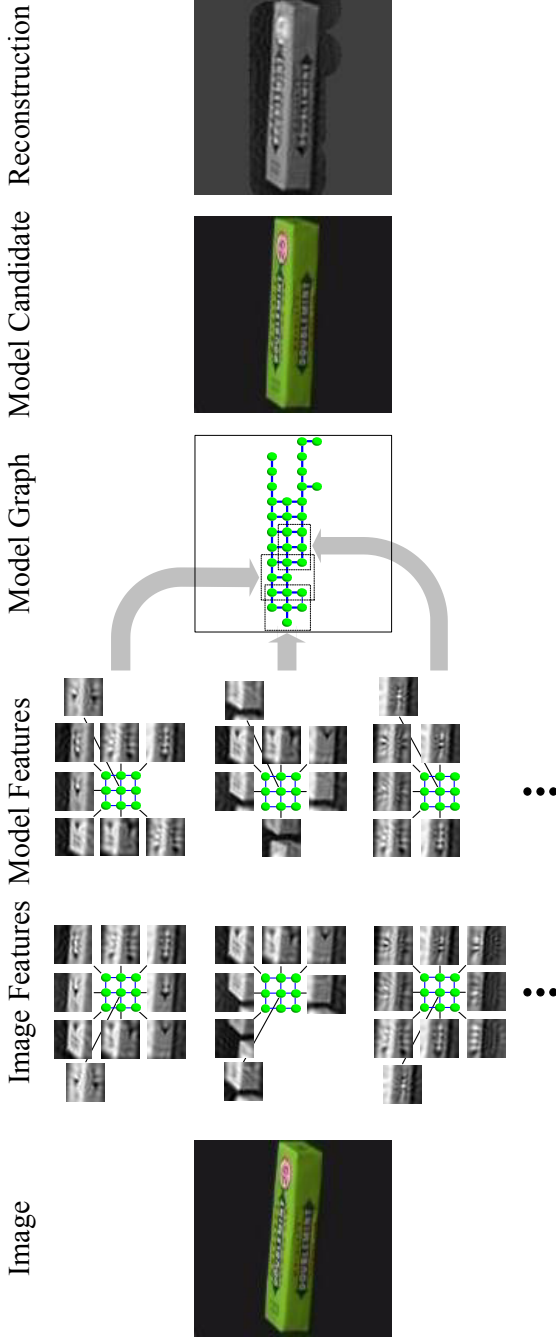
In chapter 1 we motivated that compositionality, the ability to form mental objects by rapidly binding together constituent parts (BIEDERMAN, 1987; BIENENSTOCK and GEMAN, 1995), is an important requirement for the expression of cognitive structures. In this sense, one may conceive the brain's data structure to have the form of graphs whose nodes are labeled with local image features. These graphs are termed model graphs.

This data format has been used for visual object recognition (SHAPIRO and HARALICK, 1981; BUNKE, 1983; ESHERA and FU, 1986; MESSMER and BUNKE, 1998) and in the Dynamic Link Matching approach (VON DER MALSBURG, 1981, 1988, 2002; LADES et al., 1993; WISKOTT et al., 1997). In all these approaches the data structure of stored object views has the form of model graphs. They provide a versatile data format with the capability to render the structure of any object. Because of the multitude of possible object variations like changes in identity, pose, or illumination, the graphs are required to be dynamic with respect both to shape and attributed features.

Upon presentation of an image a so-called model graph should rapidly emerge by binding together memorized subgraphs derived from earlier learning examples driven by the image features. Emergence of model graphs is a laborious task which, in computer vision, has most often been disregarded in favor of employing model graphs tailored to specific object categories like faces in frontal pose (LADES et al., 1993; WÜRTZ, 1997; WISKOTT et al., 1997). Recognition or categorization of arbitrary objects, however, demands dynamic graphs, i.e., more emphasis must be laid on the question of how model graphs are created from raw image data.

Relatively little work has been done on the dynamic creation of model graphs. The object recognition system proposed in (VON DER MALSBERG and REISER, 1995) is based on Dynamic Link Matching supplied with object memory. While learning novel objects a so-called fusion graph is created through iteratively matching image graphs with the fusion graph and grafting non-matched parts of the image graphs into the fusion graph. When an object is to be recognized, one or more image graphs are compared against model memory via graph matching, implemented by dynamic links. The matching parts of the fusion graph thus constitute the model graph for the object contained in the input image. The system has proven to perform well for a small number of object views. During both learning and recognition the objects are required to be placed in front of a plain background.

A different approach is the creation of model graphs with minimal user-assistance (LOOS, 2002). In that method, a growing neural gas (MARTINETZ and SCHULTEN, 1991; FRITZKE, 1997) is used to determine shape and topology of a model graph. Binarized difference images derived from two consecutive images of the same moving object are used as an input to a growing neural gas whose nodes are attracted to super-threshold frame differences. Upon an user-initiated event, Gabor jets are extracted at the node positions and the produced model graph is stored in a model database. During recognition, model graphs are matched in succession with the input image. The compositional aspect is thus prominent while learning novel objects but is absent during recognition. A rudimentary version of model graph dynamics is also present in (WÜRTZ, 1997), where model graphs are adapted to segmentation masks in order to ignore background influences.



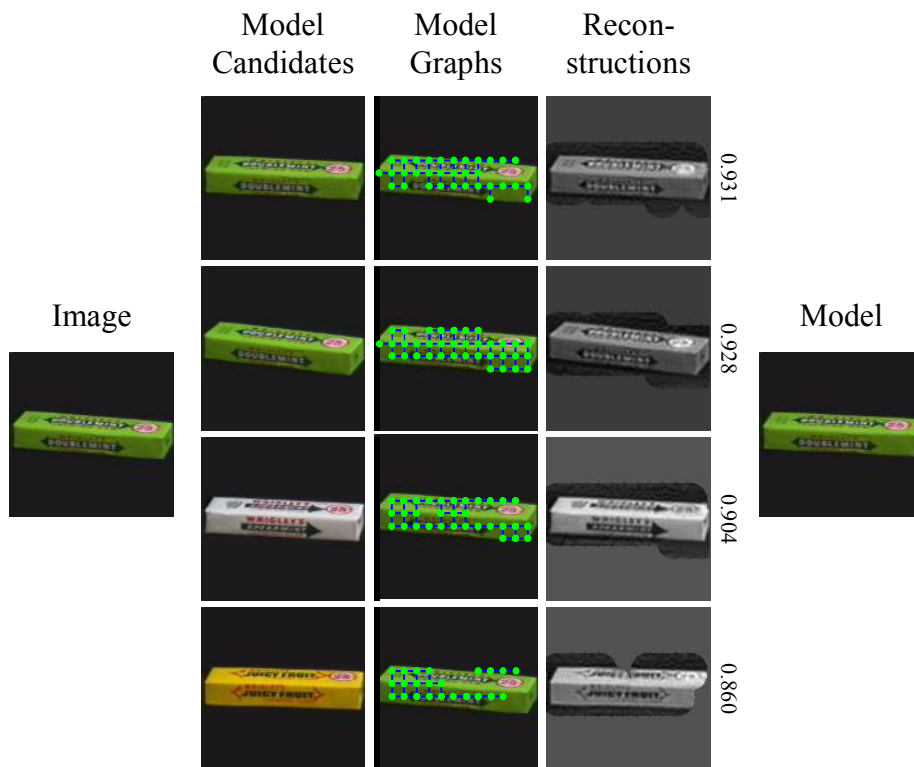
**Figure 3.1: Feature-Driven Emergence of Model Graphs** — Upon presentation of an image (first column) a model graph (fourth column) should rapidly emerge by binding together (arrows) memorized subgraphs, termed a model graph, derived from earlier learning examples (third column) that match with the image features (second column). Column six shows the reconstruction from the model graph. The graph dynamics itself proceeds in three steps. In the first step, position-invariant feature detectors are set up from training images. Through combination of these classifiers into a single-layer perceptron, a weighted majority voting scheme is implemented. It allows for preselection of salient learning examples, so-called model candidates (fifth column), and likewise for preselection of salient categories the object in the presented image supposedly belongs to. Each model candidate is verified in a third step using a rudimentary version of elastic graph matching. To further differentiate between model candidates with similar features, similar spatial arrangement for the model features is asserted. Reconstruction and the model candidate contain the same object in the same pose, which is slightly different from the one in the input image.

WEBER et al. (2000) propose a system that creates an object model in a probabilistic framework. The technique uses mixtures of collaborating probabilistic object models, termed *components*. Highly textured regions, so-called *parts*, are employed as local features. They are automatically extracted from earlier learning images. Each component is an expert for a small ensemble of object parts. In order to describe an object in an image several components need to be active. Model parameters, the parameters of the incorporated probability densities, are iteratively learned using expectation maximization (EM). Categorization of an object is based on the maximum a posteriori (MAP) decision rule: the object in the input image is supposed to belong to the category whose object model attained maximal a posteriori probability.

TANG and TAO (2005) employ a graph dynamics for object tracking. It is formulated in a maximum a posteriori framework using a hidden Markov model: the tracker estimates the object's state, expressed by a model graph, through maximization of a posterior probability. New features are added to the model graph if they can reliably be observed in the hidden Markov model's time window. Similarly, repeatedly non-matching features are removed from the model graph.

Recognition methods relying on graph matching are *correspondence-based* in the sense that image point correspondences are estimated before recognition is attempted. This estimation is usually only possible on the basis of the spatial arrangement of elementary features. There is also a class of recognition algorithms which are purely *feature-based* and completely disregard feature arrangement. A prominent example is SEEMORE (MEL, 1997). There it is shown that a simple neural network can distinguish objects in a purely feature-based way if enough feature types are employed. As a model for recognition and categorization 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 et al., 1996; THORPE and THORPE, 2001). These methods, however, encounter problems in the case of multiple objects and highly structured backgrounds. From the point of view of pattern recognition, feature-based methods are *discriminative* while graph matching is *generative* (ULUSOY and BISHOP, 2005).

It is reasonable to assume that feedforward processing is applied as far as it goes by excluding as many objects as possible and that only ambiguous cases are subjected to correspondence-based processing, which is more time-consuming.



**Figure 3.2: Selection of the Model** — Given the input image in the first column, the preselection network selects four model candidates (second column). As has been illustrated in fig. 3.1, a model graph is dynamically constructed for each model candidate by assembling matching model features into larger graphs according to their spatial arrangement (third column). The fourth column shows the reconstructions from the model graphs. 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 input image are annotated to the reconstructions. The model candidate that attains the best similarity to the input image is chosen as the recognized model (fifth column).

In this chapter we propose a form of graph dynamics, which proceeds in three steps. In the first step *position-invariant feature detectors*, which decide whether a feature is present in an image, are set up from training images. For processing arbitrary objects, features are small localized grid graphs, so-called *parquet graphs*, whose nodes are attributed with Gabor amplitudes. Through combination of these classifiers into a single layer perceptron that conforms to LINSKER's infomax principle, the so-called *preselection network*, a weighted majority voting scheme (LAM and SUEN, 1997) is implemented. The infomax 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. The preselection network allows for preselection of salient learning examples, so-called *model candidates*, and likewise for preselection of salient categories the object in the presented image supposedly belongs to. Each model candidate is verified in a third step using a rudimentary version of elastic graph matching. To further differentiate between model candidates with similar features it is asserted that the features be in similar spatial arrangement for the model to be selected. In the process model graphs are constructed dynamically by assembling model features into larger graphs according to their spatial arrangement (Fig. 3.1). Finally, the resulting model graphs are matched with a rudimentary version of elastic graph matching. The model candidate that attains the best similarity to the input image is chosen as the recognized model (Fig. 3.2).

The description of the method is accompanied by a *case study*, which exemplifies the various steps on an example, in which only two images of two objects are learned and distinguished.

### 3.1 Learning Set, Partitionings, and Categories

There are many different classifications that can be made on image data. For object recognition, all instances of the same object under different pose and/or illumination are to be put into the same class. An alternative learning problem may be the classification of illumination or pose regardless of object identity. A hallmark of human visual cognition is the classification into *categories*: we group together images of cats, dogs, insects, and reptiles into



**Figure 3.3: Case Study: Learning Set** — *The learning set comprises two images of different chewing gum packages in approximately the same pose. The images are taken from the COIL-100 database (NENE et al., 1996). In the following these images are referred to as  $I_1$  and  $I_2$ .*

the category ‘animal’ and are able to differentiate animals from non-animals with impressive speed (THORPE et al., 1996).

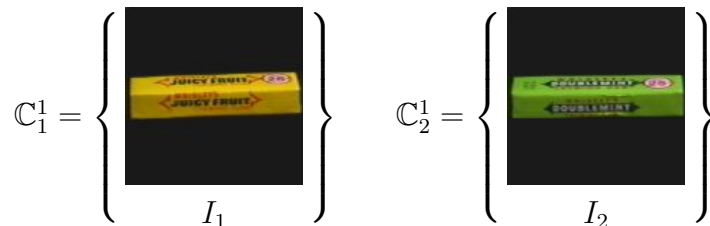
We start by considering some finite set of images  $\mathbb{I}$  and a subset  $\mathbb{D}$  of it, which we call the *learning set*. In our case study the learning set comprises two images of different chewing gum packages in approximately the same pose (Fig. 3.3).

In order to accommodate the various learning tasks that can be imposed on a single image set we consider that there exist  $K$  *partitionings*  $\Pi^k$  of the learning set (Eq. (3.1)). A partitioning  $\Pi^k$  consists of  $C^k$  pairwise disjoint partitions  $\mathbb{C}_c^k$ .

$$\Pi^k = \{ \mathbb{C}_c^k \subseteq \mathbb{D} \mid 1 \leq c \leq C^k \}$$

$$\text{with } \forall c \neq c' : \mathbb{C}_c^k \cap \mathbb{C}_{c'}^k = \emptyset \quad \text{and} \quad \bigcup_{c=1}^{C^k} \mathbb{C}_c^k = \mathbb{D} \quad (3.1)$$

The objects in the images of a particular partition are supposed to share a common semantic property, for instance, being images of animals, or having the same illumination direction. Therefore, in the following partitions are termed *categories*. *Category labels*  $c$  range between 1 and  $C^k$ ; their range implicitly depends on the number of categories in the underlying partitioning  $\Pi^k$ . For simultaneous recognition of the object’s identity and the object’s pose the learning set is subdivided into single-element categories while for object categorization purposes it might be helpful to organize the learning set in a hierarchy of categories. Fig. 3.4 shows the single partitioning of the learning set in our case study.



**Figure 3.4: Case Study: Partitioning of the Learning Set** — *In our case study there exists only  $K = 1$  partitioning  $\Pi^1$  of the learning set (Fig. 3.3). The partitioning consists of  $C^1 = 2$  single-element categories  $\mathbb{C}_1^1 = \{I_1\}$  and  $\mathbb{C}_2^1 = \{I_2\}$ .*

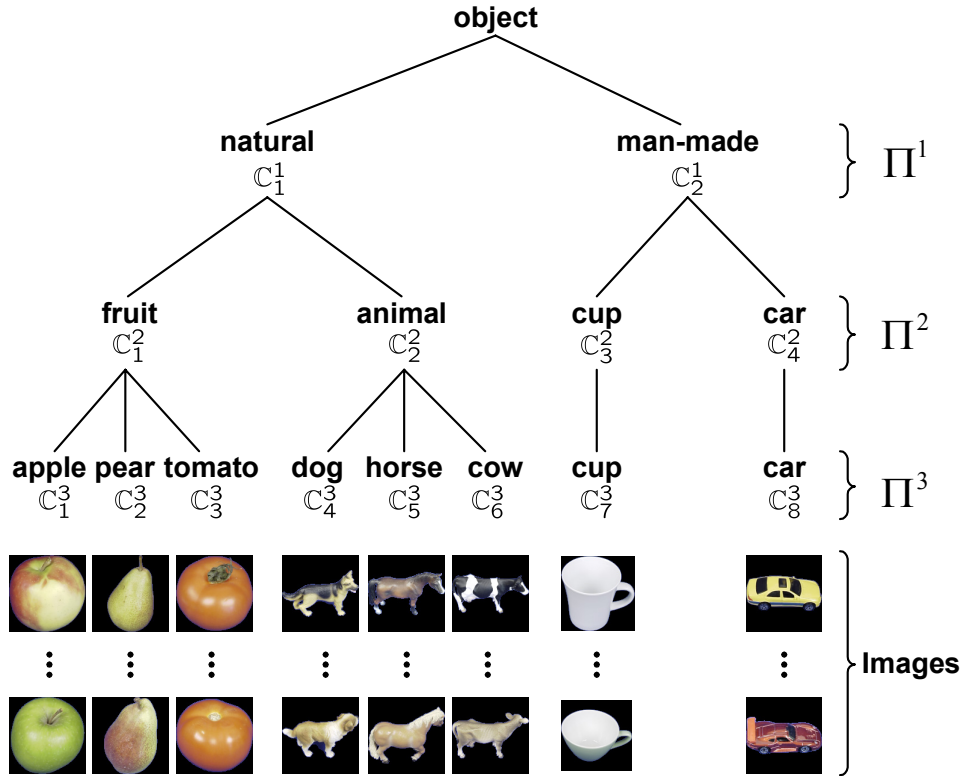
A hierarchical categorization task can be exemplified with the ETH-80 image database (LEIBE and SCHIELE, 2003). That database comprises images of apples, pears, tomatoes, dogs, horses, cows, cups, and cars in varying poses and identities and has been used for the categorization experiments in chapter 5. For those experiments we created  $K = 3$  partitionings of the learning set as shown in fig. 3.5.

## 3.2 Parquet Graphs

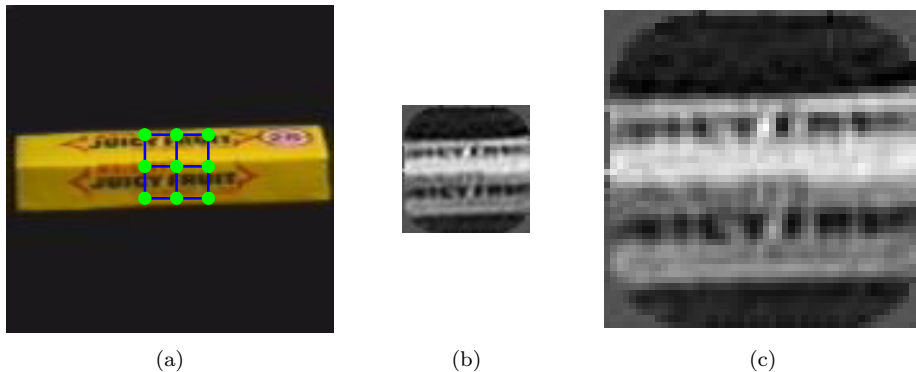
The feature-based part of the technique described in this thesis can work with any convenient feature type. A successful application employing color and multi-resolution image information is presented in (WESTPHAL and WÜRTZ, 2004). For the current combination of feature- and correspondence-based methods, we chose small regular graphs labeled with Gabor features. We call them *parquet graphs*, inspired by the look of ready-to-lay parquet tiles. These can work as simple feature detectors for preselection and can be composed to larger graph entities for correspondence-based processing.

Throughout this thesis, parquet graphs consist of  $V = 9$  nodes. In the following, a parquet graph  $f$  is described with a finite set of node attributes: 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  $\underline{x}_v$ . Computation and parameters of the Gabor features have been introduced in section 2.1.1. In order to make use of segmentation information, it is convenient to mark nodes residing in





**Figure 3.5: Hierarchical Organization of Categories** — A hierarchy of categories on the ETH-80 image database (LEIBE and SCHIELE, 2003), which contains images of apples, pears, tomatoes, dogs, horses, cows, cups, and cars in varying poses and identities, is given. We created  $K = 3$  partitionings  $\Pi^1$ ,  $\Pi^2$ , and  $\Pi^3$ . Partitioning  $\Pi^1$  comprises  $C^1 = 2$  categories of natural ( $C_1^1$ ) and man-made objects ( $C_2^1$ ). Partitioning  $\Pi^2$  comprises  $C^2 = 4$  categories of fruits ( $C_1^2$ ), animals ( $C_2^2$ ), cups ( $C_3^2$ ), and cars ( $C_4^2$ ). Finally, partitioning  $\Pi^3$  comprises  $C^3 = 8$  categories of apples ( $C_1^3$ ), pears ( $C_2^3$ ), tomatoes ( $C_3^3$ ), dogs ( $C_4^3$ ), horses ( $C_5^3$ ), cows ( $C_6^3$ ), cups ( $C_7^3$ ), and cars ( $C_8^3$ ).



**Figure 3.6: Example of a Parquet Graph** — *Figure (a) shows a parquet graph that has been placed on the object in learning image  $I_1$ . Each node of a parquet graph is attributed with Gabor amplitudes derived from an image at the node’s position. Figure (b) shows the reconstruction from the parquet graph. Figure (c) is an enlarged version of figure (b).*

the background as invalid and exclude them from further calculation in that way. For this purpose the node attributes comprise a validity flag  $b_v$  that can take the values 0 and 1, meaning ‘invalid’ and ‘valid’. Throughout this thesis, for the given parameterization of the Gabor features (Section 2.1.1), the horizontal and vertical node distances  $\Delta x$  and  $\Delta y$  are set to 10 pixels.

$$f = \left\{ (\underline{x}_v, \mathcal{J}_v, b_v) \mid 1 \leq v \leq V \right\} \quad (3.2)$$

Fig. 3.6 shows an example of a parquet graph that has been placed on the object in learning image  $I_1$ . Where appropriate, parquet graphs are more generally termed 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 node positions are irrelevant for the decision whether two images contain a similar patch of texture. Later, for verification of the selected model candidates, i.e., learning images that may serve as models for the input image, larger graphs are constructed dynamically by assembling parquet graphs derived from earlier learning images according to their spatial arrangement. Thus, within the correspondence-based part, the node positions will become important.

### 3.2.1 Similarity Function

The measure of similarity between two parquet graphs  $f$  and  $f'$  is defined as the normalized sum of the similarities between valid Gabor jets (WÜRTZ, 1997; SHAMS, 1999) attached to nodes with the same index of the given parquet graphs (Eq. (3.3)). The similarity between two Gabor jets is solely based on the amplitudes of the filter responses (Eq. (2.9)). By definition, the factors  $(b_v b'_v)$  are 1 if the respective jets  $\mathcal{J}_v$  and  $\mathcal{J}'_v$  have both been marked as valid, and 0 otherwise. These factors assert that only similarities between jets that have both been marked as valid are taken into account. If all products become 0, the similarity between the two parquet graphs yields 0. The potentials of the parquet graph similarity function are given in fig. 3.7.

$$s_{graph}(f, f') = \begin{cases} \left( \sum_{v=1}^V b_v b'_v \right)^{-1} \cdot \sum_{v=1}^V (b_v b'_v) \cdot s_{abs}(\mathcal{J}_v, \mathcal{J}'_v) & \text{if } \sum_{v=1}^V b_v b'_v > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

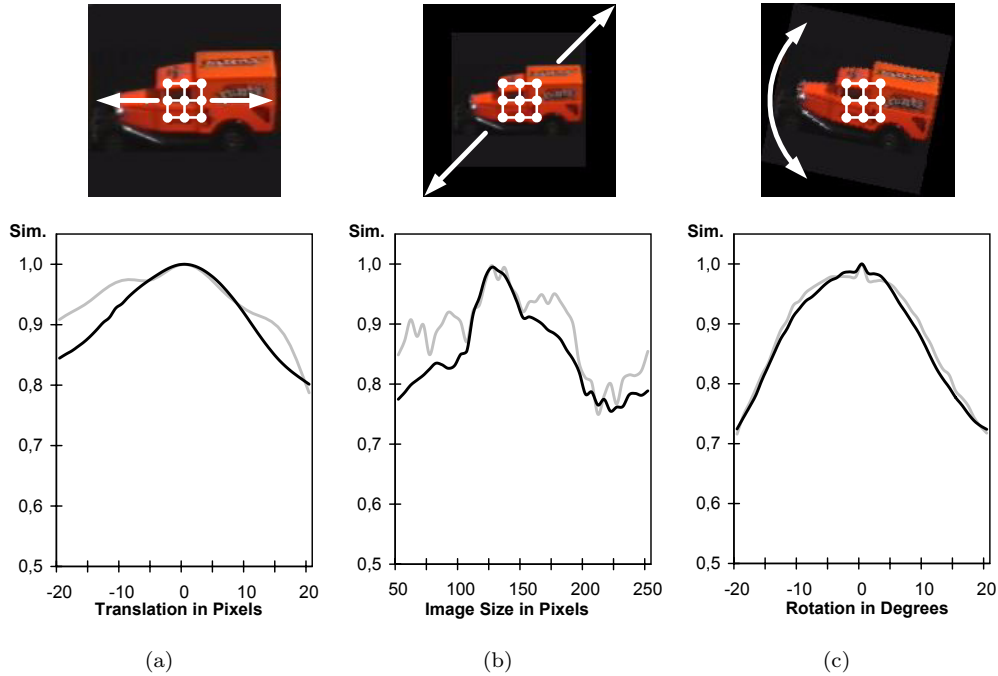
From the viewpoint of the correspondence problem, two parquet graphs in different images establish a local *array* of contiguous point-to-point correspondences. The similarity measure assesses how well points in two images specified by the given parquet graphs actually correspond to each other. It is 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.

### 3.2.2 Local Feature Detectors

For the assessment whether two parquet graphs  $f$  and  $f'$  convey similar patches of texture with respect to a given sensitivity profile we introduce local feature detectors that return 1 if the similarity between the given parquet graphs is greater or equal than a given similarity threshold  $\vartheta$  with  $0 < \vartheta \leq 1$ , and 0 otherwise (Eq. (3.4)). We say that two parquet graphs *match* with respect to a given similarity threshold if the local feature detector returns 1.

$$\varepsilon(f, f', \vartheta) = \begin{cases} 1 & \text{if } s_{graph}(f, f') \geq \vartheta \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

Matching features are one argument for point-to-point *correspondences*, which needs to be backed up by the spatial arrangement of several matching features.



**Figure 3.7: Potentials of the Parquet Graph Similarity Function**

— The potentials of the parquet graph similarity function for a parquet graph taken from the center of the same image as in fig. 2.4 are displayed. Throughout, black lines give the similarity potentials of the parquet graph similarity function while, for the sake of comparability, grey curves give the respective potentials of the jet similarity function (Eq. (2.9)); these are taken from fig. 2.4. In the translation case (a) a parquet graph has been extracted with its center node placed on the image center. It is compared to parquet graphs derived at positions along a horizontal line of increasing distance from the original position. The sensitivity to scaling (b) has been tested by comparing the original parquet graph to parquet graphs at corresponding positions in scaled images without scaling of parquet graphs. The original image is  $128 \times 128$  pixels in size. Rotation (c) has been tested by comparing the original parquet graph to parquet graphs located at the same position in rotated images. In the (a) translation (b) scale cases the measure of similarity of parquet graphs turned out to be more sensitive than the measure of similarity of Gabor jets. Nevertheless, in the case of scaled images the parquet graph similarity function performed very favorably compared to the jet similarity function: its similarity potential has a distinct global maximum and far less local maxima. The potentials differ only marginally in the (c) rotation case.

### 3.3 Learning a Visual Dictionary

Our goal is to formulate a graph dynamics that, upon image presentation, lets a model graph rapidly emerge by binding together memorized subgraphs derived from earlier learning examples. To this end we need to compute a repertoire of parquet graphs from learning examples in advance. These play the role of a *visual dictionary*. Parquet graphs derived from an input image during classification are looked up in the dictionary to find out which image and model features match. Each coincidence of a matching feature in the image and model domain may then be accounted as a piece of evidence that the input image belongs to the same categories as the learning image which contains the model feature as well.

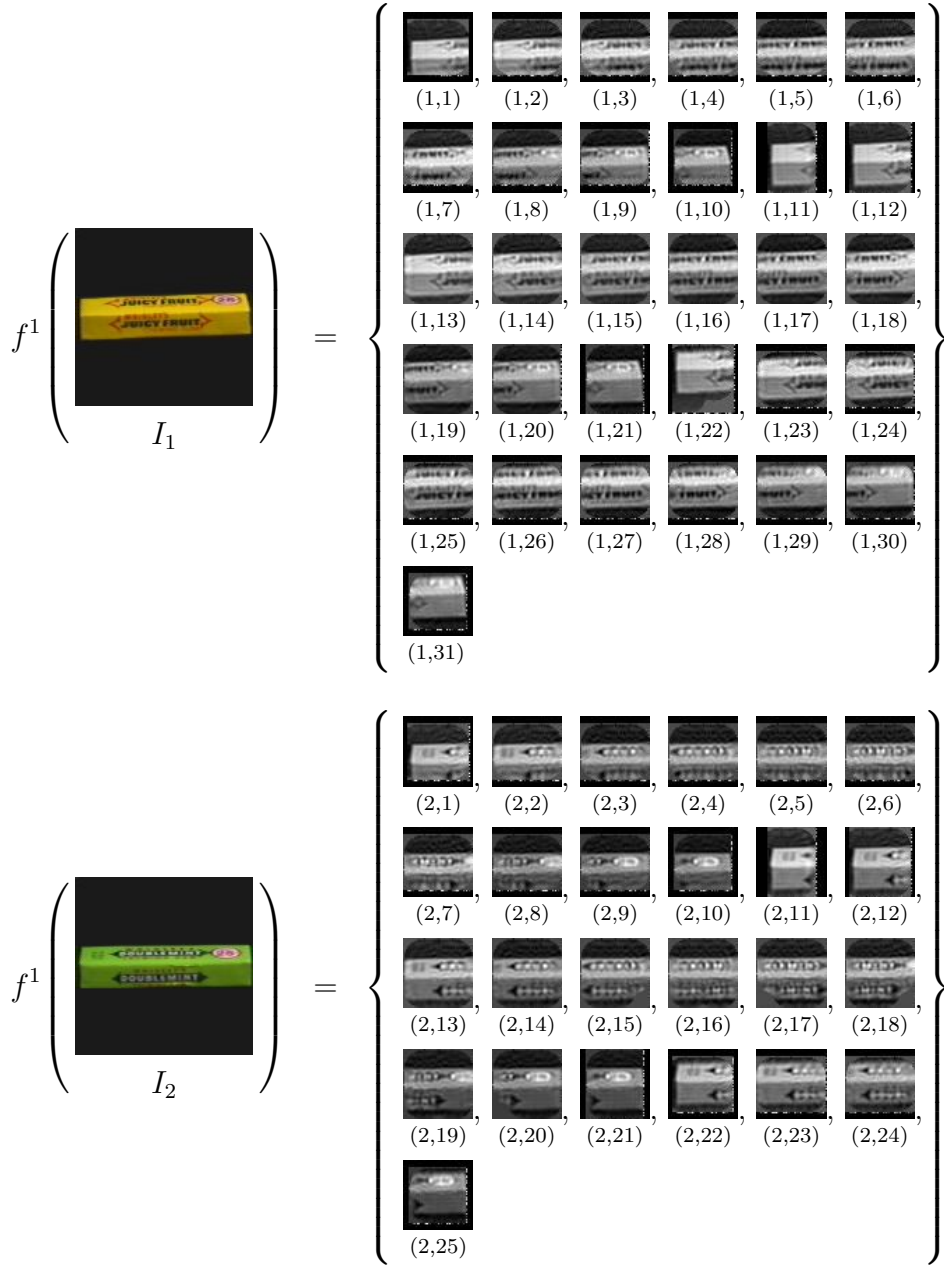
#### 3.3.1 Feature Calculators

In eq. (3.5) we define  $R$  functions  $f^r$ , each of them capable of extracting a set of features out of an image. In this thesis parquet graphs are exclusively used as local image features. Let  $\mathbb{F}$  denote the set of all possible features and let  $\wp(\mathbb{F})$  denote the power set of  $\mathbb{F}$ . In the following these functions will be called *feature calculators*. The index  $r$  implicitly specifies the parameterization of the parquet graphs returned from the respective feature calculator  $f^r$ . For instance, this applies to the similarity threshold  $\vartheta^r$ , which is employed in the local feature detectors (Eq. (3.4)). Generally, feature calculators are not restricted to parquet graph features. Other feature types have been used in (WESTPHAL and WÜRTZ, 2004; SCHMIDT and WESTPHAL, 2004; WESTPHAL, 2004; ARENTZ, 2006).

$$f^r : \mathbb{I} \rightarrow \wp(\mathbb{F}) \text{ with } r \in \{1, \dots, R\} \quad (3.5)$$

For extraction of parquet graphs, the inter-node distances  $\Delta x$  and  $\Delta y$  are also used to specify a grid in the image plane. At each grid position allowing for placement of a whole parquet graph, a parquet graph is extracted. Scanning of the image starts in the upper left corner from left to right to the lower right corner. If the image is known to be figure-ground segmented, parquet graphs with the majority of nodes residing in the background will be disregarded, the others have background points marked as invalid.

In the case study, we employ only  $R = 1$  feature calculator  $f^1$ . The feature calculator returns a set of parquet graphs with ten pixels distance between two neighbored nodes in horizontal and in vertical direction, respectively. Fig. 3.8 shows the result of applying this feature calculator to both learning examples.



**Figure 3.8: Case Study: Application of the Feature Calculator to the Learning Images** — *The thumbnail images in the returned sets on the right hand side are reconstructions from the extracted parquet graphs. Each reconstruction is uniquely labeled with a tuple. The first component addresses the learning image the parquet graph stems from while the second component is a sequential number. Application of the feature calculator to learning images  $I_1$  and  $I_2$  resulted in a different number of extracted features, even though the contained objects have approximately the same size. The difference is caused by a slight variation in pose.*

### 3.3.2 Feature Vectors

Looking at the number of parquet graphs that have been extracted from just two images (Fig. 3.8), it is clear that for learning sets with thousands or even ten thousands of images the total number of features would grow into astronomical dimensions. Consequently, we have to limit the total number of features to a tractable number. For this task we employ a simple variant of *vector quantization* (GRAY, 1984), which is given as pseudo code in fig. 3.9. A vector quantizer maps data vectors in some vector space into a finite set of *codewords*, which are supposed to represent the original set of input vectors well. A collection of codewords that purposefully represents the set of input vectors is termed *codebook*. The design of an optimal codebook is NP-hard.

Using the vector quantization given in fig. 3.9, each of the  $R$  feature calculators is used to compute a feature vector  $\underline{f}^r$  with  $r \in \{1, \dots, R\}$ . In the following  $T^r$  denotes the number of features in feature vector  $\underline{f}^r$ . All  $R$  feature vectors constitute the visual dictionary. Let, as a shorthand,  $f_t^r$  address the feature with index  $t$  in the feature vector with index  $r$ , throughout.

In our case study, application of the vector quantization algorithm using feature calculator  $f^1$  with a similarity threshold of  $\vartheta^1 = 0.92$  yields the result presented in tab. 3.1. The final feature vector  $\underline{f}^1 = (f_t^1)_{1 \leq t \leq 8}$  comprises  $T^1 = 8$  parquet graphs. The visual dictionary of our case study contains only this single feature vector.

## 3.4 Preselection Network

In this section we will present the second step of the proposed form of graph dynamics: a feedforward neural network that allows for preselection of salient learning examples, so-called *model candidates*, and likewise for preselection of *salient categories* the object in the presented image supposedly belongs to. This network will be called the *preselection network*. 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 and PAULS, 1995). The preselection network’s output neurons take the part of these view-tuned units.

The preselection network is a fully-connected single layer perceptron (ROSENBLATT, 1958) that implements a weighted majority voting scheme (LAM and SUEN, 1997). In the network’s input layer *position-invariant feature detec-*

**Algorithm 1:** *vectorQuantization***Parameter:** Learning Set:  $\mathbb{D}$ **Parameter:** Feature Calculator:  $f^r : \mathbb{I} \rightarrow \wp(\mathbb{F})$ **Parameter:** Similarity Threshold:  $\vartheta^r$ ;  $0 < \vartheta^r \leq 1$ **Result** : Feature Vector of Length  $T^r$ :  $\underline{f}^r$ 

```

1  $\mathbb{F}^r \leftarrow \emptyset$ 
2  $T^r \leftarrow 0$ 

3 forall  $I \in \mathbb{D}$  do
4   forall  $f \in f^r(I)$  do
5     if  $\forall f' \in \mathbb{F}^r : \varepsilon(f, f', \vartheta^r) = 0$  then
6        $\mathbb{F}^r \leftarrow \mathbb{F}^r \cup \{f\}$ 
7        $T^r \leftarrow T^r + 1$ 
8     end
9   end
10 end

11  $\underline{f}^r =: (f_t^r)_{1 \leq t \leq T^r} \leftarrow (0)_{1 \leq t \leq T^r}$ 
12  $t \leftarrow 0$ 
13 forall  $f \in \mathbb{F}^r$  do
14    $f_t^r \leftarrow f$ 
15    $t \leftarrow t + 1$ 
16 end

17 return  $\underline{f}^r$ 

```

**Figure 3.9: Vector Quantization Method** — *The algorithm computes a codebook of codewords. In our case parquet graphs become employed as codewords while the codebook is a set of these parquet graphs. The size of the feature set depends considerably on the value of the similarity threshold  $\vartheta^r$ . For lower values of  $\vartheta^r$  many features will be disregarded and the final feature set will become rather small. Conversely, values of  $\vartheta^r$  close to one lead to low compression rates and large feature sets. We demand random access to each particular feature in the computed codebooks. Therefore, we translate the codebook into a feature vector  $\underline{f}^r$  of length  $T^r$ , where  $T^r$  terms the number of codebook features. Let, as a shorthand,  $f_t^r$  address the feature with index  $t$  in the feature vector with index  $r$ .*



CODEWORDS		DISREGARDED FEATURES								
$f_1^1 =$		0.96	0.93	0.97	0.95	0.92	0.93	0.95	0.93	
	(1,1)									
		(1,2)	(1,3)	(1,12)	(1,13)	(1,14)	(1,22)	(2,1)	(2,12)	
$f_2^1 =$		0.97	0.97	0.96	0.94	0.95	0.93	0.93	0.94	0.94
	(1,4)									
		(1,5)	(1,6)	(1,7)	(1,8)	(1,15)	(1,16)	(2,2)	(2,3)	(2,4)
		0.93	0.93	0.93	0.92	0.92	0.93	0.93	0.92	0.92
		(2,5)	(2,6)	(2,7)	(2,8)	(2,14)	(2,15)	(2,16)	(2,17)	(2,18)
$f_3^1 =$		0.95	0.94	0.96	0.96	0.94	0.94	0.95		
	(1,9)									
		(1,10)	(1,19)	(1,20)	(2,9)	(2,10)	(2,19)	(2,20)		
$f_4^1 =$		0.94	0.94	0.97						
	(1,11)									
		(1,21)	(1,31)	(2,11)						
$f_5^1 =$		0.97	0.94	0.93	0.93	0.95	0.93			
	(1,17)									
		(1,18)	(1,24)	(1,25)	(1,26)	(1,27)	(1,28)			
$f_6^1 =$		0.93	0.92	0.93	0.94					
	(1,23)									
		(1,29)	(1,30)	(2,22)	(2,23)					
$f_7^1 =$		0.93								
	(2,13)									
		(2,24)								
$f_8^1 =$		0.96								
	(2,21)									
		(2,25)								

**Table 3.1: Case Study: Computation of the Feature Vector** — The table shows the result of applying the vector quantization algorithm given in fig. 3.9 using feature calculator  $f^1$  with a similarity threshold of  $\vartheta^1 = 0.92$ . The table’s left column comprises parquet graphs that have been chosen as codewords. The column on the right shows the disregarded parquet graphs. The lower labels have been introduced in fig. 3.8, the upper labels are the similarities between the disregarded parquet graph and the respective codeword. The final feature vector  $\underline{f}^1 = (f_t^1)_{1 \leq t \leq 8}$  comprises  $T^1 = 8$  parquet graphs.

*tors* submit their assessments whether their reference feature is present in an image to dedicated input neurons while the output layer comprises one neuron for each predefined category. 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 feed-forward 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 to directly assign synaptic weights. The time-consuming adaption of synaptic weights becomes unnecessary at the expense of having to set up the preselection network in batch mode, i.e., the complete learning set has to be presented. This network setup in conjunction with the application of the winner-take-most or winner-take-all nonlinearity as decision function (RIESENHUBER and POGGIO, 2000) implements a weighted majority voting scheme that allows for the desired preselection of salient categories and model candidates.

Here, the selection of salient categories and model candidates is only based on feature coincidences in image and model domain. As their spatial arrangement is disregarded, false positives are frequent among the selected model candidates. To rule them out similar spatial arrangement of features will be asserted for the model to be selected in the correspondence-based verification part (Section 3.5).

### 3.4.1 Neural Model

In the preselection network we employ two types of generalized McCulloch & Pitts neurons (MCCULLOCH and PITTS, 1943), variant A with identity and variant B with a Heaviside threshold function  $H(\cdot)$  as output function. The output of a neuron of type A is equal to the weighted sum of its inputs  $\sum_{n=1}^N x_n w_n$  with  $x_n$  being the presynaptic neurons’ outputs and the  $w_n$  being synaptic weights. The output of a neuron of type B is 1 if the weighted sum of its inputs is greater than 0, and 0 otherwise.

### 3.4.2 Position-Invariant Feature Detectors

To test the presence of a particular feature from the visual dictionary, in the following called *reference* or *model feature*, in an image we construct a *position-invariant feature detector* out of local feature detectors (Section

3.2.2). For this task, we distribute instances of local feature detectors uniformly over the image plane. For a given reference feature, combining these local feature detectors in a linear discriminant yields a position-invariant feature detector that returns 1 if the reference feature is observed at at least one position in the image plane, and 0 otherwise. Position-invariance is thus achieved with a logical OR. In the same fashion invariance to scale may be implemented. Fig. 3.10 shows how a position-invariant feature detector is constructed for a feature taken from the visual dictionary. For a given feature  $f_t^r$ , the symbol  $\tau_t^r$  denotes the respective position-invariant feature detector and  $\tau_t^r(I)$  (Eq. (3.6)) its result. We will say that a position-invariant feature detector  $\tau_t^r$  has *found* or *observed* its feature  $f_t^r$  in an input image  $I$  if  $\tau_t^r(I) = 1$ . From now on, we use the term *feature detector* only for the position-invariant version.

$$\tau_t^r : \mathbb{I} \rightarrow \{0, 1\}; \tau_t^r(I) = H \left( \sum_{f \in f^r(I)} \varepsilon(f, f_t^r, \vartheta^r) \right) \quad (3.6)$$

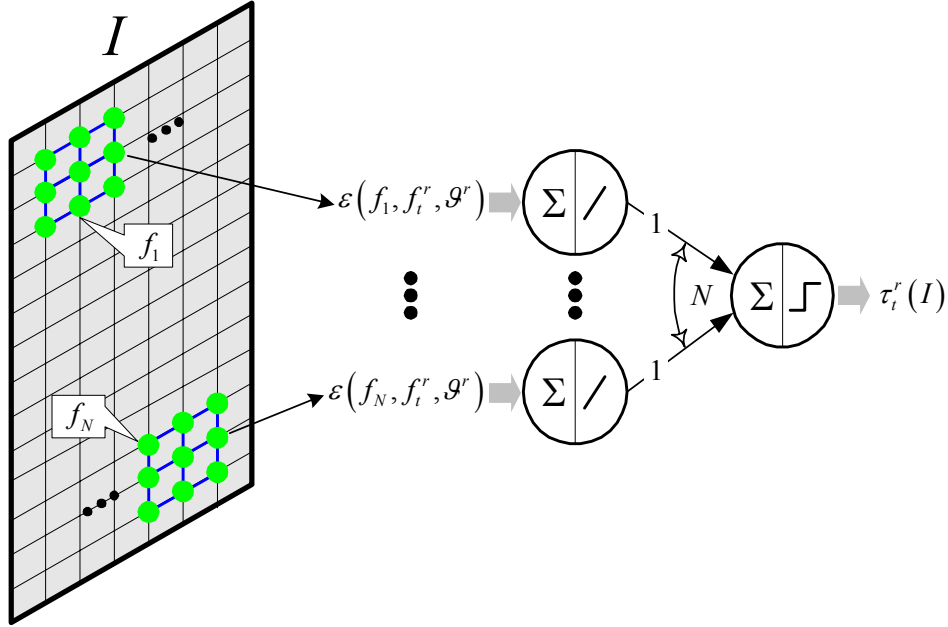
For the sake of simplicity we regard the feature detectors as the perceptron's processing elements (ROSENBLATT, 1958), rather than an additional layer.

Each time a feature detector has found its reference feature  $f_t^r$  in the input image, we add a pair of matching features  $(f, f_t^r)$  to a table, where  $f$  stems from the input image. That table is used for efficient construction of image and model graphs in the correspondence-based verification part (Section 3.5). The table is cleared before each image presentation.

$$\mathcal{F}_{match}(I) \leftarrow \mathcal{F}_{match}(I) \cup \bigcup_{f \in f^r(I)} \left\{ (f, f_t^r) \mid \varepsilon(f, f_t^r, \vartheta^r) = 1 \right\} \quad (3.7)$$

### 3.4.3 Weighting of Feature Detectors

From the example in tab. 3.1 it becomes clear that the feature detectors have varying relevance for the selection of salient categories. In the following the contributions of the feature detectors to the decision about choosing salient categories are described through *measures of information*. SHANNON has defined information as the decrease of uncertainty (SHANNON, 1948). In this sense, a natural definition of the measures of information is presented in eq. (3.8). For a given feature detector  $\tau_t^r$  that has found its reference feature  $f_t^r$  in the input image and for a given partitioning  $\Pi^k$ , the information  $i_t^{r,k}$  that feature detector contributes to the decision about choosing



**Figure 3.10: Position-Invariant Feature Detector** — *The position-invariant feature detector returns 1 if a given feature  $f_t^r$  is present in image  $I$ , and 0 otherwise. Position-invariance is thus achieved with a logical OR. At each grid position allowing for placement of a whole parquet graph a local feature detector is installed that compares the local parquet graph with the reference feature  $f_t^r$ . Technically, this has been implemented by applying feature calculator  $f^r$  to the given image  $I$ . If the feature calculator returns a set of  $N$  parquet graphs  $\{f_n | 1 \leq n \leq N\}$ , each local feature detector compares its feature  $f_n$  with the reference feature  $f_t^r$  with respect to similarity threshold  $\vartheta^r$ . Then, each local feature detector passes its result into a single layer perceptron with  $N$  input units of type A, one output unit of type B, and feedforward connections of strength 1 between each input unit and the output neuron. The net's output is 1 if at least one of the local feature detectors has found its reference feature in the given image, and 0 otherwise. In this fashion a position-invariant feature detector is instantiated for each feature in the visual dictionary. In the same manner invariance to scale may be achieved.*

categories of partitioning  $\Pi^k$  is defined by the difference between the largest possible amount of uncertainty,  $\ln C^k$ , and the feature detector's amount of uncertainty encoded by the SHANNON entropy  $\mathcal{H}_t^{r,k}$ .  $\mathcal{P} [\mathbb{C}_c^k | f_t^r]$  describes the conditional probability that the genuine category is  $\mathbb{C}_c^k$  given that feature  $f_t^r$  has been observed. Probabilities equal zero are excluded from calculation. In this fashion measures of information are calculated for all features in the visual dictionary with respect to all partitionings of the learning set. Similar approaches are proposed in (ULLMAN and SALI, 2000; FRITZ et al., 2004).

$$i_t^{r,k} = \ln C^k - \mathcal{H}_t^{r,k} = \ln C^k + \sum_{\substack{c=1 \\ \mathcal{P}[\mathbb{C}_c^k | f_t^r] \neq 0}}^{C^k} \left( \mathcal{P} [\mathbb{C}_c^k | f_t^r] \cdot \ln \mathcal{P} [\mathbb{C}_c^k | f_t^r] \right) \quad (3.8)$$









For a given partitioning  $\Pi^k$ , the measures of information range between 0 and  $\ln C^k$ . If a feature occurs in all categories of that partitioning, the respective feature detector cannot make a contribution and, accordingly, its measure of information is 0. Conversely, if a feature occurs in only one category, the respective feature detector contributes maximally; its measure of information is  $\ln C^k$ .

For the derivation of the conditional probabilities  $\mathcal{P} [\mathbb{C}_c^k | f_t^r]$  we start with the definition of a shorthand: let  $n_t^r(\mathbb{C})$  denote the total number of observations of feature  $f_t^r$  in the images of the parameterized category  $\mathbb{C}$  (Eq. (3.9)).

$$n_t^r(\mathbb{C}) = \sum_{I \in \mathbb{C}} \sum_{f \in f^r(I)} \varepsilon(f, f_t^r, \vartheta^r) \quad (3.9)$$

Assuming that all prior probabilities for choosing a category of a given partitioning  $\Pi^k$  are the same, the conditional probabilities  $\mathcal{P} [\mathbb{C}_c^k | f_t^r]$  are calculated through application of Bayes' rule as given in eq. (3.10)). For a given category  $\mathbb{C}_c^k$  and a given feature  $f_t^r$  we may interpret this probability as the frequency of that feature across the categories of partitioning  $\Pi^k$ . Tab. 3.2 demonstrates the calculation of the measures of information in our case study.

$$\mathcal{P} [\mathbb{C}_c^k | f_t^r] = \frac{n_t^r(\mathbb{C}_c^k)}{\sum_{c'=1}^{C^k} n_t^r(\mathbb{C}_{c'}^k)} \quad (3.10)$$

FEATURE INDEX ( $t$ )	FEATURE ( $f_t^1$ )	$n_t^1(\mathbb{C}_1^1)$	$n_t^1(\mathbb{C}_2^1)$	$\mathcal{P}[\mathbb{C}_1^1 f_t^1]$	$\mathcal{P}[\mathbb{C}_2^1 f_t^1]$	$i_t^{1,1}$
1	 (1,1)	7	2	$\frac{7}{9}$	$\frac{2}{9}$	0.1634
2	 (1,4)	7	12	$\frac{7}{19}$	$\frac{12}{19}$	0.035
3	 (1,9)	4	4	$\frac{1}{2}$	$\frac{1}{2}$	0
4	 (1,11)	3	1	$\frac{3}{4}$	$\frac{1}{4}$	0.1307
5	 (1,17)	7	0	1	0	0.6931
6	 (1,23)	3	2	$\frac{3}{5}$	$\frac{2}{5}$	0.0201
7	 (2,13)	0	2	0	1	0.6931
8	 (2,21)	0	2	0	1	0.6931

**Table 3.2: Case Study: Calculation of Measures of Information** — The table demonstrates the calculation of the feature detectors’ measures of information. The parquet graphs in the second column stem from the visual dictionary and can be looked up in tab. 3.1. The number of feature occurrences  $n_t^1(\mathbb{C}_1^1)$  and  $n_t^1(\mathbb{C}_2^1)$  in columns three and four can be verified by counting the occurrences of the respective reference feature  $f_t^1$  within categories  $\mathbb{C}_1^1$  and  $\mathbb{C}_2^1$  (Tab. 3.1). The probabilities  $\mathcal{P}[\mathbb{C}_1^1|f_t^1]$  and  $\mathcal{P}[\mathbb{C}_2^1|f_t^1]$  in columns five and six have been calculated using eq. (3.10), and, finally, the measures of information  $i_t^{1,1}$  in column seven have been calculated using eq. (3.8). One can easily verify that the measures of information scale proportionally with the feature detectors’ contributions about choosing salient categories.

### 3.4.4 Neurons, Connectivity, and Synaptic Weights

The preselection network is a single-layer perceptron comprising a layer of input and a layer of output neurons. In the network’s input layer, we assign neurons of type A to the feature detectors. Thus, the network comprises  $V_{\text{in}} = \sum_{r=1}^R T^r$  input neurons. By definition, each input neuron passes the result of its feature detector into the network. In the network’s output layer, we assign neurons of type A to the predefined categories. Accordingly, the network contains  $V_{\text{out}} = \sum_{k=1}^K C^k$  output neurons.

For fulfillment of the infomax principle, we define the synaptic weight  $w_{t,c}^{r,k}$  between the presynaptic neuron assigned to a feature detector  $\tau_t^r$  and the postsynaptic neuron assigned to a category  $\mathbb{C}_c^k$  as follows. Imagine that feature  $f_t^r$  can both be observed in the input image and in at least one image of that category. Then, this may be considered as a piece of evidence that the object in the input image belongs to that category. Consequently, feature detector  $\tau_t^r$  should contribute its quantitative amount of information  $i_t^{r,k}$  to the output of the postsynaptic neuron assigned to category  $\mathbb{C}_c^k$ . Conversely, if that category contains only images in which feature  $f_t^r$  cannot be observed, the feature detector should never be allowed to make a contribution at all.

Using this construction rule, we define  $R \cdot K$  matrices of synaptic weights  $\underline{\underline{W}}^{r,k}$ : one matrix per feature vector/partitioning combination. For a given feature vector  $\underline{f}^r$  and a given partitioning  $\Pi^k$ , weight matrix  $\underline{\underline{W}}^{r,k}$  (Eq. (3.11)) is of dimensions  $(C^k \times T^r)$ . It comprises the synaptic weights  $w_{t,c}^{r,k}$  of the connections between the input neurons assigned to feature detectors  $\tau_t^r$  and the output neurons assigned to categories  $\mathbb{C}_c^k$ . The indices  $t$  of presynaptic neurons range between 1 and  $T^r$  and indices  $c$  of the postsynaptic neurons between 1 and  $C^k$ .

$$\underline{\underline{W}}^{r,k} = \left( H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_{t'}^r(I') \right) \cdot i_t^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} =: \left( w_{t,c}^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} \quad (3.11)$$

In our case study, feature vector  $\underline{f}^1$  comprises eight features and the learning set has been partitioned into two categories. Accordingly, weight matrix  $\underline{\underline{W}}^{1,1}$  is of dimensions  $(2 \times 8)$ . The matrix is shown in fig. 3.11.

### 3.4.5 Saliencies

The output of the postsynaptic neuron dedicated to category  $\mathbb{C}_c^k$  will be called the *saliency* of that category and is denoted by  $s_c^k(I)$ . With respect to an

$$\begin{aligned}
\underline{\underline{W}}^{1,1} &= \left( H \left( \sum_{I \in \mathbb{C}_c^1} \tau_t^1(I) \right) \cdot i_t^{1,1} \right)_{\substack{1 \leq c \leq 2 \\ 1 \leq t \leq 8}} \\
&= \begin{pmatrix} 0.1634 & 0.035 & 0 & 0.1307 & 0.6931 & 0.0201 & 0 & 0 \\ 0.1634 & 0.035 & 0 & 0.1307 & 0 & 0.0201 & 0.6931 & 0.6931 \end{pmatrix} \\
&=: (w_{t,c}^{1,1})_{\substack{1 \leq c \leq 2 \\ 1 \leq t \leq 8}}
\end{aligned}$$

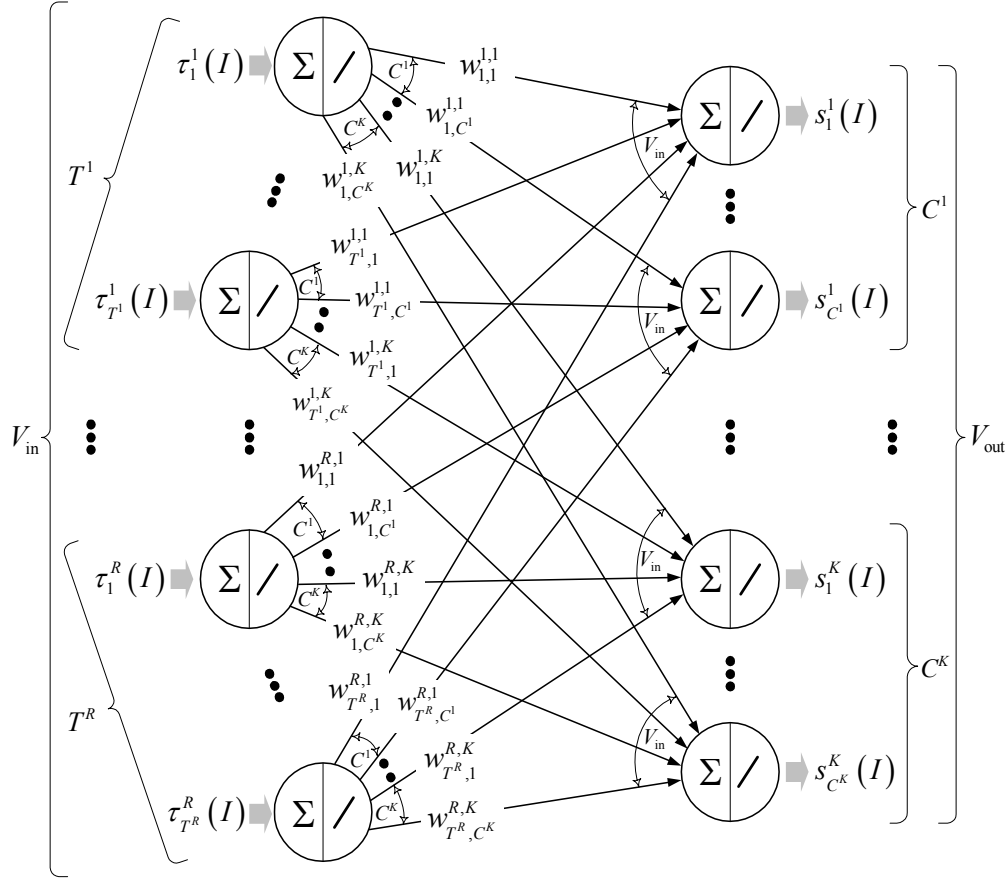
**Figure 3.11: Case Study: Weight Matrix** — *In our case study, feature vector  $f^1$  comprises eight features and the learning set has been partitioned into two categories. Accordingly, weight matrix  $\underline{\underline{W}}^{1,1}$  is of dimensions  $(2 \times 8)$ . The measures of information  $i_t^{1,1}$  of feature detectors  $\tau_t^1$  can be looked up in tab. 3.2.*

input image  $I$ , that saliency is defined as the sum of the measures of information  $i_t^{r,k}$  of those feature detectors  $\tau_t^r$  whose reference feature coincides in the input image and in at least one image of category  $\mathbb{C}_c^k$ . Thus, a saliency is the accumulated evidence contributed by activated feature detectors: the more pieces of evidence have been collected, the more likely the input image belongs to that category. For each partitioning of the learning set we can calculate a *saliency vector*  $\underline{s}^k$  with  $C^k$  saliencies by summing up the matrix vector products of the weight matrices  $\underline{\underline{W}}^{r,k}$  with the vector of feature detector responses  $(\tau_t^r(I))_{1 \leq t \leq T^r}$  over all  $R$  feature vectors (Eq. (3.12)). Fig. 3.12 shows the complete preselection network.

$$\underline{s}^k : \mathbb{I} \rightarrow \mathbb{R}^{C^k}; \quad \underline{s}^k(I) = \sum_{r=1}^R \underline{\underline{W}}^{r,k} \cdot (\tau_t^r(I))_{1 \leq t \leq T^r} =: (s_c^k(I))_{1 \leq c \leq C^k} \quad (3.12)$$

We exemplify the calculation of saliencies with our case study. First, the only feature calculator  $f^1$  is applied to input image  $I'_2$  which contains the same object as learning image  $I_2$  but in a slightly different pose. The result of feature extraction is shown in fig. 3.13. Second, the visual dictionary is traversed in search of matching model features. The outcome of this step, the vector of feature detector responses, is given in tab. 3.3. Passing this vector to the preselection network of our case study yields saliencies of  $s_1^1(I'_2) = 0.6931$  for category  $\mathbb{C}_1^1$  and  $s_2^1(I'_2) = 1.3862$  for category  $\mathbb{C}_2^1$ . Thus, the input image belongs more likely to category  $\mathbb{C}_2^1$ .





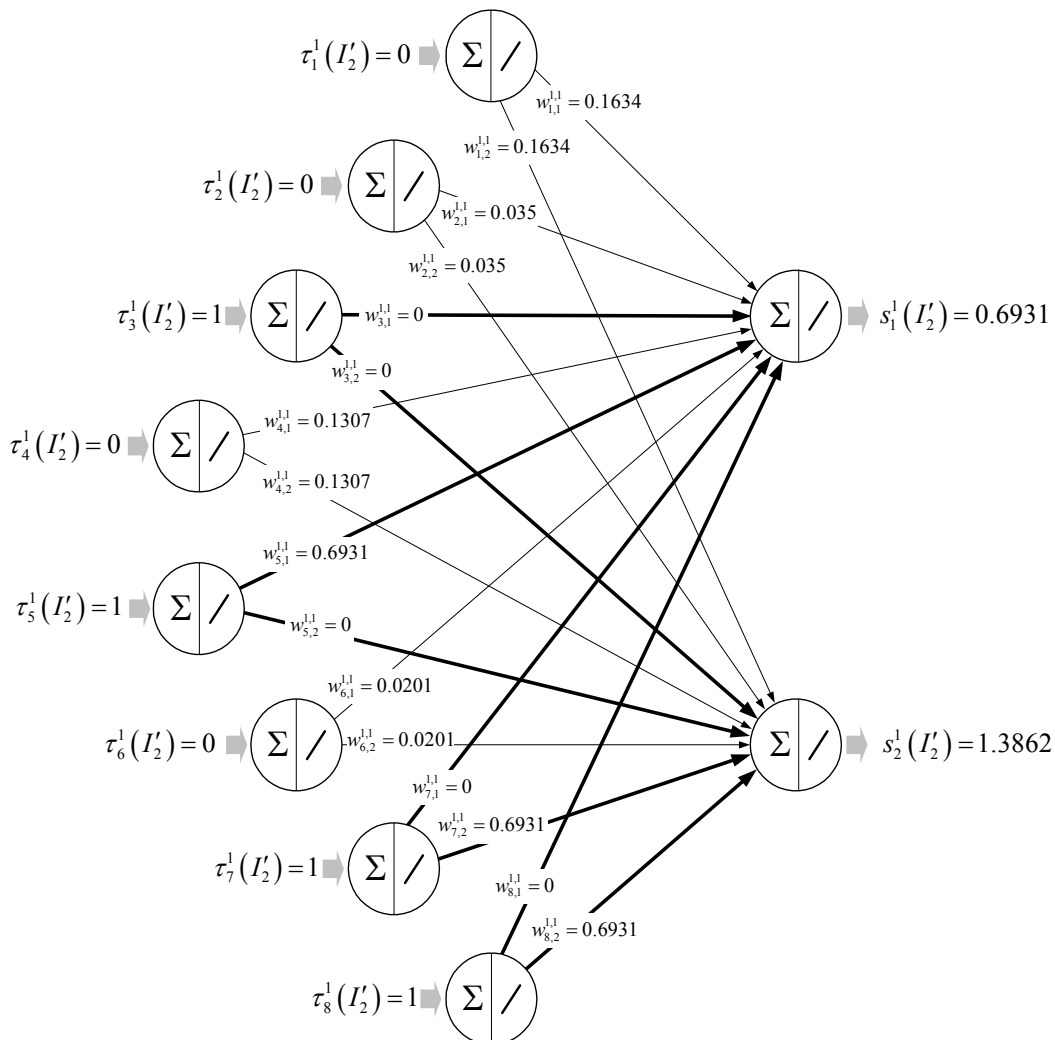
**Figure 3.12: Preselection Network** — The preselection network is a fully-connected single-layer perceptron. In its input layer neurons of type A have been assigned to the feature detectors. Accordingly, the network comprises  $V_{\text{in}} = \sum_{r=1}^R T^r$  input neurons. Each input neuron passes the binary result of its feature detector into the network. In the network's output layer neurons of type A have been assigned to the predefined categories. Accordingly, the network contains  $V_{\text{out}} = \sum_{k=1}^K C^k$  output neurons. The synaptic weights  $w_{i,c}^{r,k}$  are chosen in a way such that the whole network conforms to LINSKER's infomax principle. The output of the postsynaptic neuron that has been assigned to a given category  $C_c^k$  will be called the saliency of that category and is denoted by  $s_c^k(I)$ .

$$f^1 \left( \begin{array}{c} \text{Image of a green box} \\ I'_2 \end{array} \right) = \left\{ \begin{array}{l} \begin{array}{cccccc} \text{Thumbnail 1} & \text{Thumbnail 2} & \text{Thumbnail 3} & \text{Thumbnail 4} & \text{Thumbnail 5} & \text{Thumbnail 6} \\ (2, 1)' & (2, 2)' & (2, 3)' & (2, 4)' & (2, 5)' & (2, 6)' \end{array} \\ \begin{array}{cccccc} \text{Thumbnail 7} & \text{Thumbnail 8} & \text{Thumbnail 9} & \text{Thumbnail 10} & \text{Thumbnail 11} & \text{Thumbnail 12} \\ (2, 7)' & (2, 8)' & (2, 9)' & (2, 10)' & (2, 11)' & (2, 12)' \end{array} \\ \begin{array}{cccccc} \text{Thumbnail 13} & \text{Thumbnail 14} & \text{Thumbnail 15} & \text{Thumbnail 16} & \text{Thumbnail 17} & \text{Thumbnail 18} \\ (2, 13)' & (2, 14)' & (2, 15)' & (2, 16)' & (2, 17)' & (2, 18)' \end{array} \\ \begin{array}{cccccc} \text{Thumbnail 19} & \text{Thumbnail 20} & \text{Thumbnail 21} & \text{Thumbnail 22} & \text{Thumbnail 23} & \text{Thumbnail 24} \\ (2, 19)' & (2, 20)' & (2, 21)' & (2, 22)' & (2, 23)' & (2, 24)' \end{array} \\ \begin{array}{ccc} \text{Thumbnail 25} & \text{Thumbnail 26} & \text{Thumbnail 27} \\ (2, 25)' & (2, 26)' & (2, 27)' \end{array} \end{array} \right.$$

**Figure 3.13: Case Study: Application of Feature Calculator  $f^1$  to Image  $I'_2$**  — The thumbnail images in the returned set on the right hand side are reconstructions from the extracted parquet graphs and serve for visualization purposes. Each reconstruction is uniquely labeled with a tuple. The first component addresses the testing image while the second component is a sequential number. The dash after each label indicates that the parquet graph stems from the input image.

FEATURE INDEX ( $t$ )	MODEL FEATURE ( $f_t^1$ )	MATCHING IMAGE FEATURES	FEATURE DETECTOR ( $\tau_t^1(I_2')$ )
1	(1,1)	_____	0
2	(1,4)	_____	0
3	(1,9)	0.92 0.92 0.93 0.92 0.94 0.94	1
4	(1,11)	_____	0
5	(1,17)	0.92 0.92 0.92	1
6	(1,23)	_____	0
7	(2,13)	0.95 0.97	1
8	(2,21)	0.93 0.95 0.97 0.93 0.94 0.94	1

**Table 3.3: Case Study: Matching Features** — The table shows matching model (second column) and image parquet graphs (third column) with respect to similarity threshold  $\vartheta^1 = 0.92$ . The upper labels of the reconstructions in the third column are the similarities between matching model and image parquet graphs. For clarity, non-matching parquet graphs have been disregarded. The feature detectors  $\tau_t^1$  return 1 if they have observed their reference feature  $f_t^1$  in the input image, and 0 otherwise (column four).



**Figure 3.14: Case Study: Computation of Saliencies** — *The preselection network of our case study comprises eight input and two output neurons. Synaptic weights are taken from weight matrix  $\underline{\underline{W}}^{1,1}$  (Fig. 3.11). Connections with an activated presynaptic neuron, i.e., input neurons whose feature detector has found its reference feature in the input image, are drawn as bold arrows. Taking  $I'_2$  as an input image yields  $(\tau_t^1(I'_2))_{1 \leq t \leq 8} = (0, 0, 1, 0, 1, 0, 1, 1)^\top$  as the vector of feature detector responses (Tab. 3.3). This vector serves as an input to the preselection network and yields saliencies of  $s_1^1(I'_2) = 0.6931$  for category  $\mathbb{C}_1^1$  and  $s_2^1(I'_2) = 1.3862$  for category  $\mathbb{C}_2^1$ . Thus, the input image belongs more likely to category  $\mathbb{C}_2^1$ .*

### 3.4.6 Synaptic Plasticity

Subject to constraints, the infomax principle allows to directly assign synaptic weights. However, in classical neural architectures learning is modeled by synaptic plasticity: the change of synaptic weights under the control of neural signals. In the following we briefly describe that the preselection network’s synaptic weights can be learned in a Hebbian fashion (HEBB, 1949). Hebbian synaptic plasticity states that connections between two simultaneously firing neurons should be strengthened.

From the viewpoint of information theory PFAFFELHUBER (1972) describes learning as a process in which the system’s own uncertainty, its *subjective* entropy, or, equivalently, its missing information decreases in time. In information theory the SHANNON entropy is used as a measure of uncertainty. It is based on so-called *objective* probabilities which are only known by an ideal observer with full world knowledge but are unknown to the learning biological system itself. That measure is thus unsuited to characterize the system’s own uncertainty.

For the derivation of a Hebbian-style weight dynamics we focus on a given synaptic weight  $w_{t,c}^{r,k}$  and follow PFAFFELHUBER’s line of thought. The contribution of feature detector  $\tau_t^r$  to the decision about choosing a category of partitioning  $\Pi^k$  is described through a measure of information  $i_t^{r,k}$  which is based on the SHANNON entropy  $\mathcal{H}_t^{r,k}$  (Eq. (3.8)). That entropy in turn expresses the feature detector’s uncertainty about choosing categories of  $\Pi^k$ . Since it is exclusively based on *objective* probabilities  $\mathcal{P} [\mathbb{C}_c^k | f_t^r]$  (Eq. (3.10)) it is termed the *objective* entropy.

In the same fashion we define the synaptic weight which is based on the system’s own or *subjective* entropy  $\tilde{\mathcal{H}}_t^{r,k}$  about choosing a category of partitioning  $\Pi^k$ . It is based on, yet unknown, *subjective* probabilities  $\mathcal{Q} [\mathbb{C}_c^k | f_t^r]$  that the true category is  $\mathbb{C}_c^k$  given that feature  $f_t^r$  has been observed. According to PFAFFELHUBER, the subjective entropy is defined as the expectation value of subjective information contents (BELIS and GUIASU, 1968), taken with the objective probabilities (Eq. (3.13)). Subjective probabilities equal zero are excluded from calculation.

$$\tilde{\mathcal{H}}_t^{r,k} = - \sum_{\substack{c'=1 \\ \mathcal{Q}[\mathbb{C}_{c'}^k | f_t^r] \neq 0}}^{\mathbb{C}^k} \left( \mathcal{P} [\mathbb{C}_{c'}^k | f_t^r] \cdot \ln \mathcal{Q} [\mathbb{C}_{c'}^k | f_t^r] \right) \quad (3.13)$$

The connection strength  $\tilde{w}_{t,c}^{r,k}$  that is based on the system's subjective entropy is defined in eq. (3.14).

$$\tilde{w}_{t,c}^{r,k} = H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \cdot \left( \ln C^k - \tilde{\mathcal{H}}_t^{r,k} \right) \quad (3.14)$$

From the viewpoint of information theory PFAFFELHUBER defines learning as the process in which the system's subjective entropies converge towards the objective ones in time. This is equivalent to the convergence of the missing information (Eq. (3.15)) about choosing a category of partitioning  $\Pi^k$  given that feature  $f_t^r$  has been observed towards zero in time. The missing information is defined as the difference between subjective and objective entropy. Objective and subjective probabilities equal zero are excluded from calculation.

$$\tilde{\mathcal{H}}_t^{r,k} - \mathcal{H}_t^{r,k} = \sum_{\substack{\mathcal{P}[\mathbb{C}_{c'}^k | f_t^r] \cdot \mathcal{Q}[\mathbb{C}_{c'}^k | f_t^r] \neq 0 \\ c'=1 \\ C^k}} \left( \mathcal{P}[\mathbb{C}_{c'}^k | f_t^r] \cdot \ln \frac{\mathcal{P}[\mathbb{C}_{c'}^k | f_t^r]}{\mathcal{Q}[\mathbb{C}_{c'}^k | f_t^r]} \right) \quad (3.15)$$

Through analysis of eq. (3.15) we learn that the missing information becomes zero if all objective and subjective probabilities coincide:  $\forall c' \in \{1, \dots, C^k\} : \mathcal{P}[\mathbb{C}_{c'}^k | f_t^r] = \mathcal{Q}[\mathbb{C}_{c'}^k | f_t^r]$ . Hence, in eq. (3.16) we introduce a fairly simple dynamics that lets a given subjective probability converge towards the respective objective one with a learning velocity  $\alpha \in ]0, 1]$ . Note that the subjective probability is adapted only if both the pre- and the postsynaptic neuron are active, i.e., if feature  $f_t^r$  can be observed both in the input image  $I$  and in at least one image of category  $\mathbb{C}_c^k$ . Further note that the dynamics in eq. (3.16) cannot describe a real learning process since it is still a function of the objective probability, which is, however, unknown to the system. Thus, the learning system does not only have to learn its own subjective probabilities but in addition has to compute estimates of the objective ones. In eqs. (3.18) and (3.19) we give a method to compute these estimates.

$$\dot{\mathcal{Q}}[\mathbb{C}_c^k | f_t^r] = \alpha \cdot \tau_t^r(I) \cdot H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \cdot \left( \mathcal{P}[\mathbb{C}_c^k | f_t^r] - \mathcal{Q}[\mathbb{C}_c^k | f_t^r] \right) \quad (3.16)$$

Since the dynamics in eq. (3.16) lets the subjective probabilities converge to the objective ones in time, the missing information converges to 0 and,

consequently, the synaptic weight  $\tilde{w}_{t,c}^{r,k}$  converges to  $w_{t,c}^{r,k}$ . From the viewpoint of information theory, choosing the synaptic weights as in eq. (3.11) is thus the best choice. The final dynamics of synaptic weights is given in eq. (3.17). Note that it conforms to Hebbian synaptic plasticity.

$$\dot{\tilde{w}}_{t,c}^{r,k} = \tau_t^r(I) \cdot H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \cdot \left( \left( \ln C^k - \tilde{\mathcal{H}}_t^{r,k} \right) - \tilde{w}_{t,c}^{r,k} \right) \quad (3.17)$$

Estimates of the objective probabilities  $\tilde{\mathcal{P}}[\mathbb{C}_c^k | f_t^r]$  can, for instance, be iteratively computed as the frequencies of feature occurrences in the categories of the learning set partitionings. They are updated each time a new input image is presented (Eq. (3.18)). Let  $m_{t,c}^{r,k}$  denote the accumulated number of coincidences of feature  $f_t^r$  in the presented images and in category  $\mathbb{C}_c^k$  and let  $n_t^r(\{I\})$  denote the number of occurrences of feature  $f_t^r$  in the current image (Eq. (3.9)). For evaluation purposes noise  $\rho$  may optionally be added. Note that the input images are not necessarily taken from a predefined learning set. We rather conceive that learning biological or technical systems receive a continual stream of images while visually exploring objects.

$$\tilde{\mathcal{P}}[\mathbb{C}_c^k | f_t^r] = \frac{m_{t,c}^{r,k}}{\sum_{c'=1}^{C^k} m_{t,c'}^{r,k}} \quad (3.18)$$

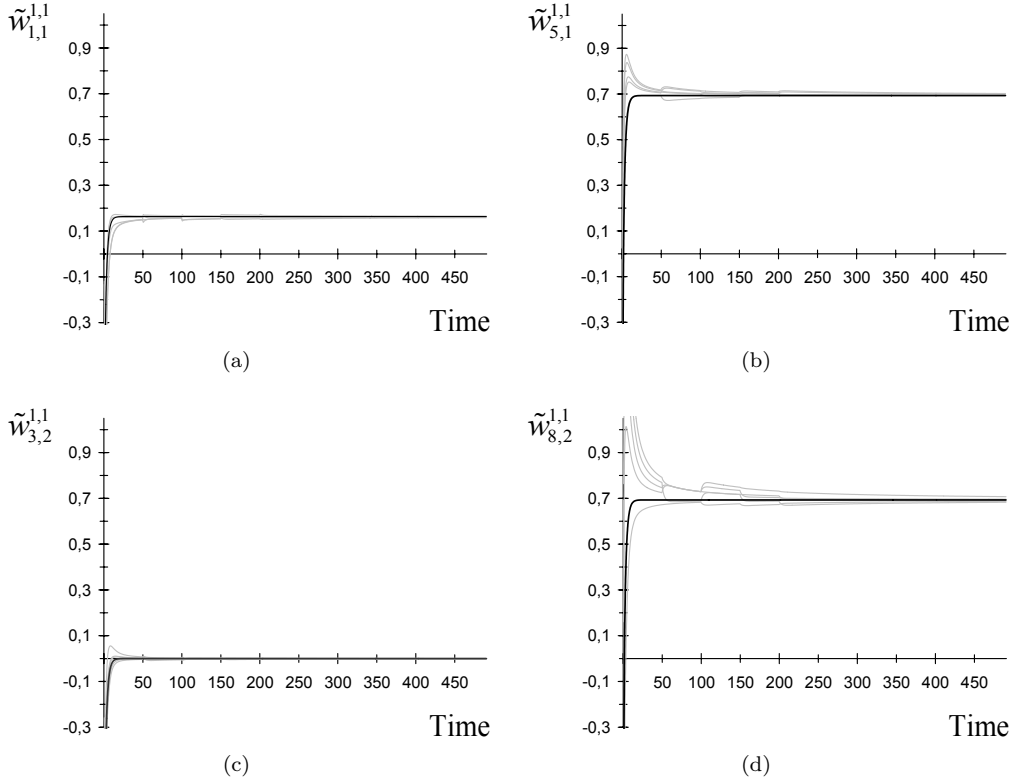
$$\dot{m}_{t,c}^{r,k} = \tau_t^r(I) \cdot H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \cdot n_t^r(\{I\}) + \rho \quad (3.19)$$

If we replace the objective probabilities,  $\mathcal{P}[\mathbb{C}_c^k | f_t^r]$ , in eq. (3.16) by their estimates,  $\tilde{\mathcal{P}}[\mathbb{C}_c^k | f_t^r]$ , we finally yield a learning system that is able to reduce its missing information solely based upon its own beliefs.

Fig. 3.15 shows the development of a selection of synaptic weights in the preselection network of our case study.

### 3.4.7 Selection of Salient Categories and Model Candidates

For selection of salient categories for the input image  $I$  we apply a winner-take-most nonlinearity as a decision rule (RIESENHUBER and POGGIO, 2000).



**Figure 3.15: Case Study: Synaptic Plasticity** — *The figure shows the development of synaptic weights (a)  $\tilde{w}_{1,1}^{1,1}$ , (b)  $\tilde{w}_{5,1}^{1,1}$ , (c)  $\tilde{w}_{3,2}^{1,1}$ , and (d)  $\tilde{w}_{8,2}^{1,1}$  in the preselection network of our case study. We parameterized the dynamics in eq. (3.16) with a learning velocity of  $\alpha = 0.3$ . All objective and subjective probabilities were initialized with random values taken from a standard normal distribution. Each subfigure comprises six test runs: In five test runs (grey curves) we periodically added noise  $\rho$  to the estimates of the objective probabilities (Eq. (3.18)). Every 50 cycles we added random values ranged between -10 and 10, after 200 cycles no more noise was added. In the sixth test run (black curve) no noise at all was added. Through inspection of the respective synaptic weights in fig. 3.11 we learn that the dynamics lets the ‘subjective’ weights always converge to the ‘objective’ ones in time.*



For a given partitioning  $\Pi^k$ , the set  $\Gamma^k(I)$  comprises all categories of the partitioning with super-threshold saliencies. The threshold is defined relative to the maximal saliency with a factor  $\theta^k$  with  $0 < \theta^k \leq 1$  (Eq. (3.20)), i.e., the  $\theta^k$  are relative thresholds. For  $\theta^k = 1$  only the most salient category will be selected, the decision rule becomes the winner-take-all nonlinearity.

$$\Gamma^k(I) = \left\{ \mathbb{C}_c^k \in \Pi^k \mid s_c^k(I) \geq \theta^k \cdot \max_{1 \leq c' \leq C^k} \{s_{c'}^k(I)\} \right\} \quad (3.20)$$

A set of *model candidates*  $\mathbb{M}(I)$  for the input image  $I$ , i.e., learning images of objects that reasonably may become models for the object in the input image, are calculated by set intersection on salient categories (Eq. (3.21)). The selected model candidates will be passed to the correspondence-based verification part for further selection.

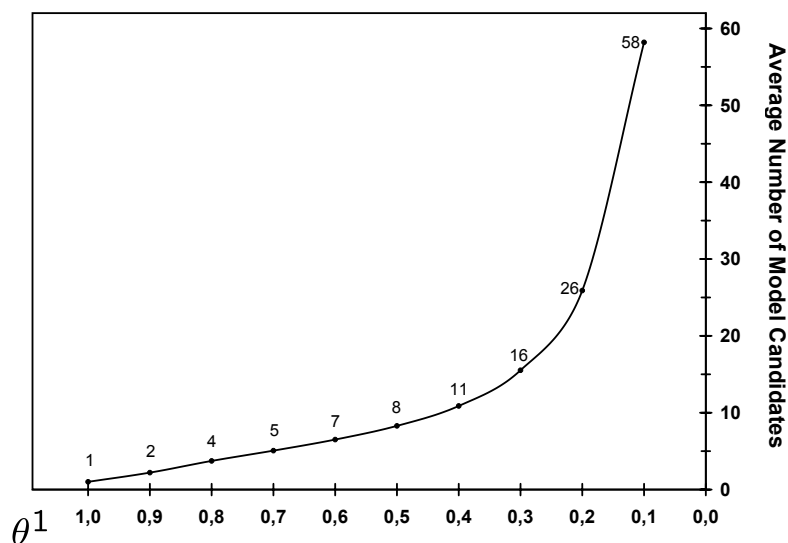
$$\mathbb{M}(I) = \bigcap_{k=1}^K \bigcup_{\mathbb{C} \in \Gamma^k(I)} \mathbb{C} \quad (3.21)$$

Fig. 3.16 gives the average numbers of model candidates in dependence on a relative threshold  $\theta^1$ . The experiment was carried out with the object recognition application proposed in the following chapter. The learning set comprised 5600 images taken from the COIL-100 database (NENE et al., 1996). From these images  $K = 1$  partitioning  $\Pi^1$  with  $C^1 = 5600$  single-element categories was created. We learn that, on average, the preselection network favorably rules out most irrelevant matches, i.e., the average numbers of model candidates are small relative to the total number of learning images, and that the average number of model candidates grows rapidly with decreasing relative thresholds. The average numbers of model candidates are, however, subjected to considerable mean variations, especially for small values of  $\theta^1$ .

In our case study, choosing  $\theta^1 = 1$  yields  $\Gamma^1(I'_2) = \{\mathbb{C}_2^1\}$  as the set of salient categories of partitioning  $\Pi^1$  and  $\mathbb{M}(I'_2) = \{I_2\}$  as the set of model candidates.

### 3.4.8 Accelerated Feature Search

In section 3.4.5 calculation of saliencies has been exemplified as a three-step procedure: extraction of features, search of matching model features in the visual dictionary, and computation of the saliencies. While steps one and three are efficiently executable on today's powerful hardware, the second step



**Figure 3.16: Average Number of Model Candidates in Dependence on a Relative Threshold** — *The average number of model candidates in dependence on the relative threshold  $\theta^1$  is given. The experiment was carried out with the object recognition application proposed in chapter 4. The learning set comprised 5600 images taken from the COIL-100 database (NENE et al., 1996). From these images one partitioning of single-element categories was created. We learn that, on average, the preselection network favorably rules out most irrelevant matches and that the average number of model candidates grows rapidly with decreasing relative thresholds. The given averages are, however, subjected to considerable mean variations, especially for small values of  $\theta^1$ . For clarity of presentation, we disregarded error bars.*

turns out to be a bottleneck in terms of execution time. This step is conceived to be performed in parallel for each feature of the visual dictionary (Section 3.4.2). However, on a general purpose computer this inherently parallel task is sequentially executed. Facing the amount of experiments to be conducted, an efficient implementation is desirable.

We propose a method that draws its efficiency from three sources. First, the method employs a coarse-to-fine strategy with respect to the number of features in the visual dictionary’s feature vectors: at the beginning, features are searched in a feature vector with relatively few features. Based on matching features that have been found in that feature vector, category

labels of salient categories of a chosen partitioning  $\Pi^{\tilde{k}}$  are collected in a set  $\mathbb{C}$ . That set is iteratively refined by searching for matching features in more and more detailed feature vectors. The set of salient categories  $\mathbb{C}$  is a global variable shared with read/write access among the algorithms explained below. Second, through application of dynamic thresholds on the measures of information, the method significantly reduces the number of model features to be compared with image features. Third, the method incorporates top-down knowledge encoded in the preselection network to assess which further model features appear in the salient categories and are thus considered worth being matched with image features.

### Computation of Saliencies

The computation of saliencies, given as pseudo code in algorithm 2 (Fig. 3.17), is implemented as a forward pass through the preselection network. The algorithm receives four parameters: the image  $I$  of an object to be recognized, an index  $\tilde{k}$  of a partitioning of the learning set, an integral value  $\theta_{hyst} > 0$ , and a factor  $\theta_{max}$ ,  $0 < \theta_{max} \leq 1$ . The function returns  $K$  saliency vectors  $\underline{s}^k(I)$ ,  $k \in \{1, \dots, K\}$ , one for each partitioning of the learning set. Parameter  $\tilde{k}$  specifies the partitioning of the learning set whose labels of salient categories are collected in  $\mathbb{C}$ ,  $\theta_{hyst}$  is an upper threshold of a hysteresis computed in algorithm 3 (Fig. 3.18), and  $\theta_{max}$  is an upper bound of the relative threshold used in algorithm 4 (Fig. 3.19).

For efficient computation of model graphs in the correspondence-based verification part (Section 3.5), pairs of matching image and model features are collected in a table of matching features  $\mathcal{F}_{match}(I)$ , which is also a global variable with read/write access.

At the beginning, the set of salient categories  $\mathbb{C}$  and the table of matching features  $\mathcal{F}_{match}(I)$  are emptied, and all saliencies are reset to zero. The saliencies are computed in nested loops over all feature vectors, over all matching model features, and over all categories: the current category's saliency is incremented by the connection strength of the synapse between the input neuron assigned to the feature detector that employs the current model feature as reference feature and the output neuron assigned to the current category. If the entire visual directory has been traversed, the algorithm returns the saliency vectors. Model candidates can be selected using eq. (3.21).

**Algorithm 2:** *computeSaliencies*

**Parameter:** Input Image:  $I$   
**Parameter:** Index of Partitioning:  $\tilde{k}$   
**Parameter:** Threshold of Hysteresis:  $\theta_{hyst}$   
**Parameter:** Upper bound of relative threshold:  $\theta_{max}$   
**Result** : Vectors of Saliencies:  $\underline{s}^1(I), \dots, \underline{s}^K(I)$

```

1  $\mathbb{C} \leftarrow \emptyset$ 
2  $\mathcal{F}_{match}(I) \leftarrow \emptyset$ 
3 for  $k \leftarrow 1$  to  $K$  do
4   | for  $c \leftarrow 1$  to  $C^k$  do
5   |   |  $s_c^k(I) \leftarrow 0$ 
6   |   end
7   end

8 for  $r \leftarrow 1$  to  $R$  do
9   |  $\mathbb{F}_{Image} \leftarrow f^r(I)$ 
10  |  $\mathbb{T}_{match} \leftarrow searchMatchingFeatures(\mathbb{F}_{Image}, r, \tilde{k}, \theta_{hyst}, \theta_{max})$ 
11  | forall  $t \in \mathbb{T}_{match}$  do
12  |   | for  $k \leftarrow 1$  to  $K$  do
13  |   |   | for  $c \leftarrow 1$  to  $C^k$  do
14  |   |   |   |  $s_c^k(I) \leftarrow s_c^k(I) + w_{t,c}^{r,k}$ 
15  |   |   |   end
16  |   |   end
17  |   end
18 end

19 return  $\underline{s}^1(I), \dots, \underline{s}^K(I)$ 

```

**Figure 3.17: Computation of Saliencies** — *The computation of saliencies is implemented as a forward pass through the preselection network: the saliency of the current category is incremented by the connection strength of the synapse between the input neuron assigned to the feature detector that employs the current model feature as reference feature and the output neuron assigned to the current category. If the entire visual directory has been traversed, the algorithm returns the saliency vectors. Parameters  $\tilde{k}$  and  $\theta_{hyst}$  are passed to algorithm 3 (Fig. 3.18).*

### Search Matching Features

Algorithm 3 (Fig. 3.18) searches for matching model features in a so-called *search space*  $\mathbb{T}_{search}$ , a set containing the indices of model features of the current feature vector considered worth being matched with image features. This set is computed by algorithm 4 (Fig. 3.19). The indices of matching model features are collected in a set  $\mathbb{T}_{match}$ . The algorithm is further concerned with the assessment whether calculation can be terminated without having processed all image features in order to accelerate execution. For this purpose, the algorithm computes a hysteresis  $hyst$  with respect to the set of salient categories  $\mathbb{C}$  of partitioning  $\Pi^{\tilde{k}}$ . That set is expanded by the labels  $c$  of those categories that comprise at least one image in which the current model feature  $f_t^r$  can be observed. If that set does not change for the current feature,  $hyst$  is incremented, and reset to 0 otherwise. If  $\mathbb{C}$  has not changed for  $\theta_{hyst}$  iterations it is assumed that it will not change for the remaining image features as well. In that case the algorithm returns the set of indices of matching features without having processed all image features.

### Definition of Search Spaces

Algorithm 4 (Fig. 3.19) computes the search space  $\mathbb{T}_{search}$ , a set containing the indices of model features of the current feature vector considered worth being matched with image features. The computation is based on the application of dynamic thresholds and on the incorporation of top-down knowledge which model features can be observed in the images of salient categories. That knowledge is encoded in the preselection network. If the set of salient categories  $\mathbb{C}$  is empty the search space comprises the indices of all features of the current feature vector. Otherwise, a relative threshold  $\theta$  is computed that ranges between 0 and  $\theta_{max}$ ; its value scales proportionally with the current number of salient categories. Given that relative threshold, all categories with a saliency below  $\theta$  times the maximal saliency are discarded from the set of salient categories. The search space contains the indices of those model features that can be observed in at least one image of a salient category and whose measure of information is greater or equal  $\theta$  times the maximal measure of information.

**Algorithm 3:** *searchMatchingFeatures*

**Parameter:** Set of Image Features:  $\mathbb{F}_{Image}$   
**Parameter:** Index of the Current Feature Vector:  $r$   
**Parameter:** Index of Partitioning:  $\tilde{k}$   
**Parameter:** Threshold of Hysteresis:  $\theta_{hyst}$   
**Parameter:** Upper bound of relative threshold:  $\theta_{max}$   
**Result** : Set of Matching Features:  $\mathbb{T}_{match}$

```

1  $\mathbb{T}_{match} \leftarrow \emptyset$ 
2  $\mathbb{T}_{search} \leftarrow \text{defineSearchSpace}(r, \tilde{k}, \theta_{max})$ 
3  $hyst \leftarrow 0$ 
4 forall  $f \in \mathbb{F}_{Image}$  do
5   forall  $t \in \mathbb{T}_{search}$  do
6     if  $\varepsilon(f, f_t^r, \vartheta^r) = 1$  then
7        $\mathbb{T}_{match} \leftarrow \mathbb{T}_{match} \cup \{t\}$ 
8        $\mathcal{F}_{match}(I) \leftarrow \mathcal{F}_{match}(I) \cup \{(f, f_t^r)\}$ 
9        $\mathbb{C}' \leftarrow \mathbb{C}$ 
10       $\mathbb{C} \leftarrow \mathbb{C} \cup \{c \mid 1 \leq c \leq C^{\tilde{k}} \wedge w_{t,c}^{r,\tilde{k}} \neq 0\}$ 
11      if  $\mathbb{C}' = \mathbb{C}$  then
12         $hyst \leftarrow hyst + 1$ 
13      else
14         $hyst \leftarrow 0$ 
15      end
16      if  $hyst > \theta_{hyst}$  then
17        return  $\mathbb{T}_{match}$ 
18      end
19    end
20  end
21 end
22 return  $\mathbb{T}_{match}$ 

```

**Figure 3.18: Search Matching Features** — *The algorithm searches for matching model features in the search space and returns a set of indices of matching model features. Moreover, for efficient construction of model graphs, pairs of matching features are collected in a table. For the assessment whether the algorithm may be terminated without having processed all image features a hysteresis is computed.*

**Algorithm 4:** *defineSearchSpace*

**Parameter:** Index of the Current Feature Vector:  $r$   
**Parameter:** Index of Partitioning:  $\tilde{k}$   
**Parameter:** Upper bound of relative threshold:  $\theta_{max}$   
**Result** : Search Space:  $\mathbb{T}_{search}$

```

1 if  $\mathbb{C} = \emptyset$  then
2   |  $\mathbb{T}_{search} \leftarrow \{t \mid 1 \leq t \leq T^r\}$ 
3 else
4   |  $\theta \leftarrow \theta_{max} \cdot |\mathbb{C}| \cdot (C^{\tilde{k}})^{-1}$ 
5   |  $\mathbb{C} \leftarrow \left\{ c \in \mathbb{C} \mid s_c^{\tilde{k}}(I) \geq \theta \cdot \max_{1 \leq c' \leq C^{\tilde{k}}} \{s_{c'}^{\tilde{k}}(I)\} \right\}$ 
6   |  $\mathbb{T}_{search} \leftarrow \bigcup_{c \in \mathbb{C}} \left\{ t \mid 1 \leq t \leq T^r \wedge w_{t,c}^{r,\tilde{k}} \geq \theta \cdot \ln C^{\tilde{k}} \right\}$ 
7 end
8 return  $\mathbb{T}_{search}$ 

```

**Figure 3.19: Definition of Search Spaces** — *The algorithm computes the search space, a set containing the indices of model features of the current feature vector considered worth being matched with image features. The computation is based on the application of dynamic thresholds on the measures of information and on the saliencies. The algorithm incorporates top-down knowledge which model features can further be observed in the images of salient categories. This knowledge is encoded in the preselection network.*

---

## Performance

We analyze the performance of the accelerated feature search in terms of recognition rate and execution time with the object recognition application proposed in the following chapter. The learning set comprised 5600 images drawn from the COIL-100 database (NENE et al., 1996). From these images the same number of single-element categories have been built:  $\Pi^1 = \{\mathbb{C}_c^1 \mid 1 \leq c \leq 5600\}$ . The only learning example in the most salient category was chosen as the model for the input image. We considered an object to be correctly recognized if test and model image contained the same object. We analyze the performance for  $\theta_{hyst} = 1, 3, 5, 10,$  and  $15,$  for  $\tilde{k} = 1,$  and

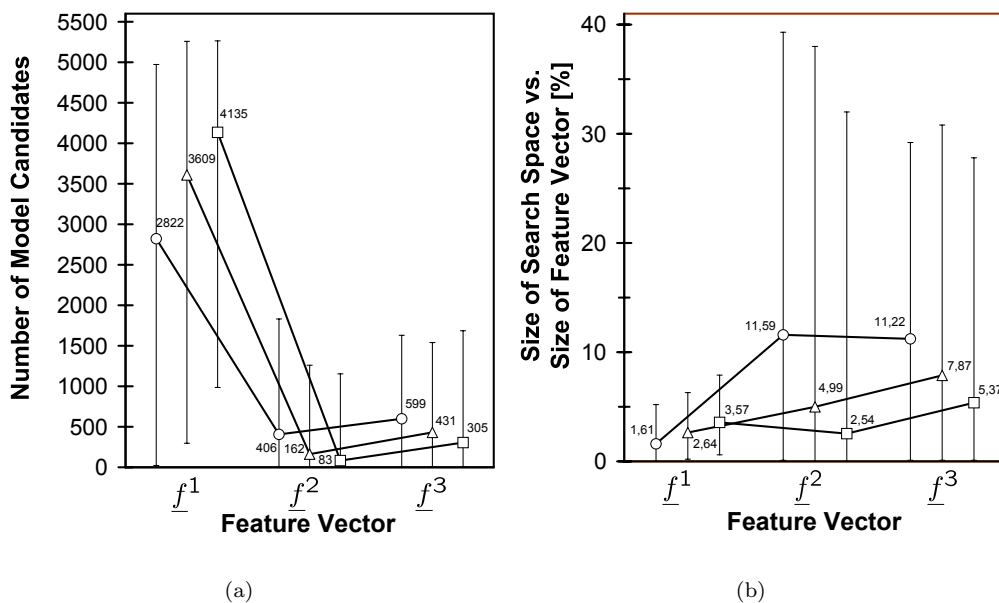
$\theta_{hyst}$	RECOGNITION RATE [%]	EXECUTION TIME [s]
1	98.2	4.76
3	98.8	5.29
5	99.0	11.91
10	99.0	21.38
15	98.6	28.63
$\infty$	97.7	93.03

**Table 3.4: Performance of the Accelerated Feature Search** — *The table lists the recognition rates and execution times attained by the object recognition application of the following chapter. We analyze the performance of the accelerated feature search for  $\theta_{hyst} = 1, 3, 5, 10,$  and  $15,$  for  $\tilde{k} = 1,$  and for  $\theta_{max} = 0.3.$  Moreover, in row  $\theta_{hyst} = \infty$  the table lists the attained recognition rate and execution time if the accelerated feature search was disabled, i.e., all three feature vectors were fully traversed in search of matching features. The false alarm rate of accidental feature matches depends proportionally on the hysteresis threshold  $\theta_{hyst}$ : for larger values of  $\theta_{hyst}$  recognition rates decrease and execution times increase while smaller values of  $\theta_{hyst}$  allow for faster execution at the expense of lower recognition rates. Since accidental feature matches are frequent in the case of disabled accelerated feature search, the worst recognition rate and the slowest execution time was attained for that case.*

for  $\theta_{max} = 0.3.$  We designed the visual dictionary to contain three feature vectors, sorted according to their length in ascending order: feature vector  $\underline{f}^1$  comprised 32,972,  $\underline{f}^2$  comprised 89,127, and  $\underline{f}^3$  comprised 210,189 parquet graphs. They were computed using the procedure proposed in section 3.3 using similarity thresholds of  $\vartheta^1 = 0.85,$   $\vartheta^2 = 0.9,$  and  $\vartheta^3 = 0.95.$

The method’s performance in terms of recognition rate and execution time is given in tab. 3.4. The performance with respect to the number of model candidates and to the sizes of search spaces is given in fig. 3.20. Generally, recognition rates decrease and execution times increase for larger values of  $\theta_{hyst}$  while smaller values of  $\theta_{hyst}$  allow for faster execution at the expense of lower recognition rates.





**Figure 3.20: Performance of the Accelerated Feature Search** — The figure shows the performance of the accelerated feature search with (a) respect to the number of model candidates and with (b) respect to the size of the search space. We give the results for  $\theta_{hyst} = 5, 10, 15$ . In both subfigures the results for  $\theta_{hyst} = 5$  are marked with circles, those for  $\theta_{hyst} = 10$  are marked with triangles, and those for  $\theta_{hyst} = 15$  are marked with boxes. Figure (a) displays the number of salient categories each time a feature vector of the visual dictionary has been processed. Since the learning set has been subdivided into one-element categories the current number of model candidates is equal to the current number of salient categories. The method allows for selection of a rather small number of model candidates relative to the total number of learning examples. The recognition rates in tab. 3.4 prove that the selection is sound. Figure (b) displays the sizes of search spaces versus the size of the current feature vector. This is, informally speaking, the percentage of the current feature vector to traverse in search of matching model features. The search spaces are always much smaller than the current feature vectors even though the mean variation is considerable. The number of salient categories and the sizes of search spaces are determined by dynamic thresholds which keep them in manageable ranges in terms of execution time.

## 3.5 Verification of Model Candidates

Up to here, model candidates have been selected by set intersection on salient categories (Eq. (3.21)). The categories' saliencies as computed by the preselection network are solely based on the detection of coincidental features in the model and image domain. 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 through asserting 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 assembling 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 input image. Its model graph is the closest possible representation of the object in the input image with respect to the learning set.

### 3.5.1 Construction of Graphs

Construction of graphs proceeds in three steps. First, from the table of matching features (Eq. (3.7)) 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 unlabeled nodes. Number and positioning of nodes is determined by the valid nodes of image and model parquet graphs. Third, at each node position, separately for image and model graph, a bunch of Gabor jets is assembled whose jets stem from node labels of valid-labeled parquet graph nodes located at that position. The respective nodes of the image or model graph become attributed with these bunches.

#### Table of Corresponding Features

During calculation of the categories' saliencies pairs of matching features have been collected in a table of matching features  $\mathcal{F}_{match}(I)$  (Eq. (3.7)). Given a model candidate  $M \in \mathbb{M}(I)$  for the input image  $I$  (Eq. (3.21)), all feature

pairs whose model feature stems from  $M$  are transferred to a table of *corresponding* features  $\mathcal{F}_{corr}(I, M)$ , which will be used for efficient aggregation of parquet graphs into larger model and image graphs. We assume that the table comprises  $N$  feature pairs, a number that depends implicitly on the model candidate. Let  $f_n^I$  denote the image and  $f_n^M$  the model parquet graph of the  $n$ -th feature pair. 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.

$$\mathcal{F}_{corr}(I, M) = \left\{ (f_n^I, f_n^M) \in \mathcal{F}_{match}(I) \mid 1 \leq n \leq N \wedge H \left( \sum_{r=1}^R \sum_{f \in f^r(M)} \varepsilon(f, f_n^M, 1) \right) = 1 \right\} \quad (3.22)$$

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 (Section 3.2). For being able 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.

$$\begin{aligned} f_n^I &= \{ (\underline{x}_{n,v}^I, \mathcal{J}_{n,v}^I, b_{n,v}^I) \mid 1 \leq v \leq V \} \\ f_n^M &= \{ (\underline{x}_{n,v}^M, \mathcal{J}_{n,v}^M, b_{n,v}^M) \mid 1 \leq v \leq V \} \end{aligned} \quad (3.23)$$

### 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. The creation of graph templates is illustrated in fig. 3.21.

$$\begin{aligned} \mathbb{X}^I &= \bigcup_{n,v} \{ \underline{x}_{n,v}^I \mid b_{n,v}^I = 1 \} \\ \mathbb{X}^M &= \bigcup_{n,v} \{ \underline{x}_{n,v}^M \mid b_{n,v}^M = 1 \} \end{aligned} \quad (3.24)$$

## 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, the jets stem from 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. Whenever possible we omit the position  $\underline{x}$  and write  $\beta^I$  and  $\beta^M$ . The assembly of Gabor jets into bunches is also illustrated in fig. 3.21.

$$\begin{aligned}\beta^I(\underline{x}) &= \bigcup_{n,v} \{ \mathcal{J}_{n,v}^I \mid \underline{x}_{n,v}^I = \underline{x} \wedge b_{n,v}^I = 1 \} \\ \beta^M(\underline{x}) &= \bigcup_{n,v} \{ \mathcal{J}_{n,v}^M \mid \underline{x}_{n,v}^M = \underline{x} \wedge b_{n,v}^M = 1 \}\end{aligned}\quad (3.25)$$

For the assessment whether a point in the image corresponds to a point in the model candidate a measure of similarity between two bunches is needed. The similarity between two bunches 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 eq. (2.9), which is based on the Gabor amplitudes.

$$s_{bunch}(\beta, \beta') = \begin{cases} 0 & \text{if } \beta = \emptyset \vee \beta' = \emptyset \\ \max_{\mathcal{J} \in \beta, \mathcal{J}' \in \beta'} \{ s_{abs}(\mathcal{J}, \mathcal{J}') \} & \text{otherwise} \end{cases} \quad (3.26)$$

## Graphs

Like parquet graphs, image and model graphs are specified by a set of node labels. Node labels comprise an absolute position in the input or model image drawn from the sets of node positions (Eq. (3.24)) and the bunch assembled at that position (Eq. (3.25)). The image graph is decorated with a superscript  $I$  while the model graph receives a superscript  $M$ .

$$\begin{aligned}\mathcal{G}^I &= \bigcup_{\underline{x} \in \mathbb{X}^I} \{ (\underline{x}, \beta^I(\underline{x})) \} \\ \mathcal{G}^M &= \bigcup_{\underline{x} \in \mathbb{X}^M} \{ (\underline{x}, \beta^M(\underline{x})) \}\end{aligned}\quad (3.27)$$

Model graphs of suited model candidates provide an approximation of the object in the input image by features present in the visual dictionary. Fig. 3.2

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 four demonstrate that the emerged model graphs describe the object in the input image well.

The constructed graphs are to some extent reminiscent of bunch graphs (WISKOTT, 1995; WISKOTT et al., 1997). Nevertheless, since they represent single model candidates, we rather speak of model instead of bunch graphs. It is, however, worthwhile mentioning that the proposed procedure may as well serve for the construction of bunch graphs. To this end the table of corresponding features has to provide feature pairs of model candidates picked from a carefully chosen subset  $\tilde{\mathbb{M}}(I)$  of the set of model candidates  $\mathbb{M}(I)$ . The alternative computation of the table of corresponding features is given in eq. (3.28). The graph construction procedure is then as well applicable to the construction of bunch graphs.

$$\mathcal{F}_{corr}^{bunch}(I, \tilde{\mathbb{M}}(I)) = \bigcup_{M \in \tilde{\mathbb{M}}(I)} \mathcal{F}_{corr}(I, M) \quad (3.28)$$

### 3.5.2 Matching

In order to assert 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). It is also very similar to multidimensional template matching (WÜRTZ, 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 leftmost node is not necessarily 0. The same is the case for the  $x$  coordinate of the uppermost node.

$$\underline{t}_0 = \left( \min_{n,v} \left\{ (\underline{x}_{n,v}^M)_x \right\}, \min_{n,v} \left\{ (\underline{x}_{n,v}^M)_y \right\} \right)^T \quad (3.29)$$

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.

$$s(I, M, \underline{t}) = |\mathcal{G}^M|^{-1} \cdot \sum_{(\underline{x}^M, \beta^M) \in \mathcal{G}^M} s_{bunch}(\beta^I(\underline{x}^M - \underline{t}_0 + \underline{t}), \beta^M) \quad (3.30)$$

In order to find the object in the input image the model graph is iteratively translated about a displacement vector in the image plane so that the measure of similarity between model and image graph becomes maximal. The model graph of a suited model candidate moves to the object's position in the input image. Let  $s_{best}(I, M)$  denote the similarity attained at that position. The displacement vectors  $\underline{t}$  stem from a set  $\mathbb{G} = \{(n\Delta x, n\Delta y) | n \in \mathbb{N}_0\}$  of all grid points defined by the given distances  $\Delta x$  and  $\Delta y$  between neighbored parquet graph nodes (Section 3.3).

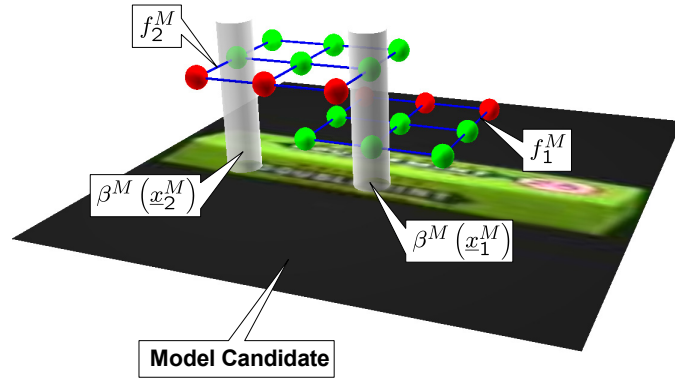
$$s_{best}(I, M) = \max_{\underline{t} \in \mathbb{G}} \left\{ s(I, M, \underline{t}) \right\} \quad (3.31)$$

### 3.5.3 Model Selection

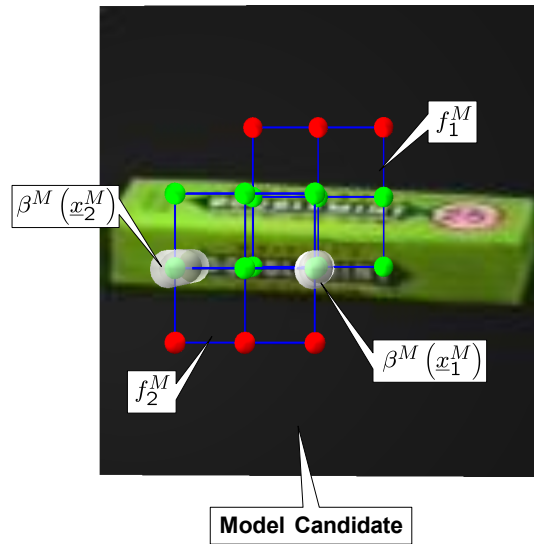
For selection of the model, the most similar learning image for the given input image, an image and a model graph are constructed for each model candidate. The model candidate that attains the best similarity between its model and image graph is chosen as the model for the input image.

$$M_{best} = \arg \max_{M \in \mathbb{M}(I)} \left\{ s_{best}(I, M) \right\} \quad (3.32)$$

In fig. 3.2 four model candidates (column two) have been computed for the given input image (column one). The similarities attained through matching image against model graphs are annotated to the reconstructions from the model graphs (column four). Since the first model candidate yields the highest similarity, it is chosen as the model for the object in the input image.

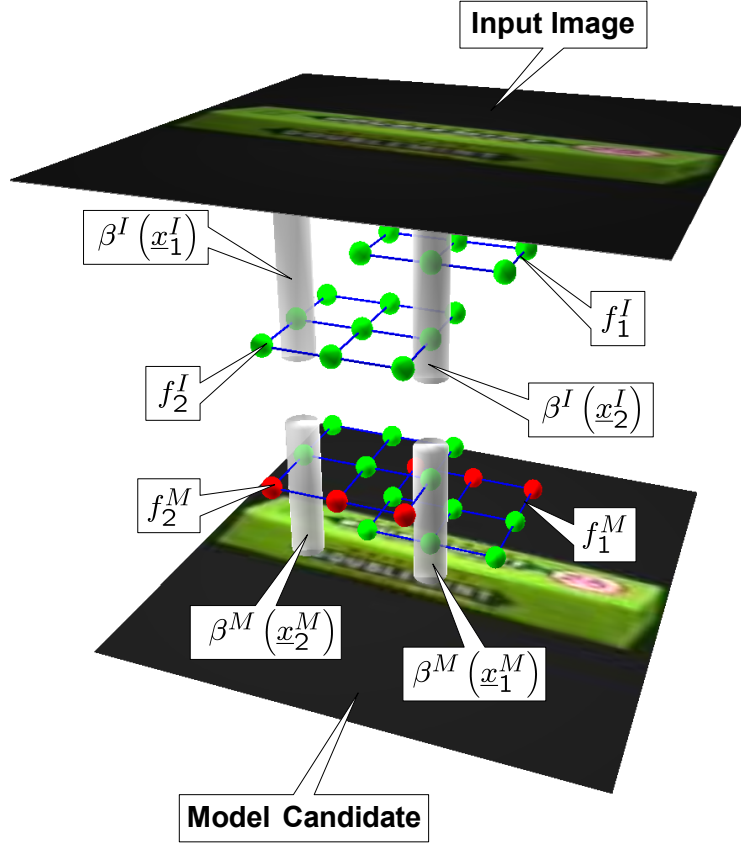


(a)



(b)

**Figure 3.21: Construction of Model Graphs** — *Figure (a) provides a side, figure (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 (green nodes). Nodes that reside in the background have been marked as invalid (red nodes). In figure (b) the shape of the emerging model graph can be predicted. 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 figure (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 very same fashion.*



**Figure 3.22: 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  $f_2^M$ . Like in fig. 3.21, green nodes represent nodes that have been marked as valid and red nodes represent nodes that have been marked as invalid for residing in the background. Since only learning images provide figure-ground information, invalid nodes appear only in the model parquet graphs. The compilation of bunches is illustrated for two 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.*



## 3.6 Parameterization

Except where mentioned otherwise, the following parameters are used throughout this thesis.

### 3.6.1 Gabor Features

The parameterization of Gabor features as given below is the same as in (LADES et al., 1993; WISKOTT, 1995).

$L = 8$	number of directions
$M = 5$	number of frequencies
$k_{step} = \sqrt{2}$	factor used for the sampling of frequencies
$k_{max} = \frac{\pi}{2}$	maximal frequency
$\sigma = 2\pi$	ratio of the width of the Gaussian window to wavelength, i.e., this parameter specifies the number of oscillations under the Gaussian envelope

### 3.6.2 Parquet Graphs

For the given parameterization of the Gabor features, the parameterization of the parquet graph features (Section 3.2) is as follows.

$V = 9$	number of nodes
$\Delta x = 10$	distance in pixels between two neighbored nodes in horizontal direction
$\Delta y = 10$	distance in pixels between two neighbored nodes in vertical direction

### 3.6.3 Visual Dictionaries

Visual dictionaries (Section 3.3) comprise  $R = 2$  feature vectors  $\underline{f}^1$  and  $\underline{f}^2$ . These are computed by the vector quantization method given in algorithm 1 (Fig. 3.9) using two feature calculators  $f^1$  and  $f^2$  that return parquet graph features, which are parameterized as given above.

- $R = 2$       number of feature calculators
- $\vartheta^1 = 0.9$     similarity threshold of feature calculator  $f^1$
- $\vartheta^2 = 0.95$    similarity threshold of feature calculator  $f^2$

### 3.6.4 Accelerated Feature Search

The accelerated feature search (Section 3.4.8) is parameterized as given below. We consider that  $\tilde{k}$  specifies a partitioning of the learning set with single-element categories. If not explicitly given, we consider that such a partitioning is implicitly defined. The given parameterization allows for high recognition or categorization rates at the expense of relatively slow execution (Tab. 3.4).

- $\theta_{hyst} = 10$     threshold of hysteresis (Algorithm 3, Fig. 3.18)
- $\theta_{max} = 0.3$     upper bound of relative threshold (Algorithm 4, Fig. 3.19)

# Chapter 4

## Object Recognition

*How can I qualify my faith in the inviolability of the design principles? Their virtue is demonstrated. They work.*

Edgar A. Whitney

In this chapter we apply the graph dynamics proposed in the previous chapter to the task of invariant object recognition. Following PALMERI and GAUTHIER (2004) the term *recognition* refers to a decision about an object's unique identity. Visual recognition thus requires to discriminate between similar objects and involves generalization across some object variations as well as translation, scale, pose, occlusion, illumination, and noise. Because of the multitude of possible object variations, recognition of objects is a difficult task.

### 4.1 Review of Literature

Theories of human object recognition come in two main types: viewpoint-independent and view- or appearance-based ones. Viewpoint-independent theories posit that objects are represented as an object-centered arrangement of volumetric primitives like cylinders, cones etc., much like the solid geometrical models used in computer-aided design. These representations are viewpoint-invariant in the sense that the same three-dimensional representa-

tion is derived over a wide range of viewing orientations. Theories following this paradigm have, for instance, been proposed in (MARR and NISHIHARA, 1978; BIEDERMAN, 1987). They state that recognition performance is independent of the particular viewpoint, which appears compatible with our intuition as recognition of familiar objects from unfamiliar viewpoints seems effortless. However, psychophysical experiments (BÜLTHOFF and EDELMAN, 1992; TARR, 1995) suggest otherwise: observers that have learned to recognize novel objects from specific viewpoints are both faster and more accurate at recognizing these same objects from those familiar viewpoints relative to unfamiliar viewpoints. Moreover, recognition performance at unfamiliar viewpoints is related to familiar views: subjects progressively needed more time and were less accurate if the distances between the familiar and unfamiliar views were gradually increased. These results suggest that human object recognition relies on multiple *views*, where a view encodes the appearance of an object under specific viewing conditions, such as pose, illumination and so on. It is therefore reasonable to assume that the mental representation of a given object is constituted by a collection of memorized views. The assumption that the brain transforms unfamiliar views into familiar ones through mental rotation has, however, been disproved in (GAUTHIER et al., 2002). BÜLTHOFF and EDELMAN (1992) rather suggested that the brain interpolates between familiar views of a given object during recognition.

There exist two types of computational appearance-based models for invariant visual object recognition: feature-based and correspondence-based. Both start with the extraction of *features*. In feature-based recognition systems, invariance to position, scale and so on is achieved feature-wise, with the help of a logical OR: parameter-dependent feature detectors pass their assessment whether their parameter-dependent reference feature is present 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. For instance, the feature detector given in section 3.4.2 is invariant to the reference feature's position in the image plane. Object recognition is achieved by comparing the list of activated master units to stored lists 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 on the spatial arrangement of local features is given up. Examples of feature-based systems include the Neocognitron (FUKUSHIMA et al., 1983), Edelman (EDELMAN, 1995), Murase & Nayar (MURASE and NAYAR, 1995), SEEMORE (MEL, 1997), VisNet (ELLIFFE et al., 2002), and (WERSING and KÖRNER, 2003). Since these methods do not solve the binding problem (VON DER MALSBERG, 1981, 1999), they en-

counter problems when confronted with more sophisticated input images, for instance, images with structured backgrounds, multiple objects, or occluded objects. As especially the spatial arrangement of features is disregarded, they leave the door open for the confusion of objects that agree in features but differ in the features' spatial arrangement, scale or orientation. It is, however, argued that non-ambiguous representations can be achieved through introduction of combination-coding units, see, for instance, (MEL, 1997).

In correspondence-based model objects are represented as ordered arrays of local features. For instance, in elastic graph matching memorized object views are represented by model graphs (Chapter 2). Models are matched with the image by solving the correspondence problem, i.e., through establishment of an organized set of point-to-point correspondences between points in the image and in the object model. Examples of correspondence-based systems include (VON DER MALSBERG, 1988; ULLMAN, 1989; HUMMEL and BIEDERMAN, 1992; LADES et al., 1993; OLSHAUSEN et al., 1993; WÜRTZ, 1995; VON DER MALSBERG and REISER, 1995; WISKOTT, 1995; WISKOTT et al., 1997; MESSMER and BUNKE, 1998). Correspondence-based methods usually encounter problems when applied to larger repertoires of general objects. As proposed and experimentally supported by BIEDERMAN (1987) objects have to be represented as structured arrays of object parts, i.e., object models are required to be dynamic with respect both to shape and features.

## 4.2 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 et al., 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 five degrees. This corresponds to 72 poses per object identity and 7200 images for the whole collection. All images are  $128 \times 128$  pixels in size. The images were 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 the same as for the COIL-100 database. All images are  $192 \times 144$  pix-



**Figure 4.1: Example Images of the COIL-100 and of the ALOI Image Database** — *The figure gives example images drawn from the COIL-100 (NENE et al., 1996) and from the ALOI image database (GEUSEBROEK et al., 2005). The images in (a) and (b) stem from the COIL-100, those in (c) and (d) stem from the ALOI database. In contrast to the ALOI images, the objects in the COIL-100 images are normalized in size.*

els in size. As the employed procedure of feature extraction (Section 3.3) failed to compute visual dictionaries with manageable numbers of features in terms of execution time for more than 100 objects, we selected a subset of 100 objects from the database’s object view collection. Since the images of the first 200 objects were considered as too dark, we decided for objects number 200-299. The chosen subset consists of 7200 images. Compared to the COIL-100 database, less effort was invested in image preprocessing in the case of the ALOI database. Especially, the images are much darker, the objects are not normalized in size, and cover a much smaller fraction of the image. 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 forthcoming experiments will show. Some example images of both databases are given in fig. 4.1.

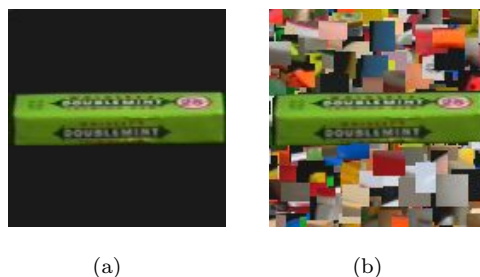
Experimental results were obtained with fivefold cross-validation (WITTEN and FRANK, 2000). In  $N$ -fold cross-validation, the data is split into  $N$  partitions of approximately the same size; we decided for  $N = 5$  partitions. Each of them is once used for testing while the remaining  $N - 1$  partitions are used for training. 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. The learning sets comprised 56, the testing sets 14 views per object, thus, 5600 or 1400 images in total, respectively. From each learning set a visual dictionary with two feature vectors was calculated. We used the default parameter set given in section 3.6. The learning images were perfectly segmented, i.e., the objects were placed in front of a plain black background. In some experiments we added structured backgrounds to the test images before presentation. The object recognition application was designed to simultaneously recognize the presented object's identity and pose. This was achieved by creating  $K = 1$  partitioning of the learning set. That partitioning consisted of  $C^1 = 5600$  single-element categories.

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  $\theta^1$  (Eq. (3.20)) that ranged between 0.1 and 1, sampled in 0.1-steps.  $\theta^1$  determined the final number of model candidates that were passed to the correspondence-based verification part. For  $\theta^1 = 1$  only one model candidates was selected while for low values the set of model candidates encompassed a large number of the original training images. This parameter thus allowed to adjust the balance between the feature- and correspondence-based parts.

### 4.3 Experiments

We present the results of seven experiments: The first experiment (Section 4.3.1) was concerned with the recognition of single objects with respect to object identity and pose. The second experiment (Section 4.3.2) was supposed to demonstrate the usefulness of the correspondence-based verification of model candidates in that recognition rates obtained using that method were compared with recognition rates achieved with an alternative method, in which a majority vote was implemented. The third experiment (Section 4.3.3) dealt with the recognition of single scaled objects. The fourth experiment (Section 4.3.4) determined the system's recognition performance if it was trained on sparse learning sets, i.e., the learning sets contained fewer training examples per object identity. The fifth experiment (Section 4.3.5) was concerned with the recognition of single objects in the case of sparse visual dictionaries, i.e., the number of features per learning example was



**Figure 4.2: Input Images of a Single Object** — *The figure shows an object from the COIL-100 database (NENE et al., 1996) as (a) segmented and (b) unsegmented image. Since the images of that database are perfectly segmented, unsegmented test images were manually created before presentation by pasting the object in the segmented image into a cluttered background consisting of arbitrarily chosen image patches of random size derived from the images of the current testing set.*

---

reduced. In the sixth experiment (Section 4.3.6) recognition performance was evaluated in the case of input images that contained multiple, non-overlapping objects. Finally, the seventh experiment (Section 4.3.7) dealt with the recognition of partially occluded objects.

### 4.3.1 Recognition of Single Objects

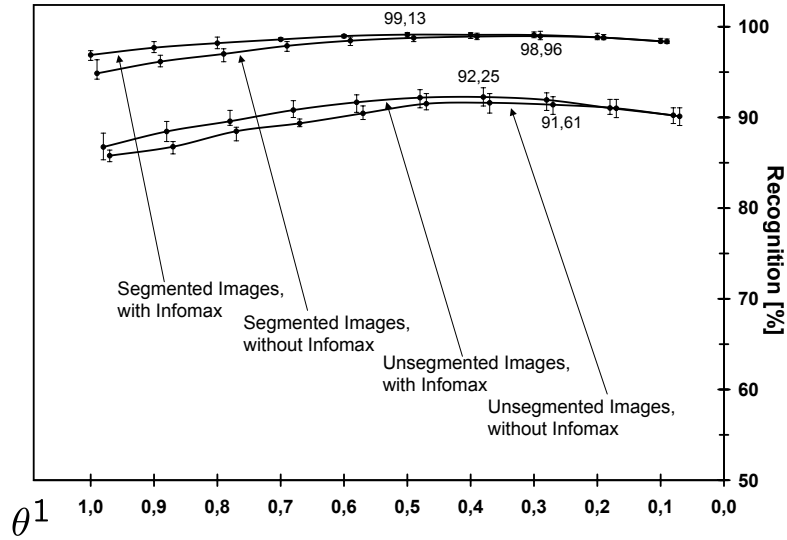
In this experiment we presented images containing a single object and evaluated the recognition performance with respect to object identity and pose. We analyzed the system's performance for each of the combinations segmented/unsegmented images and preselection network conforming/non-conforming to the infomax principle (Section 3.4). The experiment was subdivided into eight test cases per database. In the first four test cases, recognition performance with respect to object identity was evaluated for each of the mentioned combinations while the system's ability to recognize the objects' poses was investigated in the remaining four test cases. 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 the worst possible background for feature-based systems. Fig. 4.2 shows an example of



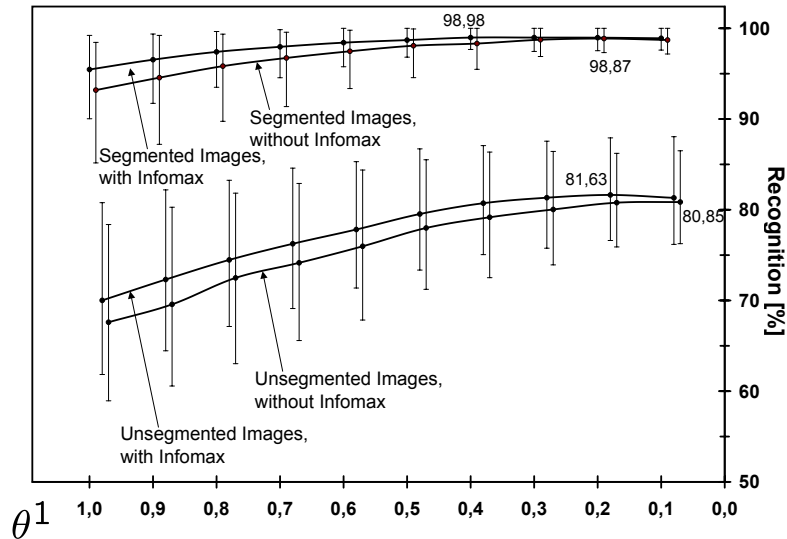
a segmented and of an unsegmented test image. In order to assess the usefulness of the choice of synaptic weights according to eq. (3.11), the preselection networks were made incompatible to the infomax principle by putting their weights out of tune according to eq. (4.1). Choosing the synaptic weights in this fashion let the categories' saliencies become simple counters of feature coincidences, the weighted majority voting scheme degenerated to a non-weighted one.

$$\underline{\hat{W}}^{r,k} = \left( H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} =: \left( \hat{w}_{t,c}^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} \quad (4.1)$$

The recognition performance with respect to object identity is shown in fig. 4.3. We considered the object in the test image to be correctly recognized if test and model image showed the same object identity regardless of the object's pose. Throughout, better recognition rates were attained if segmented images were presented. Moreover, the infomax principle always slightly improved performance. That improvement, however, decreased when the correspondence-based part was emphasized, i.e., the achieved improvement was continually used up while moving from the left to the right hand side of fig. 4.3. Most interestingly, a well-balanced combination of the feature- and correspondence-based parts led to optimal performance, throughout. Only for such well-balanced combinations the selection of model candidates was optimally carried out in the sense that neither too few nor too many learning 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 became 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 learning images of weakly salient categories 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. The same findings apply for the performance with respect to object pose given in fig. 4.4. The average pose errors were calculated over the absolute values of angle differences of correctly recognized, non-rotation-symmetric objects. Note that two consecutive learning images of the same object were at least five 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 example, for robot grasping, see, for instance, (SCHMIDT and WESTPHAL, 2004), the number of misclassified poses is more relevant than the mean pose error.

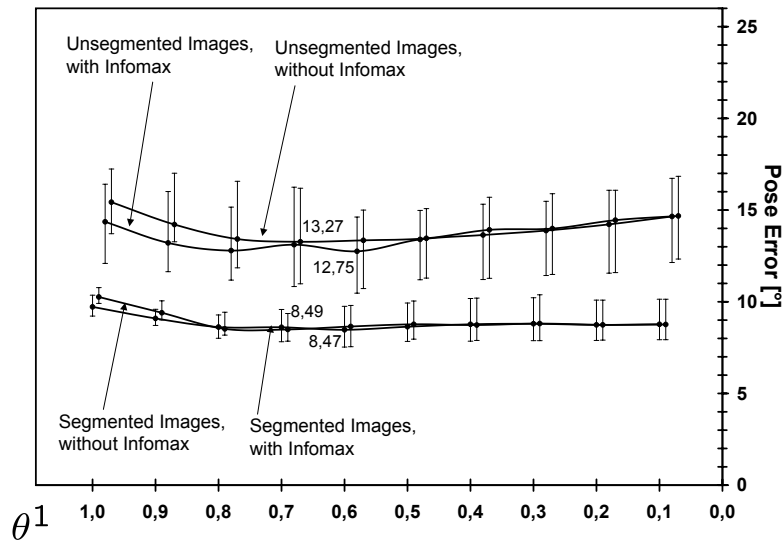


(a)

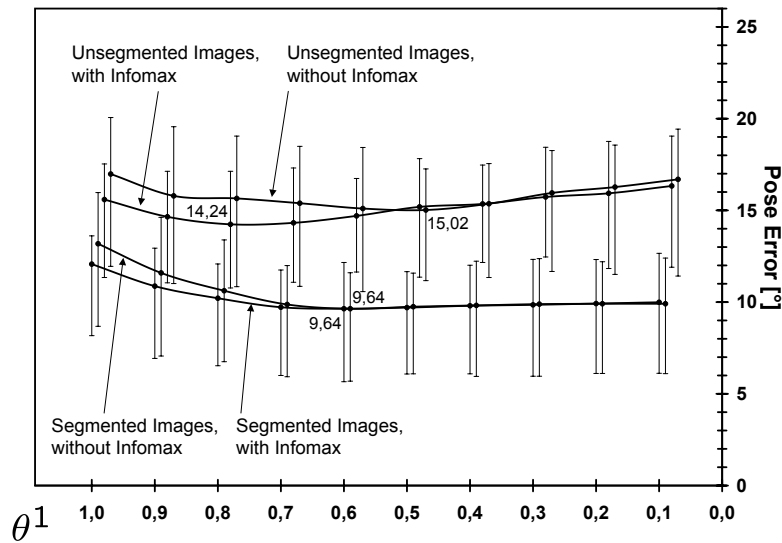


(b)

**Figure 4.3: Recognition of Single Objects (Identity)** — The figure shows the recognition performance with respect to object identity. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. The recognition performance is shown in dependence on the relative weighting of the feature- and correspondence-based parts controlled by  $\theta^1$ . This parameter determined the final number of model candidates that were passed to the correspondence-based verification part. The best results are annotated to the respective data points. The results were better for segmented images. Optimal performance was attained by satisfying the infomax principle and with a well-balanced combination of the feature- and correspondence-based parts.



(a)



(b)

**Figure 4.4: Recognition of Single Objects (Pose)** — The figure shows the recognition performance with respect to object pose. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. Again, the results were better for segmented images and optimal performance, like in the identity case (Fig. 4.3), was attained by satisfying the infomax principle and with a well-balanced combination of the feature- and correspondence-based parts.

### 4.3.2 Recognition of Single Objects using Majority Vote as Verification Method

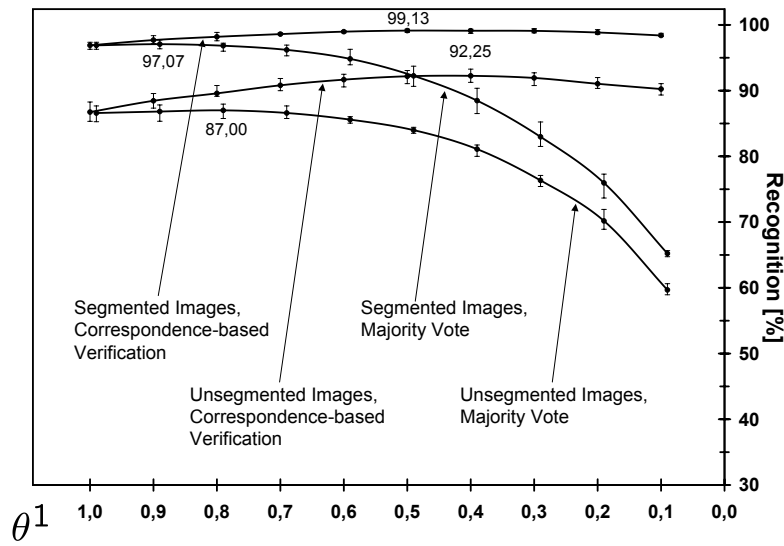
This experiment was supposed to demonstrate the usefulness of correspondence-based verification of model candidates. The recognition rates with respect to object identity achieved with that method were compared to recognition rates attained using an alternative method, in which a majority vote was implemented. In the alternative method, the set of model candidates (Eq. (3.21)) was partitioned into subsets of model candidates with the same object identity after each image presentation. The largest of these subsets was assumed to specify the identity of the object in the input image. The experiment was organized into four test cases per database: in the first two test cases the input images contained a single object, placed in front of a homogeneous and in front of a cluttered background, model candidates were verified with the correspondence-based method. In the remaining test cases, the input images contained a single object, with and without background, but model candidates were verified with the alternative method.

The result of this experiment is given in fig. 4.5. Experimental results achieved with correspondence-based verification were taken from the first experiment (Fig. 4.3). Only in the case of unsegmented input images created from the original ALOI images (Fig. 4.5 (b)), majority vote was slightly better than its correspondence-based counterpart for optimal weighting of the feature- and correspondence-based parts. In all remaining test cases the correspondence-based verification clearly outperformed the alternative method. As backgrounds were randomly created and as the system was more sensitive when confronted with unsegmented ALOI images relative to unsegmented COIL-100 images (Fig. 4.3), the start points of curves differed in the case of unsegmented ALOI images (Fig. 4.5 (b)).

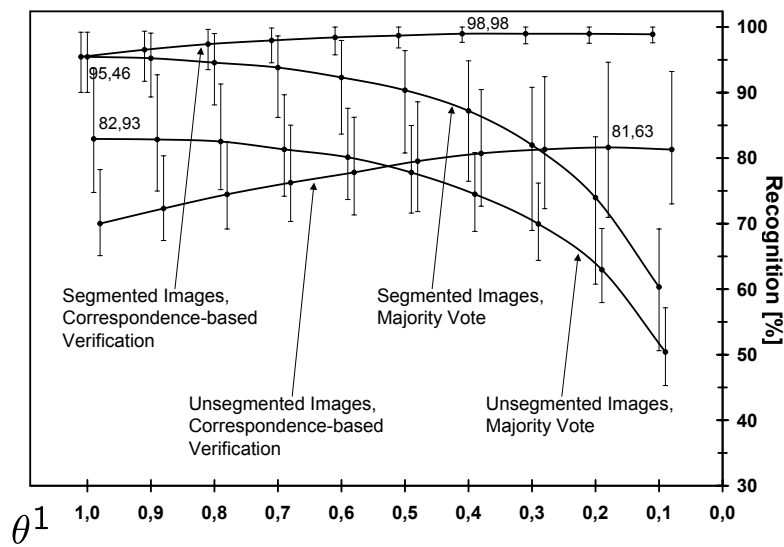
### 4.3.3 Recognition of Scaled Single Objects

This experiment was concerned with the recognition of single scaled objects with respect to object identity. It was organized into four test cases per database. The object in the input images was placed in front of a homogeneous black background and scaled to 100%, 95%, 90%, and 85% of its original size before presentation; fig. 4.6 gives examples.

The result of this experiment is given in fig. 4.7. Recognition performance depended considerably on object size. The recognition performance in the

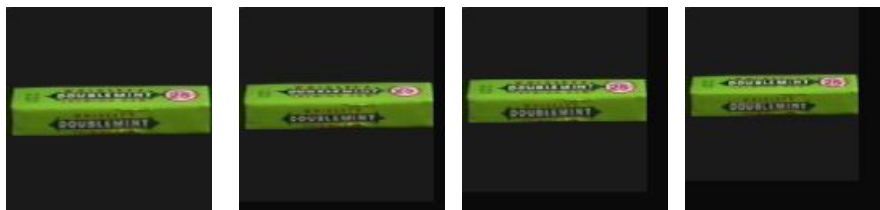


(a)



(b)

**Figure 4.5: Recognition of Single Objects using Majority Vote as Verification Method** — The figure gives the recognition performance with respect to object identity attained with correspondence-based and alternative verification of model candidates, which is a simple majority vote. The results were attained (a) with the COIL-100 and (b) with the ALOI database. Input images contained a single object with and without structured background. Only (b) in the case of unsegmented ALOI images, majority vote performed slightly better than its correspondence-based counterpart for optimal weighting of the feature- and correspondence-based parts. In all remaining test cases the correspondence-based verification clearly outperformed the alternative method.

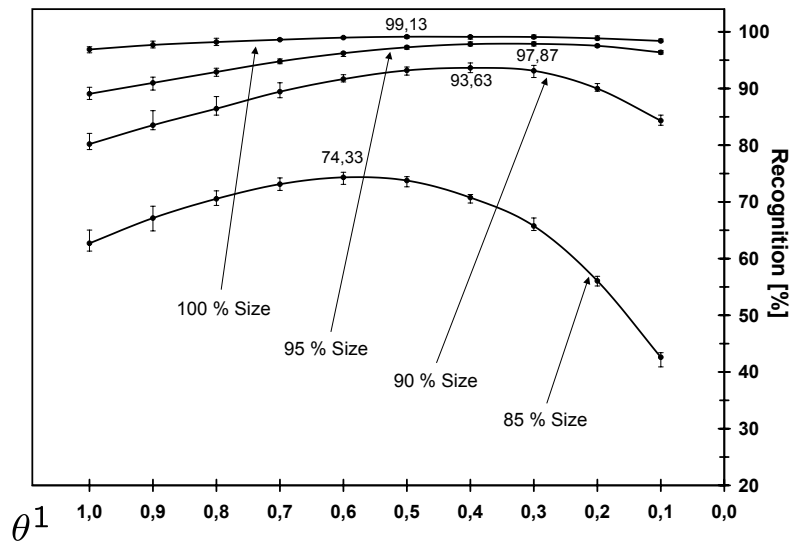


**Figure 4.6: Input Images of Scaled Objects** — *The figure shows example input images of scaled objects. From left to right the object in the input image is scaled to 100%, 95%, 90%, and 85% of its original size.*

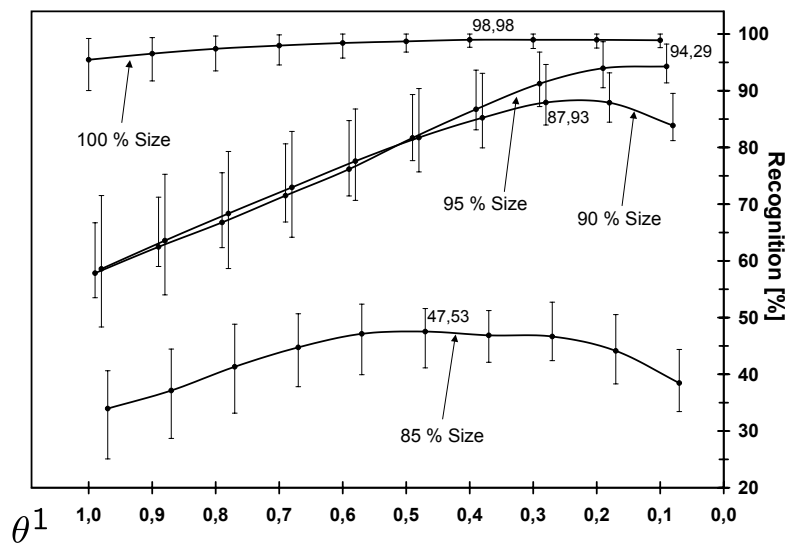
first three test cases, in which the object in the input images was scaled to 100%, 95%, and 90% of its original size, was respectable: for these scale factors recognition rates were above 93% for the COIL-100 and above 87% for the ALOI database. If the object in the test images was scaled to 85% of its original size, test case four, recognition performance dropped considerably. The main reason for this behavior is the design of the feature detectors, which are position- but not size-invariant (Section 3.4.2). Moreover, the parquet graph features were in no way adapted to changes in object size: the horizontal and vertical distances between neighbored nodes (Section 3.2) and the attributed Gabor features remained unchanged. As the objects in the ALOI images covered a much smaller fraction of the image, the system was more sensitive to changes in object size relative to images taken from the COIL-100 database.

#### 4.3.4 Recognition of Single Objects with Sparse Learning Sets

This experiment was concerned with the assessment to what extent the system was able to interpolate between learning examples. To this end, the system was trained on sparse learning sets, i.e., learning sets with fewer training examples per object identity. Input images showed the object in front of a plain black background. The experiment was subdivided into  $N = 9$  test cases per database. For each test case, five pairs of learning/testing sets were created. In a learning set the distance between two consecutive learning images was sampled homogeneously while the initial pose angle was chosen randomly. In the  $n$ -th test case,  $n \in \{1, \dots, N\}$ , two consecutive



(a)



(b)

**Figure 4.7: Recognition of Scaled Single Objects** — The figure shows the recognition performance with respect to object identity in the case of scaled single objects. The results in (a) were attained with the COIL-100, those in (b) with the ALOI database. The experiment was organized into four test cases per database in which the object in the input images was placed in front of a homogeneous background and scaled to 100%, 95%, 90%, and 85% of its original size before presentation. Recognition performance depended considerably on object size. In the first three test cases it was respectable. In the fourth test case the recognition performance dropped considerably.

learning images of an object were  $10 \cdot n$  degrees apart. The learning sets thus comprised 36, 18, 12, 9, 8, 6, 6, 5, and 4 learning images per object identity, respectively. After construction of a learning set, all remaining images of the database were assigned to the respective testing set, i.e., testing sets comprised 3600, 5400, 6000, 6300, 6400, 6600, 6600, 6700, and 6800 images. Thus, smaller learning sets implied larger testing sets.

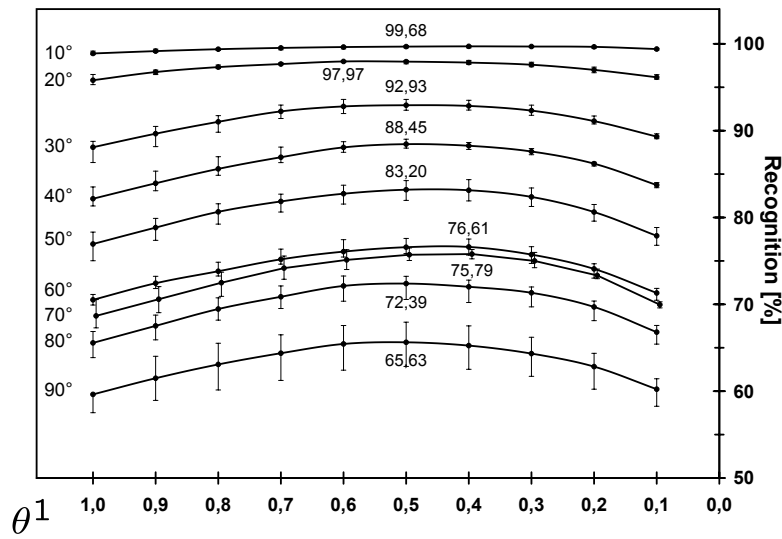
The result of this experiment is given in fig. 4.8. Like in the first experiment, optimal recognition performance was achieved for a well-balanced combination of the feature- and the correspondence-based parts. Recognition performance degraded smoothly with increasing angles between consecutive learning images. The system interpolated well between learning example not too far apart from each other. For instance, if learning images were 30 degrees apart the average recognition rate was still above 90% for both databases. The results achieved in the ninth test case, in which learning images were 90 degrees apart, were still respectable. Due to the regular sampling of object pose, results for the first test case, in which learning images were ten degrees apart, were better relative to the first experiment (Section 4.3.1).

### 4.3.5 Recognition of Single Objects with Sparse Visual Dictionaries

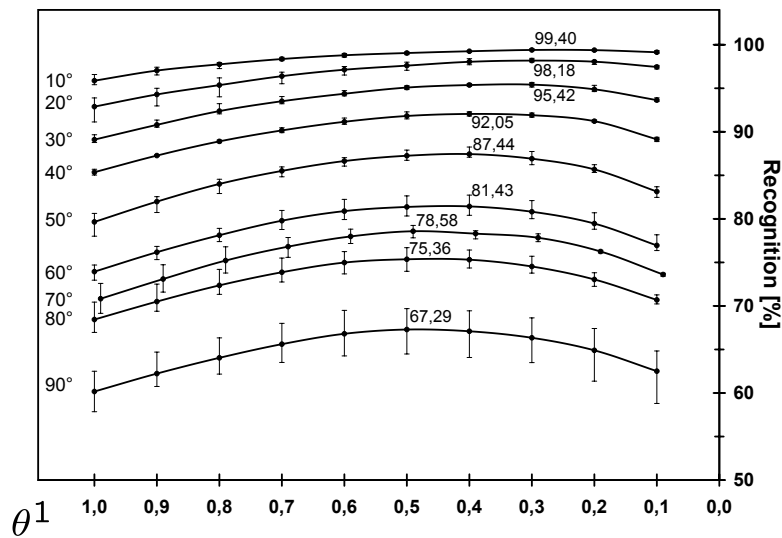
This experiment was concerned with the recognition of single objects in the case of sparse visual dictionaries, i.e., they contained fewer features per learning example. Since, upon image presentation, less model features have to be compared to image features in the case of sparse visual dictionaries the average execution time per recognition could benefit. We analyzed the impact of different degrees of sparseness on the recognition performance. The input images showed the object in front of a plain black background. The experiment was organized in five test cases per database. In each test case the fivefold cross-validation (Section 4.2) was repeated with thinned-out visual dictionaries created from the original ones. The sparse dictionaries comprised a single feature vector with 1, 2, 4, 10, and 20 features per learning example, respectively. Only the respective number of the most informative features were transferred from the original visual dictionary's second feature vector (Section 3.3), the more detailed one, to the new sparse one.

The recognition performance with respect to object identity is given in fig. 4.9. It degraded smoothly with increasing sparseness, i.e., with decreasing numbers of features per learning example. The poorer quality of ALOI relative to





(a)



(b)

**Figure 4.8: Recognition of Single Objects with Sparse Learning Sets** — The figure shows the recognition performance with respect to object identity if the system was trained on sparse learning sets, i.e., learning sets with fewer training examples per object identity. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. The experiment was subdivided into  $N = 9$  test cases. In the  $n$ -th test case,  $n \in \{1, \dots, N\}$ , two consecutive learning images of an object were  $10 \cdot n$  degrees apart. The system interpolated well between learning examples not too far apart from each other. Recognition performance degraded smoothly with increasing angles between two consecutive learning images.

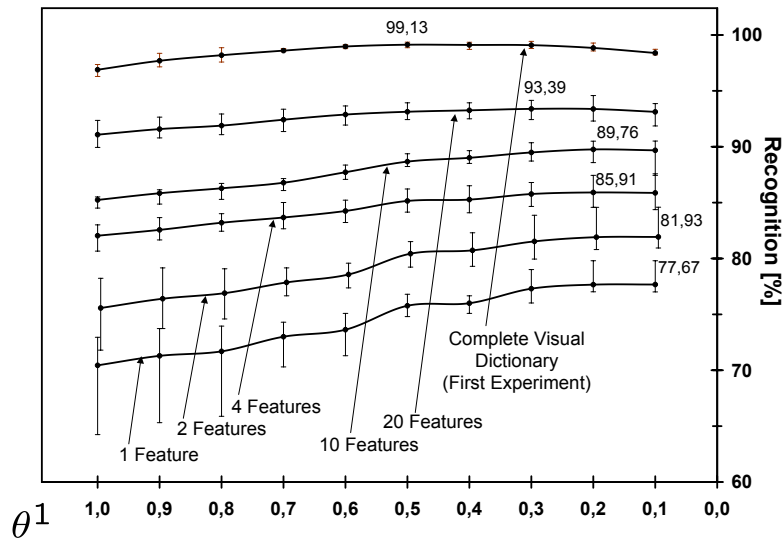
COIL-100 images was especially harmful in this experiment: the mean variation of the results attained with the ALOI database was always considerably larger relative to the results achieved with COIL-100 database.

The average execution times are given in fig. 4.10. For the sake of comparability, the average execution time measured in the first experiment (Section 4.3.1) is given here as well. For both databases the average execution times were almost independent of the relative weighting of the feature- and correspondence-based parts controlled by the relative threshold  $\theta^1$ . They scaled proportionally with the number of features per learning example in the visual dictionaries. The average execution time measured in the first experiment was below that measured in the fifth test case, 20 features per learning example, since in the first experiment the accelerated feature search (Section 3.4.8) was able to fully exploit all of its three sources of efficiency as the visual dictionary comprised two feature vectors. In contrast, in the current experiment visual dictionaries contained only a single feature vector which was traversed in a linear fashion (Algorithm 3, Fig. 3.18) in search of matching model features for the given image features.

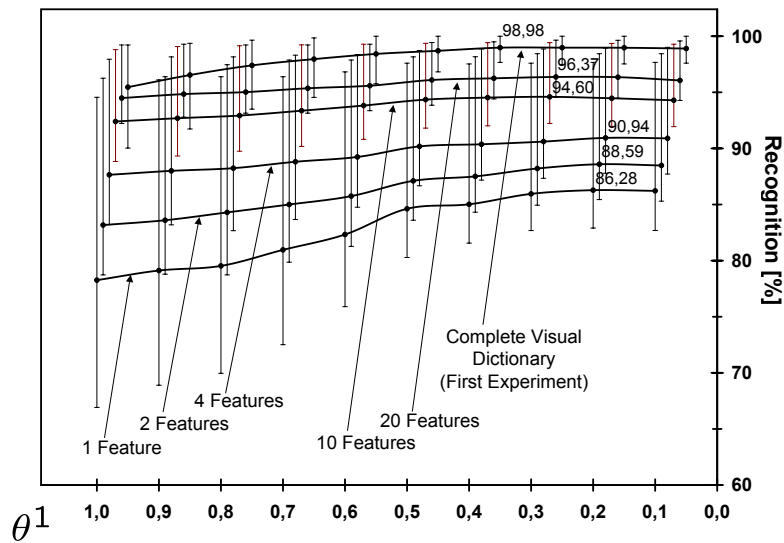
### 4.3.6 Recognition of Multiple Objects

This experiment was concerned with the recognition of multiple, simultaneously presented, non-overlapping 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  $N \in \{2, 3, 4\}$  objects placed in front of a plain black background while in the last three test cases cluttered background was manually added. The procedure of background construction was the same as in the first experiment. Fig. 4.11 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. In a test case 1400 input images were presented. The system returned the  $N$  most similar models. Each coincidence with one of the presented objects was accounted as a successful recognition response. The average recognition rates were calculated over all responses.

The result of this experiment is given in figs. 4.12 and 4.13. We learn that, compared to the single-object experiments (Section 4.3.1), the point of optimal recognition performance considerably moved to the right: putting more

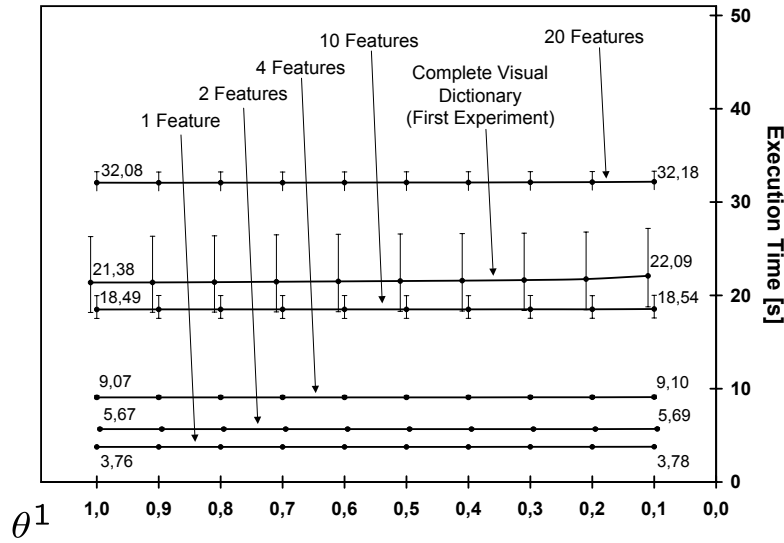


(a)

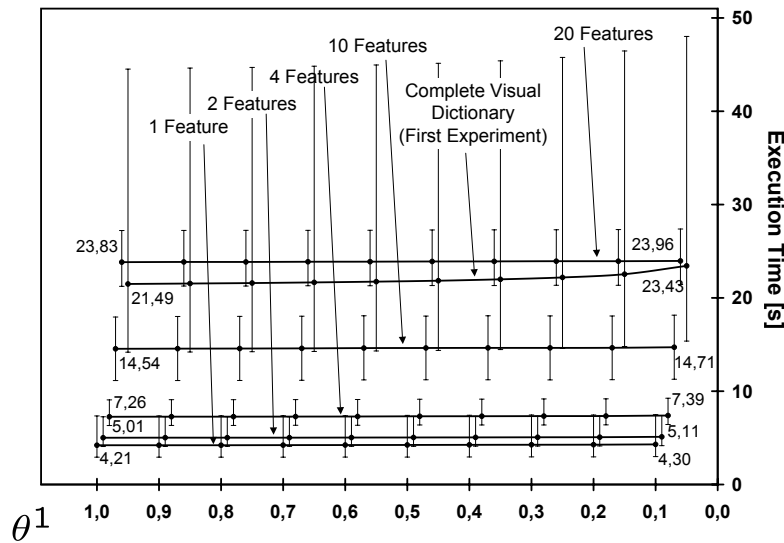


(b)

**Figure 4.9: Recognition of Single Objects with Sparse Visual Dictionaries (Identity)** — The figure shows the recognition performance with respect to object identity in the case of sparse visual dictionaries. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. The sparse visual dictionaries comprised only one feature vector with 1, 2, 4, 10, and 20 features per learning image, respectively. For the sake of comparability, the result of the first experiment (Section 4.3.1) is repeated here. The recognition performance degraded smoothly with increasing sparseness, i.e., with decreasing numbers of features per learning example.



(a)



(b)

**Figure 4.10: Recognition of Single Objects with Sparse Visual Dictionaries (Execution Time)** — The figure shows the recognition performance with respect to execution time in the case of sparse visual dictionaries. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. The sparse visual dictionaries comprised only one feature vector with 1, 2, 4, 10, and 20 features per learning image, respectively. The average execution time measured in the first experiment (Section 4.3.1) is given here as well. For both databases average execution times were almost independent of the relative weighting of the feature- and correspondence-based parts. They scaled proportionally with the number of features per learning example in the visual dictionaries.



(a)

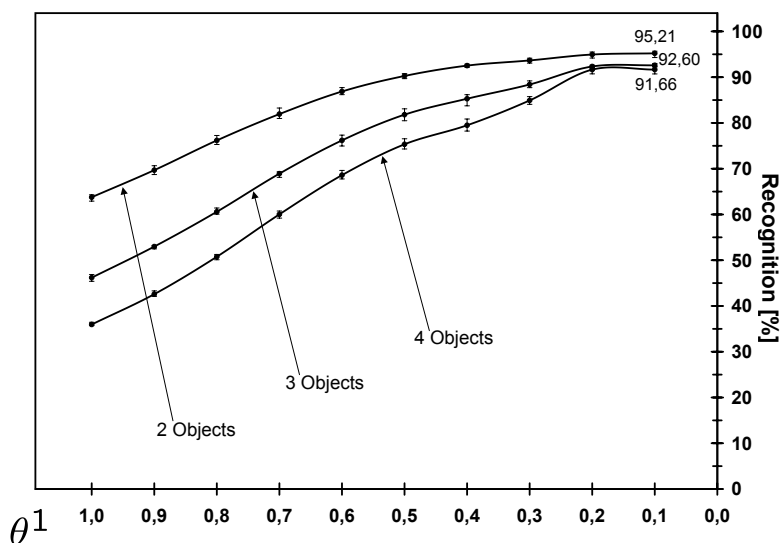


(b)

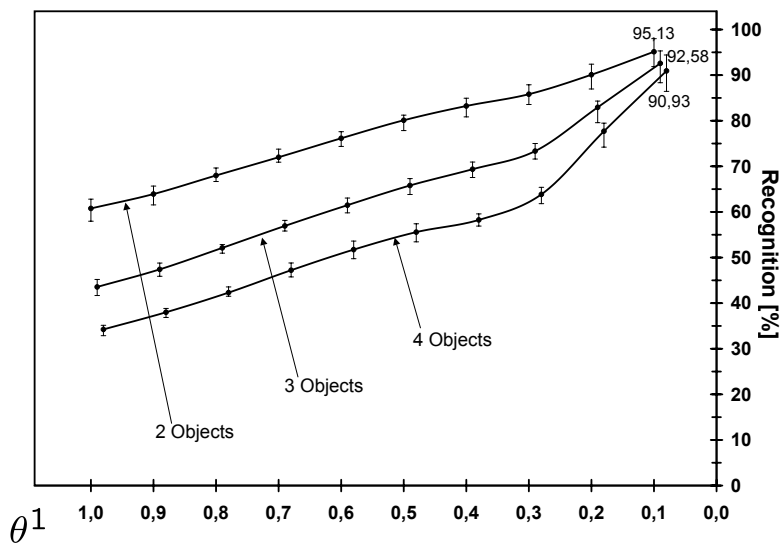
**Figure 4.11: Input Images of Multiple Objects** — *The figure shows an example of (a) a segmented and (b) an unsegmented input image containing four objects drawn from the COIL-100 database. Backgrounds were constructed in the same fashion as in the first experiment.*

---

emphasis on the correspondence-based verification part thus improved recognition performance in this experiment. This finding can be explained with the assumption that a solution of the binding problem (VON DER MALSBERG, 1981, 1999) is required in the case of multiple objects. More specifically, model graphs provide a means to purposefully navigate in the image, which seems to be crucial in the case of input images with 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 further been improved by putting more emphasis on the correspondence-based part by choosing  $\theta^1 < 0.1$ . For performance reasons this was, however, not carried out.



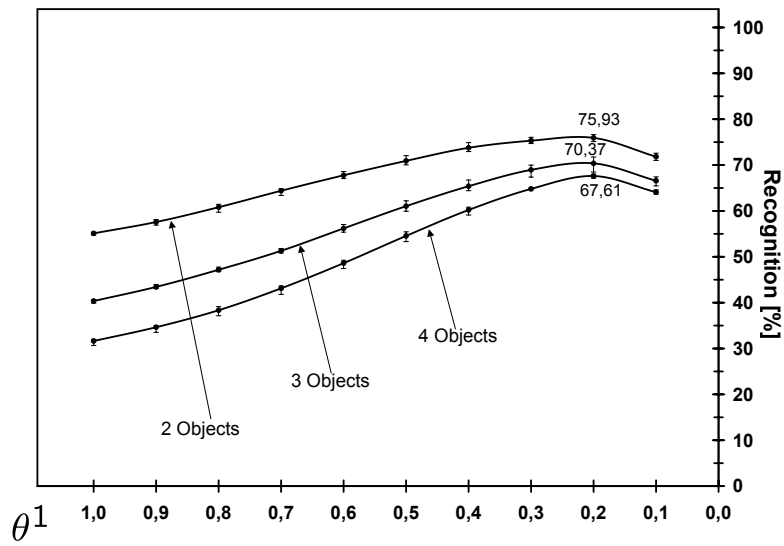
(a)



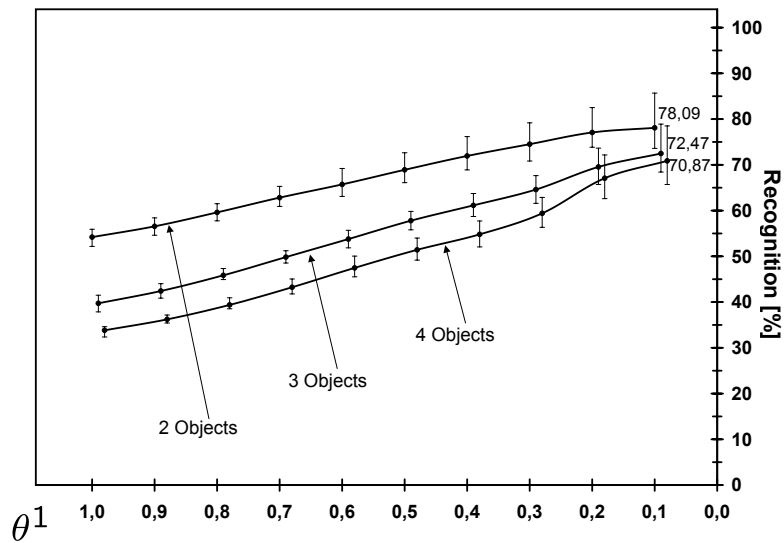
(b)

### Figure 4.12: Recognition of Multiple Objects (Segmented Images)

— The figure shows the recognition performance with respect to object identity in the case of multiple non-overlapping objects where the objects in the input images were placed in front of a plain black background. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. Compared to the first experiment (Section 4.3.1), the point of optimal recognition performance considerably moved 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 4.13: Recognition of Multiple Objects (Unsegmented Images)** — The figure shows the recognition performance with respect to object identity in the case of multiple non-overlapping objects where the objects in the input images were placed in front of a structured background. The results in (a) were attained with the COIL-100, those in (b) were attained with the ALOI database. Like in the case of segmented input images (Fig. 4.12), recognition performance degraded smoothly with the number of simultaneously presented objects.

### 4.3.7 Recognition of Partially Occluded Objects

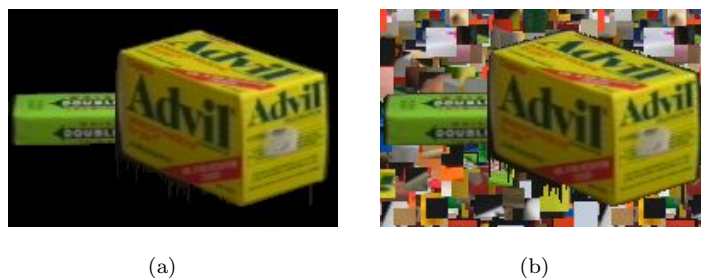
While in the previous experiment (Section 4.3.6) the objects were presented in a non-overlapping manner, this final object recognition experiment was concerned with the recognition of partially occluded objects. Only the recognition performance with respect to object identity was evaluated. The experiment was organized into twelve test cases per database. In the first six test cases we simultaneously presented two objects where 0-50% 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, each object appeared at least once as occluded. In a test case 1400 images were presented. In the remaining six 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. Fig. 4.14 shows example input images of a partially occluded object with and without added background.

The result of this experiment is given in figs. 4.15 and 4.16. In fig. 4.15 the objects in the input images were placed in front of a plain black background while the result given in fig. 4.16 was attained with unsegmented images. Like in the previous experiment, emphasis of the correspondence-based part improved recognition performance: solution of the binding problem is also demanded in the case of partially occluded objects as well. 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 non-overlapping objects. Like in the experiment with multiple objects, one can expect that recognition rates could have further been improved by putting more emphasis on the correspondence-based part by choosing  $\theta^1 < 0.1$ .

## 4.4 Discussion

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. Throughout, a well-balanced combination of the feature-



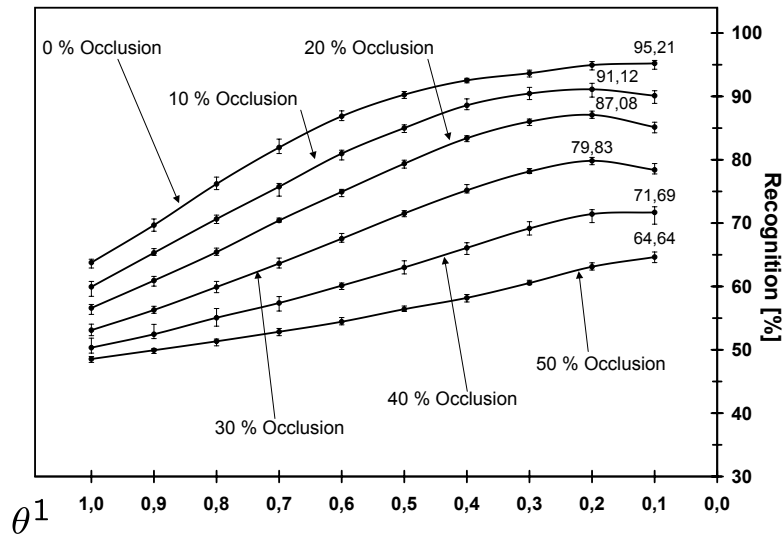


**Figure 4.14: Input Images of a Partially Occluded Object** — *The figure shows (a) a segmented and (b) an unsegmented input image of a partially occluded object. In this example, the occluding object covers about fifty percent of the occluded object.*

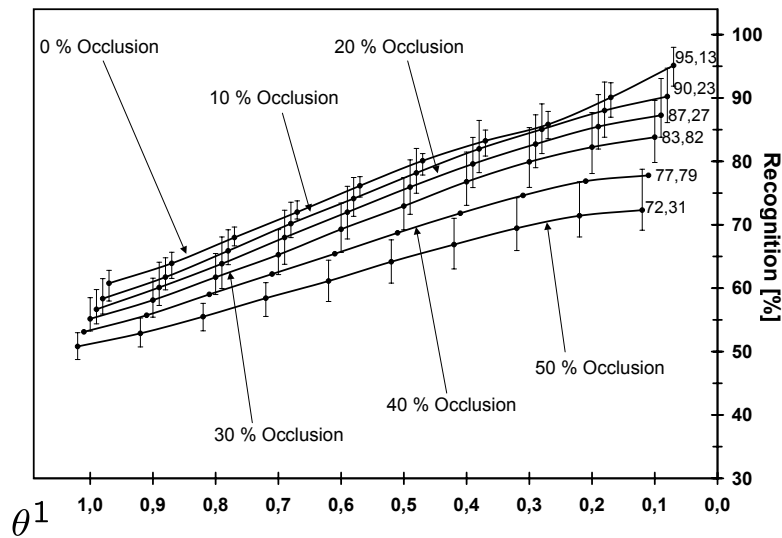
---

based and correspondence-based parts produced optimal results in terms of recognition rate and pose error. Unlike many other methods, the presented technique was able to cope with varying background, multiple objects, and partial occlusion. In all test cases the system's performance degraded smoothly with the increasing complexity of the recognition tasks and with increasing sparseness of learning sets and visual dictionaries. The system turned out to be quite sensitive to changes in object size. To this end the feature detectors presented in section 3.4.2 should be made position- and size-invariant, i.e., they should be realized as logical disjunctions of position- and size-variant feature detectors. In a qualitative sense the results attained with the COIL-100 database are comparable with 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.

Our system performed favorably compared with other techniques. The original system of MURASE and NAYAR (1995), that 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 unscaled objects drawn from the COIL-100 database. Our system attained a recognition rate of 99.13% in the same test case (Section 4.3.1). The recognition performance of the MURASE and NAYAR system is, however, unclear if it would be confronted with more sophisticated recognition task, for instance,

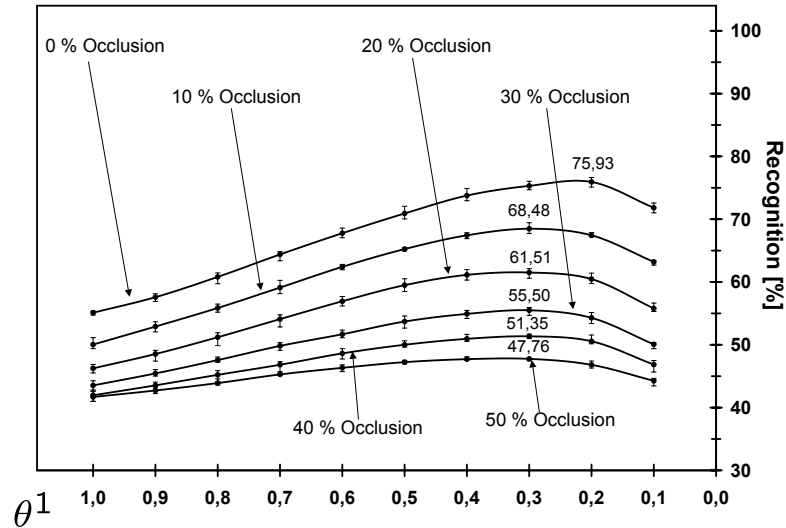


(a)

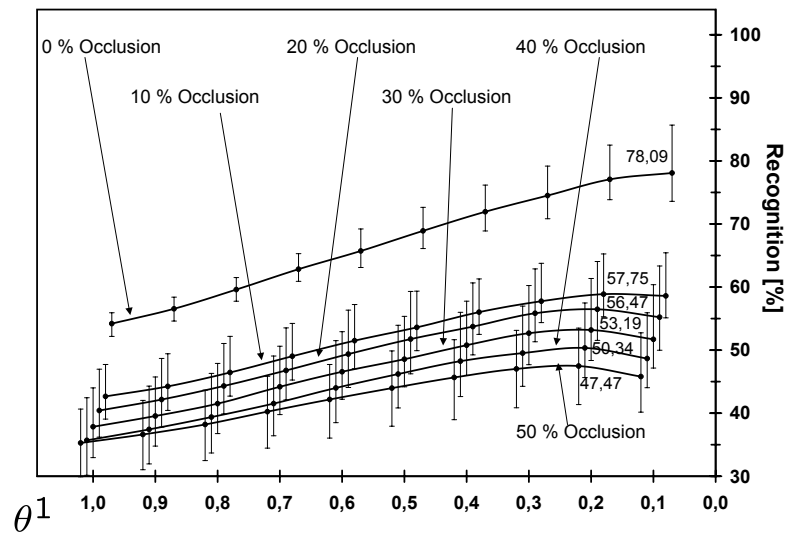


(b)

**Figure 4.15: Recognition of Partially Occluded Objects (Segmented Images)** — The figure shows the recognition performance with respect to object identity in the case of partially occluded objects that were placed in front of a plain black background. The results in (a) were attained with the COIL-100 database, those in (b) were attained with the ALOI database. Like in the case of multiple objects (Section 4.3.6), 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 4.16: Recognition of Partially Occluded Objects (Unsegmented Images)** — The figure shows the recognition performance with respect to object identity in the case of partially occluded objects that were placed in front of a structured background. The results in (a) were attained with the COIL-100 database, those in (b) were attained with the ALOI database. Like in the case of segmented input images (Fig. 4.15), recognition performance degraded smoothly with the amount of occlusion.

images with structured backgrounds, with multiple objects, or with occluded objects.

In (WERSING and KÖRNER, 2003) the performance of the MURASE and NAYAR system is compared with their method of setting up the feature extraction layers in an evolutionary fashion. The authors conducted their experiments on the COIL-100 database. In the case of segmented images and on sparse learning sets their system and ours performed about equally well, see fig.4 (b) and tab.1 in (WERSING and KÖRNER, 2003) and figs. 4.4 (a) and 4.8 (a). Provided that the learning sets contained enough training views per object, both systems achieved recognition rates above 99%. The system of WERSING and KÖRNER performed, however, better than ours if the number of training views was reduced below 18 views per object. For instance, their system achieved recognition rates of 79.0% and approximately 95% if the learning sets contained four and twelve training views per object, respectively, while, in the corresponding test cases, our system peaked at 65.63% and 92.93%. WERSING and KÖRNER report that the system of MURASE and NAYAR (1995) achieved a recognition rate of 77.0% in the case of four training views per object.

The system of WERSING and KÖRNER (2003) performed slightly better in the case of scaled objects, see fig. 4 (c) in (WERSING and KÖRNER, 2003) and fig. 4.7. A systematic comparison, however, cannot be carried out since in their experiment images of objects were randomly scaled about +/- 10% before presentation while in our experiment the scale factors were constant. The system of WERSING and KÖRNER achieved recognition rates between approximately 95% and 98% for 36 training examples per object, depending on the mode of feature selection. Our system attained recognition rates of 93.63% and 97.87% if the objects were scaled to 90% and 95% of their original size, respectively.

In the case of unsegmented images our system outperformed the system of WERSING and KÖRNER, see fig. 6 (a) in (WERSING and KÖRNER, 2003) and fig. 4.4 (a): with 36 training views per object identity our system attained a recognition rate of 92.25% while the system of 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.

As an intermediate result our system produces model graphs, which are the closest possible representations of a presented object in terms of memorized features. A variety of further processing can build on these graphs. The simple graph matching in the correspondence-based verification part can be replaced by the more sophisticated methods of LADES et al. (1993); WISKOTT et al. (1997); WÜRTZ (1997); TEWES (2006), which should lead to an increased robustness under shape and pose variations.



# Chapter 5

## Object Categorization

*Success is the ability to go from one failure to another with no loss of enthusiasm.*

Sir Winston Churchill

In this chapter we apply the graph dynamics to the task of categorizing objects. According to PALMERI and GAUTHIER (2004) *categorization* refers to a decision about an object's kind. Categorization thus requires generalization across members of a class of objects with different shapes. Especially, generalization over object identity is required. Categorization of objects is more difficult than recognition of objects, since in addition to the multitude of possible object variations such as translation, scale, pose, occlusion, illumination, noise and so forth, intra-category variations among the captured objects must be accounted for. These can be considerable. For instance, human faces can have glasses, beards, different expressions, different age, gender, or face form.

We conducted experiments on the ETH-80 database (LEIBE and SCHIELE, 2003), which contains images of objects from eight categories. The task is to categorize unknown objects into these categories.

## 5.1 Review of Literature

Computational models for object categorization, like models for visual recognition of objects (Chapter 4), can be subdivided into feature-based and correspondence-based approaches. These are afflicted with the same conceptual advantages and disadvantages.

In feature-based models, invariance to position, size and so on is achieved in the same fashion as in feature-based models of object recognition (Chapter 4). Categorization is achieved by comparing activated master units to stored lists of activated master units for objects with known memberships to predefined categories; recognition thus precedes categorization. Examples of feature-based categorization systems include (SCHNEIDERMAN and KANADE, 2000; ULLMAN and SALI, 2000; LEIBE and SCHIELE, 2003; VIOLA and JONES, 2001, 2004).

In correspondence-based models of object categorization, categories are represented as ordered arrays of local features and categorization is performed by solving the correspondence problem. In (WISKOTT et al., 1997) a model for the category of human faces in frontal pose, a so-called *bunch graph*, is proposed. That method has been outlined in section 2.3. This approach focuses on the integration of intra-category variations in the object model but ignores global object variations such as changes in pose, illumination and so on. WEBER et al. (2000) propose a system that is able to categorize objects in a probabilistic framework. A summary of that method has been given in the introduction of chapter 3. FEI-FEI et al. (2003) propose a probabilistic method, similar to that of WEBER et al. (2000), which is able to categorize objects from few learning examples.

## 5.2 Experimental Setting

The ETH-80 database (LEIBE and SCHIELE, 2003) contains images of eight categories namely apples, pears, tomatoes, dogs, horses, cows, cups, and cars of ten identities per category and 41 images in different poses per identity. The whole collection consists of 3280 images.

Experimental results were attained with leave-one-object-out cross-validation (WITTEN and FRANK, 2000). This means that the system was trained with the images of 79 objects and tested with the images of one unknown object. We thus created 80 pairs of disjoint learning and testing sets. The learning



sets contained 3239, the testing sets 41 images. From each learning set a visual dictionary with two feature vectors was calculated. We used the default parameter set given in section 3.6.

## 5.3 Experiments

An interesting question related to object categorization is whether category information imposed on the learning sets can be harnessed to improve categorization performance. We conducted two experiments: First, we evaluated categorization performance if the decision about the final category relied on a given hierarchical organization of predefined categories. Second, we evaluated categorization performance without hierarchical organization of categories.

### 5.3.1 Categorization Using Hierarchically Organized Categories

In the first experiment we hierarchically organized the images into categories of  $K = 3$  partitionings as given in fig. 3.5. The relative thresholds  $\theta^k$  for selection of salient categories of partitionings  $\Pi^k$ ,  $k \in \{1, 2, 3\}$ , were all set to 0.4 (Eq. (3.20)). For partitionings  $\Pi^1$  and  $\Pi^2$  we considered an object to be correctly categorized if exactly one category out of these was selected as salient and the presented object belonged to that category. For partitioning  $\Pi^3$  a set of model candidates was calculated by intersection of salient categories (Eq. (3.21)). The model candidates of that set were passed to the correspondence-based verification part. We considered the presented object to be finally correctly categorized if it belonged to the same of the original eight categories as the object in the model image.

Fig. 5.1 displays the averaged categorization rates computed with the leave-one-object-out cross-validation for each of the original eight categories of apples, pears, tomatoes, dogs, horses, cows, cups, and cars. Each data point was averaged over  $10 \cdot 41 = 410$  single measurements. Three types of curves can be observed. First, for the fruit categories the categorization curves have a clear minimum at partitioning  $\Pi^2$ . The system perfectly categorized input images of fruits into the categories of partitioning  $\Pi^1$  (natural and man-made) but experienced difficulties with the categories of partitioning  $\Pi^2$  (fruit, animal, cup, car), especially for pears. Since the intra-category variations of fruits were well-sampled, the correspondence-based verification

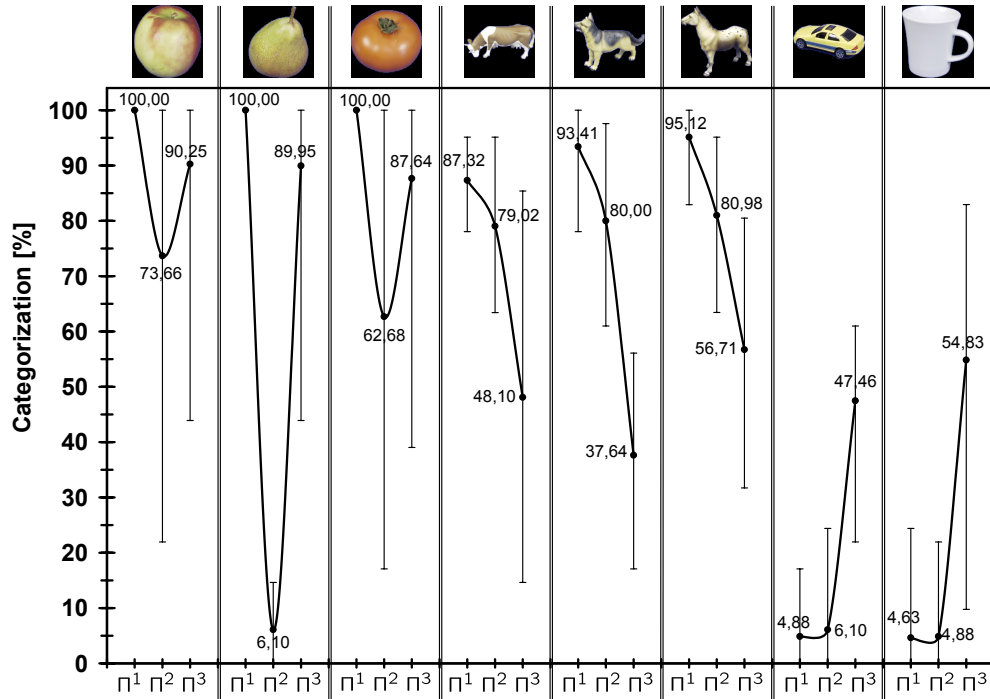
of model candidates was able to compensate for this shortcoming such that input images of fruits were well categorized into the categories of partitioning  $\Pi^3$  (apple, pear, tomato, dog, horse, cow, cup, car). Second, for the animal categories the categorization curves are strictly monotonically decreasing. The system fairly categorized input images of animals into the categories of partitionings  $\Pi^1$  and  $\Pi^2$ , albeit categorization performance for partitioning  $\Pi^2$  always fell short relative to partitioning  $\Pi^1$ , but experienced extreme difficulties for the categories of partitioning  $\Pi^3$ . This behavior is founded in the poor sampling of the animal categories; the learning data is much too sparse to make the fine distinctions between the categories of partitioning  $\Pi^3$ . Third, for the categories of cars and cups the categorization curves are monotonically increasing. Due to the imbalance between natural and man-made objects in the database, six categories of natural vs. two categories of man-made objects, the system failed to unambiguously assign input images of cars and cups to the correct categories of partitionings  $\Pi^1$  and  $\Pi^2$ . Again, the correspondence-based verification of model candidates was to some extent able to compensate for this shortcoming as the considerable increases in categorization rates for the categories of partitioning  $\Pi^3$  demonstrate.

Generally, categorization performance depended considerably on the sampling of categories. In this respect the system was able to categorize apples, pears, and tomatoes well but experienced difficulties with cows, dogs, horses, cars, and cups; in the latter cases the intra-category variations among category members are too large. It is thus reasonable to assume that categorization performance can be improved by adding more learning examples to those categories.

### 5.3.2 Categorization Using Single-Element Categories

For evaluation of the system's performance without predefined hierarchical organization of categories we arranged the learning set into  $K = 1$  partitioning of single-element categories. We considered the object in the input image to be correctly categorized if it belonged to the same category as the object in the model image. The attained results depending on  $\theta^1$  are given in fig. 5.2. For clarity, the curves are spread over two subfigures. All other parameters were the same as in the previous experiment.

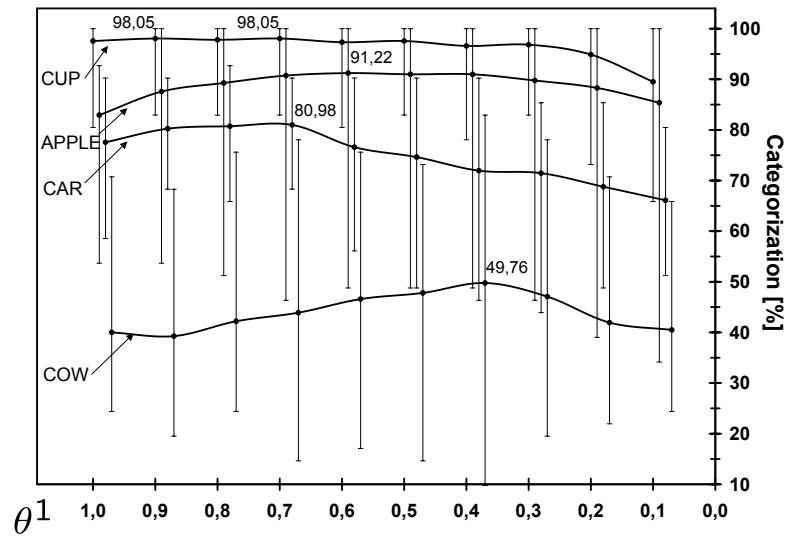
As in the object recognition experiments (Section 4.3), a well-balanced combination of the feature- and the correspondence-based parts allowed for optimal categorization performance. The expectation that categorization perfor-



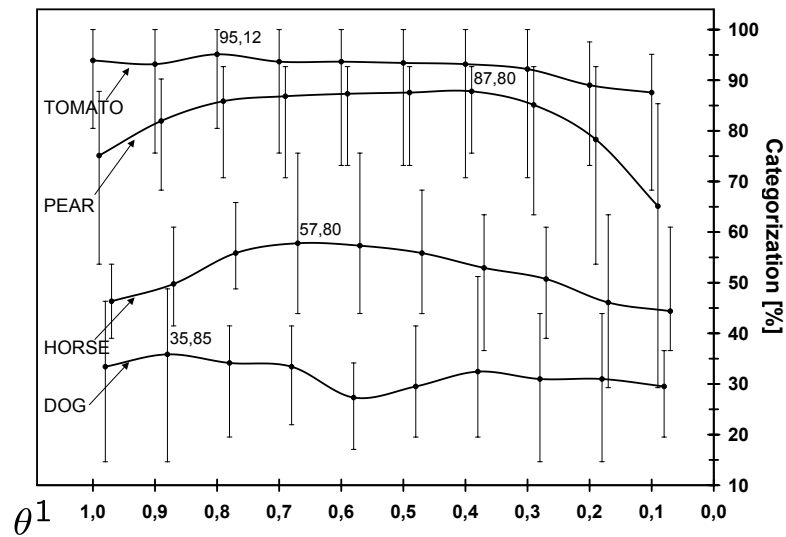
**Figure 5.1: Categorization of General Objects Using Hierarchically Organized Categories** — *The averaged categorization rates computed within the leave-one-object-out cross-validation are displayed. Each data point was averaged over 410 single measurements. Categorization performance depended considerably on the sampling of categories. The feature-based part's ability to unambiguously assign the object in the input image to the categories of partitionings  $\Pi^1$  and  $\Pi^2$  is obviously limited. For most cases, the correspondence-based verification part was able to compensate for this shortcoming, but not for the shortage of learning examples, especially in the animal categories.*

mance would benefit from hierarchical organization of categories could not be substantiated. In the case of apples, tomatoes, cows, horses, cars, and cups average categorization performance was considerably better without the hierarchy. Only for pears and dogs categorization could slightly benefit. The optimal weightings of the feature- and correspondence-based parts turned out to be category-specific. The attained categorization rates are below or close to those of LEIBE and SCHIELE (2003). Their object categorization system, however, integrates color, texture, and shape features while our system only relies on local texture information. At least the feature-based part of the technique described in this paper can work with any convenient feature type (WESTPHAL and WÜRTZ, 2004). One can thus expect to further improve categorization performance if more feature types become incorporated.

Fig. 5.3 gives a confusion matrix of the categorization performance in the case of single-element categories and optimal weightings of the feature- and correspondence-based parts. The optimal weightings were category-specific (Fig. 5.2). Categorization performance depended considerably on the degree of intra-category variations: for categories with relatively small intra-category variations, for instance, fruits, cups, and cars, the system performed well while the system's performance degraded in a remarkable fashion when confronted with images of categories with larger variations among category members. This is especially prominent for the animal categories. The system performed particularly poorly for the category of dogs. However, in 75.12% (10.00%+29.27%+35.85%) of all cases the system assigned an input image of a dog to the category of animals vs. 80.00% in the hierarchical case (Fig. 5.1). Images of horses and cows were assigned to that category in 84.87% and 86.10% of all cases in the non-hierarchical case vs. 80.98% and 79.02% in the hierarchical case, respectively. In sum, 82.03% of all cases input images of animals were correctly assigned to the category of animals in the non-hierarchical case while that number was 80.00% = (79.02% + 80.00% + 80.98%) / 3 with hierarchical organization of categories. These results once more confirm our original statement that the data is much too sparse to make the fine distinctions between the categories of partitioning  $\Pi^3$ .

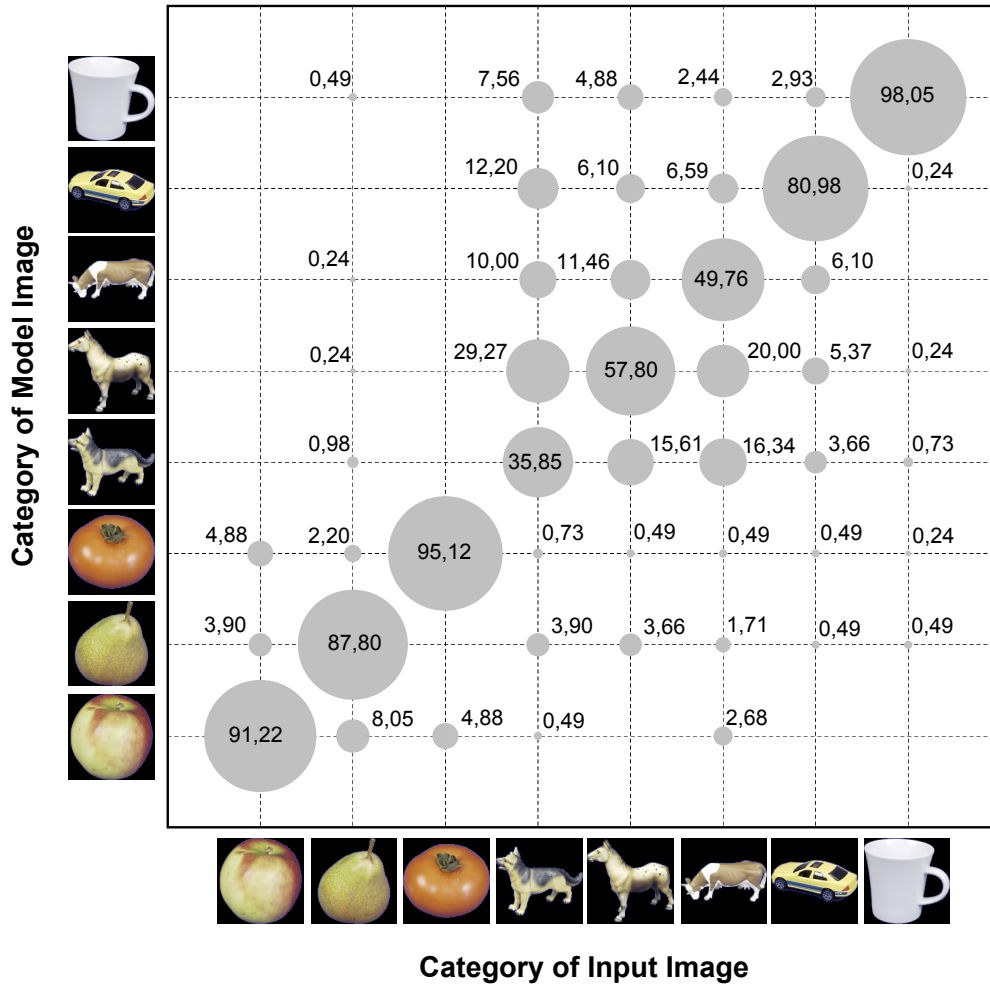


(a)



(b)

**Figure 5.2: Categorization of General Objects Using Single-Element Categories** — The averaged categorization rates in the case of single-element categories attained in the leave-one-object-out cross-validation are displayed. For clarity, the curves are spread over two subfigures. Each data point was averaged over 410 single measurements. Optimal categorization performance was achieved for well-balanced combinations of the feature- and correspondence-based parts. In most cases categorization performance was clearly better relative to the hierarchical case. The optimal weightings of the feature- and correspondence-based parts turned out to be category-specific.



**Figure 5.3: Confusion Matrix of Categorization Performance** — A confusion matrix of the categorization performance in the case of single-element categories and optimal weightings of the feature- and correspondence-based parts is given. The optimal weightings were category-specific (Fig. 5.2). The axes are labeled with the categories of the ETH-80 database (LEIBE and SCHIELE, 2003), symbolized by images of arbitrarily chosen representants. The horizontal axis codes the categories of the object in the input images while the vertical axis codes the categories of the object in the model images. The given categorization rates are relative to the categories of the object in the input images. In each column they sum up to 100%. In order to improve readability, blobs were assigned to the categorization rates whose surface areas scale proportionally with the amount of their associated categorization rates.

## 5.4 Discussion

Much work remains to be done on the categorization capabilities. In our experiment we have seen that the categories employed by human cognition were not helpful to improve the categorization capability when employed to structure the recognition process. This finding is, however, compatible with experimental results, which find that in human perception recognition of a single object instance precedes categorization (PALMERI and GAUTHIER, 2004).

Another reason for the relatively poor performance is that in some cases the data was much too sparse to really cover the intra-category variations: if the variations across category members were poorly sampled, categorization failed frequently for input images supposed to be assigned to these categories. For instance, the system performed poorly for the animal categories, but categorized input images of fruits well. Categorization can always be improved by using additional cues like color and global shape. This hypothesis is substantiated by the experimental results of LEIBE and SCHIELE (2003). This would, however, also require larger databases, because much more feature combinations would need to be tested. Nevertheless, the method presented here is well suited to accommodate hierarchical categories. Their impact on categorization quality as well as methods to learn the proper organization of categories from image data are subject to future studies.

As model graphs only represent a single object view they cannot possibly cover larger spectra of individual variations among category members. In this respect bunch graphs provide a more promising concept. As briefly mentioned in section 3.5, the graph dynamics is able to construct bunch graphs provided that the model features stem from carefully chosen model candidates. It is reasonable to assume that categorization performance can further be improved by using bunch graphs instead of model graphs.





## Chapter 6

# Estimation of Pose and Illumination of Human Faces

*Human beings, for all their pretensions, have a remarkable propensity for lending themselves to classification somewhere within neatly labelled categories. Even the outrageous exceptions may be classified as outrageous exceptions.*

W.J. Reichmann

In this chapter we apply the proposed form of graph dynamics to the task of estimating head pose and illumination type of human faces. These are so-called *extrinsic* object parameters. The estimation task is set up in terms of a categorization task in which faces of unknown individuals are subject to be assigned to predefined categories according to head pose and illumination type, i.e., generalization over identity is required. Our experiments were conducted on the CMU-PIE database (SIM et al., 2002, 2003). Even though the proposed method does not allow for estimation of fine pose and illumination changes, it may serve for the purposeful initialization of more sophisticated but slow estimation techniques or of face recognition applications.



**Figure 6.1: Preprocessed Images of Human Heads with Variation in Pose and Illumination** — *The preprocessed images of one individual with the chosen variations in pose and illumination are given. The unprocessed images were taken from the PIE database (SIM et al., 2002, 2003). We selected 15 example images per individual: five head poses (lines) and three illuminations per pose (columns). The original images have been preprocessed in two steps. First, as, unfortunately, the backgrounds of the unprocessed images point to the actual head poses, backgrounds were manually replaced by homogeneous black ones. Second, the square region in the original image where the head is located was automatically selected and scaled to  $128 \times 128$  pixels. The annotated identifiers of pose ( $P$ ) and illumination ( $I$ ) are the same as in (SIM et al., 2002).*

## 6.1 Experimental Setting

The PIE database (SIM et al., 2002, 2003) provides images of human faces in varying poses, illuminations, and facial expressions. For our experiments we decided for the subcollection of faces that are illuminated by flashes only. It contains images of 68 individuals. We selected 15 images per individual: five head poses, three illuminations per pose, all with neutral facial expression.

We selected poses 34 and 22 (full profile left/right), 11 and 37 (half-profile left/right), 27 (frontal pose) and illuminations 17 (illumination from the left), 02 (illumination from the right), and 11 (frontal illumination). The original images have been preprocessed in two steps. Unfortunately, the backgrounds of the original PIE images point to the actual head poses. Therefore, backgrounds were manually replaced by homogeneous black ones. This made the categorization task much more difficult, as will be demonstrated in one of the forthcoming experiments. Second, the square region in the original image where the head is located was automatically selected and scaled to  $128 \times 128$  pixels. This failed for six individuals. Preprocessing of the whole image set thus resulted in a collection of  $(68 - 6) \cdot 15 = 930$  images. Fig. 6.1 gives the preprocessed images of one individual with all variations in pose and illumination.

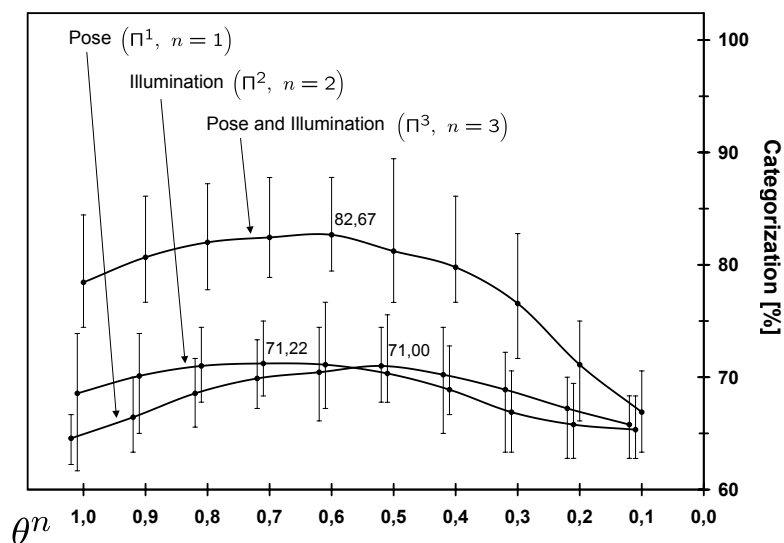
Experimental results were, like the results of the object recognition experiments (Section 4.3), achieved with fivefold cross-validation (WITTEN and FRANK, 2000). We thus created five pairs of disjoint learning and testing sets. The learning sets comprised preprocessed images of 48, the testing sets of 12 individuals with all variations in pose and illumination, thus 720 and 180 images in total, respectively. From each learning set a visual dictionary with two feature vectors was calculated. We used the default parameter set given in section 3.6.

## 6.2 Experiments

Like in the object categorization experiments (Section 5.3), we conducted two experiments. In the first experiment the learning sets were partitioned into predefined categories while in the second experiment no categories were defined beforehand.

### 6.2.1 Estimation of Pose and Illumination Type Using Predefined Categories

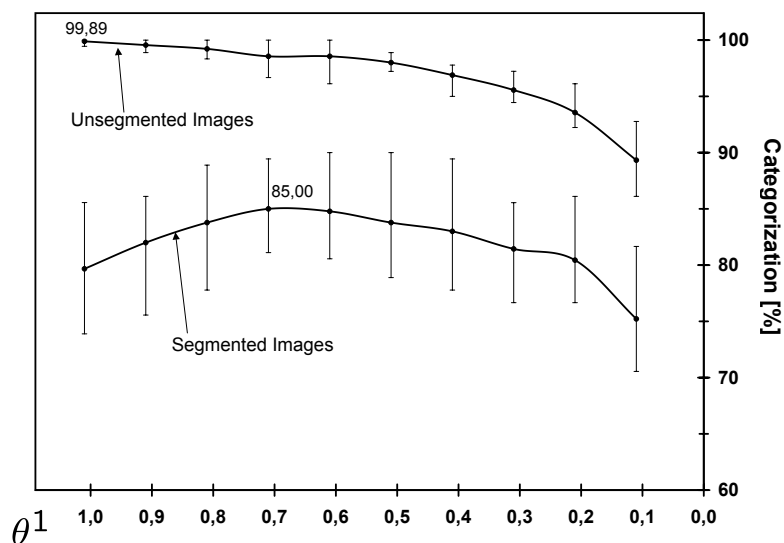
In the first experiment we created  $K = 3$  partitionings of the learning sets: Partitioning  $\Pi^1$  consisted of  $C^1 = 5$  categories, one category for each of the five head poses. Partitioning  $\Pi^2$  consisted of  $C^2 = 3$  categories, one category for each illumination type. Finally, partitioning  $\Pi^3$  consisted of  $C^3 = 15$  categories, one category for each combination of head pose and illumination type. We organized the experiment into  $N = 3$  test cases: in the  $n$ -th test case,  $n \in \{1, 2, 3\}$ , the set of model candidates was calculated



**Figure 6.2: Estimation of Pose and Illumination Type Using Pre-defined Categories** — *The system’s performance in categorizing human faces according to pose and illumination type using predefined categories is given. For optimal weightings of the feature- and correspondence-based parts, controlled by the respective relative threshold  $\theta^n$ , where  $n \in \{1, 2, 3\}$  codes the test case, the system performed best using partitioning  $\Pi^3$  for optimal weighting of the feature- and correspondence-based parts relative to partitionings  $\Pi^1$  and  $\Pi^2$ .*

by intersection of salient categories of partitioning  $\Pi^n$ , depending on the corresponding relative threshold  $\theta^n$  (Eq. (3.20)). Throughout, these ranged between 0.1 and 1, sampled in 0.1-steps. The selected model candidates were passed to the correspondence-based verification part. We considered the face in the input image to be correctly categorized if its pose and illumination type matched with the face in the model image.

The result of this experiment is given in fig.6.2. For optimal weightings of the feature- and correspondence-based parts, controlled by the respective relative threshold  $\theta^n$ , where  $n \in \{1, 2, 3\}$  codes the test case, the system performed best using partitioning  $\Pi^3$  for optimal weighting of feature- and correspondence-based parts relative to partitionings  $\Pi^1$  and  $\Pi^2$ .



**Figure 6.3: Estimation of Pose and Illumination Type Using Single-Element Categories** — *The system’s performance in categorizing human faces according to pose and illumination type using single-element categories is given. The system performed better than in the previous experiment. For the sake of comparison, the categorization performance in the case of unsegmented images is given here as well. A feature-based system would suffice to almost perfectly solve the categorization task in the case of unsegmented test images. In case of the PIE database, figure-ground segmentation thus complicates the categorization task.*

## 6.2.2 Estimation of Pose and Illumination Type Using Single-Element Categories

For evaluation of the system’s performance without predefined categories we arranged the learning set into  $K = 1$  partitioning of single-element categories. We considered the face in the input image to be correctly categorized if its pose and illumination type matched with the face in the model image.

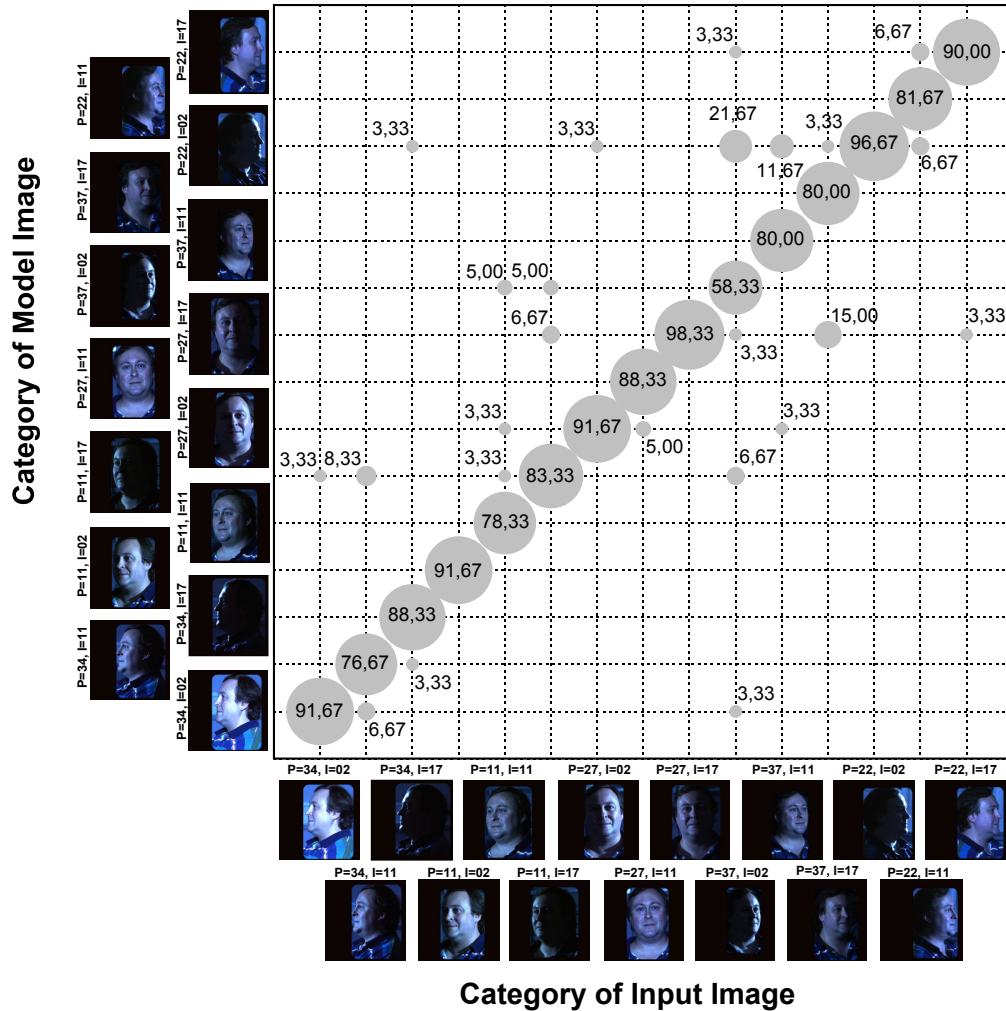
The attained results depending on the relative threshold  $\theta^1$  (Eq. (3.20)) are given in fig. 6.3. Like in the object categorization experiments (Chapter 5), disregarding predefined category information imposed on the learning sets improved categorization performance: the system performed better in the case of single-element categories relative to the previous experiment. In or-

der to prove our original statement that, ironically, figure-ground segmentation of the original PIE images complicates the categorization task, we give the system's performance for that case as well. A feature-based system would suffice to almost perfectly solve the categorization task in the case of unsegmented test images. This proves our original statement.

Fig. 6.4 gives a confusion matrix of the categorization performance in the case of single-element categories. In general the system was able to categorize human faces according to head pose and illumination type. As bunch graphs provide a means to integrate individual variations into the object representation, it is reasonable to assume that categorization performance can further be improved by using bunch instead of model graphs. As briefly mentioned in section 3.5, the graph dynamics is able to construct bunch graphs provided that the model features stem from well-chosen model candidates.

### 6.3 Discussion

In contrast to the categorization of general objects the system was in the majority of cases able to categorize unknown human faces according to head pose and illumination type, especially in view of the quality of the employed image data. Like in the categorization experiments with general objects, partitioning of the learning sets into categories employed by human cognition were not helpful to improve categorization capability. As model graphs only represent a single object view they cannot cover the whole spectrum of intra-category variations. It is therefore reasonable to assume that the system's performance can further be improved by using bunch graphs instead of model graphs. Moreover, application of more sophisticated correspondence-based techniques in the correspondence-based verification part, for instance (LADES et al., 1993; WISKOTT et al., 1997; WÜRTZ, 1997; TEWES, 2006), should allow for the same effect.



**Figure 6.4: Confusion Matrix of Categorization Performance** — A confusion matrix of the categorization performance in the case of single-element categories and optimal weighting of the feature- and correspondence-based parts is given, which was the case for  $\theta^1 = 0.7$  (Fig. 6.3). The axes are annotated with the captured head poses and illumination types, symbolized by images of an arbitrarily chosen individual. The horizontal axis codes head poses and illumination types of the face in the input images while the vertical axis codes head poses and illumination types of the face in the model images. For clarity of presentation, categorization rates less than 3% have been disregarded. In general, the system was able to categorize human faces according to head pose and illumination type.





# Chapter 7

## Summary and Future Work

*‘And what about you, Mr Stevens? What does the future hold for you back at Darlington Hall?’*

*‘Well, whatever awaits me, Mrs Benn, I know I’m not awaited by emptiness. If only I were. But oh no, there’s work, work and more work.’*

Kazuo Ishiguro — The Remains of the Day

We presented a form of graph dynamics that, upon image presentation, lets a model graph rapidly emerge by binding together memorized subgraphs derived from earlier learning examples driven by the image features. From the viewpoint of pattern recognition, the proposed technique is a combination of feature- and correspondence-based methods. The preselection network, implemented in the method’s feature-based part, is well suited to quickly rule out most irrelevant matches and only leaves the ambiguous cases, so-called model candidates, to be processed in the correspondence-based verification part, which is a rudimentary version of elastic graph matching. In the course of this model graphs emerge that describe the analyzed object well. This hybrid method outperformed both purely feature-based and purely correspondence-based approaches, especially when confronted with more sophisticated recognition tasks.

The proposed graph dynamics was applied to the tasks of visual object recognition, visual object categorization, and to the task of estimating pose and illumination type of human faces, which was set up as a categorization task.

In the object recognition experiments, the method was, unlike many other object recognition systems, not only very good at solving simple recognition tasks but also performed well when confronted with more sophisticated tasks, such as the recognition of objects in images with structured backgrounds, simultaneous recognition of multiple objects in simple visual scenes, and recognition of partially occluded objects. Moreover, it performed well in the case of sparse learning sets and visual dictionaries. In all experiments, the system's performance degraded smoothly with the complexity of the recognition task. The categorization experiments were, however, not that successful. The system's performance depended considerably on the amount of training data available for the categories. If the variations across category members were poorly sampled, categorization of objects from these categories failed frequently. Categories employed by human cognition were not helpful to improve the categorization capability, a finding which is compatible with published experimental results. The system performed, however, better for the categorization of unknown human faces according to head pose and illumination type. Partitioning of the learning sets into predefined categories employed by human cognition were again not helpful to improve the system's categorization capability.

What is left to do? Although the system performed favorably in the object recognition experiments, much work needs to be done to improve the categorization capabilities. In the medium term, these can be improved in three ways. First, one can expect that better databases and the integration of additional feature types, such as shape, color and so on, should allow for an improved categorization performance. Second, as bunch graphs provide a means to cover individual variations across category members, it is reasonable to assume that categorization performance would benefit from using bunch instead of model graphs. The correspondence-based part of the graph dynamics is prepared to construct bunch graphs provided that the model features stem from carefully chosen model candidates. The selection of appropriate model candidates is, however, subject to further studies. Third, the rudimentary version of elastic graph matching implemented in the system's correspondence-based part should be replaced by more sophisticated methods, which should lead to increased robustness under shape and pose variations. In our experiments we have seen that imposed predefined category information on the learning sets employed by human cognition were not helpful to improve categorization performance. In this thesis it has not been clarified if this is generally the case or only for the partitionings we used in our experiments. In order to address this question a method to retrieve categories from raw image data is required. Another problem is the amount of

features in the visual dictionary, see, for instance, the number of features in the feature vectors given in section 3.4.8: it appears biologically implausible that the system requires more than 300,000 features for the pose invariant recognition of only 100 objects, i.e., the simple variant of vector quantization employed in this thesis is not sufficient to reduce the total number of features to a manageable number. For this task far more powerful clustering techniques are demanded. A growing neural gas (MARTINETZ and SCHULTEN, 1991; FRITZKE, 1997) might be a good candidate. Finally, the system has shown to be quite sensitive towards changes in object scale. In order to achieve scale-invariance scale-dependent feature detectors have to be inserted in the feature detectors' input layers (Section 3.4.2), which to date are only position-invariant.

To conclude, we proposed a system that conforms to the principle of compositionality. The presented results, especially the system's object recognition capabilities, demonstrate that the principle is fundamental for the expression of cognitive structures. It provides a straightforward approach to handle visual scenes of multiple, possibly occluded objects.



# Appendix A

## Anhang in deutscher Sprache

*I have a prejudice against people who print things in a foreign language and add no translation. When I am the reader, and the other considers me able to do the translating myself, he pays me the quite a nice compliment — but if he would do the translating for me I would try to get along without the compliment.*

Mark Twain — A Tramp Abroad

### A.1 Zusammenfassung der Dissertation

Dieses Kapitel beinhaltet eine kurze Zusammenfassung der englischsprachigen Dissertation. Wo immer geeignete deutsche Fachbegriffe fehlen, werden die englischen Bezeichnungen beibehalten. Um das Auffinden von Details im Text zu erleichtern, entspricht die Einteilung in Unterkapitel der Kapiteleinteilung der Arbeit. Dem Dekan der Technisch-Naturwissenschaftlichen Fakultät der Universität zu Lübeck, Herrn Prof. Dr. Enno Hartmann, gilt mein Dank für die Erlaubnis, diese Arbeit in englischer Sprache einreichen zu dürfen.

### A.1.1 Einleitung

Eine wichtige Voraussetzung für die Beschreibung kognitiver Funktionen, zum Beispiel die visuelle Erkennung von Objekten, ist die Fähigkeit zur Konstruktion mentaler Objektrepräsentationen aus gespeicherten Teilen, sogenannten *Konstituenten*. In dieser Hinsicht ist die Datenstruktur des Gehirns als Graph vorstellbar, dessen Knoten mit elementaren Bildmerkmalen attribuiert sind, und dessen Kanten Relationen zwischen diesen Bildmerkmalen beschreiben.

Kognition wird also als aktiver Konstruktionsprozess aufgefasst, eine Notwendigkeit, die sich aus der Beschränktheit der immensen, aber dennoch begrenzten Ressourcen des Gehirns ergibt. Diese Grundannahme, die in frühen Modellen zur Beschreibung der Hirnfunktion völlig fehlt, wurde in psychophysikalischen Experimenten bestätigt. Die Lösung des Bindungsproblems, die Fähigkeit, Konstituenten in korrekter Weise miteinander zu assoziieren, d.h. aneinander zu *binden*, ist grundlegende Voraussetzung für den Wahrnehmungsprozess.

### A.1.2 Elastische Graphenanpassung

Eine sehr erfolgreiche Methode zur Erkennung bekannter Objektansichten unter leichten Variationen ist die elastische Graphenanpassung. Die zugrundeliegende Idee ist hier, dass das so genannte Korrespondenzproblem gelöst werden muss, bevor zwei Ansichten miteinander verglichen werden können. Das Korrespondenzproblem umfasst die Frage, welche Punkte in den Bildern zweier Objektansichten von einem gemeinsamen Punkt auf dem physikalischen Objekt stammen. Zu diesem Zweck werden Objektansichten durch einen Graphen beschrieben, dessen Knoten mit lokalen Bildmerkmalen attribuiert sind und dessen Kanten Relationen zwischen lokalen Merkmalen beschreiben. Die lokalen Bildmerkmale werden aus einer Gaborwavelettransformation gewonnen; die Antworten eines definierten Satzes von Gaborfiltern, angewandt auf ein Bild an einer bestimmten Position, werden in einem Merkmalsvektor, einem sogenannten *Jet* zusammengefasst. Das Korrespondenzproblem wird durch optimale Platzierung des Modellgraphen im Eingabebild durch Maximierung eines globalen Ähnlichkeitsmaßes gelöst, das auf den lokalen Jetähnlichkeiten an den korrespondierenden Punkten basiert. Der Prozess der Ähnlichkeitsmaximierung wird als *Matching* bezeichnet. Er besteht aus mehreren Schritten, sogenannten *Moves*, wobei jeder Move die Position der Modellgraphknoten im Eingabebild unter der Prämisse der Ähn-

lichkeitsmaximierung in spezifischer Weise variiert. Reihenfolge und Parametrierung der Moves ist in einer a priori festgelegten Liste, dem sogenannten *Matching Schedule*, festgelegt. Dort ist das Matching typischerweise als Grob-zu-Fein-Suche organisiert.

Unter der impliziten Annahme, dass die Ansichten des Modellobjekts und des zu erkennenden Objekts nur leicht variieren, hat sich die elastische Graphen Anpassung als korrespondenzbasierte Methode zur visuellen Erkennung von Objekten, insbesondere von menschlichen Gesichtern, bewährt. Es kommt allerdings zu Problemen, wenn diese Annahme nicht mehr zutreffend ist. Dies ist zum Beispiel der Fall, wenn eine größere Anzahl beliebiger Objekte mit voller Poseninvarianz zu erkennen ist. Die triviale Lösung, jede Objektansicht durch einen Modellgraphen zu beschreiben und diese der Reihe nach mit dem Eingabebild zu vergleichen, stellt sich schnell als nicht praktikabel und zudem biologisch unplausibel dar. Vielmehr ist eine Modellgraphendynamik wünschenswert, die, initiiert durch Präsentation eines Bilds, einen Modellgraphen aus Teilgraphen konstruiert, die zuvor aus Trainingsbildern der zu erkennenden Objekte gewonnen wurden.

### A.1.3 Emergenz von Modellgraphen

In diesem Abschnitt wird eine dreischrittige Modellgraphendynamik vorgestellt. Im ersten Schritt werden positionsinvariante Merkmalsdetektoren anhand von Trainingsbildern der zu erkennenden Objekte gelernt. Als Bildmerkmale werden durchweg kleine lokale Gridgraphen, sogenannte *Parkettgraphen* verwendet, deren Knoten mit den Amplituden der Gaborwavelettransformierten attribuiert sind. Parkettgraphen eignen sich sowohl als lokale Bildmerkmale als auch als Konstituenten für Modellgraphen. Die Menge der aus den Trainingsbildern extrahierten Parkettgraphen wird mittels einer Vektorquantisierung begrenzt. Die nach der Vektorquantisierung verbleibenden Merkmale werden in einem Vektor abgespeichert. Für jedes dieser Merkmale wird ein positionsinvarianter Merkmalsdetektor mittels Disjunktion lokaler Merkmalsdetektoren konstruiert. Das einem lokalen oder positionsinvarianten Merkmalsdetektor zugeordnete Merkmal wird als Modell- oder Referenzmerkmal bezeichnet. Die lokalen Detektoren signalisieren die Existenz eines ausreichend ähnlichen Bildmerkmals an einer definierten Position im Eingabebild, während die positionsinvariante Merkmalsdetektoren die Existenz eines oder mehrerer solcher Bildmerkmale bezogen auf die gesamte Bildebene anzeigen.

Die positionsinvarianten Merkmalsdetektoren werden in paralleler Weise kombiniert. Jedem Merkmalsdetektor wird ein verallgemeinertes McCulloch & Pitts-Neuron aus der Eingabeschicht eines Einschichtenperzeptrons zugeordnet. Die Ausgabeschicht dieses Netzwerks enthält je ein verallgemeinertes McCulloch & Pitts-Neuron für a priori festgelegte Kategorien von Trainingsbildern, d.h. Mengen von Beispielbildern mit einer gemeinsamen semantischen Eigenschaft. Das Netzwerk ist vollvernetzt, alle Eingabe- sind mit allen Ausgabeneuronen über vorwärtsgerichtete Synapsen verbunden. Die Ausgabe eines Ausgabeneurons, in dieser Arbeit *Saliency* genannt, skaliert mit der Wahrscheinlichkeit, dass das Objekt im Eingabebild zur Kategorie gehört, die dem Ausgabeneuron zugeordnet wurde. Dieses Netzwerk wird fortan als *Preselection Network* bezeichnet. Das Preselection Network erfüllt LINSKER's Infomax-Prinzip. Dieses Prinzip besagt, dass sich die synaptischen Gewichte eines Mehrschichtennetzwerks mit ausschließlich vorwärtsgerichteten Verbindungen zwischen benachbarten Schichten mittels HEBB'scher synaptischer Plastizität derart entwickeln, dass die Ausgabe eines jeden Neurons maximal informationserhaltend bezüglich seiner Eingaben ist. Unter Berücksichtigung dieser Randbedingungen und unter der Annahme, dass sich das System in einer stationären Umgebung befindet, können die synaptischen Gewichte direkt zugewiesen werden. Das zeitraubende iterative Einstellen der Gewichte entfällt. Den synaptischen Gewichten werden Informationsmaße zugewiesen, die auf der SHANNON-Entropie basieren. Diese sind charakterisiert durch den Beitrag des Merkmalsdetektors, der dem präsynaptischen Neuron zugewiesen wurde, zur Entscheidung über die Zugehörigkeit des präsentierten Objekts zur Kategorie, die dem postsynaptischen Neuron zugewiesen wurde. Durch Definition eines Schwellenwertes auf den Saliencies lassen sich nach Präsentation eines Eingabebildes nun leicht Kategorien ermitteln, zu denen das Objekt im Eingabebild vermutlich gehört. Diese werden als *salient*, die darin enthaltenen Trainingsbeispiele als *Modellkandidaten* bezeichnet. Die Anzahl der Modellkandidaten ist üblicherweise erheblich kleiner als die Gesamtzahl der Trainingsbeispiele.

Jeder Modellkandidat wird mit einer rudimentären Version der elastischen Graphenanpassung verifiziert, die nur den sogenannten *Scan Global Move* umfasst. Es wird überprüft, ob die räumliche Anordnung der Bild- und Modellmerkmale übereinstimmt. Hierzu wird jeweils ein Bild- und ein Modellgraph aus den im zweiten Schritt der Modellgraphendynamik gesammelten matchenden Bild- und Modellparkettgraphen konstruiert. Bild- und Modellgraph werden miteinander verglichen. Der Modellkandidat, dessen Modellgraph die höchste Ähnlichkeit mit dem Eingabebild erzielt, wird als Modell für das Eingabebild angenommen.



### A.1.4 Objekterkennung

Die Modellgraphendynamik wurde auf das Problem der visuellen Erkennung von Objekten angewandt. *Objekterkennung* bedeutet eine Entscheidung über Identität und erfordert Diskriminierung zwischen Identitäten und Verallgemeinerung über Objektvariationen wie Verschiebung, Skalierung, Tiefenrotation, Verdeckung, Beleuchtung, Rauschen und so weiter. Wegen der Vielzahl möglicher Variationen, die auch in Kombination auftreten können, ist die Erkennung von Objekten ein schwieriges Problem.

Es wurden Experimente mit zwei Datenbanken durchgeführt. Die Experimente umfassten die Erkennung einzelner Objekte, mit und ohne Hintergrund, mit Größenvariation, mit wenigen Trainingsbeispielen und mit wenigen Merkmalen, sowie die gleichzeitige Erkennung mehrerer Objekte, mit und ohne Hintergrund und mit und ohne Verdeckung.

Die erzielten Ergebnisse stimmten qualitativ für beide Datenbanken überein. Generell wurden mit dem getesteten System gute, teils sehr gute Ergebnisse erzielt. In allen Experimenten nahm die Leistung des Systems gleichmäßig mit der Komplexität der Erkennungsaufgabe ab.

### A.1.5 Objektkategorisierung

Die Modellgraphendynamik wurde ebenfalls auf das Problem der Kategorisierung von Objekten angewandt. *Objektkategorisierung* bedeutet eine Entscheidung über die Art eines Objekts, die Entscheidung über eine bestimmte semantische Eigenschaft. Das System muss also zusätzlich zu Objektvariationen wie Pose, Beleuchtung etc. über die Identität des Objekts generalisieren, obwohl die individuellen Unterschiede unter den Kategorieelementen erheblich sein können. Daher ist die Kategorisierung von Objekten eine wesentlich anspruchsvollere Aufgabe als das Erkennen von Objekten.

Die für die Experimente verwendete Datenbank enthielt Bilder allgemeiner Objekte, unterteilt in acht Kategorien. Die Kategorisierungsaufgabe bestand darin, das Objekt im Eingabebild einer dieser acht Kategorien zuzuordnen.

Wie zu erwarten, sank die Leistung des Systems im Vergleich zu den Ergebnissen der Objekterkennung teilweise erheblich. Die Kategorisierungsleistung hing stark von der Variation unter den Kategorieelementen ab. So konnten zum Beispiel Bilder von Früchten recht gut kategorisiert werden, das System versagte aber weitestgehend bei die Kategorisierung von Tieren.

Die Vermutung, dass sich die Kategorisierungsaufgabe durch vorgegebene Partitionierungen der Trainingsmengen vereinfachen ließe, wurde in unseren Experimenten eindeutig widerlegt. Dieser Befund stimmt mit der bisher vorliegenden Literatur überein. Die Kategorisierungsleistung könnte durch größere Datenbanken, durch Integration weiterer Merkmalstypen und durch die Verwendung von Bunch- anstelle von Modellgraphen vermutlich noch erheblich gesteigert werden. Diese Erweiterung wird in der vorliegenden Arbeit allerdings nicht mehr realisiert.

### **A.1.6 Schätzung von Pose und Beleuchtung menschlicher Gesichter**

In einem weiteren Experiment wurde die Modellgraphendynamik auf das Problem der Posen- und Beleuchtungsschätzung anhand von Bildern menschlicher Gesichter angewandt. Pose und Beleuchtung sind sogenannte extrinsische Objektparameter.

Die Schätzaufgabe wurde als Kategorisierungsaufgabe formuliert. Die verwendete Datenbank, die Bilder menschlicher Gesichter in unterschiedlichen Kopfposen und unter verschiedenen Beleuchtungen enthielt, wurde in Kategorien von Gesichtern mit jeweils gleichen zugrundeliegenden Posen- und Beleuchtungsparametern aufgeteilt. Die Aufgabe bestand darin, das Gesicht im Eingabebild einer dieser Kategorien zuzuordnen.

Insgesamt war das System gut in der Lage, insbesondere im Hinblick auf die Qualität der Eingabedaten, Gesichter hinsichtlich Pose und Beleuchtung zu kategorisieren und damit die Schätzaufgabe zu lösen. Wie bei der Kategorisierung allgemeiner Objekte, konnte die Kategorisierungsaufgabe mittels vorgegebener Partitionierungen der Trainingsmengen nicht vereinfacht werden. Die Leistungsfähigkeit des Systems ließe sich mit den in Abschnitt A.1.5 aufgeführten Mitteln vermutlich noch weiter steigern.

### **A.1.7 Zusammenfassung und Ausblick**

In dieser Arbeit wurde eine Modellgraphendynamik vorgestellt, die nach Präsentation eines Eingabebildes, einen Modellgraphen durch Aggregation von Teilgraphen konstruiert, die aus Trainingsbildern extrahiert wurden. Dieser repräsentiert das Objekt im Eingabebild in geeigneter Weise. Aus Sicht der Mustererkennung ist das vorgestellte Verfahren eine Kombinati-

on von merkmals- und korrespondenzbasierten Methoden. Das Preselection Network, implementiert im merkmalsbasierten Teil des Systems, sortiert diejenigen Trainingsbeispiele aus, die als Modell für das präsentierte Bild nicht in Frage kommen. Die verbleibenden Trainingsbeispiele, Modellkandidaten genannt, werden im korrespondenzbasierten Teil mittels einer rudimentäre Graphenanpassung verifiziert. Im Zuge dieser Verifikation entstehen Modellgraphen. Dieser hybride Ansatz war ausschließlich merkmals- und korrespondenzbasierten Verfahren überlegen, insbesondere wenn das System mit schwierigeren Erkennungsaufgaben konfrontiert wurde.

Die Modellgraphendynamik wurde auf das Problem der visuellen Erkennung und Kategorisierung von Objekten, sowie auf die Schätzung von Posen- und Beleuchtungsparametern in Bildern menschlicher Gesichter angewandt, wobei die Parameterschätzung als Kategorisierungsaufgabe formuliert wurde. In den Objekterkennungsexperimenten erzielte die Methode gute, teils sehr gute Ergebnisse, die Ergebnisse in den Kategorisierungsexperimenten konnten insgesamt zufriedenstellen. In den Objekterkennungsexperimenten konnte die Methode, im Gegensatz zu vielen anderen Objekterkennungssystemen, nicht nur einfache, sondern auch kompliziertere Erkennungsaufgaben, wie die Erkennung mehrerer bekannter Objekte in einfachen Szenen oder die Erkennung teilverdeckter Objekte bewältigen. In allen Experimenten nahm die Leistung des Systems gleichmäßig mit der Komplexität der Erkennungsaufgabe ab. Die Kategorisierungsexperimente waren hingegen nicht sehr erfolgreich, ein Umstand, der sich aus der Komplexität der Aufgabenstellung aber auch aus zukünftig abzustellenden Schwächen des Systems ergibt. In unseren Experimenten hing die Kategorisierungsleistung erheblich vom Grad der Abdeckung individueller Variationen unter den Kategorieelementen mit Trainingsbeispielen ab. Die Kategorisierungsaufgabe ließ sich in keinem der durchgeführten Experimente durch vorgegebene Partitionierungen der Trainingsmengen vereinfachen, ein Befund, der mit der bisher vorliegenden Literatur übereinstimmt. Etwas erfolgreicher, insbesondere im Hinblick auf die Qualität der verwendeten Bilder, waren die Experimente zur Schätzung von Posen- und Beleuchtungsparametern. Auch hier waren vorgegebene Partitionierungen der Trainingsmengen nicht hilfreich, um die Leistung des Systems zu verbessern.

Die Kategorisierungsleistung des Systems ließe sich durch die Integration weiterer Bildmerkmale, wie zum Beispiel Farbe oder Form, durch die Verwendung von Bunch- anstelle von Modellgraphen und durch die Verwendung besserer korrespondenzbasierter Verfahren im Verifikationsteil sicherlich noch deutlich steigern. Da in keinem dieser Experimente die Kategorisierungslei-

stung durch vorgegebene Partitionierungen der Trainingsmengen verbessert wurde, ergibt sich die Frage, ob dies generell oder nur für die in unseren Experimenten verwendeten Kategorien der Fall ist. Zur Beantwortung dieser Frage ist ein Verfahren erforderlich, mit dem Kategorien aus Beispielbildern unüberwacht gelernt werden können. Ein weiteres Problem ist die biologisch unplausible Anzahl von Merkmalen im merkmalsbasierten Teil. Es ist offensichtlich, dass die in dieser Arbeit verwendete einfache Variante der Vektorquantisierung die Merkmalsflut nur unzureichend einzudämmen vermag. Zudem wäre die größeninvariante Erkennung von Objekten wünschenswert. Hierzu müssen skaleninvariante Merkmalsdetektoren in die Eingabeschichten der bis dato nur positionsinvarianten Merkmalsdetektoren eingefügt werden (Section 3.4.2).

Abschließend bleibt festzustellen, dass ein System realisiert wurde, das, nach Präsentation eines Eingabebildes, Objektrepräsentationen in Form von Modellgraphen aus gespeicherten Konstituenten zu konstruieren vermag. Die erzielten Ergebnisse, insbesondere die der Objekterkennungsexperimente, demonstrieren, dass dieses Prinzip grundlegend für die Beschreibung kognitiver Strukturen ist. Es ermöglicht in direkter Weise die Erkennung mehrerer, möglicherweise teilverdeckter Objekte in Szenen.

## A.2 Lebenslauf

### Persönliche Daten

Name	Günter Westphal
Geburtsdatum	16. Juli 1970
Geburtsort	Anholt, jetzt Isselburg
Adresse	Braunschweigweg 4, 46395 Bocholt
Telefon	02871/183480, 0234/32-27845
E-Mail	guenter.westphal@neuroinformatik.rub.de

### Schulbildung

1977 — 1981	Pfarrer-Wigger-Grundschule, Bocholt
1981 — 1990	Städtisches Mariengymnasium, Bocholt
Mai 1990	Abitur

### Zivildienst

1990 — 1991	Malteser Hilfsdienst, Bocholt
-------------	-------------------------------

### Studium

1991 — 1998	Studium der Informatik an der Universität Koblenz-Landau, Abteilung Koblenz Diplomarbeit: <i>Training mehrschichtiger feedforward Netze mit Integerarithmetik</i> bei Dr. W. Schiffmann
-------------	--

### Beruflicher Werdegang

1994 — 1996	Studentische Hilfskraft am Institut für Sozialwissenschaftliche Informatik der Universität Koblenz-Landau
1996	Studentische Hilfskraft am Institut für Informatik der Universität Koblenz-Landau (Übungen in Theoretischer Informatik)

- 1996 — 03/1998      Werkstudent bei der Firma Technische Informationssysteme, Bocholt
- 04/1998 — 06/2002      Software-Entwickler bei der Firma Technische Informationssysteme, Bocholt, Projekte u.a. für die Siemens AG und für die Infineon Technologies AG
- 07/2002 —      Wissenschaftlicher Mitarbeiter am Institut für Neuroinformatik, Lehrstuhl Systembiophysik von Prof. Dr. Christoph von der Malsburg, Ruhr-Universität Bochum

# List of Figures

2.1	Model Graph . . . . .	11
2.2	Gabor Function . . . . .	12
2.3	Creation of Jets . . . . .	13
2.4	Similarity Potentials of the Jet Similarity Function . . . . .	15
2.5	Overview of Moves . . . . .	17
2.6	Overview of Elastic Graph Matching . . . . .	18
2.7	Bunch Graph . . . . .	19
2.8	Reconstruction from a Bunch Graph . . . . .	20
2.9	Examples of Emerged Model Graphs . . . . .	21
3.1	Feature-Driven Emergence of Model Graphs . . . . .	25
3.2	Selection of the Model . . . . .	27
3.3	Case Study: Learning Set . . . . .	29
3.4	Case Study: Partitioning of the Learning Set . . . . .	30
3.5	Hierarchical Organization of Categories . . . . .	31
3.6	Example of a Parquet Graph . . . . .	32
3.7	Potentials of the Parquet Graph Similarity Function . . . . .	34
3.8	Case Study: Application of the Feature Calculator to the Learning Images . . . . .	36
3.9	Vector Quantization Method . . . . .	38

---

3.10	Position-Invariant Feature Detector . . . . .	42
3.11	Case Study: Weight Matrix . . . . .	46
3.12	Preselection Network . . . . .	47
3.13	Case Study: Application of Feature Calculator $f^1$ to Image $I'_2$ . . . . .	48
3.14	Case Study: Computation of Saliencies . . . . .	50
3.15	Case Study: Synaptic Plasticity . . . . .	54
3.16	Average Number of Model Candidates in Dependence on a Relative Threshold . . . . .	56
3.17	Computation of Saliencies . . . . .	58
3.18	Search Matching Features . . . . .	60
3.19	Definition of Search Spaces . . . . .	61
3.20	Performance of the Accelerated Feature Search . . . . .	63
3.21	Construction of Model Graphs . . . . .	69
3.22	Matching Setup . . . . .	70
4.1	Example Images of the COIL-100 and of the ALOI Image Database . . . . .	76
4.2	Input Images of a Single Object . . . . .	78
4.3	Recognition of Single Objects (Identity) . . . . .	80
4.4	Recognition of Single Objects (Pose) . . . . .	81
4.5	Recognition of Single Objects using Majority Vote as Verifi- cation Method . . . . .	83
4.6	Input Images of Scaled Objects . . . . .	84
4.7	Recognition of Scaled Single Objects . . . . .	85
4.8	Recognition of Single Objects with Sparse Learning Sets . . . . .	87
4.9	Recognition of Single Objects with Sparse Visual Dictionaries (Identity) . . . . .	89



---

4.10	Recognition of Single Objects with Sparse Visual Dictionaries (Execution Time) . . . . .	90
4.11	Input Images of Multiple Objects . . . . .	91
4.12	Recognition of Multiple Objects (Segmented Images) . . . . .	92
4.13	Recognition of Multiple Objects (Unsegmented Images) . . . . .	93
4.14	Input Images of a Partially Occluded Object . . . . .	95
4.15	Recognition of Partially Occluded Objects (Segmented Images)	96
4.16	Recognition of Partially Occluded Objects (Unsegmented Im- ages) . . . . .	97
5.1	Categorization of General Objects Using Hierarchically Orga- nized Categories . . . . .	105
5.2	Categorization of General Objects Using Single-Element Cat- egories . . . . .	107
5.3	Confusion Matrix of Categorization Performance . . . . .	108
6.1	Preprocessed Images of Human Heads with Variation in Pose and Illumination . . . . .	112
6.2	Estimation of Pose and Illumination Type Using Predefined Categories . . . . .	114
6.3	Estimation of Pose and Illumination Type Using Single-Element Categories . . . . .	115
6.4	Confusion Matrix of Categorization Performance . . . . .	117



# List of Tables

3.1	Case Study: Computation of the Feature Vector . . . . .	39
3.2	Case Study: Calculation of Measures of Information . . . . .	44
3.3	Case Study: Matching Features . . . . .	49
3.4	Performance of the Accelerated Feature Search . . . . .	62



# Bibliography

- ARENTZ, M. *Integration einer merkmalsbasierten und einer korrespondenzbasierten Methode zur Klassifikation von Audiodaten*. Master's Thesis, Computer Science, University of Dortmund, D-44221 Dortmund, Germany, 2006.
- BELIS, M. AND GUIASU, S. A Quantitative-Qualitative Measure of Information in Cybernetic Systems. *IEEE Transactions on Information Theory*, 14: 593–594, 1968.
- BIEDERMAN, I. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94: 115–147, 1987.
- BIENENSTOCK, E. AND GEMAN, S. Compositionality in Neural Systems. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, 223–226. MIT Press, Cambridge, Massachusetts, London, England, 1995.
- BÜLTHOFF, H.H. AND EDELMAN, S. Psychological Support for a 2-D View Interpolation Theory of Object Recognition. *Proceedings of the National Academy of Science*, 89: 60–64, 1992.
- BUNKE, H. Graph Grammars as a Generative Tool in Image Understanding. In M. Nagl H. Ehrig and G. Rozenberg, editors, *Graph Grammars and their Application to Computer Science*, 153 of *LNCS*, 8–19. Springer, 1983.
- EDELMAN, S. Representation, Similarity, and the Chorus of Prototypes. *Minds and Machines*, (5): 45–68, 1995.
- ELLIFFE, M.C.M., ROLLS, E.T., AND STRINGER, S.M. Invariant Recognition of Feature Combinations in the Visual System. *Biological Cybernetics*, 86: 59–71, 2002.
- ESHERA, M.A. AND FU, K.S. An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5): 604–618, 1986.

- FEI-FEI, L., FERGUS, R., AND PERONA, P. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2, 1134–1141. 2003.
- FRITZ, G., SEIFERT, C., PALETTA, L., AND BISCHOF, H. Entropy based Saliency Maps for Object Recognition. In *Early Cognitive Vision Workshop*. 2004.
- FRITZKE, B. A Self-Organizing Network That Can Follow Non-Stationary Distributions. In *International Conference on Artificial Neural Networks (ICANN 1997)*, 613–618. Springer, 1997.
- FUKUSHIMA, K., MIYAKE, S., AND ITO, T. Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13(5): 826–834, 1983.
- GAUTHIER, I., HAYWARD, W.G., TARR, M.J., ANDERSON, A., SKUDLARSKI, P., AND GORE, J.C. Bold Activity During Mental Rotation and Viewpoint-Dependent Object Recognition. *Neuron*, 34(1): 161–171, 2002.
- GEUSEBROEK, J.M., BURGHOUTS, G.J., AND SMEULDERS, A.W.M. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61: 103–112, 2005.
- GRAY, R. Vector Quantization. *IEEE Signal Processing Magazine*, 1(2): 4–29, 1984.
- HEBB, D.O. *The Organization of Behavior*. Wiley, New York, 1949.
- HUBEL, D.H. AND WIESEL, T.N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *Journal of Physiology*, 160: 106–154, 1962.
- HUMMEL, J.E. AND BIEDERMAN, I. Dynamic Binding in a Neural Network for Shape Recognition. *Psychological Review*, 99(3): 480–517, 1992.
- JONES, J.P. AND PALMER, L.A. An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex. *Journal of Neurophysiology*, 58(6): 1233–1258, 1987.
- LADES, M., VORBRÜGGEN, J.C., BUHMANN, J., LANGE, J., VON DER MALSBERG, C., WÜRTZ, R.P., AND KONEN, W. Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Transactions on Computers*, 42(3): 300–310, 1993.

- LAM, L. AND SUEN, S.Y. 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, 1997.
- LEIBE, B. AND SCHIELE, B. Analyzing Appearance and Contour Based Methods for Object Categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR'03)*, 2, 409–415. IEEE Press, Madison, Wisconsin, USA, 2003.
- LINSKER, R. Self-Organization in a Perceptual Network. *IEEE Computer*, 105–117, 1988.
- LOGOTHETIS, N.K. AND PAULS, J. Psychophysical and Physiological Evidence for Viewer-Centered Object Representation in the Primate. *Cerebral Cortex*, 3: 270–288, 1995.
- LOOS, H.S. *User-Assisted Learning of Visual Object Recognition*. PhD Thesis, University of Bielefeld, Germany, 2002.
- MARR, D. AND NISHIHARA, H.K. Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. *Proceedings of the Royal Society of London, Section B*, 200: 269–294, 1978.
- MARTINETZ, T. AND SCHULTEN, K.J. A 'Neural-Gas' Network Learns Topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, 397–402. Amsterdam, The Netherlands, 1991.
- MCCULLOCH, W.S. AND PITTS, W.H. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5: 115–133, 1943.
- MEL, B.W. SEEMORE: Combining Color, Shape, and Texture Histogramming in a Neurally Inspired Approach to Visual Object Recognition. *Neural Computation*, 9: 777–804, 1997.
- MESSER, K., KITTLER, J., SADEGHI, M., HAMOUZ, M., KOSTIN, A., CARDINAUX, F., MARCEL, S., BENGIO, S., SANDERSON, C., POH, N., RODRIGUEZ, Y., CZYZ, J., VANDENDORPE, L., MCCOOL, C., LOWTHER, S., SRIDHARAN, S., CHANDRAN, V., PALACIOS, R. P., VIDAL, E., BAI, L., SHEN, L., WANG, Y., YUEH-HSUAN, C., HSIEN-CHANG, L., YI-PING, H., HEINRICH, A., MÜLLER, M., TEWES, A., VON DER MALSBERG, C., WÜRTZ, R., WANG, Z., XUE, F., MA, Y.,

- YANG, Q., FANG, C., DING, X., LUCEY, S., GOSS, R., AND SCHNEIDERMAN, H. Face Authentication Test on The Banca Database. In Josef Kittler, Maria Petrou, and Mark Nixon, editors, *17th International Conference on Pattern Recognition (ICPR 2004)*, 4, 523–532. IEEE Press, Cambridge, UK, 2004.
- MESSMER, B.T. AND BUNKE, H. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5): 493–504, 1998.
- MURASE, H. AND NAYAR, S.K. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14: 5–24, 1995.
- NENE, S.A., NAYAR, S.K., AND MURASE, H. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996.
- OLSHAUSEN, B.A., ANDERSON, C.H., AND VAN ESSEN, C. A Neurobiological Model of Visual Attention and Invariant Recognition Based on Dynamic Routing of Information. *The Journal of Neuroscience*, 13(11): 4700–4719, 1993.
- OLSHAUSEN, B.A. AND FIELD, D.J. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature*, 381(6583): 607–609, 1996.
- PALMERI, T.J. AND GAUTHIER, I. Visual Object Understanding. *Nature Reviews Neuroscience*, 5: 291–304, 2004.
- PERRET, D.I., SMITH, P.A.J., POTTER, D.D., MISTLIN, A.J., HEAD, A.S., AND MILNER, A.D. Visual Cells in the Temporal Cortex Sensitive to Face View and Gaze Direction. *Proceedings of the Royal Society B*, 223: 293–317, 1985.
- PETERS, G. *A View-Based Approach to Three-Dimensional Object Perception*. PhD Thesis, University of Bielefeld, Germany, 2001.
- PFAFFELHUBER, E. Learning and Information Theory. *International Journal of Neuroscience*, 3: 83–88, 1972.
- PHILLIPS, P.J., MOON, H., RIZVI, S.A., AND RAUSS, P.J. The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10): 1090–1103, 2000.



- PIAGET, J. *Das Erwachen der Intelligenz beim Kinde*, chapter 4, 21. Klett, Stuttgart, 1975a.
- PIAGET, J. *Der Aufbau der Wirklichkeit beim Kinde*, chapter 10. Klett, Stuttgart, 1975b.
- PITTS, W. AND MCCULLOCH, W. How we know Universals: The Preception of Auditory and Visual Forms. *Bulletin of Mathematical Biophysics*, 9: 127–147, 1947.
- PÖTZSCH, M., MAURER, T., WISKOTT, L., AND VON DER MALSBURG, C. Reconstruction from Graphs Labeled with Responses of Gabor Filters. In C. von der Malsburg, W. von Seelen, J. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of the ICANN 1996*, 845–850. Berlin, Heidelberg, New York: Springer, 1996.
- RIESENHUBER, M. AND POGGIO, T. Models of Object Recognition. *Nature Neuroscience*, 3: 1199–1204, 2000.
- ROSENBLATT, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65: 386–408, 1958.
- ROSENBLATT, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington DC: Spartan, 1962.
- SCHMIDT, P.A. AND WESTPHAL, G. Object Manipulation by Integration of Visual and Tactile Representations. In Uwe J. Ilg, Heinrich H. Bülthoff, and Hanspeter A. Mallot, editors, *Dynamic Perception*, 101–106. infix Verlag/IOS press, 2004.
- SCHNEIDERMAN, H. AND KANADE, T. A Statistical Approach to 3D Object Detection Applied to Faces and Cars. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, 746–751. 2000.
- SHAMS, L.B. *Development of Visual Shape Primitives*. PhD Thesis, University of Southern California, 1999.
- SHANNON, C.E. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27: 623–656, 1948.
- SHAPIRO, L.G. AND HARALICK, R.M. Structural Descriptions and Inexact Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5): 504–519, 1981.

- SIM, T., BAKER, S., AND BSAT, M. The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces. Technical Report CMU-RI-TR-01-02, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.
- SIM, T., BAKER, S., AND BSAT, M. The CMU Pose, Illumination, and Expression Database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12): 1615–1618, 2003.
- TANG, F. AND TAO, H. Object Tracking with Dynamic Feature Graph. In *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 25–32. Beijing, China, 2005.
- TARR, M.J. Rotating Objects to Recognize Them: A Case Study of the Role of Viewpoint Dependency in The Recognition of Three-Dimensional Objects. *Psychonomic Bulletin and Review*, 2(1): 55–82, 1995.
- TARR, M.J. How Experience Shapes Vision. *Psychological Science Agenda*, 19(7). URL [http://www.apa.org/Science/psa/jul05\\_mainprnt.html](http://www.apa.org/Science/psa/jul05_mainprnt.html), 2005.
- TEWES, A. *A Flexible Object Model for Encoding and Matching Human Faces*. PhD Thesis, Physics Department, University of Bochum, Germany, 2006.
- THORPE, S., FIZE, D., AND MARLOT, C. Speed of Processing in the Human Visual System. *Nature*, 381: 520–522, 1996.
- THORPE, S. AND THORPE, M.F. Seeking Categories in the Brain. *Neuroscience*, 291: 260–263, 2001.
- TREISMANN, A. AND GELADE, G. A Feature Integration Theory of Attention. *Cognitive Psychology*, 12: 97–136, 1980.
- ULLMAN, S. Aligning Pictorial Descriptions: An Approach to Object Recognition. *Cognition*, (32): 193–254, 1989.
- ULLMAN, S. AND SALI, E. Object Classification Using a Fragment-Based Representation. In S.-W. Lee, H.H. Bülthoff, and T. Poggio, editors, *First IEEE International Workshop on Biologically Motivated Computer Vision, Lecture Notes in Computer Science 1811*, 73–87. Springer, Berlin, Heidelberg, 2000.

- ULUSOY, I. AND BISHOP, C.M. Generative Versus Discriminative Methods for Object Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, California, USA, 2, 258–265. IEEE Press, 2005.
- VIOLA, P. AND JONES, M.J. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 1, 511–518. 2001.
- VIOLA, P. AND JONES, M.J. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2): 137–154, 2004.
- VON DER MALSBURG, C. The Correlation Theory of Brain Function. Internal report 81-2, Max-Planck-Institute for Biophysical Chemistry, Department of Neurobiology, 1981.
- VON DER MALSBURG, C. Pattern Recognition by Labeled Graph Matching. *Neural Networks*, 1: 141–148, 1988.
- VON DER MALSBURG, C. The What and Why of Binding: The Modeler’s Perspective. *Neuron*, 24: 95–104, 1999.
- VON DER MALSBURG, C. The Dynamic Link Architecture. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, 1002–1005. MIT Press, Cambridge, Massachusetts, London, England, second edition, 2002.
- VON DER MALSBURG, C. AND REISER, K. Pose Invariant Object Recognition in a Neural System. In F. Fogelmann-Soulié, J. C. Rault, P. Gallinari, and G. Dreyfus, editors, *International Conference on Artificial Neural Networks (ICANN 1995)*, 127–132. EC2 & Cie, Paris, France, 1995.
- WEBER, M., WELLING, M., AND PERONA, P. Unsupervised Learning of Models for Recognition. In *Proceedings of the 6th European Conference on Computer Vision (ECCV)*, 18–32. Dublin, Ireland, 2000.
- WERSING, H. AND KÖRNER, E. Learning Optimized Features for Hierarchical Models of Invariant Object Recognition. *Neural Computation*, 15: 1559–1588, 2003.
- WESTPHAL, G. Classification of Molecules into Classes of Toxicity. Technical report, Dr. Holthausen GmbH, Bocholt, Germany, 2004.

WESTPHAL, G. AND WÜRTZ, R.P. Fast Object and Pose Recognition Through Minimum Entropy Coding. In Josef Kittler, Maria Petrou, and Mark Nixon, editors, *17th International Conference on Pattern Recognition (ICPR 2004)*, 3, 53–56. IEEE Press, Cambridge, UK, 2004.

WIEGHARDT, J. *Learning the Topology of Views: From Images to Objects*. PhD Thesis, Physics Department, University of Bochum, Germany, 2001.

WISKOTT, L. *Labeled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis*. PhD Thesis, Physics Department, University of Bochum, Germany, 1995.

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N., AND VON DER MALSBERG, C. Face Recognition by Elastic Bunch Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 775–779, 1997.

WITTEN, I.H. AND FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, 2000.

WUNDRICH, I.J. *Parametrisierte zweidimensionale Modelle für dreidimensionale Gesichtserkennung*. PhD Thesis, Department of Electrical Engineering, University of Bochum, Germany, 2004.

WÜRTZ, R.P. *Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.

WÜRTZ, R.P. Object Recognition Robust Under Translations, Deformations, and Changes in Background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 769–775, 1997.

ZHU, J. AND VON DER MALSBERG, C. Maplets for Correspondence-Based Object Recognition. *Neural Networks*, 17: 1311–1326, 2004.

# Previously Published Contents of this Thesis

Parts of chapter 3 have been published in

G. Westphal, *Classification of Molecules into Classes of Toxicity*, Technical Report, Dr. Holthausen GmbH, Bocholt, Germany, 2004.

Parts of chapters 3 and 4 have been published in

G. Westphal and R. P. Würtz, *Fast Object and Pose Recognition Through Minimum Entropy Coding*, Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), J. Kittler, M. Petrou, and M. Nixon (editors), vol. 3, pp. 53–56, IEEE Press, Cambridge, UK, 2004.

P. A. Schmidt and G. Westphal, *Object Manipulation by Integration of Visual and Tactile Representations*, Proceedings of the Dynamic Perception Workshop (DP 2004), U. J. Ilg, H. H. Bülthoff and H. A. Mallot (editors), Tübingen, Germany, 2004.

G. Westphal and R. P. Würtz, *Verfahren zur schnellen Vorauswahl von Referenzmustern*, Patent Application, Deutsches Patent- und Markenamt, Munich, Germany, 2004.

Parts of chapters 3 and 6 have been published in

G. Westphal and R. P. Würtz, *Recognition of Pose and Illumination of Human Faces Through Combination of Feature-based and Correspondence-based Pattern Recognizers*, Proceedings of the 8th Tübingen Perception Conference (TWK 2005), H. H. Bülthoff, H. A. Mallot, R. Ulrich, and F. A. Wichmann (editors), Tübingen, Germany, 2005.

Parts of chapters 3, 4, and 5 have been published in

G. Westphal, C. von der Malsburg, and R. P. Würtz, *Feature-Driven Emergence of Model Graphs for Object Recognition and Categorization* in Applied Pattern Recognition, A. Kandel, H. Bunke, and M. Last (editors), 2006. Accepted.

A preprint of this is published on the website of the Dagstuhl Seminar 06031 (<http://www.dagstuhl.de>).