Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften
in der Fakultät für Physik und Astronomie
der Ruhr-Universität Bochum

von

Rolf P. Würtz
aus Heidelberg

Tag der Disputation: 19.12.1994
Gutachter: Prof. Dr. Christoph von der Malsburg
Prof. Dr. Günter Wunner

Abstract

The major tasks for automatic object recognition are segmentation of the image and solving the correspondence problem, i.e. reliably finding the points in the image that belong to points in a given model. Once these correspondences are found, the local similarities can be used to assign one model out of a set of known ones to the image.

This work defines a suitable representation for models and images based on a multiresolution transform with Gabor wavelets. The properties of such transforms are discussed in detail.

Then a neural network with dynamic links and short-term activity correlations is presented that estimates these correspondences in several layers coarse-to-fine. It is formalized into a nonlinear dynamical system. Simulations show its capabilities that extend earlier systems by background invariance and faster convergence.

Finally, the central procedures of the network are put into an algorithmic form, which allows fast implementation on conventional hardware and uses the correspondences for the successful recognition of human faces out of a gallery of 83 independent of their hairstyle. This demonstrates the potential for the recognition of objects independently of the background, which was not possible with earlier systems.

Keywords: Neural network, dynamic link architecture, correspondence problem, object recognition, face recognition, coarse-to-fine strategy, wavelet transform, image representation

Preface

Ich halte dafür, daß das einzige Ziel der Wissenschaft darin besteht, die Mühseligkeit der menschlichen Existenz zu erleichtern. Wenn Wissenschaftler, eingeschüchtert durch selbstsüchtige Machthaber, sich damit begnügen, Wissen um des Wissens Willen aufzuhäufen, kann die Wissenschaft zum Krüppel gemacht werden, und eure neuen Maschinen mögen nur neue Drangsale bedeuten. Ihr mögt mit der Zeit alles entdecken was es zu entdecken gibt, und euer Fortschritt wird doch nur ein Fortschritt von der Menschheit weg sein. Die Kluft zwischen euch und ihr kann eines Tages so groß werden, daß euer Jubelschrei über irgendeine neue Errungenschaft von einem universalen Entsetzensschrei beantwortet werden könnte.

Berthold Brecht, Leben des Galilei

This thesis has been developed in the context of neural computation and will therefore present some truly interdisciplinary work. The basic ideas and techniques come from physics, the object to be modeled is a biological one, the main material tool is a computer, some concepts are most readily used by electrical engineers, and sometimes the choice of terminology reveals the author's training as a mathematician.

I take the liberty to abuse this preface for a thought which is beyond the range of science but nevertheless its consequence. My work is but a tessera of a huge worldwide effort on the general theme of understanding human cognition and modeling it in an artificial system, in other words equipping a computer with *senses*, not only sensors. If this goal will eventually be reached, the limits of machine employment may be pushed to currently inconceivable extremes.

Last winter Germany experienced the highest post-war unemployment rate, coming hand in hand with a scaring rise of fascist ideas, movements and actions. It may be doubted that this can be satisfactorily explained by the German reunion or by normal fluctuations of economic factors. More probably it reflects a transition to an industrial production which is widely accomplished by autonomous machines or very cheap labour from the underprivileged regions of our world. This historical change can only be compared to the transition from an agricultural to an industrialized society with all its pains and horrors as well as its huge improvements to human life.

So are the scientists who do this work to be blamed for dooming millions to a life without a job? Maybe yes. The easy way out of the responsibility, namely claiming that we are only doing basic research and the effects on the society are the society's problem, is blocked by the fact that our resources and our paychecks are not granted for discovering truth but for the technologies that will come out of our efforts.

But it is also possible that the historical changes will take place regardless of the wishes of individuals and that refusing to support them would only decouple our country from the mainstream of development with even worse effects on the economical and and social environment. We cannot see the future. But certainly we will have to face the problems our scientific results are likely to bring about.

As human beings we can only experience the changes that are going on with impressive speed, but in our rôle as scientists we are also culprits. It is our very own responsibility to think about the implications of our research, because others do not know enough about it. May God help us to do the right thing.

Bochum, June 1994

Acknowledgments

> *Certains auteurs, parlant de leurs ouvrages, disent:* «Mon *livre,* mon *commentaire,* mon *histoire, etc.*» *Ils sentent leurs bourgeois qui ont pignon sur rue, et toujours un «chez moi» à la bouche. Ils feraient mieux de dire:* «Notre *livre,* notre *commentaire,* notre *histoire, etc.», vu que d'ordinaire il y a plus en cela du bien d'autrui que du leur.*
>
> Blaise Pascal, Pensées

In the years that led me to writing this thesis I had the privilege to work in intellectually very fertile environments and to know people who influenced my development and my thoughts to unmeasurable extents. Therefore, expressing my gratitude to the following persons is my personal pleasure much more than only a stylistic requirement.

Christoph von der Malsburg for providing superb material working conditions as well as an intellectually challenging environment, for uncountable fruitful debates and for his never ending enthusiasm that the questions we are tackling are indeed tractable and all technical problems can be overcome.

Charles Anderson for teaching many basics of signal and image processing and for sharing his ideas and knowledge about multiresolution representations that have influenced this thesis considerably.

Wolf Singer for sharing his immense biological knowledge, for his belief in interdisciplinary research and for the wonderful working conditions at the Max-Planck-Institute for Brain Research at Frankfurt.

Günter Wunner for taking the trouble to read the dissertation and provide the second report.

Werner von Seelen for inspiring discussions and constant encouragement for me to finish my thesis in spite of falling short of some of my expectations.

Michael Neef for his tireless efforts to maintain and perfect the institute's computer system and for kind, fast and often unconventional help with the ubiquitous computer problems.

Jan Vorbrüggen for many years of excellent cooperation, for invaluable help in software problems of all kind and for many critical remarks.

Wolfgang Konen for discussions of details of the dynamic link matching and for critical comments on the manuscript.

Laurenz Wiskott for sharing his unpublished results about the running-blob matching.

Christian Kaernbach for intense discussions in spite of his distinct preference of modeling acoustic rather than visual cognition.

Peter König for many exciting discussions and valuable hints on various details of the algorithms and neurobiological relevance.

Uta Schwalm for helping to remove many administrative obstacles, for her forbearance when my scientific and personal needs contradicted the requirements of a smooth and well-ordered course of business in a university institute, and for a stylistic polish of the manuscript.

Irmgard Lenhardt-Würtz for her love and care, the patience she had with me before this thesis was finished, and for her constant challenge for me not to narrow my view of the world by doing science.

Waltraud and Heinz Würtz for making my studies possible and for bringing me up not only to be curious but also to search for answers.

Gerda Schermer for my contact with physics from early age on.

the colleagues at the Neurocomputing Institute of the Ruhr-University at Bochum, the Max-Planck-Institute for Brain Research in Frankfurt and the University of Southern California, Los Angeles for discussions, free flow of ideas, friendship and good company.

This work would have hardly been possible without the financial support given by the *German Minister of Science and Technology (BMFT)* under the grants ITR-8800-H1 (Informationsverarbeitung in neuronaler Architektur) and 01 IN 101 B/9 (Neuronale Architekturprinzipien für selbstorganisierende mobile Systeme).

This work has been typeset by the author in LaTeX, so thanks are due to the creators of TeX and LaTeX, who offer this high quality textprocessing as public domain software. All simulations have been carried out on a Sun workstation using the IDL language whose quality and power are also gratefully acknowledged.

Thanks are also due to the physics department of the Ruhr-University at Bochum for their permission to submit the dissertation in English and thereby making it available to a larger community of international researchers.

Finally, I would like to thank the readers who appreciate the interdisciplinary nature of this work and are willing to overlook the obvious shortcomings in every single discipline.

Contents

List of Figures

List of Tables

1. Introduction

> *Wie ich an anderer Stelle ausgeführt habe, halte ich es für müßig, darüber zu spekulieren, was zuerst da war, die Idee oder das Experiment. Ich hoffe, daß niemand mehr der Meinung ist, daß Theorien durch zwingende logische Schlüsse aus Protokollbüchern abgeleitet werden, eine Ansicht, die in meinen Studententagen noch sehr in Mode war. Theorien kommen zustande durch ein vom empirischen Material inspiriertes* Verstehen, *welches am besten im Anschluß an* Plato *als Zur-Deckung-Kommen von inneren Bildern mit äußeren Objekten und ihrem Verhalten zu deuten ist. Die Möglichkeit des Verstehens zeigt aufs Neue das Vorhandensein regulierender typischer Anordnungen, denen sowohl das Innen wie das Außen des Menschen unterworfen sind.*
>
> Wolfgang Pauli, Phänomen und physikalische Realität

Three reasons render the above statement by Wolfgang Pauli suited to serve as a motto for this introduction. First the theories that will be outlined here and described in detail in later chapters certainly are too speculative to allow even the idea that they have been derived in a compelling way from experimental data. Secondly, it emphasizes the importance of *matching* internal representations with objects in the outside world, and this is also the theme of this dissertation. Finally, it supports the opinion that physics will be incomplete if it ignores the interactions between the world or the experimental apparatus with the "machinery" that constitutes the observer. Again, the functioning of this machinery is our theme in the wider sense.

1.1 Object Recognition

The recognition of familiar objects in the environment is a task carried out with such ease that at first glance even the word "task" may seem inappropriate. A closer view reveals the problems behind this act of perception. The (visual) environment interacts with our brain via the distribution of light intensity on the retinae in the eyes. This distribution can vary considerably, although we are seeing "the same thing". A slight move of an object in space changes the distribution severely. The same holds for changes in lighting. Figure 1.1 shows some images of an object which "obviously" is the same all the time. This

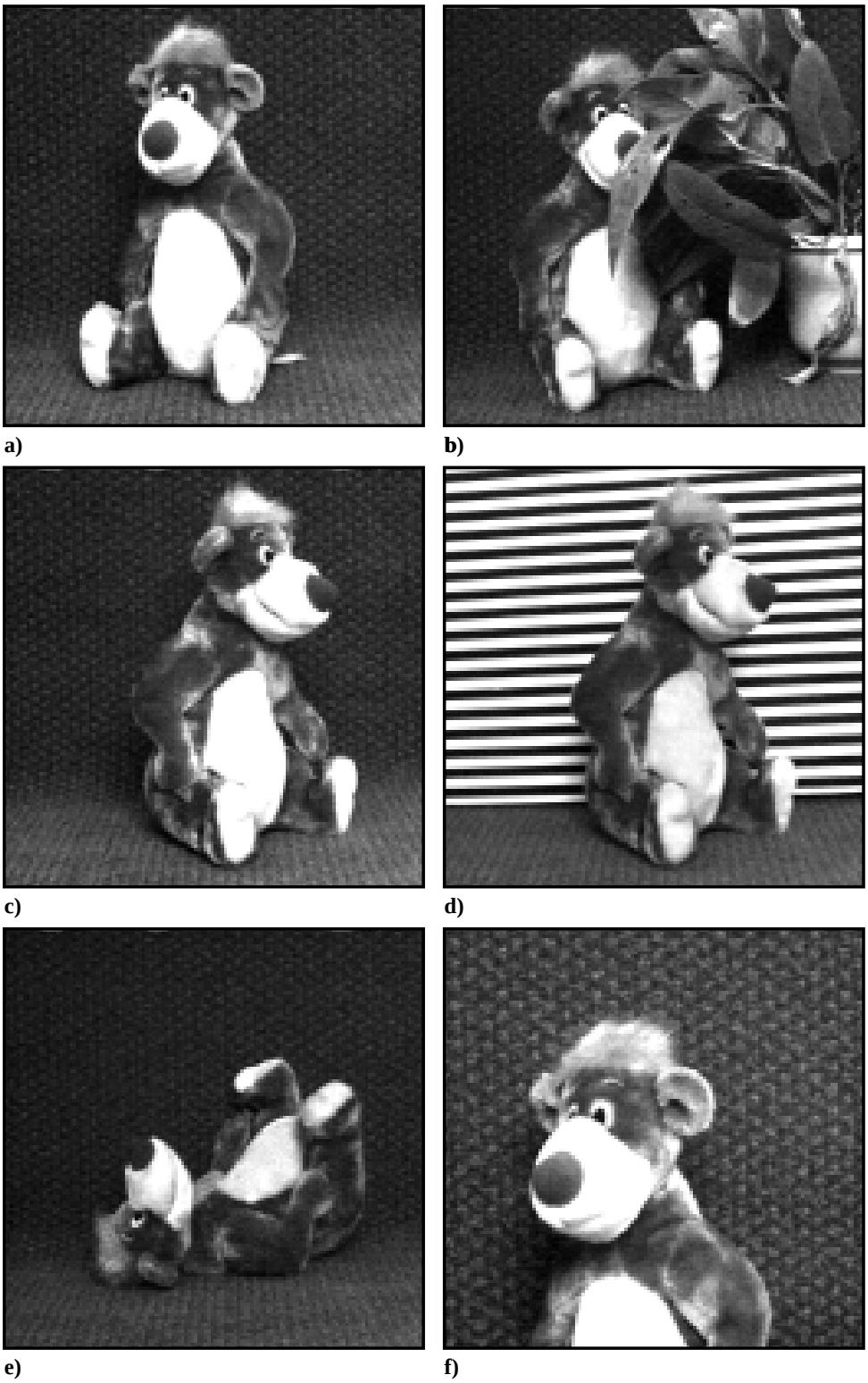

Figure 1.1: Several aspects of "obviously" the same object. The images as grey-value maps, however, seem to have little in common.

illustrates that the percept "object" is a huge *equivalence class* of images created inside the observer rather than a physical entity in the narrow sense. To make things even more complicated, such equivalence classes differ considerably between individuals and even change with time. The practiced eye can classify a respectable number of elementary particles or particle reactions from their traces in a cloud chamber, where naïve people would only see a confusing agglomeration of straight and bent lines.

The sheer size of the classes defining one object makes it impossible to store all the possible views or even a representative subset in memory. This is much more than a practical argument, because previously unseen objects can be recognized after the presentation of only one or a couple of views. If not all possible views are available the storage problem becomes less serious, but the need arises for an active process that compares a known view with the actual retinal image. This process will be called *matching*.

Given this constraint the obvious (but certainly not the only) way to assign a model class to a given object is to represent each class by a number of representative elements and find a suitable similarity function between an object presented and class representatives. This function can (in principle) be evaluated for all representatives of all classes and ought to achieve its optimum between the object presented and one representative of the correct class. Furthermore, statistics about the distribution of similarities among classes can result in estimates of how reliable the choice of class is. Single representatives will also be called *models* throughout this text.

Another complication which must be mentioned but will not be dealt with later on is the existence of *class hierarchies*. Object classes may be very rough (e.g., vehicles, trees, houses, persons, pets, streets, ...), or very fine (e.g., blue Volkswagen, red Mazda, white Mercedes Benz, my own car), or anything in between. The recognition results will be quite different depending on the choice of class division. Cognition is obviously capable of rapidly moving through various class divisions and choosing the sort of granularity that is currently appropriate.

1.1.1 The Correspondence Problem

When undergoing, e.g., perspective changes the points on the surface of an object change their absolute spatial positions as well as the positions relative to each other. If occlusion occurs, some of them will vanish completely, others will appear. In order to be able to compare the resulting images it is necessary to know which pairs of points in the images belong to the same point on the physical object, or, as the true physical object is not accessible, belong to each other. This question has turned out to be a central yet very difficult problem for computer vision. Because it is also crucial for reasonable performance it acquired the name *correspondence problem*.

> **Correspondence problem:** ***Given two images of the same object, decide which pairs of points correspond to the same point on the physical object. The fact that some points do not have corresponding partners in the other image must also be established.***

This formulation of the correspondence problem is actually the hardest possible form. Strictly speaking, it is even more general than the one given in the glossary of (Haralick and Shapiro, 1993) in the sense that the images here may be taken at different times.

For many applications additional knowledge can be exploited. Two examples are the matching of stereo images, i.e. finding correspondences between image points from the two eyes, or the computation of optical flow, where the task is to track points during movement (or from one camera frame to the next). In the first case the vertical degree of freedom is removed and the relative position of both eyes is known (see (Marr and Poggio, 1979)). For tracking during movement continuity heuristics can be applied with success, because if the frame rate is high enough, points can move only very little from frame to frame. For detailed information about tracking see, e.g., (Tölg, 1992).

Nevertheless, even in those simpler cases the problem is not solved to general satisfaction. This should be kept in mind when the results about the procedures presented here will be evaluated. Most of this thesis will be concerned with solving the correspondence problem. The general procedure will be two-stage: 1) Find a suitable representation for the image information around image points. 2) Define a similarity function for such point representations and specify a process or algorithm that finds its optimum. As stated above the actual similarity values between the correspondences found can be combined to establish a similarity function between pairs of images and thus a way to classify objects.

It can be debated if this approach is really useful. The more natural way seems to be the construction of three-dimensional representations of objects. However, psychophysical experiments that test the performance of humans to generalize from known to unknown object views suggest that cognition does indeed rely on stored views (Bülthoff and Edelman, 1992).

1.1.2 Segmentation

One possible variation in images of the same object has not yet been discussed. In figure 1.1 c) and d) the aspects of the object are identical and the lighting is very similar. However, it would not be reasonable to attempt to find correspondences between points in the respective *backgrounds*, because these are very different.

The models representative for an object therefore must be independent of the background. When an image is presented that either contains a known object which must be classified or an unknown object for which a new class must be created, this object must first be peeled out of the irrelevant background by another active process called *figure-ground- segmentation* or simply *segmentation*.

Psychophysical experiments show that there are many different clues that provide a basis for segmentation. Examples are common color, common texture, common movement, and many more. These clues are independent of the objects and therefore lead to *bottom up segmentation* , because no high-level knowledge is required.

A different way of achieving segmentation is a precise knowledge of the object and a means of finding a copy of the object representation in the image. Everything outside this copy then counts as background. This way of segmentation is called *top-down* .

Segmentation in human vision makes use of both ways of segmentation using all the clues that are available. It is clear that for unknown objects only bottom-up procedures can be employed. For a description of some segmentation procedures see, e.g. (von der Malsburg and Buhmann, 1992; Wiskott and von der Malsburg, 1993) and (Vorbrüggen, 1994). The latter also treats the problem of the incorporation of various clues that may

partly contradict each other. In (von der Malsburg and Schneider, 1986; Schneider, 1986) a neural model is presented that can solve the segmentation problem for acoustic data.

In this work we will not deal with such processes. For storing the models the background has always been removed by hand. For the matching/recognition task the image presented is segmented top-down with the stored object representatives. This will be described in sections 5.3.1 and 4.7.1.

1.1.3 Face Recognition

The object classes we will work with are human faces. This choice has several advantages.

- Faces are individual. It can easily be checked by a human if two photos show the same person. The identity of the person shown on a photo is a very natural classification of the photo. Thus the additional problems posed by class hierarchies can be easily avoided.
- They typically show internal distortions, i.e. transformations between views that cannot be modeled by a simple geometrical group. The other way around, however, sufficiently small geometrical movements in three dimensional space can be modeled as distortions.
- The surface texture of faces makes lighting effects less problematic than, e.g. objects with metallic surfaces.
- There is enough local structure (texture) to allow the sort of representations we will be using.
- The object-background problem can be studied on several levels — for a recognition independent of hairstyle, the hair has to be regarded as background, otherwise as part of the object.
- The problem is difficult yet tractable, and there is sufficient practical interest to make it more than purely academical.

Nevertheless, we will never make explicit use of the fact that we are trying to recognize human faces. The procedures can easily be applied for other object classes that either dont present the above problems or that allow extra tricks to alleviate them. We will discuss special questions at the appropriate points in the text.

One aim of this work is a closer understanding of how object recognition is *really* performed in human or animal brains. For this goal the choice of faces may not be optimal. Face recognition is one of the most important things for newborn babies, and so it may be suspected that nature has applied different procedures than for the recognition of arbitrary objects.

This suspicion is underpinned by a series of investigations that demonstrated the existence of brain cells that seem to be specialized to the detection and recognition of faces. These experiments have been carried out with monkeys who had to recognize monkey faces (Perrett et al., 1982; Rolls et al., 1985) or human faces (Baylis et al., 1985; Perrett et al., 1984). In both cases the cells were found to respond best to faces,

in (Baylis et al., 1985) even a sensitivity to the identity of the person was found. The study (Kendrick and Baldwin, 1987) finds similar behavior in sheep. For a review of the results see (Perrett et al., 1987).

The interpretation of these experiments, however, is still under debate among neurobiologists. Unfortunately, the universe of possible stimuli is much too large to really find the stimulus that a given cell prefers best. Even if this were possible and one could prove that stimulus A evokes the largest response in cell X, this would not necessarily imply that the property to be an "A-detector" would capture the "true" function of the cell. Apparently face-sensitive cells may have quite a different meaning.

Furthermore, our models will not include speculation on how the information that a certain object is recognized is represented for further processing. We will be content with having established good correspondences between image points and being able to recognize persons this way.

1.2 Is the Brain a Dynamical or a Rule-based System?

For a long time, the human mind has been metaphorically identified with the most complicated machinery that was available. While Descartes still hesitantly divided the human nature into a machine part *(res extensa)* and a thinking part *(res cogitans)* later philosophers and scientists have adopted the radical view that even the mental and spiritual expressions are simply functions of an, admittedly very complicated, machinery. Probably the first one to radically support that view was Julien Offray de La Mettrie (1748). Later the point has been pushed to various extremes, currently amounting to computer scientists like Marvin Minsky or Hans Moravec who not only believe that all mental functions *can* be carried out by machines but that they duly *should* be passed over to machines thus making the utterly imperfect human race obsolete. This work is certainly not the right platform for a discussion of the mind-body problem (and neither is the author's competence sufficient to engage on it) but one more aspect will be touched because it helps define the position of the work presented here within science.

On the background of the general inclination to use the most complicated machinery available as a metaphor for the brain the late twentieth century has two quite different paradigms to offer: the *computer* and the *nonlinear dynamical system*.

The computer metaphor for the mind has been at the heart of artificial intelligence research for several decades now. This branch of computer science has celebrated excellent success at simulating behavior which is located in the upper range of what is supposed to be intelligent. The way to success was exactly the same as is common in computer programming: Analyze the problem, formalize it (put it into rules), think up solution strategies and code them. Examples are chess playing where there is hardly any doubt that a computer will be world champion very soon, or formula manipulation, where, e.g., the integration skills of many a scientist are outclassed by commercially available programs.

These and many more achievements of artificial intelligence share one common feature: They work within an artificial world where structure and rules are completely defined or

at least definable. If human intelligence is measured by mathematical or chess playing skills, then its meaning is reduced to the capability of acting in situations that are well suited for a computer but not for a human.

There is quite a different set of tasks which to date has resisted formalization and is generally called *real-world-problems* . In the current context this does not mean problems like growth of world population, sparseness of resources or the destruction of nature. It rather stands for a computer being linked to the outside world via a video camera, a microphone or any other sensor, and devices to move around and manipulate objects. Examples for intelligent behavior would include, e.g., putting such a machine in an average European kitchen and asking it to "get a cup of coffee ready" or to "clean that mess up". If current computer systems were capable of doing this, everybody would know it from the commercials. In spite of large efforts they are not, and if our self-esteem requires that we define ourselves as superior to machines we had better talk about our everyday life instead of our written math exams.

Let us now switch our attention to the second metaphor — the dynamical system. This is the description method which is absolutely central to physics. The basic axiom is that the state of a system can in principle be described by a set of variables whose temporal evolution is governed by differential equations.

This can be applied to the brain in an obvious way: According to a simple model it consists of a number ($\sim 10^{10}$) of nerve cells that are linked to each other by a larger number of connections ($\sim 10^{14}$). The dynamics of the single nerve cell is fairly well known (with the usual caveat that biology is *always* more complicated). This together results in a dynamical system which is huge but still many orders of magnitude below Maxwell's demon in charge of a couple of liters of gas.

The discussion on how to best describe the human brain — in artificial intelligence terms, as a dynamical system, or even better in chemical, neurobiological, psychological, or theological terms — usually results in the insight that they represent quite different levels of description which can and must coexist even if their methods are very different, their results may be contradictory and their proponents often have a hard time communicating with each other.

After this peaceable compromise I will now elaborate on the view which we will take for the time of this thesis. We will try and create dynamical systems, that seem to be reasonable sketches of the biological reality and exhibit behavior that resembles cognitive capabilities. During this text this desired behavior will be solving the correspondence problem and object recognition. As there seems to be no chance to treat our dynamical systems analytically we will make extensive use of computer simulations. Once in the context of a computer simulation we will often use methods that are not direct simulations of a given system but simplifications dictated by practical requirements.

This twofold procedure is reflected in chapters 4 and 5. In chapter 4 we will stay very close to simulating a given dynamical system and show that it is capable of solving the correspondence problem. In chapter 5 we will simplify the matching procedure in a way suited for numerical evaluation and prove that it can be extended to perform realistic object recognition.

This intermediate approach of taking the best of the dynamical systems world and conventional computing techniques seems to gain many supporters from both sides. For

the artificial intelligence community the essay (Chalmers et al., 1991) may serve as an example. From the connectionist side, many of the publications of our institute may be used as a reference, e.g. (Tölg, 1992; Giefing, 1993).

Besides the practical problems posed by computer programs as models for cognition it can be argued that dynamical systems are in a very fundamental way more powerful than computer programs. Pour-El and Richards (1981) show that even a simple linear wave equation has the capability to evolve from a state which is computable by a Turing machine to a non-computable state in short time. As a Turing machine is an idealization of real computers this implies non-computability on any computer. The proof is not difficult, and the paper can be recommended for anyone interested in fundamental problems of computer science.

Also, since the 1960s intense study of nonlinear dynamical systems has revealed the awesome richness of behaviors of which they are capable. So, in believing that the human brain can be adequately modeled by a dynamical system, we are probably part of the tradition to model the mind as the most complicated structure currently known.

All this may sound like good news for the dynamical systems point-of-view, but on the other hand it is sometimes hard enough to write a computer program with desired properties. If dynamical systems have an even wider range of behavior, even more problems can be expected for the attempt to find a dynamical system that models what it is supposed to model. Indeed, in our hands dynamical systems are much harder to manipulate than computer programs. But it may be hoped that this situation will change with experience and that later generations will handle such systems with the same casualness as we are using conventional computer programs.

Beyond the visual object recognition which is our subject here it can be speculated that neuronal matching algorithms may present a key to a deeper understanding of intelligent behavior in general. Although it has been argued that rule-based systems may soon outclass all humans at, e.g., chess playing, there is no doubt that their internal structure is very different from the way any human player would proceed. Experience has equipped him or her with a number of known situations and suitable reactions. Although no two games are identical, this treasure can be efficiently searched for situations *similar* to the given one, and in many cases useful actions can be deduced. All that is needed to model this sort of intelligence is a good way to represent situations and a matching procedure. Simple as this idea may sound, its realization is far beyond the range of this thesis.

1.3 Outline of Thesis

My thesis describes processes that achieve the construction of mappings between two images. Such processes typically consist of two loosely coupled parts: a preprocessing of the image contents and a mechanism that solves the correspondence problem. They can usually be extended to yield not only a mapping but also a *similarity* between two images. Finding the similarities of one image to a stored set of objects finally leads to recognition of the corresponding object. We will try and answer the following questions:

- How can object classes be represented by one or a few examples?
- What is a reasonable measure for the similarity of a given image and a stored model?

- How can a dynamical system with the known local properties of the brain achieve classification?

- Are these principles useful for an artificial recognition system?

Chapter 2 will start with some general considerations about preprocessing, define Gabor functions, prove their optimality in the sense of phase space localization and then proceed to wavelet transforms based on these functions.

Chapter 3 explicitly describes the representation of images and models for the matching task including the parameters which will be used throughout the rest of the thesis.

Chapter 4 will give a short introduction about neural networks and then briefly outline Christoph von der Malsburg's (1981) correlation theory of brain function. On the basis of this framework a dynamical system based on neuronal properties that naturally sets up ordered mappings is described. This model will then be extended to initialize itself on a coarse resolution and then be refined to higher resolutions. The need for such a process will also be motivated.

Chapter 5 will present an algorithmic version of such a coarse-to-fine matching system which is suited to be run on sequential workstations in reasonable time. Nevertheless, the algorithm is formulated in a massively parallel manner and would achieve huge speedups on parallel hardware.

In chapter 6 the matching system from the previous chapter is used to build a hierarchical object recognition system. Its capabilities are tested under a variety of circumstances.

In chapter 7 the results will be discussed and a comparison of the performance with other methods will be attempted. This includes comparison with the FACEREC-System, an object recognition scheme in the development of which the author was involved earlier. It therefore shares some features of the system described in chapters 3 and 5. Finally, some potentially fruitful directions for further investigations will be described.

Chapter 8 contains the bibliography, and chapter 9 will give an abstract of the thesis in German and the author's curriculum vitae.

1.4 Notational Conventions and List of Symbols

Vectors are denoted by arrows. They are usually two dimensional, sometimes four dimensional (phase space variables). Feature vectors can have arbitrary finite dimension.

All quantities throughout this thesis will be free of physical units for the sake of simplicity. If necessary, the formulae can be assigned units in a consistent way, measuring, e.g., space in meters or pixels and spatial frequency in meters^{-1} or pixels^{-1}.

The important objects in this work are four dimensional, namely phase space representations and link structures between two-dimensional neuronal layers. The illustrations therefore suffer from the impossibility to display a four-dimensional object in two dimensions. Two- or at most tree-dimensional cuts must suffice.

Integrals will be understood in the sense of measure theory, i.e. the same symbol will be used for Lebesgue-integrals (where the underlying measure is the volume of measurable subsets of $\mathbf{R}^n$) and for discrete summation (where the measure of a set of points is just the number of points). The scalar product between finite-dimensional vectors, however,

will be written as $\vec{x}^{\mathsf{T}}\vec{y}$ rather than $\langle \vec{x} \mid \vec{y} \rangle$ to avoid confusion. Equality of two functions will mean that they are pointwise identical except for a set of measure 0. In the discrete cases this means pointwise identity at all points.

The following table presents the most important symbols and variables:

$\mathbf{Z}, \mathbf{R}, \mathbf{C}$	Integers, real and complex numbers
d	Dimension of the image space, usually $d = 2$
$\mathbf{U}$	Unit circle in $\mathbf{R}^2$ or the interval $(-\pi, \pi]$, respectively
$\mathbf{U}_{1/2}$	Upper half of the unit circle in $\mathbf{R}^2$ or the interval $[0, \pi)$, respectively
$\Re(c), \Im(c), \lvert c\rvert, \arg(c)$	Real part, imaginary part, modulus, and phase of a complex number
$\langle \cdot \mid \cdot \rangle, \langle \cdot \mid \cdot \rangle_{\vec{x}}$	Scalar product between functions, the variable denotes the integration variable
$\mathcal{F}$	Fourier transform $\mathcal{L}^2(\mathbf{R}^2) \to \mathcal{L}^2(\mathbf{R}^4)$
$\mathcal{F}_f$	Finite Fourier transform $\mathcal{L}^2\left(\{1, \ldots, n\}^2\right) \to \mathcal{L}^2\left(\{1, \ldots, n\}^2\right)$
$\mathbf{I}$	Image domain, typically $[0, 1) \times [0, 1)$ or a discrete subset thereof
$\mathbf{M}$	Model domain, typically a discrete subset of $[0, 1) \times [0, 1)$
$\vec{x}, \vec{y}$	Image space variables, Cartesian coordinates
$\vec{k}, \vec{\omega}$	Frequency space variables, polar coordinates
$\vec{u}$	Unit or phase space atom $\in \mathbf{R}^4$
n_{lev}, n_{dir}	Number of levels and directions in a discretized wavelet transform
$\gamma(\vec{x})$	Gabor function
$\psi_{\vec{k}}, \psi_{kp}$	Wavelet with center frequency $\vec{k}$
$\mathcal{W}$	Wavelet transform $\mathcal{L}^2(\mathbf{R}^d) \to \mathcal{L}^2(\mathbf{R}^{2d})$
$\mathbf{S}$	Sampling set in phase space $\mathbf{S} \subseteq \mathbf{R}^4$
$\mathbf{S}_f$	Sampling set for center frequencies $\mathbf{S} \subseteq \mathbf{R}^2$
$\mathcal{W}_{\mathbf{S}}$	Wavelet transform restricted to a sampling set $\mathbf{S} \subseteq \mathbf{R}^4$
$\mathcal{A}$	$= \lvert\mathcal{W}\rvert$ Absolute value of $\mathcal{W}$
$\mathcal{A}_{\mathbf{S}}$	$= \lvert\mathcal{W}_{\mathbf{S}}\rvert$ Absolute values of $\mathcal{W}_{\mathbf{S}}$
$\mathcal{P}$	$= \arg(\mathcal{W})$ Phase of $\mathcal{W}$
$\mathcal{P}_{\mathbf{S}}$	$= \arg(\mathcal{W}_{\mathbf{S}})$ Phase of $\mathcal{W}_{\mathbf{S}}$
t_a	Relative amplitude threshold
t_s	Threshold for kernels in space domain
t_f	Threshold for kernels in frequency domain
$\chi_A(\vec{x})$	Characteristic function of the set A, which is zero if $\vec{x} \notin A$ and one if $\vec{x} \in A$
$\mathcal{R}(I), \mathcal{R}(M)$	Representation of image and model
$\mathcal{K}$	Frequency level
$\mathcal{M}(M, I)$	Model-image mapping
$\mathcal{S}$	Various similarity functions
$\mathcal{S}_{glob}(M, I, \mathcal{M})$	Similarity of model and image on the basis of mapping $\mathcal{M}$
$\vec{D}(\mathcal{M})$	Distortion of a mapping $\mathcal{M}$

t_{qn}	Similarity threshold on level n
$\kappa_1, \kappa_2, \kappa_3$	Significance criteria for recognition
ξ	White noise in a dynamical system
$f * g$	Convolution, either continuous or discrete
$\kappa(\vec{x})$	Interaction kernel
$a(n), a(\vec{x})$	Activity of neuron with number n or of neuron at location $\vec{x}$
$o(n), o(\vec{x})$	Output of neuron with number n or of neuron at location $\vec{x}$
$s(n), s(\vec{x})$	Input to neuron with number n or to neuron at location $\vec{x}$
$h(\vec{x})$	Self inhibition of neuron at location $\vec{x}$
$d(\vec{x})$	Input to neuron at location $\vec{x}$
$W(n,m)$, $W(\vec{x},\vec{y})$	Short term synaptic weight between two neurons
$W_p(n,m)$, $W_p(\vec{x},\vec{y})$	Permanent synaptic weight between two neurons
$\mathrm{Corr}(\vec{x},\vec{y})$	Correlation between layer neurons in two layers
$\vartheta(x)$	Neuronal transfer function $\mathbf{R} \to [0,1]$
$\Theta(x), \delta(x)$	Heaviside and Dirac distributions
$\mathrm{supp}(f)$	Support of the function f, i.e. the set of all points where f is nonzero

2. Wavelet Preprocessing

Die ewig Unentwegten und Naiven
Ertragen freilich unsre Zweifel nicht.
Flach sei die Welt, erklären sie uns schlicht,
Und Faselei die Sage von den Tiefen.

Denn sollt es wirklich andre Dimensionen
Als die zwei guten, altvertrauten geben,
Wie könnte da ein Mensch noch sicher wohnen,
Wie könnte da ein Mensch noch sorglos leben?

Hermann Hesse, Das Glasperlenspiel

In this chapter the preprocessing of visual data will be described. In order to motivate the choice of preprocessing we will discuss wavelet transforms within a framework given by the algebraic formulation of quantum mechanics. After a short description of the formalism of position and momentum representation we will direct our attention to three different interpretations:

1. A representation is a change of variable.
2. A representation reflects the action of a group on the Hilbert space.
3. The single functionals used in a representation describe measurement devices.

2.1 Representations of a Wave Function

Quantum mechanics describe a particle at a fixed time as an element in some abstract Hilbert space. This description is unique and complete and will be denoted by Ψ.

The elements of this Hilbert space will be called *functions*, the elements of the dual *functionals*. If this terminology appears inconvenient, throughout this chapter the word "function" will be a synonym for "ket", "functional" for "bra". The latter terminology has been introduced by P. Dirac. His book (Dirac, 1967) is still very worthwhile reading, chapter III about representations covers the issues we will discuss here. Other textbooks include (Messiah, 1970) and (von Neumann, 1932). For connections between wavelet transforms and quantum mechanics that go deeper than needed for our purposes see, e.g., (Antoine, 1989; Bertrand and Bertrand, 1989; Paul and Seip, 1992; Battle, 1992).

For any practical purpose, a *representation* of the Hilbert space must be used. For a single particle without spin this is usually the space $\mathcal{L}^2(\mathbf{R}^3)$, i.e. square integrable functions on the three-dimensional Euclidean space. The scalar product between two functions will be:

$$\langle f(\vec{x}) \mid g(\vec{x}) \rangle_{\vec{x}} := \int \overline{f(\vec{x})} g(\vec{x}) \, d^d x \tag{2.1}$$

We will give the formulae for arbitrary dimension where possible throughout this chapter. Later on only $d = 2$ will be required. In the context of visual processing we will treat a stationary image as an element of $\mathcal{L}^2(\mathbf{R}^2)$ and use the terms *image space* and *frequency space*. Frequency will always mean spatial frequency unless otherwise stated.

The $\mathbf{R}^d$ underlying the function space can be interpreted as either *position space* or as *momentum space*. The first interpretation amounts to analyzing a wave function with Dirac-functionals on position space, because the value of the wave function at a location $\vec{x}_0$ is given by the scalar product with the Dirac-functional centered at $\vec{x}_0$. In other words, those Dirac-functionals are eigenfunctionals of the position operator.

$$\langle \delta(\vec{x} - \vec{x}_0) \mid \Psi \rangle_{\vec{x}} = \Psi(\vec{x}_0) \, . \tag{2.2}$$

If we use a different (but of course isomorphic) copy of $\mathcal{L}^2(\mathbf{R}^3)$ as momentum space then the momentum representation takes exactly the same form, just in momentum variables:

$$\langle \delta(\vec{\omega} - \vec{\omega}_0) \mid \Psi \rangle_{\vec{\omega}} = \Psi(\vec{\omega}_0) \, . \tag{2.3}$$

If we insist on representing Ψ in space coordinates but with eigenfunctionals of the momentum operator, then they take the form of complex exponentials:

$$\left\langle e^{i\vec{\omega}\vec{x}} \,\middle|\, \Psi(\vec{x}) \right\rangle_{\vec{x}} = \Psi(\vec{\omega}) \, . \tag{2.4}$$

Formula (2.4) can be interpreted in different ways:

1. As a definition of the Fourier transform.
2. By comparison with equation (2.3) as the spatial representation of Dirac-functionals on momentum space.
3. As a change of position variable to momentum variable.

We have seen that the choice of variable or the choice of representation is a fairly arbitrary procedure and guided only by practical considerations, while the abstract wave function Ψ is the important thing. Nevertheless, to avoid confusion and for compatibility with usual notations we will assume that Ψ is given in position space coordinates unless stated otherwise and write the Fourier transform and its inverse on $\mathcal{L}^2(\mathbf{R}^d)$ as follows

$$\begin{aligned}
\mathcal{F}(\Psi)(\vec{\omega}) &:= \left\langle e^{-i\vec{\omega}\vec{x}} \,\middle|\, \Psi(\vec{x}) \right\rangle_{\vec{x}} && (2.5)\\
&= \int \Psi(\vec{x}) e^{i\vec{\omega}\vec{x}} \, d^d x \, , && (2.6)\\
\mathcal{F}^{-1}(\Psi)(\vec{x}) &:= (2\pi)^{-d} \left\langle e^{i\vec{\omega}\vec{x}} \,\middle|\, \Psi(\vec{\omega}) \right\rangle_{\vec{\omega}} && (2.7)\\
&= (2\pi)^{-d} \int \Psi(\vec{\omega}) e^{-i\vec{\omega}\vec{x}} \, d^d \omega \, . && (2.8)
\end{aligned}$$

We see that the pair of position and momentum space representation and the Fourier transform are actually the same thing. More generally, for every representation we can define the transform that calculates it from the space representation which will be done in equation (2.9). The representations in position or momentum space are only very special choices and plenty of others can be applied when convenient. E.g., the task of finding the possible energy values of a quantum mechanical system is equivalent to transforming the space representation to a representation in terms of the eigenfunctionals of the Hamilton operator. This shows that a good choice of representation makes the desired analyses much easier.

In chapter 3 we will choose an image representation that suits our needs for the matching task. Before, we will give a general definition of a representation (or transform). Any family of functionals $\{ f_{\vec{p}} \mid \vec{p} \in \mathbf{P} \}$, where $\mathbf{P}$ is any suitable parameter set, defines a *transform* or a *representation* by means of:

$$\left(\mathcal{T}_{\mathbf{P}} \Psi\left(\vec{x}\right)\right)\left(\vec{p}\right) := \left\langle f_{\vec{p}} \mid \Psi\left(\vec{x}\right)\right\rangle_{\vec{x}} . \tag{2.9}$$

As in the case of the Fourier transform this can be interpreted as a transformation that changes the variable from space coordinates to parameter coordinates or from space representation to the representation given by the set of functionals $\{ f_{\vec{p}} \mid \vec{p} \in \mathbf{P} \}$.

Without wishing to dive deep into functional analysis the following properties of transforms are important. A transform is said to be *orthogonal* if the corresponding functionals are pairwise orthogonal:

$$\vec{p} \neq \vec{q} \implies \left\langle f_{\vec{p}} \mid f_{\vec{q}} \right\rangle = 0 \tag{2.10}$$

A transform is said to be *complete* if a function f can be recovered from its transform. In other words there must be a linear operator $\mathcal{T}^{-1}$ with the property

$$\left\| f - \mathcal{T}^{-1}\left(\mathcal{T}\left(f\right)\right) \right\| = 0 \tag{2.11}$$

For subtleties about ranges and domains of linear operators the reader should refer to standard texts about functional analysis, e.g. (Yosida, 1980).

The statement that a suitable representation can be chosen to meet the needs for the desired analysis implies that this choice will be dictated by the *physical meaning* of the various representations rather than formal reasons or, in other words, by semantic aspects. Therefore, a short glance at the physical interpretation of the functionals in the transform is in order.

Application of the spatial Dirac-functional $\delta(\vec{x} - \vec{x}_0)$ on a wave function $\Psi(\vec{x})$ in space representation gives the value of $\Psi(\vec{x}_0)$ and can be said to *measure* this value. Equally, the application of a complex exponential would give the value of $\Psi(\vec{\omega}_0)$, i.e. the value at the corresponding location in momentum space.

The use of the word measurement here is not to be confused with actual physical measurement of wave functions because it is ignoring the fact that the complex phases cannot be measured. For our purposes in image processing this does not pose a problem, because we can rely on having positive real data only.

In general, each functional in an arbitrary transform can be interpreted as a measuring device that surveys a certain subset of *phase space*, i.e. it is sensitive with a certain characteristic for a some range of space/momentum combinations. In this sense, the

eigenfunctionals of the momentum operator are "responsible" for a subspace of constant momentum and of arbitrary spatial location. As they form representations the ensembles of Dirac-functionals in position or momentum space can "build up" the whole phase space and are examples for what we will call *phase space atoms*. This word will be used as an illustrative synonym for the functionals in a transform.

2.2 Wavelet Transforms

2.2.1 Definition

For the definition of wavelet transforms and for the visual processing we will only use functionals that are themselves dual to functions in $\mathcal{L}^2(\mathbf{R}^n)$. In other words we require the functionals to have finite norm. This makes things much easier because now the distinction between functions and functionals is no longer needed, and the scalar product may be used between two functions as usual. In other words, there is a standard way to turn bras into kets and vice versa. This implies that functions and functionals decay rapidly as their variables go to infinity and are therefore *localized in phase space*.

The above definition of a transform is very general if the set of functionals has no structure. Of course, this is not the case with the position and momentum representations. All functionals $\delta(\vec{x} - \vec{x}_0)$ can be derived from a single one by a translation of position space by the amount of $\vec{x}_0$. Therefore, the set of functionals for the position representation can be regarded as the action of the group of all translations of $\mathbf{R}^3$ on the functional $\delta(\vec{x})$. Group action on a function means that the variable of the function is transformed by the group elements.

Now we will define a *wavelet transform* as a special transform generated by a single functional $\psi(\vec{x})$ which is called a *mother wavelet* and a geometrical group. Formally:

Let $\mathbf{G}$ be a group of mappings of $\mathbf{R}^n$ onto itself and $\psi(\vec{x})$ a square integrable function on $\mathbf{R}^n$. Then the set of functionals $\left\{\psi\left(g\left(\vec{x}\right)\right) \middle| g \in \mathbf{G}\right\}$ defines a wavelet transform by:

$$\left(\mathcal{W}\left(f\right)\right)\left(g\right) := \left\langle \psi\left(g(\vec{x})\right) \mid f\left(\vec{x}\right)\right\rangle . \tag{2.12}$$

Note that the parameters of the family of functionals are now the group elements. If the group itself is represented by vectors or matrices, this representation may also appear.

For reasons that will become clear in section 2.7 the mother wavelet is required to be *admissible* , i.e. to have zero integral, or a Fourier transform with a zero at $\vec{\omega} = \vec{0}$:

$$\int \psi(\vec{x})\, d^d x = \left(\mathcal{F}\left(\psi\right)\right)\left(\vec{0}\right) = 0\,. \tag{2.13}$$

It must be noted that the definition of a wavelet transform we have given here is not the most general one possible. The research on wavelets has only been going on for some years now, so canonical definitions are not available yet. Some authors, e.g., relax the requirement of having a single mother wavelet and allow two or more of them. This way, the first orthogonal wavelet transform on $\mathcal{L}^2(\mathbf{R}^2)$ was constructed (Mallat, 1988a; Mallat, 1989).

2.2.2 Properties and Taxonomy

The definition of wavelet transforms from geometrical groups immediately yields classifications of such transforms by the properties of the generating group. Especially, we will talk of *discrete* or *continuous* wavelet transforms if they are generated by a discrete or continuous group, respectively.

In general, this group will contain all the *translations* of the underlying space. These translations are conveniently expressed by means of the *convolution*, which is given here together with the most important properties:

$$\begin{aligned}(f_1 * f_2)(\vec{y}) &:= \int f_1(\vec{x}) f_2(\vec{y}-\vec{x})\, d^d x && (2.14)\\ &= \left\langle \overline{f_1(\vec{y}-\vec{x})} \mid f_2(\vec{x}) \right\rangle_{\vec{x}} && (2.15)\\ &= \mathcal{F}^{-1}\left(\mathcal{F}(f_1)\cdot\mathcal{F}(f_2)\right) . && (2.16)\end{aligned}$$

Further properties that are easily derived from the above include linearity in each argument, commutativity and associativity.

What we need is the application of a shifted functional to the function $f(\vec{x})$:

$$\begin{aligned}\mathcal{W}_{\text{transl}} f(\vec{x}) &= \langle \psi(\vec{x}-\vec{y}) \mid f(\vec{x}) \rangle_{\vec{x}} && (2.17)\\ &= \int \overline{\psi(\vec{x}-\vec{y})} f(\vec{x})\, d^2 x && (2.18)\\ &= \int \psi(\vec{y}-\vec{x}) f(\vec{x})\, d^2 x && (2.19)\\ &= \psi(\vec{x}) * f(\vec{x}) . && (2.20)\end{aligned}$$

The transition from (2.19) to (2.20) makes use of the property

$$\psi(-\vec{x}) = \overline{\psi(\vec{x})} , \tag{2.21}$$

which is true for all the wavelets we will use throughout this work. It is equivalent to the fact that the kernels have real-valued Fourier-transforms.

In the following we will use the convolution form rather than the form with the arbitrary group. If the group is $\mathbf{G}$ we will denote the factor group modulo the translations by $\overline{\mathbf{G}}$.

In our analyses we will only use continuous wavelet transforms because they are much more flexible and more suitable for describing biological facts. For computer simulations, however, the parameter groups must be discretized which leads to the apparent contradiction in terms of a *discretized continuous wavelet transform*. This is not the same thing as a discrete transform because the actual parameter set, the *sampling set* $\mathbf{S} \subseteq \mathbf{G}$, is not required to be (and usually is not) a group. To avoid the strange term we will talk about *sampled* transforms.

A very important property of a transform (representation) is its *completeness*, i.e. the possibility to *reconstruct* a function from its transform. It is important to note that this property is independent of the orthogonality of the functionals. Not even linear independence is required. It only means that there are enough of them to cover the phase space.

The convolution of an $\mathcal{L}^2$ function $f(\vec{x})$ with a second one $\kappa(\vec{x})$ (called a *kernel*) is not an invertible action. This seems surprising in the light of equation (2.16), because it would suffice to divide the product by $\mathcal{F}(\kappa)$ and apply the inverse Fourier transform to the result. This procedure, which is called *deconvolution* fails for two reasons. The first one is the occurrence of zeros in $\mathcal{F}(\kappa)$. If they are isolated the quotient attains isolated poles which may or may not be well behaved. If there are sets of measure greater than zero where $\mathcal{F}(\kappa) = 0$ it is straightforward to construct a function with nonzero norm such that its convolution with the kernel is zero. This contradicts the requirement of unique reconstruction.

The second and crucial reason is the following. $\mathcal{F}(\kappa)$ has the same (finite) norm as the κ itself. That means that it decays rapidly if $|\vec{x}|$ goes to infinity. The quotient of $\mathcal{F}(f)$ and $\mathcal{F}(\kappa)$, however, must have finite norm in order to apply the inverse Fourier transform. So all functions f for which $\mathcal{F}(f)$ does not decay rapidly enough to compensate for the growth of $\mathcal{F}(\kappa)^{-1}$ can not be reconstructed. In discretized cases, this problem leads to severe numerical instability.

Although the reconstruction from convolution with a single kernel is not possible the reconstruction from convolutions with many kernels (e.g. from a wavelet transform) becomes simple. This is because the *sum* of the Fourier transforms of all kernels need not decay if they are chosen right.

The mathematical formulation of the invertibility of a (sampled) wavelet transform is the notion of a *frame*. Given a sampling set $\mathbf{S}_f \subseteq \overline{\mathbf{G}}$ the sum of the Fourier transforms of all kernels must be bounded away from zero and from infinity. Then the quotient in the deconvolution procedure does not cause any problems.

$$0 < A \leq \int\limits_{\mathbf{S}_f \subseteq \overline{\mathbf{G}}} \left|\mathcal{F}(g(\kappa))\right|^2 dg \leq B < \infty\,. \tag{2.22}$$

This formula, like the following ones, of course relies on a suitable measure dg for the group representation used. The constants A and B are called the *frame bounds*.

If those bounds A and B exist that means that the integral over all the group elements has a finite and non-zero value at almost every point in frequency space, and thus reconstruction can be achieved by division by this integral. This leads to the reconstruction formula with *dual wavelets*, which stands for the kernels that result from the division by the integral from (2.22) in frequency space:

$$\mathcal{F}\left(f\right)\left(\vec{\omega}\right) = \int\limits_{\overline{\mathbf{G}}} \mathcal{F}\left(\left(\mathcal{W}f\right)\right)\left(\vec{\omega}, g\right) \cdot \frac{\mathcal{F}\left(g(\psi)\left(\vec{\omega}\right)\right.}{\int\limits_{\overline{\mathbf{G}}} \left|\mathcal{F}(g(\psi))(\vec{\omega})\right|^2 dg} dg\,. \tag{2.23}$$

This is indeed a reconstruction formula for f, because inverting the Fourier transform is straightforward.

If the sampling set is large enough and the kernels are well behaved it is possible that both frame bounds can be chosen to be equal. That means that the integral in inequality (2.22) is a constant. If the group $\overline{\mathbf{G}}$ is continuous and the mother wavelet is admissible (equation (2.13)) it can be shown that $\int\limits_{\overline{\mathbf{G}}} \left|\mathcal{F}(g(\psi))(\vec{\omega})\right|^2 dg$ is independent of $\vec{\omega}$, and therefore A and B can be chosen equal to that value and the reconstruction formula

simplifies to:

$$f(\vec{x}) = \left(\int\limits_{\mathbf{G}} |\mathcal{F}(g(\psi))(\vec{\omega})|^2 \, dg \right)^{-1} \int\limits_{\mathbf{G}} (\mathcal{W}f)(\vec{x}, g) * g(\psi)(\vec{x}) dg \tag{2.24}$$

Here it has been given in the space domain, because the deconvolution is trivial.

Except for the constant factor this formula looks familiar in the case that the functionals form an orthogonal basis. Indeed, continuous wavelet transforms allow the same reconstruction formula although the functionals are usually neither a basis nor pairwise orthogonal. The first factor in the formula shows the importance of the admissibility condition. It is nonzero only if the Fourier transform vanishes at $\vec{\omega} = \vec{0}$.

For a detailed discussion of the reconstruction from wavelet transforms see (Daubechies et al., 1986; Murenzi, 1989; Murenzi, 1990; Pötzsch, 1994)

2.3 The Uncertainty Principle and Gabor Functions

Although the notion of a $2n$ dimensional phase space to represent functions from $\mathcal{L}^2(\mathbf{R}^n)$ is convenient the $2n$ phase space coordinates are, of course, not independent variables. The most famous consequence of this is the *uncertainty principle* . Stripping away all physical interpretation it can be stated as follows:

> **Uncertainty principle:** ***The localization of a phase space atom is limited in that the product of the variances in position space and momentum space cannot be arbitrarily close to zero.***

All that is needed for the proof is the fact that the Fourier transform converts one representation into the other, together with the property

$$\mathcal{F}\left(\frac{\partial f}{\partial x_j}\right) = i\omega_j \mathcal{F}(f)\,. \tag{2.25}$$

The description partly follows an article by Dennis Gabor (1946), who realized that the uncertainty relation was equally important for information theory as for physics and presented all the functions that actually occupy the smallest possible phase space volume in the one-dimensional case (Gabor functions). Parts are also taken from (Böge, 1980) and (MacLennan, 1988). In the latter the uncertainty principle has been derived in arbitrary dimensionality and the optimality of the Gabor functions has been proven. Our proof here extends McLennan's proof by establishing the fact that the Gabor functions defined in (2.49) are indeed the only functions that achieve that minimum. Probably the first rigorous formulation of the uncertainty principle for vision can be found in (Daugman, 1985).

In one dimension, the space occupied by a function can be defined as its standard deviation. Analogously, we will define the mean $E_{\vec{x}}(f)$ with coordinates $E_{x_j}(f)$and the effective widths $\Delta_{x_j}(f)$ in position space:

$$E_{\vec{x}}(f) \;:=\; \frac{\int \vec{x} |f(\vec{x})|^2 \, d^d x}{\int |f(\vec{x})|^2 \, d^d x} \tag{2.26}$$

$$= \frac{\|\vec{x}f\|}{\|f\|}, \tag{2.27}$$

$$\Delta_{x_j}(f) := \sqrt{\frac{\int (x_j - E_{x_j}(f))^2 |f(\vec{x})|^2 \, d^d x}{\int |f(\vec{x})|^2 \, d^d x}}. \tag{2.28}$$

We will always assume that the function f decays fast enough for these values to be finite.

There seems to be no reasonable definition of the effective volume which would correspond to the effective width in one dimension. Nevertheless, we will define it as the product of the widths in the single Cartesian coordinates:

$$V_{\vec{x}}(f) := \prod_{j=1}^{d} \Delta_{x_j}(f). \tag{2.29}$$

This is convenient but not quite satisfactory, because, in general, it is not invariant under rotations of the coordinate system. This aesthetic flaw can probably be accepted keeping in mind that the "effective width" in itself is a fairly sloppy concept.

The same definition can, of course, be applied in momentum space by simply replacing the variable $\vec{x}$ by $\vec{\omega}$. The product of the volume of f in position space and its Fourier transform $\mathcal{F}(f)$ in momentum space will then be called the *phase space volume*:

$$V_{phase}(f) := V_{\vec{x}}(f) \cdot V_{\vec{\omega}}(\mathcal{F}(f)) \tag{2.30}$$

$$= \prod_{j=1}^{d} \Delta_{x_j}(f) \Delta_{\omega_j}(\mathcal{F}(f)). \tag{2.31}$$

A close look at the definitions shows that shifting a function in position space does neither change its volume in position or momentum space nor its mean in frequency space. This is because the shift results in multiplication of a phase factor in the momentum representation which is removed by the absolute values in equations (2.26) and (2.28). Therefore, without loss of generality we may assume that the means of f and $\mathcal{F}(f)$ vanish. Then we can derive the uncertainty relation (2.32) as follows:

$$V_{phase}(f) \stackrel{!}{\geq} 2^{-d}; \tag{2.32}$$

$$\left(\Delta_{x_j} f\right)^2 \cdot \left(\Delta_{\omega_j} \mathcal{F}(f)\right)^2 = \|x_j f(\vec{x})\|^2 \cdot \|\omega_j(\mathcal{F}(f))(\vec{\omega})\|^2 \tag{2.33}$$

$$= \|x_j f(\vec{x})\|^2 \cdot \left\| \mathcal{F}\left(\frac{\partial f}{\partial x_j}\right) \right\|^2 \tag{2.34}$$

$$= \|x_j f(\vec{x})\|^2 \cdot \left\| \frac{\partial f}{\partial x_j} \right\|^2 \tag{2.35}$$

$$\geq \left| \left\langle x_j f(\vec{x}) \,\middle|\, \frac{\partial f}{\partial x_j} \right\rangle \right|^2 \tag{2.36}$$

$$\geq \left(\Re \left\langle x_j f(\vec{x}) \,\middle|\, \frac{\partial f}{\partial x_j} \right\rangle \right)^2 \tag{2.37}$$

$$= \left(\frac{1}{2}\left(\left\langle x_j f(\vec{x}) \,\middle|\, \frac{\partial f}{\partial x_j}\right\rangle + \overline{\left\langle x_j f(\vec{x}) \,\middle|\, \frac{\partial f}{\partial x_j}\right\rangle}\right)\right)^2 \tag{2.38}$$

$$= \frac{1}{4}\left(\int \left(x_j \overline{f}\frac{\partial f}{\partial x_j} + x_j f\overline{\frac{\partial f}{\partial x_j}}\right) d^d x\right)^2 \tag{2.39}$$

$$= \frac{1}{4}\left(\int \left(\int x_j \frac{\partial\left(\overline{f}f\right)}{\partial x_j} dx_j\right) d^{d-1}x\right)^2 \tag{2.40}$$

$$= \frac{1}{4}\left(\int \left(x_j \overline{f}f\Big|_{-\infty}^{+\infty} - \int \overline{f}f dx_j\right) d^{d-1}x\right)^2 \tag{2.41}$$

$$= \frac{1}{4}\left(\int \left(\int \overline{f}f dx_j\right) d^{d-1}x\right)^2 \tag{2.42}$$

$$= \frac{1}{4}\|f\|^4 . \tag{2.43}$$

Dividing the inequality by $\|f\|^4$, applying the square root to both sides and multiplying over all dimensions yields the desired inequality (2.32).

The advantage of this derivation compared with the standard one with the commutator of Hermitian operators lies in the fact that it can easily be extended to classify *all* functions with *optimal* localization in phase space. This optimality is achieved if the inequalities (2.36) and (2.37) are both equalities. Furthermore, in order to achieve a necessary condition, possibly nonvanishing means of f and $\mathcal{F}(f)$ must be incorporated.

Inequality (2.36) (Schwartz inequality) fulfills the equal sign if and only if the two functions are linearly dependent, in other words they must be multiples of each other (the negative sign is just for convenience; so far σ_j is an arbitrary complex number):

$$x_j f(\vec{x}) = -\sigma_j \cdot \frac{\partial f}{\partial x_j} . \tag{2.44}$$

This is easily solved to:

$$f(\vec{x}) = N_j \exp\left(-\frac{{x_j}^2}{2\sigma_j^2}\right) , \tag{2.45}$$

where the integration constant turns into the normalization factor N_j. Since we need a square integrable function we can already deduce that $\Re(\sigma_j^2) > 0$.

For inequality (2.37) to fulfill the equal sign the scalar product $\left\langle x_j f \,\middle|\, \frac{\partial f}{\partial x_j}\right\rangle$ must be real. Substituting by equation (2.45) this gives

$$\left\langle x_j f \,\middle|\, \frac{\partial f}{\partial x_j}\right\rangle = \left\langle x_j N_j \exp\left(-\frac{{x_j}^2}{2\sigma_j^2}\right) \,\middle|\, -\frac{1}{\sigma_j^2} x_j N_j \exp\left(-\frac{{x_j}^2}{2\sigma_j^2}\right)\right\rangle \tag{2.46}$$

$$= -\frac{1}{2\overline{\sigma_j}^2} \cdot \left\| N_j x_j \exp\left(-\frac{{x_j}^2}{2\sigma_j^2}\right)\right\|^2 . \tag{2.47}$$

So $\overline{\sigma_j}^2$ must be real, or σ_j must be either real or imaginary. Together with $\Re(\sigma_j^2) > 0$ we conclude that all σ_j must be positive reals and the only functions that satisfy the equal

sign in the uncertainty relation for every component *and* have zero mean in position and momentum space are of the form

$$f(\vec{x}) = N \exp\left(-\sum_j \frac{{x_j}^2}{2\sigma_j^2}\right) \ , \ \sigma_j > 0\,; \tag{2.48}$$

or, simpler:

$$f(\vec{x}) = N \exp\left(-\frac{1}{2}\vec{x}^{\mathsf{T}} D \vec{x}\right) , \tag{2.49}$$

where D is a positive definite diagonal matrix with entries σ_j^{-2}. The normalization factor N is still an arbitrary complex number.

Finally, we have to incorporate the possibility of non-zero means of f and $\mathcal{F}(f)$, which will be done by a change of coordinates in position and momentum space. Let $\vec{x}_0$ be the mean of $f(\vec{x})$ and $\vec{\omega}_0$ the mean of $(\mathcal{F}(f))(\vec{\omega})$. Then the function

$$f_0(\vec{x}) = \exp\left(-i\vec{\omega}_0^{\mathsf{T}}\vec{x}\right) f(\vec{x} - \vec{x}_0) \tag{2.50}$$

has the same variances in both representations as $f(\vec{x})$ and the means in both representations are zero. The second statement is obvious. The first statement is true because a shift in one representation results in the multiplication by a (spatially varying) phase factor in the other. This is then removed by the modulus in the definition of the variance.

By applying equation (2.49) to $f_0(\vec{x})$ and solving for $f(\vec{x})$ we finally get the most general form of a function with optimal localization in phase space:

$$\gamma(\vec{x}) = N \exp\left(-\frac{1}{2}(\vec{x} - \vec{x}_0)^{\mathsf{T}} D (\vec{x} - \vec{x}_0)\right) \exp\left(-i\vec{\omega}_0^{\mathsf{T}}\vec{x}\right) , \tag{2.51}$$

with D any positive definite diagonal matrix and N an arbitrary complex number that can be adjusted in order to norm $\gamma(\vec{x})$ suitably. These functions are called *Gabor functions*. Free parameters are $\vec{x}_0$, $\vec{\omega}_0$, the normalization factor N and the positive definite diagonal matrix D whose entries are the squared inverses of the widths in each dimension. $\vec{\omega}_0$ will also be called the *center frequency*. For simplicity we introduce the following abbreviation:

$$\|\vec{x}\|_D := \vec{x}^{\mathsf{T}} D \vec{x} \tag{2.52}$$

With the help of the following formula, which holds for real positive a and arbitrary complex b and c, integrals of Gabor functions are analytically well tractable:

$$\int \exp\left(-ax^2 + bx + c\right) dx = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a} - c\right) . \tag{2.53}$$

This formula is verified by quadratic completion with some extra care concerning integration in the complex plane. It immediately yields the Fourier transform and the integral of the Gabor functions:

$$\mathcal{F}(\gamma)(\vec{\omega}) = \frac{N(2\pi)^{d/2}}{\sqrt{\det(D)}} \exp\left(-\frac{1}{2}\|\vec{\omega} - \vec{\omega}_0\|_{D^{-1}}\right) \tag{2.54}$$

$$\int \gamma(\vec{x})\, d^d x = \mathcal{F}(\gamma)(\vec{0}) = \frac{N(2\pi)^{d/2}}{\sqrt{\det(D)}} \exp\left(-\frac{1}{2}\vec{\omega}_0^{\mathsf{T}} D^{-1} \vec{\omega}_0\right) \tag{2.55}$$

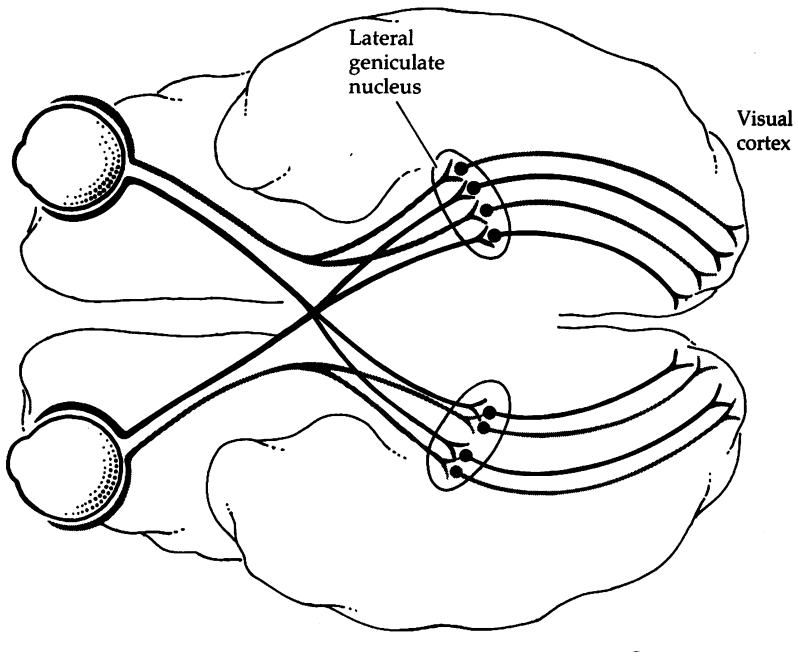

Figure 2.1: Schematic description of the early stages of visual processing. After (Nicholls et al., 1980).

2.4 Phase Space Representation in Early Vision

Now we will use the above view for the representation of visual information. This will be very close to neurobiological knowledge about early vision. It must be noted that neurophysiological data is usually not accurate enough to support more than qualitative models of neuronal properties. Heated debates about Gabor functions vs. Hermite functions or derivatives of Gaussians must be put into this perspective. Also there is currently no possibility to underpin statements about the distribution of certain properties, because neurophysiological analysis is restricted to small numbers of cells.

On the retinae in our eyes visual information about the outside world is represented by the light intensity falling onto photoreceptors with sensitivity maxima at three different wavelengths. Instead of light intensity distributions we will use the psychological (biological) term *stimulus*. Correspondingly, the activity of a detector cell will also be called its *response*.

We will simplify the situation by ignoring the different colors as well as binocularity and assuming an intensity distribution on *one* densely sampled and infinitely large sheet of point receptors — in other words we will describe the incoming light intensity as a function from $\mathcal{L}^2(\mathbf{R}^2)$. All temporal properties of the stimulus will be ignored.

Some clarification is required when we are talking below about "cells". Nerve cells or neurons in the brain are usually meant to have a positive activity value (for a closer description of neuronal properties see section 4.1). The transforms we will introduce, will regularly produce negative or even complex values. These are quantities that describe

small local assemblies of cells, rather than a single cell. In the simplest case, positive or negative values could be accomplished by a pair of cells, one of which has the receptive field corresponding to the positive part of the functional, the other to the negative one. Their activities are then transmitted to following neurons in an excitatory (positive part) or inhibitory fashion (negative part). Similarly, a complex-valued functional corresponds to (at least) four cells, one pair for the real part, and one for the imaginary part.

The first stages of processing in the visual system can be adequately described by transforms in the above sense. The *retinal ganglion cells*, which represent the first stage of visual processing, have circular symmetric functionals which can be adequately modeled, up to a normalization factor, with a difference of Gaussians:

$$\rho(\vec{x}) = \sigma_- \cdot \exp\left(-\frac{\vec{x}^\mathsf{T}\vec{x}}{2\sigma_+^2}\right) - \sigma_+ \cdot \exp\left(-\frac{\vec{x}^\mathsf{T}\vec{x}}{2\sigma_-^2}\right) \tag{2.56}$$

The coefficients of the exponentials are adjusted such that $\int \rho(\vec{x})\, d^d x$ vanishes. If $\sigma_+ < \sigma_-$ this functional has the form of a positive peak surrounded by a negative annulus. Such cells are therefore called "on-center-off-surround". For $\sigma_+ > \sigma_-$ the opposite kind of cells results. This model dates back to (Rodieck, 1965).

The retinal ganglion cells connect to the *lateral geniculate nucleus* which serves as a relay station and is concerned with merging information from both eyes, control of the visual pathway during wake and sleep phases, and feed-back control by descending cortical connections. As we are neither concerned with stereo vision nor with time-variant signals the influence of this structure on the neuronal signals will be ignored.

The next processing stage is the *primary visual cortex*. Here the situation starts getting complicated. There is a variety of cells with different properties. For our models we only pick the simplest of them, which have been named *simple cells* by (Hubel and Wiesel, 1962). Those simple cells have the following properties:

1. **Linearity:** The response to excitatory stimuli is linear in good approximation.

2. **Admissibility:** There is no response to a spatially constant illumination.

3. **Localization in image space:** Each cell has a certain area on the retina, called its *receptive field*, outside of which the stimulus does not affect the response.

4. **Orientation selectivity:** Cells respond best to light bars and edges and their response depends on the orientation of the edge or bar in the image space. Each cell has a preferred orientation and is insensitive to bars or edges perpendicular to this orientation.

5. **Localization in frequency space** When stimulated with sine-shaped intensity distributions they show the same preferred direction as with the bars and edges. Each cell has a preferred spatial frequency. If the frequency of the sine differs too much from it there is no response.

The first property means that linear functionals applied to the stimulus data are a good model for the responses of these cells. The second one requires the functionals to vanish at $\vec{\omega} = \vec{0}$ in frequency space. We will deal with this closely in section 2.5.1. Property 3 is the

localization in image space, 4 and 5 together mean localization in frequency space, where 4 requires a tuning for the orientation, 5 a tuning for the length of the center frequency.

The applicability of Gabor functions for these functionals has been proposed by (Daugman, 1985). In (Marčelja, 1980; Jones and Palmer, 1987; de Valois and de Valois, 1990) precise measurements are presented that prove this within the possible accuracy. In (Pollen and Ronner, 1981) experiments are described that find cells with odd and even symmetries and otherwise very similar parameters in close neighborhood. These underpin the usefulness of the complex-valued Gabor-functions, because their real and imaginary parts do have these symmetries.

This model is harshly criticized by (Stork and Wilson, 1990). The most serious criticism is, in the author's opinion, the fact that the real parts of Gabor functions do respond to constant illumination. We will account for this problem by modifying the Gabor functions in section 2.5.1. The other arguments do not seem very convincing.

In (Hubel and Wiesel, 1974) the distribution of simple cells with different parameters such as receptive field size and orientation tuning has been examined. They have coined the term *hypercolumn* for the set of simple cells whose receptive fields have the same sizes and are centered on the same point in image space.

In our view of simple cells as phase space atoms the hypercolumns present a simple example of small assemblies, or *phase space molecules*, to use the same metaphor. We will use various forms of such molecules for or models in chapters 4 and 5.

If the responses of all simple cells in a hypercolumn are arranged to a vector this can be viewed as a *feature vector*. It will turn out to be very useful to use such feature vectors for matching instead of the responses of the cells.

2.5 Turning Gabor Functions into a Wavelet Transform

The Gabor functions introduced in section 2.3 have been shown to be optimal in terms of phase space localization and to present a good model for a class of neurons in early vision. In order to construct a wavelet transform, two things are left to do. First, the admissibility condition as well as the experimental data require that the integral over the kernels (or, equivalently, over the mother wavelet) must be zero. A fixed function must be picked as mother wavelet, and the group that generates the transform must be specified.

2.5.1 Admissibility Correction

Equation (2.54) shows that the Fourier transform of a Gabor function is a Gaussian centered in its center frequency with widths defined by D^{-1}. Figure 2.2 a) shows a section through the line through the origin in the direction $\vec{\omega}_0$.

We will present two different methods of making the the Gabor functions admissible, i.e. modifying them for their integral to vanish. Both work in the frequency domain. The simplest approach to just define $\mathcal{F}(\psi)(\vec{0}) = 0$ is not recommendable because this would amount to subtracting a δ-functional, and we wish to use functions from $\mathcal{L}^2(\mathbf{R}^2)$. For the following discussion we will assume $\vec{x}_0 = \vec{0}$ without loss of generality.

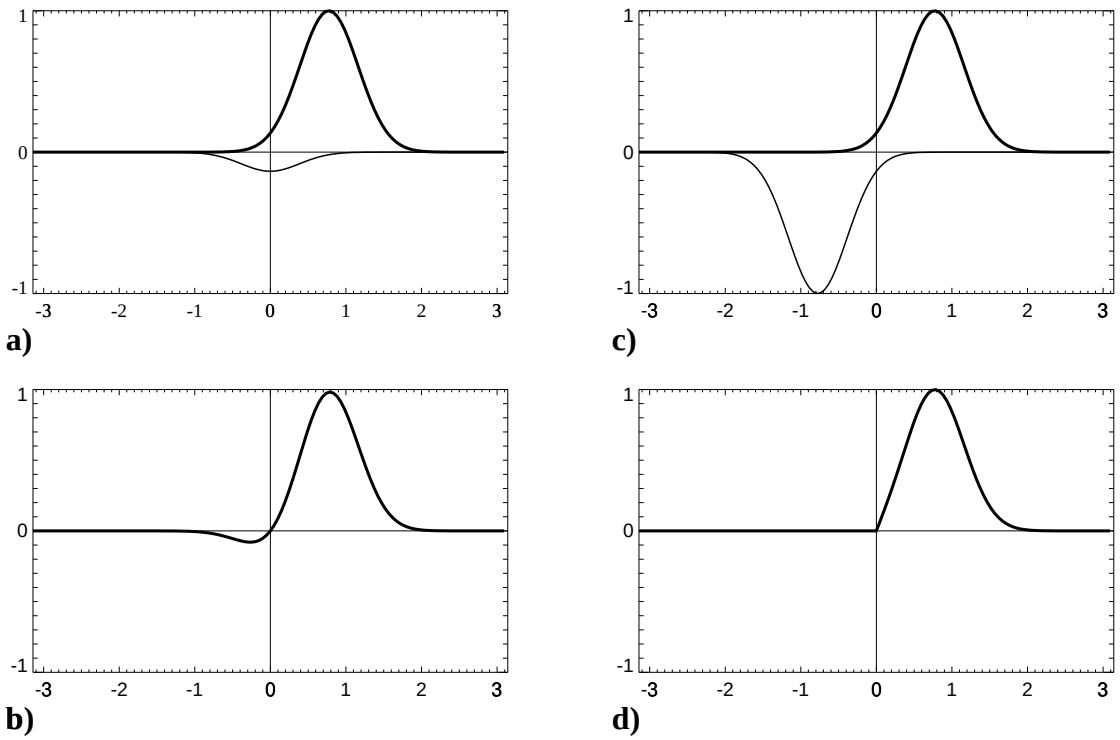

Figure 2.2: Methods for making the Gabor kernels admissible. The fat graphs in **a)** and **c)** show a section through the the Gabor kernel for $k = 0.775$ in frequency space. The value at frequency 0 is positive, therefore the kernel is not admissible. In **a)** a Gaussian of the same width and appropriate height is subtracted to yield the admissible kernel shown in **b)**. In **c)** the kernel is inflected at the origin, subtracted from the original one and all values at negative frequencies are set to zero. The result is the kernel in **d)**.

The first one follows (Murenzi, 1989; Murenzi, 1990) and subtracts a Gaussian of the same widths centered at zero. This yields:

$$\mathcal{F}(\psi_{\mathrm{M}}) = \frac{N(2\pi)^{d/2}}{\sqrt{\det D}} \left[\exp\left(-\frac{1}{2} \left\| \vec{\omega} - \vec{\omega}_0 \right\|_{D^{-1}} \right) - \exp\left(-\frac{1}{2} \left(\left\| \vec{\omega}_0 \right\|_{D^{-1}} + \left\| \vec{\omega} \right\|_{D^{-1}} \right) \right) \right] , \tag{2.57}$$

and

$$\psi_{\mathrm{M}} = n \exp\left(-\frac{1}{2} \left\| \vec{x} \right\|_{D} \right) \left[\exp\left(-i \vec{\omega}_0^{\mathsf{T}} \vec{x} \right) - \exp\left(-\frac{1}{2} \left\| \vec{\omega}_0 \right\|_{D^{-1}} \right) \right] . \tag{2.58}$$

This way of assuring admissibility is convenient for analytical considerations, because the subtraction of a Gaussian keeps the form simple. A big disadvantage is the fact that the Fourier transform has negative values, even if the normalization factor was positive (see figure 2.2 b)).

The second method is inspired by the one-dimensional Hilbert transform , and can therefore, with some phantasy, be regarded as its multidimensional generalization. A given kernel with a well-defined center frequency $\vec{\omega}_0$ is reflected about the line orthogonal to $\vec{\omega}_0$ through the origin and subtracted from the original kernel. Thus the frequency plane

is divided into two halfplanes separated by that line. Then the values in the halfplane not containing $\vec{\omega}_0$ are set to zero.

If the given kernel is nonnegative and continuous and decays monotonously with the distance from $\vec{\omega}_0$ (as is the case for the Gabor functions) this leads to a nonnegative continuous kernel which is zero in the origin. Differentiability, however, is no longer guaranteed.

$$\mathcal{F}(\psi_{\mathrm{H}})(\vec{\omega}) = \begin{cases} \exp\left(-\frac{1}{2}\left\|\vec{\omega}-\vec{\omega}_0\right\|_{D^{-1}}\right) - \exp\left(-\frac{1}{2}\left\|A\vec{\omega}-\vec{\omega}_0)\right\|_{D^{-1}}\right) & : \quad \vec{\omega}^{\mathsf{T}}\vec{\omega}_0 > 0 \\ 0 & : \quad \text{otherwise} \end{cases}, \tag{2.59}$$

where A denotes the reflection about the mentioned axis:

$$A\vec{\omega} = \vec{\omega} - 2\frac{\vec{\omega}^{\mathsf{T}}\vec{\omega}_0}{\vec{\omega}_0^{\mathsf{T}}\vec{\omega}_0}\vec{\omega}_0 \tag{2.60}$$

The formulae indeed work in arbitrary dimension. In the considerations above the line orthogonal to $\vec{\omega}_0$ must be replaced by the orthogonal complement of $\vec{\omega}_0$.

ψ_{H} itself does not have a nice analytic form. It contains error integrals, which are notoriously hard to treat analytically. But for numerical calculation it is very convenient. Its Fourier transform is not only real but strictly nonnegative.

It must be noted that neither method can work if $\vec{\omega}_0 = \vec{0}$. If the Gaussian kernel is centered at zero, removing the integral must result in drastic qualitative changes to the form. However, we are not interested in this case.

One drawback behind both methods is the fact that they cannot be generalized to arbitrary kernels. They rely on the kernels to be localized in frequency space around a point which is in a certain distance from the origin. For kernels like difference of Gaussians or Laplacian of Gaussian that are used to describe retinal ganglion cells, the vanishing of the integral must be enforced differently (see, e.g. (2.56)).

In the light of the optimality of the Gabor functions the question arises which functions are optimal in the sense of phase space localization under the extra condition that they are admissible. The author has not been able to find the answer, but this has no practical consequences for this work.

2.5.2 Choice of Wavelet Functionals

As mentioned above the neurophysiological data are not good enough to decide between various mathematical forms of functionals to describe simple cell responses. Some authors prefer the product of a polynomial and a Gaussian, or, equivalently, various spatial derivatives of Gaussians (see, e.g., (Lindeberg, 1994)). Atick and Redlich (1990) propose a neural wiring scheme that builds functionals for simple cells from DOG-models of retinal ganglion cells. This results in Jacobi-ϑ-functions, which as they state, for realistic parameter values look quite similar to Gabor functions.

We will use Gabor functions modified to fulfill admissibility by either (2.58) or (2.59) throughout the rest of the thesis. For all practical purposes, there is not much difference between the results of both methods. Because the extra complication does not improve our results we will choose *isotropic* Gaussians in the Gabor functions. This means that

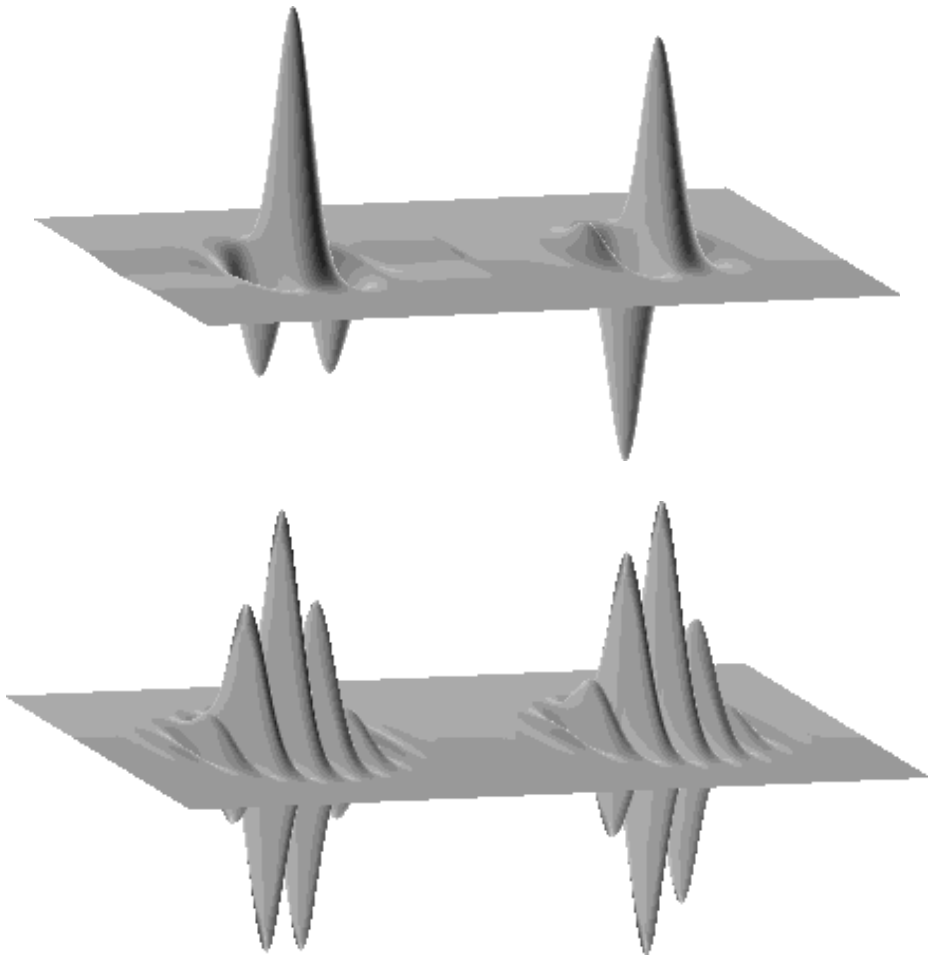

Figure 2.3: The form of the admissible Gabor kernels. The upper figure shows the Gabor kernel with $\sigma = 2$ as it is found as receptive field profile in the visual cortex and is used for visual preprocessing in this work. On the left the real part is shown, on the right the imaginary part. The lower figure shows the kernel for $\sigma = 2\pi$.

---

the diagonal matrix D must be a multiple of the unit matrix and we will replace it by σ^{-2}. For mother wavelet we choose the function centered at $\vec{\omega}_0 = \binom{1}{0}$ (For the apparent inconsistency of physical units see section 1.4):

$$\psi_0 = \exp\left(-\frac{\vec{x}^{\mathsf{T}}\vec{x}}{2\sigma^2}\right)\left[\exp\left(-ix_1\right) - \exp\left(\sigma^{-2}\right)\right] \tag{2.61}$$

The geometrical group that creates the functionals for our wavelet transform will be the group of all translations, rotations and scalings of the image plane, which is called $IG(2)$. It can be represented by $(s, \varphi, \vec{y}) \in \mathbf{R}^+ \times \mathbf{U} \times \mathbf{R}^2$, where the first parameter s stands for the scaling factor, the second for the rotation angle and the third for the translation vector.

We will derive a more compact representation of the wavelet transform generated, which is also very well suited for numerical implementation. The factor group of $IG(2)$

modulo translations consists of all combinations of rotations and scaling. These are uniquely determined by the action of their transposed matrix on the unit vector $\binom{1}{0}$. We need the transposed matrix, because we wish to represent the group action by the *column vector* $\vec{k}$:

$$\vec{k} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}, \tag{2.62}$$

corresponding to the group element

$$g = \begin{pmatrix} k_1 & k_2 \\ -k_2 & k_1 \end{pmatrix}. \tag{2.63}$$

Applying this group element to ψ_0 yields:

$$\psi_{\vec{k}}(\vec{x}) := \psi_0\left(g(\vec{x})\right) = N \exp\left(-\frac{\vec{k}^{\mathsf{T}}\vec{k}\vec{x}^{\mathsf{T}}\vec{x}}{2\sigma^2}\right)\left[\exp\left(-i\vec{k}^{\mathsf{T}}\vec{x}\right) - \exp\left(\sigma^{-2}\right)\right] \tag{2.64}$$

We will now apply identity (2.20) and replace the action of translations by convolution. Transforming a function $f(\vec{x})$ with the wavelet transform generated by the mother wavelet $\psi_0(\vec{x})$ and the group $IG(2)$ is then equivalent to *convolution* of f with all kernels of the form (2.64) with $\vec{k} \in \mathbf{R}^2 \backslash \{\vec{0}\}$.

$$(\mathcal{W}f)(\vec{x}, \vec{k}) = I(\vec{y}) * \psi_{\vec{k}}(\vec{y}) \tag{2.65}$$

The origin for $\vec{x}$ is arbitrary, the origin for $\vec{k}$ stands out by the fact that it is not part of the parameter set. Therefore, we will usually measure $\vec{k}$ in polar coordinates.

This very convenient coding for the different kernels generated by the group action can also be carried out for nonisotropic Gaussians, i.e., $D \neq \sigma^{-2}$. The formulae look a bit more difficult and we do not need them for our purposes.

2.5.3 Choice of Normalization Factors

The rule that the various functionals are derived from a single one by a geometrical group already fixes the normalization factors. Deviating from this we will choose the normalization factors depending on $\vec{k}$ in order to have a transform which is better adapted to the properties of visual data. This will not change the properties of the transform besides leading to a slightly modified reconstruction formula.

Natural images have far more specific properties than just producing $\mathcal{L}^2$-functions on the retinae. Unfortunately, it is unknown how the concept of a natural image can be formalized. This may be a major cause of trouble in computer vision, because visual algorithms implemented in the brain are probably optimized for a natural visual environment. In the absence of a mathematical notion for this environment finding vision algorithms that could mimic the cognitive capabilities of living beings is very hard. In any case, it seems a wise idea to incorporate the available knowledge, sparse as it may be.

In the work (Field, 1987) the spectra of various natural images are analyzed and it is found that the amplitudes decay like $1/|\omega|$. In the images that will be used in this thesis we found roughly the same behavior. We will therefore modify our transform such that

this dependency cancels out and we get comparable response amplitudes for all values of $|\vec{k}|$. This means that the Fourier transforms of our kernels must have norms proportional to $|\vec{k}|$. This is achieved by adjusting their normalization factors to be independent of k, or for simplicity, equal to 1:

$$\left(\mathcal{F}\psi_{\vec{k}}\right)(\vec{\omega}) = \exp\left(-\frac{\sigma^2\left(\vec{\omega}-\vec{k}\right)^2}{2\vec{k}^2}\right) - \exp\left(-\frac{\sigma^2\left(\vec{\omega}^2+\vec{k}^2\right)}{2\vec{k}^2}\right) . \tag{2.66}$$

Form this we derive the normalization in image space:

$$\psi_{\vec{k}}(\vec{x}) = \frac{\vec{k}^2}{\sigma^2}\exp\left(-\frac{\vec{k}^2\vec{x}^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}\vec{x}\right) - \exp\left(-\sigma^2/2\right)\right] . \tag{2.67}$$

2.6 Sampling Issues For Wavelet Transforms

Until now we have always considered continuous functions on $\mathbf{R}^d$. For the modeling of the responses of nerve cells this is only an idealization motivated by the fact that mathematics without infinity is no fun. Once we have to simulate our transforms on a digital computer we are forced to discretize them again. Therefore, some reflection about discretization or *sampling* is necessary.

2.6.1 The Sampling Theorem

The main theorem about the required sampling density of a function is the *sampling theorem*. It is best known for one-dimensional (electric or acoustic) signals and usually only formulated for those. The extension to arbitrary dimension, however, is straightforward, and we will give the general form here. The question to be answered is: "How many samples does it take to represent a given function?". Obviously, this is an impossible task in the general case. According to general usage, $\mathcal{L}^2$-functions are identical if they differ only on a set of measure zero, which any finite or discrete sampling set will certainly be. Additional constraints are needed. Continuity or differentiability are of no use, because the areas where they have any effect on the function (i.e. the convergence radii of their Taylor series) can be made arbitrarily small. On the other hand, it would be possible (in the two-dimensional case) to interpret the image plane as a complex plane and require the functions to be analytic. Then the reconstruction from a discrete set of points would be guaranteed by the theory of complex functions (Conway, 1978). However, this would constrain images in unnatural ways.

The possibility of local manipulations of functions that render the differentiability constraints useless have one common property: If one wishes to construct counterexamples for a sampling set of a certain density the resulting functions attain very high frequencies. The correct constraint is therefore to exclude this possibility by prohibiting high frequencies. Formalizing this results in the following definition and a theorem about the required sampling density:

> A function from $\mathcal{L}^2(\mathbf{R}^n)$ is called *band limited* by the frequencies $\vec{\eta}$ if $\mathcal{F}(f)(\vec{\omega}) = 0$ outside the hyperrectangle $[-\eta_1, \eta_1] \times \ldots \times [-\eta_d, \eta_d]$. The frequencies η_j are called *Nyquist-frequencies*.

Sampling theorem: ***A band limited function is uniquely determined by the sequence of values*** $f\left(\frac{k_1}{2\eta_1}, \cdots, \frac{k_d}{2\eta_d}\right), \vec{k} \in \mathbf{Z}^d$.

The proof can be found with varying degree of rigor in signal processing textbooks, e.g. (Kunt, 1980; Blahut, 1988). It rests on interpolation with sinc-functions:

$$f(x) = \sum_{k=-\infty}^{\infty} f\left(\frac{k}{2\eta}\right) \frac{\sin(2\pi(x-k))}{2\pi(x-k)} \tag{2.68}$$

Some care is required with the properties of the functions involved. If the closed intervals are used in the definition of band limited, functions from outside $\mathcal{L}^2(\mathbf{R}^n)$ can not be admitted.

2.6.2 Sampling of Wavelet Transform

One reason that the Fourier transform is popular not only for theoretical considerations but also for practical computations is the fact that it can be computed efficiently. At first glance, the Fourier transform as well as the convolution look like general linear transformations which, after discretization, turn into a matrix-vector multiplication, where the evaluation of each functional is a scalar product. This needs $O(n)$ operations, where n is the dimension of the data. So a transform needs, in general, $O(n \cdot m)$ operations to execute, where m is the result dimension. Fourier transform as well as convolution, however, have symmetries that allow a faster computation, and their computational complexity is only $O(n \log n)$. The details are very well described in (Press et al., 1988), an in-depth treatment can be found in (Nussbaumer, 1982).

This is the reason why we formulated the wavelet transform as a convolution in (2.65). But it also dictates part of the choice of the sampling set. In principle, the sampling could be arbitrary in the four-dimensional space spanned by the two spatial and the two frequency directions. But we can only take advantage of the convolution form if many sampling points lie on planes of constant center frequency $\vec{k}$. Most fast convolution algorithms (particularly the one using the FFT, which we will always use) require a regular grid for each of those planes. For the complete sampling set we have to specify the sampling of the *magnitude* of center frequencies, the sampling of their *directions*, and the spatial sampling of the *translations* or, equivalently, of the single convolutions belonging to each center frequency.

2.6.3 An Efficient and Intuitive Way to Sample Convolutions

In this section we will discuss three different methods to sample convolution results. For each method we will start with an image I sampled on a rectangular grid. To take full advantage of the speed of the FFT the number of sampling points usually is chosen as a power of two in each dimension, but our considerations are valid for all numbers.

The simplest form to arrive at a sampled convolution result consists in combining equations (2.16) and (2.65) and replacing the continuous Fourier transform by the finite FFT $\mathcal{F}_f$.

$$I * \psi_{\vec{k}} = \mathcal{F}_f^{-1}\left(\mathcal{F}_f(I) \cdot \mathcal{F}_f(\psi_{\vec{k}})\right) . \tag{2.69}$$

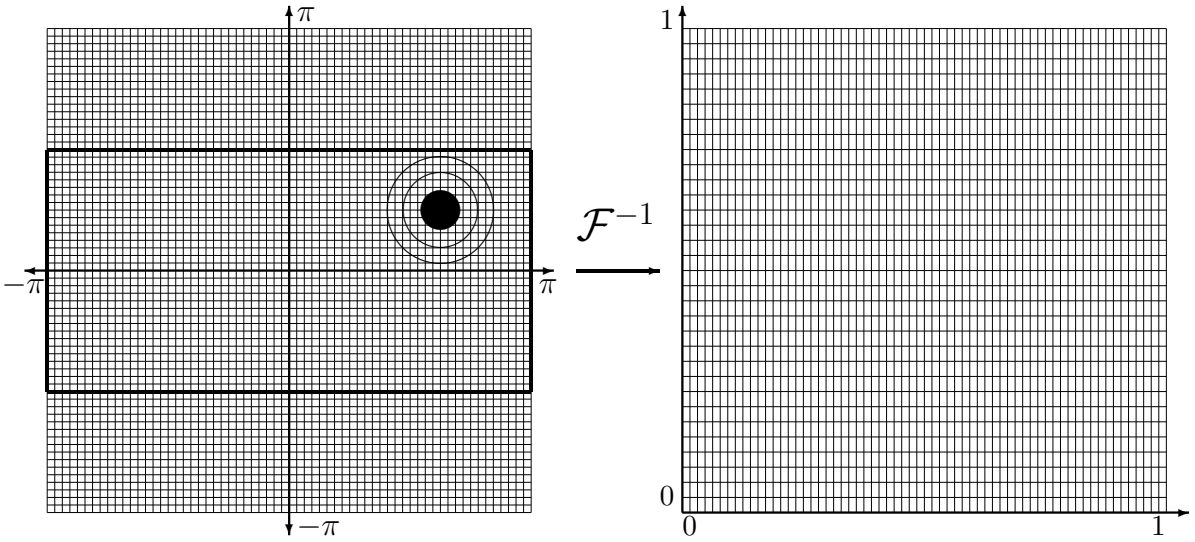

Figure 2.4: Nyquist-sampling in frequency space. The kernel in frequency space is symbolized by the concentric circles. Outside the largest circle its value is below the threshold t_f. The thresholded kernel is band limited by the dark rectangle. Inverse Fourier transform of this rectangle leads to a different sampling in image space.

As the FFT does not change the dimension of the data this leads to the same sampling for image and convolved image. This method will be called *full sampling* .

The wavelet kernels are localized in frequency space, and therefore one can make use of the sampling theorem. Strictly speaking, $I * \psi_{\vec{k}}$ is not band limited, because the support of $\mathcal{F}(\psi_{\vec{k}})$ is the whole plane. However, they decay fast enough to be approximated very well by a band limited function. In other words, we will fix a threshold t_f and approximate $\mathcal{F}(\psi_{\vec{k}})$ by the modified kernels:

$$\mathcal{F}(\psi^{bl}_{\vec{k}}) = \begin{cases} \mathcal{F}(\psi_{\vec{k}}) & : \quad \mathcal{F}(\psi_{\vec{k}}) > t_f \\ 0 & : \quad \text{otherwise} \end{cases} \quad . \tag{2.70}$$

Now we consider the smallest rectangle that is centered in the origin and contains the support of $\mathcal{F}(I) \cdot \mathcal{F}(\psi^{bl}_{\vec{k}})$. No information is lost if the values outside this rectangle are discarded. The sampling strategy now simply consists in taking the inverse FFT *of the data in this rectangle*. This will automatically lead to a subsampled convolution result without loss of information. We will call this method *Nyquist sampling*.

The full sampling can easily be recovered from this by inverting all the steps: Take the FFT of the rectangle, pad the resulting rectangle with zeroes up to the desired size (resolution), and take the inverse FFT. This is accurate up to numerical errors in the FFT (which are small). Of course, this method can also be used to achieve higher resolution than the original image had. This constitutes a method for interpolation with band limited functions. This sampling procedure is nothing more than an illustrative way to

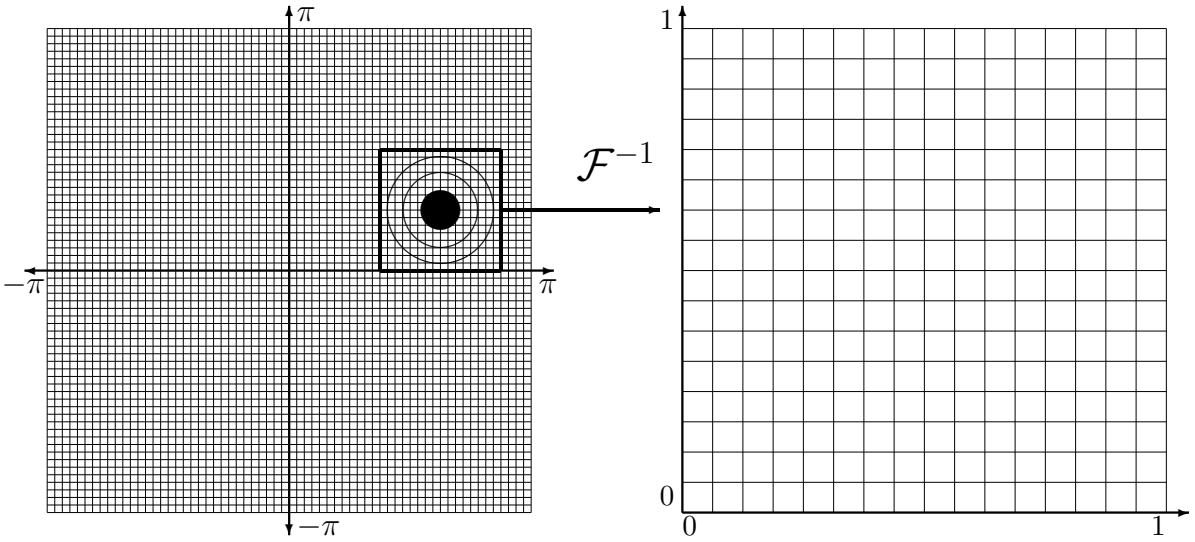

Figure 2.5: Sparse sampling in frequency space. The kernel in frequency space is symbolized by the concentric circles. Outside the largest circle its value is below the threshold t_f. Values of the transform which lie outside the dark square are zero. Therefore, it suffices to apply the inverse Fourier transform to this square only, which leads to a much sparser sampling than the one shown in figure 2.4. The fact that the center of the square is not at $\omega = 0$ is accounted for by subsequent multiplication with the wave corresponding to that shift.

---

apply the sampling theorem: The required sampling density is 2π divided by the Nyquist frequency. See figure 2.4 for a graphical illustration.

The reason why the full sampling can be recovered from the Nyquist sampling is that chopping off the zeroes is an invertible action. A second look at figure 2.4 reveals that inside the rectangle defined by the Nyquist-frequencies there are still many zeroes left. If the support of the kernel is limited to *some* rectangle defined by $\vec{\eta}_1$ (lower left corner) and $\vec{\eta}_2$ (upper right corner) the sampling theorem can be applied after shifting the origin into the center of the rectangle, i.e.

$$\vec{\eta}_c = \frac{1}{2}\left(\vec{\eta}_1 + \vec{\eta}_2\right) . \tag{2.71}$$

Nyquist sampling can now be applied to this shifted function. Afterwards, the shift can be reversed by multiplication of the result by $\exp(i\vec{\eta}_c^{\mathsf{T}}\vec{x})$. If only the amplitudes of the result are required, this multiplication can be omitted because it only changes the phase. This sampling strategy will be called *sparse sampling*. It is clear from figure 2.5 that this leads to a much sparser representation. Numerical values for the number of units resulting from each strategy can be found in section 3.6.1.

The sparse and Nyquist sampling strategies lead to a representation with few entries for low frequencies and many for high ones. If those (planar) representations for the

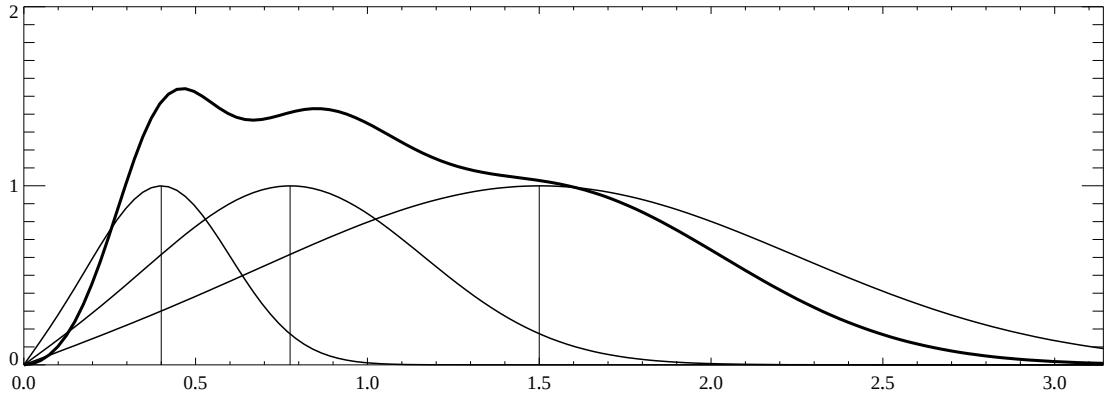

Figure 2.6: Frequency space covering of sampled transform. The Gaussian-like curves show a section through the kernels $\mathcal{F}(\psi_{hilb})$ in frequency space at the center frequencies $k = 0.4$, $k = 0.775$, and $k = 1.5$. The fat curve is the squared sum of the three kernels. This is the relevant curve for reconstruction. In the ideal case of infinite frequency space sampling it would be a constant.

---

single frequencies are stacked on top of each other (highest frequencies at the bottom) this resembles a pyramid. *Pyramid representations* have been introduced by Burt and Adelson (1983) and form a standard technique in computer vision (Cantoni and Levialdi, 1986). Our presentation here is special in the sense that the pyramid structure is directly derived from the form of the kernels, and the possibility of an arbitrary number of directions of center frequencies is included.

2.7 Reconstruction From Sampled Wavelet Transform

For reconstruction of the image from the wavelet transform a discretized version of the continuous reconstruction formula (2.24) can be used. What needs to be specified is the measure dg which must be some function of d^2k or better $d|k|d\varphi$, because our frequency sampling will be in polar coordinates. First, our wavelets are not normalized, their norm is proportional to $|\vec{k}|$. So, equation (2.24) must not be applied for $\psi_{\vec{k}}$ but for $|\vec{k}|^1\psi_{\vec{k}}$, instead. Second, the sampling of the length of $\vec{k}$ is logarithmic, so the area element d^2k is not $|k|\,d|k|\,d\varphi$, as it would usually be, but $|k|^2\,d|k|\,d\varphi$. Inserting all this into (2.24) yields:

$$I(\vec{x}) = \int_{\mathbb{S}} \mathcal{W}(\vec{x},\vec{k}) * \psi_{\vec{k}} d|k|\,d\varphi\,, \tag{2.72}$$

which leads to the following reconstruction procedure:

1. For each value of $\vec{k}$ compute $\mathcal{F}_f(\mathcal{W}(\vec{x},\vec{k}))$ with an FFT.
2. Multiply the results of 1. with $\mathcal{F}_f(\psi_{\vec{k}})$ (separately for each value of $\vec{k}$).

3. Pad the results of 2. with zeros to reach the desired resolution (separately for each value of $\vec{k}$).
4. Add the padded results up for all values of $\vec{k}$.
5. Calculate the inverse FFT of this sum.

Results of this reconstruction from the various sampling strategies can be found in 3.6.1. They will show that for reasonable parameters the reconstructions can be very similar to the original. Of course, no discretized reconstruction formula can recover the integral of the image (or the absolute grey values), because it has been carefully removed by the choice of the kernels. In the continuous case, this is recovered by the limit $|\vec{k}| \to 0$.

2.8 Multiresolution Transforms: Wavelets and Beyond

From the connectionist point of view the concept of wavelets with the condition to have a geometrical group act on a single functional may seem too narrow. The totality of all functionals or phase space atoms is probably not governed by mathematical elegance but rather by the visual experience of the brain during development. This may result in a variety of forms of functionals as well as very incomplete groups, using only the subsets that are relevant for the visual tasks.

The newest mathematical developments in this direction are presented by (Mallat and Zhang, 1993). Here the data are evaluated with what is called a *dictionary* of function(al)s and only the most significant responses are kept to represent the data. This leads to an elegant way of telling signal and noise apart without having to make very explicit assumptions about the structure of either.

This transform has been applied successfully to sound analysis. Here, the extension of the wavelet functional dictionary by sines and Dirac-functionals yielded very good results, because these are often present in sound data. The extension to two-dimensional image data is straightforward but computationally very expensive. Also, the choice of a dictionary is much less obvious here. In the long run, it should be possible to learn the dictionary from visual experience. A very crude version of this idea is used in the adaptive sampling scheme where we will keep only the wavelet responses whose amplitudes exceed a threshold (see section 3.6). An alternative, but similar concept which insists on more structure is that of *wavelet packets* (Coifman and Wickerhauser, 1992). These are usually optimized for tasks in data compression.

3. Representation of Images and Models

Entweder ein Ding hat Eigenschaften, die kein anderes hat, dann kann man es ohne weiteres durch eine Beschreibung aus den anderen hervorheben und darauf hinweisen; oder aber, es gibt mehrere Dinge, die ihre sämtlichen Eigenschaften gemeinsam haben, dann ist es überhaupt unmöglich auf eines von ihnen zu zeigen.
Denn, ist das Ding durch nichts hervorgehoben, so kann ich es nicht hervorheben, denn sonst ist es eben hervorgehoben.

Ludwig Wittgenstein, Tractatus logico-philosophicus

This chapter gives a complete description of the image representation that will be used throughout the thesis together with the procedures that are used to calculate it efficiently. Furthermore, a simple edge representation will be discussed.

3.1 Image Processing

Images are taken by a video camera, low-pass filtered by averaging over a 4×4 neighborhood and reduced to a resolution of 128×128 pixels.

A sampled continuous wavelet transform with Gabor kernels modified to fulfill admissibility is applied. The method for this (in all our numerical applications) will be the one described by equation (2.59). As described in section 2.5.2 the kernels are parameterized by their center frequency $\vec{k}$. The normalization factors in frequency space are constant for all center frequencies as we have motivated in section 2.5.3. The Gaussian envelopes have been chosen to be circularly symmetric, therefore the reflection about the axis orthogonal to $\vec{k}$ (see section 2.5.1) can be replaced by inflection about the origin. Finally, the Gabor kernels are forced to be band limited as described in section 2.6.

Putting all this together yields the final form of the kernels in frequency space:

$$\mathcal{F}(\psi_{\vec{k}})(\vec{\omega}) = \begin{cases} \exp\left(-\frac{(\vec{\omega}-\vec{k})^2}{2\sigma^2|\vec{k}|^2}\right) - \exp\left(-\frac{(\vec{\omega}+\vec{k})^2}{2\sigma^2|\vec{k}|^2}\right) & : \quad \vec{\omega}^{\mathsf{T}}\vec{k} > 0 \\ 0 & : \quad \text{otherwise} \end{cases}, \tag{3.1}$$

$$\mathcal{F}(\psi^{bl}_{\vec{k}})(\vec{\omega}) \;=\; \begin{cases} \mathcal{F}(\psi_{\vec{k}})(\vec{\omega}) & : \quad \mathcal{F}(\psi_{\vec{k}})(\vec{\omega}) > t_f \\ 0 & : \quad \text{otherwise} \end{cases} . \tag{3.2}$$

For the rest of the thesis we will drop the "bl" and simply use $\psi_{\vec{k}}$ for the kernels.

The transform is executed by taking the Fast Fourier Transform of the image, multiplying by the kernels in frequency space, reducing the size of the frequency representation according to the procedure described in 2.6.2, and finally applying the inverse FFT.

3.2 Suppressing the Background

In this section we will collect all the steps that lead to our image and model representations and describe in full detail how they are generated. All parameters will be specified in section 3.4.5. These parameters will be used throughout the thesis unless stated otherwise.

When working with a phase space representation all operations that require exact localization are bound to cause problems, because the information available in the representation at one point is always some weighted average of the image at the surrounding points. In the course of visual processing this becomes serious when the model has to be separated from the background. Without engaging on the question how this separation can be achieved we have to discuss the consequences for our representation.

In the vast majority of cases model and background are separated by a clear cut line. This does not imply that this line must always be visible, but there are always pairs of points one of which belongs to the model and the other to the background, but nevertheless they are direct neighbors in the image plane. Therefore, the corresponding phase space atoms centered at model points close to the border will always be influenced by the background. When the background changes, these atoms will match poorly.

Several levels of sophistication can be applied to overcome this problem. In (Pötzsch, 1994) a linear transformation is estimated and applied that achieves independence of the background as well as possible. Here we will take the radical view that these atoms represent contaminated information and have to be discarded altogether.

The model is defined by a mask in the image plane:

$$\mu^M(\vec{x}) = \begin{cases} 1 & : \quad \vec{x} \text{ is part of the model} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{3.3}$$

In other words, we require the model to be an arbitrary subset of the image plane and define μ^M as its characteristic function.

The next step will be to decide which phase space atoms do belong to the model in the sense that they are independent of the background. Because the support of the Gabor kernels is the whole image plane this condition will not be met by any atom with absolute accuracy. If the background contains arbitrary huge peaks of light intensity, it is possible that these contaminate all atoms in the whole image. In this case the representation is empty and the whole matching procedure fails. We will therefore restrict the discussion to "reasonable" images, where the Gabor responses in the background are smaller or in the same order of magnitude as in the model.

Then we will define an effective support for the Gabor kernels. Because this is only an approximation, anyway, we will make the further simplification to assume a circular

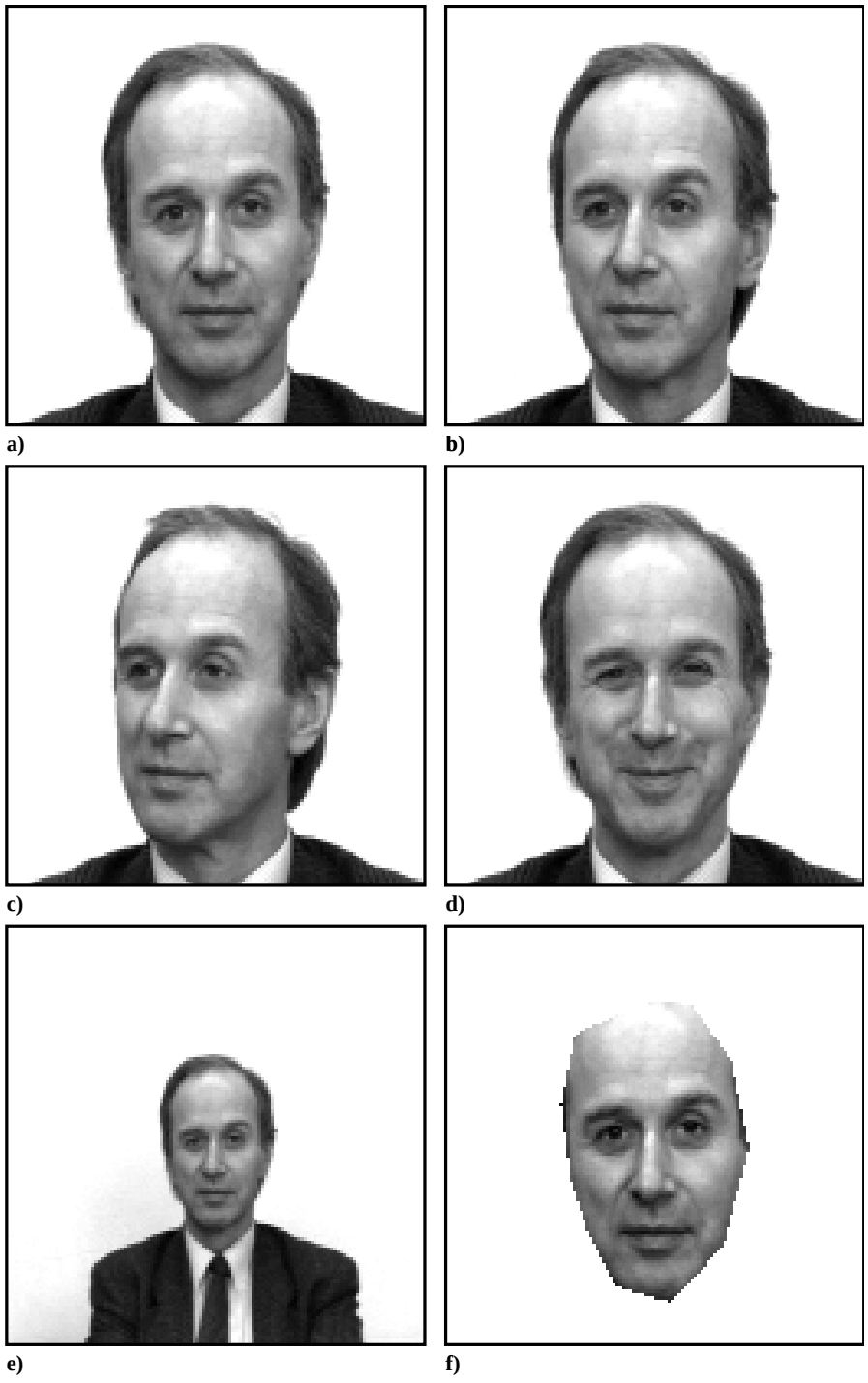

Figure 3.1: The aspects of one person in the various databases. a) is the standard position that is used to create the models. **f)** is an example for a model with segmentation. **b)** through **d)** show the pictures in the test databases **I1** through **I3** (see section6.1.2), where the persons looked 15° or 30° to their left or showed a facial expression of their choice. In **e)** the picture was taken with a different focal length resulting in roughly half the size of the face in the image.

support. Its radius is defined as that distance from the center where the Gaussian envelope of the kernel decays to t_s, the *spatial* threshold under which the kernel is sufficiently close to zero to be ignored. Besides this parameter the radius is dependent on the magnitude of the center frequency $|\vec{k}|$ and the relative bandwidth σ, and can be easily calculated:

$$R(\vec{k}) = \frac{\sigma}{|\vec{k}|}\sqrt{-2\ln(t_s)}\,. \tag{3.4}$$

Now the model/background separation for the phase space atoms can be clearly defined: A unit belongs to the model representation if and only if its central location has a distance to the background larger than the corresponding R.

This introduces a clear asymmetry between model and background. If it should be necessary (which for our purposes is not the case) to treat the background as a separate object this procedure would yield three distinct types of units: The ones belonging to the model representation, the ones belonging to the background representation and the intermediate ones which are discarded.

Fortunately, for the classification of the units it is not necessary to calculate the distances between unit centers and all image points. It suffices to convolve the mask μ^M with the characteristic function χ_R of a disk of radius R. This is due to the fact that the integral over the product of the characteristic functions of two sets is equal to the characteristic function of the intersection of the sets:

$$(\vec{x},\vec{k}) \in \mathbf{S}^M \iff \int \mu^M(\vec{y})\chi_R(\vec{y}-\vec{x})\,d^2y = \int \chi_R(\vec{y})\,d^2y\ , \tag{3.5}$$

$$\text{where}\quad \chi_R(\vec{y}) = \begin{cases} 1 & : \quad |\vec{y}| < R \\ 0 & : \quad \text{otherwise} \end{cases}\ . \tag{3.6}$$

In spite of the wrong sign in $\chi_R(\vec{y}-\vec{x})$ this can be implemented as a convolution, because χ_R is inflection-symmetric.

3.3 Amplitude Thresholding

In order to keep the representations as compact as possible the influence of units with very low response amplitudes must be discussed. These amplitudes, which are smaller than a certain threshold, can be set to zero in good approximation. Before discarding those units completely, it must be assured that their phases do not carry important information. In section 5.4.1 we will discuss in detail that this is indeed the case. Low amplitude responses are close to complex zeros in the continuous transformation, and therefore the corresponding phases are ill-defined or, in other words, very unstable under minimal shifts of the sampling grid. It is therefore advisable to set a threshold and discard all amplitudes below that threshold with the double effect of making the representation more compact and improving the reliability of the phase matches (see section 5.4). We introduce the (relative) threshold t_a and include only such units in the representation with amplitudes not smaller than this threshold times the maximal amplitude in the whole representation (the sampling set before thresholding is denoted as $\mathbf{S}_0$:

$$(\vec{x},\vec{k}) \in \mathbf{S} \Longleftrightarrow \mathcal{A}(\vec{x},\vec{k}) \geq t_a \cdot max(\mathcal{A}_{\mathbf{S}_0}) \tag{3.7}$$

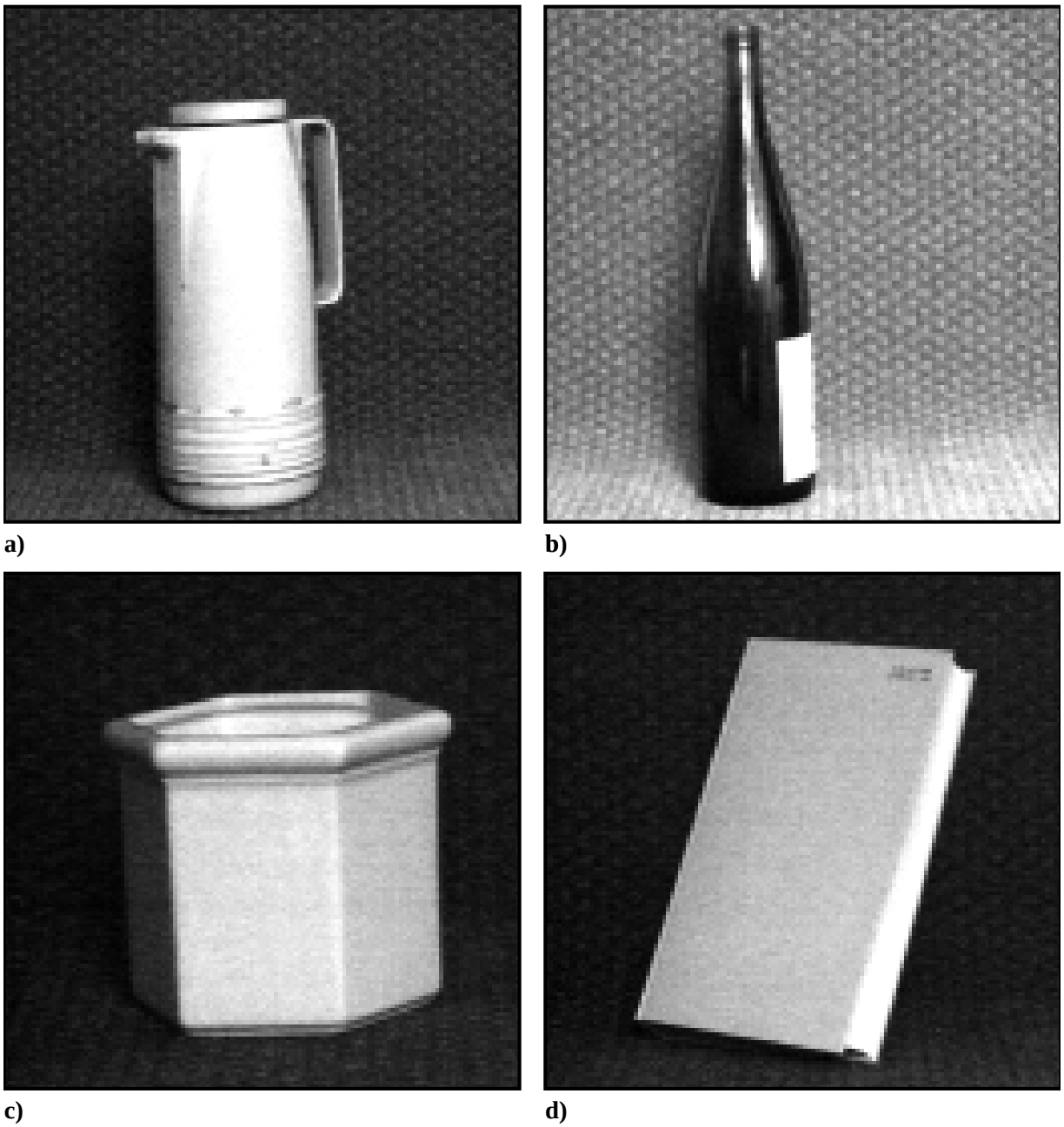

Figure 3.2: Problematic objects for the model representation.. The representation with background suppression relies on sufficient structure inside the object. For objects like the ones shown here this is not suitable. They can only be matched on the basis of their edge information.

3.4 Generating the Representations

3.4.1 Model Representation

1. A picture of the object to be modeled is taken with the video camera and subsampled to the standard resolution. The result is $M(\vec{x})$.
2. The area in the picture which belongs to the object is defined by a segmentation algorithm or by hand. The corresponding mask μ^M is stored.
3. The picture is Fourier transformed.
4. For each $\vec{k} \in \mathbf{S}_F$ the Fourier transform of the picture is multiplied with the discrete kernel $\mathcal{F}(\psi_{\vec{k}})$ as defined in equation (2.66).
5. The result is reduced in size according to the sampling strategy used (see section 2.6.2).
   (a) In the case of Nyquist sampling the smallest rectangular area which is centered in the origin and where the result is effectively nonzero (i.e. $\geq t_f$) is cut out.
   (b) Full sampling is a special case of (a), namely for $t_f = 0$
   (c) In the case of sparse sampling the smallest rectangular area where the result is effectively nonzero (i.e. $\geq t_f$) is cut out. Then it is shifted to be centered at the origin, the shift vector is stored.
6. The inverse Fourier transform is applied to this and yields complex unit responses, which are then separated into amplitude and phase. In the case of sparse sampling the phase is corrected in order to take the shift vector applied before into account.
7. The sampling set is reduced by discarding all units that are influenced by the background with the procedure described in section 3.2 using the mask μ^M.
8. The sampling set is further reduced by discarding all units whose amplitudes are smaller than $t_a \cdot \max \mathcal{A}_{\mathbf{S}^M}$ (see section 3.3).

The (complex) responses of all remaining units are stored as six-tuples with the entries:

1. horizontal location $0 \leq x_1 < 1$
2. vertical location $0 \leq x_2 < 1$
3. length of center frequency $k_{min} \leq k \leq n_{lev}$
4. direction of center frequency $0 \leq d < n_{dir} \cdot \pi$
5. amplitude value $0 \leq \mathcal{A}(x_1, x_2, k, d)$
6. phase value $-\pi < \mathcal{P}(x_1, x_2, k, d) \leq \pi$

Those entries will also be referred to as $(u_1, \ldots, u_6)$.

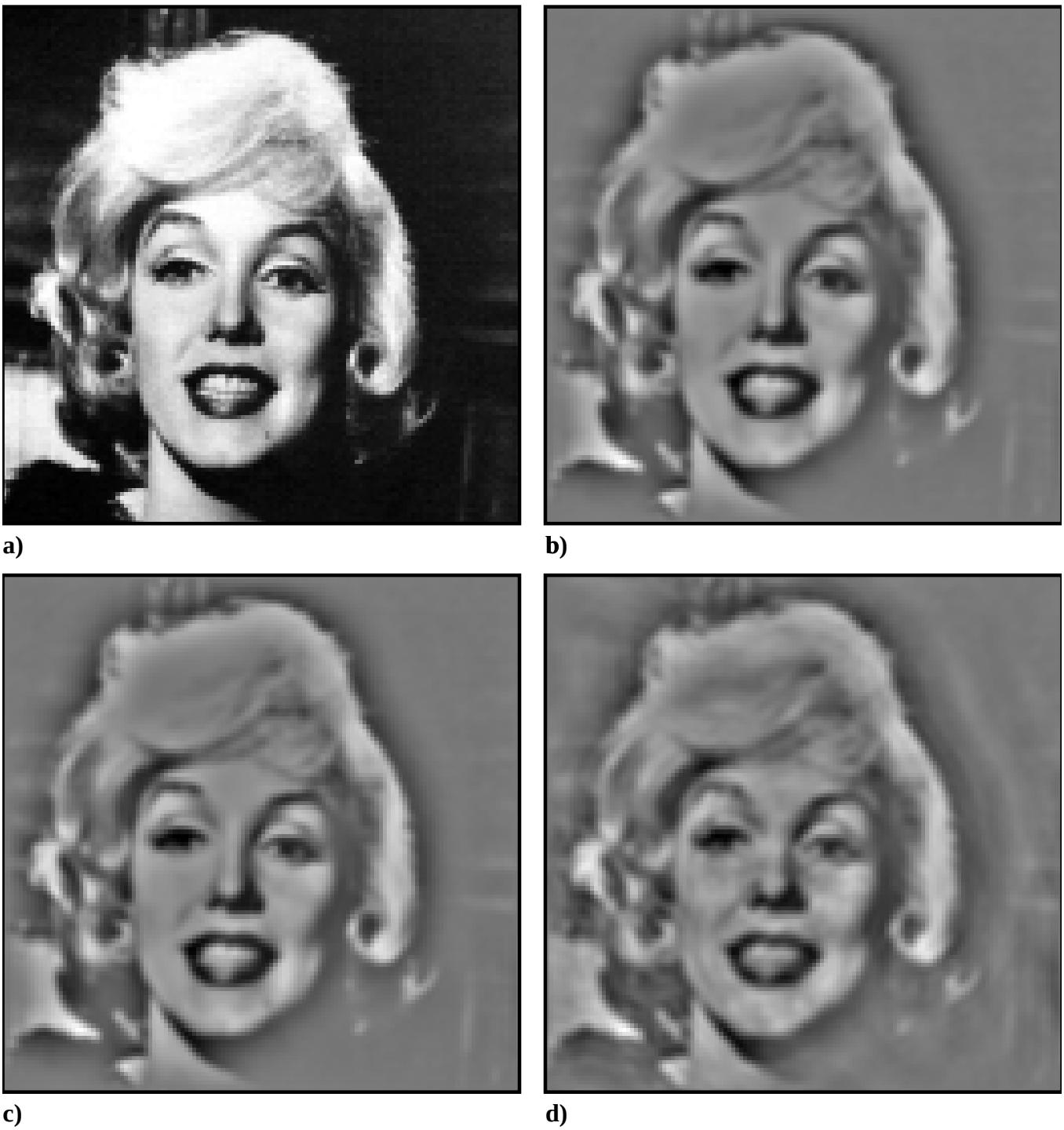

Figure 3.3: Reconstruction from representations with various sampling schemes. The original image in **a)** has been transformed and the units influenced by the wrap around at the border have been removed. **b)** shows the reconstruction from the transform with full resolution (consisting of 172,140 units), **c)** from the one with Nyquist sampling (60,776 units), and **d)** the one from sparse sampling (20,620 units)

3.4.2 Image Representation

The image representation is generated exactly in the same way as the model representation except for the fact that there is no segmentation. The mask μ^I is chosen as identical to the image area and step 5 is only used in order to remove the artifacts introduced by the wrap-around of the finite Fourier transform.

3.4.3 Subrepresentations

For the matching procedures described in chapters 4 and 5 we will have to use subsets of the image and model representations. These are simply subsets of units that adhere to certain properties. Specifically, we will use the subrepresentations of all units which have a given modulus of the center frequency. These will be called *frequency levels* $\mathcal{K}_{|\vec{k}|}(I)$, $\mathcal{K}_{|\vec{k}|}(M)$ or simply *levels*. They are defined as the set of all units in the representation whose length of center frequency (or third component in the six-tuple) is equal to $|\vec{k}|$. For further simplification we replace the index $|\vec{k}|$ by the corresponding number i if $|\vec{k}| = k_{min} \cdot (k_{max}/k_{min})^i$. Then the whole representation is the union of all levels:

$$\mathcal{R} = \bigcup_{i=0}^{n_{lev}} \mathcal{K}_i \tag{3.8}$$

3.4.4 Image Database

Now we have completely specified how models and image are prepared for the matching procedures described in chapters 4 and 5. For simplicity each stored object will be represented by *one* model in the form of an array of six-tuples.

The image to be classified is another array of six-tuples which, due to the lacking segmentation will usually be much longer than the ones belonging to the models.

For evaluating the abilities for object recognition we are using 5 databases with faces of 83 persons. The first one contains one frontal view and is used to create the model representations. The second and third ones contain images of the persons looking 15° and 30° to their left, respectively. Database 4 contains an arbitrary facial expression and is used for evaluating the deformation tolerance. In 5 the pictures were taken with a different focal length, resulting in a size of the face 50%. The possibility of including size-invariant matching will be shown in section 5.8. Figure 3.1 shows the five views of one person, including the segmentation used for the matching.

The databases 1, 2 and 4 are the same as in (Lades et al., 1993), with two persons omitted for whom the pictures 3 and 5 were not available. The results can therefore be compared with that system, which will also be briefly described in 7.1. Most results for the single mappings, however, have been obtained with persons not in the databases in order to avoid the danger to optimize systems with respect to the special data at hand.

3.4.5 Standard Parameters

In the following all free parameters of the representations are fixed. The image resolution will be 128×128 pixels. The sizes of the center frequencies for the wavelet transform will

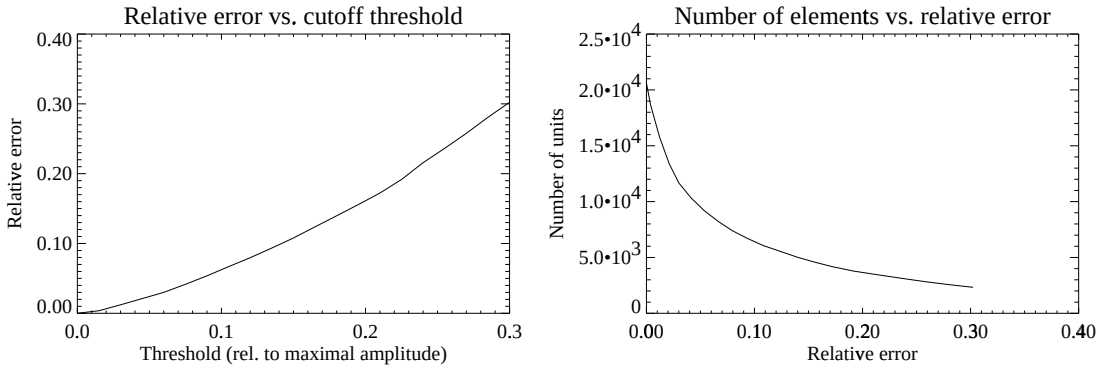

Figure 3.4: Effects of amplitude thresholding. The image from figure 3.3 has been transformed with sparse sampling. All amplitudes below t_a times the maximum of all amplitudes have been discarded. The reconstructed image has been compared with the one obtained without thresholding (Figure 3.3 **d)**) by taking the squared pixelwise difference and dividing it by the norm. This relative error is plotted on the left hand side vs. the relative threshold. The right hand side shows the number of remaining units vs. the relative error. It can be seen that our default value of $t_a = 0.05$ leads to compact representations without introducing too much deviation in the image information.

---

be sampled logarithmically, their directions uniformly:

$$\mathbf{S}_f = \left\{ k_{min} \cdot \left(\frac{k_{max}}{k_{min}}\right)^{\frac{k}{n_{lev}}} \begin{pmatrix} \cos\left(\frac{d\pi}{n_{dir}}\right) \\ \sin\left(\frac{d\pi}{n_{dir}}\right) \end{pmatrix} \;\middle|\; k \in \{0, \cdots, n_{lev}\}\, d \in \{0, \cdots, n_{dir} - 1\} \right\}, \quad (3.9)$$

$$\begin{aligned} n_{dir} &= 4\,, && (3.10)\\ n_{lev} &= 2\,, && (3.11)\\ k_{min} &= 0.4\,, && (3.12)\\ k_{max} &= 1.5\,. && (3.13) \end{aligned}$$

In the following we specify the standard values for the relative bandwidth, the threshold for the amplitudes, the spatial cutoff for the kernel values and the cutoff for the kernels in frequency space: will be:

$$\begin{aligned} \sigma &= 2.0\,, && (3.14)\\ t_a &= 0.05\,, && (3.15)\\ t_s &= 0.125\,, && (3.16)\\ t_f &= 0.125\,. && (3.17) \end{aligned}$$

3.5 A Simple Edge Representation

The representations described here have an obvious limitation: They cannot be applied to objects with hardly any internal structure, because here the wavelet responses will

be very close to zero except for model boundaries. The boundaries, however, have been discarded because the wavelets centered there are influenced by the background.

In cases like the one shown in 3.2 one can rely only on the edge structure to get a matching. Therefore, we describe a simple edge representation which is calculated similarly to the model representation. It is clear, that the background suppression can not be applied in this case.

After transformation and subsampling exactly as above the results for each center frequency are scanned for local maxima of the amplitudes. Each response amplitude is compared with its spatial neighbors in the direction $\pm\vec{k}$ and discarded unless both of them are smaller than itself.

The concept of neighbor in various direction does not pose problems if, as in our case, the directions are 0°, 45°, 90°, and 135°. If more directions are required, this probably is hard to generalize.

After removing everything beside local maxima amplitude thresholding can still be applied to remove maxima that do not stem from real edges but rather from numerical noise in the absence of any local structure.

This leads to a very compact image code which is inspired by and similar to Mallat's multiscale edges (Mallat and Zhong, 1991). The differences are that four directions instead of two are used and that the spatial sampling density is frequency dependent in our case.

3.6 First Experiments with the Model Representation

Before engaging on the matching task we will describe some experiments with the representations defined above. They will also motivate the choice of the parameters σ and t_a.

3.6.1 Reconstruction

Although the reconstruction of an image from a representation will not be needed for the matching procedures it is important for the choice of parameters and for giving a quick impression of the information content in the representation. The method we are using follows the reconstruction formula (2.24), which basically states that each unit must be multiplied with the corresponding kernel and the result be integrated over all center frequencies.

So the procedure is as follows. For each center frequency the corresponding units are extracted and the (complex) responses are arranged into a rectangular matrix, whose size depends on the maximal density (resolution) of units present in the whole representation. If the representation has been calculated as described in section 3.2 this is straightforward. If further manipulations have been applied it may require some rounding of the locations (x_1, x_2) of the units. The responses of all units that are missing in the representation are set to zero. This matrix is Fourier-transformed and multiplied with the kernel in the frequency domain; the results are added up over all center frequencies. A final inverse Fourier transform yields the reconstructed image.

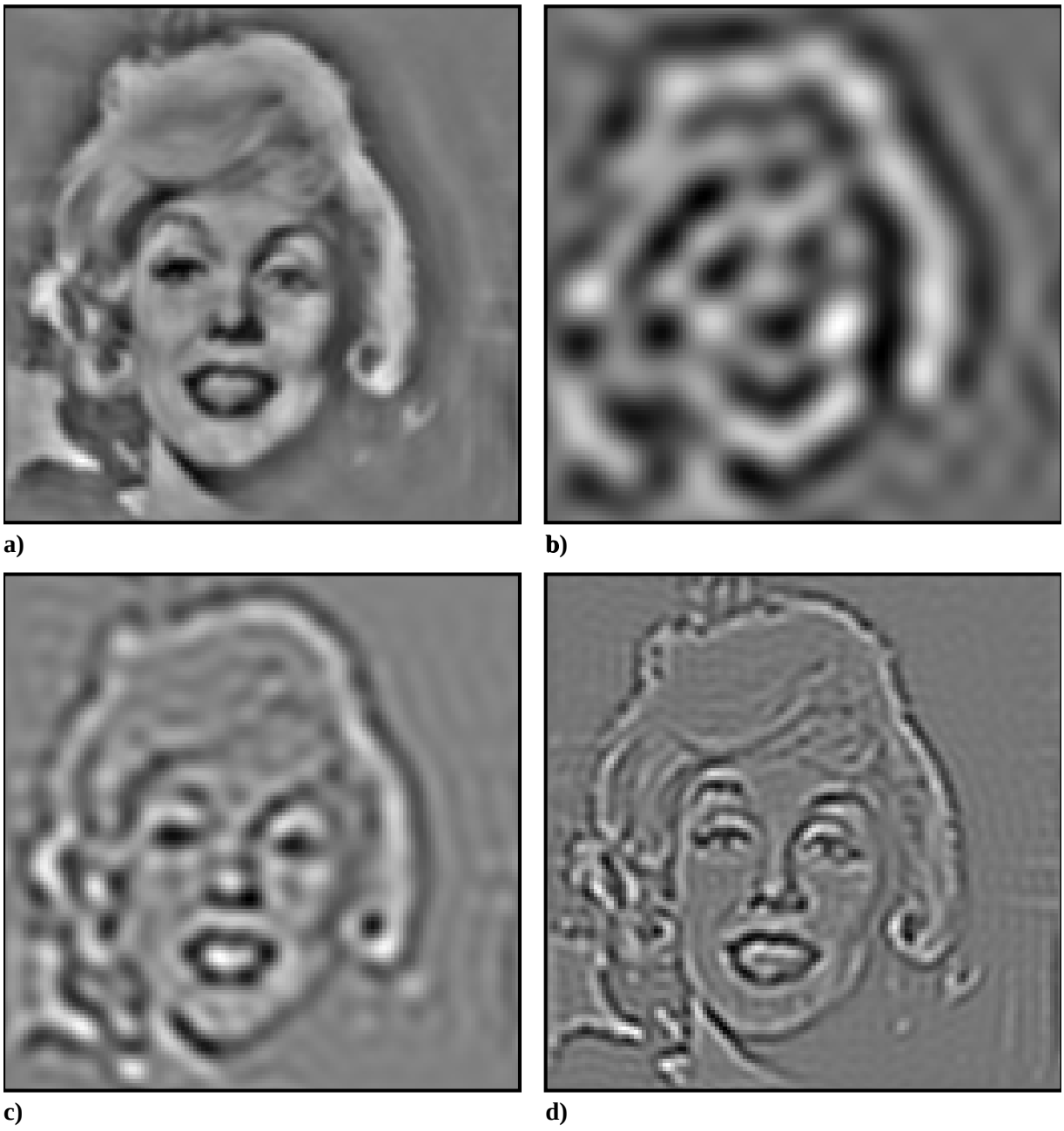

Figure 3.5: Reconstruction from single frequency levels. Picture **a)** shows the reconstruction from the full representation, **b)** from $\mathcal{K}_0$, **c)** from $\mathcal{K}_1$ and **d)** from $\mathcal{K}_2$ only.

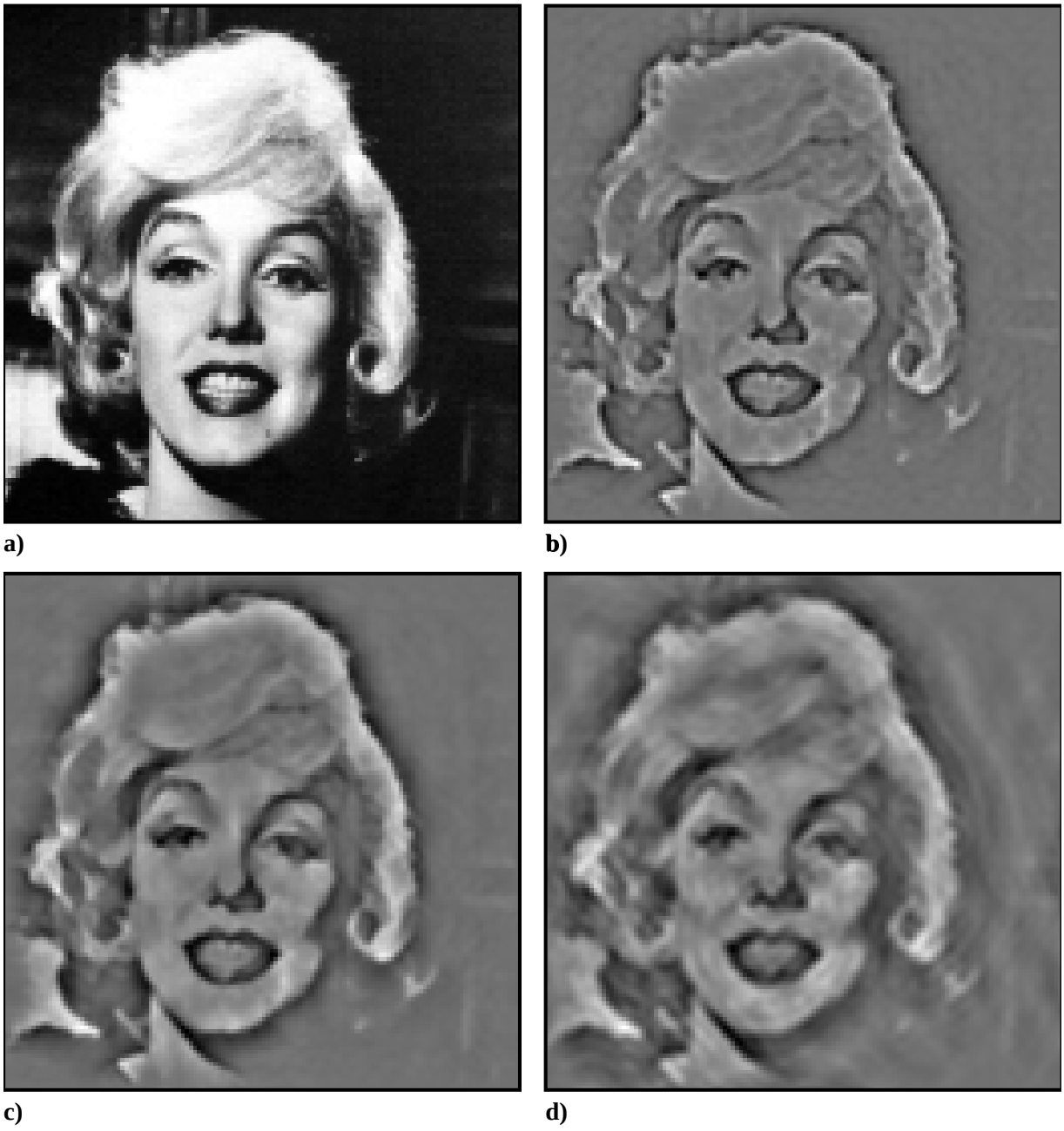

Figure 3.6: Reconstruction from edge representation. The original image in **a)** has been transformed and the units influenced by the wrap around at the border have been removed. Then only the units with local amplitude maxima as defined in section 3.5 have been kept. **b)** shows the reconstruction from the transform with full resolution (21,735 units), **c)** from the one with Nyquist sampling (12,520 units), and **d)** the one from sparse sampling (5,176 units). Although the last reconstruction does not look particularly good it is worth noticing that this representation has much fewer units than the original image had pixels.

When comparing the result with the original image, e.g. in figures 3.3 or 3.7 it has to be kept in mind that the absolute gray value cannot be reconstructed because all kernels have vanishing integral. Due to the finite sampling of the center frequencies also the very low frequency components will be missing from the reconstruction. This could be easily remedied by storing the total grey values and a few low-frequency components in addition to the representation. This has not been done, because the reconstructions are by far good enough to allow human recognition.

3.6.2 Reconstruction from subrepresentations

The reconstruction procedure has been formulated in such a way that it can be applied to any subrepresentation. In this subsection we treat five interesting cases of subrepresentations, namely the ones remaining after amplitude thresholding with various thresholds, the single frequency levels and the edge representation from section 3.5. In order to get some visualization of the information contained in the frequency levels we present the results of reconstruction from this particular kind of subrepresentations in figure 3.5.

In (Mallat and Zhong, 1991) a very sophisticated reconstruction procedure is applied that recovers the image from the multiscale edges with excellent quality. Here, we are not so much interested in image compression but rather in matching. So for our purposes, any reconstruction algorithm will do that reproduces a version of the image that is easily recognizable by a human. This makes sure that we did not lose too much information when choosing a compact representation.

These considerations led to the attempt to simply use the same reconstruction procedure as for the normal representation, which is described in detail in section 3.6.1. Figure 3.6 shows the results. From this example (out of several that we have tested the algorithm with) it can be concluded that this simple-minded reconstruction procedure suffices to fulfill the above requirement. It could certainly be improved by applying iterative improvement like the one used by Mallat.

3.6.3 Affine Image Transforms

Due to its construction, the continuous wavelet transform allows a simple formula for the behavior under the affine transformations of image space that constitute its geometrical group. In other words:

$$\mathcal{W}\left(f(A\vec{x}+\vec{b})\right)(\vec{y},\vec{k}) = \mathcal{W}(f(\vec{x}))\left((A\vec{y}+\vec{b},\left(A^{\mathsf{T}}\right)^{-1}\vec{k}\right), \tag{3.18}$$

if $A^{\mathsf{T}}A$ is a multiple of the unit matrix. In two dimensions those matrices are arbitrary combinations of rotation, scaling and reflection. This yields the following method to apply such a transform to a representation.

Translation: The translation vector is added to (u_1, u_2).

Scaling: (u_1, u_2) are multiplied with the scaling factor, u_3 is divided by the scaling factor.

Rotation: The vector (u_1, u_2) is rotated by the rotation angle, which is also added to u_4.

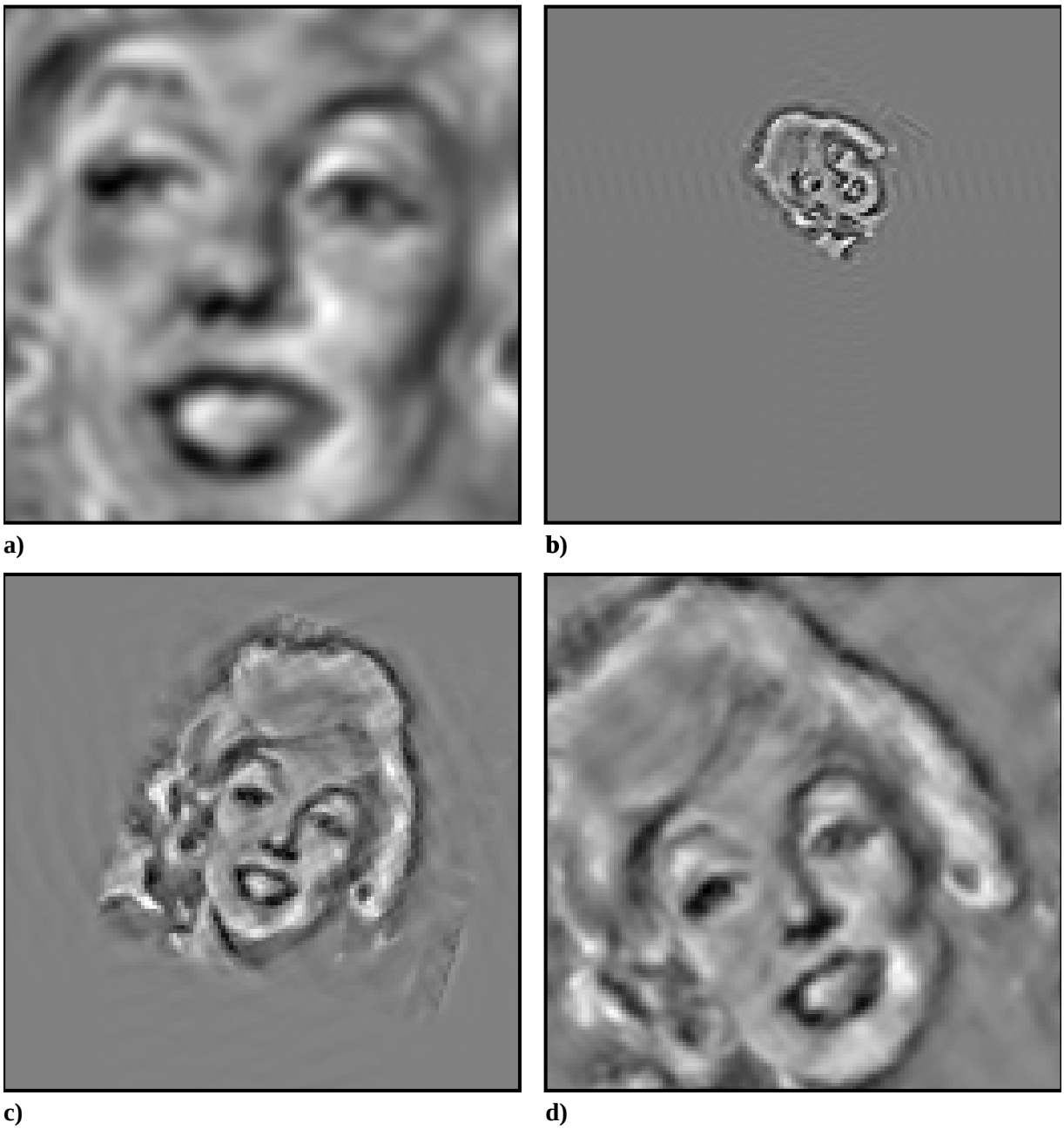

Figure 3.7: Reconstruction from affine transforms of an image. The representation as a collection of unit responses is very well suited for creating scaled, rotated and shifted versions of the image. Note that neither scale factor nor rotation angle need to suit the sampling of center frequencies.

Reflection: (u_1, u_2) is reflected about the axis, the response phase u_6 is replaced by $u_6 + \pi$.

These steps can be combined in order to apply all combinations of the four transformations. In order to keep a representation with units centered only within $[0, 1) \times [0, 1)$ all units falling outside this area are discarded.

This easy way to apply geometrical transformations to a representation can eventually be used for the matching procedure to include scale- and rotation invariance, once the scaling factor and rotation angle have been estimated somehow. Here we only present the reconstruction of transformed representations to demonstrate the success of the method (figure 3.7).

4. Hierarchical Dynamic Link Matching

Zwar ists mit der Gedanken-Fabrik
Wie mit einem Weber-Meisterstück,
Wo ein Tritt tausend Fäden regt,
Die Schifflein herüber hinüber schießen,
Die Fäden ungesehen fließen,
Ein Schlag tausend Verbindungen schlägt.

Johann Wolfgang von Goethe, Faust

4.1 Neural Networks

In this section we will give a brief overview of the basics of neural network modeling. This approach is central to the attempt to describe cognitive capabilities as dynamical systems. Therefore, it is necessary to outline the building elements that will constitute our dynamical system.

4.1.1 Dynamics of Model Neurons

The main physiological substrate of the brain consists of *neurons* or *nerve cells*. As usual in science these can be studied on various description levels. Basically, any neuron consists of a cell body, a complicated branching pattern that receives input from other cells, the *dendrites* or *dendritic tree* and an *axon* that transports the activity to other neurons. The research on the functioning of this transport works has revealed a fascinating universe of molecular channels in the cell membrane that can transport electric activation. Viewed on this level, a single nerve cell may well have a higher degree of complexity and organization than a complete computer (disregarding the fact that both terms are pretty ill-defined). This should be kept in mind when evaluating claims about artificial brains that can soon be expected. Another fascinating piece of architecture are the *synapses* that make the contacts between the axons of sending neurons and the dendrites of receiving ones. These are complex systems of various chemical and electric processes.

In our context we are not interested in those details but rather in the properties of *networks* of neurons. It is clear, that we must abandon nearly all detail about single neurons in order to study the network effects. All we will keep is the notion of a *model neuron* that receives input from a number of other model neurons (eventually including

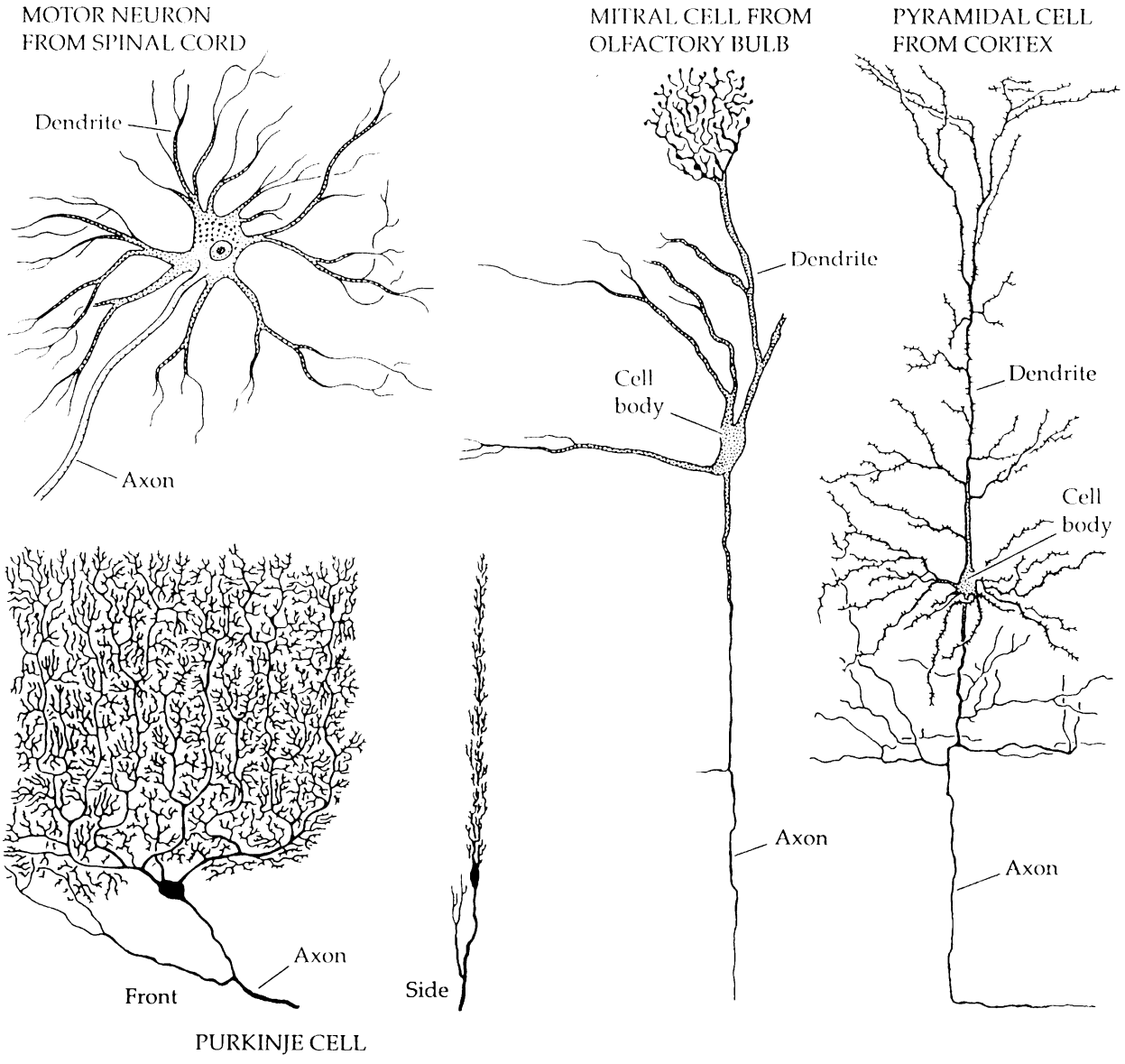

Figure 4.1: Some biological neurons. From (Nicholls et al., 1980).

itself), apply some simple transformation to that input and transport the resulting output to other neurons, which in turn do the same thing.

The important objects of study then are the connections between the cells and the transformation inside the cell. The situation is further simplified in the following way: There is an activity of the neuron which leads to an output signal via a simple nonlinear function. Neurons are connected by synapses that are characterized by a single number, their *synaptic strength* or their *synaptic weight* or simply *weight*. If there are n the cells in a system the weights, in general, can be represented by a $n \times n$ matrix. Output values are transported to other cells via synapses, which multiply them by their weights and thus turn them into input signals for the other cells. Finally, each cell calculates its activity from the summation of all input signals by certain dynamics. This model will be put into formulae in the next subsection. Once the dynamics of the neurons, the nonlinearities and the matrix of the weights are specified, this describes the behavior of the net completely. The main issue of neural network research is how to specify the weight matrices in order to achieve a desired behavior.

Our model neurons will differ from biological neurons in one more way, that has already been discussed in section 2.4. Their activities may (in principle) take negative values, which can model the activities of biological neurons only if this activity describes at least a pair of cells, one of which has an excitatory, the other an inhibitory effect on their target cells. This is acceptable because it makes the modeling so much easier.

The activity a, the output o and the input (stimulus) s of a model are ruled by the following dynamical system:

$$\begin{aligned} \tau_a \frac{d}{dt} a &= -a + s && (4.1) \\ o &= \vartheta(a) && (4.2) \\ s &= \sum_i W(i) o(i) && (4.3) \end{aligned}$$

The summation in (4.3) runs over all neurons that have a forward connection to the special neuron considered.

The nonlinearity $\vartheta(x)$ is supposed to be monotonically increasing and map $\mathbf{R}$ into the interval $[0, 1]$. Typical choices are the *Fermi function*

$$\vartheta(a) = \frac{1}{1 + \exp(-\lambda a)}\,, \tag{4.4}$$

which converges to a Heaviside function for $\lambda \to \infty$, or

$$\vartheta(a) = \begin{cases} 0 & : \quad a < 0 \\ a & : \quad 0 \le a \le 1 \\ 1 & : \quad a > 1 \end{cases} . \tag{4.5}$$

The latter is very practical for fast numerical implementation and does not share the drastic effects of a Heaviside function. Mappings onto $\mathbf{R}^+$ like $\vartheta(a) = \max(0, a)$ are also in use, but the complete absence of nonlinearities would lead to boring (and useless) behavior of the models (von der Malsburg, 1973).

Equation (4.2) needs some further comments because here the first difference between *conventional* neural networks and our dynamic networks becomes clear. Depending on the position in the network, the input s is either a sensory input or a summed output from other neurons or a mixture of both. Conventional networks, including, e.g., back-propagation schemes, use the situation where the switching of activities is instantaneous, or $\tau_a = 0$ for all units. In this case the resulting network is not really a dynamical system but only a stimulus-response system that follows the development of the input pattern without delay and without any internal dynamics. Neural networks of that kind have been applied successfully to invariant object recognition, e.g. in (Pitts and McCulloch, 1947; Rosenblatt, 1961; Fukushima, 1980). Nevertheless, we will argue in section 4.2 that this paradigm it too narrow to yield a brain model that could capture the task of object recognition or solve the correspondence problem under realistic conditions.

4.1.2 Connections Between Model Neurons

In equation (4.3) the connection strengths between model neurons have already been introduced. A neural network is a *directed graph* with neurons as vertices and connections as edges, which are labeled with their synaptic weights. In principle, every network can be modeled by a complete graph, with a connection between every pair of neurons, because the non existing edges can be given zero weights. This modeling strategy, however, is not advisable, because the connectivity of neural nets is usually a limiting factor for their simulation. Furthermore, connections may have to bear more information than only a single weight, namely if they are modifiable or not and on which time scales. This further enhances the storage and computation time problems. For those reasons predefined structure must be exploited wherever possible.

The most common structure is a network of several *layers*. These are subsets of neurons arranged in planes where the connections inside a plane are regular in the sense that the connection strength between the neurons is only a function of their distance vector. This has the advantage that these *intralayer connections* can be modeled by convolution with a connectivity kernel which is much simpler than having to model an arbitrary connection matrix. The intralayer connections are usually not modifiable. This scheme contains the (important) case that the kernels are identically zero, i.e. the layers have no internal connections at all.

In order to describe neuronal layers, the variables a, o, and s for the single neurons are replaced by $a(\vec{x}_L)$, $o(\vec{x}_L)$, and $s(\vec{x}_L)$, where $\vec{x}$ denotes the position inside the layer, and L is a symbol that specifies the layer. According to our general standard, $\vec{x}_L$ will be viewed as continuous or discrete whatever is more suitable. Also we will drop the index L wherever the layer is clear from the context.

In the following we will be mainly concerned with *interlayer connections* , i.e. synaptic connections between two layers. Here, in general, we have to model full weight matrices which we will write as $W(\vec{y}, \vec{x})$. The activities $a(\vec{x})$ of neurons in one layer serve as inputs $d(\vec{x})$ for the ones in other layer, here denoted by the variable $\vec{y}$:

$$d(\vec{x}) = \int W(\vec{y}, \vec{x}) \cdot \vartheta(a(\vec{y}))\, d^2y \tag{4.6}$$

After the introduction of short-time weights in section 4.2 they will serve as a code for

image point correspondences in the sense that corresponding pairs of image points will attain a high weight in their connection, noncorresponding ones will have no connection or very low weights.

4.1.3 Unsupervised Learning

A simple counting argument shows that the connectivity of the brain can not be completely determined by genetic information. The number of cells in the cortex has an order of magnitude of 10^{10}. The genetic code of a human, however, consists of "only" about 10^9 base pairs and is therefore simply too small to code even for the connectivity between brain cells, let alone for the corresponding weights.

Beyond this consideration which is contestable from various sides there is plenty of neurobiological evidence that the development of the brain relies heavily on stimulation from the environment. This can be shown already on the lowest levels of vision, so there is hardly any doubt that the same applies to higher brain functions. This statement does not directly touch the old and heated debate if our very high cognitive and social functions are dictated by genetic or environmental influences because it is completely beyond the current reach of neuroscience to follow either trace up to behavior.

During the course of brain development there is usually no agent that would tell the brain that it is doing the right thing. Most of this developmental learning must rely on extracting statistical information from the environment, in other words, brains are probably optimized to deal with regular environments, and to make use of those regularities. This strategy constitutes a very important evolutionary advantage, because changes in the environment, i.e. the vanishing of old regularities and the emergence of new ones, can be very rapid. If the brain structure was stored genetically, only the usual evolutionary mechanisms could lead to adaptation. If the brain development happens in close interaction with the environment, one generation can suffice for that adaptation. This special form of brain development is generally called *unsupervised learning*, because no supervisor is necessary to form the brain. (The opposite, *supervised learning* is used to name methods that make neural networks behave like a desired function in the stimulus-response paradigm. This is mentioned only for completeness and is not relevant in our context.)

The simple cells that have been discussed in detail in section 2.4 present an example for unsupervised learning. Even though they represent a very early stage of vision and their function is very similar in all healthy animals of one species experiments show that they are not hardwired from birth on. There is a certain time during development of the animal, the *critical period*, when their orientation selectivity is formed. For kitten this period ranges from 3 to 13 weeks postnatally. If they are kept in an environment which lacks variety in orientations and forced not to move around, the cells sensitive for the absent orientations do not develop. The adult cats will have a "selective blindness" and are unable to see the orientations they have not learned during their critical period (Hirsch and Spinelli, 1970; Blakemore and Cooper, 1970).

A dynamical system that models the development of simple cells has been proposed in (von der Malsburg, 1973). It shows exactly the behavior described above: If stimulated with a rich set of patterns, the full range of orientation selectivity develops, with a reduced

set, only a reduced range of orientations is found, although all cells have developed some specificity.

The obvious question of how such a system can get off the ground at all has a very simple answer which has been given by Donald Hebb (1949) in a speculative way: The synaptic weights in the immature system are randomly distributed. Thus any stimulus leads to a diffuse response in the sense that all target cells react in a mediocre way to all stimuli. Then the weights of the synapses that connect active feature detectors and target cells with an activity slightly above average are slightly strengthened. The idea behind this is that the connections between cells active in response to the same stimulus are good connections in the sense that they reflect important knowledge about the world and are worthwhile reinforcing. Put into a differential equation that reads

$$\frac{d}{dt}W(n,m) = \lambda W(n,m)o(n)o(m)\,. \tag{4.7}$$

This equation leads to the growth of a weight if the two cells n and m connect by the corresponding synapse are active at the same time. The speed of this growth is governed by the *learning rate* λ.

If many stimuli are presented to the net there would be a statistical chance for every single weight to grow, and all of them would simply explode towards infinity. Therefore, Hebb's principle must be accompanied by another one which restrains the unlimited growth of all synaptic weights. Several authors have applied various principles here, the simplest include competition among all synapses that either target on the same cell or originate from the same cell or both. This transcribes to the dynamical system as a constraint, which, for the three possibilities mentioned, takes the forms:

$$\sum_n W(n,m) \le W_{max} \tag{4.8}$$

$$\sum_m W(n,m) \le W_{max} \tag{4.9}$$

$$\sum_n W(n,m) \le W_{max} \;\wedge\; \sum_m W(n,m) \le W_{max} \tag{4.10}$$

This constraint can, in principle, be enforced by an evolution equation (Eigen, 1971; Hofbauer and Sigmund, 1988). Such equations describe systems that are self-amplifying and have to compete with other self-amplifying systems for a limited resource. This would lead back to a dynamical system without constraints, which would be more convenient for analytical treatment. As our system is too complicated for analytical treatment, anyway, we do not need this complication. In our simulations, we will simulate small time steps of the dynamical system and enforce the constraint by normalization after every step. Also this formulation is very plausible in the biological sense. The linear dynamics of link growth is limited by the finite physiological resources of a cell to receive or produce synaptic connections.

4.2 The Dynamic Link Architecture

We will now return to the problem of invariant recognition. The neural network models that adhere to the stimulus-response scheme run into problems when they are confronted

with realistic problems, i.e. real images. The reason for that 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, 1980). However, in order to achieve a realistic system, the amount of new cells to be introduced to cover the whole spectrum of invariances soon exhaust the total number of cells available.

The clue to a possible solution lies in a closer analysis of this spectrum. Although there are many invariances that the brain can achieve it can not achieve them equally well. Some tasks take distinctly more time than others. This is not described by a structure like the neocognitron, because (once it has developed to its final state as a pattern recognizer) the processing time needed to classify an input pattern is practically constant. This leads to the idea that recognition in the mature system is an *active process*, a convergence of the system to an ordered state.

From the vast psychophysical literature scrutinizing the processing times of human subjects performing recognition tasks we will only sketch one example to back the argument. The explicit experiments are described, e.g. in (Treisman and Gelade, 1980).

Human subjects were presented combinations of green and red crosses and circles. 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 can be interpreted in the following way: Assuming that there are hardwired detectors for the simple features like "cross" or "circle" or "red" or "green" those probably do not exist for the combined features "red cross" etc.. Instead, the combination of low features into higher ones constitutes an active process that takes time, and it takes more time than a simple, hardwired stimulus-response scheme would require.

Rather than going through more psychophysical experiments we will consult our everyday experience to convince ourselves that specialized detectors can not exist for all complex objects. This is because we can recognize new and unusual feature combinations without any problem. Everybody can recognize a purple cow on the first occurrence even if neither experience nor evolution had any need to develop neurons responsible for such a strange creature.

Stimulus-response neural networks cannot account for the experiments described above. In order to overcome this shortcoming the *Dynamic Link Architecture* (von der Malsburg, 1981) postulates that there must be a mechanism that can *bind* simple features into complex ones, and that this mechanism should exist on a very low physiological level. This requires the introduction of a new set of dynamical variables that can code for the presence or absence of binding between two neurons. The proposed solution consists in introducing a second set of synaptic weights that are modifiable on the very short time scale of cognitive processes and are constrained by the long-term weights (that code the long-time experience). They will be called *short-time weights* or *dynamic links* . For no-

tational convenience the symbols $W(n,m)$ and $W(\vec{y}_L,\vec{x}_L)$ will be used for these weights in the following, the long-term (permanent) weights will be denoted by W_p.

For a complete description, the dynamics of these short-time weights must be specified. This choice is very much constrained by the requirement that the dynamics of a synapse can make use only of dynamic variables whose values are accessible to that synapse, namely the activities of the pre- and postsynaptic cells and the distribution of all the synapses on either cell. Throughout the rest of this chapter, the same equations that have been introduced above for unsupervised learning, namely equations (4.7) and one out of (4.8), (4.9), and (4.10), will govern these dynamics, with the extra constraint

$$W(\vec{y}_L,\vec{x}_L) \le W_p(\vec{y}_L,\vec{x}_L)\,. \tag{4.11}$$

To date a problem with this postulate is that the physiological basis is not very clear. This leads to some freedom in the choice of dynamics. So some authors find it preferable to use rapidly switching *gating neurons* instead of dynamic links (Hinton and Lang, 1985; Phillips et al., 1988; Olshausen et al., 1993). This leads to the need for more neurons, but the cell counting techniques are too poor to allow an empirical decision between these theories. From the theorist's point of view, this debate is not too important, because the central issue is the dynamics of the connections and not their physical basis. Of course, the latter would constrain the first, but the required biological data is not available yet.

We close this section by summarizing the fundamental principle of the dynamic link architecture: Additional to the long time synaptic weights there are short-time synaptic weights that are dynamic variables with time constants in the same order of magnitude as the cell activities. Their strengths code for the degree of binding that exists between the neurons involved. Their dynamics are ruled by short-time correlations of neuronal activities and competition among each other. The complete system converges rapidly from an unordered initial state to a highly organized state which corresponds to a percept.

4.3 Dynamic Link Matching for Object Recognition

In this section we review a first dynamical system that can solve the correspondence problem on the basis of the Dynamic Link Architecture. It has been described under various aspects and with varying detail in (von der Malsburg, 1988b; Lades et al., 1993; Konen and Vorbrüggen, 1993; Konen et al., 1994).

The system architecture consists of two neuronal layers, an image layer and a model layer, both of which are equipped with feature detectors.

The interaction kernels that describe the intralayer connections have the form:

$$\kappa(\vec{x}) := \alpha \exp\left(-\frac{\vec{x}^{\mathsf{T}}\vec{x}}{2\rho^2}\right) - \beta\,, \quad 0 < \beta < \alpha \tag{4.12}$$

This means that neighboring cells have excitatory (positive) connections while distant cells have inhibitory (negative) ones.

Then the dynamics of a single layer attain the form

$$\frac{d}{dt}a(\vec{x}) = \tau^{-1}a(\vec{x}) + (g_\kappa \kappa(\vec{x}) - c_g) * \vartheta(\vec{x}) + s(\vec{x})\,, \tag{4.13}$$

where $s(\vec{x})$ denotes a possible input to the layer.

It can be shown that if the input is small this equation converges to an asymptotically stable state where only one disc-shaped region of the layer has positive activity (Amari, 1989; Konen et al., 1994). The size of this region is governed by the parameters α and β. For brevity we will refer to this region as a *blob* in the following. The location of the blob is determined by asymmetries in the initial conditions. As a consequence, only the cells inside the blob have a nonzero output. A spatially constant activity also constitutes a stationary state of the system. However, this is an unstable one and can be avoided by adding an arbitrarily small noise term.

For the whole system two such layers are interconnected by dynamic links in the way that each cell in the image layer has a connection to all cells in the model layer. If the weights are sufficiently small (which is easily regulated by adjusting W_{max}) both layers still adhere to the above dynamics and form blobs. Once two blobs are formed the weight dynamics are updated. For solving the correspondence problem, not only the activities of the cells contribute to the weight modification but also the feature similarities. The growth rate of a weight is proportional to the product of both. The appropriate formula reads:

$$\frac{d}{dt}W(\vec{y},\vec{x}) = \lambda W(\vec{y},\vec{x})T(\vec{y},\vec{x})o(\vec{y})o(\vec{x})\,. \tag{4.14}$$

The growth is again limited by one of the equations (4.8), (4.9), or (4.10). The matrix T codes the feature similarities. The exact features are not of interest here, the ones defined in (7.2) with the similarity function from equation (7.3) are well suited (Konen and Vorbrüggen, 1993).

This system solves the correspondence problem in the following way: The similarity matrix T has many local maxima, corresponding to the feature ambiguities which turn the correspondence problem into a problem. The blobs in each layer now constrain the area where the weights can grow in the way that only *neighboring* cells contribute to the growth of weights. This disambiguates the feature correspondences if the blobs are small enough.

The whole process of blob formation and weight update is repeated many times with random noise as initial condition. Then the blobs statistically have covered all image locations and all model locations. The growing weights, however, influence the blob formation in the model layer because corresponding areas get a higher and higher probability to be simultaneously active in a blob pair. This constitutes a self-organizing system that indeed converges to the desired one-to-one mapping between corresponding image and model points.

The system described in this section has two shortcomings. First, the resetting of the blob formation has to be done by some control unit which is not part of the dynamical system. Second, many iterations of blob formation and weight update must be applied until the final mapping is achieved. The first shortcoming will be remedied by the introduction of *running blob dynamics* in section 4.5, the second will be alleviated by a hierarchical scheme that starts by establishing rough correspondences on small layers of low-frequency features and then successively refines that mapping using the information from higher frequency bands.

4.4 The Need for Hierarchical Processing

The use of pyramidal representations is often disparagingly referred to as being but a technical trick to save computer resources and having no significance for the description of biological systems. The main argument is that in such systems all processing is so highly parallel that considerations of processing time are pointless. This is certainly the case for recognition systems such as the neocognitron (Fukushima, 1980; Fukushima et al., 1983) which, after the training phase, are hardwired, simple stimulus-response systems. In such a system implemented on completely parallel hardware, the processing time should indeed only depend on the time constants of the neurons and the propagation times. If we adopt the view that recognition needs an active matching process as proposed by the Dynamic Link Architecture, the situation is quite different. The only way we see to circumvent the feature ambiguities by introducing topological constraints is to work off the various locations sequentially. This requires more processing time if the neuronal layers involved get larger.

If a rough mapping can be established on the basis of few cells and the refinement steps already can build on this rough approximation to the final mapping, some parallelity can be reintroduced. This is necessary, because human object recognition can be, under the right circumstances, extremely fast. Experiments that are currently carried out by Irving Biedermann at the University of Southern California in Los Angeles suggest that object recognition is possible with a presentation time shorter than 70 milliseconds (Biederman et al., 1994). Although it is not clear that the complete recognition process is finished within the presentation time the brute force dynamic link matching gets its problems when realistic time constants are incorporated.

Psychophysical experiments by (Watt, 1987) demonstrate that coarse to fine processing indeed does occur when a stimulus is presented. Experiments in our institute have shown that for the special case of stereo matching the visual system can make use of coarse-to-fine strategies but does not always do so (Mallot et al., 1994).

The experiment proposed in figure 4.2 shows that recognition can be achieved on the basis of low-frequency information but is severely impeded if information on higher levels contradicts the right interpretation. In our context this can be interpreted as follows. A mapping process is initialized on a low frequency level and successively refined to higher ones. In the low-pass filtered image soon no further levels are available and the result is taken for a recognition. In the complete image, the patch boundaries introduce high frequency components that have nothing to do with the face. The visual system, however, doesn't notice the trick and recognizes a pattern of squares with different grey values. This is certainly not the only possible interpretation. Experiments would be interesting that explore the recognition of the unfiltered image with very short presentation times. Unfortunately this sort of experiments is probably hard to do once the subjects know the trick.

There is evidence that the dynamic link matching as well as its algorithmic caricatures (see sections 4.3, 7.1, 7.2.1, and chapter 5) receive much of their power from the use of local feature vectors. This means that the local elements of each of the two layers of the system described in section 4.3 must contain as many feature detectors as the length of the vector requires, and those must have one-one connections between the layers. In order

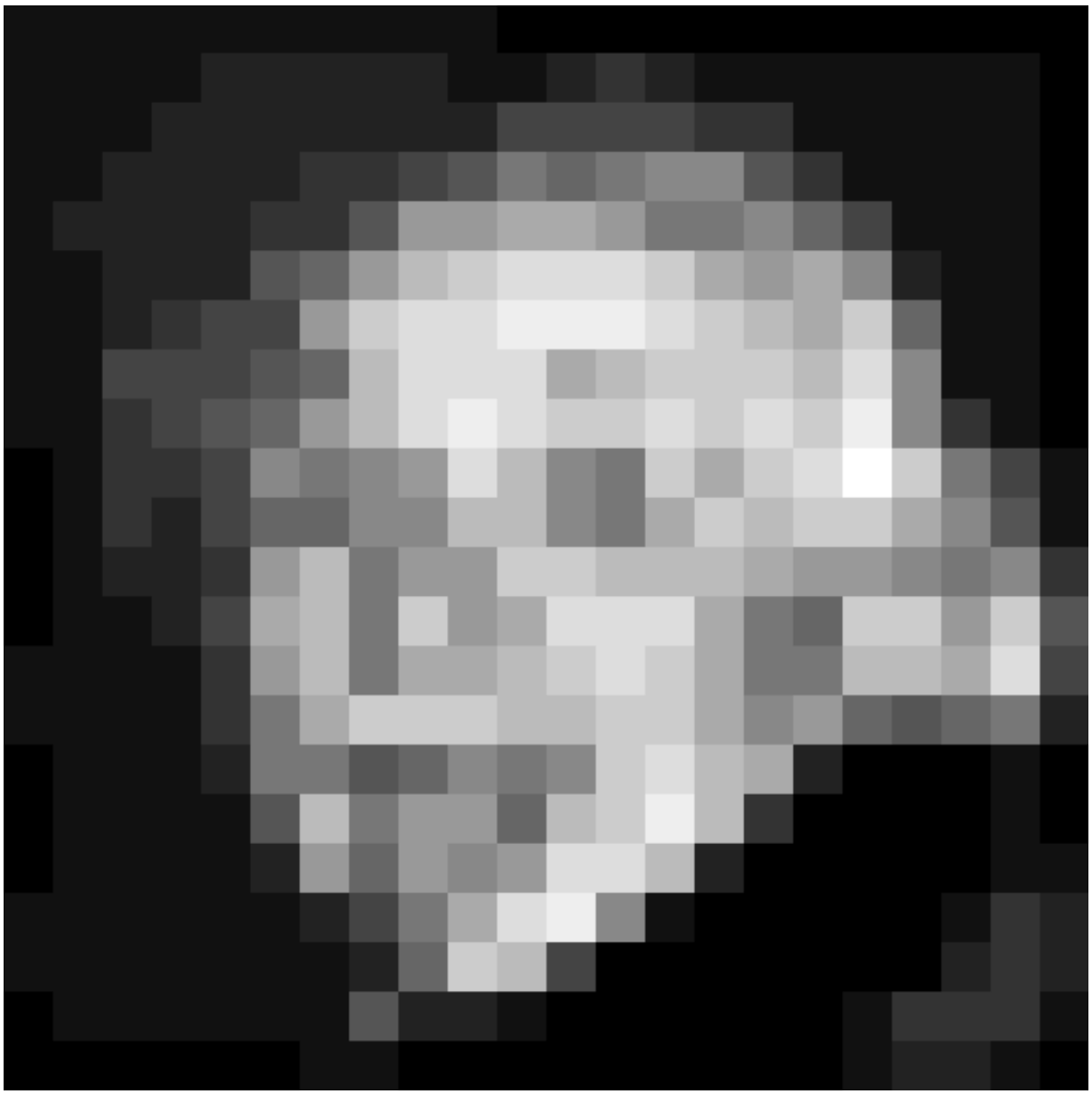

Figure 4.2: Experiment for multiresolution recognition. If watched from usual reading distance this person cannot be recognized. Introducing low-pass filtering by squinting or moving it to a distance of several meters from the eyes can lead to recognition.

for size-invariant matching to be possible this connectivity must extend to connections between each feature detector and the corresponding ones on all different scales. This is quite a lot of neuronal machinery, and it seems logical, within the framework of the correlation theory, that part of it is not hardwired but worked off sequentially. This is also proposed by the model presented here. The feature vectors include only the various directions on one scale, the different scales are treated sequentially.

There is another argument from the computational side. Due to the partially sequential nature of the dynamic link matching it is not only technical but applies also to the biological model. The matching in the dynamic link framework takes processing times proportional to the number of connections between layers if strict one-one mappings are required. This means a computational complexity of n^4 with n the linear size of a layer (Behrmann, 1993). This argument can be put the other way around: Short processing times on high resolution layers will produce spatially extended mappings. These may be viewed as an interpolation of more precise mappings on subsampled layers.

As a final argument I should like to turn the tables and state that the view of neuronal tissue as continuous fields is only a technical trick, because mathematics (or theoretical physics) without infinity is no fun at all. The brain is obviously (spatially) discrete, and there is no reason to believe that nature should waste resolution on information in low frequency bands. Although the statistical data about the distributions of receptive field properties in primary visual cortex seems to be too sparse to make an empiric judgment here, this sort of thrift has been established in the optic nerve (Wässle et al., 1986).

4.5 Layer Dynamics

The dynamics in this section follow the general model for the activity of neural layers (Wilson and Cowan, 1973; Amari, 1980). The special setup with the self-inhibiting and reciprocally connected layers has been developed by (Wiskott and von der Malsburg, 1994). There is so far no complete analytical treatment of these dynamics so we will restrict ourselves to a qualitative description and simulations.

The general form of dynamics for a neuronal layer will be the following:

$$\tau_a \frac{d}{dt} a(\vec{x}) = -a(\vec{x}) + c_\kappa \left(\kappa(\vec{x}) - c_g\right) * \vartheta(a(\vec{x})) + c_c - c_h h(\vec{x}) + c_s s(\vec{x}) + c_\xi \xi \quad (4.15)$$

$$\frac{d}{dt} h(\vec{x}) = \begin{cases} \tau_{h+}^{-1} \left(a(\vec{x}) - h(\vec{x})\right) & : \quad a(\vec{x}) > 0 \\ \tau_{h-}^{-1} \left(a(\vec{x}) - h(\vec{x})\right) & : \quad a(\vec{x}) \leq 0 \end{cases} \quad (4.16)$$

The single terms in the activity dynamics have the following meaning. In the absence of any connections (all terms beside $-a(\vec{x})$ are equal to zero) the activity decays to zero with the time constant τ_a. The kernel κ represents the internal connections in the layer in accordance to equation (4.3). The convolution represents the notion that the connection strength between two layer neurons depends only on their distance vector, not on their absolute positions. The treatment of the boundaries in this convolution requires extra consideration. The constant c_g reflects a global inhibition. It models a (linear) cell that collects the outputs from all layer neurons and inhibits all of them. The constant c_c is an input from an extra cell that has a constant activity. Beside the form of the kernel c_g and c_c are the most important parameters to get a desired behavior from the dynamics.

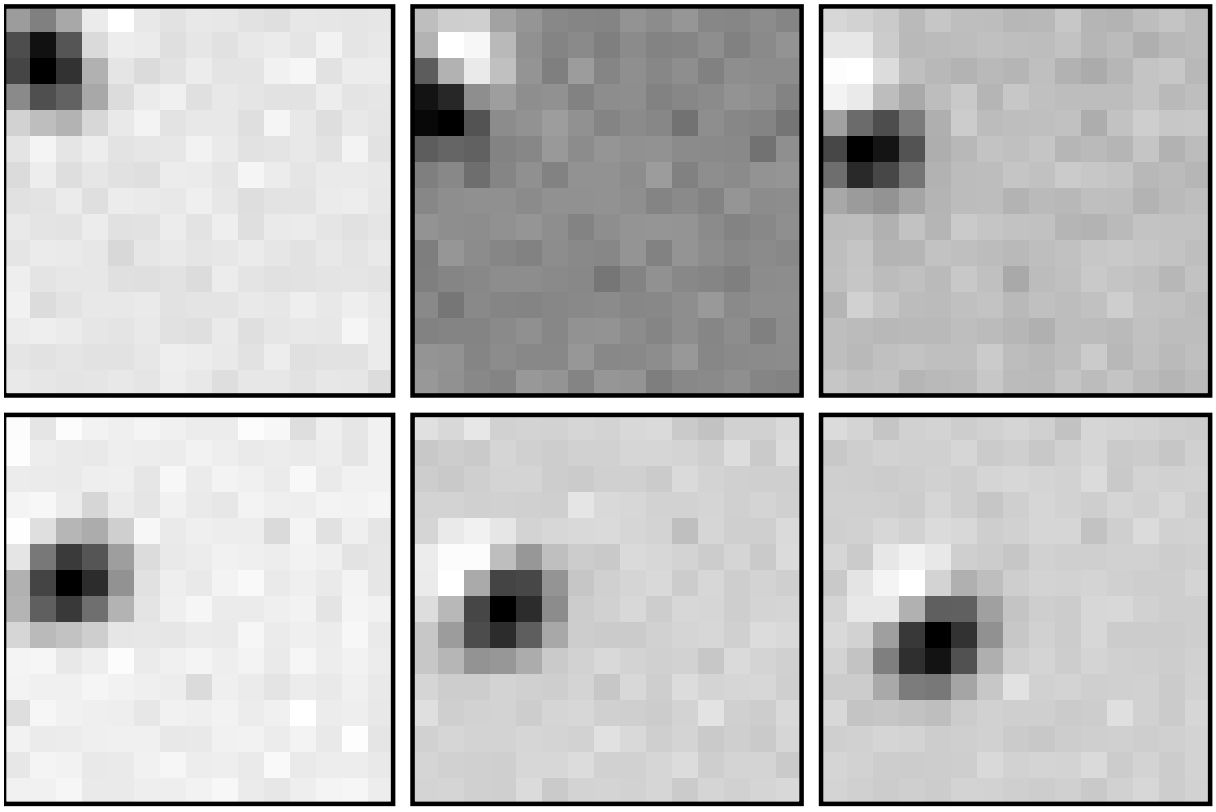

Figure 4.3: Layer dynamics on level 0. As a visualization of the dynamic activity on layer zero this figure shows six snapshots of a moving blob spaced by 10 simulation time steps.

The term $c_h h(\vec{x})$ models a delayed self-inhibition of the layer neuron, whose dynamics are described by equation (4.16). It converges to the activity itself with time constants τ_{h+} for positive activities and τ_{h-} for negative activities. These different time constants must be interpreted keeping in mind that the positive or negative activities are in fact a simplified description of several cells (see section 2.4 for a closer discussion).

The input $s(\vec{x})$ comes from the activities in a different layer, here denoted by the variable $\vec{y}$, and connected by the weight matrix W:

$$s(\vec{x}) = \int W(\vec{y}, \vec{x}) \cdot \vartheta(a(\vec{y}))\, d^2y \tag{4.17}$$

For all our simulations $\vartheta(a)$ will be the function specified in equation (4.5).

ξ is a noise term which is meant to describe spontaneous activities of neurons and the influence of possible connections from other neurons which are not part of the model. Technically, it is needed for symmetry breaking between different possible solutions. Its presence in the model shows the robustness of the dynamics. This is important for a biologically plausible model, because in the brain the subsystems are certainly not as nicely separated that other processes do not influence the neurons at all.

Even in the absence of external input $c_s = 0$ these layer dynamics can show a universe of different behaviors. We will list only some of them that are used in the hierarchical

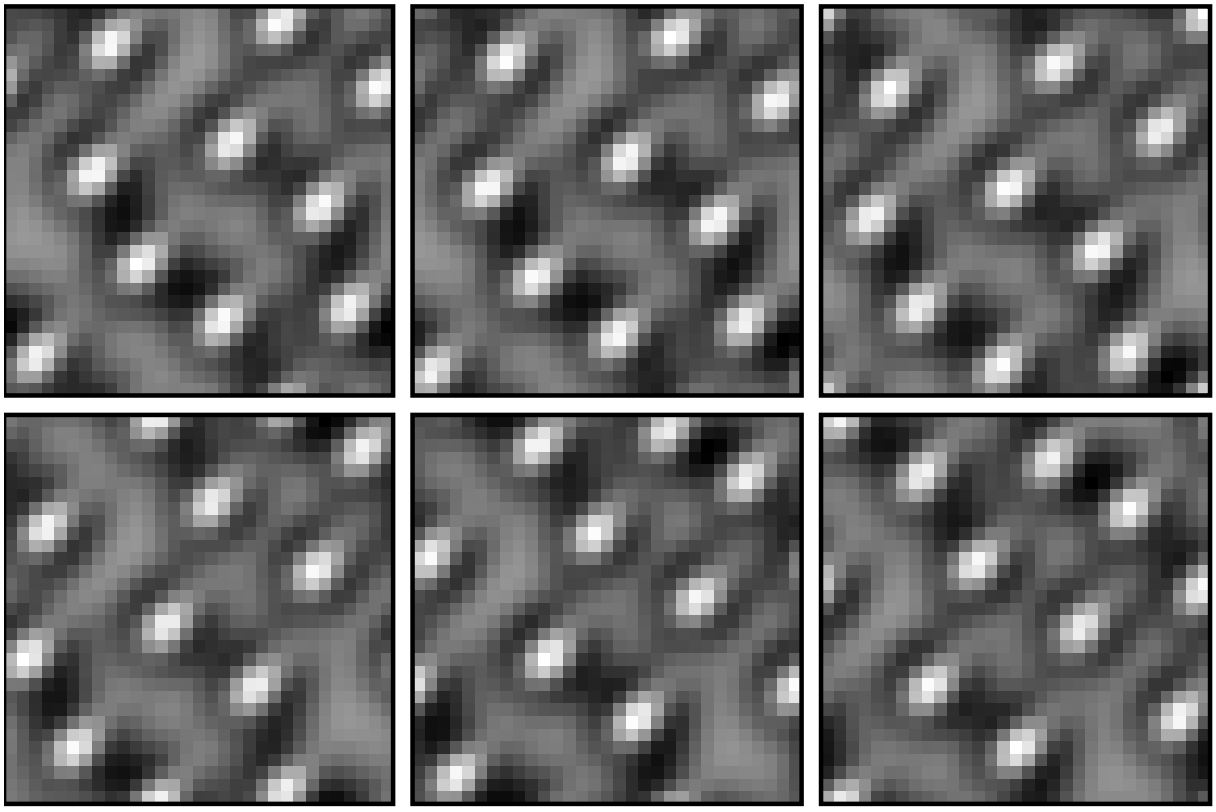

Figure 4.4: Layer dynamics on level 1. On the higher levels several blobs move coherently across the layer to strengthen links that have good feature similarity and are supported by the links from level zero. The six snapshots are 5 simulation time steps apart.

---

model. First we qualitatively describe two important cases which converge to a stationary state.

One stationary blob If the kernel is chosen to be a Gaussian centered at zero, the global inhibition c_g and the constant input c_c take suitable positive values, and self-inhibition as well as the external input are zero, the system converges to a stationary blob like already discussed in section 4.3.

Several stationary blobs Replacing the Gaussian kernel (and the global inhibition) by a difference of Gaussians which has the same form as in equation (4.24) leads to more complicated patterns. Depending on the parameter settings those include stripes of positive activity or a multitude of small blobs in a relatively regular arrangement. It is the latter case that will be of interest to us.

The parameter settings with kernels centered at zero and without self-inhibition converge to steady states and are not useful for a system which actively covers all regions of image and model in order to establish strong links between corresponding points.

There are two quite different approaches to make the self-organized patterns move. The first one is the delayed self-inhibition ($c_h > 0$). Once a blob has formed the self-inhibition builds up and keeps it from staying in the same position. The layer dynamics, however, insist on building up the blob. Two different things can happen. The first possibility is that the blob moves smoothly to a neighboring location. This is the case in the complete absence of external input to the layer. The direction of the movement is arbitrary, i.e. influenced only by the noise or the external inputs. If the input is inhomogeneous it also occurs that the blob decays and immediately builds up in a different location. Both behaviors will be important. An in-depth discussion of this dynamics can be found in (Wiskott and von der Malsburg, 1994).

The second possibility uses interaction kernels whose maximum is shifted away from the zero position. This shift does not influence the shape of the blob pattern. Due to the fact that an active cell excites its neighbor more than itself, the pattern will move in the direction of the shift vector. This way of forcing a pattern to move across the layer is more appropriate in the multiblob case, because it leads to a coherent motion of all blobs.

The first kind of dynamics (one blob driven by delayed self-inhibition) will be used on level 0 to find the subset of the image that corresponds to the model. The second kind (several blobs driven by a shifted kernel) governs the higher levels.

4.6 Weight Dynamics Between Two Layers

Local elements of two layers of the same resolution level are interconnected by a pair of links. These are certainly not simple synapses but sets of synapses that interconnect the cells constituting the local elements. Nevertheless, they are treated as simple dynamic variables.

As described in section 4.1.3 their dynamics are a combination of Hebb's rule and competition on both source and target cells.

$$\tau_W \frac{d}{dt} W(\vec{x},\vec{y}) = W(\vec{x},\vec{y})\,\mathrm{Corr}(\vec{x},\vec{y})\,, \tag{4.18}$$

$$\int W(\vec{x},\vec{y})\,d^2x \le 1\,, \tag{4.19}$$

$$\int W(\vec{x},\vec{y})\,d^2y \le 1\,. \tag{4.20}$$

The correlation Corr is not a correlation in the mathematical sense but a measure for the coherent activity of the neurons as well as their attached feature detectors. This means it incorporates the feature similarity as well as the synchronicity of activities. The feature vector is the vector of all response amplitudes of units located at the respective position and having a center frequency belonging to the actual level. We give only the formula for the feature similarities here, its form will be discussed in detail in section 5.2.1.

$$\mathcal{S}(\vec{f},\vec{g}) := \begin{cases} 0 & : \quad \vec{f}=\vec{0} \;\text{ or }\; \vec{g}=\vec{0} \\ \left(\frac{\vec{f}^{\mathsf{T}}\vec{g}}{|\vec{f}||\vec{g}|}\right)^4 & : \quad \text{otherwise}\,. \end{cases} \tag{4.21}$$

This definition differs from the one used in chapters 5 and 6 only in the exponent 4.

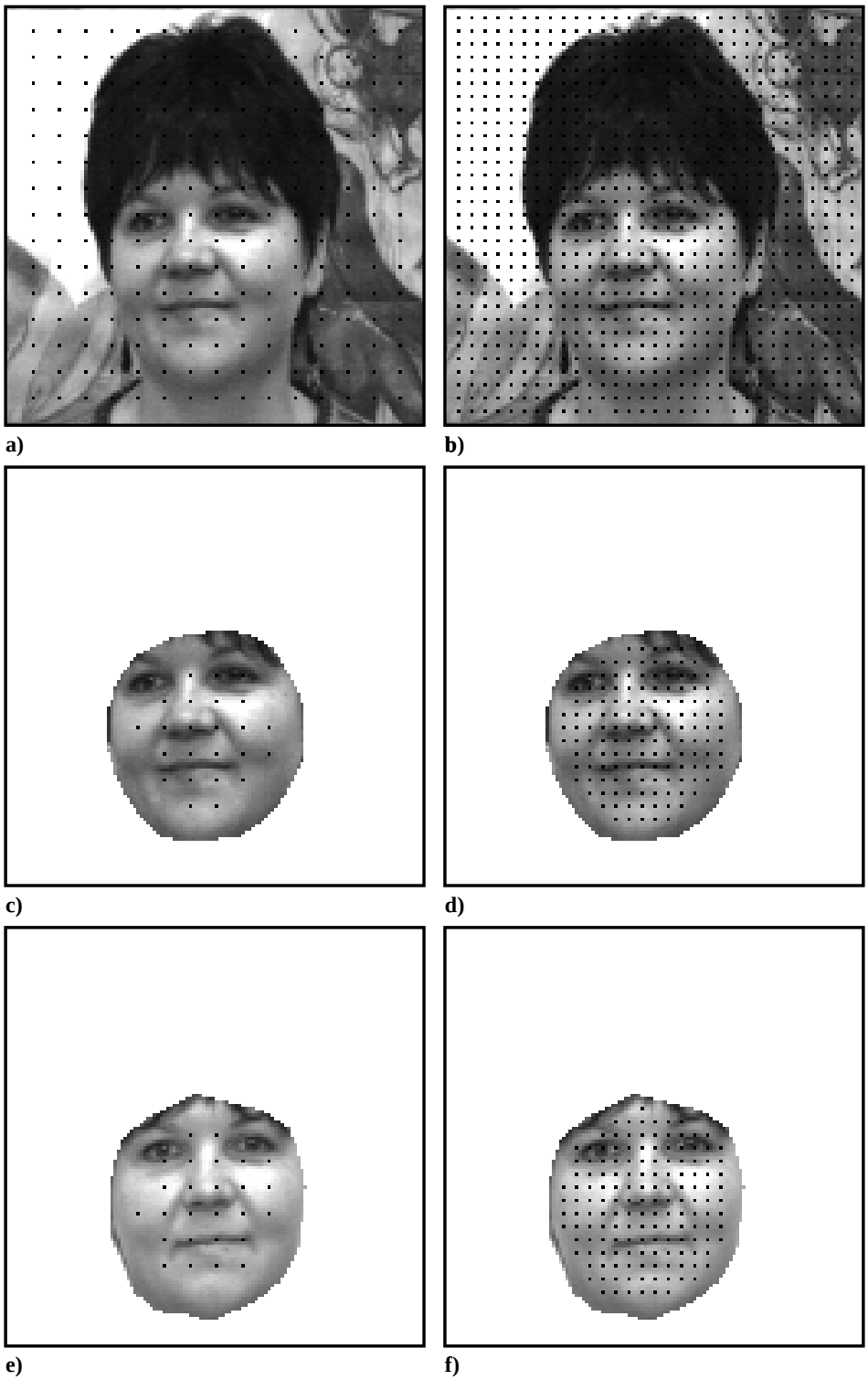

Figure 4.5: The location of the layer neurons. The black spots mark the location of the layer neurons in image and model, respectively. **a)** and **b)** show the image representations on level 0 and 1. The model representation shown in **c)** and **d)** was used for the simulation with identical pictures, the one in **e)** and **f)** for the simulation with different ones. The layers are rectangular, those neurons without a location in the model representation are part of the layer dynamics but do not make or receive dynamic links.

This exponent would not be absolutely necessary here, but it makes the maxima in the similarity landscape sharper, and the dynamics can converge faster.

4.7 The Complete Model

After having discussed previous models for dynamic link matching for the solution of the correspondence problem and the qualitative properties of the dynamical systems involved we will now present a complete model that finds correspondences hierarchically, i.e. starting with low frequency information and low resolution and then refining the found mapping to higher resolution using the next frequency level. This model can be extended to several frequency levels — as a proof of concept and given the limitation of computational resources we restrict ourselves to the initialization and one refinement step for the actual simulations.

Starting from the representations described in chapter 3 each frequency level is represented by a pair of model and image layers which are interconnected by a full matrix of dynamic links in each direction. This means that each feature vector from the levels of a representation is assigned a layer neuron with the capability to connect via dynamic links to the respective other layer. If $c_h \neq 0$, which will be the case in the layers belonging to the lowest frequency level, this layer neuron comes equipped with a pair of neurons that mediate the delayed self-inhibition. For simplicity of simulation, the layers are rectangular, even if some locations in that rectangle do not have a feature vector due to background suppression or amplitude thresholding. Those locations take part in the layer dynamics but can not build dynamic links. The layers for the simulations are shown in figure 4.5.

The coupling of the levels is done in the following way. The link dynamics on level $n+1$ are constrained such that the links can can only grow once some link on level n exceeds a threshold. Then their growth rates are determined by a combination of the correlations of their local elements and the correlated activity of the topology cells on level n.

4.7.1 Rough Mapping with low frequencies

The layer dynamics on the lowest frequency level follow equations (4.15) and (4.16). The self-inhibition is positive, the kernel a Gaussian centered at zero:

$$\kappa_0(\vec{x}) := \exp\left(-\frac{\vec{x}^{\mathsf{T}}\vec{x}}{2\rho^2}\right) \tag{4.22}$$

The width of the kernel and the global inhibition are adjusted such that a blob results with an area of roughly one quarter of the size of the model layer. Due to the self-inhibition these blobs move across model and image layer, respectively. In the absence of strong dynamic links their movements are not correlated.

The link dynamics are governed by equation (4.18), the correlation on this level takes the form:

$$\mathrm{Corr}_0\left(\vec{x}_n, \vec{y}_n\right) = o\left(\vec{x}_n\right) o\left(\vec{y}_n\right) + c_{\mathcal{S}} \frac{1}{9} \sum_{3\times 3} \mathcal{S}\left(\vec{f}\left(\vec{x}_n\right), \vec{f}\left(\vec{y}_n\right)\right) , \tag{4.23}$$

where the sum runs over a 3×3 neighborhood.

In the beginning both blobs move freely and independently on their corresponding layers. Correlations make some links between the layers grow, others decay. After some time, the links have become strong enough that the image blob can only exist inside the region which corresponds to the model. From then on, the blob decays outside this region after a while and spontaneously reforms inside the region. When the links have grown even stronger, the image blob does not leave the region any more, and the correct links grow until a one-one mapping has been reached.

4.7.2 Mapping Refinement

For the layer dynamics on the nth level the interaction kernel is a difference of Gaussians with a maximum slightly off zero:

$$\kappa_n(\vec{x}) = \sigma_- \cdot \exp\left(-\frac{(\vec{x}-\vec{x}_0)^2}{2\sigma_+^2}\right) - \sigma_+ \cdot \exp\left(-\frac{(\vec{x}-\vec{x}_0)^2}{2\sigma_-^2}\right) . \tag{4.24}$$

This leads to the desired behavior of several blobs moving across the layer. If $\vec{x}_0$ is chosen appropriately, the blobs reach every point of the layer. This choice is, of course, dependent on the layer size and, consequently, on the level number n.

The link dynamics are partly the same as on the level $n-1$. The difference is that they are influenced by the links on that level. This is necessary for these links to refine the mapping built on the lower level. The influence is twofold. The value of the maximal link on level $n-1$ must reach a threshold c_t to trigger the link dynamics on level n. After this, the correlation on level n is given by a weighted sum of the correlation of the outputs on level n, the feature similarities on the same level, and the correlation of the outputs on the lower level $n-1$ of those cell pairs, whose dynamic link exceeds the threshold c_t. The latter is accounted for by the Heaviside function Θ.

$$\begin{aligned} \mathrm{Corr}_n\left(\vec{x}_n, \vec{y}_n\right) = \quad & o\left(\vec{x}_n\right) o\left(\vec{y}_n\right) + c_{\mathcal{S}} \mathcal{S}\left(\vec{f}\left(\vec{x}_n\right), \vec{f}\left(\vec{y}_n\right)\right) \\ & + c_{low} \Theta\left(W\left(\vec{x}_{n-1}, \vec{y}_{n-1}\right) - c_t\right) o\left(\vec{x}_{n-1}\right) o\left(\vec{y}_{n-1}\right) . \end{aligned} \tag{4.25}$$

From the start of the simulation the blobs wander across the model and image layer. Once the mapping on level $n-1$ has developed far enough for the first link to reach threshold their links start developing, too. Only the ones that connect neurons with good feature similarity *and* good correlated activity on the same locations on level $n-1$ are competitive and eventually reach threshold to trigger the link dynamics on the next level up.

4.8 Simulations

The complete model has been simulated using an Euler discretization of the dynamical system on levels 0 and 1. The time discretization has been $\Delta t = 0.3$. The noise term has been modeled by random numbers with the appropriate prefactor $1/\sqrt{\Delta t}$. The link dynamics, which use most of the simulation time have been updated only every third time

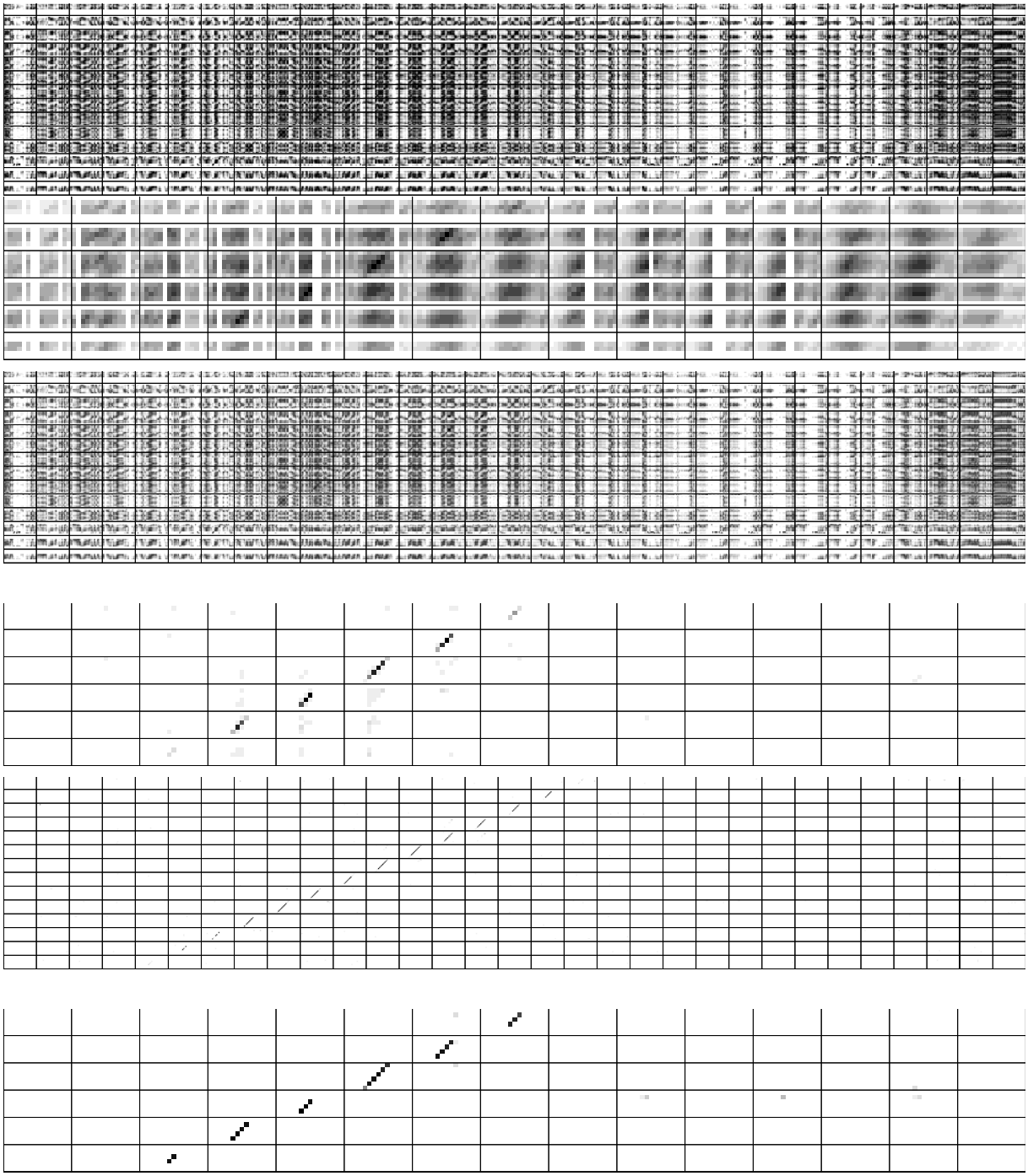

Figure 4.6: The development of the dynamic links for identical images. In the beginning (top two figures) the links reflect only the feature similarities, which are highly ambiguous. After 270 time steps the lower level has sorted out the correct correspondences, and the first links have grown above the threshold where they are allowed to influence the higher frequency level. In the bottom figures (a snapshot after 1000 time steps) the links on both frequency levels are restrained to the correct correspondences. See section 4.8.1 for an explanation of the visualization method used for the link structures.

step. The product of the outputs of the topology cells, which is the time variant part of the correlations driving the dynamics have been added up over these three steps.

The simulations started with 100 time steps for the layer dynamics to reach their behavior from an initialization with random numbers between 0 and c_ξ. Then the link dynamics were turned on. The convolutions that model the intralayer connections were simulated with zero padding on level 0 and with wrap around on the higher levels.

The links were initialized to the feature similarities divided by the number of elements in the larger layer. Thus, they were already primed a little bit towards supporting the ones with good similarities. To avoid underflow and the occurrence of links with strength exactly zero that are unable to grow anymore the links always keep a minimal value of $2^{-7}/N$ with N the total number of links between the respective layers.

The inequalities (4.19) and (4.20) have been enforced by dividing every link by the total of incoming (outgoing) links after each update step of the link dynamics, if this total exceeded the threshold 1.

4.8.1 Visualization of the Link Structures

The distribution of links between two layers is a four-dimensional structure which is hard to visualize on a screen or on paper. Therefore, they are represented as follows. Both layers are cut up into horizontal lines, which are then reassembled as a long, one-dimensional array, starting with the bottom line and ending with the top one. The links between these arrays form a two-dimensional matrix. Their values can then be represented as grey values, the minimal value in the matrix corresponding to white, the maximal one to black.

Figures 4.6 and 4.7 show such link matrices. In order to make the evaluation (a little) easier the single horizontal lines have been separated by black lines. Thus each little rectangle contains the links between one line of the model layer and one of the image layer. A one-one mapping between a model line and a contiguous part of an image line shows up as a diagonal within the rectangle.

4.8.2 Choice of Parameters

Here we present the complete set of parameters that led to the results described in this chapter. The behavior of the system allows considerable changes before it changes qualitatively.

For the layer dynamics on **level 0** the interaction kernel was a Gaussian of 0.8 in width (measured in units of pixels). The other parameters were:

$$\tau_a = 1 \tag{4.26}$$
$$c_\kappa = 3 \tag{4.27}$$
$$c_g = 0.015 \tag{4.28}$$
$$c_c = 0.1 \tag{4.29}$$
$$c_h = 1.2 \tag{4.30}$$
$$c_s = 0.8 \tag{4.31}$$
$$c_\xi = 0.01 \tag{4.32}$$

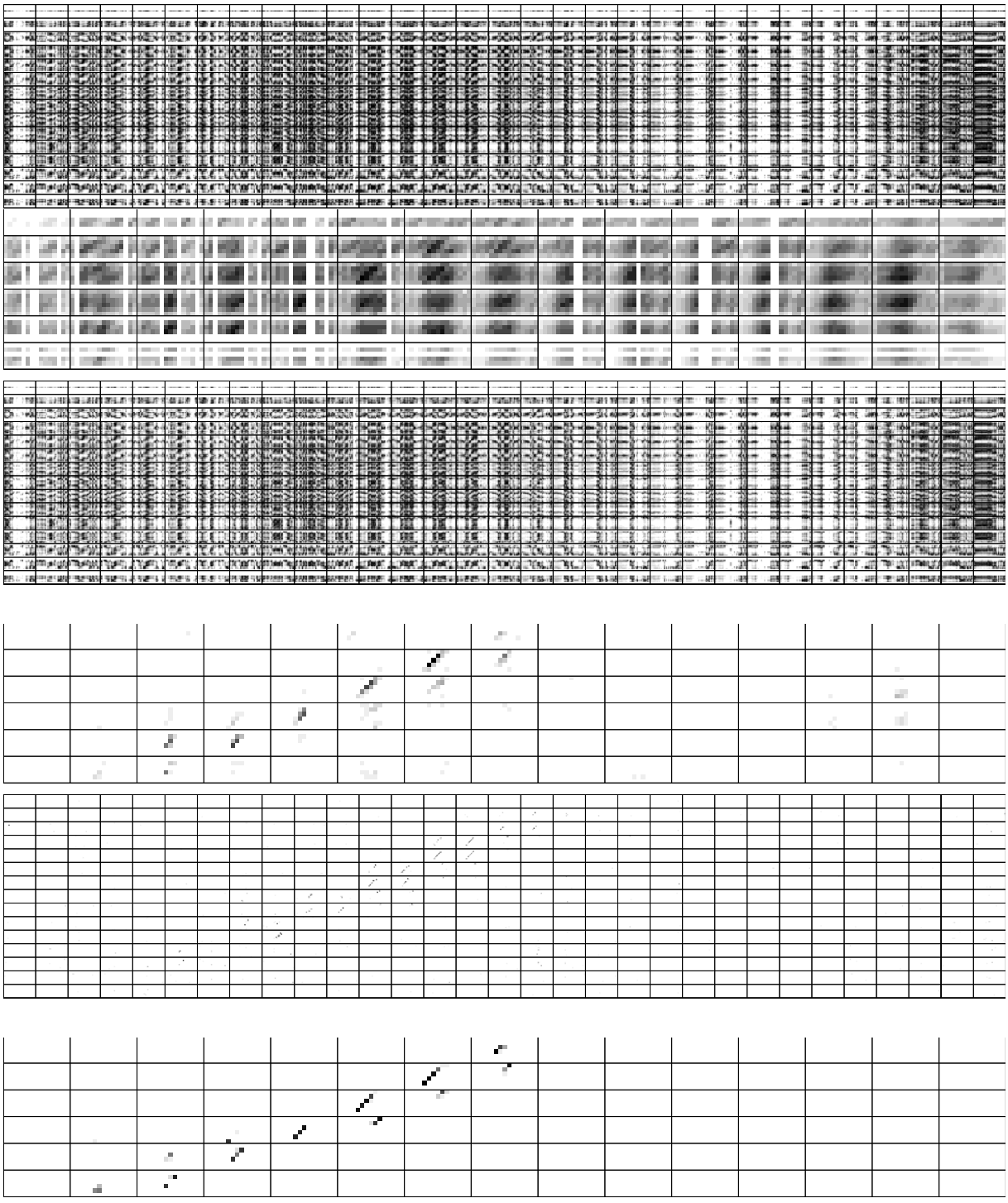

Figure 4.7: The development of the dynamic links between different images. For different images the feature ambiguities are, of course, worse than in figure 4.6. As a consequence, the network takes more time to establish the correct correspondences, and some erroneous correspondences survive. The link matrices shown are the states at time steps 0, 390, and 1000. See section 4.8.1 for an explanation of the visualization method used for the link structures.

$$\tau_{h+} = 1 \tag{4.33}$$
$$\tau_{h-} = 0.3\,. \tag{4.34}$$

The link dynamics on level zero are influenced by the relative weight of the feature similarity $c_{\mathcal{S}}$ and the learning rate τ_W. The values for those are:

$$c_{\mathcal{S}} = 1.5 \tag{4.35}$$
$$\tau_W = 0.1\,. \tag{4.36}$$

On **level 1** the interaction kernel was a difference of Gaussians with $\sigma_+ = 1.5$ and $\sigma_- = 2.0$. The coordinates of the maximum were chosen to be 0.3 and 0.1 in horizontal and vertical direction, respectively. The delayed self-inhibition was turned off ($c_h = 0$), thus the constants τ_{h+} and τ_{h+} do not appear in the parameter list:

$$\tau_a = 1 \tag{4.37}$$
$$c_\kappa = 3 \tag{4.38}$$
$$c_g = 0 \tag{4.39}$$
$$c_c = 0 \tag{4.40}$$
$$c_h = 0 \tag{4.41}$$
$$c_s = 3.5 \tag{4.42}$$
$$c_\xi = 0.01\,. \tag{4.43}$$

The definition of the correlation on level 1 includes the threshold c_t for the links on the lower level and the strength of the influence of the lower level c_{low}. The complete list of parameters here is:

$$c_{\mathcal{S}} = 1.5 \tag{4.44}$$
$$\tau_W = 0.1 \tag{4.45}$$
$$c_t = 0.5 \tag{4.46}$$
$$c_{low} = 1\,. \tag{4.47}$$

4.9 Results

Figures 4.6 and 4.7 demonstrate that the dynamics described above manage to solve the correspondence problem on the basis of the model and image representations of chapter 3. The results are perfect when model and image are taken from identical pictures (figure 4.6). Due to the massive feature ambiguities this is not as trivial as it may sound. For different pictures the refined mapping still contains some erroneous links (figure 4.7). The reasons for that will be discussed below.

The system has the following generalizations and improvements compared with other models for dynamic link matching, which are described in (Konen and Vorbrüggen, 1993; Rinne, 1994; Wiskott and von der Malsburg, 1994). First it is able to deal with a structured background. This is very important for a model to work under realistic circumstances. The other models only work with layers of equal size and therefore get severe problems if the backgrounds are different in model and image.

Secondly, the hierarchical structure leads to shorter convergence times. The dynamics on the lowest level can converge relatively rapidly because the layer sizes are small. On the larger layers assigned to the higher levels the longer convergence times are shortened by parallel sorting out of suboptimal correspondences in the multiple blob dynamics. Due to the fact that the link dynamics on the higher levels are triggered a considerable amount of time before convergence on the lower level (i.e., when the first link reaches half its maximal value) the link dynamics on the different levels run partly in parallel which reduces the convergence time further. The two intermediate states in figures 4.6 and 4.7 show that the link structure on level 0 still changes substantially when the dynamics on level 1 have already started.

Thirdly, it has been shown that the information on the lowest frequency level suffices to obtain a coarse initialization of the model-image correspondences. Consequently, the model works in accordance to the experiment described in section 4.4 and thus captures at least some properties of human visual cognition.

It can be concluded that the goal to build a dynamical system which uses dynamic links on the basis of short-time correlations and does not show the limitations of earlier systems concerning background influence and convergence time has been reached.

Some minor shortcomings of this system remain to be discussed. In the case of different images (figure 4.7) some erroneous links remain on level zero. This is mainly a problem of the layer dynamics, because the multiple blobs on this layer are not suppressed in areas without correspondence. Some further research will be necessary to find layer dynamics which produce multiple moving blobs *and* a possibility to suppress them in areas already ruled out by the link dynamics on the lower level.

Also the link distribution does not show the desired clear diagonals within all the little rectangles. This is partly a (desired) consequence of distortions between the pictures. A major reason for the weaker performance, however, is that excellent correspondences can not be established because the sampling is too sparse. If the best correspondence to a model point would lie between four image points weaker links will establish to all of these points. Nevertheless, the dynamics are able to reduce the initial ambiguities to a relatively ordered state. In the next chapter we will find an elegant way to circumvent the problem of the sparse sampling by using the phase information of the feature vectors. However, it is not clear yet how this phase matching can be carried out by neural dynamics.

5. Algorithmic Pyramid Matching

Du hast wohl recht; ich finde nicht die Spur
von einem Geist und alles ist Dressur.

Johann Wolfgang von Goethe, Faust

5.1 Matching of Phase Space Molecules

In this chapter we will abstract from the self-organizing machinery and drop the possibility of keeping several links active from the same model location. This is motivated by the fact that using sequential workstations the self-organization cannot exhibit its full power and simulations are too slow to exploit the mappings for recognition out of a large database.

The basic task executed by the combined blob/weight dynamics is to find the location in the image where the configuration of features is identical (or at least similar) to the configuration of model features highlighted by the blob. In computer vision this problem is commonly solved by a procedure called *template matching*, which will be described in section 5.2.

Although solving the correspondence problem would mean to find the corresponding pairs of phase space atoms this cannot be achieved by comparing the atoms alone, because there is too little information in a single (scalar) response to disambiguate between many possible pairs. Therefore, matching must rely on suitable phase space molecules. One possible choice of molecules are the jets described in section 7.1.2. Here we will use smaller ones that are restricted to center frequencies of a fixed length. The great advantage of those is that for larger frequencies their spatial extent gets smaller and smaller. Therefore, moving up gradually with the center frequency, they lead to very good approximations of spatial one-one mappings.

If we work with a uniform sampling set (which is the case for single frequency levels) or simple thresholding a small local feature vector describes the texture or the image structure close to the corresponding point. The matching procedure amounts to finding corresponding texture elements. This cannot be the most general recognition scheme because it would completely fail in the absence of texture. Nevertheless, it is sufficient for human faces, and we will study its performance in detail. For applications with object classes with little texture, the choice of molecules as well as the local similarity functions must be modified, but it can be speculated that the dynamic link matching as well as the matching described here would work equally well. For the dynamic link matching this has been partly shown in (Rinne, 1994).

5.2 Template Matching

In this section we will reduce the segmentation and refinement steps of the matching procedure to the central mechanism of template matching, which can be well formalized and easily and efficiently implemented. The task is to find a copy of a small pattern (a *template*) in a larger function f, which will be referred to as *data* in the apparent absence of a better term. A template $t(\vec{x})$ is an arbitrary $\mathcal{L}^2$-function with finite support, the corresponding data any $\mathcal{L}^2$-function. What is required is a displacement vector $\vec{y}$ such that:

$$t(\vec{x}-\vec{y}) \neq 0 \Longrightarrow f(\vec{x}) = \lambda t(\vec{x}-\vec{y}) \tag{5.1}$$

Allowing an arbitrary factor λ gives invariance under global changes in contrast.

Two functions are equal up to the constant factor λ if and only if their normed scalar product is equal to one. This property is applied to the template and the restriction of the data to the support of the shifted template. To ensure the latter, f must be multiplied with the characteristic function of the support of $t(\vec{x}-\vec{y})$. This can be omitted in the scalar product but not in the norm of f in the denominator. Thus, we define the following function:

$$\mathcal{S}(f,t)(\vec{y}) = \frac{\langle f(\vec{x}) \mid t(\vec{x}-\vec{y})\rangle_{\vec{x}}}{\|t(\vec{x})\|_{\vec{x}} \cdot \left\|f(\vec{x})\chi_{\mathrm{supp}(t)}(\vec{x}-\vec{y})\right\|_{\vec{x}}}. \tag{5.2}$$

Now, condition (5.1) is fulfilled if and only if $\mathcal{S}(f,t)(\vec{y}) = 1$ for some $\vec{y}$. In real-world applications this equality will never be exact. Nevertheless, $\mathcal{S}(f,t)(\vec{y})$ can be used as a similarity function. Its maximum (a value between -1 and 1 and close to 1 in the presence of a reasonable match) reflects the maximal similarity of the template to a part of the data. We define as *template matching* any procedure which delivers the displacement $\vec{y}$ (or possibly several of them) where $\mathcal{S}(f,t)(\vec{y})$ achieves its maximum. All subtleties about compact domains or the like to guarantee the existence of such a $\vec{y}$ will be disregarded because finally all this will be applied only to discretized domains.

The use of the scalar product here results from the necessity to derive a *global* similarity measure from the *local* similarities given by the simple product of the function values. From this point of view the idea of template matching can be easily extended to compare vector-valued functions by first defining a local similarity function $\mathcal{S}_{loc}$ between the vectors, extending this to a modified scalar product and then applying the same formula.

5.2.1 Local Similarity Function

For template matching purposes throughout this work the similarity between two vectors, or the pointwise similarity of vector-valued functions will be defined as follows:

$$\mathcal{S}_{loc}(\vec{f},\vec{g}) = \begin{cases} 0 & : \quad \vec{f}=\vec{0} \ \text{ or } \ \vec{g}=\vec{0} \\ \frac{\vec{f}^{\mathsf{T}}\vec{g}}{|\vec{f}||\vec{g}|} & : \quad \text{otherwise}\,. \end{cases} \tag{5.3}$$

The scalar product in the numerator is a straightforward extension of the simple product between real (or complex) numbers. The normalization is motivated by the

following properties which hold for any vector $\vec{g}$:

$$\mathcal{S}_{loc}(\vec{g},\vec{g}) = \begin{cases} 0 & : \quad \vec{g} = \vec{0} \\ 1 & : \quad \vec{g} \neq \vec{0} \end{cases} . \tag{5.4}$$

Furthermore, two nonzero vectors are equal up to a constant factor if and only if their local similarity is equal to one. Thus, two vector-valued functions are equal up to a spatially variant factor if and only if the local similarities are equal everywhere on the intersection of the supports of their moduli. This makes some things a lot easier. The possible factors allow for local contrast changes and alleviate illumination problems.

5.2.2 Multidimensional Template Matching

In order to transfer equation (5.2) to the multidimensional case the pointwise product used in the scalar product of two functions must be replaced by $\mathcal{S}_{loc}$. This means that the numerator of the right hand side in equation (5.2) takes the form:

$$\int \mathcal{S}_{loc}\left(\vec{f}(\vec{x}), \vec{t}(\vec{x}-\vec{y})\right) \, d^2x \, . \tag{5.5}$$

Using the fact that the norms in the denominator are the scalar products of the respective arguments with themselves, the first one turns into:

$$\int \mathcal{S}_{loc}\left(\vec{t}(\vec{x}), \vec{t}(\vec{x})\right) \, d^2x \tag{5.6}$$

$$= \int \chi_{\mathrm{supp}(|\vec{t}(\vec{x})|)} \, d^2x \, . \tag{5.7}$$

This equality is a consequence of property (5.4) and shows that the first norm in the denominator is the size of the area where $\vec{t}(\vec{x})$ is nonzero.

The second norm in the denominator of equation (5.2) turns into:

$$\int \mathcal{S}_{loc}\left(\vec{f}(\vec{x})\chi_{\mathrm{supp}(|\vec{t}|)}(\vec{x}-\vec{y}), \vec{f}(\vec{x})\chi_{\mathrm{supp}(|\vec{t}|)|}(\vec{x}-\vec{y})\right) \, d^2x \tag{5.8}$$

$$= \int \chi_{\mathrm{supp}(|\vec{t}(\vec{x}-\vec{y})||\vec{f}(\vec{x})|)} \, d^2x \, . \tag{5.9}$$

This term is identical to the size of the area where the shifted template *and* the corresponding values of $\vec{f}$ are nonzero.

Assembling the terms (5.5), (5.7), and (5.9) we are ready to define the procedure of *multidimensional template matching*, or *MTM*. It consists of finding the displacement $\vec{y}$ (or possibly several of them) where the following function achieves its maximum:

$$\mathcal{S}\left(\vec{f},\vec{t}\right)(\vec{y}) = \frac{\int \mathcal{S}_{loc}\left(\vec{f}(\vec{x}), \vec{t}(\vec{x}-\vec{y})\right) \, d^2x}{\int \chi_{\mathrm{supp}(|\vec{t}(\vec{x})|)} \, d^2x \cdot \int \chi_{\mathrm{supp}(|\vec{t}(\vec{x}-\vec{y})||\vec{f}(\vec{x})|)} \, d^2x} \, . \tag{5.10}$$

5.2.3 Implementation of Multidimensional Template Matching

There are several efficient methods for the implementation of (scalar) template matching, including FFT-methods and gradient-based methods. In our case, the templates as well as the corresponding data will be relatively small (see sections 5.3.1 and 5.3.2). As a consequence, it is sufficient to find the optimal shift vector by exhaustive search, i.e. the template is positioned at every possible location within the given discretization, the similarity evaluated and the optimal shift determined.

If (for an application) the initial estimation of the mapping has to be carried out on a much larger data area, more sophisticated methods for the matching may be considered. On the other hand, this matching constitutes only one crude cue for the segmentation task; for recognition procedures supposed to work well in large images under realistic conditions many segmentation cues must be combined into one powerful segmentation method to precede any attempt to the solution of the correspondence problem. This is not our task here, for the description of such a system, which can be easily combined with the procedure described here, the reader is referred to (Vorbrüggen, 1994).

5.2.4 Choice of Multidimensional Templates

As pointed out at the beginning of this chapter the scalar responses of scale space atoms do not suffice to disambiguate between the many possible correspondences. Obviously, the use of phase space molecules with vectors of responses will, at least statistically, alleviate the problem. This made it necessary to extend the notion of template matching from scalar-valued functions to vector-valued ones. Here we will define explicitly how the multidimensional templates are retrieved from representations as defined in chapter 3. Given a representation $\mathcal{R}$ of either an image or a model $\vec{h}(\vec{x})$ shall be the vector of all complex unit responses located at $\vec{x}$. This means, that the two frequency components are, for notational simplicity, mangled into one single component. How this mangling is done is of no importance, because all our further manipulations will be invariant under permutation of components. However, it is of course important that it be done in a consistent way in both image and model representation. This implies that the components must cover all the center frequencies present in image *or* model representation, if some of them should be missing from either one.

As a consequence of this definition there will possibly be undefined components in $\vec{h}(\vec{x})$. Good care has to be taken of the treatment of these missing components. In our implementation, they will be coded as having a response amplitude which is precisely zero. This is possible only because the amplitudes are nonnegative and amplitudes close to zero have been eliminated (see section 3.3), and therefore makes the matching procedures less general than they ought to be. Nevertheless, special treatment of those components, e.g. as NaNs (not a number) would have increased the programming effort as well as the execution times by untenable amounts.

In the following, the complex vectors $\vec{h}(\vec{x})$ will again be separated into amplitude and phase, these will be referred to as $\mathcal{A}(\vec{h}(\vec{x}))$ and $\mathcal{P}(\vec{h}(\vec{x}))$. With all the mentioned precautions , these are well-defined entities for all representations and for each $\vec{x}$ which is part of the representation. For other values of $\vec{x}$ $\mathcal{A}(\vec{h}(\vec{x}))$ and $\mathcal{P}(\vec{h}(\vec{x}))$ will be zero.

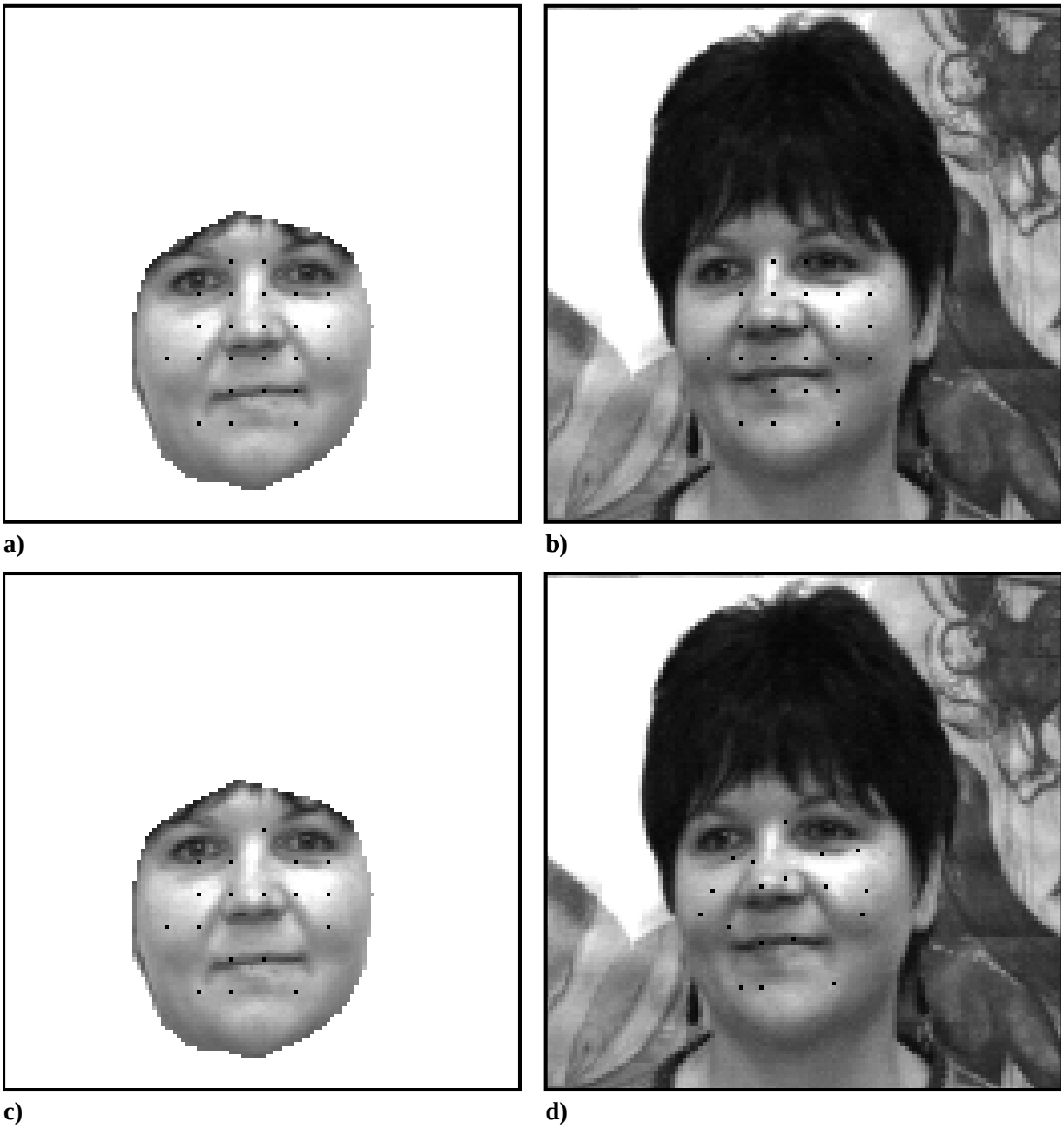

Figure 5.1: Mappings on the lowest level. $\left(|\vec{k}| = 0.4\right)$ **a)** and **b)** show the mapping points from the MTM, in model and image. In **c)** and **d)** the phases have been matched and the points with poor similarity have been discarded from the mapping. The mapping from **c)** to **d)** is used for further refinement.

The representation of the local feature vectors as amplitude and phase naturally leads to the introduction of two local similarity functions. The first one $\mathcal{S}_\mathcal{A}$ compares only the amplitudes and is used for the MTM-procedures. Following equation (5.10) it is defined as follows:

$$\mathcal{S}_\mathcal{A}\left(\vec{h}(\vec{x}^M), \vec{h}(\vec{x}^I)\right) := \frac{\left\langle \mathcal{A}(\vec{h}(\vec{x}^M)) \,\middle|\, \mathcal{A}(\vec{h}(\vec{x}^I))\right\rangle}{\left\|\mathcal{A}(\vec{h}(\vec{x}^M))\right\| \cdot \left\|\mathcal{A}(\vec{h}(\vec{x}^I))\right\|} \,. \tag{5.11}$$

The second one $\mathcal{S}_\mathcal{P}$ compares the amplitudes and the phases and will be used to evaluate the actual similarities of mapped point pairs. $\mathcal{S}_\mathcal{P}$ will be defined in equation (5.36) in section 5.4, when it is clear how the phases are matched.

5.3 Creation of Mappings

The matching procedure described so far is concerned with finding a correspondence for a single point or a template of points. The collection of all correspondences found between a given pair of image and model are combined into a *model-image-mapping* or, simply, *mapping*. Formally, a mapping is a set of quadruples of coordinates:

$$\mathcal{M}(M, I) := \left\{ (\vec{x}_i^M, \vec{x}_i^I) \,\middle|\, i = 1, \ldots, N \right\} \tag{5.12}$$

It will be required that the respective coordinates are part of the respective representations, i.e. for each i there must be a unit in M whose first two (out of six) entries are identical to $\vec{x}_i^M$, and the same must be true when M is replaced by I. For further evaluation we now define some *geometrical characteristics* of a mapping $\mathcal{M}$: The number of correspondences will be called the *size* of the mapping:

$$|\mathcal{M}| := N \,, \tag{5.13}$$

where N is the number from equation (5.12). If $|\mathcal{M}| = 0$ the mapping is called *empty*. This is, obviously, not a very interesting case, but it can, in principle, result from our mapping procedures and must therefore be included in the definitions. Because the spatial coordinates in the model are fairly arbitrary (a model may have been derived from an arbitrary location in an image) the *average displacement* of the mapping describing the center of the coordinate system is important:

$$|\mathcal{M}| > 0 \implies \vec{A}(\mathcal{M}) := \frac{1}{|\mathcal{M}|} \sum_{i=0}^{|\mathcal{M}|} \left(\vec{x}_i^I - \vec{x}_i^M\right) \,, \tag{5.14}$$

$$|\mathcal{M}| = 0 \implies \vec{A}(\mathcal{M}) := \vec{0} \,. \tag{5.15}$$

If a mapping represents a simple shift between model and image its average displacement will be the shift vector. The deviation from a shift, which we will call the *distortion* of the mapping is given by the standard deviations in the two directions. The standard deviations are taken componentwise which makes the following definition a bit awkward to write down. The index m takes the values 1 and 2 and indicates the directions in image or model domain, respectively:

$$\vec{D}(\mathcal{M}) := \begin{pmatrix} D_1 \\ D_2 \end{pmatrix} \tag{5.16}$$

$$|\mathcal{M}| > 1 \implies D_m(\mathcal{M}) := \frac{1}{|\mathcal{M}| - 1} \sqrt{\sum_{i=0}^{|\mathcal{M}|} \left(x^I_{mi} - x^M_{mi} - A_m(\mathcal{M})\right)^2} \tag{5.17}$$

$$|\mathcal{M}| \leq 1 \implies \vec{D}(\mathcal{M}) := \vec{0} \tag{5.18}$$

This distortion is a two-dimensional vector with nonnegative components and zero if and only if the mapping is a simple shift or empty.

Including the local similarity functions $\mathcal{S}$ from equations (5.11) or (5.36), respectively we define the *global similarity* $\mathcal{S}_{glob}(M, I, \mathcal{M})$ of a mapping, together with its standard deviation $\mathcal{D}(M, I, \mathcal{M})$. For empty mappings both of these measures will be defined as zero, as well as $\mathcal{D}(M, I, \mathcal{M})$ for mappings with only one entry.

$$\mathcal{S}_{glob}(M, I, \mathcal{M}) := \frac{1}{|\mathcal{M}|} \sum_{i=0}^{|\mathcal{M}|} \mathcal{S}\left(\vec{h}\left(x^I_i\right), \vec{h}\left(\vec{x}^M_i\right)\right) . \tag{5.19}$$

$$\mathcal{D}(M, I, \mathcal{M}) := \frac{1}{|\mathcal{M}| - 1} \sqrt{\sum_{i=0}^{|\mathcal{M}|} \left(\mathcal{S}\left(\vec{h}\left(x^I_i\right), \vec{h}\left(\vec{x}^M_i\right)\right) - \mathcal{S}(M, I, \mathcal{M})\right)^2} . \tag{5.20}$$

In order to distinguish between the similarity functions $\mathcal{S}_{\mathcal{A}}$ and $\mathcal{S}_{\mathcal{P}}$ these measures will attain indices $\mathcal{A}$ or $\mathcal{P}$, respectively.

5.3.1 Matching Amplitudes on the Lowest Level

The first part of the mapping procedure consists in finding the part of the image where the object is located. For this it is sufficient to restrict the model and image representations to the lowest frequency level. If the image and the model differ in the frequency levels contained, which may happen due to the background suppression, the lowest level common to both representations has to be used here. Nevertheless, for simplicity of notation we will assume that $\mathcal{K}_0$ is this lowest level.

The procedure simply consists of choosing the amplitudes at the lowest level of the image representation as the data for multidimensional template matching and the amplitudes at the lowest level of the model representation as the template. The result is a shift vector $\vec{y}_0$, which is added to every model point to yield a first estimate of the mapping.

$$\mathcal{M}_0(M, I) := \{ (\vec{x}, \vec{x} + \vec{y}_0) \mid \vec{x} \in \mathcal{K}_0(M) \} \tag{5.21}$$

Although the reconstruction from this lowest level does not yield a recognizable picture of the model (see figure 3.5) the information contained here suffices to find the location. Of course, it is possible to artificially construct counterexamples that show an ambiguity on the lowest level but not on the higher ones, but this was generally not the case in the pictures we used. For a general discussion of such limitations of our algorithm see section 7.2.2. In the limited range of natural images this turns into an advantage, because in the lowest level the image information is spread out so far spatially that the local

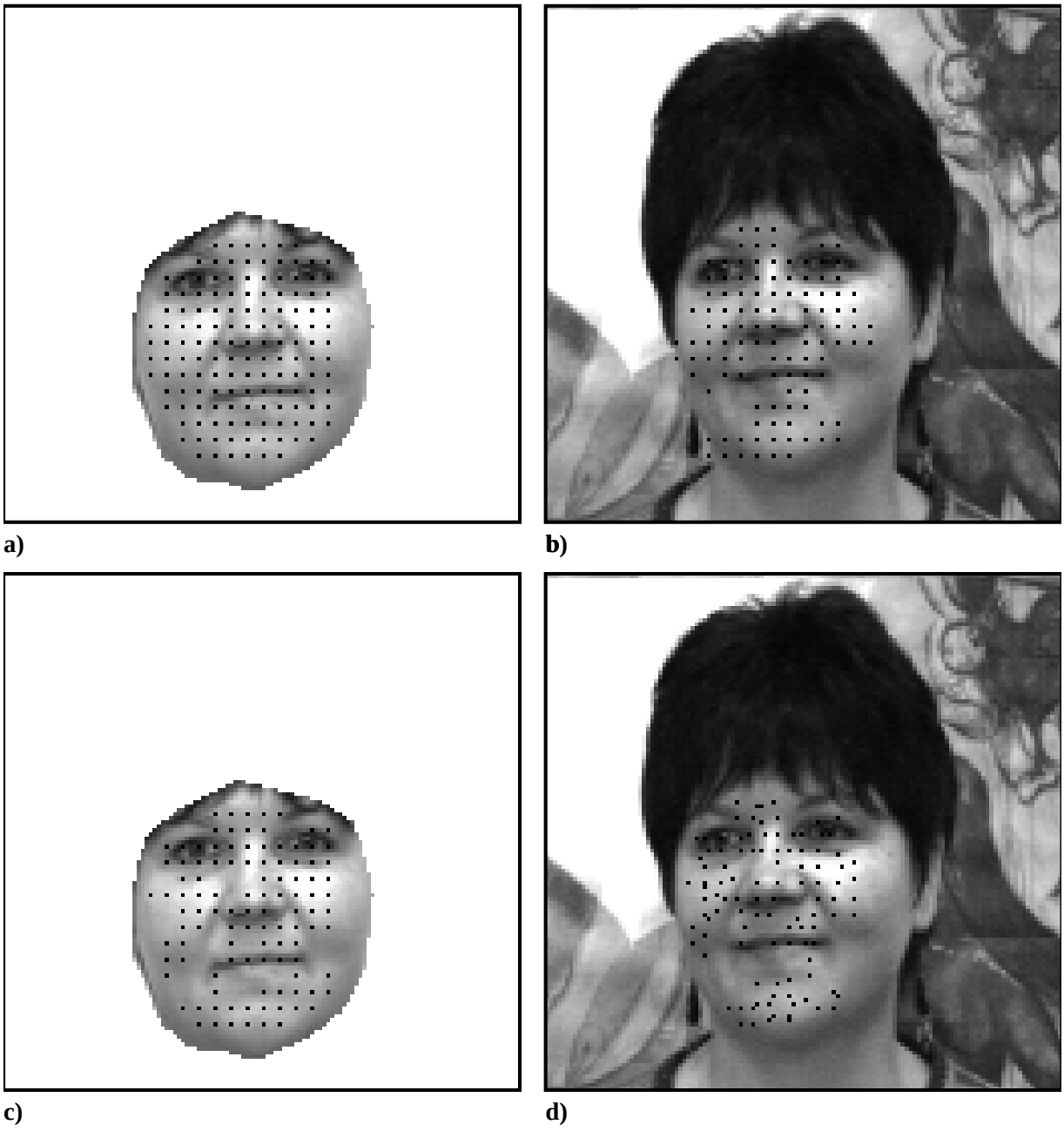

Figure 5.2: Mappings on the middle level. $\left(|\vec{k}| = 0.775\right)$ **a)** and **b)** show the mapping points from the MTM, in model and image. In **c)** and **d)** the phases have been matched and the points with poor similarity have been discarded from the mapping. The mapping from **c)** to **d)** is used for further refinement.

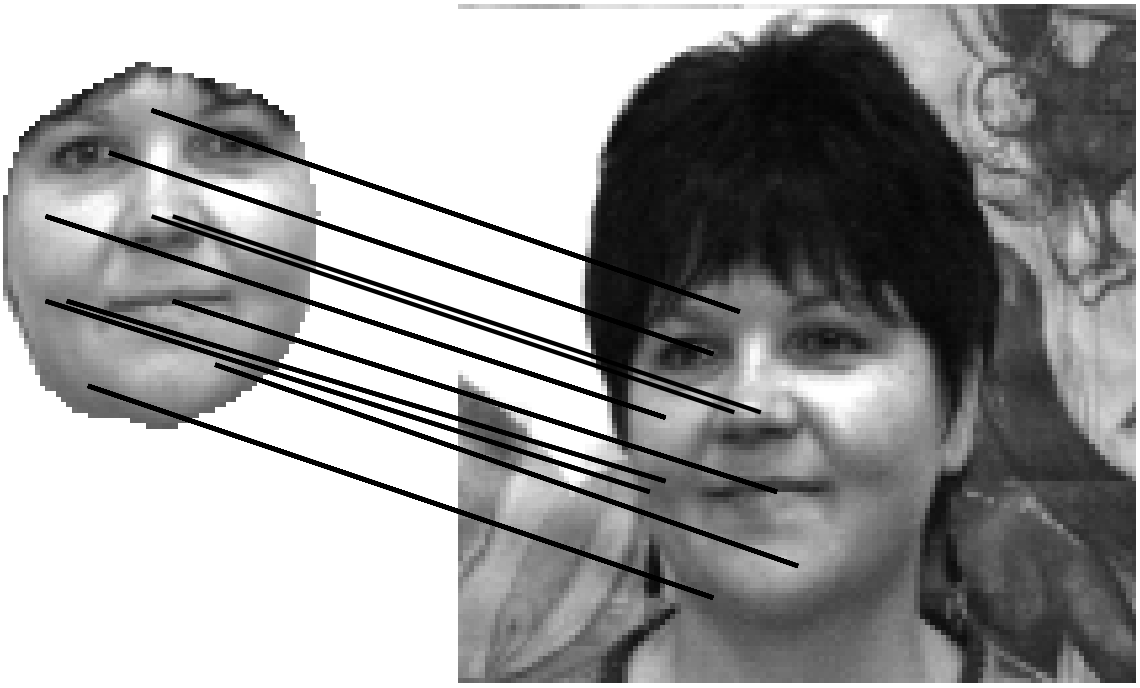

Figure 5.3: Correspondences on the middle level. $\left(|\vec{k}| = 0.775\right)$ This figure shows selected correspondences from the mapping in figure 5.2 in order to illustrate the accuracy of the mapping.

---

distortions between model and image (which will be measured in the refinement steps) do not impede the template matching.

The background suppression will in general produce model representations whose spatial extent is smaller than the one of the image. If this is not the case the MTM cannot produce reasonable results, and the shift vector $\vec{y}_0$ is defined as $\vec{0}$.

5.3.2 Mapping Refinement

In this section we present a method to refine a mapping $\mathcal{M}_n$ that has been produced using the frequency levels $\mathcal{K}_0 \ldots \mathcal{K}_n$ to a mapping $\mathcal{M}_{n+1}$ using the information from level $\mathcal{K}_{n+1}$. Of course, this will only be a true refinement if the spatial resolution at level $\mathcal{K}_{n+1}$ is higher than the one at $\mathcal{K}_n$, which need not necessarily be the case because the resolution is determined automatically from the center frequency as described in section 2.6.3 and may be the same if the sampling of the center frequencies is dense. Nevertheless, the method will also work in this case.

This is achieved by *local* multidimensional template matching of the response amplitudes in the levels $\mathcal{K}_{n+1}(M)$ and $\mathcal{K}_{n+1}(I)$. We must now specify how the model and image templates are chosen.

Both levels are arranged into a rectangular matrix of feature vectors. The size of the matrix is determined by the resolution on this level. If the resolutions happen to be different for the several directions of center frequencies, the largest resolution is chosen. As

we have already discussed while describing the recognition procedure in section 3.6.1 this does not pose any problems for representations that have been constructed as described in section 3.2. If further manipulations have been applied, again some rounding of the locations may be necessary.

The matrix of the model level is then divided up into small rectangles in a non-overlapping way. The *size* $\vec{s}^M$ of the rectangles depends on the level resolution. For all following operations it will be chosen such that they contain, in general, 2×2 feature vectors. On the model borders or at possible holes in the representation (which may occur as a result of amplitude thresholding) some of the rectangles may contain no feature vectors or 1, 2 or 3 of them. In the first case, they are not considered further. In the other cases, they are filled up with feature values $\vec{0}$.

Each little square (or more generally rectangle) now serves as a template for a local MTM. The data field is chosen in the following way. First, the point pair from the mapping $\mathcal{M}_n$ is chosen whose model point lies closest to the center of the template. If this point happens to be the center of the template, the corresponding image point will be the center of the data field. In this case, the data field also attains a fixed size $\vec{s}^I$, which must be larger than the template size $\vec{s}_M$ (in each component). We have chosen to introduce a fixed ratio s_M^I for the sizes of template and data, so in this case we get $\vec{s}^I = s_M^I \cdot \vec{s}^M$. That means that the data will (in general, like above) contain 3×3 feature vectors.

The local density of the mapping is not known, and many model locations will frequently be missing from a mapping. Thus, extra considerations are necessary for the case that the center of the template is not part of the mapping. At such points the mapping is not known with good quality, which has two consequences. First, some heuristic must be applied in order to determine the center of the data field. Second, the data field must be larger to account for the uncertainty in the correspondence.

Let $\vec{c}^M$ be the center of the model template, s_1^M and s_2^M its sizes in both directions, $\vec{x}^M$ the model point closest to $\vec{c}^M$ which is part of the mapping $\mathcal{M}^n$, and $\vec{x}^I$ the corresponding image point. Then center $\vec{c}^I$ and sizes s_1^I and s_2^I of the data area are defined as follows:

$$\vec{c}^I := \vec{x}^I + (\vec{c}^M - \vec{x}^M) \tag{5.22}$$

$$s_i^I := s_M^I \cdot s_i^M + 2 \cdot \left| c_i^M - x_i^M \right| , \; i = 1, 2 \tag{5.23}$$

The above case that the template center hits a mapping point is, of course, included in this definition.

Equation (5.22) reflects the idea that if the correspondence is not known at a model location the best one can do is assume a constant deviation from the closest known mapping location. Higher levels of sophistication might, of course, be applied here, e.g. using a combination of several surrounding mapping points. In the important case of the model boundaries that would not mean much progress, because the mapping will only be known in a direction perpendicular to the model boundary and pointing to the interior of the model.

In equation (5.23) the uncertainty in the mapping is accounted for by allowing a data field of at least twice the size of the deviation of the template center and the known mapping point in each component. This suffices to justify the simple assumption leading to equation (5.22), because the actual correspondence is checked over a wide range.

Now that the rectangle constituting the data field is defined the actual data field consists of all the feature vectors that fall inside this rectangle. If this is not the case for any one, no correspondence is assigned to the points in the template. Otherwise, the existing feature vectors are arranged into a rectangular matrix, missing locations are assigned zero amplitude and phase.

Then MTM is applied to the pair of template and data, yielding a local shift vector $\vec{y}_0$. This shift is relative to the mapping already known. So for each $\vec{x}$ which is part of the level $\mathcal{K}_{n+1}(M)$ *and* inside the current template the pairs

$$\left(\vec{x}, (\vec{c}^{\,I} + \vec{y}_0) + (\vec{x} - \vec{c}^{\,M})\right) \tag{5.24}$$

are included in the mapping $\mathcal{M}_{n+1}$. The first addition in the image components applies the shift to the center of the data field, the second one corrects for the location of $\vec{x}$ inside the template.

This procedure is executed for all the templates that make up the level $\mathcal{K}_{n+1}(M)$. It is worth noting that the single MTMs are completely independent of each other and can therefore be executed in parallel. This is due to the fact that the templates have been chosen to be non overlapping. The data fields, however, may overlap. This can lead to mappings that are unique but not invertible, i.e. several model points may be mapped to the same image point. Furthermore, the mapping need not be strictly neighborhood preserving, i.e. local "crossovers" of correspondences may occur. Both problems will be greatly alleviated by the removal of poor matches in section 5.5

5.4 Treatment of Phases

The matching of wavelet amplitudes alone can already lead to very good recognition performance (see also section 7.1). However, if a good spatial resolution of the mapping is required, the phases have to be included, because they carry the fine geometrical information. In this section we will describe a way to match the phases after the amplitudes have found their best correspondences. This part of the matching process is the only one that does not have a counterpart in the self-organizing process from chapter 4.

5.4.1 The Structure of the Wavelet Phases

For a close analysis of the matching process it is necessary to have some idea of the global properties of the wavelet responses. This is problematic because of the unformalized notion of a "natural image" (see section 7.2.2). Nevertheless, inspection of the responses has unraveled some qualitative properties, which have also been studied and verified in detail in (Fleet, 1992).

Figure 5.4 d) gives examples of the response amplitudes for various values of $\vec{k}$. It can be seen that the amplitudes take the form of smooth hills with maxima at edges perpendicular to $\vec{k}$. The spatial extent of these hills is mainly dictated by the width of the Gaussian window in the definition of the Gabor functions, i.e. by $\left|\vec{k}\right|^{-1}$. This form makes it plausible to use MTM for establishing correspondences, because small shifts will usually produce small variations in the amplitudes.

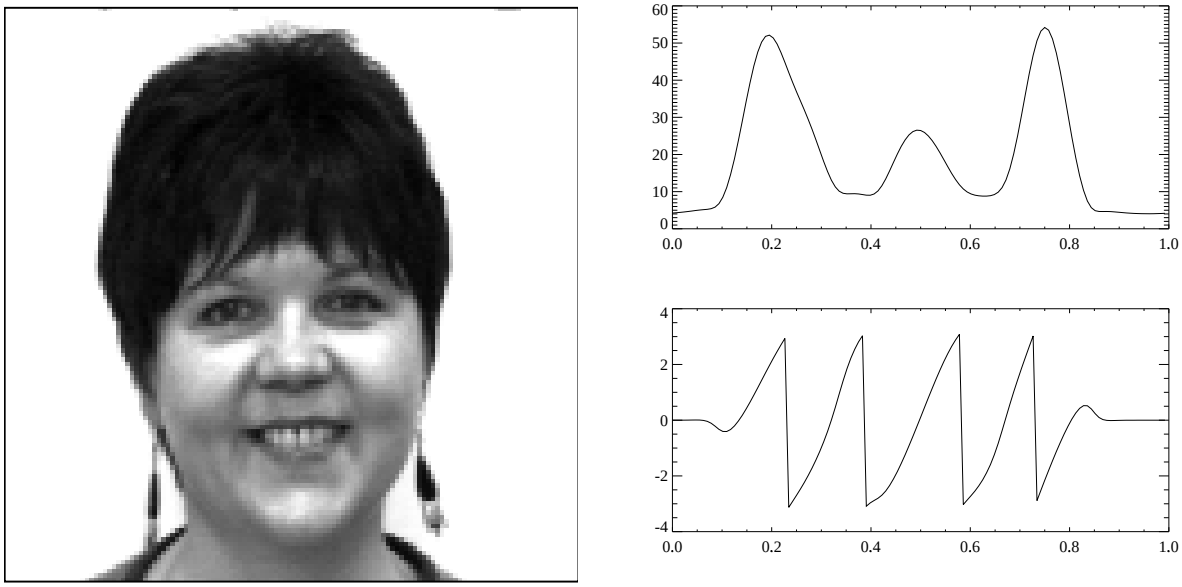

Figure 5.4: The structure of the wavelet responses. The image on the left hand side has been convolved with a kernel of center frequency 0.4 in horizontal direction. The plots show the amplitudes and the phases of the result on a scanline through the eyes. The amplitudes have the form of smooth hills, the phases behave like the phases of a plain wave with roughly the same frequency. The largest deviations are in the neighborhood of low amplitude values.

For the phases the situation is quite different. As shown in figure 5.4 d) at most locations they resemble the phases of a plane wave. Fleet (1992) has shown that this is indeed the case except for locations where the amplitude is close to zero. There, the phases change arbitrarily with just a little shift in the image plane. As a consequence, they are numerically extremely unstable and not usable for matching near points of small amplitude.

The frequency of the (local) waves is given by the gradient in the frequency components $\vec{k}$ of the wavelet space. The computation of this gradient is very costly, because fine resolution in $\vec{k}$ is required. Fleet also shows that it is close to the center frequency of the kernel.

5.4.2 Phase Matching

From the preceding section it is clear that the matching procedure for the phases must be quite different from the one for the amplitudes.

The phases, in the absence of zeros, have the structure of plain waves, i.e. they rotate with a frequency which is close to the center frequency of the generating kernel. For the matching task we assume that the phase frequency is equal to the center frequency, except for points with small amplitude. Fleet (1992) reports that it has proven sufficient to exclude points with amplitudes smaller than 5% of the maximal amplitude to get

reliable phases.

As the phases are only known on a coarse grid, this hypothesis can hardly be tested from the pyramid representation. However, the deviations have been tested on a variety of images and, after amplitude thresholding, have not been drastic (see figure 5.4).

For matching units more reliably we assume that the phase difference between two units found to be corresponding by amplitude matching are caused by small local shifts on the scale of the discretization, which can, of course, not be detected by the MTM.

The phase difference for the matching is defined as follows, already taking into account the instability of the phases around amplitude zeros (and their resulting uselessness for matching):

$$\Delta\left(\vec{u}^I,\vec{u}^M\right)=\begin{cases}\mathcal{P}(\vec{u}^I)-\mathcal{P}(\vec{u}^M) & : \quad \mathcal{A}(\vec{u}^I)>t_a\max\mathcal{A}(\vec{u}^I)\wedge\mathcal{A}(\vec{u}^M)>t_a\max\mathcal{A}(\vec{u}^M)\\ 0 & : \quad \text{otherwise}\end{cases}. \tag{5.25}$$

Once these assumptions have been made it is clear how the phases of two *units* have to be matched. If the phase difference is caused only by a displacement, then it must be equal to the product of the displacement vector and the center frequency. Yet, our task is to match a whole feature vector. Each of its n_{dir} units votes for one displacement and they have to reach an agreement. This is done by choosing the displacement $\vec{X}$ which gives the least squared deviation from the given phase differences.

$$E=\sum_{j=0}^{n_{dir}-1}\left[\vec{X}\cdot\vec{k}_j-\Delta\left(\vec{u}^I,\vec{u}^M\right)\right]^2\stackrel{!}{=}\min \tag{5.26}$$

If the directions of the center frequencies are given by $j\pi/n_{dir},\ j=0,\ldots,n_{dir}-1$ (as usual) and their length is constant $\left(|\vec{k}_j|=k\right)$ the minimization can be solved. The single phase differences $\Delta\left(\vec{u}_j^I,\vec{u}_j^M\right)$ as defined by (5.25) will be abbreviated as Δ_j. In order to keep the terms more compact we will replace n_{dir} by D for the time of the derivation:

$$\begin{aligned}
E &= \sum_{j=0}^{D-1}\left[\vec{X}\cdot\vec{k}-\Delta_j\right]^2 && (5.27)\\
&= \sum_{j=0}^{D-1}\left[kX_1\cos\left(\frac{j\pi}{D}\right)+kX_2\sin\left(\frac{j\pi}{D}\right)-\Delta_j\right]^2 && (5.28)\\
\frac{\partial E}{\partial X_1} &= 2k^2\sum_{j=0}^{D-1}\left[X_1\cos\left(\frac{j\pi}{D}\right)+X_2\sin\left(\frac{j\pi}{D}\right)-\frac{\Delta_j}{k}\right]\cos\left(\frac{j\pi}{D}\right) && (5.29)\\
\frac{\partial E}{\partial X_2} &= 2k^2\sum_{j=0}^{D-1}\left[X_1\cos\left(\frac{j\pi}{D}\right)+X_2\sin\left(\frac{j\pi}{D}\right)-\frac{\Delta_j}{k}\right]\sin\left(\frac{j\pi}{D}\right) && (5.30)
\end{aligned}$$

So we have to solve the linear system:

$$\begin{pmatrix}\sum_{j=0}^{D-1}\cos^2\left(\frac{j\pi}{D}\right) & \sum_{j=0}^{D-1}\cos\left(\frac{j\pi}{D}\right)\sin\left(\frac{j\pi}{D}\right)\\ \sum_{j=0}^{D-1}\cos\left(\frac{j\pi}{D}\right)\sin\left(\frac{j\pi}{D}\right) & \sum_{j=0}^{D-1}\sin^2\left(\frac{j\pi}{D}\right)\end{pmatrix}\begin{pmatrix}X_1\\X_2\end{pmatrix}=\begin{pmatrix}\sum_{j=0}^{D-1}\frac{\Delta_j}{k}\cos\left(\frac{j\pi}{D}\right)\\ \sum_{j=0}^{D-1}\frac{\Delta_j}{k}\sin\left(\frac{j\pi}{D}\right)\end{pmatrix} \tag{5.31}$$

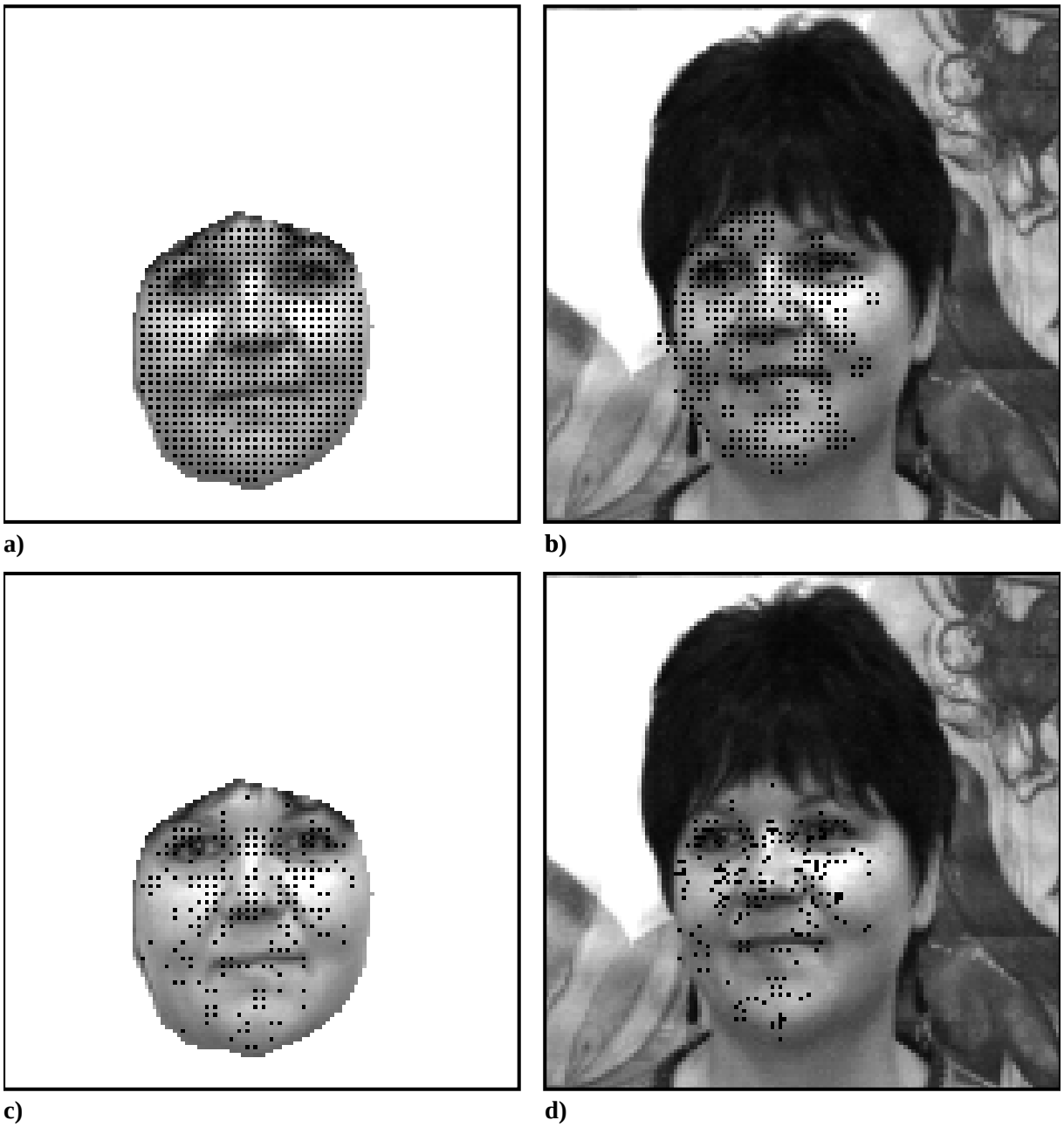

Figure 5.5: Mappings on the highest level. $\left(|\vec{k}| = 1.5\right)$ **a)** and **b)** show the mapping points from the MTM, in model and image. In **c)** and **d)** the phases have been matched and the points with poor similarity have been discarded from the mapping. The mapping from **c)** to **d)** is used for further refinement.

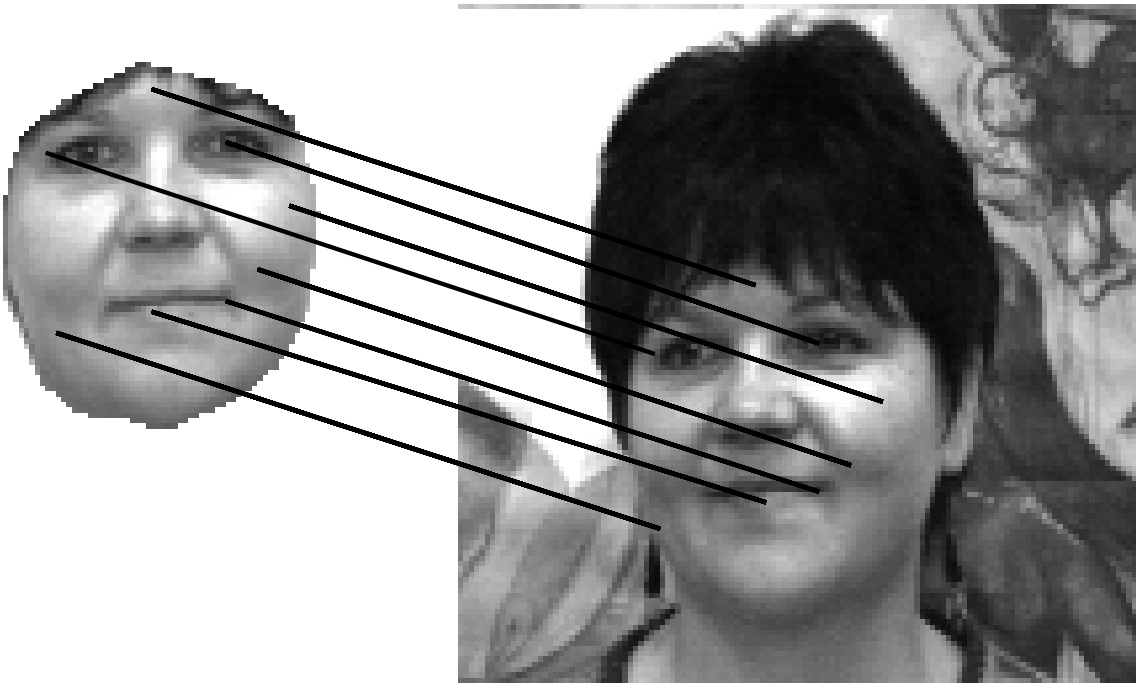

Figure 5.6: Correspondences on the highest level. $\left(|\vec{k}| = 1.5\right)$ This figure shows selected correspondences from the mapping in figure 5.5 in order to illustrate the accuracy of the mapping.

Elementary trigonometric operations for the matrix yield:

$$\begin{pmatrix} \sum_{j=0}^{D-1} \cos^2\left(\frac{j\pi}{D}\right) & \sum_{j=0}^{D-1} \cos\left(\frac{j\pi}{D}\right)\sin\left(\frac{j\pi}{D}\right) \\ \sum_{j=0}^{D-1} \cos\left(\frac{j\pi}{D}\right)\sin\left(\frac{j\pi}{D}\right) & \sum_{j=0}^{D-1} \sin^2\left(\frac{j\pi}{D}\right) \end{pmatrix} = \begin{pmatrix} \frac{D}{2} & 0 \\ 0 & \frac{D}{2} \end{pmatrix} , \tag{5.32}$$

and we get the following solution to (5.31):

$$X_1 = \frac{2}{D}\sum_{j=0}^{D-1} \frac{\Delta_j}{k} \cos\left(\frac{j\pi}{D}\right) \tag{5.33}$$

$$X_2 = \frac{2}{D}\sum_{j=0}^{D-1} \frac{\Delta_j}{k} \sin\left(\frac{j\pi}{D}\right) \tag{5.34}$$

$$\vec{X} = \frac{2}{D}\sum_{j=0}^{D-1} \frac{\Delta_j}{k}\frac{\vec{k}_j}{k} \tag{5.35}$$

The single constituents of the sum in (5.35) are the shifts predicted by the single phase differences multiplied with the unit vectors in the directions of the center frequencies. The prefactor $1/n_{dir}$ is just the average of these shifts. The extra factor 2 reflects the dimension of the image space. If $n_{dir} = 2$, the displacements are independent of each other and may add freely.

For a reasonable local similarity function after the phase correction the remaining phase difference must become part of $\mathcal{S}_\mathcal{P}$. We have chosen to do this in the following way:

$$\mathcal{S}_\mathcal{P}\left(\vec{h}(\vec{x}^M), \vec{h}(\vec{x}^I)\right) := \frac{\sum_i \mathcal{A}(\vec{h}(\vec{x}_i^M))\mathcal{A}(\vec{h}(\vec{x}_i^I)) \cos\left(\Delta\vec{h}(\vec{x}_i^M, \vec{x}_i^I) - \vec{k}\cdot\vec{X}\right)}{\left\|\mathcal{A}(\vec{h}(\vec{x}^M))\right\| \cdot \left\|\mathcal{A}(\vec{h}(\vec{x}^I))\right\|} \tag{5.36}$$

This is identical to $\mathcal{S}_\mathcal{A}$ in the ideal case that the phase differences after the applied local shifts are zero. Remaining phase differences lead to a penalty, because the cosine becomes smaller than one.

5.5 Exclusion of Erroneous Matches

The mapping procedures described so far still have one serious drawback: They enforce that every point in the model (on the various levels) must find a correspondence in the image. If some model point is occluded in the image this is not useful because it will find some point with structure which is similar by chance. The requirement posed in the definition of the correspondence problem, namely that the non-existence of a corresponding point must also be detected by the procedure, is still violated.

The mapping procedures described in sections 5.3.1 and 5.3.2 have assured that point pairs with good similarity are excluded from the mapping if their geometrical arrangement is incompatible with the mapping in neighboring parts. If the mapping still contains noncorresponding points this can be due to occlusion or strong distortion. In the case of occlusion a model point will find a point in the occluding object. In the presence of strong distortion the local features between the model point and the corresponding image point will differ significantly, and therefore this part of the mapping is no longer reliable.

The only grounds on which these two cases can be detected is the actual local similarity between features of the corresponding points. In order to exclude mismatches we introduce a *quality threshold* t_{qn} and exclude all points from the mapping $\mathcal{M}$ for which the condition

$$\mathcal{S}_{loc}\left(\vec{h}(\vec{x}^M), \vec{h}(x^I)\right) \geq t_{qn} \tag{5.37}$$

is violated. The threshold t_{qn} is defined as follows:

$$t_{q0} := \alpha_q \mathcal{S}\left(M, I, \mathcal{M}_0\right) \tag{5.38}$$

$$t_{q(n+1)} := \alpha_q \mathcal{S}\left(M, I, \mathcal{M}_n\right) . \tag{5.39}$$

$\mathcal{S}\left(\mathcal{M}_n\right)$ is the global similarity from equation (5.19), $\mathcal{S}_{loc}$ is the local similarity function from equation (5.11) or (5.36), respectively. For all levels except the lowest one this threshold is created using only information from the mappings already known. This has the advantage that, again, no global information about the mapping currently being established is necessary.

The choice of the factor α_q is dictated by the following tradeoff. If it is small then most points will be kept in the mapping and the mapping will remain dense. This will lead to many erroneous correspondences for the reasons mentioned above. If the value is too high few points with high similarities will survive and the mapping can become very sparse. Then the refinement steps will eventually become reliable. Experience has shown that $\alpha_q = 1$ constitutes a reasonable compromise between these two problems.

5.6 Overall Mapping Procedure

We now have a method to initialize a mapping, a method to refine any mapping using the information from a higher frequency level and two methods for manipulating a given mapping, namely phase matching and dropping correspondences with poor similarity.

These are combined in the following way. The initialization yields a mapping $\mathcal{M}_0^{\mathcal{A}}$. Then the phases are matched leading to $\mathcal{M}_0^{\mathcal{P}}$. The quality threshold t_{q0} is derived from $\mathcal{M}_0^{\mathcal{P}}$ Then the poor matches are removed from $\mathcal{M}_0^{\mathcal{P}}$, which leads to $\mathcal{M}_0^{\mathcal{F}}$, the final mapping on this level.

The mapping $\mathcal{M}_0^{\mathcal{F}}$ together with the levels $\mathcal{K}_1(M)$ and $\mathcal{K}_1(I)$ is then used for the refinement step, which results in $\mathcal{M}_1^{\mathcal{A}}$. Phase matching yields $\mathcal{M}_1^{\mathcal{P}}$ and with t_{q1} as derived from $\mathcal{M}_0^{\mathcal{F}}$ this is reduced to $\mathcal{M}_1^{\mathcal{F}}$. The same step is executed one more time, resulting in the mappings $\mathcal{M}_2^{\mathcal{A}}$, $\mathcal{M}_2^{\mathcal{P}}$, and $\mathcal{M}_2^{\mathcal{F}}$.

There have also been experiments that created a third mapping on each level which, for every model point, selected the better match of $\mathcal{M}_i^{\mathcal{P}}$ and $\mathcal{M}_i^{\mathcal{A}}$. This has not led to significant improvements.

The four mapping procedures mentioned at the beginning of this section could, of course, be combined in many more ways. Some of them have been tested and generally led to worse results for the mapping quality as well as for the recognition rate.

5.7 Quality of Mappings

The system described above has been tested on many model-image pairs. All results in this and the following chapter have been obtained with the sparse sampling scheme. The representation parameters were the ones specified in section 3.4.5. The parameters specific to the matching were chosen as:

$$\alpha_q = 1.0\,, \tag{5.40}$$

$$s_M^I = 1.5\,. \tag{5.41}$$

Many experiments have shown that the mapping procedure actually finds the correct correspondences between model and image points. Some examples can be found in figures 5.3 and 5.6. Complete mappings are shown in figures 5.1, 5.2, and 5.5. Close inspection of figures 5.2 and 5.5 reveals that not all correspondences are as excellent as the ones visualized, but in general nearly all of them are acceptable. Unlike for the recognition presented in the next chapter the author was unable to find an objective test for the quality of the mappings. Entering a complete correspondence mapping by hand was beyond his patience and would not have been objective, either. So the inspection of the figures must suffice to convince the reader.

It is important to observe that some model points that did not have a good correspondence on the lowest level may attain one on the higher levels. This indicates that a multilevel procedure is very adequate for the task.

Another important observation that can be made is the importance of the phase matching. In figure 5.1 it can be seen that many of the points found acceptable correspondences after phase matching already on this level although most of the correspondences produced by the MTM were fairly poor.

Figure 5.6 illustrates that very good correspondences are found from the lowest up to the third level. It may be concluded that the combination of low-level coarse matching, phase matching, mapping refinement and dropping poor correspondences constitutes a powerful method for solving the correspondence problem.

5.8 Extension to Size Invariant Mappings

The mapping methods described so far require that the image and the model be roughly identical in size and orientation, because otherwise the units must be matched in a different way. As pointed out in section 3.6.3, a rotation or scaling in the image plane leads to a relocation of the respective units and to a rotation or inverse scaling of the length of center frequency.

The most powerful way to deal with such invariances is to estimate the parameters for scale and orientation and then transform the image, accordingly. These estimates must come from presegmentation or from the initialization step.

The size invariant matching procedures causes several new problems like the different samplings in image and frequency space in model and image that we will not discuss in detail here. In principle, they can be accounted for by a sufficiently high resolution in the image transform, from which the model can choose the subsets required for the matching. This has not been attempted here due to the high amount of computing resources required.

Nevertheless, in order to demonstrate this capability we shall describe a modified initialization that is able to produce acceptable mappings from the model to an image which contains the image at 50% of its linear size. This is, of course, a well-behaved case, because the spatial resolutions generally advance by a factor of two when moving up one level.

The idea is simply the following. Instead of initializing the mapping only from $\mathcal{K}_0(M)$ to $\mathcal{K}_0(I)$ mappings $\mathcal{M}_{0i}$ are initialized from $\mathcal{K}_0(M)$ to *all* levels $\mathcal{K}_i(I)$. Because of the favorable ratio of resolutions this can be done with a practically unmodified algorithm, because the matrices which represent the template and the data already have the correct arrangement of the responses.

The best mapping from $\mathcal{M}_{0i}$ can then be refined to a mapping $\mathcal{M}_{1(i+1)}$ using the standard refinement method. That this method works may appear trivial at the first glance, but it must be kept in mind that the factor of 50% is not at all exact and the images are real images with much more background. An example of a resulting mapping is shown in figure 5.7. All experiments conducted have shown that $\mathcal{S}(\mathcal{M}_{01})$ is significantly larger than $\mathcal{S}(\mathcal{M}_{00})$ and $\mathcal{S}(\mathcal{M}_{02})$.

This is, of course, only a very crude demonstration that basically the same method may work for objects seen at different distances as well.

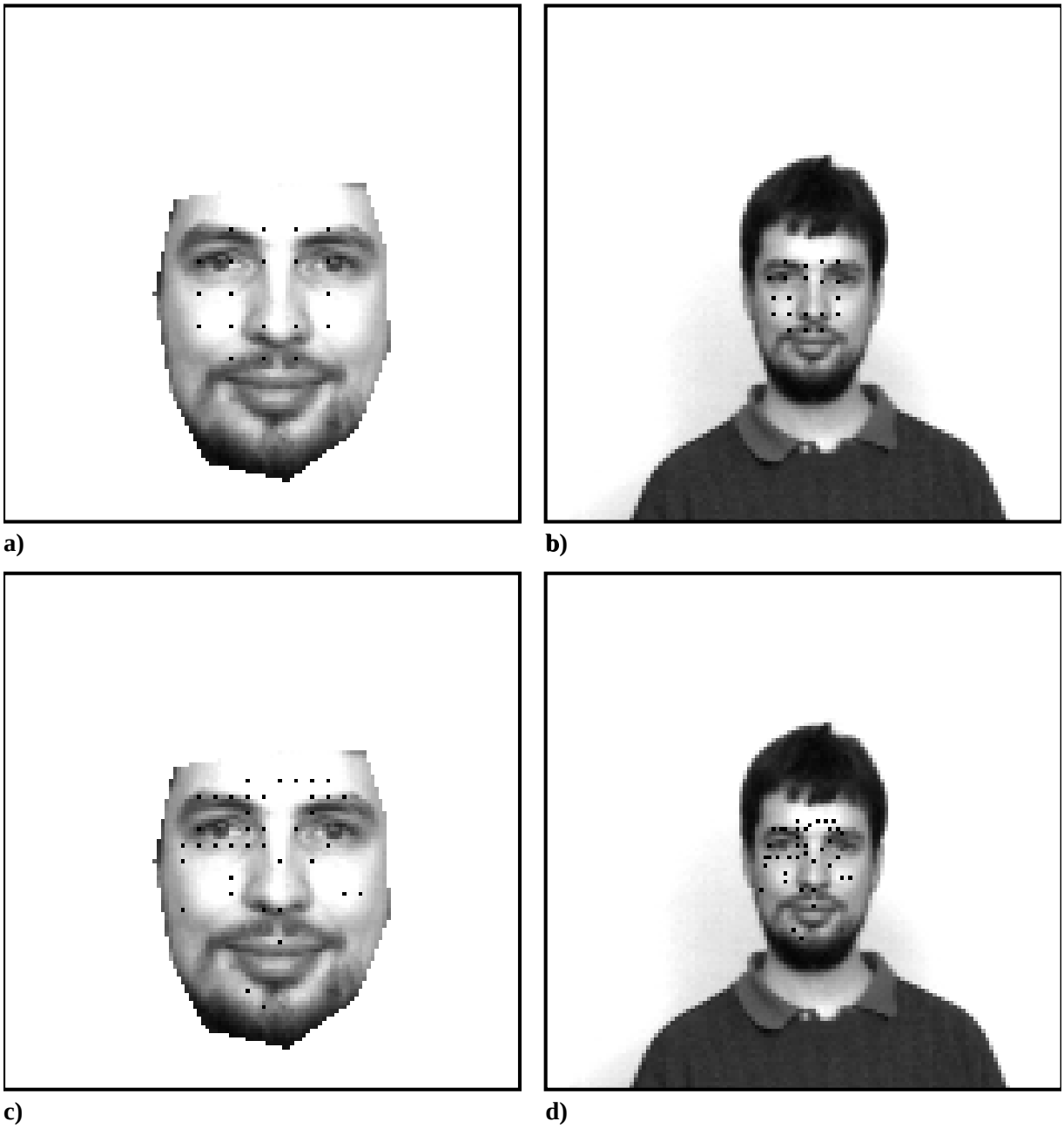

Figure 5.7: Examples of size invariant mappings. **a)** and **b)** show the mapping points from the lowest level in the model to the first level in the image. That image level has been selected by the quality of the mappings only. In **c)** and **d)** that mapping has been refined as usual. In this procedure the relative quality threshold α_q has been set to 0.9.

6. Hierarchical Object Recognition

For a few seconds he sat in stunned silence as the images rushed around his mind and tried to find somewhere to settle down and make sense.
Part of his brain told him that he knew perfectly well what he was looking at and what the shapes represented whilst another quite sensibly refused to countenance the idea and abdicated responsibility for any further thinking in that direction.
The flash came again, and this time there could be no doubt.

Douglas Adams, The Hitchhiker's Guide to the Galaxy

We have now presented two different ways of solving the correspondence problem. The first one gave a detailed neuronal formulation and the second one was a simplified version designed to run in reasonable time on a workstation. In order to fulfill the final claim that has been made in the introduction we must now show that an established correspondence mapping can be used for object recognition independent of the background. For these recognition experiments we will use only mappings created by the procedure from chapter 5, because the simulation of the dynamics currently requires too much computation to use them on databases of realistic size.

6.1 Recognition procedure

6.1.1 Image-Model Similarity

Once a correspondence mapping has been established it can be used to define a global similarity between the model and the image. Applying the mapping procedure to an image and a whole gallery of models then leads to an object recognition procedure. There is, of course, a lot of freedom in the definition of such a measure. The simplest possibility would be the average similarity from equation (5.19). Other possibilities one could imagine would include detailed evaluation of the local distortions which could be used for a weighting of the local similarities. This reflects the notion that in the presence of stronger distortion the similarities are expected to be worse. For future developments beyond the scope of this work even the single pairs of corresponding points should be weighted according to their importance for recognition. As it seems very hard to invent such a weighting the

probably best way to do this is to learn such weights from experience. This research is currently being started at our institute.

Here we restrict ourselves to a linear combination of the average similarity and (with negative weight) the length of the distortion vector. This reflects the fact that if the distortion of a mapping is high the probability for remaining erroneous correspondences is also high. Therefore, we introduce as the *model-image similarity* for the recognition decision:

$$\mathcal{S}_{rec}(M, I, \mathcal{M}) = \mathcal{S}_{glob}\left(\mathcal{M}(M, I)\right) - \lambda \left|\vec{D}(\mathcal{M}(M, I))\right| . \tag{6.1}$$

The factor λ has been determined experimentally. A value of 1.0 yielded the best results in all experiments. This shows that the pure similarities are not enough to discriminate well between the models. However, in the mapping procedure itself only those similarities have been used in order to keep the required information as local as possible.

Evaluating this image-model similarity for a database of models $\{ M_i \mid i = 0 \ldots N\}$ yields a series of similarities, whose maximum corresponds to the recognized model. The detailed results will be presented in section 6.2.

6.1.2 Significance of Recognition

The process of comparing an image with all models stored in a database always yields a best value for the global similarity of one model to the given image, irrespective of whether or not a corresponding image of the same person is contained in the database. For a recognition mechanism to be of use, a criterion to evaluate the significance of a match must be applied. This is an important difference to many other recognition schemes that can not decide well that an unknown object has been presented.

Our results show that the answer can, with some reliability, be extracted from the statistics of the series of all global similarities. Let the series $\mathcal{S}_i$ denote these values ordered in ascending sequence, i.e. $\mathcal{S}_i < \mathcal{S}_{i+1}\ \forall i \in \{0, 1, \ldots, N-1\}$, and M_i be the model which gave the result $\mathcal{S}_i$. For the recognition to be significant we expect $\mathcal{S}_0$, which corresponds to M_0, the "candidate" model, to be clearly distinct from all the other values. This has been formalized as follows: If s the standard deviation of the series $\{\mathcal{S}_i \mid i = 1, 2, ...N-1\}$ (not containing the candidate model), then we define a first criterion for the acceptance of a match:

$$\kappa_1 := [r_1 > t_{\kappa 1}] \ , \text{ where } r_1 := \frac{\mathcal{S}_1 - \mathcal{S}_0}{s} . \tag{6.2}$$

This means that the difference of the best similarity and the second best must exceed a threshold (in units of the standard deviation of the whole series).

This criterion has been developed for the recognition system described in (Lades et al., 1993). In the current context it has turned out to be crucial to include the absolute value of the similarity as a second criterion, whereas the second criterion from that system turned out to be quite useless here.

$$\kappa_2 := [r_2 > t_{\kappa 2}] \ , \text{ where } r_2 := \mathcal{S}_0 . \tag{6.3}$$

Both criteria can be combined in order to keep more significant recognitions while ruling out all incorrect ones thus improving the performance of the system further:

$$\kappa_3 := [\kappa_1 \vee \kappa_2] \ . \tag{6.4}$$

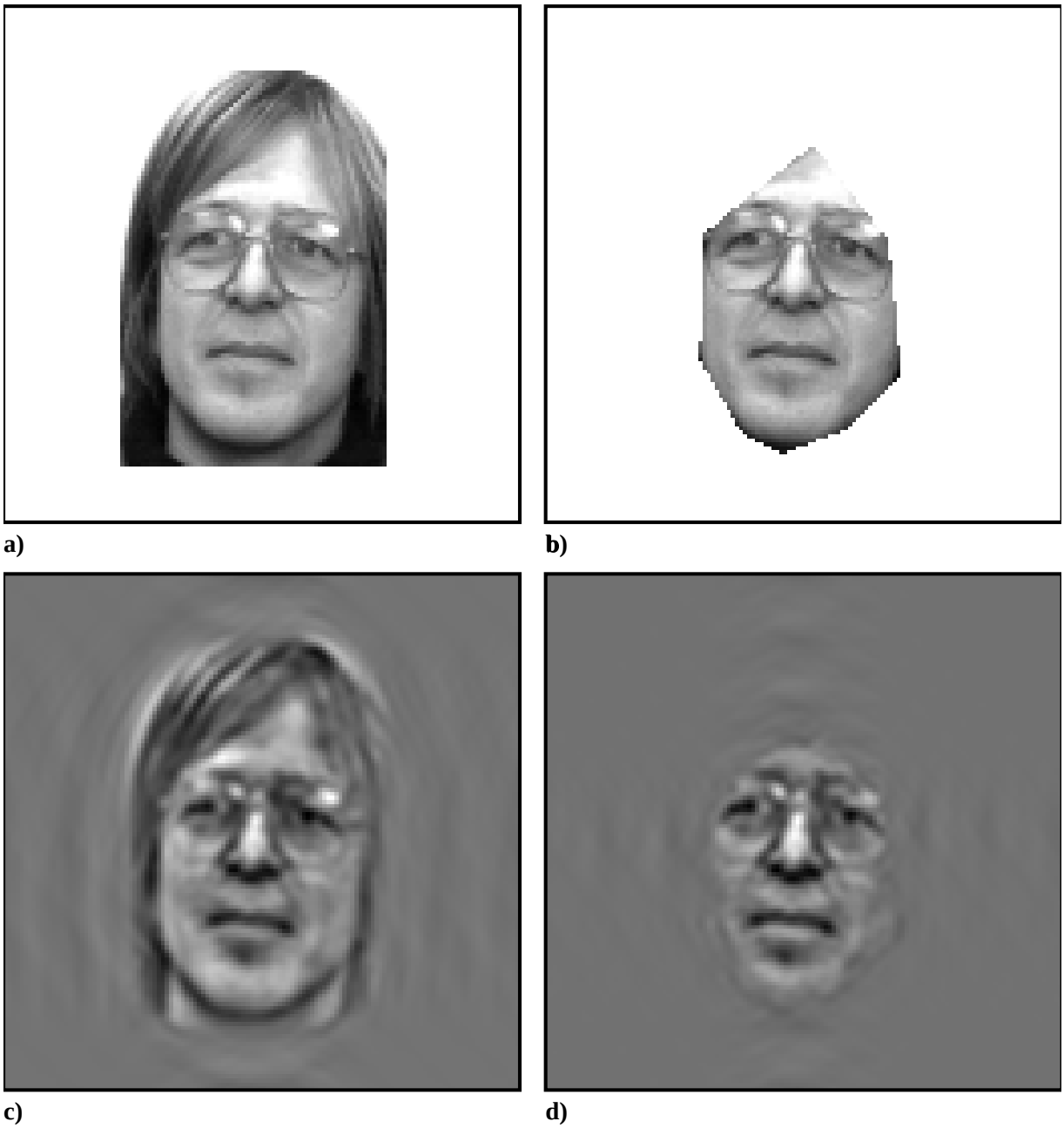

Figure 6.1: Segmentation for model databases. a) shows a segmented model from database **M1**, **b)** from database **M2**. In **c)** and **d)** the corresponding reconstructions are presented. If a much higher value of σ had been chosen for the transform (which would mean a wider spatial extent of the Gabor kernels) these reconstructions would be hardly recognizable. The representation for this model in **M1** consists of 8,388 units, the one in **M2** of 3,576.

Experiment 1.1: Rectangular segmentation, 83 images								
Level	**Criterion**	**C**	**F**	**CA**	**CR**	**FR**	**FA**	**R**
Hierarchy level 0	$\kappa_1 = [r_1 > 0.472]$	71	29	51	20	29	0	49
	$\kappa_2 = [r_2 > 0.964]$	71	29	20	51	29	0	80
	$\kappa_3 = \kappa_1 \vee \kappa_2$	71	29	54	17	29	0	46
Hierarchy level 1	$\kappa_1 = [r_1 > 0.768]$	43	6	33	11	6	0	17
	$\kappa_2 = [r_2 > 0.874]$	73	6	48	25	6	0	31
	$\kappa_3 = \kappa_1 \vee \kappa_2$	40	6	29	11	6	0	17
Hierarchy level 2	$\kappa_1 = [r_1 > 0.483]$	13	4	11	2	4	0	6
	$\kappa_2 = [r_2 > 0.816]$	25	6	14	11	6	0	17
	$\kappa_3 = \kappa_1 \vee \kappa_2$	13	4	12	1	4	0	5
Hierarchy total	κ_1	99	1	94	0	1	0	6
	κ_2	99	1	83	5	1	0	17
	κ_3	99	1	95	0	1	0	5
Level 2 only	$\kappa_1 = [r_1 > 0.483]$	94	6	87	7	6	0	13
	$\kappa_2 = [r_2 > 0.816]$	94	6	83	11	6	0	17
	$\kappa_3 = \kappa_1 \vee \kappa_2$	94	6	89	5	6	0	11

Table 6.1: Recognition results for rectangularly segmented models and images of persons looking 15° to their left. The numbers are percentages of the whole image database. Significance criteria and cases are explained in section 6.1.2.

Of course, there is a trade-off between ruling out all false recognitions and accepting all correct ones. Here we will use a simpler evaluation method than in (Lades et al., 1993), because the experience has shown that including the extra cases does not make much difference in the result. The reasons for that are that the databases used are fairly big and the recognition rates are not too high in the most interesting cases, where structured background is present. Thus we will only work with databases which contain the correct person. Then the following cases are possible:

CA The correct model was picked as the best match, and the match was judged *significant*.

CR The correct model was picked as the best match, but the match was rejected.

FA The wrong model was picked as the best match, and the match was accepted (significant).

FR The wrong model was picked as the best match, and the match was rejected (insignificant).

Experiment 1.2: Rectangular segmentation, 249 images								
Level	**Criterion**	**C**	**F**	**CA**	**CR**	**FR**	**FA**	**R**
Hierarchy level 0	$\kappa_1 = [r_1 > 0.764]$	68	32	20	48	32	0	80
	$\kappa_2 = [r_2 > 0.964]$	68	32	13	55	32	0	87
	$\kappa_3 = \kappa_1 \vee \kappa_2$	68	32	23	45	32	0	77
Hierarchy level 1	$\kappa_1 = [r_1 > 0.768]$	65	15	45	20	15	0	35
	$\kappa_2 = [r_2 > 0.876]$	71	15	38	34	15	0	49
	$\kappa_3 = \kappa_1 \vee \kappa_2$	62	15	45	17	15	0	33
Hierarchy level 2	$\kappa_1 = [r_1 > 0.970]$	22	13	6	17	13	0	30
	$\kappa_2 = [r_2 > 0.816]$	36	13	16	20	13	0	33
	$\kappa_3 = \kappa_1 \vee \kappa_2$	20	12	8	12	12	0	24
Hierarchy total	κ_1	93	7	70	7	7	0	30
	κ_2	93	7	67	11	7	0	33
	κ_3	93	7	76	5	7	0	24
Level 2 only	$\kappa_1 = [r_1 > 0.970]$	86	14	63	23	14	0	37
	$\kappa_2 = [r_2 > 0.821]$	86	14	62	24	14	0	38
	$\kappa_3 = \kappa_1 \vee \kappa_2$	86	14	69	17	14	0	31

Table 6.2: Recognition results for rectangularly segmented models and images of persons in different poses. The numbers are percentages of the whole image database. Significance criteria and cases are explained in section 6.1.2.

Additionally, we shall consider the simple cases **C** (correct recognition) and **F** (false recognition) and the "rejected" or "no decision yet" case **R**, which is the union of **CR** and **FR**.

In the ideal recognition algorithm, only the case **CA** should occur. Any safe recognition algorithm must rule out case **FA**, because this is a serious mistake. Case **CR** shows an imperfection of the presented image or the algorithm. The quality of the recognition can thus be judged by counting the number of **CA** cases once the thresholds have been adjusted such that no **FA** cases remain. This choice of thresholds may sound debatable but it produces the best results that can be expected from the algorithm given that false positive recognitions are unacceptable. A living system probably would not live up to this expectation. So the idea about the thresholds is that they have been learned in a long process of false and correct recognitions using feedback from the environment and adjusted in a way that suits the respective organism best — anxious individuals may well have higher thresholds than rather careless ones.

Experiment 2.1: Segmentation without hair, 83 images								
Level	**Criterion**	**C**	**F**	**CA**	**CR**	**FR**	**FA**	**R**
Hierarchy level 0	$\kappa_1 = [r_1 > 0.553]$	42	58	10	33	58	0	90
	$\kappa_2 = [r_2 > 0.970]$	42	58	13	29	58	0	87
	$\kappa_3 = \kappa_1 \vee \kappa_2$	42	58	17	25	58	0	83
Hierarchy level 1	$\kappa_1 = [r_1 > 0.398]$	81	10	61	19	10	0	29
	$\kappa_2 = [r_2 > 0.904]$	76	11	23	53	11	0	64
	$\kappa_3 = \kappa_1 \vee \kappa_2$	73	10	54	19	10	0	29
Hierarchy level 2	$\kappa_1 = [r_1 > 0.330]$	24	5	18	6	5	0	11
	$\kappa_2 = [r_2 > 0.799]$	58	6	55	2	6	0	8
	$\kappa_3 = \kappa_1 \vee \kappa_2$	24	5	23	1	5	0	6
Hierarchy total	κ_1	99	1	89	1	1	0	11
	κ_2	99	1	92	1	1	0	8
	κ_3	99	1	94	0	1	0	6
Level 2 only	$\kappa_1 = [r_1 > 0.330]$	94	6	88	6	6	0	12
	$\kappa_2 = [r_2 > 0.799]$	94	6	92	2	6	0	8
	$\kappa_3 = \kappa_1 \vee \kappa_2$	94	6	93	1	6	0	7

Table 6.3: Recognition results for models without their hair and images of persons looking 15° to their left. The numbers are percentages of the whole image database. Significance criteria and cases are explained in section 6.1.2.

6.1.3 Hierarchical Recognition

With the notion that insignificant recognition reflects the fact that no reliable decision was possible yet the multilevel structure of our algorithm can be used to improve average recognition time and recognition quality. First, a recognition is attempted using only the mapping $\mathcal{M}_0^{\mathcal{F}}$ on the lowest level. For all the **R** cases the next mapping $\mathcal{M}_1^{\mathcal{F}}$ is used and for the cases where this level did not lead to a decision, the recognition is again attempted using the mapping $\mathcal{M}_2^{\mathcal{F}}$.

Tables 6.1 through 6.5 will show two things. First, correct recognition is indeed possible from the lowest level on. That means that the average recognition times can be greatly reduced by the hierarchical approach. Secondly, hierarchical recognition always yields significantly more significant recognitions than the recognition from $\mathcal{M}_2^{\mathcal{F}}$ alone. This gives interesting insights into the distribution of prominent recognition cues across the spatial frequency range.

These practical advantages underpin the usefulness of the philosophy of hierarchical recognition outlined in section 4.4. It is also capable of interpreting the recognition experiment shown there. On the frequency levels present in the low pass filtered image a

Experiment 2.2: Segmentation without hair, 249 images								
Level	**Criterion**	**C**	**F**	**CA**	**CR**	**FR**	**FA**	**R**
Hierarchy level 0	$\kappa_1 = [r_1 > 0.563]$	41	59	8	33	59	0	92
	$\kappa_2 = [r_2 > 0.972]$	41	59	7	34	59	0	93
	$\kappa_3 = \kappa_1 \vee \kappa_2$	41	59	11	30	59	0	89
Hierarchy level 1	$\kappa_1 = [r_1 > 1.273]$	62	30	15	47	30	0	77
	$\kappa_2 = [r_2 > 0.904]$	63	31	10	53	31	0	83
	$\kappa_3 = \kappa_1 \vee \kappa_2$	59	30	18	41	30	0	71
Hierarchy level 2	$\kappa_1 = [r_1 > 1.252]$	56	21	14	42	21	0	63
	$\kappa_2 = [r_2 > 0.833]$	61	22	24	37	22	0	59
	$\kappa_3 = \kappa_1 \vee \kappa_2$	50	21	23	27	21	0	48
Hierarchy total	κ_1	85	15	37	12	15	0	63
	κ_2	85	15	41	14	15	0	59
	κ_3	85	15	52	6	15	0	48
Level 2 only	$\kappa_1 = [r_1 > 1.252]$	78	22	33	46	22	0	67
	$\kappa_2 = [r_2 > 0.833]$	78	22	41	37	22	0	59
	$\kappa_3 = \kappa_1 \vee \kappa_2$	78	22	49	30	22	0	51

Table 6.4: Recognition results for models without their hair and images of persons in different poses. The numbers are percentages of the whole image database. Significance criteria and cases are explained in section 6.1.2.

correct recognition is possible but it is not significant enough for the visual system to be satisfied with it. If no higher frequency information is available, this is the final result of the recognition attempt. In the presence of faulty high frequency information recognition is again tried on the next higher level, where it completely fails.

6.2 Tests of the Recognition Performance

In order to evaluate the recognition performance two different model databases and two different image databases have been set up. In this section we shall present all the results of the recognition experiments.

Model database **M1** consists of all 83 persons looking straight into the camera (database 1 from section 3.4.4. Their images have been segmented by a simple rectangle which has the same size for all models. This has been done to attempt a relatively fair comparison with the system described in 7.1 and in (Lades et al., 1993). For an example see figure 6.1 **a)**.

Model database **M2** consists of the same 83 persons looking straight into the camera. Their images have been segmented by hand such that their hair is invisible and only the

Experiment	Images		Hierarchy total				Level 2 only			
	Database	#	C	F	CA	R	C	F	CA	R
1.1	I1	83	99	1	95	5	94	6	89	11
1.2	I1, I2, I3	249	93	7	76	24	86	14	69	31
1.3	I2	83	80	20	58	42	70	30	37	63
1.4	I1, I2	166	89	11	68	32	82	18	62	38
1.5	I3	83	100	0	96	4	95	5	93	7
1.6	I1, I3	166	99	1	95	5	95	5	89	11
1.7	I2, I3	166	90	10	75	25	83	17	61	39
2.1	I1	83	99	1	94	6	94	6	93	7
2.2	I1, I2, I3	249	85	15	52	48	78	22	49	51
2.3	I2	83	71	29	40	60	65	35	36	64
2.4	I1, I2	166	85	15	61	39	80	20	59	41
2.5	I3	83	86	14	57	43	76	24	54	46
2.6	I1, I3	166	92	8	68	32	85	15	65	35
2.7	I2, I3	166	78	22	42	58	70	30	39	61

Table 6.5: Overview of all recognition results. The numbers are percentages of the whole image database. Experiments in the upper half use models with rectangular segmentation, in the lower half the models are segmented without their hair. Database **I1** contains persons looking 15° to their left, **I1** the same persons looking 30° to their left and **I3** contains them showing an arbitrary facial expression. Cases are explained in section 6.1.2.

---

plain faces remain (See figure 6.1 **b)**). Generally, this demonstrates the capabilities for recognition independent of the background. In this special case it evaluates the possibility to recognize persons independent of their hair, which is a much harder task also for humans.

Image database **I1** is used to test the performance under moderate conditions and consists of the 83 persons looking 15° to their right. Database **I123** introduces hard conditions, namely including image database 1, 2 and 3. Thus it contains 249 images including the head orientations of 15°, 30° and the arbitrary facial expressions.

In order to allow a closer analysis of the strengths and weaknesses of the algorithm, the results for the databases I2 and I3 alone are also shown.

For each experiment the thresholds t_{r1} and t_{r2} have been adjusted such that false positive recognitions (**FA** cases) are reliably excluded. For the factor λ, which weights the distortion in the definition of the recognition similarity, a value of 1 has been found to be optimal.

In the hierarchical case there is an extra row for the total of each column over the

levels. The numbers in these columns have the following interpretations. The **C** cases are the ones where the correct model was recognized at some level during the process. The **CA** cases were accepted at one (and therefore only one) level. The **F**, **FA**, and **FR** cases are the ones where false recognition (accepted or rejected) happened on *all* levels. Also the number of **R** cases shows the ones that have been rejected on all levels (and is therefore identical to the number of **R** cases in the highest hierarchy level).

Image data base **I1** does not pose serious problems for recognition with either model segmentation. For the other databases the number of correct and significant recognitions drops sharply from model database **M1** to **M2**. This shows that the feature vectors inside the face are distorted strongly, and the recognition must rely more on the hairstyle. This does not sound like a serious restriction. However, the attempted invariance under changes in hairstyle was not a goal in itself. It has mainly been demonstrated for a method to reliably exclude background influence. So other methods will probably face the same problem when being confronted with objects in front of an arbitrary background.

7. Discussion

"All right," said the Cat; and this time it vanished quite slowly, beginning with the end of the tail, and ending with the grin, which remained some time after the rest of it had gone.
"Well, I've often seen a cat without a grin," thought Alice; "but a grin without a cat! It's the most curious thing I ever saw in my life!"

Lewis Carroll, Alice's Adventures in Wonderland

7.1 Comparison With Labeled Graph Matching

This section describes a recognition system to the development of which the author contributed before engaging on the topic of this thesis. Enough detail is covered to see similarities and differences to the system of chapter 6. For a full description including parameters and detailed performance figures see (Lades et al., 1993).

In this system, database models are represented by *labeled graphs*. Recognition is performed by finding an optimal match for the model graphs in the image domain. The model graphs have a number of points in the model domain as vertex set $\mathbf{V}$. The graph topology is defined by the edge set $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$.

Vertices and edges are both labeled by vectors defined below. Labels are compared with similarity functions which are added up to yield a similarity function between the graphs.

7.1.1 Vertex labels

The vertices in our graph are simply points in the two-dimensional image domain. Each point is labeled with the array of the amplitudes of all wavelet responses centered at that point. This choice requires that the points in $\mathbf{V}$ belonging to the graph are sampling points for the wavelet transform at all frequencies considered. In other words, the sampling sets in the model domain must be the Cartesian product of $\mathbf{V}$ and a space-independent frequency sampling set:

$$\mathbf{S}^M = \mathbf{V} \times \mathbf{S}_f \,. \tag{7.1}$$

$\mathbf{S}_f$ is the same as in equation (3.9). with different parameters: $n_{dir} = 8$, $n_{lev} = 5$, $k_{min} = 2.356$, $k_{min} = 0.589$. This difference is due to a different value of σ in the wavelets. The choice of σ will be discussed in section 7.1.7.

Experiment	Images		Hierarchy total				FACEREC			
	Database	#	C	F	CA	R	C	F	CA	R
1.1	I1	83	99	1	95	5	95	5	93	7
1.2	I1, I2, I3	249	93	7	76	24	92	8	81	19
1.3	I2	83	80	20	58	42	86	14	61	39
1.4	I1, I2	166	89	11	68	32	90	10	77	23
1.5	I3	83	100	0	96	4	94	6	94	6
1.6	I1, I3	166	99	1	95	5	95	5	92	8
1.7	I2, I3	166	90	10	75	25	90	10	77	23
2.1	I1	83	99	1	94	6	19	81	1	99
2.2	I1, I2, I3	249	85	15	52	48	14	86	1	99
2.3	I2	83	71	29	40	60	16	84	4	96
2.4	I1, I2	166	85	15	61	39	17	83	2	98
2.5	I3	83	86	14	57	43	7	93	0	100
2.6	I1, I3	166	92	8	68	32	13	87	1	99
2.7	I2, I3	166	78	22	42	58	11	89	1	99

Table 7.1: Results of the hierarchical and the FACEREC system. Databases and experiment numbers are the same as in table 6.5. FACEREC performs slightly better on model database **M1** (rectangular segmentation) but fails completely on database **M2** (hair removed). This shows that only the hierarchical system is capable of background-independent recognition.

The vector of all amplitudes at one single image point is called a *jet*:

$$\vec{J}(\vec{x}) = \mathcal{A}(\vec{x}) \Big|_{\vec{k}\in\mathbf{S}_f} \tag{7.2}$$

In the image domain full sampling is used in order to have a jet attached to every image location (pixel).

Similarity of vertex labels is defined as the normed scalar product between two jets exactly as our similarity of local feature vectors in equation (5.3):

$$\mathcal{S}_v\left(\vec{J}_1, \vec{J}_2\right) = \frac{\vec{J}_1 \cdot \vec{J}_2}{\left\|\vec{J}_1\right\| \cdot \left\|\vec{J}_2\right\|} \tag{7.3}$$

7.1.2 Edge Labels

Edge labels are introduced to enforce a weak geometrical similarity in the graphs to be matched. This is justified by the notion that, although there may be distortions between

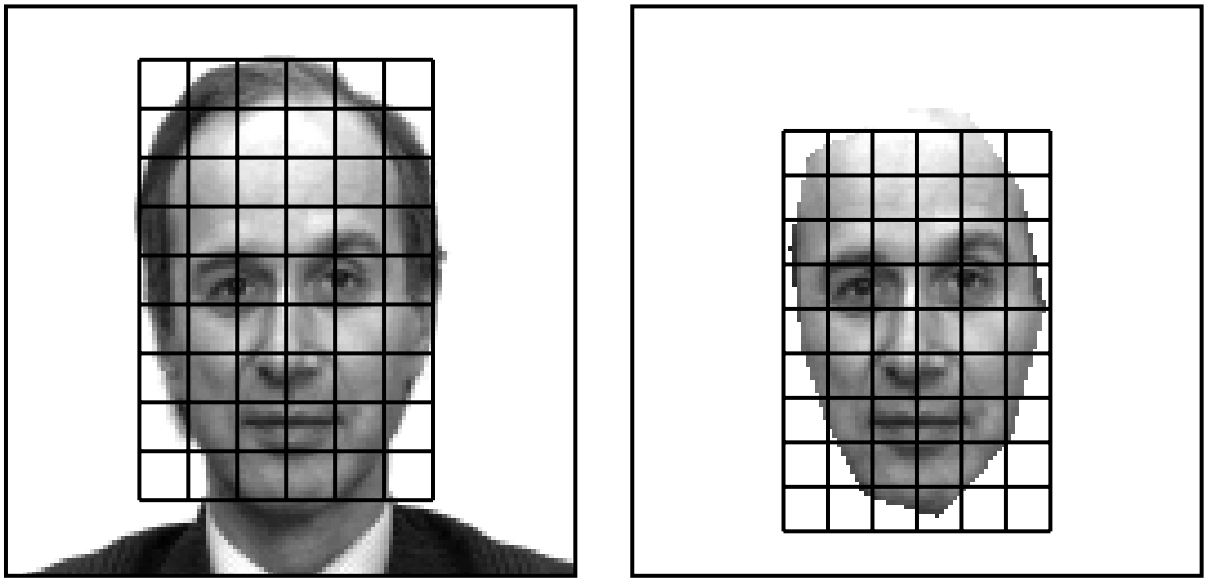

Figure 7.1: Model graphs for the FACEREC system. On the left a model graph for the standard system is shown (database **M1**). For a comparison with the hierarchical system graphs have been produced from the segmented images put on white background (database **M2**).

---

the image graph and the model graph, the graphs are not expected to be arbitrarily scrambled. In the system described here the edges $(i, j) \in E$ with the distance vector of the vertices they connect:

$$\vec{\Delta}_{(i,j)} := \vec{x_j} - \vec{x_i} \tag{7.4}$$

Those labels are compared by the similarity function:

$$\mathcal{S}_e\left(\vec{\Delta}^I, \vec{\Delta}^M\right) := -\left(\vec{\Delta}^I - \vec{\Delta}^M\right)^2 . \tag{7.5}$$

7.1.3 Graph Similarity

Finally, the comparison functions for both types of labels are added up over the whole graph and linearly combined into one similarity function between the stored model and the image graph.

$$\mathcal{S}_\Gamma(\Gamma^I, \Gamma^M) := \lambda \sum_{(i,j)\in E} \mathcal{S}_e\left(\vec{\Delta}^I_{(i,j)}, \vec{\Delta}^O_{(i,j)}\right) + \sum_{i\in V} \mathcal{S}_v\left(J^I_i, J^O_i\right) . \tag{7.6}$$

The factor λ can be adjusted to make the enforcement of the neighborhood preservation more or less strict. It is varied during the process.

7.1.4 Graph Dynamics

The function $\mathcal{S}_\Gamma$ defined in the previous section can be interpreted as a function on the set of all possible image graphs. Its maximum over the whole set measures the similarity of the

stored model to the presented image. It can be approximated by a suitable optimization procedure described in the next paragraph.

The graphs describing the models are rectangular grids with 7 points horizontally and 10 points vertically. The distance between two neighboring points (in either direction) is 10 pixels. (The image size is 128×128 pixels). First the given model graph is copied into the image domain, λ is set to a very high value which enforces that the only variability in the mapping is the center of gravity of the graph. This is optimized by a random walk procedure. After this step, the corresponding graph in the image domain is roughly located on the object.

In the second step λ is relaxed to a suitable finite value and the correspondences of single points are optimized by a random walk. After some convergence criterion has been fulfilled, the resulting value of $\mathcal{S}_\Gamma$ is taken as similarity of the image and the model.

As usual, this similarity value is calculated for all models, and the best one is the recognized object. Like in section 6.1.2 a combination of two significance criteria is used to judge the reliability of the recognition. The first one is the same as κ_1 from equation (6.2), the second one is defined as the difference of the best similarity to the *mean* of all other similarities, divided by their standard deviation.

7.1.5 Performance

Here we present the results of the FACEREC matching with the same databases as in section 6.2. Figure 7.1 shows model graphs from model databases **M1** and **M2**. Table 7.1 compares them with the ones of the hierarchical recognition scheme. The results show that for model database **M1** the graph matching performs slightly better, and significantly better under more difficult circumstances.

For database **M2** FACEREC produces only very poor results. Although the graphs have already been shrunk in order to account for the smaller regions the results may be slightly improved by model graphs which are better adapted. Nevertheless, it is obvious that this system can not deal with the background in a reasonable way.

7.1.6 Advantages of the hierarchical system

Although the FACEREC system has better recognition rates in the absence of structured background than the hierarchical system the latter has the following advantages.

FACEREC does not produce good correspondence mappings. One of the reasons for this is that it relies only on the amplitudes of the wavelet responses. Due to the large variety of spatial frequencies present in a jet it has proven very difficult to include the phase information in a convincing way. The second reason is that the graph similarity has a deformation term. This makes the procedure relatively robust in the presence of strongly deformed jets by placing the vertex where it optimally fits the structure of the model graph. These points are probably not corresponding well. Furthermore, the correspondences delivered are restricted to the vertices of the model graph and therefore very sparse.

The precise and dense mappings delivered by the hierarchical matching system will probably allow many important extensions. In the scheme described in chapter 6 all the

geometrical information in the mapping is collapsed into the single number $\left|\vec{D}\right|$ (equation (6.1)). More advanced versions will possibly be able to estimate the true three-dimensional geometrical transform from the correspondences. This in itself is an interesting piece of information, which human cognition is also able to extract. Furthermore, it can be used to modify the representations (or the similarity functions) taking into account that the geometrical transformation did not only change the position of the units but also their values. (See section 3.6.3 for simple examples.) This can be turned into an iterative algorithm that optimizes both locations of corresponding points and the similarities of their representing units and will certainly constitute a much more robust recognition procedure.

Another possible improvement would be to assign different weights to the different regions. For face recognition, e.g., the eyes and the mouth will be much more important than the cheeks. This weighting, however, must rely on good correspondences at least for the points with high weights.

The jets in the FACEREC system pick up very much background information, especially the low frequency components. This is shown impressively by the results in table 7.1. Dealing with this problem the way that has been proposed in this work (section 3.2) is not possible with the jet representation used in FACEREC, because most of the jets would have to be discarded.

For a massively parallel implementation the hierarchical system has two great advantages. The pyramidal representation is optimally suited for implementation on a convolver with a fixed maximal kernel size (such as, e.g., the Datacube system).

The topological costs in the FACEREC system are a *global* measure that has to be evaluated at every update step of the matching. This makes it hard to implement the matching process in a truly parallel manner. In the hierarchical scheme, all refinement steps, as well as all phase adjustments and the exclusion of the poor matches are completely independent of each other. That means that they can be carried out on separate processors, with a need for communication only after the establishment of a complete mapping on one frequency level.

7.1.7 Which Relative Bandwidth for Gabor Functions?

A difference between the hierarchical matching and the FACEREC system that deserves special consideration is the relative bandwidth of the Gabor kernels used. In FACEREC, the value has been $\sigma = 2\pi$ with a much higher localization in frequency space. Figure 2.3 shows both kernels.

In this work we have chosen $\sigma = 2$, a value which is close to the properties of simple cells. This leads to a relatively high localization in image space. For good matching this localization poses a problem because these small areas lead to more ambiguities in the feature similarity. (The probability that small areas are similar in the image is higher than for larger ones). In our system this problem has been alleviated by matching templates of such responses, which again enlarges the area.

A qualitative argument that can lead to a better understanding of the relationship of feature vectors produced with the different values of σ is the notion that the features corresponding to the higher values are linear combinations of the ones from the lower val-

ues. The reason is that for the Gabor kernels before admissibility correction the following relationship holds:

$$\psi_{2\sigma;\vec{k}} = \psi_{\sigma;\vec{k}} * \psi_{\sigma;\vec{k}} \tag{7.7}$$

The validity of this can be seen immediately from the Fourier transform (equation (2.54)). It shows that kernels with high values of σ can be reached by successive convolution with ones with lower σ and the same holds for the actual values of the image representation. If a higher value is needed, just another convolution with a suitable kernel is necessary.

7.2 Outlook

7.2.1 What has been achieved?

In contrast to earlier and other works the need for a hierarchical scheme has been motivated and two working systems have been proposed. The dynamic link matching has been extended to finding a coarse match within an image containing background and refining it.

Previously unsolved issues like the suppression of the background and the inclusion of phase information have been tackled successfully. The background suppression has led to a method that can recognize faces independently of the person's hairstyle, although with a lower rate of significant recognitions. The FACEREC system is not able to do this without substantial modifications.

Many attempts to solve the correspondence problem have been lacking the successes of the FACEREC system described in 7.1 or the matching scheme presented in this thesis. A major reason for our success seems to be the use of feature vectors instead of scalar features. In this light the use of orientation selective filters may well be an important step towards better systems and a better understanding of object recognition.

7.2.2 What is left to do?

For a first answer to this question the reader is invited to look around and enjoy the ease with which human cognition works. None of the severe limitations of our computer vision systems poses a problem here. So it looks as if everything is left to do. The gap between neural dynamics and convincing demonstrations of cognitive capabilities is still awesome.

Nevertheless it can be hoped that this work is at least a step to a closer understanding of cognition and describe it. In this case some successive steps can be outlined. The most important problem to solve seems to be a convincing integration of a memory of known objects into a system. Here we have treated all objects separately without any interconnections between them. Consequently, the recognition algorithms have a linear complexity in the number of models. This is certainly not the case for recognition in the brain. There are already several models of associative memory, but it is currently unclear how they can cooperate with the matching schemes described here.

A related challenge is the incorporation of more abstract categories. Most probably face recognition would first recognize that there is a face in the presented image and then "take a closer look" and decide about the identity of the person. It is currently unclear how the abstract notion of a "general face" can be represented. Furthermore, categories

like “female” or “grinning” can be recognized without any knowledge of the identity of the person. Finding a suitable representation for such categories seems hopeless, but our brains do it all the time. So somewhere hidden in our brains there must be a code for “a grin without a cat”.

A different part of the problem is the lack of a classification of “natural images”. It is difficult enough to invent a good algorithm for object recognition, but it is fairly easy to construct examples where a given algorithm completely fails. This means that computer vision algorithms (or the unknown algorithms carried out in the brains of living beings) are only valid for a very small subclass of light distributions. This subclass can, most probably, not be defined mathematically but depends on the history of all things this being has seen. So maybe the definition should be turned around, stating that a natural image is one where all the algorithms work well. On one hand this view makes the set of “natural images” depend on the individual. On the other hand it shifts the emphasis from the actual algorithm to the principles that *create* this algorithm.

Although the systems described here are only crude caricatures of living brains they are already very complicated. The neuronal machinery is a complex structure of layers of different resolution and size and links between them, not to mention the parameters that must be adjusted in order to keep it within a regime that exhibits useful behavior. Such sophisticated machinery is probably not present at birth but builds up and adapts during growth using all the visual experiences made. So the ultimate challenge seems to be a deeper understanding of the principles that govern the development of the neural machinery in real brains. Once this will be achieved it may become feasible to let computer vision algorithms develop from a crude to a truly useful state. The scientific question that may turn up then is if their internal functioning remains understandable.

8. Bibliography

Amari, S. (1980). Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339–364. Reprinted in (Anderson et al., 1990).

Amari, S. (1989). Dynamical stability of formation of cortical maps. In Arbib, M. and Amari, S., editors, *Dynamic Interactions in Neural Networks: Models and Data*. Springer.

Anderson, C. and van Essen, C. (1987). Shifter circuits: A computational strategy for dynamic aspects of visual processing. *Proc Natl. Acad. Sci. USA*, 84:6297–6301.

Anderson, J., Pellionisz, A., and Rosenfeld, E., editors (1990). *Neurocomputing II. Directions for Research*. MIT Press, Cambridge, MA.

Anderson, J. and Rosenfeld, E., editors (1988). *Neurocomputing. Foundations of research*. MIT Press, Cambridge, MA.

Antoine, J.-P. (1989). Poincaré coherent states and relativistic phase space analysis. In *Wavelets, Time-Frequency Methods and Phase Space*, pages 221–231. Springer, Berlin, Heidelberg, New York.

Arndt, P. A., Mallot, H. A., and Bülthoff, H. (1993). Stereovision without localized image features. Technical Report 1, Max-Planck-Institut für biologische Kybernetik, Spemannstraße 36, D-7206 Tübingen.

Atick, J. and Redlich, A. (1990). Mathematical model of the simple cells in the visual cortex. *Biological Cybernetics*, 63:99–109.

Bajcsy, R. and Kovačič, S. (1989). Muliresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21.

Battle, G. (1992). Wavelets: A renormalization group point of view. In et al., M. B. R., editor, *Wavelets and their applications*, pages 323–349. Jones and Bartlett Publishers, Boston.

Baylis, B. and Rolls, E. (1987). Responses of neurons in the inferior temporal cortex in short term and serial recognition memory tasks. *Experimental Brain Research*, 65:614–622.

Baylis, G., Rolls, E., and Leonhard, M. (1985). Selectivity between faces in the responses of a population of neurons in the cortex in the superior temporal sulcus in the monkey. *Vision Res.*, 25(8):1021–1035.

Behrmann, K.-O. (1993). Leistungsuntersuchungen des „Dynamischen Link-Matchings“ und Vergleich mit dem Kohonen-Algorithmus. Technical Report IR-INI 93-05, Ruhr-Universität Bochum, Diploma thesis, Universität Karlsruhe.

Bergen, J. R. and Julesz, B. (1983). Parallel versus serial processing in rapid pattern discrimination. *Nature*, 303(5919):696–698.

Bertrand, J. and Bertrand, P. (1989). A relativistic Wigner function affiliated with the Poincaré group. In *Wavelets, Time-Frequency Methods and Phase Space*, pages 232–246. Springer, Berlin, Heidelberg, New York.

Biederman, I., Subramaniam, S., and Madigan, S. F. (1994). Chance forced choice recognition memory for identifiable RSVP object pictures. Paper presented at the meetings of the Psychonomics Society, St. Louis.

Bienenstock, E. and von der Malsburg, C. (1987). A neural network for invariant pattern recognition. *Europhysics Letters*, 4:121–126.

Blahut, R. E. (1988). *Principles and Practice of Information Theory*. Addison Wesley.

Blakemore, C. and Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, 228:477–478.

Blakemore, C. and Mitchell, D. E. (1973a). Environmental modification of the visual cortex and the neural basis of learning and memory. *Nature*, 241:467–468.

Blakemore, C. and Mitchell, D. E. (1973b). Environmental modification of the visual cortex and the neural basis of learning and memory. *Nature*, 241:467–468.

Böge, S. (1980). Skript zur Vorlesung „Analysis III“. Mathematisches Institut der Universität Heidelberg.

Buhmann, J., Lades, M., and von der Malsburg, C. (1990). Size and distortion invariant object recognition by hierarchical graph matching. In *Proceedings of the IJCNN International Joint Conference on Neural Networks*, pages II 411–416, San Diego. IEEE.

Buhmann, J., Lange, J., and von der Malsburg, C. (1989). Distortion invariant object recognition by matching hierarchically labeled graphs. In *IJCNN International Joint Conference on Neural Networks, Washington*, pages I 155–159. IEEE.

Buhmann, J., Lange, J., von der Malsburg, C., Vorbrüggen, J. C., and Würtz, R. P. (1992). Object recognition with Gabor functions in the Dynamic Link Architecture — parallel implementation on a Transputer network. In Kosko, B., editor, *Neural Networks for Signal Processing*, pages 121–159. Prentice Hall, Englewood Cliffs, NJ.

Bülthoff, H. H. and Edelman, S. (1992). Psychophysical support for a two-dimensional view interpolation theory of object recognition. *Proc. Natl. Acad. Sci. USA*, 89:60–64.

Burr, D., Morrone, M., and Spinelli, D. (1989). Evidence for edge and bar detectors in human vision. *Vision Research*, 29(4):419–431.

Burt, P. and Adelson, E. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, COM-31:532–540.

Cantoni, V. and Levialdi, S., editors (1986). *Pyramidal Systems for Computer Vision*, NATO-ASI Series. Springer.

Chalmers, D. J., French, R. M., and Hofstadter, D. R. (1991). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. Technical Report 49, CRCC, Indiana University, Bloomington, IN47405.

Chui, C., editor (1992a). *An Introduction to Wavelets*. Jones and Bartlett Publishers, Inc.

Chui, C., editor (1992b). *Wavelets – A Tutorial in Theory and Applications*. Jones and Bartlett Publishers, Inc.

Cohen, A. (1989). Ondelettes, analyses multirésolutions et filtres miroirs en quadrature. Technical report, CEREMADE, Université Paris IX - Dauphine.

Coifman, R. and Wickerhauser, M. (1992). Entropy based methods for best basis selection. *IEEE Transactions on Information Theory*, 32:712–718.

Combes, J., Grossmann, A., and Tchamitchian, P., editors (1989). *Wavelets, Time-Frequency Methods and Phase Space. Proceedings of the International Conference, Marseille, France, December 14–18, 1987*. Springer, Berlin, Heidelberg, New York.

Conway, J. B. (1978). *Functions of One Complex Variable*. Graduate Texts in Mathematics. Springer-Verlag.

Crick, F. (1984). Function of the thalamic reticular complex: The search light hypothesis. *Proceedings of the National Academy of Sciences, USA*, 81:4568–4590.

Damasio, A. (1989). The brain binds entities and events by multiregional activation from convergence zones. *Neural Computation*, 1(1):123–132.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications of Pure and Applied Mathematics*, XLI:909–996.

Daubechies, I., Grossmann, A., and Meyer, Y. (1986). Painless nonorthogonal expansions. *Journal of Mathematical Physics*, 27(5):1271–1283.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1362–1373.

Daugman, J. G. (1988). Complete discrete 2-d gavor transforms by neural networks for image analysis and compression. *IEEE Trans. ASSP*, 36(7):1169–1179.

de La Mettrie, J. O. (1748). *L' homme machine*. Leiden.

de Valois, R. and de Valois, K. (1990). *Spatial Vision*. Oxford University Press.

Dirac, P. A. (1967). *The Principles of Quantum Mechanics*. Clarendon Press, Oxford.

Doursat, R., Konen, W., Lades, M., von der Malsburg, C., Vorbrüggen, J. C., Wiskott, L., and Würtz, R. (1993). Neural mechanisms of elastic pattern matching. Technical Report IR-INI 93-01, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York.

Ebeling, W., Engel, H., and Herzel, H. (1990). *Selbstorganisation in der Zeit*. Akademie-Verlag Berlin.

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., and Reitboeck, H. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, 60:121–130.

Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523.

Engel, A. K., König, P., Gray, C. M., and Singer, W. (1990). Synchronization of oscillatory responses: A mechanism for stimulus-dependent assembly formation in cat visual cortex. In Eckmiller, R., Hartmann, G., and Hauske, G., editors, *Parallel Processing in Neural Systems and Computers*, pages 105–108. North Holland, Amsterdam.

Field, D. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394.

Flaton, K. (1992). *2D Object Recognition By Adaptive Feature Extraction and Dynamical Link Graph Matching*. PhD thesis, University of Southern California.

Fleet, D. J. (1992). *Measurement of Image Velocity*. Kluwer Academic Publishers, Dordrecht, Netherlands. Foreword by Allan D. Jepson.

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *Intl. Journal of Computer Vision*, 5(1):77–104.

Fleet, D. J., Jepson, A. D., and Jenkin, M. R. (1991). Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210.

Freeman, W. J. (1975). *Mass Action in the Nervous System*. Academic Press, New York.

Frohn, H., Geiger, H., and Singer, W. (1987). A self-organizing neural network sharing fratures of the mammalian visual system. *Biological Cybernetics*, 53:333–343.

Froment, J. and Mallat, S. G. (1992). Second generation compact image coding with wavelets. In Chui, C., editor, *Wavelets - A Tutorial in Theory and Applications*. Academic Press.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.

Fukushima, K., Miyake, S., and Ito, T. (1983). Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Trans. SMC*, 13:826–834.

Gabor, D. (1946). Theory of communication. *J. Inst. Elec. Eng. (London)*, 93:429–457.

Garey, M. and Johnson, D. (1979). *Computers and Intractability*. W.H. Freeman and Co., New York.

Gaudiot, J., von der Malsburg, C., and Shams, S. (1988). A data-flow implementation of a neurocomputer for pattern recognition applications. In *Proc. of the 1988 Aerospace Applications of Artificial Intelligence Conference, Dayton, Ohio*.

Giefing, G. (1993). *Foveales Bildverarbeitungssystem zur verhaltensorientierten Szenenanalyse*. VDI Verlag, Düsseldorf.

Glünder, H. (1988). *Invariante Bildbeschreibung mit Hilfe von Autovergleichs-Funktionen*. PhD thesis, Technische Universität München.

Görz, G., editor (1993). *Einführung in die künstliche Intelligenz*. Addison-Wesley.

Gray, C. M., König, P., Engel, A. K., and Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties. *Nature*, 338:334–337.

Grossmann, A. (1988). Wavelet transforms and edge detection. In et al., S. A., editor, *Stochastic Processes in Physics and Engineering*. D. Reidel Publishing Company.

Grossmann, A., Kronland-Martinet, R., and Morlet, J. (1989). Reading and understanding continuous wavelet transforms. In *Wavelets, Time-Frequency Methods and Phase Space*, pages 2–20. Springer, Berlin, Heidelberg, New York.

Grossmann, A. and Morlet, J. (1985). Decomposition of functions into wavelets of constant shape, and related transforms. In *Mathematics and Physics, Lecture on Recent Results*. World Scientific Publishing, Singapore.

Guckenheimer, J. and Holmes, P. (1986). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Applied Mathematical Sciences 42. Springer.

Haken, H. (1978). *Synergetics – An Introduction*. Springer, Berlin, Heidelberg, New York.

Haken, H. and Olbrich, H. (1978). Analytical treatment of pattern formation in the Gierer–Meinhardt model of morphogenesis. *J. Math. Biol.*, pages 317–331.

Halmos, P. and Sunder, V. (1978). *Bounded Integral Operators on L^2 Spaces*. Springer.

Haralick, R. M. and Shapiro, L. G. (1992,1993). *Computer and Robot Vision*, volume 1,2. Addison Wesley.

Hauske, G. and Zetsche, C. (1990). Die Bedeutung des analytischen Signals in Bildanalyse und Bildkodierung. *Frequenz*, 44(2):68–73.

Häussler, A. F. and von der Malsburg, C. (1983). Development of retinotopic projections — an analytical treatment. *Journal of Theoretical Neurobiology*, 2:47–73.

Hayes, M. (1982). The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30(2):140–154.

Hebb, D. O. (1949). *The Organization of Behavior*. Wiley, New York. Partly reprinted in (Anderson et al., 1990).

Heil, C. E. and Walnut, D. F. (1989). Continuous and discrete wavelet transforms. *SIAM Review*, 31(4):628–666.

Hinton, G. and Lang, K. (1985). Shape recognition and illusory conjunctions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 252–260.

Hirsch, H. V. and Spinelli, D. (1970). Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats. *Science*, 168:869–871.

Hofbauer, J. and Sigmund, K. (1988). *The Theory of Evolution and Dynamical Systems*. Cambridge University Press.

Hofstadter, D. R. (1980). *Gödel, Escher, Bach — An Eternal Golden Braid*. Vintage Books.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558.

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

Hubel, D. H. and Wiesel, T. N. (1974). Uniformity of monkey striate cortex: A parallel relationship between field size, scatter, and magnification factor. *Journal of Comparative Neurology*, 158:295–306.

Jaffard, S. and Meyer, Y. (1989). Bases d'ondelettes dans des ouverts de $\mathbf{R}^n$. *J. Math. pures et appl.*, 68(4):95–108.

Jähne, B. (1989). *Digitale Bildverarbeitung*. Springer.

Jetschke, G. (1989). *Mathematik der Selbstorganisation*. Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden.

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

Jörgens, K. (1970). *Lineare Integraloperatoren*. B.G. Teubner.

Kendrick, K. and Baldwin, B. (1987). Cells in temporal cortex of conscious sheep can respond preferentially to faces. *Science*, 236:448–450.

Kirby, M. and Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12.

Kohonen, T. (1990). Internal representations and associative memory. In Eckmiller, R., Hartmann, G., and Hauske, G., editors, *Parallel Processing in Neural Systems and Computers*, pages 177–182. North Holland, Amsterdam.

Konen, W., Maurer, T., and von der Malsburg, C. (1994). A fast dynamic link matching algorithm for invariant pattern recognition. *Neural Networks*, 7(6/7):1019–1030.

Konen, W. and von der Malsburg, C. (1992). Unsupervised symmetry detection: A network which learns from single examples. In Aleksander, I., editor, *Proceedings of the International Conference on Artificial Neural Networks*, pages 121–125. North-Holland, Amsterdam.

Konen, W. and von der Malsburg, C. (1993). Learning to generalize from single examples in the dynamic link architecture. *Neural Computation*, 5:719–735.

Konen, W. and Vorbrüggen, J. (1993). Applying dynamic link matching to object recognition in real world images. In Gielen, S., editor, *Proceedings of the International Conference on Artificial Neural Networks*. North-Holland, Amsterdam.

Kunt, M. (1980). *Digital Signal Processing*. Artech House.

Küppers, B.-O. (1990). *Der Ursprung biologischer Information*. Piper, München, Zürich.

Lades, M., Buhmann, J., and Eeckmann, F. (1994). Distortion invariant object recognition under drastically varying lighting conditions. In Becks, K.-H. and Perret-Gallix, D., editors, *New Computing Techniques in Physics Research III*, pages 339–346. World Scientific Publ. Co.Pte.Ltd.

Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311.

Lades, M., Vorbrüggen, J. C., and Würtz, R. P. (1991). Recognizing faces with a Transputer farm. In Durrani, T., Sandham, W., Soraghan, J., and Forbes, S., editors, *Applications of Transputers*, pages 148–153. IOS Press; Amsterdam, Oxford, Washington, Tokio.

Li, Z. and Atick, J. J. (1994). Toward a theory of the striate cortex. *Neural Computation*, 6(1):127–146.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers.

Lyre, H. E. (1993). Bildverarbeitung mit Wavelets zur Extraktion von Mehrskalenkanten. Technical Report IR-INI 93-03, Ruhr-Universität Bochum, Diploma thesis, Universität Dortmund.

MacLennan, B. (1988). Gabor representations of spatiotemporal visual images. Technical Report CS-91-144, University of Tennessee, Knoxville, TN 37996.

Mallat, S. and Zhong, S. (November 1991). Characterization of signals from multiscale edges. Technical report, Courant Institute, New York University.

Mallat, S. G. (1988a). *Multiresolution Representations and Wavelets*. PhD thesis, University of Pennsylvania, Philadelphia, PA 19104-6389.

Mallat, S. G. (1988b). Review of multifrequency channel decompositions of images and wavelet models. Technical Report 412, Courant Institute, NY 10012.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7).

Mallat, S. G. and Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415.

Mallot, H. A., Dartsch, S., and Arndt, P. A. (1994). Is correspondence search in human stereo vision a coarse-to-fine process? Technical Report 4, Max-Planck-Institut für biologische Kybernetik, Spemannstraße 36, D-7206 Tübingen.

Marko, H. and Giebel, H. (1970). Recognition of handwritten characters with a system of homogeneous layers. *Nachrichtentechnische Zeitschrift*, 9:455–459.

Marr, D. and Poggio, T. (1979). A computational theory of human stereo vision. *Proc. Roy. Soc. (London) B*, 204:301–328.

Marčelja, S. (1980). Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America*, A 70(11):1297–1300.

Messiah, A. (1970). *Quantum Mechanics*. North Holland, Amsterdam. Translated from the French by G.M. Tenner.

Meyer, Y. (1989). Orthonormal wavelets. In *Wavelets, Time-Frequency Methods and Phase Space*, pages 21–37. Springer, Berlin, Heidelberg, New York.

Miller, K. D. and MacKay, D. J. (1994). The role of constraints in Hebbian learning. *Neural Computation*, 6(1):100–126.

Mjolsness, E., Gindi, G., and Anandan, P. (1989). Optimization in model matching and perceptual organization. *Neural Computation*, 1.

Morgan, M., Ross, J., and Hayes, A. (1991). The relative importance of local phase and local amplitude in patchwise image reconstruction. *Biological Cybernetics*, 65:113–119.

Morrone, M. and Burr, D. (1988). Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society London*, B 235:221–245.

Murenzi, R. (1989). Wavelet transforms associated to the n-dimensional euclidean group with dilations: Signals in more than one dimension. In *Wavelets, Time-Frequency Methods and Phase Space*, pages 239–246. Springer, Berlin, Heidelberg, New York.

Murenzi, R. (1990). *Ondelettes multidimensionelles et applications a l'analyse d'images*. PhD thesis, Université catholique de Louvain, Chemin du Cyclotron, 2, B–1348 Louvain-la-Neuve, Belgium.

Murray, J. (1989). *Mathematical Biology*. Springer.

Nakayama, K. (1988). The iconic bottleneck and the tenuous link between early visual processing and perception. In Blakemore, C., editor, *Vision: Coding and Efficiency*. Cambridge Universtiy Press.

Nicholls, J. G., Martin, A. R., and Wallace, B. G. (1980). *From Neuron to Brain*. Sinauer Associates, Sunderland, Massachusetts, 3 edition.

Nussbaumer, H. J. (1982). *Fast Fourier Transform and Convolution Algorithms*. Springer Verlag, second edition.

Olshausen, B. A., Anderson, C. H., and van Essen, D. C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719.

Omnès, R. (1990). From Hilbert space to common sense: A synthesis of recent progress in the interpretation of quantum mechanics. *Annals of Physics*, 201:354–447.

Pascal, B. (1907). *Pensées de Pascal sur la religion et sur quelques autres sujets*. Société Française d' Imprimerie et de Libraire, Paris.

Paul, T. and Seip, K. (1992). Wavelets and quantum mechanics. In Ruskai, M. B. and other, editors, *Wavelets and their applications*. Jones and Bartlett Publishers, Boston.

Pauli, W. (1961). Phänomen und physikalische Realität. In *Aufsätze über Physik und Erkenntnistheorie*. Vieweg, Braunschweig.

Perrett, D., Mistlin, A., and Chitty, A. (1987). Visual neurons responsive to faces. *Trends in the Neurosciences*, 10:358–364.

Perrett, D., Rolls, E., and Caan, W. (1982). Visual neuron responses to faces in the monkey temporal cortex. *Exp. Brain Res.*, 47:329–342.

Perrett, D., Smith, P., Mistlin, A., Milner, A. H. A., and Jeeves, M. (1984). Neurones responsive to faces in the temporal cortex: studies of functional organisation, sensitivity to identity and relation to perception. *Human Neurobiology*, 3:197–208.

Phillips, W., Hancock, P., Wilson, N., and Smith, L. (1988). On the acquisition of object concepts from sensory data. In Eckmiller, R. and von der Malsburg, C., editors, *Neural Computers*, volume 41 of *NATO ASI series F*. Springer Verlag.

Pitts, W. and McCulloch, W. (1947). How we know universals: the perception of auditory and visual forms. *Bulletin of Mathematical Biophysics*, 9:127–147. Reprinted in (Anderson and Rosenfeld, 1988).

Pollen, D. A. and Ronner, S. F. (1981). Phase relationships between adjacent simple cells in the visual cortex. *Science*, 212:1409–1411.

Pötzsch, M. (1994). Die Behandlung der Wavelet-Transformation von Bildern in der Nähe von Objektkanten. Technical Report IR-INI 94-04, Ruhr-Universität Bochum, Diploma thesis, Universität Dortmund.

Pour-El, M. B. and Richards, I. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, pages 215–239.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1988). *Numerical Recipes in C — The Art of Scientific Programmming*. Cambridge University Press.

Rinne, M. (1994). Matchen von Kantenbildern mit einem dynamischen Neuronennetz. Technical Report, Ruhr-Universität Bochum, Diploma thesis, Universität Karlsruhe.

Rodieck, R. (1965). Quantitative analysis of cat retinal ganglion cells. *Vision Res.*, 5:583–601.

Rolls, E., Baylis, G., and Leonhard, M. (1985). Role of low and high spatial frequencies in the face-selective responses of neurons in the cortex in the superior temporal sulcus in the monkey. *Vision Res.*, 25(8):1021–1035.

Rosenblatt, F. (1961). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, D.C.

Rosenfeld, A. (1984). *Multiresolution Image Processing and Analysis*. Springer Verlag.

Rumelhart, D., McClelland, J., and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press; Cambridge, MA; London.

Sagi, D. and Julesz, B. (1985). "where" and "what" in vision. *Science*, 228:1217–1219.

Schillen, T. B. and König, P. (1990). Coherency detection and response segregation by synchronizing and desynchronizing delay connections in a neuronal oscillator model. In *IJCNN 90 International Joint Conference on Neural Networks, San Diego*. IEEE.

Schneider, W. (1986). *Anwendung der Korrelationstheorie der Hirnfunktion auf das akustische Figur-Hintergrund-Problem (Cocktailparty-Effekt)*. PhD thesis, Universität Göttingen, 3400 Göttingen, F.R.G.

Seip, K. (1992). Wavelets in $H^2(\mathbf{R})$: Sampling, interpolation, and phase space density. In Chui, C., editor, *Wavelets – A Tutorial in Theory and Applications*. Academic Press.

Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4.

Stork, D. G. and Wilson, H. R. (1990). Do Gabor functions provide appropriate descriptions of visual cortical receptive fields? *Journal of the Optical Society of America A*, 7(8):1362–1373.

Tölg, S. (1992). *Strukturuntersuchungen zur Informationsverarbeitung in neuronaler Architektur am Beispiel der Modellierung von Augenbewegungen für aktives Sehen*. PhD thesis, Ruhr-Universität Bochum.

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

Turing, A. (1952). The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc.* **B**, 237.

Turk, M. A. and Pentland, A. P. (1991a). Face recognition using eigenfaces. *Journal of Cognitive Neuroscience*.

Turk, M. A. and Pentland, A. P. (1991b). Face recognition using eigenfaces. In *Proceedings of CVPR'91*. IEEE Press.

Ventriglia, F., editor (1975). *Neural Modeling and Neural Networks*. Pergamon Press, Oxford, New York, Seoul.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100. Reprinted in (Anderson and Rosenfeld, 1988).

von der Malsburg, C. (1979). Development of ocularity domains and growth behaviour of axon terminals. *Biol. Cybernetics*, 32:49–62.

von der Malsburg, C. (1981). The correlation theory of brain function. Technical report, Max-Planck-Institute for Biophysical Chemistry, Postfach 2841, Göttingen, FRG. Reprinted 1994 in: Schulten, K., van Hemmen, H.J. (eds.), *Models of Neural Networks*, Vol. 2, Springer.

von der Malsburg, C. (1983). How are nervous structures organized? In Başar, E., Flohr, H., H.Haken, and Mandell, A., editors, *Synergetics of the Brain, Proceedings of the International Symposium on Synergetics*, pages 238–249. Springer, Berlin, Heidelberg, New York.

von der Malsburg, C. (1985). Nervous structures with dynamical links. *Ber. Bunsenges. Phys. Chem.*, 89:703–710.

von der Malsburg, C. (1986). Am I thinking assemblies? In Palm, G. and Aertsen, A., editors, *Proceedings of the Trieste Meeting on Brain Theory*. Springer, Berlin, Heidelberg.

von der Malsburg, C. (1988a). Goal and architecture of neural computers. In Eckmiller, R. and von der Malsburg, C., editors, *Neural Computers*. Springer, Berlin.

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

von der Malsburg, C. (1990a). Considerations for a visual architecture. In Eckmiller, R., editor, *Advanced Neural Computers*, pages 303–312, Amsterdam. North-Holland.

von der Malsburg, C. (1990b). Network self-organization. In Zornetzer, S., Davis, J., and Lau, C., editors, *An Introduction to Neural and Electronic Networks*, pages 421–432. Academic Press, 1st edition.

von der Malsburg, C. (1990c). A neural architecture for the representation of scenes. In McGaugh, J., Weinberger, N., and Lynch, G., editors, *Brain Organization and Memory: Cells, Systems and Circuits*, pages 356–372. Oxford University Press, New York.

von der Malsburg, C. and Bienenstock, E. (1986). Statistical coding and short-term synaptic plasticity: A scheme for knowledge representation in the brain. In Bienenstock, E., Fogelman, F., and Weisbuch, G., editors, *Disordered Systems and Biological Organization. NATO Advanced Research Workshop*, pages 247–272. Springer, Berlin, Heidelberg, New York.

von der Malsburg, C. and Bienenstock, E. (1987). A neural network for the retrieval of superimposed connection patterns. *Europhysics Letters*, 3:1243–1249.

von der Malsburg, C. and Buhmann, J. (1992). Sensory segmentation with coupled neural oscillators. *Biological Cybernetics*, 67:233–242.

von der Malsburg, C. and Schneider, W. (1986). A neural cocktail-party processor. *Biological Cybernetics*, 54:29–40.

von der Malsburg, C. and Singer, W. (1988). Principles of cortical network organization. In Rakic, P. and Singer, W., editors, *Neurobiology of the Neocortex*, pages 69–99. John Wiley.

von der Malsburg, C., Würtz, R., and Vorbrüggen, J. (1991). Bilderkennung mit dynamischen Neuronennetzen. In Brauer, W. and Hernández, D., editors, *Verteilte Künstliche Intelligenz und kooperatives Arbeiten*, Informatik-Fachberichte 291, pages 519–529. Springer.

von Neumann, J. (1932). *Mathematische Grundlagen der Quantenmechanik*. Springer Verlag. English translation published by Princeton University press.

Vorbrüggen, J. C. (1994). *Zwei Modelle zur datengetriebenen Segmentation visueller Daten*. PhD thesis, Ruhr-Universität Bochum. In preparation.

Wässle, H., Peichl, L., and Boycott, B. (1986). Dendritic territories of cat retinal ganglion cells. *Nature*, 292:344–345.

Watt, R. (1987). Scanning from coarse to fine spatial scales in the human visual system after the onset of a stimulus. *Journal of the Optical Society of America A*, 4:2006–2021.

Whitten, G. (1993). Scale space tracking and deformable sheet models for computational vision. *IEEE Trans. PAMI*, 15(7):697–706.

Wiener, N. (1948, 1961). *Cybernetics*. MIT Press, Cambridge, MA. Partly reprinted in (Anderson et al., 1990).

Willshaw, D. J. and von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society, London*, B 194:431–445. Reprinted in (Anderson et al., 1990).

Willshaw, D. J. and von der Malsburg, C. (1979). A marker induction mechanism for the establishment of ordered neural mappings. *Philosophical Transactions of the Royal Society, London*, B 287:203–243.

Wilson, H. and Cowan, J. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.*, 12:1–23.

Wilson, H. and Cowan, J. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13:55–80.

Wilson, H., MacFarlane, D., and Phillips, G. (1983). Spatial frequency tuning of orientation selective units estimated by oblique masking. *Vision Research*, 23:873–882.

Wilson, H. R. and Giese, S. C. (1977). Threshold visibility of frequency gradient patterns. *Vision Research*, 17:1177–1190.

Wiskott, L. and von der Malsburg, C. (1993). A neural system for the recognition of partially occluded objects in cluttered scenes. *Int. J. of Pattern Recognition and Artificial Intelligence*, 7(4):935–948.

Wiskott, L. and von der Malsburg, C. (1994). Dynamic link matching with running blobs. in preparation.

Witkin, A., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144.

Wittgenstein, L. (1963). *Tractatus logico-philosophicus*. Suhrkamp-Verlag.

Würtz, R. P. (1992). Gesichtserkennung mit dynamischen neuronalen Netzen. *Spektrum der Wissenschaft*, pages 18–22.

Würtz, R. P. (1993). Gesichtserkennung mit dynamischen neuronalen Netzwerken. In Ahlers, R., editor, *Bildverarbeitung '93 – Forschen, Entwickeln, Anwenden*, pages 169–180. Eigenverlag Technische Akademie Esslingen, Ostfildern.

Würtz, R. P., Vorbrüggen, J., and von der Malsburg, C. (1990). A transputer system for the recognition of human faces by labeled graph matching. In Eckmiller, R., Hartmann, G., and Hauske, G., editors, *Parallel Processing in Neural Systems and Computers*, pages 37–41. North Holland, Amsterdam.

Würtz, R. P., Vorbrüggen, J. C., von der Malsburg, C., and Lange, J. (1991). A Transputer-based neural object recognition system. In Burkhardt, H., Neuvo, Y., and Simon, J., editors, *From Pixels to Features II – Parallelism in Image Processing*, pages 275–294. North Holland, Amsterdam.

Würtz, R. P., Vorbrüggen, J. C., von der Malsburg, C., and Lange, J. (1992). Recognition of human faces by a neuronal graph matching process. In Schuster, H., editor, *Applications of Neural Networks*, pages 181–200. VCH, Weinheim.

Yosida, K. (1980). *Functional Analysis*. Springer.

Zetzsche, C. and Caelli, T. (1989). Invariant pattern recognition using multiple filter image representations. *Computer Vision, Graphics and Image Processing*, 45:251–262.

Zhang, J. (1991). Dynamics and formation of self-organizing maps. *Neural Computation*, 3:54–66.

9. Anhang in deutscher Sprache

And the LORD came down to see the city and the tower, which the children of men builded.
And the LORD said, Behold, the people is *one, and they have all one language; and this they begin to do: and now nothing will be restrained from them, which they have imagined to do.*
Go to, let us go down, and there confound their language, that they may not understand one another's speech.
So the LORD scattered them abroad from thence upon the face of all the earth; and they left off to build the city.

Genesis 11:5–8

9.1 Zusammenfassung der Dissertation

Dieses Kapitel beinhaltet eine kurze Zusammenfassung der englischsprachigen Dissertation. Wo immer geeignete deutsche Fachbegriffe nicht zur Verfügung stehen, werden die englischen Bezeichnungen beibehalten. Die Einteilung in Unterkapitel entspricht der Kapiteleinteilung der Arbeit, um das Auffinden von Details im Text zu erleichtern. Dem Promotionsausschuß der Fakultät für Physik und Astronomie gilt mein Dank für die Erlaubnis, diese Arbeit in englischer Sprache einzureichen.

9.1.1 Einleitung

Diese Arbeit handelt von visueller Objekterkennung, d.h. der Fähigkeit von Lebewesen, Objekte in ihrer Umwelt wiederzuerkennen. Diese Fähigkeit ist uns so selbstverständlich, daß es überraschend ist, daß die Computerwissenschaften trotz langjähriger intensiver Bemühung noch keine Systeme hervorgebracht haben, die menschlichen oder auch tierischen Leistungen auf diesem Gebiet auch nur nahekommen.

Dabei stellt die Aufnahme und das Digitalisieren der visuellen Daten mit moderner Elektronik keinerlei Problem dar. Vielmehr liegt die Schwierigkeit darin, daß das gleiche Objekt unter verschiedenen Blickwinkeln, Beleuchtungssituationen oder teilweiser Verdeckung durch andere Objekte völlig verschiedene Helligkeitsverteilungen auf der Netzhaut bzw.

dem Kamerachip erzeugt (Abbildung 1.1 auf Seite 14). Ein Objekt ist also eine riesige Äquivalenzklasse von solchen Helligkeitsverteilungen, die wir als *Bilder* bezeichnen. Diese Äquivalenzklassen werden vom Gehirn definiert und ein gesehenes Objekt schnell und effektiv einer Klasse zugeordnet. Auf welche Weise die Repräsentation einer solchen Klasse im Gehirn stattfindet ist noch weitgehend unbekannt.

Hier werden wir von der Modellvorstellung ausgehen, daß jede Klasse durch einen (oder mehrere) Repräsentanten definiert ist, den wir *Modell* nennen werden. Die Aufgabe besteht dann darin, in einem Bild zunächst den Ausschnitt zu finden, wo ein bekanntes Objekt vorhanden ist und dann herauszufinden, welches Modell diesem Ausschnitt am ähnlichsten ist. Der erste Teil wird Segmentation genannt, der zweite Erkennung.

Wegen der zahlreichen Veränderungen, dem das Bild ein und desselben Objektes unterworfen ist, hat es sich für die Erkennung als entscheidend herausgestellt zu entscheiden, welche Punkte im Modell welchen im Bild entsprechen, bzw. Abbilder desselben Objektpunktes sind. Diese Frage wird als *Korrespondenzproblem* bezeichnet. Ein Großteil der vorliegenden Arbeit beschäftigt sich mit seiner Lösung. Das Auffinden der geeigneten Punktepaare werden wir auch *matching* nennen.

Von der Vielzahl möglicher Objektklassen wurden menschliche Gesichter ausgewählt, da hier die menschliche Erkennung, also die Beurteilung der Leistung von Modellen keine Probleme aufwirft. Außerdem unterscheiden sich Gesichter von starren Objekten dadurch, daß auch interne Verzerrungen durch Mienenspiel zu ihren natürlichen Veränderungen gehören, was das Problem besonders interessant macht. Dafür sind Beleuchtungsprobleme weniger gravierend als z.B. bei metallischen Gegenständen.

In der künstlichen Intelligenz wurde und wird versucht, Probleme des Computersehens durch Aufstellen geeigneter Regeln zu lösen. Diese Versuche hatten wenig Erfolg, da die Anzahl der Ausnahmen rasch ins unermeßliche wächst, wenn Bilder aus einer natürlichen Umwelt zu bearbeiten sind. In neuerer Zeit wird daher versucht, solche Probleme mit Hilfe von dynamischen Systemen, die aus zahlreichen lokalen einfachen und gleichen Elementen bestehen und die Komplexität ihres Verhaltens nur aus dem Zusammenspiel einer sehr großen Zahl dieser Elemente beziehen. Solche Systeme werden in der Physik seit Jahrhunderten mit Erfolg untersucht. Das Gehirn als eine große Zahl verschalteter Nervenzellen sollte dieser Art von Beschreibung im Prinzip zugänglich sein.

9.1.2 Waveletvorverarbeitung

Ausgehend von verschiedenen Darstellungen einer Wellenfunktion in der Quantenmechanik werden hier Darstellungen von Bildern diskutiert. Weder die Darstellung als Grauwertbild (Ortsraumdarstellung) noch die Fouriertransformation davon (Impulsraumdarstellung) sind für das Bildverstehen geeignet. Es wird argumentiert, daß *gemischte* oder *Phasenraumdarstellungen* geeigneter sind. Diese bestehen aus einem Satz von quadratintegrierbaren Funktionalen, die den gesamten Phasenraum abdecken.

Das Volumen dieser Funktionale im Phasenraum ist genau wie in der Quantenmechanik durch die Unschärferelation nach unten beschränkt. Es wird gezeigt, daß *Gaborfunktionen*,

d.h. Produkte aus einer ebenen Welle und einer Gaußglocke die einzigen Funktionen sind, die dieses Minimalvolumen einnehmen.

Eine elegante Art, Phasenraumdarstellungen zu verwirklichen, stellen die *Wavelettransformationen* dar. Diese bestehen aus einer geometrischen Gruppe und einer *Mutterfunktion*. Die Transformation ist dann einfach die *Faltung* der Wellenfunktion (oder des Bildes) mit allen Funktionen, die bei Anwendung der Gruppe auf die Mutterfunktion entstehen. In unserem Fall wählen wir die Gruppe aus Skalierung und Rotation der Bildebene und wenden sie auf eine Gaborfunktion an, um die Funktionale für die Wavelettransformation zu erhalten. Vorher muß jedoch die Gaborfunktion noch *zulässig* gemacht werden, d.h. ihr Integral muß verschwinden. Dies ist nötig, damit das Bild aus seiner Wavelettransformation auf einfache Weise rekonstruiert werden kann und auch allgemein für die Bildverarbeitung sehr nützlich. Schließlich wird die Wavelettransformation aus den gedrehten und skalierten Versionen einer zulässig gemachten Gaborfunktion aufgebaut, wobei die einzelnen Funktionale durch ihre *Mittenfrequenz* $\vec{k}$, das ist die Raumfrequenz der zugehörigen Welle, parametrisiert werden. Das Verhältnis σ aus der Breite der Gaußglocke und der Wellenlänge bleibt ein Parameter der ganzen Transformation.

Unabhängig von den obigen Überlegungen spricht für diese Art der Vorverarbeitung, daß sie im Gehirn tatsächlich verwirklicht ist. In der visuellen Großhirnrinde kann man sog. *einfache Zellen* finden, die auf Lichtreize genauso reagieren wie ein solches (zulässig gemachtes) Gaborfunktional.

Nach der Behandlung der wichtigsten Eigenschaften von Wavelettransformationen werden Fragen ihrer geeigneten Diskretisierung diskutiert. Es wird eine elegante Methode angegeben, um so sparsam wie möglich zu diskretisieren. Da sie auf der schnellen Fouriertransformation beruht, ist damit auch eine Möglichkeit gegeben, die Transformation effizient zu berechnen. Danach wird ein Verfahren beschrieben, das es erlaubt, aus diskretisierten Waveletdaten das ursprüngliche Bild zu rekonstruieren.

9.1.3 Darstellung von Bildern und Modellen

In diesem Kapitel werden die Überlegungen des vorhergehenden zusammengefaßt und eine geeignete Darstellung von Modellen und Bildern entwickelt. Es enthält die kompletten Vorschriften für deren Berechnung. Besonderes Augenmerk liegt hier auf der Behandlung des Bildhintergrunds. Da die einzelnen Funktionale der Transformation immer eine gewisse räumliche Ausdehnung haben, ist die Darstellung am Rand der Modelle gestört. Da die Erkennung unabhängig vom Hintergrund erfolgen soll, müssen diese Störungen ausgeschaltet werden. Hier wir dies dadurch erreicht, daß sie völlig aus der Modelldarstellung entfernt werden. Eine Konsequenz davon ist, daß die Darstellung auf niedrigen Frequenzebenen nur an Punkten in der Nähe des Objektzentrums bekannt ist, auf höheren Frequenzen können die Punkte näher an den Rand rücken.

Weiter werden alle Elemente aus der Darstellung entfernt, die eine sehr kleine Antwortamplitude aufweisen. Dies ist einerseits nützlich, um den Speicherbedarf zu verkleinern, andererseits für ein verläßliches Matching notwendig, da die wichtigen komplexen Phasen der Transformation an Stellen kleiner Amplitude sehr instabil sind.

Nachdem Modell- und Bilddarstellung vollständig definiert sind, werden erste Experimente damit durchgeführt. Diese bestehen im einzelnen aus Messungen der Qualität der Rekonstruktion bei verschiedenen Schwellwerten für die Amplituden, aus der Durchführung von affinen Abbildungen wie Translation, Rotation und Skalierung der Bildebene und Rekonstruktion aus einzelnen Frequenzebenen. Die Ergebnisse werden in den Abbildungen 3.4 bis 3.7 auf den Seiten 57 bis 62 dargestellt.

Für Objekte, die im Gegensatz zu Gesichtern nur sehr wenig interne Struktur aufweisen, wird eine weitere Darstellungsform beschrieben, die nur die lokalen Amplitudenmaxima der oben beschriebenen Transformation enthält. Erstaunlicherweise genügt auch diese Darstellung, um eine erkennbare Rekonstruktion des Bildes zu erhalten.

9.1.4 Hierarchisches Dynamic Link Matching

Hier werden zunächst die wichtigsten Grundlagen der Theorie neuronaler Netzwerke dargestellt, soweit sie für unser dynamische System, das das Korrespondenzproblem löst, wichtig sind. Dies sind im einzelnen die Dynamik eines Modellneurons, die Interaktion über synaptische Verbindungen sowie das unüberwachte Lernen, ein Modell dafür wie Organismen ihren kognitiven Apparat mit Hilfe der Informationen aus der visuellen Umwelt organisieren.

Das klassische Modell neuronaler Netzwerke betrachtet die Dynamik einer großen Zahl von Neuronen, die mit festen Verbindungsstärken vernetzt sind. Das Wissen über die Umwelt, bzw. der Algorithmus des Netzwerks steckt in diesen Verbindungsstärken. Sie sind nur langsam veränderlich, d.h. in dem Maße wie das Netzwerk sein Verhalten bzw. seine Kenntnisse über die Umwelt verändert. Ein Perzept, bzw. die Ausgabe des Netzwerks, besteht im allgemeinen in der Aktivitätsverteilung in einer spezialisierten Gruppe von Neuronen.

Dieses Modell scheint aus verschiedenen theoretischen wie experimentellen Gründen zu kurz zu greifen. Daher schlug Christoph von der Malsburg (1981) vor, daß eine geeignete Dynamik nicht nur die Aktivitäten der Neuronen sondern über die oben besprochenen Verbindungsstärken hinaus noch *dynamische Verbindungsstärken* oder *dynamische Links* enthalten muß, deren Dynamik etwa die gleichen Zeitkonstanten hat wie die der Neuronen selbst. Dabei ist die Änderung eines dynamischen Links durch die Korrelation der Aktivitäten der beiden verbundenen Zellen gegeben, d.h. gleichzeitig oder synchron aktive Zellen verstärken ihre Links, asynchron aktive Zellen schwächen sie ab. Starke Links wiederum fördern die gleichzeitige Aktivität der beiden Zellen. Damit wird ein Prozeß schneller Selbstorganisation in Gang gesetzt, der zu hochgeordneten Zuständen aus Zellaktivitäten und Linkstärken führt, die dann einem Perzept entsprechen.

Ein System zur Lösung des Korrespondenzproblems auf der Basis dieser Architektur kann wie folgt aussehen. Bild und Modell sind durch eine dichte zweidimensionale Schicht von Neuronen repräsentiert, zwischen zwei Neuronen der beiden Schichten besteht eine dynamische Verbindung. Mit jedem Neuron ist ein Satz von Merkmalsdetektoren verbunden, der die Bild- bzw. Modellinformation kodiert. Diese Kombination wird als lokales Element bezeichnet. Die Schichten sind intern so verdrahtet, daß ihre Aktivität auf eine Scheibe

lokalisiert ist, die sich über die Schicht bewegt. Die Wachstumsraten der Links sind bestimmt durch die Korrelationen der lokalen Elemente, d.h. eine geeignete Kombination aus gleichzeitiger Aktivität und Merkmalsähnlichkeit. Elementpaare mit hoher Korrelation verstärken ihre Verbindungen, solche mit niedriger schwächen sie ab. Diese Dynamik konvergiert aus einem Anfangszustand, in dem die Linkstärken nur von der Merkmalsähnlichkeit abhängen, zu einer stationären Linkverteilung, wo nur korrespondierende Punkte durch starke Links verbunden sind.

Diese Dynamik hat noch zwei Schwächen. Bei für realistische Bilder notwendigen Auflösungen sind sehr viele Zellen und entsprechend viele Links notwendig. Da die Aktivitätsscheibe jede Zelle mehrmals überstreichen muß, bis sich die korrekten Korrespondenzen herausgebildet haben, dauert dies sehr lange. Dies ist nicht nur für Simulationen unangenehm sondern würde auch in einem biologischen System zu langen Verarbeitungszeiten führen. Die zweite Schwäche besteht darin, daß das Modell nur zu einem Teil des Bildes korrespondiert und der Hintergrund abgetrennt werden muß.

Daher wird eine Dynamik vorgeschlagen, die das Korrespondenzproblem auf der in Kapitel 3 vorgestellten Darstellungen in hierarchischer Weise löst. Hierzu wird jeder Frequenzebene ein Paar von neuronalen Schichten zugeordnet. Auf der niedrigsten Ebene folgen diese Schichten der oben beschriebenen Dynamik, wobei die Modellschicht hier kleiner ist als die Bildschicht. Der Selbstorganisationsprozeß ist dann in der Lage, starke Links zwischen den Modellpunkten und den korrespondierenden Bildpunkten auszubilden. Nach einiger Zeit verläßt die Aktivitätsscheibe den zum Modell passenden Bildbereich nicht mehr und bildet hier die Korrespondenzen aus. Da diese Frequenzebene durch wenige Zellen repräsentiert werden kann, ist dies in relativ kurzer Zeit möglich.

Die Neuronenschichten der nächsthöheren Frequenzebene gehorchen einer Dynamik, die mehrere kleine Aktivitätsscheiben gleichzeitig über Bild und Modell laufen läßt. Die Linkdynamik wird hier durch die Korrelation der lokalen Elemente sowie durch die Linkstärken auf der darunterliegenden Frequenzebene getrieben. Diese Links können sich erst entwickeln, wenn die der niedrigeren Ebene eine bestimmte Schwelle überschreiten. Dadurch werden die früher gefundenen groben Korrespondenzen verfeinert. Die verschiedenen gleichzeitig auftretenden Aktivitätsscheiben können keine falschen Korrespondenzen verstärken, da sie weiter voneinander entfernt sind als der Einflußbereich der darunterliegenden Links reicht. Durch diese parallele Verarbeitung wird das Gesamtsystem deutlich beschleunigt.

Dieses Modell wurde komplett für zwei Frequenzebenen simuliert und ist in der Lage, die richtigen Korrespondenzen zwischen zwei Bildern zu finden. Zwischen einem Bild und einem Ausschnitt aus demselben Bild als Modell sind die Korrespondenzen perfekt. Für verschiedene Bilder sind sie z.T. nicht besonders genau, was erstens an Schwächen der Verfeinerungsdynamik liegt, aber auch daran, daß nur die Amplitudeninformation der Darstellungen verwendet wird. Für genaue Korrespondenzen sind jedoch die Phasen unverzichtbar, was in Abschnitt 5.4 berücksichtigt wird. Jedenfalls ist es gelungen, ein dynamisches System zu entwerfen, das auf der Basis dynamischer Links das Korrespondenzproblem hintergrundunabhängig und schneller als frühere Systeme löst.

9.1.5 Algorithmisches Matching von Bildpyramiden

Um die komplizierte Selbstorganisationsmaschine aus dem vorangegangenen Kapitel effizient auf einem sequentiellen Computer implementieren zu können, sind einige Vereinfachungen notwendig.

Die zentrale Aufgabe sowohl des Grobmatchings als auch der Verfeinerungsschritte ist es, zusammenhängende Teile der Modelldarstellung in der Bilddarstellung wiederzufinden. Für skalare Funktionen wird diese Aufgabe gemeinhin durch *Template Matching* gelöst, d.h. eine Funktion mit kleinem Träger das Template wird so lange verschoben, bis ihr normiertes Skalarprodukt mit einer Funktion größeren Trägers ihr Maximum annimmt.

Dieses Verfahren läßt sich auf vektorwertige Templates ausdehnen, wie in Abschnitt 5.2.2 gezeigt wird. Dieses Verfahren bezeichnen wir als *MTM* oder *multidimensional template matching*.

Die Templates und Datenfelder, auf die das MTM angewandt wird, werden aus den Modell- bzw. Bilddarstellungen gewonnen, indem nur die Beiträge der Filter mit Mittenfrequenzen einer festen Länge ausgewählt werden. Die verschiedenen Richtungen der Filter bilden dann die Komponenten der Vektoren, aus denen sich die Templates zusammensetzen.

9.1.5.1 Grobes Auffinden des Objektes

Ungeachtet der Tatsache, daß realistische Segmentierung sehr viel größeren Aufwand erfordert, ist es für unsere hiesige Aufgabe ausreichend, die Korrespondenzen zwischen den Punkten zuerst auf der niedrigsten Frequenzebene durch MTM zu schätzen. Da im Modell der Hintergrund unterdrückt wurde, ist die Modellebene kleiner als die Bildebene und kann als ganzes ins Bild gematcht werden. Es hat sich gezeigt, daß dies in den meisten Fällen ausreicht, um das Gesicht im Bild grob zu lokalisieren, selbst beim Vorliegen eines kompliziert strukturierten Hintergrundes (s. Abb. 5.1). Da auf der niedrigsten Frequenzebene auch die Ortsauflösung gering ist, erfordert dieses Verfahren relativ wenig Aufwand.

9.1.5.2 Verfeinerung auf der nächsthöheren Frequenzebene

Jede gegebene Korrespondenzabbildung kann auf die nächsthöhere Frequenzebene verfeinert werden, wo im allgemeinen die Ortsauflösung höher ist. Hier wäre es nicht angebracht, ein MTM auf die ganze Frequenzebene von Modell bzw. Bild anzuwenden, da das erstens keine lokalen Verzerrungen zulassen, zweitens die schon gewonnene Information aus den niedrigeren Frequenzebenen nicht ausnutzen und drittens einen relativ großen Aufwand darstellen würde.

Daher wird die Modellebene in kleine Templates eingeteilt, die im allgemeinen 2×2 Punkte enthalten. Jedes Template erhält ein Datenfeld in der Bildebene zugeordnet. Dessen Ort richtet sich nach der nächsten aus der tieferen Frequenzebene bekannten Korrespondenz. Es hat eine Größe von mindestens 3×3 Punkten und um so mehr je weiter die nächste Korrespondenz vom Zentrum des Templates entfernt war (an solchen Stellen gibt diese

Korrespondenz nur eine sehr grobe Schätzung der gesuchten wieder.) All diese Templates suchen nun in ihren Datenbereichen nach dem am besten passenden Ort. Die einzelnen MTMs sind völlig unabhängig voneinander, was bedeutet, daß sie massiv parallel implementiert werden können. Auf der anderen Seite hat es zur Folge, daß durchaus mehrere Modellpunkte zum gleichen Bildpunkt korrespondieren können (aber nicht umgekehrt).

9.1.5.3 Phasenanpassung

Für die beiden bisher besprochenen Schritte zum Erzeugen einer Korrespondenzkarte wurden nur die Amplituden der (komplexwertigen) Gaborantworten verwendet. Für eine genaue räumliche Auflösung der Korrespondenzen sind jedoch deren Phasen von besonderer Bedeutung. Die Analyse dieser Phasen ergibt folgende Eigenschaften.

In der Nähe von Nullstellen der Amplitude sind die Phasen nicht definiert, bzw. numerisch sehr instabil. Dies gilt sowohl für isolierte Nullstellen als auch für ganze Gebiete, in denen die Amplitude sehr klein ist. Wo die Amplitude eine signifikante Größe hat (z.B. größer als 5% ihres Maximalwerts ist) verhalten sich die Phasen in etwa wie die Phasen der Mittenfrequenz selbst. D.h. lokal ändern sich die Phasen etwa mit dieser Mittenfrequenz.

Damit bietet sich folgendes Verfahren zur Phasenanpassung an. Sobald klar ist, daß zwei Phasenraumatome aufgrund ihrer Amplituden und derer der Umgebung eine ungefähre Korrespondenz haben, ergibt der Quotient aus ihrer Phasendifferenz und dem Betrag der Mittenfrequenz des zugrundeliegenden Funktionals eine Verschiebung in Richtung der Mittenfrequenz. Wird der Punkt im Bild um diesen Vektor verschoben, so passen die Phasen perfekt zusammen. Für solche Antworten, deren Amplitude klein ist, so daß ihre Phasen keine Information enthalten, wird die Phasendifferenz künstlich auf null gesetzt, diese bewirken also keine Verschiebung.

Da jedoch nicht Atome sondern ganze Merkmalsvektoren einander angepaßt werden, müssen sich die Mittenfrequenzen verschiedener Orientierung noch auf einen gemeinsamen Verschiebungsvektor einigen. Dieser Vektor wird so ermittelt, daß die Summe der quadratischen Abweichungen von perfekter Phasenanpassung über die verschiedenen Orientierungen minimal wird. Dafür kann eine geschlossene Formel angegeben werden, mit deren Hilfe diese Anpassung schnell und effektiv durchzuführen ist. Mit Hilfe der Phasenanpassung ergeben sich schon auf der tiefsten Frequenzebene erstaunlich genaue Korrespondenzen, wie man in Abbildung 5.1 sehen kann.

9.1.5.4 Ausschluß von fehlerhaften Korrespondenzen

Die bisher beschriebenen Schritte zur Herstellung einer Korrespondenzkarte haben noch einen Nachteil. Bei teilweisen Verdeckungen des Objektes gibt es zwangsläufig Punkte im Modell, deren korrespondierende im Bild nicht sichtbar sind. Solche Punkte können nur aufgrund der tatsächlichen Ähnlichkeiten der als korrespondierend erkannten Merkmalsvektoren ausgeschieden werden. Dazu wird eine Schwelle für die Ähnlichkeit eingeführt und alle Korrespondenzen, deren lokale Ähnlichkeit unter dieser Schwelle liegt, aus der Karte eliminiert. Wird diese Schwelle zu hoch angesetzt, so bleiben zuwenig Korrespon-

denzen übrig, ist sie zu niedrig, so sind die Korrespondenzen nicht verläßlich.

Die Wahl dieser Schwelle ist unterschiedlich für die Karten auf der niedrigsten Frequenzebene und die durch Verfeinerung entstandenen. Auf der niedrigsten Ebene hat sich der Mittelwert der lokalen Ähnlichkeiten als brauchbar erwiesen. Beim Verfeinerungsschritt ist dies ein ungünstiges Maß, da es lokal an den Stellen, wo verfeinert wird, nicht bekannt ist. Hier dient der Mittelwert der Ähnlichkeiten auf der vorhergehenden Ebene als Schwelle.

Zahlreiche Experimente haben ergeben, daß diese Art der Bewertung von Korrespondenzen geeignet ist, einerseits eine relativ dichte Karte zu erhalten und andererseits zuverlässig Punkte auszuschließen, die keine Korrespondenz im Bild haben.

9.1.5.5 Das vollständige Verfahren

Wir haben nun vier Teilaspekte eines Verfahrens zur Erstellung einer Korrespondenzkarte vorgestellt, nämlich Templatematching auf der niedrigsten Frequenzebene, Verfeinerung einer gegebenen Karte mit Hilfe der Information auf der nächsthöheren Ebene, Anpassung der Phasen sowie Ausschluß der fehlerhaften Korrespondenzen. Diese werden folgendermaßen zu einem Gesamtverfahren kombiniert.

Um die grobe Position des Objektes im Bild zu finden, wird die entsprechende Frequenzebene aus dem Modell extrahiert und der am besten passende Ort durch Template Matching gefunden. Daraus ergeben sich erste grobe Korrespondenzen. Diese werden durch Phasenanpassung verbessert, danach werden die Punktpaare mit schlechter Ähnlichkeit eliminiert. Der Mittelwert der verbleibenden Ähnlichkeiten liefert eine Schwelle für die Ähnlichkeiten der nächsten Ebene.

Diese Karte wird dann durch lokales Template Matching verfeinert, die Phasen werden wieder angepaßt und die Paare mit unterschwelliger Ähnlichkeit eliminiert. Dasselbe Verfahren wird wiederholt, um eine Karte auf der dritten Ebene zu erhalten.

Für jede dieser Karten stehen nun die Punktpaare sowie die lokalen Ähnlichkeiten zur Verfügung. Als globale Bewertungsgrößen einer Karte werden die mittlere Ähnlichkeit, deren Standardabweichung, der mittlere Verschiebungsvektor sowie die Standardabweichungen seiner Komponenten definiert.

Eine wichtige Beobachtung ist, daß Punkte, die auf einer bestimmten Frequenzebene keine Korrespondenz finden konnten, dies teilweise auf höheren bzw. tieferen Ebenen tun. Dies gibt interessante Einblicke in die Verteilung der relevanten Bildinformation über die Frequenzebenen.

9.1.6 Hierarchische Objekterkennung

Nachdem zwei Verfahren zur Lösung des Korrespondenzproblems vorgeschlagen wurden, kann nun das Problem der Objekterkennung bearbeitet werden. Dazu werden die Karten auf den verschiedenen Ebenen nicht nur zwischen einem Modell und einem gegebenen Bild sondern zwischen einer ganzen Datenbank von Modellen und dem Bild hergestellt.

Eine Linearkombination zwischen mittlerer Ähnlichkeit und dem Betrag der (vektoriellen) Standardabweichung der Verschiebungsvektoren dient als globales Ähnlichkeitsmaß zwischen dem jeweiligen Modell und dem Bild. Das Modell mit maximaler Ähnlichkeit zum Bild ist dann das erkannte Objekt. Dies liefert für jede Frequenzebene einen Erkennungsmechanismus.

Da nicht sicher ist, ob für ein gegebenes Bild das richtige Objekt überhaupt in der Modelldatenbank vorhanden ist, muß die Qualität dieser Erkennung noch überprüft werden. Dazu werden zwei Kriterien definiert, die eine verläßliche Erkennung garantieren sollen. Das erste fordert, daß die Differenz der besten Ähnlichkeit zur zweitbesten (in Einheiten der Standardabweichung aller Ähnlichkeiten außer der besten) eine Schwelle überschreitet. Das zweite setzt eine Schwelle für die Ähnlichkeit selbst. Beide Kriterien können mit einem logischen „oder“ zu einem noch mächtigeren Kriterium verbunden werden. D.h. für eine verläßliche Erkennung muß entweder die Ähnlichkeit des erkannten Modells signifikant über der nächsten Ähnlichkeit liegen oder diese Ähnlichkeit selbst muß sehr hoch sein. Die Schwellen für diese Kriterien können so eingestellt werden, daß keine falsche Erkennung signifikant möglich ist. Die Qualität des Erkennungsverfahrens ergibt sich dann aus der Anzahl der signifikant richtig erkannten Modelle (bei Experimenten mit vielen Bildern).

Mit diesen Signifikanzkriterien kann nun ein hierarchischer Erkennungsprozeß durchgeführt werden. Die Erkennung wird zunächst auf niedriger Frequenzebene versucht. Ist sie dort signifikant, so gilt das entsprechende Modell als erkannt. Für die nicht signifikanten Erkennungen wird die Korrespondenzkarte auf der nächsten Frequenzebene verwendet. Die hier signifikant erkannten gelten ebenfalls als erkannt. Für die übrigen wird die dritte Frequenzebene herangezogen.

Dieses Verfahren hat den Vorteil, daß eine Erkennung im Durchschnitt schneller abläuft. Weiter hat sich gezeigt, daß es eine etwa 10% größere Ausbeute an signifikant richtigen Erkennungen liefert als etwa die Erkennung auf der höchsten Ebene allein.

9.1.7 Diskussion

Hier werden zunächst die hier vorgestellte Matchingverfahren mit der Methode des Matchens bewerteter Graphen, einer früheren Arbeit unter Beteiligung des Autors, verglichen und die neuen Möglichkeiten beschrieben. Es stellt sich heraus, daß die beiden Verfahren bei homogenem Hintergrund in Bild und Modell etwa gleich gut abschneiden. Bei strukturiertem Hintergrund im Bild, der anhand einer frisurinvarianten Modelldarstellung erprobt wurde, bricht das Graphmatchingverfahren völlig zusammen, während die hierarchische Erkennung noch respektable Ergebnisse liefert. Es wird diskutiert, welcher Wert von σ, d.h. des Verhältnisses von Breite zu Wellenlänge, für die Erkennung besonders geeignet sind. In der vorliegenden Arbeit wie auch im visuellen Cortex ist ein Wert von etwa $\sigma = 2$ verwirklicht, beim Matchen bewerteter Graphen wurden die besten Ergebnisse mit $\sigma = 2\pi$ erzielt. Die Gründe dieser Diskrepanz werden diskutiert, und es wird gezeigt, daß das hier vorgestellte Verfahren allgemeiner ist.

Eine große Schwäche des ganzen Computersehens ist das Fehlen einer klaren Formalisie-

rung von „natürlichen“ Bildern. Es ist sicher leicht, Bilder zu konstruieren, für die die hier vorgestellten Verfahren völlig versagen müssen, diese würde aber ein Betrachter nicht als Abbilder von natürlichen Objekten akzeptieren.

Abschließend bleibt zu sagen, daß die hier vorgestellten Verfahren einen Schritt auf dem Weg zum Verständnis kognitiver Funktionen als dynamischer Systeme darstellen. Die Vorteile sind relativ einfache formale Behandelbarkeit und vor allem die Möglichkeit, solche Systeme in Hardware zu realisieren, womit die hohen Rechenzeiten ihrer Simulation auf realistische Werte zusammenschrumpfen werden, da die massive Parallelität dann voll ausgenutzt werden kann.

9.2 Lebenslauf

Persönliche Daten

Name	Rolf Peter Würtz
Geburtsdatum	18. November 1959
Geburtsort	Heidelberg
Adresse	Wiemelhauser Straße 193, 44799 Bochum
Telefon	0234/308038, 0234/700-7996
Email	rolf@neuroinformatik.ruhr-uni-bochum.de

Schulbildung

04/66 – 09/69	Waldparkschule in Heidelberg
10/69 – 05/78	Helmholtzgymnasium in Heidelberg
05/78	Abitur

Studium

04/78	Aufnahme in die Studienstiftung des deutschen Volkes
10/78 – 04/86	Studium der Mathematik und Physik an der Universität Heidelberg
	Diplomarbeit: *Holomorphe Differentiale und Weierstraßpunkte von Artin-Schreier-Erweiterungen eines rationalen Funktionenkörpers* bei Prof. P. Roquette
04/86	Diplomprüfung mit Nebenfächern Physik und Angewandte Informatik
04/86 – 03/87	Weltreise

Beruflicher Werdegang

11/81 – 10/82, 10/84 – 03/86	Tätigkeit als studentische Hilfskraft am mathematischen Institut der Universität Heidelberg
10/85– 04/86, 04/87 – 05/88	Tätigkeit als freiberuflicher Softwareentwickler u.a. bei BASF AG, Ludwigshafen, R&R Magazines, Leimen

06/88 – 12/89	Wissenschaftlicher Mitarbeiter von Prof. Wolf Singer am Max-Planck-Institut für Hirnforschung, Frankfurt
10/88 – 12/88, 10/89–12/89	Forschungsaufenthalte an der University of Southern California, Los Angeles
seit 01/90	Wissenschaftlicher Mitarbeiter von Prof. Christoph von der Malsburg am Institut für Neuroinformatik der Ruhr-Universität Bochum
06/92, 06/93	Lehraufträge an der Technischen Hochschule Ilmenau
05/93	Mündliche Diplomprüfungen in Theoretischer Physik und Experimentalphysik an der Ruhr-Universität Bochum zwecks Zulassung zur Promotion

Veröffentlichungen

1990	(Würtz et al., 1990)
1991	(Würtz et al., 1991; Lades et al., 1991; von der Malsburg et al., 1991)
1992	(Buhmann et al., 1992; Würtz, 1992; Würtz et al., 1992)
1993	(Lades et al., 1993; Doursat et al., 1993; Würtz, 1993)

Index