Reconstruction of Objects from Images with Partial Occlusion

DISSERTATION
zur
Erlangung des Grades
„Doktor der Naturwissenschaften“
an der Fakultät für Physik und Astronomie
der Ruhr-Universität Bochum

von
Roushdi Abdelhamid

aus
Sharkia (Ägypten)

Bochum

2005
Promotionskommission
Prof. Dr. Köhler (Vorsitzender)
Prof. Dr. von der Malsburg
Prof. Dr. Koch
Prof. Dr. Goeke
PD Dr. Bomans
Abstract

Occlusion is a difficult problem in the computer vision. The system that we present here is a step to build a system that is able to deal with the occlusion problem in computer vision. The reconstruction of objects in the presence of partial occlusion is in general a hard problem for two reasons: first, we try to reconstruct an unknown object and second, the correspondence problem is not exactly solved at boundaries of the parts to keep the right configuration in the original object. In order to solve the occlusion problem in a given scene we need to segment its parts and then we can cut the occluding object and collect information about part of the occluded one.

We present in this thesis a semi-supervised algorithm to finding occluding objects and separate the visible part of the occluded object and then we reconstruct the whole object. The algorithm that we discuss here relies on the elastic graph matching that has been used successfully in face and object recognition.
Contents

1 Introduction 8
  1.1 Problem ......................................................... 8
  1.2 Occluded Objects Problem ...................................... 10
  1.3 Combination of Features in the Correct Spatial Configuration
      in Brain ......................................................... 12
  1.4 Machine Vision Systems ........................................ 13
  1.5 Outline of the Thesis .......................................... 16
  1.6 List of Mathematics Symbols ................................... 17

2 Wavelets and Object Representations 18
  2.1 Introduction ................................................... 18
  2.2 Fourier Transform .............................................. 20
      2.2.1 Windowed Fourier Transform (WFT) ..................... 21
  2.3 The 1-D Continuous Wavelet Transform ...................... 23
      2.3.1 1-D Discrete Wavelet Transform ..................... 25
  2.4 The 2-D continuous Wavelet Transform ...................... 29
      2.4.1 The 2-D Discrete Wavelet Transform ................ 30
  2.5 Gabor Filters ................................................ 31
      2.5.1 Image Representation with Gabor Wavelets .......... 33
  2.6 Properties of Gabor Wavelets ................................ 34
  2.7 Wavelet and Human Visual System ............................ 35

3 Elastic Graph Matching 36
  3.1 Motivation .................................................. 36
  3.2 Object Representation ........................................ 38
  3.3 Comparing Feature Vectors ................................... 39
  3.4 Graph matching ............................................... 41
  3.5 Bunch Graph ................................................ 41
  3.6 Discussion .................................................. 42
4 Learning Objects from their Components 44
   4.1 Introduction ............................................. 44
   4.2 Data Representation ..................................... 47
   4.3 Object Decomposition into Partial Components .......... 47
   4.4 Computation of Correlation Matrix ....................... 48
   4.5 Clustering Algorithm .................................... 49
   4.6 Grouping Partial Geons .................................. 54
   4.7 The Result with Real Objects ............................ 55
   4.8 Discussion ................................................. 55

5 Grip Finding in Partially Occluded Scenes 58
   5.1 Introduction ............................................. 58
   5.2 Object Representation .................................... 59
   5.3 Multi-Feature Vector ..................................... 60
      5.3.1 Skin-Similarity Gabor Transformed Local Image .... 60
      5.3.2 Gabor Wavelet-Transformed Local Image .......... 61
      5.3.3 Color-Averaged Local Image ....................... 61
   5.4 Complex Feature Vectors .................................. 62
   5.5 Generalization of the Algorithm .......................... 63
   5.6 Experiments and Results .................................. 65
      5.6.1 Object Free from the Hand ......................... 65
   5.7 Discussion ................................................. 71

6 Reconstruction of Object from Partially Occluded Views 72
   6.1 Occlusion Understanding .................................. 73
   6.2 The Description of the Problem ............................ 75
   6.3 Reconstruction from a Model Graph ....................... 75
      6.3.1 Reconstruction from Gabor Feature .................. 76
      6.3.2 Reconstruction from Color Feature .................. 77
   6.4 Combine the Partial Graphs .............................. 78
   6.5 Discussion ................................................. 79

7 Discussion and Outlook 82
   7.1 Discussion ................................................. 82
   7.2 Outlook .................................................. 84

Appendices 84

A Zusammenfassung in deutscher Sprache 85
   A.1 Das Ziel der Dissertation ................................. 85
   A.2 Einleitung ................................................ 86
A.3 Bildvorverarbeitung durch Gabor Wavelets . . . . . . . . . . . 86
A.4 Elastische Graphenanpassung . . . . . . . . . . . . . . . . . 87
A.5 Erkennung von Objektbestandteilen . . . . . . . . . . . . . 88
A.6 Extraktion von teilverdeckten Objekten . . . . . . . . . . . . . 89
A.7 Rekonstruktion eines Objekts aus Teilansichten . . . . . . . 90
A.8 Diskussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
Acknowledgements

Many people, other than the authors, contribute to the making of a book, from the first person who had the bright idea of alphabetic writing through the inventor of movable type to the lumberjacks who felled the trees that were pulped for its printing. It is not customary to acknowledge the trees themselves, though their commitment is total.

Forsyth and Rada

First of all, I would like to thank Prof. C. von Der Malsburg very much for his friendly treatment and giving me the chance to work with his group at the Institute für Neuroinformatik of the Ruhr-Universität Bochum, Germany for his broad experience, for his way of always finding time for discussions, for his encouraging support, and for a cheerful stimulating research environment. Furthermore, his insistence on deeper insight as well as to more precise and discussion of the experiments and their results.

I am grateful to Dr. Rolf Würtz for his constant help and giving me many useful comments.

I wish also to thank my colleagues Andreas Tewes, Bram Bolder and all developers of software in the Institute for their help to me in the starting of my work.

I wish to thank Mr. Michael Neef for the excellent administration of the workstations of the Institute.

My special thanks go to my wife for her help to take care of and rearing my children, especially my handicapped child, and I also would like to thank my parent and my parent in law to take care of my handicapped child.
To My Children and My Parents
Chapter 1

Introduction

*I would have imagine, then, that there exists in the mind of man a block of wax · · · and that we remember and know what is imprinted as long as the image lasts; but when the image is effaced, or can not be taken, then we forget or do not know.*

Plato, Theaetetus,191

*Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.*

Albert Einstein

1.1 Problem

We all live in a world of objects, mostly opaque, whose fronts hide their backs, may partly hide some other objects and, in turn, be partly hidden by others. On one hand, as humans, our perception hardly suffers when objects are partially hidden. On the other hand, the human brain does not only perceive the pieces or fragments of the objects (the parts that are directly visible), but it is usually also able to recognise the whole object from the visible parts. How can we do that without knowing anything about the whole object? What are the features that we have used to do that? Can we construct a machine that is able to do that? People should feel encouraged to try out their ideas and time will tell what works and what does not.
We humans are able to complete occluded portions of objects, because our vision system is able to temporarily store previous information about parts visible in a different views in our limbic system, specially in the hippocampus. As an example, if someone holds an object, part of this object will be invisible and occluded by the hand. If we have not seen the object before, then we have to guess the shape of the whole object, see figure 1.1. On the other hand, if we have seen different views of an object, we can collect the features and with the help of some features and our ability to restore temporary information for a short time in our vision system we can reconfigure the whole object.

In this thesis, the above process has been imitated and we have introduced a system that is able to learn to reconstruct the whole object if it has seen different visible parts of the object at least one time. The system depends on wavelet transform to model responses of simple neurons in visual cortex.
V1 (Jones and Palmer, 1987) and an Elastic Graph Matching algorithm. The EGM has been applied successfully in object recognition, especially face recognition and face identification (von der Malsburg, 1988). We would like to emphasise that the system proposed here has no any previous knowledge about the whole object but only the different parts that are directly visible. How can our vision system deal with occluded objects and what is the related work in machine vision? In order to be able to answer this question and understand the occlusion problem, we will in the next sections introduce an overview of the problem in machine vision and human systems.

1.2 Occluded Objects Problem

What is the occlusion problem? The occlusion problem means that a part of one object is hidden by another object. In other words, occluded regions in an image are spatially coherent parts that cannot be seen in that object. These regions generate discontinuities in depth, are important for any which process which must take object boundaries into account, such as pattern recognition, segmentation, motion analysis, and object identification.

The occlusion problem nowadays is one of the obstacles in many scientific fields like computer vision, computer graphics, and brain research. It has been attacked from many scientific views.

For instance, in machine vision. On one hand, the recognition of individual objects with complete shapes has been studied for a long time, can be handled without much difficulty with many existing techniques, such as shape signature, moments and Fourier descriptors (Hu, 1962; Persoon and Fu, 1977). However, problems arise when the object is partially occluded. Those aforementioned global techniques are incapable to solve the occlusion problem. This problem has importance in the industrial environment and robot recognition. Hence it has attracted plenty of interest since the 1980s. Many structural methods have been reported. Some researchers use dominant-point based recognition methods (Han and Jang, 1990). However dominant points alone are insufficient to form a complete integrated representation of an object. Polygon representation (Liu and Srinath, 1990) is another branch. However, this method has the drawback that it is unstable in finding break points for non-polygonal objects. Some researchers used one or a combination of some basic geometric features, such as line, arc, corner and end to describe a contour (Bolles and Cain, 1982; Tsang et al., 1992), but some complex object may not be fully represented by the basic geometric features.

Spectral descriptors overcome the problem of noise sensitivity and bound-
ary variation by analyzing shape in the spectral domain. Gorman proposed a partial shape recognition technique utilising local features described by Fourier descriptors. However, this approach needs approximated scale information to set the threshold for curve partitioning. Other than Fourier descriptor, the wavelet descriptor is another spectral descriptor. It has the advantage of multi-resolution approximation in both spatial and spectral domain over Fourier descriptors. Some pattern recognition approaches using wavelets have been reported (Khalil and Bayoumi, 2000). These approaches focus on the non-occluded case which rely on global information, hence they are inapplicable to solve occlusion problems. On the other hand, some researchers have generally either ignored occlusion analysis entirely or treated it as a secondary process that is postponed until matching is completed and smoothing is underway (Barnard, 1982). Other studies that have tried to find the contour of a segment of the object and try to complete the whole object are discussed in (Krolupper, 2004). A few authors have proposed techniques that indirectly address the occlusion problem by minimising spurious mismatches resulting from occluded regions and discontinuities (Hannah, 1989; Park et al., 2003; Wiskott and von der Malsburg, 1993).

Belhumeur has considered occlusion in several papers. In (Belhumeur and Mumford, 1992) point out that occluded regions, not just occlusion boundaries, must be identified and incorporated into matching. Using this observation and Bayesian reasoning, an energy functional is derived using pixel intensity as the matching feature, and dynamic programming is employed to find the minimal-energy solution. In (Belhumeur, 1993) the Bayesian estimator is refined to deal with sloping and creased surfaces. Penalty terms are imposed for proposing a break in vertical and horizontal smoothness or a crease in surface slope. Belhumeur’s method requires the estimation of several critical prior terms which are used to suspend smoothing operations. Geiger, Ladendorf, and Yuille (Geiger et al., 1995) also directly address occlusion and occlusion regions by defining an a priori probability for the disparity field based upon a smoothness function and an occlusion constraint. Also Tiehua Du and Hao Zheng have proposed a 2-D object recognition algorithm applicable for partially occluded objects see (Tiehua et al., 2004). At first the boundary of the object of interest is extracted and then the boundary is segmented into curve segments using dominant points, followed by a proportional extension. After that, each segment is represented by its wavelet descriptors at multi-scale. A hierarchical iterative matching is performed to identify the object from low to high resolution.

In psychology, Kaufman and Johnson have shown a neural correlate of object permanence in six-month-old infants: a burst of gamma-band EEG (electroencephalogram) activity over the temporal lobe that occurs during
an occlusion event and when an object is expected to appear from behind an occluder. They interpret this burst as being related to the infants representation of occluded object, For more details see (Kaufman et al., 2003). There is another psychophysical evidence that the human visual system uses geometrical occlusion relationships during binocular stereopsis (Anderson and Nakayama, 1995; Shimojo and Nakayama, 1990) to reason about the spatial relationships between objects in the world. In binocular imagery, we encounter occlusion twice. Stereo images contain occluding edges that are found in monocular views and occluded regions that are unique to a stereo pair (Belhumeur and Mumford, 1992). In this work we are interested in how to use the information from different frames to visualise the whole object.

1.3 Combination of Features in the Correct Spatial Configuration in Brain

One aim of this research is a closer understanding of how occluded object recognition takes place in human or animals brains. Studies have shown that many cells in the processing stream respond to combinations of features (including objects), but not to single features presented alone, and the features must have the correct spatial arrangement. This has been shown, for example, with faces, for which it has been shown by masking out or presenting parts of the face (for example mouth, eyes, hair) in isolation, or by jumbling the features in faces, that some cells in the cortex respond only if two or more features are present and are in the correct spatial arrangement (Perrett and Chitty, 1987). (Edmund and Alessandro, 1998) stated that evidence has been found for non-face cells. For example, some posterior inferior temporal neurons might only respond to the combination of an edge and a small circle if they were in the correct spatial relation to each other, and also an evidence consistent with this suggestion, that neurons are responding to combination of a few variables represented at the stage of cortical processing flanked by inhibitory subregions, or to combinations of colors.

Neurons that respond to several features but not to a single feature indicate that the system is non-linear (Field and Wu, 2004). Also the visual system must be able to segregate overlapping objects from one another (segmentation in machine vision) (Mata and Ringach, 2005). Evidence from lesions in humans and monkeys suggests that perceptual separation of occluded or overlapping objects involves extra-striate visual cortex. In monkeys, area V4 has been shown to play an important role in recognising occluded or poorly salient shapes. In humans, a retinotopic homologue of ventral V4 (V4v) has
been described, but it is not known whether this area is also functionally homologous to area V4 in monkeys. In another study, they tried to localise the visual cortical regions involved in perceptual segregation of overlapping shapes using positron emission tomography, see (Larsson et al., arch).

1.4 Machine Vision Systems

Computer vision refers to the automated extraction of information regarding the objects or scene in one or more images. This is an extremely useful task. With rapid advances in computer hardware and visualization systems, taking pictures is usually non-destructive and sometimes discreet, and computer vision is creeping into virtually every corner of science. The descriptions that users seek can differ widely between applications. For example, in medical imaging: One builds software systems that can enhance imagery, or identify important phenomena or events, or visualize informations obtained by imaging. Another one, as we propose in this work is to reconstruct or combine the parts of one object to integrate information about the whole object. A third example is in interpreting satellite images and analyse the objects in these images. How does the machine do that? As I mentioned above, the procedures for each application are different and depend on the intended aim, but they have steps in common.

The machine vision stages as shown in figure 1.2 and we can summarize them as following:

**Image Acquisition:** Image acquisition is the first key step to master in the computer vision process. The software used for acquisition can appear quite different depending on the type of image source and the interface. This can vary from video cameras with appropriate frame grabbers (either on a card internal to the computer or an external device connected to it by a parallel or serial interface), to digital cameras with their own serial or SCSI interfaces, to slide or flat bed scanners usually with a SCSI or USB interface. In all ways we can easily somehow pick up images or a sequence of images with different image formats. In our institute we use a firewire camera connected with a PC with double processors eight hundred megahertz and return an image with a resolution $320 \times 240$ pixels.

**Image Processing** Even with careful attention to image acquisition, control of the illumination source for uniformity, adjustment of the camera for optimum exposure and dynamic range, etc., some processing may be required to improve the image. The most common problems are noise either due to insufficient signal or inherent in the camera and electronics when high gain is use, and nonuniform illumination or brightness across the image.
Image pixel values may not cover the full dynamic range or be affected by problems in the digitisation process. These are generally adjusted using point processes that replace pixel values based on the individual values, perhaps using the histogram of all pixel values in the image to select optimum replacements.

The local operators like weighting or differencing operators may be used to decrease or increase the variance of pixel gray values respectively. Many other local operator may be used to enhance certain features like corner and edges in an image.

**Image Segmentation** A central problem, called segmentation, is to distinguish objects from background. For intensity images (i.e, those represented by point-wise intensity levels) four popular approaches are: threshold techniques, edge-based methods, region-based techniques, and connectivity-
preserving relaxation methods.

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc.

Edge-based methods centre around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring.

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighbouring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Hybrid techniques using a mix of the methods above are also popular.

A connectivity-preserving relaxation-based segmentation method (Asano and Kimura, 1995), usually referred to as the active contour model, was proposed recently. The main idea is to start with some initial boundary shape represented in the form of spline curves, and iteratively modify it by applying various shrink or expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an elastic contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is not an easy task.

Model-based Recognition In a typical recognition task, one has an image of an unknown object, and the objective is to match it with a known set of objects, given in an appropriate representation such as models or reference images. Since the viewpoint from which the image was taken and other camera parameters are unknown, the matching involves searching in a prohibitively large search space.

Of course, the representation of the object model is extremely important. Clearly, it is impossible to keep a database that has examples of every view of an object under every possible lighting condition. Thus, object views will be subject to certain transformations; certainly perspective transformations depending on the viewpoint, but also transformations related to the lighting conditions and other possible factors. Wavelets are somehow an invariant model-based representation descriptor and we use it to represent our objects see the next chapter.

From the above introduction of the properties and the recent studies on both the human and machine vision systems, we can say the human vision system is a complex system and needs a lot of effort to understand it for
the benefit of machine systems. On the other hand, over the last decade researchers succeeded to clarify many of the characteristics of the human vision system.

1.5 Outline of the Thesis

This work is aimed to simulate the brain in reconstructing an object from its visible parts, and we will try to answer the questions that we gave at the beginning of this chapter. I have organized this thesis as follows

In chapter (2), we will introduce the definition of continuous and discrete wavelet transforms and then we discuss the Gabor wavelets and their role in image processing and how we can convert the Gabor functions to DC-free wavelets.

In chapter (3) we will discuss Elastic Graph Matching as the basic element of our algorithm and the different similarity functions, specially the lateral similarity, because it was proved that the lateral similarity function plays a good role in the matching process in the case of occluded objects.

In chapter (4), we will re-implement an algorithm proposed by Ladan Shams on artificial images and try to test it on the real images, and we will also discuss a clustering algorithm that is able to cluster the correlated features.

In chapter (5) we propose a multi-feature algorithm that is able to find the fingers in objects partially occluded by a holding hand.

In chapter (6) we have examine our algorithm to reconstruct the whole object from its visible parts. Finally in chapter (7) we discuss our results and give an outlook about future work.

In appendix (A) we summarize our work in a few pages in German.
## 1.6 List of Mathematics Symbols

We use the following notation throughout this work.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{N}$</td>
<td>set of natural numbers</td>
</tr>
<tr>
<td>$\mathbb{Z}$</td>
<td>set of integer numbers</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>set of real numbers</td>
</tr>
<tr>
<td>$\mathbb{C}$</td>
<td>set of complex numbers</td>
</tr>
<tr>
<td>$\mathcal{F}$</td>
<td>Fourier operator</td>
</tr>
<tr>
<td>$f$</td>
<td>signal in spatial domain</td>
</tr>
<tr>
<td>$E$</td>
<td>energy function of signal $f$</td>
</tr>
<tr>
<td>$\hat{f}$</td>
<td>signal in Fourier domain</td>
</tr>
<tr>
<td>$I$</td>
<td>image in spatial domain</td>
</tr>
<tr>
<td>$a$</td>
<td>is the scale factor of 1-D wavelet</td>
</tr>
<tr>
<td>$b$</td>
<td>is the translation of 1-D wavelet</td>
</tr>
<tr>
<td>$a_0, a_k, b_k$</td>
<td>Fourier series coefficients</td>
</tr>
<tr>
<td>$\hat{f}(\omega, t)$</td>
<td>Windowed Fourier Transform</td>
</tr>
<tr>
<td>$\langle \cdot, \cdot \rangle$</td>
<td>scalar product</td>
</tr>
<tr>
<td>$\psi$</td>
<td>mother wavelet</td>
</tr>
<tr>
<td>$\hat{\psi}(\omega)$</td>
<td>Fourier Transform of $\psi$</td>
</tr>
<tr>
<td>$\psi_{a,b}$</td>
<td>wavelet family in 1-D</td>
</tr>
<tr>
<td>$S$</td>
<td>a sampling grid</td>
</tr>
<tr>
<td>$B_\psi = { \psi_{m,n}, (m, n) \in S }$</td>
<td>a discrete family of wavelets in 1-D</td>
</tr>
<tr>
<td>$\delta_{i,j}$</td>
<td>Dirac delta</td>
</tr>
<tr>
<td>$\psi_{S,x_0,R_\theta}$</td>
<td>wavelet family in 2-D</td>
</tr>
<tr>
<td>$\theta$</td>
<td>rotation angle</td>
</tr>
<tr>
<td>$R_\theta$</td>
<td>rotation matrix</td>
</tr>
<tr>
<td>$x_0$</td>
<td>translation vector</td>
</tr>
<tr>
<td>$S$</td>
<td>dilation matrix</td>
</tr>
<tr>
<td>$S_{\text{skin}}$</td>
<td>skin color similarity</td>
</tr>
<tr>
<td>$DC$</td>
<td>$\mathcal{F}(0, 0)$</td>
</tr>
<tr>
<td>$DFT$</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>$IDFT$</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>$WFT$</td>
<td>Windowed Fourier Transform</td>
</tr>
<tr>
<td>$EEG$</td>
<td>Electroencephalogram</td>
</tr>
<tr>
<td>$EGM$</td>
<td>Elastic Graph Matching</td>
</tr>
<tr>
<td>$GSD$</td>
<td>Geon Structural Descriptions GSD</td>
</tr>
</tbody>
</table>
Chapter 2
Wavelets and Object Representations

The graduate with a science degree asks, why does it work? The graduate with an engineering degree asks, how does it work? The graduate with an accounting degree asks, how much will it cost? The graduate with a liberal arts degree asks, do you want fries with that? - anon.

C. Garbett

2.1 Introduction

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology. Interchanges between these fields during the last ten years have led to many new wavelet applications such as image processing, turbulence, human vision, radar, and earthquake prediction.

In this chapter, we introduce wavelets to the interested technical person of the digital signal processing field. We describe the history of wavelets beginning with Fourier, compare wavelet transforms with Fourier transforms, state properties and other special aspects of wavelets, and finish with an
interesting application such as image processing with Gabor wavelets. The wavelet transform is often referred to as the wavelet decomposition. We will use the term wavelet transform hereafter.

Before 1930, the main branch of mathematics leading to wavelets began with Joseph Fourier (1807) with his theories of frequency analysis, now often referred to as Fourier synthesis. He asserted that any $2\pi$-periodic function $f(x)$ is the sum

$$f(x) = a_0 + \sum_k (a_k \cos(kx) + b_k \sin(kx)) \quad (2.1)$$

of its Fourier series. The coefficients $a_0$, $a_k$, and $b_k$ are calculated by

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x)dx,$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx)dx,$$  \hspace{1cm} (2.2)

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx)dx$$

Fourier’s assertion played an essential role in the evolution of the ideas mathematicians had about the functions. He opened up the door to a new functional universe.

After (1807), by exploring the meaning of functions, Fourier series convergence, and orthogonal systems, mathematicians gradually were led from their previous notion of frequency analysis to the notion of scale analysis. That is, analyzing $f(x)$ by creating mathematical structures that vary in scale. How? Construct a function, shift it by some amount, and change its scale. Apply that structure in approximating a signal. Now repeat the procedure. Take that basic structure, shift it, and scale it again. Apply it to the same signal to get a new approximation. And so on. It turns out that this sort of scale analysis is less sensitive to noise because it measures the average fluctuations of the signal at different scales.

The first mention of wavelets appeared in an appendix to the thesis of A. Haar (1909). One property of the Haar wavelet is that it has compact support, which means that it vanishes outside of a finite interval. Unfortunately, Haar wavelets are not continuously differentiable which somewhat limits their applications.

By using a scale-varying basis function called the Haar basis function Paul Levy, a 1930s physicist, investigated Brownian motion, a type of random
signal (Meyer, 1993). He found the Haar basis function superior to the Fourier basis functions for studying small complicated details in the Brownian motion.

Another 1930s research effort by Littlewood, Paley, and Stein involved computing the energy of a function $f(x)$:

$$ E = \frac{1}{2} \int_0^{2\pi} |f(x)|^2 \, dx $$

(2.3)

The computation produced different results if the energy was concentrated around a few points or distributed over a larger interval. This result disturbed the scientists because it indicated that energy might not be conserved. The researchers discovered a function that can vary in scale and can conserve energy when computing the functional energy. Their work provided David Marr with an effective algorithm for numerical image processing using wavelets in the early 1980s.

In (Mallat, 1999), Stéphane Mallat gave wavelets an additional jump-start through his work in digital signal processing. He discovered some relationships between quadrature mirror filters, pyramid algorithms, and orthonormal wavelet bases. Inspired in part by these results, Y. Meyer constructed the first orthogonal wavelets. Unlike the Haar wavelets, the Meyer wavelets are continuously differentiable; however they do not have compact support. A couple of years later, Ingrid Daubechies used Mallat’s work to construct a set of wavelet orthonormal basis functions that are perhaps the most elegant, and have become the cornerstone of wavelet applications today, see (Daubechies, 1992; Mallat, 1999; Kaiser, 1994).

### 2.2 Fourier Transform

Fourier transform is an important image processing tool which is used to decompose an image into its sine and cosine components and defined as:

$$ \mathcal{F}(f(\vec{x})) = \hat{f}(\vec{\omega}) = \int f(\vec{x}) \exp(-i\vec{\omega}^T \vec{x}) \, d^2\vec{x} $$

(2.4)

the output of the transformation represents the signal in the fourier domain (frequency domain), while the input signal is represented in spatial domain. In Fourier signal, each point represents a particular frequency contained in the spatial domain signal. As we are only concerned with digital images, we are interested in the Discrete Fourier Transform (DFT). The DFT is the sampled Fourier transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully
describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image. For an image $I(n)$ of size $N_1 \times N_2$, the two dimensional DFT is given by the formula:

$$
\hat{f}(\omega_1, \omega_2) = \frac{1}{N_1N_2} \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} I(n_1, n_2) \exp^{-2\pi \left(\frac{n_1 \omega_1}{N_1} + \frac{n_2 \omega_2}{N_2}\right)},
$$

where $I(n_1, n_2)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $\hat{f}(\omega_1, \omega_2)$ in the Fourier space. Equation (2.5) can be interpreted as the value of each point $\hat{f}(\omega_1, \omega_2)$ is obtained by multiplying the spatial image with the corresponding base function and summing the result. The basis functions are sine and cosine waves with increasing frequencies. The value $\hat{f}(0, 0)$ represents the DC component of the image, which corresponds to the average brightness, and $\hat{f}(N_1/2, -1, N_2/2, -1)$ represents the highest frequency. As in figure 2.1, the image was transformed as in equation (2.5) and then we have replaced each pixel with its natural logarithm with the form:

$$
\hat{f}'(\omega_1, \omega_2) = \kappa \log(1 + |\hat{f}(\omega_1, \omega_2)|)
$$

(2.6)

We have added one, since the logarithm is not defined for zero, and the scaling constant $\kappa$ is chosen so that the maximum output value is 255. That means if $\lambda$ is the maximum value in the input image, $\kappa$ is given by:

$$
\kappa = \frac{255}{\log(1 + |\lambda|)}
$$

(2.7)

On the other hand, the Fourier image can be re-transformed to the spatial domain to get the original input image by the Inverse Discrete Fourier Transform (IDFT) that can be computed as in the form:

$$
I(n_1, n_2) = \frac{1}{N_1N_2} \sum_{\omega_1=0}^{N_1} \sum_{\omega_2=0}^{N_2} \hat{f}(\omega_1, \omega_2) \exp^{2\pi \left(\frac{n_1 \omega_1}{N_1} + \frac{n_2 \omega_2}{N_2}\right)}
$$

(2.8)

From the above equation the signal can be re-transformed into the spatial domain without loss of information.

### 2.2.1 Windowed Fourier Transform (WFT)

Suppose we want to analyze a signal $f(t)$ for its frequency content. If the signal $f(t)$ consists of a single, steady note with fundamental frequency $\omega_1$, then $f(t)$ is periodic with period $P = \frac{1}{\omega_1}$ and the natural description of its
frequency contents is the Fourier series. If the signal is not periodic, there are two approaches to analyze the signal to its frequency content.

The first approach is to compute the Fourier transform \( \hat{f} \) of \( f(t) \) over the entire domain. The other approach is to divide \( f(t) \) onto subintervals and analyze each subinterval to its frequency separately. This approach is known as WFT or short-time Fourier transform and can be summarized as in the following definition see (Daubechies, 1992).

**Window Function:** Let \( g(u) \) be a function that vanishes outside the interval \( -T \leq u \leq 0 \), such that \( \sup g \subset [-T, 0] \). The function \( g \) will be a weight function, or window, which will be used to localise the signal \( f(t) \).

We allow \( g \) to be complex-valued, although in many applications it may be real. For every \( t \in \mathbb{R}^+ \) define

\[
 f_t(u) \equiv \bar{g}(u - t) f(u),
\]

where \( \bar{g}(u - t) \) is complex conjugate of \( g(u - t) \). Then \( \sup f_t \subset [t - T, t] \), where \( \sup f_t \) is the support of function \( f_t \) and we think of \( f_t \) as a localised version of \( f \) that depends only on the values \( f(u) \) for \( t - T \leq u \leq t \). If \( g \) is continuous, then the values \( f_t \) with \( u \cong t - T \) and \( u \cong t \) are small. This means that the above localisation is smooth rather than abrupt see (2.9).
We now define the WFT of the signal \( f \) as the Fourier transform of \( f_t \):

\[
\hat{f}(\omega, t) = \int_{-\infty}^{\infty} du \exp(-2\pi i \omega u) f_t(u)
\]

\[
= \int_{-\infty}^{\infty} du \exp(-2\pi i \omega u) \bar{g}(u-t) f(u)
\]

The formula (2.10) can be understood as follows; first multiply the function \( g(x) \) with the signal \( f(x) \) and compute the Fourier transform of the product \( \bar{g}(u-t)f(u) \). Because the window function \( g(u) \) has a short time duration, the Fourier transform of \( \bar{g}(u-t)f(u) \) reflects the signal’s local frequency properties. By moving \( g(u) \) and repeating the same process, we could obtain a rough idea how the signal’s frequency contents evolve over time see figure 2.2.

If we define

\[
g_{w,t}(u) \equiv \exp(2\pi i \omega u) g(u-t)
\]

then we can express the shift-time Fourier transform as inner product of \( f \) with \( g_{w,t} \)

\[
\hat{f}(\omega, t) = \langle g_{w,t}, f \rangle
\]

which makes sense if both functions are in \( L^2(\mathbb{R}) \). For more details about WFT (Kaiser, 1994; Daubechies, 1992).

### 2.3 The 1-D Continuous Wavelet Transform

In this section, we present the definition of a wavelet as in (Louis et al., 1994; Daubechies, 1992; Lee, 1996; Kaiser, 1994). A function \( \psi(x) \in L^2(\mathbb{R}) \) that satisfies the condition

\[
0 < C_\psi = 2\pi \int_{\mathbb{R}} d\omega \frac{||\hat{\psi}(\omega)||^2}{||\omega||} < \infty
\]

is called an admissible wavelet where \( \hat{\psi}(\omega) \) is the Fourier transform of the spacial wavelet \( \psi(x) \). Equation (2.13) is often referred to as the admissibility condition. In the following when we use the term wavelet we assume that the wavelet is admissible. The admissibility condition ensures that the Fourier transform of \( \psi(x) \) decays sufficiently fast when approaching to zero.
Figure 2.2: The figure shows in, a the original signal, b the window, and in c and d the localised versions $f_3(u)$ and $f_7(u)$ of the original signal as in equation (2.9).

For any function $f \in L^2(\mathbb{R})$ the continuous 1-D wavelet transform is given by:

$$
(L\psi f)(a, b) = \frac{1}{\sqrt{|a|}} \int_{\mathbb{R}} f(x) \psi \left(\frac{x - b}{a}\right) dx = \langle \psi_{a,b}, f \rangle,
$$

(2.14)

where $a \in \mathbb{R} - \{0\}$, $b \in \mathbb{R}$, $\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi \left(\frac{x - b}{a}\right)$ and $\langle .,. \rangle$ denoting the $L^2(\mathbb{R})$-inner product. The function $\psi$ is often called the mother wavelet and the functions $\psi_{a,b}$ are called a family of wavelets (Jense and la Cour-Harbo, 2001). The wavelets transform in equation (2.14) can be understood as follows: the mother wavelet $\psi$ is scaled and shifted with $a$, and $b$ respectively and then multiplied with the $f$ and summed over the range of $f$. 

24
The corresponding inverse wavelet transform, resolution of identity formula, or Calderón equation that reconstructs a function \( f(x) \) from its wavelet coefficients is

\[
f(x) = \frac{1}{C_\psi} \int_{\mathbb{R}} \int_{\mathbb{R}} (L_\psi f)(a, b) \frac{1}{\sqrt{|a|}} \psi \left( \frac{x - b}{a} \right) \frac{dadb}{a^2}
\]

\[(2.15)\]

Equation (2.15) was first proved in 1964. The integration with respect to \( a \) and \( b \) is done over the entire continuous phase space. The parameters \( a, b \) are continuous over \( \mathbb{R} \) and control the dilation and translation of the mother wavelet function \( \psi \). The term phase space is borrowed from physics and refers to the two-dimensional time frequency space, considered as a geometric whole (Daubechies, 1992). It can be seen that when the integral in the equation (2.15) diverges the function \( f \) cannot be reconstructed. Equation (2.15) can be interpreted in two ways: It shows

1. that a function \( f \) can be uniquely represented in terms of its wavelet coefficients \( (L_\psi f)(a, b) \) and that there is a one-to-one correspondence between functions \( f \in L^2(\mathbb{R}) \) and vectors in the infinite-dimensional vector space over the wavelets \( \psi_{a,b} = \psi \left( \frac{x-b}{a} \right) \)

2. that a function \( f \) can be written as a superposition of the wavelets \( \psi_{a,b} \)

### 2.3.1 1-D Discrete Wavelet Transform

It is known that the representation \( (L_\psi f)(a, b) \) of equation (2.14) is highly redundant and that the continuous phase space can be discretized without loss of information (Daubechies, 1992; Louis et al., 1994). In this sense, let \( S \subset \mathbb{R} - \{0\} \times \mathbb{R} \) be a discrete set. Then \( B_\psi = \{\psi_{m,n}, (m,n) \in S\} \) defines a discrete family of wavelets. The set \( S \) can be understood as a sampling grid of the phase space. Using the family of wavelet \( B_\psi \), the wavelet coefficients \( (L_\psi f)(m,n) = \langle \psi_{m,n}, f \rangle \) for \( (m,n) \in S \) are calculated by applying equation (2.14), the double integral is then replaced by a double sum. However, there does not exist a discrete version of equation (2.15). Hence before we can write a function \( f \) in terms of its discrete wavelet coefficients \( (L_\psi f)(m,n) \), we have to introduce some more concepts.

Obviously, for a given wavelet function \( \psi \), how well a function \( f \in L^2(\mathbb{R}) \) can be represented by its discrete wavelet coefficients \( (L_\psi f)(m,n), (m,n) \in \)
S, depends on the sampling grid S, or equivalently on the discrete family of wavelets \( B_\psi \). In order to quantify this, the term frame needs to be introduced. It is usually defined in a more general manner see (Daubechies, 1992).

**Frame :** Let \( \psi \in L^2(\mathbb{R}) \) be a wavelet, \( S \) be a sampling grid, and \( B_\psi = \{ \psi_{m,n}, (m,n) \in S \} \) a discrete family of wavelets. We say that \( B_\psi \) is a frame if there exist constants \( A > 0 \) and \( B < \infty \) such that for all \( f \in L^2(\mathbb{R}) \)

\[
A \| f \|^2_{L^2} \leq \sum_{(m,n) \in S} | \langle \psi_{m,n}, f \rangle |^2 \leq B \| f \|^2_{L^2}, \tag{2.16}
\]

where \( \| f \|^2_{L^2} = \int_{-\infty}^{\infty} | f(x) |^2 \, dx \) is the energy of the function \( f \), \( A \) and \( B \) are called frame bounds.

When a discrete family of wavelets forms a frame, it provides a complete and lossless representation of any function \( f \in L^2(\mathbb{R}) \) see (Daubechies, 1992). In order to provide more detail, we introduce some additional terms: \( B_\psi \) is called orthogonal in \( L^2(\mathbb{R}) \) if for all \( \psi_i, \psi_j \in B_\psi \)

\[
\langle \psi_i, \psi_j \rangle = \delta_{i,j} \tag{2.17}
\]

A frame \( B_\psi \) is called a basis for \( L^2(\mathbb{R}) \) if for all \( f \in L^2(\mathbb{R}) \) the linear combination \( f = \sum_k c_k(f) \psi_k \) is unique. A family of functions in \( L^2(\mathbb{R}) \) that is both orthogonal and a basis is called an orthogonal basis.

The expression \( \frac{A + B}{2} \) measures the redundancy of the frame while \( \frac{B}{A} \) measures its tightness. A frame is called tight when \( B = A \). For frame bounds \( A = B = 1 \) and \( \| \psi_i \| = 1 \), the family of functions \( B_\psi \) forms an orthogonal basis of \( L^2(\mathbb{R}) \), and any function \( f \in L^2(\mathbb{R}) \), can be uniquely written as

\[
f(x) = \frac{2}{A + B} \sum_{(m,n) \in S} (L_\psi f)(m,n) \psi_m \left( \frac{x-n}{m} \right)
\]

\[
= \frac{2}{A + B} \sum_{(m,n) \in S} \langle \psi_{m,n}, f \rangle \psi_{m,n}(x), \tag{2.18}
\]

even for frame bounds \( 0 < A \leq B < 2 \), \( B_\psi \) can still be considered to be an orthogonal frame and equation (2.18) is fairly exact. For frame bounds \( A = B > 1 \), equation (2.18) is exact, but \( B_\psi \) no longer constitutes a basis, so that the linear combination in equation (2.18), cannot be unique. In cases where \( B_\psi \) does not constitute a tight frame, i.e \( A < B \), we have to write \( f \) in
terms of the dual frame \( \tilde{B}_\psi = \{ \tilde{\psi}_{m,n} \mid (m, n) \in S \} \)

\[
f(x) = \sum_{(m,n) \in S} \langle \tilde{\psi}_{m,n}, f \rangle \psi_{m,n}(x) \\
= \sum_{(m,n) \in S} \langle \psi_{m,n}, f \rangle \tilde{\psi}_{m,n}(x).
\]

(2.19)

The two families of functions \( B_\psi \) and \( \tilde{B}_\psi \) are called dual when for each \( \psi_i \in B_\psi \) and \( \tilde{\psi}_j \in \tilde{B}_\psi \) we have

\[
\langle \psi_i, \tilde{\psi}_j \rangle = \begin{cases} 
1 & \text{if } i = j; \\
0 & \text{if } i \neq j.
\end{cases}
\]

(2.20)

For large sets \( S \), the dual wavelets in (2.19) can be computed only approximately (Daubechies, 1992). It should be kept in mind that a frame we can be reconstructed for function \( f \in L^2(\mathbb{R}) \).

In this thesis, we are not interested in finding a wavelet representation for every function \( f \in L^2(\mathbb{R}) \). Instead, we will deal with only a small subset of \( L^2(\mathbb{R}) \), so that we will not actually have to deal with wavelet families that constitute frames. But equation (2.19) still holds for non-frame wavelet families and allows to approximately reconstruct a function \( f \) with minimal error with respect to the \( L^2 \) norm.

It may be mentioned that for the discrete wavelet transform, the function \( f \) and the wavelet \( \psi \) are continuous functions. It is the phase space that is discrete here. That becomes clearer when we look at (2.14). In (2.14) the integral remains discrete because the wavelet coefficient is calculated by integration over the continuous function parameters at discrete phase space coordinates. Consequently, in equation (2.18) and (2.19), function \( f \) is written a sum over all the discrete phase space coordinates \( (m, n) \in S \).

In multi-resolution signal analysis or multi-frequency channel decomposition, as discussed in (Louis et al., 1994; Daubechies, 1992) one exploits the properties of the discrete wavelet transform to analyse signal in scale-pyramid like fashion. For this, the phase space is usually sampled with a wavelet-like grid, where the support of \( \psi \) is essentially proportional to \( a_0^m \):

\[
S = \{ (nb_0a_0^m, a_0^{-m}k_0) : m, n \in \mathbb{Z} \} \subset \mathbb{R} - \{0\} \times \mathbb{R}
\]

\[
B = \{ \psi_{m,n}^{a_0,b_0}(x) = a_0^{-m} \psi (a_0^{-m}x - nb_0) : m, n \in \mathbb{Z} \}
\]

(2.18)
Figure 2.3: Phase space sampling scheme corresponding to the discrete wavelets transform. The constant $k_0$ is given by $k_0 = \int_{0}^{\infty} |\tilde{\psi}(k)|^2 \frac{dk}{k}$; $\psi$ was chosen to be even and $a_0 = 2$ (Louis et al., 1994)

where $a_0 > 1$, $b_0 > 0$ and $k_0 = \int_{0}^{\infty} |\tilde{\psi}(k)|^2 \frac{dk}{k}$. The choice of $a_0$ and $b_0$ is directly related to the choice of the mother wavelet $\psi$: For multi-resolution signal analysis, the dilation step $a_0$ and translation step width $b_0$ are usually chosen to be 2 and 1, respectively, while the wavelet $\psi$ is often chosen such that $\psi$ is well localised in both the spatial and the Fourier domain and such that $B$ constitutes an orthonormal basis. What is exploited here is essentially the fact that the support of $\psi_{m,n}$ is proportional to $a_0^m$. As a consequence, high-frequency wavelets $\psi_{m,n}$, with $m << 0$, are greatly concentrated and involve a very small time translation step $b_0a_0^m$ which is also proportional to $a_0^m$. This means that the wavelet transform is able to “zoom in” on the signal data by using more and more concentrated wavelets $\psi_{m,n}$. In contrast to first choosing $a_0$ and $b_0$ and then the mother wavelet, as above, one might be interested in choosing the mother wavelet $\psi$ first and then finding the parameters $a_0$ and $b_0$. This allows one to investigate how the phase space should be sampled in order to achieve a frame.
2.4 The 2-D continuous Wavelet Transform

It is possible to extend the 1-D continuous wavelet transform to 2-D. For this, we need to introduce rotation $\theta$ in addition to dilation $a$, and 2-D translation $\vec{b}$. For $\psi(\vec{x}) \in L^2(\mathbb{R}^2), \vec{x} \in \mathbb{R}^2$.

The admissibility condition similar to equation (2.13) becomes.

$$0 < C_\psi = 4\pi^2 \int_0^\infty \int_0^{2\pi} d\omega d\theta \frac{\|\hat{\psi}(\omega \cos(\theta), \omega \sin(\theta))\|^2}{\|\vec{\omega}\|^2} < \infty,$$

(2.21)

where $\hat{\psi}$ is Fourier transform of $\psi$, and $\theta$ is the rotation angle. In words the function is called admissible if its integration over the whole frequency domain is bounded.

The 2-D wavelet family generated from mother wavelet $\psi$ with dilation $a$, rotation $\theta$, and translation $\vec{b}$ can be written as:

$$\psi_{a,\vec{b},\theta}(\vec{x}) = \frac{1}{a} \psi \left( R_\theta \left( \frac{\vec{x} - \vec{b}}{a} \right) \right)$$

(2.22)

The 2-D wavelet transform of the function $f(\vec{x})$ defined by the form

$$L_\psi f(a, \vec{b}, \theta) = \int_0^\infty \int_0^\infty f(\vec{x}) \psi_{a,\vec{b},\theta}(\vec{x}) d^2\vec{x}$$

$$= \left\langle \psi_{a,\vec{b},\theta}, f \right\rangle,$$

(2.23)

In words it is meant that the signal or image is convolved with a scaled, rotated, and translated kernels at the same position. The reconstruction formula similar to (2.15) then becomes

$$f = \frac{1}{C_\psi} \int_0^\infty \int_0^\infty \int_0^{2\pi} (L_\psi f)(a, \vec{b}, \theta) \frac{1}{a^3} \psi_{a,\vec{b},\theta} d\theta d^2\vec{b}.$$

(2.24)

Note that the dilation parameter $a_x = a_y = a$ is the same for both dimension. This, however can be relaxed for functions $f \in L^2(\mathbb{R}^2)$ that are separable in every coordinate (Zhang and Benveniste, 1992).

If a function $f(\vec{x})$ is separable i.e

$$f(\vec{x}) = f_1(x_1) f_2(x_2)$$

(2.25)
For such function each component is handled separately in the integral, so that for any such function \( f \in L^2(\mathbb{R}^2) \) the continuous 2-D wavelet transform is given by

\[
L_\psi f(\vec{c}, s, \theta) = \int_{\mathbb{R}^2} f(\vec{x}) \psi(SR(\vec{x} - \vec{c})) d^2\vec{x} \\
= \langle f, \psi_{\vec{c}, s, \theta} \rangle = \langle f, \psi_{\vec{n}} \rangle
\] (2.26)

with the rotation matrix \( R \), the dilation matrix \( S \), and the translation vector \( \vec{c} \) given by

\[
R = \begin{pmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{pmatrix}
\] (2.27)

and dilation or (scaling) matrix \( S \) defined by

\[
S = \begin{pmatrix}
s_x & 0 \\
0 & s_y
\end{pmatrix}
\] (2.28)

and translation vector is

\[
\vec{c} = (c_x, c_y)^T.
\] (2.29)

Here \( \theta \) denotes the rotation angle of the wavelet \( \psi(\vec{x}) \), \( s_x, s_y \) the scalings in the \( x \) and \( y \) directions, and \( c_x, c_y \) the translation in the \( x, y \) directions. In this sense, the wavelets \( \psi_{\vec{n}} \) are dilated, rotated, and translated version of the mother wavelet \( \psi \). The five-dimensional parameters vector \( \vec{n} \) is given by these parameters:

\[
\vec{n} = (c_x, c_y, \theta, s_x, s_y).
\] (2.30)

The function \( f \) can be reconstructed by integration over all wavelet parameters:

\[
f = \frac{1}{C_\psi} \int (Lf)_{\vec{n}} \psi_{\vec{n}} \frac{d\vec{n}}{|s_x s_y|} = \frac{1}{C_\psi} \int \langle f, \psi_{\vec{n}} \rangle \psi_{\vec{n}} \frac{d\vec{n}}{|s_x s_y|}
\] (2.31)

2.4.1 The 2-D Discrete Wavelet Transform

A natural way to define the discrete wavelet transform is to discretize the phase space and to assign the discrete values to the wavelet parameters as
follows (Lee, 1996; Louis et al., 1994; Daubechie, 1992): \( s_x = (s_{x0})^m, s_y = (s_{y0})^m, c_x = n s_0 (s_{x0})^m, c_y = k s_0 (s_{y0})^m, \) and \( \theta = \theta_0 = l \theta_0, \) with \( m, n, k, l \in \mathbb{Z}. \) The discrete wavelet transform is then given by:

\[
(L_d^d)(m, n, k, l) = \langle f, \psi_{mnkl} \rangle. \tag{2.32}
\]

Equation (2.32) can be interpreted as an abstract representation of \( f \) by its coefficients. To represent \( f \) uniquely (if it is possible at all), huge numbers of wavelet coefficients are generally needed. How well \( f \) is represented by its coefficients \( (L^d_d)(m, n, k, l) \) and how many are needed depends on the chosen wavelet and on the values of \( s_{x0}, s_{y0}, S_0, \) and \( \theta_0. \)

In another approach (Wiskott and von der Malsburg, 1993) the 2-D discrete wavelet transform is also used in the bunched graph approach only a few prominent feature points are represented by their wavelet coefficients. Of course, only a limited reconstruction of the image is possible in this case. The equation

\[
f = \sum_{mnkl} \omega_{mnkl} \psi_{mnkl}. \tag{2.33}
\]

allows two interpretations:

- Given the wavelet \( \psi_{mnkl}, \) an image \( f \) can be represented by the set of weights \( \omega_{mnkl}. \) Where each wavelet \( \psi_{mnkl} \) is a feature of \( f, \) the weights \( \omega_{mnkl} \) give the importance of \( \psi_{mnkl} \) is the description of \( f. \)

- The function \( f \) is approximated by a linear combination of weighted wavelets. Equation (2.33) therefore defines a template for \( f, \) with approximation quality as an additional degree of freedom.

In (Wiskott et al., 1995), the representational aspect of equation (2.33) is emphasised in the sense that the goal was to represent individual properties of faces.

\section*{2.5 Gabor Filters}

As an important example of wavelet in 1-D and 2-D wavelet we examine complex Gabor functions, that restricted by a Gaussian envelope function, were first introduced by (Gabor, 1946). In one dimension, their impulse response is given by

\[
G_{\sigma, \omega_0}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{x^2}{2\sigma} \right) \exp \left( i\omega_0 x \right). \tag{2.34}
\]
Figure 2.4: left shows a plot of the 2-dimensional Gabor functions for the even part, and in the right the odd part with two different frequencies.

In two dimensions the mathematical expression of the filter response looks like:

\[
G_{\sigma,\omega_0}(\vec{x}) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right) \exp(i\omega_0(\vec{x})) .
\] (2.35)

In the above 2-D equation, the rotation and translation parameters are omitted. The parameters \( \sigma \) and \( \theta \) are chosen beforehand as constants. Dilation, rotation, and translation are done through the wavelet parameters in (2.38, 2.39) can be split into an even real part \( \psi^e \) and odd imaginary part \( \psi^o \):

\[
G^e_{\sigma,\omega_0}(\vec{x}) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right) \cos (\omega_0(\vec{x}))
\] (2.36)

\[
G^o_{\sigma,\omega_0}(\vec{x}) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right) \sin (\omega_0(\vec{x}))
\] (2.37)
2.5.1 Image Representation with Gabor Wavelets

The Gabor function discussed above and have the shape of plane waves restricted by a Gaussian envelope function have been proven to be optimal in terms of phase space localisation and present a good model for class of neurons in early vision. To transform Gabor functions to wavelet, they should satisfy the admissible conditions (2.13, 2.21). As in (Jones and Palmer, 1987; Würtz, 1995) a term should be subtracted from equations (2.34, 2.35) and assume $\sigma_x = \sigma_y = \sigma$ to become.

$$\psi_\omega(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{x^2}{2\sigma} \right) \left( \exp (i\omega x) - \exp \left( -\frac{\sigma^2}{2} \right) \right).$$

(2.38)

And in two dimensions becomes

$$\psi_\omega(x) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{||x||^2}{2\sigma^2} \right) \left( \exp (i\omega^T x) - \exp \left( -\frac{\sigma^2}{2} \right) \right).$$

(2.39)

Now transforming or convolving an image $I(x)$ at a position $x$, with a family of Gabor wavelets of different size and orientation provides a set of complex coefficients at each pixel. This set is called a jet or feature vector $J(x)$ (we use hereafter feature vector) which represents a small patch of grey values in an image $I(x)$, and defined as

$$J_j(x) = \int I(\tilde{y}) \psi_j(x - \tilde{y}) d\tilde{y}^2$$

(2.40)

Where $\psi_j$ is a family of discrete Gabor wavelets kernels given by:

$$\psi_j(x) = \frac{k_j^2}{\sigma^2} \exp \left( \frac{k_j^2 x^2}{2\sigma^2} \right) \left( \exp (ik_j^x x) - \exp \left( -\frac{\sigma^2}{2} \right) \right)$$

(2.41)

where $k_j$ is the wave vector. By rotating and scaling the wave vector $L$ and $M$ respectively a family of Gabor wavelets can be reconstructed. If the frequency varies $0, \cdots, M - 1$, and the angle varies $0, \cdots, L - 1$ then the wave vector $k_j$ can be written in the form.

$$k_j = \begin{pmatrix} k_m \cos \phi_l \\ k_m \sin \phi_l \end{pmatrix}$$

(2.42)

Where $k_m = \frac{K_{\text{max}}}{k_{\text{step}}}$, $\phi_l = l \frac{\pi}{L}$ with index $j = l + Lm$. A feature vector $J_j(x)$ is defined as the set $J_j(x)$ of $M \times L$ complex coefficients (Lades et al., 1993) as.

$$J_j = a_j \exp(i\phi_j)$$

(2.43)

with amplitude $a_j(x)$ slowly varying with position and phases $\phi_j(x)$ varying with the spatial frequency given by the characteristic wave vector $k_j$. 

33
Figure 2.5: The figure shows that, in the left is the original image, in the middle Gabor filter at a certain direction and scale, and in the right is the result of convolution of the original image with Gabor filter and feature vector.

2.6 Properties of Gabor Wavelets

The Gabor wavelets and their associated transforms have some unique properties, which make them well suited for coding local image features, we can briefly summarized it as follows:

- Since they are DC-free, they provide robustness against varying illumination in the image.
- Robustness against varying contrast can be obtained by normalising the feature vector (2.43).
- The limited localisation in frequency as well as space domain yields a certain amount of robustness against translation, distortion, rotation, and scaling.
- Only the phase changes drastically with translation, but that can be used for estimating displacement.
- A disadvantage of the large kernels is their sensitivity to background variation, but as shown by Würtz (1995), if the object is known, the influence of the background can be suppressed.
• the Gabor wavelets are closely related to the receptive fields of simple cells in the vertebrate visual cortex (Manjunath, 1992).

2.7 Wavelet and Human Visual System

At this point, one may ask, what is the relation between wavelets and human visual system?.

The primary visual cortex V1 of mammals has been the subject of intense study for at least four decades (Smyth et al., 2003; Olshausen, 2003; Sharpee et al., 2004; Wirth et al., 2003). Hubel and Wiesel’s original studies in the early 1960’s created a paradigm shift by demonstrating that the responses of the single neurons in the cortex could be tied to distinct image properties such as the local orientation (Hubel and Wiesel, 1962; Hubel and Wiesel, 2005).

In (Jones and Palmer, 1987; David et al., 2004), the response of the simple receptive field in cat striate cortex was measured and compared with the responses of Gabor wavelets. The result was an neurophysiological evidence that, the filter response profiles of the main class of linearly-responding cortical neurons (called simple cells) are best modelled as a family of self-similar 2D Gabor wavelets. Also these results provide convincing empirical support for Marcelja’s original hypothesis (Marcelja, 1980) and Daugmans’s extension of it to two spatial dimensions (Daugman, 1985).

In (Olshausen, 2005; Shapley, 2003), a survey about almost all what is studied in the visual cortex and how we close to understand the primary visual cortex V1 is reviewed and discussion of the five non-understandable problems was given. From the above discussion we will close our chapter, and ask a question. How can we compare the representation of different intensities? The answer will be discussed in the next chapter.
Chapter 3

Elastic Graph Matching

Jeff Stilson

In this chapter we discuss the elastic graph matching algorithm as in (Wiskott, 1995). Visual pattern can be represented as a labelled graph. Nodes are labelled with local features; edges are labelled with relational features distance vectors. The process to match and compare labelled graphs is achieved by Dynamic Link Matching. DLM is a self-organising topographic mapping between a template image and a data image. The elastic graph matching has been successfully applied for many computer vision problems like identification, tracking, pose estimation, and recognition of objects.

3.1 Motivation

Graphs, particularly labeled graphs are a very universal data structure, which is used nowadays in many computer applications. Two projections of the same object, as seen in two images, may differ strongly, depending, among other things, on the point of view, the current illumination, or occlusion by other objects. The task to identify those points in the two projections which stem from the same points on the objects is defined as the correspondence problem. A full correspondence can be defined as follows:
Correspondence Between Two Images: The mapping which determines a point of an object in an image or frame to its unique corresponding point in another. Those pairs of points are considered corresponding which are projections of the same point on the object.

The correspondence problem occurs in a number of tasks related to vision, such as stereo matching, motion estimation, and object recognition, because vital information about the physical world is contained in the correspondence. In stereo matching point correspondences need to be established between two images taken at the same moment in time of the identical physical scene from two different points of view in order to determine relative differences in depth from the disparities of those pairs of points. In motion estimation tasks correspondences can be used to extract motion from two images received by the same observer at different instances in time (sequence of images). In object recognition correspondences between a stored model and new input must be established to allow identification by comparing local image features. As much as a solution to the correspondence problem is desirable as difficult it has turned out to obtain one. One of the two fully automatic systems performed significantly worse without the additional information about the location of the eyes. The only system which performed satisfactorily even when no solution to the correspondence problem was based on the Elastic Graph Matching approach (Wiskott, 1995; Würtz, 1995; Lades et al., 1993), which also serves as a basis to the work presented here.

In order to establish the required point correspondences a point description is needed which allows to retrieve a given point from an image. Characterizing a point in terms of its grey value is clearly not sufficient. In general each gray value is shared by a number of points in an image, and the description is thus highly ambiguous. The gray value is also sensitive to changes in illumination and is thus not a reliable feature. It is therefore not sufficient to describe a point in relation to other points. But this introduces new problem, because these relations between gray values may be subject to change, if for example, the object shown in an image is rotated or scaled. To deal with these changes in relation all the possible different relations can be included in the point description or alternatively the transformation properties of these relation can be included. If the ensemble of points, whose relation are considered, is well chosen these relation transformations are subject to stringent constraints. If for example, relations between points belonging to a single rigid object are considered, only those relation transformation are feasible which are consistent with the transformation properties of rigid bodies. If on the other hand, relations between points on different objects are considered nothing constrains the possible relation transformation, thus pro-
Figure 3.1: In the left is one feature vector, and in the right, the figure shows the pattern of a model graph: each node labelled with a feature vector providing no advantages over a single-point description. It is therefore desirable to increase the number of coded relations as long as no newly introduced relation transformation offset the gain in ambiguities reduction. Following this argument correspondence must be established for a whole ensemble of points, such as all points on a single object. This way the question of, (which point in one image corresponds to which point in another), turns to, (which one of given set of mappings $F$ is the actual correspondence).

3.2 Object Representation

The goal of a local image description must be to reduce the ambiguity in the description of a single point, in order to facilitate the retrieval of correspondences. Moreover, such a representation should exhibit some degree of invariant under local transformations to allow for generalization over different projections of the same feature. This and more is provided by features based upon a Gabor wavelet transform as mentioned in the previous chapter.

Let $I(\vec{x})$ be the distribution of lighting of the input image. We have represented the input image as an undirected graph $G$ with a set of vertices $V \equiv \{J_1, ..., J_n\}$. At each node in the graph $G$, we have extracted the different local features such as texture and color at this node see figure 3.1.
3.3 Comparing Feature Vectors

To compute how similar two feature vectors are which are extracted from two different images, we need a kind of similarity measurement which satisfying certain conditions. It should be invariant under those changes in the image, which are irrelevant to the correspondence problem, such as for instance illumination, local transformation, such as rotation and scaling, it should yield smooth similarity potentials to aid the search for correspondences in the transformation space, and the similarity function should not introduce new unsolicited ambiguities. We introduce four different similarity functions.

1. The cosine of the angle between the two feature vectors which introduced by (Buhmann et al., 1992; Lades et al., 1993) and defined by the formula:

\[ S_{\text{abs}}(J, J') = \frac{J \cdot J'}{||J|| \cdot ||J'||} \]  

(3.1)

Where \( ||.|| \) denotes the norm of the feature vector. This similarity function is useful as it is provides robustness to variations in degree of contrast.

2. The similarity function that contain phases and absolute values of feature vectors and invented by (Lades et al., 1993) and given by the equation

\[ S_{\text{phase}}(J, J') = \frac{J J' \cos(\phi_1 - \phi_2)}{||J|| \cdot ||J'||} \]  

(3.2)

The similarity generated by \( S_{\text{phase}} \) is very sensitive towards small image transformations, but it can be used effectively to discriminate between two feature vectors with similar amplitudes, and secondly since phase change quickly with location, it provides a mean to locate feature vectors in an image precisely.

3. The similarity function based on compensated phases and absolute values (Wiskott, 1995; Lades et al., 1993) and defined by the formula:

\[ S_{\text{dis}}(J, J') = \frac{J J' \cos(\phi_k - \phi_k - \vec{k}^T \vec{d})}{||J|| \cdot ||J'||} \]  

\[ \vec{d} = \arg \max_{\vec{d}'} \left( \frac{J J' \cos(\phi_k - \phi_k - \vec{k}^T \vec{d}')}{||J|| \cdot ||J'||} \right) \]  

(3.3)
where $\vec{k}$ is the wave vector, and $\vec{d}$ is the displacement. $S_{\text{dis}}$ is less sensitive towards small image change. $S_{\text{dis}}$ function gives a straightforward method for displacement or disparity estimation $\vec{d}$ between two feature vectors derived from close image location.

4. The lateral similarity is a non-linear sum of similarity function for example equations (3.3,3.1, 3.2). This non-linearity has a very simple form, and has a close biological analog — lateral excitation (Gilbert, 1992). To compute the similarity between two feature vectors, we augmented the graph similarity function with an element that emphasises the topological coherence of the correspondence. The lateral similarity function $S_{\text{lat}}$ involves the enhancement of each pairwise similarity value $S_{\text{sim}}$ by its neighbouring similarity values as in the equation (3.4).

$$S_{\text{lat}}(J_i, J_j') = S_{\text{sim}}(J_i, J_j') + S_{\text{sim}}(J_i, J_j') \left( \sum_l S_{\text{sim}}(J_l, J_k') \right),$$

where $l$ is the index of immediate neighbours of $J_i$, $J_k'$ is the feature vector corresponding to $J_l$ and $S_{\text{sim}}$ is one of the similarity functions $S_{\text{abs}}, S_{\text{phase}}, S_{\text{dis}}$.

The lateral similarity function $S_{\text{lat}}$ is similar to the lateral excitatory interactions among the neighbouring in V1. The second term in equation (3.4) is the excitation received by $S_{\text{lat}}(J_i, J_j')$. The amount of excitation directly depends on the value of $S_{\text{lat}}(J_i, J_j')$. This is consistent with the physiological findings in the visual cortex. The function of this lateral excitation during graph matching is to favour matches which lead to topographically smooth high similarity profiles over matches which contain topographically isolated high similarity values which tend to be accidental. The motivation behind augmentation of the simple similarity measure with lateral excitation was to fortify the algorithm against noise and occluded.

As we mentioned in our introductory chapter, we need to find correspondence features in two frames/images that have occluded parts and as we will discuss in chapter (6), our algorithm involves matching of the model graph with another images while only part of the model graph has a corresponding match in the input graph. That is, there are parts of the model graph which do not have corresponding match in the input graph. These occlusions cause the total graph similarity to be low, causing the total feature vectors similarity of the desired match to potentially become comparable to those of false correspondences which accidentally contain some disparate high similarity nodes.
In such instances, lateral similarity favours the correct match due to its contiguity of high similarity nodes. In our algorithm we have use the lateral similarity for the above reasons.

3.4 Graph matching

As we have mentioned above each visual pattern can be represented via a model graph $G(V, E)$ with local features and links encoding the topological relationship between the features (von der Malsburg, 1988). With this assumption we can formulate the problem of object recognition as graph matching, where the goal of the graph matching process is to find one-to-one correspondence between feature vectors in the input model graph and those of stored model (Bienenstock and von der Malsburg, 1987).

To find the best match of a model graph in an image, the model graph is moved and scaled over a feature image (I mean an image that have the same feature of the features of the model graph). The best position of the model graph in the image is the position at which the graph feature vectors have highest similarity with the feature vectors extracted from the image. The process of finding the best matching based on the matching parameters that be used in the matching process.

The correspondence between a model and an image by elastic graph matching (Wiskott, 1995) is achieved in six different moves. First the rigid model graph is moved over the entire image on a coarse grid (scan global move). In the surrounding area of the best position a scan global move is executed again, but with finer grid. In the new determined place in the image the graph is scaled vertically and horizontally until it fits the best (scan scale move). Again a scan global move without any grid take place, after that a disparity scale move take place. The last adjustment of the model graph is performed by the scan local move see figure 3.2.

3.5 Bunch Graph

Rigid objects or objects that have common structures like human faces can be represented with a model graph (Wiskott et al., 1999). The problem of face recognition can be described as a graph matching as following. Each face is represented as a model graph with $N$ nodes at the face landmark and stored in a model graph, see figure 3.3, each model graph can be added to generate a set of model graphs that are arranged in a specific order called bunch graph. Bunch graph matching depends on the idea that objects within one
class have common fiducial points, that can be identified for each member of the class, and further that these fiducial points have approximately identical geometrical relations for all members and only differ in their appearance. One major advantage of bunch graph matching over elastic graph matching is that recognition of a specific member of a class is decoupled from solving the correspondence problem. Once the correspondence is established by a class specific bunch graph, the similarities needed for recognition can be obtained by simple comparison of one set of local image features, which is only once extracted from the image and all available models.

3.6 Discussion

After we have introduced the idea of the elastic graph matching and its extension bunch graph matching, we need to discuss why we used the graph
Figure 3.3: The figure shows, how the bunch graph is generated from different faces. The upper row contains images of faces, the middle row contains feature vectors extracted at the right eye of each face. These are added to form a stack-like structure of feature vectors which we call bunch jet. Finally, bottom row a graph of facial landmarks is generated by attaching at each node the corresponding feature vector. Combined together they generate the bunch graph, as displayed in the bottom row to the right.

matching to be the basis of our algorithm. In the last few years graph matching has been successfully used in many branches of computer vision like medical images, satellite image, and face recognition. EGM has been proved (Irniger, 2005; Würtz, 1995; Wiskott and von der Malsburg, 1993) to be one of the best algorithms for face recognition and recognition systems that uses a pre-stored information about the model class. On the other hand, system like template matching, Principal Component Analysis, Independent Component Analysis have not been successfully applied for non-faces objects, while neural network systems need a lot of time to learn the examples.
Chapter 4

Learning Objects from their Components

If asked what aspect of vision means the most to them, a watchmaker may answer 'acuity', a night flier 'sensitivity', and an artist 'color'. But to animals which invented the vertebrate eye and hold the patents on most features of the human model, the visual registration of motion was of the greatest importance.

Gordon Lynn Walls

4.1 Introduction

In this chapter, we introduce an algorithm for object recognition which supported by a psychological evidence as discussed by Biederman (Biederman, 1987). Biederman suggests a theory (called Geon Structural Descriptions GSD) of human pattern recognition. The theory suggests that there are three stages in human recognition.

1. We sub-divide the object into its individual components.

2. Each sub-object is then classified. Biederman suggests that there are thirty-six such categories, called geons. These geons are a kind of alphabet for composing objects, such as letters serve as an alphabet for building up words. Biederman states, recognizing a geon involves recognizing the features that define it where these features describe elements
of its generation. He draws a similarity between recognising a geon and recognising a letter.

3. Once we have identified the pieces of an object and their configuration together, we recognize the object as the pattern composed from these pieces. Just as we build up words by recognizing each letter in turn, so the same happens when we try to recognize an object. We recognize each letter to build the word, so we recognise each geon to recognize the object.

In (Shams et al., 2000; Shams, 1999) the above procedure was applied to point-view invariant objects like cone, cylinder, and cube (called geons or shape primitives). The algorithm uses synthetic gray-level objects, each composed of several parts or geons. As Biederman has proposed to learn the whole object, it should be sub-divided into its geons parts. Some of data that were used in the above algorithm is displayed in figure 4.1.

We have re-implemented the above algorithm to test it with the real objects, and we have found that: the results in the case of graphical shapes is better than the results in the real one. The reasons are the inter-correlation between real occluded shapes is not perfectly recognised and then we have difficulty to separate the shapes in the same view.

A part-based representation in which objects are represented in terms of the constituent shape primitives and the topological relationship amongst the parts (Biederman, 1987) possesses many important in-variances consistent with those of the brain. This representation accounts for a body of psychophysical data on object recognition (Biederman and Cooper, 1991) and seems to be an appropriate model of object representation employed by the brain at least for a considerable class of objects. Entry-level object recognition, for instance, discrimination between a book and a chair, seems to be robust to rotation in depth even for novel objects, and this model is the only one which can account for this capacity. Of course, it is likely that for subordinate object classification, e.g., we mean discrimination of one type of book and another. For irregular objects, other representations and recognition mechanisms are used. However, even for such recognition tasks, the human brain would have to exploit the regularities in the varying patterns in order to achieve invariant recognition. Thus, whether the regularities are in the form of object shape primitives, or other constellations of feature which may not be as verbally describable and intuitive as volumetric object parts, the recurring patterns, or feature combinations seem to be the basis for invariant recognition. In this chapter, we focus primarily on regular object shape primitives.
Despite extensive efforts focusing on the recovery of shape primitive-like structures (Brooks, 1983), no systematic investigation of the acquisition of the shape primitives in the brain has yet been conducted, neither on the experimental front nor within the computational modelling field. Therefore, it is not clear whether such representation, which seems to underlie much of the adult object recognition mechanisms, are the result of a learning process or arise through maturation encoded by genes. The first step in answering this question is to investigate whether, given the biological constraints, such a learning task is computationally feasible. The learning model presented here uses segmented texture free, uniform color object representations as input. Each object is represented by a single static image shown in figure 4.1.
4.2 Data Representation

As we have mentioned in chapter (3) and as discussed in detail in (Wiskott, 1995), each object is represented by a lattice of nodes, where each node is labeled with a Gabor feature vector. A feature vector is a set of extracted wavelets with different orientation and scales. We model visual feature cells by the magnitude of complex-valued Gabor wavelet responses (Lades et al., 1993). In figure 4.2 a grid graph was displayed on an object where each node is labelled by Gabor feature vector defined in (4.1).

\[
J_k(n) = \sum_{m} I(m) \psi(n - m), \tag{4.1}
\]

where \( \vec{k} \) is the wave vector, \( \vec{n}, \vec{m} \) are pixel positions, and \( \psi(\vec{n}) \) is the Gabor wavelet response at position \( \vec{n} \) given by equation (2.41).

4.3 Object Decomposition into Partial Components

After we have represented each object as a grid graph, each object is matched against all other objects in the data base. To compute the similarity between two graphs, we have used the lateral total similarity between objects \( k \) and \( l \), which is given by the form:

\[
S_{lat}(G_k, G_l) = \frac{1}{n} \sum_{i} \bar{s}(f_i, f'_j), \tag{4.2}
\]

where

\[
\bar{s}(f_i, f'_j) = s(f_i, f'_j) + \frac{1}{n} \sum_{k} s(f_k, f'_p) \tag{4.3}
\]

and where \( k \) is the index of immediate neighbours of \( f_i \), \( f'_p \) is the correspondence feature vector with \( f_k \), and \( s(f_i, f'_j) \) is one of the similarity functions that we have discussed in chapter (3), in our case \( S_{abs} \). If the similarity \( S_{lat} \) is below a threshold \( t \), the match is discarded. If it is above that threshold, then we compute the similarity of feature vector \( i \) in object \( k \), \( f'^k_i \), and its matching feature vector in object \( l \), \( f'^l_j \) with running index \( l \). We refer to it as \( H_{kl}^{il} \), given by the form:

\[
H_{kl}^{il} = \bar{s}(f'^k_i, f'^l_j), \quad S_{lat}(G_k, G_l) > t \tag{4.4}
\]

The similarity vector is displayed in figure 4.3.
Success in the matching process is achieved mostly when there is at least one geon in common between the two objects and the match has found this correspondence. The match is not a trivial task, as a given geon in general is partially occluded differently in another object, lowering the similarity. Also, even if the two matching geons cause a very high similarity the match may still be discarded based on low total similarity $S_{tot}$ due to the presence of non-matching geons or non-geonic parts.

4.4 Computation of Correlation Matrix

To find the subgraph correspondence to an object $k$, we consider its similarity feature vector $H^{kl}_i$ with running index $l$. We denote for the set of similarity
Figure 4.3: This figure shows how we compute the similarity for one feature vector \( i \) in object \( k \) of feature vector \( i \) in object \( k \) with its correspondence in objects \( l \) with \( H^k_i \) see figure 4.3. It is to be expected that two feature vectors \( i \) and \( j \) located within a geon region in object \( k \) either both have high similarity or have both low similarity. On this expectation we calculated the pairwise correlations \( R^k_{ij} \) with the form:

\[
R^k_{ij} = \frac{H^k_i \cdot H^k_j}{\|H^k_i\| \cdot \|H^k_j\|}. \quad (4.5)
\]

Figure 4.4 shows the correlation matrix obtained for the object shown on 4.2. The correlation matrix is a square matrix with equal number of column and rows equal to the number of nodes of the model graph as it is clear from equation (4.5).

In order to group the highly correlated nodes in a model graph from the correlated matrix, we try to use a suitable clustering algorithm. In the next section we will introduce the clique finding clustering algorithm.

### 4.5 Clustering Algorithm

After we have computed the correlation matrix \( R^k \) for object \( k \), and thresholding it, we represent it by a graph where the nodes represent the feature vector of the model graph, and links represent the correlation between the nodes \( i \) and \( j \). The cluster algorithm that do this process can be defined as follows:

\[
C^k = C(R^k) = \{ g_p : |g_p| > 4 \}; \quad \bigcup_p g_p = G^k; \quad g_p \cap g_q = \emptyset \quad (4.6)
\]

where \(| . |\) represent the set cardinality. The cluster finding algorithm \( C \) partitions the graph \( G^k \) in such a way that the nodes with highly intercorrelated
Figure 4.4: This figure shows how we compute the correlation matrix for one object.
Figure 4.5: This figure shows how we compute the correlation matrix for one object \( k \) and the result of the clustering algorithm given in equation (4.6).
similarity vectors $H_i^k$ get clustered together forming subgraph $g_p$ and returns only clusters with 4 or more nodes. How can we compute the intercorrelated subgraph?

The problem of binding the object nodes together based on their pairwise data can be framed in graph context: the features to be bound together are interpreted as graph nodes, and the pairwise data are interpreted as graph links. The problem of finding intercorrelated groups of features would be equivalent to the problem of finding cliques in a graph. A clique is the largest fully interconnected subgraph in a graph.

We describe an algorithm for solving the clique problem outlined above. The intuition behind this heuristic is as follow: interpret each node in the graph as a mass, and imagine a gravitational force between those nodes which are connected to each other via an edge and no such force otherwise. Such gravitational system should lead to the collapse of the nodes belonging to a clique into one point in space, due to the high connectivity among those nodes. On the other hand, the nodes which do not belong to any cliques would most likely get pulled by different sporadic masses to which they are connected but never strongly enough in one coherent direction so that they get absorbed by a clique. Thus, one can expected that the members of each clique in the graph would collapse into one point and the dynamics would converge to state where the only remaining nodes are either those consisting of a clique or, those which do not belong to any cliques. The cliques in this scheme can then be identified by excluding the masses which are composed of more than a couple of nodes. What remains is those nodes which are the result of the collapse of several nodes into one point, each corresponding to a clique.

At each iteration, a node $i$ is selected at random, and its position $x_i$ is changed by

$$\Delta x_i = \alpha \frac{f_i}{\|f_i\|}, \quad (4.7)$$

where

$$f_i = \sum_j \frac{d_{ij}}{\|d_{ij}\|}, \quad (4.8)$$

where $j$ represents the nodes which are connected to node $i$, and $d_{ij}$ the vectors connecting node $i$ to nodes $j$. Equation (4.8) illustrates the net force which is exerted to node $i$ by all nodes $j$. Each node $j$ pulls node $i$ towards itself. The sum of such forces can add up to large amount, pulling node $i$ in the correct direction, however, far too much moving to a position far beyond the location of nodes $j$. To prevent this the amount of the move should be re-adjusted such that node $i$ would not pass through the cloud of nodes $j$,
but rather only get closer to them. To this end, we move node $i$ in direction of $f_i$, however for a distance $\alpha$ which is half of the median distance between node $i$ and nodes $j$.

The choice of the criterion for stopping the iteration is not trivial. It turned out, however, that a very simple condition was highly adequate for our object decomposition application. The iteration was stopped as soon as a move $\Delta x_i$ became smaller than a threshold which is a very small positive number. The algorithm is highly robust to this threshold. The reason for success of this convergence criterion is that when a move is nearly zero, all members of a given clique have already converged to the same position. This is the case because each member of the clique always moves towards approximately the centre of mass of the clique and the amount of the move is proportional to its distance to the clique members; the farther a node is from the centre of a nodes $j$ cloud, the larger the move. This way all clique members collapse into one point roughly at the same time. The stability of one node would signal the stability of all. Some result of this cluster algorithm is displayed in figure 4.5.
4.6 Grouping Partial Geons

In the above steps, we have separated each object into its individual components (cube, cone, and cylinder) and non-components parts. Now we have to collect similar components together (we mean the cone together, cylinders together, and cubes together). We repeat the process of decomposition of each object into its individual components, but this time the matching process is performed on different components and non-components parts with the formula:

\[ m_{kl} = S_{lat}(g_k, g_l), \]  

(4.9)

where \( k \) and \( l \) are indices for all partial geons resulted from the decomposition of the all object graphs, see 4.5. After that, we compute the cross correlation between the match similarities of pairs of partial geons (4.9).

\[ r_{kl} = \frac{m_k^T m_l}{\|m_k\| \|m_l\|} \]  

(4.10)

where \( m_x \) denotes a vector of \( m_{xy} \) with running index \( y \). To group together the partial geons which are intercorrelated in terms of their matching pattern, we apply the clustering algorithm to the cross correlation matrix computed by (4.10).
After we have collected similar partial geons separately, the parts of the cone are collected together and the same are done for parts of the cube and the cylinder. To this end, we fuse the model graphs within one aggregate by overlaying them with each other in the relative position found in the match and by averaging the feature vectors coming to lay on top of each other. If you need more information about the algorithm that we have discussed above, you maybe should to back to (Biederman, 1987), and (Shams, 1999).

4.7 The Result with Real Objects

We have tested the above algorithm with geon-similar real objects (e.g. real objects but similar to that used by Ladan Shams), see figure 4.8, and we have found that the correspondence problem at the boundary of the overlapped objects is difficult to solve and then the separation of the whole objects into their geons parts. From our experiment, we conclude that for computer generated geons which are texture-less and whose background is very simple, we can easily find a good correspondence in a rotated view of the same geon. In the case of real objects, the process of finding the right correspondence is very difficult because the feature vectors extracted from rotated views are not similar to the input feature vectors and we then get a few intercorrelated nodes and the correlation matrix is very ambiguous. Due to this ambiguity, it is hard to find the right threshold to control the clustering algorithm. In the case of computer generated texture-less geons the algorithm can succeed to find the high intercorrelated nodes because the geons are view-independent objects and if the extracted node at one position in a geon in one view is the same as in a different position in another view of the same geon. We can say that this algorithm has the following shortcomings.

- The input data must be learned in two dimensions.
- Constancy of goon’s orientation in plane.
- It is applicable only to simple and not to general objects.

4.8 Discussion

We would like to emphasise that the focus of this dissertation is not recognize the geons in a scene, but reconstruction of objects from images with partial occlusion, but we have discussed this algorithm to use it in the process of separating two partially occluded objects in general. As we have mentioned
above the algorithm is very difficult to apply to real objects. What remains an open question is where GSD theory of human object recognition falls along this continuum. Given Biederman and Gerhardstein’s demonstrations of viewpoint invariance it would be tempting to associate GSD theory with the viewpoint invariant effects that occur in some recognition judgements. However, our analysis indicates that:

**Generality.** The conditions proposed for obtaining viewpoint invariance do not characterize everyday object recognition. What these conditions define is an instance of recognition by unique features. Moreover, there is little evidence to indicate that the features specifying geons remain unique under typical recognition conditions or beyond demonstrations using only restricted sets of objects.

**Current evidence.** The GSD theory is inconsistent with the wide range
of studies that find viewpoint-dependent recognition performance. There are currently no well-grounded reasons to discount viewpoint-dependent recognition effects as arising from non-recognition systems or experimental artifacts. Moreover, demonstrations of viewpoint invariance using familiar common objects necessarily fail to distinguish between previously learned multiple-view and view-invariant structural descriptions. In contrast there are many specific results that are explained by multiple-views, but not GSD theory.

**Explanatory power.** GSD theory does not provide an account of entry-level recognition. In some cases, GSD theory represents different entry-level items as the same objects, in other cases it represents the same entry-level items as different objects. Biederman and Gerhardstein’s results indicate that different exemplars of the same entry-level category are encoded as distinct representations.

These are some notes about the GSD theory, more details are found on (Biederman, 1987; Biederman and Cooper, 1991; Tarr and Bülthoff, 1995). After we have found that, the GSD theory is not general for all objects, we will try to find an algorithm that succeeds to separate two occluded objects, for instance, to separate an object from the hand. How do we do that? The answer of this question we will discuss in the next chapter.
Chapter 5

Grip Finding in Partially Occluded Scenes

Nothing is more humbling than to look with a strong magnifying glass at an insect so tiny that naked eye sees only the barest speck and to discover that nevertheless it is sculled and articulated and striped with the same care and imagination as a zebra. Apparently it does not occur to nature range of vision, and the suspicion arises that even the zebra was not designed for our benefit.

Rudolf Arnheim

In this chapter we will discuss a system for finding and separating the human hand which is holding an object. This problem is one of the very difficult computer vision problems, because of many degrees of freedom, the flexibility of the hand and the occluding problem. In order to solve the correspondence problem, we use lateral similarity, by not only comparing different features at one point but the features at that point and its neighbours, see chapter (3).

5.1 Introduction

Recognising non-rigid objects like the human hand is a very hard computer vision problem. Different methods have been presented in the literature, whose main drawbacks is low robustness or high computational load in the analysis of cluttered scenes (Wiskott and von der Malsburg, 1993). In order
to achieve immersive human-computer interaction, human body parts, e.g.,
the hand, could be considered as a natural input which motivates research
on tracking, analysing, and recognising human body movements.

As an application example, the gestures could be used to represent some
commanding inputs such as pointing, rotating, starting, stopping,... etc. One
of the important characteristics of human body parts is skin color. Many
good algorithms are based on the the skin color to detect the hand in the
scene (Wu and Huang, 2002), (Triesch, 1999). We have used the skin color
feature in our algorithm, and found that lighting and background (as in many
algorithms) play a big role in the response of the feature. The human hand
has received more attention by many researchers in the last decade. Many of
the researchers have concentrated in the hand tracking as in (Utsumi et al.,
2002), fingerprint identification as in (Pankanti et al., 2001), hand sign recog-
nition (Downton and Drouet, 1991), and hand segmentation as in (Cui and
Weng, 1996). In our system we have looked for the characteristics of the
hand and tried to find the correspondence of these characteristics by using
the Elastic Graph Matching (EGM). The rest of this chapter is organised as
follows: In the next section we discuss the object representation with Gabor
wavelets and describe the different features that we have used in our algo-
rithm. In section three we explain the generation of the algorithm described
in section two with the EGM algorithm. In section four we introduce our
results and finally discuss the advantages and disadvantages of our system.

5.2 Object Representation

As we discussed before in chapter (3), each visual pattern can be represented
by a graph containing nodes labelled with local features and links encoding
the topological relationship between features (Wiskott, 1995). Based on this
representation, the problem of pattern recognition can be formulated as one-
to-one correspondence between the nodes of an input graph and another one
in the database. A good correspondence is one that respects the topological
relationships between the node and finds high similarity between the labels
of the correspondence nodes. To achieve high correspondence, object class
has an important role in the success of the whole system. An object class
that has a common structure is the human face. EGM has been successfully
applied. In our system we represent each grip as undirected graph with nodes
positioned at the landmark of the hand like fingers tips (Poinzer et al., 1981),
other landmarks in the hand, and labelled with a compound feature vector.
The compound feature vectors consists of three different types of features.
We introduce them separately in the next section.
5.3 Multi-Feature Vector

In the last few years many articles have discussed different ways to derive local image descriptions, some of which are based on statistics such as averaging, statistical moments like (Zhao1 et al., 2002), Principal Component Analysis (PCA)(Chen et al., 2002), or Independent Component Analysis (ICA). Some analyses are based on wavelets, others are based on the advantages of the multi-features vectors and combinations of different features as in (Triesch and von der Malsburg, 2001). In this section, we describe three local image descriptions and we combine the different features to generalise the elastic graph matching algorithm in order to find the grip in the occluded view, then to copy the corresponding region and extract the hand from the view.

5.3.1 Skin-Similarity Gabor Transformed Local Image

It seems that skin color of the human hand is an important feature that helps to solve the correspondence problem. We describe an algorithm that is able to find the skin color of the hand in the scene and then we transform the skin-similarity image with Gabor wavelets. Let \((H_0, S_0)\) be the prototype of the skin color, we compute the similarity region of the prototype, then we call the result image skin-similarity local image.

To compute the similarity between pixels in the image and the prototype we use the formula:

\[
R_{\text{skin}} = \chi_t \left( 1 - \sqrt{\left( \frac{H - H_0}{\sigma_H} \right)^2 + \left( \frac{S - S_0}{\sigma_S} \right)^2} \right)
\]  

(5.1)

Where \(\sigma_H\) and \(\sigma_S\) are normalization factors and \(\chi_t\) is a threshold function parameterised with a constant \(t > 0\) and defined with the equation:

\[
\chi(x) = \begin{cases} 
1 & \text{if } x \geq 0; \\
0 & \text{if } x < 0.
\end{cases}
\]  

(5.2)

The result of the above skin classifier is displayed as a binary image in figure 5.1. As we can see from the above formula, the skin-similarity image is sensitive for objects that have a color similar to the skin color. We have transformed the image with Gabor wavelet to extract an image: in it each pixel carries a local feature vector at this position. To compute the similarity between two feature vectors we have used the amplitude phase similarity function defined in equation (3.2).
Figure 5.1: The figure shows the skin similarity image according to (5.1) and 5.2) respectively. In The left is the original image and right is the skin similar region.

5.3.2 Gabor Wavelet-Transformed Local Image

As we discussed in (2.5), we have converted the input image to the HSI color space and then we transform the intensity (gray level) image with DC-free Gabor filter equation (2.39). The resulting image has the same size as the input image and each pixel is a feature vector with size equal to the convolution of different scaled and oriented Gabor filters with the original pixels around this position. To compare the two feature vectors, we have used the $S_{abs}$ similarity function defined in equation (3.1). The result of the similarity function of one feature vector with another Gabor transformed image is displayed in figure 5.2.

5.3.3 Color-Averaged Local Image

The third feature that we have used is a local color average in the HSI color space (for transforming from RGB color space to the HSI color space), see (Gonzalez and Richard, 1992; Chamorro-Martinez et al., 1998). Two image regions are compared by their color averages. We consider regions of the 8-neighbouring to compare color regions with this formula:

$$S_{col}(J, J') = \frac{\langle \Gamma (J), \Gamma (J') \rangle}{\| \Gamma (J) \| \cdot \| \Gamma (J') \|},$$

(5.3)

where
Figure 5.2: This figure shows the result of the different descriptors that we have discussed and how each feature vector alone is good to find correspondences. $A$ is the source image with a circle around the position from which we extracted the feature vector, $A'$ is the target image in which we have tried to find the most similar feature vector to the extracted one from $A$. In $B'$ the color similarity image between the color feature vector extracted from $A$ circled position was compared with the color feature vector extracted at each pixel in $A'$, in $B$ is the most similar correspondence. In $C'$ the similarity of only Gabor feature vector extracted around the circled position in $A$ and $A'$, $C$ is the most similar position.

$$\Gamma(J) = (\tau_H H, \tau_S S, \tau_I I)^T.$$ The result of this feature has been displayed in figure 5.2.

5.4 Complex Feature Vectors

The three different feature vectors described above, are extracted at each pixel and combined into a compound entity and we call complex feature vector, see 5.4. As we can see from figure 5.2, the more feature vectors in the complex feature vector, the better correspondence match. The most straightforward way of combining the local image descriptions is to simply concatenate them, see 5.4. In order to compute the similarities between two complex feature vectors $J$ and $J'$, first we have to compute the similarity of their corresponding constituents. Then we add individual similarities with
Figure 5.3: This figure shows the rest of the different descriptors that we have discussed as in figure 5.2. In \( A' \) displayed the skin-similar Gabor similarity of the feature vector taken at circled position in 5.2, \( A \) and skin-similar Gabor feature vector in the target image \( A' \). The same procedure was performed for the compound feature vector of the three different feature vector in \( B, B' \) and in \( C, C' \) is the similarity without the skin-similar Gabor feature vector certain normalised weighting factors as in this form:

\[
S_{\text{com}}(J, J') = \sum_n \omega_n S_n(J, J'), \quad \sum \omega_n = 1,
\]

where \( n \) is an index over the different features, \( S_n \) is the similarity between different feature vectors and the \( \omega_n \) are normalised weighting factors. In figure 5.3 the result of finding correspondences with complex feature vector is displayed in \( B' \). The maximum of the distribution is a proper place as indicated by a circle in the target \( B \). Clearly, this represents a significant improvement compared to using a Gabor feature vector alone. In this example, the improvement is mainly due to the skin-similarity Gabor feature vector as can be seen from the differences between \( C, C' \) and \( B, B' \).

## 5.5 Generalization of the Algorithm

Our aim is not to solve the correspondence problem for one feature vector as we have done with the complex feature vector above, but to locate the
hand, which is partly hidden by another object, and to free the object from the hand. To use the advantages of the complex feature vector to find the hand, we have represented the hand in a certain view with a model graph. At each node of the model graph, we extract a complex feature vector and its edges to reserve the topology of the hand. In order to find the hand in another view where it may be partly occluded with another object, we have used a matching schedule as we discussed in chapter (3). The search process is achieved by scanning the model graph extracted of the hand grip view over the other scene and search for the position of highest similarity of the graph on the scene. To reserve the topology we have used the lateral similarity function as in this equation

$$S_{\text{lat}}(G, G') = \frac{1}{n} \sum_{i=1}^{n} \bar{S}_{\text{lat}}(J_i, J'_c(i)),$$

(5.5)

$$\bar{S}_{\text{lat}}(J_i, J'_c(i)) = S_{\text{com}}(J_i, J'_c(i)) + S_{\text{com}}(J_i, J'_c(i)) \sum_{k} S_{\text{com}}(J_k, J'_c(k)),$$

(5.6)

where $\bar{S}_{\text{com}}(J_i, J'_c(i))$ is the lateral excitation from the neighbours $k$. The motivation behind augmentation of the simple similarity measure with lateral excitation was to fortify the algorithm against noise and clutter. Our algorithm involves matching of the model graph with another image while only part of the model graph has a corresponding match in the other image or graph. That is, there are parts of the model graph which do not have a corresponding match in the input graph. This is similar to the task of matching two instances of an object where the two instances are partially occluded in different ways in the two scenes. These occlusions cause the total graph
similarities to drop, causing the total feature vector similarities of the desired match to potentially become comparable to those of false correspondences, which accidentally contain some high similarity nodes. In such cases, lateral excitation favours the correct match due to its contiguity of high similarity nodes, and in effect, penalises the false matches by suppressing the accidental isolated high similarity nodes. This phenomenon also relates to the Gestalt principle of continuity or nearness and is also consistent with the spirit of the dynamic link matching in that neighbouring nodes cooperate in establishing correspondences.

5.6 Experiments and Results

We have tested our system on a data base picked up from a Sony robotics firewire camera with a constant background, different lighting degrees, and complex background, see figure 5.5. In our test we have found that the system is sensitivity against background and the illumination conditions as in many different algorithms. Another condition that we have tried to control is the pose of the grip of the hand. Due to the flexibility of the hand and different characteristics of each pose of the hand, we have found that the system is tolerant to slight pose variation. The use of the lateral similarity in equation (5.5,5.6) is practically important, because the second term in equation (5.6) is the excitation received by \( S_{\text{com}}(J_i, J'_c(i)) \). As it can be seen, the amount of excitation directly depends on the value of \( S_{\text{com}}(J_i, J'_c(i)) \). The matching process become bad when the pose of the hand is great. In our experiment we have found that the matching depends on the background, lighting, and the pose of the grip. Some of our results have been shown in figure 5.6. As it can be seen, the correspondence in the first example (the first row in figure 5.6 is perfectly matched because we have tried to control the pose as well as possible, and then we have succeeded to segment the hand perfectly. In the second row in figure 5.6 the pose is different than the grip of the object, and the correspondence only successful at higher characteristic points, for example finger tips.

5.6.1 Object Free from the Hand

In order to free the object from the hand we have applied an algorithm to find the hand under the model graph and separate it in a new frame. After the separation process we get the region that represents the hand and the rest is considered to be the object.
Figure 5.5: The figure shows the different types of objects that we have used it in our experiment. In middle we display some object with complex background and in the upper and bottom we show some of objects with simple background
Figure 5.6: The left column shows the hand posture with a free graph at each node attached a complex feature vector, in the right column displays the matching graph after searching for the best correspondence of the hand posture. As you can see, the system worked successfully on the simple background.
Figure 5.7: The left column shows the hand posture with a free graph at each node attached a complex feature vector, and in the right column displays the matching graph after searching for the best correspondence of the hand posture. As you can see, the system worked successfully on complex background
Figure 5.8: In this figure we have displayed the result of the object free from the hand. In the upper row from left to right, the hand posture, the best match of the hand. In the lower row, the hand region in a new frame, and the part of the object after freeing it from the hand are shown.

In the next chapter we introduce an algorithm to combine these parts of the object. The result of the separation process is displayed in figure 5.8.

In the case of complex background, we have encountered a problem with the reconstruction of the object from its parts due to the other parts in the background. The segmentation process of the other object return other segments that is similar to the part of the object, which make the matching process fail to reconstruct the right object. For this reason, we have subtracted the background in the pick up of the images as shown in the figure 5.5. With this preprocessing the algorithm is stable and able to reconstruct the object from their parts.
Figure 5.9: In this figure we have displayed the result of the object free from the hand. In the upper from left to right, the hand posture, the best match of the hand. In the lower row, the hand region in a new frame, and the part of the object after freeing it from the hand are shown.
5.7 Discussion

Analysis of the non-rigid parts of the human body is not a trivial computer vision tasks. The main aim for our system was to find the hand which holding another objects, and then trying to separate the object from the hand by segmenting the hand. We have found that the task is not trivial because the hand has no fixed characteristics, but we have tried to extract features at the most important parts of the hand like finger tips, finger joints and other important positions. The system for the constant background, and the slightly pose gives a good results. The slighter the pose is, the better the result is. the EGM (von der Malsburg, 1988) was introduced over a decade ago as a biologically inspired method for pattern recognition. Our results demonstrated that despite its age, EGM remains a competitive algorithm, even in comparison with the modern analytically developed statistical information processing methods. EGM with lateral similarity in equation (5.6) reveals the following characteristic. Given equal similarity values across all the nodes in the graph for a given match, the nodes at the graph boundary receive less excitation from their neighbours than the nodes in the centre because they have fewer neighbours. In other words, given random similarity values, the nodes at the object boundary are weighted less than those in the centre, and thus their contribution to the total graph similarity is reduced. Of course this argument only holds on average. In cases where a boundary node is surrounded by very high similarity values, or where a centre node is surrounded by low similarity nodes, this will no longer be the case. However, this weighting scheme can be important in situations where the objects are to be detected in cluttered scences, as in ours. The feature vectors located at or near the boundary of the object are affected by the structure in the background, hence leading to low similarity with the boundary feature vector in a model graph. Lateral excitation in effect weights down the contribution from the boundary feature vectors and improves the robustness of graph matching.
Chapter 6

Reconstruction of Object from Partially Occluded Views

The more efficient computers become at inducing new knowledge, the more widely that knowledge will be applied, even in matters of life and death. It is essential that such knowledge be open to inspection. This means that designers of learning systems have a public duty to use comprehensible description languages - even if that means sacrificing performance. Otherwise we run the risk of generating truly 'unknowable knowledge.'


Occlusion is a major cause of information loss: even in moderately complicated scenes it is either impossible or impractical to obtain complete range scans (Sanchiz and Fisher, 2000). On the other hand, an exhaustive description of observed objects or environment is needed for some applications, like construction of a 3D model (Wundrich, 2004) and environment object recognition.

In this chapter we will describe an algorithm to recombine parts of an object to reconstruct a complete view of the whole object. The system we discuss here depends on the elastic graph matching that we have described in chapter (3)
6.1 Occlusion Understanding

Our world is full of occlusion. In many scenes, we are likely to find several if not hundred occluded parts reconstructing. The occluded parts of an object in a scene is one a difficult problem in computer vision and impossible to solve in one single scene (Dell’Acqua and Fisher, 2002). We describe in this section some different types of occlusion, which may arise when a scene is picked up. Lots of these are not resolvable without domain specific knowledge, or models, helping to derive structural interpretation of images. As we have mentioned in chapter (1) and in figure 1.1, this type of occlusion is very complex, because the hand hides a big part of the object under the fingers, and not reserve the boundary of the object. In order to learn or write a program to complete the object from one view is impractical, because the information about the occluded object is lost, but Fisher in (Dell’Acqua and Fisher, 2002) has proposed a procedure to fill in the gaps without performing extra scans by extending the boundary of the object without to reconstruct the information in the occluded region.

Previous research on occlusion reconstruction focused on the reconstruction of a single large area occluded by one object. In that context two cases were considered: occlusion preserving surfaces, and occlusion breaking surfaces.

**Occlusion preserving surfaces:** In this type of occlusion different image regions correspond to different objects, and the occluding region is entirely surrounded by the occluded one. For example, a book lying in the middle of a table scanned by a range sensor from above. Figure 6.1 a shows an example of these occlusions: surface $B$ occludes surface $A$.

Detection and reconstruction of these occlusions is based on searching for regions entirely contained within the boundaries of another region, so that they can then be extended across the occluding area.

**Occlusions breaking surfaces:** In this case the occluding parts breaks the occluded surface into two regions which correspond to a single surface. For instance, figure 6.1 b the surface $B$ is occluding the surface $A$ and divided $A$ to two surfaces without cutting the boundaries.

The detection and the reconstruction is based on identifying compatible regions which can then be merged in order to reconstruct the missing occluded part (Stulp et al., 2001).

In both the above cases the occluding object does not obscure the missing boundaries of the occluded one. The work of (Dell’Acqua and Fisher, 2002) explored the case when the occluding part partially obscures boundaries of an occluded object. This case is called occlusion breaking boundaries.
Figure 6.1: The figure shows that, in the above row left, surface $B$ occludes surface $A$, in the right the surface $B$ occludes surface $A$ and divide it into two surface $A$ and $A'$ without cutting to the boundaries, and in the bottom the surface $B$ obscures the surface $A$ with cutting the boundaries of $A$. 
6.2 The Description of the Problem

The problem that we will solve here is how to reconstruct the parts of the object which are hidden. The problem of occlusion in general is one of the hard problems in computer vision, for instance in segmentation and contour detection. In the previous chapter we have tried to separate two overlapping objects (in our case the hand and another object). If we have parts of an object as in figure 6.2, can we expect the whole object without seeing it before? Yes, we explain in the rest of this chapter an algorithm that is able to achieve this task. At first, each part is represented as a grid graph, and we describe in the next section the main idea.

6.3 Reconstruction from a Model Graph

As we have explained in chapter (2), the region surrounding a given pixel in the image is represented by responses of a set of Gabor filters of different frequencies and orientations, all centered at the same pixel position. This set of responses is called feature vector. As in (Lades et al., 1993; Buhmann et al., 1992; Wiskott et al., 1995; Wiskott et al., 1999), objects as well as human faces are represented by graphs (described in chapter (3) whose nodes are labelled with feature vector, and whose edges describe topographical relations.

How can we reconstruct the object from the model graph? The answer of
this question has attracted the attention of many researchers in the last few years. In the next few pages we will describe two algorithms that reconstruct the objects from the model graph labeled with Gabor responses at each node. We try to summarize the two algorithm as follows:

### 6.3.1 Reconstruction from Gabor Feature

The reconstruction of an object from a model graph labeled with responses of Gabor filters is proposed by (Pötzsch et al., 1996; Pötzsch, 1994) as follows:

1. Let $I$ be the input image.
2. Let $J_\nu$ be the Gabor Transform of $I$ at pixel $\vec{x}$ with equation (2.40)
3. Let $l, d$ be the number of scales and orientations of Gabor kernels $\psi$.

4. Let $G(V, E)$ be the model graph with vertices labeled with $l \times d$ dimensional feature vector.

5. Let $I^R$ be the reconstructed image.

6. For all nodes of the model graph

   (a) Compute $\Psi_{\nu\mu} := \langle \psi_{\nu}, \psi_{\mu} \rangle = \sum_j \bar{\psi}_{\nu}(-\vec{x}_j)\psi_{\mu}(-\vec{x}_j)$, with $\vec{x}_j$ running over all pixels.

   (b) Compute $I^R = \sum_{\nu} \sum_{\mu} (\Psi^{-1})_{\nu\mu} \bar{\psi}_{\mu}$.

The result is displayed in figure 6.3.

### 6.3.2 Reconstruction from Color Feature

The second algorithm is invented in this work, and we have used the color feature vector that we have described in the previous chapter to visualize the object. Our algorithm can be summarized in the following steps:

1. Given a model graph $G(V, E)$, where $V$ is the number of vertices and $E$ the number of edges.
Figure 6.4: The figure displays the reconstruction of image from a model graph labeled with color feature vector inside of Gabor wavelets, $A$ is the original image, $B$ is the reconstruction of the object from the color feature vector.

2. From the vertices of model graph, create the triangulation set $T = t_i(v_{i,1}, v_{i,2}, v_{i,3})$, where $v_{i,j}$ are neighbour vertices in the model graph.

3. For all triangles $t_i$:
   
   (a) compute the average value of color feature vector at its vertices $f_m$.
   
   (b) Fill the pixels inside each triangle $t_i$ with the mean value $f_m$

The result of is displayed in figure 6.4.

### 6.4 Combine the Partial Graphs

To combine the different visible parts of the object, for each part, we construct a grid graph $G^i(V^i, E^i)$, where $i = 1, \ldots, N$, for each graph node $V^i$ related a compound features vector. We compare the two graphs $G^i$ with color similarity function, and $G^j$ to find if they have a common visible parts of the object or not. If the related features similarity is greater than a threshold $g_t$

$$S_g(G^i, G^j) > g_t,$$  \hspace{1cm} (6.1)

where $g_t$ is a constant not dependent on the type of object. Then the feature vectors that locate at the same part of the object and are visible in the two parts will have a high similarity. So we compute similarity of features at each node $V^i_{n'}$ in the graph $G^i$ with features vector at node $V^j_{n'}$ in graph $G^j$ if

$$S_n(V^i_{n'}, V^j_{n'}) > n_t$$ \hspace{1cm} (6.2)
Now after we expect where may be the visible nodes in common, we would collect them in one graph. We call Fusion Graph and we refer to it as $G^{\text{new}}(V^{\text{new}}, E^{\text{new}})$. In the object that homogeneous and do not have different features at each location, it is possible to find high similarity at the same position, i.e., may be we have two feature vectors at the same position. For this reason, we copy $V^i_n$ to the new model graph $G^{\text{new}}(V^{\text{new}}, E^{\text{new}})$ such that, for all nodes in $G^{\text{new}}$

$$V^i_n \neq V^i_k \quad \forall k$$

(6.3)

The fusion graph is a collection of all visible parts of the object each at the original position, and we will try to reconstruct the object from the fusion graph with different feature vectors at each node.

### 6.5 Discussion

In this chapter, we have discussed the problem of how to deal with the parts of the object that are hidden by another object. The difficulty of the problem depends on how the occluding object hides the occluded one. As we have classified in the introduction of this chapter, the problem of occlusions preserving surfaces is different from occlusions breaking surfaces and occlusions breaking boundaries. We have concerned on the occlusions breaking boundaries which is the hardest one in the occlusion problems.
Figure 6.6: The figure shows, the same result as in figure 6.5 but for another object.

Figure 6.7: In this figure we have displayed the reconstruction of the fusion graph that is labeled with color feature vector using two ideas. In the middle we have triangle the fusion graph and we have filled each triangle with the mean value of color feature vectors at the triangulated nodes, and in the right we have filled neighbouring pixels with color feature vectors at each node.
Our algorithm depends on the elastic graph matching. The elastic graph matched has applied successfully on face and object recognitions, but the correspondence problem for nodes that located at the boundaries is hard to solve. The correspondence problem is attacked from many researchers like (Würtz, 1995) and (Pötzsch, 1994; Lourakis et al., 2004). In our system we have treated the nodes in boundaries as in (Würtz, 1995). The result that we have got from our algorithm is good to recognize the objects. We have tested the system with different objects like three dimension coloured object like cube of wood as in figure 6.5. In this example and as we have mentioned in our introduction for the problem in chapter (1) different parts of the object are occluded with other object and then we can not see the part of the object under the hand for example. Examples with different objects are displayed on the above. As a final question that we would answer it. How can we use or what are the applications of such a system in the reality? The answer to this question seems to me in the generalisation. If this algorithm can be generalised to an image sequence, we can use it in many application like follow an objects moves partially hidden by other object (cars on a highway) and many others, but for this system I think the application is only applicable on the static object views.
Chapter 7

Discussion and Outlook

The known is finite, the unknown infinite; intellectually we stand on an island in the midst of an illimitable ocean of inexplicability. Our business in every generation is to reclaim a little more land.

T.H. Huxley

7.1 Discussion

The primary goal of the visual system is to recognize objects so that we may interact with them appropriately. Although our perception of objects appears instantaneous and automatic, complex processing is required by the visual system for recognition of objects in the real world. One of the primary obstacles faced by visual system is the fact that objects occlude parts of themselves and parts of neighbouring objects which makes the information describing objects is incomplete.

In general, to deal infants or baby to represent the object in the occluded objects. We should know how object representations are affected by times of non-visibility and how infants represent the location of attended objects (spatial representation) when non-visible. We are interested in two questions related to the occlusion process. Can infants accurately represent the durations of occlusion and if so, how do these representations change over time?

Psychological studies (Gredebäck, 2004) have been performed on infants and adults to answer the above questions. The result of these studies suggest that on one hand, the young infants are able to take information of the
stimulus from the visible parts of it for a short time; on the other hand, infants extrapolate the target trajectory during occlusion. The processes that govern accurate predictions are still inadequately documented or discovered. It is still unclear to what extent infants actually extrapolate the trajectory or whether infants formulate and apply rules based on previous experience to predict the reappearance of the target.

If infants extrapolate the motion of the occluded target it is still uncertain if these are based on the tangent at disappearance or infants have the ability to produce more complex extrapolations. If rules are used to predict the reappearance of the target this cannot be done on their first trial; experience is needed to form stable representations. Infants might rely on simple rules of reappearance or form associations between pre- and post-occlusion trajectories. By applying simple rules repeated occlusion events can be accurately predicted. However, if the reappearance location changes constantly the resulting prediction will be inaccurate and based on the average reappearance location.

Associating computational models with psychophysical is notoriously difficult. The similarity between the data retrieved from biological systems and a computational model are easily over-interpreted. It still makes sense to point out some parallels between the model presented in our work and the biologically relevant findings. In chapter 2 we have describe how the spatial information are transferred in simple cell in visual cortex V1 as described with (Jones and Palmer, 1987). In chapter 3 we discuss how we can store information of a visible objects in what is called the elastic graph model. In one attempt to simulate how the brain can correlate information that can be extracted from one visible part and if our brain follow the geometrical structural theory, we have discussed in chapter 4 an viewpoint-independent algorithm that can work only for the non-real objects and structure less. The result of this algorithm had reoriented our attention to another idea of how our brain may deal with the occluded objects. As in chapter 5, we have proposed an algorithm that is able to free the object from the hand in many different situations of occlusion. In chapter 6 we have proposed the main idea how the baby may extrapolate the trajectory of a shape in the partially occluded objects by keeping the configuration of the object in the matching process. At this point we can say our baby algorithm should be developed more and more to control the real situation in our world.
7.2 Outlook

At the end of our journey and to be clear for the followers, we need to ask the final question. What is the right way to solve the complexity of the occlusion problem? We think the right way is to understand how our brain does that at first and then we can simulate it with the computer. In order to generalize the algorithm we need to develop the occluder finding algorithm (in our case the hand) and exactly separate the visible part of the occluded object, and generalize the reconstruction algorithm to solve the correspondence problem near the boundaries. This generalization has many applications in the real life. For instance, in robots football, the robot can follow the ball in partially occluded situations and, in graphics industry to enable the graphics chips to deal with occluded objects and many other situations.
Appendix A

Zusammenfassung in deutscher Sprache

The three-legged stool of understanding is held up by history, languages, and mathematics. Equipped with these three you can learn anything you want to learn. But if you lack any one of them you are just another ignorant peasant with dung on your boots

Robert A. Heinlein

A.1 Das Ziel der Dissertation

Textursynthese an den Rändern berücksichtigt.

A.2 Einleitung


In dieser Arbeit wird eine neuartige Methode vorgestellt, welche vollständige Objekte aus ihren Teilen rekonstruieren kann. Alle Teile des Objektes, die aus Bildern gewonnen werden, in denen das Objekt lediglich teilverdeckt sichtbar ist, müssen hierfür vorliegen (siehe auch Abbildung(6.5)).

A.3 Bildvorverarbeitung durch Gabor Wavelets

Ein Nachteil der Fourier-Transformation liegt darin, dass sie nur die lokalen Eigenschaften eines Signals berücksichtigt. Dies liegt darin begründet, dass die Fourier-Transformation ein Signal in ebene Wellen zerlegt, die nicht lokalisiert sind.

Grossman und Morlet veröffentlichten zwei Arbeiten (Grossmann et al., 1984; Grossmann and Morlet, 1985), in der sie eine neue Transformation
zur Frequenzanalyse von Signalen vorstellten und damit erzielte Ergebnisse diskutierten. Diese neue Transformation, die mittlerweile als Wavelet-Transformation bekannt ist, zerlegt ein Signal mittels Funktionen, die sowohl im Frequenzraum als auch im Ortsraum hinreichend lokalisiert sein können.

Bei der Wavelet Transformation wird mehr Flexibilität dadurch erreicht, dass eine fast beliebig wählbare Funktion, das Mutterwavelet, zur Analyse des Signals verschoben, rotiert und gestaucht wird. Wie es ihr Name suggeriert, lassen sich Wavelets als verallgemeinerte Schwingungen interpretieren, was sich abstrakt durch ihren verschwindenden Mittlewert ausdrückt. Der Preis für die Vielseitigkeit ist das Auftreten zusätzlicher Variablen in der Transformation, die Ort, Skalierung und Orientierung beschreiben. Die Wavelet-Transformation kann schließlich als Faltung des Signals mit der Menge aller sich in Orientierung und Skalierung unterscheidenden Wavelets verstanden werden.

In dieser Arbeit wurden Gabor-Wavelets unterschiedlicher Skalierung und Orientierung verwendet.

In (Jones and Palmer, 1987) wird gezeigt, dass die Antwort der s.g. Simple Cells im visuellen Kortex der Katze mit guter Übereinstimmung als Filterantwort eines Gabor-Wavelets verstanden werden können. Die Verwendung von Gabor-Wavelets ist somit auch aus neurophysiologischen Gesichtspunkten naheliegend.

**A.4 Elastische Graphenanpassung**


Die elastische Graphenanpassung beschreibt ein Objekt also zum einen
in Bezug auf seine visuellen Merkmale, wie sie im Graphen repräsentiert sind, und zum anderen in Bezug auf seine Transformationseigenschaften, wie sie in der Hierarchie von Abbildungen gegeben sind. Während die visuellen Merkmale relativ mühelos aus Beispielbildern gewonnen werden können, sind die Transformationseigenschaften nicht so leicht aus einem einzelnen Beispiel zu extrahieren und müssen deshalb von Hand vorgegeben werden. Da das Aufsetzen von objektspezifischen Transformationen aber detailiertes Wissen über die dreidimensionale Struktur eines Objektes verlangt, wird bei der elastischen Graphenanpassung häufig die vereinfachende Annahme gemacht, dass allgemeine geometrische Transformation wie Translation und Skalierung so dominierend sind, dass es ausreicht, nur diese explizit zu behandeln.

A.5 Erkennung von Objektbestandteilen

Das Lernen eines Objektmodells ausgehend von teilverdeckten Beispielen sowie eine darauf basierende Erkennung von teilverdeckten Objekten ist ein schwieriges, jedoch häufig anzutreffendes Problem in der Szenenanalyse. In (Biederman, 1987) wurde die sogenannte Geon-Structural-Decomposition Theorie vorgestellt und als eine mögliche Form der Mustererkennung im Gehirn diskutiert. Die Theorie erläutert drei Stadien in der Objekterkennung

1. Zerlegung des Objekts in seine Bestandteile


3. Sobald die Teile eines vollständigen Objekts und deren relative Konfiguration zueinander vorliegen, erkennen wir den Gegenstand als Muster.

Ergebnissen, die unter Verwendung der realen Objekte erzielt worden sind. Im folgenden sollen diese Ergebnisse interpretiert werden:

- Die künstlichen Objekte sind weitestgehend texturlos, was zur Folge hat, dass im wesentlichen die Kontour kodiert wird. Dies scheint eine Verbesserung des Matchings zur Folge zu haben.

- Eine verlässliche Graphenanpassung ist die Voraussetzung dafür, die Korrelationsmatrix hinreichend genau zu bestimmen. Diese wiederum ist Voraussetzung dafür, mittels eines Clustering-Verfahrens ein Objekt in seine Teilobjekte zerlegen zu können.

Im Fall der realen Objekte, die sich auch hinsichtlich ihrer Textur unterscheiden können, gestaltet sich die Lösung des Korrespondenzproblems deutlich schwieriger. Dies wirkt sich zunächst auf die Korrelationsmatrix aus, in der sich Cluster deutlich weniger etablieren, als es bei den künstlichen Objekten der Fall gewesen ist.

### A.6 Extraktion von teilverdeckten Objekten

A.7 Rekonstruktion eines Objekts aus Teilansichten

Basierend auf den unterschiedlichen Teilen eines Objekts, die durch beschrieben werden, wird nun ein vollständiges Modell erstellt. Dies geschieht dadurch, dass die Teilmodelle (Graphen) einander so zugeordnet werden, dass die Überlappung benachbarter Objektteile (Knoten) zu einer signifikanten Erhöhung der lateralen Ähnlichkeit führt. Ist dies der Fall, so wird davon ausgegangen, dass die Objektteile im Bereich des Überlappes miteinander korrespondieren.

A.8 Diskussion


List of Figures

1.1 The main problem ............................................. 9
1.2 Machine Vision Summery ........................................ 14

2.1 Fourier Transform ............................................. 22
2.2 Chirp Signal ................................................... 24
2.3 Definition of Frame ............................................. 28
2.4 Shape of Gabor function ........................................ 32
2.5 Gabor Transform ............................................... 34

3.1 Feature Vector and Face Graph .................................. 38
3.2 Matching schedule ............................................... 42
3.3 Bunch Graph ................................................... 43

4.1 Geon data base .................................................. 46
4.2 Geon Representation as Grid Graph ............................ 48
4.3 The similarity vector ............................................ 49
4.4 The correlation matrix .......................................... 50
4.5 The correlation matrix and cluster .............................. 51
4.6 The correlation matrix and cluster .............................. 53
4.7 Geon Results .................................................... 54
4.8 Real geons data .................................................. 56
4.9 Result of real geon data .......................................... 57

5.1 Skin Similar Region .............................................. 61
5.2 Detecting Correspondence with Color and Gabor Features .... 62
5.3 Detecting Correspondence with Skin-Similarity and Compound Feature Vector ........................................... 63
5.4 Combine different feature vectors ................................ 64
5.5 Different Grip data ............................................... 66
5.6 Simple finding with complex background ........................ 67
5.7 Grip find with simple background ................................ 68
5.8 Object free from the hand one ................................... 69
5.9 Another example of object free from the hand

6.1 Types of Occluding

6.2 The figure shows different parts of an object

6.3 Reconstruction from Gabor Feature

6.4 Reconstruction from Color Feature Vector

6.5 Combination of cue parts

6.6 Reconstruction of hand from its parts

6.7 Fusion of car parts
Bibliography


Index

A
acquisition . . . . . . . . . . . . . . . . . . . . . 12
admissibility condition . . . . . . . . . 28
admissible . . . . . . . . . . . . . . . . . . . . . 22
aforementioned . . . . . . . . . . . . . . . . . .9
algorithm . . . . . . . . . . . . . . . . . . . . . 74
ambiguities . . . . . . . . . . . . . . . . . . . . 37
ambiguity . . . . . . . . . . . . . . . . . . . . .37
application . . . . . . . . . . . . . . . . . . . .80
automate . . . . . . . . . . . . . . . . . . . . .12
average . . . . . . . . . . . . . . . . . . . . . . 77

B
baby . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
background . . . . . . . . . . . . . . . 13, 74
Bayesian . . . . . . . . . . . . . . . . . . . . 10
binocular imagery . . . . . . . . . . . . .11
blurred region . . . . . . . . . . . . . . . . .14
boundaries . . . . . . . . . . . . . . . .9, 74, 80
boundary variation . . . . . . . . . . . 10
brain . . . . . . . . . . . . . . . . . . . . .15, 82
brightness . . . . . . . . . . . . . . . . . . . . 20
bunch graph . . . . . . . . . . . . . . . . . . . . 42

camera . . . . . . . . . . . . . . . . . . . . . . . .12
cells . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
clique . . . . . . . . . . . . . . . . . . . . . . . . . . 51
cluster . . . . . . . . . . . . . . . . . . . . . . . .15, 48
– algorithm . . . . . . . . . . . . . . . . . . . .52
– clustering . . . . . . . . . . . . . . . . . . . 15
clutter . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
cluttered scenes . . . . . . . . . . . . . . . . . . 57
coarse grid . . . . . . . . . . . . . . . . . . . . . 40
cohere nt . . . . . . . . . . . . . . . . . . . . . . . . 9
color . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
– feature vector . . . . . . . . . . . . . . . .76
complex conjugate . . . . . . . . . . . . . . 21
complex background . . . . . . . . . . . . 68
complex feature vector . . . . . . . . . . .61
compound feature . . . . . . . . . . . . . . 77
compound feature vector . . . . . . . . . 58
computer vision . . . . . . . . . . . . . . . . 1
configuration . . . . . . . . . . . . . . . . . . . . .82
constraints . . . . . . . . . . . . . . . . . . . . 36
contour . . . . . . . . . . . . . . . . . . . . . . . . .14, 74
convolution . . . . . . . . . . . . . . . . . . . . 33, 60
correlation . . . . . . . . . . . . . . . . . . . . . . 48
– matrix . . . . . . . . . . . . . . . . . . . . . . . . .48
Correspondence . . . . . . . . . . . . . . . . . 1
correspondence . . . . . . . . . . . . . . . . . . 35, 36, 47, 54, 80

d
DC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15, 20
DC:free . . . . . . . . . . . . . . . . . . . . . . . . . 33
descriptor . . . . . . . . . . . . . . . . . . . . . . . 14
descriptors . . . . . . . . . . . . . . . . . . . . . .9, 61
DFT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
digital signal . . . . . . . . . . . . . . . . . . . . . 17
dilation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24, 27
discontinuities . . . . . . . . . . . . . . . . . . . .9, 17
disparities . . . . . . . . . . . . . . . . . . . . . . . .36
distribution . . . . . . . . . . . . . . . . . . . . . . 37
DLM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
dual frame . . . . . . . . . . . . . . . . . . . . . . . . 26
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>viewpoint</td>
<td>14</td>
</tr>
<tr>
<td>viewpoint invariant</td>
<td>55</td>
</tr>
<tr>
<td>visible</td>
<td></td>
</tr>
<tr>
<td>- nodes</td>
<td>78</td>
</tr>
<tr>
<td>visual cortex</td>
<td>11</td>
</tr>
</tbody>
</table>

**W**

<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavelet</td>
<td>8</td>
</tr>
<tr>
<td>wavelets</td>
<td>14, 17</td>
</tr>
<tr>
<td>- coefficients</td>
<td>24</td>
</tr>
<tr>
<td>- family</td>
<td>26</td>
</tr>
<tr>
<td>- parameters</td>
<td>29</td>
</tr>
<tr>
<td>WFT</td>
<td>21, 22</td>
</tr>
</tbody>
</table>
Lebenslauf

Persönliche Daten

Name: Roushdi Mohamed Farouk Abdelhamid
Geburtsdatum: 20. Oktober 1970
Geburtsort: Sharkia, Ägypten
Adresse: Mahata str.5, Dandit, Mit Gahmr, Dakahlia, Ägypten
Telefon: 002/0506010032 oder 002/0552702228
Email: roushdi7@neuroinformatik.rub.de oder rmfarouk@yahoo.com

Schulbildung

09/1977-06/1984: Shararahschule in Sharkia
09/1984-06/1990: Al-Hausuniagymnasium in Sharkia
09/1990: Abitur

Studium

03/1996-02/1997: Postgraduiertenkolleg
02/00: Magisterprüfung
Preise/Auszeichnung

05/1994 der Preis des ersten Kursteilnehmers in der Wissenschaftlichen Fähigkeit, Abteilung für Mathematik, Universität Zagazig, Ägypten

1996 der Preis für einen der besten 10 Kursteilnehmer in der Universität Zagazig

Beruflicher Werdegang


seit 11/2001 Stependiat an lehrstuhl von Prof. Dr. Christoph von der Malsburg am Institut für Neuroinformatik der Ruhr-Universität Bochum

Veröffentlichungen