

A Biologically Plausible Model for Corner-based Object Recognition from Color Images

Tino Lourens

Key words: corner detection / color vision / graph matching.

© Shaker 1998

Alle rechten voorbehouden. Niets van deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm, zonder schriftelijke toestemming van de uitgever.

ISBN 90-423-0034-5

RIJKSUNIVERSITEIT GRONINGEN

A Biologically Plausible Model for Corner-based Object Recognition from Color Images

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus Dr. F. van der Woude
in het openbaar te verdedigen op
maandag 30 maart 1998
des namiddags te 4.15 uur

door

Tiete Lourens

geboren op 2 maart 1970
te Ee

Promotor: Prof. Dr. G. R. Renardel de Lavalette
Referent: Dr. R. P. Würtz

Contents

Acknowledgments	ix
1 Introduction	1
1.1 Outline of this thesis	3
1.2 Contributions	5
1.3 List of symbols	6
2 Visual Pathway	9
2.1 Retina and lateral geniculate nucleus	10
2.2 Visual cortex	12
2.3 Building block structure	14
3 Mathematical Models of Different Cells in Early Vision	17
3.1 Relation between receptive field center size and dendritic tree of a ganglion cell	18
3.1.1 Receptive field sizes of monkey M and P cells	19
3.2 Center-surround cells	20
3.2.1 Receptive field profiles of center-surround cells	20
3.2.2 Responses of center-surround cells	21
3.3 Simple cells	22
3.3.1 Receptive field profiles of simple cells	22
3.3.2 Responses of simple cells	26
3.4 Complex cells	27
3.4.1 Responses of complex cells	27
3.4.2 Additional complex cell based operators	28
3.5 Selection of scales in a multi-scale representation	29
3.6 Mapping the visual field onto the cortex	31
3.6.1 Magnification between cortex and visual field	31
3.6.2 Determining the relation between eccentricity and cortex radius . .	34
3.6.3 Eccentricity dependent receptive fields	35
3.6.4 Results of other operators	40
3.7 Summary and discussion	40
4 Localization Properties of Cortical Corner Detectors	43
4.1 End-stopped cells	45

4.1.1	Responses of end-stopped cells	45
4.1.2	False response elimination	49
4.2	Relational position invariance and rotation robustness of end-stopped cells .	53
4.2.1	Changing the preferred orientation of simple cells	53
4.2.2	Rotation of input image	55
4.3	Position of strongest response of ES-cells with different receptive field sizes	57
4.4	Related work	61
4.5	Summary and discussion	67
5	Improving Corner Detection using Multiple Scales	71
5.1	Related work	73
5.2	Responses at different corners	73
5.2.1	Convex angles	73
5.2.2	Junctions	75
5.2.3	Rounded corners	76
5.2.4	Conflicting corners	77
5.3	Detected corners at a single scale	78
5.3.1	Potential corner features and marked corners	79
5.3.2	Comparison with other corner operators	81
5.4	Combining scales	84
5.4.1	Equidistant versus scale-dependent scale sampling	87
5.4.2	Classifying a corner by its scale signature	87
5.4.3	Combining scales with different operators	88
5.5	Summary and discussion	90
6	Graph Extraction from Images	95
6.1	Segment extraction algorithm	96
6.2	Segment extraction	98
6.2.1	Orientation selection	98
6.2.2	Following a segment	100
6.2.3	Stop criteria	100
6.2.4	Optimizations	102
6.3	Results	103
6.4	Related work	104
6.5	Summary	106
7	Inclusion of Color Opponent Cells in the Model	107
7.1	Color opponent cells in the visual system	108
7.2	Representation of color	110
7.2.1	Matrix transformations	111
7.3	Opponent cells	112
7.4	Reconstruction of color using three channels	115
7.5	Double opponent color sensitive and orientation selective cells	117
7.5.1	Graph extraction for multiple channels	119

7.6	Summary and discussion	122
8	Object Recognition using Graph Matching	123
8.1	Graph attributes	125
8.2	Matching algorithm	127
8.3	Results	129
8.3.1	Complexity analysis of a case study	132
8.4	Related work and discussion	136
8.5	Summary	137
9	Concluding Remarks	139
9.1	Summary and conclusions	139
9.2	Future research	140
A	Properties of Gaussian and Mexican-hat Functions	141
A.1	Properties of Gaussians	141
A.1.1	Cut-off radius of a Gaussian	142
A.2	Properties of the Mexican-hat function	142
A.2.1	The cut-off radii of the Mexican-hat function	142
A.2.2	Normalization of the Mexican-hat function	143
A.3	The Mexican-hat function using the difference of two Gaussians	144
A.3.1	One-dimensional Mexican-hat function based on DOGs	145
A.3.2	Controlling the difference of two Gaussians	145
A.3.3	Two-dimensional Mexican-hat function based on DOGs	147
A.3.4	Minima of the Mexican-hat function based on DOGs	148
	Bibliography	149
	Index	161
	Samenvatting	165

Acknowledgments

As a Christian I would first thank God for the talents he gave me. Secondly I thank my wife Emilia Ivanova Barakova for her love, patience, encouragement, and understanding.

I want to thank all the people who contributed to the completion of this thesis. In particular the following people:

The members of the Ph.D.-committee, Prof. Dr. Reinhard Eckhorn, Prof. Dr. Frans Groen, and Prof. Dr. Doekele Stavenga for carefully reading the manuscript and their useful comments.

Promotor Prof. Dr. Gerard Renardel de Lavalette for giving me the opportunity to finalize this thesis.

Referent Dr. Rolf Würtz, my scientific advisor for the last two years, for gently guiding my work and carefully reading early and final versions of this thesis. My enthusiasm and ideas were always nicely criticized and supplied by useful hints. In you I did not only find a good scientific partner but also a very good friend. During our visits we got demonstrations of the percept-ask-recognize learning principle by your son. Felix, thanks.

Dr. Hans van Hateren, Prof. Dr. Wim Hesselink, Dr. Jos Roerdink, and Dr. Gert Vegter for reading parts of this thesis and giving useful comments.

Thanks also to Prof. Dr. Nikolay Petkov Turkedijev, whose attitude definitely improved my self criticism about this thesis.

Finally, I am indebted to my parents and brothers for their support.

1

Introduction

And God said, "Let there be light," and there was light. God saw that the light was good, and he separated the light from the darkness.

Genesis 1:3-4 (NIV)

THE RECOGNITION OF OBJECTS is a task carried out with such ease by a human that we hardly consider it as a difficult. A closer view reveals the problems behind this act of perception. A distribution of light intensities on the retinas in our eyes is processed by our brain. This distribution of light intensities can change considerably, for example if the lighting changes. Even when the object is translated, scaled, rotated, partly occluded, placed in a different environment, or a combination of them, the object is still recognized as being the same. Since all these actions cause a change in the distribution of light intensities in the eye, the changes must be compensated for in a later stadium, i.e. in our brain. Thus there is a huge number of distributions which are all perceived as the same object.

Another aspect is that concrete unknown objects (like my friend's new car) are recognized to belong to a known abstract class of objects (here cars). It is likely that specific areas in the brain are responsible for an equivalence class. The representation of such a class must be compact, for otherwise it is impossible to store enough classes in the brain. Not all possible views have to be stored, because the brain probably has an active process that compares the actual scene with a stored scene, a process which is called *matching*.

Another complication is the representation of classes. Depending on the environment, the result of recognition can be quite different dependent on the task, goal, or interest of a person. For instance, when a person is interested in details of one or a few objects it is likely that a class containing details of this object is accessed or activated. In completely different environments where one can expect the same object but also any arbitrary other object, it is not likely that the class containing details of this single object is accessed but more likely that other classes will be accessed. Cognition is capable of rapidly moving through various different classes and choosing the class which is most appropriate for the current task. Cognition is complex because contents of classes change with time and the content of a class is unique for every individual since it is dependent on the environment and the interest of the

individual.

The goal of this thesis is to develop an artificial vision model which is able to recognize a limited class of objects and is inspired by biological models. The interest in developing such a model is twofold. Firstly, we want to find robust algorithms for technical systems, and secondly, we want to provide a computational model for human vision. We do not strive for a completely faithful simulation of biological perception but for a model which is robust to translation, rotation, scaling, and small distortions. The model has a compact representation of objects and uses a matching process to recognize objects.

Let us give the similarities and differences between artificial and natural visual systems. Since inputs of both systems are similar, and the outputs of the artificial system should be similar to the outputs of the natural system, both must deal with comparable (but not necessarily identical) image processing problems. The natural systems of mammals have been developed under the needs for survival. Therefore they can adequately deal with a broad range of problems. Artificial visual systems on the other hand are often specialized in one or a few aspects of vision.

In this thesis, we will create an object recognition model which uses principles discovered in natural visual systems. Although characteristics of many cells in the eyes and brains of different species are known, it is often not clear what role these cells play. Physical limits of organic material differ from those of the hardware of a machine. It follows that exact modeling of a natural visual system is not always desirable and also impossible when we are aiming at a working recognition system. The organic material in a natural system has a limited lifetime. If cells die their function is replaced by other cells. Therefore a natural system is dynamic, in contrast with most artificial systems which are static. Our goal is to model the response of different cells and not their exact physical and chemical behavior. The brain works highly parallel and is currently faster than any parallel machine. Implementations of models based on natural vision will be partly carried out on a serial machine and partly on a parallel machine with very few processors (32 in our case) and can be relatively slow due to limitations of these machines.

A human obtains knowledge by learning. During this process general representations of different classes are created and adjusted somewhere in the brain. As we mentioned before, the brain probably stores objects in a compact way. Since edges, corners, colors, and movements play an important role in biological vision systems, it seems likely that they form the cornerstones of these compact representations. In the model developed here we assume that compact representations contain edges and corners only. We do not model the learning aspects of recognition but assume that the system has been taught a limited set of objects. The model can be extended so that it learns to recognize objects by presenting the model images which contain a single object.

An object is represented as a set of connected corners and edges. We choose this representation because it is compact and invariant under translation, scale, and rotation. The disadvantage of such a representation: we cannot describe a unique model for certain objects, e.g. round objects, like a ball. Although a round object contains an edge, it is not connected by one or more corners, which are the key for rotation and scale invariance. Object detection therefore will be limited to objects in which all edges are spanned by corners.

Basically the model receives as input a two-dimensional image which represents a two-dimensional pattern of light intensities projected at the retina. Image processing techniques, which mimic (up to a certain extent) parts of a natural visual system, are applied to enhance corners and edges in the input image. An additional step is taken to extract these corners and edges to create an abstract representation. We choose a graph representation where corners and edges are represented as vertices and edges, respectively. Finally, learned objects are located (matched) in the graph.

In this thesis we create a model that is able to find and recognize certain known objects in a color image. Since we (man) can very accurately complete the task of perception followed by recognition, the model will be partly based on existing neurophysiological knowledge of the behavior of cells in the retina, the lateral geniculate nucleus, and the primary visual cortex. The functionality of one or more cell types of the visual system has been used before in models of image processing. Most of these models contain operators that model the response behavior of one cell type, and in the present context we are doing the same.

It is known that in the biological visual pathway so-called complex cells have strong responses at edges and end-stopped cells have strong responses at corners. Responses of these cells are modeled by operators that are a composition of linear operators (filtering) with simple nonlinear ones (e.g. thresholding). Models for end-stopped cells are rare and we use one of the few good models for these cells. We discuss the properties and shortcomings of this model and extend it by using multiple scales.

Vision does not end with edge and corner enhancement. By looking at a scene we are able to recognize objects in the scene. This suggests that there is a kind of matching process acting in the brain. In neurophysiology, very little is known about the temporal and parietal areas (see Figure 2.5) and beyond. This means that when detection and recognition are involved one must speculate how the brain might work. Since we use edges and corners, graph matching would be a logical step. Before graph matching can take place, corners need to be extracted from the “enhanced corner image”. These corners can then be used for the extraction of edges from the “enhanced edge image”, where the extracted corners and edges represent the vertices and edges of the graph. Finally, a graph matching algorithm will be applied to find potentially inexact copies of different objects (also represented as a graph) in the image graph. It must be emphasized that these extraction and matching methods are purely hypothetical and not based on any biological evidence.

1.1 Outline of this thesis

The flow of visual information via the M or magno-cellular (from *magnus* = large) pathway is basically used to order the chapters in the thesis, with one exception: Chapter 7 is about color which belongs to the P or parvo-cellular (from *parvus* = small) pathway.

We know that our visual system detects and locates corners and edges accurately. In Chapter 2 we take a closer view of the main flow of visual information from eye to primary visual cortex. This information flow is known as *early vision*.

Chapter 3 gives a general overview of the most widely used models for different cells in early vision. The modeled properties of most of these cells form the basis for our approach of

artificial vision. We model the so-called center-surround, simple, and complex cell types. In the visual system these cell types can differ in their spatial (and temporal) resolution. Some cells only respond to very small parts of the visual field and are involved with highly detailed vision, while other cells respond to large parts of the visual field. These cells interact at different levels of accuracy (scales). Interaction at different scales raises the questions which scales are useful and how these scales can be ordered. In natural vision systems the spatial accuracy drops with eccentricity, but in most static artificial systems accuracy is uniform. We do pay attention to this phenomenon, however, for the sake of completeness. The decrease in spatial accuracy with eccentricity highly reduces the amount of information. A reduction of information is important since it makes the system relatively fast when motion is involved. Motion is not included in this thesis but it will be added to the model in future research.

In Chapter 4 a corner operator based on responses of cortical end-stopped cells proposed by Heitger et al. [71] will be modeled (with some minor but important changes) and compared to six standard corner enhancing operators. We are aiming at a robust operator with respect to the position under different conditions (e.g. rotation of the image), since it forms the basis for the graph. Rotation and position at different scales are analyzed to determine the robustness of the operator.

Chapter 5 improves the end-stopped operator by using multiple scales. This is necessary since the operator is noise sensitive at small scales and does not respond to all different corners at a single scale. The response of the operator will be examined for different scales at convex corners, rounded corners, and several junctions. We discuss the choice for a proper operator to combine multiple scales and motivate the choice for the number of scales. Finally the multiple scale corner operator will be compared with standard operators at both single and multiple scales.

Chapter 6 describes a line-segment extraction algorithm where the corners obtained with the end-stopped operator at multiple scales are used together with the “edge enhanced image”. Edges between a pair of corner points will be extracted only, hence edges without two corner points are not detected. The content of this chapter is meant as an intermediate step towards a graph representation and should be regarded as preprocessing for graph matching (see Chapter 8).

Chapter 7 extends the model with color. We use two “color-opponent” channels which are found in natural color vision. In previous chapters we gave a model which is based on achromatic vision. In this chapter we use this model but apply it to two different opponent channels. Hence models for biologically plausible color-opponent cells are proposed: one opponent cell type which responds to edges of a preferred orientation and a type which responds to corners. In natural vision two opponent color channels are found; in combination with the achromatic channel, every color can be reconstructed. The three channels are combined to yield the final edge and corner detection model.

Chapter 8 gives a graph matching algorithm which searches copies of different known objects in the input graph. The representation of these objects accomplishes scale, rotation, and translation invariance. The matching algorithm is based on a standard back-tracking algorithm which is a time consuming (NP-hard) problem. Hence angle and length ratio attributes which are found in every two-dimensional graph are added to speed up the search.

Appendix A describes some functions used in linear filtering, such as the Gaussian, Laplacian of a Gaussian, and the difference of Gaussians. This appendix aims at the reader who is interested in the differences and properties of these functions.

Parts of Chapter 3 and Appendix A have been published in [120, 121], parts of Chapters 4, 5, and 7 in [198], and parts of Chapters 6 and 8 in [122].

1.2 Contributions

In this section we give the contributions of this thesis.

1. An overview of early vision from a computational point of view is given. With this overview an artificial vision system based on line and corner enhancement can be constructed (Chapter 3).
2. The corner detecting qualities of the model of end-stopped cells, proposed by Heitger et al. [71], are assessed (Chapter 4).
3. We propose a new corner detector by a multi-scale combination of the modeled end-stopped cells (Chapter 5), which yields:
 - a physiological model for the percept of a corner and
 - a useful corner operator for computer vision.
4. Edge, and corner enhancement algorithms are generalized to color channels. We use the properties of the complex and end-stopped cells and assume that these cells respond excitatory to one color and inhibitory to another color (Chapter 6).
5. We develop a line detection algorithm, based on the assumption that corners are more stable than lines, and use it to extract line-segments by following enhanced edges from one corner to another (Chapter 7).
6. We develop an attributed graph format for views of objects, which is suited for objects in which all edges are spanned by corners.
7. A graph matching algorithm will be used for object recognition (Chapter 8), where the choice of attributes leads to:
 - invariance under translation, rotation and scaling,
 - robustness under small perspective changes and undetected lines, and
 - reduction of evaluations from $N!$ to less than N^3 .

Points 4-6 apply also to physiologically motivated color channels and complete color images. This is still done rarely in computer vision.

1.3 List of symbols

Function or operator $X_{\sigma,\theta}$ denotes that X is used at a single scale σ and single orientation θ .
 Function or operator X_{σ} denotes the use of a single scale σ (and multiple orientations).

The most important variables:

σ	scale.
θ	orientation.
T	threshold.
ω	$\in \Omega = \{\sigma, \text{avg}, \text{all}, \dots\}$.

Functions used to model receptive field profiles:¹

M_{σ}	Difference of Gaussians or a Mexican-hat function representing center-surround cell receptive field profile.
$\Re G_{\sigma,\theta}$	Real part of a Gabor function.
$\Im G_{\sigma,\theta}$	Imaginary part of a Gabor function.
$\Re \hat{G}_{\sigma,\theta}$	Real part of a normalized Gabor function representing the symmetrical simple cell receptive field profile.
$\Im \hat{G}_{\sigma,\theta}$	Imaginary part of a normalized Gabor function representing the anti-symmetrical simple cell receptive field profile.
G_{σ}	Gaussian function representing center-only cell receptive field profile.

Operators to model the response of cells:

M_{σ}	M or center-surround operator representing the response of a center-surround cell.
N_{σ}	N or magno-operator representing the response of a ganglion cell (M type).
P_{σ}	P or parvo-operator representing the response of a ganglion cell (P type).
$S_{\sigma,\theta}^s$	Ss or symmetrical simple cell operator representing the response of the symmetrical simple cell.
$S_{\sigma,\theta}^a$	Sa or anti-symmetrical simple cell operator representing the response of the anti-symmetrical simple cell.
$C_{\sigma,\theta}$	C or complex cell operator representing the response of the complex cell.
C_{σ}	C operator (at multiple orientations).
O_{σ}	O or principal orientation operator.
O	O or principal orientation operator (for multiple scales).
L_{θ}^R	L or eccentricity dependent operator.
$E_{\sigma,\theta}^s$	Es or single end-stopped operator representing the response of a single end-stopped cell.
$E_{\sigma,\theta}^d$	Ed or double end-stopped operator representing the response of a double end-stopped cell.
I_{σ}^t	It or tangential inhibition operator.

¹The term receptive field will be introduced in Chapter 2.

I_{σ}^r	Ir or radial inhibition operator.
E_{σ}^s	Single end-stopped operator.
E_{σ}^{dl}	Double end-stopped operator.
E_{σ}	End-stopped operator.
G_{σ}	G or center-only operator.
$PCF_{T,\omega}$	Potential corner feature operator.
$MC_{T,\omega}$	Marked corner operator.
E_{avg}	E operator for multiple scales by averaging.
C_{avg}	C operator for multiple scales by averaging.

Operators to model the response of color opponent cell types, where e and i denote the excitatory and inhibitory color, respectively:

$CS_{\sigma}^{e,i}$	CS or center-surround opponent operator.
$SC_{\sigma}^{e,i}$	SC or surround-center opponent operator.
$CO_{\sigma}^{e,i}$	CO or center-only opponent operator.
$DO_{\sigma}^{e,i}$	DO or double opponent center-surround operator.
$DO_{\sigma,\theta}^{e,i} S_{\sigma,\theta}^s$	DOSs or double opponent symmetrical simple operator.
$DO_{\sigma,\theta}^{e,i} S_{\sigma,\theta}^a$	DOSa or double opponent anti-symmetrical simple operator.
$DO_{\sigma,\theta}^{e,i} C_{\sigma,\theta}$	DOC double opponent complex operator.
$DO_{\sigma}^{e,i} E_{\sigma}$	DOE or double opponent end-stopped operator.
$DO_{\sigma}^{e,i} E_{\text{avg}}$	DOE or double opponent end-stopped operator for multiple scales by averaging.
$DO_{\sigma}^{e,i} E_{\text{all}}$	DOE or double opponent end-stopped operator for multiple scales, orientations, and color channels.

Notations:

$[x]^{\geq 0}$	$= \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{elsewhere} \end{cases}$
$[x]^{\leq 0}$	$= \begin{cases} 0 & \text{if } x \geq 0 \\ -x & \text{elsewhere} \end{cases}$
$[\vec{x}]$	is the truncation to the nearest integer for every element of \vec{x} .

2

Visual Pathway

And God said, "Let the land produce living creatures according to their kinds: livestock, creatures that move along the ground, and wild animals, each according to its kind." And it was so. God made the wild animals according to their kinds, the livestock according to their kinds, and all the creatures that move along the ground according to their kinds. And God saw that it was good. Then God said, "Let us make man in our image, in our likeness, and let them rule over the fish of the sea and the birds of the air, over the livestock, over all the earth, and over all the creatures that move along the ground." So God created man in his own image, in the image of God he created him; male and female he created them.

Genesis 1:24-27 (NIV)

THE MAIN OBJECTIVE OF THIS CHAPTER is to present some basic knowledge about the elements and pathways in natural visual systems. The visual system of mammals processes different attributes of a visual scene, such as color, form, and motion, in parallel by parallel but interacting channels (DeYoe and van Essen [31], Zeki and Shipp [201], Livingstone and Hubel [119], and Fukushima and Kikuchi [52]). Each channel has a specialized functionality, but physically it is not clearly segregated from the other channels.

The image of the outside world that is projected on the retina provides the eye with information that initiates signals that lead to perception. The sequence of steps by which the neuronal signals are evoked, transmitted, and combined to produce a scene, can be followed step by step through the retina via the optic nerve, chiasm, and optic tract to the first relay station, (the lateral geniculate nucleus, LGN), on the way to the primary visual cortex. This is illustrated in Figure 2.1. In all three structures (retina, LGN, and cortex) the cells are arranged in layers, and individual types of neurons can be identified by their anatomical and physiological characteristics.

A single cell has one or more input channels (dendrites) and one output channel (axon). The response of such channels can be measured as impulses per second. The response of the cell is either excitatory or inhibitory. The output channel of a single cell is active when it has an impulse rate between 100-200 per second (Hubel [89]).

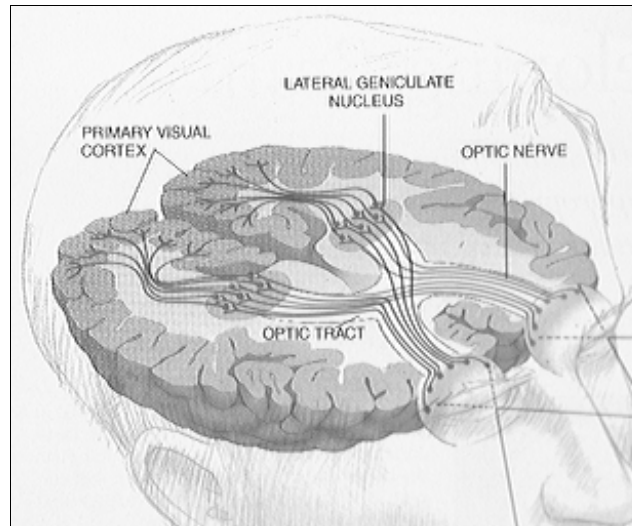


Figure 2.1: Outline of the visual pathways (Scien. Am.). The right side of each retina (found at the backside of the eye) projects to the right lateral geniculate nucleus, and the right visual cortex receives information exclusively from the left half of the visual field. The inputs from each eye end in separate layers of the lateral geniculate nucleus. From the lateral geniculate nucleus the information proceeds to the primary visual cortex at the back of the head.

2.1 Retina and lateral geniculate nucleus

The neuronal responses start at receptors known as *rods* and *cones*. Rods are responsible for night vision and contain molecules of the visual pigment rhodopsin that absorb light most effectively in the short and intermediate wavelength range of the spectrum. Absorption of light results in a generation of electrical signals.

Color and daylight vision depends on the cones. Three cone types, known as short wavelength (blue), intermediate wavelength (green), and long wavelength (red) sensitive, contain pigments that differ in molecular composition of the protein. In rods and cones light breaks down the visual pigment molecules. This releases energy leading to electrical signals at the synapses. It would lead too far to go to all details. For the reader who is interested in the underlying chemistry we refer to works of e.g. Baylor [11, 12], Dowling [34], Hubel [89], or Nicholls et al. [141].

The receptors are the first stage in the visual sensory pathway. Axons from the receptors make contacts with a second array of cells, called the *bipolar* cells. These cells contact a third array of cells called the *ganglion* cells. Two more cell types are found in the retina, the *horizontal* cells, which link the receptors and bipolar cells by relatively long connections which are parallel to the retinal layers, and the *amacrine* cells, which link bipolar and retinal ganglion cells in a similar way (Figure 2.2). The information flow in the retina can be subdivided into two paths: a direct path, from the receptors to bipolar cells to ganglion cells, and

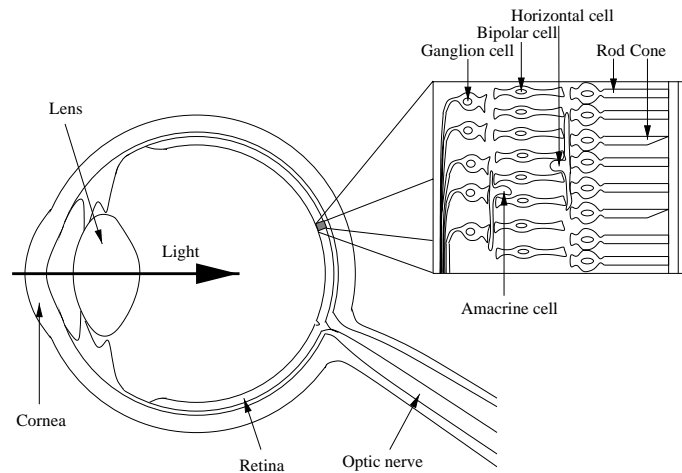


Figure 2.2: *The eye (Hubel [89]). The enlarged retina shows the relative positions of the retinal layers. Note that light has to pass all cells before it reaches the rods and cones.*

an indirect path, in which horizontal and amacrine cells are involved too.

The retina is a two-dimensional layer of about a quarter millimeter thick. The number of rods and cones varies across the retina. In the center we only have cones (fovea) and see fine detail. With increasing eccentricity the retina becomes thinner and visual accuracy becomes less (van Buren [18] and Fischer [45]). A simple experiment reveals that our visual system loses spatial accuracy quickly: if we keep our eyes fixed at a certain character from the current line it is apparent that we see a few characters sharp on the left and right side of this character. Hence we have to move our eyes from left to right to read this text (Lettvin [109]).

In 1953, Kuffler [101] and Barlow [9] were the first to record the activity from the output channels (axons) of retinal ganglion cells of cats. These cells are grouped in two main categories, which are denoted as M and P type¹ (Kaplan et al. [96], Shapley et al. [171], and Livingstone and Hubel [118]). The criteria for these groups are both anatomical and physiological. The P ganglion cells project to the four dorsal layers (parvo-cellular layers) of smaller cells in the lateral geniculate nucleus and the M cells project to the larger cells in the two ventral layers (magno-cellular layers) in the lateral geniculate nucleus. In the cat, which mainly lacks color vision, the classification is different. The ganglion cells are divided in X, Y, and W groups (Enroth-Cugell and Robson [38]).

The optic nerve is a bundle of axons from the ganglion cells, which carries the entire output of the retina. The outputs of each eye terminate on cells in the left and right LGN. In the monkey the LGN contains six layers (Figure 2.3), but the LGN of the cat has only three. Most of the cells in the LGN send axons to the cerebral cortex. They do not only receive input from the optic nerve fibers but also back from the cerebral cortex and from the brain-stem. To

¹In the terminology of Perry et al. [148], M or magno (large) cells are called P α for primate alpha, while P or parvo (small) cells are called P β for primate beta.

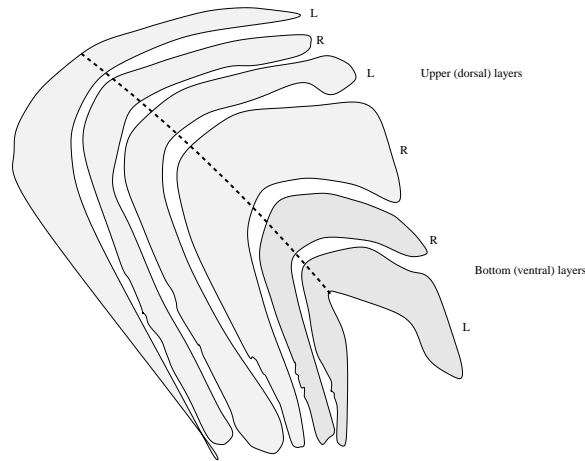


Figure 2.3: The left lateral geniculate (“bent like a knee”) nucleus of the monkey (Hubel [89]). In the four parvo-cellular layers the cells are smaller than in the two magno-cellular layers. Cells in the LGN receive input either from the left eye or from the right eye. The cells are segregated in alternating layers, denoted by L or R for input from left or right eye, respectively. Movement along a layer implies movement in the visual scene. A perpendicular movement to the layers (dotted line) implies a switch between eyes but no movement in the visual scene.

keep our model simple we will only consider the feed-forward stream of visual information: from retina to LGN to cortex.

2.2 Visual cortex

From the LGN the visual information proceeds to the primary visual cortex (also called striate cortex, area 17, or visual area V1), which is part of the cerebral cortex. The striate cortex is divided into six layers. Most of the axons of the LGN terminate in layer 4C (Hubel and Wiesel [87]). These cells are of the center-surround type and are strictly monocular. At subsequent stages most of the cells show orientation specificity and about half of them is binocular. In visual area V1 (primary visual cortex), four different cell types are distinguished: the center-surround, the simple, complex, and hyper-complex (end-stopped) cells. The simple cells are found mainly in layers 3, 4A, 4B, and 4C α (Hubel and Wiesel [82]). Complex and hyper-complex cells are found in layers 2, 3, 5, and 6 (Hubel and Wiesel [86]). A sketch of the main primary visual pathway is given in Figure 2.4.

From the retina via the LGN to V1 of monkeys there are two main streams called, the *M pathway* and *P pathway*. The M layers of the LGN are connected to layer 4B of V1 via layer 4C α (Blasdel et al. [15]). The outputs of the P layers of the LGN are relayed to the so-called *blob* and *inter-blob* columns which are found in layers 2 and 3, where the blobs are separated from each other by the inter-blob regions (Hubel [89]). Within the blobs, cells show color selectivity but lack orientation selectivity (Hubel and Livingstone [89, 117]). Inter-blob cells

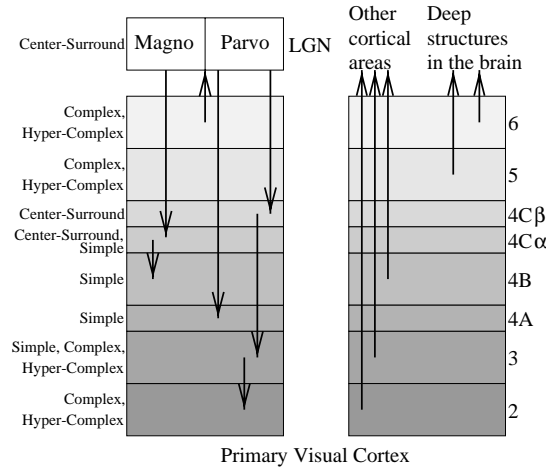


Figure 2.4: A schematic overview of the main visual pathway in the primary visual cortex (Hubel [89]). The primary visual cortex is subdivided in six layers. The main connections are made by axons from the LGN to the primary visual cortex and from the primary visual cortex to other brain regions.

are both color and orientation selective (Gouras and Krüger [58] and Thorell et al. [183]).

In Visual area V2 (also called area 18) alternating thick and thin stripes are found, separated from each other by inter-stripes (Livingstone and Hubel [119] and Zeki and Shipp [201]). The P pathway splits the two upper layers of V1 into one pathway concerned primarily with color and another primarily with form (Livingstone and Hubel [117]). Both pathways lead to V4, which is specialized for color and form (Zeki [204]). The M pathway terminates in layer 4B of V1, where at least two new categories of orientation and direction selectivity are generated which project to the thick stripes of V2. The thick stripes are characterized by a concentration of directionally selective cells and an absence of wavelength selectivity (Hubel and Livingstone [80]). The main feature of V3, which gets inputs from V1 and V2, is orientation selectivity whereas most cells in V5 (MT or middle temporal area) are directionally selective (Zeki [204]).

Orientation selectivity is generated in both major pathways, so it seems that both are involved with form perception: the P pathway through its inputs to V4 and the M pathway through V3. Most cells in V4 are (unlike cells in V3) also color selective, so there is a relationship between color and form processing in V4.

The M cells are more concerned with the form of moving objects and generating structure from motion, whereas the P pathway is more concerned with a detailed static (or slowly moving) image that contains both color and luminance information.

Figure 2.5 yields a view on visual areas V1 to V5. A subdivision of V1 is made into layers 2, 3, 4B, and 4C. Area V2 is subdivided into thin stripes and inter stripes (which outputs form the input for area V4) and thick stripes (which connects areas V3 and V5). There is evidence for direct connections between V3, V4, and V5 (Zeki and Shipp [201]). For more detailed information about areas V1 to V5 we refer to Zeki [204].

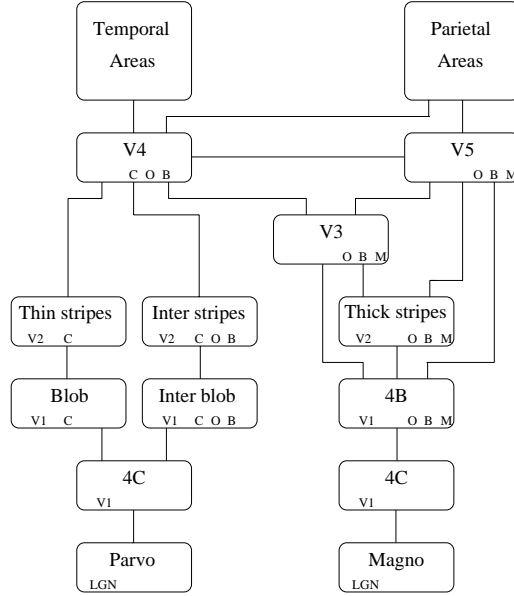


Figure 2.5: A diagram of the visual system (DeYoe and van Essen [31] and Fukushima and Kikuchi [52]). The symbols C, O, B, M are used for Color, Orientation, Binocular disparity, and Motion sensitivity, respectively.

2.3 Building block structure

In the past forty years much work on visual sensory areas of mammals (mostly cats and monkeys) has been done by neuro-physiologists. The functionality of groups of cells in different sensory areas are now reasonably well understood.

The technique of illuminating selected areas of the retina introduced the concept of the receptive field, a concept that provided a key for understanding the significance of the signals in the brain. The term receptive field was coined by Sherrington in relation to reflex actions and was reintroduced by Hartline [65]. The *receptive field* of a cell can be defined as the area of the visual field (or the area on the retina) which is able to influence the activity of cell by light. The *receptive field profile* of a cell, which Hubel [89] calls “mapping out a cell’s receptive field”, is the sensitivity profile of the cell which is modeled by a spatial filter function.

Kuffler [101] and Barlow [9] were the first to record the activity from the axons of the retinal ganglion cells. Their experiments revealed the type of receptive field profiles the retinal ganglion cells possess. They discovered that a retinal ganglion cell or a geniculate cell responds well to a circular spot of light² and found two basic cell types: the *center-on*

²This implies that the ganglion cells have circular shaped receptive fields, we use the terms *receptive field size* or *radius* to give the boundary of the receptive field.

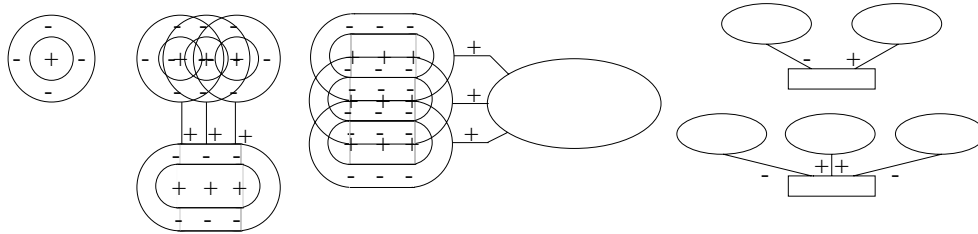


Figure 2.6: The building block structure. At the left the center-surround cells (with center on and surround off) is given. A number of these cells are making excitatory synaptic connections with a cell of higher order. If we assume that instead of three center-surround cells many more cells are along a line, then we get the receptive field profile of a simple cell, which will consist of a long narrow excitatory region with inhibitory flanks. A number of simple cells (only three are illustrated) make excitatory synapses with a single complex cell (which is represented as a white ellipse). Complex cells make excitatory and inhibitory synapses with a single end-stopped cell (represented as a rectangle). The top-right rectangle has one excitatory and one inhibitory part and therefore represents the receptive field of a single end-stopped cell. Similarly the bottom-right rectangle represents a double end-stopped cell which has two inhibitory parts.

surround-off cells and the *center-off surround-on* cells. The *receptive field center* of a center-surround cell is that part of the receptive field that is excitatory in a center-on surround-off cell and inhibitory in a center-off surround-on cell.

Kaplan et al. [96] and Shapley et al. [171] point out that there are two ganglion cell types. The first type are the monkey *M cells*. These cells are retinal ganglion cells which project to the magno-cellular layers (ventral layers) in the LGN. The second type are monkey *P cells* which project to the parvo-cellular layers (dorsal layers) in the LGN. The two ventral layers in the LGN are more sensitive to luminance contrast than the cells in the four dorsal layers of the LGN. The differences between these layers are due to the fact that there are differences in the retinal ganglion cells which provide excitatory synaptic input to the LGN-neurons, and not because of the organization (pattern of connectivity) in the LGN. The P cells in the monkey are color sensitive (wavelength selective), have small concentric center-surround receptive fields, and are not very contrast sensitive. The M cells are not wavelength selective, and have concentric center-surround receptive fields. Their receptive fields are larger than the receptive fields of the P cells and are highly sensitive to contrast (Shapley et al. [171], Livingstone et al. [119], and Nicholls et al. [141]).

Hubel and Wiesel [79, 83, 84] did pioneering work on the cat's striate cortex. They explored various visual cortical regions with micro-electrodes and divided the recorded cells into four distinct classes based on a so-called *building block structure* of these classes. An illustration of the building block architecture is given in Figure 2.6.

The first class are the *center-surround cells*. The ganglion cells belong to this class. These cells are found in the retina, the lateral geniculate body, and in layer 4C α of the striate cortex. Kuffler [101] discovered that when the receptive field center of a center-on surround-off cell gets a stimulus the cell starts firing immediately. When the inhibitory part is stimulated it does not fire until the inhibitory part is not stimulated anymore. Then the cell starts firing.

The second class are the *simple cells*, which were found coincidentally in the late 1950s by

Hubel and Wiesel (taken from Hubel [88]):

We were inserting the glass slide with its black spot into the slot of the ophthalmoscope when suddenly over the audio monitor the cell went off like a machine gun. After some fussing and fiddling we found out what was happening. The response had nothing to do with the black dot. As the glass slide was inserted its edge was casting onto the retina a faint but sharp shadow, a straight dark line on a light background. That was what the cell wanted, and it wanted it, moreover, in just one narrow range of orientations.

This was unheard of. It is hard, now, to think back and realize just how free we were from any idea of what cortical cells might be doing in an animal's daily life.

The simple cells receive their inputs from the center-surround cells (Hubel and Wiesel [87]). It is well known that a large amount of these cells react strongly to oriented lines and are found in layer 4B and 4C β of the striate cortex (Hubel and Wiesel [79]).

Probably two types of simple cells exist, a symmetric and an anti-symmetric type. Pollen and Ronner [152] recorded simultaneously the response from pairs of adjacent cells which had identical frequency and orientation tuning properties. Such paired cells differed in spatial phase by approximately 90 degrees. This suggests that the simple cells always come in pairs: a symmetric and an anti-symmetric cell with overlapping receptive fields (Atick and Redlich [6]).

The third class is that of the *complex cells*. The difference between the simple cells and the complex cells is that the latter have larger receptive fields and, instead of having discrete regions from which one can obtain excitatory and inhibitory responses (to a bar stimulus in the preferred orientation), responses can be obtained from every part of the receptive field (Hubel [89] and Zeki [204]). The characteristics of these cells can be accounted for by supposing that several simple cells excite one complex cell (Hubel and Wiesel [79]). Complex cells are the most common cells in the striate cortex, they make up about 75 percent of the population (Hubel [89]). The first cell Hubel and Wiesel recorded from (the one that responded to the edge created by the glass slide) was most probably a complex cell.

Simple and complex cells show length summation; the longer the line segment in the input stimulus the stronger the response. Of course, when the length of the line segment is longer than the diameter of the receptive field of the cell it does not change its response anymore, because nothing can influence the cell outside its receptive field.

Even more complicated cells were first referred to as *hyper-complex cells*, the fourth class of cells. They behave as if they receive input from two or more complex cells. When there are two input channels, there is one excitatory and one inhibitory input to the hyper-complex cell (Hubel and Wiesel [85]). Like simple and complex cells, these cells respond to lines of a specific orientation but unlike the former the length of the line is critical. Presenting a longer line segment increases the response up to a certain limit. If this limit is exceeded the response decreases. The hyper-complex cells respond strongly to line-ends, corners, and highly curved segments. The cells are found in layer 2 and 3 of monkey area V1, where they make up about 20 percent of the cells. Later these cells became known as *end-stopped cells*, because of their properties (Hubel [89]).

Mathematical Models of Different Cells in Early Vision

THE AIM OF THIS CHAPTER is to give a general overview of the most widely used models for different cells in early vision. The mathematical models used here can have modifications, partly simplifications, partly extensions, of models known in literature. We shall discuss the reasons for specific choices and modifications.

In Section 3.1 we describe a relation between the receptive field size and the dendritic tree of a ganglion cell. Discussion of this relation is necessary, since in some papers (e.g. Shapley and Perry [171]) a dendritic tree is used, while most of the papers use a receptive field. In the same section the average receptive field size of both P and M cells is discussed. In Section 3.2 operators are defined to model the responses of these ganglion cells. Center-surround cells respond well to differences in contrast but are not orientation selective. The outputs of these cells form the inputs for the simple cells, which are orientation selective and form the basis of our object recognition model. Atick and Redlich [6], e.g., made a mathematical model of simple cells in the P pathway. In their model the simple cell sums its input from a set of ganglion cells. Hence it is good to get some insights to what kind of stimuli these cells respond well. In Sections 3.3 and 3.4 we model the response of simple and complex cells, respectively. First we model the receptive field profiles of the simple cells. For these profiles we use two-dimensional Gabor functions. To obtain a better approximation of the simple cell responses, these functions are normalized so that the simple cell operator has zero output if the image is uniform. This property is called admissibility (Würtz [197]). The modeled simple cell operator does not always have a strong response at the exact position of the edge. The operator used for the complex cells responds strongly to the exact location of edges and is therefore suitable as edge and line detector. In Section 3.4 we also introduce a principal orientation operator which gives the preferred orientation of the complex operator.

In Section 3.5 we introduce two scale spaces, one with a linear increasing and one with a non-linear increasing scale space parameter. A scale space is used in Chapter 5 for corner detection. In Section 3.6, a model is given for the mapping between visual field and cortex. The first “neuronal” image is created in the retina, which is a two-dimensional grid of rods

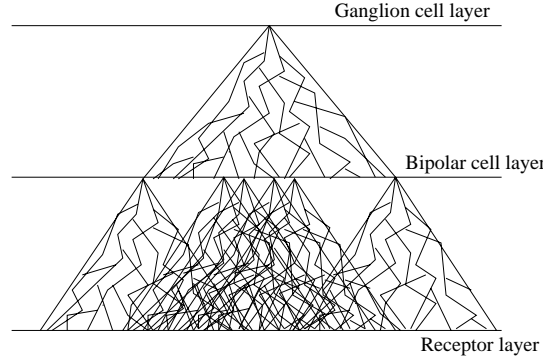


Figure 3.1: The dendritic tree of the ganglion cell plus the dendritic trees of the bipolar cells which are directly connected with this ganglion cell is defined as the dendritic tree of the ganglion cell to the receptors.

and cones. This grid is not uniform (we lose visual acuity with eccentricity), but it is not deformed, unlike the striate cortex. Hubel and Wiesel [91] described a topographic mapping of the visual field onto the cortex and the relation between average receptive field size and magnification factor. The magnification factor is the relation between a movement in the cortex and the displacement in the visual field (Daniel and Whitteridge [22] and Hubel and Freeman [90]). The receptive field size of a cell in the primary visual cortex can be derived from the magnification factor and the fact that a two millimeter movement in the cortex is needed to displace from one receptive field region to an entirely new region, i.e. such that the receptive fields do not overlap. This deformed map is of importance when motion is involved. In this thesis we limit ourselves to static images but we plan to investigate motion in the future. The chapter ends with a summary and discussion.

3.1 Relation between receptive field center size and dendritic tree of a ganglion cell

All direct connections from the bipolar cells to a ganglion cell together form the *dendritic tree* of a ganglion cell. The maximum size or radius of such a tree is called the *dendritic tree size* or radius. The dendritic tree from the receptors to a ganglion cell is defined as the tree of connections that can be reached by the direct pathway, from receptors via bipolar cells to ganglion cells (Hubel [89]). This is illustrated in Figure 3.1.

The direct pathway from receptors to a ganglion cell is responsible for the receptive field center of that ganglion cell. The indirect pathway, which is the path via horizontal cells and amacrine cells, is responsible for the surround of the receptive field. We conclude that the dendritic tree radius from the receptors to a ganglion cell is equal to the receptive field center radius of the same ganglion cell. This implies that the average variation in the dendritic tree size with retinal eccentricity of the cells is equal to the average variation in receptive field

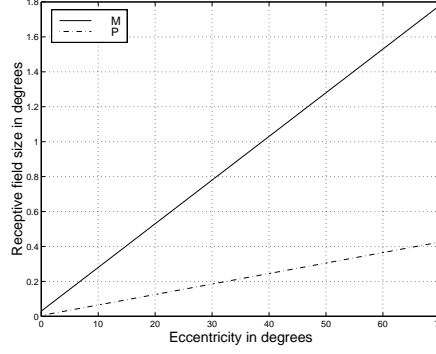


Figure 3.2: The average receptive field center radius grows linearly with eccentricity for both monkey M and P cells. Note that receptive fields of M cells are about a factor four larger than that of the P cells.

center size of the cells.

According to Shapley and Perry [171], the dendritic tree sizes of the ganglion cells grow linearly with retinal eccentricity. We also assume that the average dendritic tree size of the bipolar cells grows in a linear way with increasing retinal eccentricity. Assume that the average dendritic tree radius between ganglion and bipolar cells is

$$gb(\alpha) = a\alpha + c \quad (3.1)$$

and that the average dendritic tree between bipolar cells and the receptors is

$$br(\alpha) = b\alpha + d, \quad (3.2)$$

where α is the eccentricity in degrees and a , b , c , and d are constants. The total dendritic tree radius from ganglion cells to receptors, which is equal to the receptive field center radius of the ganglion cell, will be:

$$\begin{aligned} gr(\alpha) &= gb(\alpha) + br(\alpha + gb(\alpha)) \\ &= a\alpha + c + b(\alpha + a\alpha + c) + d \\ &= (a + b + ab)\alpha + bc + c + d. \end{aligned} \quad (3.3)$$

This implies that the dendritic tree between ganglion cell and receptors also grows in a linear way with increasing retinal eccentricity.

3.1.1 Receptive field sizes of monkey M and P cells

Six categories of cells exist in the different layers of the retina (rods and cones included). One of these layers is the ganglion cell layer, which can be subdivided by type of cell. One of the earliest subdivisions of ganglion cells in the cat retina was made by Enroth-Cugell and Robson [38]. The presence of color vision in the monkey aided the classification

considerably, since one type projects to the P pathway, which is responsible for color, and the other type projects to the M pathway (Leventhal [110] and Perry et al. [148]).

Let us make the assumption that given the eccentricity, both ganglion and bipolar cells have exactly the same average dendritic tree radius. Shapley and Perry [171] elaborated the dendritic tree size for ganglion cells to bipolar cells with retinal eccentricity. If we assume that both ganglion and bipolar cells have the same average dendritic tree radius, then the average dendritic tree radius from ganglion cells to the receptors is twice the size of the average dendritic tree radius of the ganglion cells. The receptive field center radius for a given eccentricity α for the M cells (Figure 3.2) is

$$r_{Mc}(\alpha) = 2.5 \cdot 10^{-2} \alpha + 3.0 \cdot 10^{-2}, \quad (3.4)$$

and the smaller receptive field center radius of the P cells (Figure 3.2) is

$$r_{Pc}(\alpha) = 6.0 \cdot 10^{-3} \alpha + 5.0 \cdot 10^{-3}, \quad (3.5)$$

where α is the eccentricity in degrees.

In the cat, the ganglion cell classes are labeled W, X, and Y. They are distinguished by the different retinal subnetworks that provide their input. X-cells are driven by a single linear receptive field center mechanism. Y-cells receive center and surround signals from nonlinear subunits in their receptive fields. The monkey M cells and the cat X-cells have several similarities, but their receptive field radii differ in the center of the visual field ($\alpha = 0$). The M cells have a center radius of 0.03° , whereas the X-cells have radii of 0.05° . Both increase approximately in the same way with retinal eccentricity. The receptive field centers of the Y-cells are three times larger than those of the X-cells (Shapley et al. [171]).

3.2 Center-surround cells

Most primate retinal ganglion cells exhibit a center-surround organization of their receptive fields, where the center and surround responses are antagonistic. Most commonly, the central region is flanked by an antagonistic surrounding region that is spatially more extended than the central region.

3.2.1 Receptive field profiles of center-surround cells

The best known, physiologically based receptive field profile of the center-surround cells is the difference of two Gaussians (DOGs), which was introduced by Rodieck [161] to describe the spatial sensitivity to a dot of light of retinal ganglion cells. The combination of these two functions is also called, because of its shape, a *Mexican-hat* function. Enroth-Cugell and Robson [38] were the first who used the function for the receptive field profiles of these cells in the spatial frequency domain.

Instead of using the difference of two Gaussians (for its definition and properties we refer to Section A.3) also the Laplacian of a two-dimensional Gaussian function, obtained by multiplying two one-dimensional Gaussian functions (A.3), can be used to get a Mexican-hat function (A.7). The Laplacian of a Gaussian is zero-normalized (A.11), and the center

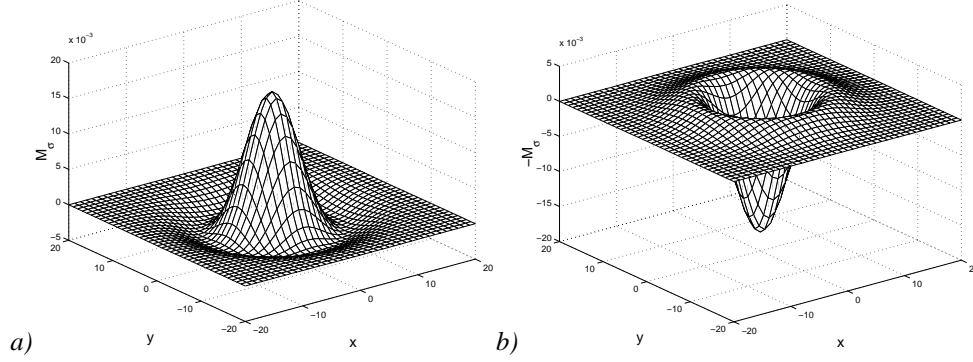


Figure 3.3: Receptive field profiles of modeled center-surround cells. *a)* The center-on, surround-off cell. *b)* The center-off surround-on cell. We used a normalized Laplacian of a Gaussian (A.7) with $\sigma = 5$.

(or the absolute function) can be normalized (A.13). Normalization is used if the function is used as a filter for different scales σ (which determine the sizes or spatial frequency of the receptive fields) and the outputs for different σ have to be combined or compared. An illustration of such a normalized function is given in Figure 3.3. The choice for a Laplacian or a difference of Gaussians depends on the choice for the ratio between center and surround. If a constant ratio is desired, the choice for a Laplacian of a Gaussian (Section A.2) would be a proper choice. The ratio can be varied for the difference of Gaussians, but the ratio between center and surround is at least three for the two-dimensional function (Figure A.4a).

3.2.2 Responses of center-surround cells

The receptive fields of both M and P cells have center-surround properties and their responses depend on the difference in contrast between that part of the visual field which influences the excitatory part of the cell and the part of the visual field which influences the inhibitory part of the cell.

The spatial impulse response of a two-dimensional layer of center-surround cells is modeled by convolving the input stimulus (a two-dimensional image) I with function M_σ (A.7). Hence one can say that the response of a cell can be approximated by a function from $\mathbb{R}^2 \rightarrow \mathbb{R}$. In formula:

$$M_\sigma(x, y) = \int_{x_1} \int_{y_1} I(x_1, y_1) M_\sigma(x - x_1, y - y_1) dx_1 dy_1, \quad (3.6)$$

where we assume that the receptive field sizes remain constant for all $(x, y) \in I$. The average receptive field size (σ) for every (x, y) is approximately the same, since I is usually a small part of the visual field.

We do not use spatio-temporal properties, since we are interested in pattern recognition in static images. For the spatio-temporal properties in the cat's retinal ganglion cells we refer

to e.g. Enroth-Cugell et al. [39] or, more recently, Somers et al. [176]. Cell responses (firing rates) are by definition positive, whereas idealized linear cells¹ can have positive or negative responses. An interpretation of the linear model (like (3.6)) is that the positive values are interpreted as direct firing and negative output as firing with a delay. We interpret the positive and negative values in the linear model as a tuple of two cells with opposite properties. One cell responsible for the positive (center-on surround off) and one cell responsible for the negative part (center-off surround-on). This makes the model computationally attractive, because both center-on surround-off and center-off surround-on cell responses can be obtained by a single convolution. Atick and Redlich [6] built in such a non-linearity by thresholding the output at zero and use a tuple of center-on surround-off and center-off surround-on cells.

The M cell and P cell respond differently to contrast, so an additional step has to be taken to model this difference. The response of the M cell, which is highly contrast sensitive, is as follows:

$$N_{\sigma}(x, y) = \frac{a_{\sigma} M_{\sigma}(x, y)}{0.13 + M_{\sigma}(x, y)}, \quad (3.7)$$

where N is the response for an M cell and a_{σ} is the highest possible response of the cell achieved at a maximum contrast difference. The P cell is a low-contrast sensitive filter. Its response is modeled as follows:

$$P_{\sigma}(x, y) = \frac{a_{\sigma} M_{\sigma}(x, y)}{1.74 + M_{\sigma}(x, y)}. \quad (3.8)$$

Both (3.7) and (3.8) are taken from Kaplan and Shapley [96].

3.3 Simple cells

Hubel and Wiesel [87] discovered that most cells in the primary visual cortex, layer 4 of cats, are orientation selective. These cells, called simple cells, do not respond to diffuse light and respond only weakly to spots of light. To obtain a strong response from them, the activating stimulus must be an edge of the proper orientation and must fall within the excitatory part of the receptive field (Hubel and Wiesel [79]). Because of their properties in image processing, the simple cells can be seen as line and edge enhancers.

3.3.1 Receptive field profiles of simple cells

The spatial receptive field profiles of simple cells can be fitted well by Gabor (Gabor [53]) functions² (Marčelja [130], Daugman [23, 24], Jones and Palmer [95], and Lee [108]), the first derivatives of Gaussians, or other similar functions (Stork and Wilson [178, 179, 194])

¹A cell is linear when its response depends linearly on a single quantity: the weighted linear sum of the local luminosity over its receptive field (Atick and Redlich [6]).

²A Gabor function is defined by a sine (or cosine) wave modulated by a Gaussian envelope.

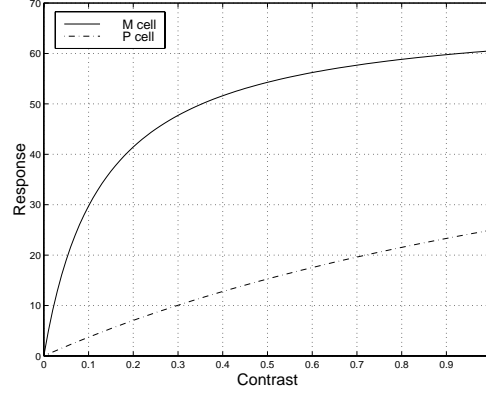


Figure 3.4: For both P and M cells, the contrast-response functions N and P are given. The magno-cellular cells (M cells) are highly sensitive to contrast, whereas the parvo-cellular cells (P cells) are less sensitive to contrast. The response is measured in impulses per second.

and Hawken and Parker [66]). Stork and Wilson propose a model with difference of Gaussians

$$S(x, y) = A \left(e^{-\frac{x^2}{\sigma_1^2}} - B e^{-\frac{x^2}{\sigma_2^2}} + C e^{-\frac{x^2}{\sigma_3^2}} \right) e^{-\frac{y^2}{\sigma_4^2}}. \quad (3.9)$$

Like in previous work, Lourens [120], we have chosen to model the receptive field profiles of the symmetrical and anti-symmetrical simple cells by the real and imaginary part of a two-dimensional Gabor function $G_{\sigma, \theta}$:

$$\Re G_{\sigma, \theta}(x, y) = \cos \left(\frac{\pi}{\sigma \sqrt{2}} (x \cos \theta + y \sin \theta) \right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3.10)$$

and

$$\Im G_{\sigma, \theta}(x, y) = \sin \left(\frac{\pi}{\sigma \sqrt{2}} (x \cos \theta + y \sin \theta) \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (3.11)$$

where θ denotes the preferred orientation, and σ the receptive field size.

The symmetrical two-dimensional Gabor function $\Re G$ does not have a zero integral. If such a function is used as a filter, it will also respond to homogeneous fields of constant intensity and is sensitive to changes of absolute intensity. The function needs to be zero-normalized. Since simple cells do not respond to uniform stimuli but only to edges of a preferred orientation. We normalize this function by multiplying the negative part of the function with a factor γ^- and the positive part with a factor $\gamma^+ = 1$. The normalization factor γ^- is as follows:

$$\gamma^- = \frac{\int_x \int_y [\Re G_{\sigma, \theta}(x, y)]^{\geq 0} dx dy}{\int_x \int_y [\Re G_{\sigma, \theta}(x, y)]^{\leq 0} dx dy}. \quad (3.12)$$

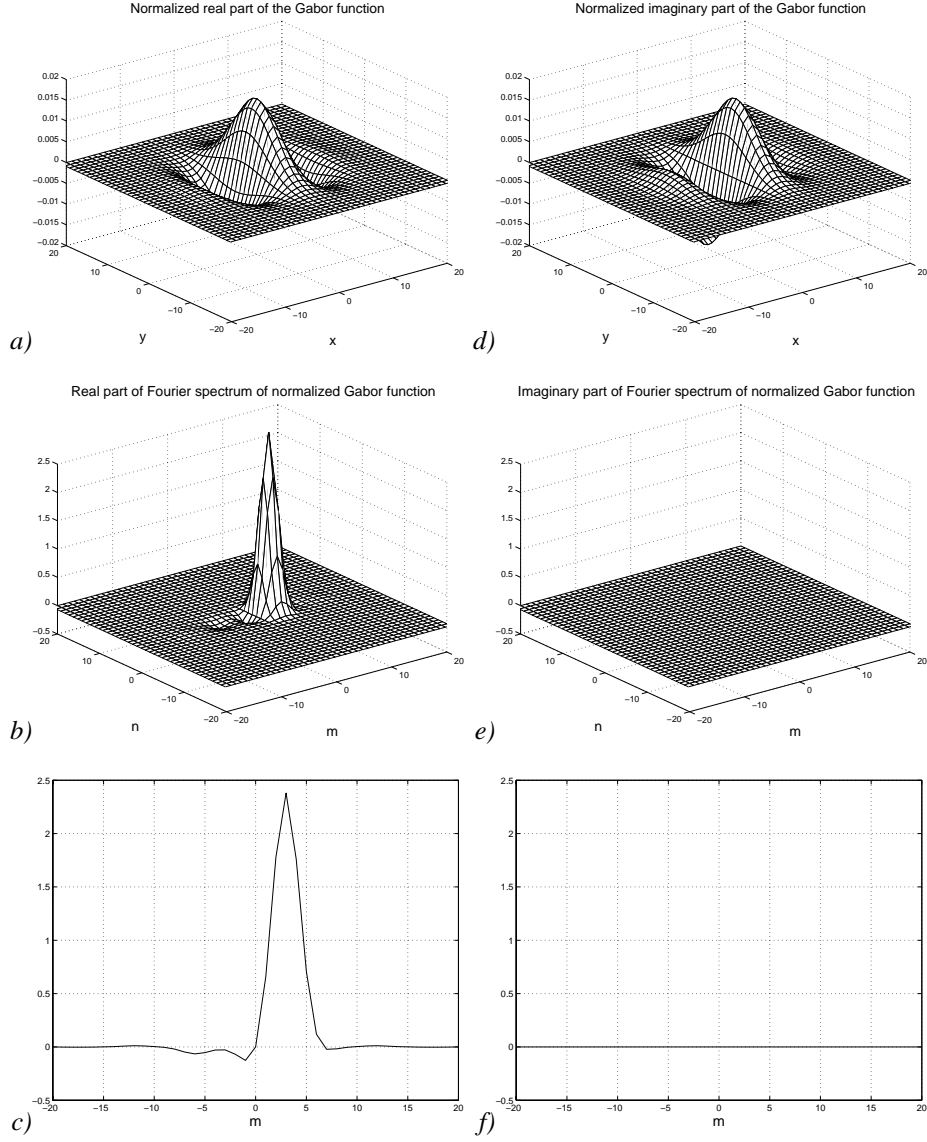


Figure 3.5: *a), d)* The real and imaginary part of a Gabor function for $\sigma = 5$ and $\theta = 0$. *b), e)* Real and imaginary part of the Fourier transform of the normalized Gabor function ($\Re G + i\Im G$). *c), f)* A cross section through the zero axis of *b)* and *e)*, respectively.

The function $[x]^{\geq 0}$ is defined to be 0 where its argument is negative and x elsewhere. The function $[x]^{\leq 0}$ is 0 where its argument is positive and $-x$ elsewhere. Orientation θ can be chosen arbitrarily since the normalization factor γ^- is orientation independent. Due to the

factor $\frac{1}{\sqrt{2\sigma^2}}$, the size σ does not affect the normalization factor. The factor γ^- is approximately 1.3 for all orientations and sizes.³ This normalization of the two-dimensional Gabor functions does not mean that the integral over the positive or absolute values of the function is equal for different σ . The amplitude of the function for different receptive field sizes is different. This fact forces us also to normalize the positive part of the function, otherwise one cannot compare the results of an image convolved with two functions which have a different amplitude. The normalization factors for the amplitude of the Gabor functions are as follows:

$$\eta_{\Re} = \int_x \int_y [\Re G_{\sigma,\theta}(x,y)]^{\geq 0} dx dy \quad (3.13)$$

and

$$\eta_{\Im} = \int_x \int_y [\Im G_{\sigma,\theta}(x,y)]^{\geq 0} dx dy, \quad (3.14)$$

for the symmetrical (η_{\Re}) and anti-symmetrical (η_{\Im}) two-dimensional Gabor function, respectively. The symmetrical two-dimensional Gabor function (3.10) is multiplied with $\frac{\gamma}{\eta_{\Re}}$ and the anti-symmetrical two-dimensional Gabor function (3.11) is divided by η_{\Im} . Note that instead of integrating the positive part of the Gabor function for amplitude normalization we also could have taken the integral over the absolute values of the function. Figure 3.5 illustrates the normalized Gabor function in both spatial and (spatial) frequency domain. In this figure we also illustrate the function in frequency domain since for computational reasons it is often calculated in Fourier domain. For normalization in Fourier domain we refer to Würtz [197]. The normalized function $\hat{G}_{\sigma,\theta}$ is as follows:

$$\Re \hat{G}_{\sigma,\theta}(x,y) = \frac{\gamma^+}{\eta_{\Re}} [\Re G_{\sigma,\theta}(x,y)]^{\geq 0} - \frac{\gamma^-}{\eta_{\Re}} [\Re G_{\sigma,\theta}(x,y)]^{\leq 0} \quad (3.15)$$

and

$$\Im \hat{G}_{\sigma,\theta}(x,y) = \frac{1}{\eta_{\Im}} \Im G_{\sigma,\theta}(x,y). \quad (3.16)$$

The integral $\int_x \int_y \Re \hat{G}_{\sigma,\theta}(x,y) dx dy = 0$ because

$$\begin{aligned} & \frac{\gamma^+}{\eta_{\Re}} \int_x \int_y [\Re G_{\sigma,\theta}(x,y)]^{\geq 0} dx dy - \frac{\gamma^-}{\eta_{\Re}} \int_x \int_y [\Re G_{\sigma,\theta}(x,y)]^{\leq 0} dx dy \\ &= \frac{1}{\eta_{\Re}} \eta_{\Re} - \frac{\eta_{\Re}}{\eta_{\Re} \int_x \int_y [\Re G_{\sigma,\theta}(x,y)]^{\leq 0} dx dy} \int_x \int_y [\Re G_{\sigma,\theta}(x,y)]^{\leq 0} dx dy \\ &= 0 \end{aligned}$$

³In the discrete case γ^- is approximately the same for every orientation if σ is large. If a small σ is used the discretization effects will be strong. For small σ it is useful to calculate γ^- for every orientation independently.

and $\int_x \int_y |\Re \hat{G}_{\sigma, \theta}(x, y)| dx dy = 2$ is independent of σ since

$$\begin{aligned} & \int_x \int_y \left| \frac{\gamma^+}{\eta_{\Re}} [\Re G_{\sigma, \theta}(x, y)]^{\geq 0} - \frac{\gamma^-}{\eta_{\Re}} [\Re G_{\sigma, \theta}(x, y)]^{\leq 0} \right| dx dy \\ &= \int_x \int_y \frac{1}{\eta_{\Re}} [\Re G_{\sigma, \theta}(x, y)]^{\geq 0} dx dy + \int_x \int_y \frac{\gamma^-}{\eta_{\Re}} [\Re G_{\sigma, \theta}(x, y)]^{\leq 0} dx dy \\ &= 2. \end{aligned}$$

Since the main objective of Stork and Wilson [179] “the fact that the real parts of Gabor functions do respond to constant illumination” is solved, there seems to be no serious difference between their model and normalized Gabor functions and the choice is more a matter of taste.

3.3.2 Responses of simple cells

The response of a simple cell can be modeled by convolving the input stimulus (image) I with receptive field function $\Re \hat{G}$ or $\Im \hat{G}$. The response of the symmetrical simple cell is modeled as follows:

$$S_{\sigma, \theta}^s(x, y) = \int_{x_1} \int_{y_1} I(x_1, y_1) \Re \hat{G}_{\sigma, \theta}(x - x_1, y - y_1) dx_1 dy_1. \quad (3.17)$$

Equation (3.17) is known as the convolution of I and $\Re \hat{G}_{\sigma, \theta}$, also denoted as $S_{\sigma, \theta}^s = I * \Re \hat{G}_{\sigma, \theta}$. We use a periodic convolution since it is in a standard software library. Using a periodic convolution means that we get wrap around effects. This implies that artifacts can appear at the boundaries of the images. The boundary artifacts can be eliminated by removing the information at the boundaries of the resulting image or by zero padding. The latter has the disadvantage that the data that actually have to be convolved are becoming larger. The response of the anti-symmetrical simple cell is modeled as follows:

$$S_{\sigma, \theta}^a = I * \Im \hat{G}_{\sigma, \theta}. \quad (3.18)$$

The difference between (3.17) and (3.18) is that in the former, image I is convolved with $\Re \hat{G}_{\sigma, \theta}$ while in the latter I is convolved with $\Im \hat{G}_{\sigma, \theta}$. We call them $S_{\sigma, \theta}^s$ - and $S_{\sigma, \theta}^a$ -operator, respectively, since $S_{\sigma, \theta}^s$ and $S_{\sigma, \theta}^a$ are operators on input signal I , and yield a function of (x, y) . To write equation (3.17) correctly we should use $S_{\sigma, \theta}^a I$, but for abbreviation we will skip I for every operator used in this thesis.

In analogy with the ganglion (center-surround) structure, Atick and Redlich [6] assume that the simple cell's input comprises two components, one excitatory and the other inhibitory, summated separately. In the mathematical model of simple cells in its general form, Atick and Redlich again use a non-linearity by thresholding at zero. Experiments however indicate that a very large class of simple cells is linear with respect to spatial summation (Maffei and Fiorentini [123], Movshon et al. [137], Andrews and Pollen [4], Kulikowski and Bishop [102], and Pollen and Ronner [153]). Therefore Atick and Redlich extended their model to restore linearity.

3.4 Complex cells

Complex cells share with simple cells the property of responding only to specifically oriented lines and edges. Like the simple cells, they respond over a limited region of the visual field but unlike simple cells their behavior cannot be explained by a subdivision of the receptive field into excitatory and inhibitory regions. The receptive fields of complex cells are somewhat larger than simple cells. There are complex cells that only respond properly if an oriented line, edge, or slit of light is moved across the receptive field. Complex cells that do react to a stationary line show a response wherever the line is placed within the receptive field, provided that the orientation is correct (Hubel [89]). In the building block architecture this means that a number of simple cells make excitatory synapses with a single complex cell.

3.4.1 Responses of complex cells

A possible operator to model the response of the complex cells is:

$$C_{\sigma,\theta}(x,y) = \max_{(x_1,y_1) \in \Xi} \left(S_{\sigma,\theta}^s(x_1,y_1), S_{\sigma,\theta}^a(x_1,y_1) \right), \quad (3.19)$$

where Ξ is a small area with center (x,y) . A logical choice for Ξ would be an elliptic area with the long axis perpendicular to the orientation of the line.

Movshon et al. [137] and Heggelund [69] describe complex cells as a non-linear pooling of responses from a spatially distributed group of subunits within the cell's receptive field. Wilson [195] sums the different inputs of the complex cell instead of using the max-operator (3.19), to describe its response.

Adelson and Bergen [1] and Morrone and Burr [136] combined the $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operators to a "local energy" representation, which is defined by

$$C_{\sigma,\theta}(x,y) = \sqrt{S_{\sigma,\theta}^s(x,y)^2 + S_{\sigma,\theta}^a(x,y)^2}, \quad (3.20)$$

where orientation $\theta \in [0, \pi)$ since $C_{\sigma,\theta} = C_{\sigma,\theta+\pi}$. Heitger et al. [71] used this representation and called it the $C_{\sigma,\theta}$ -operator in analogy to complex receptive fields.

Although the modeled response of a complex cell (3.20) is not exactly according to the model suggested by Hubel, it gives a reasonable approximation. If we modify this equation using excitatory input, we get

$$C_{\sigma,\theta}(x,y) = \max \left(S_{\sigma,\theta}^s(x,y), -S_{\sigma,\theta}^s(x,y), S_{\sigma,\theta}^a(x,y), S_{\sigma,\theta+\pi}^a(x,y) \right). \quad (3.21)$$

This equation is not exactly the same as (3.20) but gives a good approximation. In Figure 3.6 an example for the output of the $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operator is given. The strongest response of the former operator is at the proper position at an edge but the strongest response of the latter is shifted. In the two-dimensional case this shift is perpendicular to preferred orientation θ . When a line is used the $S_{\sigma,\theta}^s$ -operator gets the strongest responses at the proper position and the $S_{\sigma,\theta}^a$ -operators strongest responses are shifted. Using a $C_{\sigma,\theta}$ -operator from (3.19), (3.20),



Figure 3.6: Response of different operators at a slice from an image (I) with an up- and down-going edge and two lines. In the second and third row the outputs of the S^a - and S^s -operator are given. The fourth row gives the output of the C -operator from (3.20) and in the last row the output of the operator from (3.21) is given.

or (3.21) avoids shifting of the strongest response. Although the difference in output between the $C_{\sigma,\theta}$ -operators (3.20) and (3.21) is small, we will use the operator from (3.20) because of its smoothness. This property is illustrated in Figure 3.6 where Ca (for complex amplitude) shows a smooth response profile, in contrast to Cm (complex maximum) which shows a ragged response profile.

3.4.2 Additional complex cell based operators

The $C_{\sigma,\theta}$ -operator responds to edges and lines of a specific orientation. For example, if we consider an image with a circle, the operator would respond only to parts of the circle. These parts depend on the preferred orientation θ . That information is split into different orientation channels is not always desirable (it depends on the goal one has in mind). In case of a circle one could prefer a non-orientation selective edge filter. When N different orientations of the $C_{\sigma,\theta}$ -operators are combined with a max-operator, to get the strongest responses over all orientations, we get a non-orientation selective edge filter. The operator thus obtained we also call complex operator. This C_{σ} -operator is given by:

$$C_{\sigma}(x, y) = \max_{i \in [0, \dots, N)} (C_{\sigma, \theta_i}(x, y)), \quad (3.22)$$

where N is the number of orientations and $\theta_i = \frac{i\pi}{N}$. Since all orientation specific information is available we introduce another operator which can be used together with (3.22) to facilitate edge detection. The operator selects the orientation which has the strongest response and thus selects the orientation that is closest to one of the preferred orientations of the $C_{\sigma,\theta}$ -operators:

$$O_{\sigma}(x, y) = \theta_i(x, y), \quad (3.23)$$

where $i \in [0, \dots, N)$ is a value which satisfies

$$\forall_{j \in [0, \dots, N)} \left(C_{\sigma, \theta_i}(x, y) \geq C_{\sigma, \theta_j}(x, y) \right).$$

We call this the principal orientation or O_σ -operator.

3.5 Selection of scales in a multi-scale representation

As mentioned before we will use multiple scales for corner detection (Chapter 5). All operators we used up to now contain a parameter σ which determines the size of the receptive field. A single *scale* is interpreted as a discrete two-dimensional layer of one type of operator which all have the same σ . A multi-scale representation consists of several discrete two-dimensional layers of identical size, where every layer has a different σ . We use a multi-scale representation which is called a *scale space* where the image size (two-dimensional layer) is kept constant and the size of the filter is changed. Another method for creating a scale space is keeping the filter size constant and resizing the image, an approach used by Koenderink and van Doorn [98] in their stack model. These scale spaces mostly use resizing factors of two since this is the simplest way to scale a discrete image.

Lindeberg [112, 113] defines a scale space $L(x, y; t) = (g(\cdot, \cdot; t) * f(\cdot, \cdot))(x, y; t)$ as a convolution of a two-dimensional signal f with a Gaussian kernel g , where scale parameter $t = \sigma^2$. Creating a scale space by using parameter σ is more flexible than scaling a discrete image, since σ can have any arbitrary value larger than zero.

In practice a number of scales is selected. We elaborate two types of multiple scales: *linear scaling* in spatial domain and linear scaling in frequency domain. The latter scales non-linearly in spatial domain.

In either case we use receptive field radius r as scale parameter.⁴ Let $s + 1$ be the number of scales considered and let r_0 be the minimal scale and r_s be the maximal scale. For linear scaling the other scales are then given by

$$r_i = r_0 + i \frac{r_s - r_0}{s}. \quad (3.24)$$

We will call this a *linear scale space*.

In this thesis most operators are directly or indirectly based on a convolution. An image is convolved with a Gaussian G_σ , a Mexican-hat M_σ , or a Gabor function $\hat{G}_{\sigma, \theta}$ when the operators are directly based on a convolution. All three functions are based upon Gaussians. Therefore Gabor and Gaussian functions in the spatial domain show identically shaped Gaussians in frequency domain. Normalization of Gabor functions, as described in this chapter, causes the Gaussian to have negative values in the frequency domain (Figure 3.7). Since a Gaussian is not band limited it always causes some aliasing. To eliminate aliasing negative frequencies can be set to zero⁵ (Würtz [197]).

⁴The relation between σ and the receptive field radius r is linear (A.5).

⁵In all our experiments we did not set the negative frequencies to zero.

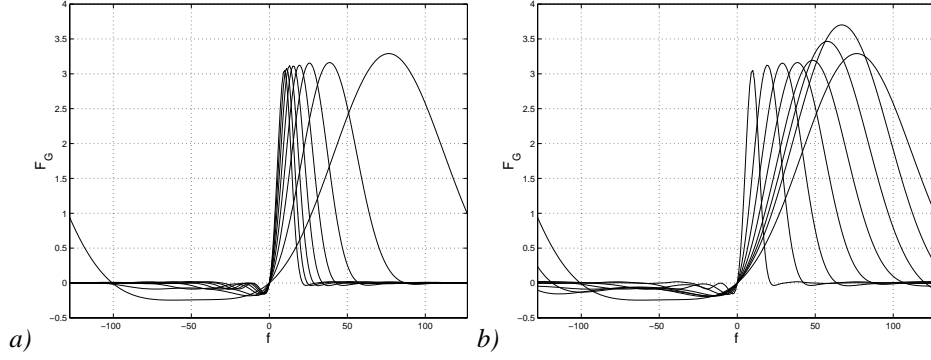


Figure 3.7: Frequency spectra of the real part of the Gabor function (similar to Figure 3.5c) for 8 different receptive field radii. *a)* Linear scaling gives linearly increasing receptive field radii (3.24). *b)* Linear scaling in frequency domain gives non-linear scaling in spatial domain (3.27). The following constants are used: $r_0 = 5$, $r_s = 40$, and $s = 7$.

Let us Fourier transform the Gaussian function from time (or spatial) domain to frequency domain:

$$\begin{aligned}
 F_G(f) &= \int_{-\infty}^{\infty} G(t) e^{-\sqrt{-1}2\pi f t} dt \\
 &= e^{-2(\sigma\pi f)^2} \\
 &= \left\{ \tau = \frac{1}{2\sigma\pi} \right\} \\
 &= e^{-\frac{f^2}{2\tau^2}}.
 \end{aligned} \tag{3.25}$$

This gives a Gaussian function without normalization factor.

Linear scaling in the frequency domain is commonly used in literature for information technical reasons. The formula is similar to (3.24):

$$\tau_i = \tau_0 + i \frac{\tau_s - \tau_0}{s}. \tag{3.26}$$

A linearly increasing scale space parameter τ with increasing i in frequency domain is not increasing linearly in spatial domain. If we use $\tau = \frac{1}{2\sigma\pi} = \frac{c}{r}$, where constant $c = \frac{\sqrt{-2\log \epsilon}}{2\pi}$, then scale space parameter r in spatial domain is as follows:

$$\begin{aligned}
 \tau_i &= \tau_0 + i \frac{\tau_s - \tau_0}{s} \\
 &\equiv \frac{c}{r_i} = \frac{c}{r_0} + i \frac{\frac{c}{r_s} - \frac{c}{r_0}}{s} \\
 &\equiv r_i = \frac{sr_0 r_s}{sr_s - i(r_s - r_0)}.
 \end{aligned} \tag{3.27}$$

We will call this a *non-linear scale space*.

The maximum radius r_s is at most half of the size of the square image since the convolution kernel should be smaller than the size of the signal (image). In practice the radius is smaller, because the receptive field radius of a cell is smaller than the whole visual field. Figure 3.7 illustrates that the minimum radius r_0 should be approximately eight to avoid aliasing effects in frequency domain.

3.6 Mapping the visual field onto the cortex

Using our visual system we have the feeling that everything we see is detailed, although this is not true, as mentioned in Chapter 2. The illusion that everything is detailed is created by the movement of the eyes. The decreased number of cells per degree of visual field in the periphery is therefore not a disadvantage, but rather an advantage since the total number of cells can be highly reduced.⁶

We can get a deeper understanding of the geometry of the cortex by comparing it with the visual field or retina.

For practical reasons the eye is a sphere, and that is consequently the shape of the retina. With a spherical eye, retinal magnification is constant: the movement in degrees of visual field per millimeter movement in the retina is constant throughout the retina. For the human eye it is 3.5 degrees per millimeter (Hubel [89]). Because the retina is spherical, its layers are not uniform. The ganglion cells in the retina are several layers thick in the fovea (center of the visual field) but in the far periphery there are holes in a single layer (van Buren [18]).

Unlike the retina, the cortex does not have to be spherical. In the cortex the foveal part is expanded relatively to the periphery. It presumably expands so that the thickness remains the same throughout. Daniel and Whitteridge [22] predicted that the unfolded cortex has a shape like a pear. Tootell et al. [185] illustrated how a two-dimensional stimulus is mapped on two-dimensional flattened sections of cortex. A model of this deformation between cortex and visual field will be given in this section.

A location dependent operator is created to model the response of a cortical cell at a position (x_c, y_c) in the cortex. This is possible since there is a relation between coordinate (x_v, y_v) in the visual field and coordinate (x_c, y_c) in the cortex. We transform both coordinates to polar coordinates (r_v, ϕ) and (r_c, ϕ) , respectively. Here $x_v = r_v \cos \phi$, $y_v = r_v \sin \phi$, $x_c = r_c \cos \phi$, and $y_c = r_c \sin \phi$ are the transformation functions from Cartesian to polar coordinates. Finding a relation between r_v and r_c will be sufficient, since the size of the receptive field changes with eccentricity only (Daniel and Whitteridge [22]).

3.6.1 Magnification between cortex and visual field

Before we start with the magnification between the primary visual cortex and the visual field, let us first consider the visual range of a human being. A human being has two visual ranges, one for the left and one for the right eye. In Figure 3.8 the visual range of the left eye is

⁶Standard CCD cameras need a little less than half a million light sensitive units where a camera based on a decreasing number of units with eccentricity would only need 8,000.

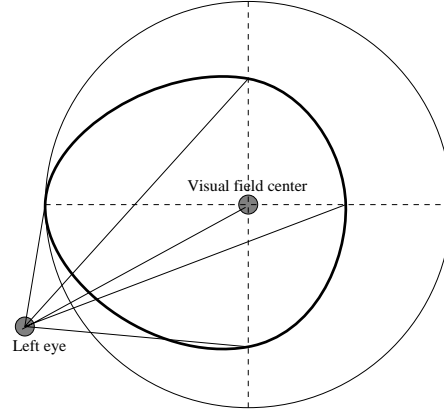


Figure 3.8: The visual range for the left eye of a human being. The maximum horizontal left angle is about 70° and the maximum horizontal right angle is about 35° (due to physical limitations: our nose prohibits us from an angle which would be about the same as the horizontal left angle). The vertical upper angle and vertical lower angle are about 40° and 50° degrees, respectively. The vertical angles are also both limited due to the fact that the physical structure of the face prohibits larger angles. In the model only the largest possible visual field angle is used, which implies that one gets a circular shaped visual range.

shown. In the model, for simplicity, we do not take into account the limited physical visual field angles but the maximal visual field angle reached by the horizontal left angle of the left eye. We define the maximum visual field angle as $\alpha_{VF,max}$. A radius r can be obtained from angle α if one radius and its corresponding angle are known. In formula:

$$r = \frac{r_k \tan \alpha}{\tan \alpha_k}, \quad (3.28)$$

where r_k with corresponding α_k are known.

The magnification is a relation between an angle in the visual field and a distance in the primary visual cortex. In the center (fovea) of the visual field a human being has a more detailed image than in the periphery, suggesting that more spatial information from the center of the visual field is (temporarily) represented in the primary visual cortex in comparison with the information from the periphery of the visual field. This implies that a larger area of the cortex is used for an area in the center of the visual field than for an identical sized area in the periphery. Daniel and Whitteridge [22] show that the visual field maps systematically on the cortex. They defined *magnification* as the distance in the cortex corresponding to a distance of one degree in the visual field. As we go out from the fovea, a given amount of visual field corresponds to a progressively smaller and smaller area of cortex. In the fovea a movement of approximately $\frac{1}{7}^\circ$ corresponds to a movement of 1 millimeter in the cortex and at the far periphery (a visual field angle of $\alpha_{VF,max}$) a one millimeter movement in the cortex corresponds to a shift of about 8° in the visual field (Daniel and Whitteridge [22]).

Daniel and Whitteridge not only showed that the magnification drops logarithmically if one moves from the center of the visual field to the far periphery, but also that this relation is

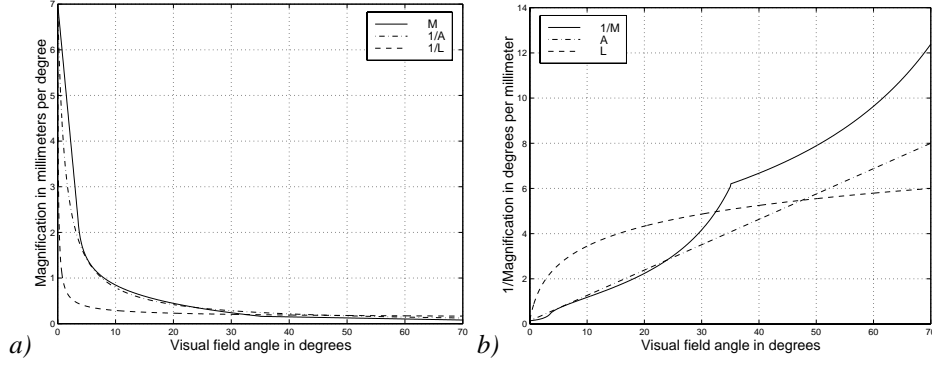


Figure 3.9: a) The magnification factor M (3.29) and the inverted functions L^{-1} and A^{-1} . b) The inverted magnification M^{-1} , logarithmic function L (3.30), and linear approximation A (3.31).

approximately valid in every direction. This means that the magnification only depends on the angle between the fovea and a certain position in the visual field.

Hubel et al. [91] also showed a relation between magnification and eccentricity. In contrast to Daniel and Whitteridge, who measured up to an eccentricity of about 65° , they measured within a maximal angle of 22° from the fovea and derived an expression for the inverse of magnification in degrees per millimeter. Thus they express a movement of one millimeter in the cortex as a displacement of the inverse of magnification in degrees in the visual field. Their results are comparable with those of Daniel and Whitteridge with respect to the maximal angle of 22° .

We make an approximation M (Figure 3.9a) of the magnification factor from the results reported by Daniel and Whitteridge [22], by using two linear functions and one logarithmic function:

$$M(\alpha) = \begin{cases} 7.00 - 1.34\alpha & \alpha \in [0 - 3.5) \\ 1.69 - 0.44 \log(\alpha - 3.25) & \alpha \in [3.5 - 35) \\ 2.42 \cdot 10^{-1} - 2.3 \cdot 10^{-3}\alpha & \alpha \in [35 - 65) \end{cases} \quad (3.29)$$

Where α is the retinal eccentricity or visual field angle in degrees. If an inverted magnification is used, as suggested by Hubel, and we assume that the inverted magnification shows a logarithmic behavior then the following function L (Figure 3.9b) can be used:

$$L(\alpha) = \frac{1}{6} + \frac{35}{6 \cdot \log(\alpha_{VF, \max} + 1)} \log(\alpha + 1) \quad \alpha \in [0, \alpha_{VF, \max}]. \quad (3.30)$$

From the inverted magnification M^{-1} the third linearly increasing approximation function A (Figure 3.9b) was derived. (See also Figure 6A from Hubel et al. [91] for a linearly increasing inverted magnification factor.) This function A is given by:

$$A(\alpha) = 0.14 + 0.11\alpha. \quad (3.31)$$

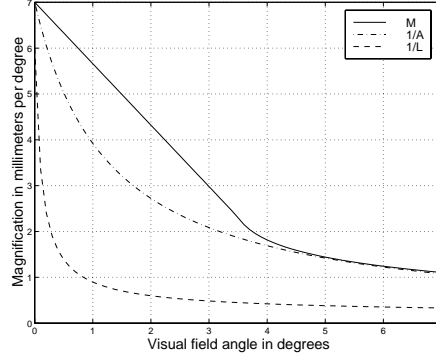


Figure 3.10: An enlargement of the center of the visual field from Figure 3.9a, to emphasize the difference between the magnification factor M and the inverted approximated linearly increasing function A^{-1} .

Hubel and Freeman [90] use linear inverse magnification factor $A(\alpha) = 0.11 + 0.06\alpha$. Schwartz used $a(x+b)/((x+b)^2 + y^2)$ with constants $a = 7.7$ and $b = 0.33$ which leads to an inverse factor of $A(\alpha) = 0.04 + 0.13\alpha$ (Swindale [181]).

If we consider the inverse function approximation M^{-1} , we conclude that the function is rather linear and can be approximated by A , although at first sight there seems to be a big difference if the visual field angle is larger than 25° (Figure 3.9b). This in fact is true, but if the approximation A is inverted (A^{-1}) it gives a good approximation of M as shown in Figure 3.9a. The difference is not in the periphery but near the fovea (i.e. a visual field angle $< 5^\circ$), as illustrated in Figure 3.10. Since A and M^{-1} hardly differ for visual field angles larger than 25 degrees and although Hubel et al. did only measure up to an eccentricity of 22 degrees we assume that using A for these angles is valid.

3.6.2 Determining the relation between eccentricity and cortex radius

We define the “center of the primary visual cortex” as the position where the center of the visual field is mapped. With this definition we are able to define the relation between the radius from the center of the primary visual cortex and the visual field angle. With a given cortex radius r_c (in millimeters) the visual field angle α is

$$\alpha = \frac{d_0}{d} \left(e^{dr_c} - 1 \right), \quad (3.32)$$

where d_0 and d are constants, which can be, e.g., obtained from (3.31). Suppose that the cortex radius $r_c = n$ millimeters and that we move with a step of ϵ millimeters, in such a way that the radius becomes $n + \epsilon$. Using function A , we obtain the relation between visual field angle α and the cortex radius r_c from (3.32) with the following function:

$$\alpha(n + \epsilon) = \alpha(n) + \epsilon A(\alpha(n))$$

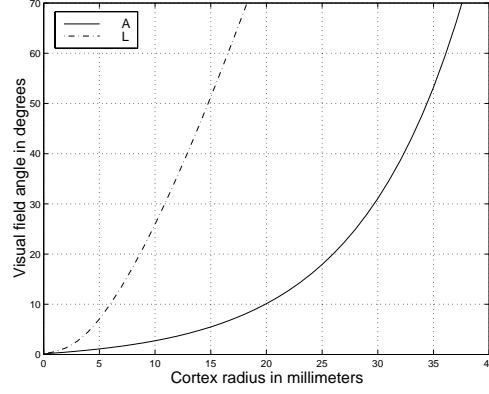


Figure 3.11: Relation between eccentricity and cortex radius, using linear magnification function A (solid line) or the logarithmic inverted magnification function L (dashed line).

$$\begin{aligned}
 &\equiv \frac{\alpha(n+\varepsilon) - \alpha(n)}{\varepsilon} = A(\alpha(n)) \quad \{\varepsilon \rightarrow 0 \text{ and } A(\eta) = d_0 + d\eta\} \\
 &\equiv \alpha'(n) = d_0 + d\alpha(n) \\
 &\equiv \alpha(n) = Ce^{dn} - \frac{d_0}{d} \quad \{\alpha(0) = 0\} \\
 &\equiv \alpha(n) = \frac{d_0}{d} (e^{dn} - 1).
 \end{aligned}$$

If the visual field angle α is known then the cortex radius is:

$$r_c = \frac{\log\left(\frac{d}{d_0}\alpha + 1\right)}{d}, \quad (3.33)$$

which is obtained by using (3.32).

Figure 3.11 illustrates that for a visual field angle of 40° a radius of 32 millimeters is needed for the linear inverted magnification functions. The part actually used in the cortex is smaller. This is because the receptive field center of a cell can be smaller $\alpha_{VF, \max}$ and still cover the maximum possible visual field angle. We can use a radius approximately 2 millimeters less and still cover the whole visual field. If we assume that the striate cortex has a circular shape and that on average the visual field angle for both monkey and human is about 40° (see Figure 3.8) then we need approximately 2800 mm^2 of cortical area. This is surprising, because according to Daniel and Whitteridge [22] the primary visual cortex of a macaque monkey has a surface of about $1300\text{-}1400 \text{ mm}^2$.

3.6.3 Eccentricity dependent receptive fields

In Figure 3.2, we illustrated that the receptive field centers of the M and P cells grow linearly with eccentricity. This implies that small receptive fields appear in the fovea, where we dis-

tinguish with high detail. With increasing eccentricity, information is less and less detailed. An illustration of this decrease of detail by blurring is illustrated by Beaudot et al. [14].

Let receptive field angle ρ be related to the linear inverted magnification factor A :

$$\rho = cA(\alpha), \quad (3.34)$$

where c is a constant. Note that the receptive field radius can be obtained from ρ by using (3.28) and that ρ grows linearly with eccentricity.

Hubel [88] described that roughly a two millimeter movement in the cortex is needed to displace fields from one region to an entirely new region. We may conclude that, if one moves two millimeters in the cortex that the receptive fields do not overlap with each other. Let us use assume that all cells in the cortex which are at a two millimeter distance from each other have no receptive field overlap but border on each other then we can determine the constant c as follows:

$$\alpha(n) + cA(\alpha(n)) = \alpha(n+2) - cA(\alpha(n+2)) \quad (3.35)$$

$$\equiv c = \frac{\alpha(n+2) - \alpha(n)}{A(\alpha(n)) + A(\alpha(n+2))} = \frac{d+2}{(d+1)^2 + 1} \quad (3.36)$$

$$\equiv \{d = 0.11\}$$

$$c \approx 0.945.$$

Equation (3.35) is illustrated in the left part of Figure 3.12. With constant c receptive field angle ρ is known but the relation between two receptive fields which have the same visual field angle α is still not defined. So the last step for covering the whole visual field is the relation between two adjacent receptive fields who have both the same receptive field angle α but a different position in the visual field, which is illustrated in the right part of Figure 3.12. For this relation again a two millimeter movement is used to be in an entire new field region. We assume that two adjacent receptive fields with an equal visual field angle also have a two millimeter position difference in the cortex.

Angle ϕ (Figure 3.12), which is needed to cover the visual field for a constant visual field angle α given by

$$\phi = \arcsin\left(\frac{\rho}{\alpha}\right). \quad (3.37)$$

The number $N(\alpha)$ of receptive fields for a given constant visual field angle α is:

$$N(\alpha) = \left\lfloor \frac{2\pi}{\phi} + \frac{1}{2} \right\rfloor. \quad (3.38)$$

From the properties described in the previous sections, a cell whose receptive field size depends on the eccentricity can be modeled. An eccentricity dependent receptive field is as follows:

$$L_0^R(x, y) = \int_{x_1=-r}^r \int_{y_1=-r}^r I(x_1, y_1) R_{r,\theta}(x - x_1, y - y_1) dx_1 dy_1, \quad (3.39)$$

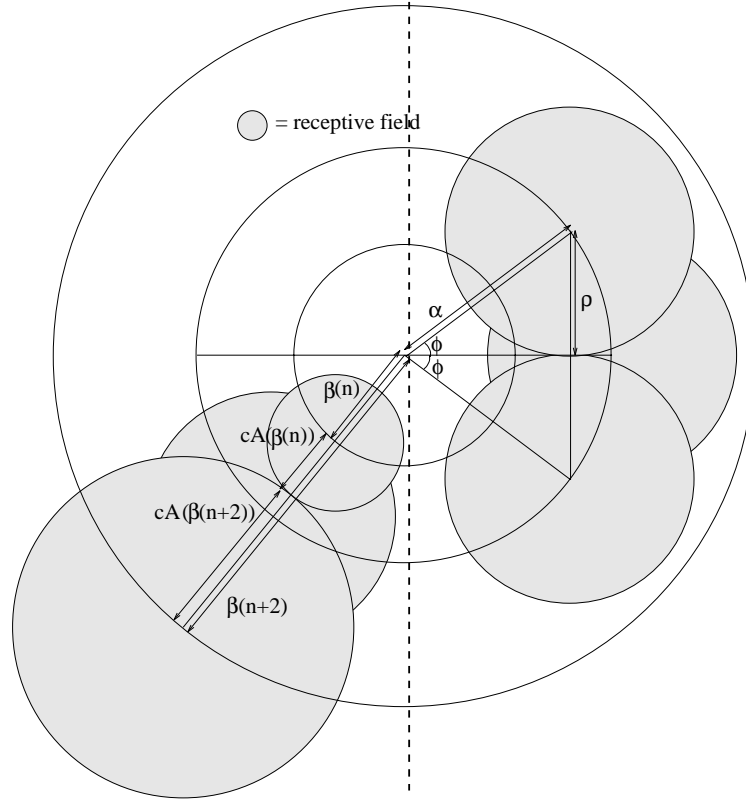


Figure 3.12: The receptive fields are represented by grey discs. By using the neighboring receptive fields in the cortex with a radius of n , $n+1$, and $n+2$ millimeters, the maximum receptive field angle $cA(\alpha(n))$ can be calculated. With the maximum receptive field is meant that two receptive fields are adjacent when their centers are at a 2 millimeter distance in the cortex (left part). With a constant visual field angle α and the receptive field angle ρ , ϕ can be calculated (right part). Note that we visualized radii but depicted the corresponding angles.

where r is the radius of the receptive field, I is a two-dimensional visual field (image) and R is one of the center-surround, simple, or complex receptive field profiles.

By using the L_θ^R -operator, a cortical response map is obtained. We apply this operator to a natural input image in Figure 3.13a. Also for this operator we validate the negative output by using a tuple. A tuple is illustrated by mapping the output of one cell in a range from gray (zero output) to white (strong response) and the other cell from gray (again zero output) to black. The results of the eccentricity dependent operator are illustrated in Figure 3.13c-d. The difference between the two is due to the use of a different magnification factor. In Figure 3.13c we use the linear increasing magnification factor (3.31) and in Figure 3.13d the logarithmic factor (3.30). For better visualization we enlarged the results obtained with the linearly increasing factor (Figure 3.13b).

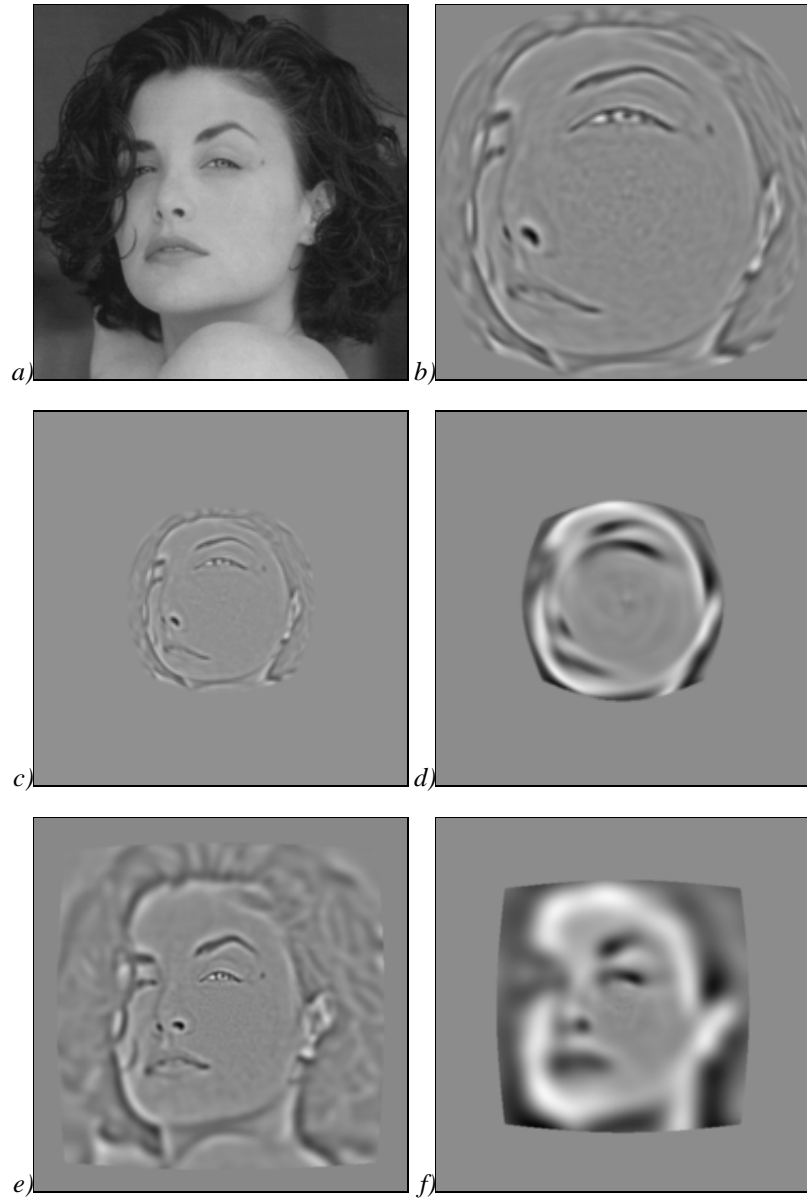


Figure 3.13: Response maps obtained with the L^M -operator where M is a Mexican-hat function. *a)* A natural input image. The center of the visual field is a slightly below the right eye in the image. *c)-d)* Response maps of the striate cortex area are represented as a two-dimensional grid. In fact, this grid should be folded over the pear shaped striate cortex. A linear and logarithmic inverted magnification factor are used, respectively. *b)* An enlargement of *c)*. *e)-f)* Responses of a non-distorted map, for a linear and logarithmic inverted magnification function, respectively.

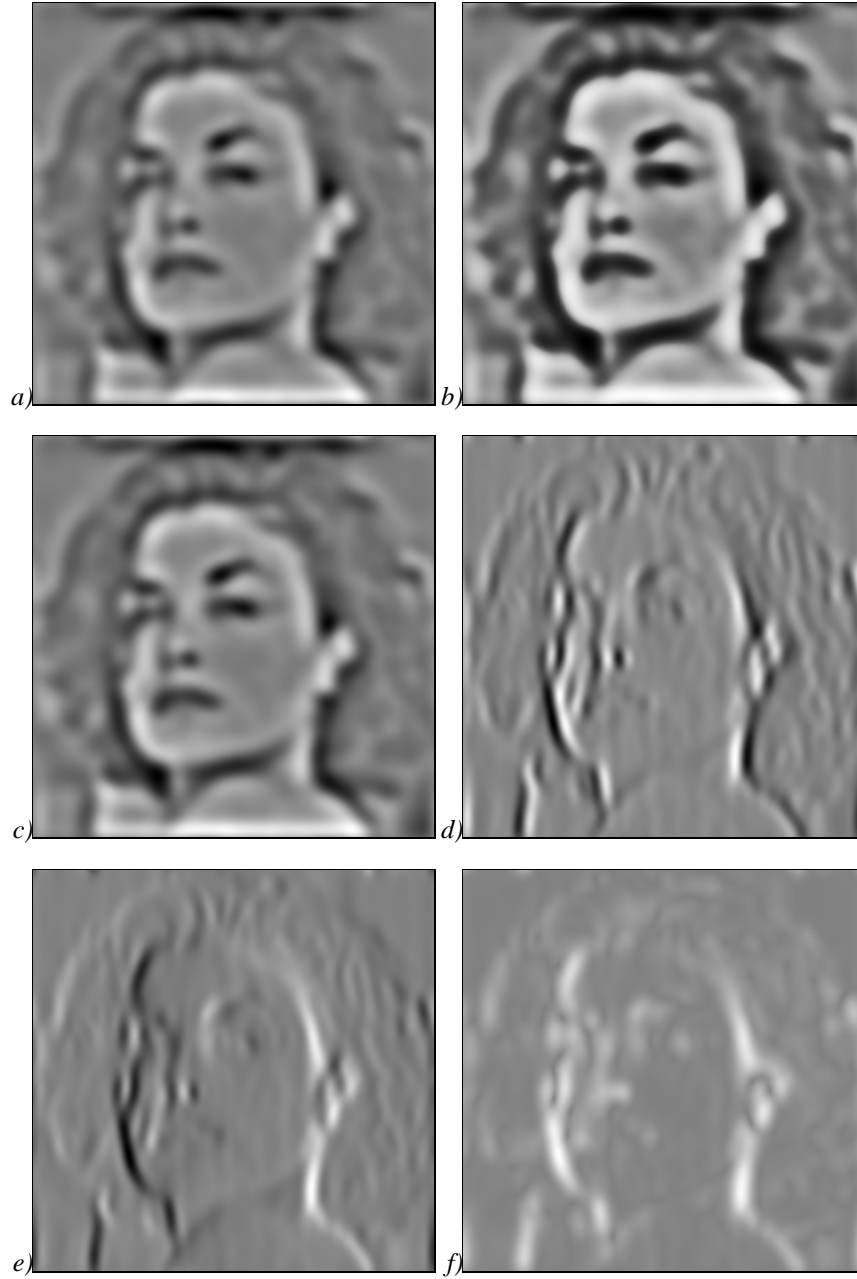


Figure 3.14: Response maps of different operators applied to the input image of Figure 3.13a. a)-f) Results obtained with M_{σ^-} , N_{σ^-} , P_{σ^-} , $S_{\sigma,\theta}^s$, $S_{\sigma,\theta}^a$, and $C_{\sigma,\theta}$ -operator, respectively. The used parameters are: $\sigma = 5$ and $\theta = 0$.

The retinal map is not distorted because of the ordering of the ganglion cells. Therefore, the ganglion cell density decreases with retinal eccentricity (van Buren [18] and Stone [177]). Because the receptive fields of retinal ganglion cells are small in the center of the visual field, detailed information of the visual field will be obtained here. As receptive fields grow linearly with eccentricity, the information gets less detailed and this means that the two-dimensional response map of ganglion cells will be blurred more and more as one moves away from the center of the visual field. This map is illustrated in Figure 3.13e-f. We used the same magnification factors to illustrate the difference between a distorted and a non-distorted map.

Layer 4C of the monkey striate cortex contains concentric center-surround cells, but these cells are ordered on a pear shaped map (Tootell et al. [185]). This distorted map will give a completely different view (Figure 3.13c-d) compared to the resulting maps from the retinal ganglion cells or cells in the LGN. The distorted map uses in comparison to the ganglion cell map a large surface for the center, where details are visible. Note that the image which is used as input stimulus has a size of 256×256 pixels, which gives a maximum visual field angle of 7.9 degrees. The angle is obtained by using equation (3.28), where $\alpha_k = 22.6^\circ$ is the horizontal visual angle of the used camera lens, which results in a row r_k of 768 pixels.

3.6.4 Results of other operators

Figure 3.14a gives the response map of the M_σ -operator (3.6). The output of the M_σ - and P_σ -operators are about the same (Figure 3.14a,c), which is expected since the response of the P cells increase in an almost linear way with increasing contrast. The output of the N_σ -operator (3.7) responds strongly to differences in contrast and differs in this sense clearly from the P_σ -operator (3.8). We should emphasize that the P cells most probably serve another task, since they are wavelength sensitive and their receptive fields center radii are about four times smaller than the M cells.

The outputs of the $S_{\sigma,\theta}^s$ - and $S_{\sigma,\theta}^a$ -operator (3.17) and (3.18), illustrated in Figure 3.14d-e, both respond well to a preferred orientation, but the latter can also discriminate between a down- or up-going edge. Finally, the $C_{\sigma,\theta}$ -operator (3.20) responds to edges of a preferred orientation at the proper location (Figure 3.14f).

3.7 Summary and discussion

In this chapter several different cells in early vision have been modeled with respect to their spatial filtering properties, using the building block architecture proposed by Hubel. We explained the relation between receptive field and dendritic tree of a ganglion cell. The receptive field profiles of center-surround cells have been modeled with a Mexican-hat function, but can also be modeled with a difference of two Gaussians. The difference in contrast response in monkey M and P cells has been modeled. There is a difference of about a factor four in receptive field size between these two types. The wavelength selectivity of the P cells is not considered here, but is elaborated in Chapter 7.

The receptive field profiles of simple cells can be fitted well by Gabor functions. In the following chapters we will use these functions, although the receptive field profiles of sim-

ple cells can be approximated well by differences of Gaussians too. In several models the simple cells get their inputs from the center-surround cells, e.g. the model used by Atick and Redlich [6]. The response of simple cells is modeled by convolving Gabor functions. For computational reasons we did not use the output of the center-surround cells as input for the simple cells. Several models for the simple cells have been proposed and the degree of linearity exhibited has been debated by many authors. In the “emergent” model of Somers et al. [176], synaptic potentials add and subtract linearly. This model also contains three sources of non-linearity: action potential thresholds, voltage dependent driving forces on synaptic potentials, and cortical amplification of responses. The “normalization” model of Heeger [67, 68] employs strongly oriented LGN inputs with no cortical excitation. It has three forms of non-linearity: rectifying threshold, normalizing (divisive inhibition), and squaring of output responses. Our choice for the linear model is based on two points: it is widely used and most simple cells show a linear behavior with respect to spatial summation.

The complex cells get their inputs from the simple cells. We use the “local energy” $C_{\sigma,\theta}$ -operator (3.20) because of its smoothness. The $C_{\sigma,\theta}$ -operator enhances lines and edges with a certain orientation at the proper location. This makes the operator suitable for line enhancement. The operator corresponds to the complex cell model suggested by Pollen and Ronner [153] and is in agreement with the observation that complex cells usually exhibit no phase dependence in response to gratings (Maffei and Fiorentini [123] and Movshon et al. [137]). The choice for this operator is that it models a variety (but not all) of different types of cells in the complex category and that it is computationally inexpensive. An additional complex cell-based operator has been defined and called the principal orientation operator. It will be used in Chapter 6 to facilitate the line detection mechanism.

We introduced two types of multiple scales, a scale space with linearly increasing receptive fields r and a non-linear one (which is linear in frequency domain). Multiple scales are used to obtain better results. E.g., in Chapter 5 the results obtained with the multiple scale corner operator are better than the results with the same operator at a single scale. Also a multiple scale C -operator is used to get better edge enhancement as compared with the results at a single scale.

We modeled receptive fields growing linearly in size with retinal eccentricity and also the distorted map between visual field and cortex. Experimentally a distorted map in a macaque monkey was determined by Tootell et al. [185]. Models of this so-called “spatial mapping” and “retinoptic mapping” are made by e.g. Schwartz [168] and Mallot et al. [124, 125, 126, 127].

We did not include the growing receptive fields with eccentricity and the distorted mapping. Growing receptive fields indicate that there is a reduction of information. When motion is involved this reduction is desired since it makes the system faster. If spatio-temporal properties are involved we can use the movement of the eyes to get the illusion that the visual scene is perceived sharply. A typical application where a reduction of information can be desired is a moving robot which rapidly moves its “eyes” from one point of interest to another. The distorted map is not used because it will be more difficult to compare two identical objects which are at different positions in the visual field. This map can become of real interest if we know what kind of location independent features are extracted.

4

Localization Properties of Cortical Corner Detectors

Jesus said to them, “Have you never read in the Scriptures: ”‘The stone the builders rejected has become the cornerstone; the Lord has done this, and it is marvelous in our eyes’?

Matthew 21:42 (NIV)

THE USE OF THE TERM “CORNER” in the vision literature is not uniform. We consider a *junction* to be a point where two or more differently oriented line segments (or edges) end. A *corner* is a junction where one of the angles between the line segments is larger than 180 degrees. With larger we mean that a human must be able to see two different line segments and not a straight line.¹

Corners are important features because they provide useful segmentation clues, and their robustness with respect to changes in perspective or small distortions in objects makes them useful for object recognition. Matching corners is fast and easy compared to matching line segments. It is fast, not only because the number of corners is generally less than the number of line segments in a discrete image, but also because a corner contains only one pixel (in a discrete image) and a line segment at least one (but usually a lot more). Matching a single point to another point is easy compared to a line segment since the latter has length and orientation. Also the direction of movement of a single straight edge through an aperture is ambiguous since the perceived movement is orthogonal to the edge. This ambiguity of the direction of motion is known as the *aperture problem*. The direction of movement of a corner is not ambiguous and therefore avoids this problem.

The robustness and speed of corner matching can be improved by adding specific feature attributes to the corners (Rosin [165]). From the definition of a corner we know that at least two line segments start from (or end at) the corner. This means that we can use the number of outgoing line segments and the *angle* between two neighboring line segments as feature

¹This angle is very small (hyperacuity).

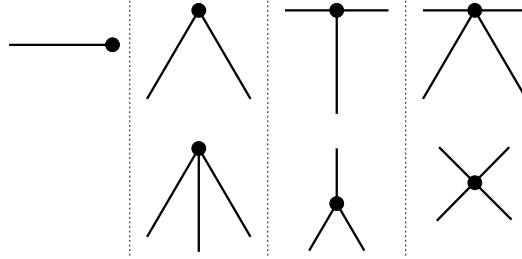


Figure 4.1: Different junction types. The left type is called the *I-junction*. It is not a real junction since there is only one line segment but can be interpreted as a junction with zero degree angle. The second column are the corner junctions types, since there is one angle between two neighboring line segments which is larger than 180 degrees. The third column contains junctions with 3 line segments or more brief 3-junctions, illustrated are the T- and Y-junctions. In the rightmost column the K- and X-junctions are shown, they are 4-junctions.

attributes for a corner. Also the *sharpness* of the corner (sharp, round, or blunt), its *size* by which is meant the range of scales over which a corner exists,² and the *colors* or *grey-values* of foreground and background are corner attributes. Finally the *junction type*, as illustrated in Figure 4.1, is used as extra information about the corner (e.g. V, Y, \downarrow , T, K, or X).

Our method for detecting corners will yield position, sharpness and size (Chapter 5), color and contrast (Chapter 7). The junctions for a corner are limited to the V, \downarrow , or I (line-end) types because our definition of corner does not allow other mentioned junction types. Once line segments that end at a corner point are detected, the angle between neighboring line segments and orientation of the corner can be derived too.

It is clear that corners also play an important role in human perception. A triangle, e.g., can be constructed easily when 3 corner points are given which are not collinear. A perceptual construction of a triangle from three non-connected curves (or line segments) which are part of its outline is much harder because the corner information (sharpness) is missing. Since we are able to detect corners very accurately, the choice for a biologically motivated corner operator is a logical one. Our choice is to use the response of *end-stopped cells*, found in monkeys (Hubel [89]). Peterhans et al. [149, 190] studied the role of end-stopped cells in mechanisms of figure-ground segregation and found that these cells respond well to terminations (line-ends and corners) and to T-junctions, but weakly or not at all to X-junctions. Two types of end-stopped cells are found. The so-called *single end-stopped cells*, which have receptive field profiles with one inhibitory and one excitatory lobe, and the *double end-stopped cells*, which have one excitatory lobe and two inhibitory lobes. Examples of both types of cells are given by Zeki [204] and Hubel [89], respectively.

In this chapter an operator which models the functionality of end-stopped cells, proposed by Heitger et al. [71] is used as corner detector. The strongest responses of end-stopped cells for different receptive field sizes occur at corners and line-ends. The position of the strongest response is scale invariant and robust under rotations of the visual input stimulus.

²For example, the curvature of the corner is scale dependent.

In Section 4.1 the responses of end-stopped cells are modeled. Two types of end-stopped cells are modeled: the single and double end-stopped cells. Both receive excitatory and inhibitory responses from complex cells. The used operator is not sufficient in the two-dimensional case, i.e. it gives false responses, and therefore an additional inhibition mechanism is added. Sections 4.2 and 4.3 elaborate on the position invariance and rotation robustness of these operators. In Section 4.4 a number of standard corner operators are implemented and compared with the end-stopped operator. The chapter ends with a summary and discussion.

4.1 End-stopped cells

End-stopped cells that receive their input from the complex cells show the following behavior: when a longer line segment is presented the response increases up to a certain limit and after this limit is exceeded the response decreases. This indicates that end-stopped cells respond best to short line segments, line-ends or corners. Since end-stopped cells respond well to corners, we will check the response properties of modeled end-stopped cells at different types of corners. The end-stopped cells are found in layer 2 and 3 of monkey area V1 and also in area V2 (Peterhans and von der Heydt [150]). A schematic illustration of both types of end-stopped cells is given in Figure 2.6. The single end-stopped cells respond well to line-ends. The double end-stopped cells respond best to very short line-segments or small spots, small circular objects, blobs, or sharp bends of contours.

4.1.1 Responses of end-stopped cells

Heitger et al. [71] proposed a model for the two types of end-stopped cells. Since their model shows responses at the line-ends and corners where we expect them, we will follow their model. They used the $C_{\sigma,\theta}$ -operator defined in (3.20) to model the responses of the complex cells. The $C_{\sigma,\theta}$ -operator localizes intensity discontinuities, but its output does not carry information about their type (edge or line) and polarity (light or dark), as is illustrated in Figure 4.2d.

The responses of simple and complex cells to a long straight line are constant along the line, and can therefore be characterized completely by their variation in the orthogonal direction. They do not provide an explicit representation of terminations of an edge or line nor of deviations from straightness (e.g. corners). The end-stopped cells will respond only to line terminations and to corners. The definition of a line termination is obviously the point where the line ends, but a proper definition for a corner is not trivial. Our definition excludes T-, Y-, K-, or X-junctions, since there should be an angle between two neighboring line segments which is larger than 180 degrees. With a *rounded corner* we mean that two straight line segments with different orientations are connected to each other by a curved segment. A highly rounded corner is connected by a small part of an ellipse since their line segments end close to each other but not exactly at the same position as the line segments at “standard” corners do.

A combination of output responses of complex cells is used as input for the modeled end-

stopped cells. Our notation differs slightly from Heitger et al. [71] in when thresholding is applied. The response of single end-stopped cells, is modeled as follows:

$$\bar{E}_{\sigma,\theta}^s(x,y) = C_{\sigma,\theta}(x+d\sigma\sin\theta, y-d\sigma\cos\theta) - C_{\sigma,\theta}(x-d\sigma\sin\theta, y+d\sigma\cos\theta), \quad (4.1)$$

where $\theta \in [0, 2\pi)$ is the orientation and d a positive constant. The constant d represents the distance between the excitatory and inhibitory part of the receptive fields of the end-stopped cells relative to the size of the receptive field. Obviously, it cannot be chosen arbitrarily (Figure 4.2l). If for example a large d is chosen, the line segment (or edge) will only be eliminated in the middle and two shorter line segments will remain. When d is chosen too small a corner will be detected at the wrong position. The correct position or *exact locations* of corners we determined manually. Every response of the modeled end-stopped cells other than at these corners is called a *false response*.

The double end-stopped or $\bar{E}_{\sigma,\theta}^d$ -operator which models the response of double end-stopped cells approximates the second derivative of the $C_{\sigma,\theta}$ -operator in the direction orthogonal to θ and is as follows:

$$\bar{E}_{\sigma,\theta}^d(x,y) = C_{\sigma,\theta}(x,y) - 0.5C_{\sigma,\theta}(x+2d\sigma\sin\theta, y-2d\sigma\cos\theta) - 0.5C_{\sigma,\theta}(x-2d\sigma\sin\theta, y+2d\sigma\cos\theta). \quad (4.2)$$

The $\bar{E}_{\sigma,\theta}^s$ -operator, which models the receptive field profile of a single end-stopped cell is an approximation of the first derivative of the $C_{\sigma,\theta}$ -operator in the direction orthogonal to θ . As illustrated in Figure 4.2d the strongest responses of the $C_{\sigma,\theta}$ -operator for orientation $\theta = 0$ occur at the vertical edges. Since these responses are not at the positions of the corners, they must be suppressed by the $\bar{E}_{\sigma,\theta}^s$ - and $\bar{E}_{\sigma,\theta}^d$ -operators.³ Therefore not position (x,y) is taken but positions at a distance dependent on parameter d , orthogonal to orientation θ of the modeled complex cell.

The performance of the end-stopped operators strongly depends on the choice of the distance parameter d . To get the strongest responses at the exact positions of the corners, as is illustrated in Figure 4.2k and 4.2l, it becomes clear that the choice of d depends on the exact location of the corner and the position of strongest response of the $C_{\sigma,\theta}$ -operator (which is most close to the exact position of the corner). If the building block architecture is used, the end-stopped cells receive input from complex cells and complex cells in turn receive their input from simple cells. We then can conclude that the distance depends on the receptive field size of the simple cells. It follows from the arbitrarily chosen receptive field size ($\sigma = \sqrt{5}$) and the corresponding end-stopped distance $d\sigma = 4$ that

$$d = \frac{4}{\sqrt{5}} \approx 1.8. \quad (4.3)$$

With this choice of d the $\bar{E}_{\sigma,\theta}^s$ - and $\bar{E}_{\sigma,\theta}^d$ -operators gave the strongest response at the manually marked positions of the corners.

³Equations (4.1) and (4.2) are similar to equations (6) and (7) of Heitger et al. [71].

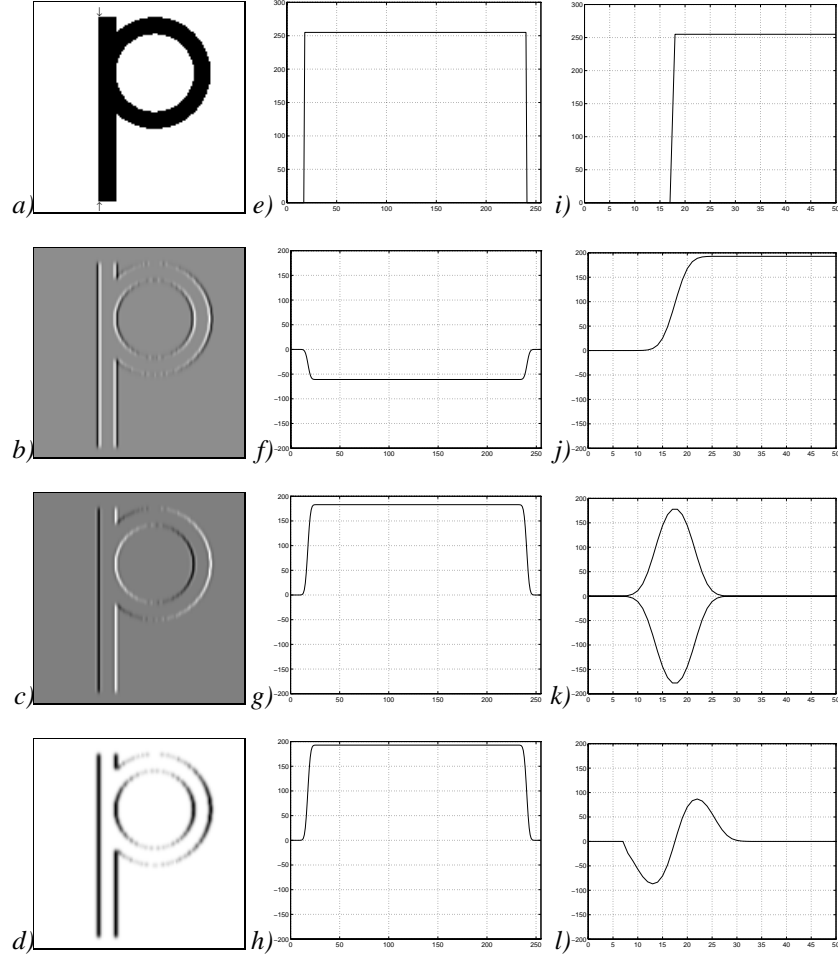


Figure 4.2: Responses of different types of modeled cells. *a)* The character *P* is used as input stimulus. *b)-c)* The output of the $S_{\sigma,\theta}^s$ and $S_{\sigma,\theta}^u$ -operators, respectively. *d)* The output of the $C_{\sigma,\theta}$ -operator. Black denotes a positive output and white zero output in *a* and *d*. Black, gray, and white represent positive, zero, and negative output, respectively, in *b* and *c*. The parameters used for the Gabor functions (3.10) and (3.11) are $\sigma = \sqrt{5}$ and $\theta = 0$ and the discrete image contains 256x256 pixels. *e)* The intensity profile of the vertical slice between the arrows at the left most side of *P* in *a*, which is just in the black part of the character *P*. *f)-h)* The output of the operators for the positions on the vertical slice. *i)* An enlargement of *e* but only part of it is shown. It starts at the top of *a* and ends 32 pixels below the left upper corner of character *P*. *j)* Part of *h*. The strongest responses of the $C_{\sigma,\theta}$ -operator are not exactly at the corner of *P* but shifted downward (from position 18 to position 25). *k)* The strongest response of the $E_{\sigma,\theta}^s$ -operator is at the correct position, since these operators are anti-symmetrical. We used orientations $\theta = 0$ and $\theta = \pi$ together to be sure to always have one positive response. *l)* The strongest responses of the $E_{\sigma,\theta}^u$ -operator are shifted 4 positions downward. This is exactly the value $d\sigma$ used in the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^u$ -operators.

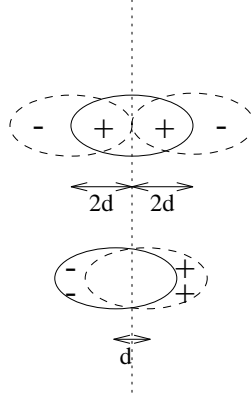


Figure 4.3: The $E_{\sigma,\theta}^d$ -operator is constructed by three displaced $C_{\sigma,\theta}$ -operators with preferred orientation $\theta = 0.5\pi$. The excitatory part in the center is twice as strong as the two inhibitory parts. Below, the $E_{\sigma,\theta}^s$ -operator is constructed by two $C_{\sigma,\theta}$ -operators, the excitatory part shifted to the right and inhibitory part to the left. The performance of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators is dependent on displacement d of the $C_{\sigma,\theta}$ -operators; when d is chosen using (4.3) the receptive fields of the modeled complex cells ($C_{\sigma,\theta}$ -operator) have a considerable overlap.

To validate why we need two different types of end-stopped cells, we use the outputs of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators for a single orientation θ , because the responses of the operators can be illustrated more accurately in the one-dimensional case. The one-dimensional $E_{\sigma,\theta}^s$ -operator can be obtained from (4.1) by substituting the orientation $\theta = \frac{\pi}{2}$. The 1-D operator is then as follows: $E_{\sigma}^s(x) = C_{\sigma}(x + d\sigma) - C_{\sigma}(x - d\sigma)$. The second E_{σ}^s -operator, with $\theta = \frac{3\pi}{2}$, gives an identical result as for $\theta = \frac{\pi}{2}$, but has opposite sign. The results of both E_{σ}^s -operators are shown together in Figure 4.2k. It gives the desired maximum responses at edges, this in contrast to the response of E_{σ}^d -operator, which show a maximum response (Figure 4.2l) at a position which is shifted over a distance $d\sigma$, at the same edge. The E_{σ}^d -operator will give a maximum response at the proper position if a line segment fits entirely in the excitatory part of the receptive field. That means that the operator responds properly to thin line segments, spots, blobs, and small circularly shaped objects. However, on blob input stimuli the E_{σ}^s -operators give false responses (Heitger et al. [71]). This validates that both types of end-stopped cells are needed to get a strong response on the proper location. The shifts of the strongest responses in the one-dimensional case can be compensated by using the amplitude $E_{\sigma}^u = \sqrt{(E_{\sigma}^s)^2 + (E_{\sigma}^d)^2}$ of the two end-stopped operators. The receptive fields of complex cells (3.20) are modeled by taking the amplitude of the symmetrical and anti-symmetrical simple operator, with the same purpose: to have the strongest responses at the exact positions of the edges and line segments. Figure 4.2f shows a negative output of the $S_{\sigma,\theta}^s$ -operator, where one would desire positive output, which is the case in Figure 4.2g for the $S_{\sigma,\theta}^d$ -operator. Thus the positive response of the $S_{\sigma,\theta}^s$ -operator in Figure 4.2b is shifted when the bar is wide. If the bar had been smaller (i.e. a line) then an edge going up and down would have fitted

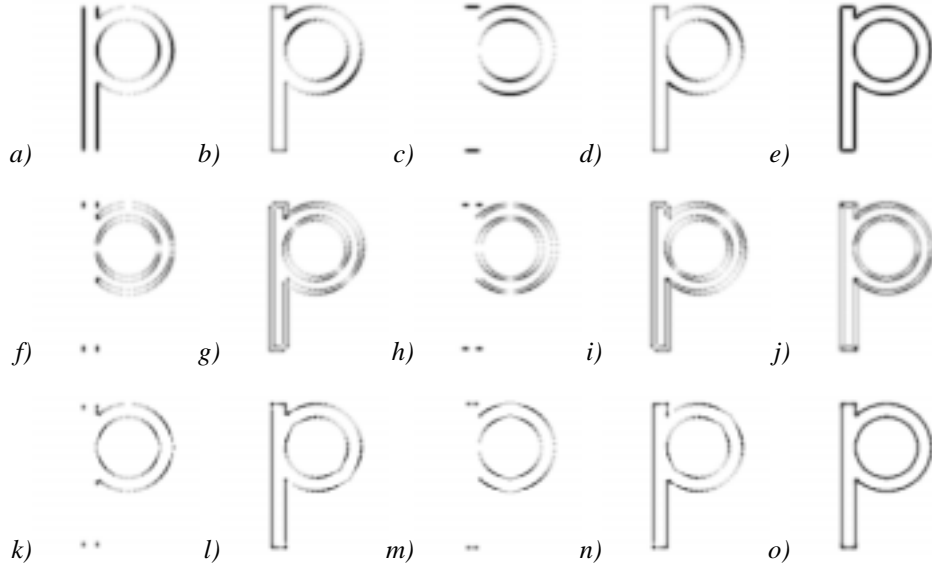


Figure 4.4: Responses of $C_{\sigma,\theta}$ -, $E_{\sigma,\theta}^s$ -, and $E_{\sigma,\theta}^d$ -operators for different orientations at input stimulus character P from Figure 4.2a. a-d) The outputs of the $C_{\sigma,\theta}$ -operators for orientations $\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}$, and $\frac{3\pi}{4}$, respectively. e) The output of all four orientations combined with the max-operator. f-i) The output of the $E_{\sigma,\theta}^s$ -operators for the same orientations. j) The output of the four different orientations combined with the max-operator. k-o) The outputs of the $E_{\sigma,\theta}^d$ -operators again for the same orientations and combined with the max-operator. We used $\sigma = \sqrt{5}$ for the $S_{\sigma,\theta}^s$ - and $S_{\sigma,\theta}^a$ -operators defined in (3.10) and (3.11), and for the $E_{\sigma,\theta}^s$ -operator (4.1) and $E_{\sigma,\theta}^d$ -operator (4.2).

into one receptive field of a symmetrical simple cell ($S_{\sigma,\theta}^s$ -operator) and would have had the strongest response on the proper place. In such a case, when a bar is so thin that its width fits in the receptive field of the modeled simple cells, the $S_{\sigma,\theta}^a$ -operator responds at the wrong location.

4.1.2 False response elimination

From our test image (Figure 4.2a) we took a vertical slice and illustrated that the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators responded only at and near the position of a line-end (Figure 4.2k-l). Since the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators have a preferred orientation θ , the response will be inhibited by one or two inhibitory lobes orthogonal to θ . If Figure 4.2d (which is identical with Figure 4.4a) is considered, we notice that the responses of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators (Figure 4.4f and k) differ considerably from the response of the $C_{\sigma,\theta}$ -operators at the same receptive field center at the vertical edges of P. The responses of the $C_{\sigma,\theta}$ -operator are strong also at the circular part of P, since these responses are not at the optimum vertical orientation the inhibitory lobes of

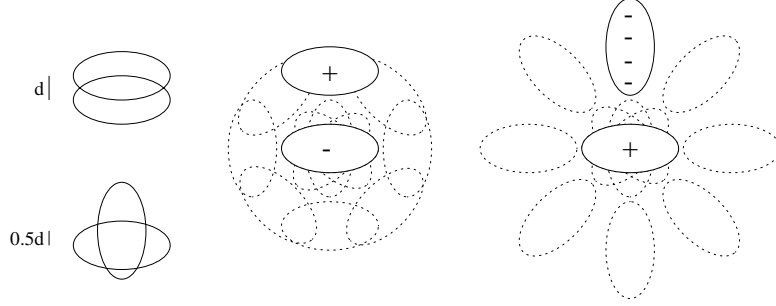


Figure 4.5: Two inhibition operators. The I_G^t -operator is illustrated in the middle and the I_G^r -operator at the right. On the left a more realistic distance is given between the excitatory and inhibitory parts of the two operators, the receptive fields give a considerable overlap. For better illustration we used a larger distance between the excitatory and inhibitory parts.

$\bar{E}_{\sigma,\theta}$ -operators do not suppress the responses at the circular part sufficiently. These responses are called false responses, since they are not desired and therefore need to be inhibited.

In the two-dimensional case the amplitude of the receptive fields of the two types of modeled end-stopped cells will not be sufficient to have no response at wrong positions. This has to do with the fact that every modeled complex cell with a certain preferred orientation which is near to the orientation of a line segment (or edge) also responds well, although the response of these cells is weaker than the response of the cells which have a preferred orientation identical to the orientation of the line segment. Figure 4.4a-d shows this phenomenon: a response at the horizontal edges in all orientations except $\theta = 0$ (Figure 4.4a). The modeled end-stopped cells will respond strongly at the edge with the preferred orientation ($\theta = \frac{\pi}{4}$) in the sense that the $\bar{E}_{\sigma,\theta}^s$ -operators respond strongly at the two positions which are exactly where the edges end.⁴ This is illustrated in Figure 4.4h where four black “dots” appear. The strong responses are shifted inwards for the $\bar{E}_{\sigma,\theta}^d$ -operators (Figure 4.4m). However, the responses of these two operators with preferred neighboring orientations will not be suppressed by the inhibitory lobes and false responses appear all over the edges as shown in Figure 4.4g and 4.4i for the $\bar{E}_{\sigma,\theta}^s$ -operator and in Figure 4.4l and 4.4n for the $\bar{E}_{\sigma,\theta}^d$ -operator.

It is most likely that end-stopped cells as they are found in the brain avoid the problem of false responses; either excitatory and inhibitory subunits are composed differently or additional inhibitory input eliminates the false responses. Heitger et al. [71] assume the latter. They solve the problem of false responses by adding a surround inhibition mechanism and propose two operators. The first operator gives the differences between peripheral and central activities within the same orientation channel and is therefore called tangential inhibition or I_G^t -operator. This operator is chosen in such a way that its responses are parallel to the edges, but there are, however, no responses where the $C_{\sigma,\theta}$ -operators respond well, since the response is strongly inhibited by this operator. An illustration of the output of the operator applied to the P-image is in Figure 4.6a. The second operator is the radial inhibition or I_G^r -operator, since it gives the difference between two orthogonal orientation channels.

⁴The strength of the responses is coded in gray level, white denotes no response and black strong response.

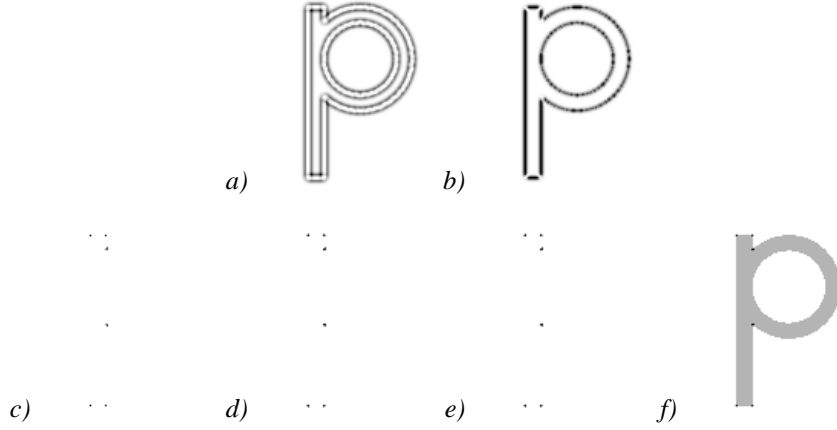


Figure 4.6: a)-b) The results of the surround inhibition I_G^l - and I_G^r -operators applied to the P-image. c)-d) The output of the E_G^s - and E_G^d -operators (4.11) and (4.12), respectively. For both operators we used the max-operator to combine different orientations of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators defined in (4.6) and (4.7). e) The output of the E_G -operator (4.14). f) The results of e combined with the input image. For visualization character P has a gray color. The parameters are: $\sigma = \sqrt{5}$, $w_r = 4$, $w_l = 1$, $g = 2$, and $N = 8$.

This operator responds strongly where the $C_{\sigma,\theta}$ -operator responds strongly, but it does not respond at corner coordinates since it is inhibited here by the $C_{\sigma,\theta+\pi}$ -operator. An illustration of the operator is given in Figure 4.6b. For both operators at every orientation a pair of $C_{\sigma,\theta}$ -operators is used which are first multiplied with a weighting factor, then subtracted and thresholded at zero. The operators defined below are slightly different from the two proposed by Heitger et al. [71]. In the I_G^l -operator the weights $+1$ and $-w_l$ are swapped, we multiply the first $C_{\sigma,\theta}$ -operator with $-w_l$ and the second with $+1$. In the I_G^r -operator we used $0.5d\sigma$ in the second $C_{\sigma,\theta}$ -operator as distance where Heitger et al. used d . The operators are as follows:

$$I_G^l(x, y) = \sum_{i=0}^{2N-1} \left[-w_l C_{\sigma,\theta_{i \bmod N}}(x, y) + C_{\sigma,\theta_{i \bmod N}}(x + d\sigma \cos \theta_i, y + d\sigma \sin \theta_i) \right] \geq 0 \quad (4.4)$$

and

$$I_G^r(x, y) = \sum_{i=0}^{2N-1} \left[C_{\sigma,\theta_{i \bmod N}}(x, y) - w_r C_{\sigma,\theta_{(i+\frac{N}{2}) \bmod N}}(x + 0.5d\sigma \cos \theta_i, y + 0.5d\sigma \sin \theta_i) \right] \geq 0, \quad (4.5)$$

where N is the number of orientations, $\theta_i = \frac{i\pi}{N}$, and w_l and w_r are constants. The weights should be chosen properly, otherwise they do not suppress all false responses. The number of orientations should always be even, otherwise $\frac{N}{2}$ in (4.5) is not an integer. The weights $w_l = 1$ and $w_r = 4$ are found empirically, and seem to be independent of input stimulus,

$\varsigma = \sigma^2$	N=4	N=6	N=8	N=10
1.25	[1.2, 1.4]		[1.6, 1.8]	
2.50	≥ 2.2		≥ 1.0	
5.00	≥ 1.5	≥ 0.8	≥ 0.6	≥ 0.5
10.00	≥ 1.2		≥ 0.6	≥ 0.5
20.00	≥ 1.2	≥ 0.8	≥ 0.6	≥ 0.5
30.00	≥ 1.2		≥ 0.6	

Table 4.1: The interval for a proper gain factor g for different σ and different number of orientations N .

receptive field size, and number of orientations. Both inhibition operators are illustrated in Figure 4.5, where $N = 4$ orientations are used.

Figure 4.2h and 4.2j illustrated that the responses of the $C_{\sigma,\theta}$ -operators at the manually marked position of the corner is about half of the response on the edge. This makes it suitable for an inhibition mechanism because it will not much affect the responses at line-ends.

The I_{σ}^l -operator suppresses all responses parallel and adjacent to the edge if we move along the strongest responses of the $C_{\sigma,\theta}$ -operator. This operator therefore inhibits all false responses of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators from neighboring directions. The I_{σ}^r -operator inhibits all responses where the $C_{\sigma,\theta}$ -operators respond well, except at those positions where the operators with orthogonal orientation respond well.

The $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators inhibited by the I_{σ}^l - and I_{σ}^r -operators are as follows:

$$E_{\sigma,\theta_i}^s = \left[\left[E_{\sigma,\theta_i}^s \right]^{\geq 0} - g(I_{\sigma}^l + I_{\sigma}^r) \right]^{\geq 0} \quad i = 0, \dots, 2N-1 \quad (4.6)$$

and

$$E_{\sigma,\theta_i}^d = \left[\left[E_{\sigma,\theta_i}^d \right]^{\geq 0} - g(I_{\sigma}^l + I_{\sigma}^r) \right]^{\geq 0} \quad i = 0, \dots, N-1, \quad (4.7)$$

where g is the gain factor.

Like the two weight factors w_l and w_r , also the gain factor g can be chosen independently from the receptive field size. In Table 4.1 the two smallest values for ς have a different gain factor, this is mainly caused by discretization of the coordinates in the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators. The coordinates $(x, y) \in \mathbb{Z}^2$ of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators depend on distance $d\sigma$. If this distance is small, the relative error of rounding a real number to an integer is much larger than it is with a larger distance. The gain factor is independent of the number of orientations. The notation “ $\geq x$ ” in the table means that g should be at least x to avoid false responses. Note that the maximum gain factor we tried is $g = 25$, so formally $\geq x$ means that $g \in [x, 25]$, since we did not test for larger g .

The minimum gain factor which still avoids false responses becomes lower if the number of orientations increases. This has to do with the inhibition I_{σ}^l - and I_{σ}^r -operators because their response is the sum over an excitatory and an inhibitory modeled complex cell over all

orientations. A possibility for having a constant minimum suitable g independent from the number of orientations N is to take the maximum or average of all orientations instead of the sum in (4.4) and (4.5).

Table 4.1 indicates that g is independent of the number of orientations and receptive field size. We used a constant gain factor ($g = 2$) in all our model experiments.

4.2 Relational position invariance and rotation robustness of end-stopped cells

In this section we elaborate on the position invariance and rotation robustness of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators, because relative relations of corners must remain stable under translation and rotation. This stability of a corner is measured by checking how the responses of these operators change when a constant number of different orientations θ_i for the $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operators is used and the input image is rotated over an angle ϕ .

We start with the rotation of the Gabor function by adjusting θ , we do not rotate the input image, and use the results as reference material. Rotating the Gabor function has the advantage that this causes negligible discretization effects compared to the rotation of a two-dimensional image. In fact, rotating the function and taking the discrete image from a camera is comparable to taking a discrete image from a rotated camera and use the non-rotated Gabor function. We check the responses of the modeled complex and end-stopped cells, and expect a smooth decrease or increase in response when θ is adjusted.

Rotation of the input stimulus is trivial when the stimulus is continuous but is more difficult when it is discrete. We use a standard rotation algorithm and a rotation algorithm with bilinear interpolation. With both methods we checked the responses of the $C_{\sigma,\theta}$ -, $E_{\sigma,\theta}^s$ -, and $E_{\sigma,\theta}^d$ -operators.

4.2.1 Changing the preferred orientation of simple cells

Rotating a square discrete image gives loss of information due to discretization effects. Also, part of the background is lost because the image should keep its original size. Therefore we change the orientations θ_i in the Gabor functions. By rotating the function we choose different preferred orientations for the modeled simple cells. Rotation is possible since the following property of (3.17) and (3.18) in continuous coordinates holds:

$$\begin{aligned} S_{\sigma,\theta}(x, y) &= \int_{x_1} \int_{y_1} I_0(x_1, y_1) G_{\sigma,\theta}(x - x_1, y - y_1) dx_1 dy_1 \\ \equiv S'_{\sigma,\theta}(x_\phi, y_\phi) &= \int_{x_1} \int_{y_1} I_\phi(x_1, y_1) G_{\sigma,\theta+\phi}(x - x_1, y - y_1) dx_1 dy_1, \end{aligned} \quad (4.8)$$

where $(x_\phi, y_\phi) = (x \cos \phi + y \sin \phi, -x \sin \phi + y \cos \phi)$ and I_ϕ is the input image rotated around the center of the original image, i.e. the center of the discrete image has coordinates $(0, 0)$.

In (4.8) both input image I and Gabor function G are rotated by ϕ compared to $S_{\sigma,\theta}$. Intuitively it is clear that the responses of $S_{\sigma,\theta}(x, y)$ and $S'_{\sigma,\theta}(x_\phi, y_\phi)$ are identical. Note that

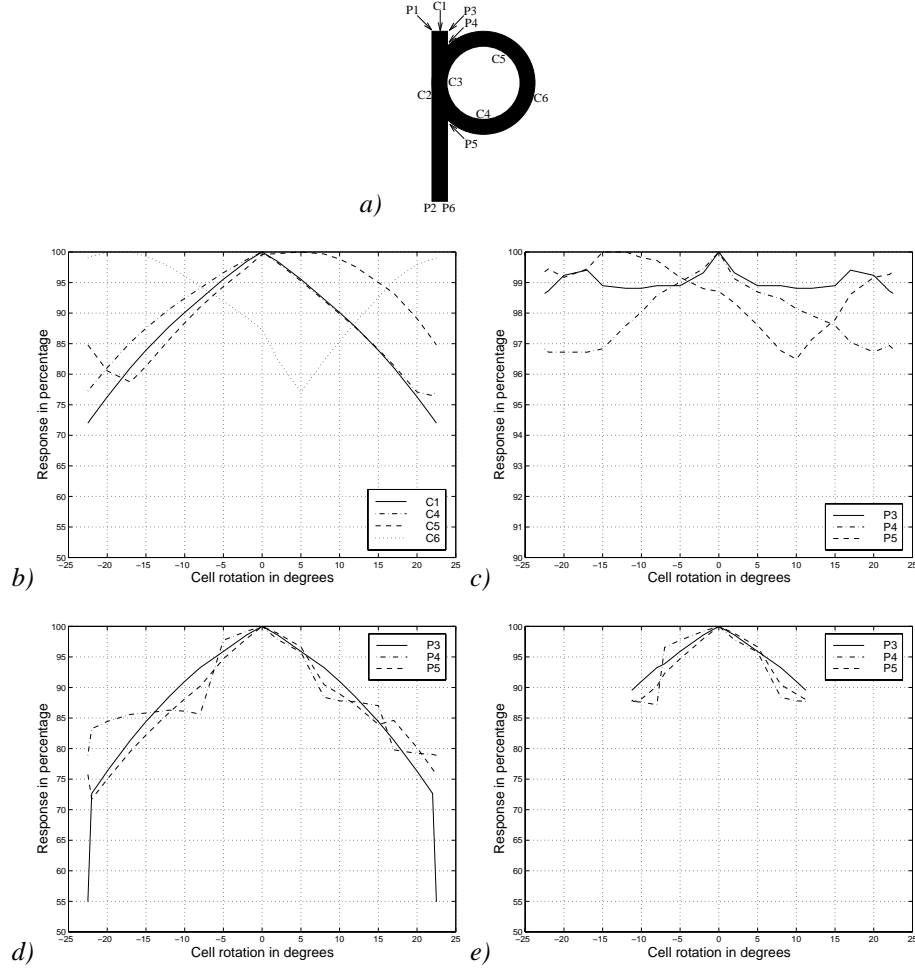


Figure 4.7: Responses upon cell rotation. a) Character *P* is again used as input image. Six corners are labeled P1 to P6. The other coordinates C1-C6 are chosen arbitrarily on the edges of *P*. b) The response of the C_σ -operators. Since the responses at C1-C3 are all equal, only C1 is displayed. c) The responses of the same operators at the coordinates where the corners appear. (Note the difference in scale.) d) The responses of the E_σ -operators at corners P3, P4, and P5 for different preferred orientations. e) Identical to d, only the number of orientations N is doubled to 8. Note that the responses at corners P1, P2, and P6 are not displayed since they are equal with the response of P3. The used parameters are $\sigma = \sqrt{5}$, $w_r = 4$, $w_t = 1$, and $g = 2$.

$S'_{\sigma,\theta}$ should be rotated back over an angle ϕ to be exactly equal to $S_{\sigma,\theta}$. Fortunately the last step is not necessary since we are interested in the output of the E_σ -operator at corners only.

Consider that N orientations $\theta_i = \frac{i\pi}{N} + \theta_0$ starting at orientation $\theta_0 = 0$ are used, then the $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operators will respond best if line segments (or edges) in the image have

orientations $\frac{i\pi}{N}$, $i \in [0, N)$. The response drops if the line segment is not exactly in one of these orientations. By rotating the input stimulus around $\phi \in [-\frac{\pi}{2N}, \frac{\pi}{2N})$ we notice that the response of the C_σ - and E_σ -operators change. The results of change in response are shown in Figure 4.7b-d.

When N different orientations are used there are intervals of $\frac{\pi}{N}$ between two preferred orientations. If the preferred orientations are rotated $\frac{\pi}{N}$, it gives exactly the same response as when it is not rotated at all. Therefore the maximum useful rotation is $\pm \frac{\pi}{2N}$.

The strongest responses of the E_σ -operators were remarkably stable when we changed the preferred orientation, considering the maximum useful rotation. For all tested orientations, 23 samples in $\pm \frac{\pi}{8}$ and 15 samples in $\pm \frac{\pi}{16}$ for $N = 4$ and $N = 8$, the strongest responses were at the manually marked positions of the corners.

Figure 4.7b illustrates that the C_σ -operators have an almost linear decrease in response (worst case example C1). The response drops to about 75 percent of the maximum. Note that the orientations of C5 and C6 are not at one of the four preferred orientations $(0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4})$ but rotated -17 and 5 degrees away from one of the four preferred orientations. The C_σ -operators have a similar decrease in response as the S -operators. The latter have about 95 and 75 percent of the strongest response at $\frac{\pi}{16}$ and $\frac{\pi}{8}$, respectively (Petkov et al. [151]). However when the responses of the C_σ -operators are measured at the positions of corners, they show a remarkably stable response for all rotated input responses. Although the responses of P1-P6 are about 60 percent of the strongest responses of C1-C6, they do not drop below 96 percent of the strongest response (Figure 4.7c). The strongest response is obtained when there is no rotation.

Figure 4.7d-e illustrate that there is almost no difference between the number of orientations, if the stimulus is rotated within $\pm \frac{\pi}{16}$. The worst response with 8 orientations is approximately 10 percent less than the strongest response, this is the same for 4 orientations if we take the same maximum rotation of $\pm \frac{\pi}{16}$ into account. This means that if the number of orientations is once more doubled to sixteen, the weakest response of the modeled end-stopped cells will be at approximately 95 percent of the maximum response. The human brain is able to discriminate between more than 16 different orientations and therefore we conclude that with this set of orientations the E_σ -operators are almost insensitive to rotation of the input image.

Four different orientations are not recommended because there is a remarkable drop (Figure 4.7) in response when the stimulus is rotated over more than ± 21 degrees. Probably, when the four different Gabor functions are transformed into the frequency domain there is a small gap between two different orientations which cause the dramatical decrease in response. For most applications eight different orientations will be enough to find all corners and strongly curved edges.

4.2.2 Rotation of input image

In the previous subsection we mentioned the discretization problem when rotating an image and that rotation of the preferred orientation of modeled simple cells causes fewer problems. Unfortunately the different preferred orientations of simple cells are fixed, which means that

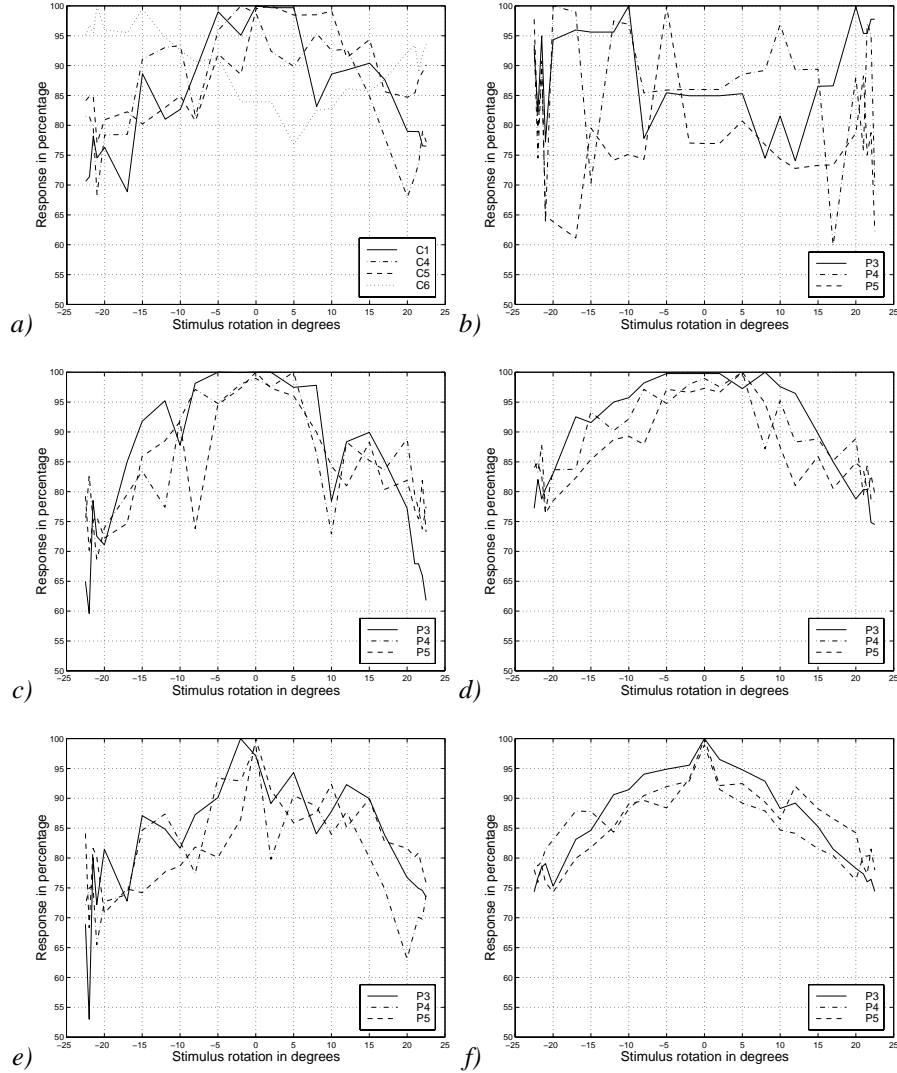


Figure 4.8: Responses upon stimulus rotation. The preferred orientations are kept constant. The six corners P1-P6 and the other coordinates C1-C6 on the edges of character P, as illustrated in Figure 4.7a, are used again. a) The responses of the C_σ -operators for different rotations of the input image. Since the responses of C1-C3 are all equal, again only C1 is displayed. b) The responses of the same operators at the coordinates where corners appear. c) The responses of the E_σ -operators at corners P3, P4, and P5 for different rotations of the input image. d) The max-operator applied to $E_\sigma(x, y)$ and its eight neighbors. The responses of corners P1, P2, and P6 are again not displayed since they are equal with the response of P3. The used parameters are identical to the ones in Figure 4.7. e-f) are similar to c and d, respectively. The image is rotated with the bilinear interpolation technique.

it is not possible to change the preferred orientation. Since we want a biologically plausible model we are now rotating the input image and keeping the preferred orientation fixed.

We use $N = 4$ different orientations $\theta_i = \frac{i\pi}{N}$ where $i \in [0, N)$ and character P again as input image I . The image rotated by angle φ is denoted as I_φ and is obtained as follows:

$$I_\varphi(x, y) = I(x_\varphi, y_\varphi). \quad (4.9)$$

From the rotated input image the discrete coordinates (x, y) are used to find the best integer coordinate (x_φ, y_φ) in the original image. Hence the image is rotating clockwise with increasing angle φ . The coordinates (x_φ, y_φ) are obtained as follows:

$$\begin{pmatrix} x_\varphi \\ y_\varphi \end{pmatrix} = \left\lfloor \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2} \right\rfloor. \quad (4.10)$$

The responses of the C_σ -operators in coordinates C1-C6, illustrated in Figure 4.8a show a capricious behavior when the difference between two rotated stimuli is small. The decrease in response becomes clear when the stimulus is rotated from 0 to $\pm \frac{\pi}{8}$. Comparing Figures 4.7b and 4.8a both show a similar decrease when rotating from 0 towards $\pm \frac{\pi}{8}$. Although the latter figure shows a non-smoothly decreasing response due to discretization effects.

The constant responses of the C_σ -operators at corners and line-ends seem promising for using their constant responses as corner detection mechanism (Figure 4.7c). However the responses shown in Figure 4.8b are by far not so stable as in the previous section. The discretization effects have a significant influence on the responses of these operators. They drop to 75 and 60 percent of the maximum response in the best (P3) and worst (P4) case, respectively.

The responses of the E_σ -operators as shown in Figure 4.8c show results comparable with the responses of Figure 4.7d. The discretization effects in Figure 4.8c give strong decreases in response for certain orientations, e.g. coordinate P4 at 10° and P5 at -8° and 10° . To avoid a large influence of discretization, the maximum response of the expected position plus its eight neighbors is taken. That means that a movement of ± 1 in horizontal and vertical direction of the calculated position of the corner is allowed. The reason for taking the maximum response of the coordinate and its neighbors is that the coordinates are integer values. By rounding the rotated coordinates to the nearest integers we can get the wrong coordinate, therefore this discretization effect is compensated by assuming that one of its direct neighbors also could have been the proper position of the corner. The response where a drift of ± 1 is allowed (Figure 4.8d) definitely gives better results than when the calculated position is allowed only, as shown in Figure 4.8c. The strong decreases in response are eliminated but the response is still not decreasing as smoothly as in Figure 4.7d.

4.3 Position of strongest response of ES-cells with different receptive field sizes

In this section the response and improvement of the strongest response of the E_σ -operator is examined with increasing receptive field size. The idea of using different receptive field sizes

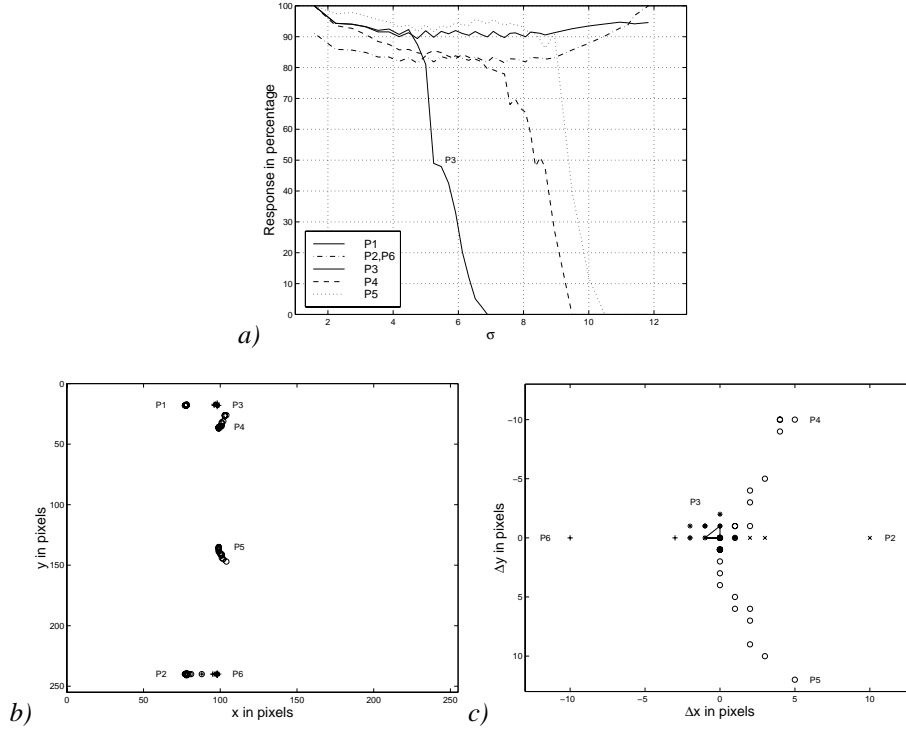


Figure 4.9: a) The strongest responses of the E_σ -operator at coordinates P1-P6 with increasing receptive field size. b) The movement of the strongest responses of the same operator at coordinates P1-P6 for different receptive field sizes. c) The relative movement of coordinates P2-P6. Note that the strongest responses at coordinate P1 are hardly moving and is therefore displayed with a solid line. The following parameters are used: $w_r = 4$, $w_t = 1$, and $g = 2$.

is to obtain corners and line-ends which are very reliable. For example, if a complex input image is presented and small receptive fields are used, undesirable corners and line-ends can be detected due to the background or texture of an object. These responses at corners and line-ends are weak or zero when large receptive fields are used, but the disadvantage with large receptive fields is that details from the input stimulus are lost and that corner localization is poor.

In Figure 4.9b the strongest response of the E_σ -operator at coordinate P1 stays at the same position for all the receptive field radii from 6.7 to 50.2. The responses at this coordinate are constant over all the sizes (Figure 4.9a) and can therefore be marked as a very stable corner. The responses are also strong at corner coordinates P2 and P6, when a visual scene is presented at a short distance (observed in detail). When the same visual scene is seen from a larger distance (observed in less detail) everything will appear smaller on the retina. In such case character P will be smaller and can appear for example as a printed character P. When such a P is considered, a human observer will find one corner at position P1 and one line-end

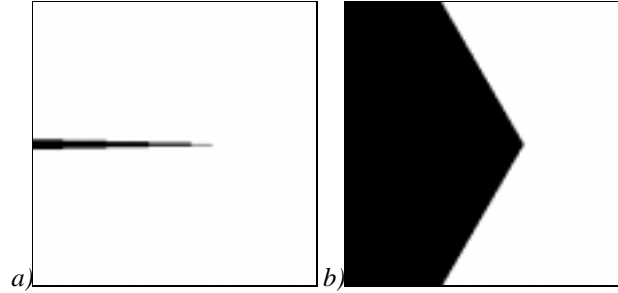


Figure 4.10: We applied the E_{σ} -operator to images containing a convex corner, to check the response near the corner and position of the strongest response. The response of the E_{σ} -operator of angles α between 3 and 180 degrees have been elaborated. a) The corner has an angle of 3 degrees and near the corner it is a line-end, smaller angles will not change anything near the corner and are therefore not applied to the E_{σ} -operator. b) The largest angle $\alpha = 120$ degrees where the responses of the E_{σ} -operator around the corner are still strong (see Figure 5.1).

between coordinates P2 and P6. The effect of differently sized input of the same object is modeled by using several different receptive field sizes. The corners located at P2 and P6 are fused to one new line-end exactly in the middle of P2 and P6 when large receptive fields are used. The fusion of the coordinates causes an increasing response of about 15 percent (Figure 4.9a).

The response of coordinate P3 vanishes first because the coordinate is at a distance 20 in horizontal and 18 in vertical direction from P1 and P4, respectively and therefore influenced by these coordinates. The receptive fields of the modeled simple cells with centers at coordinates P1, P3, and P4 start overlapping each other when they have a radius r larger than 9. At a receptive field radius of 19 the response of P3 starts decreasing because strongly inhibitory parts are located in P1 and P4. With increasing size of receptive fields the strongest response at coordinate P4 starts moving upwards and is disappearing quickly. The C_{σ} -operators are responding accurately to edges for small σ , but when larger and larger σ are used, fine detail at the edges near corner P4 is vanishing. When σ is large the C_{σ} -operators respond well to the upper horizontal edge but the response for the E_{σ} -operators is completely zero at coordinate P4. The identical phenomenon is repeating for larger σ with coordinate P5.

We also elaborate on the position of the strongest response of the end-stopped operators at convex corners with different angles. These responses are expected to be around the exact location of the corner. To be able to combine the end-stopped operators with the same receptive field center (x, y) but different receptive field radii, we need to know how the response at position (x, y) for different scales will be. Even more important is the *drift* of a corner which is defined as the deviation of the position of the strongest response from the exact location of the corner. If the drift is large and in different directions for different scales, the receptive fields do not or only partly overlap and combining will be difficult.

Since end-stopped cells respond well to corners and line-ends we used a synthetic corner as input image (Figure 4.10), where a line-end can be interpreted as a corner with a small angle.

α	Δx	Δy	α	Δx	Δy
3	1.5 ± 2.5	0.0 ± 2.0	80	0.0 ± 0.0	0.0 ± 1.0
5	3.0 ± 4.0	0.0 ± 1.0	90	0.5 ± 0.5	0.5 ± 0.5
10	0.0 ± 3.0	0.0 ± 0.0	100	0.5 ± 0.5	0.0 ± 0.0
15	-1.0 ± 3.0	0.0 ± 0.0	110	-1.5 ± 1.5	0.0 ± 2.0
20	-1.5 ± 2.5	0.0 ± 0.0	120	-2.0 ± 1.0	0.0 ± 1.0
30	-1.5 ± 2.5	0.0 ± 0.0	125	-2.0 ± 1.0	0.0 ± 1.0
40	0.5 ± 0.5	0.0 ± 1.0	130	0.0 ± 1.0	0.0 ± 1.0
50	0.5 ± 0.5	0.0 ± 2.0	135	$?? \pm ??$	$?? \pm ??$
60	1.0 ± 1.0	0.0 ± 2.0	140	$?? \pm ??$	$?? \pm ??$
70	-0.5 ± 0.5	0.0 ± 2.0	150	$?? \pm ??$	$?? \pm ??$

Table 4.2: In the leftmost and fourth column the convex angle α of the corner is given. In Figure 4.10 two corners with angles, $\alpha = 3$ and 120 degrees are illustrated. The drift in pixels of the strongest output of the E_σ -operator in x and y direction is in the second and third column for $\alpha \leq 70$, respectively. In the fifth and sixth column the drift for $\alpha > 70$ is given. The question marks $?? \pm ??$ indicate that at these angles the strongest output of the E_σ -operator is zero and no position (x, y) can be marked as the maximum output.

For the simulation $N = 8$ orientations $\theta_i = \frac{i\pi}{N}$, $i \in [0, N)$ for the $C_{\sigma, \theta}$ -operators are used. The outputs of the $E_{\sigma, \theta}^s$ - and $E_{\sigma, \theta}^d$ -operators at position (x, y) are combined with a max-operator for all orientations:

$$E_\sigma^s(x, y) = \max_{\theta} (E_{\sigma, \theta}^s(x, y)) \quad (4.11)$$

and

$$E_\sigma^d(x, y) = \max_{\theta} (E_{\sigma, \theta}^d(x, y)). \quad (4.12)$$

That different orientations need to be combined is obvious since a single orientation channel never contains all corners. We choose the max-operator since it is simple and independent of the number of orientations and is biological plausible (winner-takes-all).

Since $E_{\sigma, \theta}^s = -E_{\sigma, \theta+\pi}^s$ and because we use the max-operator: $\max(E_{\sigma, \theta}^s, E_{\sigma, \theta+\pi}^s)$, equation (4.6), can be written as

$$E_{\sigma, \theta_i}^s = \left[|E_{\sigma, \theta_i}^s| - g(I_\sigma^r + I_\sigma^r) \right]^{\geq 0}, \quad i = 0, \dots, N-1. \quad (4.13)$$

This has the advantage that only half of the E_{σ, θ_i}^s -operators need to be calculated, $i = 0, \dots, N-1$ instead of $i = 0, \dots, 2N-1$ and also that the orientation parameter θ of both $E_{\sigma, \theta}^s$ - and $E_{\sigma, \theta}^d$ -operators is in the same domain.

The E_σ^s - and E_σ^d -operators are combined with another maximum operator to what we call general end-stopped or E_σ -operator:

$$E_\sigma(x, y) = \max(E_\sigma^d(x, y), E_\sigma^s(x, y)), \quad (4.14)$$

to obtain the strongest response from both types of modeled end-stopped cells for all possible orientations. By using the building block architecture the response of such an E_σ -operator can represent the functionality of a biological cell in the brain. The outputs from all single and double end-stopped cells for all different preferred orientations but the same receptive field center and the same receptive field size make excitatory synapses with a single “E”-cell. That makes the E-cell a biologically plausible cell.

Fifteen different receptive field radii r from (A.5) ranging from $r = 5$ to $r = 40$ with steps of 2.5 have been used to measure the drift of the E_σ -operator. Table 4.2 gives the drift to the exact location of the corner where the strongest responses of the E_σ -operator appeared. The drift in y direction is given by Δy . The largest drift in x direction is when the corner has an angle smaller than 30 degrees. With the small receptive fields ($r = 5$ and $r = 10$) the strongest response is exactly at the location of the corner, when the radius is increased with steps of five the position of the best response remains the same or moves one position to the left. Exactly the same happens with large angles. The corners with angles smaller than 10 degrees show a different behavior, they tend to move to the right because they are almost identical to line-ends.

We conclude that the strongest response of the E_σ -operator at corners drifts a little. This is important if different receptive field sizes should be combined since it would be difficult if corners drift considerably with increasing σ . In Chapter 5 we are improving corner detection by combining end-stopped operators with different σ .

4.4 Related work

Corner detection methods in the literature can be broadly divided into two groups. The first group consists of approaches that work on images indirectly, whereas the second group of approaches works on the image directly.

Approaches in the *first group* involve extracting edges as a chain code (e.g. using the Freeman indices [48, 49]), and subsequently either searching for points of maximum curvature (Asada and Brady [5], Deriche and Faugeras [27], and Medioni and Yasumoto [132]) or performing a polygonal approximation on the chains and then searching for the line segment intersections (Horaud et al. [77]). The majority of these approaches involves the application of local operations along the length of a curve (Rutkowski [166], Anderson and Bezdek [3], Ray and Ray [158], and Seeger and Seeger [169]).

Let a sequence of n coordinates describe a contour C :

$$C = \{p_i = (x_i, y_i), \quad i = 1, \dots, n\} \quad (4.15)$$

where p_{i+1} is the successor of p_i . The contour is “eight-connected” which implies that if position p_i is known its successor can have eight possible positions. The Freeman chain code describes these possible positions with a number ranging from zero to seven, where zero is in downward direction and with increasing number the direction moves counter-clockwise. When a_i is the Freeman chain code number between p_i and p_{i+1} then the local curvature (Seeger and Seeger [169]) K_i at i is approximated as difference of neighboring chain code

numbers:

$$K_i \approx a_{i+1} - a_{i-1}. \quad (4.16)$$

Since the resolution of the Freeman indices is only $\pi/4$, the quantization error is high. In order to reduce the effect of quantization and smooth out the local fluctuation the k -curvature method is used. An orientation function ζ_i is introduced:

$$\zeta_i = \arctan \left(\frac{y_{i+k} - y_{i-k}}{x_{i+k} - x_{i-k}} \right). \quad (4.17)$$

Parameter k is called the smoothing factor, generally chosen to be two or three (Lee et al. [107]). Non-maximum suppression is used to associate corners with pixels whose ζ values are local maxima (exceeding a chosen threshold). The scale at which corners are detected is dependent on k . If k is too small, a rounded corner will not be detected, if k is too large problems may arise for corner pairs separated by less than k pixels. Fischler and Bolles [46] use several different values of k to detect corners at different scales. Since corners can exist at many different scales, Hartley and Rosenfeld [64] and Baugher and Rosenfeld [10] used a pyramid in order to detect corners at different scales. Hartley and Rosenfeld require a corner to lie at the intersection of the approximating lines, but Baugher and Rosenfeld require the corner to lie on the curve.

The classical iterative end-point algorithm, proposed by Duda and Hart [36] is a method using corner detection with splitting methods of two-dimensional image curves. The method splits a curve into sub-curves separated by corner-points. In each iteration a corner-point is identified to be the curve point most distant from the chord between the sub-curve's two end-points. Faster algorithms are proposed by Han et al. [60] and Espelid and Jonassen [42]. Although splitting curve algorithms are fast they have the property to detect corners on a circle. This indicates that they do not show rotation invariance.

Seeger and Seeger [169] use a real time corner detection method. Only few integer operations on a 3x3 pixel-matrix are required. Within a given direction quantization, local curvature is approximated by finite differences. The extrema of curvature are classified and subsets are selected as corners. Because a 3x3 matrix is used the corners are detected on a small scale. Consequently, a straight ragged line will yield corner points.

A common problem with k -curvature methods is that they fail to detect and localize what humans perceive as corners (Sohn et al. [175]).

The *second group* consists of approaches that work on grey level images directly (e.g. Dreschler and Nagel [35], Noble [143], Harris and Stephens [63], Guiducci [59], Singh and Shneier [172], Deriche and Giraudon [28], Rohr [162], Kitchen and Rosenfeld [97], Zuniga and Haralick [205], Rangarajan et al. [156], Deriche and Giraudon [29], and Wang and Brady [192]).

In 1978, Beaudet [13] proposed a rotationally invariant operator called DET, derived using a second-order Taylor expansion of the intensity surface $I(x, y)$:

$$\text{DET} = I_{xx}I_{yy} - I_{xy}^2. \quad (4.18)$$

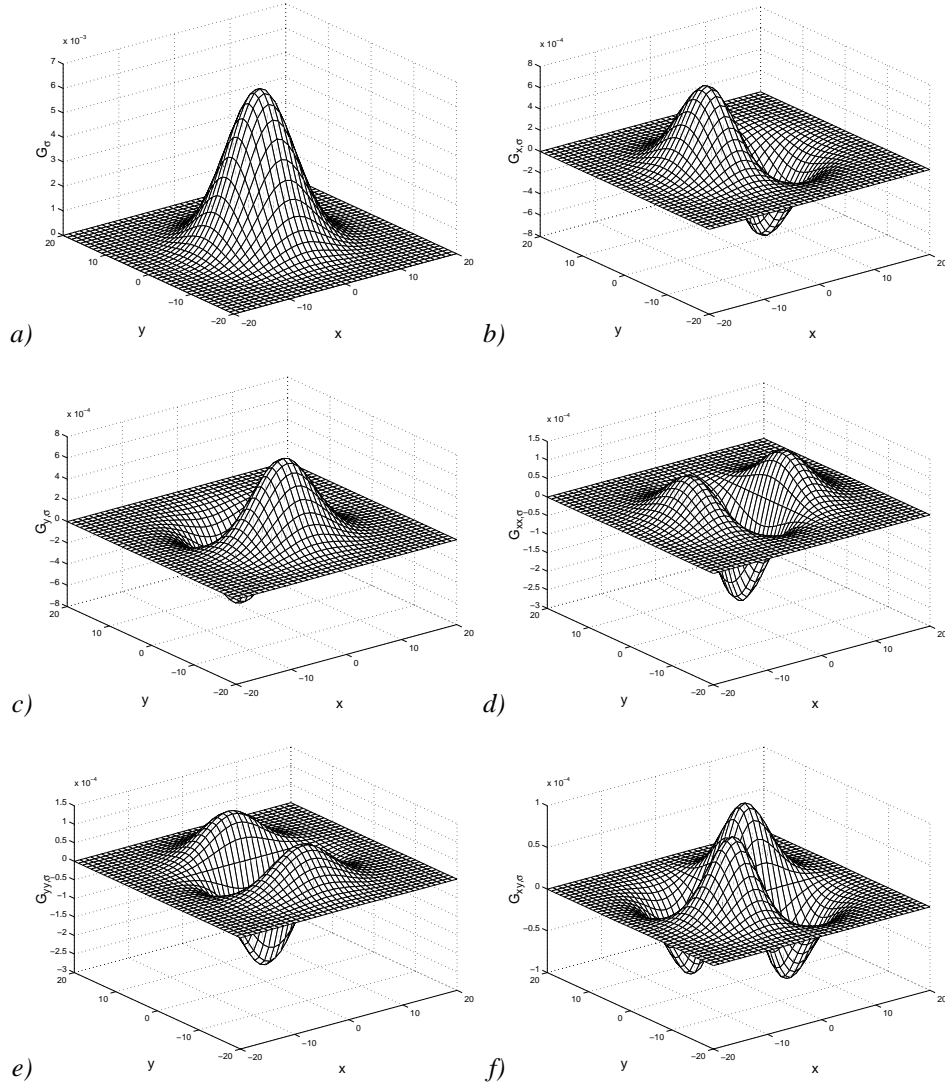


Figure 4.11: Different kernels used for the standard corner operators. *a)* Gaussian function. *b)* Partial derivative of the Gaussian in x -direction. *c)* Partial derivative in y -direction. *d)* Second order derivative, twice in x -direction. *e)* Second order derivative, twice in y -direction. *f)* Derivative once in x - and once in y -direction. For the Gaussian function (A.3) constant σ was chosen to be five.

The best realization of I_z , $z \in \{x, y, xx, xy, yy\}$ is to take a kernel as illustrated in Figure 4.11 and convolve it with an input image I , for example $I_{xx} = I * G_{xx}$. This is faster than blurring the image by convolving it with a Gaussian and then calculate the derivatives.

Corner detection using the DET-operator is based on thresholding of the absolute value of

the extrema of this operator. In fact the operator equals the determinant of the Hessian

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}, \quad (4.19)$$

which is related to the product of the principal curvatures $\kappa_{\min}\kappa_{\max}$ and is called the Gaussian curvature, Lipschutz [115]:

$$\kappa_{\min}\kappa_{\max} = \frac{\text{DET}}{(1 + I_x^2 + I_y^2)^2}. \quad (4.20)$$

Dependent on the output, these curvatures are divided into three groups (elliptic, hyperbolic, or parabolic). Dreschler and Nagel [35] proposed an operator based on this Gaussian curvature principle.

Kitchen and Rosenfeld [97] proposed a measure of “cornerness” based on the change of gradient direction along an edge contour multiplied by the local gradient magnitude:

$$KR = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2}. \quad (4.21)$$

The local maximum of this measure isolates corners using a non-maximum suppression process applied on the gradient magnitude before its multiplication with the curvature. In fact KR is the explicit representation for the second derivative in the direction orthogonal to the gradient (Torre and Poggio [186]). The measure of cornerness is sometimes expressed as $KR/\sqrt{I_x^2 + I_y^2}$ (e.g. Noble [143] or Lindeberg [113]).

Noble [143] has shown how the Plessey corner detector of Harris and Stephens [62] estimates image curvature and has proposed an image representation that is based on differential geometry:

$$P = \frac{\text{trace } M}{\det M}, \quad (4.22)$$

where $\text{trace } M = \sum_i M_{i,i}$, $\det M$ is the determinant of M , and M is the matrix introduced by Nagel [139] in calculations concerning optical flow estimation:

$$M = \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} + \mu^2 \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}^2, \quad (4.23)$$

with a suitable constant μ . Detector P is only suitable for V-junctions, and its performance on higher-order structures is unpredictable.

A slightly modified version of the Plessey corner detector, what has become known as the Plessey feature point detector, is considered by Harris and Stephens [63]. They have the following operator:

$$P_{HS} = \det M - k(\text{trace } M)^2, \quad (4.24)$$

for providing discrimination against high-contrast pixel step edges ($k = 0.04$). The output of the operator is thresholded for corner detection.

Many gray-level corner detectors do not extract feature attributes of a corner apart from the strength of the response (e.g. Kitchen and Rosenfeld [97], Zuniga and Haralick [205], Rangarajan et al. [156], Deriche and Giraudon [29], and Wang and Brady [192]) but assume an idealized corner that is sharply pointed and has straight, steep edges. In natural images most corners do not have such properties. Sometimes edges are ragged and corners rounded or blurred. Also contrast, adjacent edges, color, and texture have influence on the detection of corners.

Rosin [165] provides several post-processing techniques for determining additional corner features, such as color, orientation, angle, and contrast. Additional information about corners can be used to resolve otherwise ambiguous correspondences during matching, e.g. corners whose additional features do not match certain criteria. Rosin derives cornerness from a subset of the following properties: *position*, *subtended angle*, *orientation*, *edge shape*, *edge texture*, *contrast*, *edge profile*, *sharpness*, *color*, *junction type*, and *size*.

There are a few corner detectors which do measure some of those feature attributes. Li and Madhavan [111] use a set of basis functions which also provides the orientation. Rosin [165] gives a definition of *corner orientation*, which is the average of the orientation of two edges forming the corner. The *subtended angle* is the angle between those two edges. The same definitions can be used for two neighboring edges with an angle larger than 180 degrees.

Mehrotra et al. [133] detect half edges, from which orientation and subtended angle can be calculated. Liu and Tsai [116] determine orientation and subtended angle, while Rohr [163] additionally provides blur and contrast. Guiducci [59] also estimates orientation, subtended angle, blurring, and contrast based on differential geometry. Wang and Brady [192] use Beaudet's [13] DET-operator to estimate corner sharpness. Rosin [165] describes methods for calculating color (light vs. dark), orientation, contrast, and subtended angle between two edges.

Koenderink and van Doorn [99] and ter Haar Romeny et al. [182] use methods of differential geometry. Their differential operator

$$\text{Diff} = I_{xx}I_y^2 + 2I_xI_yI_{xy} - I_{yy}I_x^2 \quad (4.25)$$

enhances corners in an image. Their approach (like the others given before) is an elegant mathematical formalism, but only part of the model is neuro-physiologically inspired.

Perona [146] used Gaussian derivative based kernels with an elongated shape to achieve high orientation selectivity. The real part is a Gaussian: $G(x, y, \sigma_x, \sigma_y) = \exp(-((x/\sigma_x)^2 + (y/\sigma_y)^2))$ differentiated twice along the y-axis using a $\sigma_x : \sigma_y$ ratio of 1 : 3. The imaginary part of the kernel is the real part of the Hilbert transform taken along the y-axis (Perona [147]).⁵ The first derivative along the axis of the initial kernel yields the end-stopped kernel.

Five different standard operators have been implemented and applied to the P-image of Figure 4.2a. The outputs of the operators are set to zero if the absolute value does not exceed a certain threshold. The Plessey operator forms an exception, its absolute value should not exceed a certain threshold, otherwise it is set to infinity.

⁵The "gauss-3" kernel used by Perona looks similar to the Gabor kernel, illustrated in Figure 3.5a-b.

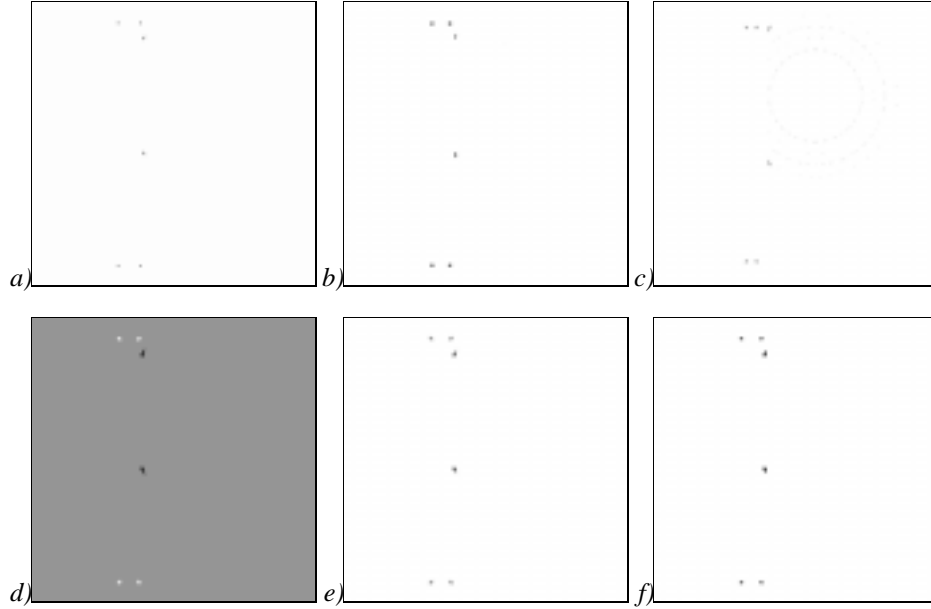


Figure 4.12: The thresholded results of the outputs of different operators which are applied to the smoothed synthetic P image, illustrated in Figure 4.2a (without the two small arrows). *a)* Diff-operator. *b)* DET-operator. *c)* Gaussian curvature. *d)* Kitchen-Rosenfeld operator. *e)* Plessey feature point operator. *f)* Plessey-operator. We used $\sigma = 2$ and $\mu = 1$, the outputs are thresholded at 20000, 50, 0, 10, 0, and 0.01, respectively.

In Figure 4.12b-d black, grey, and white represent a positive, zero, and negative response, respectively. The first implemented operator is the “Diff” or corner enhancing operator from (4.25). The strongest responses of this operator are moved towards the inside of the vertical bar of P . The results of the DET-operator (4.18) are almost at the proper place. Although smaller than the strongest responses of the Diff-operator, the strongest responses of the DET-operator are displaced in to the same direction as well. The strongest responses of the Gaussian curvature operator (4.20) are not at the proper position. The Kitchen-Rosenfeld operator (4.21) and the Plessey feature point operator (4.24) give the strongest responses at the proper locations. For the Plessey feature point operator we used $\mu = 1$. The Plessey-detector (4.22) gives the weakest responses at the proper locations (Figure 4.12f), also with $\mu = 1$. The output of the Plessey-operator is treated differently from the other operators in the sense that a point is marked as a corner if and only if the absolute value of the output is smaller than a certain threshold.

Only a few explicit models for end-stopped cells have been published. Fukushima [51] computed differences between offset complex cell responses and both single and double end-stopped cells were considered. The complex cells used by Fukushima are different, they are specially designed for representation of lines. More recently Dobbins et al. [32, 33] constructed a model which only considered double end-stopped cells. The inputs of these cells

were obtained from simple cells and not from complex cells. The excitatory and inhibitory inputs also had different receptive field sizes.

Fukushima [51] recognized the problem of false responses, to eliminate these responses inhibitory end-zones were made larger and given higher weighting. Recently, Weitzel et al. [193] proposed a neuronal model for contour segmentation which includes neurons with end-stopped properties.

4.5 Summary and discussion

In this chapter we used a family of two-dimensional Gabor functions to model the receptive fields of simple cells, which were defined in Chapter 3. The Gabor functions used are different from those used by Heitger et al. [71] but their functionality differs marginally. The symmetrical and anti-symmetrical part of the Gabor function are combined in the same way as Heitger et al. do. The $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators are slightly different in the sense that they are defined for all orientations and are not thresholded at zero. We thresholded the outputs of the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators after inhibition by the surround inhibition mechanism for computational reasons. When different orientations of the operators are combined with the max-operator, instead of $2N$ only N different orientations have to be calculated.

The strongest responses of the $C_{\sigma,\theta}$ -operators appear at line segments (or edges) but the response decreases before the line-end is reached, as illustrated in Figure 4.2j. The $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators have their strongest responses exactly at the locations of the corners but inhibit the response at line segments at or near θ only and false responses appear. Hence the output is inhibited by two additional operators: I_{σ}^l - and I_{σ}^r -operators.

The main contribution in the chapter is in the properties (rotation and relative movement of the strongest response) the $E_{\sigma,\theta}^s$ - and $E_{\sigma,\theta}^d$ -operators have. Also the choice of the four constants d , w_r , w_l , and g is motivated. The choice $d = 1$ of Heitger et al. [71] was not clear and for our purpose (responses at junctions only) also not optimal. We do not get responses at circles but if the parameters are used as proposed by Heitger et al. responses appear at circles as illustrated by Fellenz [44]. The weights w_r and w_l are found empirically and are independent of orientation and receptive field size. The gain factor g was examined for a large number of receptive field sizes and for 4 different numbers of orientations ($N = 4, 6, 8$, or 10). The number of orientations N should be at least four in order to be able to eliminate all false responses. It turned out that g can be chosen rather arbitrarily if the only criterion is false response elimination. In Table 4.1 is illustrated that for $\sigma \geq \sqrt{5}$ every $g \geq 1.5$ is sufficient. In order to have almost the same responses over a limited range of receptive field radii the gain factor should quadratically increase with the receptive field radius. The range is limited since responses to corners disappear when large receptive fields are used as shown in Figure 4.9.

We examined the discretization effects of rotated input stimuli with and without bilinear interpolation and used the rotation of simple cells as reference material. It became obvious that the discretization effects of a rotated input stimulus have considerable influence on the output of the E_{σ} -operator. Globally the responses at the rotated input stimulus are the same as the response of the S -operators with a rotated preferred orientation. However the difference

between two rotated input stimuli which are rotated over slightly different angles is unpredictable. The response of the E_{σ} -operator decreases to 60 percent of the maximum response when 4 orientations are used. This decrease can be explained by comparing the responses with the response of the C -operators, when an edge does exactly fit one of the preferred orientations the response is optimal and when it does not exactly fit, the response decreases. Although at first sight the responses of these operators seem to be constant over all orientations (rotating the preferred orientation of the modeled simple cells), we should not compare the responses of E -operators with these responses since the excitatory and inhibitory parts of the E -operators are at the edges of the input stimulus and not at the corners.

A denser sampling of preferred orientations for S -operators implies that each edge is closer to one of the preferred orientations, which means that the decrease of the response of E_{σ} -operators is small. When the input stimulus is rotated it is recommended not only to take the manually marked position of the rotated corner but also its eight neighbors, i.e. to allow a drift of the corner of ± 1 position in order to obtain more stable responses. When the number of orientations is 8 or 16 and the drift of ± 1 position is taken into account the responses are robust to rotations.

Finally we checked the responses of E_{σ} -operator on the stability of the position of corners (i.e. where do these operators have the strongest responses) with increasing receptive field sizes. For a convex corner the maximum drift of a corner is small for all different angles and scales. The drift is strongest when the angle is small, the corner looks like a line-end and the strongest response of the E_{σ} -operator is found a little before the position where the line actually ends. We showed that the responses of the E_{σ} -operator at corners are stable up to a certain receptive field radius. When the receptive field radius is increasing more, accuracy is lost and the strongest response at these corners start drifting and their responses are decreasing rapidly. Figure 4.9a illustrates this phenomenon for corners P3, P4, and P5. If receptive fields are made larger and larger strong responses of the E_{σ} -operator spread and the exact position of the corner becomes less and less accurate, even responses at corners are becoming weak, because different corners of the input signal appear in the same receptive field and inhibit each other. Finally, the receptive fields are so large that they respond to a few corners. Instead of having a rapid decrease of the response it also happens that the response at a certain position increases, e.g., when two corners are fused to one line-end. If receptive fields would be larger than the input stimulus, the stimulus can be seen as a blob. In such a case the operator will respond best in the center of the blob.

We conclude that when sixteen different $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operators with respect to the preferred orientation are chosen arbitrary but equally spread over 180 degrees the responses of the E_{σ} -operator drop in the worst case to 95 percent of the maximum response but the strongest response is obtained from the same position of the input stimulus. Therefore it does not matter which sixteen equally spread preferred orientations we take. Although discretization effects play an important role, comparable results are obtained by rotating the input stimulus and keeping the preferred orientations fixed. In both cases, rotating these orientations and keeping the input stimulus fixed or keeping the orientations fixed and rotating the input stimulus, the positions of the strongest responses of the E_{σ} -operator have been found to be invariant.

For comparison, we implemented five different standard corner operators, the results of these operators are good (i.e., only corners are enhanced) when applied to simple images (e.g. the P image) and give almost identical results when compared with the results of the E_σ -operator. The advantage of most of these standard corner operators is that they are calculated fast. There is also a disadvantage: a threshold needs to be tuned in order to obtain proper results. In the following chapter these standard operators will be applied to more complex images and their properties will be discussed in more detail. The large computation time of the E_σ -operators as compared with the standard operators (Section 4.4) is a disadvantage in the current implementation. The computational time can be cut down since the $S_{\sigma,\theta}^a$ - and $S_{\sigma,\theta}^s$ -operators can be calculated in parallel for every θ (and σ). After the S -operators are calculated all $C_{\sigma,\theta}$ -operators can be calculated in parallel for every θ (and σ). Finally the $E_{\sigma,\theta}^s$, $E_{\sigma,\theta}^l$, I_σ^l , and I_σ^r -operators can be calculated in parallel for every θ (and σ). In terms of parallel complexity the time needed to obtain corners is the sequential flow from center-surround, simple, complex, to end-stopped cells.

5

Improving Corner Detection using Multiple Scales

So this is what the Sovereign Lord says: “See, I lay a stone in Zion, a tested stone, a precious cornerstone for a sure foundation; the one who trusts will never be dismayed.”

Isaiah 28:16 (NIV)

SINGLE SCALE CORNER DETECTION as described in the previous chapter is usually not sufficient in the sense that there is a large possibility that some corners are not detected at all and others appear at a position where they are not expected. If a corner appears at a place where no corner exists then it is a false response (see also Page 46) and if no response appears at the exact location of a corner it is called a *missing response*. When a single scale is used for the E_{σ} -operators and a corner appears at a certain position one is not able to detect automatically if such a corner is correct or false. In this chapter different scales are used to reduce the chances of missing or falsely detected corners or line-ends to a minimum. If the response of $E_{\sigma}(x, y)$ is strong for several different σ (scales), there is more evidence that there is a corner at position (x, y) .

Corner detection at small scales for instance is very accurate but highly sensitive to every small detail, e.g. fine texture or camera noise. At large scales the detection is more reliable but inaccurate, also two or more corners can fit into one receptive field. The choice for scales which are neither small nor large seems to be the best one, although those operators do not respond well to rounded corners like the large scales do, and are not as accurate as the small scales.

The choice for a combination of scales improves the judgment whether there is a reliable corner or a noise response. A corner is *reliable* if it appears in more than a certain number of scales with a response higher than a certain threshold. Combining scales also gives additional information about the corner. For example noise responses appear in small scales and the E_{σ} -operators for large σ will respond well to rounded corners where small σ will give no response at all. Figure 4.6 illustrated the strong responses at the exact locations of corners and

line-ends of the E_σ -operators when applied to a simple image created by hand, but complex images (of natural scenes) can contain fine textures, e.g. plaster on a wall.

On a single scale different properties of the E_σ -operators must be elaborated first. Important are the position of the strongest responses of these operators. Are the strongest responses close to the exact locations of the corners? Does the operator respond well to any arbitrary corner? If it happens that this is not the case, can or should we improve the operator so that it responds to a wider range of corners? Does the operator respond to junctions?

In Chapter 4 we showed that the drift of the strongest responses of the E_σ -operators at different angles for a number of scales is small. This means that it is possible to combine different scales. For combining scales we need to answer the following questions: How many scales should we take? Which scales are needed? Should a linear or non-linear scale space be used? The most important question: how to combine the responses of the E_σ -operators at different scales? What is the advantage of using multiple scales? It is clear that with multiple scales the aim is to find all corners and have no false responses, but are we really going towards this goal by using multiple scales or is a single scale good enough? Do we introduce other problems by using multiple scales? For example is there a possibility that there is a conflict between two or more corners at different scales? In this chapter we will try and find an answer to these questions.

The chapter is organized as follows: in Section 5.1 a limited overview of related work of multiscale corner detection methods is given. In Section 5.2 a synthetic input image with one convex corner is presented and the response of the E_σ -operator for different scales presenting different angles of the corner between 3-180 degrees is studied. Corners smaller than 3 degrees are not considered anymore because the corner will look like a line-end. Junctions and rounded corners are examined as well. In this section we also discuss a specific example of drift. In Section 5.3 a potential corner feature (PCF) area and marked corner (MC) are introduced, since we want one single (pixel) point to be marked as a corner. The choice for a coordinate (x, y) to be marked as a corner is based on two criteria, the first one is that the response of $E_\sigma(x, y)$ should be stronger than responses at the neighboring positions and secondly the response should exceed a known threshold. The responses of the end-stopped operators which exceed this threshold are called potential corner features. In a PCF area there is at least one position which is marked as a corner, outside the PCF areas no corners are marked. The results of the E_σ -operator and standard corner operators (from Chapter 4) are applied to complex images and compared with each other. Combining different scales will be discussed in Section 5.4. One of the advantages of using multiple scales is that we get an additional feature attribute, which indicates on which scales a corner is found. For example the responses at the corner can be found in the small, middle, or all scales. Such an attribute can be used to facilitate an object detection task (Rosin [165]). The multiple scale end-stopped operator will be applied to several different natural images. The chapter ends with a summary and discussion.

5.1 Related work

Baugher and Rosenfeld [10] use an image pyramid for corner detection. A pyramid is a multi-layer data structure used to represent a digital image at different scales. Each level of the pyramid consists of a square array, with a difference in image resolution of a factor two between each level. They apply a curvature algorithm to all levels of the pyramid. The corners detected in every level are combined using a max-operator.

Fischler and Bolles [46] use smoothing factors k in order to detect corners at k different scales, as mentioned in Chapter 4.

Deriche and Giraudon [29] consider a corner model based on a scale space approach that combines useful properties from the Laplacian and Beaudet's measure and is then used in order to detect the exact corner position.

Rattarangsi and Chin [157] use a technique for detecting and localizing corners of planar curves. The technique is based on Gaussian scale space, which consists of the maxima of absolute curvature of the boundary function presented at all scales. This scale space is transformed into a tree that provides a simple but concise representation of corners at multiple scales. A multiple scale corner detection scheme is developed using a coarse-to-fine tree parsing technique.

A scale space can be created for the standard corner operators also by using different σ for the Gaussian kernels. We combined different scales of the standard corner operators with an average operator and compared the results with the multi scale end-stopped operator.

Also Lindeberg [113] uses a scale space representation for junction detection but does not combine these scales.

Our approach is based on convolving a grey-scale image with Gabor functions with different σ . Using filters with different σ has the advantage that any scale sampling can be used and not only factors of two. Unlike Rattarangsi and Chin who used a max-operator to find all corners, we use an average operator to combine different scales since the max-operator does not rule out strong responses to noise or tiny features at small scales.

5.2 Responses at different corners

5.2.1 Convex angles

In the previous chapter we checked the drift at different angles of synthetic images, an illustration of such an image is given in Figure 4.10. In this section the responses of these images are elaborated.

In Figure 5.1 the strongest responses of different receptive field radii are illustrated. It is remarkable that the responses are almost identical for all scales at angles between 10 and 120 degrees, only the smallest receptive field $r = 5$ is an exception.

The smallest receptive field has a strong response at corners with a small angle (3 degrees), the response drops rapidly with increasing receptive field radii. Its response is only half for a receptive field with radius $r = 10$ and it is less than fifth for radii larger than 35, compared to the strongest response. The decrease is caused by the thin line, as illustrated in Figure 4.10a.

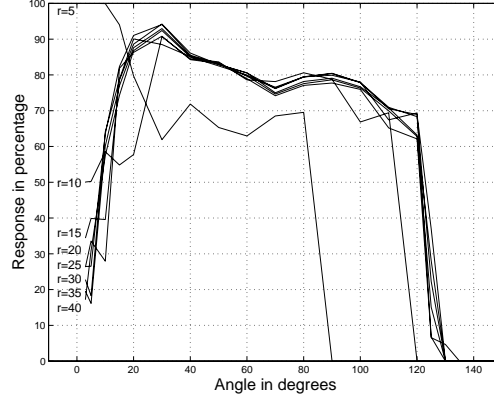


Figure 5.1: The relative output of the E_G -operator at corners with different angles. The response of the E_G -operator with receptive field radius $r = 5$ responds best to a corner with an angle of 3 degrees, this response is scaled to 100 and all others are related to this response. The lines labeled from $r = 5$ to $r = 40$ represent the different receptive field radii.

Such a line influences the main part of a small receptive field at the corner and therefore it responds well but if the receptive fields get larger the part of the receptive field that influences the thin line gets relatively smaller and smaller, this implies that the influence of such a line in the receptive field gets smaller and smaller, and therefore also its response.

The response E_G -operator decreases rapidly at corners with an angle between 120 and 125 degrees and vanishes completely for larger angles. Large angles are not recognized anymore, this phenomenon is caused by the orientation selectivity of the S -operators and the I_G' -operator. The S -operators respond best to a preferred orientation ϕ but still respond well (fifty percent of the response at the preferred angle) at an angle $\phi \pm \frac{2\pi}{9}$ (Petkov et al. [151]). Hence S -operators will respond to both lines if the angle between them is between 120 and 180 degrees. This means that the I_G' -operator has little or no influence on the response, at corners with such angles, since both edges give a good response at the same orientation. An increase of the number of orientations will not have an improving effect. For corners with an angle between 120-140 degrees, we can improve the response of the I_G' -operator at corner points. This is possible because if an edge differs 40° from the preferred orientation the response is still about half of the response compared to the response of the edge which has an orientation which is equal to the preferred orientation. When the edge differs 30° from the preferred orientation the response is still 60 percent of the maximum.

We conclude that the response at every different scale for a corner with the same convex angle is almost the same. In the previous chapter we already mentioned that the drift is small. The combination of a small drift and the equal response of different scales makes the E_G -operator suitable for combining different scales. The E_G -operators respond well to angles from 20 to 125 degrees and are scale independent. For angles larger than 130 degrees the E_G -operators fail to respond at every scale. The E_G -operator fails here while a human is still able to mark the corner. Although it does not necessarily mean that the end-stopped cells should

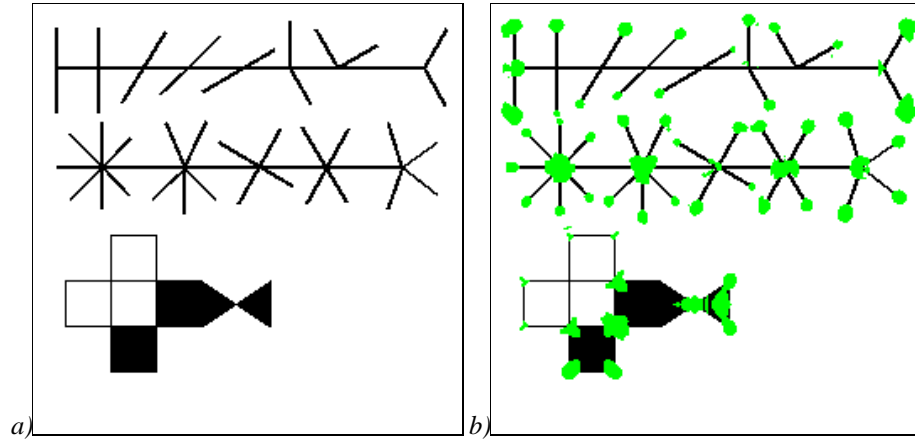


Figure 5.2: a) An input image with different junctions. A T-junction and Y-junction (3-junctions) are at the top row on the left and right, respectively. In the middle of the top row are different X- and K-junctions (4-junctions). In the second row other n-junctions ($n \geq 5$) are illustrated. The squares and triangles at the bottom are used to show how the E_G -operators respond to different combinations of edges and line segments. b) A maximum operator is used to combine the responses of the E_G -operator at different scales. The thresholded output of the maximum operators is illustrated in gray. Scales ranging from $r = 5$ to $r = 40$ with a step-size of 2.5 are used. At large scales the black triangle is like a blob therefore the E_G -operator responds well at this position.

respond at angles larger than 130 degrees we interpret this as a limitation of the modeled end-stopped cells. Using “simple cells” with different bandwidths does not improve the response at angles larger than 130 degrees. The responses at small angles are good at small scales but decrease with increasing scale, these angles are therefore scale dependent.

The difference in response at corners with angles between 20 and 120 degrees is 20 percent only, from 90 percent at a corner with an angle of 20 degrees to 70 percent of the maximum response at 120 degrees. This indicates that the E_G -operator responds well to a corner of any angle between 20 and 120 degrees and no compensation is necessary to be able to compare corners with a different convex angle.

5.2.2 Junctions

Peterhans et al. [149] and van der Zwan et al. [190] studied the role of end-stopped cells in mechanisms of figure-ground segregation. They recorded the responses of single neurons in areas V1 and V2 of rhesus monkeys, that were trained on a visual fixation task that reinforced foveal viewing, to stimuli that mimicked occlusion cues (line-ends, corners, and T-junctions). Of the 100 end-stopped cells studied 98 did respond to terminations (line-ends and corners). About half of these neurons gave a stronger response when the stimulus covered one half of the receptive field, and only half of that response or less when it covered the other half. The remaining neurons had symmetrical receptive fields giving similar responses independent of which side of the receptive field was covered by the stimulus. When tested, both types of

neuron also responded to T-junctions and only weakly or not at all to X-junctions.

We applied the E_σ -operators, with scales ranging from a radius of 5 to 40 with a linear increasing step size of 2.5, to different junctions which are in one input image, illustrated in Figure 5.2a. The output of the $E_\sigma(x, y)$ is taken and marked if its response was higher than a threshold T , which is about two percent of the strongest response. The results are shown in Figure 5.2b.

The E_σ -operators respond well to T-junctions and do not respond at all to X-junctions when they are formed by line segments. In one case there is a response but the response is weak in three scales only and is not at the position of the X-junction. Remarkable are the responses at X-junctions which are formed by edges instead of line-segments, which are illustrated in the lower part of Figure 5.2a. The responses at these junctions are strong at all scales. The responses at X-junctions which are a combination of two edges and two line segments are strong also, but at the middle and large scales only. These responses are not at the exact location of the X-junction but moved towards the inside of the black square.

The E_σ -operators respond at all scales to the 3-junction in the upper row at the right of Figure 5.2a. Like the 3-junction the response at the 8-junction, second row at the left, and the symmetrical 6-junction are good at all scales also. Strong responses at 5- and 7-junctions are in the middle and large scales. The non-symmetrical 6-junction, second row in the middle evoked a response at large scales but the responses at these scales are weak.

We conclude that the E_σ -operators respond to all n -junctions ($n \geq 3$), except K- and X-junctions ($n = 4$) where all parts of the junction are line segments.

5.2.3 Rounded corners

Corners in natural scenes are not always sharp, therefore it is important to examine the response of the E_σ -operators at rounded corners. It is expected that the operators with small σ do not respond to rounded corners because their receptive fields are too small to contain the entire corner area. The operators with large σ do not only contain the complete corner area but also parts of the straight line segments. In such case the response is expected to be strong at the corner. Figure 5.3d shows that the E_σ -operator with small receptive fields ($r = 5$) respond to rounded corners up to a radius of two. The operators with large receptive fields ($r = 40$) respond well to rounded corners with a radius up to 15.

If 80 percent of the strongest response is used as a threshold for corner detection then the maximum radius of the rounded corner and end-stopped distance $d\sigma$ are almost the same. This is expected because the receptive fields must contain the entire corner plus parts of the straight lines and because the output of I_σ^r - and I_σ^l -operators are suppressed at line-ends. We conclude that the end-stopped distance $d\sigma$ gives the upper bound for the radius of the rounded corner. Since $d\sigma$ is proportional to the receptive field radius, every rounded corner can be detected. However, when large receptive fields are used the position of the rounded corners can be determined less accurately because there are a lot of receptive fields with different receptive field centers which can be influenced by the same corner. However, determining the exact location of a rounded corner, even marked manually, is difficult since one has a choice to mark the corner on the edge in the middle of the rounded part or at the

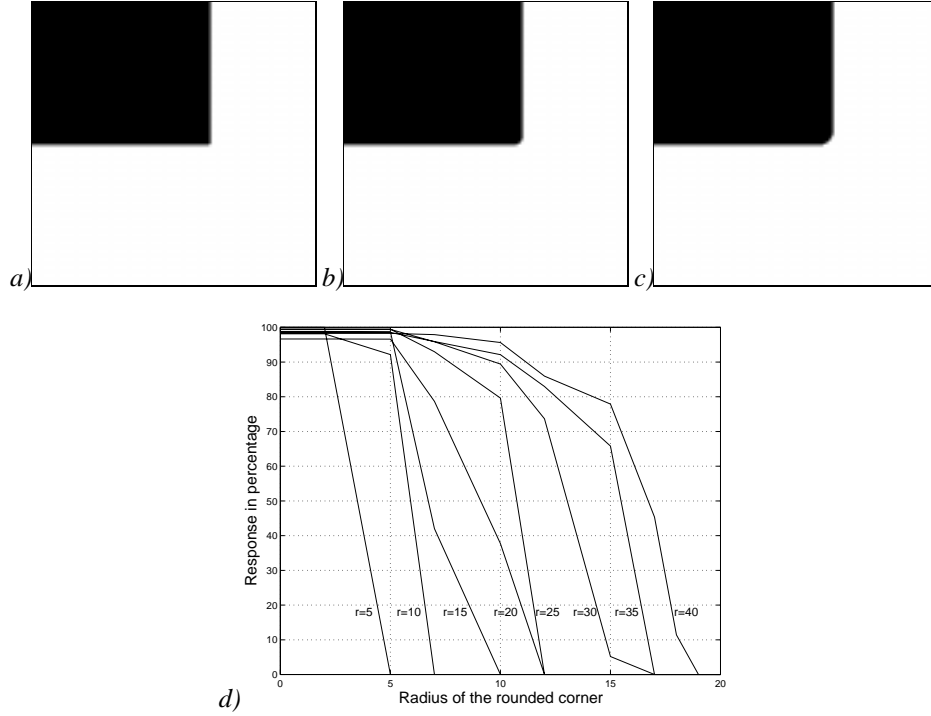


Figure 5.3: Images with different rounded corners. a), b), and c) have radii of 5, 10, and 15, respectively. d) Responses of the E_{σ} -operator at rounded corners with different receptive field radii. The lines labeled from $r = 5$ to $r = 40$ represent the different receptive field radii of the simple cells. The parameters used are identical to the ones used in Figure 5.1. The I'_{σ} -operator is used.

imaginary intersection of the two straight edges.

5.2.4 Conflicting corners

In Figure 5.4 an image is created (without the two small grey spots) which models corners like P4 and P5 from Figure 4.9b-c. Using again 8 orientations and identical parameters for the E_{σ} -operator as in Section 5.2, the spots in Figure 5.4a indicate that two areas with strong responses are found. Strong responses at the lower spot of Figure 5.4a are appearing in small scales, as illustrated in Figure 5.4b. Strong responses at the upper spot are found in large scales. We conclude that this is a special form of drift which we will call *conflicting corners* since it depends on the scale and the context in which the image is shown.

Combining scales in order to find a reliable corner with conflicting corners is a harder problem. One can decide to take a position which depends on the direction of the drift or just to take one of the conflicting corners. The latter has the advantage that different scales can be determined by combining different scales at one coordinate (x, y) in the visual field.

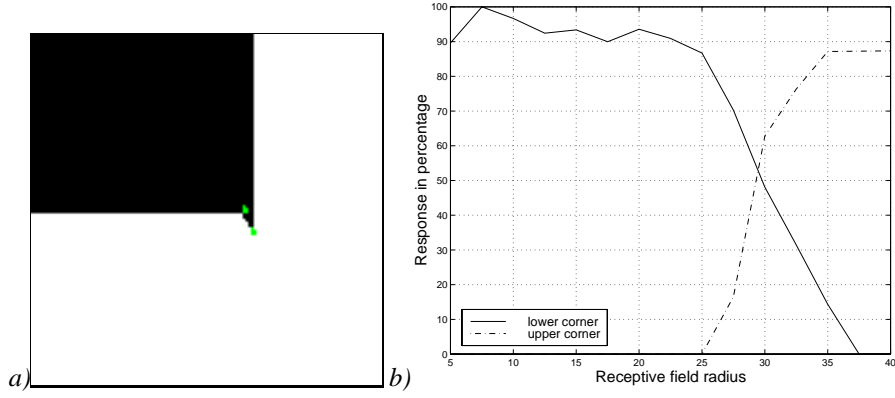


Figure 5.4: a) An input image which has two separable areas where the E_σ -operators respond well. This are the two grey areas in the image. At small scales the operators respond well to the lower area and at the large scales it respond well to the upper area. b) Response of the E_σ -operators at different scales for the upper and lower corner.

If the first method is chosen, the combination of different scales at position (x, y) is not sufficient. The drift of such a corner must be traced starting at a large scale since their responses are most reliable, moving towards the best response of a smaller scale, predicting the direction for the next scale.

5.3 Detected corners at a single scale

The scales we are using range from receptive field radius $r = 5$ to $r = 40$. The minimal possible radius is $r \approx 2.5$ because the end-stopped distance $d\sigma$ should be at least one pixel when discrete images are used. Discretization effects for such small radii are so severe that their results are not predictable when applied to a discrete image. In the Fourier domain the highest possible frequencies without overlap (aliasing) are obtained when $r = 7$. For more details about the choice of different scales in the frequency domain we refer to Section 3.5. The maximum radius is mainly based on the resolution of the used input images. We use square input images with a resolution of 256×256 pixels. A circularly shaped receptive field with radius $r = 40$ is influenced by almost eight percent of the image. Larger receptive fields are not useful, first of all because it will not give better or additional information compared with a scale which is a little smaller, secondly the implementation of filters with such large radii can cause considerable wraparound artifacts. These artifacts can be avoided by enlarging the image but will take more (computer) memory and computation time. The use of larger receptive fields will be useful when larger images are used.

In Chapter 4 and the previous sections of this chapter we checked the response of the E_σ -operator at several different corners and junctions and analyzed their drift at different scales. The current section gives criteria when a position (x, y) is marked as a corner, and a real-



Figure 5.5: An image with five objects on a fine texture background.

world image is used to test the “quality” of the E_σ -operator. The response of the operator is compared with the output of the standard operators from Chapter 4.

The response of the E_σ -operator at position (x, y) in the visual field is marked as a *potential corner feature* (PCF) if it satisfies the following condition:

$$\text{PCF}_{T,\omega}(x, y) \equiv [E_\omega(x, y) \geq T], \quad (5.1)$$

where constant T is a threshold and $\omega \in \Omega$. The set Ω contains σ and all possible scale combinations, e.g. the average (*avg*) of all scales. The square brackets denote boolean variables throughout this chapter. By searching the *local maxima* of the E_σ -operators in a PCF area we usually get the exact position or a position quite near to the manually marked corner or line-end. In formula a *marked corner* (MC) satisfies the following condition:

$$\text{MC}_{T,\omega}(x, y) \equiv \left[\text{PCF}_{T,\omega}(x, y) \wedge \forall_{(x_1, y_1) \in B_n(x, y)} (E_\omega(x, y) \geq E_\omega(x_1, y_1)) \right], \quad (5.2)$$

where $B_n(x, y)$ is a disc with center (x, y) and radius n . In the discrete case it means that coordinate (x, y) and its neighbors within radius n are taken, when the response $E_\sigma(x, y)$ is equal or stronger than all neighbors it is marked as a corner, line-end, or blob. The definition of marked corner is chosen in such way that in one PCF area at least one position is marked as a corner and at most all the possible positions in the PCF area. There is always one since we can always find a maximum $E_\omega(x, y)$, where (x, y) is a coordinate in the PCF area. In case $E_\omega(x, y)$ are equal for all coordinates (x, y) in the PCF area, all coordinates will be marked.

5.3.1 Potential corner features and marked corners

Figure 5.5 is a difficult test-case for corner detection, because it has a fine texture as background and some corners are rounded, vague, or with low contrast. If we take different scales

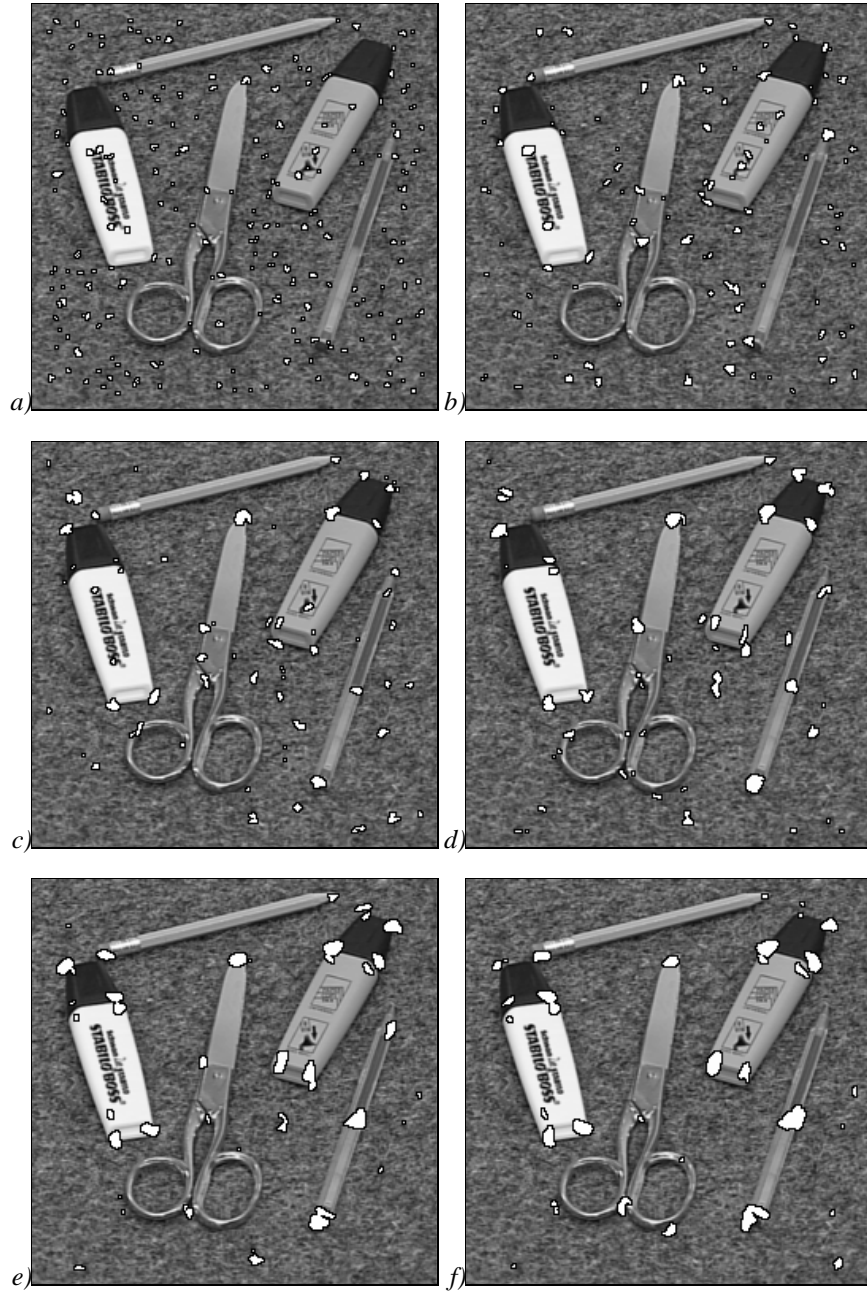


Figure 5.6: a)-f) Potential corner features of the image illustrated in Figure 5.5 using receptive field radii from $r = 10$ (a) to 35 (f) with linearly increasing steps of 5.

separately as illustrated in Figure 5.6, we obtain PCF areas in the background for small scales. Also middle scales show over 50% percent falsely detected potential corners features and even the most reliable scales still show false responses at the round part of the scissors.

If small scales are used, we get very small PCF areas, responses at a corner are very accurate since the areas contain a single point only, but most of the PCF areas do not contain a corner but small details of the background. As the receptive fields increase, the size of the PCF area increases too, and the number of PCF areas in the texture which are not perceived as a corner or line-end decrease. Figure 5.6c shows areas at all corner-points but also a considerable amount of non-desired PCF areas. In the largest scale the areas are large and some corners are not covered by a PCF area, but almost all areas where we do not perceive a corner or line-end are gone.

It is important to notice that for example the corner at the top of the pen is covered by a PCF area in the small and middle scales. The corners at the top of the left marker appear in the middle and large scales only, this makes it clear that multiple scales are not only necessary to obtain all corners but also to eliminate all PCF areas which do not contain a corner or line-end.

The MCs at the smallest scales are at the proper locations of the prints on the markers. Most of the corners in the image are slightly rounded therefore there are no marked corners at the smallest scales. The drift of corners is small at the left marker in the image. All six possible corners are marked at the same position in Figure 5.7a-e. Usually the positions of the MCs are well positioned if we do not consider the marked corners at the background.

5.3.2 Comparison with other corner operators

Using a single scale

In Chapter 4, we used six different standard corner operators. The outputs of these operators are at a single scale, multiple scales can be obtained by blurring the input image with Gaussians with a constant σ and by down-sampling the input image, which is a pyramid structure, or by taking different σ for the Gaussian kernels, which is called a scale space.

In Figure 5.8 we used a Gaussian kernel with $\sigma = 3$ for I_x , I_y , I_{xx} , I_{xy} , and I_{yy} , with such a σ no responses appear in the background and it is small enough to have still enough detail to detect most of the corners. The choice of different Gaussians does not improve the responses at the corners. Responses at ragged edges are caused if a smaller σ is chosen. The choice for a larger σ will cause lack of response to proper corners.

The outputs of five different standard operators, illustrated in Figure 5.8, all show good corner responses at the texture of the markers. The Diff- and DET-operators are sensitive to large differences in contrast, this is illustrated well at the light-dark edge of the left marker. They both respond well to most of the positions on this edge. The Diff-operator performs poorly at the left background edge, where several positions are marked as a corner. The Kitchen-Rosenfeld operator does not respond well to any corners of the markers, it responds mainly at positions with strong contrast. The Plessey feature point- and Plessey-operator respond rather well to the corners of the markers and the pencil. The responses at the round parts of the scissors are numerous and most of these responses are at a wrong location. All

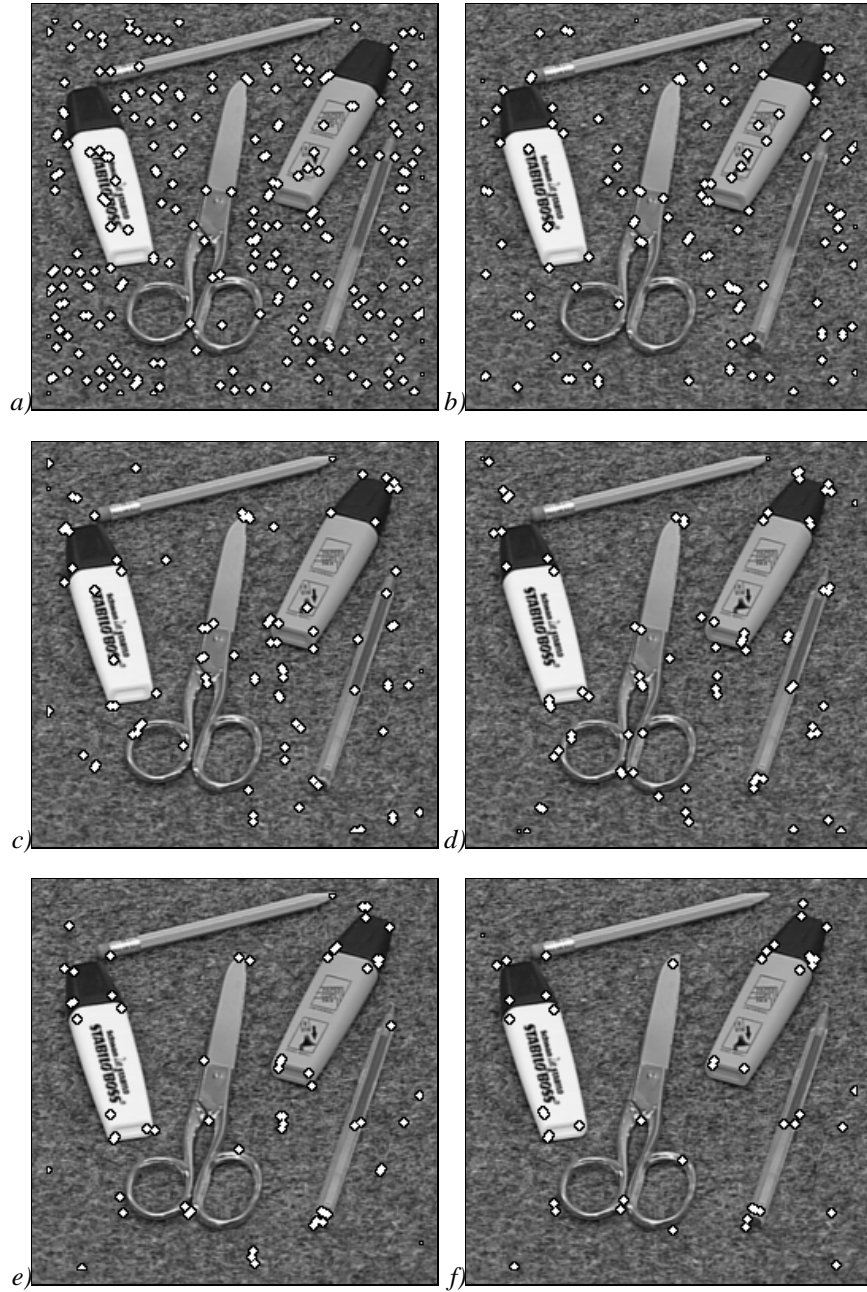


Figure 5.7: Marked corners (MC) of the potential corner feature areas of Figure 5.6a-f. For better visualization the MCs are displayed larger.

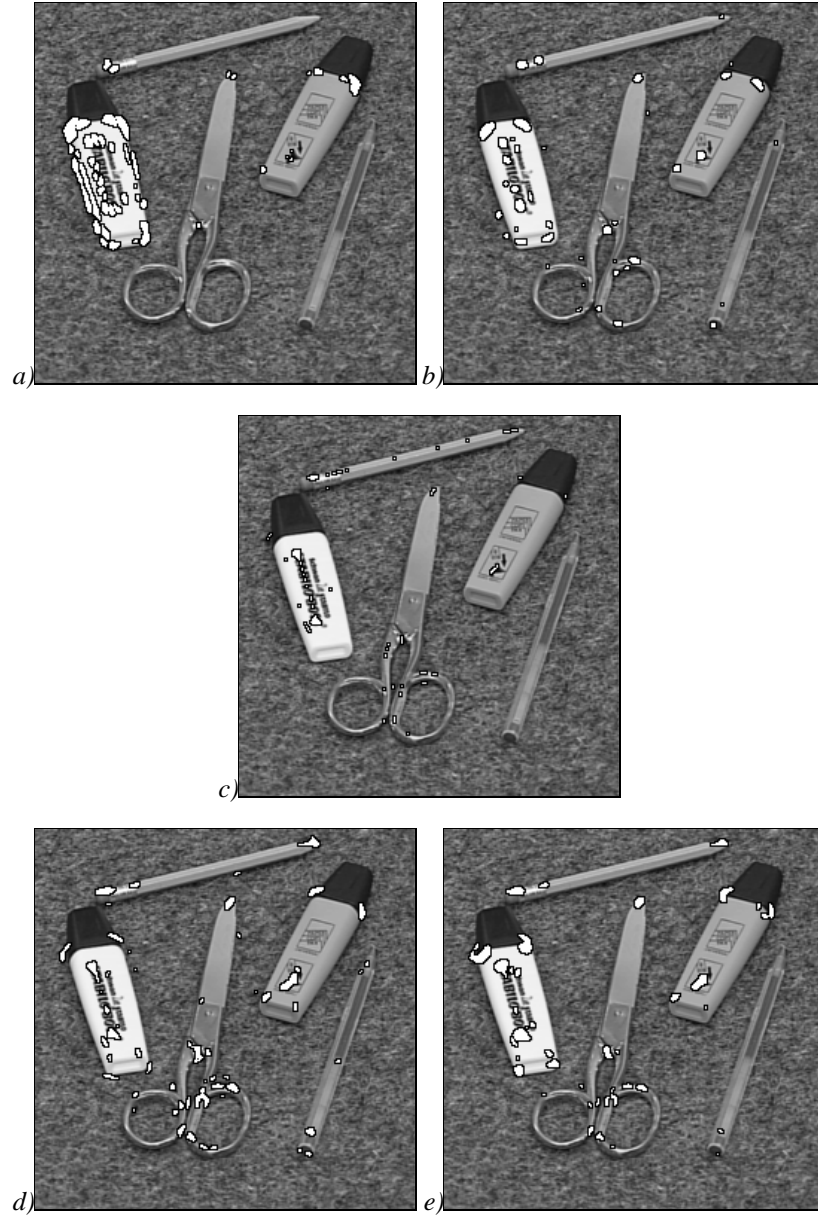


Figure 5.8: The thresholded results of the outputs at a single scale of different operators which are applied to Figure 5.5. a)-e) The PCF areas obtained with the Diff-, DET-, Kitchen-Rosenfeld, Plessey feature point, and Plessey-operator, respectively. The outputs of the operators are thresholded at 100, 2, 3, 15, and 0.4. Lowering the threshold will give more PCF areas but not at the desired positions, i.e. the perceived corners. The parameters used are: $\mu = 1$, and $\sigma = 3$ for the Gaussian kernel.

operators do not respond to corners with low contrast, for example the pen at the right of the image, the two Plessey operators respond to one corner of the pen only. The operators all respond to the rounded parts of the scissors. Local detail is caused by reflection of light, which causes a strong contrast. The Gaussian curvature operator, from Chapter 4, did not respond at all to proper positions, the responses were placed at random in the input image, therefore it is not illustrated.

If we compare the E_σ -operator with the standard operators, we conclude that the E_σ -operator responds properly to almost all corners in the image but also detects a lot of noisy corners. The standard operators do not respond at all to the fine background texture but fail to respond to certain corners. The performance of the E_σ -operator will be far better than the standard operators if we succeed in separating the response to fine texture from the real corners.

Using multiple scales

Of course, a fair comparison is only possible if a multiscale combination is also used for standard corner operators. To improve the results of the standard operators we combined the operator outputs at different scales. Before the different scales are combined, the positive parts of the derivatives of the Gaussian kernels are normalized, like we did with the Gabor functions (3.13) and (3.14). The responses at different scales are then comparable, if the kernels would not have been normalized the responses at small scales are far stronger than the responses at large scales. The different scales are combined with an average operator. The choice for this operator is motivated in Section 5.4. The results of the multiscale standard corner operators are illustrated in Figure 5.9.

Combining different scales for the standard operators have improving effects in the sense that more marked corners are found near the exact locations of the corners. By combining scales the operators do not lose their properties, which means that if the operator responds well to strong contrast at a single scale, it will still respond well to contrast when multiple scales are used. The Diff-, DET-, and Kitchen-Rosenfeld operators at multiple scales are not very useful when applied to Figure 5.5 as illustrated in Figure 5.9a-c, there are too many responses at wrong positions. The performance of the two Plessey-operators is better but still a considerable number of false responses appear in the image (Figure 5.9d-e). The false responses can be reduced if the threshold is increased but then proper corners are lost too and that is not desired. It is better to detect all corners and to have false responses, than to miss some corners, because false responses can still be eliminated at higher level processing, see Chapter 8.

5.4 Combining scales

The combination of scales is supposed to improve the results we obtained in the previous sections with single scales. Different scales should be combined in such a way that responses to noise or fine texture disappear and that the corners found are more reliable than at single scales. The results depend on the number of scales, the receptive field radii of the different

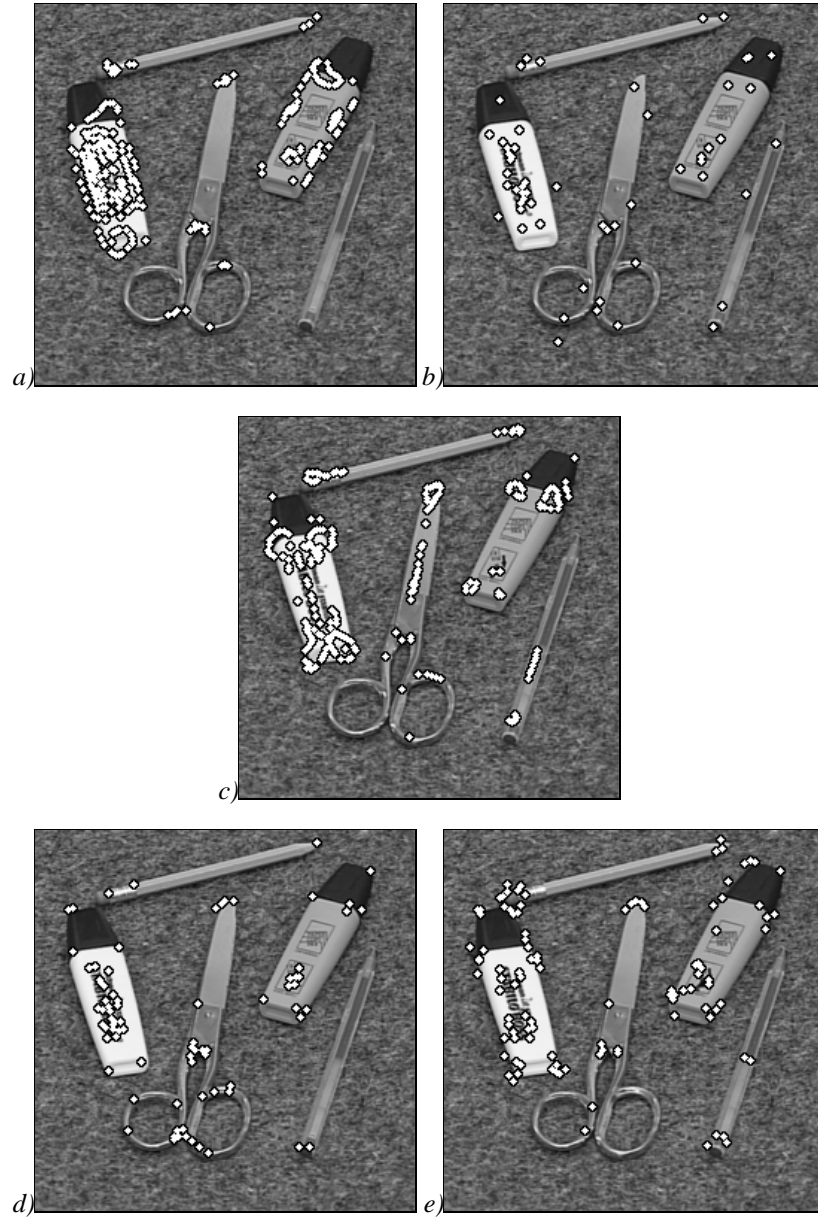


Figure 5.9: The marked corners at multiple scales (combined with the avg-operator) of different standard operators which are applied to Figure 5.5. a)-e) The marked corners obtained with the Diff-, DET-, Kitchen-Rosenfeld, Plessey feature point, and Plessey-operator, respectively. The outputs of the operators are thresholded at $5 \cdot 10^4$, 200, 25, 10^6 , and 0.005. The parameters used are: $\mu = 1$, and linear increasing receptive fields ranging from $r = 5$ to $r = 40$ with a step size of 2.5.

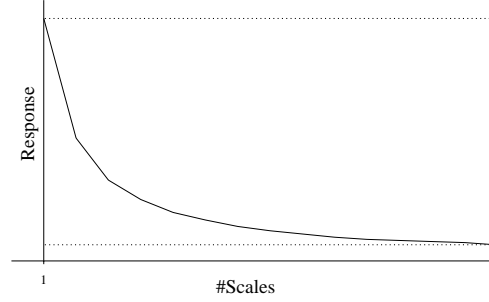


Figure 5.10: The threshold (solid line) for the average output of the E_{σ} -operators with identical centers but S different scales. If the responses are above this threshold the corner will be accepted and otherwise rejected. Horizontally the number of scales is illustrated and vertically the average response. The two horizontal dotted lines denote the minimum and maximum criteria, i.e. if a response at a single scale is below this minimum it will be rejected or if a single scale is above the maximum criterion this response will be accepted.

scales, and the kind of operator used to combine different scales. The following properties of the responses of E_{σ} -operators should be taken in account when combining different scales. We have found that the observations made in the previous sections carry over well to real-world images:

1. The outputs of the E_{σ} -operators at a corner or line-end are (almost) identical at all different receptive field sizes.
2. Usually if a corner is detected with a receptive field radius r , it is detected also in the “neighboring” receptive field radii $r \pm r'$ (In the simulations $r' = 2.5$ or 5 are used).
3. The E_{σ} -operators respond well to sharp corners at all different scales but they do not respond to rounded corners, except for large r .
4. The responses of the E_{σ} -operator at large scales give large spots as potential corner feature areas. These spots cover the proper locations.
5. The E_{σ} -operators at small scales locate corners very well but are very sensitive to local changes (high frequency noise) and therefore less reliable.

A coordinate is marked as a corner if the output of the combining operator of the E_{σ} -operator with the same centers but different scales is stronger than a certain threshold. This threshold usually depends on the type of combining operator. The combining operator is between two thresholds, a minimum threshold which is a constant value larger than zero and a maximum threshold which is a constant value smaller than the strongest response possible response. For example if we consider the average-operator then the minimum threshold is reached at the maximum number of scales and the maximum threshold at one single scale,

as illustrated in Figure 5.10. The solid line in the figure gives a criterion for marking a corner, if it is above the line the corner is marked and otherwise it is rejected.

Elimination of responses to noise or a fine texture seems to be quite easy since they usually occur in small scales but when a response is considered not to be noise but a real corner depends on the choice of the thresholds, for example if the response of the E_σ -operator in one of the small scales is very strong (above the maximum threshold) it can be marked as a corner. The minimum threshold should be chosen in such a way that a corner with low contrast and therefore weak responses which appear in all scales should be marked as a corner. This implies that the E_σ -operators which give weak responses at a wide range of scales will be marked and that small responses at certain scales which are usually no corners are ignored.

5.4.1 Equidistant versus scale-dependent scale sampling

The combination of different scales does not only depend on the type of combining operator but also on the number of scales and the sizes of these scales. The scales are proceeding in a certain way. Two types of progression are possible: linear, or non-linear scaling.

We tested two methods: a linearly increasing scale and an increasing scale which is ordered in such a way that we have a constant step-size in frequency domain. A constant step-size in the frequency domain is used to spread the information equally, therefore it is used more often. Shifting such a non-linear scale is more easy, for example if the smallest scale causes a lot of responses to noise one can “shift” the scales and eliminate the smallest scale. The linear scale has the advantage that rounded corners are detected better since it yields relatively more large receptive fields. In Section 3.5, differences between a linear and non-linear scale in both spatial and frequency domain have been elaborated. Most receptive fields of the non-linear scale are small in spatial domain, small scales are sensitive to fine textures or noise. Combining scales will give numerous useless corners if the majority has a small scale, as illustrated in Figure 5.12a. Hence less high frequency scales should be taken since they are noise sensitive. To avoid responses in noisy areas, we should make a choice which of the different scales to take. By selecting specific scales in frequency domain we can avoid the problem of fine texture or noise responses but it is easier to take linearly increasing receptive fields to get a non-linearly increasing frequency map (Figure 5.12b) with relatively little small scales, since the actual sampling is less important than the relative influence of large and small scales.

In the simulations fifteen different scales are used, they perform considerably better than when eight or four different scales are used. The results obtained with 31 or 63 scales are not much better than with fifteen scales since the overlap of two neighboring scales gets bigger when 31 or 63 scales are used and therefore additional scales do not add extra information.

5.4.2 Classifying a corner by its scale signature

Before combining different scales it is useful to see how the responses are distributed over different scales for corners. We used 15 different linear increasing fields ranging from 5 to

40 with a step size of 2.5. By using multiple scales we can evaluate the response of the E_σ -operator at a manually marked corner. For example if the responses appear in large scales only we can conclude that such a corner is different from another marked corner where the responses of the E_σ -operator appear in small scales only. Corners are classified by their responses at different groups of scales. By classifying corners we get an additional feature attribute. A coarse subdivision of different groups of corners is made by evaluating the responses at different scales. The results of the responses at corners are shown in Figure 5.11. Note that the use of attributes speeds up the search if a set of corners needs to be compared with another set of corners.

The group with small scales are responding at receptive fields with a radius smaller than 20, and is called *group small* (GS). In the group with large scales *group large* (GL) responses for radii that are at least 15 are found only. Two groups have responses in the “middle” scales, in the first group with middle-large scales *group middle large* (GML) for radii that are at least 10 and there is response at one or more receptive fields with radius 25 or larger. In the other group middle-small scales *group middle small* (GMS) are found, i.e. corners in this group respond at scales with radii less than 30 but there is at least one scale with a radius of at least 20 that responds. The fifth group *group all* (GA) is for the corners which show a response at all scales, all except the smallest scale, or all except the largest scale. The sixth group is called the *rest group* (GR), it is for non standard responses, for example when responses appear in two or more clusters of scales.

Corners found in GS are typically “detail” corners, for example the positions where a keyhole begins and ends, where the black text on a marker ends, or the end-points of corners where thin line-segments end. Corners in GL respond to rounded corners and ends of rounded bars. In the group GML a background corner was found, coincidentally three black line segments came together in this point which caused the response. In this group, rounded corners are found also. Corners found in GMS are sharp corners with usually small line-segments or edges, the top of a ball-point is such a corner. In GA corners with long straight edges, line segments, and also corners with low contrast are found. In Figure 5.11e there is a corner where the response of the E_σ -operators does not exceed 12 percent of the strongest response of all tested corners in Figure 5.11. This figure also illustrates that in natural images corners frequently occur in all scales except the smallest scale. The rest group (GR) contains corners which are split into two clusters: they respond well at small scales and at large scales but not at middle scales. These corners are found at positions where e.g. at detailed scales it is a corner and at large scales it is a line-end.

5.4.3 Combining scales with different operators

Since we know what kind of response can be expected at different scales simple operators are defined to combine different scales. The operators should be biologically plausible.

Let us introduce a general operator which sums over all receptive fields:

$$E_\Sigma(x,y) = \sum_{\sigma} c_\sigma \vartheta(E_\sigma(x,y)), \quad (5.3)$$

where c_σ is constant and ϑ is a neural function. The output of E_Σ is strong enough even

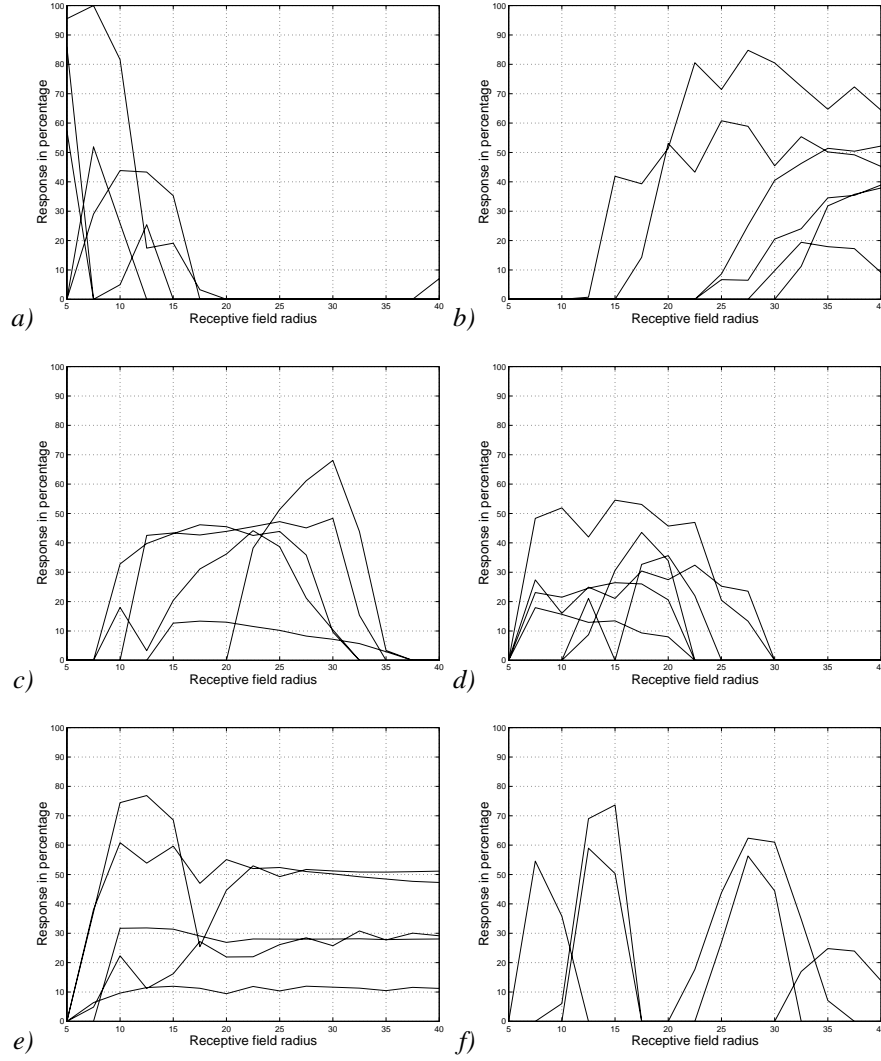


Figure 5.11: The responses at different MCs using 15 linearly increasing scales ranging from 5 to 40. The responses are divided into five different groups. The division depends on the response of the E_σ -operators at the 15 different scales. a) Responses of the E_σ -operators are found at small scales only. b) Responses found at large scales only. c) Responses found at middle and large scales. d) Responses at small and middle scales. e) Responses at all scales. f) Responses found in different clusters of scales belong to the rest group.

if the responses of all different E_σ are weak, which is the case when the contrast around a corner is low. If the response of a single scale or a few scales is high, the output of E_Σ is also high enough. The operator will respond properly to rounded corners because the sum of the

responses in large scales will be sufficient. Corners in small objects will be found since they respond strongly at small scales.

In the simplest case $c_\sigma = 1$ (for all c_σ) and $\vartheta(x) = x$, the operator will sum over all scales and the total response is used for corner detection. Since a comparable response is desired when the operator is used for a different number of scales another operator is introduced. If we assume that all $c_\sigma = \frac{1}{S}$ then the operator scale independent:

$$E_{\text{avg}}(x, y) = \frac{1}{S} \sum_{\sigma} E_{\sigma}(x, y), \quad (5.4)$$

which we call the average or avg-operator. This simple operator is biologically plausible and has only one parameter: the number of scales. The number of scales should be chosen such that the overlap in Figure 3.7b is sufficient, but also the overlap should not be too large that results of neighboring scales are almost identical. Increasing the number of scales has the advantage that a strong response in a single scale has a less strong influence. For example consider a threshold of $T = 5$ and a number scales $S = 15$, the response of a single scale must be at least 75 to be marked as potential corner feature. The results improve considerably when 15 scales instead of 8 in the range from 5-40 are used, because the absolute threshold ST decreases the chance that a single scale coordinate is marked as a corner.

Figure 5.12 illustrates marked corners of 5 different images using the avg-operator. Figure 5.12a shows the MCs of a non-linear scale from (3.27) with radii ranging from $r = 5$ to $r = 40$. This choice of radii is not useful since too many small scales are used and MCs appear everywhere in the background. In Figure 5.12b-f a linear scale with radii $r = 5 - 40$ are used. The choice of a linear scale with these radii is a good one. In Figure 5.12b, there are only 3 of the 55 marked corners in the background.

In the “Door” image (Figure 5.12d) sixteen corners are found. Using a single scale in this image will give several responses at the area to the right of the door, using multiple scales avoids this problem. Although the image has no false responses, there are two missing corners near the door handle. However when not only a gray channel is used for detecting corners but also different color channels the two corners are found. In Chapter 7 different color channels will be introduced and used for corner detection. Figure 5.12e contains a rounded corner and a corner with low contrast both will not be detected if a single scale is used. The edges in the image seem smooth but they are in fact ragged, which gives false responses at single scales.

The “Building” and “Text” images, Figures 5.12c and f, respectively, contain a lot of detailed corners close to each other, although corners appear almost all at proper locations it is recommended to enlarge the image. Two of the three marked corners classified in the rest group are found in the “Text” image between the O and G in the word “programming”.

5.5 Summary and discussion

In this chapter we elaborated on the output of the E_σ -operator for corners with convex angles between 3 and 180 degrees. The output of the operator was the same at every scale, except in the smallest scale but this is due to aliasing effects. At angles larger than 125 degrees

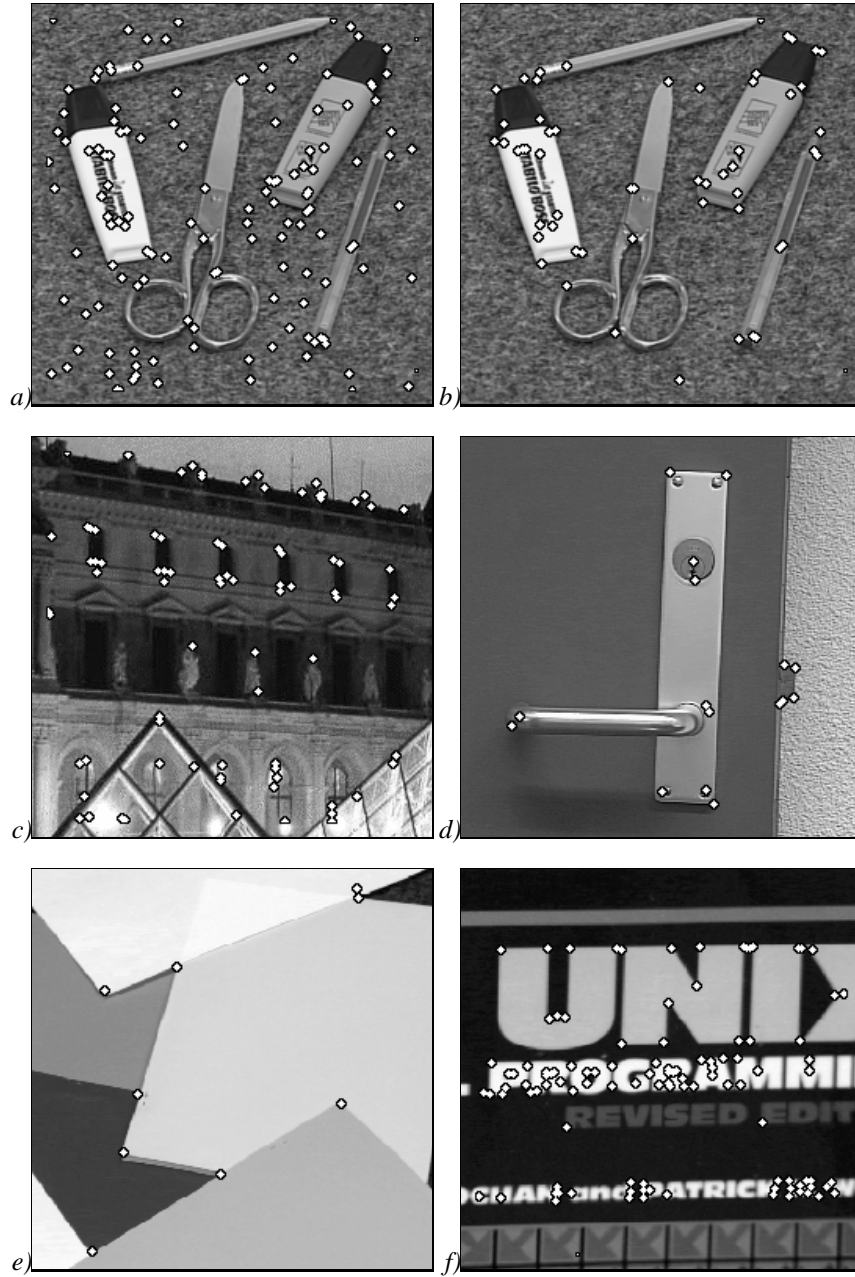


Figure 5.12: a) Detected corners at a non-linear scale. b)-f) Detected corners at a linear scale applied to different natural images. The avg-operator from (5.4) is used with parameter $S = 15$. For marking corners a threshold $T = 5$ is used.

the operator did not respond at all. This is a limitation of the operator. The response of the operator can be improved by changing the S -operators in such a way that they are sensitive to a smaller orientation bandwidth.

The drift of a corner is strong when two (or more) positions can be marked as a corner, which we called conflicting corners. At such a corner the position is dependent on the scale, actually this means that the position of the corner differs when we look at the details in the image or look at the image globally.

The operator responds well to all junctions, except 4-junctions (crossings). Peterhans et al. [149] and van der Zwan et al. [190] confirm that the end-stopped cells respond to T-junctions and only weakly or not all to X-junctions when they studied the role of end-stopped cells in mechanisms of figure-ground segregation. Until now nobody studied biological end-stopped cells for other junctions, so there is no biological evidence that end-stopped cells do respond to n-junctions, where $n \geq 5$.

A relation between the rounded corners and end-stopped distance $d\sigma$ was found. If 80 percent of the maximum response is used as a threshold for a corner to be marked (Figure 5.3d), the maximum radius for a detected rounded corner is almost equal to $d\sigma$. This is because the entire corner plus parts of the straight lines must fit in the receptive field. Thus, they should be rather large and therefore the position of the corner is less accurate compared to the location of the corner when a smaller receptive field is used. Although large scales are less accurate, it is still better than to detect nothing at all. If a response is found in large scales only, it is most probably a rounded corner and the location of such a corner is ill-defined anyway.

The goal of corner detection is to mark a certain location where a corner is found, and therefore two terms have been introduced: a potential corner feature (PCF) area which is the area where responses of the E_σ -operator exceed a certain threshold, within this area the positions of the locally maximal responses of the E_σ -operator are marked as corners (MCs). This implies that the MCs are a subset of the PCF areas, within a PCF area there is always at least one MC. By definition, MCs contain a single point, although in the figures they are displayed larger. In the smallest scale (Figure 5.7a) it is clearly visible that at the borders of the image there are no MCs, neither PCF areas in Figure 5.6. The PCF areas at the borders can contain artifacts since the convolutions are implemented with wrap around conditions and the E_σ -operators at distance $d\sigma$ from the border of the image are set to zero, so if we consider the largest used scale about six percent of the boundary is omitted.

If we compare the results of the E_σ -operator at a single scale (Figure 5.6e) with the standard corner operators we find that the E_σ -operator performs better than all standard corner operators, when the best scale is chosen for both operators (Figure 5.8). However the E_σ -operator is more time consuming compared to the standard operators in the current implementation. The E_σ -operator detects almost all corners where the standard operators lack to respond to a lot of corners.

The use of multiple scales can improve corner detection when the proper samples and scale combining operator are taken. The actual sampling, the choice for a linear or non-linear scale, is less important than the relative influence of small and large receptive fields. We chose an average operator to combine different scales, this is simple, independent on the

number of scales, and biologically plausible. Intuitively, it seems reasonable to have different receptive field sizes, which have the same receptive field center and functionality. There is neurophysiological evidence that there exist receptive fields with the same (or almost the same) receptive field center and functionality but with different receptive field sizes. From the work of Hubel [89] the sizes do not differ much, say about a factor of two in a simple cortex area between the smallest and largest receptive field. In area V5, Zeki [204] studied the receptive fields of cells in a long oblique penetration and found the first sixteen cells to have relatively small receptive fields and the next group of cells to be significantly larger (about 10 times). Probably in most areas receptive fields also differ in size but we should be careful, since there is no direct evidence that the difference between the smallest and the largest receptive field (with similar functional properties) is large and that every size between the smallest and largest receptive field is available.

The results obtained with a multiple scale corner detector (avg-operator) give better results than the standard multiple scale corner operators since the avg-operator detects almost all corners and has only a few false responses. The standard operators are not able to find certain corners. This has to do with the type of corner (e.g. rounded or low contrast) and not with the number of scales. Using multiple scales for the standard operators is recommendable but less successful than the avg-operator.

In all our simulations we used 15 scales in the range from 5-40 with a step size of 2.5. To combine different scales the avg-operator and a constant threshold of 5 were used to mark the possible corner features. The use of multiple scales not only gives better results but also an extra feature attribute which delivers information about the scales a corner is detected and thus some information about the corner. We created six different groups to give additional information about the corner. They are classified by checking where the responses of the E_{σ} -operators appear: at small, middle, large scales, or a combination of them. The size of a PCF area can be used as another feature attribute, to give an indication how big the possibility is that there is a corner in the area.

A more stable response at location (x,y) for a certain scale can be obtained by taking a maximum local neighborhood. Such neighborhood can be constant at all scales but also increase with the receptive field size. Small receptive fields respond to detailed information in the input image and are not location robust, large receptive fields show a response at a certain location (x,y) that does not differ much from its direct neighbors. We tried the maximum response for a constant size, with its 8 direct neighbors the results were good. There were however artifacts: the possibility that a corner is marked which has no response at all and the possibility that all the neighboring coordinates get the same maximum response from the center coordinate and therefore all points are marked as a corner. With 24 neighbors the positions of the corners were not as accurate as the positions of the corners for 8 neighbors. We conclude that a small neighborhood can improve the stability of the responses but that one can get some non-desired artifacts.

6

Graph Extraction from Images

Then the Lord answered Job out of the storm. He said: "Who is this that darkens my counsel with words without knowledge? Brace yourself like a man; I will question you, and you shall answer me." Where were you when I laid the earth's foundation? Tell me, if you understand. Who marked off its dimensions? Surely you know! Who stretched a measuring line across it? On what were its footings set, or who laid its cornerstone— while the morning stars sang together and all the angels shouted for joy?

Job 38:1-7 (NIV)

A SIMPLE ALGORITHM WILL BE PROPOSED that extracts segments (lines, edges, and curves) from an image. The detected corners obtained by the corner operator described in Chapter 5 and an edge enhanced image are used by the algorithm to extract the segments. A segment will only be added to the set of segments, which is represented as a chain of edges, if the segment has a corner as start-point and a corner as end-point. The output of the algorithm is a set of segments. Corners and the set of segments represent vertices and edges of a graph, respectively. These graphs will be used for object detection and recognition by using graph matching, which will be elaborated on in Chapter 8.

Standard algorithms for edge extraction like the border-tracking algorithm (Haralick and Shapiro [61]) or edge following as graph searching (Ballard and Brown [8]) are known. A more widely used algorithm is the Hough transform (Hough [78]). This transform is used for detecting straight lines in images (Duda and Hart [37]) and has been further generalized to detect curved segments which are specified by a sequence of boundary points (Ballard [7]). The algorithm we propose here differs from these methods because the set of corners is used as an input parameter, and the algorithm gives as result a set of segments. Since every segment has a corner as start- and end-points, postprocessing (e.g., edge linking) to compensate for gaps and noise is not necessary. In Chapter 8 we reduce the extracted segments to straight edges but one can obtain an extra matching criterion by comparing two segments with each other.

In Section 6.1 we give a global description of the algorithm and give definitions for enhancement, detection, and extraction. In this section we also define our representation of

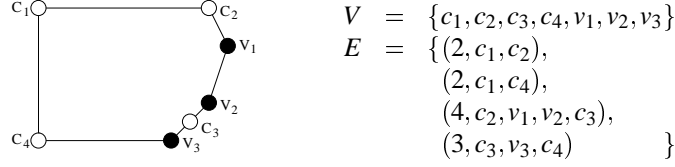


Figure 6.1: Representation of a graph. The vertices c_i represent the corners and the other vertices v_j are used as additional vertices to approximate the segments. Note that vertices v_j are not necessary if the segment is straight.

the graph, which differs marginally from the standard graph representation. Section 6.2 describes the algorithm in more detail. Section 6.3 illustrates some results of extracted graphs from real-world images. Section 6.4 describes related work, and the chapter ends with a summary.

6.1 Segment extraction algorithm

The terms edge enhancement, edge detection, and edge extraction are not used uniformly in literature. We assume that a square $N \times N$ sized discrete image I has elements $I(x, y) \in \mathbb{R}$, for all $0 \leq x < N$ and $0 \leq y < N$. An *edge enhancing* operator is an operator on images I , yielding an edge enhanced image I_{enh} with all elements $I_{\text{enh}}(x, y) \in \mathbb{R}$. In this image high values appear at edges and lines and low values elsewhere. An *edge detecting* operator applied to image I results in a binary image I_{det} where all elements $I_{\text{det}}(x, y) \in \mathbb{B}$. If $I_{\text{det}}(x, y)$ is true (or 1) then point (x, y) belongs to a line or edge. An *edge extracting* operator is an operator on images I , I_{enh} , or I_{det} yielding a set of extracted edges or segments. An *extracted edge* and an *extracted segment* are represented as a pair of points and a chain of points, respectively.

Our algorithm includes an edge extracting operator that operates on an edge enhanced image. A *segment* is represented as $(n + 2, c_i, v_{k_1}, v_{k_2}, \dots, v_{k_n}, c_j)$, where c_i and c_j are corner points and n gives the number of points v_{k_l} needed to give a proper approximation of the segment. Note that a segment can be described also with a set of $n - 1$ edges. The set of segments and the set of corners are used to represent a graph.

A *graph* $G(V, E)$ consists of a set V of elements called *vertices* and a set E of *edges*. In our approach, we initially represent a set of corners as a set of vertices V and represent an edge by a tuple of vertices. We slightly adjust the graph representation by augmenting the vertices with two *coordinate attributes* x and y to indicate the location of the corner in a two-dimensional image. Since we use segments instead of edges we add a set of vertices v_{k_l} to set V , which are used to give proper approximations of the segments. The set E represents the set of segments, since this is more compact than writing every segment as a set of edges. An example of a representation of a such graph G with this notation is given in Figure 6.1. The vertices of the graph are represented as circles, and the segments as a chain of edges.

It is known that segment detection and extraction are difficult problems in discrete color or gray-scale images. It is difficult since a segment can have arbitrary length and is not guaranteed to be straight. The quality of segment detection and extraction strongly depends

```

1  Procedure ExtractGraph ( $C, I, \Upsilon, G$ )
2    segmentset :=  $\emptyset$ 
3    forall  $c \in C$ 
4      forall  $\sigma \in \Upsilon$ 
5         $O := \text{SetOfAnglesAtDistFromCornerWhereSegmIsDetInImage}(d\sigma, c, I)$ 
6        forall  $o \in O$ 
7           $p := c + \text{DistanceInDirection}(d\sigma, o)$ 
8          Store ( $p, P$ )
9           $o' := o$ 
10         repeat
11           FollowSegmentInDirection ( $o'$ )
12           Adjust ( $o'$ ) /* if necessary */
13            $p := p + \text{DistanceInDirection}(d\sigma, o')$ 
14           Store ( $p, P$ )
15         until StopCriterionFulfilled
16         if StopCriterionIsCorner ( $C \setminus c$ )
17           segment := Chain ( $P$ )
18           AddTo (segment, segmentset)
19   Optimize (segmentset)
20    $G := \text{Graph}(\text{segmentset}, C)$ 

```

Figure 6.2: Pseudo algorithm for extracting a graph G from an edge enhanced image I by using a set of corners C and a set Υ of scaling values.

on the results obtained by the edge enhancing operator. The image should have high values at edges and lines of the input image and low values elsewhere and, e.g., the values on a line should be high and not drop suddenly at a corner only. Also one would not desire high values where no edge or line is perceived.

We enhance segments in the image by using $C_{\sigma, \theta}$ -operators from Chapter 3. In order to obtain high values at lines and edges, we used the output of the C_{σ} -operators at multiple scales. Similar to the E_{avg} -operator also here different scales are combined with an averaging operator to a so called C_{avg} -operator. Combining is possible since strong responses (high values) at edges and lines of these operators are not shifting at different scales, unlike the S -operators. The C_{avg} -operator together with the set of corners which are obtained by using the MC-operator (5.2) is used as input for the algorithm.

The pseudo algorithm for graph extraction is illustrated in Figure 6.2, it contains five stages and uses as input a set of corners C , an edge enhanced image I , and a set of scaling values Υ , and gives as output a graph G .

The first three stages are the edge extracting operator (lines 2-18). Basically the first stage, line 5, is the initialization: the selection of orientations at a distance $d\sigma$ from the corner, where d is the distance constant (see (4.3)). In the direction of every selected orientation a segment will be followed (traced). This following of a segment is the next stage (lines 10-14). A segment is followed until it satisfies a stop criterion (line 15). A segment is added to

the set of segments if the stop criterion is another corner (lines 16-18). These lines represent the third stage of the algorithm.

During the following of a segment at every step a point is selected and stored. Normally it is not necessary to keep all the selected points to represent a segment. In the fourth stage redundant points are eliminated from every segment (line 19). The final stage (line 20) is the representation of the graph.

6.2 Segment extraction

In this section we will give a more detailed description of the segment extraction algorithm but first we give a complexity analysis of the algorithm in terms of K , the number of corners. The algorithm contains four loops. The three inner loops take constant time since Υ contains a constant number of scaling values σ , the selected number of orientations is bounded by the constant number of sample points, and also line following is bounded by a constant maximum number of steps. Therefore the algorithm is $O(K)$. Note that the total number of sample points will be never more than there are pixels in the discrete image.

6.2.1 Orientation selection

A corner (or line-end) contains implicit information (one or more segments start from this position) but it does not carry any information about the direction in which a segment is starting. In Chapter 4 we described that the response of the C_σ -operator near the location of corners is relatively weak. We showed that the responses of the C_σ -operator are strong when it is at some distance from the corner. Therefore we start searching for possible segments using a circle with radius $d\sigma$ and the corner as center. On this circle a number of samples are taken where the response of the C_σ -operator is determined. If samples at a certain position (x, y) are taken the chance that response of the C_σ -operators for a known σ is strong at a “noise” point is realistic. To reduce non-desired strong responses we combine the responses of the C_σ -operator at S different scales with an averaging operator:

$$C_{\text{avg}}(x, y) = \frac{1}{S} \sum_{\sigma} C_{\sigma}(x, y), \quad (6.1)$$

and take sample points from the output of this operator. In the previous chapter we subdivided corners into groups; some of the corners are found in all scales while others appeared in large or small scales only. If a number of samples on a single circle are used to determine the orientations of different segments then it is possible that the radius of the circle is taken too small and therefore the responses at the samples are all weak and no orientation is selected. It is important that all possible orientations are found, if a certain orientation is not found the segment with this orientation will not be detected.

We do not use a single circle but a number of circles with radii $d\sigma = \frac{4}{\sqrt{-10 \log e}} r$ (see (4.3) and (A.5)), where $r \in [10, 15, \dots, 40]$. To avoid discretization effects we omit the smallest radius $r = 5$. Figure 6.3a illustrates samples, represented as dots, on these circles. The number

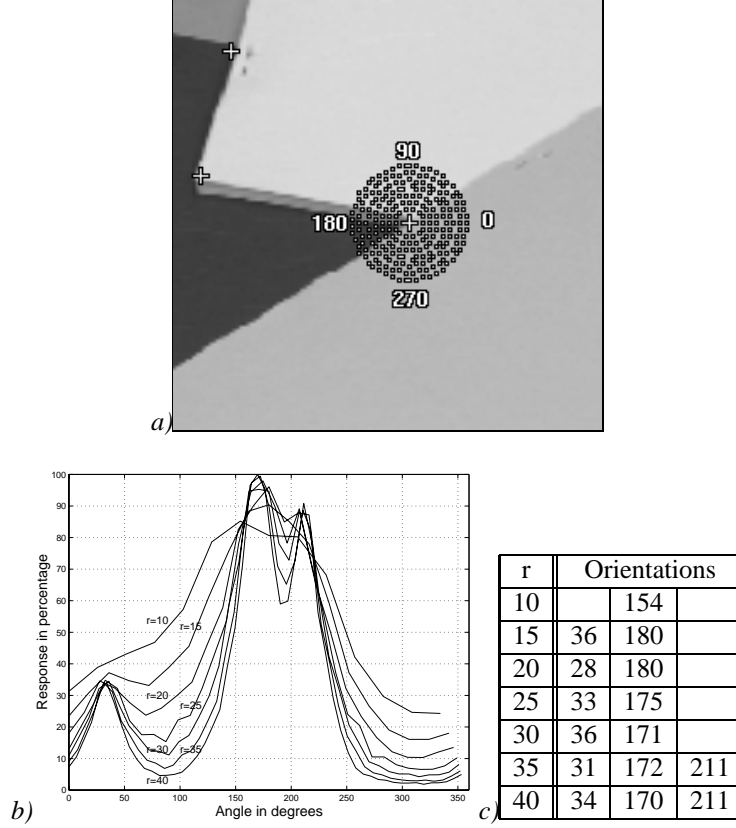


Figure 6.3: Finding the orientations of different segments starting from corner c_5 . a) Samples of seven circles for the determination of orientations are taken from $r = 10$ to $r = 40$ with steps of 5. b) The responses of the C_{avg} -operator along the circle starting from the right on the horizontal axis going round counter-clockwise. c) The orientations in degrees found for different radii r .

of samples Q_σ on a circle with radius $d\sigma$ is obtained in the following way:

$$Q_\sigma = \frac{\pi}{\arcsin\left(\frac{1}{d\sigma}\right)}. \quad (6.2)$$

The samples are ordered counter-clockwise starting at the horizontal axis on the right-most position. The angle of a sample is obtained by $\phi_j = 2\pi j/Q_\sigma$, $j \in [0, \dots, Q_\sigma)$. Pair $(d\sigma, \phi_j)$ should then be transformed from polar- to Cartesian-coordinates by using the corner as center. The responses of the C_{avg} -operator at these sample points on the circle are taken and illustrated in Figure 6.3b. For $r = 15$ to $r = 40$ the responses are all rather similar at the same orientation. This is expected since the edges are straight, so for every different radius the orientation of the edges should be the same. An exception is $r = 10$, only 14 sampling points are on this circle, this implies that a sample is taken with steps of almost 26 degrees.

A sample point with orientation ϕ_j is selected if the response of the C_{avg} -operator at the

sample point is a local maximum, is above a low constant threshold T , and is clearly stronger than one of the responses at the other sample points on the circle. In Figure 6.3a the measured angles of the three segments have an angle with the horizontal axis of 30, 167, and 211 degrees, respectively. The selected orientations for corner c_5 (for labeling of corners see Figure 6.6b) are given in Figure 6.3c. The selected orientations are close to one of the three measured orientations. In the largest circle the number of sampling points is far more than in the smallest circle and therefore the orientations selected are most accurate.

6.2.2 Following a segment

The selected orientations described in Section 6.2.1 are used to follow a potential segment. A condition must be satisfied: if we assume that a corner is located at (x_0, y_0) in the image and that the sample point on the circle with orientation ϕ_j is (x_1, y_1) , then the value of the \mathcal{O} -operator (see (3.23)) at (x_1, y_1) should be near to the selected orientation ϕ . Since the segment is not necessarily straight, ϕ does not need to be the optimal orientation, therefore we take a number of samples s_k , $k \in [0, \dots, K]$ between $\phi - \delta$ and $\phi + \delta$, all at a distance $d\sigma$ from (x_1, y_1) , as illustrated in Figure 6.4c-d. We choose $\delta = \pi/8$. We take the responses of the \mathcal{C}_{avg} -operator at these sample points and pick the sample which has the strongest response.

Assume that sample point (x_2, y_2) has the strongest response then we found a new position in the process of following. A schematic image is given in Figure 6.4d where one of the black circles is point (x_2, y_2) . We started at a corner (x_0, y_0) and then moved via (x_1, y_1) to (x_2, y_2) . Orientation ϕ is adjusted by taking the angle between the horizontal axis and the vector from (x_1, y_1) through (x_2, y_2) . We are ready to do a next step, until a stop criterion is satisfied.

At each step i a point (x_i, y_i) is stored, in this way we get a chain, i.e., an ordered set of points (x_i, y_i) , $i \in [0, \dots, n]$. This chain represents a curve from (x_0, y_0) to (x_n, y_n) .

6.2.3 Stop criteria

At every step we check if one of the five following different stop criteria is satisfied:

1. Another corner is found.
2. One of the sample points which is used for following is outside the image.
3. There is no sample where the output of the \mathcal{C}_{avg} -operator is above threshold T and with an orientation similar to the output of the \mathcal{O} -operator.
4. The response of the \mathcal{C}_{avg} -operator differs too much from the response at the previous step.
5. The segment is too long.

The criteria are applied in the order as they are described above. The first criterion is satisfied when a corner is found within a distance $d\sigma$ from the current point and is within an angle δ from the current orientation.

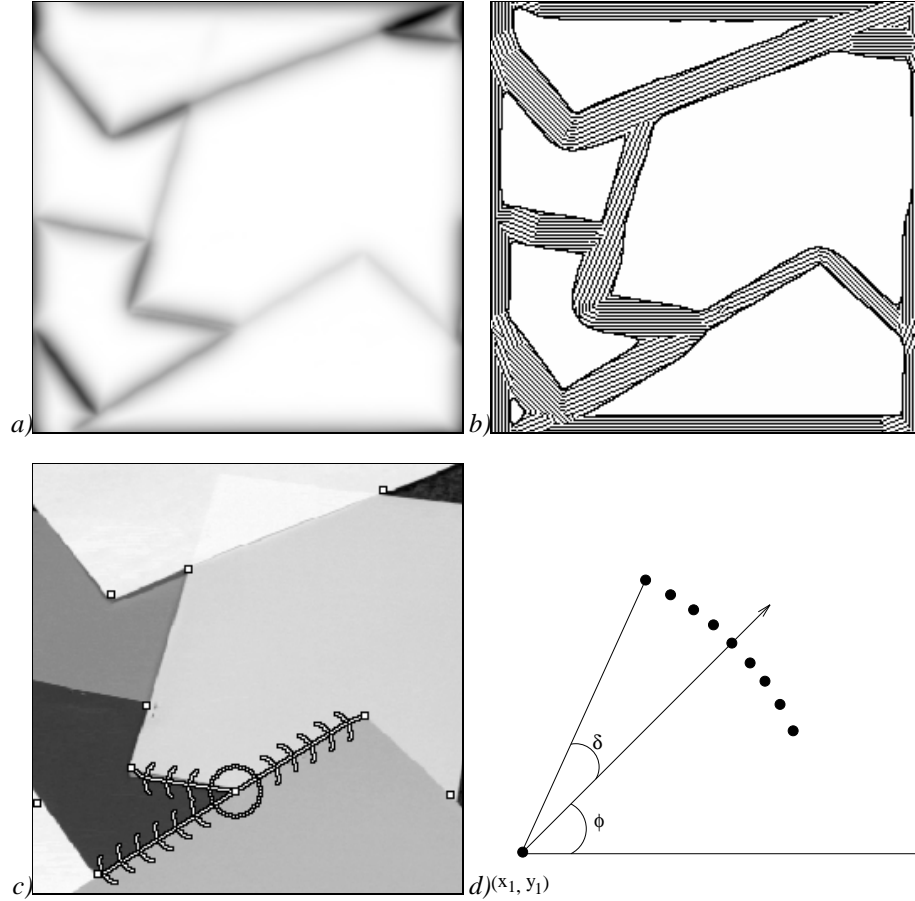


Figure 6.4: a) The outputs of the C_{avg} -operator, white denotes no response and black strong response. b) The output of the O -operator which gives the preferred orientations of the C_{avg} -operator. The image has 9 different patterns: completely white is the response below threshold $T = 5$. The eight striped patterns denote the preferred orientations with steps of $\pi/8$. c) Three possible orientations are followed for $r = 35$. Segment following is illustrated with sample points on a part of a circle. d) A schematic representation of one step in the process of following a segment.

The second criterion is a rather trivial one and can be avoided by zero padding the image such that the third criterion is satisfied before being outside the extended image.

The third criterion is fulfilled when the output of all C_{avg} -operators at the sampled points (illustrated as black circles in Figure 6.4d) do not exceed threshold T , or the current orientation ϕ between differs too much from the principal image orientation.

The fourth criterion is the difference in response of the C_{avg} -operator at position (x_{i-1}, y_{i-1}) and (x_i, y_i) , if the response differs too much, the possibility that this is still the same segment is rather small, since in such case we expect a corner to be found.

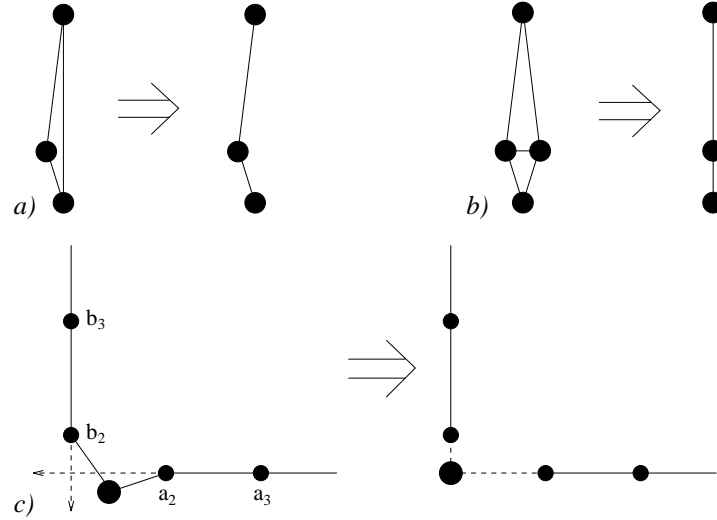


Figure 6.5: Graph improvements. *a) Eliminating a segment. b) Fusing two corners. c) Moving a corner to the most plausible position.*

The last criterion is the length of a segment, it should not be too long, i.e. if a segment is for example ten times longer than the width or height of the image one can doubt if the segment is a proper one. In such case the largest probability is that a circular path is being followed.

6.2.4 Optimizations

During the process of following a segment we keep a point (x_i, y_i) , $i \in [0, \dots, n]$ at every step. Point (x_0, y_0) is the position of the corner where we start from and (x_n, y_n) is the end-point of the segment, which is another corner. When a straight line between these two corners can be drawn, it is not necessary to keep all n coordinates, in fact we eliminate all coordinates $(x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ and keep the two corner points.

We apply the segment following algorithm separately for every scale ranging from $r = 10$ to $r = 40$. By using the same algorithm at several scales independently the chance that a segment is missed is rather small but it is possible that a segment is detected more than once. To avoid doubly detected segments we mark the pixels in the image where we have been before. This increases the speed of searching, avoids that the algorithm follows a circular path, and validates that at most all pixels in the image will be checked.

Further optimizations can be made by eliminating segments and by fusing corners. Figure 6.5a-b gives an example for both eliminating and fusing. Another improvement is to move a corner to a more predictable position by intersecting two extrapolated segments (Figure 6.5c).

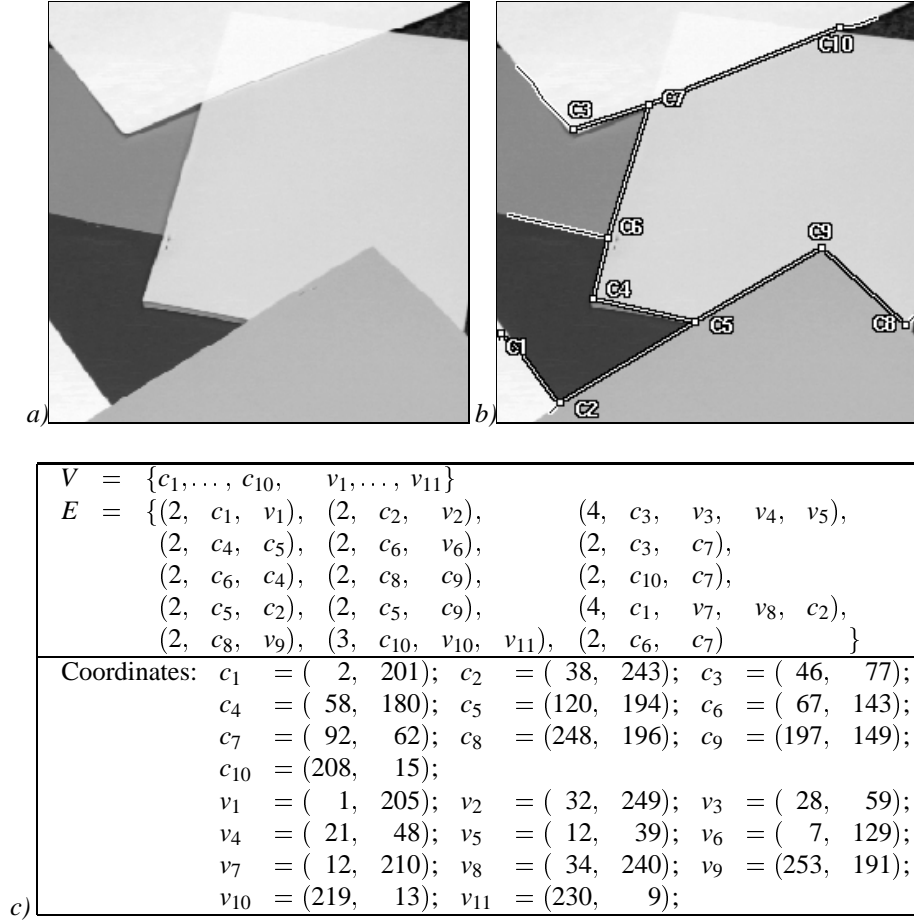


Figure 6.6: a) An input image sized 256×256 pixels. b) Exactly 10 corners are used as input. Detected segments between two corners are marked black-white-black and between one corner and the border of the image the segments are marked white-black-white. c) The graph extracted from the input image.

6.3 Results

In images where corners are not too close to each other and lengths of segments not too small, and at such a distance from each other that during the process of following there is no confusion possible between two or more segments the algorithm performs well, as illustrated in Figure 6.6. Although at first sight Figure 6.6a seems a trivial image, there are a number of difficulties to solve such as low contrast at edges and ragged edges. Also strong response differences between two different edges which are close to each other cause problems since the strongest responses is selected. This implies that the edge where the response is weaker will not be detected. An example of a non detected segment is the edge in the upper right

corner of Figure 6.6b. At starting corner c_{10} two line segments were found, one going towards c_7 and the other going towards the upper corner of the image. When a circle with sample points is drawn at a distance $d\sigma$ from corner c_{10} there are two orientations found only. The third orientation is not found because its sample point is too close to the sample point of the selected orientation. Normally when a segment is in between two corner points this is not a problem, because if for one corner the orientation is not found there is still a good chance that the segment will be detected if the other corner is used as starting point. The results illustrated in Figure 6.6b show that this is the only missing segment. The others are well on their places and close to or on the edge.

From Figure 6.6 we conclude that the results are very good since all, except one segment, are detected and all segments, except one, contain corner points only.

The performance is not always that good, for example in Figure 6.7a, we see that certain segments seem to be detected several times (left marker). The reason that a segment is detected several times is caused by corners which are relatively close to each other. The segment which will be followed is the one with the strongest contrast. This means that all corners near this edge will get the same segment. The pen and pencil have two parallel edges which are close to each other, close in the sense that during the process of following, there is at every step a possibility that the other edge will be selected. The pen has such a detected segment which “jumps” from one to the other edge. Similar problems are encountered in Figure 6.7c. In Figure 6.7b the lack of corners prohibits the algorithm to detect all segments. When edges are close to each other, the results can be improved by enlarging the image, for example Figure 6.7c shows segments which are “jumping” between an edge of U and an edge of R and O. When the image is enlarged two times these jumping segments disappeared (Figure 6.7d).

6.4 Related work

In literature there are a huge amount of line and edge enhancing operators. Well known are the operators proposed by Roberts [160], Sobel [174], and Prewitt [154]. Later Marr and Hildreth [129] proposed a Laplacian of Gaussian zero-crossing edge operator, its is because of its shape also known as the Mexican-hat operator. In 1986 Canny [19] proposed an operator which is almost identical with the operator suggested by Marr and Hildreth in the one-dimensional case.

Edges are usually detected by thresholding the edge enhanced image (e.g., Gonzalez and Wintz [56] and Haralick and Shapiro [61]). Choosing a threshold leads to broken edge contours. Pentland [145] solved this problem by averaging the edge strength of a contour over part of its length. If the average is above a threshold, the entire segment is marked, otherwise no part of the contour appears in the output. Canny [19] mentioned that it is very difficult to set a threshold so that there is a small probability of marking noise edges while retaining high sensitivity. Good results are obtained with the Canny edge detector, and later with the logical/linear operators (Iverson and Zucker [92]) and the suppression and enhancement scheme (Heitger [70]). However, these operators all aim at high sensitivity and do not guarantee closed edge contours, while we aim at moderate sensitivity and desire closed contours.



Figure 6.7: a)-f) Detected segments in complex images. For the selection of different orientations we used $r \in [15, 20, 25, 30]$.

To reduce the noise responses to a minimum we use the C_{avg} -operator as edge enhancer and get a moderate sensitivity.

The edge extraction algorithms like the border-tracking algorithm (Haralick and Shapiro [61]), the edge following as graph searching (Ballard and Brown [8]), and the generalized Hough transform (Ballard [7]) are all sensitive to gaps and noise. Therefore for all these algorithms postprocessing is necessary.

Up to now there is not an algorithm (found by the author) that detects segments by using corners as start- and end-points of a segment. Therefore no postprocessing such as segment linking is necessary.

6.5 Summary

We presented a segment extraction algorithm that uses the output of the corner detection algorithm from Chapter 5 as input. At a corner we try to find all possible orientations in which to follow a segment by using the response of the C_{avg} -operator. Corners do not only form the starting point but they are also essential for the first and most important stop criterion in the process of following a segment. When the algorithm is applied to all corners we make a graph representation. A technique called graph matching will be used to recognize objects in this graph (Chapter 8).

The proposed algorithm has several nice features:

1. The begin and end position of every segment, which are corners, are a priori known. Therefore no postprocessing is necessary.
2. Threshold T which is used to eliminate noise responses is a low constant which is the same for every image.
3. The responses to noise and background edges are highly reduced by using the C_{avg} -operator.
4. The way of selecting a number sample points avoids noise because the sample with the strongest response is selected only.

The results can be improved by using color to find additional corners. Due to undetected corners the segments between these corners will not be extracted too. This is the problem for the image illustrated in Figure 6.7b. In the next chapter we will use color input images, to improve corner-detection and therefore line-extraction.

Inclusion of Color Opponent Cells in the Model

And God said, "This is the sign of the covenant I am making between me and you and every living creature with you, a covenant for all generations to come: I have set my rainbow in the clouds, and it will be the sign of the covenant between me and the earth. Whenever I bring clouds over the earth and the rainbow appears in the clouds, I will remember my covenant between me and you and all living creatures of every kind. Never again will the waters become a flood to destroy all life. Whenever the rainbow appears in the clouds, I will see it and remember the everlasting covenant between God and all living creatures of every kind on the earth." So God said to Noah, "This is the sign of the covenant I have established between me and all life on the earth."

Genesis 9:12-17 (NIV)

COLOR PLAYS AN IMPORTANT ROLE in human vision. In the macaque monkey (and presumably in the human visual system) some 70-80% of the cells in the LGN are transmitting color information (DeValois [189]). In artificial color vision it can be illustrated with a simple example that information is lost when a color image is converted to a gray scale image. In previous chapters we used gray scale images with good results. This is not surprising since we are color blind in dim light and still find most corners and edges accurately. That gray scale images are not always sufficient is illustrated in Figure 6.7b, where two corners are missing. The corners will be found if color channels are used. With a simple image we will show that one achromatic channel is not sufficient but that chromatic channels are needed to find all corners and segments. It is conceivable, then, that many of the cortical complex and end-stopped cells respond to lines or corners, but that the information upon which they detect the presence of a line or corner may be chromatic as well as achromatic information (DeValois [189]).

In Section 7.1 we give a brief overview of color and color opponent cells. In Section 7.2 the necessary transformations from the (r, g, b) -format, which will be used throughout the

chapter, to other triples will be given, among them the response of the tristimulus. In Section 7.3 the four different types of color opponent cells will be modeled. An image is used to illustrate the response of these different cell types. In Section 7.4 we present a reconstruction model to show that the black-white, red-green, and blue-yellow opponency is sufficient to reconstruct the (r, g, b) -input image. In fact we create a transformation scheme from the three opponent types to the (r, g, b) -format. In Section 7.5 biological plausible double opponent color simple-, complex-, and end-stopped cells are modeled to find edges and corners in the chromatic channels. Although these cells are hypothetical they can be made plausible by using the building block architecture. The chapter ends with a summary and discussion.

7.1 Color opponent cells in the visual system

Color is commonly used, but what is color exactly? Fundamentally *light* is electromagnetic energy in the 400- to 700 nanometer wavelength part of the spectrum. *Color* is the perceptual result of light. The visible part of the spectrum ranges from violet to indigo, blue, green, yellow, orange, to red, which is illustrated in Figure 7.1. In 1704 Newton [140] reported a new theory about light and colors to the royal society. He split up white light with a prism and recombined the light with a second prism, obtaining white again. His discoveries led to the recognition that ordinary light is made up of a continuous mixture of different wavelengths in the visual part of the spectrum. Over the eighteenth century, it came to be realized that any color could be obtained by mixtures of light of three different wavelengths. The fact that color can be produced using three parameters (wavelengths, cones) was termed *trichromacy*. In 1802 Thomas Young [199] put forward a simple theory to explain trichromacy. He proposed that at each point in the retina there must be at least three light sensitive “particles” (cones) sensitive to the colors red, green, and violet. The physiological basis for the trichromacy of color vision was discovered in 1964 when Marks, Dobbie, and MacNichol [128] and also Brown and Wald [16] showed that as first suggested by Thomas Young, the human retina contains three cone types with different pigments.

Another theory is proposed by Hering [72, 73, 74]. In his theory, color vision depends on three opponent processes, black-white, yellow-blue, and red-green. Jameson and Hurvich [93, 94] generalized Hering’s theory to incorporate spatial interactions in the red-green and yellow-blue systems and to give physiological evidence for inhibitory mechanisms in sensory systems. Their work gives results similar to Land’s retinex (= retina-cortex) theory [105, 106]. His theory provides a framework for computing the color seen at a particular part of the retina on the basis of relative intensities of three wavelengths and their spatial interactions.

Cells which respond excitatory to one color and inhibitory to another color are called *color opponent* cells. These cells are found at the first levels of processing after the photoreceptors (DeValois et al. [30]). The activity of these cells follows Hering’s conjecture that color is analyzed with red-green, yellow-blue, and black-white opponents (Hering translated by Hurvich and Jameson [75]).

In the parvo-cellular layer of the visual system of the monkey¹ three different opponent

¹In this chapter with monkey we mean the species which have color vision, e.g. macaques.

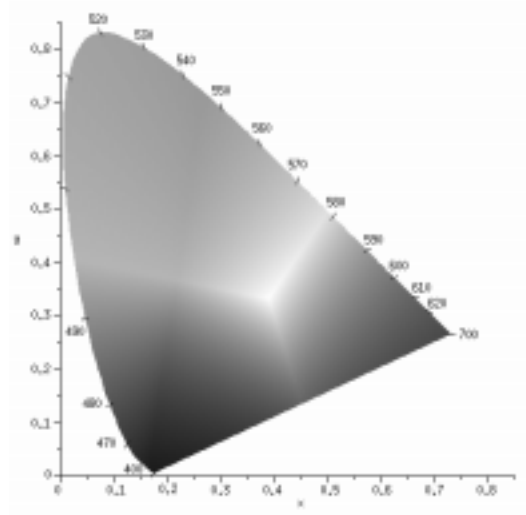


Figure 7.1: The CIE chromacity diagram. The diagram shows the visible part of the spectrum. The wavelengths around the periphery are in nanometer.

cell types are found (Hubel [89]). Cells of the first type are called *center-surround color-opponent cells*. This type is (like the other two color-opponent types) divided into two groups, the first group has red-green color opponency and the second blue-yellow opponency.

Cells of the second type are *center-only color-opponent cells*, and make up about ten percent of the four upper geniculate (parvo-cellular) layers.

Cells of the third type, about 15 percent of the cells in the parvo-cellular layers, have center-surround receptive fields and show no color preferences. Their receptive field size is about the same size as the first type, where the receptive fields of the second type are more like the receptive fields of the magno-cells which are about four times larger than the first and third type (Shapley and Perry [171]).

The opponent-color theory uses the above three cell types and combines the outputs from red, green, and blue (R,G,B) channels to give an achromatic channel (black and white) and two chromatic channels (red-green) and (blue-yellow). Analysis made by Gottschalk and Buchsbaum [57] points out that within the framework of information theory the color-opponent system is optimal. This implies that its components are orthogonal and therefore do not contain redundant information. In Section 7.4 we present a reconstruction scheme to show that these three channels are sufficient.

Another type of color-opponent cells are the *double-opponent cells*. Daw [25, 26] discovered them in the goldfish. These cells are found in the cortex of the monkey but not in the lateral geniculate nucleus (Hubel [89]).

Probably the most important phenomenon in the perception of color from the point of view of both every day life and survival, is the fact that the color of an object stays the same when the spectral content of the scene is varied. This phenomenon is called *object color constancy*.

Two other phenomena in the perception of color need a physiological explanation: these are *simultaneous color contrast* and *successive color contrast*. Many investigators have recognized that there is a relationship between object color constancy and simultaneous color contrast. This can be illustrated by considering the activity of a double opponent cell described in the visual system of several species. Its receptive field has a center and a surround with opposing responses for different colors in both parts of the receptive field. This accounts for its name: it has opposing responses for different colors and for different areas. A typical double opponent cell gives excitatory responses to red light and inhibitory responses to green light in the center of its receptive field, and inhibitory responses to red light and excitatory responses to green light in the periphery of its receptive field.

The double opponent cells are found in the cortical upper-layer cells in so-called *blobs*. Within the blobs of the macaque, Livingstone and Hubel found three groups of double opponent cells, known as *red-green*, *yellow-blue*, and *black-white* double opponent cells. The most common double-opponent cell has red-green color opponency (Livingstone and Hubel [117, 119]). It seems likely that cells in these blobs project to cells around the blobs, a few of which are specific for both color and orientation (Hubel and Wiesel [86] and Livingstone and Hubel [117]). In all species that have been studied, double opponent cells are found one stage higher than opponent cells and are probably formed by convergence of the opponent color cells (Hubel [86]).

7.2 Representation of color

The *tristimulus* theory, which is a part of the trichromacy theory, is based on the hypothesis that the retina has three kinds of color sensors (cones). These three cone types are sensitive to red (long wavelength), green (middle wavelength), or blue (short wavelength) lights. The terms red and green are a little misleading since their peaks are at 580 nanometer (orange) and 545 nanometer (yellow-green), respectively. The trichromacy or tristimulus theories are intuitively attractive because they correspond loosely to the notion that colors can be specified by weighted sums of red, green, and blue (the primary colors). Three color matching functions l_λ , m_λ , and s_λ are used to match a color of constant luminance, for all values of dominant wavelength in the visible spectrum.

In 1931 the Commission Internationale de l'Éclairage (CIE) defined three standard primaries X , Y , Z , to replace red, green, and blue. The three color-matching functions x_λ , y_λ , and z_λ for the X , Y , Z primaries have better mathematical properties than the color matching functions l_λ , m_λ , and s_λ in the tristimulus theory, since they do not have negative values. In fact the three CIE color matching functions x_λ , y_λ , and z_λ are linear combinations of the l_λ , m_λ , and s_λ color matching functions based on the spectral response of the three cone types in the human retina (Foley et al. [47]).

The amounts of X , Y , Z primaries needed to match a color with spectral energy distribution $P(\lambda)$ are:

$$\tilde{X} = k \int P(\lambda) x_\lambda d\lambda, \quad \tilde{Y} = k \int P(\lambda) y_\lambda d\lambda, \quad \tilde{Z} = k \int P(\lambda) z_\lambda d\lambda, \quad (7.1)$$

where k is measured in lumens/watt. For self-luminous objects k is equal to 680. By applying

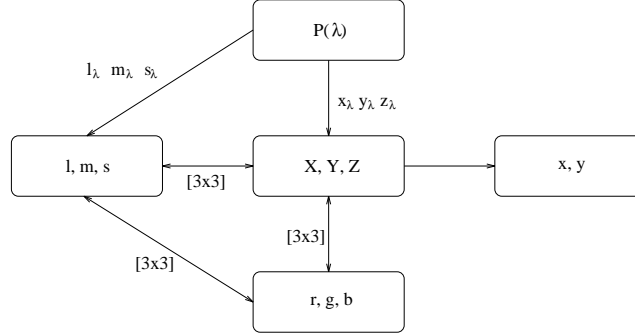


Figure 7.2: Color transformation scheme. With color matching functions l_λ , m_λ , and s_λ , the spectral response functions for the three cone types (l, m, s) can be obtained from the spectral energy distribution $P(\lambda)$. The CIE defined three other matching functions x_λ , y_λ , and z_λ which have positive weights only. To obtain the CIE chromaticity diagram illustrated in Figure 7.1, triple (X, Y, Z) is transformed to tuple (x, y) . The transformation function used is given in (7.2). For transformations from (X, Y, Z) to (r, g, b) or (l, m, s) and vice versa matrix multiplications of different 3×3 matrices are needed.

\tilde{X} , \tilde{Y} , \tilde{Z} to X, Y, Z a color C can be matched: $C = \tilde{X}X + \tilde{Y}Y + \tilde{Z}Z$. The chromaticity values x , y , and z are defined by normalizing against $\tilde{X} + \tilde{Y} + \tilde{Z}$:

$$x = \frac{\tilde{X}}{\tilde{X} + \tilde{Y} + \tilde{Z}}, \quad y = \frac{\tilde{Y}}{\tilde{X} + \tilde{Y} + \tilde{Z}}, \quad z = \frac{\tilde{Z}}{\tilde{X} + \tilde{Y} + \tilde{Z}}. \quad (7.2)$$

Notice that $x + y + z = 1$. That is x , y , and z are on the $\tilde{X} + \tilde{Y} + \tilde{Z} = 1$ plane. An illustration of the (x, y) tuple is given in Figure 7.1.

7.2.1 Matrix transformations

The r, g, b primaries are used in Cathode Ray Tube (CRT) monitors and color raster graphics. A color camera connected to a frame-grabber delivers images which are stored in (r, g, b) -format, or other triples which can be easily transformed to (r, g, b) -format.

The transformation from r, g, b primaries to the X, Y, Z primaries is a matrix multiplication:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}, \quad (7.3)$$

where X_r, X_g, X_b are the weights applied to the monitor's r, g, b colors to find X . The same yields for Y_r, Y_g, Y_b to find Y and for Z_r, Z_g, Z_b to find Z .

The first image in vision is obtained in the retina which consist of a two-dimensional array of cones. Suppose that at all positions (x, y) there exist a combination of three different cone types, then the response of all the cones can be interpreted as the first "neural color image" in early vision. Data generated by a color camera should be first transformed to a triple

which represents the response of the three different cone types at a certain position (x, y) in order to use it as input for early vision. In real life a color image is displayed (on a screen) first in order to be seen by a human observer. Therefore the monitor also plays a role in the transformation from one triple to another. Since every monitor has its own properties, it is not trivial to find a suitable 3×3 transformation matrix.

Hubel and Livingstone [81] used a 3×3 matrix prepared by Smith and Pokorny [173] to convert from (r, g, b) values to relative human cone excitations (l, m, s) :

$$\begin{pmatrix} l \\ m \\ s \end{pmatrix} = \begin{pmatrix} 0.293 & 0.603 & 0.104 \\ 0.134 & 0.704 & 0.162 \\ 0.046 & 0.099 & 0.854 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}. \quad (7.4)$$

Inverting the 3×3 matrix, the following matrix is obtained:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 5.575 & -4.808 & 0.233 \\ -1.019 & 2.339 & -0.320 \\ -0.182 & -0.012 & 1.195 \end{pmatrix} \begin{pmatrix} l \\ m \\ s \end{pmatrix}. \quad (7.5)$$

The responses of l , m , and s are positive and, like r , g , and b , do not exceed 255.

The conversion from r , g , b to l , m , s shifts the peak sensitivities to orange, yellow-green, and blue. Using relative responses l , m , and s responses makes l-m opponency (red-green opponency) almost ineffective since both l and m get a strong green component: 60 and 70 percent, respectively. With almost ineffective we mean that cells which have l-m opponency will respond hardly if they get l and m as input. The cells with lm-s opponency (yellow-blue opponency) respond well since there is a green component and an opposite blue component.

Ineffective l-m opponency and effective lm-s opponency might explain the difference in amount of cells with l-m and lm-s opponency, the ratio is approximately 3:1 (Livingstone and Hubel [117]).

It is more realistic to use the absolute human cone excitations. This makes cells with l-m opponency more effective since there is a difference in peak sensitivity between the cones of about ten percent. The difference in peak sensitivity between l and s cones is approximately a factor ten (Foley et al. [47]). With such a huge difference we can hardly speak of lm-s opponency.

To avoid the problems of ineffective opponency in one of the opponent channels and for information technical reasons we assume that $r = l$, $g = m$, and $b = s$.

7.3 Opponent cells

Like in Chapter 3 where we modeled the center-surround cells we are modeling the opponent cells, by first describing a function, which is convolved with an input image to model the response of the cell. For the opponent cell's receptive field profiles we need two functions, one for the excitatory and one for the inhibitory part.

The first type of cell is a center-surround opponent cell. For both excitatory and inhibitory parts we use a Gaussian to model the receptive field profile. For the excitatory part Gc_σ

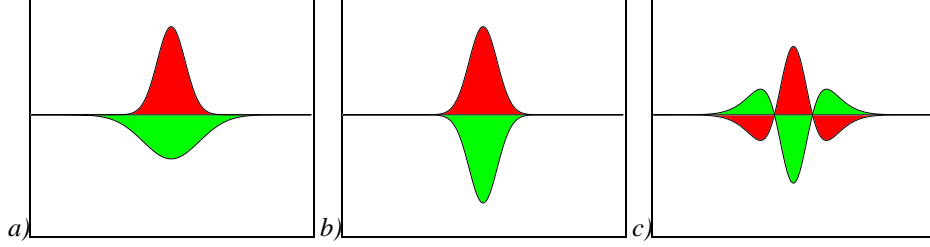


Figure 7.3: Opponent receptive field profiles. The profile above the horizontal line is excitatory and below inhibitory. Receptive field of the a) center-surround opponent cell, b) center-only opponent cell, and c) double-opponent cell. For visualization, the functions are displayed in one dimension only.

(A.15) and for the inhibitory part $G_{S\sigma}$ (A.22). Both functions are illustrated in Figure 7.3a. The response image of the center-surround opponent cell is modeled as follows:

$$CS_{\sigma}^{e,i} = I^e * (2\pi\sigma^2 G_{\sigma}) - I^i * (2\pi\sigma^2 G_{S\sigma}) = I^e * G_{C\sigma} - I^i * G_{S\sigma}, \quad (7.6)$$

where $(e, i) \in \{(r, g), (g, r), (b, y), (y, b)\}$. I^a , $a \in \{r, g, b, y\}$, denotes which color (red, green, blue, or yellow) component of input image I is used, where

$$I^y = \frac{1}{2} (I^r + I^g). \quad (7.7)$$

Since the excitatory and inhibitory functions in Figure 7.3a are not symmetrical in the x-axis, an additional operator is needed when the excitatory and inhibitory functions are swapped. We will call this the surround-center opponent cell. The response image is as follows:

$$SC_{\sigma}^{e,i} = I^e * G_{S\sigma} - I^i * G_{C\sigma}. \quad (7.8)$$

The receptive field profile for the center-only opponent cell is modeled by two Gaussians G_{σ} (A.3) and is illustrated in Figure 7.3b. The response image of the center-only opponent cell is as follows:

$$CO_{\sigma}^{e,i} = I^e * G_{\sigma} - I^i * G_{\sigma}. \quad (7.9)$$

The third type of cells does not show color preferences, the response image of the center-only broad band cell is modeled by the following operator:

$$G_{\sigma} = I^w * G_{\sigma}, \quad (7.10)$$

where image $I^w = \frac{I^r + I^g + I^b}{3}$. An illustration of the output of this operator is shown in Figure 7.4b.

The receptive field profile of double opponent cells is modeled by two Mexican-hat functions based on the differences of two Gaussians Mcs_{σ} (A.17). The functions are illustrated in Figure 7.3c and the image response of these cells is as follows:

$$DO_{\sigma}^{e,i} = I^e * Mcs_{\sigma} - I^i * Mcs_{\sigma}. \quad (7.11)$$

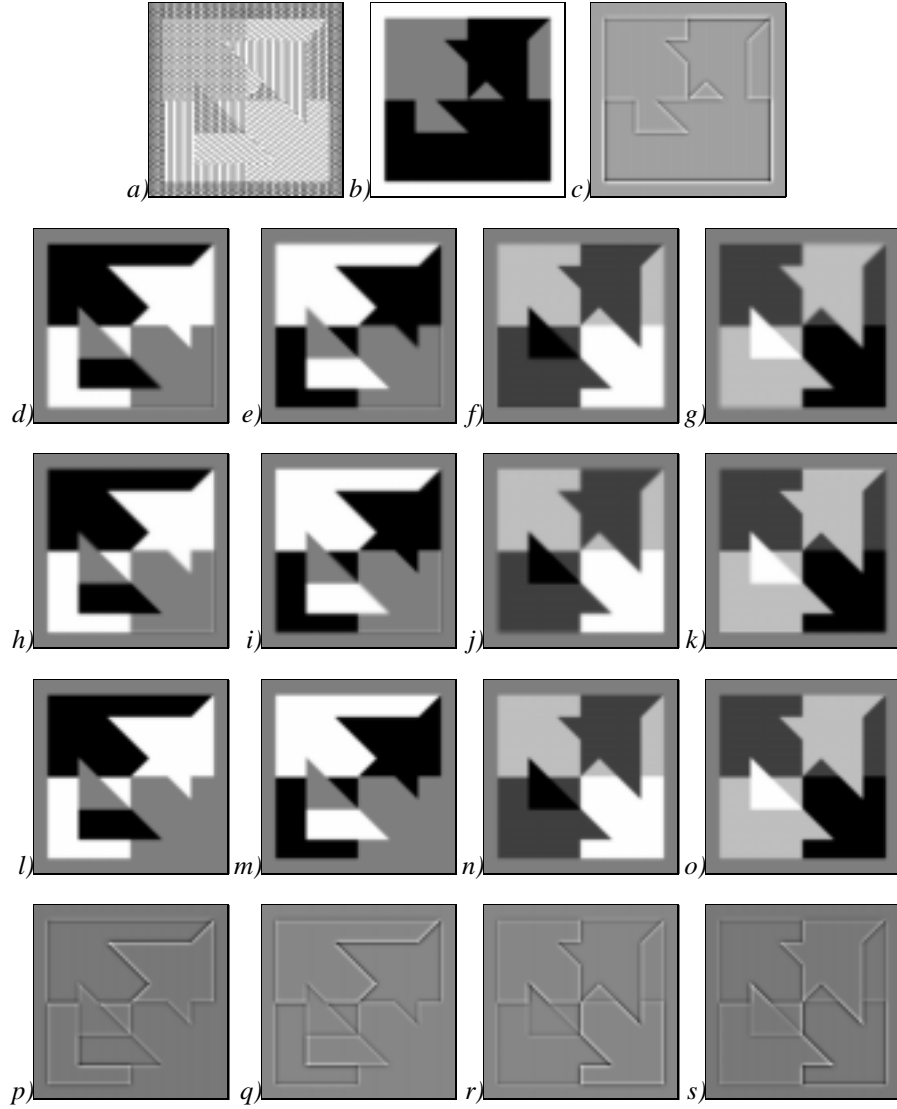


Figure 7.4: a) A simple “color” image. Red, green, and blue components are coded with $|$, $/$, \backslash oriented lines, respectively. A line denotes that the component is 255, no line denotes that the component is zero. This implies that white contains all three oriented lines. b) The output of the G_σ -operator (achromatic channel). c) The output of the M_σ -operator (achromatic channel) defined in (3.6). d)-g) Outputs of the $CS_\sigma^{g,g}$ -, $CS_\sigma^{g,r}$ -, $CS_\sigma^{b,y}$ -, and $CS_\sigma^{b,b}$ -operators, respectively. h)-k) Outputs of the $SC_\sigma^{g,g}$ -, $SC_\sigma^{g,r}$ -, $SC_\sigma^{b,y}$ -, and $SC_\sigma^{b,b}$ -operators, respectively. l)-o) Outputs of the $CO_\sigma^{g,g}$ -, $CO_\sigma^{g,r}$ -, $CO_\sigma^{b,y}$ -, and $CO_\sigma^{b,b}$ -operators, respectively. p)-s) Outputs of the $DO_\sigma^{g,g}$ -, $DO_\sigma^{g,r}$ -, $DO_\sigma^{b,y}$ -, and $DO_\sigma^{b,b}$ -operators, respectively. For all operators we used $\sigma = 1$. Black denotes inhibitory, medium gray no, and white excitatory response.

Instead of using Mcs_σ we can also use a Mexican-hat function given by a Laplacian of a Gaussian M_σ (A.7) which needs to be divided by a center normalization factor (A.13).

Note that the receptive field sizes of the center-surround cells are about four times smaller than the center-only and double-opponent cell types, but this is not shown in Figure 7.3 and is neither clear from the equations above.

The opponent operators are applied to a simple image which is illustrated in Figure 7.4a. The following opponent channels are used: red-green, green-red, blue-yellow, and yellow-blue. The first mentioned color is the excitatory and the second the inhibitory color. The results for the CS_σ -, SC_σ -, CO_σ -, and DO_σ -operators are illustrated in Figure 7.4d-g, h-k, l-o, and p-s, respectively.

Although the responses of the $CS_\sigma^{e,i}$ - and $SC_\sigma^{e,i}$ -operators look the same, they are different. If we subtract the second from the first operator we get an output which is similar to the output of the $DO_\sigma^{e,i}$ -operator. The $CO_\sigma^{e,i}$ -operators also have a similar response as the $CS_\sigma^{e,i}$ - and $SC_\sigma^{e,i}$ -operators, but their difference is at the edges, since they have center opponency only.

The $DO_\sigma^{e,i}$ -operator only responds at edges. The difference in gray in Figure 7.4p-s are optical illusions. The brain fills in the color between the edges. If the edges are bright at the inside, the gray color appears to be brighter than normal. Similarly when the edges are dark inside, the gray color appears to be darker.

7.4 Reconstruction of color using three channels

The cells found by neurophysiologists are of broadband type (achromatic) or have red-green or blue-yellow opponency. In this section we will show that these three channels are sufficient to reconstruct every color. For the reconstruction we use the $CO_\sigma^{e,i}$ -operators, since for these operators holds: $-CO_\sigma^{e,i} = CO_\sigma^{i,e}$.

When a Gaussian is convolved with an image its result is a blurred image. If we choose $\sigma = \epsilon$ to be very small such that the convolved result is identical (in the discrete case) to the original image

$$I^a = I^a * G_{\sigma_\epsilon} \quad (7.12)$$

then we can make an exact reconstruction.

In Figure 7.5 we start with an (r, g, b) color image. In the second and third column the image is split into five different color components (red, green, blue, yellow, and gray). The fourth column gives the output of the $CO_\sigma^{r,g}$ -, $CO_\sigma^{b,y}$ -, and G_σ -operators, which model the responses of the cells in the three different channels. By combining the $CO_\sigma^{r,g}$ - and $CO_\sigma^{b,y}$ -operators we can create two operators with primary color opponency only:

$$\begin{aligned} & CO_\epsilon^{r,b} \\ = & \{(7.9) \text{ and } \{(7.12)\} \\ & I^r - I^b \\ = & \end{aligned}$$

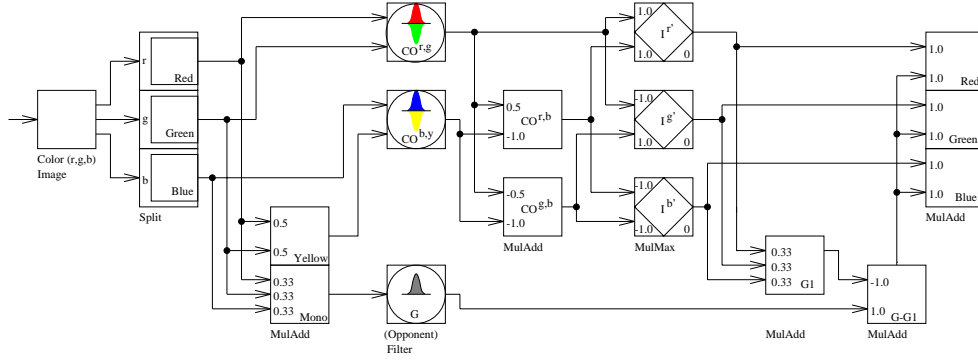


Figure 7.5: Scheme of reconstruction.

$$\begin{aligned}
 & \frac{1}{2}I^r - \frac{1}{2}I^g + \frac{1}{2}I^r + \frac{1}{2}I^g - I^b \\
 = & \{(7.7)\} \\
 & \frac{1}{2}(I^r - I^g) + I^r - I^b \\
 = & \{(7.12) \text{ and } (7.9)\} \\
 & \frac{1}{2}CO_{\epsilon}^{r,g} - CO_{\epsilon}^{b,y}
 \end{aligned} \tag{7.13}$$

and similarly

$$CO_{\epsilon}^{g,b} = -\frac{1}{2}CO_{\epsilon}^{r,g} - CO_{\epsilon}^{b,y}. \tag{7.14}$$

The two additional operators are illustrated in the fifth column of Figure 7.5. To make a reconstruction (sixth column) we take the differences between the different operators:

$$I^{r'} = \max(CO_{\epsilon}^{r,g}, CO_{\epsilon}^{r,b}, 0), \tag{7.15}$$

$$I^{g'} = \max(-CO_{\epsilon}^{r,g}, CO_{\epsilon}^{g,b}, 0), \tag{7.16}$$

$$I^{b'} = \max(-CO_{\epsilon}^{r,b}, -CO_{\epsilon}^{g,b}, 0). \tag{7.17}$$

This gives us back already a reconstruction for pure colors. Suppose that we have a triple $(r, g, b) = (1, 1, \frac{1}{2})$ and a triple $(r', g', b') = (\frac{1}{2}, \frac{1}{2}, 0)$ which appear as light - and dark yellow, respectively. The reconstruction using (7.15), (7.16), and (7.17) gives for both colors (r', g', b') and that would imply that with two different opponent cells we are not able to discriminate between these two different colors. It is clear that in this example we lost the gray component.

When the exact (r, g, b) triple should be reconstructed, we need a broad-band cell with center-only properties. The exact reconstruction of image I can be obtained by adding the

output of $G_e - \frac{I^{r'} + I^{g'} + I^{b'}}{3}$ to (7.15), (7.16), and (7.17), which is illustrated in the three right-most columns of Figure 7.5.

7.5 Double opponent color sensitive and orientation selective cells

Double opponent cells with center surround properties were first found in the goldfish (Daw [25]). Later these cells were found also in monkey area V1 and V2.² Their receptive fields are rather small in comparison with receptive fields of cells in monkey area V4. The cells in area V1 are connected to area V4, probably in a convergent fashion, with the result that the recipient cells in area V4 have much larger receptive fields (Zeki [204]).

The cells in area V4 which Zeki passed first when he did his measurements in area V5 are more difficult to activate, but a great majority of cells which he succeeded to activate showed color selectivity in the sense that they responded better to certain colors than to others (Zeki [202, 203]). There are also orientation selective cells found in area V4. The combination of both color and orientation selections are found too in this area (Zeki [204]). Also Hubel, Wiesel, and Livingstone mention that it seems likely that cells in blobs project to cells around the blobs, a few of which are specific for both color and orientation. These cells we term: *color sensitive, orientation selective cells*.

Although orientation selective cells with double opponent color properties are not found in area V4, it can be made plausible that these cells exist since cells in V4 receive their input from area V1 and V2 which have double opponent properties. We term these hypothetical cells: *double opponent, color sensitive, orientation selective cells*. Like for the color opponent cells in area V1 and V2 we assume that there are 3 classes of cells: the broadband or achromatic type and red-green and blue-yellow opponent types.

Using the building block principle for the P pathway *double opponent, color sensitive, simple cells* can be obtained in a similar way as the simple cells. If we assume that color sensitive simple cells are in the visual cortex, and we use the S -operators (3.17) and (3.18) then the response image of the symmetrical double opponent simple cell can be modeled as follows:

$$DO^{e,i}S_{\sigma,\theta}^s = I^e * \Re \hat{G}_{\sigma,\theta} - I^i * \Re \hat{G}_{\sigma,\theta} \quad (7.18)$$

and the response image of the anti-symmetrical double opponent simple operator is

$$DO^{e,i}S_{\sigma,\theta}^a = I^e * \Im \hat{G}_{\sigma,\theta} - I^i * \Im \hat{G}_{\sigma,\theta}. \quad (7.19)$$

Applying the building block principle to the double opponent color sensitive cells we have the following operator for the double opponent color sensitive complex cell:

$$DO^{e,i}C_{\sigma,\theta}(x,y) = \sqrt{\left(DO^{e,i}S_{\sigma,\theta}^s(x,y)\right)^2 + \left(DO^{e,i}S_{\sigma,\theta}^a(x,y)\right)^2}. \quad (7.20)$$

²The responses of cells in V2 have not been studied in sufficient detail to tell whether they are similar to the responses of the counterparts in area V1 (Zeki [204]).

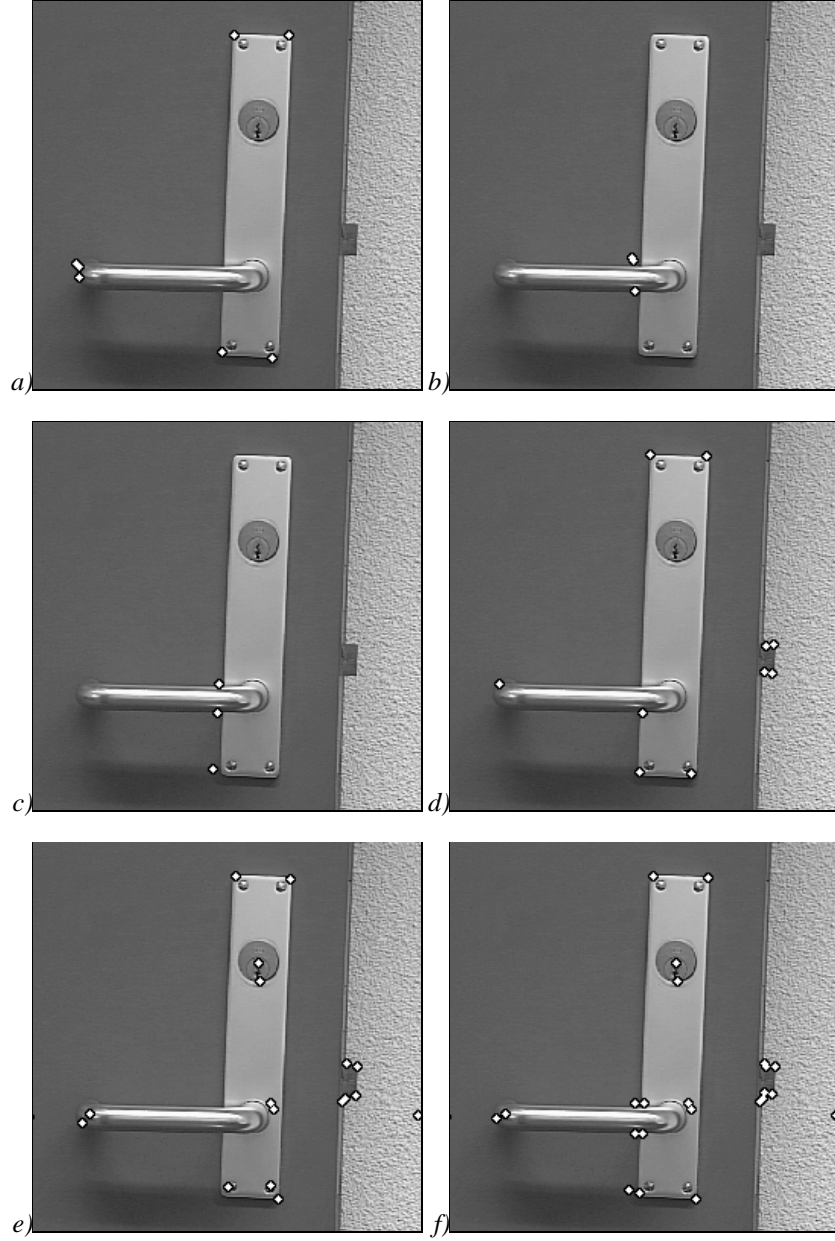


Figure 7.6: Detected corners obtained by using the MC-operator. a)-d) In the MC-operator we use $DO^{e,i}E_0$ instead of E_0 . Detected corners for $MC_{5,avg}$ with $\{e, i\} = \{\text{red, green}\}$, $\{\text{green, red}\}$, $\{\text{blue, yellow}\}$, and $\{\text{yellow, blue}\}$, respectively. e) Results of the $MC_{5,avg}$ -operator (also illustrated in Figure 5.12d). f) Detected corners ($MC_{5,all}$ -operator) by combining all channels with the E_{all} -operator.

This operator is identical to (3.20) with the exception that we use the double opponent wavelength sensitive simple cell operators instead of the simple cell operators. Similarly the following double opponent wavelength operators are used: $DO^{e,i}C_{\sigma}$, $DO^{e,i}E_{\sigma}$, $DO^{e,i}E_{\text{avg}}$, which are similar to the operators from (3.22), (4.14), and (5.4), respectively.

We introduce a new combining operator

$$\begin{aligned} E_{\text{all}}(x, y) = & E_{\text{avg}}(x, y) + \\ & DO^{r,g}E_{\text{avg}}(x, y) + DO^{g,r}E_{\text{avg}}(x, y) + \\ & DO^{b,y}E_{\text{avg}}(x, y) + DO^{y,b}E_{\text{avg}}(x, y) \end{aligned} \quad (7.21)$$

to combine all used channels: broadband, red-green, green-red, blue-yellow, and yellow-blue. Corners in color images can now be found by using the PCF-operator (5.1) and MC-operator (5.2) for $\omega = \text{all}$.

If a single broadband channel is used only one loses information, which can cause corners not to be detected. With the addition of two chromatic channels we are sure that no information is lost, since we can make a reconstruction. The advantage of using multiple channels is that all corners should be detected. If corners are not detected it is due to short-comes in the model and not due to loss of information. Most corners appear in more than one channel, summing the outputs of different channels (7.21) gives a good indication about the robustness of the corner. Another advantage is that we can assign another feature attribute to the corner. This feature attribute indicates in which channels the corner is found.

In Figure 7.6 the detected corners are illustrated by using the MC-operator, for better visualization we illustrated the corners larger. Figure 7.6e shows that the results obtained separately are not sufficient but that the combination of the channels is necessary to find all corners (Figure 7.6f). Combining corners at multiple channels can cause a local maximum in a channel to disappear in the combination of channels and vice versa a local maximum can appear due to summing the responses of different channels which are not in one of the single channels.

7.5.1 Graph extraction for multiple channels

In Figure 7.7 it becomes clear that most of the segments are detected in every channel. The combination of all channels improves the results but it is not so remarkable as the improvement of the detected corners.

In Figure 7.7f the detected segments which separate the door from the background are not displayed because we will consider segments between two different corners only. In the other images of the same figure these segments are displayed with a white-black-white segment.

The corners obtained with the $MC_{7,\text{all}}$ -operator are used as input and applied to the line detection algorithm from Chapter 6 to all channels separately. For the MC-operator we used a local neighborhood with $n = 4$ to avoid local maxima in a PCF area which are very close to each other and represent the same corner. Two examples of such corners are in Figure 7.6.

The difference between Figure 6.7b and Figure 7.8b is large. In the achromatic channel some corners are not found and therefore also the corresponding lines. Using multiple channels has the advantage that almost all corners are detected. However if we compare the

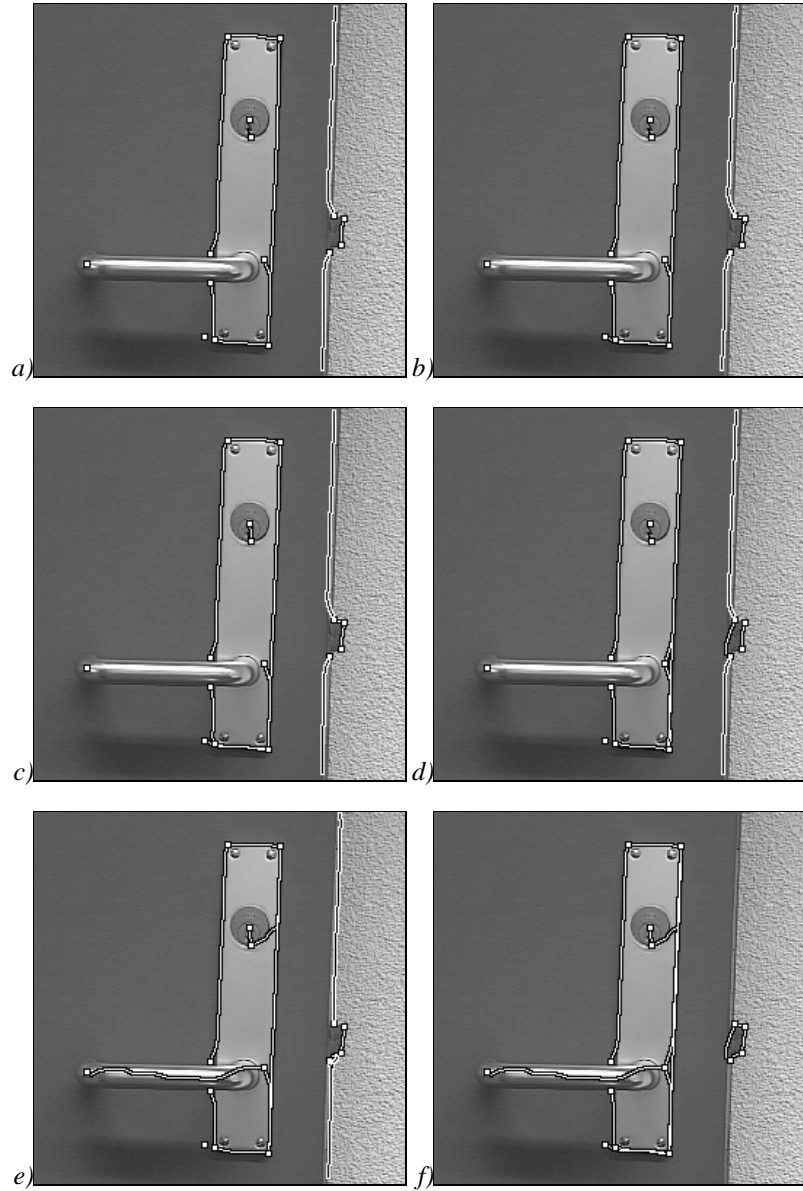


Figure 7.7: Detected segments of a) red-green, b) green-red, c) blue-yellow, d) yellow-blue, and e) achromatic channel. f) Detected segments combining all five channels.

detected graphs at an achromatic channel with the graphs found at multiple channels there is not always an obvious improvement. For example if we compare the image graph illustrated

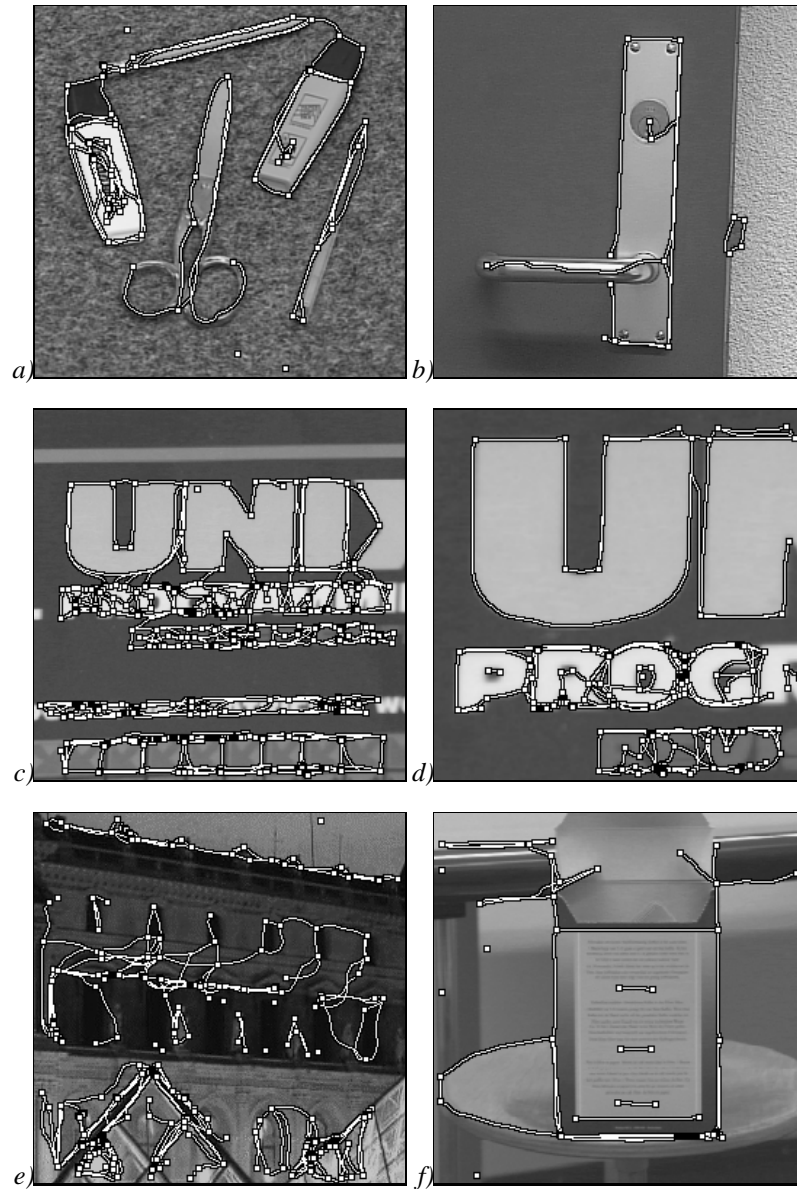


Figure 7.8: *Graphs found for different color images using multiple channels.*

in Figure 6.7c with the graph of Figure 7.8c we notice that there are more corners detected at multiple channels and also that the number of detected segments increased considerably but it is not clear which of them is better.

7.6 Summary and discussion

We introduced a multi-channel multi-scale edge and corner detection model. The model uses as input a color image in (r, g, b) -format. Note that with a matrix multiplication this triple can be transformed to another triple.

In the chapter we first modeled the response of known opponent color cells. The center-surround, center-only, and double opponent cells. With a simple image we illustrated that the first two types respond well to specific colors and the latter type responded well to edges only. In the model our primary goal is to find corners and edges. This implies that we should use double-opponent cells. However the center-surround and center-only cells are very useful too, they can be used to determine color. Hence they can serve as color-attributes to facilitate object recognition. The double opponent cells are the chromatic equivalent of the center-surround cells in the achromatic channel, i.e. the response of both cell types is similar.

We used two chromatic channels (red-green and blue-yellow opponency) and one achromatic channel. These three channels are found in monkeys and it is proven that these channels are sufficient. Additional opponent channels are therefore not necessary. Nevertheless, if an industrial application uses only a few colors, it can be useful to define other opponent colors.

In monkey area V4, Zeki [204] found color sensitive and orientation selective cells. In 1978, the concept of a modular organization of the primate area 17 (layers two and 3) took a new dimension with the discovery of Wong-Riley of a pattern of regularly repeating blob-like structures. The cells within these blobs are color selective. Livingstone and Hubel [89, 117] mentioned that it seems likely that cells in blobs project to cells around the blob cells. This implies that some cells are both color and orientation selective.

A pair of such cells with opposite color pair can be combined to the equivalent of a simple cell for the color pair in question. These cells have not been found yet, but it is quite plausible that they exist. We call them double opponent simple cells. Their output is combined in a similar way as the complex cells for the achromatic channel, this results in the double opponent complex cells. Their existence has not been shown experimentally but the repetition of the construction of the achromatic pathway seems plausible. By using the repetition we also introduced the double opponent end-stopped cells which respond well to corners and line-ends with red-green or blue-yellow opponent colors.

Combining the two chromatic channels with the achromatic channel improved the results at corners considerably compared to the results with the achromatic channel only, an illustration of the difference is given in Figure 7.6e and f.

The improvement is less obvious for edge detection (complex cells) but it is important that most and preferably all edges in the image are detected.

In Chapter 8 a graph matching algorithm is used to find objects by using model graphs. An object is found if there is a copy of the model in the image graph. Since we are searching for copies, falsely detected lines are not a problem. Of course there is a possibility, although not likely, that objects are falsely detected. Hence center-surround and center-only cells can be useful to make sure that the detected object is correctly detected.

8

Object Recognition using Graph Matching

The carpenter measures with a line and makes an outline with a marker; he roughs it out with chisels and marks it with compasses. He shapes it in the form of man, of man in all his glory, that it may dwell in a shrine.

Isaiah 44:13 (NIV)

RECOGNITION OF ALL A-PRIORI KNOWN OBJECTS in a two-dimensional color image is the goal of this chapter. In pattern recognition and image analysis, graphs are used for the representation of objects. By representing the object as a graph we obtain translation, rotation, and scale invariance. Prototypes or *model graphs* of known objects are used to find objects in an *image graph* which is obtained by preprocessing the image as described in previous chapters. Thus the problem of object recognition is transformed into a graph matching problem, which is applied successfully by, e.g., Eshera and Fu [41], Bunke and Messmer [17, 135], Gold and Rangarajan [55], and Rosin [165].

Generally, the term *graph matching* refers to the process of trying to find correspondences between graphs. Our goal is slightly different: we want to find *all* similar copies of some collection of model graphs in the image graph. Hence, we consider a specific instance of graph matching, namely error-tolerant subgraph isomorphism.

Two graphs are *isomorphic* if there is a one-to-one correspondence function between the vertices of the two graphs such that the structure of the edges is preserved by the function (McHugh [131]). If one of the graphs involved in the matching process is larger than the other, i.e. the image graph $G(V, E)$ contains more vertices than the model graph $G_m(V_m, E_m)$, then we are looking for a *subgraph isomorphism* from G_m to G . In fact, we are looking for all inexact subgraphs in G that are isomorphic with G_m . In many applications, the encoding of objects as graphs will not be perfect due to, e.g, noise or limitations of the graph extraction algorithm. Hence, it is not realistic to require isomorphic matching. A natural idea is to introduce cost functions and incorporate the concept of errors into graph matching. Graphs are then compared to each other by means of the *error-tolerant subgraph isomorphism*.

One of the drawbacks of graph matching is its computational complexity. It is well known that subgraph isomorphism testing is NP-complete (van Leeuwen [191]). The subgraph isomorphism problems are NP-complete even if G and G_m range over planar graphs of valence 3 (Lingas and Proskurowski [114]). Subgraph isomorphism can be solved in polynomial time if G is a forest and G_m is a tree (Reyner [159]) but remains NP-complete if G is a tree and G_m is a forest (Garey and Johnson [54]). Consequently, no algorithm can be constructed that guarantees to find (error-correcting or error-tolerant) subgraph isomorphisms in polynomial time.

Because of the representational power of graphs, much effort has gone into the development of efficient matching algorithms. Dependent on the application different attributes are assigned to the vertices and edges in a graph. Heuristics that use these attributes can decrease the search space to a low order polynomial (Gold and Rangarajan [55]). The algorithms used in literature can globally be divided into two groups.

The *first group* involves the construction of a state-space which is then searched. Searching by using a state-space always guarantees the *optimal* solution. One of the best known methods for graph and subgraph isomorphism detection is based on depth-first backtracking search (Corneil and Gotlieb [21]). The number of backtracking steps in a search tree can be greatly reduced by a combination of backtracking and forward checking (Ullman [188]). Another approach for subgraph isomorphism detection is based on building a so called association graph and searching for maximal cliques in such a graph (Falkenhainer et al. [43], Horaud and Skorda [76], and Myaeng and Lopez-Lopez [138]).

In the problem of *error-correcting* subgraph isomorphism an algorithm is optimal if it is guaranteed to find the sequence of edit operations with *minimal* cost such that a subgraph isomorphism exist (Messmer [134]). Most of these algorithms are based on the A^* -algorithm (Nilsson [142]). By introducing a heuristic cost estimation function, the size of the search tree can be greatly reduced. Various cost functions have been proposed (e.g., Tsai and Fu [187], Shapiro and Haralick [170], Fu [50], Sanfeliu and Fu [167], Eshera and Fu [40], and Wong [196]).

All the algorithms in the first group are guaranteed to find the graph and subgraph isomorphisms. The *second group* of algorithms is based on *probabilistic* or non-linear approximation methods. These algorithms generally have a lower computational complexity but do not guarantee to find the best solution. The algorithms are based on relaxation (e.g., Rosenfeld et al. [164], Peleg [144], Ton and Jain [184], Christmas et al. [20], and Gold and Rangarajan [55]), neural networks (e.g., Kuner and Ueberreiter [103], Yu and Tsai [200], and Suganthan et al. [180]), linear programming (Almohamad and Duffuaa [2]), genetic algorithms (Krcmar and Dhawan [100]), or Lagrangian optimization (Rangarajan and Mjolsness [155]).

Since we want to recognize all objects in the database that differ at most an a-priori known error-value from the exact object in the image, we have the problem of *error-tolerant* subgraph isomorphism. In this problem an algorithm is defined to be optimal if it is guaranteed to find all subgraph isomorphisms, where the sequence of edit operations is less than a limited cost (Messmer [134]).

The algorithm we use belongs to the first group since we want all possible objects in the image graph to be found. To obtain an efficient algorithm we add two attributes (angle and

length) to the graphs. Cost functions based on these attributes and on tolerating missing edges should cut down the number of evaluations severely, and only accept (inexact) matches where the object differs a little from the known object. Inexact matches of missing vertices are not allowed. The system is therefore error tolerant with respect to edges. In an image small differences in length and angle of the same object can be caused by perspective projections. We tolerate matches with small differences in the angles and lengths of an object, i.e., the algorithm is tolerant to small distortions and changes in view.

The chapter is organized as follows: Section 8.1 describes the use of attributes and gives the cost functions which are used to find inexact matches. Section 8.2 gives the graph matching algorithm. The results and time-complexity obtained with this algorithm are discussed in Section 8.3. Section 8.4 gives some related work and discussion. The chapter ends with a summary.

8.1 Graph attributes

The graphs we use are extracted from images. We assign length and angle attributes to such a graph. These attributes give a more accurate description of the objects in the image. Hence, the search space for matching an attributed model graph with an attributed image graph is highly reduced. In this section we give the representation of such an attributed graph and introduce the functions and constants that bound the search space.

We take the graph which is illustrated in Figure 8.1a as an example. Initially we start with corners which have (x, y) -coordinates. To keep a graph translation invariant we must not consider the absolute coordinates but use them relatively to each other. This is obtained by taking the lengths of the edges and the angles between edges. To let a graph be scale independent we choose the lengths relative to each other. This is illustrated in Figure 8.1b, where the longest edge is 100.00 and the others are a percentage of the longest edge. Angles are measured counterclockwise: as a consequence, we have, e.g., in Figure 8.1b that the angle at vertex 0 is different from the angle at vertex 1, although the graph is symmetrical in the vertical axis. The choice of these attributes keeps the graph translation, rotation, and scale invariant. The attributes of the graph are represented in a table. An example of such a table is given in Figure 8.1c.

An efficient representation of a graph is the adjacency matrix. This matrix used by, e.g., van Leeuwen [191] is a boolean matrix B . The elements $B(i, j)$, $i < j$ and $i, j \in V$, of this matrix are chosen to be true (or 1) if and only if $(i, j) \in E$, and false (or 0) otherwise. We choose a different adjacency matrix A in the sense that the elements $A(i, j) \in \mathbb{R}$ give the cost for tolerating a missing edge in the image graph. Due to noise in the color input image and due to limitations of the graph extraction algorithm not every edge will be extracted. For the image graph one would desire that, if an edge is missing during the process of matching, one can tolerate such a missing edge. Since we do not want to tolerate *all* missing edges, a cost function is introduced. The *cost function* $C(v_1, v_2)$, $v_1, v_2 \in V$, for tolerating a missing edge yields a real. During the process of matching we sum the costs for these tolerated missing edges. Since tolerating a certain missing edge between two vertices can be more likely than tolerating another missing edge, there should be a difference in cost. Hence, we do not use

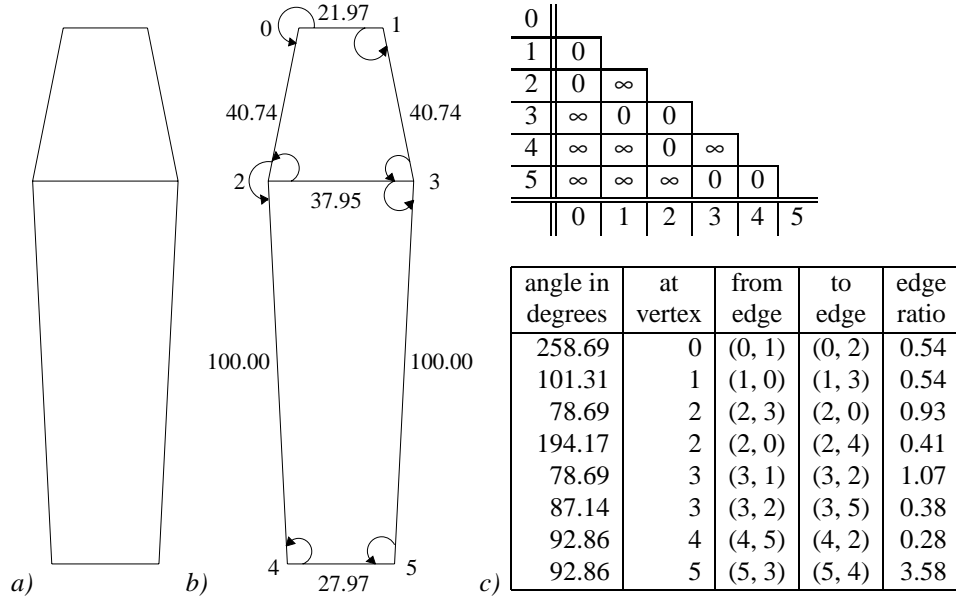


Figure 8.1: a) An example of a (model) graph. b) The same graph augmented with attributes, which are the (relative) length of an edge and the angle between two edges. c) Two tables are used to represent the attributes of the graph. In the table at the top the adjacency matrix is displayed, zero denotes that there is an edge and ∞ denotes that there is no edge between the two corresponding vertices. The table at the bottom gives the angular attribute plus the ratio between the lengths of two edges, which is obtained by dividing the length of the “from edge” by the length of the “to edge”.

a boolean adjacency matrix. For the model graph no cost function is necessary, so only two values are needed: the value 0 denotes $(i, j) \in E_m$ and otherwise it is ∞ . Since we use undirected graphs we only need the lower- or upper-diagonal of the adjacency matrix which means $\frac{1}{2}\#V(\#V - 1)$ elements, where $\#V$ denotes the number of vertices in V throughout this chapter.

To cut down evaluation costs during matching we try to evaluate as little as possible without losing proper solutions and allowing inexact matches. During the matching process we tolerate one or more missing edges in the image graph. Only inexact matches will be considered where the cost function for missing edges is lower than an a-priori known limited cost.

In the lower table of Figure 8.1c, the vertices are augmented with a set of angle and ratio attributes. In the matching process these attributes are used to reduce the search space. When a vertex is matched we tolerate an angle and a ratio at the vertex in the image graph to differ at most a known constant angle and a known constant ratio from the corresponding angle and ratio in the model graph. Additionally we allow the average angle difference of all angles to differ at most a known constant angle. Similarly the average ratio difference is also bounded by a constant. We introduce four bounding constants:

explanation	name	type
number of currently matched vertices	mv	\mathbb{N}
vertex which is currently evaluated	cv	\mathbb{N}
angle of cv which needs to be evaluated	cva	\mathbb{N}
list of matched vertices of the image graph	L	$L_i \in \mathbb{N} \quad i \in [0, \dots, mv)$
list of parsed vertices of the model graph	Lm	$Lm_i \in \mathbb{N} \quad i \in [0, \dots, mv)$
summed edge cost	De	\mathbb{R}
summed angle difference	Da	\mathbb{R}

Table 8.1: The data structure of the stack of the algorithm.

1. the angle tolerance,
2. the average angle tolerance,
3. the ratio tolerance, and
4. the average ratio tolerance.

8.2 Matching algorithm

The basic matching algorithm is a tree search algorithm. For efficiency reasons we do not create the complete tree but we use a stack. The data structure used in the stack is illustrated in Table 8.1.

In the data structure we keep a list of matched vertices from the model graph and a list of matched vertices from the image graph. The summed edge cost parameter De is the cost for missed but tolerated edges. The summed angle difference Da is used for the average difference. Both should not exceed a chosen maximum. We use parameter cv for the current vertex, whose attributes have to be evaluated. Since a single vertex can have more than one angle and ratio attribute we need an additional parameter cva to indicate which angle and ratio have to be evaluated.

A pseudo code for the graph matching algorithm is illustrated in Figure 8.2. The algorithm finds all inexact matches, which satisfy known criteria, of model graph Gm in image graph G . Lines 2-11 are the initial stage of the algorithm. We start with an empty stack and after that push all vertices of the image graph one after another on the stack since all vertices of the image graph can be matched with the first vertex of the model graph. Line 12-31 are the actual matching. In line 13 we take a possible partial solution from the top of the stack and check if we have a complete match (line 14). If it is a complete match, the maximum and average relative length differences between model and match are calculated (line 15). If both are less than some known constants, the match is accepted and displayed (line 16). If we do not have a complete match we go to line 18. Here we check if the current angle and ratio can be evaluated. With evaluation we mean that model angle and ratio can be compared with image angle and ratio. If we can not evaluate, because one or both vertices to form the angle

```

1  Procedure MatchGraph ( $G, Gm$ )
2      stack :=  $\emptyset$ 
3      forall  $v \in V$ 
4           $L_0 := v$ 
5           $Lm_0 := \text{FirstVertex}(Vm)$ 
6           $mv := 1$ 
7           $cv := Lm_0$ 
8           $cva := \text{FirstAngle}(cv)$ 
9           $De := 0.0$ 
10          $Da := 0.0$ 
11         Push ( $mv, cv, cva, L, Lm, De, Da$ )
12     while stack  $\neq \emptyset$ 
13         Pop ( $mv, cv, cva, L, Lm, De, Da$ )
14         if  $mv = \#Vm \wedge \text{Last}(cv, Lm)$  /* All vertices matched */
15             if ProperMaximumAndAverageRelativeLengthDifferences
16                 Accept
17             else
18                 if CanEvaluateAngleFormVertex ( $cva, cv, \&r, \&\alpha, \&mv$ )
19                     if ProperRatioAndAngle ( $r, \alpha, \max Da, Da, \&a$ )
20                         if  $cva = \text{LastAngle}(cv)$ 
21                              $cv := \text{NextVertex}(cv, Vm)$ 
22                              $cva := \text{FirstAngle}(cv)$ 
23                         else
24                              $cva := \text{NextAngle}(cva, cv)$ 
25                             Push ( $mv, cv, cva, L, Lm, De, Da + a$ )
26                     else /* Add missing vertex  $nv$  */
27                          $Lm_{mv} := nv$ 
28                         forall  $v \in V \setminus L$  /* Unused vertices only */
29                              $L_{mv} := v$ 
30                             if ProperVertex ( $Ce, \max De, De, \&e$ )
31                                 Push ( $mv + 1, cv, cva, L, Lm, De + e, Da$ )

```

Figure 8.2: Algorithm for graph matching.

are missing, then we match the missing model vertex by adding it to the list (line 27) and find all possible vertices in the image graph (lines 28-31). A vertex v is added if it is not matched yet and if the cost Ce of tolerating missing edge (L_{cv}, v) is smaller than the allowed cost.

The speed of the algorithm depends mainly on the condition of line 18. If the maximum tolerated angle and ratio differences are chosen properly most of the partial matches will be rejected here, by not pushing it back on top of the stack.

The speed of matching depends on the total amount of pushes. In worst case there are

$$\#V \times (\#V - 1) \times (\#V - 2) \times \cdots \times (\#V - \#V_m) = \frac{\#V!}{(\#V - \#V_m)!} \quad (8.1)$$

possible one-to-one correspondences between the vertices of a pair of graphs. Since every partial match is pushed on the stack the total amount of pushes is:

$$\sum_{i=1}^{\#V_m} \frac{\#V!}{(\#V-i)!}. \quad (8.2)$$

In this situation the memory requirements for the stack are

$$\#V + (\#V - 1) + (\#V - 2) + \dots + (\#V - \#V_m) = (\#V + 1)\#V_m - \frac{1}{2}\#V_m(\#V_m + 1). \quad (8.3)$$

This gives a memory complexity of $O(\#V(\#V_m)^2)$ since the data structure is of $O(\#V_m)$.

8.3 Results

We bound the search space of the matching algorithm by five functions as mentioned in Section 8.1. The edge and ratio functions need to be explained in some more detail. We assume that the cost of missing edges all have the same cost. Which means that for the image graph an adjacency matrix is created like the one for the model graph but all ∞ s are replaced by 1s.

The difference in ratio δr between two pairs of edges is as follows:

$$\delta r = \max \left(\frac{r_m}{r}, \frac{r}{r_m} \right), \quad (8.4)$$

where r_m denotes the ratio of an edge pair in the model graph and r denotes the corresponding matched ratio in the image graph.

Two images are illustrated in Figure 8.3a-b and their extracted image graphs in Figure 8.3c-d. We will use the model graph illustrated in Figure 8.1 to find all the markers in the two different images.

The results are illustrated in Figure 8.4a-b. The parameters for the fault tolerances are chosen too large (on purpose) such that false matches are found in Figure 8.4b. We tolerated at most five missing edges from the in total seven edges of the model graph and a maximum δr of five. We tolerated an angle difference of at most 10 percent (which corresponds with 36 degrees) and also an average angle difference of at most 10 percent, since there are 10 angles in the model graph this means that the summed angular difference should not exceed 360 degrees. When a match is found we tolerate a maximum relative length difference of 50 percent and an average relative length difference of 10 percent. The matching times for both image graphs are approximately 13 and 17 seconds. If a more reasonable $\delta r = 2$ is chosen correct matches are found only. If the maximum number of edges to be tolerated is reduced to two, finding all markers in the image graph takes less than a second.

Tables with matches and differences of length and angle compared to the model graph are given in Figure 8.4c-d. The correct matches show an average angle difference which is at most 2.82 where we allow at most 10 percent and the average length difference is at most

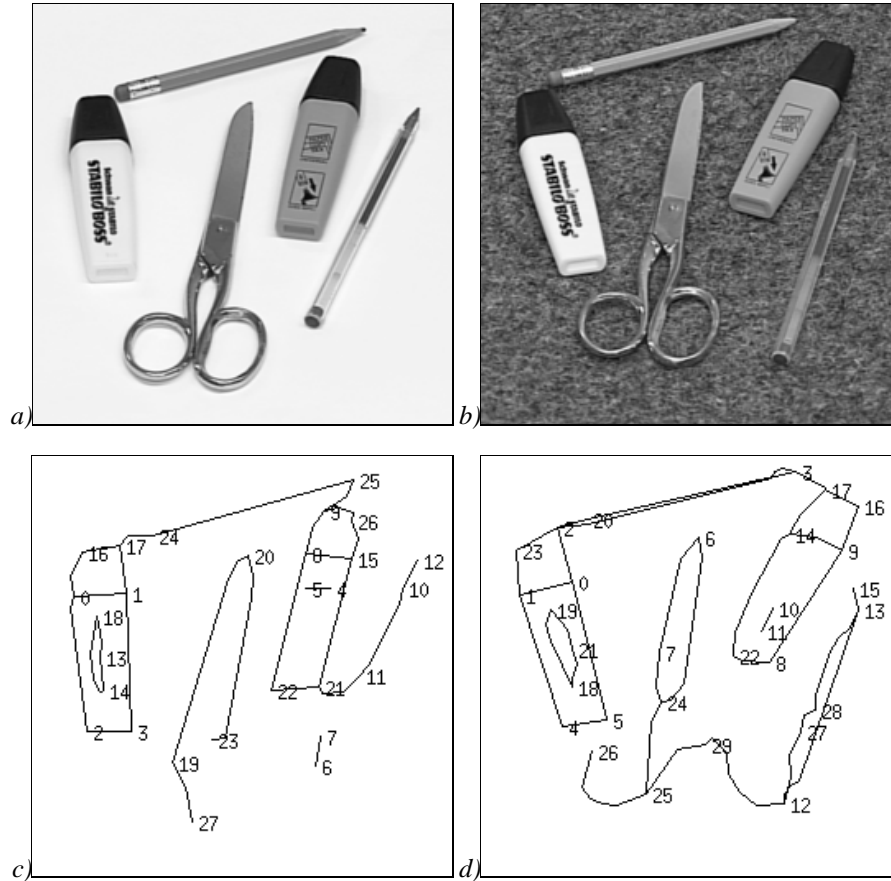


Figure 8.3: a)-b) Input images. c)-d) Extracted image graphs from a) and b), respectively, with numbered vertices.

5.65 percent, also here we allow at most 10 percent. To speed up to matching process we can tolerate a smaller average angle and length.

Since speed is an important factor in graph matching, we applied the algorithm using different criteria. The speed is measured in number of pushes¹ since this number is machine and time independent. Let us take a closer look to the number of pushes we need for inexact matching and assume that the maximum number of missing edges that is tolerated is equal to the number of edges in the model graph. By matching the first vertex of the model graph we need $n = \#V$ pushes. For the second and third vertex we still cannot evaluate angle and ratio since three vertices are needed and therefore we need $n(n-1)$ and $n(n-1)(n-2)$ pushes, respectively. When the third vertex is added to the partial match, we start evaluating angle

¹Approximately 25,000 pushes can be evaluated in one second on a HP 9000-735 workstation.

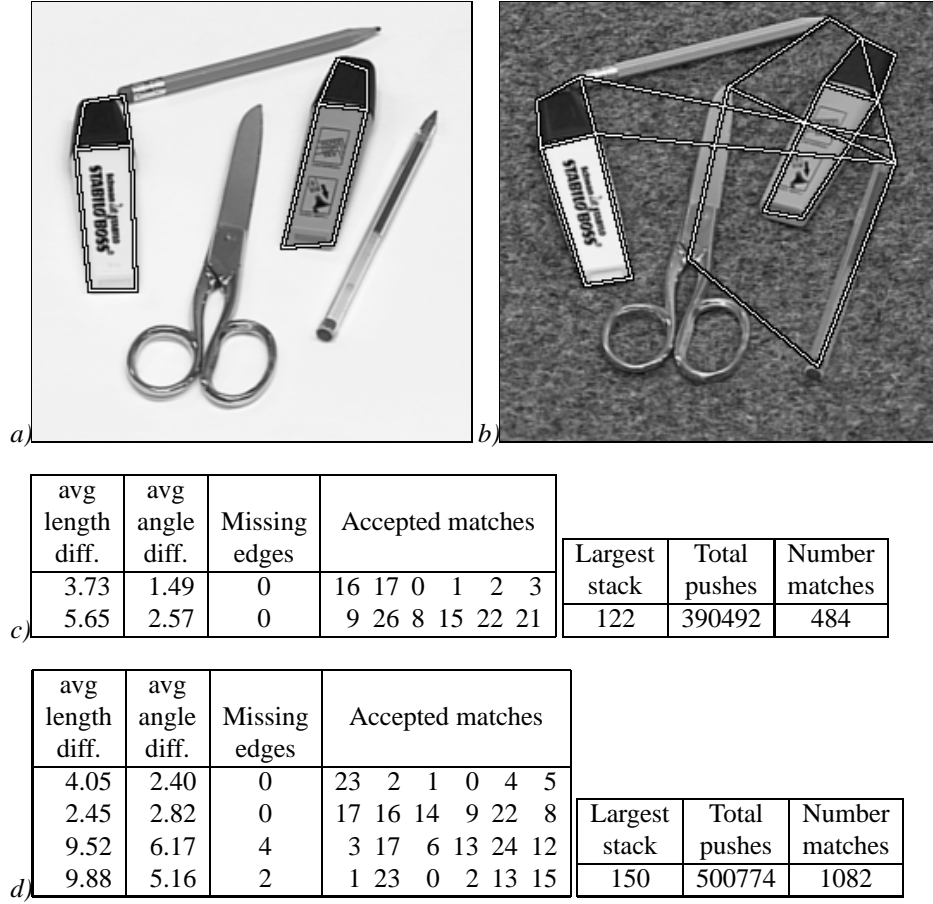


Figure 8.4: a)-b) Found matches of the markers. The used parameters are: the maximum number of missing edges that is tolerated is five, the ratio between two edge pairs $\delta r = 5$, the average angle tolerance is 10 percent, the maximum angle tolerance is 10 percent, the maximum length tolerance is 50 percent, and the average length tolerance is 10 percent. c)-d) The found matches with their average length and angle difference in percentage and number of missing edges. In the three rightmost columns the memory requirements and the total evaluation time in pushes are given.

and ratio. At this stage a large number of partial matches is rejected and for the forth vertex we do not need $(n-3)$ pushes for every partial solution but say c_4 , where $0 \leq c_4 \leq n-3$. This implies that at this stage $n(n-1)(n-2)c_4$ pushes are needed. Similarly for the k^{th} vertex ($k \geq 4$) we need $n(n-1)(n-2) \prod_{i=4}^k c_i$ pushes, with $0 \leq c_i \leq n+1-i$. The total amount of pushes is

$$\sum_{i=1}^3 \frac{n!}{(n-i)!} + \frac{n!}{(n-3)!} \sum_{j=4}^m \prod_{i=4}^j c_i, \quad (8.5)$$

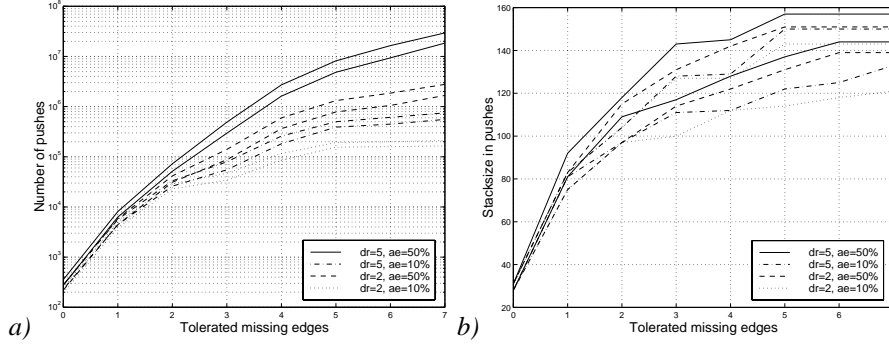


Figure 8.5: Timing results of matching the model graph (Figure 8.1)a with the two image graphs of Figure 8.3c-d. a) Time in number of pushes. b) Maximum stack size, measured in number of pushes which are pushed on the stack and not popped from the stack.

where $m = \#V_m$. If $c_i = n + 1 - i$ (for all i) then the number of pushes is equal to the maximum number of pushes (8.2).

It is difficult to give a good estimation of c_i since it depends on the choice of the angle and ratio tolerance. If the angle is chosen to be so large that it is always accepted and the ratio tolerance is large then c_i will be approximately $n + 1 - i$. The model graph of Figure 8.1a which contains six vertices is matched with the two different image graphs of Figure 8.3c-d which contain 28 and 30 vertices, respectively. In worst case we need $2.84 \cdot 10^8$ and $4.45 \cdot 10^8$ pushes for these image graphs. In Figure 8.5a we used $\delta r = 5$ and a maximum angle error $ae = 50\%$ and needed about a factor twenty less than the maximum possible amount of pushes.

In a normal situation we choose parameters $\delta r = 2$ and $ae = 10\%$ which means that we allow the ratio to differ at most a factor two ($0.5l - 2.0l$) and the angle to differ at most 36 degrees ($-\pi/5 - \pi/5$). For these parameters we found $c_4 \approx 1.7$ (7% of $n - 3$), $c_5 \approx 2.5$ (10% of $n - 4$), and $c_6 \approx 0.15$ (0.6% of $n - 5$) for the two image graphs and the marker as model graph. It is expected that c_i becomes smaller with increasing i since the partially matched possible solutions need to satisfy more and more angles and ratios with increasing number of matched vertices and therefore becomes more specific. Partial matches are already so specific at the sixth vertex that about a factor seven less pushes are needed compared to the previous vertex. It is expected that model graphs with more vertices are that specific that few additional pushes (less than the pushes at the sixth vertex) are needed for the rest.

The memory requirements are small compared to the time complexity, see (8.3) and Figure 8.5b.

8.3.1 Complexity analysis of a case study

In this section we present a case study to get insights in the actual matching time needed for graphs up to 250 vertices. We try to recognize traffic signs in a normal environment.

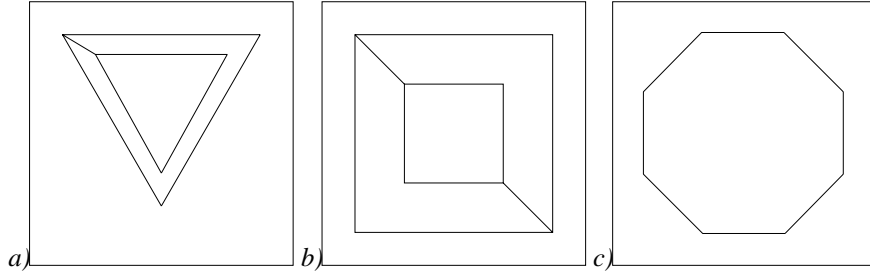


Figure 8.6: The three used model graphs. *a) Danger sign, since inner and outer triangle are related to each other an extra edge is needed. b) Precedence sign, here we used two extra edges. c) Stop sign.*

We choose traffic signs because they are flat, hence they do not change dramatically when viewed under different angles.

In total we used 39 different image graphs and three model graphs (Figure 8.6). From those image graphs 26 contain one or more traffic signs, the other 13 contain no traffic signs. We also used the three model graphs as image graphs. The three model graphs are: the sign for approaching a precedence road, the sign for being on a precedence road, and the stop sign. They are abbreviated to danger, precedence, and stop, respectively. Extra edges in the model graphs of Figure 8.6a-b are needed to retain the relative position and size of the different parts of the graph.

We expect that model danger will be matched faster than the other two because six instead of eight vertices need to be evaluated. From the latter two models the precedence model is expected to be matched faster since it has more angles and edges than the stop model.

Cost function and results

In the previous sections we used a simple cost function for the image graph. For the traffic signs we have extended the cost function for tolerating missing edges in an image graph. The cost function for an existing edge is still zero but the cost for a non-existing edge between a and b is one if there exist a chain between a and b , i.e. there are one or more vertices v_i on or near the imaginary line-segment (a, b) for which there exists edges between (a, v_0) , (v_0, v_1) , \dots , (v_n, b) . If there is no edge and no chain between vertices a and b the cost for tolerating the edge is two plus the number of intersections (a, b) makes with existing edges. The choice for this cost function decreases the matching time.

In total we used 42 image graphs. Three of them are the model graphs which are transformed to image graphs. All three are (of course) detected correctly. In the 13 graphs which do not contain traffic signs nothing is detected. In the other images all signs except two were detected. Both undetected signs are due to a K-junction, which are not detected as illustrated in Figure 5.2. This implies that a corner is missing and the match is not found. Although two signs are not found, there is no falsely detected sign in any of the image graphs. The missed traffic signs are therefore not due to the matching algorithm but due to missing corners. This problem can be solved by improving corner detection or to tolerate missing vertices in the

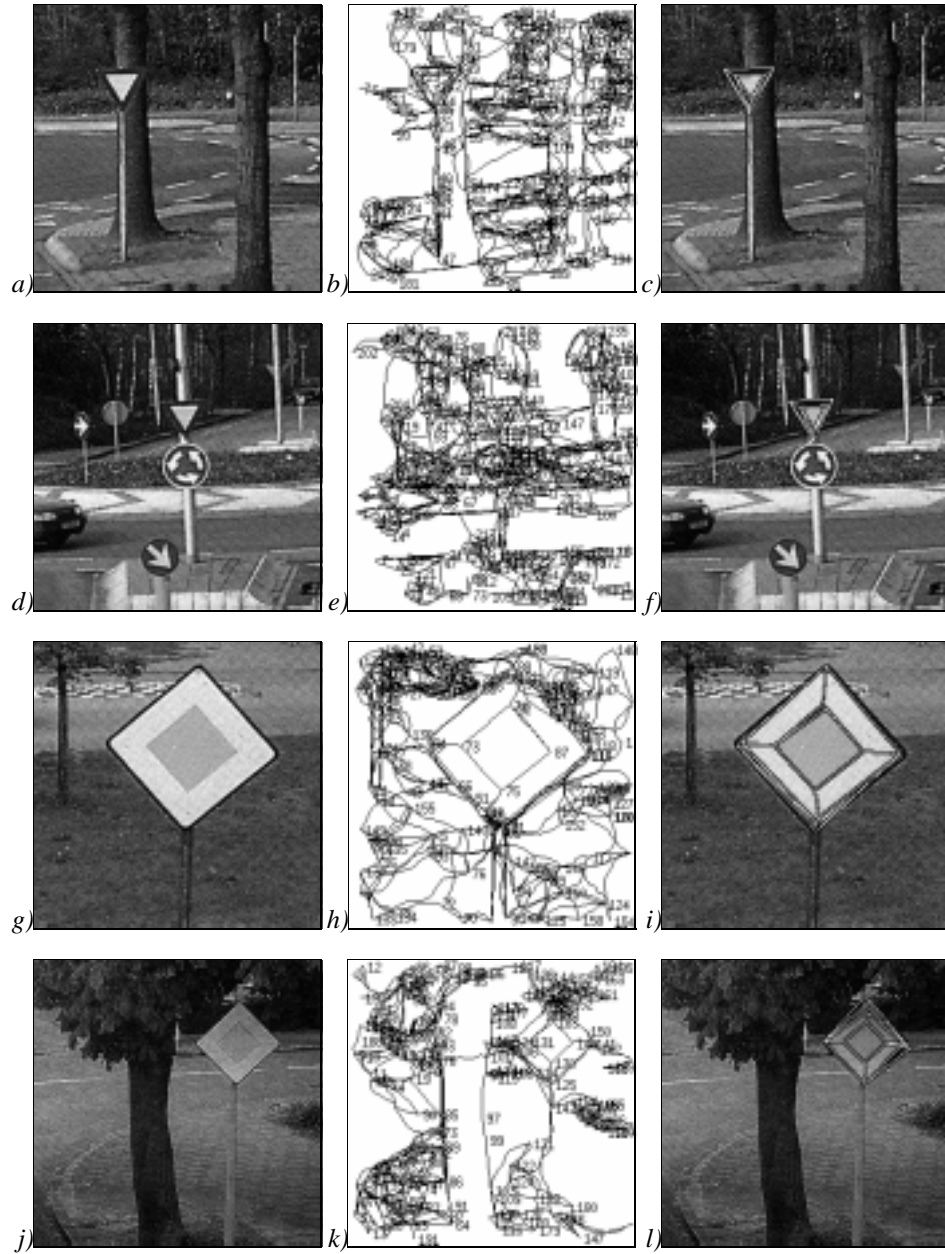


Figure 8.7: Input images, extracted graphs, and found matches, left, middle, and right column, respectively. The parameters used are: the average angle error is three, the average length error is five, the maximum angle error is five, the maximum length error is 14, the ratio tolerance is one, and the maximum cost for tolerating missing edges is 15.

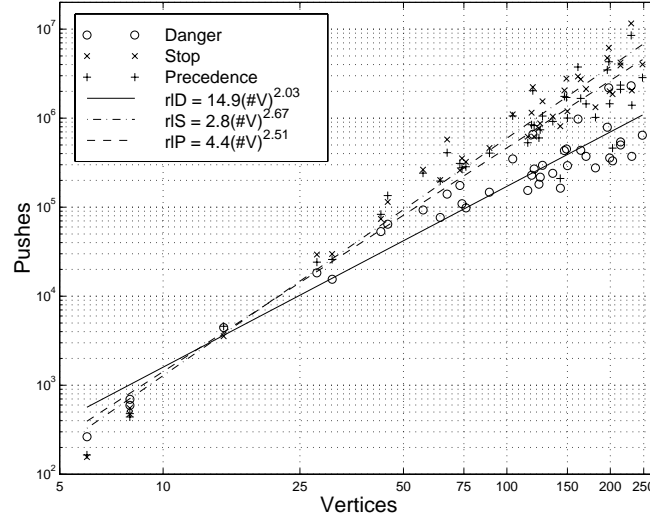


Figure 8.8: Matching time measured in pushes and the regression lines rID , rIS , and rIP of the three used model graphs. The correlations found are 0.96, 0.98, and 0.97, respectively.

matching algorithm.

Four images containing traffic signs are illustrated in the left column of Figure 8.7. In the middle column the corresponding image graphs are illustrated. The image graphs contain 214 and 586, 248 and 681, 161 and 455, and 196 and 461 corners and edges for the first, second, third, and fourth row, respectively. The rightmost column illustrates the detected traffic signs.

Resulting complexity

Figure 8.8 gives the graph matching time measured in number of pushes, for all 42 image graphs. The matching time has an upper bound of approximately $\#V^3$. This is not what one would expect directly since three vertices are needed for the evaluation of the first angle and ratio, see also (8.5). This would suggest already $\#V^3$ but we should take into account that not every triple is possible. For example if one or both edges should be tolerated the cost can be higher than we tolerate. For all three models we tolerate an edge cost of 15 which can be exceeded directly by the cost of tolerating the first edge, therefore a partial match containing a single vertex can be rejected.

On average 89 matches are obtained (but not all accepted) for the danger sign which has six corners, seven edges, and eight angles (6-7-8). On average 39 matches are found for the stop sign (8-8-8). This is almost the most matching time consuming model graph that can be constructed with eight vertices, only a chain is worse (8-7-6). The precedence model graph (8-10-12) has eight matches on average. Extending model graphs with more vertices does increase the matching time when the number of vertices is small and almost constant

for a larger number of vertices since almost all partial matches are already rejected before. Extending the number of edges in the model graph will decrease the matching time. This implies that the matching time is expected not to increase dramatically with large model graphs.

In Figure 8.8 the regression lines indicate the average amount of pushes needed to match a model. The average number of pushes needed are $15(\#V)^{2.0}$, $3(\#V)^{2.7}$, and $4(\#V)^{2.5}$ for the danger, stop, and precedence model, respectively. The correlations found are good which implies that these regression lines are reliable.

8.4 Related work and discussion

In literature, most of the inexact matching algorithms deal with error-correcting subgraph isomorphisms and very little are about error tolerating subgraph isomorphisms. The difference between the two is that the first always finds one subgraph isomorphism with a distance or cost and the second delivers all subgraph isomorphisms within a known distance or cost. Hence, it is very difficult to compare the time complexity of these problems with each other.

Messmer [134] used an error-tolerant subgraph isomorphism for detection and recognition of graphic symbols. His algorithm is tolerant to missing vertices and edges, and even merges vertices during matching. The angle attribute is used to increase the speed of matching, these angles are assigned to the edges and therefore a directed graph is necessary. Unfortunately nothing is mentioned about the time complexity.

Eshera and Fu [40, 41] proposed an algorithm for the error-correcting subgraph isomorphism. They assigned attributes to vertices and edges to impose restrictions to the graphs. They called these graphs attributed relational graphs (ARGs), which use three different features types (straight line, arc, and curve) and have relations (joint, intersection, and facing) between these features. For these relations the distance and angle attributes are used. This algorithm is of polynomial time, approximately $O(l^3m^2)$, where l and m are the number of links in the two graphs. The main problem of this algorithm is the preprocessing. It is difficult to match the segments with one of the three feature types.

More recently Gold and Rangarajan [55] used ARGs for image representation and analysis. Their probabilistic algorithm has a time complexity of $O(lm)$. They applied the algorithm to the error-correcting subgraph isomorphism problem to match an image containing a cup (model) with an image that contains a coffee pot and a cup. They manually marked 10 corners and 8 additional points in the model and approximately 40 corners and approximately 40 additional points in the other image. They needed about 30 seconds to find the match. If we assume that we need $\#V^3$ pushes and that we can evaluate 10,000 pushes on the same machine then we need about 6 seconds to find the match. With our algorithm we are faster and guarantee that if there is an (inexact) subgraph isomorphism it will be found.

A complementary method is labeled graph matching which is applied successfully for, e.g. invariant face recognition (Lades et al. [104]). In this method, vertices are assigned local texture elements and edges carry the geometrical information about relative locations. This method is limited to richly structured or textured objects like faces. Objects with homogeneous surfaces do not provide the sort of vertex labels required there and can only be

matched using their outlines.

8.5 Summary

In this chapter we presented an inexact graph matching algorithm. The algorithm has as input an image graph and a set of model graphs and returns as output all possible inexact matches of the different model graphs in the image graph. The model graphs represent an object or part of it, therefore these graphs all contain useful corner and edge information. We used the angle between two edges and the ratio between these edges (length), to speed up the matching process. The algorithm is edge and distortion tolerant but we do not allow missing vertices. Hence, we need a good corner detector.

In worst case this algorithm has exponential time complexity. The worst case situation is a fact if the fault tolerances of angle and ratio are chosen in such way that any arbitrary choice is accepted. If we choose the tolerances for angle and ratio properly, such that an object is perceived correctly, we cut down computational costs considerably and will detect proper objects only. The algorithm has an empirical upper bound of $\#V^3$, which is fast enough for our purpose since the number of corners in an image is usually less than 250. At most 11 minutes on a workstation is needed if we assume both upper bounds.

The matching speed can be increased more by assigning more attributes to the vertices. In Chapter 5 we divided the corners into six different classes which gives a vertex attribute and in Chapter 7 we could use the information in which color channels the corner is found which gives another vertex attribute.

The implementation of the matching algorithm is based on a stack with units containing partial solutions (Table 8.1). Every partial solution can be evaluated independently and this makes it suitable for a parallel implementation. The choice for one global stack or several local stacks (for every process or processor one) can be elaborated on in more detail to get the optimal performance.

When a match is found, we use a cost function as selection criterion based on straight edges between the corners. The selection criterion becomes more accurate when the curved segments are used (Chapter 6). They can also be used to get a better description of the model graphs. When curved segments are used we can apply a selection criterion by surface difference.

Concluding Remarks

IN THIS THESIS we developed an artificial vision model, inspired by biological vision models. It is able to recognize objects that are spanned by corners from color images. This chapter summarizes the contents of the developed model and gives some directions for future extensions and improvements of the model.

9.1 Summary and conclusions

In Chapter 3 we gave an overview of early vision from a computational point of view. From this chapter we included (in the model) the S -operators and C -operators which model the responses of simple and complex cells, respectively. We used the C -operator for line and edge enhancement, since this operator responds strongly and exactly at the positions where the lines and edges appear.

In Chapter 4 we used a corner enhancement operator, as proposed by Heitger et al. [71]. This so-called E -operator models the response of end-stopped cells. The operator responds well at most corners and junctions, but is sensitive to noise at small scales. To avoid noise responses we proposed a new corner detector by a multi-scale combination of the E -operator (Chapter 5). Other advantages of the multi-scale E -operator is that it responds well to rounded corners and that it performs better than the standard corner enhancement operators.

In Chapter 7 the C -operator and E -operator are generalized to color channels. We used the same properties of the complex and end-stopped cells and assumed that these cells respond excitatory to one color and inhibitory to another color. This generalization made the corner operator perform significantly better than by using a single achromatic channel.

In Chapter 6 we developed a line detection algorithm, based on the assumption that corners are more stable than lines, and used it to extract line-segments by following enhanced edges from one corner to another.

The extracted lines and corners are represented as a graph. This representation is suited for objects in which all edges are spanned by corners. We chose this representation because it is compact and leads to invariance under translation, rotation, and scaling.

A graph matching algorithm is used to recognize a-priori known objects (Chapter 8). Since

graph matching is an NP-hard problem, we used an attributed graph format. The choice of the attributes (angles and lengths) in the graph led to a reduction of evaluations from $N!$ to less than N^3 , and robustness under small perspective changes and undetected lines. The used graph matching algorithm is optimal, i.e., it recognizes all known objects in the image within known constraints.

The human visual system has limited space for the storage of different objects. Hence, objects should be stored in a compact and quickly accessible way. How an object is represented in the brain is unknown. It is likely that specific features are extracted from a visual scene (image) and that a set of these features, which represent an object, are compared with stored (learned) objects. Hence, attributed graphs may be part of the representation used in the brain, and graph matching may be a part of the matching process as it takes place in the brain.

To summarize, the model is sequentially divided into:

1. the *enhancement* of features (e.g., edges, corners, and color) in the input image,
2. the *extraction* of the features,
3. and *recognition* by matching these features.

9.2 Future research

In a few cases the model fails to recognize objects, due to undetected corners. In Chapter 5 we reported that the E -operator does not respond to corners with an angle larger than 130 degrees, and lacks to respond to 4-junctions. A possible way to detect these corners and 4-junctions is to extend the operator. Another option is to add missing corners when edges are extracted.

In the current model we used corners and edges only. The model can be extended by filling in the areas enclosed by edges using color and texture. Texture and color can be used to improve and facilitate edge and corner detection, and gives additional information to the model and image graph. Additional attributes make the graph more specific, which will increase the speed of matching.

The model can be easily extended for stereo vision, since both images can be processed separately until from both images an image graph is extracted. From these two graphs we should create a new graph which describes a three-dimensional scene by solving the *correspondence problem* for corners. Since these graphs contain the same attributes as two-dimensional graphs a similar graph matching algorithm can be used.

Furthermore, we want to extend the model in such a way that it automatically extracts model graphs. If a model graph is extracted automatically and is not known, it will ask its teacher what kind of object it is. Similar to a young child who perceives an unknown object will ask what kind of object it is.

All the extensions up to now deal with static images; our final goal is to use a model which detects objects in a dynamic vision environment. However, including temporal dynamics is a really difficult task and with the current speed of computers hardly possible.

A

Properties of Gaussian and Mexican-hat Functions

DEFINITIONS for one- and two-dimensional Gaussian and Mexican-hat functions are given and their properties discussed. The properties should be known because in the used models both two-dimensional Gaussian and Mexican-hat functions are used as filters with different radii. This means that if one of these functions is used as a windowed filter, the radius of such a filter should be known, i.e. the cut-off radius, since these functions will never reach zero exactly. If filters with different radii are used, the normalization factor should also be known if one wants to compare the filtered results.

In this appendix we analyze the cut-off radius of both the Gaussian and the Mexican-hat function. For the Mexican-hat function, based on the second derivative or Laplacian of the Gaussian, we determine the center radius and normalization factors. The ratio between center and surround radius is constant if a Mexican-hat based on the second derivative or Laplacian is used, therefore also the Mexican-hat based on the difference of two Gaussians (DOGs) is used. These DOGs are chosen so that they are normalized. We determine the possible ratios between center and surround radii and the minima of the DOGs.

A.1 Properties of Gaussians

The Gaussian function is as follows:

$$G_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (\text{A.1})$$

The function is normalized, which is useful when it is used as a filter and necessary if a scale space is used and the results within this scale space are compared with each other.¹ If we use a Gaussian filter, μ is always set to zero. The one and two-dimensional Gaussian functions

¹The outputs can of course also be compensated for afterwards but this is computationally less attractive.

then are as follows:

$$G_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (\text{A.2})$$

and

$$G_{\sigma}(x, y) = G_{\sigma}(x) \cdot G_{\sigma}(y) = \frac{1}{\sigma^2 2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (\text{A.3})$$

A.1.1 Cut-off radius of a Gaussian

The Gaussian function approaches zero very fast, but it never reaches zero exactly. We approximate the Gaussian function by zero at those points where the exponential function is $\leq \varepsilon$, for some small constant $\varepsilon > 0$. The corresponding argument r is called the *cut-off radius*. So we have

$$e^{-\frac{r^2}{2\sigma^2}} = \varepsilon \quad (\text{A.4})$$

$$\equiv r = \sigma \sqrt{-2 \log \varepsilon}. \quad (\text{A.5})$$

This is identical for both one and two-dimensional Gaussian functions. For the latter, we substitute $r = \sqrt{x^2 + y^2}$.

A.2 Properties of the Mexican-hat function

The standard Mexican-hat function is the second derivative of the Gaussian, when it is a one-dimensional function, and the Laplacian, $\nabla^2 G_{\sigma} = \frac{\partial^2 G_{\sigma}}{\partial x^2} + \frac{\partial^2 G_{\sigma}}{\partial y^2}$, of a Gaussian if it is a two-dimensional function. Since in both cases we obtain a Mexican-hat which is upside down, we multiply it with -1 and get:

$$M_{\sigma}(x) = \frac{\sigma^2 - x^2}{\sigma^5 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}. \quad (\text{A.6})$$

and

$$M_{\sigma}(x, y) = \frac{2\sigma^2 - (x^2 + y^2)}{2\sigma^6 \pi} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (\text{A.7})$$

A.2.1 The cut-off radii of the Mexican-hat function

The *center radius* of the one-dimensional Mexican-hat function is

$$\sigma^2 - r^2 = 0 \quad \equiv \quad r = \sigma. \quad (\text{A.8})$$

For the two-dimensional Mexican-hat function the center radius is

$$2\sigma^2 - r^2 = 0 \quad \equiv \quad r = \sqrt{2}\sigma. \quad (\text{A.9})$$

The surround cut-off radius is chosen to be the same for both one- and two-dimensional Mexican-hat functions and are chosen to be identical to the cut-off radius of the Gaussians from (A.5).

From the equations (A.8), (A.9), and (A.5), we conclude that the ratio between center and surround is $1 : \sqrt{-2\log \epsilon}$ for the one-dimensional and $1 : \sqrt{-\log \epsilon}$ for the two-dimensional Gaussian or Mexican-hat function. To minimize aliasing effects $\epsilon = e^{-9}$ is a good choice, which is found empirically when the two-dimensional Mexican-hat function was used as a discrete filter (Lourens [120, 121]).

A.2.2 Normalization of the Mexican-hat function

Since the one-dimensional Mexican-hat function is the derivative of a function which vanishes (rapidly) at $\pm\infty$, its integral is zero:

$$\int_{r=-\infty}^{\infty} \frac{\sigma^2 - r^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}} dr = 0. \quad (\text{A.10})$$

The integral of the two-dimensional Mexican-hat function (without constant $\frac{1}{2\sigma^2\pi}$) is

$$\begin{aligned} & \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} \frac{2\sigma^2 - (x^2 + y^2)}{\sigma^4} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} dx dy \\ &= \int_{\varphi=0}^{2\pi} \int_{r=0}^{\infty} \frac{2\sigma^2 - r^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}} r dr d\varphi \\ &= \left\{ a = 2\sigma^2, r^2 = at \right\} \\ &= \left(\int_{\varphi=0}^{2\pi} d\varphi \right) \int_{t=0}^{\infty} \frac{2a^2}{a^2} (1-t) e^{-t} dt \\ &= 4\pi \int_{t=0}^{\infty} (1-t) e^{-t} dt \\ &= \{\text{partial integration}\} \\ &= 0. \end{aligned} \quad (\text{A.11})$$

It can be useful to normalize the Mexican-hat function in such a way that its central part has integral zero. For instance if we use Mexican-hat filters with different radii, where the results need to be compared with each other. As an example we give a continuous input image filtered with a Mexican-hat filter with radius r and the same input image which is a factor two larger and filtered with a Mexican-hat filter with radius $2r$. If the results of the two filtered images should be compared, it is necessary to normalize the center of the Mexican-hat function.

The normalization factor for the center of a one-dimensional Mexican-hat function can be obtained by calculating the integral over the center part of the Mexican-hat function:

$$\int_{-\sigma}^{\sigma} \frac{\sigma^2 - r^2}{\sigma^5 \sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}} dr$$

$$\begin{aligned}
&= \int_{-1}^1 \frac{\sigma^2 - \sigma^2 t^2}{\sigma^5 \sqrt{2\pi}} e^{-\frac{t^2}{2}} \sigma dt \\
&= \frac{1}{\sigma^2 \sqrt{2\pi}} \int_{-1}^1 (1 - t^2) e^{-\frac{t^2}{2}} dt \\
&= \{\text{partial integration}\} \\
&= \frac{1}{\sigma^2 \sqrt{2\pi}} \left(\int_{-1}^1 e^{-\frac{t^2}{2}} dt + t e^{-\frac{t^2}{2}} \Big|_{-1}^1 - \int_{-1}^1 e^{-\frac{t^2}{2}} dt \right) \\
&= \frac{1}{\sigma^2} \sqrt{\frac{2}{e\pi}}.
\end{aligned} \tag{A.12}$$

The normalization factor for the center of a two-dimensional Mexican-hat function is as follows:

$$\begin{aligned}
&\int_x \int_y \frac{2\sigma^2 - (x^2 + y^2)}{\sigma^6 2\pi} e^{-\frac{x^2 + y^2}{2\sigma^2}} dx dy \\
&= \int_{\varphi=0}^{2\pi} \int_{r=0}^{\sigma\sqrt{2}} \frac{1}{\sigma^6 2\pi} (2\sigma^2 - r^2) e^{-\frac{r^2}{2\sigma^2}} r dr d\varphi \\
&= \int_{\varphi=0}^{2\pi} \left\{ r = \sigma\sqrt{2}t, dr = \frac{\sigma}{\sqrt{2}} dt \right\} \\
&= \frac{1}{2\pi\sigma^6} \left(\int_0^{2\pi} d\varphi \right) \int_{t=0}^1 (2\sigma^2 - 2\sigma^2 t) e^{-t} \sigma^2 dt \\
&= \frac{2}{\sigma^2} \left(\int_0^1 (1 - t) e^{-t} dt \right) \\
&= \{\text{partial integration}\} \\
&= \frac{2}{\sigma^2} \left(\int_0^1 e^{-t} dt + t e^{-t} \Big|_0^1 - \int_0^1 e^{-t} dt \right) \\
&= \frac{2}{\sigma^2 e}.
\end{aligned} \tag{A.13}$$

A.3 The Mexican-hat function using the difference of two Gaussians

The Mexican-hat function based on the difference of two Gaussians (DOGs) has the advantage that we can control both center and surround, this in contrast to the Mexican-hat function in Section A.2 where the ratio between center and surround is constant.

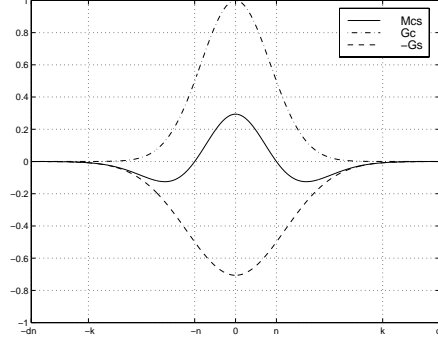


Figure A.1: Gaussian functions $G_{c\sigma}$ and $G_{s\sigma}$ have cut-off radii of k and dn , respectively. The Mexican-hat function $M_{cs\sigma}$ is the sum of $G_{c\sigma}$ and $-G_{s\sigma}$. Note that the center radius of the Mexican-hat function is not equal with the cut-off radius of $G_{c\sigma}$.

A.3.1 One-dimensional Mexican-hat function based on DOGs

Instead of the Mexican-hat function defined by formula (A.7) one can also use a Mexican-hat function based on the difference of two Gaussian functions:

$$M_{\sigma,m}(r) = G_{\sigma}(r) - G_{m\sigma}(r), \quad (\text{A.14})$$

for some $m > 1$. Since both Gaussians are normalized this implies that $\int_{-\infty}^{\infty} M_{\sigma,m}(r) dr = 0$. Normalization of $\int_{-\infty}^{\infty} |M_{\sigma,m}(r)| dr = 0$ is only possible in a numerical way. Since normalization is not possible analytically we multiply $M_{\sigma,m}$ with a factor $\sigma\sqrt{2\pi}$ to be able to control the two Gaussians and to simplify both Gaussians.

For convenience we will write the first Gaussian in equation (A.14) multiplied with this factor as

$$G_{c\sigma}(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad (\text{A.15})$$

the second Gaussian as

$$G_{s\sigma}(r) = \frac{1}{m} e^{-\frac{r^2}{2\sigma^2 m^2}}, \quad (\text{A.16})$$

and the Mexican-hat as

$$M_{cs\sigma}(r) = G_{c\sigma}(r) - G_{s\sigma}(r). \quad (\text{A.17})$$

A.3.2 Controlling the difference of two Gaussians

If a Mexican-hat function is used as a filter and one wants to control both center and surround of the filter then the function should be normalized because when used as a filter will have better properties.

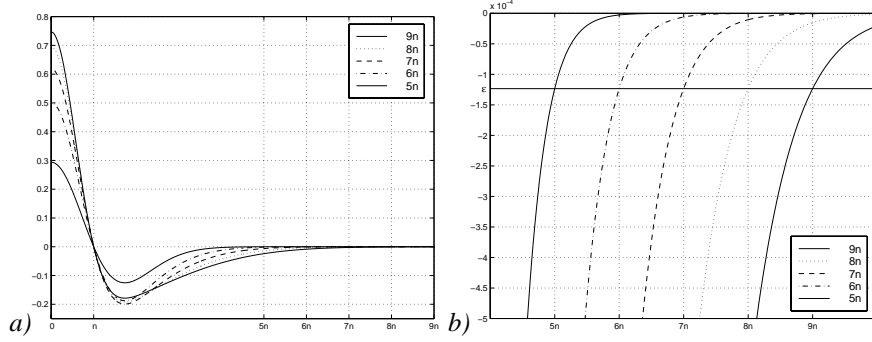


Figure A.2: a) Five different Mexican-hat functions all with the same center radius n but different surround radii, the different surround radii are $5n$, $6n$, $7n$, $8n$, and $9n$, respectively. b) The five different Mexican-hat functions, enlarged at one of the surround borders. The horizontal line is $\varepsilon = -e^{-9}$.

Let us assume that the center of the Mexican-hat function has radius n and that the surround has a radius dn (see also Figure A.1). Given the center and surround Gaussians from (A.15) and (A.16), we should find out how σ and m should be chosen to get a Mexican-hat center radius n and a surround radius of dn (Figure A.2).

For the center radius n of the Mexican-hat function yields:

$$Mcs_{\sigma}(n) = 0 \quad (\text{A.18})$$

$$\equiv Gc_{\sigma}(n) = Gs_{\sigma}(n)$$

$$\equiv e^{-\frac{n^2}{2\sigma^2}} = \frac{1}{m} e^{-\frac{n^2}{2\sigma^2 m^2}}$$

$$\equiv -\frac{n^2}{2\sigma^2} = -\frac{n^2}{2\sigma^2 m^2} - \log m$$

$$n = \sigma \sqrt{\frac{2 \log m}{1 - \frac{1}{m^2}}}. \quad (\text{A.19})$$

From which we conclude that there is a relation between σ and m . For the surround radius of the Mexican-hat function yields:

$$Mcs_{\sigma}(dn) = -\varepsilon \quad (\text{A.20})$$

$$\equiv Gc_{\sigma}(dn) - Gs_{\sigma}(dn) = -\varepsilon$$

$$\approx \{Gc_{\sigma}(dn) \approx 0\}$$

$$Gs_{\sigma}(dn) = \varepsilon$$

$$\equiv e^{-\frac{d^2 n^2}{2\sigma^2 m^2}} = m\varepsilon$$

$$\equiv \{(A.19)\}$$

$$e^{\frac{d^2 \log m}{1 - m^2}} = m\varepsilon$$

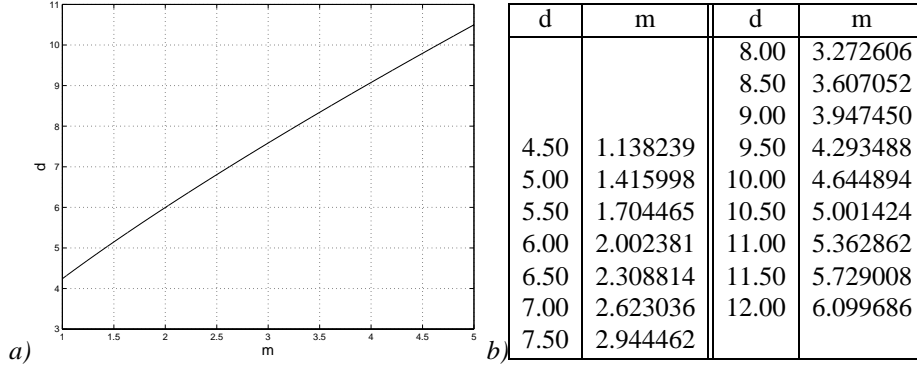


Figure A.3: a) The relation between m and d . Note that d is at least four. This means that only a ratio between center and surround can be realized where the surround radius is at least four times larger than the center radius. b) The relation between m and d .

$$\begin{aligned}
 &\equiv \frac{d^2 \log m}{1 - m^2} = \log(m\epsilon) \\
 &\equiv d = \sqrt{\frac{\log(m\epsilon)(1 - m^2)}{\log m}}.
 \end{aligned} \tag{A.21}$$

Note that if we had used the Mexican-hat function from (A.14) we would have had $\log(\sigma\sqrt{2\pi m\epsilon})$ instead of $\log(m\epsilon)$ in (A.21).

A.3.3 Two-dimensional Mexican-hat function based on DOGs

Similar to the one-dimensional Mexican-hat based on DOGs we use two Gaussians. To simplify these two-dimensional Gaussians we multiply with a factor $2\pi\sigma$ and obtain a $Gc_\sigma(r)$ which is identical with (A.15) and

$$Gs_\sigma(r) = \frac{1}{m^2} e^{-\frac{r^2}{2\sigma^2 m^2}}, \tag{A.22}$$

where $r = \sqrt{x^2 + y^2}$. Equation (A.19) changes, since $\log(m^2) = 2\log m$, it will be multiplied by a factor $\sqrt{2}$ and becomes:

$$n = 2\sigma \sqrt{\frac{\log m}{1 - \frac{1}{m^2}}}. \tag{A.23}$$

Equation (A.21) changes into:

$$d = \sqrt{\frac{\log(m^2\epsilon)(1 - m^2)}{2\log m}}. \tag{A.24}$$

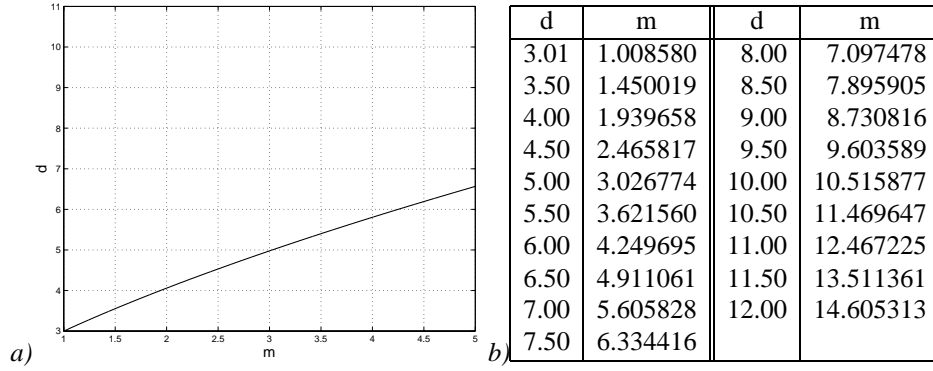


Figure A.4: a) The relation between m and d in the two-dimensional case. Note again that d is at least three. This means that only a ratio between center and surround can be realized where the surround radius is at least three times larger than the center radius. b) The relation between m and d in the two-dimensional case.

A.3.4 Minima of the Mexican-hat function based on DOGs

By construction the maximum in the Mexican-hat is at the origin ($r = 0$). But in Figure A.2 one can see that the minimum value of the Mexican-hat seems to be constant independent of the ratio between center and surround. Let us calculate the derivative of the Mexican-hat function to check if the location of the minimum is indeed independent of the ratio between center and surround.

The minimum of the one-dimensional Mexican-hat is at:

$$\begin{aligned}
 & \frac{d}{dr} M_{cs\sigma}(r) = 0 & (A.25) \\
 \equiv & -\frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} + \frac{r}{\sigma^2 m^2} \frac{1}{m} e^{-\frac{r^2}{2\sigma^2 m^2}} = 0 \\
 \equiv & \frac{r}{\sigma^2} \left(\frac{1}{m^3} e^{-\frac{r^2}{2\sigma^2 m^2}} - e^{-\frac{r^2}{2\sigma^2}} \right) = 0 \\
 \equiv & \{r = 0 \text{ gives the maximum}\} \\
 & \frac{1}{m^3} e^{-\frac{r^2}{2\sigma^2 m^2}} = e^{-\frac{r^2}{2\sigma^2}} \\
 \equiv & r = \sigma \sqrt{\frac{6 \log m}{1 - \frac{1}{m^2}}} \\
 \equiv & \{(A.19) \text{ and } r \geq 0\} \\
 & r = n\sqrt{3}. & (A.26)
 \end{aligned}$$

This proves that the minimum is center surround ratio independent. In the two-dimensional case the minimum of the Mexican-hat is at $r = n\sqrt{2}$.

Bibliography

- [1] E. H. Adelson and J. R. Bergen (1985). Spatio-temporal energy models for the perception of motion. *Journal of the Optical Society of America*, **2**, 284–299.
- [2] H. A. Almohamad and S. O. Duffuaa (1993). A linear programming approach for the weighted graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 522–525.
- [3] I. M. Anderson and J. C. Bezdek (1984). Curvature and tangential deflection of discrete arcs: a theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 27–40.
- [4] B. W. Andrews and D. A. Pollen (1979). Relationship between spatial frequency selectivity and receptive field profile of simple cells. *J. Physiol.*, **287**, 163–176.
- [5] H. Asada and M. Brady (1986). The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 2–14.
- [6] J. J. Atick and A. N. Redlich (1990). Mathematical model of the simple cells in the visual cortex. *Biological Cybernetics*, **63**, 99–109.
- [7] D. H. Ballard (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, **13**, 111–122.
- [8] D. H. Ballard and C. M. Brown (1982). *Computer Vision*. Prentice-Hall Inc.
- [9] H. B. Barlow (1953). Summation and inhibition in the frog's retina. *Journal of Physiology, London*, **119**, 69–88.
- [10] S. Baugher and A. Rosenfeld (1992). Corner detection and localization in a pyramid. In *Computer Vision and Image Processing*, pp. 103–121.
- [11] D. A. Baylor and M. G. F. Fuortes (1970). Electrical responses of single cones in the retina of the turtle. *J. Physiol.*, **207**, 77–92.
- [12] D. A. Baylor, B. J. Nunn, and J. L. Schnapf (1987). Spectral sensitivity of cones of the monkey macaca fascicularis. *J. Physiol.*, **390**, 145–160.
- [13] P. R. Beaudet (1978). Rotational invariant image operators. In *4th International Conference on Pattern Recognition*, pp. 579–583, Tokyo.
- [14] W. Beaudot, P. Palagi, and J. Hérault (1993). Realistic simulation tool for early vision processing including space, time and colour data. In *New Trends in Neural Computation, Proceedings of IWANN'93*, edited by J. Mira, J. Cabestany, and A. Prieto, *Lecture Notes in Computer Science*, **686**, 370–375. Springer-Verlag.
- [15] G. G. Blasdel, J. S. Lund, and D. Fitzpatrick (1985). Intrinsic connections of macaque striate cortex: axonal projections of cells outside lamina 4c. *J. Neurosci.*, **5**, 3350–3369.
- [16] P. K. Brown and G. Wald (1964). . *Science*, **144**, 45–52.

- [17] H. Bunke and B. T. Messmer (1995). Efficient attributed graph matching and its application to image analysis. In *8-th International Conf. ICIAP'95, Lecture Notes in Computer Science -Image Analysis and Processing*, edited by C. Braccini, L. DeFloriani, and G. Vernazza, **974**, 45–55, San Remo, Italy.
- [18] J. M. Van Buren (1963). *The Retinal Ganglion Cell Layer*. Charles C. Thomas, Springfield, Illinois, U.S.A.
- [19] J. Canny (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 679–698.
- [20] W. J. Christmas, J. Kittler, and M. Petrou (1995). Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 749–764.
- [21] D. G. Corneil and C. C. Gotlieb (1970). An efficient algorithm for graph isomorphism. *Journal of the ACM*, **17**, 51–64.
- [22] P. M. Daniel and D. Whitteridge (1961). The representation of the visual field on the cerebral cortex in monkeys. *Journal of Physiology*, **159**, 203–221.
- [23] John G. Daugman (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Amer.*, **2**, 1160–1169.
- [24] John G. Daugman (1988). Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36**, 1169–1179.
- [25] N. W. Daw (1968). Colour-coded ganglion cells in the goldfish retina: Extension of their receptive fields by means of new stimuli. *J. Physiol.*, **197**, 567–592.
- [26] N. W. Daw (1984). The psychology of colour vision. *Trends in Neuroscience*, **7**, 330–335.
- [27] R. Deriche and O. D. Faugeras (1990). 2-d curve matching using high curvature points: Application to stereo vision. In *Proceedings of the 10th International Conference on Pattern Recognition*, pp. 240–242, Atlantic City.
- [28] R. Deriche and G. Giraudon (1990). Accurate corner detection: An analytical study. In *Proceedings of the 3rd International Conference on Computer Vision*, pp. 66–70, Osaka.
- [29] R. Deriche and G. Giraudon (1993). A computational approach for corner and vertex detection. *International Journal of Computer Vision*, **10**, 101–124.
- [30] R. L. DeValois, C. J. Smith, S. T. Kitai, and A. J. Koroly (1958). . *Science*, **127**, 238–259.
- [31] E. A. DeYoe and D. C. Van Essen (1988). Concurrent processing streams in monkey visual cortex. *TINS*, **11**, 219–226.
- [32] A Dobbins, S. W. Zucker, and M. S. Cynader (1987). Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature, London*, **329**, 438–441.
- [33] A. Dobbins, S. W. Zucker, and M. S. Cynader (1989). Endstopping and curvature. *Vision Research*, **29**, 1371–1387.
- [34] J. E. Dowling (1987). *The retina –An Approachable Part of the Brain*. Harvard University Press, Cambridge, Mass.
- [35] L. S. Dreschler and H. H. Nagel (1982). On the selection of critical points and local curvature extrema of region boundaries for interframe matching. In *International Conf. on Pattern Recognition*, pp. 542–544.

- [36] R. O. Duda and P. E. Hart (1972). *Pattern Classification and Scene Analysis*. Wiley, New York.
- [37] R. O. Duda and P. E. Hart (1972). Use of the hough transform to detect lines and curves in pictures. *Communications of the ACM*, **15**, 11–15.
- [38] C. Enroth-Cugell and J. G. Robson (1966). The contrast sensitivity of the retinal ganglion cells of the cat. *Journal of Physiology*, **187**, 517–552.
- [39] C. Enroth-Cugell, J. G. Robson, D. E. Schweitzer-Tong, and A. B. Watson (1983). Spatio-temporal interactions in cat retinal ganglion cells showing linear spatial summation. *Journal of Physiology*, **341**, 279–307.
- [40] M. A. Eshera and King-Sun Fu (1984). A graph distance measure for image analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, **14**, 398–408.
- [41] M. A. Eshera and King-Sun Fu (1986). An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 604–618.
- [42] R. Espelid and I. Jonassen (1991). A comparison of splitting methods for the identification of corner-points. *Pattern Recognition Letters*, **12**, 79–83.
- [43] B. Falkenhainer, K. D. Forbus, and D. Gentner (1990). The structure-mapping engine: Algorithms and examples. *Artificial Intelligence*, **41**, 1–63.
- [44] W. A. Fellenz (1997). *Ein neuromorphes System für die datengetriebene Szenenanalyse*, volume 479 of *10: Informatik/Kommunikationstechnik*. VDI Verlag GmbH.
- [45] B. Fischer (1973). Overlap of receptive field centers and representation of the visual field in the cat's optic tract. *Vision Research*, **13**, 2113–2120.
- [46] M. Fischler and R. Bolles (1983). Perceptual organization and curve partitioning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 38–46.
- [47] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, second edition.
- [48] H. Freeman (1961). On the encoding of arbitrary geometric configurations. *IEEE Trans. Elec. Computers*, **10**, 260–268.
- [49] H. Freeman and L. Davis (1977). A corner-finding algorithm for chain coded curves. *IEEE Transactions on Comput.*, **26**, 297–303.
- [50] K. S. Fu (1983). A step towards unification of syntactic and statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**, 200–205.
- [51] K. Fukushima (1970). A feature extractor for curvilinear patterns: A design suggested by the mammalian visual system. *Kybernetik*, **7**, 153–160.
- [52] Kunihiro Fukushima and Masayuki Kikuchi (1995). Binding of form and motion by selective visual attention: A neural network model. In *Brain Processes, Theories, and Models. An International Conference in Honor of W.S. McCulloch 25 Years after his Death*, edited by R. Moreno-Díaz and J. Mira-Mira, pp. 436–445, Las Palmas de Gran Canaria, Spain. MIT press, Cambridge USA.
- [53] D. Gabor (1946). Theory of communication. *J. Inst. Elec. Eng.*, **93**, 429–457.
- [54] M. R. Garey and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York.
- [55] S. Gold and A. Rangarajan (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 377–388.

- [56] R. C. Gonzalez and P. Wintz (1987). *Digital Image Processing*. Addison Wesley, second edition.
- [57] A. Gottschalk and G. Buchsbaum (1983). Information theoretic aspects of color signal processing in the visual system. *IEEE Transactions on Syst. Man Cybernetics*, **13**, 864–873.
- [58] Peter Gouras and Jürgen Krüger (1979). Responses of cells in foveal visual cortex of the monkey to pure color contrast. *Journal of Neurophysiology*, **42**, 850–860.
- [59] A. Guiducci (1988). Corner characterization by differential geometry techniques. *Pattern Recognition Letters*, **8**, 311–318.
- [60] M. H. Han, D. Jang, and J. Foster (1989). Identification of cornerpoints of two-dimensional images using a line search method. *Pattern Recognition*, **1**, 13–20.
- [61] R. M. Haralick and L. G. Shapiro (1992). *Computer and Robot Vision*. Addison-Wesley.
- [62] C. Harris (1987). Determination of ego-motion from matched points. In *Proc. Alvey Vis. Conf.*, Cambridge, UK.
- [63] C. Harris and M. Stephens (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vis. Conf.*, pp. 189–192, Manchester.
- [64] R. Hartley and A. Rosenfeld (1985). Hierarchical line linking for corner detection. In *Integrated Technology for Parallel Image Processing*, edited by S. Leviali, pp. 101–119, New York. Academic Press.
- [65] H. K. Hartline (1940). The receptive field of the optic nerve fibers. *Amer. J. Physiol.*, **130**, 690–699.
- [66] M. J. Hawken and A. J. Parker (1987). Spatial properties of neurons in the monkey striate cortex. In *Proceedings of the Royal Society of London, B*, **231**, 251–288.
- [67] D. J. Heeger (1992). Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, **9**, 181–198.
- [68] D. J. Heeger (1994). The representation of visual stimuli in primary visual cortex. *Current Directions in Psychological Science*, **3**, 159–163.
- [69] P. Heggelund (1985). Receptive field organization of simple and complex cells. In *Models of the visual cortex*, edited by D. Rose and V. G. Dobson, pp. 358–365. Wiley, New York.
- [70] F. Heitger (1995). Feature detection using suppression and enhancement. Technical Report TR-163, Image Science Lab, ETH-Zurich, Switzerland.
- [71] Friedrich Heitger, Lukas Rosenthaler, Rüdiger von der Heydt, Esther Peterhans, and Olaf Kübler (1992). Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, **32**, 963–981.
- [72] E. Hering (1874). Sitzungsberichte der wiener akademie. *Mathem.-Naturwissensch.*, **69**, 131.
- [73] E. Hering (1878). *Zur Lehre vom Lichtsinne*. Springer-Verlag, Berlin.
- [74] E. Hering (1880). Zur erklärang der farbenblindheit aus der theorie der gegenfarben. *Jb. für Naturwiss.*, **1**, 76–107.
- [75] E. Hering (1964). *Outlines of a Theory of the Light Sense*. Harvard Univeristy Press, Cambridge Mass., U.S.A. Translated by L. M. Hurvich and D. Jameson.
- [76] R. Horaud and T. Skordas (1989). Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 1168–1180.
- [77] R. Horaud, F. Veillon, and T. Skordas (1990). Finding geometric and relational structures in an image. In *Proc. 1st Europ. Conf. Comput. Vis.*, pp. 374–384, Antibes, France.

- [78] P. V. C. Hough (1962). Method and means for recognizing complex patterns. *U.S. Patent 3,069,654*.
- [79] D. Hubel and T. Wiesel (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology, London*, **160**, 106–154.
- [80] D. H. Hubel and M. S. Livingstone (1985). Complex-unoriented cells in a subregion of primate area 18. *Nature*, **315**, 325–327.
- [81] D. H. Hubel and M. S. Livingstone (1990). Color and contrast sensitivity in the lateral geniculate body and primary visual cortex of the macaque monkey. *Journal of Neuroscience*, **10**, 2223–2237.
- [82] D. H. Hubel and T. N. Wiesel (1959). Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.*, **148**, 574–591.
- [83] D. H. Hubel and T. N. Wiesel (1961). Integrative action in the cat's lateral geniculate body. *Journal of Physiology, London*, **155**, 385–398.
- [84] D. H. Hubel and T. N. Wiesel (1963). Receptive fields of cells in striate cortex of very young visually inexperienced kittens. *Journal of Neurophysiology*, **26**, 994–1002.
- [85] D. H. Hubel and T. N. Wiesel (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, **28**, 229–289.
- [86] D. H. Hubel and T. N. Wiesel (1968). Receptive fields and functional architecture of monkey striate cortex. *J. Physiology*, **195**, 215–243.
- [87] D. H. Hubel and T. N. Wiesel (1977). Ferrier Lecture functional architecture of macaque visual cortex. *Proc. R. Soc. Lond. B.*, **198**, 1–59.
- [88] David H. Hubel (1982). Exploration of the primary visual cortex, 1955–78. *Nature*, **299**, 515–524.
- [89] David H. Hubel (1988). *Eye, Brain, and Vision*. Scientific American Library, New York.
- [90] David H. Hubel and David C. Freeman (1977). Projection into the visual field of ocular dominance columns in macaque monkey. *Brain Research*, **122**, 336–343.
- [91] David H. Hubel and Torsten N. Wiesel (1974). Uniformity of monkey striate cortex: A parallel relationship between field size, scatter, and magnification factor. *Journal of Comparative Neurology*, **158**, 295–306.
- [92] L. A. Iverson and S. W. Zucker (1995). Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 982–996.
- [93] D. Jameson and L. M. Hurvitch (1959). Perceived color and its dependence on focal, surrounding, and preceding stimulus variables. *J. Opt. Soc. Am.*, **49**, 890–898.
- [94] D. Jameson and L. M. Hurvitch (1961). Opponent chromatic induction: Experimental evaluation and theoretical approach. *J. Opt. Soc. Am.*, **51**, 46–53.
- [95] J. Jones and L. Palmer (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, **58**, 1233–1258.
- [96] E. Kaplan and R. M. Shapley (1986). The primate retina contains two types of ganglion cells, with high and low contrast sensitivity. *Proc. Natl. Acad. Sci. U.S.A.*, **83**, 2755–2757.
- [97] L. Kitchen and A. Rosenfeld (1982). Gray-level corner detection. *Pattern Recognition Letters*, **1**, 95–102.

- [98] J. J. Koenderink and A. J. van Doorn (1978). Visual detection of spatial contrast; influence of location in the visual field, target extent and illuminance level. *Biological Cybernetics*, **30**, 157–167.
- [99] J. J. Koenderink and A. J. van Doorn (1987). Representation of local geometry in the visual system. *Biological Cybernetics*, **55**, 367–376.
- [100] M. Krcmar and A. Dhawan (1994). Application of genetic algorithms in graph matching. In *Proceedings of the International Conference on Neural Networks*, **6**, 3872–3876.
- [101] S. W. Kuffler (1953). Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*, **16**, 37–68.
- [102] J. J. Kulikowski and P. O. Bishop (1981). Linear analysis of the responses of simple cells in the cat visual cortex. *Exp. Brain Res.*, **44**, 386–400.
- [103] P. Kuner and B. Ueberreiter (1988). Pattern recognition by graph matching combinatorial versus continuous optimization. *International Journal on Pattern Recognition and Artificial Intelligence*, **2**, 527–542.
- [104] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, **42**, 300–311.
- [105] E. H. Land (1964). The retinex. *Scien. Am.*, **52**, 247–264.
- [106] E. H. Land (1974). The retinex theory of colour vision. *Proc. R. Inst. Gr. Brit.*, **47**, 23–58.
- [107] J.S. Lee, Y.N. Sun, C.H. Chen, and C.T. Tsai (1993). Wavelet based corner detection. *Pattern Recognition*, **26**, 853–865.
- [108] Tai Sing Lee (1996). Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 959–971.
- [109] J. Y. Lettvin (1995). oral lecture. In *Brain Processes, Theories, and Models. An International Conference in Honor of W.S. McCulloch 25 Years after his Death*.
- [110] A. G. Leventhal (1979). Retinal ganglion cells in the old-world monkey: morphology and central connections. *Science*, **213**, 1139–1142.
- [111] X. Li and R. Madhavan (1986). Correlation measures for corner detection. In *Conf. Computer Vision Pattern Recognition*, pp. 643–646.
- [112] T. P. Lindeberg (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 234–254.
- [113] T. P. Lindeberg (1992). Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. Technical report, Royal Institute of Technology, Stockholm University, Stockholm, Sweden.
- [114] A. Lingas and A. Proskurowski (1989). On parallel complexity of the subgraph homeomorphism and the subgraph isomorphism problem for classes of planar graphs. *Theoretical Computer Science*, **68**, 155–173.
- [115] M. M. Lipschutz (1969). *Differential Geometry*. McGraw-Hill, New York.
- [116] S. T. Liu and W. H. Tsai (1990). Moment preserving corner detection. *Pattern Recognition*, **23**, 441–460.
- [117] M. S. Livingstone and D. H. Hubel (1984). Anatomy and physiology of a color system in the primate visual cortex. *J. Neurosci.*, **4**, 309–356.

- [118] M. S. Livingstone and D. H. Hubel (1987). Connections between layer 4b of area 17 and the thick cytochrome oxidase stripes of area 18 in the squirrel monkey. *Journal of Neuroscience*, **7**, 3371–3377.
- [119] Margaret Livingstone and David Hubel (1988). Segregation of form, color, movement, and depth: Anatomy, physiology, and perception. *Science*, **240**, 740–749.
- [120] T. Lourens (1995). A model of spatial filtering by visual cortical simple cells. In *Brain Processes, Theories, and Models. An International Conference in Honor of W.S. McCulloch 25 Years after his Death*, edited by R Moreno-Díaz and J. Mira-Mira, pp. 391–400, Las Palmas de Gran Canaria, Spain. MIT press, Cambridge USA.
- [121] T. Lourens (1995). Modeling retinal high and low contrast sensitivity filters. In *Proceedings of the International Workshop on Artificial Neural Networks, IWANN '95*, edited by J. Mira and F. Sandoval, *Lecture Notes in Computer Science*, **930**, 61–68, Torremolinos (Málaga), Spain. Springer-Verlag Berlin Heidelberg.
- [122] Tino Lourens and Rolf P. Würtz (1998). Object recognition by matching symbolic edge graphs. In *Proceedings of the Third Asian Conference on Computer Vision*, Hong Kong. In press.
- [123] L. Maffei and A. Fiorentini (1973). The visual cortex as a spatial frequency analyser. *Vision Res.*, **13**, 1255–1267.
- [124] H. A. Mallot (1985). An overall description of retinotopic mapping in the cat's visual cortex areas 17, 18, and 19. *Biological Cybernetics*, **52**, 45–51.
- [125] H. A. Mallot (1987). Point images, receptive fields, and retinotopic mapping. *Trends in Neurosciences*, **10**, 310–311.
- [126] Hanspeter A. Mallot and Gerd-Jürgen Giefling (1990). Retinal sampling grids and space-variant image compression. In *Parallel Processing in Neural Systems and Computers*, edited by R. Eckmiller, G. Hartmann, and G. Hauske, pp. 125–132. Elsevier Science Publishers B.V.
- [127] Hanspeter A. Mallot, Werner von Seelen, and Fotios Giannakopoulos (1990). Neural mapping and space-variant image processing. *Neural Networks*, **3**, 245–263.
- [128] W. B. Marks, W. H. Dobelle, and E. F. MacNichol (1964). . *Science*, **143**, 1181–1183.
- [129] D. C. Marr and E. Hildreth (1980). Theory of edge detection. *Proc. Roy. Soc. London*, **B 207**, 187–217.
- [130] S. Marčelja (1980). Mathematical description of the responses of simple cortical cells. *J. Opt. Soc. Amer.*, **70**, 1297–1300.
- [131] J. A. McHugh (1990). *Algorithmic Graph Theory*. Prentice-Hall International, Inc.
- [132] G. Medioni and Y. Yasumoto (1986). Corner detection and curve representation using cubic b-spline. In *IEEE International Conference on Robot. Autom.*, pp. 764–769, San Francisco.
- [133] R. Mehrotra, S. Nichani, and N. Ranganathan (1990). Corner detection. *Pattern Recognition*, **23**, 1223–1233.
- [134] B. T. Messmer (1996). *Efficient Graph Matching Algorithms for Preprocessed Model Graphs*. PhD thesis, University of Bern.
- [135] B. T. Messmer and H. Bunke (1993). A network based approach to exact and inexact graph matching. Technical report, Institut für Informatik und angewandte Mathematik, Universität Bern.
- [136] M. C. Morrone and D. C. Burr (1988). Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London*, **235**, 335–354.

- [137] J. A. Movshon, I. D. Thompson, and D. J. Tolhurst (1978). Receptive field organization of complex cells in the cat's striate cortex. *Journal of Physiology*, **283**, 53–77.
- [138] S. H. Myaeng and A. Lopez-Lopez (1992). Conceptual graph matching: a flexible algorithm and experiments. *Journal of Experimental and Theoretical Artificial Intelligence*, **4**, 107–126.
- [139] H. H. Nagel (1987). On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*, **33**, 299–324.
- [140] I. Newton (1704). *Opticks: Or, a treatise of the reflexions, refractions, inflexions, and colours of light*. S. Smith and B. Walford, London.
- [141] John G. Nicholls, A. Robert Martin, and Bruce G. Wallace (1992). *From Neuron to Brain -A Cellular Molecular Approach to the Function of the Nervous System*. Sinauer Associates, Inc., third edition.
- [142] N. J. Nilsson (1980). *Principles of Artificial Intelligence*. Tioga, Palo Alto.
- [143] J. A. Noble (1988). Finding corners. *Image Vis. Comput.*, **6**, 121–128.
- [144] S. Peleg (1980). A new probabilistic relaxation scheme. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 362–369.
- [145] A. P. Pentland (1982). *Visual inference of shape: Computation from local features*. PhD thesis, Department of Psychology, Massachusetts Institute of Technology, Cambridge, MA.
- [146] P. Perona (1992). Steerable-scalable kernels for edge detection and junction analysis. In *Proc. Second European Conf. Computer Vision*, edited by G. Sandini, LNCS, **588**, 3–18. Springer-Verlag.
- [147] P. Perona (1995). Deformable kernels for early vision. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, **17**, 488–499.
- [148] V. H. Perry, R. Oehler, and A. Cowey (1984). Retinal ganglion cells that project to the dorsal lateral geniculate nucleus in the macaque monkey. *Neuroscience*, **12**, 1101–1123.
- [149] E. Peterhans, R. van der Zwan, and R. Baumann (1995). Sensitivity to stereoscopic depth of end-stopped cells in monkey visual cortex. In *European Conference on Visual Perception, ECVP '95 (Physiology)*, p. 6, Tübingen.
- [150] E. Peterhans and R. von der Heydt (1991). Elements of form perception in the monkey prestriate cortex. In *Representation of vision. Trends and tacit assumptions in vision research*, edited by A. Gorea, Y. Frégnac, Z. Kapoula, and J. Findlay, pp. 111–124. Cambridge University Press, Cambridge.
- [151] N. Petkov, P. Kruizinga, and T. Lourens (1993). Orientation competition in cortical filters -an application to face recognition. In *Proceedings of Computing Science in The Netherlands, CSN '93*, edited by H.A. Wijshoff, pp. 285–296, Utrecht, The Netherlands.
- [152] D. A. Pollen and S. F. Ronner (1981). Phase relationships between adjacent simple cells in the visual cortex. *Science*, **212**, 1409–1411.
- [153] D. A. Pollen and S. F. Ronner (1982). Spatial computation performed by simple and complex cells in the visual cortex of the cat. *Vision Res.*, **22**, 101–118.
- [154] J. Prewitt (1970). Object enhancement and extraction. In *Picture Processing and Psychopictorics*, edited by B. Lipkin and A. Rosenfeld, pp. 75–149. Academic Press, New York.
- [155] A. Rangarajan and E. Mjolsness (1994). A lagrangian relaxation network for graph matching. In *Proceedings of the International Conference on Neural Networks*, **7**, 4629–4634.

- [156] K. Rangarajan, M. Shah, and D. V. Brackle (1989). Optimal corner detection. *Comput. Vision Graphics Image Process.*, **48**, 230–245.
- [157] A. Rattarangsi and R.T. Chin (1992). corners of planar curves corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 430–449.
- [158] K. R. Ray and K. S. Ray (1992). Detection of significant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, **13**, 443–452.
- [159] S. W. Reyner (1977). An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, **6**, 730–732.
- [160] L. G. Roberts (1965). Machine perception of three dimensional solids. In *Optical and Electro-optical Information Processing*, edited by J. T. Tippet, pp. 159–197. MIT Press, Cambridge MA.
- [161] R. W. Rodieck (1965). Quantitative analysis of cat retinal ganglion cell response to visual system. *Vision Research*, **5**, 583–601.
- [162] K. Rohr (1990). über die modellierung und identification charakteristischer grauwertverläufe in realweltbildern. In *12. DAGM-Symposium Mustererkennung*, edited by R. E. Großkopf, **254**, 217–224, Oberkochen-Aalen. Informatik-Fachberichte, Springer-Verlag Berlin Heidelberg.
- [163] K. Rohr (1992). Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, **9**, 213–230.
- [164] A. Rosenfeld, R. Hummel, and S. Zucker (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, **6**, 420–433.
- [165] Paul L. Rosin (1996). Augmenting corner descriptors. *Graphical Models and Image Processing*, **58**, 286–294.
- [166] W. Rutkowski (1977). A comparison of corner-detection techniques for chain-coded curves. Technical Report 623, University of Maryland.
- [167] A. Sanfeliu and K. S. Fu (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, **13**, 353–363.
- [168] E. L. Schwartz (1980). Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, **20**, 645–669.
- [169] U. Seeger and R. Seeger (1994). Fast corner detection in gray-level images. *Pattern Recognition Letters*, **15**, 669–675.
- [170] L. G. Shapiro and R. M. Haralick (1981). Structural descriptions and inexact matching. *IEEE Transactions on pattern analysis and machine intelligence*, **3**, 504–519.
- [171] Robert Shapley and V. Hugh Perry (1986). Cat and monkey retinal ganglion cells and their visual functional roles. *Trends in Neuroscience (TINS)*, **9**, 229–235.
- [172] A. Singh and M. Shneier (1989). Grey level corner detection: A generalization and a robust real time implementation. *Computer Vision, Graphics, and Image Processing*, **51**, 54–69.
- [173] V. C. Smith and J. Pokorny (1986). In *Handbook of perception and human performance, Sensory processes and perception*, edited by K. R. Boff, L. Kaufman, and J. P. Thomas, chapter Colorimetry and color discrimination, **1**. Wiley, New York.
- [174] I. E. Sobel (1970). *Camera Models and Machine Perception*. PhD thesis, Electrical Engineering Department, Stanford University, Stanford, CA.

- [175] K. Sohn, W.E. Alexander, J.H. Kim, and W.E. Snyder (1994). A constrained regularization approach in robust corner detection. *IEEE Transactions on Systems, Man, and Cybernetics*, **24**, 820–82.
- [176] David C. Somers, Sacha B. Nelson, and Mriganka Sur (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience*, **15**, 5448–5465.
- [177] J. Stone (1965). A quantitative analysis of the distribution of ganglion cells in the cat's retina. *Journal of Comparative Neurology*, **124**, 337–352.
- [178] D. G. Stork and H. R. Wilson (1988). Analysis of gabor function descriptions of visual receptive fields. *Investigative Ophthalmology & Visual Science*, **29**, 398.
- [179] D. G. Stork and H. R. Wilson (1990). Do gabor functions provide appropriate descriptions of visual cortical receptive fields. *J. Opt. Soc. Am. A*, **7**, 1362–1373.
- [180] P. Suganthan, E. Teoh, and D. Mital (1995). Pattern recognition by graph matching using the potts mft neural networks. *Pattern Recognition*, **28**, 997–1009.
- [181] N. V. Swindale (1996). The development of topography in the visual cortex: a review of models. *Network: Computation in Neural Systems*, **7**, 161–247.
- [182] B. M. ter Haar Romeny, L. J. M. Florack, J. J. Koenderink, and M. A. Viergever (1991). *Scale Space: Its Natural Operators and Differential Invariants, Second Quingennial Review 1986-1991*. Dutch Society for Pattern Recognition and Image Processing.
- [183] Lisa G. Thorell, Russell L. DeValois, and Duane G. Albrecht (1984). Spatial mapping of monkey v1 cells with pure color and luminance stimuli. *Vision Research*, **23**, 751–769.
- [184] J. Ton and A. Jain (1989). Registering landsat images by point matching. *IEEE Transactions on Geoscience and Remote Sensing*, **27**, 642–651.
- [185] R. B. Tootell, E. Switkes, M. S. Silverman, and S. L. Hamilton (1988). Functional anatomy of macaque striate cortex. retinoptic organization. *The Journal of Neuroscience*, **8**, 1500–1593.
- [186] V. Torre and T. A. Poggio (1986). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 147–163.
- [187] W. H. Tsai and K. S. Fu (1979). Error-correcting isomorphisms of attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 757–768.
- [188] J. R. Ullman (1976). An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, **23**, 31–42.
- [189] R. L. De Valois (1973). In *Handbook of Sensory Physiology*, edited by R. Jung, chapter 3, Central mechanisms of color vision, **713**, 209–253. Springer-Verlag, Berlin.
- [190] R. van der Zwan, R. Baumann, and E. Peterhans (1995). End-stopped cells in the visual cortex of the alert monkey. In *European Conference on Visual Perception, ECVP '95 (Animal electrophysiology)*, p. 43, Tübingen.
- [191] J. van Leeuwen (1990). In *Handbook of Theoretical Computer Science*, edited by J. van Leeuwen, chapter 10, Graph Algorithms, pp. 527–631. Elsevier Science Publishers.
- [192] H. Wang and M. Brady (1994). A practical solution to corner detection. In *International Conference on Image Processing*, **I**, 919–923.
- [193] L. Weitzel, K. Kopecz, C. Spengler, R. Eckhorn, and H. J. Reitboeck (1997). Contour segmentation with recurrent neural networks of pulse-coding neurons. In *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns*, edited by G. Sommer and J. J. Koenderink. In press.

- [194] H. R. Wilson (1989). Psychophysical models of spatial vision and hyperacuity. In *Spatial form vision*, edited by D. Regan, London. MacMillan.
- [195] H. R. Wilson, D. Levi, L. Maffei, J. Rovamo, and R. DeValois (1989). In *Visual perception -The neurophysiological foundations*, edited by L. Spillmann and J. S. Werner, chapter 10, The perception of form -Retina to striate cortex. Academic Press.
- [196] E. K. Wong (1990). Model matching in robot vision by subgraph isomorphism. In *Syntactic and Structural Pattern Recognition -Theory and Applications*, edited by H. Bunke and A. Sanfeliu, pp. 381–414. World Scientific.
- [197] Rolf P. Würtz (1995). *Multilayer dynamic link networks for establishing image point correspondences and visual object recognition*, volume 41 of *Reihe Physik*. Verlag Harri Deutsch, Thun, Frankfurt am Main.
- [198] Rolf P. Würtz and Tino Lourens (1997). Corner detection in color images by multiscale combination of end-stopped cortical cells. In *Proceedings of the International Conference on Artificial Neural Networks, ICANN'97*, edited by Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud, *Lecture Notes in Computer Science*, **1327**, 901–906, Berlin Heidelberg New York. Springer Verlag.
- [199] T. Young (1802). The bakerian lecture. on the theory of light and colours. *Phil. Trans. R. Soc., London*, pp. 12–48.
- [200] S. S. Yu and W. H. Tsai (1992). Relaxation by the hopfield neural network. *Pattern Recognition*, **25**, 197–208.
- [201] S. Zeki and S. Shipp (1988). The functional logic of cortical connections. *Nature*, **335**, 311–317.
- [202] S. M. Zeki (1973). Colour coding in rhesus monkey re striate cortex. *Brain Res.*, **53**, 422–427.
- [203] S. M. Zeki (1977). Colour coding in the superior temporal sulcus of rhesus monkey visual cortex. *Proc. R. Soc. London*, **B 197**, 195–223.
- [204] Semir Zeki (1993). *A Vision of the Brain*. Blackwell science Ltd., London.
- [205] O. A. Zuniga and R. M. Haralick (1983). Corner detection using the facet model. In *Proc. Conference on Pattern Recognition and Image Processing*, pp. 30–37.

Index

A

aperture 43
 area 17 *see* primary visual cortex
 area 18 *see* visual area V2
 axon 9

B

blob 12, 110
 building block structure 15

C

C_{avg} 98
 cell
 amacrine 10
 bipolar 10
 center-only 113
 center-surround 15
 color opponent 108
 center-only 109
 center-surround 109, 112, 113
 color sensitive and orientation selective 117
 complex 16
 double opponent 109
 black-white 110
 orientation selective 117
 red-green 110
 simple 117
 yellow-blue 110
 end-stopped 16, 44
 double 44
 single 44
 ganglion 10
 horizontal 10
 hyper-complex *see* end-stopped
 M 11, 15
 P 11, 15

 receptive field 14
 simple 15
 W 20
 X 20
 Y 20
 $CO_{\sigma}^{e,i}$ 113
 $CS_{\sigma}^{e,i}$ 113
 cerebral cortex 11, 12
 chain code 61
 chromaticity 111
 color 108
 constancy 109
 contrast 110
 $C_{\sigma,\theta}$ 27–29
 cone 10
 conflicting corner 77
 corner 43
 rounded 45
 corner attribute
 angle 43
 color 44, 65
 contrast 65
 corner orientation 65
 edge profile 65
 edge shape 65
 edge texture 65
 junction type 44, 65
 orientation 65
 position 65
 sharpness 44, 65
 size 44, 65
 subtended angle 65
 correspondence problem 140
 cost function 125
 cut-off radius 142

D

- dendrites 9
- dendritic tree 18
 - size 18
- difference of Gaussians . . *see* Mexican-hat
- distance 46
- DOG *see* Mexican-hat
- $DO_{\sigma}^{e,i}$ 113
- $DO_{\sigma}^{e,i}C$ 117
- $DO_{\sigma}^{e,i}E$ 119
- $DO_{\sigma,\theta}^{e,i}S_{\sigma,\theta}^a$ 117
- $DO_{\sigma,\theta}^{e,i}S_{\sigma,\theta}^s$ 117
- drift 59

E

- early vision 3
- E_{avg} 90
- edge 96
 - detection 96
 - enhancement 96
 - extraction 96
- E_{σ}^d 60
- $E_{\sigma,\theta}^d$ 52
- $\hat{E}_{\sigma,\theta}^d$ 46
- E_{σ} 60
- E_{σ}^s 60
- $E_{\sigma,\theta}^s$ 52
- $\hat{E}_{\sigma,\theta}^s$ 46
- E_{Σ} 88
- exact location 46

F

- false response 46
- Fourier transform 30

G

- Gabor function 22
- gain factor 53
- Gaussian curvature 64
- Gaussian function 141–142
- G_{σ} 113
- Gc_{σ} 145
- graph 96
 - attribute 96

- angle 125
- coordinate 96
- length 125
- ratio 129
- matching 123

- G_{σ} 141
- Gs_{σ} *see* 145, 147

I

- image graph 123
- $\Im G_{\sigma,\theta}$ 23
- inter-blob 12
- I_{σ}^r 51
- isomorphism
 - graph 123
 - subgraph 123
 - error-tolerant 123
- I_{σ}^l 51

J

- junction 43

L

- Laplacian of Gaussians . . *see* Mexican-hat
- lateral geniculate nucleus 9, 11–12
- light 108
- L_{θ}^R 36

M

- magnification 32
- marked corner *see* $MC_{T,\omega}$
- matching 1
 - algorithm 128
 - cost function 125
 - data structure 127
 - memory requirements 129
 - speed 129
- Mcs_{σ} 145
- $MC_{T,\omega}$ 79
- Mexican-hat 20, 142–148
 - center radius 142
 - difference of Gaussians 144–148
 - minimum 148
 - normalization 143–144
- middle temporal area . . *see* visual area V5

- missing response 71
model graph 123
 M_σ 142
 \bar{M}_σ 21
 $M_{\sigma,m}$ 145
MT *see* visual area V5
- N**
NP-complete 124
 N_σ 22
- O**
optic chiasm 9
optic nerve 9
optic tract 9
 O_σ 28
- P**
pathway
 M 3, 12
 P 3, 12
PCF $_{T,\omega}$ 79
potential corner feature *see* PCF $_{T,\omega}$
primary visual cortex 9, 12–13
 P_σ 22
- R**
 $\Re G_{\sigma,\theta}$ 23
receptive field 14
 center 15
 profile 14
 anti-symmetrical simple . *see* $\Im G_{\sigma,\theta}$
 center-surround 15, 20–22
 complex 27–28
 simple 22–26
 symmetrical simple *see* $\Re G_{\sigma,\theta}$
 radius 14
 size 14
reliable corner 71
response
 center-only *see* G_σ
 center-surround *see* M_σ
 color opponent
 center-only *see* $CO_{\sigma}^{e,i}$
 center-surround *see* $CS_{\sigma}^{e,i}$
 surround-center *see* $SC_{\sigma}^{e,i}$
 complex *see* $C_{\sigma,\theta}$, *see* C_{avg}
 double opponent *see* $DO_{\sigma}^{e,i}$
 complex *see* $DO_{\sigma}^{e,i}C$
 end-stopped *see* $DO_{\sigma}^{e,i}E$
 simple (asymm.) *see* $DO_{\sigma}^{e,i}S_{\sigma,\theta}^a$
 simple (symm.) *see* $DO_{\sigma}^{e,i}S_{\sigma,\theta}^s$
 end-stopped *see* E_σ
 double *see* $E_{\sigma,\theta}^d$, *see* $E_{\sigma,\theta}^d$, *see* E_{σ}^d
 single *see* $E_{\sigma,\theta}^s$, *see* $E_{\sigma,\theta}^s$, *see* E_{σ}^s
M *see* N_σ
P *see* P_σ
simple
 anti-symmetrical *see* $S_{\sigma,\theta}^a$
 symmetrical *see* $S_{\sigma,\theta}^s$
retina 9–11
retinex theory 108
rod 10
rotation 55–57
- S**
scale 4, 29
scale space 29
 linear 29
 non-linear 31
segment 96
 $S_{\sigma,\theta}^a$ 26
 $S_{\sigma,\theta}^s$ 26
standard corner operator 81–84
 DET 62
 Diff 65
 Kitchen Rosenfeld 64
 Plessey 64
striate cortex *see* primary visual cortex
 $SC_{\sigma}^{e,i}$ 113
- T**
trichromacy 108
tristimulus 110
- V**
vertex 96
visual area V1 *see* primary visual cortex
visual area V2 13

visual area V3	13
visual area V4	13
visual area V5	13

Samenvatting

Dit proefschrift beschrijft onderzoek aan objectherkenning door kunstmatige visuele systemen. Op het eerste gezicht lijkt het construeren van een dergelijk systeem een triviaal probleem: immers, deze vaardigheid beheersen wij mensen zodanig goed dat ze als vanzelfsprekend wordt gezien. Toch is het menselijke visuele systeem superieur als het wordt vergeleken met de huidige kunstmatige visuele systemen, dit ondanks jarenlang onderzoek door vele wetenschappers.

Bij de meeste kunstmatige visuele systemen worden initieel beelden opgenomen met behulp van een camera (soms ook wel eens het elektronische oog genoemd). Zo'n beeld bestaat uit een twee-dimensionaal rooster van beeldpunten (pixels), waarbij elk beeldpunt een bepaalde kleur bevat die wordt gerepresenteerd door een getal. Daarentegen bevat het biologische oog een rooster van lichtgevoelige cellen, het netvlies. Elke cel zet lichtinformatie om in elektrische informatie, hetgeen in een getal uit te drukken is, en kan in dit opzicht worden vergeleken met een beeldpunt.

Het is duidelijk dat de moeilijkheid bij een kunstmatig visueel systeem niet zozeer bij het creëren van het beeld zit maar veel meer bij het interpreteren van zo'n beeld. Immers een voorwerp wordt onder verschillende kijkhoeken, belichtingen of gedeeltelijk verborgen achter een ander voorwerp toch nog steeds als hetzelfde voorwerp herkend. De mens is in staat om zeer snel en met grote nauwkeurigheid een beeld te interpreteren. Het ligt daarom voor de hand om biologische visuele systemen nader te analyseren en de daarbij ontdekte principes te gebruiken binnen de kunstmatige visuele systemen.

Het menselijke visuele systeem heeft begrensde mogelijkheden met betrekking tot het opslaan van objecten. Objecten zullen daarom op een compacte en snel toegankelijke manier moeten worden opgeslagen. Hoe deze representatie er uit ziet is voor een groot deel onbekend. Het is aannemelijk dat, bij het zien van een voorwerp, er bepaalde kenmerken uit dit voorwerp worden gehaald en dat deze kenmerken worden vergeleken met de kenmerken van opgeslagen (geleerde) voorwerpen. Het vergelijken van deze kenmerken noemen we "matching" en het vinden van vergelijkbare kenmerken "herkenning".

Het herkennen van objecten binnen visuele systemen kan globaal worden opgedeeld in een aantal fasen:

1. het construeren van een beeld,
2. het benadrukken van kenmerken (bijvoorbeeld intensiteitsranden, hoekpunten of een kleur) in dit beeld,
3. het extraheren van deze kenmerken uit het beeld (met de benadrukte kenmerken)

4. en het detecteren van verschillende combinaties van kenmerken die verschillende objecten representeren.

Het in dit proefschrift ontwikkelde model wordt opgebouwd volgens bovenstaande punten.

In fase 2 maken we gebruik van de voorwaartse stroom van informatie in het biologische visuele systeem. We simuleren de respons van cellen in het biologische systeem op zodanige manier dat het resultaat van een groep cellen de invoer vormt voor een volgende groep cellen en gebruiken hierbij de “building block” architectuur die door Hubel en Wiesel werd geïntroduceerd. Het biologische visuele systeem is dermate complex dat slechts de eerste fase en een deel van de tweede fase bekend zijn. Het eerste deel van het model is dan ook beperkt tot datgene wat bekend is door middel van metingen van neurofysiologen.

In de laatste 40 jaar is er veel onderzoek gedaan naar de visuele systemen van katten en apen, met name aan het oog en de visuele cortex. Dit is goed te verklaren doordat zowel het oog als de visuele cortex zich bijna aan de oppervlakte bevinden en daardoor gemakkelijk van buitenaf te bereiken zijn. De visuele informatie gaat via het oog naar de LGN (nucleus corporis geniculati lateralis) om zijn weg te vervolgen naar de visuele cortex, die aan de achterkant van het hoofd is gelocaliseerd. Daarna gaat de informatiestroom naar dieper gelegen lagen in de hersenen.

Een zeker type cellen die in het netvlies wordt gevonden reageert als een kleine lichtvlek als stimulus wordt aangeboden (vgl. een zwart beeld met een witte cirkel). De respons neemt toe naarmate de vlek groter wordt totdat een zekere omvang bereikt is, daarna neemt de respons af. Deze cellen staan dan ook bekend als cellen met “center-surround” eigenschappen.

In de visuele cortex vonden neurofysiologen groepen cellen die zeer goed reageren op randen met een bepaalde oriëntatie, terwijl andere groepen cellen alleen reageren op hoekpunten. Hubel en Wiesel noemden de verschillende, in complexiteit toenemende, groepen cellen “simpel”, “complex” en “hypercomplex”. De building block architectuur gaat uit van de voorwaartse stroom van informatie. Binnen deze architectuur leveren een aantal center-surround cellen op een rechte lijn met een zekere oriëntatie de invoer voor de simpele (randgevoelige) cellen. Een combinatie van deze cellen levert de invoer voor de complexe (randgevoelige) cellen die vervolgens op hun beurt weer de invoer leveren voor de hypercomplexe (hoekpuntgevoelige) cellen. In het model zijn complexe en hypercomplexe cellen volgens de building block architectuur opgenomen. Met deze architectuur leggen we ons de beperking op dat de informatiestroom geen terugkoppeling bevat. Uit onderzoek is gebleken dat er wel degelijk een terugwaartse stroom van informatie plaatsvindt, maar dat is binnen dit proefschrift buiten beschouwing gelaten. De resultaten van de gemodelleerde hoekpuntgevoelige cellen zijn vergeleken met een aantal hoekpunt-detectoren die gebruikelijk zijn in de beeldbewerking. Het blijkt dat de gemodelleerde hoekpuntgevoelige cellen betere resultaten geven dan deze hoekpunt-detectoren, zeker als ze op meerdere schalen wordt toegepast.

Alle hierboven genoemde celtypen zijn achromatisch, d.w.z. niet gevoelig voor kleur. Uit neurofysiologisch onderzoek blijkt dat het veel moeilijker is om kleurgevoelige cellen te stimuleren, waarschijnlijk doordat ze een veel complexere functionaliteit hebben. Wel blijkt dat kleurgevoelige cellen goed reageren op een bepaalde kleur terwijl in dezelfde cel de respons wordt onderdrukt als een andere kleur als stimulus wordt aangeboden. Deze eigenschap wordt kleuropponentie genoemd. Er zijn twee soorten kleuropponentcellen, de ene heeft rode

en groene, en de andere blauwe en gele kleuroponenten. Het blijkt dat de combinatie van deze twee soorten tezamen met de achromatische cellen voldoende is om elke kleur te representeren.

Binnen de kleuroponentcellen zijn diverse soorten te onderscheiden. Eén van deze soorten is de center-surround kleuroponentcel. Daarnaast zijn er nog cellen die reageren op een kleur in het centrum, maar buiten het centrum drukt diezelfde kleur de respons. Deze cel heeft nog een andere kleur die de respons drukt in het centrum maar weer sterk reageert buiten dit centrum. Deze center-surround cellen staan bekend als dubbel kleuroponent cellen. Naast deze cellen zijn er ook nog kleuroponent cellen gevonden die oriëntatiegevoelig zijn. Over cellen die kleurgevoelig zijn en reageren op hoekpunten is tot nu toe nog niets bekend. De functionaliteit van zulke cellen kan aannemelijk worden gemaakt door de buiding block architectuur toe te passen op kleurgevoelige cellen. Alle genoemde bekende functionaliteiten van kleurgevoelige cellen tezamen met de hypothetische kleur- én hoekpuntgevoelige cellen zijn opgenomen in het model. Al deze gemodelleerde kleurgevoelige cellen worden tot de tweede fase gerekend.

Overigens worden niet alle groepen cellen gebruikt in het model. Een voorbeeld hiervan zijn de cellen die gevoelig zijn voor beweging. Deze zijn niet opgenomen in het model omdat alleen met statische beelden wordt gewerkt.

In dieper liggende lagen in de hersenen wordt de informatie afkomstig van bovengenoemde cellen verwerkt. Een object wordt gerepresenteerd als een verzameling van kenmerken die op een bepaalde manier aan elkaar gerelateerd zijn. Hoe deze representaties er in een biologisch systeem uitzien is onbekend. Omdat vele cellen in de visuele cortex gevoelig zijn voor randen en hoeken is het waarschijnlijk dat deze kenmerken een rol zullen spelen in de representatie van een object. Binnen de wiskunde en informatica staat een zekere representatie van hoeken en randen (kanten) bekend als een graaf. Deze representatie zal dan ook in het model worden gebruikt, mede omdat de nodige eigenschappen en algoritmen over grafen bekend zijn.

Aleer deze representatie kan worden gebruikt dienen eerst de in het beeld benadrukte aspecten (zoals randen en hoekpunten) te worden gelocaliseerd en geëxtraheerd. In het model is een mechanisme opgenomen dat, uitgaande van de gevonden hoekpunten, randen probeert te extraheren. Op deze manier wordt een graaf gecreëerd.

Nadat uit een kleurenbeeld een verzameling kenmerken is gehaald die wordt gerepresenteerd als een graaf, kunnen bestaande objecten (ook als graaf gerepresenteerd) met delen van de graaf worden vergeleken. Deze techniek staat bekend als het matchen van grafen. Het vinden van objecten in deze representatie is zeer rekenintensief (NP-volledig), en daarom zijn er kenmerken aan de hoekpunten toegevoegd die de snelheid van het matchen versnellen. Deze kenmerken zijn zo gekozen dat ze de translatie, rotatie en schaal invariantie van de graaf behouden. Het matching proces wordt door deze kenmerken zodanig versneld dat ze praktisch toepasbaar en bovendien ook nog fouttolerant is.