Linköping Studies in Science and Technology
Thesis No. 877

Multiscale Curvature Detection in Computer Vision

Björn Johansson

INSTITUTE OF TECHNOLOGY
LINKÖPINGS UNIVERSITET

LIU-TEK-LIC-2001:14

Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden

Linköping , March 2001

Multiscale Curvature Detection in Computer Vision

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping
Sweden

ISBN 91-7219-999-7 ISSN 0280-7971

Abstract

This thesis presents a new method for detection of complex curvatures such as corners, circles, and star patterns. The method is based on a second degree local polynomial model applied to a local orientation description in double angle representation. The theory of rotational symmetries is used to compute curvature responses from the parameters of the polynomial model. The responses are made more selective using a scheme of inhibition between different symmetry models. These symmetries can serve as feature points at a high abstraction level for use in hierarchical matching structures for 3D estimation, object recognition, image database search, etc.

A very efficient approximative algorithm for single and multiscale polynomial expansion is developed, which is used for detection of the complex curvatures in one or several scales. The algorithm is based on the simple observation that polynomial functions multiplied with a Gaussian function can be described in terms of partial derivatives of the Gaussian. The approximative polynomial expansion algorithm is evaluated in an experiment to estimate local orientation on 3D data, and the performance is comparable to previously tested algorithms which are more computationally expensive.

The curvature algorithm is demonstrated on natural images and in an object recognition experiment. Phase histograms based on the curvature features are developed and shown to be useful as an alternative compact image representation.

The importance of curvature is furthermore motivated by reviewing examples from biological and perceptual studies. The usefulness of local orientation information to detect curvature is also motivated by an experiment about learning a corner detector.

Acknowledgments

The front page of this thesis should really be covered with the names of present and past people at the Computer Vision Laboratory. Their presence provided a creative and friendly athmosphere which was cruical for the work in this thesis. I would especially like to thank the following persons:

Professor Gösta Granlund, head of the research group and my supervisor, for introducing me to this interesting field of research, for proofreading, giving constructive criticism, and in general sharing his ideas.

Gunnar Farnebäck, for proofreading parts of the manuscript, discussing numerous mathematical issues, and for introducing me to the polynomial model which is the basis for much of the work in this thesis.

Per-Erik Forssén and Anders Moe, for proofreading parts of the manuscript and giving constructive comments.

Our secretary Catharina Holmgren, for proofreading the manuscript and helping me to improve my English skills. This was much appreciated and well needed.

Professor Hans Knutsson, for many fruitful discussions and an endless stream of ideas. Some of the work in this thesis are a direct result of these discussions.

Dr. Magnus Borga, for many discussions concerning learning algorithms.

Dr. Mats Andersson and Johan Wiklund, for sharing their experience on filter optimization. Johan also for helping me fight the never ending battle against the computers.

Dr. Klas Norberg, for taking time to discuss all sorts of things, ranging from image operators to pedagogical teaching issues.

Finally, I want to thank my leisure-friends for all spare time adventures - there is more to life than nine to five.

This work was sponsored by the Swedish Foundation for Strategic Research (SSF), within the Swedish strategic research initiative VISIT (VISual Information Technology).

Contents

Chapter 1

Introduction

1.1 Motivation

The work in this thesis has been carried out within the project “Contents based search in image and video databases”. This project is part of VISIT (VISual Information Technology), which is a national Swedish strategic research initiative within the field of Visual Information Technology, supported by the Swedish Foundation for Strategic Research (SSF). The goal of this particular project is to develop methods and tools for contents based image and video database search.

Imagery will be an essential type of information in the future, especially in the emerging IT-networks. Large image and video databases will be common and serve as key sources of information for private people in their everyday life, as well as for professionals in their work. The research field concerning Content Based Image Retrieval (CBIR) has therefore increased considerably during the last decade, see e.g. [Smeulders et al., 2001], [Johansson, 2000b], [Rui et al., 1999], [Marsicoi, 1997]. Almost every existing computer vision algorithm has been applied in CBIR. But still, todays CBIR-systems are not very capable of mimicking human retrieval and need to be combined with traditional textual search. Manual annotation of keywords to every image in a large database is however a tedious work. Since the annotator is only human, he is bound to forget useful keywords. Also, keywords cannot capture abstract concepts and feelings. The old saying “One picture is worth a thousand words” definitely still holds. Humans also tend to abstract query images for some conceptual information. We tend to associate objects in terms of our ability to interact with them. This phenomenon can also be traced in text-based systems where the categories often represent actions (or corresponding nouns). For example, *glasses* can look very different from each other but are still associated because we can perform a common action on them, namely *drink*. A truly useful system for general browsing has to be able to perform this association, but this is a very difficult task to accomplish in practice.

The basic idea and motivation for this thesis is that local, high level, and selective image features will play an important role in future CBIR systems. This kind of features have so far been sparsely applied in CBIR, but the attention around them has increased in the last few years. They should be more descriptive than most of the low level features applied today, e.g. lines and edges.

Large image databases can consist of thousands, or even millions of images. The need for efficient algorithms is therefore crucial. As an example, the company behind the text based Internet search engine AltaVista estimates that, in a CBIR system for the net, the computational time for each image should be at most around a second. A large part of the work in this thesis is therefore focused on efficient algorithms.

As systems grow more complex, it is also important from a practical perspective that algorithms are easy to understand and to implement. The feature detection strategy presented in this thesis takes a unified approach to single and multiscale detection of complex curvature such as corners, circles, and star patterns. The algorithm employs a polynomial model of a local orientation image to detect these features. The same model applied on gray-level images has been used for detection of edges and lines and estimation of local orientation, and for a number of other applications. The polynomial model therefore serves as a unified approach to these tasks. Furthermore, an efficient multiscale algorithm for computing the polynomial model is presented.

1.2 Contributions

The computer vision community consists of people from many different research disciplines, and the terminology varies to a great extent. Hence it is difficult to review related work and say what is really new and not. The list below contains the contributions that are likely to be new.

- The local polynomial expansion model in chapter 4 is not new, but a new efficient approximative algorithm is developed to estimate the model in several scales. The algorithm uses Gaussian filters and derivative filters to compute partial derivatives. This idea to compute partial derivatives is not new, but it has probably never been used before as a way to estimate polynomial models.

  The approximative algorithm has some relation to [Burt, 1988]. This reference was found very recently, and a comparison has not yet been made. The approach is very different from the one in this thesis, but the results may be similar.

- The idea in section 5.3.4 to detect curvature in several scales by using a local polynomial model based on a local orientation description, is also new. Polynomial models have been used before to detect curvature, but then applied on gray-level images directly. By using local orientation, it is easier to represent more complex curvatures. It should be noted that using local orientation to detect curvature is not a new idea, the review in chapter 5 of the rotational symmetry theory deals with this issue. The polynomial model approach should be seen as an efficient way to detect these symmetries.

Polynomial models have been used for a number of computer vision tasks, and this model serves as a unified framework to these tasks.

- The idea to use normalized inhibition in section 5.3.3 to make the rotational symmetry responses more selective is also new.

  The other approach, normalized convolution in section 5.3.2, has previously been used in a special case. The generalization is natural, but it has probably never been explored before.

  These two ideas have also been published in [Johansson and Granlund, 2000] and [Johansson et al., 2000].

- The idea of using canonical correlation to learn feature detectors in section 6.3 is not new, but it has previously only been used to learn local orientation detectors. In this section it is shown that the same idea can be used to learn a corner detector. In addition, it is shown that parameters from a polynomial model of the image representation can be used instead of the representation itself. This greatly reduces the amount of data and makes the learning easier.

  Section 6.3 is published in [Johansson et al., 2001].

- The idea to use phase histograms on rotational symmetry responses as an image representation, section 6.4.2, is also new.

- The back-projection idea in appendix A is a method to transform local orientation descriptions in double angle representation to corresponding gray-level patterns. Part of the idea is inspired from [Bigün, 1997], but otherwise assumed to be new.

1.3 Thesis outline

Figure 1.1 contains an overview of the thesis outline and in which order the sections should be read. The thesis is mainly divided into three parts: introduction, theory, and experiments.

Chapters 1 and 2 introduce and motivate the work in the thesis and also review some related work.

The theory and algorithm part is divided into three chapters. Chapter 3 contains a short description of the basic signal processing tools used in this thesis; normalized convolution and canonical correlation. These tools will be used in the remaining theory chapters and in the experiments. Chapter 4 reviews an algorithm for local polynomial expansion using normalized convolution and also describes a new approximative method, which more efficiently estimates the polynomial model in several scales. Chapter 5 reviews the rotational symmetry theory and describes a new algorithm to detect the symmetries using the local polynomial model. Chapter 6 contains experiments based on the theories and algorithms, except section 6.4.2 about phase histograms, which also may be viewed as 'theory'.

All experiments are gathered in chapter 6. Sections 6.1 and 6.2 evaluate the new algorithms developed in chapters 4 and 5 respectively. Section 6.4 uses the rotational symmetry detection algorithm in an object recognition experiment. Section 6.5 discusses some other possible applications for this algorithm. Section 6.3 contains an experiment on learning a detector for corner orientation. This experiment may seem a bit off track from the rest of the thesis, but it should be viewed as an additional attempt to motivate the use of rotational symmetries and local orientation as information for curvature detection.

Finally, chapter 7 summarizes the work and discusses some ideas for future research.

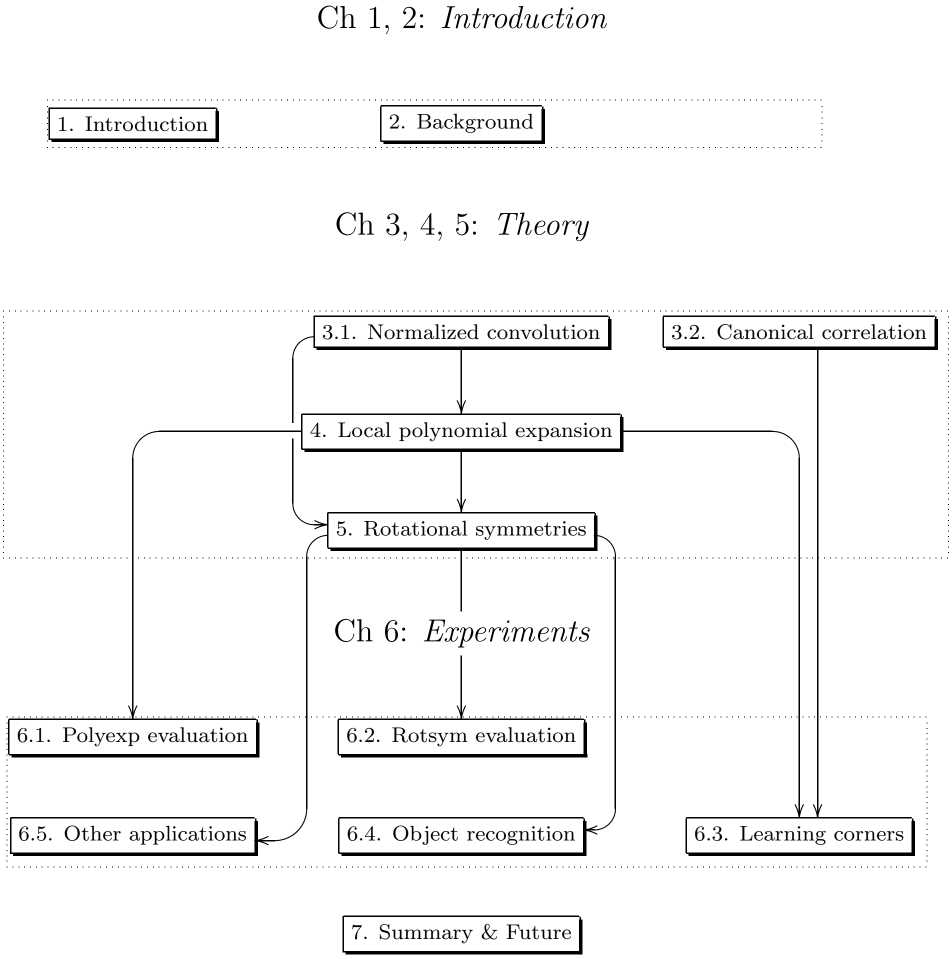


Figure 1.1: Thesis outline

1.4 Notations

Below follows a list of notations used in this thesis:

- Italic letters (e.g. I and z) denote real or complex functions or scalars. Lowercase letters in boldface (e.g. $\mathbf{f}$) denote vectors, and uppercase letters in boldface denote matrices (e.g. $\mathbf{P}$).
- Partial derivatives are sometimes denoted using subscripts, e.g. $f_x = \frac{\partial f}{\partial x}$, $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$, etc.
- Conjugate transpose for complex vectors and matrices is denoted by *. For real vectors and matrices transpose is also denoted as ^{T}.
- $\mathbf{a} \cdot \mathbf{b}$ denotes pointwise multiplication of the elements of the two vectors $\mathbf{a}$ and $\mathbf{b}$.
- $|z|$ denotes magnitude and $\angle z$ denotes argument, or phase, of the complex value z.
- $\hat{z}$ for complex values denotes normalized value, i.e. $\hat{z} = z/|z|$ and $\hat{\mathbf{a}}$ for a vector denotes unit length.
- Two scalar (inner) products are used in this thesis. One unweighted and one weighted:

  $$\begin{array}{rcl} \langle \mathbf{a},\ \mathbf{f} \rangle & = & \mathbf{a}^*\mathbf{b} \\ \langle \mathbf{a},\ \mathbf{b} \rangle_W & = & \mathbf{a}^*\mathbf{W}\mathbf{b} \end{array} \tag{1.1}$$

  where $\mathbf{W}$ is a positive semidefinite matrix.

Additional notations are introduced when needed.

Chapter 2

Background

This chapter reviews some work related to this thesis.

2.1 Biology

The use of local curvature for object recognition is still limited in the field of computer vision. Designing curvature detectors can be motivated by looking at biological systems. There are a number of studies and perceptual experiments that indicate the importance of curvature in biological vision. This section presents some of them. Unfortunately they give no clues to how to further use curvature information for recognition tasks. The figures in this section are copied from the cited articles.

Attneave ([Attneave, 1954]) views point features in an information theoretical perspective. Point features such as curvature points and corners contain more information than lines and edges because they cannot as easily be predicted from neighboring points. He also shows that objects can be recognized from simplified drawings using straight lines between high curvature points, see figure 2.1.

Oster ([Oster, 1970]): *Phosphenes* are subjective images which result from internal activity in the eye and the brain. They can arise spontaneously as moving specks of light, for example when you close your eyes or enter a dark room. Other patterns can be induced by pressing the eyeballs, star patterns can arise from a blow on the head (hence the expression 'seeing stars'), and still other patterns can be induced from chemical drugs or from electrical stimulation. These patterns are interesting because they must be related to the visual pathway, and the visual cortex. Curvature, circles and star patterns seem to be among the patterns that appear when electrical impulses are sent into the brain through electrodes placed on the head, see figure 2.2.

Oster also points out that scribbles from children during their early years are

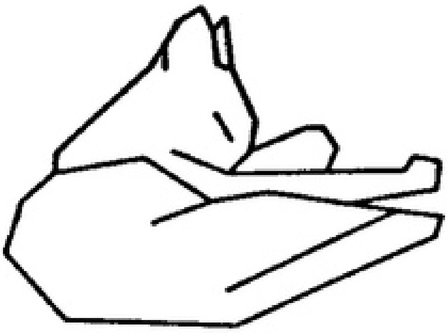


Figure 2.1: From [Attneave, 1954]. Perceptual experiment showing the importance of curvature. Quote: “Drawing made by abstracting 38 points of maximum curvature from the contours of a sleeping cat, and connecting these points appropriately with a straight edge.”

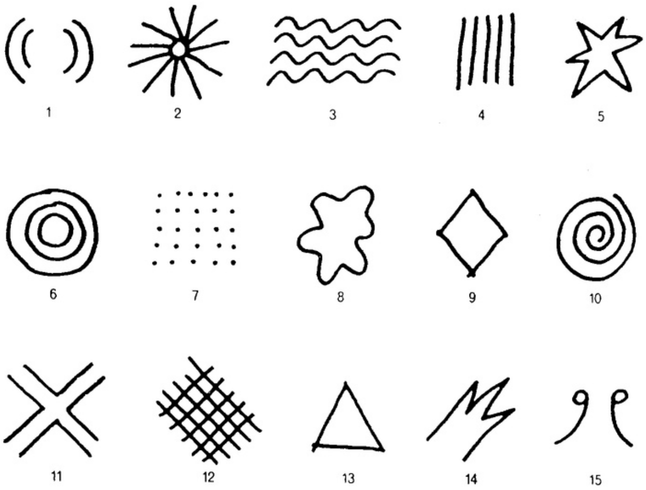


Figure 2.2: From [Oster, 1970]. Examples of phosphenes. Quote: “CLASSIFICATION of electrically induced phosphenes was undertaken by Max Knoll. On the basis of reports from more than 1,000 volunteers he grouped the phosphenes into 15 categories, each represented here by a typical example and numbered in accordance with its commonness. Certain forms are characteristic of each pulse frequency for each individual.”

similar to typical electrically induced phosphenes, as the ones in figure 2.2. As the child grows older the scribbles are combined to form more complex figures such as objects. The drawings are subsequently improved and finally relations between objects are included.

Blakemore and Over ([Blakemore and Over, 1974]) Cell *adaptation* means that cells that have been strongly activated become less responsive due to fatigue. During adaptation so called *after-effects* can occur which means that the interpretation of an event becomes biased to cell responses that have not been adapted. For example, if one looks at a yellow colored area for a about minute and then at a gray area it appears bluish, since blue is the complementary color to yellow (see [Atkinson et al., 1990]). Blakemore and Over showed that we can experience curvature after-effects. A person had looked fixedly on a curvature image for a minute, see figure 2.3, and was then asked to look at an image with a line and correct the orientation until it looked straight. If the line needed correction, the person experienced an after-effect. Blakemore and Over argued that this indicated the existence of curvature-selective cells. They also argued that the curvature cells use information from orientation-selective cells.

Biederman ([Biederman, 1987]), ([Biederman and Cooper, 1991]): An object can often be recognized by its contour alone, see the left column in figure 2.4. This means that there is a great deal of redundant information in a color or intensity image. Biederman showed that if we go one step further and only use high-curvature features, we can still recognize the objects in many cases, see the middle column in figure 2.4. The recognition task becomes much more difficult if we use other contour parts (right column).

Gallant et al ([Gallant et al., 1993]): Gallant et al studied the selectivity for polar (circle, spiral-, and star-patterns), hyperbolic (curvature patterns), and Cartesian (lines, edges, etc) image patterns in some cells in area V4 in the macaque visual cortex (also see Tanaka below). They found that many cells are more sensitive to polar and hyperbolic patterns than to Cartesian patterns. Many of the cells were tuned to one phase within the class, for example some cells were sensitive to star-patterns but not to circle patterns, some were sensitive to curvature around one direction but not to the other directions. Also, some cells were selective to more than one of the three classes of patterns but they still kept their tuning selectivity within the classes. It should also be mentioned that many of the cells were fairly invariant to the location of the pattern within the receptive field.

The results of this experiment are interesting because these patterns are the same as those found by the rotational symmetry detectors described in chapter 5.

Humphreys ([Humphreys et al., 1994]): Many conclusions about how the brain works can be made by observing persons with lesions in the brain. Humphreys studied the effects of lesions in the parietal lobe (Balints syndrom). These

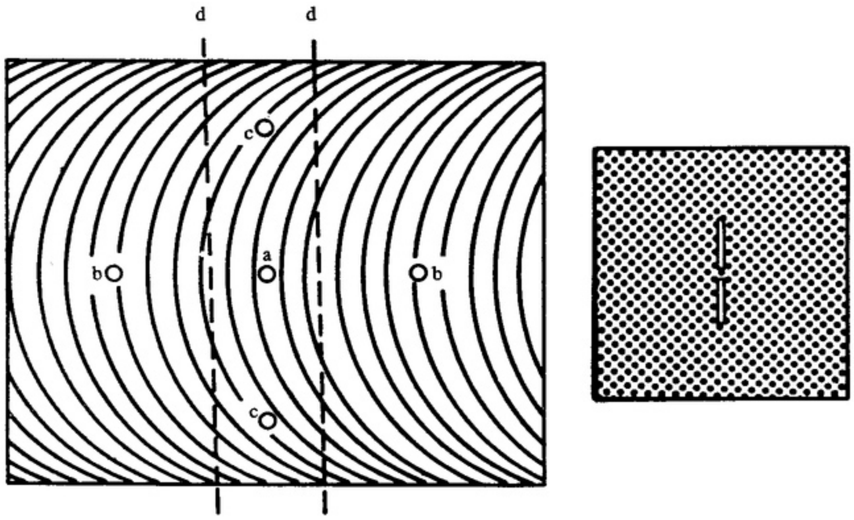


Figure 2.3: From [Blakemore and Over, 1974]. Perceptual experiment on curvature after-effects. Left: Inspection stimulus. Right: Test stimulus. Quote: "The subjects adapted by viewing the inspection stimulus and then, while fixating the dark center of the test stimulus, they adjusted the curvature of the line until it appeared straight. There were four inspection conditions: (i) steady fixation on a spot of light at a, (ii) smooth eye movements following the spot as it moved horizontally over a total excursion b-b, (iii) pursuit eye movements with a vertically moving fixation spot c-c, (iv) horizontal scanning from b-b but with the pattern blanked off at each side, beyond d, exposing only a central pattern of curves, 2-5 deg wide. Significant after-effects were generated under conditions (i), (ii) and (iv), but not with vertical scanning."

persons had for example difficulty recognizing words if there were objects present - the objects seemed to outvote the words. It was also difficult to recognize squares represented by lines when another object represented by corners was present. On the other hand, it was no problem if the square was represented by corners and the other object by lines, see figure 2.5. This implies that corners outvote lines in some sense, and that they therefore are more important. This may also be because corners have a stronger influence than lines upon the attention mechanism.

Tanaka et al ([Tanaka, 1996]), ([Kobatake and Tanaka, 1994]): It is assumed that there are two visual pathways in the brain, popularly called the *where* and *what* pathways. The first one deals with location of objects. The second one, also called the ventral visual pathway, presumably deals with recognition of objects. One simplified model is that the visual information is processed

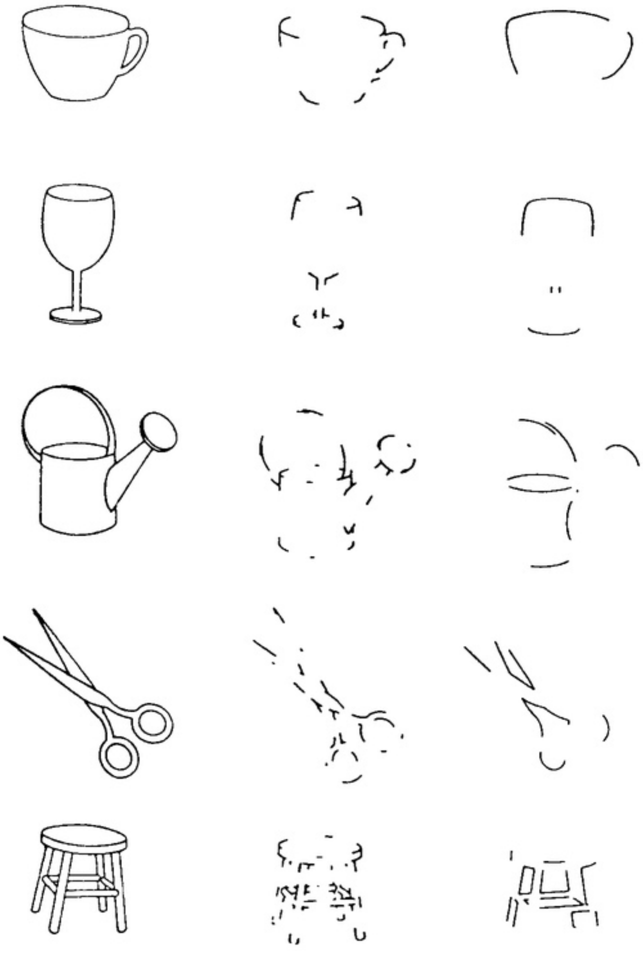


Figure 2.4: From [Biederman, 1987]. Perceptual experiment showing the importance of curvature. Quote: “Example of five stimulus objects in the experiment on the perception of degraded objects. (The left column shows the original intact versions. The middle column shows the recoverable versions. The contours have been deleted in regions where they can be replaced through collinearity or smooth curvature. The right column shows the nonrecoverable versions. The contours have been deleted at regions of concavity so that collinearity or smooth curvature of segments bridges the concavity. In addition, vertices have been altered, for example, from Ys to Ls, and misleading symmetry and parallelism have been introduced.)”

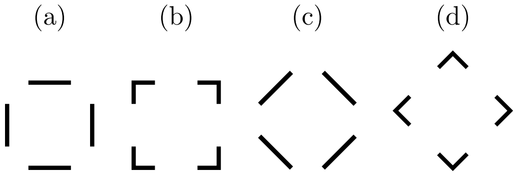


Figure 2.5: From [Humphreys et al., 1994]. Square and diamond test patterns represented by lines and corners. Objects (a),(b),(c),(d) were shown either isolated or in pairs (a,d) and (b,c) for a short duration of time, and two persons with lesions in the parietal lobe was asked to detect whether a square was present. They failed the task for the pair (a),(d) but not for isolated objects or for the pair (b,c).

approximately in a sequence through five local areas in the brain:

$$V1 \Rightarrow V2 \Rightarrow V4 \Rightarrow TEO \Rightarrow TE \tag{2.1}$$

TE (inferotemporal cortex) is assumed to be the last area in the pathway specifically involved in visual processing. The information from TE goes out to other parts of the brain. The cell responses become more refined and selective along the way, and their behavior is very non-linear and unpredictable at the end of the path. Tanaka et al measured individual cell responses to different image patterns. First, objects were shown, and the ones that gave a cell response were subsequently simplified in a way that the cell response remained high, to finally arrive at what they called the cell critical feature. This pattern was assumed to be the simplest, yet optimal pattern for the cell. Some of these patterns are shown in figure 2.6. The patterns should not be taken too seriously, it would be almost impossible to find the optimal pattern for an individual cell, but they may at least give a hint of the complexity in each brain area. It can also be mentioned that the cell responses were more invariant to size and position of the patterns in the later areas TEO and TE.

Horridge ([Horridge, 2000]) has shown that honeybees can be taught to discriminate between circle and star patterns. His experiments also suggest that the bees have 'tangential' and 'radial' filters, i.e. cells that are sensitive to edges directed out from the center of the eye's fixation point (e.g. edges in a star pattern) and edges directed orthogonally to those (e.g. edges in a circular pattern).

2.2 Some image feature detectors

In this thesis a strategy for detection of complex curvature is developed. There are many principles described in the literature for detection of similar features, or at least in some sense for detection of features at the same complexity level. Examples

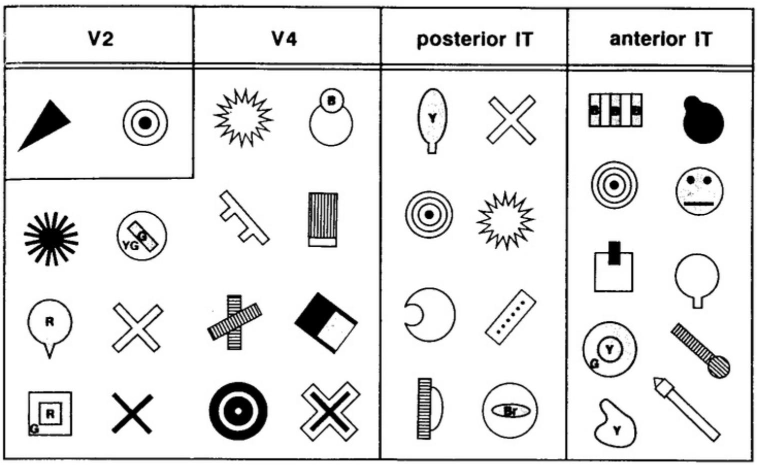


Figure 2.6: From [Kobatake and Tanaka, 1994]. Quote: "Examples of the complex critical features in the 4 regions. YG, yellow green; Br, brown."

are corners, curvature, symmetries, line endings and junctions. Applications involve motion segmentation and tracking of objects [Smith and Brady, 1995], image enhancement and restoration [Smith and Brady, 1997], and 3D surface reconstruction by tracking corners in time [Charnley and Blisset, 1989].

This section reviews some of these detectors. It is not the intention to compare the different detectors but merely to give an idea of what is available today and discuss some advantages and disadvantages. The detectors either use intensity information or local orientation information. They are presented in chronological order.

Moravec ([Moravec, 1977]): Probably the first 'points of interest'-detector. Directional variance is measured over small square windows of typically 4 to 8 pixels on a side. Sums of squares of differences of pixels adjacent in each of four directions (horizontal, vertical and two diagonals) over the window are obtained. The variance of the window is the minimum of these four sums. This variance is then used as a measure of information in the window.

Beaudet ([Beaudet, 1978]): Also one of the first attempts to detect interesting points. The local image area is approximated with its k:th order Taylor series expansion using least squares (different values of k are tried). He then evaluates some different corner detectors based on the model parameters and decides that the best one is

$$I_{xx}I_{yy} - I_{xy}^2 \tag{2.2}$$

where I_{xx}, I_{yy}, and I_{xy} are second order derivatives of the image intensity function.

Haralick, Kitchen & Rosenfeld, and Nagel ([Haralick and Watson, 1981], [Haralick, 1984], [Haralick and Shapiro, 1993], [Kitchen and Rosenfeld, 1982], [Nagel, 1983]) have all developed similar corner detectors based on polynomial expansion models. An incomplete third degree polynomial model is fitted locally to the image:

$$\begin{aligned} I(x,y) \quad \sim \quad & k_1 + k_2x + k_3y + k_4x^2 + k_5xy + k_6y^2 \\ & +k_7x^3 + k_8x^2y + k_9xy^2 + k_{10}y^3 \end{aligned} \tag{2.3}$$

This model is sometimes called the facet model. A well known curvature measure is the derivative of the contour tangent angle along the contour. It can be shown that this curvature measure can be computed as

$$\kappa = \frac{2I_{xy}I_xI_y - I_{yy}I_x^2 - I_{xx}I_y^2}{(I_x^2 + I_y^2)^{3/2}} \sim \frac{2(k_2k_3k_5 - k_6k_2^2 + k_4k_3^2)}{(k_2^2 + k_3^2)^{3/2}} \tag{2.4}$$

The corner detector is then defined as

$$\kappa |\nabla I|^\gamma \tag{2.5}$$

$|\nabla I|^\gamma$ can be viewed as a certainty measure of the curvature κ.

Rotational symmetries : The rotational symmetries are thoroughly described in chapter 5. The theory was developed around 1981 by Granlund and Knutsson and a number of people have been doing research on them, see the review in section 5.4. The basic idea is to use local orientation (in double angle representation) to detect complex curvature. A set of filters is applied on the orientation image and from the result it is possible to detect and distinguish between a number of features, such as corners, circles, and star patterns.

Harris ([Harris and Stephens, 1988]): Harris detector, also called Plessey detector, is one of the best known detectors for point features. First the image gradient ∇I is computed, for instance using differentiated Gaussian filters. Then the outer product of the gradient is computed and averaged over a local area in the image using a Gaussian filter,

$$A = \sum_{\mathbf{x}} g(\mathbf{x}) \nabla I(\mathbf{x}) \nabla I(\mathbf{x})^T \tag{2.6}$$

A is a 2×2 matrix and is sometimes called an orientation tensor. By looking at the eigenvalues of this matrix we can decide whether the local image area contains a one- or two-dimensional structure. For example, edges and lines will give one large and one small eigenvalue, while for corners both eigenvalues will be large. The Harris corner detector is defined as

$$\det(A) - 0.04\mathrm{trace}^2(A) \tag{2.7}$$

This detector is often claimed to be a corner detector but it detects a whole range of patterns and cannot distinguish between corners and other two-dimensional structures.

Noble ([Noble, 1988]) presented a detector which is closely related to the Harris detector:

$$\frac{\det(A)}{\text{trace}(A)} \tag{2.8}$$

where A is computed according to equation 2.6. It is shown that this measure is approximately the average curvature weighted with the image gradient.

Cooper ([Cooper et al., 1990]) detects corners in two steps; First, find possible corner locations by testing similarity between image patches along the edge direction. The patches differ if we are close to a corner. Second, compute the contour direction and if the absolute value of the second derivative along the contour direction is greater than zero, the region is detected as a corner.

Mehrotra, Nichani, and Ranganathan ([Mehrotra et al., 1990]) Line endings are detected in 8 different directions. First and second derivatives of a Gaussian are used as filters (where the origin is located on the edge of the filter instead of in the middle). Both corner angle and corner orientation are computed.

Deriche ([Deriche and Giraudon, 1990]): An attempt to improve the location of the response from Beaudet's detector in equation 2.2. Corners are detected in two different scales using the Beaudet detector. A line is drawn between the two responses and the Laplacian is computed along the line. The zero crossing of the Laplacian is selected as the corner location.

Bårman ([Bårman, 1991]) detects curvature from a local orientation description in double angle representation. The local orientation image is correlated with a set of quadrature filters, and the responses are used to detect curvature. This curvature is related to the first order rotational symmetries described in chapter 5. Bårman also uses a similar strategy to detect curvature in 3D and 4D data.

Rohr ([Rohr, 1992]): Rohr defines a corner by the parameters α (corner orientation), β (corner angle), a (corner amplitude), and σ (corner softness). Junctions are then modeled as a sum of corner regions. A simple point-of-interest detector is used to find preliminary locations in which the parameters are optimized from a least squares problem (the solution is found by an iterative method). Corners, T-, L-, K-, X- and arrow-junctions can be detected but large masks (20×20) are used which makes the algorithm computationally complex.

Reisfeld ([Reisfeld et al., 1995]) Describes an operator that measures symmetries using the image gradient. For each position $\mathbf{r}_0$ we will get a contributioin to the symmetry response from each pair of pixel positions $\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{r}$ and $\mathbf{r}_2 = \mathbf{r}_0 + \mathbf{r}$. The contribution C_{12} is computed as $C_{12} = D_{12}P_{12}|\nabla I(\mathbf{r}_1)||\nabla I(\mathbf{r}_2)|$ where

$$D_{12} = \frac{1}{\sqrt{2\pi}\sigma} e^{-|\mathbf{r}_1 - \mathbf{r}_2|/2\sigma} \tag{2.9}$$

is a distance function and

$$P_{12} = (1 - \cos(\theta_1 + \theta_2 - 2\alpha_{12}))(1 - \cos(\theta_1 - \theta_2)) \ , \ \begin{cases} \theta_1 = \angle \nabla I(\mathbf{r}_1) \\ \theta_2 = \angle \nabla I(\mathbf{r}_2) \\ \alpha_{12} = \angle \mathbf{r} \end{cases} \tag{2.10}$$

is a function of gradient direction. The first term of P_{12} is high when the gradients are oriented in the same direction toward each other, i.e. symmetric with respect to the line going through $\mathbf{r}_1$ and $\mathbf{r}_2$. The second term gives a low response for parallel gradients, which includes edge patterns. A circle is the optimal pattern. The highest responses were located on the eyes and the mouth when applied to an image with a face. The algorithm can be efficiently implemented.

SUSAN ([Smith and Brady, 1997],[Smith and Brady, 1995]) SUSAN stands for Smallest Univalue Segment Assimilation Nucleus. This corner detector is both fast (for example 10 times faster than the Harris detector) and noise robust. The algorithm is as follows:

1. Place a circular mask around the pixel (nucleus) in question, $\mathbf{r}_0$.
2. Compute the number of pixels within the mask which have similar brightness to the nucleus using the formula

$$n(\mathbf{r}_0) = \sum_{\mathbf{r}} c(\mathbf{r}, \mathbf{r}_0) \quad \text{where} \quad c(\mathbf{r}, \mathbf{r}_0) = e^{(\frac{I(\mathbf{r}) - I(\mathbf{r}_0)}{t})^6} \tag{2.11}$$

The set of pixels $\mathbf{r}$ with high value $c(\mathbf{r}, \mathbf{r}_0)$ is called the USAN set.

3. Threshold $n(\mathbf{r}_0)$ to get initial responses using the formula

$$R(\mathbf{r}_0) = \begin{cases} n_{max}/2 - n(\mathbf{r}_0) & \text{if } n(\mathbf{r}_0) < n_{max}/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

4. Test for false positives; the center of gravity of the USAN set should be far away from the nucleus and all pixels between the nucleus and the center of gravity should belong to the USAN set.
5. Apply non-max-suppression to find the corners.

Trajkovic and Hedley ([Trajkovic and Hedley, 1998]) This corner detector is based on the minimum intensity change (MIC) and the Corner Response Function, CRF, and can roughly be described as

$$\text{CRF} = \min_{\mathbf{r}}[(I(\mathbf{r}_0 + \mathbf{r}) - I(\mathbf{r}_0))^2 - (I(\mathbf{r}_0 - \mathbf{r}) - I(\mathbf{r}_0))^2] \tag{2.13}$$

In reality the formula is a bit more complicated and fuzzy than above. The CRF can be computed in a efficient way.

Chain code In addition to the algorithms described above there are a number of local curvature detectors based on a local or object contour descriptions, or *chain codes*. This approach requires a prior segmentation step which might work well for well behaved images but less so for natural images.

Examples of multiscale curvature detection, called curvature scale space, can be found in [Rosin, 1992], [Mokhtarian et al., 1996]. In this cases the object contour is extracted and the curvature is computed in several scales. Infliction points, which is defined as points where the curvature is switching from concave to convex, is used as feature points for the object.

There are no general evaluation criteria or test images for complex feature detectors. Results are often presented for very simple, synthetic images. Some of the developers above have compared their detectors with others on synthetic noisy images, but the results are inconclusive. Still, some general criteria have been proposed for a good feature detector:

- Good detection: minimum number of false negatives/positives.
- Good location: It is often argued that a corner detector should give highest response at the corner point and not somewhere inside the corner which is the case for example in the curvature detector presented in this thesis. This is however no problem as long as the location is consistent.
- Only one response to each single feature.
- Speed: This is an important criterion for practical applications in general and real-time applications in particular.
- Insensitive to noise.

The feature detectors presented in chapter 5 in this thesis can be very efficiently implemented to detect curvature in several scales. They are also able to detect and distinguish between a number of useful features such as curvature, circles, and star-patterns, which is more than most of the detectors presented above can handle. In addition they experience a graceful degradation with respect to many geometrical transformation such as rotation, zooming, and change of view. They should therefore be a good platform for further analysis.

Chapter 3

Fundamental tools

This chapter briefly describes the two fundamental tools used in this thesis; normalized convolution and canonical correlation. They are called fundamental because they are quite general tools in signal processing and statistical analysis.

3.1 Normalized convolution

This section contains a short summary of the normalized convolution technique. For a more thorough description, see [Farnebäck, 1999b] and [Farnebäck, 1999a]. The technique was developed about 10 years ago, see [Knutsson and Westin, 1993], [Westin, 1994] (some of the ideas can be also traced in [Knutsson et al., 1988b] and [Burt, 1988]).

3.1.1 Summary

Normalized convolution models a signal with a linear combination of a set of basis functions. It takes into account the uncertainty in the signal values and also permits spatial localization of the basis functions which may have infinite support.

Let $\{\mathbf{b_n}\}_1^N$ be a set of vectors in $\mathbb{C}^M$. Assume we want to represent, or approximate, a vector $\mathbf{f} \in \mathbb{C}^M$ with a linear combination of $\{\mathbf{b_n}\}$, i.e.

$$\mathbf{f} \sim \sum_1^N s_n \mathbf{b}_n = \mathbf{B}\mathbf{s} \tag{3.1}$$

where

$$\mathbf{B} = \begin{pmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_N \\ | & | & & | \end{pmatrix} \quad , \quad \mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} \tag{3.2}$$

With the signal vector $\mathbf{f} \in \mathbb{C}^M$ we attach a *signal certainty vector* $\mathbf{c} \in \mathbb{R}^M$ indicating our confidence in the values of $\mathbf{f}$. Similarly we attach an *applicability function* $\mathbf{a} \in \mathbb{R}^M$ to the basis functions $\{\mathbf{b}_n\} \in \mathbb{C}^M$ to use as a window for spatial localization of the basis functions.

This thesis only deals with the case of $\{\mathbf{b_n}\}_1^N$ being linearly independent and spanning a subspace of $\mathbb{C}^M$ (i.e. $N \leq M$). The approximation can then be described as a weighted least squares problem, where the weight is a function of the certainty and the applicability [1]:

$$\arg\min_{\mathbf{s}\in\mathbb{C}^N} \|\mathbf{f}-\mathbf{B}\mathbf{s}\|_W^2 = \arg\min_{\mathbf{s}\in\mathbb{C}^N} (\mathbf{f}-\mathbf{B}\mathbf{s})^*\mathbf{W}(\mathbf{f}-\mathbf{B}\mathbf{s}) \tag{3.3}$$

where $\mathbf{W} = \mathbf{W}_a\mathbf{W}_c$, $\mathbf{W}_a = \mathrm{diag}(\mathbf{a})$, $\mathbf{W}_c = \mathrm{diag}(\mathbf{c})$.

This has the effect that elements in $\mathbf{f}$ with a low certainty and elements in $\mathbf{b}_n$ with a low applicability value have less influence on the solution than elements with a high certainty and applicability. The solution becomes

$$\mathbf{s} = (\mathbf{B}^*\mathbf{W}\mathbf{B})^{-1}\mathbf{B}^*\mathbf{W}\mathbf{f} = \tilde{\mathbf{B}}^*\mathbf{f} \quad \text{where} \quad \tilde{\mathbf{B}} = \mathbf{W}\mathbf{B}(\mathbf{B}^*\mathbf{W}\mathbf{B})^{-1} \tag{3.4}$$

The columns of $\tilde{\mathbf{B}}$ are called the *dual basis* of $\{\mathbf{b}_n\}$.

In terms of inner products the solution can be written as

$$\begin{aligned}
\mathbf{s} &= \begin{pmatrix} \langle \mathbf{b}_1,\, \mathbf{b}_1\rangle_W & \dots & \langle \mathbf{b}_1,\, \mathbf{b}_N\rangle_W \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_N,\, \mathbf{b}_1\rangle_W & \dots & \langle \mathbf{b}_N,\, \mathbf{b}_N\rangle_W \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{b}_1,\, \mathbf{f}\rangle_W \\ \vdots \\ \langle \mathbf{b}_N,\, \mathbf{f}\rangle_W \end{pmatrix} \qquad (3.5)\\
&= \begin{pmatrix} \langle \mathbf{a}\cdot\mathbf{b}_1,\, \mathbf{c}\cdot\mathbf{b}_1\rangle & \dots & \langle \mathbf{a}\cdot\mathbf{b}_1,\, \mathbf{c}\cdot\mathbf{b}_N\rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{a}\cdot\mathbf{b}_N,\, \mathbf{c}\cdot\mathbf{b}_1\rangle & \dots & \langle \mathbf{a}\cdot\mathbf{b}_N,\, \mathbf{c}\cdot\mathbf{b}_N\rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{a}\cdot\mathbf{b}_1,\, \mathbf{c}\cdot\mathbf{f}\rangle \\ \vdots \\ \langle \mathbf{a}\cdot\mathbf{b}_N,\, \mathbf{c}\cdot\mathbf{f}\rangle \end{pmatrix}\\
&= \begin{pmatrix} \langle \mathbf{a}\cdot\mathbf{b}_1\cdot\bar{\mathbf{b}}_1,\, \mathbf{c}\rangle & \dots & \langle \mathbf{a}\cdot\mathbf{b}_1\cdot\bar{\mathbf{b}}_N,\, \mathbf{c}\rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{a}\cdot\mathbf{b}_N\cdot\bar{\mathbf{b}}_1,\, \mathbf{c}\rangle & \dots & \langle \mathbf{a}\cdot\mathbf{b}_N\cdot\bar{\mathbf{b}}_N,\, \mathbf{c}\rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{a}\cdot\mathbf{b}_1,\, \mathbf{c}\cdot\mathbf{f}\rangle \\ \vdots \\ \langle \mathbf{a}\cdot\mathbf{b}_N,\, \mathbf{c}\cdot\mathbf{f}\rangle \end{pmatrix}
\end{aligned}$$

where '·' denotes pointwise multiplication.

The signal $\mathbf{f}$ can for instance be a local area in an image and the basis functions $\mathbf{b}_n$ can be polynomials, Fourier functions or other useful analyzing functions. Doing this approximation in each local area of the image can be efficiently implemented by means of convolutions, hence the name normalized convolution. This

[1] In the general case the problem can be formulated as

$$\arg\min_{\mathbf{r}\in\mathcal{S}} \|\mathbf{r}\|, \quad \mathcal{S} = \{\mathbf{r} \in \mathcal{C}^M; \|\mathbf{B}\mathbf{r}-\mathbf{f}\|_{\mathbf{W}} \text{ is a minimum}\}$$

is because the left arguments in the scalar product, $\mathbf{a} \cdot \mathbf{b}_i \cdot \bar{\mathbf{b}}_j$ and $\mathbf{a} \cdot \mathbf{b}_i$, can be interpreted as filters that are to be correlated with the signals $\mathbf{c}$ and $\mathbf{c} \cdot \mathbf{f}$ respectively.

If the overall signal certainty $\mathbf{c}$ is too low we cannot rely on the result $\mathbf{s}$. For the signal $\mathbf{f}$ we had a certainty measure, $\mathbf{c}$, indicating how well we can rely on the information in $\mathbf{f}$. We can apply the same philosophy for the solution $\mathbf{s}$ and use an output certainty, $\mathbf{c}_{\text{out}}$, indicating how well we can rely on $\mathbf{s}$. There exist several suggestions for $\mathbf{c}_{\text{out}}$, see [Farnebäck, 1999a]. The one used in this thesis is from [Westelius, 1995]:

$$\mathbf{c}_{\text{out}} = \left(\frac{\det(\mathbf{B}^* \mathbf{W}_a \mathbf{W}_c \mathbf{B})}{\det(\mathbf{B}^* \mathbf{W}_a \mathbf{B})} \right)^{1/N} \tag{3.6}$$

which measures how 'less distinguishable' the basis functions becomes when we include uncertainty compared to full certainty. Note that even if the basis functions are orthogonal in the case of full certainty ($\mathbf{c} \equiv 1$) they may not necessarily be so when the certainty is varying. The $1/N$ exponent makes $\mathbf{c}_{\text{out}}$ proportional to $\mathbf{c}$.

3.1.2 Simple example

As a simple example consider a signal and two basis functions in $\mathbb{R}^3$:

$$\mathbf{f} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad , \quad \mathbf{b}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad , \quad \mathbf{b}_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \tag{3.7}$$

and their corresponding certainty and applicability respectively:

$$\mathbf{c} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad , \quad \mathbf{a} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \tag{3.8}$$

We thus have

$$\mathbf{B} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{pmatrix} \tag{3.9}$$

and

$$\mathbf{W} = \mathbf{W}_a \mathbf{W}_c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.10}$$

and the solution becomes

$$\mathbf{s} = (\mathbf{B}^T \mathbf{W} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{W} \mathbf{f} = \begin{pmatrix} 3 & -2 \\ -2 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 4 \\ -4 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \tag{3.11}$$

i.e.

$$\mathbf{f} \sim 0\mathbf{b}_1 - 2\mathbf{b}_2 = -2\mathbf{b}_2 \tag{3.12}$$

Another example can be found in section 4.3 where each local area in an image is approximated by a first degree polynomial. The basis functions in this case are 1, x, and y. The signal certainty is chosen as 1 inside the image and 0 outside image border and the applicability is chosen as a Gaussian function. The polynomial model is then used to estimate image gradient.

3.2 Canonical correlation

This section contains a short summary of the canonical correlation technique. For a more thorough description, see [Borga, 1998].

3.2.1 Summary

Assume that we have two stochastic variables

$$\mathbf{x} \in \mathbb{C}^{M_1} \quad \text{and} \quad \mathbf{y} \in \mathbb{C}^{M_2} \tag{3.13}$$

M_1 and M_2 do not have to be equal. For simplicity we can assume that they both have zero mean. Canonical correlation analysis, CCA, can be defined as the problem of finding two sets of basis vectors, one for $\mathbf{x}$ and the other for $\mathbf{y}$, such that the correlations between the projections of the variables onto these basis vectors are mutually maximized. In other words, CCA measures linear relationships between two multidimensional variables.

For the case of only one pair of basis vectors we have the projections $x = \mathbf{w}_x^* \mathbf{x}$ and $y = \mathbf{w}_y^* \mathbf{y}$ (* denotes conjugate transpose) and the correlation is written as

$$\begin{aligned} \rho &= \frac{\mathrm{E}[xy]}{\sqrt{\mathrm{E}[x^2]\mathrm{E}[y^2]}} \\ &= \frac{\mathrm{E}[\mathbf{w}_x^* \mathbf{x}\mathbf{y}^* \mathbf{w}_y]}{\sqrt{\mathrm{E}[\mathbf{w}_x^* \mathbf{x}\mathbf{x}^* \mathbf{w}_x]\mathrm{E}[\mathbf{w}_y^* \mathbf{y}\mathbf{y}^* \mathbf{w}_y]}} \\ &= \frac{\mathbf{w}_x^* \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^* \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^* \mathbf{C}_{yy} \mathbf{w}_y}} \end{aligned} \tag{3.14}$$

where E[.] denotes expectation value and

$$\mathbf{C}_{xy} = \mathrm{E}[\mathbf{x}\mathbf{y}^*] \quad , \quad \mathbf{C}_{xx} = \mathrm{E}[\mathbf{x}\mathbf{x}^*] \quad , \quad \mathbf{C}_{yy} = \mathrm{E}[\mathbf{y}\mathbf{y}^*] \tag{3.15}$$

The maximal canonical correlation is found by maximizing ρ with respect to $\mathbf{w}_x$ and $\mathbf{w}_y$. It can be shown that the maximal canonical correlation can be found by solving an eigenvalue system.

$$\begin{cases} \mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\hat{\mathbf{w}}_x &= \rho^2 \hat{\mathbf{w}}_x \\ \mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\hat{\mathbf{w}}_y &= \rho^2 \hat{\mathbf{w}}_y \end{cases} \tag{3.16}$$

The eigenvectors $\hat{\mathbf{w}}_{x1}$, $\hat{\mathbf{w}}_{y1}$ corresponding the the largest eigenvalue ρ_1^2 are the projections that have the highest canonical correlation ρ_1. The next two eigenvectors $\hat{\mathbf{w}}_{x2}$, $\hat{\mathbf{w}}_{y2}$ have the second highest correlation ρ_2 and so on.

Only one of the eigenvalue equations needs to be solved since the solutions are related by

$$\left\{ \begin{array}{lcl} \mathbf{C}_{xy}\hat{\mathbf{w}}_y & = & \rho\lambda_x \mathbf{C}_{xx}\hat{\mathbf{w}}_x \\ \mathbf{C}_{yx}\hat{\mathbf{w}}_x & = & \rho\lambda_y \mathbf{C}_{yy}\hat{\mathbf{w}}_y \end{array} \right. \quad \text{where} \quad \lambda_x = \lambda_y^{-1} = \sqrt{\frac{\hat{\mathbf{w}}_y^* \mathbf{C}_{yy}\hat{\mathbf{w}}_y}{\hat{\mathbf{w}}_x^* \mathbf{C}_{xx}\hat{\mathbf{w}}_x}} \tag{3.17}$$

It can also be shown that the different projections are uncorrelated, i.e.

$$\left\{ \begin{array}{lclcl} \mathrm{E}[x_i x_j] & = & \mathbf{w}_{xi}^* \mathbf{C}_{xx} \mathbf{w}_{xj} & = & 0 \\ \mathrm{E}[y_i y_j] & = & \mathbf{w}_{yi}^* \mathbf{C}_{yy} \mathbf{w}_{yj} & = & 0 \\ \mathrm{E}[x_i y_j] & = & \mathbf{w}_{xi}^* \mathbf{C}_{xy} \mathbf{w}_{yj} & = & 0 \end{array} \right. \tag{3.18}$$

Another property is that CCA is invariant to affine transformations. If we for instance transform $\mathbf{x}$ to $\mathbf{u} = \mathbf{A}\mathbf{x}$ we simply get the new solution $\mathbf{w}_{ui} = \mathbf{A}^* \mathbf{w}_{xi}$.

It can also be mentioned that CCA is closely related to mutual information. If $\mathbf{x}$ and $\mathbf{y}$ are Gaussian variables the mutual information can be computed from the correlations ρ_i.

3.2.2 Simple example

Let a, b, and c be three independent stochastic variables with zero mean and standard deviations σ_a, σ_b, and σ_c respectively. Let

$$\mathbf{x} = \begin{pmatrix} a \\ b \end{pmatrix} \quad , \quad \mathbf{y} = \begin{pmatrix} a+c \\ a-c \end{pmatrix} \tag{3.19}$$

We then have

$$\mathbf{C}_{xx} = \begin{pmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{pmatrix}, \; \mathbf{C}_{yy} = \begin{pmatrix} \sigma_a^2+\sigma_c^2 & \sigma_a^2-\sigma_c^2 \\ \sigma_a^2-\sigma_c^2 & \sigma_a^2+\sigma_c^2 \end{pmatrix}, \; \mathbf{C}_{xy} = \begin{pmatrix} \sigma_a^2 & \sigma_a^2 \\ 0 & 0 \end{pmatrix} \tag{3.20}$$

which gives

$$\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.21}$$

and the first eigensystem in equation 3.16 has the solution

$$\left\{ \begin{array}{l} \rho_1 = 1 \quad , \quad \hat{\mathbf{w}}_{x1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \rho_2 = 0 \quad , \quad \hat{\mathbf{w}}_{x2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{array} \right. \tag{3.22}$$

The $\mathbf{w}_{yi}$ vectors can be found from the second eigensystem in equation 3.16 or from the second system in equation 3.17:

$$\left\{ \begin{array}{lll} \mathbf{w}_{y1} = & \mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\hat{\mathbf{w}}_{x1} = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \end{pmatrix} \hat{\mathbf{w}}_{x1} = & \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \\ \hat{\mathbf{w}}_{y2} = & & \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix} \end{array} \right. \tag{3.23}$$

(The last vector $\mathbf{w}_{y2}$ cannot be computed from equation 3.17 since $\rho_2 = 0$.)
The projections onto the vectors corresponding to the largest canonical correlation become

$$x_1 = \hat{\mathbf{w}}_{x1}^T \mathbf{x} = a \quad , \quad y_1 = \hat{\mathbf{w}}_{y1}^T \mathbf{y} = a\sqrt{2} \tag{3.24}$$

and we see that they are fully correlated, as indicated by $\rho_1 = 1$.

Chapter 4

Local polynomial expansion

4.1 Introduction

Polynomials as a local signal model have been used in a number of image analysis applications including gradient edge detection, zero-crossing edge detection, image segmentation, line detection, corner detection, three-dimensional shape estimation from shading, and determination of optical flow, see [Haralick and Shapiro, 1993], [Haralick, 1984], [Haralick and Watson, 1981]. The polynomial model is fitted to a local square-shaped neighborhood in the image using non-weighted least squares.

Recently Farnebäck has shown that polynomial expansion using weighted least squares with a Gaussian weight function can give much better results on local orientation and motion estimation than other existing methods, see [Farnebäck, 1999a], [Farnebäck, 2000b], [Farnebäck, 2000a], [Farnebäck, 1999b]. The expansion can be made by means of correlations with Cartesian separable filters which make the algorithm computationally efficient. The idea of using weighted least squares for polynomial expansion has also been mentioned in [Westin, 1994] where a second degree polynomial was used for gradient estimation in irregularly sampled data, but nothing was said about efficient filtering. The idea can also be found in [Burt, 1988] where a bilinear model ($r_0 + r_1x + r_2y + r_3xy$) was used for interpolation in incomplete and irregularly sampled data. The model was efficiently estimated from image moments computed locally in the image. The model was also estimated in several scales by combining image moments in finer scales to compute moments in coarser scales.

This chapter presents the polynomial expansion theory and an alternative approximative polynomial expansion algorithm that efficiently computes the parameters of Farnebäck's polynomial model in one or several scales. The algorithm is based on the simple observation that polynomial functions multiplied with a Gaussian function can be described in terms of partial derivatives of the Gaussian.

The chapter outline is as follows: Section 4.2 summarizes the work done by

Farnebäck. Section 4.3 illustrates the theory on a simple gradient estimation experiment. Sections 4.4 and 4.5 describe the new approximative algorithm in one and several scales respectively. Sections 4.6, 4.7, and 4.8 discusses practical details, computational complexity and conclusions. An evaluation of the algorithm is found in the experiment chapter, section 6.1.

4.2 Using normalized convolution

This section contains a short summary of work done by Farnebäck. Further details can be found in [Farnebäck, 1999a]. The theory is for pedagogical reasons explained using a second degree polynomial model on a two-dimensional signal but the generalization to other polynomial orders and signal dimensionalities is straightforward.

Assume we want to model an N-dimensional signal f with a second degree polynomial:

$$f(\mathbf{x}) \sim c + \mathbf{b}^T\mathbf{x} + \mathbf{x}^T\mathbf{A}\mathbf{x} \quad , \quad \mathbf{x} \in \mathbb{R}^N \tag{4.1}$$

where c is a scalar, $\mathbf{b}$ is a vector and $\mathbf{A}$ is a symmetric matrix. In the two-dimensional case we have

$$f(x, y) \sim r_1 + r_2x + r_3y + r_4x^2 + r_5y^2 + r_6xy \tag{4.2}$$

Let P_2 denote the *second degree polynomial basis* in $\mathbb{R}^2$, i.e. the basis consisting of all 2D-monomials up to the second degree:

$$P_2 = \{1, x, y, x^2, y^2, xy\} \tag{4.3}$$

In practice the polynomial model is applied to a limited area of size $n \times n$ in a pixel-discretized image. After reshaping the local signal and basis functions into vectors we can describe them as elements in $\mathbb{R}^{n^2}$ (or $\mathbb{C}^{n^2}$). Equation 4.2 can then be rewritten as

$$\mathbf{f} \sim \mathbf{P}_2\mathbf{r} \tag{4.4}$$

where

$$\mathbf{P}_2 = \begin{pmatrix} | & | & | & | & | & | \\ \mathbf{1} & \mathbf{x} & \mathbf{y} & \mathbf{x}^2 & \mathbf{y}^2 & \mathbf{xy} \\ | & | & | & | & | & | \end{pmatrix} \quad , \quad \mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_6 \end{pmatrix} \tag{4.5}$$

If we use the normalized convolution theory in section 3.1 we can define the polynomial expansion problem as

$$\arg\min_{\mathbf{r}\in\mathbb{C}} \|\mathbf{f} - \mathbf{P}_2\mathbf{r}\|_W = \arg\min_{\mathbf{r}\in\mathbb{C}} (\mathbf{f} - \mathbf{P}_2\mathbf{r})^*\mathbf{W}(\mathbf{f} - \mathbf{P}_2\mathbf{r}) \tag{4.6}$$

where $\mathbf{W} = \mathbf{W}_a \mathbf{W}_c$, $\mathbf{W}_a = \text{diag}(\mathbf{a})$ is the applicability weight and $\mathbf{W}_c = \text{diag}(\mathbf{c})$ is the signal certainty weight. The solution becomes

$$\mathbf{r} = (\mathbf{P}_2^T \mathbf{W} \mathbf{P}_2)^{-1} \mathbf{P}_2^T \mathbf{W} \mathbf{f} \tag{4.7}$$

It is assumed that we have enough certainty so that the inverse of $\mathbf{P}_2^T \mathbf{W} \mathbf{P}_2$ exists. The choice of applicability and certainty in general depends on the application. There is however one choice of applicability that often is to be preferred due to its nice properties: the Gaussian function. The Gaussians are the only functions which are simultaneously both isotropic and Cartesian separable. Cartesian separability gives efficient computational structures while the isotropic property gives well behaved results. It has for instance been shown in an orientation estimation experiment using polynomial expansion that among a number of different choices of applicability, e.g. cube, sphere, cone, etc., the Gaussian function gave the best result (see [Farnebäck, 1999a]).

This choice of applicability will also lead to a very efficient approximative polynomial expansion algorithm as we will see in section 4.4.

The computational structure differs depending on whether we have full certainty or not. The next two subsections deal with these two cases.

4.2.1 Full certainty

In the case of full certainty we have

$$\mathbf{W}_c = \mathbf{I} \tag{4.8}$$

and the polynomial expansion solution in equation 4.7 is reduced to

$$\begin{aligned}
\mathbf{r} &= (\mathbf{P}_2^T \mathbf{W}_a \mathbf{P}_2)^{-1} \mathbf{P}_2^T \mathbf{W}_a \mathbf{f} \qquad (4.9) \\
&= \begin{pmatrix} 1 & & & \sigma^2 & \sigma^2 & \\ & \sigma^2 & & & & \\ & & \sigma^2 & & & \\ \sigma^2 & & & 3\sigma^4 & \sigma^4 & \\ \sigma^2 & & & \sigma^4 & 3\sigma^4 & \\ & & & & & \sigma^4 \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{1} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{x} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{y} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{x}^2 \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{y}^2 \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{xy} \cdot \mathbf{g},\ \mathbf{f} \rangle \end{pmatrix} \\
&= \begin{pmatrix} 2 & & & -\frac{1}{2\sigma^2} & -\frac{1}{2\sigma^2} & \\ & \frac{1}{\sigma^2} & & & & \\ & & \frac{1}{\sigma^2} & & & \\ -\frac{1}{2\sigma^2} & & & \frac{1}{2\sigma^4} & & \\ -\frac{1}{2\sigma^2} & & & & \frac{1}{2\sigma^4} & \\ & & & & & \frac{1}{\sigma^4} \end{pmatrix} \begin{pmatrix} \langle \mathbf{1} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{x} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{y} \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{x}^2 \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{y}^2 \cdot \mathbf{g},\ \mathbf{f} \rangle \\ \langle \mathbf{xy} \cdot \mathbf{g},\ \mathbf{f} \rangle \end{pmatrix}
\end{aligned}$$

where the expression for $\mathbf{P}_2^T \mathbf{W}_a \mathbf{P}_2$ is computed assuming continous functions. The matrix $(\mathbf{P}_2^T \mathbf{W}_a \mathbf{P}_2)$ does not depend on the signal. $\mathbf{P}_2^T \mathbf{W}_a \mathbf{f}$ means that we correlate the signal $f(x, y)$ with the filters

$$g \ , \ xg \ , \ yg \ , \ x^2 g \ , \ y^2 g \ , \ xyg \tag{4.10}$$

These filters can be made Cartesian separable, and it turns out that we only have to use 9 1D-filters. Figure 4.1 contain the correlator structure needed to compute $\mathbf{P}_2^T\mathbf{W}_a\mathbf{f}$. After the correlations we multiply the result with $(\mathbf{P}_2^T\mathbf{W}_a\mathbf{P}_2)^{-1}$ in each local neighborhood to get the final solution $\mathbf{r}$.

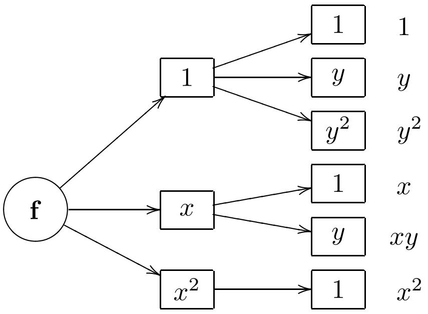


Figure 4.1: Correlator structure for polynomial expansion in 2D with full certainty. The first and second filters are 1D-filters along the x- and y-dimension respectively. There is understood to be an applicability factor in each box as well. From [Farnebäck, 1999a].

4.2.2 Uncertain data

In this case we have to compute the general solution

$$\mathbf{r} = (\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{f} \tag{4.11}$$

$\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{f}$ can as before be computed by correlating with the filters in equation 4.10 but now on the signal $c(x,y)f(x,y)$. $\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{P}_2$ now depends on the signal certainty,

$$\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{P}_2 =$$

$$\begin{pmatrix}
\langle \mathbf{1}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{c}\rangle \\
\langle \mathbf{x}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^3\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle \\
\langle \mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}^3\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^2\cdot\mathbf{g},\ \mathbf{c}\rangle \\
\langle \mathbf{x}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^3\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^4\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^3\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle \\
\langle \mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}^3\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{y}^4\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^3\cdot\mathbf{g},\ \mathbf{c}\rangle \\
\langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^2\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^3\mathbf{y}\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{xy}^3\cdot\mathbf{g},\ \mathbf{c}\rangle & \langle \mathbf{x}^2\mathbf{y}^2\cdot\mathbf{g},\ \mathbf{c}\rangle
\end{pmatrix} \tag{4.12}$$

The elements in this matrix can be computed by correlating $c(x,y)$ with the filters

$$\begin{aligned} &g\ ,\ xg\ ,\ yg\ ,\ x^2g\ ,\ y^2g\ ,\ xyg\ ,\ x^3g\ ,\ y^3g\ ,\ x^2yg\ ,\ xy^2g... \\ &\qquad\qquad\qquad ...\ x^4g\ ,\ y^4g\ ,\ x^3yg\ ,\ x^2y^2g\ ,\ xy^3g \end{aligned} \tag{4.13}$$

This can also be made by means of separable filters. Figure 4.2 contains the correlator structures needed to compute $\mathbf{P}_2\mathbf{W}_a\mathbf{f}$ and $\mathbf{P}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{P}_2$ respectively. The result from the correlations are put into equation 4.11 to get the final solution $\mathbf{r}$.

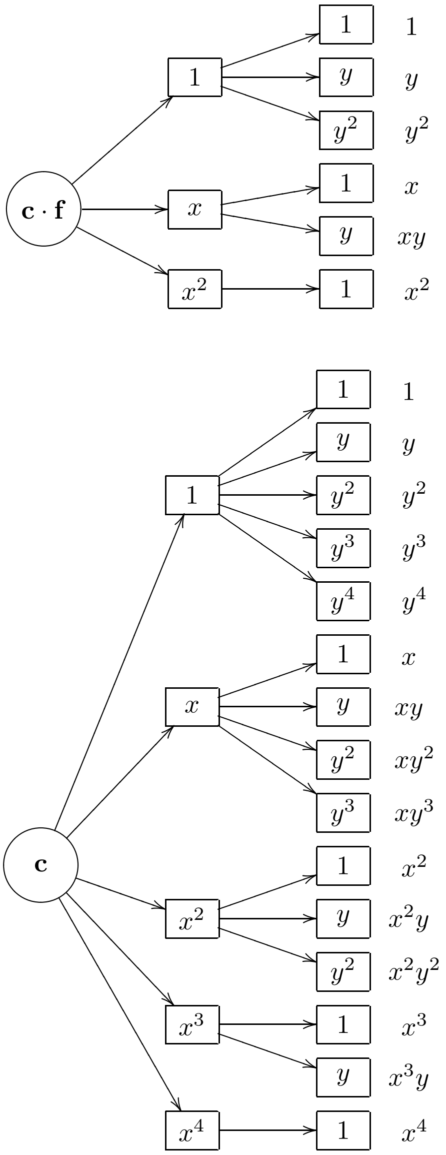


Figure 4.2: Correlator structure for polynomial expansion in 2D with uncertain data. There is understood to be an applicability factor in each box as well. From [Farnebäck, 1999a].

4.3 Example: Estimation of image gradient

For a simple example of polynomial expansion on uncertain data we turn to the problem of image gradient estimation. A very common method to estimate the image gradient is to use 'scale derivatives'. This means that the image f is first convolved with a Gaussian g,

$$f_\sigma = f * g \quad , \quad g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.14}$$

which is then differentiated. By the laws of convolution this differentiation can be computed as convolutions between the original image f and partial derivatives of g,

$$\nabla f_\sigma = \begin{pmatrix} f_{\sigma,x} \\ f_{\sigma,y} \end{pmatrix} = \begin{pmatrix} f * g_x \\ f * g_y \end{pmatrix} \quad \text{where} \quad \begin{cases} g_x = -\frac{x}{\sigma^2} g \\ g_x = -\frac{y}{\sigma^2} g \end{cases} \tag{4.15}$$

The middle image in figure 4.3 shows the result of this method using $\sigma = 10$. The method gives poor results near the image border. Usually the estimates near the border are cut off and valuable information may be lost. Another, more important problem is that this method will also give poor results if we have uncertain data within the image.

An alternative to the method above is to estimate the image gradient from a polynomial model fitted on the image. We can for example choose a first degree polynomial model:

$$f(x,y) \sim r_1 + r_2 x + r_3 y \tag{4.16}$$

and then estimate the gradient as

$$\nabla f \sim \nabla(r_1 + r_2 x + r_3 y) = \begin{pmatrix} r_2 \\ r_3 \end{pmatrix} \tag{4.17}$$

In practice we have the model

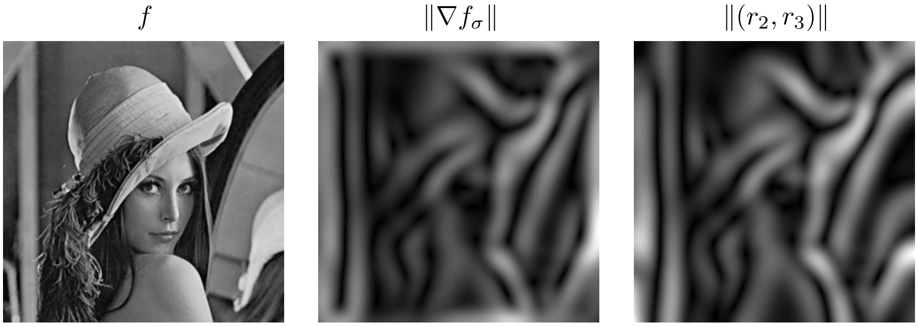
$$\mathbf{f} \sim \begin{pmatrix} | & | & | \\ \mathbf{1} & \mathbf{x} & \mathbf{y} \\ | & | & | \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \mathbf{P}_1 \mathbf{r} \tag{4.18}$$

where $\mathbf{P}_1$ denotes the first degree polynomial basis. The solution becomes

$$\begin{aligned} \mathbf{r} &= (\mathbf{P}_1^T \mathbf{W} \mathbf{P}_1)^{-1} \mathbf{P}_1^T \mathbf{W} \mathbf{f} \qquad (4.19) \\ &= \begin{pmatrix} \langle \mathbf{g} \cdot \mathbf{1},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{x},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{y},\ \mathbf{c} \rangle \\ \langle \mathbf{g} \cdot \mathbf{x},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{x^2},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{xy},\ \mathbf{c} \rangle \\ \langle \mathbf{g} \cdot \mathbf{y},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{xy},\ \mathbf{c} \rangle & \langle \mathbf{g} \cdot \mathbf{y^2},\ \mathbf{c} \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{g} \cdot \mathbf{1},\ \mathbf{c} \cdot \mathbf{f} \rangle \\ \langle \mathbf{g} \cdot \mathbf{x},\ \mathbf{c} \cdot \mathbf{f} \rangle \\ \langle \mathbf{g} \cdot \mathbf{y},\ \mathbf{c} \cdot \mathbf{f} \rangle \end{pmatrix} \end{aligned}$$

c is defined as 1 within the image and 0 outside the image. The rightmost image in figure 4.3 shows the result from this method using a Gaussian applicability with

$\sigma = 10$. Note the considerable improvement near the image borders. Of course this method has a higher computational complexity. To compute $(\mathbf{P}_1^T\mathbf{W}\mathbf{P}_1)^{-1}$ we can use the correlator structure in figure 4.1 but with $\mathbf{c}$ as input instead of $\mathbf{f}$. Then we also have to compute $\mathbf{P}_1^T\mathbf{W}\mathbf{f}$. We get totally 15 1D-filters plus solving the equation system 4.19.



f $\quad\|\nabla f_\sigma\|$ $\quad\|(r_2, r_3)\|$

Figure 4.3: Example of image gradient estimation. Left: Testimage 'lenna'. Middle: Estimated image gradient using differentiated Gaussian filters. Right: Estimated gradient using a first degree polynomial model and zero certainty outside image border.

The full complexity only has to be applied near the border though. Elsewhere we have full certainty and the method actually reduces to the 'scale gradient' method because the matrix $(\mathbf{P}_1^T\mathbf{W}\mathbf{P}_1)^{-1}$ will then become a diagonal matrix. This last insight implies that there is a strong relation between polynomial expansion using a Gaussian applicability and computing image derivatives using 'derivatives at different scales'. This idea will be used in section 4.4 to create an efficient approximative polynomial expansion algorithm.

4.4 Approximative expansion using derivative filters

The first four derivatives of a Gaussian are in the one-dimensional case

$$\begin{aligned} g(x) &= \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} \\ g'(x) &= -\tfrac{x}{\sigma^2} g(x) \\ g''(x) &= \tfrac{x^2-\sigma^2}{\sigma^4} g(x) \\ g'''(x) &= \tfrac{3\sigma^2 x - x^3}{\sigma^6} g(x) \\ g''''(x) &= \tfrac{x^4 - 6\sigma^2 x^2 + 3\sigma^4}{\sigma^8} g(x) \end{aligned} \tag{4.20}$$

Partial derivatives in higher dimensionalities can easily be derived from the 1D-case due to the Cartesian separability of the Gaussian.

Let $\mathbf{D}_2$ denote another basis for the second degree polynomial space:

$$\mathbf{D}_2 = \begin{pmatrix} | & | & | & | & | & | \\ \mathbf{1} & -\mathbf{x} & -\mathbf{y} & \mathbf{x}^2 - \sigma^2 & \mathbf{y}^2 - \sigma^2 & \mathbf{xy} \\ | & | & | & | & | & | \end{pmatrix} \tag{4.21}$$

where σ is a scalar. If we multiply this basis with a Gaussian function, $\mathbf{g}$, with standard deviation σ we can from equation 4.20 see that it is closely related to partial derivatives of the Gaussian:

$$\mathbf{W}_a\mathbf{D}_2 = \begin{pmatrix} | & | & | & | & | & | \\ \mathbf{g} & \sigma^2\mathbf{g}_x & \sigma^2\mathbf{g}_y & \sigma^4\mathbf{g}_{xx} & \sigma^4\mathbf{g}_{yy} & \sigma^4\mathbf{g}_{xy} \\ | & | & | & | & | & | \end{pmatrix} \tag{4.22}$$

where $\mathbf{W}_a = \text{diag}(\mathbf{g})$ and $\mathbf{g}_x$, ..., $\mathbf{g}_{xy}$ denote the partial derivatives up to the second degree of the Gaussian.

The relation between the basis $\mathbf{D}_2$ and the second degree polynomial basis $\mathbf{P}_2$ in equation 4.5 is

$$\mathbf{D}_2 = \mathbf{P}_2\mathbf{T}_{\text{PD}} \tag{4.23}$$

where

$$\mathbf{T}_{\text{PD}} = \begin{pmatrix} 1 & 0 & 0 & -\sigma^2 & -\sigma^2 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.24}$$

If we put this relation into the polynomial expansion solution, equation 4.7, we obtain

$$\begin{aligned} \mathbf{r} &= (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{f} \\ &= (\mathbf{T}_{\text{PD}}^{-T}\mathbf{D}_2^T\mathbf{W}\mathbf{D}_2\mathbf{T}_{\text{PD}}^{-1})^{-1}\mathbf{T}_{\text{PD}}^{-T}\mathbf{D}_2^T\mathbf{W}\mathbf{f} \\ &= \mathbf{T}_{\text{PD}}(\mathbf{D}_2^T\mathbf{W}\mathbf{D}_2)^{-1}\mathbf{D}_2^T\mathbf{W}\mathbf{f} \end{aligned} \tag{4.25}$$

This means that we first fit the signal to the basis functions $\mathbf{D}_2$ and then transform the result to the model $\mathbf{f} \sim \mathbf{P}_2\mathbf{r}$ using $\mathbf{T}_{\text{PD}}$. The gain is that the basis functions in $\mathbf{D}_2$, corresponding to filters with which we correlate the image, can be approximated with a Gaussian filter followed by small derivative filters. The gain is most obvious for large σ or for high dimensionalities of the data. As we will see in section 4.5 it is also useful in multiscale expansion. But first we consider the cases of full certainty and uncertain data in one scale.

4.4.1 Full certainty

As in section 4.2.1 we have

$$\mathbf{W}_c = \mathbf{I} \tag{4.26}$$

and the solution reduces to

$$\mathbf{r} = \mathbf{T}_{\mathrm{PD}}(\mathbf{D}_2^T\mathbf{W}_a\mathbf{D}_2)^{-1}\mathbf{D}_2^T\mathbf{W}_a\mathbf{f} \tag{4.27}$$

$$\begin{aligned}
\mathbf{r} &= \mathbf{T}_{\mathrm{PD}}(\mathbf{D}_2^T\mathbf{W}_a\mathbf{D}_2)^{-1}\mathbf{D}_2^T\mathbf{W}_a\mathbf{f} \qquad (4.28)\\
&= \mathbf{T}_{\mathrm{PD}} \begin{pmatrix} 1 & & & & & \\ & \sigma^2 & & & & \\ & & \sigma^2 & & & \\ & & & 2\sigma^4 & & \\ & & & & 2\sigma^4 & \\ & & & & & \sigma^4 \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{1}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle -\mathbf{x}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle -\mathbf{y}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle (\mathbf{x}^2-\sigma^2)\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle (\mathbf{y}^2-\sigma^2)\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{f}\rangle \end{pmatrix}\\
&= \begin{pmatrix} 1 & & & -\frac{1}{2\sigma^2} & -\frac{1}{2\sigma^2} & \\ & -\frac{1}{\sigma^2} & & & & \\ & & -\frac{1}{\sigma^2} & & & \\ & & & \frac{1}{2\sigma^4} & & \\ & & & & \frac{1}{2\sigma^4} & \\ & & & & & \frac{1}{\sigma^4} \end{pmatrix} \begin{pmatrix} \langle \mathbf{1}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle -\mathbf{x}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle -\mathbf{y}\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle (\mathbf{x}^2-\sigma^2)\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle (\mathbf{y}^2-\sigma^2)\cdot\mathbf{g},\ \mathbf{f}\rangle \\ \langle \mathbf{xy}\cdot\mathbf{g},\ \mathbf{f}\rangle \end{pmatrix}
\end{aligned}$$

$\mathbf{D}_2^T\mathbf{W}_a\mathbf{f}$ means that we correlate the image with the filters

$$g\ ,\ \sigma^2 g_x\ ,\ \sigma^2 g_y\ ,\ \sigma^4 g_{xx}\ ,\ \sigma^4 g_{yy}\ ,\ \sigma^4 g_{xy} \tag{4.29}$$

These filters can be approximated with a Gaussian filter followed by derivative filters:

$$\begin{aligned}
\sigma^2 g_x &\approx g * d_x \\
\sigma^2 g_y &\approx g * d_y \\
\sigma^4 g_{xx} &\approx g * d_x * d_x \\
\sigma^4 g_{yy} &\approx g * d_y * d_y \\
\sigma^4 g_{xy} &\approx g * d_x * d_y
\end{aligned} \tag{4.30}$$

where d_x and d_y are one-dimensional derivative filters along the x- and y-dimension respectively.

Figure 4.4 contains the correlator structure needed to compute $\mathbf{D}_2^T\mathbf{W}_a\mathbf{f}$. The result from the correlator structure is multiplied with $\mathbf{T}_{\mathrm{PD}}(\mathbf{D}_2^T\mathbf{W}_a\mathbf{D}_2)^{-1}$ in each local neighborhood to get the final solution $\mathbf{r}$.

This correlator structure only needs 2 Gaussian 1D filters of length n (n depends on σ) and additional 5 derivative filters. This should be compared to 9 1D filters of length n in the correlator structure in figure 4.1.

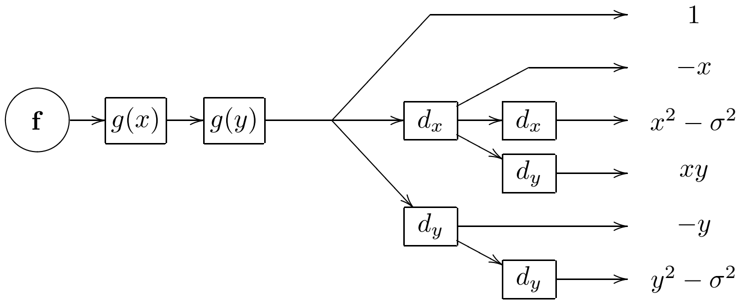


Figure 4.4: Correlator structure for approximative polynomial expansion in 2D with full certainty.

4.4.2 Uncertain data

As in section 4.2.2 we have to compute the general solution

$$\mathbf{r} = \mathbf{T}_{\mathrm{PD}}(\mathbf{D}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{D}_2)^{-1}\mathbf{D}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{f} \tag{4.31}$$

We have to use one correlator structure to compute $\mathbf{D}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{f}$ and one structure to compute $\mathbf{D}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{D}_2$. They can both be approximated with a Gaussian filter followed by small derivative filters, see figure 4.5.

This case is not further investigated in this thesis but stated here for completeness. There could be some numerical problems with the approximations regarding the higher order terms and the inversion of the matrix $\mathbf{D}_2^T\mathbf{W}_a\mathbf{W}_c\mathbf{D}_2$.

4.5 Multiscale polynomial expansion

The new approximative polynomial expansion algorithm described above is easily generalized to multiscale polynomial expansion. The correlator structure in figure 4.4 consists of a Gaussian filter, $\mathbf{g}$, followed by a derivative structure, ∂. If we have full certainty and want to fit the polynomial model in several scales we can simply compute a lowpass hierarchy using Gaussian filters and then in each scale attach a derivative structure ∂, see figure 4.6. Finally we compute the solution

$$\mathbf{r} = \mathbf{T}_{\mathrm{PD}}(\mathbf{D}_2^T\mathbf{W}_a\mathbf{D}_2)^{-1}\mathbf{D}_2^T\mathbf{W}_a\mathbf{f} \tag{4.32}$$

in each scale using the results from the correlator structure. Figure 4.7 shows an example of the resulting filters created from the multiscale structure.

In the general case of uncertain data we should attach both the correlator structures in figure 4.5 in each scale of the lowpass hierarchy. This requires a lot of computations. A greedy approximation of the general case could be to use uncertainty only when computing the lowpass hierarchy, i.e. in each scale and each

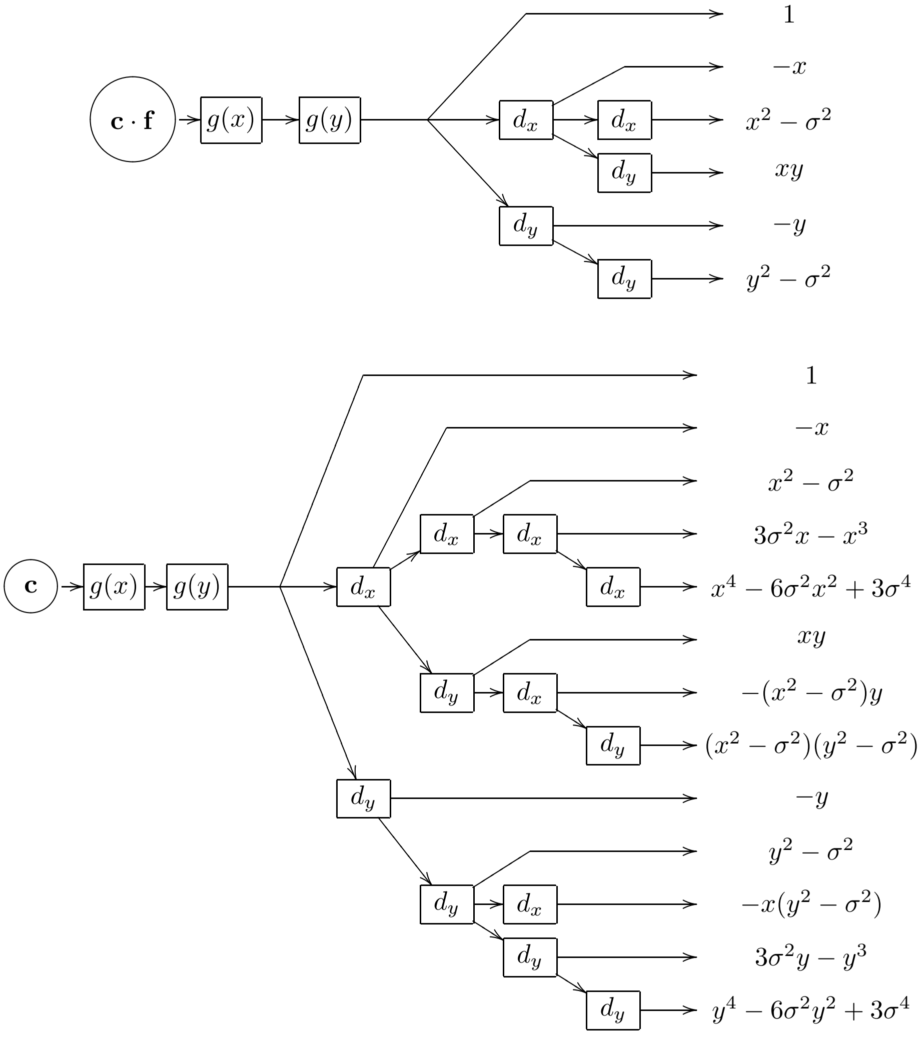


Figure 4.5: Correlator structure for approximative polynomial expansion in 2D with uncertain data.

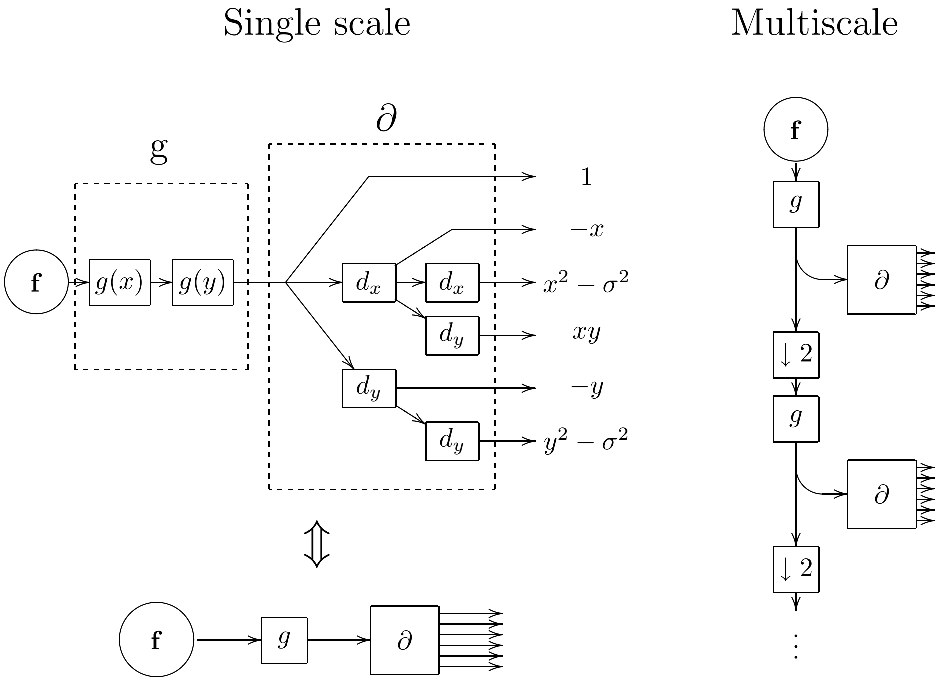


Figure 4.6: Correlator structures for single scale and multiscale approximative polynomial expansion in 2D with full certainty. ↓ 2 means down-sampling by a factor 2.

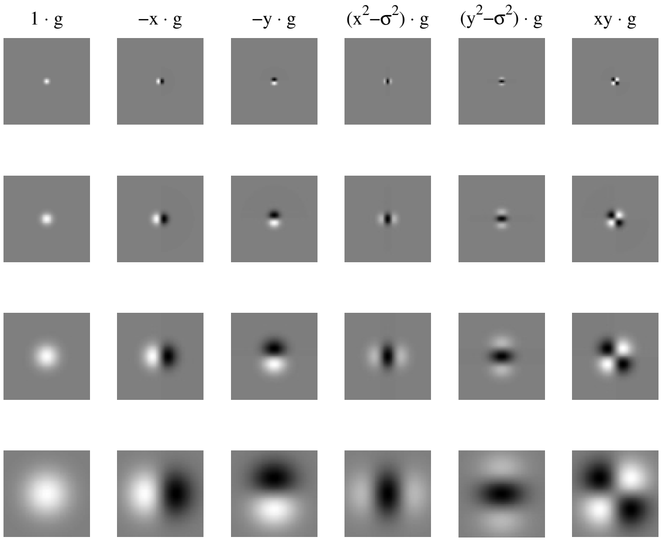


Figure 4.7: Basis functions in four scales generated by the multiscale correlator structure. Black and white colors indicate negative and positive values respectively.

local neighborhood we calculate

$$\frac{\langle \mathbf{g},\ \mathbf{c} \cdot \mathbf{f} \rangle}{\langle \mathbf{g},\ \mathbf{c} \rangle} \tag{4.33}$$

which is then used as input to the derivative structure ∂ in figure 4.6.

Comment:
It may be possible to design a multiscale algorithm by correlating the result from the lowpass hierarchy with the polynomial basis **P** instead of the basis **D**. One problem is that if we for instance correlate a Gaussian g_{σ_1} with the filter $x^2 g_{\sigma_2}$ we do not get a resulting filter $x^2 g_{\sigma_3}$ but instead we obtain

$$g_{\sigma_1} * x^2 g_{\sigma_2} = \frac{\sigma_1^2 \sigma_3^2}{\sigma} g_{\sigma_3} + \frac{\sigma_1^4}{\sigma_3^4} x^2 g_{\sigma^3} \quad \text{where} \quad \sigma_3 = \sqrt{\sigma_2^2 + \sigma_2^2} \tag{4.34}$$

This means that we have a different basis at each scale. This is not a problem for the basis **D** because if we correlate the filters **WD** with a Gaussian we get the same filters but with a larger Gaussian applicability **W**.

4.6 Practical issues

4.6.1 Filter optimization

The derivative filters d_x and d_y in figure 4.4 can be optimized in a number of ways. A very simple way is to use the 'derivatives at different scales'-idea described in section 4.3 and use $d_x = -\frac{x}{\sigma_d^2} g_{\sigma_d}$ and $d_y = -\frac{y}{\sigma_d^2} g_{\sigma_d}$ to compute approximative partial derivatives. This does not give a very good approximation if the filters are small. Another, more powerful method, is to optimize the whole correlator structure in figure 4.4 with respect to some suitably chosen goal function. The theoretical basis for these kind of optimizations can be found in e.g. [Andersson et al., 1999], [Knutsson et al., 1999]. Ideal output functions specified in both the spatial domain and in the Fourier domain are provided by the user together with filter masks that define the appearance of the filters, i.e. which filter coefficients that are allowed to be used in the optimization. All the filter coefficients in the structure are then optimized to match the spatial and the Fourier domain ideal functions. The filters usually get good spatial location in both domains which promise good behavior of the structure. This method is however not further explored in this thesis and is a topic for future research. In the experiments in section 6.1 we choose an intermediate solution:

Given the ideal output filter **xg** and an input filter **g** we can optimize an approximation of a derivative filter **d** by minimizing the error function

$$\|\mathbf{xg} - \mathbf{g} * \mathbf{d}\|^2 \tag{4.35}$$

with respect to $\mathbf{d} \in \mathbb{R}^p$, where p specify the filter size. The solution is found by solving a linear equation system. The second derivative can then be optimized using the same error function but with $\mathbf{x}^2\mathbf{g}$ as ideal output and $\mathbf{g} * \mathbf{d}$ as input filter. This optimization method is quite simple and the details are therefore omitted.

Note that special care should be taken for optimization of the filters in the multiscale structure in figure 4.6. The filters should, due to the down-sampling function, be thought of as being 'spread out' in the original image. This means that if we for example correlate with the filter

$$d = [\ -1 \quad -2 \quad 0 \quad 2 \quad 1\ ] \tag{4.36}$$

after down-sampling, the corresponding filter without down-sampling is

$$d = [\ -1 \quad 0 \quad -2 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 1\ ] \tag{4.37}$$

and the optimization should therefore be made using the 'filter mask'

$$d = [\ x \quad 0 \quad x \quad 0 \quad x \quad 0 \quad x \quad 0 \quad x\ ] \tag{4.38}$$

where x denotes non-zero coefficients.

4.6.2 Minimizing the approximation error

In practice we only have an approximation of $\mathbf{D}$, denoted $\widetilde{\mathbf{D}}$. $\widetilde{\mathbf{D}}$ and $\mathbf{P}$ do not span the same subspace so we can only use an approximate transformation matrix, $\widetilde{\mathbf{T}}_{\text{PD}}$ between the two basis sets. The polynomial expansion solution becomes

$$\tilde{\mathbf{r}} = \widetilde{\mathbf{T}}_{\text{PD}}(\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{f} \tag{4.39}$$

One way to estimate $\widetilde{\mathbf{T}}_{\text{PD}}$ is by demanding that all signals $\mathbf{f}$ in the second degree polynomial subspace should get correctly estimated parameters, i.e.

$$\tilde{\mathbf{r}} = \mathbf{r} \text{ for } \mathbf{f} = \mathbf{P}\mathbf{r}\ ,\ \mathbf{r} \in \mathbb{R}^6 \ \Rightarrow\ \widetilde{\mathbf{T}}_{\text{PD}} = (\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{P})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}} \tag{4.40}$$

Another, more general, statistical approach is to minimize

$$\mathrm{E}\left[\|\tilde{\mathbf{r}} - \mathbf{r}\|^2\right] = \mathrm{E}\left[\|\widetilde{\mathbf{T}}_{\text{PD}}(\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{f} - (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}\mathbf{f}\|^2\right] \tag{4.41}$$

with respect to $\widetilde{\mathbf{T}}_{\text{PD}}$ over a set of signals $\mathbf{f}$. E[.] denotes expectation value over the set. To simplify the computations we assume $\mathrm{E}[\mathbf{f}\mathbf{f}^T] \propto \mathbf{I}$ and the solution, derived in appendix B, becomes

$$\widetilde{\mathbf{T}}_{\text{PD}} = (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2\widetilde{\mathbf{D}}(\widetilde{\mathbf{D}}^T\mathbf{W}^2\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}} \tag{4.42}$$

Hence, after computing $\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{f}$ using the correlator structure in figure 4.4 we compute the model parameters by multiplying with the matrix

$$\widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1} = (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2\widetilde{\mathbf{D}}(\widetilde{\mathbf{D}}^T\mathbf{W}^2\widetilde{\mathbf{D}})^{-1} \tag{4.43}$$

In the worst case approximation we should still have even and odd basis functions so the three part matrices in equation 4.43 all have the structure

$$\begin{pmatrix} x_1 & & & x_2 & x_2 & \\ & x_3 & & & & \\ & & x_3 & & & \\ x_4 & & & x_5 & x_6 & \\ x_4 & & & x_6 & x_5 & \\ & & & & & x_7 \end{pmatrix} \tag{4.44}$$

where x_k denote non-zero elements. The matrix $\widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1}$ will therefore also have this structure.

4.7 Computational complexity

Table 4.1 contains time complexity and memory overhead for the different versions of the polynomial expansion algorithm described above. Included are also the case three-dimensional data with full certainty. This case will be used in an experiment in section 6.1.2.

Method		Time complexity	Memory overhead
2D, full cert	True	9n+10	6
	Approx	2n+5m+12	6
2D, uncert	True	29n+92	21
	Approx	4n+17m+92	21
3D, full cert	True	19n+16	10
	Approx	3n+9m+22	10

Table 4.1: Computational complexity for the polynomial expansion algorithms. n = size of local area, m = size of derivative filters.

The terms dependent of n and m in the time complexity are the total number of coefficients involved in the correlation structure. In the full certainty cases, the second term refer to the operations needed to multiply the correlation result with the matrices $(\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}$ and $(\mathbf{D}^T\mathbf{W}\mathbf{D})^{-1}$ respectively to get the model parameters. The sparsity of these matrices are taken into account. In the 2D uncertainty case the model parameters are instead computed by solving a 6×6 symmetric positive definite equation system, estimated at $\frac{1}{6}6^3 + \frac{3}{2}6^2 + \frac{1}{3}6 = 92$ operations.

4.8 Conclusions

There is a strong relationship between local polynomial expansion and derivatives at different scales, which has been taken advantage of in this chapter to design a very efficient approximative local polynomial expansion algorithm. The gain of using the approximative algorithm is most evident when the local area is large and/or the signal dimensionality is high. It also gives a very efficient computational structure when the model is fitted in several scales.

Section 6.1 contains two experiments of the approximative polynomial expansion algorithm described in this chapter. The algorithm appears to perform well for $\sigma > 1$ if the size of the derivating filters is at least 5. There is not much difference between using the two estimations $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ in equations 4.41 and 4.40. It seems that the simpler estimate based on the polynomial subspace is slightly better on average, but has a somewhat higher maximum error than the statistical estimate.

The introduction 4.1 mentioned another efficient approximative algorithm by Burt ([Burt, 1988]). A comparison between the two algorithms remains to be done. The approaches differ but the result may be similar. One difference is that the filter kernels used by Burt have integer values and the resulting applicability will therefore be less isotropic. Also, Burts algorithm did not allow for the scale σ to be chosen freely, but the algorithm can probably be modified to have that feature.

Chapter 5

Rotational symmetries

5.1 Introduction

Human vision seems to work in a hierarchical way in that we first extract low level features such as local orientation and color and then higher level features [Bear et al., 1996]. There also seem to exist lateral interactions between cells, perhaps to make them more selective. No one knows for sure what these high level features are but there are some indications that curvature, circles, spiral, and star patterns are among them [Gallant et al., 1993]. Indeed, perceptual experiments indicate that corners and curvature are very important features in the process of recognition and one can often recognize an object from its curvature alone [Attneave, 1954], [Biederman, 1987]. They have a high degree of specificity and sparsity and as they are point features, they do not suffer from the aperture problem usually encountered for line and edge structures [Granlund and Knutsson, 1995]. Also, these kind of features are more or less invariant to many geometrical transformations, for example circle- and star-patterns are invariant to rotation, zooming and high curvature points are to some extent invariant to change of view.

This chapter introduce tools and procedures to model and detect the features mentioned above. The features are described using the theory of rotational symmetries which define curvature from local orientation.

The procedure to detect these features works in a hierarchical way; first, a local orientation image in double angle representation is calculated. This image is more invariant to intensity which simplifies curvature detection. Second, curvature is detected using the local orientation image.

This chapter also suggests two techniques to make the responses more selective which will simplify subsequent processing. One idea is to use lateral inhibitions between responses, inspired from biological vision.

Real applications using rotational symmetries have been few, perhaps because of the lack of efficient computational structures. This chapter also describe a method to detect the features very efficiently in several scales. A local polynomial expansion model (see chapter 4) is computed on the local orientation and the rotational symmetries is detected using the model parameters.

The chapter is outlined as follows: Section 5.2 introduce the theory. Section 5.3 describes tools for selective and efficient detection of the symmetry features. Background and previous work on rotational symmetries is reviewed in section 5.4, when appropriate terminology has been introduced. Finally, the chapter is concluded in section 5.5.

Experiments to evaluate the algorithm described in this section can be found in the experiment chapter. Section 6.2 evaluates the rotational symmetry algorithm on noisy images of rotational symmetries. Section 6.4 evaluates the use of curvature features in an object recognition experiment on the Columbia Image Object Database (COIL-100). Section 6.5 discusses and reviews other applications.

5.2 Basic theory

5.2.1 Local orientation in double angle representation

A classical representation of local orientation is simply a 2D-vector pointing in the dominant direction. An example of this is the image gradient. Another representation is the *double angle representation* where we have a complex number, z, with a phase that is double the local orientation angle, see [Granlund, 1978]. In other words, if the orientation has the direction θ we represent it with a vector pointing in the 2θ-direction, i.e. $z = e^{i2\theta}$. Figure 5.1 illustrates the idea.

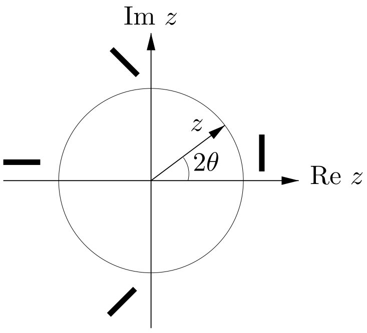


Figure 5.1: Double angle representation of local orientation.

This representation has at least two advantages:

- We avoid ambiguities in the representation of even signals. It does not matter if we choose to say that the orientation has the direction θ or, equivalently, $\theta+\pi$. In the double angle representation both choices get the same descriptor $e^{i2\theta}$.

- Averaging the double angle orientation description field makes sense. One can argue that two orthogonal orientations should have maximally different representations, e.g. vectors that point in opposite directions. This is for instance useful in color images and vector fields when we want to fuse descriptors from several channels into one description.

The double angle descriptor will throughout this thesis be denoted by a complex valued function z.

In the experiments in section 6.4 we will use the polynomial based gradient estimation in the example in section 4.3 to detect edges. From the estimated gradient, $\nabla f = (f_x, f_y)$, a double angle representation can be computed as

$$z = |\nabla f|^{\gamma} e^{i2\arctan(f_y/f_x)} = |\nabla f|^{\gamma-2}(f_x + if_y)^2 \tag{5.1}$$

The γ-parameter controls the energy sensitivity.

As an illustration, the result of using this method on a simple binary image is shown in the top row in figure 5.9. z will be used to detect complex curvature features such as corners, circles, and star-patterns. The rotational symmetry theory in the next section describes complex curvature in terms of local orientation.

5.2.2 Rotational symmetries

Figure 5.2 shows a circle- and a star-pattern together with their corresponding local orientation description in double angle representation. Notice that they have the same phase variation, except for a constant difference π.

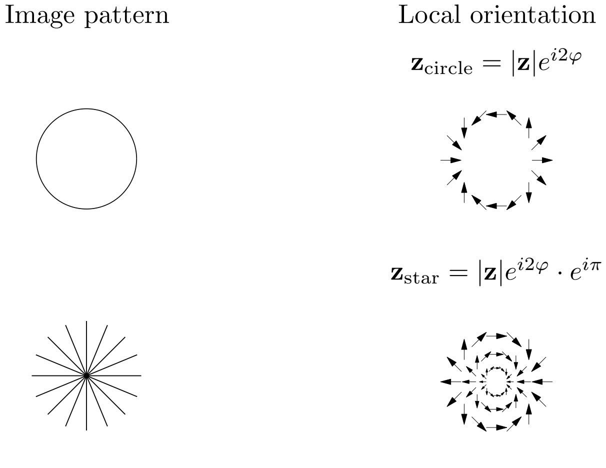


Image pattern
Local orientation
$\mathbf{z}_{\text{circle}} = |\mathbf{z}|e^{i2\varphi}$
$\mathbf{z}_{\text{star}} = |\mathbf{z}|e^{i2\varphi} \cdot e^{i\pi}$

Figure 5.2: Two image patterns, a circle and a star, together with their corresponding local orientation description in double angle representation.

The circle and star patterns belong to a family of patterns called *rotational symmetries*. They are defined as patterns with a local orientation phase that only depends on φ:

Rotational symmetry

A signal $f(r, \varphi)$ is called a rotational symmetry if $\hat{z} = z/|z|$ only depends on φ, where z is the local orientation description in double angle representation of the signal f. (5.2)

A special case of symmetry patterns are the *n:th order rotational symmetries*:

n:th order rotational symmetry

$$z(r, \varphi) = c(r, \varphi)e^{i(n\varphi+\alpha)} \quad , \quad \begin{array}{l} c(r, \varphi) \in \mathbb{R} \\ \alpha \in [-\pi, \pi] \end{array} \tag{5.3}$$

For example, the circle and star belongs to the second order symmetries. Each symmetry order n describe a whole class of patterns. Each α represent a member in the class.

What image patterns corresponds to the local orientation description $c\, e^{i(n\varphi+\alpha)}$? To answer this, one has to start with the local orientation image $z(r, \varphi) = c\, e^{i(n\varphi+\alpha)}$ and go backwards to the original image, f, from which the orientation image was calculated, i.e.

Double angle representation z $\Rightarrow$ **Grayscale image** f

This 'inverse' is of course not unique, we lost information about the sign of the image gradient when we computed the double angle. But we get a hint by making the assumption that the image gradient is parallel to the dominant orientation, $\nabla f = \pm|\nabla f|\, e^{i(n\varphi+\alpha)/2}$. (We have to divide the phase by two to get rid of the double angle representation, the price is the direction ambiguity.) This will give a differential equation system that can be solved assuming polar separability. The solution is derived in appendix A. Figure 5.3 contains a sample of these psychedelic patterns. Note that the patterns have a fairly constant magnitude $|z|$ (assuming suitable scale of the local neighborhood). n:th order symmetry patterns with varying magnitude can for instance be created from the patterns in figure 5.3 by taking one or several trajectories.

The perhaps most useful symmetries, i.e. the most common visual patterns in our daily lives, are the

- 0:th order: Describes lines.
- 1:st order: Describes hyperbolic patterns, e.g. curvature and line-endings. (They are also called parabolic symmetries).
- 2:nd order: Describes circles, stars, and spiral patterns. (Also called circular symmetries.)

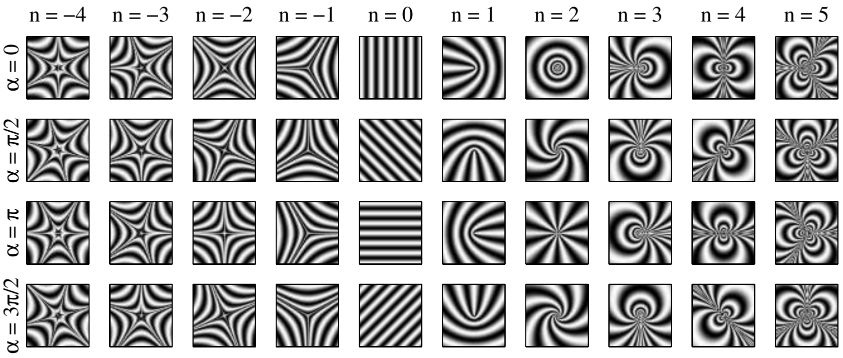


Figure 5.3: Some examples of n:th order rotational symmetry patterns, corresponding to a local orientation description $z = e^{i(n\varphi+\alpha)}$. The patterns are derived in appendix A.

Some examples of these patterns are shown in figure 5.4. Note that all patterns in a row has the same orientation description when the magnitude $|z|$ is disregarded.

There are other rotational symmetry patterns than the n:th order symmetries that might be of interest. Figure 5.5 contain some examples of general rotational symmetry patterns. The normalized orientation descriptor, $\hat{z}(\varphi)$ can be expressed as a linear combination of the n:th order symmetry descriptions,

$$\hat{z}(\varphi) \sim \sum_n s_n e^{in\varphi} \tag{5.4}$$

This is equivalent to a discrete Fourier transform of the periodic function $\hat{z}(\varphi)$. Some of the s_n coefficients and an approximative reconstruction of the pattern using only s_n, $n = -2, -1, ..., 2$, is also shown in figure 5.5.

5.3 Detection of rotational symmetries

We now turn to the problem of how to detect the rotational symmetries described in the previous section. Section 5.3.1 introduces the problems. Section 5.3.2 and 5.3.3 describes two techniques to improve selectivity. Finally section 5.3.4 combine the selectivity technique with the polynomial expansion theory described in chapter 4 to create an efficient multiscale detection algorithm.

5.3.1 Introduction

In practice the want to find patterns in a limited area of size $n \times n$ in a pixel-discretized local orientation image. After reshaping the local signal z and the polar coordinates r,φ into vectors we can describe them as elements in $\mathbb{C}^{n^2}$ and denote them as $\mathbf{z}$, $\mathbf{r}$, and $\boldsymbol{\varphi}$.

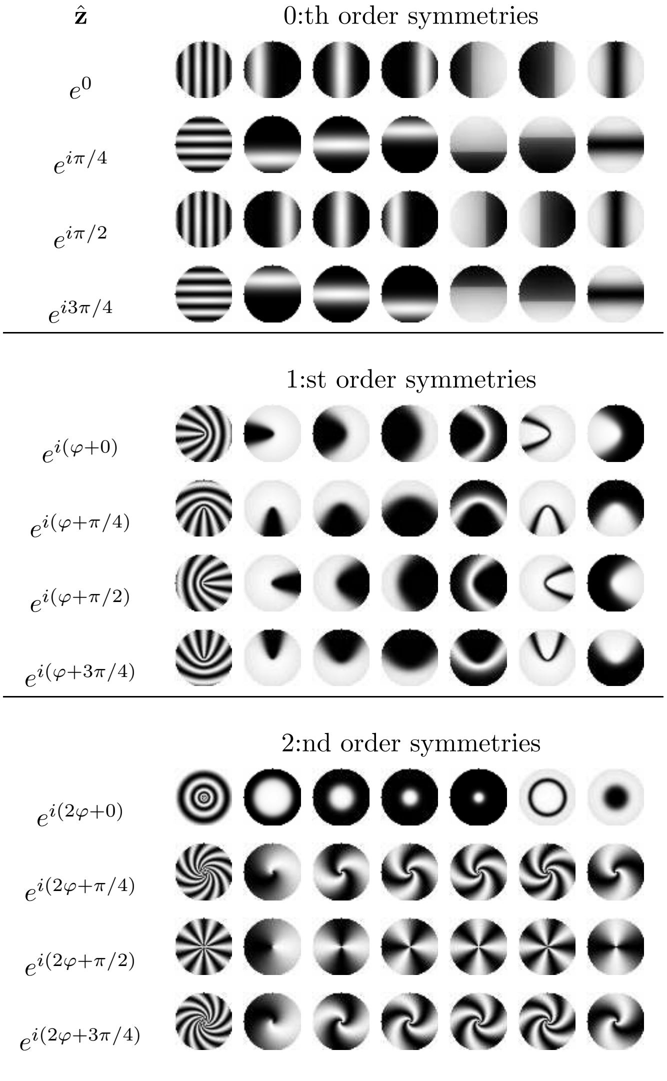


Figure 5.4: Some examples of 0:th, 1:st, and 2:nd order rotational symmetry patterns and the corresponding local orientation descriptions.

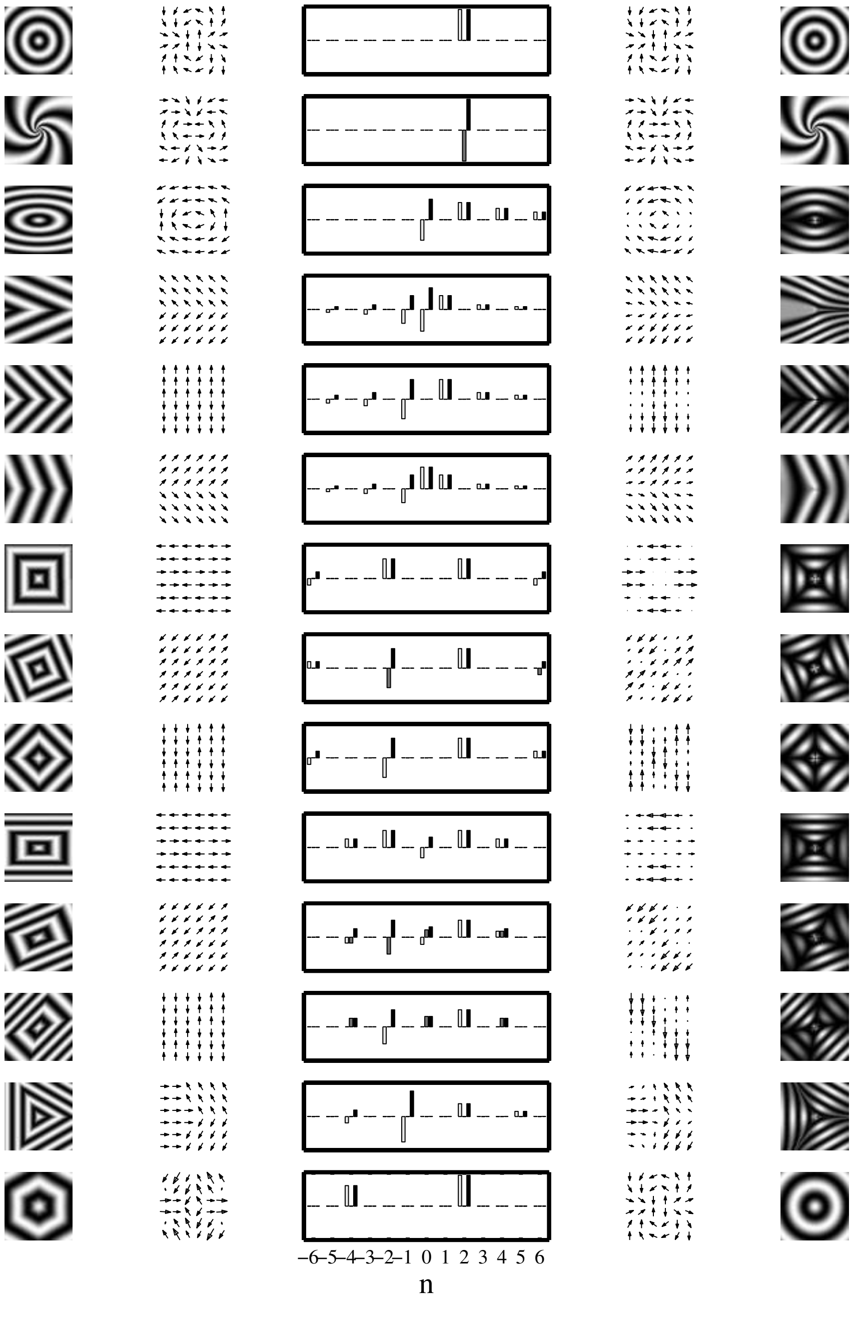


Figure 5.5: Examples of image patterns and their rotational symmetry description. Column 1: image pattern. Column 2: Corresponding local orientation in double angle representation. Column 3: Rotational symmetry components s_n (white = $\Re\{s_n\}$, gray = $\Im\{s_n\}$, black = $|s_n|$). Column 4: $\sum_{n=-2}^{2} s_n e^{i n\varphi}$. Column 5: Back-projection from column 4.

A simple way to detect an n:th order rotational symmetry pattern is to correlate $\mathbf{z}$ with a filter $\mathbf{a}\cdot\mathbf{b}_n$, where $\mathbf{a}$ is an applicability function serving as a window for the basis function $\mathbf{b}_n = e^{in\varphi}$ (sometimes called *circular harmonic function*). Figure 5.6 shows the three filters $\mathbf{a}\cdot\mathbf{b}_0$, $\mathbf{a}\cdot\mathbf{b}_1$, and $\mathbf{a}\cdot\mathbf{b}_2$ where $\mathbf{a}$ is chosen as a Gaussian function. The standard deviation, σ, of the Gaussian controls the size of the symmetries we want to detect.

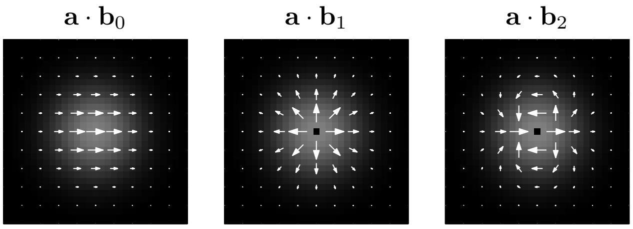


Figure 5.6: Filters $\mathbf{a}\cdot\mathbf{b}_n$ needed to detect the 0:th, 1:st, and 2:nd order rotational symmetries.

The correlation means that in each local neighborhood we calculate the scalar products

$$\left\{\begin{array}{lcl} s_0 & = & \langle \mathbf{a}\cdot\mathbf{b}_0,\ \mathbf{z}\rangle \\ s_1 & = & \langle \mathbf{a}\cdot\mathbf{b}_1,\ \mathbf{z}\rangle \\ s_2 & = & \langle \mathbf{a}\cdot\mathbf{b}_2,\ \mathbf{z}\rangle \end{array}\right. \quad \Longleftrightarrow \quad \mathbf{s} = \mathbf{B}^*\mathbf{W}_a\mathbf{z} \tag{5.5}$$

where

$$\mathbf{s} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix} \quad , \quad \mathbf{B} = \begin{pmatrix} | & | & | \\ \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 \\ | & | & | \end{pmatrix} \tag{5.6}$$

and $\mathbf{W}_a = \mathrm{diag}(\mathbf{a})$. If we have an image pattern $\mathbf{z} = |\mathbf{z}|e^{i(n\varphi+\alpha)}$ we will get $s_n = \langle \mathbf{a},\ |\mathbf{z}|\rangle e^{i\alpha}$. Therefore, high magnitude $|s_n|$ indicate an n:th order rotational symmetry and the phase $\angle s_n$ points out the class member.

The middle row in figure 5.9 shows the result when correlating the local orientation z in the top row with the filters in figure 5.6. The standard deviation σ is chosen as about half the size of the circle pattern.

Note that the responses are not as selective as one might want them to be. For example, a line in the outer area of the local neighborhood gives a high magnitude in all three responses s_0, s_1, s_2 because this pattern is approximately part of all three classes (see for instance column $n = 1$, and $n = 2$ in figure 5.3 - the trajectories in the outer area are all approximately lines). Another example is that we get high second order responses for corners, because they are approximately 'half circles'. In this thesis the goal is to find curvature, corners, circles, etc. The first and second rotational symmetry responses can be one way to detect these features, but it would be nice if they were more selective so that the first order response gave lower responses to lines and edges and that the second order response was low for both lines and corners. Sections 5.3.2 and 5.3.3 discusses two different methods to make the responses more selective. The first method models the local orientation as a linear combination of the n:th order symmetries. The second one is a fuzzy classification approach.

5.3.2 Improved selectivity using normalized convolution

This technique is also described in [Johansson et al., 2000]. The idea is to use the normalized convolution theory discussed in section 3.1 to make the responses more selective. We simply use $\mathbf{b}_n$ as basis functions and $\mathbf{c} = |\mathbf{z}|$ as signal certainty. The goal is then to approximate the signal $\hat{\mathbf{z}} = \mathbf{z}/|\mathbf{z}|$ with a linear combination of the basis functions, i.e.

$$\hat{\mathbf{z}} \sim s_0\mathbf{b}_0 + s_1\mathbf{b}_1 + s_2\mathbf{b}_2 = \mathbf{Bs} \tag{5.7}$$

using the certainty $\mathbf{c}$ and an applicability $\mathbf{a}$ as weight functions for the signal and basis functions respectively. The solution becomes

$$\mathbf{s} = (\mathbf{B}^*\mathbf{WB})^{-1}\mathbf{B}^*\mathbf{W}\hat{\mathbf{z}} = (\mathbf{B}^*\mathbf{WB})^{-1}\mathbf{B}^*\mathbf{W}_a\mathbf{z} \quad \text{where} \quad \mathbf{W} = \mathbf{W}_a\mathbf{W}_c \tag{5.8}$$

As output certainty we use equation 3.6. The top row in figure 5.10 contains the result $c_{out} \cdot s_n$ when using a Gaussian applicability with the same standard deviation as before. The result is now more selective and sparse. It is important to use the magnitude $|\mathbf{z}|$ as certainty and not part of the signal. If we had used a constant certainty and $\mathbf{z}$ as signal we would arrive at the same result as in the previous section (middle row in figure 5.9). This is because the basis functions are orthogonal in full certainty and therefore equation 5.8 simplifies to equation 5.5.

The normalized convolution technique not only improves selectivity. It can also be used to describe other patterns than the n:th order symmetries. As mentioned in section 5.2.2, the patterns in figure 5.5 can be approximately described as a linear combination of a small number of n:th order symmetries. It should be possible to for instance use $\mathbf{b}_n$, $n = -2, -1, ..., 2$, as basis functions and calculate the coefficients s_n with the normalized convolution technique described here to find squares, triangles, corners and their corresponding corner angles, etc. Experiments to confirm this is however left as future work.

$\mathbf{B}^*\mathbf{Wz}$ implies that we compute the correlations

$$(\mathbf{a} \cdot \mathbf{b}_n) \star \mathbf{z} \quad , \quad n = 0, 1, 2 \tag{5.9}$$

and $\mathbf{B}^*\mathbf{WB}$ means that the compute the correlations

$$(\mathbf{a} \cdot \mathbf{b}_m \cdot \bar{\mathbf{b}}_n) \star \mathbf{c} = (\mathbf{a} \cdot \mathbf{b}_{m-n}) \star |\mathbf{z}| \quad , \quad m - n = -2, ..., 2 \tag{5.10}$$

The computational complexity involved may be too high for some applications. The next section describes a more efficient method to create selective responses.

5.3.3 Improved selectivity using normalized inhibition

This technique is also described in [Johansson and Granlund, 2000] and it is also related to the idea of characteristic phases, see [Granlund, 1999]. To avoid some of the computations involved in the normalized convolution method we can instead let the filter responses in equation 5.5 inhibit, or punish, each other so that if one response is high the other ones becomes lower. This can be though of as a fuzzy classification into one of the n:th order symmetries. Before we inhibit we normalize with the amount of certainty, i.e

$$s_n = \frac{\langle \mathbf{a} \cdot \mathbf{b}_n,\ \mathbf{z} \rangle}{\langle \mathbf{a},\ |\mathbf{z}| \rangle} \tag{5.11}$$

This is actually equivalent to the normalized convolution result in the previous section when only one basis function is used. The normalization ensures (easy to show) that the magnitude of the response lies between 0 and 1,

$$0 \leq |s_n| \leq 1 \tag{5.12}$$

This property gives a simple inhibition rule. The normalized inhibition is computed as ('p' is short for 'punish')

$$\begin{array}{rcl} |s_0^p| & = & |s_0|\,(1-|s_1|)\,(1-|s_2|) \\ |s_1^p| & = & |s_1|\,(1-|s_0|)\,(1-|s_2|) \\ |s_2^p| & = & |s_2|\,(1-|s_0|)\,(1-|s_1|) \end{array} \tag{5.13}$$

As output certainty we can use

$$c_{out} = \langle \mathbf{a},\ |\mathbf{z}| \rangle \tag{5.14}$$

(which coincide with the output certainty from the normalized convolution method when only one basis function is used).

The result of this method when applied to the same simple test image as before is shown in the second row in figure 5.10. The response is more selective compared to the normalized convolution method, on the other hand it cannot describe patterns as linear combinations of the n:th order symmetries. The inhibition method is also much faster to calculate than the normalized convolution method described in the previous section. We still have to compute the correlations in equation 5.9, but instead of the correlations in equation 5.10 we only have to compute the output certainty $\mathbf{a} \star |\mathbf{z}|$.

However, the filters are still two-dimensional and if we want to find fairly large image pattern the filters also get large. The next section deals with the subject of efficient detection.

5.3.4 Efficient detection using polynomial expansion

Introduction

One problem with the methods described so far is that it is difficult to efficiently calculate the results. For practical use it is often important that the algorithm is as fast as possible. The filters can, for some choices of applicability, be made approximately separable into a small number of 1D-filters using SVD (Singular Value Decomposition), see [Johansson and Granlund, 2000], or if we want to detect the symmetries in several scales, create a filter-net using a technique similar to the one described in [Andersson et al., 1999], [Knutsson et al., 1999] (the results from this is not yet reported). The design of the kernel masks in the filter-net is not automatic but involves a great deal of engineering which makes it inflexible and the SVD approach is difficult to efficiently generalize to multiscale filtering.

This section describes how the rotational symmetries can be detected using the polynomial expansion algorithm in chapter 4. This approach creates less practical problems than the approaches mentioned above. The singlescale/multiscale filtering is very efficient and the filter kernels can be automatically created, which makes this method more flexible. The method is based on the observation that if the basis functions $b_n = e^{in\varphi}$ are weighted with r^n we get polynomials:

$$\left\{ \begin{array}{rcrcrcl} b_0 & = & e^{i0\varphi} & = & (x+iy)^0 & = & 1 \\ rb_1 & = & re^{i1\varphi} & = & (x+iy)^1 & = & x+iy \\ r^2 b_2 & = & r^2 e^{i2\varphi} & = & (x+iy)^2 & = & x^2 - y^2 + i2xy \end{array} \right. \tag{5.15}$$

Note that $r^n b_n$ is still a member of the n:th order symmetry class. This means that there is a close relation between the information that can be described by the basis functions b_n and the polynomial coefficients respectively. It is therefore possible to use the polynomial model parameters to detect rotational symmetries.

Single scale detection

Assume we have a local polynomial expansion model of the local orientation, using normalized convolution with full certainty (section 4.2.1):

$$\mathbf{z} \sim \mathbf{P}\mathbf{r}_z \quad , \quad \text{where } \mathbf{r}_z = (\mathbf{P}^T \mathbf{W}_a \mathbf{P})^{-1} \mathbf{P}^T \mathbf{W}_a \mathbf{z} \tag{5.16}$$

($\mathbf{r}_z$ should not be confused with the polar coordinate r). The correlations in equation 5.5 can then be approximated from the polynomial model parameters:

$$\begin{array}{rcl} \mathbf{s} & = & \mathbf{B}^* \mathbf{W}_a \mathbf{z} \\ & \sim & \mathbf{B}^* \mathbf{W}_a \mathbf{P} \mathbf{r}_z = \begin{pmatrix} 1 & 0 & 0 & \sigma^2 & \sigma^2 & 0 \\ 0 & \sigma\sqrt{\frac{\pi}{8}} & -i\sigma\sqrt{\frac{\pi}{8}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sigma^2}{2} & -\frac{\sigma^2}{2} & -i\frac{\sigma^2}{2} \end{pmatrix} \mathbf{r}_z \end{array} \tag{5.17}$$

Now we do not have to correlate the local orientation $\mathbf{z}$ with the filter kernels $\mathbf{WB}$. Instead we can compute a polynomial model, which can be done quite efficiently, and then use the polynomial parameters $\mathbf{r}_z$ to compute an approximation of $\mathbf{s}$.

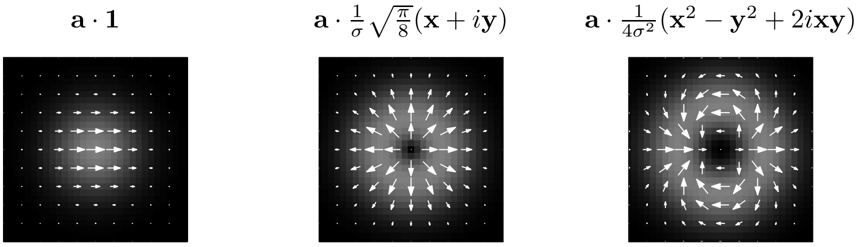
To interpret the result we notice that both the polynomial expansion and the transformation from $\mathbf{r}_z$ to $\mathbf{s}$ are linear operations which can be merged into one single linear operation. From equation 5.16 and 5.17 we have that

$$\mathbf{B}^* \mathbf{W}_a \mathbf{P} \mathbf{r}_z = \mathbf{B}^* \mathbf{W}_a \mathbf{P} (\mathbf{P}^T \mathbf{W}_a \mathbf{P})^{-1} \mathbf{P}^T \mathbf{W}_a \mathbf{z} \tag{5.18}$$

This means that $\mathbf{B}^*\mathbf{W}_a\mathbf{Pr}_z$ can be interpreted as correlation on the original signal z with the filter kernels

$$\begin{aligned}&\mathbf{W}_a\mathbf{P}(\mathbf{P}^T\mathbf{W}_a\mathbf{P})^{-1}\mathbf{P}\mathbf{W}_a\mathbf{B}\\ &= \mathbf{W}_a[\mathbf{1} \quad \tfrac{1}{\sigma}\sqrt{\tfrac{\pi}{8}}(\mathbf{x}+i\mathbf{y}) \quad \tfrac{1}{4\sigma^2}(\mathbf{x}^2-\mathbf{y}^2+2i\mathbf{xy})]\end{aligned} \tag{5.19}$$

The filter kernels are is shown in figure 5.7.



$\mathbf{a}\cdot\mathbf{1}$ $\quad$ $\mathbf{a}\cdot\frac{1}{\sigma}\sqrt{\frac{\pi}{8}}(\mathbf{x}+i\mathbf{y})$ $\quad$ $\mathbf{a}\cdot\frac{1}{4\sigma^2}(\mathbf{x}^2-\mathbf{y}^2+2i\mathbf{xy})$

Figure 5.7: Correlation of the polynomial model $\mathbf{Pr}_z$ with the filter kernels $\mathbf{a}\cdot\mathbf{b}_n$ is equivalent to correlation of the original signal $\mathbf{z}$ with the filter kernels above.

We can use the normalized inhibition technique in section 5.3.3 to make the responses more selective. First we normalize with the amount of certainty:

$$s_n \sim \frac{\langle \mathbf{a}\cdot\mathbf{b}_n,\ \mathbf{Pr}_z\rangle}{\langle \mathbf{a},\ |\mathbf{z}|\rangle} \tag{5.20}$$

It would perhaps be more correct to normalize with $\langle \mathbf{a},\ |\mathbf{Pr}_z|\rangle$ instead of $\langle \mathbf{a},\ |\mathbf{z}|\rangle$, but this variant is more computationally complex. The normalized inhibition was based on the property $0 \le |s_n| \le 1$. This does only approximately hold in equation 5.20. But we can modify the inhibition rule and instead use

$$\begin{aligned}|s_0^p| &= h(|s_0|)\,(1-h(|s_1|))\,(1-h(|s_2|))\\ |s_1^p| &= h(|s_1|)\,(1-h(|s_0|))\,(1-h(|s_2|))\\ |s_2^p| &= h(|s_2|)\,(1-h(|s_0|))\,(1-h(|s_1|))\end{aligned} \tag{5.21}$$

where

$$h(t) = \begin{cases} 1 & \text{if } t > 1\\ t & \text{otherwise}\end{cases} \tag{5.22}$$

The phase of s_n^p still equals the phase of s_n.

The result of this method on the simple test image is shown in the third row in figure 5.10. $c_{out} = \langle \mathbf{a},\ |\mathbf{z}|\rangle$ is used as output certainty. The standard deviation σ of the Gaussian applicability $\mathbf{a}$ was chosen a bit smaller than in the previous experiments because the filter kernels in figure 5.7 are slightly larger than the previous kernels in figure 5.6 for the same choice of σ. The result is approximately equal to the non-approximative normalized inhibition result in the second row.

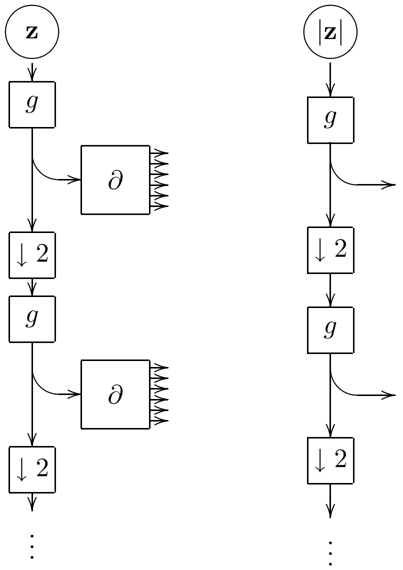


Figure 5.8: Multiscale correlator structures needed to compute rotational symmetries in several scales. The ∂-box are equal to the ∂-box in figure 4.7.

Multiscale detection

The method above is easily generalized to multiscale detection. We simply use the multiscale polynomial expansion algorithm in section 4.5. The multiscale algorithm is as follows:

1. Compute a local polynomial expansion model of $\mathbf{z}$ in several scales using the algorithm in section 4.5. In addition, compute a lowpass hierarchy of the certainty $|\mathbf{z}|$. The total correlator structure is shown in figure 5.8. We now have model parameters $\mathbf{r}_z$ and output certainties $c_{out} = \langle \mathbf{a},\ |\mathbf{z}|\rangle$ in several scales.
2. In each scale, compute the scalar product $\langle \mathbf{a} \cdot \mathbf{b}_n,\ \mathbf{P}\mathbf{r}_z\rangle$ using equation 5.17.
3. In each scale, normalize with the output certainty according to equation 5.20.
4. In each scale, compute the normalized inhibition in equation 5.21.
5. The final response is computed as $c_{out} \cdot s_n^p$.

5.4 Previous work

This section summarize most of the earlier work on rotational symmetries.

The rotational symmetry theory was developed around 1981 by Granlund and Knutsson, however first mentioned in a patent from 1986, see [Knutsson et al., 1988a], [Knutsson and Granlund, 1988], [Knutsson et al., 1988b]. An early related work is also [Hoffman, 1966].

A previous solution to achieve better selectivity is called the *consistency operation*, see [Knutsson and Granlund, 1988], [Granlund and Knutsson, 1995]. This technique is also used in a commercial image processing program by ContextVision[1]. The name refers to the idea that the operator should only respond to signals which are consistent with the signal model. The method is based on a combination of four scalar products:

$$\begin{aligned} h_{n1} = \langle \mathbf{a} \cdot \mathbf{b}_n,\ \mathbf{z} \rangle \quad &, \qquad h_{n2} = \langle \mathbf{a} \cdot \mathbf{b}_n,\ \mathbf{c} \rangle \\ h_{n3} = \langle \mathbf{a},\ \mathbf{z} \rangle \quad &, \qquad h_{n4} = \langle \mathbf{a},\ \mathbf{c} \rangle \end{aligned} \tag{5.23}$$

The filter results are combined into

$$h_n = \frac{h_{n4} h_{n1} - h_{n2} h_{n3}}{h_{n4}^{\gamma}} \tag{5.24}$$

where the denominator is an energy normalization controlling the model versus energy dependence of the algorithm. With $\gamma = 1$ the orientation magnitude varies linearly with the magnitude of the input signal. Decreasing the value increases the selectivity. The result from this operation is called *divcons* when $n = 1$ and *rotcons* when $n = 2$. The result when applying this method (with $\gamma = 1$) on the simple test image is shown in the last row in figure 5.10. The result is less selective than the normalized convolution and inhibition results, especially the rotcons result. The explanation for this, as pointed out in [Westin, 1994], is that divcons and rotcons can be fairly comparable to normalized convolution with $\{\mathbf{b}_0, \mathbf{b}_1\}$ resp. $\{\mathbf{b}_0, \mathbf{b}_2\}$ as basis functions, i.e.

$$\begin{aligned} \text{divcons:} \quad & \hat{\mathbf{z}} \sim s_0 \mathbf{b}_0 + s_1 \mathbf{b}_1 \\ \text{rotcons:} \quad & \hat{\mathbf{z}} \sim s_0 \mathbf{b}_0 + s_2 \mathbf{b}_2 \end{aligned} \tag{5.25}$$

This means for instance that the corners are not 'inhibited' in the rotcons response but are instead detected as 'half circles'. In general, using normalized convolution or normalized inhibition instead of the consistency operation produces more selective and sparse responses.

Bigün, [Bigün, 1988], [Bigün, 1997], [Bigün, 1990], [Bigün, 1987], describes detection of rotational symmetries as a special case of 'finding invariants of Lie groups of transformations'. The idea is to describe the symmetries as classes of signals that is invariant to translations in some coordinate system. A suitable coordinate system is chosen and the best translation invariant orientation is found. The solution can be described as a linear filter on the local gradient in double angle representation, $\langle \mathbf{z},\ \mathbf{w} \rangle$ where $\mathbf{w}$ can be a rotational symmetry $\mathbf{a} \cdot \mathbf{b}_n$ or some other

[1] `http://www.contextvision.se/`

prototype pattern. The difference between $|\langle \mathbf{z},\ \mathbf{w}\rangle|$ and $\langle |\mathbf{z}|,\ |\mathbf{w}|\rangle$ is used as a measure of model certainty. This is closely related to the normalized inhibition idea in section 5.3.3.

The algorithm was illustrated on sea bottom photographs and electronic circuit schemes. The object was to find sea anemones (spiral patterns) and transistors respectively.

Bigün also points out the relation to Generalized Hough Transform, GHT, which uses edges and their orientation to detect curvature, see [Duda and Hart, 1972]. The difference is that in GHT only positive votes give a contribution, while in complex valued correlation we will also get negative votes.

Bårman, [Bårman and Granlund, 1988], use the −1:st and 1:st order symmetries to detect corners. These two symmetries are well suited for this task, as can be seen in figure 5.5.

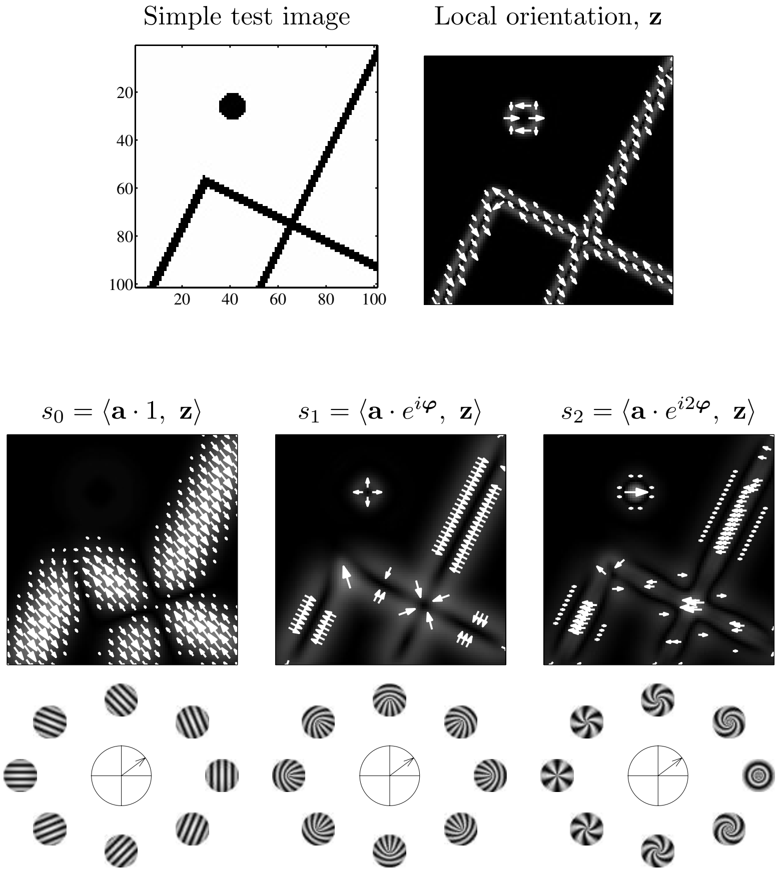
Bårman has also developed an algorithm to detect curvature using local orientation in double angle representation, see [Bårman, 1991]. In this algorithm, a set of quadrature filters is applied on the local orientation image, and the filter responses are used to detect curvature. An inhibition strategy between a straight line measure and the curvature measure is applied to improve the curvature direction estimate as well as the curvature certainty. This curvature is related to the curvature described by the first order symmetry. Bårman also points out the close relation between the strategies of detecting edges on the gray-scale image and detecting curvature on the local orientation image. This is also confirmed by equations 5.15 and 5.17 where it is shown that computing the first order response is equivalent to computing the gradient of the local orientation z.

Westelius, [Westelius, 1995], uses the rotcons operator (and a $\cos^2$-applicability) to create potential fields which are used in object recognition problems. The potential fields indicate possible locations for objects, which then can be further examined.

5.5 Conclusions

The rotational symmetry theory is a tool to describe curvature using local orientation in double angle representation, z. By representing the information as the phase $\angle z$ and using the magnitude $|z|$ as a certainty measure we can more easily handle the many variations of curvature patterns that can occur in an image.

This thesis introduces a new algorithm to detect these symmetries in one or several scales. The new algorithm uses the parameters from a polynomial expansion model, which can be very efficiently computed. The result on a simple test image shows that the new method gives subjectively the same result as the old ones but are much faster to compute. Further evaluation can be found in the experiment chapter. The first experiment in section 6.2 evaluates the performance on test images of 0:th, 1:st, and 2:nd order symmetries. The images were created



Simple test image

Local orientation, $\mathbf{z}$

$s_0 = \langle \mathbf{a} \cdot 1,\ \mathbf{z} \rangle$

$s_1 = \langle \mathbf{a} \cdot e^{i\boldsymbol{\varphi}},\ \mathbf{z} \rangle$

$s_2 = \langle \mathbf{a} \cdot e^{i2\boldsymbol{\varphi}},\ \mathbf{z} \rangle$

Figure 5.9: Top: Simple test image and its corresponding local orientation in double angle representation. Middle: 0:th, 1:st, and 2:nd order rotational symmetry responses. Bottom: Interpretation of the result in the middle row. The intensities represent the response magnitude. The vectors represent the complex values in a regular grid for the local orientation and for the 0:th order and at the maxima points for the 1:st and 2:nd order (because last two are point features).

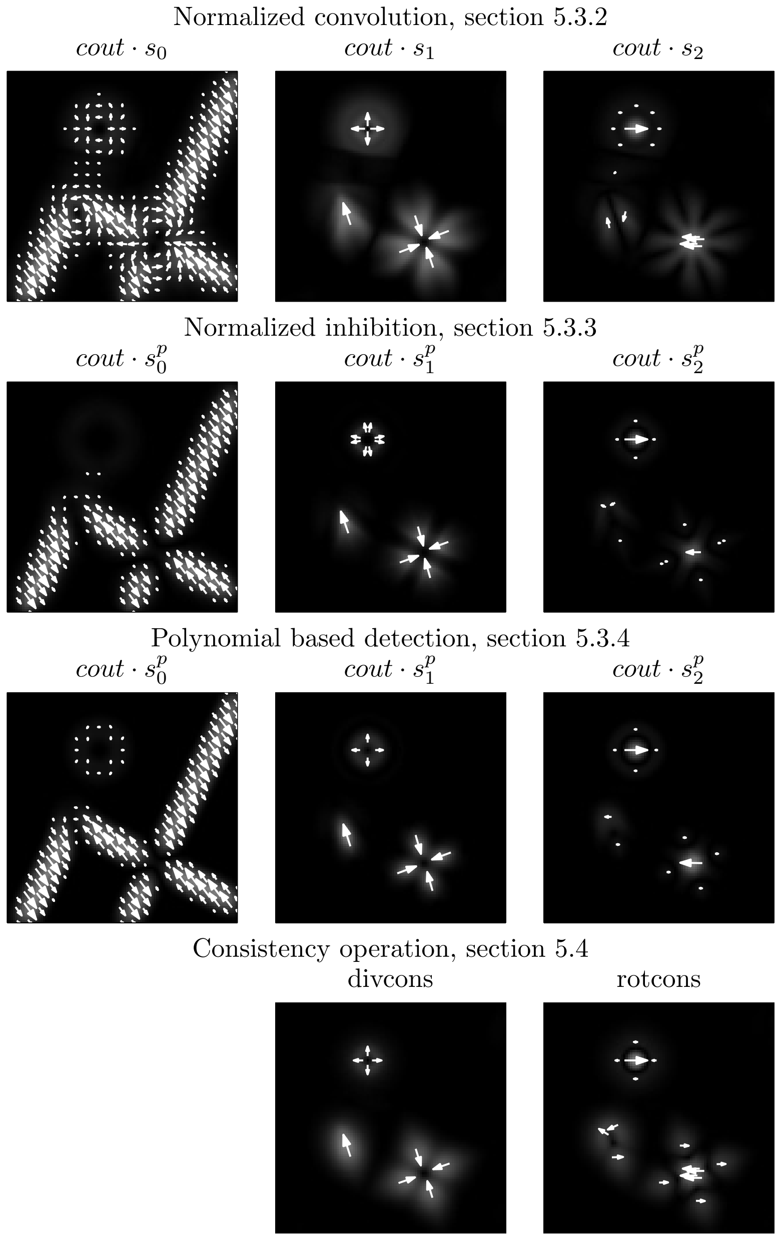


Figure 5.10: Different techniques to make the responses in figure 5.9 more sparse and selective.

using the back-projection method in appendix A. The algorithm seems to work as expected on those images. The symmetry detection algorithm is also evaluated in an application experiment in section 6.4, where the goal is to recognize objects from its curvature.

This chapter has also indicated that with suitable choices of applicabilities there is strong relationship between polar Fourier transform, polynomial expansion, and partial derivatives, i.e.

$$\left\{\begin{array}{lcl}\mathbf{b}_0 & = & 1\\ \mathbf{b}_1 & = & e^{i\varphi}\\ \mathbf{b}_2 & = & e^{i2\varphi}\end{array}\right. \leftrightsquigarrow \left\{\begin{array}{lcl}\mathbf{b}_0 & = & 1\\ \mathbf{b}_1 & = & x+iy\\ \mathbf{b}_2 & = & (x+iy)^2\end{array}\right. \leftrightsquigarrow \left\{\begin{array}{lcl}\mathbf{b}_0 & = & 1\\ \mathbf{b}_1 & = & \partial x+i\partial y\\ \mathbf{b}_2 & = & (\partial x+i\partial y)^2\end{array}\right.$$

Chapter 6

Experiments

This chapter contains experiments based on the theory presented in the previous chapters.

6.1 Evaluation of the approximative polynomial expansion algorithm

This section evaluates the approximative polynomial expansion algorithm described in chapter 4. The first experiment, section 6.1.1, evaluates the parameter approximation on natural images. It is difficult to perform the evaluation disregarding the application - sometimes an approximation of a part of the system can be quite poor without deteriorating the overall system performance. The algorithm is therefore also evaluated in an orientation estimation experiment, section 6.1.2.

6.1.1 Evaluation on natural images

The first evaluation is made by comparing the parameters from the 'ideal' algorithm in section 4.2.1 (full certainty case) with the approximative algorithm described in section 4.4.1 using the estimated transformation matrix $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ in equation 4.42 and the filter optimization method based on least squares described in section 4.6.1. How well the polynomial model is suited for image applications is not an issue here.

Since this algorithm is intended to be used on image data, it is evaluated on the same sort of data. Figure 6.1 shows the test images used in this experiment.

The polynomial model parameters were estimated in each image using both the approximative algorithm and the ideal algorithm with varying standard deviation of the Gaussian applicability. To avoid border effects, the results within reach of the image borders, corresponding to half the applicability size, were cut off before evaluation. This left about 200000–250000 neighborhoods (depending on the applicability size) for evaluation. The parameters from the approximative and

Figure 6.1: Test images for the evaluation experiment.

ideal algorithms were then compared using the following error measure:

$$e = \frac{\|\mathbf{Pr} - \mathbf{P}\tilde{\mathbf{r}}\|_W}{\|\mathbf{Pr}\|_W} = \sqrt{\frac{(\mathbf{r} - \tilde{\mathbf{r}})^T \mathbf{P}^T \mathbf{W} \mathbf{P} (\mathbf{r} - \tilde{\mathbf{r}})}{\mathbf{r}^T \mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{r}}} \tag{6.1}$$

This is a relative error between the local image neighborhoods described by the approximative parameters $\tilde{\mathbf{r}}$ and the true parameters $\mathbf{r}$ respectively. The norm is weighted with the same Gaussian function that was used as applicability to estimate the parameters. The result is shown in figure 6.2.

The parameters are also separately compared using the SNR measure in appendix C:

$$e_k = 10 \log \frac{\sum (r_k - \tilde{r}_k)^2}{\sum r_k^2} \; [dB] \tag{6.2}$$

The result is shown in figure 6.3.

Some observations can be made from the experiment:

- The approximation performs well above $\sigma = 2$. $\sigma < 2$ corresponds to image regions smaller than 5×5 and the approximation becomes worse due to discretization effects.

- The largest relative errors can be found some distance away from sharp

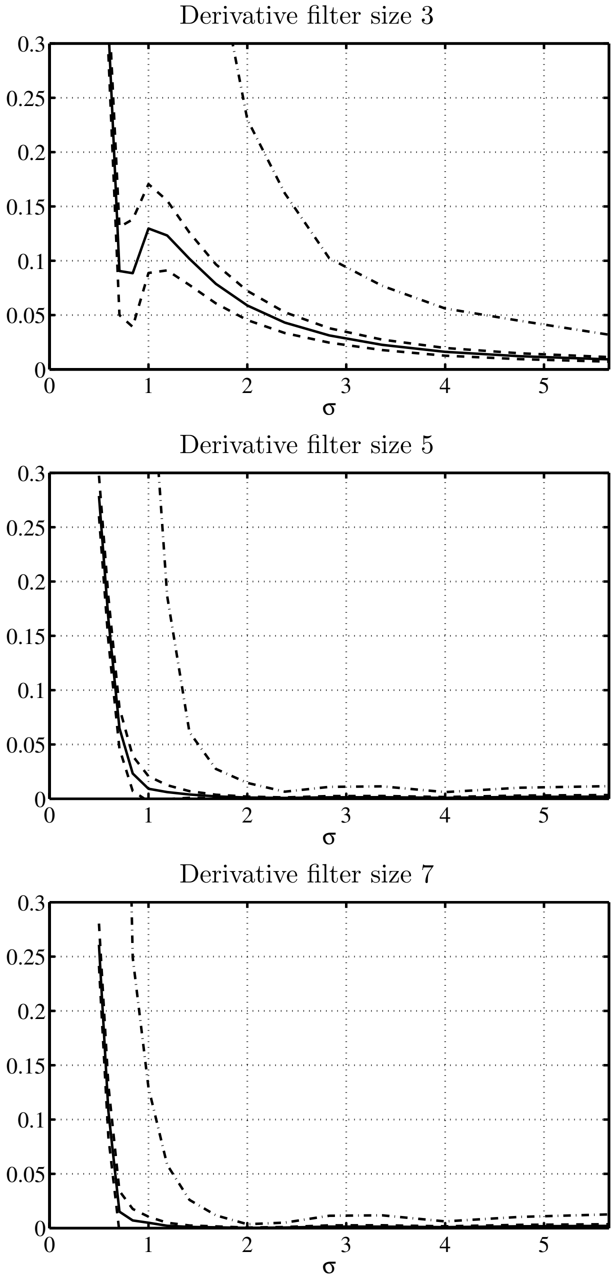


Figure 6.2: Relative error, $e = \frac{\|\mathbf{Pr}-\mathbf{P}\tilde{\mathbf{r}}\|_W}{\|\mathbf{Pr}\|_W}$, for Gaussian applicability with varying standard deviation, σ and varying size of the derivative filters. Bottom figure: Using derivative filters of size 5. Solid lines: mean(e), Dashed lines: mean(e) ± std(e), Dash-dotted lines: max(e)

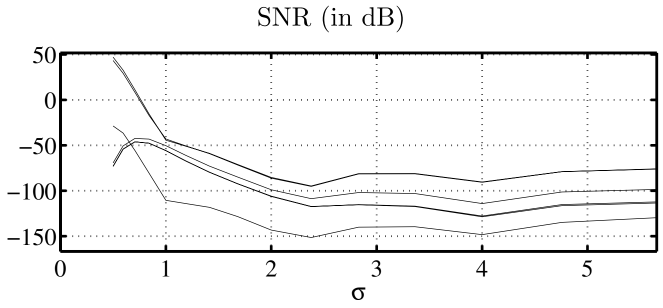


Figure 6.3: Evaluation of the model parameters separately using equation 6.2. From top to bottom: e_4, e_5, e_6, e_2, e_3, e_1 (e_4, e_5 and e_2, e_3 respectively are almost identical). Derivative filter size 5 was used.

edges. This is due to the poor performance of the derivative approximation for such patterns.

- The performance improved with increasing size of the derivative filters, but larger filter size than 7 did not significantly improve the result.

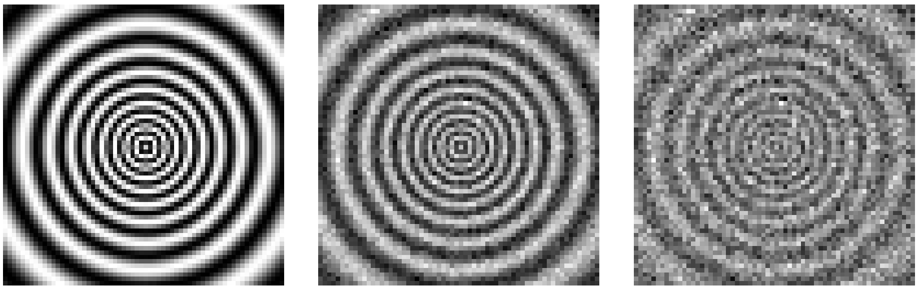
- The coefficients corresponding to x^2 and y^2 has the worst average performance. This is because the second partial derivatives are estimated by correlating the derivative filter on approximations of the first partial derivatives.

- Section 4.6.2 also mentioned another estimation, $\widetilde{\mathbf{T}}_{\mathrm{PD}}$, in equation 4.40. Experiments (not included here) indicate that using this estimation gives a slightly better average performance, while the maximum error becomes slightly higher.

- In this experiment the same filter was used to approximate both the first and the second derivative. The result when the second filter was optimized separately did not differ significantly. If anything it performed slightly worse, suggesting that the optimization method used in section 4.6.1 may not be the best one to use.

6.1.2 Local orientation estimation on 3D data

In this section the approximative polynomial expansion algorithm is evaluated in an application. The polynomial model parameters are used to estimate local orientation in a three-dimensional signal. The $64 \times 64 \times 64$ test volume, called the *onion volume*, consists of concentric spherical shells, see figure 6.4 for a cut through the center. Different amounts of noise is added and the local orientation is then estimated. Estimates near the center and the border of the volume are removed before the evaluation to avoid center irregularities and border effects.



(a) SNR = ∞ (b) SNR = 10 dB (c) SNR = 0 dB

Figure 6.4: Slice 32 of the $64 \times 64 \times 64$ test volume with different amount of noise added. The test volume can be imagined as slice 32 rotated around an axis going through the center of the volume.

The onion volume has been used before by Andersson, Wiklund and Knutsson (see [Andersson et al., 1998, Knutsson and Andersson, 1995]) and by Farnebäck (see [Farnebäck, 1999a, Farnebäck, 2000b]) to evaluate local orientation estimation algorithms. The orientation estimation was in both cases based on *orientation tensors*. In the second case the tensors were computed using parameters from a local polynomial model, which will be used as reference when evaluating the approximative polynomial expansion algorithm presented in this thesis.

Before making the experiments we summarize the theory of orientation tensors:

Orientation tensors

A more complete description of this theory can be found in [Farnebäck, 1999a], [Granlund and Knutsson, 1995].

Assume we have a *simple signal*, i.e. a signal that varies only in one direction $\mathbf{n}$ and is constant in all directions orthogonal to $\mathbf{n}$. This can be written as

$$f(\mathbf{x}) = h(\mathbf{x}^T\mathbf{n}) \tag{6.3}$$

for some function h. Figure 6.5 shows some examples of simple and non-simple two-dimensional signals.

The simple signal above is represented by the orientation tensor

$$\mathbf{T} = A\hat{\mathbf{n}}\hat{\mathbf{n}}^T \tag{6.4}$$

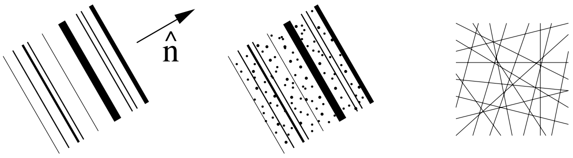


Figure 6.5: Examples of a simple signal (left) and non-simple signals (middle and right).

where A is a scalar that may depend on signal energy, certainty, or other useful information. It is easy to show that the signal orientation $\hat{\mathbf{n}}$ can be computed from the tensor as the eigenvector corresponding to the largest eigenvalue. For a non-simple signal there does not exist a clear definition of the tensor, but the dominant orientation is still estimated as the eigenvector corresponding to the largest eigenvalue. Besides dominant orientation we can also extract information about signal variation in other directions by studying the eigensystem of $\mathbf{T}$. However, in this experiment only the dominant orientation is used.

There are different ways to compute the tensor in practice. One way, used by Andersson et al., is to look at the local signal energy in different directions using a set of *quadrature filters* and weigh them together in a proper way, see [Granlund and Knutsson, 1995]. A quadrature filter is basically a complex valued filter with an even real part and an odd imaginary part and with a Fourier transform that is localized around a certain direction. The filtering can be efficiently computed by use of sequential filter trees [Andersson et al., 1998].

Farnebäck on the other hand uses the parameters from a local polynomial model to compute the tensor. In the three-dimensional case we have the model

$$f(\mathbf{x}) \sim c + \mathbf{b}^T\mathbf{x} + \mathbf{x}^T\mathbf{A}\mathbf{x} \quad , \quad \mathbf{x} \in \mathbb{R}^3 \tag{6.5}$$

where c is a scalar, $\mathbf{b}$ is a 3×1 vector and $\mathbf{A}$ is a 3×3 symmetric matrix. The orientation tensor is computed as

$$\mathbf{T} = \mathbf{A}\mathbf{A}^T + \gamma\mathbf{b}\mathbf{b}^T \tag{6.6}$$

where γ is a non-negative weight factor between the linear (odd) and quadratic (even) part of the signal model. The choice of γ depends on σ.

Experiments

In previous experiments an orientation tensor was computed in each local neighborhood and the dominant orientation was estimated as the eigenvector $\hat{\mathbf{e}}_1$, corresponding to the largest eigenvalue. The performance of the algorithms were measured by an angular RMS error

$$\Delta\phi = \arcsin\left(\sqrt{\frac{1}{2L}\sum_{l=1}^{L}\|\hat{\mathbf{x}}\hat{\mathbf{x}}^T - \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T\|^2}\right) = \arccos\left(\sqrt{\frac{1}{L}\sum_{l=1}^{L}(\hat{\mathbf{x}}^T\hat{\mathbf{e}}_1)^2}\right) \qquad (6.7)$$

where $\hat{\mathbf{x}}$ is the true orientation and L is the number of points. To avoid border effects and irregularities at the center of the volume, the sum was only computed for points at radius between 0.16 and 0.84. The results from these experiments is recalled in table 6.1. In the second case the RMS errors refer to the optimal choices of σ (≈ 1) and γ (≈ 0.1). In the table is also included the number of filter coefficients needed to compute the tensors.

In this experiment we use Farnebäcks theory to compute the orientation tensors, equation 6.6, but the polynomial expansion is computed using the approximative algorithm described in chapter 4. Figure 6.6 contains the correlator structure needed to compute the approximative polynomial expansion in 3D.

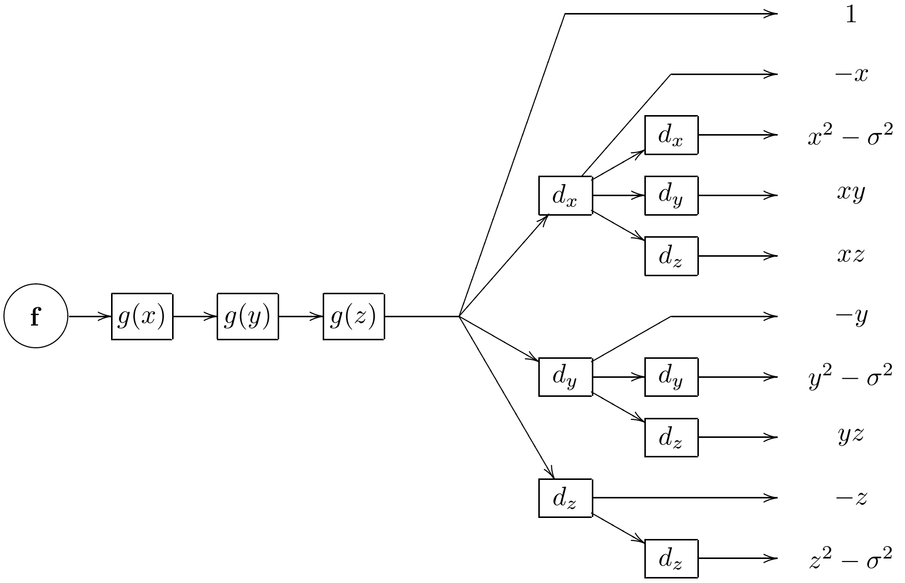


Figure 6.6: Correlator structure for approximate polynomial expansion in 3D with full certainty.

The output from the correlator structure in each local neighborhood becomes $\widetilde{\mathbf{D}}_3^T\mathbf{W}\mathbf{f}$, where $\widetilde{\mathbf{D}}_3$ is an approximation of

$$\mathbf{D}_3 = \begin{pmatrix} | & | & | & | & | & | & | & | & | & | \\ \mathbf{1} & -\mathbf{x} & -\mathbf{y} & -\mathbf{z} & \mathbf{x}^2-\sigma^2 & \mathbf{y}^2-\sigma^2 & \mathbf{z}^2-\sigma^2 & \mathbf{xy} & \mathbf{xz} & \mathbf{yz} \\ | & | & | & | & | & | & | & | & | & | \end{pmatrix} \qquad (6.8)$$

This result is then transformed to the polynomial model parameters by

$$\tilde{\mathbf{r}} = \widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}_3^T\mathbf{W}\widetilde{\mathbf{D}}_3)^{-1}\widetilde{\mathbf{D}}_3^T\mathbf{W}\mathbf{f} \tag{6.9}$$

Two estimates of $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ were mentioned in section 4.6.2. The first one, equation 4.40, minimizes the error in the polynomial subspace:

$$\widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}_3^T\mathbf{W}\widetilde{\mathbf{D}}_3)^{-1} = (\widetilde{\mathbf{D}}_3^T\mathbf{W}\mathbf{P})^{-1} \tag{6.10}$$

The second one, equation 4.43, minimizes the expectation error using the signal model $\mathbf{C}_{\mathrm{ff}} = \mathrm{E}[\mathbf{f}\mathbf{f}^T] = \mathbf{I}$:

$$\widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}_3^T\mathbf{W}\widetilde{\mathbf{D}}_3)^{-1} = (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2\widetilde{\mathbf{D}}_3(\widetilde{\mathbf{D}}_3^T\mathbf{W}^2\widetilde{\mathbf{D}}_3)^{-1} \tag{6.11}$$

Both versions are evaluated in this experiment.

Before the results are presented, it should be mentioned that the angular RMS error was computed using points at a radius between 0.22 and 0.78 instead of 0.16 – 0.84 used in previous experiments. The reason for this is that the local neighborhood used to compute the approximative polynomial expansion is slightly larger than in the non-approximative algorithm for the same size of the Gaussian filters because of the subsequent derivative filters. Therefore, the test volume had to be pruned somewhat more in order to avoid border effects. For the sake of comparison, the Farnebäck experiment (non-approximative polynomial expansion) was reproduced using the new radius interval. The new result did not significantly differ from the previous one using the old radius interval, which suggests that the result of the new algorithm in this thesis can be compared with both previous results in table 6.1.

All combinations of the following parameters were tested:

- A Gaussian filter of size 9 was used in all experiments.
- $\sigma = 0.5, 0.6, 0.7, ..., 2.0$
- $\gamma = 10^{-2}, 10^{-1.5}, 10^{-1}, ..., 10^{2}$
- Derivative filters of size $m = 3, 5, 7$
- Both estimates $\widetilde{\mathbf{T}}_{\mathrm{PD}}$, equations 6.10 and 6.11.

The result for optimal σ and γ is shown in table 6.2. Figure 6.7 shows the RMS error as a function of σ and γ for the non-approximative and the approximative algorithms.

Some observations can be made from the experiment:

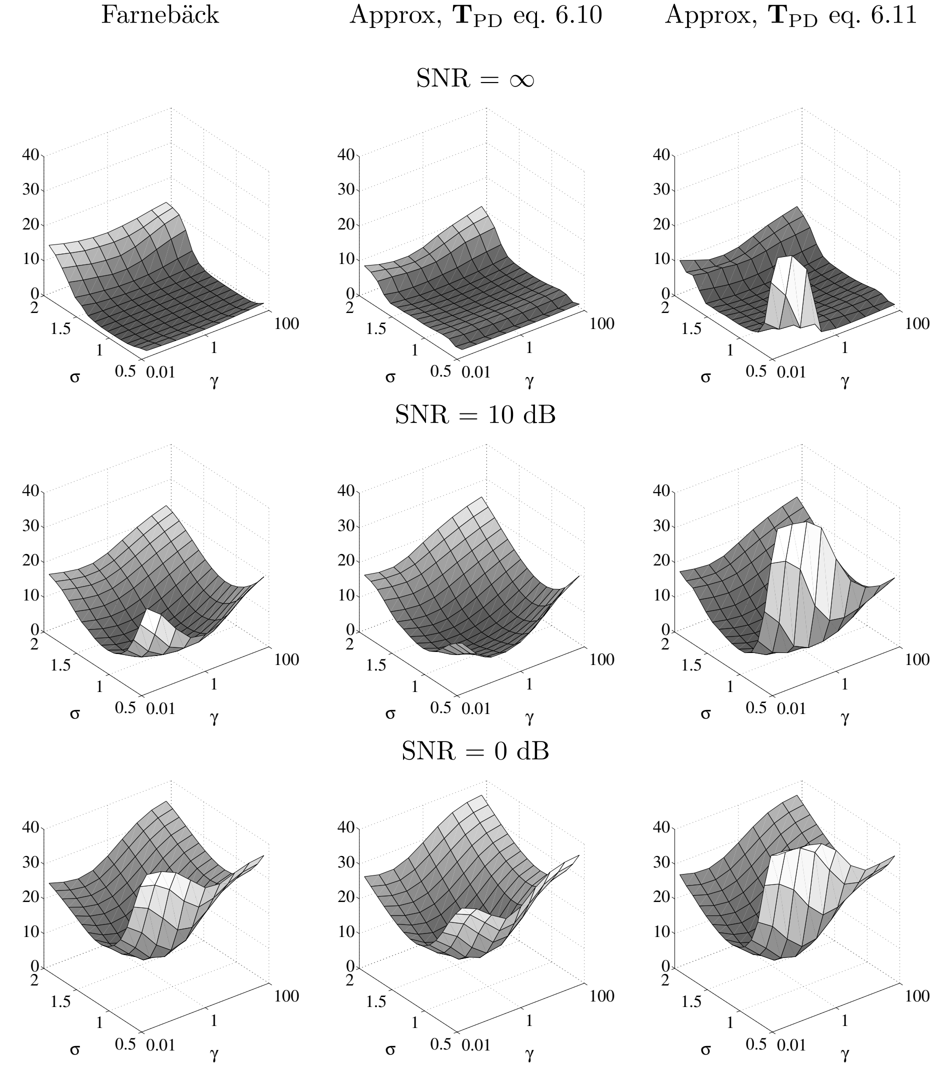
- Optimal σ (≈ 1) and γ (≈ 0.1) is similar in the non-approximative and approximative cases, except for SNR $= \infty$ where $\sigma_{opt} \approx 1.4$. The reason could be that the filter approximation is best for $\sigma > 1$, as we could see in the experiment in section 6.1.1, and the best σ in this experiment is therefore a trade-off between optimal σ and good filter approximations.

SNR	Andersson, Wiklund, Knutsson 345 coeff	Farnebäck 171 coeff
∞	0.76°	0.11°
10 dB	3.02°	3.03°
0 dB	9.35°	10.24°

Table 6.1: Performance of previous algorithms, [Andersson et al., 1998, Knutsson and Andersson, 1995] and [Farnebäck, 1999a, Farnebäck, 2000b]. The evaluation included points at a radius between 0.16 and 0.84.

SNR	Farnebäck 171 coeff	Johansson					
		pol. subspace $\widetilde{\mathbf{T}}_{\mathrm{PD}}$, eq. 6.10			statistical $\widetilde{\mathbf{T}}_{\mathrm{PD}}$, eq. 6.11		
		$m = 3$	$m = 5$	$m = 7$	$m = 3$	$m = 5$	$m = 7$
		54 coeff	72 coeff	90 coeff	54 coeff	72 coeff	90 coeff
∞	0.11°	4.39°	0.78°	0.28°	3.89°	0.78°	0.28°
10 dB	3.01°	5.79°	3.39°	3.05°	5.26°	3.41°	3.06°
0 dB	10.38°	11.42°	10.25°	10.37°	13.72°	10.40°	10.38°

Table 6.2: Comparison between non-approximative algorithm (Farnebäck) and new approximative algorithm for different parameter values. The evaluation included points at a radius between 0.22 and 0.78.



Farnebäck

Approx, $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ eq. 6.10

Approx, $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ eq. 6.11

SNR $= \infty$

SNR $=$ 10 dB

SNR $=$ 0 dB

Figure 6.7: RMS error as a function of σ and γ for the non-approximative (Farnebäck) and the approximative (new) algorithms. Derivative filter size $m = 5$ was used in all cases.

- The statistical estimate of $\widetilde{\mathbf{T}}_{\mathrm{PD}}$, equation 6.11, gives a higher error for $\sigma < 1$ and small γ than the estimate in equation 6.10, but otherwise they have about the same behavior regarding σ and γ as the non-approximative algorithm. The bad performance for the statistical estimate of $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ suggests that the signal model $\mathbf{C}_{\mathrm{ff}} = \mathbf{I}$ may be too simple.

6.1.3 Conclusions

The experiments in section 6.1.1 and 6.1.2 suggest that the approximative polynomial expansion algorithm works well for $\sigma > 1$, but more application-oriented experiments should be made before making any final conclusions. Optimal σ for the experiment in section 6.1.2 is around 1, which is a bad choice for the approximative algorithm. This should be taken into account when evaluating the results.

A derivative filter size of 5 seems to be a good trade-off between accuracy and computational complexity. The performance is comparable to the performance of the Andersson, Wiklund, and Knutsson algorithm but the new algorithm needs only about a fourth as many filter coefficients. There is no significant difference between the simpler choice of $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ in equation 4.40 and the statistical one in equation 4.43 if $\sigma > 1$, perhaps because of the signal model $\mathbf{C}_{\mathrm{ff}} = \mathbf{I}$ not being very accurate.

6.2 Evaluation of the rotational symmetry algorithm

This section contains an experiment to evaluate the new rotational symmetry detection method based on polynomial expansion, section 5.3.4. The experiment aims to evaluate how the algorithm works for the patterns it is designed for. It is difficult to define what responses the method should give for other features than the rotational symmetries. This experiment might still give an indication of the algorithm performance. Further evaluation can be found in section 6.4, where the rotational symmetries are used in an object recognition experiment.

The experiment is designed as follows:

Step 1: 15×15 test images based on the back-projection algorithm in appendix A were created. The test images contain 0:th, 1:st, and 2:nd order symmetries with random phase, see figure 6.8, 6.9, and 6.10 respectively. Noise with PSNR $= 20$ dB was added.

Step 2: The local image gradient in double angle representation, z, was computed for each test image. Image borders corresponding to half the filter size were removed in order to avoid border effects, leaving 9×9 images of z.

Step 3: A global polynomial expansion model with Gaussian applicability, $\sigma = 2$, was computed for each z-image, $\mathbf{z} \sim \mathbf{Pr}_z$.

Step 4: The normalized responses

$$s_n = \frac{\langle \mathbf{g} \cdot \mathbf{b}_n,\ \mathbf{Pr}_z \rangle}{\langle \mathbf{g},\ |\mathbf{z}| \rangle} \quad , \quad n = 0, 1, 2 \tag{6.12}$$

were computed, where $\mathbf{b}_n = e^{in\boldsymbol{\varphi}}$.

Step 5: Finally the response magnitudes were inhibited using the rule

$$\begin{array}{rcl} |s_0^p| & = & h(|s_0|)\,(1 - h(|s_1|))\,(1 - h(|s_2|)) \\ |s_1^p| & = & h(|s_1|)\,(1 - h(|s_0|))\,(1 - h(|s_2|)) \\ |s_2^p| & = & h(|s_2|)\,(1 - h(|s_0|))\,(1 - h(|s_1|)) \end{array} \tag{6.13}$$

where

$$h(t) = \begin{cases} 1 & \text{if } t > 1 \\ t & \text{otherwise} \end{cases} \tag{6.14}$$

Figures 6.8, 6.9, and 6.10 show both the non-inhibited and inhibited responses for the three sets of test images respectively. As can be seen, the non-inhibited n:th order response, s_n, gives a magnitude approximately equal to 1 for the n:th order test images. Some of the test images give a high magnitude for more than one response order, e.g. lines in the outer area of the image. These images correspond to ambiguous patterns that can be classified as more than one symmetry order.

If we use the inhibition rule, we will get a low response for these patterns, which may be desirable if we only want to detect non-ambiguous patterns. The last two rows in each figure show the patterns which after inhibition gave the lowest and highest responses respectively. Note that the responses for $|s_2^p|$ in the experiment in figure 6.10 are all fairly high, so some of the lowest response patterns should still be considered as fairly certain second order symmetries.

The phase estimate is evaluated using the following formula:

$$\mathrm{std}_{|s_n^p|}(\Delta\alpha) = \sqrt{\frac{1}{N-1}\frac{\sum |s_n^p|_k(\alpha - \overline{\alpha})^2}{\sum |s_n^p|_k}} \tag{6.15}$$

where α is the true phase and $\overline{\alpha} = \angle s_n^p$ the estimated phase. It is basically the standard deviation of the error, but weighted with the magnitude response $|s_n^p|$ which serves as a certainty measure.

To conclude, the algorithm seems to work well for the patterns it is designed to detect. Ambiguous patterns will after inhibition get a lower response, and more well-defined patterns such as clear lines, edges, and high curvature points will still have a high response for their corresponding order. The approximation with polynomials does not significantly affect the result. Performing the same experiments using $s_n = \frac{\langle \mathbf{g}\cdot\mathbf{b}_n,\ \mathbf{z}\rangle}{\langle \mathbf{g},\ |\mathbf{z}|\rangle}$ instead of the approximative normalized response in step 4 gives similar results (not included here). Also, the phase estimate seems to be fairly accurate for the test images with the highest responses.

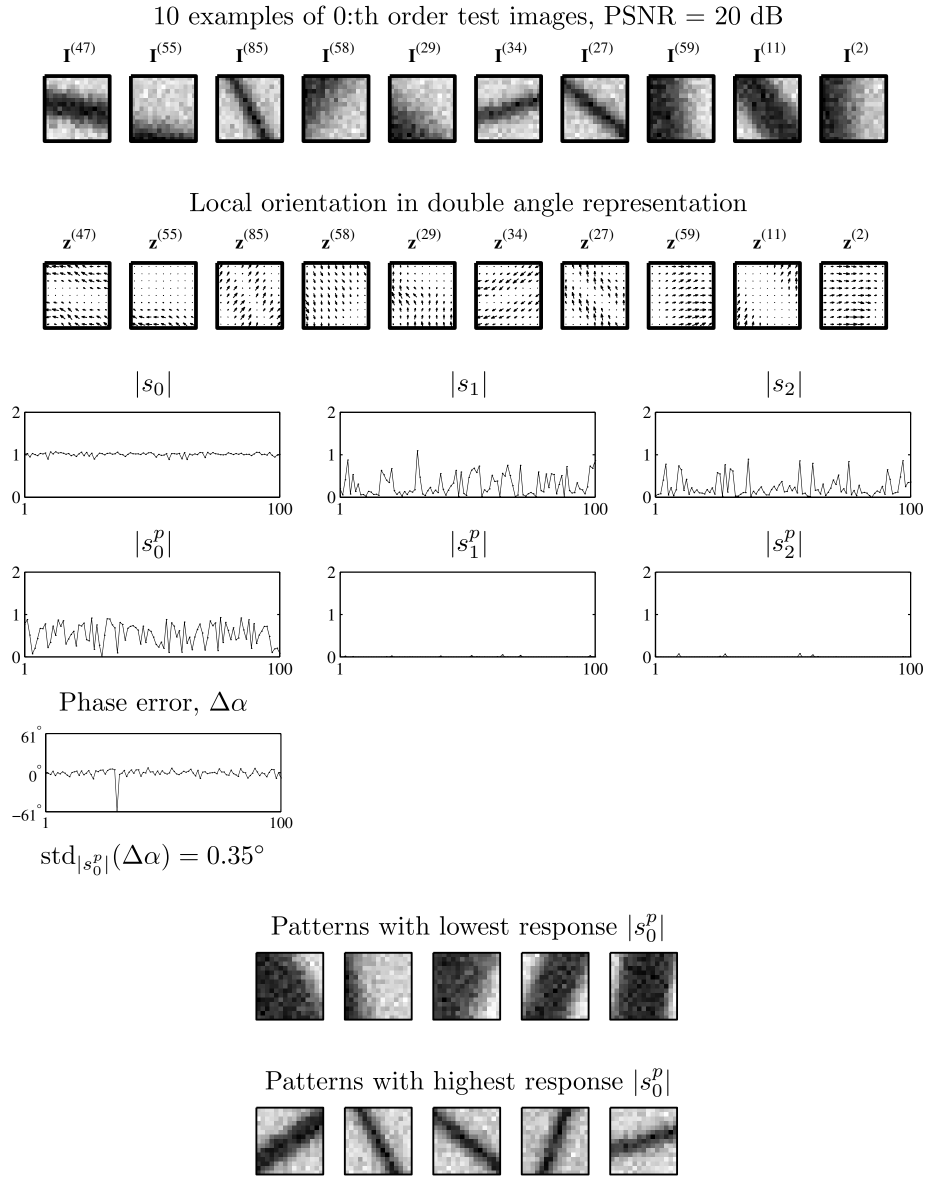


Figure 6.8: Zeroth, first, and second order responses on zeroth order symmetries.

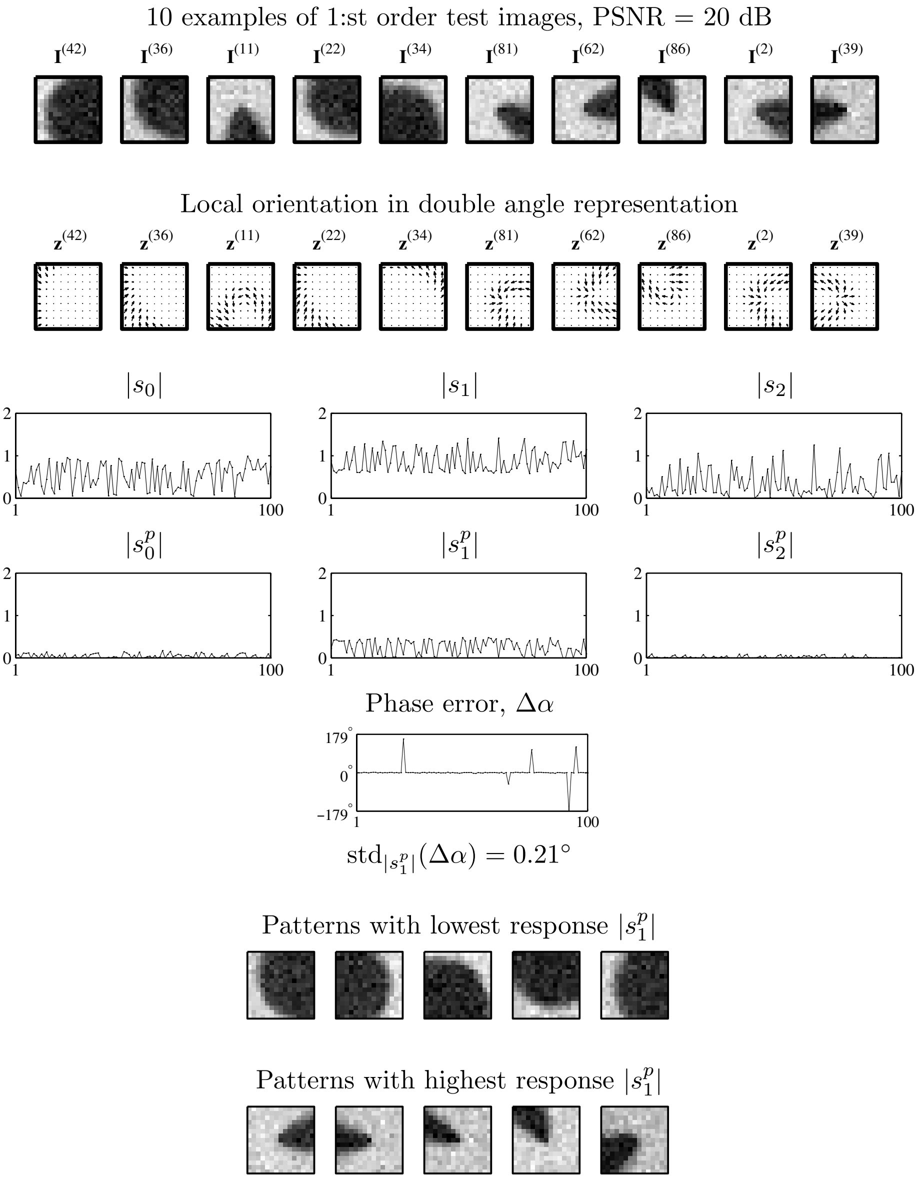


Figure 6.9: Zeroth, first, and second order responses on first order symmetries.

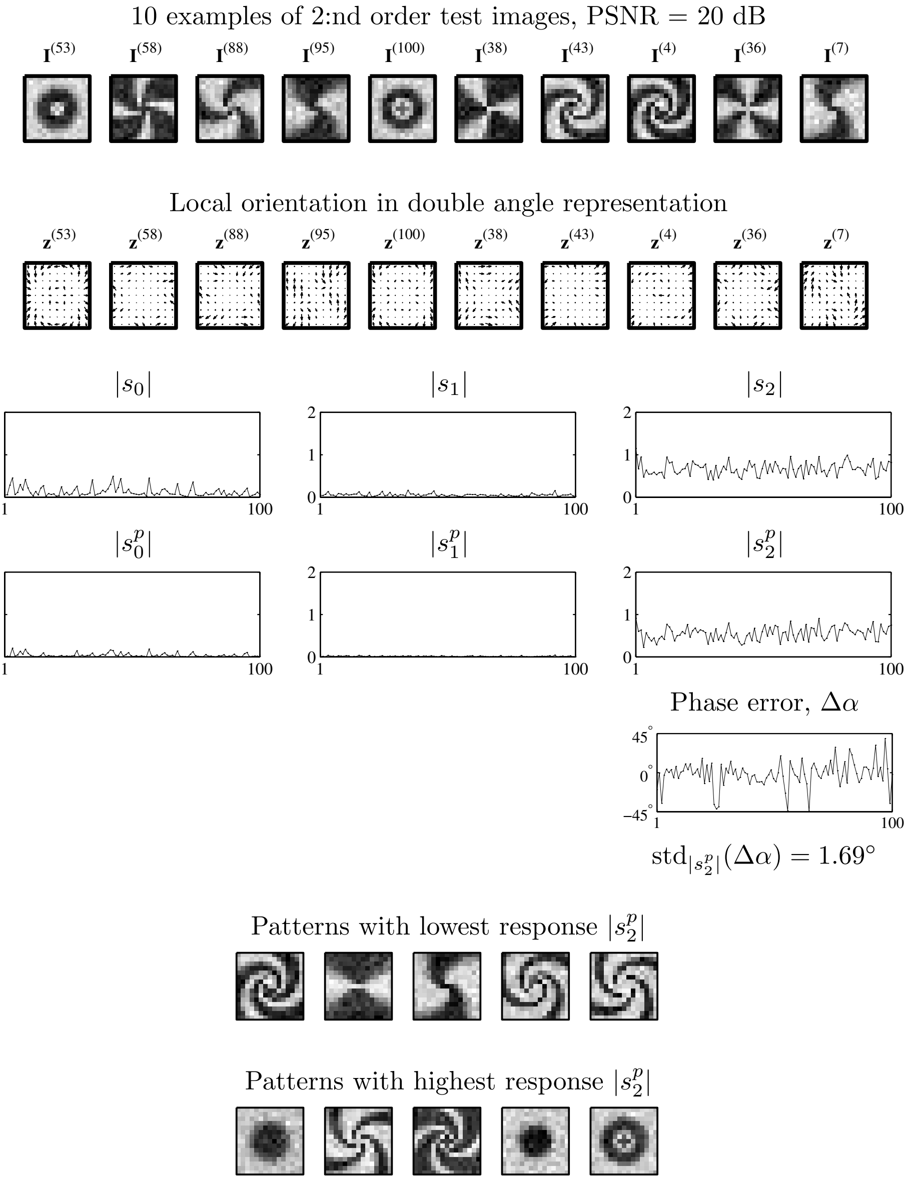


Figure 6.10: Zeroth, first, and second order responses on second order symmetries.

6.3 Learning a detector for corner orientation

6.3.1 Introduction

It is often difficult to design models and detectors for complex features analytically. The rotational symmetry detectors in chapter 5 are one exception. Another approach is to have a system learn a feature detector from a set of training data. If a suitable model is chosen, we can use the training data to find the model parameters. One feature learning approach based on canonical correlation is described in [Borga, 1998], [Knutsson and Borga, 1999], [Borga and Knutsson, 2000]. They show how a system can learn to detect local orientation invariant to signal phase by showing the system pairs of sinusoidal patterns that had the same local orientation but different phases. They used products between intensity values as input samples to the system, and the system then learned linear combinations of the input samples which in a simple manner could be decoded into the local orientation angle. It turned out that the linear combinations could be interpreted as quadrature filters.

This section (also presented in [Johansson et al., 2001]) illustrates how the same technique can be used to learn a descriptor of corner orientation which is invariant to corner angle and color (intensity). Several input data representations are examined: image intensity values, products between intensity values, and local orientation in double angle representation. The dimensionality of the input data can be quite large, especially if we use products between intensity values. Therefore, to reduce the dimensionality, parameters from a polynomial expansion model on the respective representations were also explored as input data. As shown below this reduction did not affect the performance of the system.

It may not be obvious that a quadratic model should work, but it is motivated by the following argument:

According to chapter 5 we can detect corners and other complex curvatures by applying a linear complex filter on a local orientation description in double angle representatoin. To find the local orientation we can for instance compute the image gradient, ∇I. This means that in each local neighborhood of the image we compute the scalar products

$$\nabla I = \begin{pmatrix} I_x \\ I_y \end{pmatrix} = \begin{pmatrix} \langle \mathbf{g}_x,\ \mathbf{I} \rangle \\ \langle \mathbf{g}_y,\ \mathbf{I} \rangle \end{pmatrix} \tag{6.16}$$

where $\mathbf{g}_x$ and $\mathbf{g}_y$ are derivating filters, for example derivatives of a Gaussian. From the gradient we can then for instance compute the double angle representation as

$$z = (I_x + iI_y)^2 = (\langle \mathbf{g}_x,\ \mathbf{I} \rangle + i\langle \mathbf{g}_y,\ \mathbf{I} \rangle)^2 = \langle \mathbf{g}_x + i\mathbf{g}_y,\ \mathbf{I} \rangle^2 \tag{6.17}$$

If we reshape $\mathbf{I}$, $\mathbf{g}_x$ and $\mathbf{g}_y$ into vectors we can write z as

$$z = \left((\mathbf{g}_x + i\mathbf{g}_y)^T \mathbf{I}\right)^2 = (\mathbf{g}_x + i\mathbf{g}_y)^T \mathbf{I}\mathbf{I}^T (\mathbf{g}_x + i\mathbf{g}_y) = \langle (\mathbf{g}_x + i\mathbf{g}_y)(\mathbf{g}_x + i\mathbf{g}_y)^T,\ \mathbf{I}\mathbf{I}^T \rangle \tag{6.18}$$

Hence, the double angle representation can be computed using a (complex) linear combination of products between the intensity values. Corners and other curvatures can be computed using a linear filter on the double angle representation, which means that these features too can be detected using linear combinations of products between the intensity values.

This discussion suggests that if we want to use learning to create a detector for corner orientation, it could be a good idea to use a quadratic model.

The first subsection contains a short description of the feature learning algorithm, and in the following subsections we apply this algorithm to the different image representations starting with the simplest ones, ending with the local orientation representation.

6.3.2 A canonical correlation approach

This subsection describes the basic ideas of the Borga and Knutsson feature learning algorithm based on canonical correlation analysis (CCA). A more thorough description can be found in [Borga, 1998].

Assume we have N pairs of training samples:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}^{(1)}\ \mathbf{x}^{(2)}\ ...\ \mathbf{x}^{(N)}] \\ \mathbf{Y} &= [\mathbf{y}^{(1)}\ \mathbf{y}^{(2)}\ ...\ \mathbf{y}^{(N)}] \end{aligned} \tag{6.19}$$

Each pair $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ has a number of common properties that we want the system to learn and other uncommon properties that we want the system to be invariant to. In this section each sample pair is computed from images of corners that has the same corner orientation and differ in corner angle and color, see figure 6.12 for an illustration.

If we compute the canonical correlation, described in section 3.2, between the signals $\mathbf{x}^{(n)}$ and $\mathbf{y}^{(n)}$ we will hopefully find a number of projections $\mathbf{w}_{xk}^T\mathbf{x}^{(n)}$ and $\mathbf{w}_{yk}^T\mathbf{y}^{(n)}$ which has a high correlation. The canonical correlation vectors $\mathbf{w}_{xk}$ and $\mathbf{w}_{yk}$ will then be fairly equal and correspond to a one-dimensional subspaces, that is invariant to the uncommon properties but varies with the common properties. It should therefore be possible to use $\mathbf{w}_{xk}$ to discriminate between the different values of the common properties.

If we use a quadratic model, i.e. products between intensity values or between other values, as input data the dimensionality will be fairly high. As a consequence the vectors $\mathbf{w}_{xk}$ will also be large. This is not suitable for practical applications. We can use eigenvalue decomposition to reduce the computational complexity:

Let $\mathbf{I}_x$ be $M \times M$-images corresponding to one of the training samples. If we reshape $\mathbf{I}_x$ into an $M^2 \times 1$-vector $\mathbf{i}_x$ we can write products between intensity values as an outer product, $\mathbf{i}_x\mathbf{i}_x^T$. The input to the learning algorithm is then this outer product reshape into an $M^4 \times 1$-vector, $\mathbf{x} = \text{vec}(\mathbf{i}_x\mathbf{i}_x^T)$. The projection of an image onto a canonical correlation vector $\mathbf{w}_x$ can then be written as

$$\mathbf{x}^T\mathbf{w}_x = \mathbf{i}_x^T\mathbf{W}_x\mathbf{i}_x \tag{6.20}$$

where $\mathbf{W}_x$ is the same as $\mathbf{w}_x$ reshaped into a M^2/M^2-matrix. We can now use the eigensystem to $\mathbf{W}_x$, i.e.

$$\mathbf{W}_x = \sum_k \lambda_k \mathbf{e}_k\mathbf{e}_k^T \tag{6.21}$$

and write the projection as

$$\mathbf{x}^T\mathbf{w}_x = \mathbf{i}_x^T\left(\sum_k \lambda_k \mathbf{e}_k\mathbf{e}_k^T\right)\mathbf{i}_x = \sum_k \lambda_k(\mathbf{i}_x^T\mathbf{e}_k)^2 \tag{6.22}$$

We now see that the projection of $\mathbf{x}$ onto $\mathbf{w}_x$ can be computed as projections of the image $\mathbf{i}_x$ onto $\mathbf{e}_k$ followed by a weighted sum of squares. Note that $\mathbf{e}_k$ can be viewed as a linear filter on the image.

If we can remove some of the terms in the sum, we can save a lot of computations. It would be tempting to keep the terms corresponding to the largest eigenvalues, but it turns out that

$$\left|E[\lambda_k(\mathbf{e}_k^T\mathbf{i}_x)^2]\right| \tag{6.23}$$

where $E[.]$ means expectation value, is a more relevant significance measure. It measures the average energy in the subspace defined by $\mathbf{e}_k$. It should be noticed that the two most significant terms coincide with the largest eigenvalue terms in the experiments described in this section.

Now we use this idea in a corner detection experiment:

6.3.3 Experiment setup

Figures 6.11, 6.12 and 6.13 show examples of training samples for the experiments. Each pair of training samples has the same corner orientation but differs in other properties. When using linear components of intensity values it is not possible to learn corner orientation invariant to corner color. Therefore, as a first experiment we only vary the corner angle within the pairs and keep the color constant, see figure 6.11. In the rest of the experiments both the corner angle and corner color varies, see figure 6.12 and 6.13. In the local orientation experiments a larger image size was used in order to avoid border effects when the local orientation was computed, see figure 6.13.

The noise-free data is only used as part of the evaluation. Gaussian noise was added to the image data before put into the learning system. It turns out that the noise actually helps the learning algorithm to find more smooth and robust vectors $\mathbf{w}_{xk}$. This is not surprising since the algorithm tries to find projections which are invariant to the noise (because the noise is not a common property) which implicates a low-pass characteristic of the projections.

The list below contains some facts about the training set:

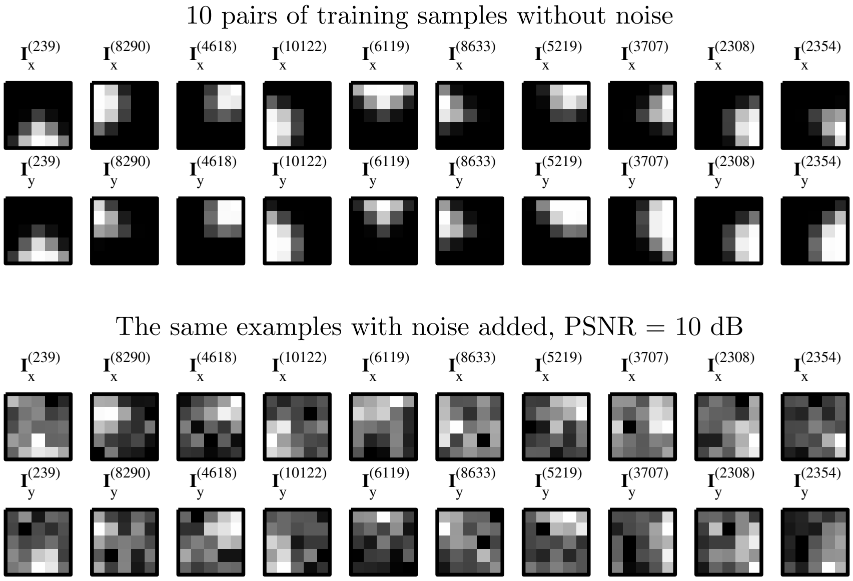

10 pairs of training samples without noise
$\mathbf{I}_x^{(239)}$ $\mathbf{I}_x^{(8290)}$ $\mathbf{I}_x^{(4618)}$ $\mathbf{I}_x^{(10122)}$ $\mathbf{I}_x^{(6119)}$ $\mathbf{I}_x^{(8633)}$ $\mathbf{I}_x^{(5219)}$ $\mathbf{I}_x^{(3707)}$ $\mathbf{I}_x^{(2308)}$ $\mathbf{I}_x^{(2354)}$
$\mathbf{I}_y^{(239)}$ $\mathbf{I}_y^{(8290)}$ $\mathbf{I}_y^{(4618)}$ $\mathbf{I}_y^{(10122)}$ $\mathbf{I}_y^{(6119)}$ $\mathbf{I}_y^{(8633)}$ $\mathbf{I}_y^{(5219)}$ $\mathbf{I}_y^{(3707)}$ $\mathbf{I}_y^{(2308)}$ $\mathbf{I}_y^{(2354)}$
The same examples with noise added, PSNR = 10 dB
$\mathbf{I}_x^{(239)}$ $\mathbf{I}_x^{(8290)}$ $\mathbf{I}_x^{(4618)}$ $\mathbf{I}_x^{(10122)}$ $\mathbf{I}_x^{(6119)}$ $\mathbf{I}_x^{(8633)}$ $\mathbf{I}_x^{(5219)}$ $\mathbf{I}_x^{(3707)}$ $\mathbf{I}_x^{(2308)}$ $\mathbf{I}_x^{(2354)}$
$\mathbf{I}_y^{(239)}$ $\mathbf{I}_y^{(8290)}$ $\mathbf{I}_y^{(4618)}$ $\mathbf{I}_y^{(10122)}$ $\mathbf{I}_y^{(6119)}$ $\mathbf{I}_y^{(8633)}$ $\mathbf{I}_y^{(5219)}$ $\mathbf{I}_y^{(3707)}$ $\mathbf{I}_y^{(2308)}$ $\mathbf{I}_y^{(2354)}$

Figure 6.11: Examples of pairs of training samples used for learning corner orientation invariant to corner angle when intensity information was used as input to the system.

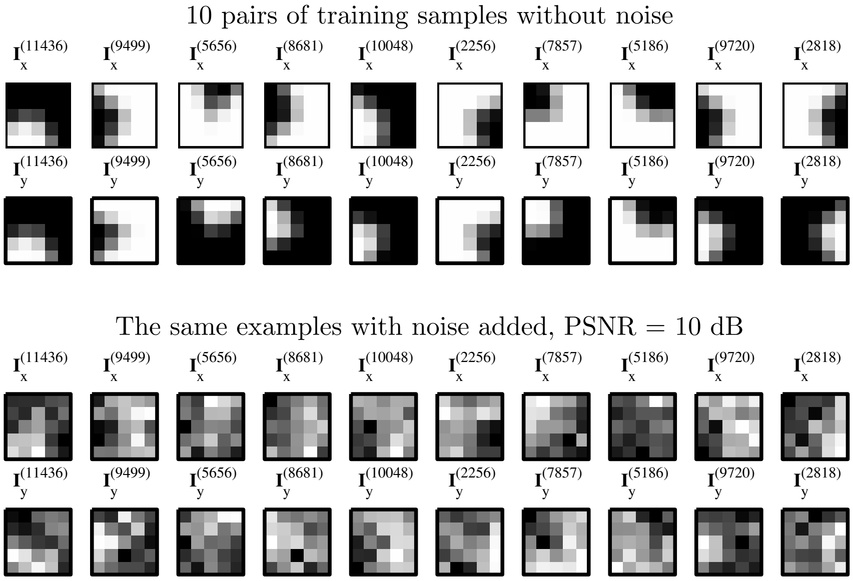

10 pairs of training samples without noise
$\mathbf{I}_x^{(11436)}$ $\mathbf{I}_x^{(9499)}$ $\mathbf{I}_x^{(5656)}$ $\mathbf{I}_x^{(8681)}$ $\mathbf{I}_x^{(10048)}$ $\mathbf{I}_x^{(2256)}$ $\mathbf{I}_x^{(7857)}$ $\mathbf{I}_x^{(5186)}$ $\mathbf{I}_x^{(9720)}$ $\mathbf{I}_x^{(2818)}$
$\mathbf{I}_y^{(11436)}$ $\mathbf{I}_y^{(9499)}$ $\mathbf{I}_y^{(5656)}$ $\mathbf{I}_y^{(8681)}$ $\mathbf{I}_y^{(10048)}$ $\mathbf{I}_y^{(2256)}$ $\mathbf{I}_y^{(7857)}$ $\mathbf{I}_y^{(5186)}$ $\mathbf{I}_y^{(9720)}$ $\mathbf{I}_y^{(2818)}$
The same examples with noise added, PSNR = 10 dB
$\mathbf{I}_x^{(11436)}$ $\mathbf{I}_x^{(9499)}$ $\mathbf{I}_x^{(5656)}$ $\mathbf{I}_x^{(8681)}$ $\mathbf{I}_x^{(10048)}$ $\mathbf{I}_x^{(2256)}$ $\mathbf{I}_x^{(7857)}$ $\mathbf{I}_x^{(5186)}$ $\mathbf{I}_x^{(9720)}$ $\mathbf{I}_x^{(2818)}$
$\mathbf{I}_y^{(11436)}$ $\mathbf{I}_y^{(9499)}$ $\mathbf{I}_y^{(5656)}$ $\mathbf{I}_y^{(8681)}$ $\mathbf{I}_y^{(10048)}$ $\mathbf{I}_y^{(2256)}$ $\mathbf{I}_y^{(7857)}$ $\mathbf{I}_y^{(5186)}$ $\mathbf{I}_y^{(9720)}$ $\mathbf{I}_y^{(2818)}$

Figure 6.12: Examples of pairs of training samples used for learning corner orientation invariant to corner angle and color (intensity) when intensity information was used as input to the system.

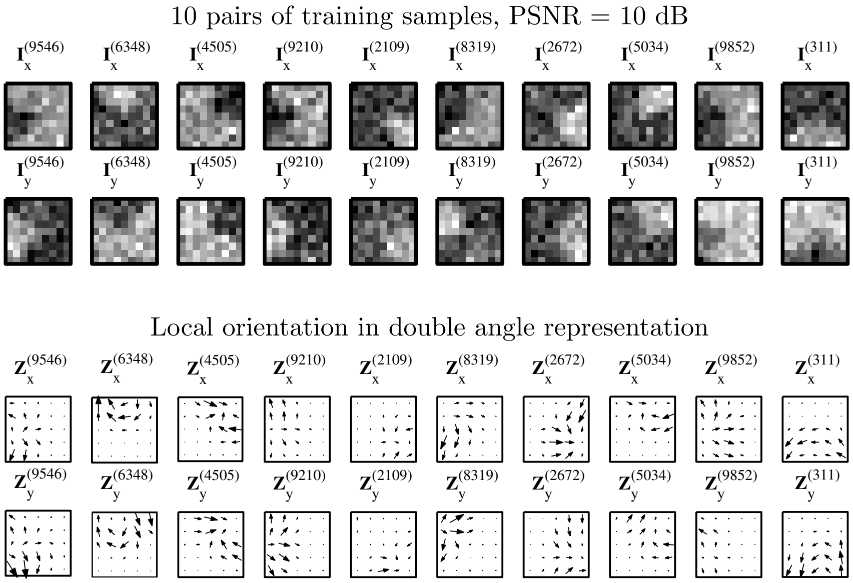


Figure 6.13: Examples of pairs of training samples used for learning corner orientation invariant to corner angle and color when local orientation information was used as input to the system.

- The orientation varies between 0° and 360° with a resolution of 5°, giving a total of 72 values.

- The corner angle varies between 60°-120° with a resolution of 5°, giving a total of 13 values.

- The pairs consist of all combinations of the images above that have the same corner orientation but differ in corner angle. For each corner orientation we have 13^2 combinations of corner angle pairs. This gives a total of $N = 72 \times 13^2 = 12168$ sample pairs.

- The corners were made from a product of two edges. The bottom of the edges has the value 0 and the top of the edge has the value 1. The transition from bottom to top was made soft with a width of 2 pixels using a $\cos^2$ function.

- Gaussian noise with PSNR=10 dB (definition, see C) was added to the data.

- In the color invariance case the corners were randomly inverted.

-
  - For the intensity value experiments an image size 5×5 was used.
  - For the local orientation experiments an image size 9×9 was used. The gradient (I_x, I_y) was then computed using 5×5 differentiated Gaussian

> filters **xg** and **yg** with $\sigma = 1.2$. The double angle representation is then computed as $z = (I_x + iI_y)^2$. The border values were finally removed leaving a 5×5 local orientation image.

As an evaluation measure 1000 images with random corner orientation, angle and color were used. Noise with PSNR=10 dB was added. The system then estimated a corner orientation for each of the images. The angular error was computed and the mean angular error was removed (the mean is not important since the system is unaware of the orientation reference value). The standard deviation of the error was finally computed.

6.3.4 Using intensity information

Linear CCA

As a first experiment we try to learn corner orientation invariant to corner angle. Intensity values are used as input data to the system, i.e.

$$\begin{cases} \mathbf{x}^{(n)} = \mathbf{i}_x^{(n)} = \mathrm{vec}(\mathbf{I}_x^{(n)}) \\ \mathbf{y}^{(n)} = \mathbf{i}_y^{(n)} = \mathrm{vec}(\mathbf{I}_y^{(n)}) \end{cases} \tag{6.24}$$

Thus we have 12168 pairs of 25-dimensional training vectors. The resulting canonical correlations are shown in figure 6.14. We get four quite large correlations. The absolute values are not critical as long as they are fairly high, they depend on the noise added.

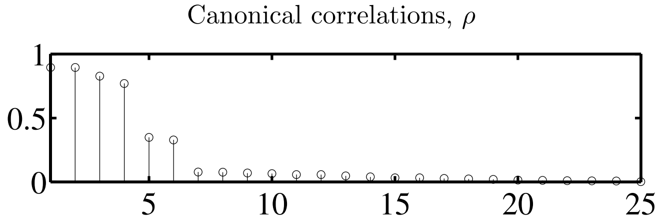


Figure 6.14: Resulting canonical correlations when using intensity value representation.

Figure 6.15 shows the six first canonical correlation vectors reshaped into matrices and their corresponding Fourier transforms. We can see that the first two vectors are some kind of edge filters, which is not too surprising. The figure also shows the projection of the noise-free data onto the vectors. The first two projections vary as sinusoidal functions with 90° phase difference. The next two are sensitive to the double angle of the orientation. Note that the projections are fairly invariant to corner angle.

To decode the projections into a corner orientation angle, we can simply take the angle of the vector $(\mathbf{w}_{x1}^T \mathbf{i}_x, \mathbf{w}_{x2}^T \mathbf{i}_x)$. This is illustrated in figure 6.16 on noise-free data. The estimate is even more invariant to corner angle than the projections

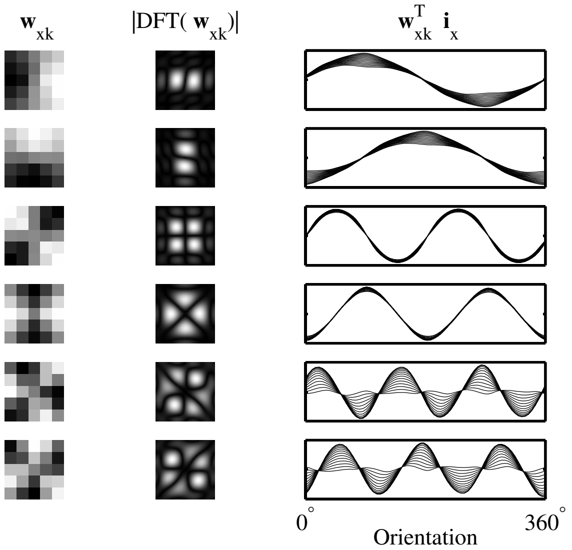


Figure 6.15: Left: The six first canonical correlation vectors, $\mathbf{w}_{x1}$ - $\mathbf{w}_{x6}$, reshaped into matrices. Middle: Their corresponding Discrete Fourier transform (absolute value). Right: Projection of the training data without noise onto the canonical correlations vectors. Note that there are several curves corresponding to different corner angles.

alone. The evaluation on 1000 noisy test images, described in the experiment setup section, is shown in figure 6.17.

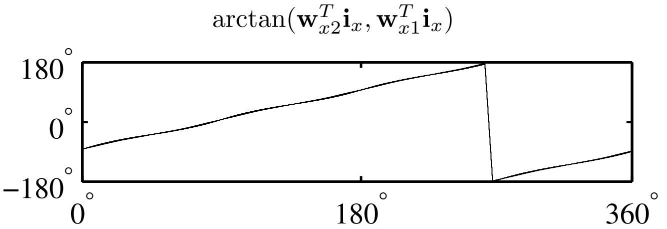


Figure 6.16: Decoding of canonical correlation projections into corner orientation on noise-free data. There are several curves, corresponding to different corner angles.

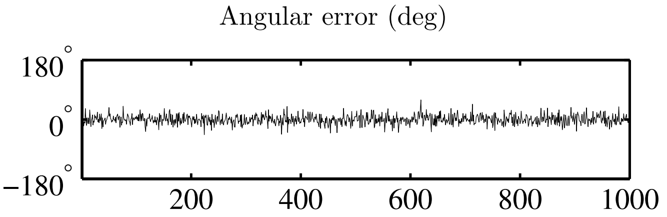


Figure 6.17: Evaluation of the decoding formula on noisy data. Standard deviation of the error was 13.8°.

Note that the resulting vectors are not designed to be selective to corners alone. For instance, the first two vectors also detect edges. But if several vectors are used, it can be possible to separate between different features. The third and the fourth vector can for example be used to distinguish corners from edges.

Linear CCA on polynomial model parameters

We now perform an experiment similar to the previous one, except that in this case we use parameters from a polynomial expansion of the images as input to the algorithm. The polynomial expansion is described in chapter 4, and we get

$$\begin{cases} \mathbf{x}^{(n)} = \mathbf{r}_x^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{i}_x^{(n)} \\ \mathbf{y}^{(n)} = \mathbf{r}_y^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{i}_y^{(n)} \end{cases} \tag{6.25}$$

where $\mathbf{P}_2$ is the second order polynomial basis and $\mathbf{W}$ is a diagonal weight matrix with a Gaussian in the diagonal. A standard deviation of 2 is chosen for the Gaussian. We now have 12168 pairs of 6-dimensional training vectors. The resulting canonical correlations are shown in figure 6.18.

Again we get four large correlations. Figure 6.19 shows the resulting canonical correlation vectors. To better understand what they are doing, we can interpret

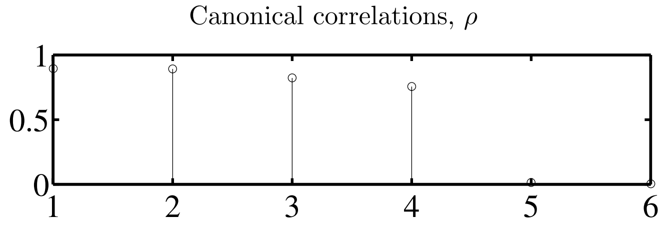


Figure 6.18: Resulting canonical correlations when using parameters from a polynomial expansion model of the intensity images.

them as filters on the image. We have

$$\mathbf{w}_x^T \mathbf{r}_x = \mathbf{w}_x^T (\mathbf{P}_2^T \mathbf{W} \mathbf{P}_2)^{-1} \mathbf{P}_2^T \mathbf{W} \mathbf{i}_x^{(n)} \tag{6.26}$$

Therefore

$$\mathbf{F}_{w_x} = \mathbf{W} \mathbf{P}_2 (\mathbf{P}_2^T \mathbf{W} \mathbf{P}_2)^{-1} \mathbf{w}_x \tag{6.27}$$

can be interpreted as the corresponding image filter. $\mathbf{F}_{w_{xk}}$ and the projections onto noise-free data are also shown in figure 6.19, and we can see that they more or less correspond to the previous experiment (figure 6.15).

The performance on noisy data can be seen in figure 6.21. The result is similar to the previous experiment (figure 6.17). The advantage of using this approach compared to the previous one is that we have reduced the input dimensionality from 25 to 6.

Quadratic CCA

We now try to learn corner orientation invariant to both corner angle and color. The training set is similar to the previous one, except that we also vary the corner color, see 6.12. As input data we use a quadratic model, i.e. products between intensity values:

$$\begin{cases} \mathbf{x}^{(n)} = \mathrm{vec}(\mathbf{i}_x^{(n)} \mathbf{i}_x^{(n)T}) \\ \mathbf{y}^{(n)} = \mathrm{vec}(\mathbf{i}_y^{(n)} \mathbf{i}_y^{(n)T}) \end{cases} \tag{6.28}$$

We have 12168 pairs of $5^4 (= 625)$-dimensional training vectors. The outer product matrix is symmetric, so in practice we can skip the elements below the diagonal and reduce the input dimensionality to $25(25+1)/2 = 325$. The 50 best canonical correlations are shown in figure 6.22.

We get six large correlations. The eigenvalues and the significance measures for these are shown in figure 6.23. Luckily, there are only two significant terms for each vector $\mathbf{w}_{xk}$, which means that the computations can be substantially reduced. The projection of $\mathbf{x}$ onto the correlations vectors approximated by their two most

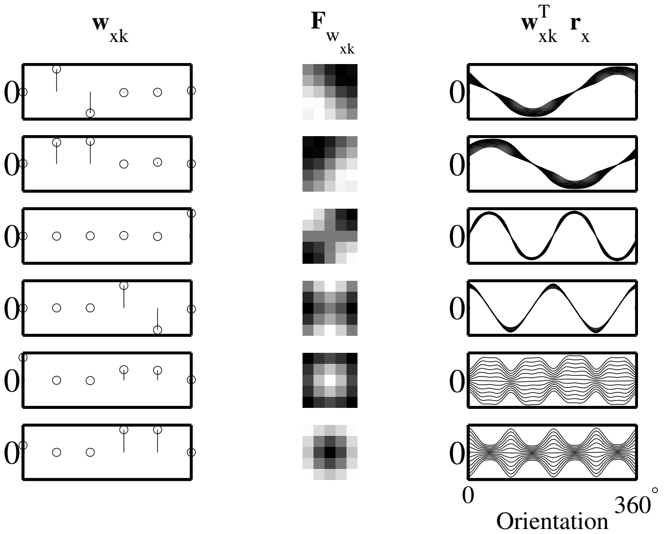


Figure 6.19: Left: The six first canonical correlation vectors, $\mathbf{w}_{x1}$ - $\mathbf{w}_{x6}$. Middle: Polynomial coefficients $\mathbf{w}_{xk}$ interpreted as filters on the original signal $\mathbf{i}$, $\mathbf{F}_{w_{xk}} = \mathbf{W}\mathbf{P}_2(\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{w}_{xk}$. Right: Projection of the training data without noise onto the canonical correlations vectors. Note that there are several curves corresponding to different corner angles.

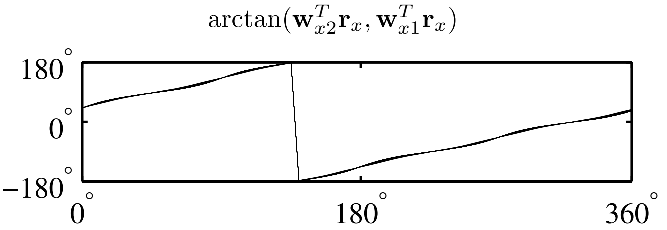


Figure 6.20: Decoding of canonical correlation projections into corner orientation on noise-free data.

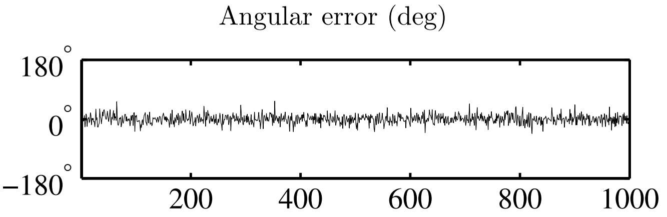


Figure 6.21: Evaluation of the decoding formula on noisy data. Standard deviation of the error was 13.5°.

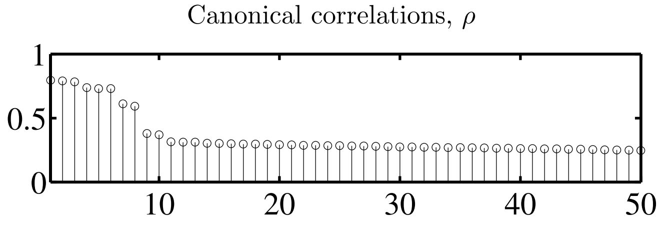


Figure 6.22: Resulting canonical correlations when using covariant components of intensity values. Only the first 50 of the 325 correlations are shown.

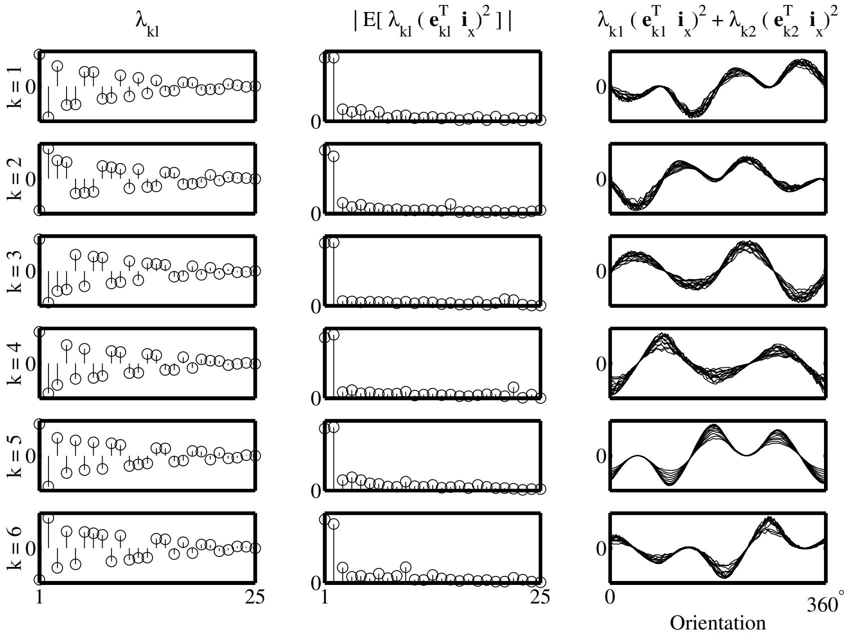


Figure 6.23: Result from the six largest canonical correlations. Left: Eigenvalues of $\mathbf{w}_{xk}$. Middle: Significance measure of each eigenvector. Right: Projection of the training data without noise onto $\mathbf{w}_{xk}$ approximated by its two most significant eigenvectors.

significant terms is also shown in figure 6.23. The true projections are not shown here but they do not differ significantly from the approximative ones.

The projections do not behave as nice as they did in the previous experiments, why it is more difficult to find a suitable decoding function. But the projections are fairly invariant to corner angle and color and thus it should theoretically be possible to find a decoding function. This is not further investigated in this thesis, though.

We can still make some interesting observations from this experiment. Figure 6.24 shows the two most significant eigenvectors together with their Fourier transform for each of the six largest correlation vectors. If closely examined, they could be interpreted as local edge filters! This suggests that local orientation could be a good choice of input data to the system. The image size is only 5 × 5 pixels and it can therefore be difficult for the system to 'invent' edge filters by itself, especially if the image is noisy. We could try to use larger images, but the complexity grows fast with increasing image size, and at present the computers runs out of memory.

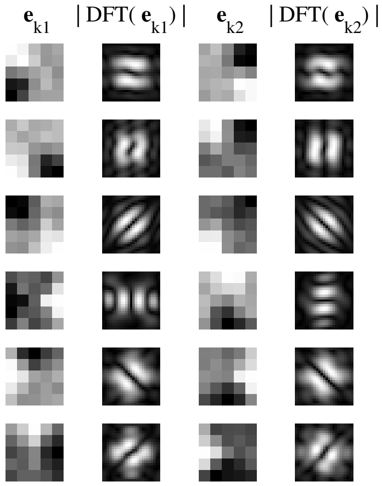


Figure 6.24: The two most significant eigenvectors and their Fourier transform for each of the six largest canonical correlation vectors.

Quadratic CCA on polynomial model parameters

For completeness, the same experiment as above can be performed using a quadratic model on parameters from a polynomial expansion model of the image. This means that we have

$$\begin{cases} \mathbf{x}^{(n)} = \mathrm{vec}(\mathbf{r}_x^{(n)}\mathbf{r}_x^{(n)T}) \\ \mathbf{y}^{(n)} = \mathrm{vec}(\mathbf{r}_y^{(n)}\mathbf{r}_y^{(n)T}) \end{cases} \quad \text{where} \quad \begin{cases} \mathbf{r}_x^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{i}_x^{(n)} \\ \mathbf{r}_y^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{i}_y^{(n)} \end{cases} \tag{6.29}$$

The dimensionality of the input data is now reduced from 325 to $6(6+1)/2 = 21$. The resulting canonical correlations are shown in figure 6.25. We get six (or maybe eight) large correlations.

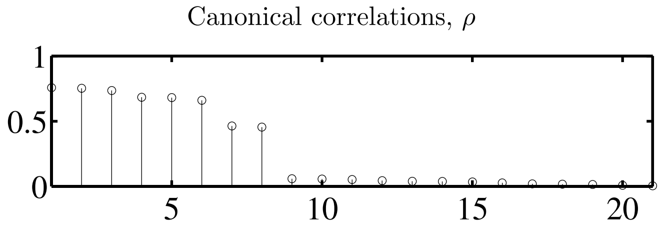


Figure 6.25: Resulting canonical correlations when using covariant components of parameters from a polynomial expansion model of the intensity images.

As before, we get two significant terms for each correlation vector, and the corresponding eigenvectors are shown in figure 6.26. They can be transformed to image filters by using equation 6.27. Again, they could be interpreted as local edge filters. The projections of the filters onto the signal (not shown here) resembles the ones in the previous experiment (figure 6.23) and the decoding function is still difficult to design.

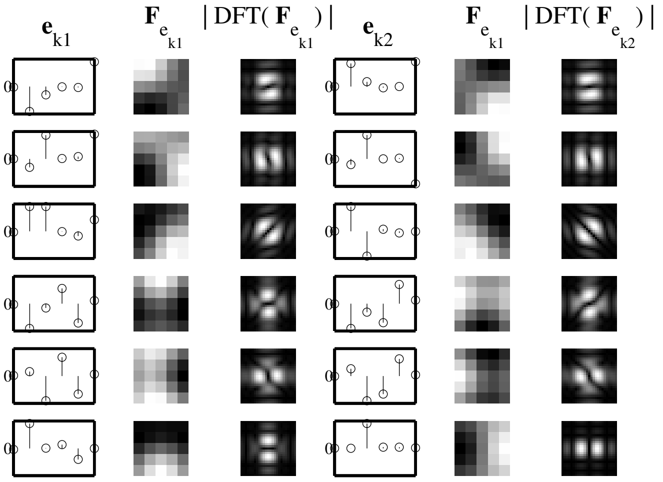


Figure 6.26: Column 1,4: The two most significant eigenvectors for each of the six largest canonical correlation vectors. Column 2,5: Polynomial coefficients $\mathbf{e}_{xk}$ interpreted as filters on the original signal $\mathbf{i}$, $\mathbf{F}_{e_{xk}} = \mathbf{W}\mathbf{P}_2(\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{e}_{xk}$. Column 3,6: Discrete Fourier transforms of the filters.

The last two experiments cannot easily be used to design a corner orientation estimator, but they can serve as a motivation to why we should use local orientation as input data to the learning system. The next two experiments therefore uses local orientation in double angle representation as input data.

6.3.5 Using local orientation in double angle representation

Linear CCA

We now use the training set in figure 6.13 as input to the system.

$$\begin{cases} \mathbf{x}^{(n)} = \mathrm{vec}(\mathbf{z}_x^{(n)}) \\ \mathbf{y}^{(n)} = \mathrm{vec}(\mathbf{z}_y^{(n)}) \end{cases} \tag{6.30}$$

We get 12168 pairs of complex valued 25-dimensional training samples (i.e. 50 real values per vector). A small amount of noise (PSNR=10 dB) was added to $\mathbf{x}$ and $\mathbf{y}$ before put into the system. This helped the system to find smoother vectors, $\mathbf{w}_x$, which were easier to interpret. The resulting canonical correlations are shown in figure 6.27. We get five fairly high correlations. The decrease in correlation for these five is partly due to the noise added, the difference is less obvious without the noise. Apart for the decrease, the noise did not change the behavior of the vectors significantly.

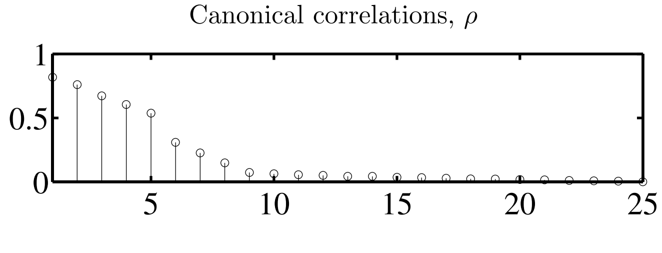
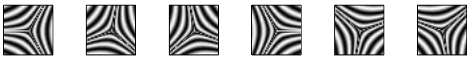
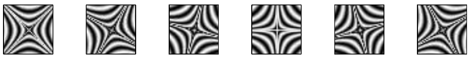
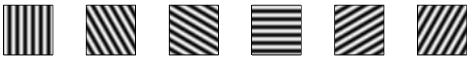


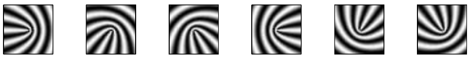
Figure 6.27: Resulting canonical correlations when using local orientation in double angle representation.

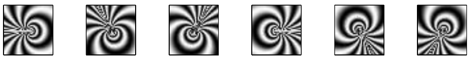
The five first complex valued correlation projections $\mathbf{w}_{x1}$-$\mathbf{w}_{x5}$ are shown in figure 6.28. Except for a constant offset phase they can be identified as rotational symmetry filters:

$\mathbf{w}_{k1} \sim e^{-1i\varphi} \iff$ 

$\mathbf{w}_{k2} \sim e^{-2i\varphi} \iff$ 

$\mathbf{w}_{k3} \sim e^{0i\varphi} \iff$  (6.31)

$\mathbf{w}_{k4} \sim e^{1i\varphi} \iff$ 

$\mathbf{w}_{k5} \sim e^{3i\varphi} \iff$ 

The gray-level patterns refer to the optimal prototype patterns for each rotational symmetry (c.f. section 5.2.2). All patterns that can be described as rotations or parts of the prototype pattern (e.g. one of the trajectories) are also detected by the corresponding filter. The projections of $\mathbf{x}$ onto the vectors $\mathbf{w}_{xk}$ are also complex valued, and the magnitude and phase of these projections are also shown in figure 6.28 (note that the offset phase for the first projection is approximately zero only by coincidence.). We can see that the phase varies very nicely with the orientation, while the magnitude is approximately constant but depends somewhat on the corner angle (i.e. the image energy).

Since the first and second projections are sensitive to the third and fourth multiple of the orientation respectively, we can for instance decode the projections into a corner orientation angle by simply taking the phase of the quotient $(\mathbf{w}_{k2}^*\mathbf{z}_x)/(\mathbf{w}_{k1}^*\mathbf{z}_x)$. We can also use the magnitudes of the projections as a certainty measure. The result of this decoding function on noise-free data is shown in figure 6.29 and on the evaluation data in figure 6.30. Another decoding function could simply be to use the phase of the fourth projection, $\mathbf{w}_{k4}$, since this is approximately the identity mapping. But the result would be less accurate, as can be inferred from the projection in figure 6.28.

Linear CCA on polynomial model parameters

Again we can use polynomial model parameters as input to the system as a way to reduce input dimensionality. This time we fit the model to the local orientation description $\mathbf{z}$:

$$\begin{cases} \mathbf{x}^{(n)} = \mathbf{r}_x^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{z}_x^{(n)} \\ \mathbf{y}^{(n)} = \mathbf{r}_y^{(n)} = (\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{W}\mathbf{z}_y^{(n)} \end{cases} \tag{6.32}$$

We get 12168 pairs of 6-dimensional complex valued training samples. The resulting canonical correlations are shown in figure 6.31.

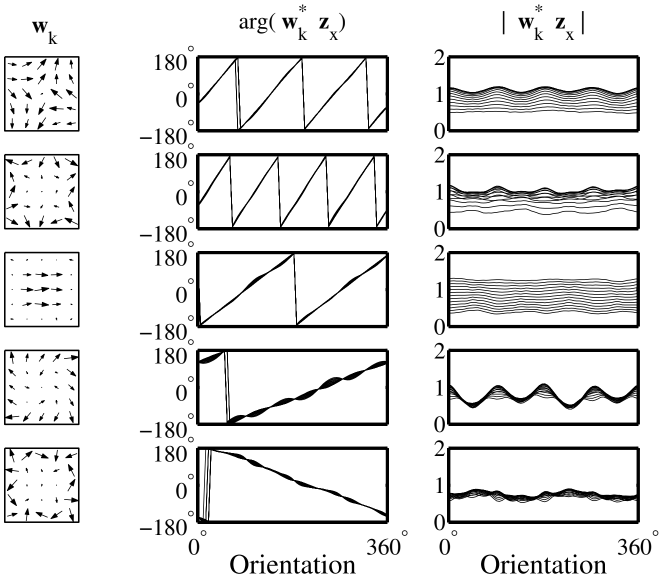


Figure 6.28: Left: The five largest canonical correlation vectors, $\mathbf{w}_{x1}$ - $\mathbf{w}_{x5}$ (reshaped into matrices). Middle and right: Angle and magnitude respectively of the projection of the training data without noise onto the canonical correlations vectors.

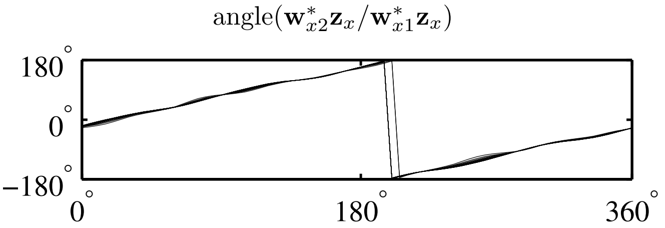


Figure 6.29: Decoding of canonical correlation projections into corner orientation on noise-free data.

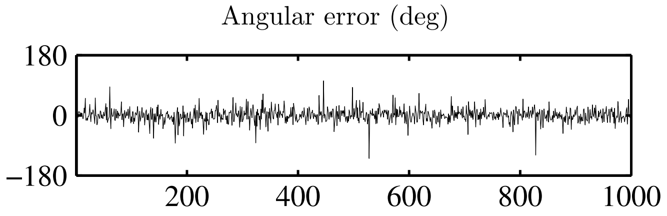


Figure 6.30: Evaluation of the decoding formula on noisy data. Standard deviation of the error was $\sigma = 19.5°$.

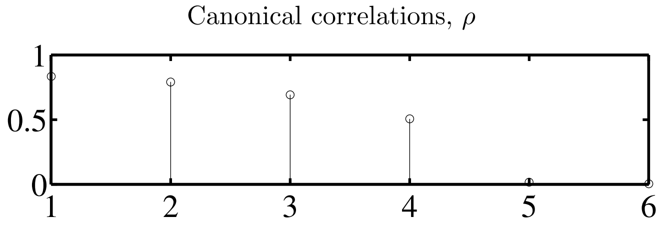


Figure 6.31: Resulting canonical correlations when using parameters from a polynomial expansion model of local orientation in double angle representation.

We have four fairly large correlations, and we can again visualize the result by converting the polynomial coefficients to corresponding filters on the local orientation image using equation 6.27. The result is shown in figure 6.32 together with the signal projections. The evaluation result is shown in figures 6.33 and 6.34. The result is very similar to the previous experiment. The fifth largest vector has changed from $e^{3i\varphi}$ to a non-significant vector, probably because the polynomial model cannot handle the fast variations in $e^{3i\varphi}$.

6.3.6 Discussion

It has often been argued, partly motivated by biological vision systems, that local orientation information should be used to detect more complex features. The experiments in this section is an attempt to further motivate this idea. It may be possible to use the result from the quadratic model experiments, but the local orientation helps the system to learn a more well behaved representation which is easier to decode.

It should be possible to use the idea in this section to learn other features and invariances. One drawback could be the amount of necessary training data. Preliminary experiments show that by using the polynomial model, the number of training pairs could be less than if we used the image or local orientation directly. This is because the number of training samples is generally proportional to the number of input parameters.

We do not have to use the double angle representation but it helps the system. Preliminary experiments show that it is sufficient to use a quadratic model of the gradient, i.e. in each position in the image we compute the vector (I_x^2, I_xI_y, I_y^2) and use them as input to the system. The practical details, including experiment setup, interpretation of the correlation vectors and decoding of the projections, become more complex and are therefore not included here. But this representation combined with the polynomial expansion to reduce dimensionality can be useful, for instance if we want to learn features in 3D-image volumes, where we do not have a definition of the double angle representation.

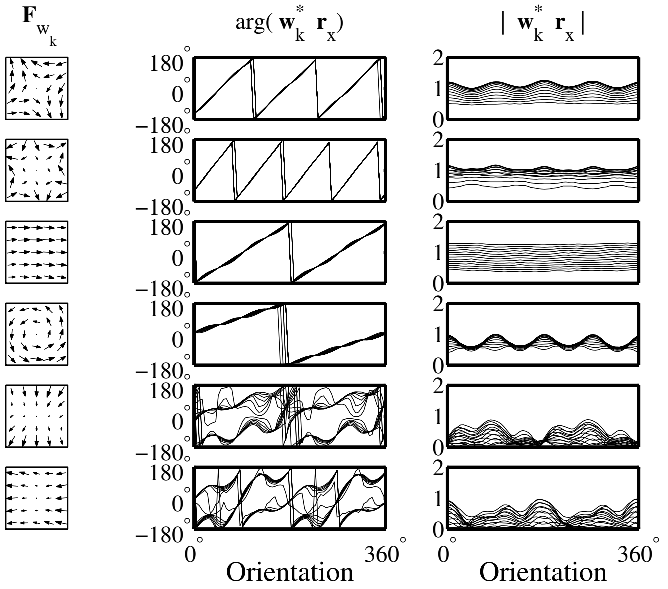


Figure 6.32: Left: The canonical correlation vectors $\mathbf{w}_{xk}$ interpreted as filters on the original signal $\mathbf{z}$, $\mathbf{F}_{w_{xk}} = \mathbf{W}\mathbf{P}_2(\mathbf{P}_2^T\mathbf{W}\mathbf{P}_2)^{-1}\mathbf{w}_{xk}$. Middle and right: Angle and magnitude respectively of the projection of the training data without noise onto the canonical correlations vectors.

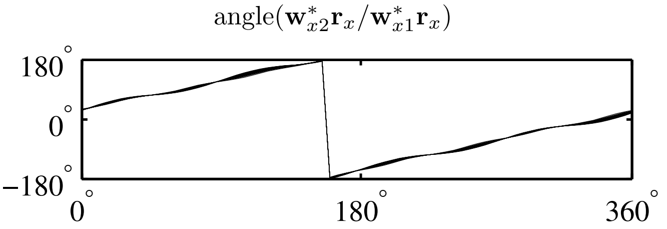


Figure 6.33: Decoding of canonical correlation projections into corner orientation on noise-free data.

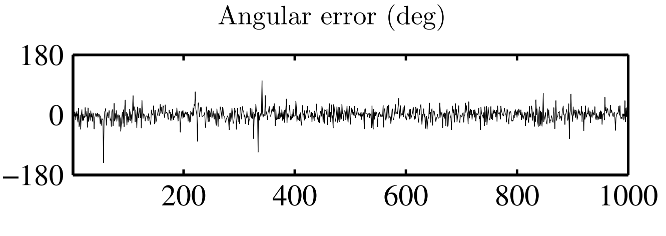


Figure 6.34: Evaluation of the decoding formula on noisy data. Standard deviation of the error was $\sigma = 18.8°$.

In chapter 5 we claim that the first order rotational symmetry, $e^{i\varphi}$, is a good descriptor for corners. It would be nice if this symmetry gave the highest correlation in the experiments above instead of only the fourth best correlation. One reason why this is not happening is that the first order symmetry is not designed to give a high response at the absolute peak of the corner, but instead gives the highest response somewhere inside the corner. If we want this symmetry to come in first, we should therefore show the learning system corners with a center position located some distance away from the image center.

6.4 Object recognition in the COIL database

6.4.1 Introduction

In order to illustrate how curvature can be used to recognize objects, an experiment is made on the Columbia Object Image Library, COIL-100, database[1]. This database consists of 7200 color images of 100 objects in 72 views (5° interval) each. The image size is 128×128. Figure 6.35 shows the 100 objects at the 0°-view.

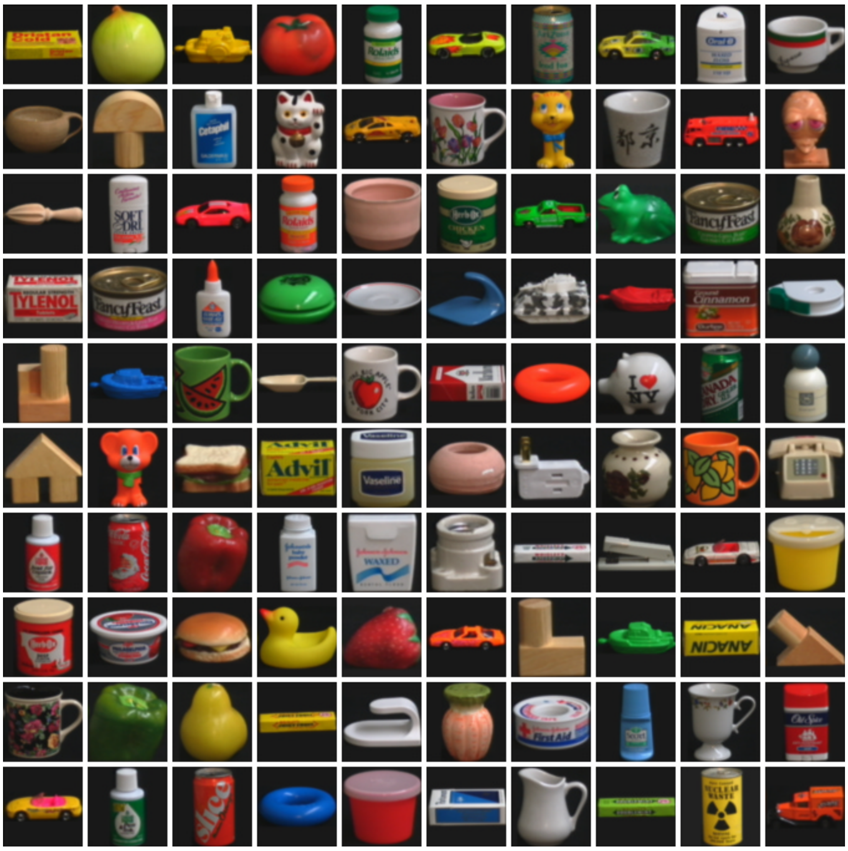


Figure 6.35: Columbia Object Image Library, COIL-100. The figure shows one of the 72 views for each of the 100 objects.

The conditions are quite controlled; there is only one object in each image, the light source is kept constant, etc. This database is too simple for a general evaluation of the performance of contents based search algorithms, but it serves well for small object recognition experiments.

[1] Available at `http://www.cs.columbia.edu/CAVE`

Method	# training views/object			
	36	18	8	4
SNoW, intensity, [Yang et al., 2000]	95.81	92.31	85.13	81.46
SNoW, edge, [Yang et al., 2000]	96.25	94.13	89.23	88.28
Linear SVM, [Pontil and Verri, 1998]	96.03	91.30	84.80	78.50

Table 6.3: Experimental results of some classifiers using the COIL-100 database. The numbers represent the correct classification rate in percent on the evaluation data.

Previous view-based recognition experiments on this database include projection of each object onto its eigenspace and learning of the object trajectory, [Nayar et al., 1996], learning of object-separation hyperplanes using Support Vector Machines (SVM), [Pontil and Verri, 1998] and [Yang et al., 2000], and Sparse Networks of Winnows (SNoW) [Yang et al., 2000]. The SNoW method tried to learn the hyperplane using either intensity data or edge information from 32×32 gray-scale images. The results of the SVM and SNoW techniques can be found in [Yang et al., 2000] and are recalled in table 6.3 for reference. The experiments use m views per object as training samples, and the remaining $72 - m$ views are used to evaluate the algorithm. The values in the table denote the fraction in percent of correctly classified evaluation images.

In the experiment in this section, different image representations are compared for their object recognition ability. The representations are:

- Image intensity, I
- Local orientation, z
- First and second order rotational symmetries, s_1 and s_2
- Phase histograms based on local orientation, H_z
- Phase histograms based on the first and second order symmetries, H_{s_1} and H_{s_2}.

The primary goal is not to compete with the results in table 6.3. These experiments include more sophisticated learning algorithms and a fair comparison is therefore difficult to make.

Section 6.4.2 introduces the concept of phase histograms. Section 6.4.3 contains details of the experiment and the result. Finally, section 6.4.4 discusses the result.

6.4.2 Phase histograms

Histograms is a technique to reduce the amount of data by only counting the presence of features and ignore their locations. Histograms have been widely used in computer vision, for example histograms of local orientation have been used as a descriptor of image content in image database search, see e.g. [Johansson, 2000b].

However, histograms of complex curvature such as rotational symmetries should be more descriptive since these features are more sparse and selective, but still contain a lot of information.

This section describes how to calculate histograms of rotational symmetries. The information in the symmetry responses s_n lies in the phase, $\angle s_n = \alpha$. Therefore we create a phase histogram by letting a bin in the histogram represent the amount of phase at a certain angle, or put in mathematical terms:

$$\begin{aligned} &\mathbf{H}(s_n) \in \mathbb{R}^N, \quad \text{where} \quad H_n(s_n) = \sum |s_n| u(\angle s_n) \\ &\text{and } u(\alpha) = \begin{cases} cos^2(\frac{N}{4}(\alpha - \frac{2\pi n}{N})) & \text{if } |\alpha - \frac{2\pi n}{N}| < \frac{2\pi}{N} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6.33}$$

Figure 6.36 illustrates the phase histogram idea. Figure 6.37 shows the u-function with $N = 8$ bins. The bins are a bit overlapping which makes the histogram more robust to rotation than if we had used non-overlapping bins.

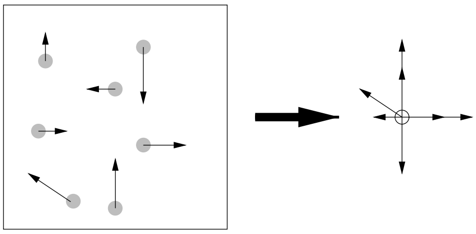


Figure 6.36: Illustration of phase histogram.

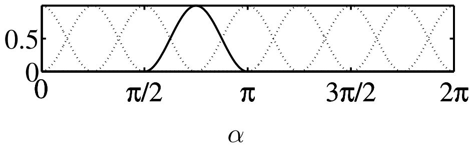


Figure 6.37: Overlapping bins using cos^2-bumps.

The histograms are an attempt to bridge the gap between the image domain and the semantic domain. A histograms on the first order symmetry is a fuzzy equivalence to the phrase “We have X_1 number of corners with orientation V_1, X_2 number of corners with orientation V_2, etc.”. Histograms on the second order symmetry correspond to the phrase “The image contains X number of circular patterns, Y number of star patterns, and Z number of spiral patterns”.

6.4.3 Experiment: Nearest neighbor classification

Figure 6.38 and 6.39 shows the different features computed on one of the views of object 60. The list below contains some computational details:

- Intensity, I: The color images are converted to gray-level images, and are down-sampled in five scales using Gaussian filters with $\sigma = 1.2$. The object contour is close to the image border. Therefore, to avoid border effects, normalized convolution with zero certainty outside the image border was used for the Gaussian filtering.

- Local orientation, z: The image gradient ∇I was estimated on the intensity images in each of the five scales. The gradient was computed from a first degree polynomial model as in the example in section 4.3. The Gaussian applicability had $\sigma = 1.2$. Again, zero certainty was used outside the image border. Finally, the double angle representation was computed as $z = |\nabla I| e^{i2\angle \nabla I}$.

- Phase histograms based on local orientation, H_z: For each scale, a global phase histogram was computed using equation 6.33 and $N = 8$ bins. The z-values were thresholded, removing all values below 10%, before the histograms were computed.

- First and second order rotational symmetries, s_1 and s_2: The rotational symmetries were computed using the multiscale algorithm in section 5.3.4, i.e. the approximative multiscale polynomial expansion algorithm in 4.5 together with a normalized inhibition scheme to make the responses more selective. The 128×128 z-image was used as input to the algorithm and the symmetries were computed in 5 scales using $\sigma = 1.2$ in the first scale.

- Phase histograms based on the first and second order symmetries, H_{s_1} and H_{s_2}: The symmetry images were thresholded, removing all values below 10%, and then non-max-suppressed (set all non-peak values to zero) before the phase histograms were computed.

To calculate all the features in figures 6.38 and 6.39 takes about 1.6 seconds on a 440 MHz SUN Ultra 10. Of these, 0.05 seconds are spent on the computing the intensity low-pass hierarchy, 0.3 are spent on computing local orientation for each scale, 0.35 seconds are spent on computing histograms of the local orientation, 0.8 seconds are spent on computing the multiscale rotational symmetries responses, and 0.1 seconds are spent on computing histograms of the symmetries (the symmetry histograms were faster to compute than the local orientation histograms because of the sparsity property of the symmetry responses).

A simple nearest neighbor classifier is used in the evaluation. Two different distance measures are used; Euclidean distance and angular distance. Each evaluation image was classified to the same object as the nearest training image. For practical reasons (memory storage and computational complexity), only scales 32×32, 16×16, and 8×8 were evaluated for the non-histogram representations.

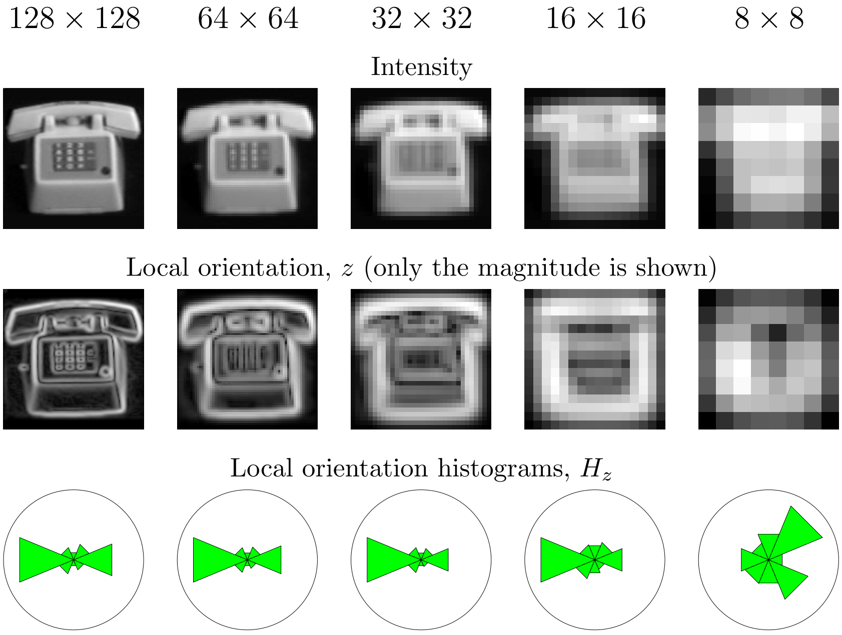

128 × 128
64 × 64
32 × 32
16 × 16
8 × 8
Intensity
Local orientation, z (only the magnitude is shown)
Local orientation histograms, H_z

Figure 6.38: Features generated from one view of object 60 in the COIL database.

In the histogram case, different combinations of scales and features were also evaluated. It was found that the histogram approach was improved if each element in the feature vector was normalized with its standard deviation before calculating the Euclidean distance. The reason is that the different features and scales have different max responses and therefore some features will have larger impact than others without the normalization.

The result using the Euclidean distance is shown in table 6.4 and the result using the angular distance is shown in table 6.5. Some of the results using the Euclidean distance are also visualized in figure 6.40. The tables also show the dimensionality of each representation (labeled as "# param") to illustrate the complexity of each representation.

6.4.4 Discussion

The experiment above indicates that high level features and their histograms are useful for object recognition, especially the first order symmetry which corresponds to high curvature points. Some observations can be made from the results:

- It is difficult to say whether the Euclidean or the angular distance measure performs better. For the non-histogram representations, it seems as if the Euclidean distance is better than the angular distance when the number of training views is large and the image resolution is low and vice versa. For the histogram representation the Euclidean distance outperforms the angular

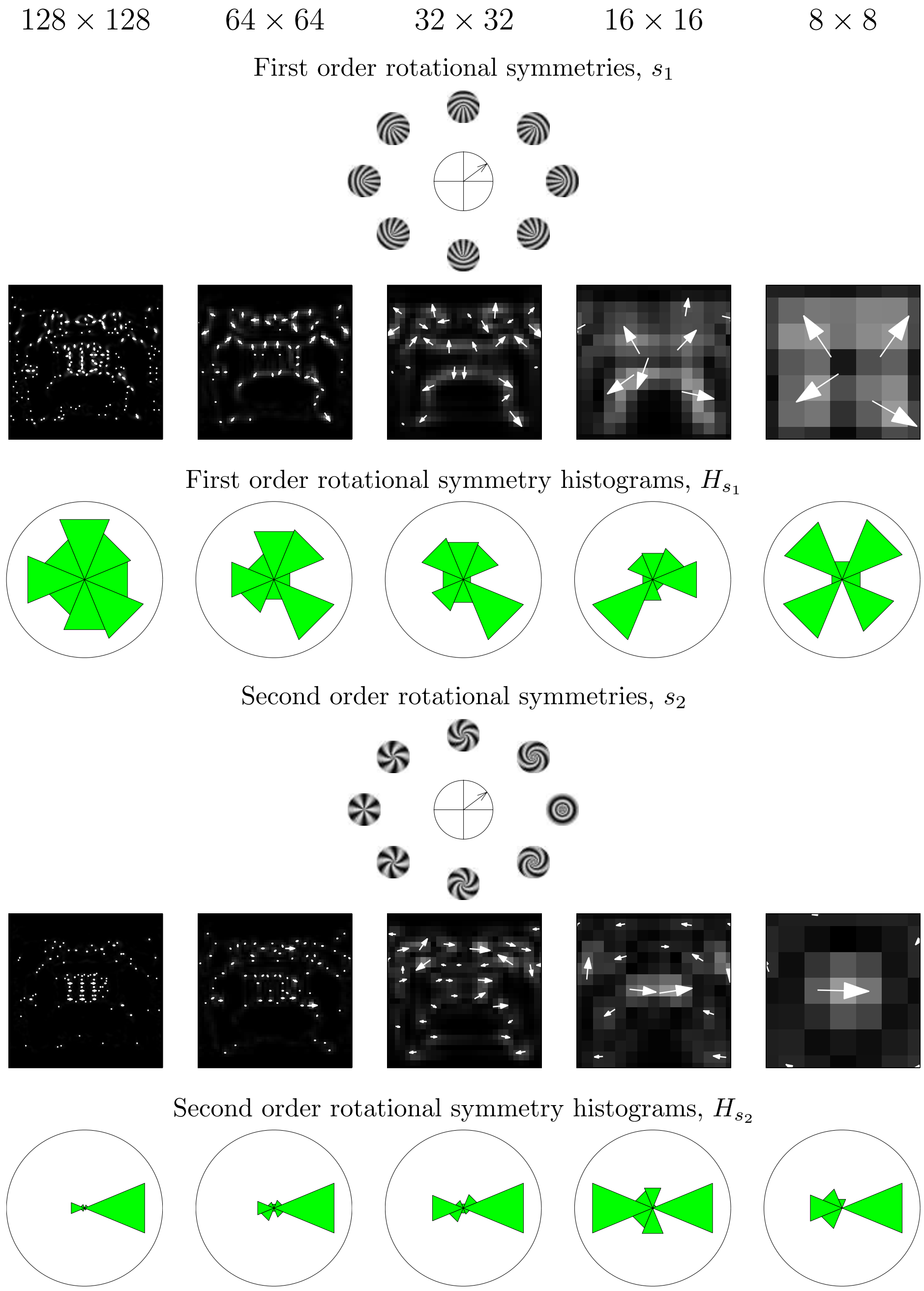

128 × 128
64 × 64
32 × 32
16 × 16
8 × 8
First order rotational symmetries, s_1
First order rotational symmetry histograms, H_{s_1}
Second order rotational symmetries, s_2
Second order rotational symmetry histograms, H_{s_2}

Figure 6.39: Features generated from one view of object 60 in the COIL database. The intensities represent the response magnitude and the vectors represent the complex values at the local maxima points.

Representation	# param	# training views/object			
		36	18	8	4
I, 32×32	1024	98.17	96.31	87.91	71.03
I, 16×16	256	98.17	96.37	88.66	72.07
I, 8×8	64	97.86	95.72	87.16	71.37
z, 32×32	2048	97.86	95.59	84.86	68.09
z, 16×16	512	98.08	96.33	87.78	70.43
z, 8×8	128	97.56	95.39	85.78	67.16
s_1, 32×32	2048	97.28	94.24	76.31	56.72
s_1, 16×16	512	98.33	97.11	87.22	67.15
s_1, 8×8	128	98.28	97.22	88.25	70.28
s_2, 32×32	2048	96.08	91.24	69.62	51.82
s_2, 16×16	512	98.19	96.78	84.30	63.69
s_2, 8×8	128	98.47	97.09	87.25	69.62
H_z, scale 1	8	88.36	82.78	66.67	49.85
H_z, scale 2	8	86.25	80.96	65.16	50.40
H_z, scale 3	8	87.42	81.54	64.39	48.07
H_z, scale 4	8	85.08	78.30	61.22	45.49
H_z, scale 5	8	83.03	76.22	60.86	45.29
H_z, scales 1–5	40	94.86	90.70	77.94	62.26
H_{s_1}, scale 1	8	54.50	47.67	37.77	29.06
H_{s_1}, scale 2	8	63.53	54.11	42.64	31.85
H_{s_1}, scale 3	8	62.94	52.96	40.22	26.25
H_{s_1}, scale 4	8	68.03	55.78	40.80	30.96
H_{s_1}, scale 5	8	73.81	62.28	47.03	34.74
H_{s_1}, scales 1–5	40	99.58	97.85	86.38	65.26
H_{s_2}, scale 1	8	34.25	29.89	25.17	20.44
H_{s_2}, scale 2	8	45.25	38.17	31.05	23.62
H_{s_2}, scale 3	8	45.19	37.46	27.62	21.85
H_{s_2}, scale 4	8	49.64	39.74	28.08	21.06
H_{s_2}, scale 5	8	65.00	52.06	37.84	27.82
H_{s_2}, scales 1–5	40	98.50	94.30	78.59	57.93
$H_z \& H_{s_1}$, scales 1–5	80	98.81	97.26	87.55	69.82
$H_{s_1} \& H_{s_2}$, scales 1–5	80	99.81	99.04	90.44	71.15
$H_z \& H_{s_1} \& H_{s_2}$, scales 1–5	120	99.08	98.31	90.28	73.00

Table 6.4: Nearest neighbor using Euclidean distance. Experimental results of some classifiers using the COIL-100 database. The numbers represent the correct classification rate in percent on the evaluation data.

Representation	# param	# training views/object			
		36	18	8	4
I, 32 × 32	1024	97.83	96.19	88.12	71.44
I, 16 × 16	256	97.81	96.50	88.83	72.60
I, 8 × 8	64	97.56	95.81	88.02	71.71
z, 32 × 32	2048	97.42	95.37	86.38	69.47
z, 16 × 16	512	97.33	95.89	87.08	69.47
z, 8 × 8	128	96.53	94.22	82.69	63.75
s_1, 32 × 32	2048	97.17	95.39	84.95	65.49
s_1, 16 × 16	512	98.11	97.09	88.61	70.22
s_1, 8 × 8	128	98.00	96.69	86.91	68.24
s_2, 32 × 32	2048	97.11	94.24	81.59	62.19
s_2, 16 × 16	512	98.14	96.83	87.53	68.28
s_2, 8 × 8	128	98.31	96.81	87.02	69.41
H_z, scale 1	8	81.11	74.89	58.70	41.99
H_z, scale 2	8	79.08	73.07	57.16	42.24
H_z, scale 3	8	80.47	73.24	55.41	39.99
H_z, scale 4	8	75.83	68.50	51.52	38.04
H_z, scale 5	8	73.39	65.39	49.62	37.03
H_z, scales 1–5	40	93.11	89.22	75.23	58.09
H_{s_1}, scale 1	8	29.56	25.11	20.20	15.72
H_{s_1}, scale 2	8	38.56	32.15	24.52	18.21
H_{s_1}, scale 3	8	43.33	34.76	26.25	18.15
H_{s_1}, scale 4	8	54.44	42.67	31.20	23.94
H_{s_1}, scale 5	8	60.58	49.96	36.72	26.71
H_{s_1}, scales 1–5	40	99.17	96.98	83.38	61.00
H_{s_2}, scale 1	8	19.56	17.65	14.67	11.82
H_{s_2}, scale 2	8	23.33	18.83	16.44	12.85
H_{s_2}, scale 3	8	24.72	19.80	15.03	11.85
H_{s_2}, scale 4	8	35.33	26.59	19.06	14.74
H_{s_2}, scale 5	8	53.28	42.50	30.89	22.51
H_{s_2}, scales 1–5	40	97.75	92.78	74.83	54.28
H_z&H_{s_1}, scales 1–5	80	98.61	97.17	87.09	67.57
H_{s_1}&H_{s_2}, scales 1–5	80	99.81	99.06	89.50	69.51
H_z&H_{s_1}&H_{s_2}, scales 1–5	120	99.19	98.33	90.20	71.79

Table 6.5: Nearest neighbor using angular distance. Experimental results of some classifiers using the COIL-100 database. The numbers represent the correct classification rate in percent on the evaluation data.

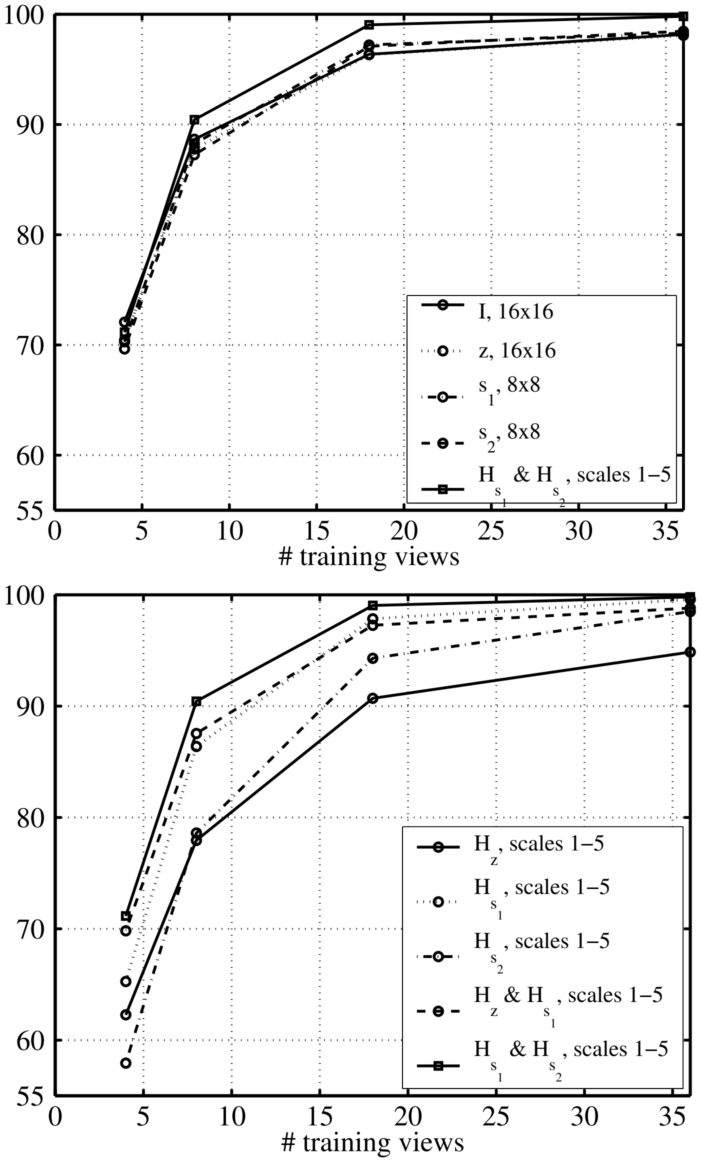


Figure 6.40: Visualization of some of the results in table 6.4. The numbers represent the correct classification rate in percent on the evaluation data.

distance in all cases, but the difference is less significant when different scales or features are combined.

As stated before, the experiment setup is quite controlled. The angular distance measure should give a better performance than the Euclidean distance when the lighting conditions varies.

- The combination of first and second order phase histograms gives the best performance, as can be seen in figure 6.40. Second place is shared by some of the non-histogram representations together with the histogram representations based on local orientation and the first order symmetry combined.
- The representations also differ in memory requirements. The representations based on a combination of $H_{s_1^p}$ and $H_{s_2^p}$ have a quite good performance and each image is represented by a feature vector of length 80, which is a fairly compact representation of an image.

  Also, the symmetry representations, s_1 and s_2, are sparse which means that the memory requirements for these representations are smaller than gray-level or orientation representations of the same size.

The performance of the highest ranked representations do not differ significantly. The difference may be more obvious using a more complex database. Still, the conclusion is that the combination of phase histograms from the symmetries works at least as well as the non-histogram feature representations for this database. In addition, the histograms have some advantages over the non-histogram representations that do not appear in this experiment. They are invariant to translation as long as the whole object is still in the image and should perform better than the non-histogram representations when the location of the object varies. It is also easier to match images invariant to rotation when we use the histograms. For example, we can calculate the Fourier series of the histograms, and then use the magnitude of the Fourier coefficients as a rotation invariant descriptor for the image, see e.g. [Forssén and Johansson, 2000]. Also, the local orientation and symmetry representations and their histograms are more invariant to lightning conditions than the intensity representation because they only depend on differences between intensity values.

Worth noticing is that the combined histograms of the first order symmetry (H_{s_1}, scales 1–5) performs much better than the combined histograms of the local orientation (H_z, scales 1–5), see figure 6.40. This suggests that histograms based on higher level features such as curvature have a higher descriptional power than histograms based on low level features such a intensity and local orientation. The superior performance when using high level features can also be confirmed from the SNoW results in table 6.3. The edge representation performs better than the intensity representation, which seems to indicate that higher level features are better suited for object recognition. The nearest neighbor classifier used in this experiment is however very inefficient and future work includes exploring how symmetry features can be combined with learning procedures to improve the performance in terms of speed and memory storage. For example, the SNoW architecture claims to be well suited for sparse features. However, the representations used in the SNoW experiments were designed mainly with the sparse property in mind. For example, the intensity information was representation by a binary vector, each element corresponded to a certain integer intensity value at a certain position. This representation lacks metric and should be less robust to changes in the environment.

There are a number of issues left open for discussion. One example is the choice of the number of bins for the histograms. This choice is a trade-off between selectivity and invariance to different transformations, e.g. rotation, deformation, etc. Another example is how to choose a suitable scale of the local orientation feature when we want to detect rotational symmetries of a certain other scale.

6.5 Other possible applications

This section presents two applications where rotational symmetries (chapter 5) are being under investigation.

6.5.1 Detecting landmarks in aerial images

The goal of the WITAS-project, see [WITAS web page,] and [Doherty et al., 2000], is to design an autonomous aircraft for traffic surveillance. One of the research issues is generation of landmark features for use in navigation. The features can be used in template matching or local histogram matching to help the aircraft find landmark objects to establish its position.

Rotational symmetries are one example of interesting landmarks which are currently under investigation. Figure 6.41 shows an aerial image and its corresponding local orientation magnitude $|z|$. From the orientation image z rotational symmetries are detected in several scales. The result is shown in figure 6.41. The left column shows the magnitude of the first order symmetries. The middle and right columns are computed as the negative and positive real part respectively of the second order symmetry responses. The first order symmetry can be used to find high curvature points, and the second order symmetry can be used to find crossroads, traffic circles, houses, etc. The rotational symmetry features are more robust to illumination and seasonal variations than the original gray-level image which makes them suitable in a matching process. In spite of the apparent complexity, the particular pattern of such features turns out to be quite specific. Such a higher level matching is likely to take place in human vision as well.

6.5.2 Autonomous truck

An early version of the rotational symmetry detector was used in a robotics project, where the goal was to have an autonomous truck locate a pallet and pick it up. Details can be found in the Masters Thesis by Roll, see [Roll, 1999].

Figure 6.42(a) shows an image of a pallet. Figure 6.42(b) shows the corresponding local local orientation magnitude. The rotational symmetries were detected in several scales and the results were combined into one scale-invariant description in order to reduce the amount of data. The second order symmetry filter gives characteristic blobs with zero phase on the pallet indicating circular-like patterns (in this case squares). The circular patterns can be extracted from the second order response by e.g. taking the real part and ignoring negative values, the result is shown in figure 6.42(c). The pallet has a characteristic signature in the shape of a number of blobs. The circle-blob image can be further processed using the Hough transform to find 'lines' formed by the blobs, and the correct line can be found by testing against the characteristic signature hypothesis. The optimal line is shown in figure 6.42(c), and the values along the line is shown in figure 6.42(d). In spite of the complexity of the image, the linear set of circular structures turns out to be quite unique. When the pallet is found in the image, it is possible to calculate

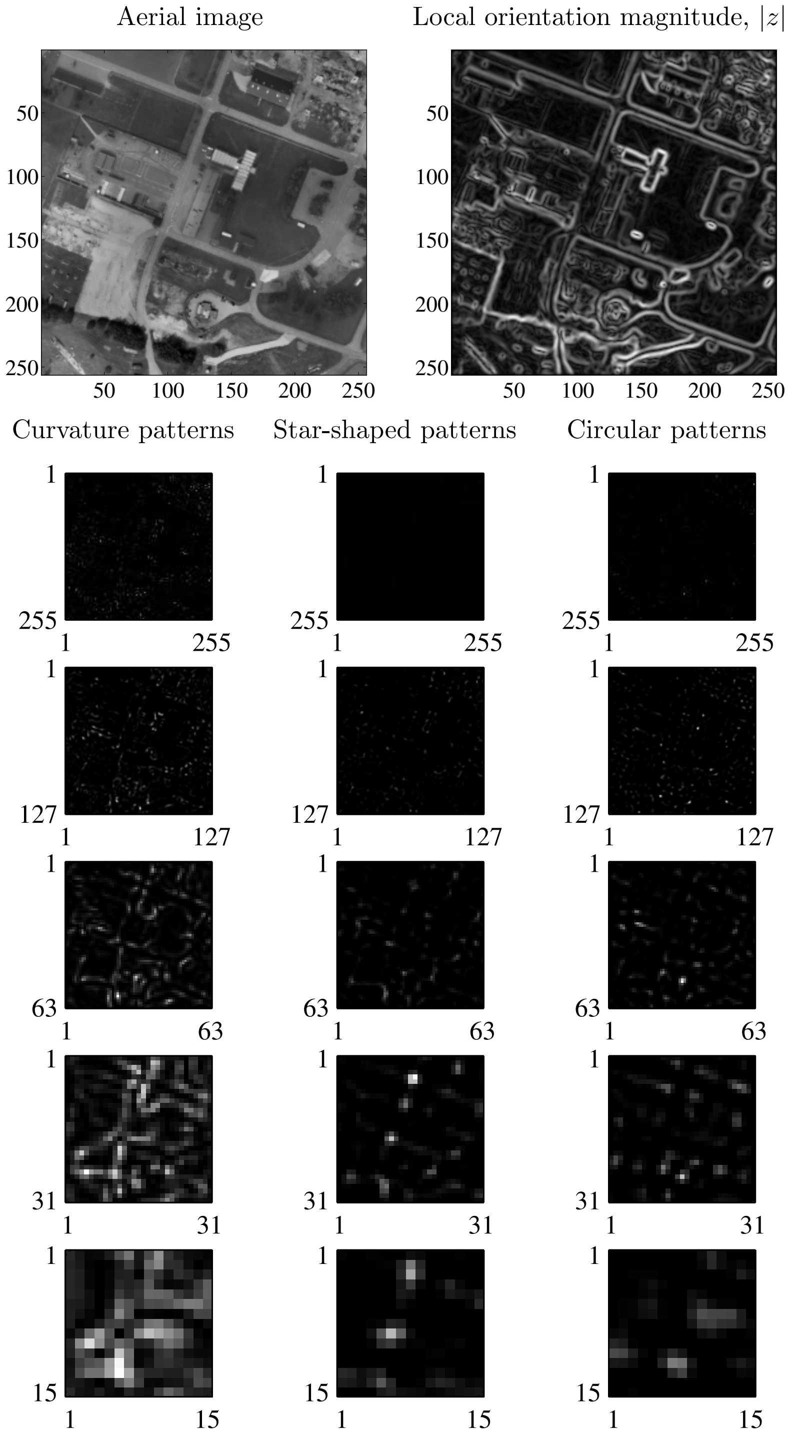


Figure 6.41: Detection of landmarks in an aerial image.

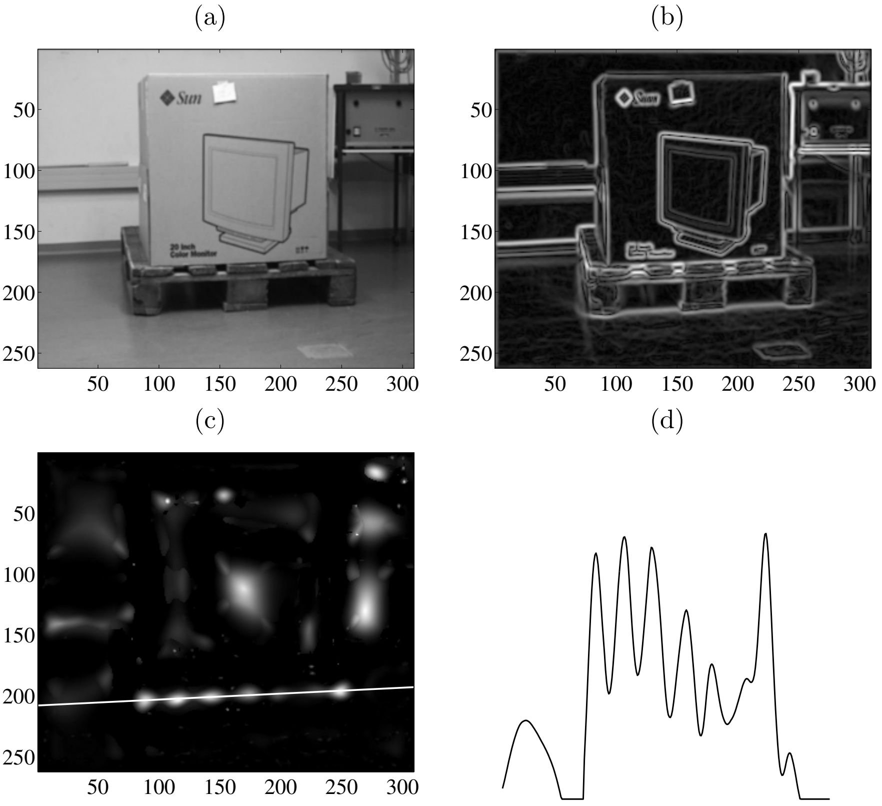


Figure 6.42: (a) Image of a pallet. (b) corresponding local orientation magnitude $|z|$ (c) Circular-like patterns extracted from second order rotational symmetry response in several scales. The circle features form a line at the pallet position. (d) characteristics along the line in (c) fulfilling structural hypothesis for the desired object.

pallet direction, distance and orientation in the real world using the information in the symmetry image and the local orientation image.

This technique has been tested on 28 images containing pallets. The distance between the camera and the pallet varied in the range of 1 to 16 meters and the orientation of the pallet relative to the camera varied between 0 to 40 degrees. The result was quite promising overall, but the number of test images is too small to reach a final conclusion. Details can be found in [Roll, 1999].

Chapter 7

Summary

7.1 Summary and discussion

This thesis describes a new method of detecting curvature in several scales. The method is based on a local polynomial model applied to a local orientation description, and the theory of rotational symmetries is used to compute curvature responses from the model parameters. The responses are made more selective using an inhibition scheme.

An approximative algorithm for multiscale polynomial expansion is developed which gives a very efficient multiscale curvature detection algorithm. This is very important in order for the algorithm to be practically useful. The approximative algorithm is evaluated in an experiment to estimate local orientation on 3D data, and the performance is comparable to previously tested algorithms which are more computationally expensive.

The curvature algorithm is evaluated in an object recognition experiment. Different image representations are compared, and it is shown that histograms of curvature give the best performance.

The importance of curvature is furthermore motivated by examples from biological and perceptual studies, and the usefulness of local orientation information to detect curvature is also motivated by an experiment about learning a corner detector.

The object recognition experiment shows that curvature is a very useful feature, but the experiment is made under a controlled environment. There remain a number of unsolved issues before a practical system for object recognition is a reality, for example how to handle multiple objects, occlusion, different scales, deformations, etc., i.e. the classical computer vision problems. One problem with many of the existing feature detectors used today is their poor selectivity, i.e. they are unable to distinguish between perceptually very different patterns. Their ability to describe the scene is therefore very crude. We would get a more ample description of the scene if we could distinguish between different features. The rotational symmetries in several scales is one example. Handling all these different features seems to be difficult in practice, but each feature map is quite sparse, i.e. few responses are active simultaneously. In recent years, interest in sparse features has increased and strategies to handle this kind of features are emerging.

7.2 Future research

Future research includes further investigation of the use of rotational symmetry features and other complex features for object recognition and content based image retrieval. Two ideas in this direction are discussed here.

7.2.1 Other features: Color and curvature combined

Perceptual experiments indicate that colors close to boundaries are important. A color illusion can be found [Cornsweet, 1970] and is recalled in figure 7.1. The color inside the circle seems darker than the color outside the circle, but this is only an illusion. The colors are the same except close to the circle border. The illusion is less obvious if we have a straight boundary, for example two areas separated by an edge. This is an indication that our notion of object color depends on the color close to high curvature points. A combined color and curvature descriptor should therefore be very useful. This will be a subject for future research.

7.2.2 Associative networks for sparse features

The object recognition experiment on the COIL database in section 6.4 shows promising results for the rotational symmetry features. The nearest neighbor classifier used in this experiment is, however, very inefficient, and future work includes exploring how symmetry features can be combined with learning procedures to improve the performance. The rotational symmetry features have a sparse behavior. This means that we can handle a large number of different features in a number of scales without running out of memory because most of the values are zero. This is not the case for intensity maps and other dense maps. There are learning structures specifically designed for sparse features, the SNoW structure mentioned in section 6.4 being one example. Another example is the associative structure developed by Granlund, see [Granlund, 2000]. Preliminary experiments show that using the later structure together with complex curvature features can give powerful descriptors.

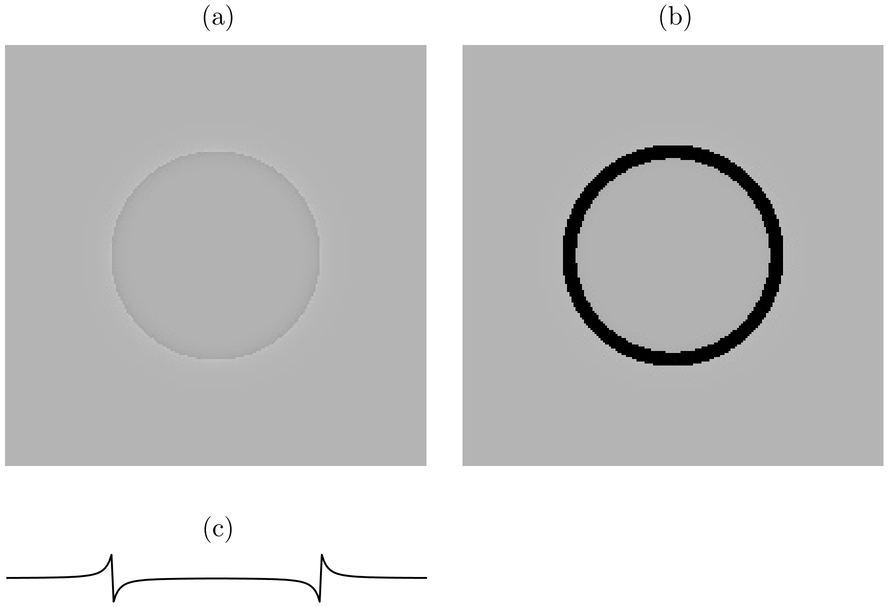


Figure 7.1: Color illusion. (a) Image with a circle. The color is the same inside and outside the circle except close to the circle border. (b) The same image as in (a) but the circle border is covered, and the color illusion disappears. (c) Intensity variation along the middle horizontal line.

Appendices

A Back-projection of rotational symmetries

This section describes the back-projection method mentioned in section 5.2.2. The method is further explained in [Johansson, 2000a]. The question was: given a description on local orientation, what class of image patterns does this description corresponds to? The question is answered for the class of rotational symmetry patterns.

This 'inverse' is of course not unambiguous, but we get a hint by making the assumption that the image gradient is parallel to the dominant orientation. This section contains a great deal of calculations but the idea is straightforward. There are two assumptions made. The first one, mentioned above, is not very restrictive. The second one assumes that the image pattern can be described as a separable function in some suitably chosen coordinate system, which depend on the selected symmetry.

A.1 General assumption

The assumption that the image gradient, ∇f, is parallel to the local orientation $\beta = \beta(x, y)$ can be written mathematically as

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} // \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix} \quad , \quad \text{where } \beta = \frac{1}{2}\arg z \tag{A.1}$$

i.e.

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \pm\sqrt{f_x^2 + f_y^2} \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix} \tag{A.2}$$

We have to divide the phase $\angle z$ by two to get rid of the double angle representation. The price is the direction unambiguity ($\pm$).
Equation A.2 squared gives

$$\left\{ \begin{array}{l} f_x^2 = \left(f_x^2 + f_y^2\right)\cos^2\beta \\ f_y^2 = \left(f_x^2 + f_y^2\right)\sin^2\beta \end{array} \right. \quad \Longleftrightarrow \quad f_x \sin\beta = \pm f_y \cos\beta \tag{A.3}$$

From equation A.2 we see that the '-'-solution is false and we arrive at the final equation:

$$\boxed{\begin{array}{c}\underline{\text{Back-projection equation:}}\\ \\ f_x \sin\beta = f_y \cos\beta\end{array}} \tag{A.4}$$

Every pattern description based on the local orientation is described by a function $\beta(x, y)$. This function can be put into equation A.4, which can be solved to get the corresponding gray-image pattern $f(x, y)$.

The next subsection solves the back-projection equation for the special case of rotational symmetries.

A.2 Rotational symmetries

Theory

A rotational symmetry is defined as

$$z = |z|e^{i2\beta(\varphi)} \tag{A.5}$$

where z is the local orientation in double angle representation. The orientation β only depends on φ and is constant along the r-dimension. We shall now solve the back-projection equation A.4 assuming $\beta = \beta(\varphi)$. In this case it is easier to switch to polar coordinates:

$$\begin{cases} x = r\cos\varphi \\ y = r\sin\varphi \end{cases} \tag{A.6}$$

The partial derivatives, f_x and f_y, can be written in polar coordinates using the chain-rule:

$$\begin{cases} f_x = \frac{\partial f}{\partial x} & = & \frac{\partial f}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial f}{\partial \varphi}\frac{\partial \varphi}{\partial x} & = & f_r \cos\varphi - f_\varphi \frac{\sin\varphi}{r} \\ f_y = \frac{\partial f}{\partial y} & = & \frac{\partial f}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial f}{\partial \varphi}\frac{\partial \varphi}{\partial y} & = & f_r \sin\varphi + f_\varphi \frac{\cos\varphi}{r} \end{cases} \tag{A.7}$$

If this is inserted in equation A.4 we get

$$\left(f_r \cos\varphi - f_\varphi \frac{\sin\varphi}{r}\right)\sin\beta = \left(f_r \sin\varphi + f_\varphi \frac{\cos\varphi}{r}\right)\cos\beta \tag{A.8}$$

After some re-shuffling we get the polar version of equation A.4:

$$\boxed{\begin{array}{c}\underline{\text{Polar back-projection equation:}}\\ \\ f_r r \sin(\beta - \varphi) = f_\varphi \cos(\beta - \varphi)\end{array}} \tag{A.9}$$

This equation is still quite difficult to solve, but if we make the assumption that f is polar separable, i.e. $f(r,\varphi) = R(r)\Phi(\varphi)$, we get

$$\begin{array}{rcl} R'(r)\Phi(\varphi)r\sin(\beta-\varphi) & = & R(r)\Phi'(\varphi)\cos(\beta-\varphi) \\ & \Leftrightarrow & \\ \frac{R'(r)}{R(r)}r & = & \frac{\Phi'(\varphi)}{\Phi(\varphi)}\coth(\beta-\varphi) \end{array} \tag{A.10}$$

Since β only depends on φ we know that the left side only depends on r, and the right side only depends on φ. Therefore both sides have to be constant:

$$\begin{cases} \frac{R'(r)}{R(r)}r = K \\ \frac{\Phi'(\varphi)}{\Phi(\varphi)}\coth(\beta-\varphi) = K \end{cases} \Rightarrow \begin{cases} R(r) = C_1 r^K \\ \Phi(\varphi) = C_2 e^{K\int\tan(\beta-\varphi)d\varphi} \end{cases} \tag{A.11}$$

And we get

$$\boxed{\begin{array}{c} \underline{\text{Rotational symmetries:}} \\ \\ f(r,\varphi) = C(re^{\int\tan(\beta(\varphi)-\varphi)d\varphi})^K \end{array}} \tag{A.12}$$

Provided a $\beta(\varphi)$, this function can be solved numerically to get the final solution. Figure 5.5 on page 47 shows some examples of images that was generated using this formula.

There are some cases for which we can solve equation A.12 analytically. Suppose z is the n:th order rotational symmetry $e^{i(n\varphi+\alpha)}$, i.e. $\beta(\varphi) = \frac{1}{2}n\varphi + \frac{1}{2}\alpha$. Then we get

$$\begin{array}{rcl} \Phi(\varphi) & = & C_2 e^{K\int\tan((\frac{n}{2}-1)\varphi+\frac{\alpha}{2})d\varphi} = /*/ = \\ & = & C_2 e^{-K\frac{1}{\frac{n}{2}-1}\ln|\cos((\frac{n}{2}-1)\varphi+\frac{\alpha}{2})|} = \\ & = & C_2|\cos((\frac{n}{2}-1)\varphi+\frac{\alpha}{2})|^{-\frac{K}{\frac{n}{2}-1}} \end{array} \tag{A.13}$$

$/*/$ There is one exception to the solution above: If $n = 2$ we get

$$\begin{array}{rcl} \Phi(\varphi) & = & C_2 e^{K\int\tan(\frac{\alpha}{2})d\varphi} = \\ & = & C_2 e^{K\tan(\frac{\alpha}{2})\varphi} \end{array} \tag{A.14}$$

If we choose $K = 1 - \frac{n}{2}$ and skip the $|\,.\,|$ in the case $n \neq 2$, and $K = 1$ in the case $n = 2$ we get the final solution:

$$\boxed{\begin{array}{ll} \multicolumn{2}{c}{\underline{n\text{:th order rotational symmetry:}}} \\ & \\ f(r,\varphi) = Cr^{(1-\frac{n}{2})}\cos((\frac{n}{2}-1)\varphi+\frac{\alpha}{2}) & n \neq 2 \\ & \\ f(r,\varphi) = Cre^{\tan(\frac{\alpha}{2})\varphi} & n = 2 \end{array}} \tag{A.15}$$

What do the patterns in equation A.15 look like?

One way to visualize them is to plot trajectories or isobars (inspired by [Bigün, 1988]). To get the trajectories we can for instance plot

$$g(r,\varphi)=\frac{1+\cos(\omega f(r,\varphi))}{2} \tag{A.16}$$

in the case $n \neq 2$. ω determines the frequency of the repetitive pattern. For the case $n=2$ is turns out that

$$\begin{aligned} g(r,\varphi) &= \tfrac{1}{2}(1+\cos(\omega\cos(\tfrac{\alpha}{2})\ln f(r,\varphi))) = \\ &= \tfrac{1}{2}(1+\cos(\omega(\cos(\tfrac{\alpha}{2})\ln r+\sin(\tfrac{\alpha}{2})\varphi))) \end{aligned} \tag{A.17}$$

is a better, well behaved, choice. It is easy to show that if $f(r,\varphi)$ is a solution to the back-projection equation A.4 then every function $g(r,\varphi)=h(f(r,\varphi))$ is also a solution. We can thus generate a larger class of functions than polar separable functions that solves the symmetry equation. Figure 5.3 on page 45 shows some examples of functions from equation A.15 using the trajectory functions A.16 and A.17.

Other choices of g could be a log-norm function:

$$g(r,\varphi)=e^{-\omega_1\ln^2(\omega_2 f(r,\varphi))} \tag{A.18}$$

or a fuzzy threshold function:

$$g(r,\varphi)=\frac{(f(r,\varphi)(1-\omega_1))^{\omega_2}}{(f(r,\varphi)(1-\omega_1))^{\omega_2}+(\omega_1(1-f(r,\varphi)))^{\omega_2}} \tag{A.19}$$

This will give non-repetitive line and edge patterns respectively. Different ω_1, ω_2 give different patterns, see figure 5.4 on page 46 for examples.

B Minimization of polynomial approximation error

This section proves the solution of the minimization problem in equation 4.41, section 4.6. The goal is to minimize

$$\mathrm{E}\left[\|\tilde{\mathbf{r}}-\mathbf{r}\|^2\right]=\mathrm{E}\left[\|\widetilde{\mathbf{T}}_{\mathrm{PD}}(\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{f}-(\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}\mathbf{f}\|^2\right] \tag{B.1}$$

with respect to $\widetilde{\mathbf{T}}_{\mathrm{PD}}$ over a set of signals $\mathbf{f}$. E[.] denotes expectation value over the set. Let

$$\begin{aligned} \mathbf{A}_{\mathrm{P}} &= (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W} \\ \mathbf{A}_{\mathrm{D}} &= (\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W} \end{aligned} \tag{B.2}$$

We can now rewrite the minimization function as

$$\begin{aligned}
\mathrm{E}\left[\|\tilde{\mathbf{r}}-\mathbf{r}\|^2\right] &= \mathrm{E}\left[\|\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}\mathbf{f}-\mathbf{A}_{\mathrm{P}}\mathbf{f}\|^2\right] \qquad \text{(B.3)}\\
&= \mathrm{E}\left[\mathbf{f}^T(\mathbf{A}_{\mathrm{D}}^T\widetilde{\mathbf{T}}_{\mathrm{PD}}^T-\mathbf{A}_{\mathrm{P}}^T)(\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}-\mathbf{A}_{\mathrm{P}})\mathbf{f}\right]\\
&= \mathrm{E}\left[\mathbf{f}^T\mathbf{A}_{\mathrm{D}}^T\widetilde{\mathbf{T}}_{\mathrm{PD}}^T\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}\mathbf{f}+\mathbf{f}^T\mathbf{A}_{\mathrm{P}}^T\mathbf{A}_{\mathrm{P}}\mathbf{f}-2\mathbf{f}^T\mathbf{A}_{\mathrm{P}}^T\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}\mathbf{f}\right]
\end{aligned}$$

The solution is found by solving the equation system

$$\begin{aligned}
\frac{\partial\mathrm{E}\left[\|\tilde{\mathbf{r}}-\mathbf{r}\|^2\right]}{\partial\widetilde{\mathbf{T}}_{\mathrm{PD}}} &= \mathrm{E}\left[2\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}\mathbf{f}\mathbf{f}^T\mathbf{A}_{\mathrm{D}}^T-2\mathbf{A}_{\mathrm{P}}\mathbf{f}\mathbf{f}^T\mathbf{A}_{\mathrm{D}}^T\right]\\
&= 2\widetilde{\mathbf{T}}_{\mathrm{PD}}\mathbf{A}_{\mathrm{D}}\mathbf{C}_{\mathrm{ff}}\mathbf{A}_{\mathrm{D}}^T-2\mathbf{A}_{\mathrm{P}}\mathbf{C}_{\mathrm{ff}}\mathbf{A}_{\mathrm{D}}^T=\mathbf{0} \qquad \text{(B.4)}
\end{aligned}$$

where $\mathbf{C}_{\mathrm{ff}}=\mathrm{E}[\mathbf{f}\mathbf{f}^T]$. The solution becomes

$$\begin{aligned}
\widetilde{\mathbf{T}}_{\mathrm{PD}} &= \mathbf{A}_{\mathrm{P}}\mathbf{C}_{\mathrm{ff}}\mathbf{A}_{\mathrm{D}}^T(\mathbf{A}_{\mathrm{D}}\mathbf{C}_{\mathrm{ff}}\mathbf{A}_{\mathrm{D}}^T)^{-1}\\
&= (\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}\mathbf{C}_{\mathrm{ff}}\mathbf{W}\widetilde{\mathbf{D}}(\widetilde{\mathbf{D}}^T\mathbf{W}\mathbf{C}_{\mathrm{ff}}\mathbf{W}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}} \qquad \text{(B.5)}
\end{aligned}$$

If we assume $\mathbf{C}_{\mathrm{ff}}\propto\mathbf{I}$ we get

$$\widetilde{\mathbf{T}}_{\mathrm{PD}}=(\mathbf{P}^T\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2\widetilde{\mathbf{D}}(\widetilde{\mathbf{D}}^T\mathbf{W}^2\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^T\mathbf{W}\widetilde{\mathbf{D}} \qquad \text{(B.6)}$$

(It can be mentioned that the solution would have been the same if we had used a weighted norm, because each dimension is optimized separately.)

C Definition of SNR, PSNR

Two measures of noise are used in this thesis, SNR and PSNR. The definitions are stated below.

Signal-to-noise ratio, SNR:

$$\mathrm{SNR}=10\log_{10}\frac{\sigma_x^2}{\sigma_e^2} \qquad \text{(C.1)}$$

where σ_x and σ_e are the standard deviations of the signal (without noise) and the noise respectively.

Peak-signal-to-noise-ratio, PSNR:

$$\mathrm{PSNR}=10\log_{10}\frac{|x_{p-p}|^2}{\sigma_e^2} \qquad \text{(C.2)}$$

where x_{p-p} is the peak-to-peak value of the signal (without noise).

Note that the SNR measure is in a sense more dependent on the signal than the PSNR measure. For example, two images of corners with different corner angle have different σ_x but the same x_{p-p}.

Bibliography

[Andersson et al., 1998] Andersson, M., Wiklund, J., and Knutsson, H. (1998). Sequential Filter Trees for Efficient 2D 3D and 4D Orientation Estimation. Report LiTH-ISY-R-2070, ISY, SE-581 83 Linköping, Sweden. URL: http://www.isy.liu.se/cvl/ScOut/TechRep/TechRep.html.

[Andersson et al., 1999] Andersson, M., Wiklund, J., and Knutsson, H. (1999). Filter Networks. In *Proceedings of Signal and Image Processing (SIP'99)*, Nassau, Bahamas. IASTED. Also as Technical Report LiTH-ISY-R-2245.

[Atkinson et al., 1990] Atkinson, R. L., Atkinson, R. C., Smith, E. E., Bem, D. J., and Hildagd, E. R. (1990). *Introduction to Psychology, 10th edition.* Harcourt Brace Jovanovich, Inc.

[Attneave, 1954] Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61.

[Bårman, 1991] Bårman, H. (1991). *Hierarchical Curvature Estimation in Computer Vision.* PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden. Dissertation No 253, ISBN 91-7870-797-8.

[Bårman and Granlund, 1988] Bårman, H. and Granlund, G. H. (1988). Corner Detection Using Local Symmetry. In *Proceedings from SSAB Symposium on Picture Processing*, Lund University, Sweden. SSAB. Report LiTH–ISY–I–0935, Computer Vision Laboratory, Linköping University, Sweden, 1988.

[Bear et al., 1996] Bear, M. F., Connors, B. W., and Paradiso, M. A. (1996). *Neuroscience: Exploring the Brain.* Williams & Wilkins. ISBN 0-683-00488-3.

[Beaudet, 1978] Beaudet, P. R. (1978). Rotational invariant image operators. In *Internation Joint Conference on Pattern Recognition*, pages 579–583.

[Biederman, 1987] Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147.

[Biederman and Cooper, 1991] Biederman, I. and Cooper, E. E. (1991). Priming contour-deleted images: Evidence for intermediate representations in visual object recognition. *Cognitive Psychology*, 23:393–419.

[Bigün, 1987] Bigün, J. (1987). Optimal Orientation Detection of Circular Symmetry. Report LiTH–ISY–I–0871, Computer Vision Laboratory, Linköping University, Sweden.

[Bigün, 1988] Bigün, J. (1988). *Local Symmetry Features in Image Processing.* PhD thesis, Linköping University, Sweden. Dissertation No 179, ISBN 91-7870-334-4.

[Bigün, 1990] Bigün, J. (1990). A structure feature for some image processing applications based on spiral functions. *Computer Vision, Graphics, and Image Processing*, 51:166–194.

[Bigün, 1997] Bigün, J. (1997). Pattern recognition in images by symmetries and coordinate transformations. *Computer Vision and Image Understanding*, 68(3):290–307.

[Blakemore and Over, 1974] Blakemore, C. and Over, R. (1974). Curvature detectors in human vision? *Perception*, 3:3–7.

[Borga, 1998] Borga, M. (1998). *Learning Multidimensional Signal Processing.* PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden. Dissertation No 531, ISBN 91-7219-202-X.

[Borga and Knutsson, 2000] Borga, M. and Knutsson, H. (2000). Finding Efficient Nonlinear Visual Operators using Canonical Correlation Analysis. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 13–16, Halmstad. SSAB.

[Burt, 1988] Burt, P. J. (1988). Moment images, polynomial fit filters and the problem of surface interpolation. In *Proc. of Computer Vision and Pattern Recognition, Ann Arbor.* Computer Society Press.

[Charnley and Blisset, 1989] Charnley, D. and Blisset, R. J. (1989). Surface reconstruction from outdoor image sequances. *Image and Vision Computing*, 7(1):10–16.

[Cooper et al., 1990] Cooper, J., Venkatesh, S., and Kitchen, L. (1990). Early jump-out corner detectors. Technical Report 90/14, Departement of Computer Science, University of Western Australia.

[Cornsweet, 1970] Cornsweet, T. N. (1970). *Visual Perception.* Academic press.

[Deriche and Giraudon, 1990] Deriche, R. and Giraudon, G. (1990). Accurate corner detection: An analytical study. In *3rd Internation Conference on Computer Vision*, pages 66–70.

[Doherty et al., 2000] Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E., and Wiklund, J. (2000). The WITAS Unmanned Aerial Vehicle Project. In Horn, W., editor, *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, pages 747–755, Amsterdam.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the Hough transform to detect lines and cures in pictures. *Communications of the Association Computing Machinery*, 15.

[Farnebäck, 1999a] Farnebäck, G. (1999a). Spatial Domain Methods for Orientation and Velocity Estimation. Thesis LiU-Tek-Lic-1999:13, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden. Lic. Thesis No. 755, ISBN 91-7219-441-3.

[Farnebäck, 1999b] Farnebäck, G. (1999b). A Unified Framework for Bases, Frames, Subspace Bases, and Subspace Frames. In *Proceedings of the 11th Scandinavian Conference on Image Analysis*, pages 341–349, Kangerlussuaq, Greenland. SCIA. Also as Technical Report LiTH-ISY-R-2246.

[Farnebäck, 2000a] Farnebäck, G. (2000a). Fast and Accurate Motion Estimation using Orientation Tensors and Parametric Motion Models. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 1, pages 135–139, Barcelona, Spain. IAPR.

[Farnebäck, 2000b] Farnebäck, G. (2000b). Orientation Estimation Based on Weighted Projection onto Quadratic Polynomials. In Girod, B., Greiner, G., Niemann, H., and Seidel, H.-P., editors, *Vision, Modeling, and Visualization 2000: proceedings*, pages 89–96, Saarbrücken.

[Forssén and Johansson, 2000] Forssén, P.-E. and Johansson, B. (2000). Fractal coding by means of local feature histograms. Report LiTH-ISY-R-2295, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden.

[Gallant et al., 1993] Gallant, J. L., Braun, J., and Essen, D. C. V. (1993). Selectivity for polar, hyperbolic, and cartesian gratings in macaque visual cortex. *Science*, 259:100–103.

[Granlund, 1978] Granlund, G. H. (1978). In Search of a General Picture Processing Operator. *Computer Graphics and Image Processing*, 8(2):155–178.

[Granlund, 1999] Granlund, G. H. (1999). The complexity of vision. *Signal Processing*, 74(1):101–126. Invited paper.

[Granlund, 2000] Granlund, G. H. (2000). An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany.

[Granlund and Knutsson, 1995] Granlund, G. H. and Knutsson, H. (1995). *Signal Processing for Computer Vision*. Kluwer Academic Publishers. ISBN 0-7923-9530-1.

[Haralick, 1984] Haralick, R. M. (1984). Digital step edges from zero crossing of second directional derivatives. PAMI-6(1):58–68.

[Haralick and Shapiro, 1993] Haralick, R. M. and Shapiro, L. G. (1993). *Computer and robot vision*, volume 2. Addison-Wesley.

[Haralick and Watson, 1981] Haralick, R. M. and Watson, L. (1981). A facet model for image data. *Computer Graphics and Image Processing*, 15(2):113–129.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151.

[Hoffman, 1966] Hoffman, W. C. (1966). The Lie algebra of visual perception. *Journal Math. Psychol.*, 3:65–98.

[Horridge, 2000] Horridge, A. (2000). Seven experiments on pattern vision of the honeybee, with a model. *Vision Research*, 40:2589–2603.

[Humphreys et al., 1994] Humphreys, G. W., Romani, C., Olson, A., Riddoch, M. J., and Duncan, J. (1994). Non-spatial extinction following lesions of the parietal lobe in humans. *Nature*, 372:357–359.

[Johansson, 2000a] Johansson, B. (2000a). Backprojection of Some Image Symmetries Based on a Local Orientation Description. Report LiTH-ISY-R-2311, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden.

[Johansson, 2000b] Johansson, B. (2000b). A Survey on: Contents Based Search in Image Databases. Report LiTH-ISY-R-2215, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden.

[Johansson et al., 2001] Johansson, B., Borga, M., and Knutsson, H. (2001). Learning corner orientation using canonical correlation. In *Proceedings of the SSAB Symposium on Image Analysis*, Norrköping. SSAB. to appear.

[Johansson and Granlund, 2000] Johansson, B. and Granlund, G. (2000). Fast Selective Detection of Rotational Symmetries using Normalized Inhibition. In *Proceedings of the 6th European Conference on Computer Vision*, volume I, pages 871–887, Dublin, Ireland.

[Johansson et al., 2000] Johansson, B., Knutsson, H., and Granlund, G. (2000). Detecting Rotational Symmetries using Normalized Convolution. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 3, pages 500–504, Barcelona, Spain. IAPR.

[Kitchen and Rosenfeld, 1982] Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102.

[Knutsson and Andersson, 1995] Knutsson, H. and Andersson, M. (1995). Optimization of Sequential Filters. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 87–90, Linköping, Sweden. SSAB. LiTH-ISY-R-1797. URL: http://www.isy.liu.se/cvl/ScOut/ TechRep/TechRep.html.

[Knutsson et al., 1999] Knutsson, H., Andersson, M., and Wiklund, J. (1999). Advanced Filter Design. In *Proceedings of the11th Scandinavian Conference on Image Analysis*, Greenland. SCIA. Also as report LiTH-ISY-R-2142.

[Knutsson and Borga, 1999] Knutsson, H. and Borga, M. (1999). Learning Visual Operators from Examples: A New Paradigm in Image Processing. In *Proceedings of the 10th International Conference on Image Analysis and Processing (ICIAP'99)*, Venice, Italy. IAPR. Invited Paper.

[Knutsson and Granlund, 1988] Knutsson, H. and Granlund, G. H. (1988). Apparatus for Determining the Degree of Variation of a Feature in a Region of an Image that is Divided into Discrete Picture Elements. US-Patent 4.747.151, 1988. (Swedish patent 1986).

[Knutsson et al., 1988a] Knutsson, H., Granlund, G. H., and Bigun, J. (1988a). Apparatus for Detecting Sudden Changes of a Feature in a Region of an Image that is Divided into Discrete Picture Elements. US-Patent 4.747.150, 1988. (Swedish patent 1986).

[Knutsson et al., 1988b] Knutsson, H., Hedlund, M., and Granlund, G. H. (1988b). Apparatus for Determining the Degree of Consistency of a Feature in a Region of an Image that is Divided into Discrete Picture Elements. US-Patent 4.747.152, 1988). (Swedish patent 1986).

[Knutsson and Westin, 1993] Knutsson, H. and Westin, C.-F. (1993). Normalized and differential convolution. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 515–523. IEEE.

[Kobatake and Tanaka, 1994] Kobatake, E. and Tanaka, K. (1994). Neuronal selectivity to complex object-features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology*, 71:856–867.

[Marsicoi, 1997] Marsicoi (1997). Indexing ictorial documents by their content: a survey of current techniques. *Image and Vision Computing*, 15:119–141.

[Mehrotra et al., 1990] Mehrotra, R., Nichani, S., and Ranganathan, N. (1990). Corner detection. *Pattern Recognition*, 23(11):1223–1233.

[Mokhtarian et al., 1996] Mokhtarian, F., Abbasi, S., and Kittler, J. (1996). Robust and efficient shape indexing through curvature scale space. In *Proceedings of the 1996 British Machine and Vision Conference BMVC'96*, Edinburgh, Scotland.

[Moravec, 1977] Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of Int. Joint. Conf. on Artificial Intelligence in Cambridge,MA*, page page 584.

[Nagel, 1983] Nagel, H.-H. (1983). Displacement vectors derived from seceond-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21:85–117.

[Nayar et al., 1996] Nayar, S. K., Nene, S. A., and Murase, H. (1996). Real-Time 100 Object Recognition System. In *Proc. of ARPA Image Understanding Workshop*.

[Noble, 1988] Noble, J. A. (1988). Finding corners. *Image and Vision Computing*, 6(2):121–128.

[Oster, 1970] Oster, G. (1970). Phosphenes. *Scientific American*, 222(2):82–87.

[Pontil and Verri, 1998] Pontil, M. and Verri, A. (1998). Support vector machines for 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646.

[Reisfeld et al., 1995] Reisfeld, D., Wolfson, H., and Yeshurun, Y. (1995). Context free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14:119–130.

[Rohr, 1992] Rohr, K. (1992). Modelling and identification of characteristic intensity variations. *Image and Vision Computing*, 10(2):66–76.

[Roll, 1999] Roll, J. (1999). A System for Visual-Based Automated Storage Robots. Master's thesis, Linköping University. LiTH-ISY-EX-2053. URL: http://www.isy.liu.se/cvl/ScOut/ Masters/Masters.html.

[Rosin, 1992] Rosin, P. L. (1992). Representing curves at their natural scales. *Pattern Recognition*, 25(11):1315–1325.

[Rui et al., 1999] Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62.

[Smeulders et al., 2001] Smeulders, A. W. M., Woring, M., Santini, S., Gupta, A., and Jain, R. (2001). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.

[Smith and Brady, 1995] Smith, S. M. and Brady, J. M. (1995). Asset-2: Real-time motion segmentation and object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820.

[Smith and Brady, 1997] Smith, S. M. and Brady, J. M. (1997). Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

[Tanaka, 1996] Tanaka, K. (1996). Representation of visual features of objects in the inferotemporal cortex. *Neural Networks*, 9(8):1459–1475.

[Trajkovic and Hedley, 1998] Trajkovic, M. and Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16:75–87.

[Westelius, 1995] Westelius, C.-J. (1995). *Focus of Attention and Gaze Control for Robot Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden. Dissertation No 379, ISBN 91-7871-530-X.

[Westin, 1994] Westin, C.-F. (1994). *A Tensor Framework for Multidimensional Signal Processing*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden. Dissertation No 348, ISBN 91-7871-421-4.

[WITAS web page,] WITAS web page. URL: http://www.ida.liu.se/ext/witas/.

[Yang et al., 2000] Yang, M.-H., Roth, D., and Ahuja, N. (2000). Learning to Recognize 3D Objects with SNoW. In *Proceedings of the 6th European Conference on Computer Vision*, volume I, pages 439–454, Dublin, Ireland.