Steerable Filters and Local Analysis of Image Structure

by

William Tafel Freeman

B.S., Physics with Honors and Distinction, Stanford University, 1979
M.S., Electrical Engineering, Stanford University, 1979
M.S., Applied Physics, Cornell University, 1981

Submitted to the Media Arts and Sciences Section,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

June 1992

Author____

Media Arts and Sciences Section
May 1, 1992

Certified by____

Edward H. Adelson
Associate Professor,
MIT Media Laboratory and Dept. of Brain and Cognitive Sciences
Thesis Supervisor

Accepted by____

Stephen Benton
Chairman, Departmental Committee on Graduate Students

Steerable Filters and Local Analysis of Image Structure

by

William Tafel Freeman

Submitted to the Media Arts and Sciences Section,
School of Architecture and Planning
on May 1, 1992, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Two paradigms for visual analysis are *top-down*, starting from high-level models or information about the image, and *bottom-up*, where little is assumed about the image or objects in it. We explore a local, bottom-up approach to image analysis. We develop operators to identify and classify image junctions, which contain important visual cues for identifying occlusion, transparency, and surface bends.

Like the human visual system, we begin with the application of linear filters which are oriented in all possible directions. We develop an efficient way to create an oriented filter of arbitrary orientation by describing it as a linear combination of *basis filters*. This approach to oriented filtering, which we call *steerable filters*, offers advantages for analysis as well as computation. We design a variety of steerable filters, including steerable quadrature pairs, which measure local energy. We show applications of these filters in orientation and texture analysis, and image representation and enhancement.

We develop methods based on steerable filters to study structures such as contours and junctions. We describe how to post-filter the energy measures in order to more efficiently analyze structures with multiple orientations. We introduce a new detector for contours, based on energy local maxima. We analyze contour phases at energy local maxima, and compare the results with the prediction of a simple model.

Using these tools, we analyze junctions. Based on local oriented filters, we develop simple mechanisms which respond selectively to "T", "L", and "X" junctions. The T and X junctions may indicate occlusion and transparency, respectively. These mechanism show that detectors for important, low-level visual cues can be built out of oriented filters and energy measures, which resemble responses found in the visual cortex.

We present a second approach to junction detection based on salient contours. We combine our contour detector with the structural saliency algorithm of Shashua and Ullman, which finds visually salient contours. To improve its descriptive power, we include a competitive mechanism in the algorithm. From the local configuration of saliencies, we form simple detectors which respond to cues for occlusion, transparency and surface bending. Using the saliency values and curve linking information, we can propagate this information along image contours.

For both algorithms, we show successful results on simple synthetic and natural images. We show results for more complicated scenes and discuss the methods do not work, and why. Each algorithm uses only local calculations applied in parallel throughout the image, and assumes little prior information about the objects it expects to see.

Thesis Supervisor: Edward H. Adelson
Associate Professor,
MIT Media Laboratory and Dept. of Brain and Cognitive Sciences

Acknowledgments

My first thanks, and many thanks, go to my advisor, Ted Adelson. It was because of him that my graduate studies were enjoyable and so intellectually rewarding. He showed me how to do research, write papers and give talks. I could always count on him to have good intuition about hard problems. The apprenticeship was a privilege and a pleasure.

Sandy Pentland and Ted, now with Roz Picard and Aaron Bobick, previously with Bernd Girod, have made the Vision and Modeling group be a great environment for vision research and have taught us all a great deal. An important character in the group was my office and discussion-mate, Eero Simoncelli. Eero taught me what I know about wavelets and what I don't know about phase. I learned from Dave Heeger, too. All the group members contributed through hardware, software, math, humor, and secretarial skills; thanks to all: Stan, Trevor, Irfan, Stanzi, Monica, George, Thad, Pawan, Laureen, Bea, Shawn, Matthew, Fang, Mr. Tsunashima, Mr. Isonishi, Mr. Ohta, Mr. Hoshino, John, Martin F., Martin B., Bradley, Ali, Alex, Stephen, Mike, Peg.

I appreciate the effort and time put in by my thesis committee, Ted Adelson, Sandy Pentland, and Rick Szeliski. They read my thesis carefully, asked hard questions at my defense, and suggested helpful improvements.

The Osman family helped out very much, from with taking care of Roz to taking care of me.

Many thanks to my family: my mother, Irv, Flo and Rick, for their support during this time. I thank my late father, too, for all his love. I wish he could see this.

Thank you so much, Franny, for all of your work, understanding, and love during these years. Our second child (to be) helped out by providing extra motivation to meet thesis deadlines. Thanks to our first child, Roz, for showing me what really matters.

This research was performed at the MIT Media Laboratory and was supported in part by DARPA/RADC contract F30602-89-C-0022, a contract with Goldstar Co., Ltd, and a contract with David Sarnoff Research Laboratories (subcontract to the National Information Display Laboratory). The opinions expressed are those of the author and do not necessarily represent those of the sponsors.

Contents

Chapter 1

Introduction

1.1 The Problem

Humans see effortlessly. Reflectance, shading, and illumination effects all change the observed light intensities, yet we can sort out which effects are responsible for which changes in the images we observe.

Computers can not yet do as well. The simple images of Fig. 1-1 (a) – (c) would stump virtually all image analysis programs. The center portions of each figure have identical intensities (see (d) – (f)), yet each one gives a very different visual percept to a human. In Fig. 1-1 (a), the center bar appears to be occluding a rectangle behind it. Figure 1-1 (b) looks like two overlaid transparent rectangles. Figure 1-1 (c) looks like a folded sheet.

Most image interpretation programs assign only one meaning to all intensity changes, and could never come up with the correct interpretation for all three images. *Shape-from-shading* programs exist which treat all intensity variations as evidence for shading, which would interpret Fig. 1-1 (c) correctly but all the other images incorrectly. An algorithm which could parse transparent overlays would only interpret Fig. 1-1 (b) correctly. An unsolved problem is how to decide what process caused the observed image intensities–are they due to shading, reflectance, or lighting changes, or transparency? Essential to solving this problem is to identify and categorize the physical origin of the different junctions and contours in Fig. 1-1. Identifying those physical origins is the goal of this work.

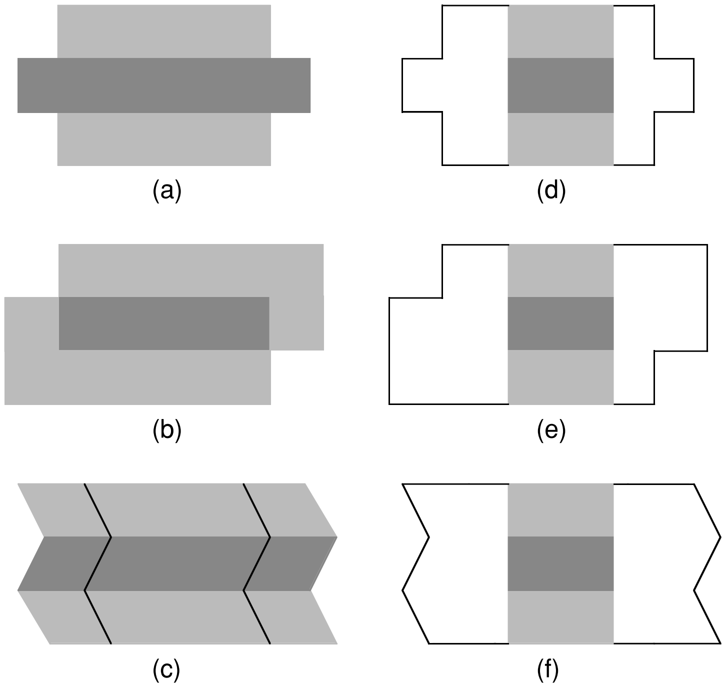


Figure 1-1: Illustration showing the insufficiency of image interpretation based on local image intensities. (a) – (c) have the same image intensities in their centers, as shown in (d) – (f). However, we assign very different interpretations to these same intensities– (a) occlusion, (b) transparency, and (c) a surface bend.

1.2 Our Approach

We want to work with digitized images using local, biologically plausible operations. We will not attempt to model the visual system. However, in restricting ourselves to some of the same constraints and representations as we believe the brain uses, we hope to gain insight into problems the brain may have to solve or approaches it may use.

Some computer vision systems are model based, and can exploit *top-down* reasoning to interpret visual information. In this work, we take the opposite approach and explore what can be done with purely *bottom-up* processing. This allows us to make few assumptions about what we expect to see. We expect that a better understanding of the bottom-up part will lead to better general vision systems which use both bottom-up and top-down analysis.

We will analyze images based on the local cues which junctions provide. Junctions can indicate, among other things, occlusion, transparency or surface bending. We begin with the same initial processing step that it is thought that the brain uses—linear filtering by a bank of oriented filters (see, e.g., [79]). In Chapter 2 we study how to apply oriented filters over a continuum of orientations. We develop an approach to oriented filtering which we call *steerable filters*. This method is efficient and analytically useful. These results have applications in many areas of image processing and computer vision.

In Chapter 3, as a first step in analyzing contours and junctions, we use steerable filters to analyze orientation in regions with one or more orientations. We identify an artifact particular to regions of multiple orientations, and propose a post-filtering step to remove the effect. The post-filtering increases the efficiency of the oriented filters. These mathematical results apply to the analysis of junctions in images, as well as the analysis of occlusion and transparency in moving sequences.

In Chapter 4 we build a contour detector from oriented filters which responds properly to lines, edges, and image contours of phase intermediate between those two. We also study statistical properties of the local phase along image contours.

We then build two different types of junction detectors. The first, described in Chapter 5, follows the orientation and contour analysis by steerable filters with additional local filtering steps. It successfully detects and categorizes junctions in simple images, showing that this important function can be done with simple filter-like operations.

The second junction detector, developed in Chapter 6, is based on salient contours and has some advantages over the first approach. We modify an existing *salient* curve finder to improve its performance near curves and junctions. The output gives a local indicator for nearby curves. By analyzing the configuration of the saliency outputs we form a junction detector with improved performance in the presence of incomplete or noisy image data. The curve-finder provides a simple way to propagate the identification made at junctions along the appropriate contours.

The resulting methods can interpret the causes of the junctions and contours in images such as Fig. 1-1. The algorithms and tools developed in this bottom-up approach are general and apply to other systems for image processing and analysis.

Chapter 2

Tools for Image Analysis – Steerable Filters

2.1 Introduction

Oriented filters are used in many vision and image processing tasks, such as texture analysis, edge detection, image data compression, motion analysis, and image enhancement [70, 27, 20, 43, 89, 33, 45, 4, 38, 57, 60]. In many of these tasks, it is necessary to apply filters of arbitrary orientation under adaptive control, and to examine the filter output as a function of both orientation and phase. We will discuss techniques that allow the synthesis of a filter at arbitrary orientation and phase, and develop methods to analyze the filter outputs. We will also describe efficient architectures for such processing, develop flexible design methods for the filters in two and three dimensions, and apply the filters to several image analysis tasks. Other reports of this work appear in [31, 32, 33].

One approach to finding the response of a filter at many orientations is to apply many versions of the same filter, each different from the others by some small rotation in angle. A more efficient approach is to apply a few filters corresponding to a few angles and interpolate between the responses. One then needs to know how many filters are required and how to properly interpolate between the responses. With the correct filter set and the correct interpolation rule, it is possible to determine the response of a filter of arbitrary orientation without explicitly applying that filter.

We use the term *steerable filter* to describe a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of *basis filters.* We will show that two-dimensional functions are steerable (see [87, 33] for higher dimensional cases), and will show how many basis filters are needed to steer a given filter.

2.2 An Example

As an introductory example, consider the 2-dimensional, circularly symmetric Gaussian function, G, written in Cartesian coordinates, x and y:

$$G(x,y) = e^{-(x^2+y^2)}, \tag{2.1}$$

where scaling and normalization constants have been set to 1 for convenience. The directional derivative operator is steerable as is well-known [25, 31, 43, 54, 60, 61, 62, 63, 73, 85]. Let us write the nth derivative of a Gaussian in the x direction as G_n. Let $(\ldots)^\theta$ represent the rotation operator, such that, for any function $f(x,y)$, $f^\theta(x,y)$ is $f(x,y)$ rotated through an angle θ about the origin. The first x derivative of a Gaussian, $G_1^{0^\circ}$, is

$$G_1^{0^\circ} = \frac{\partial}{\partial x} e^{-(x^2+y^2)} = -2xe^{-(x^2+y^2)}. \tag{2.2}$$

That same function, rotated 90 degrees, is:

$$G_1^{90^\circ} = \frac{\partial}{\partial y} e^{-(x^2+y^2)} = -2ye^{-(x^2+y^2)}. \tag{2.3}$$

These functions are shown in Fig. 2-1 (a) and (b). It is straightforward to show that a G_1 filter at an arbitrary orientation θ can be synthesized by taking a linear combination of $G_1^{0^\circ}$ and $G_1^{90^\circ}$:

$$G_1^\theta = \cos(\theta) G_1^{0^\circ} + \sin(\theta) G_1^{90^\circ}. \tag{2.4}$$

Since $G_1^{0^\circ}$ and $G_1^{90^\circ}$ span the set of G_1^θ filters we call them *basis filters* for G_1^θ. The $\cos(\theta)$ and $\sin(\theta)$ terms are the corresponding *interpolation functions* for those basis filters.

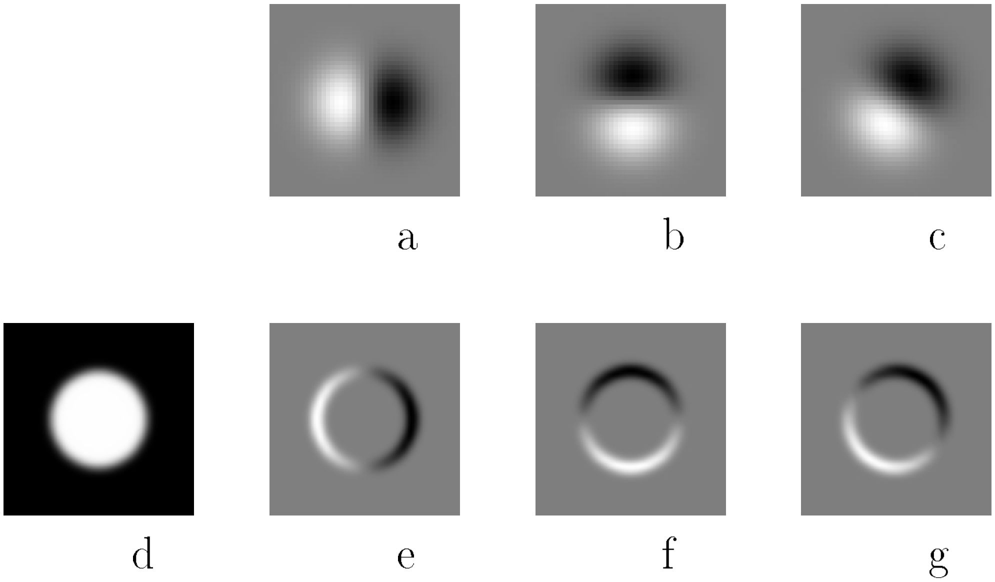


Figure 2-1: Example of steerable filters. (a) $G_1^{0^\circ}$, first derivative with respect to x (horizontal) of a Gaussian. (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^{θ}, an arbitrary rotation of the first derivative of a Gaussian. (c) $G_1^{30^\circ}$, formed by $\frac{1}{2}G_1^{0^\circ} + \frac{\sqrt{3}}{2}G_1^{90^\circ}$. The same linear combinations used to synthesize G_1^{θ} from the basis filters will also synthesize the response of an image to G_1^{θ} from the responses of the image to the basis filters: (d) Image of circular disk. (e) $G_1^{0^\circ}$ (at a smaller scale than pictured above) convolved with the disk, (d). (f) $G_1^{90^\circ}$ convolved with (d). (g) $G_1^{30^\circ}$ convolved with (d), obtained from $\frac{1}{2}$ [image e] $+\frac{\sqrt{3}}{2}$ [image f].

Because convolution is a linear operation, we can synthesize an image filtered at an arbitrary orientation by taking linear combinations of the images filtered with $G_1^{0^\circ}$ and $G_1^{90^\circ}$. Letting $*$ represent convolution, if

$$R_1^{0^\circ} = G_1^0 * I \tag{2.5}$$

$$R_1^{90^\circ} = G_1^{90} * I \tag{2.6}$$

then

$$R_1^\theta = \cos(\theta) R_1^{0^\circ} + \sin(\theta) R_1^{90^\circ}. \tag{2.7}$$

The derivative of Gaussian filters offer a simple illustration of steerability. In the next section, we generalize these results to encompass a wide variety of filters. (See also [87, 103] for recent extensions of this approach.)

2.3 Steering Theorems

We want to find the conditions under which any function, $f(x,y)$, steers, i.e., when it can be written as a linear sum of rotated versions of itself.

The steering constraint is

$$f^\theta(x,y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x,y). \tag{2.8}$$

We want to know what functions $f(x,y)$ can satisfy Eq. (2.8), how many terms, M, are required in the sum, and what the interpolation functions, $k_j(\theta)$, are.

We will work in polar coordinates $r = \sqrt{x^2+y^2}$ and $\phi = \arg(x,y)$. Let f be any function which can be expanded in a Fourier series in polar angle, ϕ:

$$f(r,\phi) = \sum_{n=-N}^{N} a_n(r) e^{in\phi}. \tag{2.9}$$

Through using the Fourier expansion for f, Eq. (2.9), in the steering constraint, Eq. (2.8), one can show [33] the following:

Theorem 1 *The steering condition, Eq. (2.8), holds for functions expandable in the* *form of Eq. (2.9) if and only if the interpolation functions $k_j(\theta)$ are solutions of:*

$$\begin{pmatrix} 1 \\ e^{i\theta} \\ \dots \\ e^{iN\theta} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ e^{i\theta_1} & e^{i\theta_2} & \dots & e^{i\theta_M} \\ \vdots & \vdots & \vdots & \vdots \\ e^{iN\theta_1} & e^{iN\theta_2} & \dots & e^{iN\theta_M} \end{pmatrix} \begin{pmatrix} k_1(\theta) \\ k_2(\theta) \\ \vdots \\ k_M(\theta) \end{pmatrix}. \tag{2.10}$$

If, for any n, $a_n(r) = 0$, then the corresponding (nth) row of the left hand side and of the matrix of the right hand side of Eq. (2.10) should be removed.

We are interested in the minimum number of basis functions which are required to steer a particular function, $f(r, \phi)$. Let T be the number of positive or negative frequencies $-N \leq n \leq N$ for which $f(r, \phi)$ has non-zero coefficients $a_n(r)$ in a Fourier decomposition in polar angle. For example, $\cos(\phi) = \frac{e^{i\phi}+e^{-i\phi}}{2}$ has $T = 2$ and $\cos(\phi)+1 = \frac{e^{i\phi}+e^{-i\phi}}{2}+e^0$ has $T = 3$. By making projections onto complex exponentials and analyzing the ranks of matrices, one can derive the minimum number of basis filters of any form which will steer $f(r, \phi)$ [33], i.e., for which the following equation holds:

$$f^\theta(r, \phi) = \sum_{j=1}^{M} k_j(\theta) g_j(r, \phi), \tag{2.11}$$

where the $g_j(r, \phi)$ can be any set of functions. Theorem 2 gives the result:

Theorem 2 *Let T be the number of non-zero coefficients $a_n(r)$ for functions $f(r, \phi)$ expandable in the form of Eq. (2.9). Then the minimum number of basis functions which are sufficient to steer $f(r, \phi)$ by Eq. (2.11) is T, i.e., M in Eq. (2.11) must be $\geq T$.*

Using rotated versions of the function itself as the basis functions, as in Eq. (2.8), the T basis function orientations θ_j must be chosen so that the columns of the matrix in Eq. (2.10) are linearly independent. In practice, for reasons of symmetry and robustness against noise, we choose basis functions spaced equally in angle between 0 and π. Note that the interpolation functions $k_j(\theta)$ do not depend on the values of the non-zero coefficients $a_n(r)$ in the Fourier angular decomposition of the filter $f(r, \phi)$.

A 1-D bandlimited function can be represented by a finite number of samples corresponding to the number of Fourier terms, which is the number of degrees of freedom. Theorems 1 and 2 show that angularly bandlimited functions behave the same way.

We illustrate the use of Theorem 1 by re-deriving the steering equation for G_1. In polar coordinates, the first derivative of a Gaussian is

$$G_1^{0^\circ}(r,\phi) = -2re^{-r^2}\cos(\phi) = -re^{-r^2}(e^{i\phi}+e^{-i\phi}). \tag{2.12}$$

Since $G_1^{0^\circ}(r,\phi)$ has two non-zero coefficients in a Fourier decomposition in polar angle ϕ, by Theorem 1, two basis functions suffice to synthesize G_1^θ. The interpolation functions are found from Eq. (2.10), with all entries but the second row removed:

$$\left(e^{i\theta} \right) = \left(e^{i\theta_1} \quad e^{i\theta_2} \right) \left(\begin{array}{c} k_1(\theta) \\ k_2(\theta) \end{array} \right). \tag{2.13}$$

If we pick one basis function to be oriented at $\theta_1 = 0^\circ$ and the other at $\theta_2 = 90^\circ$, then Eq. (2.13) gives $k_1(\theta) = \cos(\theta)$ and $k_2(\theta) = \sin(\theta)$. Thus, Theorem 1 tells us that $G_1^\theta = \sum_{j=1}^{2} k_j(\theta) G_1^{\theta_j} = \cos(\theta)G_1^{0^\circ} + \sin(\theta)G_1^{90^\circ}$, in agreement with Eq. (2.4).

Figure 2-2 shows 1-D cross-sections of some steerable basis filters, plotted as a function of angle ϕ at a constant radius. An arbitrary translation of any one curve can be written as a linear combination of the basis curves shown on the graph (rotation of the filter corresponds to translation on these graphs). Figure 2-2 (a) shows the sinusoidal variation of 1-D slices of $G_1^{0^\circ}$ and $G_1^{90^\circ}$, plotted at a constant radius. In this case, the steering property is a re-statement of the fact that a linear combination of two sinusoids can synthesize a sinusoid of arbitrary phase. Figure 2-2(b) and (c) are 1-D cross-sections of steerable basis sets for functions with the azimuthal distribution $0.25\cos(3\phi) + 0.75\cos(\phi)$ and $0.25\cos(3\phi) - 1.25\cos(\phi)$, respectively. Since each function has non-zero Fourier coefficients for two frequencies, by Theorem 1, four basis functions suffice for steering. Because both functions contain sinusoids of the same frequencies (even though of different amplitudes), they use the same $k_j(\theta)$ interpolation coefficients.

It is convenient to have a version of Theorem 1 for functions expressed as polynomials in Cartesian coordinates x and y [31]. Applying Theorem 1 to the polynomial in polar coordinates, one can show [33] the following:

Theorem 3 *Let* $f(x,y) = W(r)P_N(x,y)$*, where* $W(r)$ *is an arbitrary windowing*

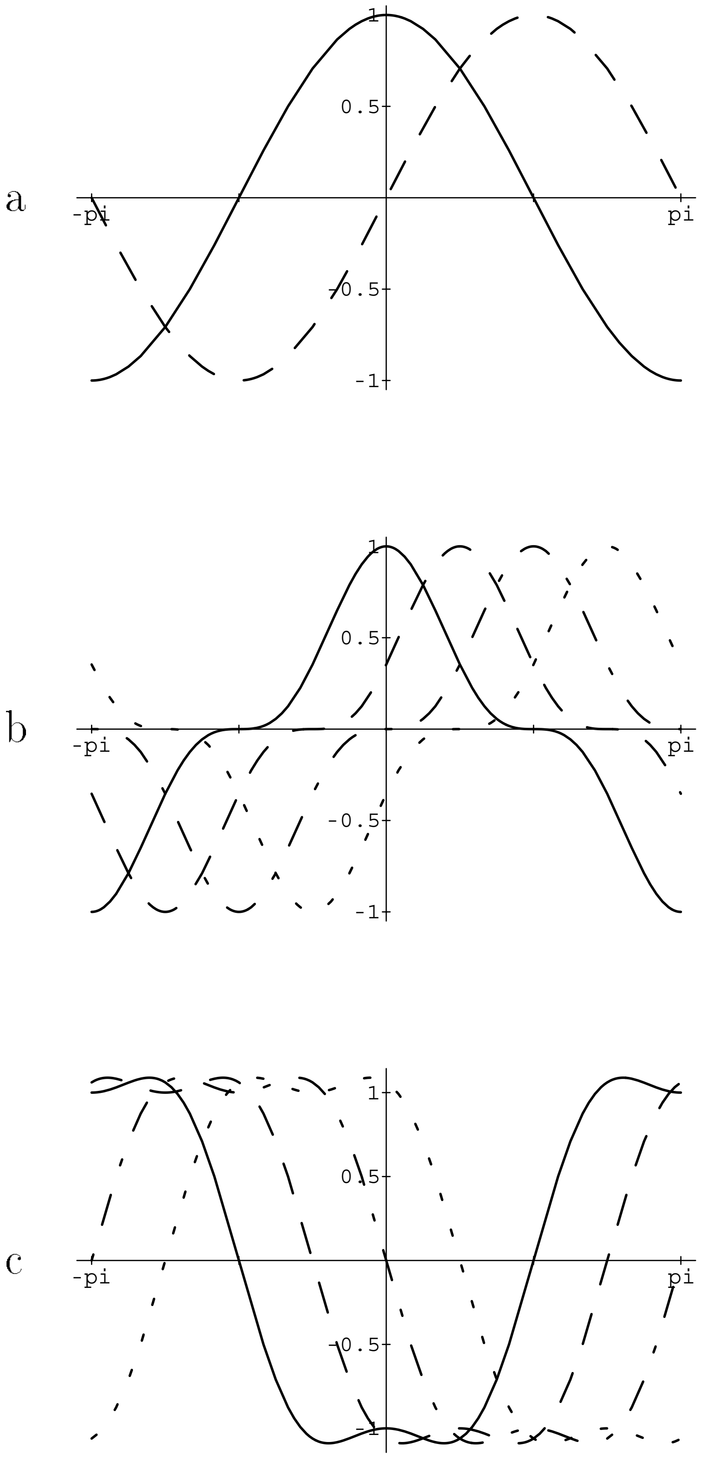


Figure 2-2: Three sets of steerable basis functions, plotted as a function of azimuthal angle, ϕ, at a constant radius. An arbitrary angular offset of each function (linear shift, as plotted here) can be obtained by a linear combination of the basis functions shown. (a) G_1 steerable basis set. (b) four basis functions for $0.25\cos(3\phi) + 0.75\cos(\phi)$; (c) four basis functions for $0.25\cos(3\phi) - 1.25\cos(\phi)$. The same interpolation functions apply for (b) as for (c).

function and $P_N(x, y)$ is an Nth order polynomial in x and y, whose coefficients may depend on r. Linear combinations of $2N + 1$ basis functions are sufficient to synthesize $f(x, y) = W(r)P_N(x, y)$ rotated to any angle. Eq. (2.10) gives the interpolation functions, $k_j(\theta)$. If $P_N(x, y)$ contains only even [odd] order terms (terms $x^n y^m$ for $n + m$ even [odd]), then $N + 1$ basis functions are sufficient, and Eq. (2.10) can be modified to contain only the even [odd] numbered rows (counting from zero) of the left hand side column vector and the right hand side matrix.

Theorem 3 allows steerable filters to be designed by fitting the desired filters with polynomials times rotationally symmetric window functions, which can be simpler than using a Fourier series in polar coordinates. However, Theorem 3 is not guaranteed to find the minimum number of basis functions which can steer a filter. Representing the function in a Fourier series in angle makes explicit the minimum number of basis filters required to steer it. In a polynomial representation, the polynomial order only indicates a number of basis functions sufficient for steering. For example, consider the angularly symmetric function, $x^2 + y^2$, written in a polar representation as $r^2 e^{0\phi}$. Theorem 2 would say that only one basis function is required to steer it; Theorem 3, which uses only the polynomial order, merely says that a number of basis functions sufficient for steering is $2 + 1 = 3$.

The above theorems show that steerability is a property of a wide variety of functions, namely all functions which can be expressed as a Fourier series in angle, or in a polynomial expansion in x and y times a radially symmetric window function. Derivatives of Gaussians of all orders are steerable because each one is a polynomial (the Hermite polynomials [78]) times a radially symmetric window function.

Figure 2-3 shows a general architecture for using steerable filters. (cf. Koenderink and van Doorn [61, 62, 63], who used such an architecture with derivatives of Gaussians, and Knutsson et al. [60] who used it with related filters.) The front end consists of a bank of permanent, dedicated basis filters, which always convolve the image as it comes in; their outputs are multiplied by a set of gain masks, which apply the appropriate interpolation functions at each position and time. The final summation produces the adaptively filtered image.

An alternative approach to the steerable filters presented here would be to project all rotations of a function onto a complete set of orthogonal basis functions, such as the Hermite functions, or the polynomials used in the facet model [43]. One could

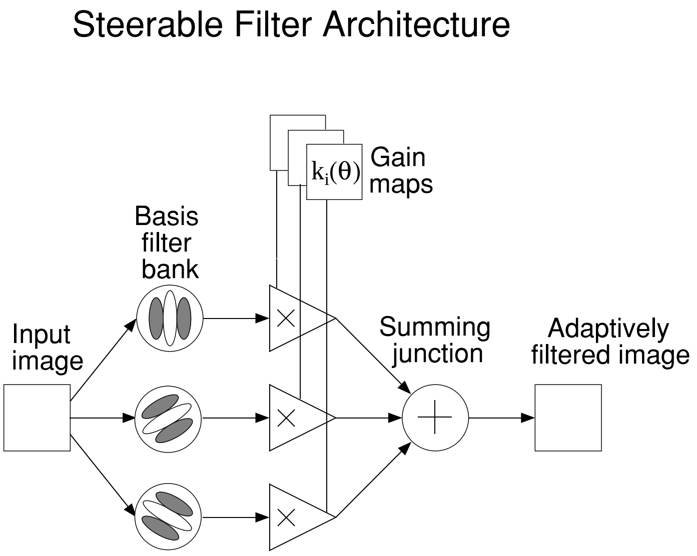


Figure 2-3: Steerable filter system block diagram. A bank of dedicated filters process the image. Their outputs are multiplied by a set of gain maps which adaptively control the orientation of the synthesized filter.

then steer the filter by changing the expansion coefficients. Such expansions allow flexible control over the filter, but for purposes of steering they generally require more basis functions than the minimum number given by Theorem 2. For example, $2N+1$ basis functions are sufficient to steer any Nth order polynomial, while a complete set of 2-D polynomial basis functions would require $(N+1)(N+2)/2$ basis functions ($n+1$ basis functions for every order $0 \leq n \leq N$). Furthermore, a general decomposition may require extra basis functions in order to fit a rotationally symmetric component of the function, which requires no extra basis functions for steering when using rotated versions of the function itself as basis functions.

2.4 Designing Steerable Filters

All functions which are bandlimited in angular frequency are steerable, given enough basis filters. But in practice the most useful functions are those which require a small number of basis filters.

As an example, we will design a steerable quadrature pair based on the frequency response of the second derivative of a Gaussian, G_2. A pair of filters is said to be in quadrature if they have the same frequency response but differ in phase by 90° (i.e. are Hilbert transforms of each other [17]). Such pairs allow for analyzing spectral strength independent of phase, and allow for synthesizing filters of a given frequency response with arbitrary phase. They have application in motion, texture, shape, and orientation analysis [4, 7, 16, 30, 37, 36, 35, 45, 47, 57, 77, 94]. Gaussian derivatives are useful functions for image analysis [61, 62, 63, 116] and a steerable quadrature pair of them would be useful for many vision tasks.

First, we design a steerable basis set for the second derivative of a Gaussian, $f(x,y) = G_2^{0^\circ} = (4x^2 - 2)e^{-(x^2+y^2)}$. This is the product of a second order, even parity polynomial and a radially symmetric Gaussian window, so, by Theorem 3, three basis functions suffice. Equation (2.10) for the interpolation functions, $k_j(\theta)$, becomes

$$\begin{pmatrix} 1 \\ e^{i2\theta} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ e^{i2\theta_1} & e^{i2\theta_2} & e^{i2\theta_3} \end{pmatrix} \begin{pmatrix} k_1(\theta) \\ k_2(\theta) \\ k_3(\theta) \end{pmatrix} . \tag{2.14}$$

Requiring that both the real and imaginary parts of Eq. (2.14) agree gives a system of three equations. Solving the system, using $\theta_1 = 0°$, $\theta_2 = 60°$, $\theta_3 = 120°$, yields

$$k_j(\theta) = \frac{1}{3}[1 + 2\cos(2(\theta - \theta_j))], \tag{2.15}$$

and we have

$$G_2^\theta = k_1(\theta)G_2^{0°} + k_2(\theta)G_2^{60°} + k_3(\theta)G_2^{120°}. \tag{2.16}$$

We can form an approximation to the Hilbert transform of G_2 by finding the least squares fit to a polynomial times a Gaussian. We found a satisfactory level of approximation (total error power was 1% of total signal power) using a 3rd order, odd parity polynomial, which is steerable by four basis functions. We refer to this approximation as H_2. Its steering formula is given with that for several other polynomial orders in [33].

Figures 2-4 (a) and (b) show 1-D slices of G_2 and H_2. The quality of the fit of H_2 to the Hilbert transform of G_2 is fairly good, as shown by the smooth, Gaussian-like energy function $(G_2)^2 + (H_2)^2$, (c), and the closeness of the magnitudes of the Fourier spectra for each function, (d).

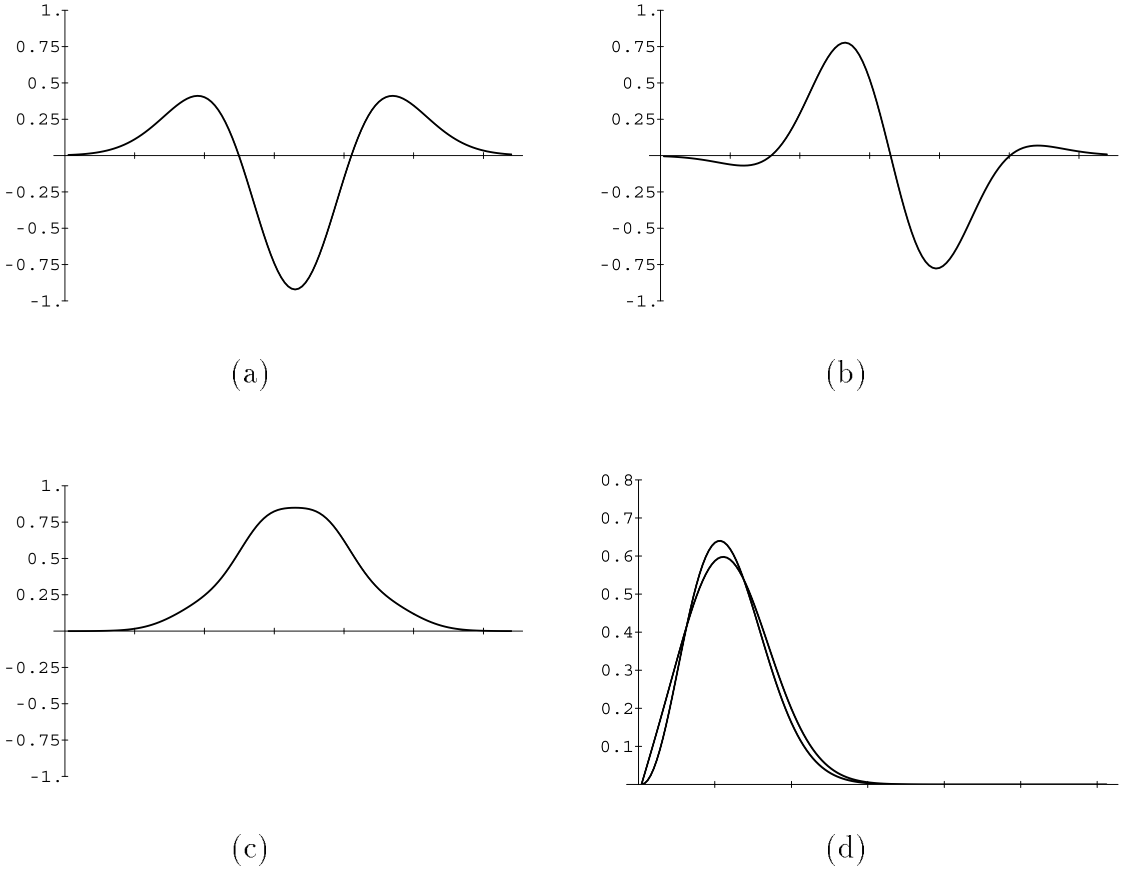
The seven basis functions of G_2 and H_2 are sufficient to shift G_2 arbitrarily in both phase and orientation. Those seven basis functions, and the magnitudes of their Fourier transforms, are shown in Fig. 2-5. The appendix of [33] lists several quadrature pairs, based on several orders of derivatives of Gaussians and fits to their Hilbert transforms.

2.5 Designing Separable Steerable Filters

For most steerable filters, the basis filters are not all x-y separable, which can present high computational costs. For machine vision applications, we would like to have only x-y separable basis functions.

We first note that for all functions f which can be written as a polynomial in x and y, there is an x-y separable basis, although it may have many basis functions. Applying the rotation formula to each x and y term of the polynomial will result in a sum of products of powers of x and y, with coefficients which are functions of the rotation angle:

$$f^\theta(x, y) = \sum_l \sum_j k_{lj}(\theta) x^l y^j. \tag{2.17}$$



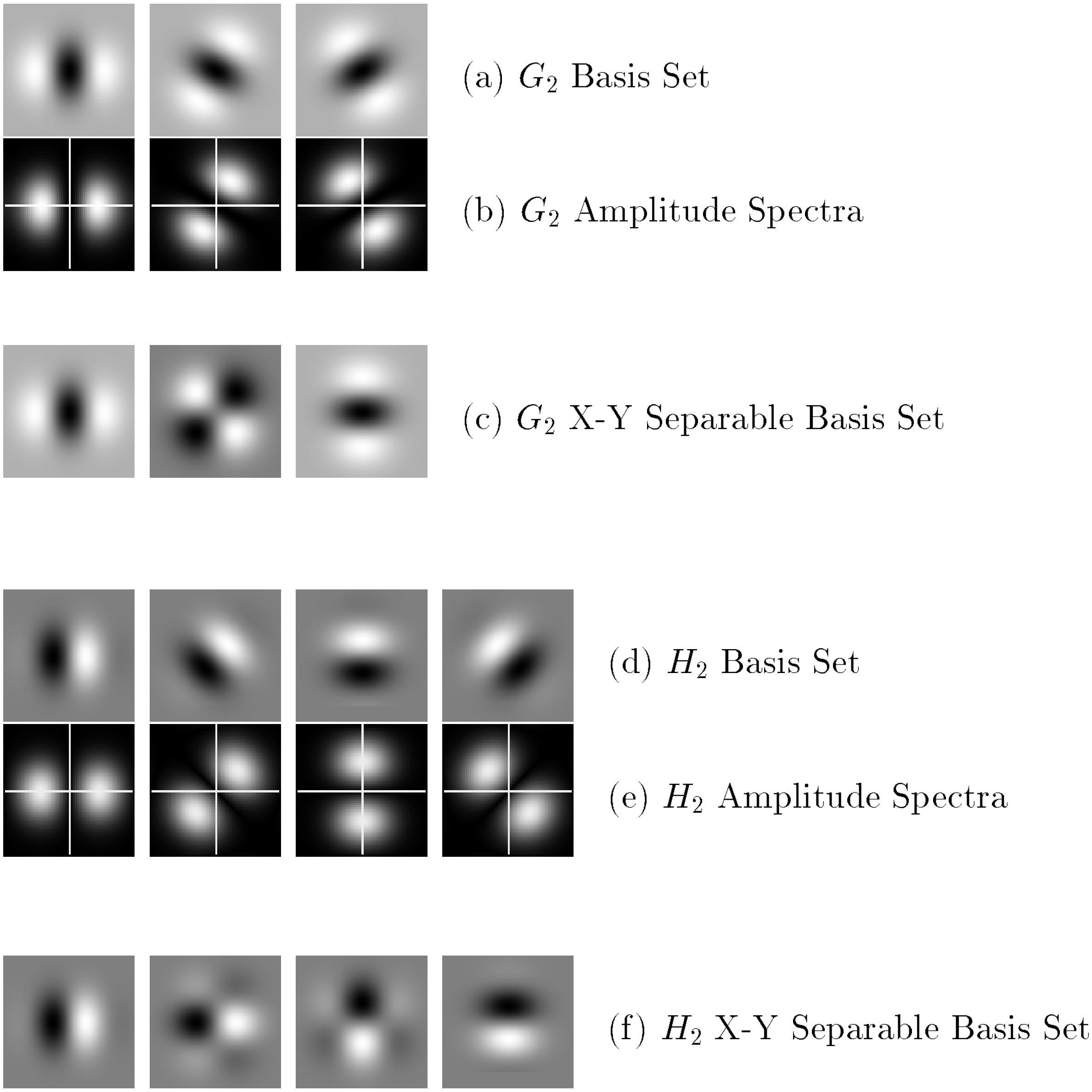
(a)

(b)

(c)

(d)

Figure 2-4: (a) G_2, 2nd derivative of Gaussian (in 1 dimension). (b) H_2, fit of 3rd order polynomial (times Gaussian) to the Hilbert transform of (a). (c) energy measure: $(G_2)^2 + (H_2)^2$. (d) magnitudes of Fourier transforms of (a) and (b).



(a) G_2 Basis Set

(b) G_2 Amplitude Spectra

(c) G_2 X-Y Separable Basis Set

(d) H_2 Basis Set

(e) H_2 Amplitude Spectra

(f) H_2 X-Y Separable Basis Set

Figure 2-5: G_2 and H_2 quadrature pair basis filters (rows (a) and (d)). The filters in rows (a) and (d) span the space of all rotations of their respective filters.. G_2 and H_2 have the same amplitude spectra (rows (b) and (e)), but 90° shifted phase. Steerable G_2 and H_2 filters can measure local orientation direction and strength, and the phase at any orientation. Rows (c) and (f) show equivalent x-y separable basis functions which can also synthesize all rotations of G_2 and H_2, respectively.

Each x and y product in the rotated polynomial can be thought of as an x-y separable basis function, with its coefficient $k_{lj}(\theta)$ the interpolation function.

In many cases, however, there exists an x-y separable basis set which contains only the minimum number of basis filters, yet spans the space of all rotations for the function of interest. Such a separable basis allows steerable filters to be applied with high computational efficiency. Rows (c) and (f) of Fig. 2-5 show x-y separable basis sets for the G_2 and H_2 filters. Reference [33] gives a derivation of the steering formulas for these x-y separable functions, shows how to find the separable basis functions, and gives the functional forms and digital filter values for x-y separable versions of the G_2, H_2, and G_4 and H_4 basis filters. See also [87] for how to make x-y separable versions of a single oriented filter.

2.6 Discrete Space Filters

The steering theorems have been derived for continuous functions, and one might be concerned that new difficulties would arise when one worked with discretely sampled functions. But if a continuous function is steerable, then a sampled version of it is steerable in exactly the same fashion, because the order of spatial sampling and steering are interchangeable. The weighted sum of a set of spatially sampled basis functions is equivalent to the spatial sampling of the weighted sum of continuous basis functions. So one can obtain digital steerable filters by simply sampling a continuous filter. Spatially sampled versions are given for G_2, H_2, G_4 and H_4 in [33].

Filters can also be designed in the frequency domain, where one may separate the radial and angular parts of the design [57]. Conventional filter design techniques [64, 82] allow the design of a circularly symmetric 2-D filter with a desired radial response. Then, one can impose on that filter the angular variation needed to make a steerable basis set by frequency sampling [64] (if the angular response is relatively smooth). Inverse transforming the frequency sampled response gives the filter kernel.

Figure 2-6 shows an example of this. The filter was designed to be part of a steerable, self-inverting pyramid image decomposition [103], described below. The constraints on the multi-scale decomposition lead to the radial frequency response shown in Fig. 2-6 (a). We used the frequency transformation method [64] to convert the 1-D filter to a nearly angularly symmetric 2-D filter, Fig. 2-6 (b).

Having selected a radial frequency band, we next divided the band into four oriented subbands by imposing an angular variation of $\cos^3(\nu)$, where ν is azimuthal angle in frequency. This function has four angular frequencies (±3 and ±1) and so, by Theorem 1, requires four basis functions to steer. We Fourier transformed the radially symmetric kernel, multiplied by the four desired $\cos^3(\nu - \theta_j)$ angular responses, and inverse transformed to obtain the basis filter impulse responses. Figure 2-6 (c - f) shows the frequency amplitude responses of the resulting digital steerable filters.

2.7 Steerable Pyramid for Multi-Scale Decomposition

The steerable filters described above were designed to form a multi-scale, self-inverting pyramid decomposition [103]. Applying each filter of the decomposition once to the signal gives the transform coefficients; applying each filter a second time (with filter tap values reflected about the origin) and adding the results reconstructs a low-passed version of the image. Because all of the filters of the pyramid are bandpass, a high-pass residue image must be added back in to reconstruct the original image (as with [109]) . To implement this decomposition, we designed the angular and radial components of the polar separable design so that the squares of the responses of each filter added to unity in the frequency plane.

Figure 2-7 shows the steerable pyramid representation. The four bandpass filters at each level of the pyramid form a steerable basis set. The pyramid basis filters were oriented at 0°, 45°, 90°, 135°, but the coefficients for any filter orientation can be found from a linear combination of the four basis filter outputs. When the basis filters are applied again at each level, the pyramid collapses back to a filtered version of the original image with near-perfect agreement. The steerable pyramid image transform allows control over orientation analysis over all scales.

The steerable pyramid is an image transform for which all of the basis functions are derived by dilation, translation, and rotation of a single function, and therefore it may be considered to be a wavelet transform [41, 71]. Most work on wavelet image decomposition has involved discrete orthogonal wavelets, in particular those known as quadrature mirror filters (QMF's) [29, 71, 101, 107]. Pyramids made from QMF's and other wavelets can be extremely efficient for image coding applications. Such representations are usually built with x-y separable filters on a rectangular lattice [6, 71, 115], which significantly limits the quality of orientation tuning that can be achieved. Simoncelli and Adelson [6, 100] have devised QMF pyramids based on filters placed on a hexagonal lattice; in addition to being orthogonal and self-similar, these pyramids have good orientation tuning in all bands. However, the basis functions are not steerable, so the representation is not optimal for orientation analysis. Non-orthogonal pyramids with orientation tuning have been described by [27, 38, 74, 109].

Unlike the pyramids based on QMF's, the steerable pyramid described here is significantly overcomplete: not counting the residual image, there are $5\frac{1}{3}$ times as many coefficients in the representation as in the original image ($1\frac{1}{3}$ times over-complete, as with the Laplacian pyramid [19], but for each of 4 orientations). The overcompleteness limits its efficiency but increases its convenience for many image processing tasks. Although it is non-orthogonal, it is still self-inverting, meaning that the filters used to build the pyramid representation are the same as those used for reconstruction.

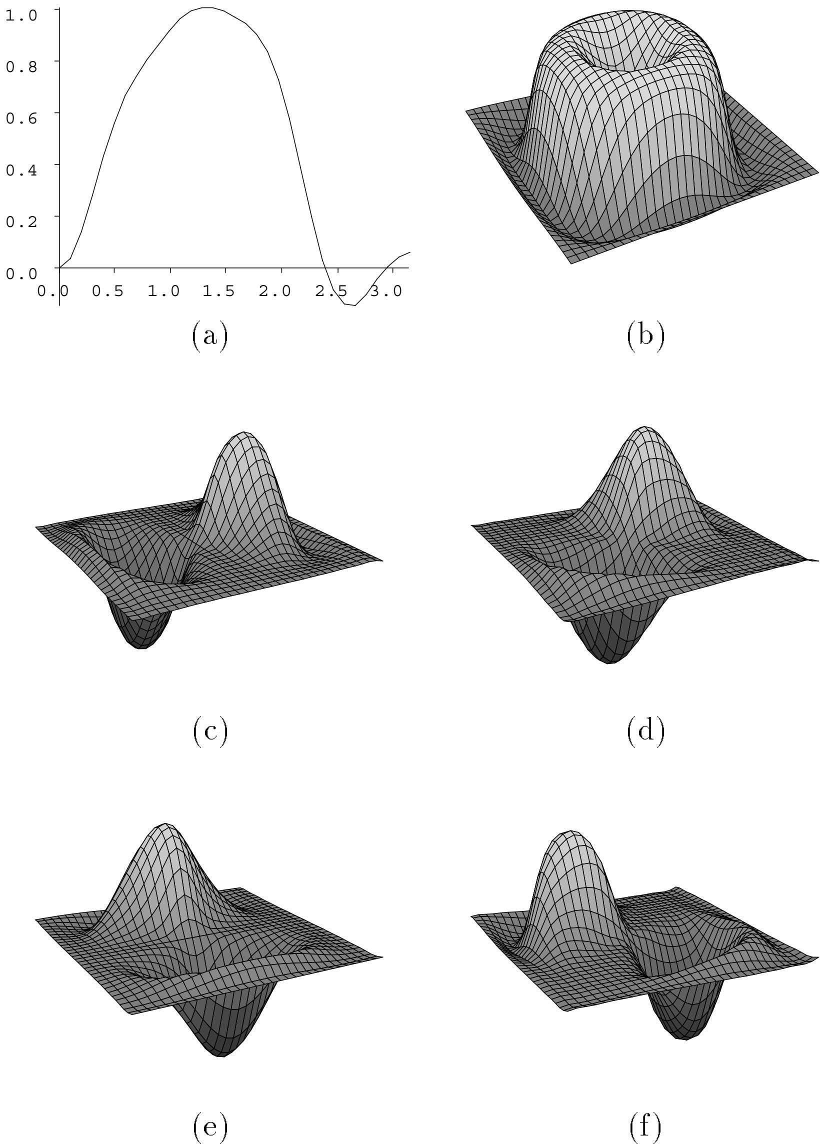
2.8 Summary of Steerable Filters

Steerable filters can be used for a variety of operations involving oriented filters. The oriented filter, rotated to an arbitrary angle, is formed as a linear combination of basis filters. Once the basis filter responses are known, the response of the filter steered (rotated) to an arbitrary angle, can easily be found. A similar technique can be used to control the phase of the filters. We have shown that most filters can be steered in this manner, given enough basis filters, and have described how to determine the minimum number of basis functions required, and how to interpolate between them in angle.

Steerable filters can be applied to many problems in early vision and image analysis, including texture and orientation analysis, image enhancement, motion analysis, noise removal, image representation, and shape from shading [57, 58, 33, 103, 36, 89]. Figures 2-8, 2-10, and 2-9 show some examples. Because the synthesis of the rotated filter is analytic and exact, steerable filters offer advantages for image analysis over *ad hoc* methods of combining oriented filters at different orientations. Many processing schemes require no additional convolution after the initial pass through the basis filters. Even to use a filter at just one orientation, it will often be more efficient to apply the entire x-y separable basis set and steer the filter to that orientation than to apply the non-separable filter.

We designed steerable quadrature pair filters which we will use later to analyze orientation and phase and to find contours. We also built a self-similar steerable pyramid representation, allowing the analysis and manipulation of oriented structures at all scales. [103, 33] describe applications of the steerable pyramid to multi-scale stereo matching, noise removal, and shape from shading.

In the two next chapters, in preparation for analyzing junctions, we use steerable filters to analyze orientation, contours, and phase.



(a) (b)

(c) (d)

(e) (f)

Figure 2-6: Frequency domain filter response plots, illustrating design procedure for digital steerable filter. (a) Desired radial frequency distribution, plotted from 0 to π. (b) Desired angularly symmetric two-dimensional frequency response, obtained through frequency transformation. The profile in (b) was multiplied by the desired $\cos^3(\theta - n\theta)$ angular frequency responses and inverse transformed to yield the steerable basis set. (c) – (f) The imaginary component of the frequency responses of the resulting steerable filters.

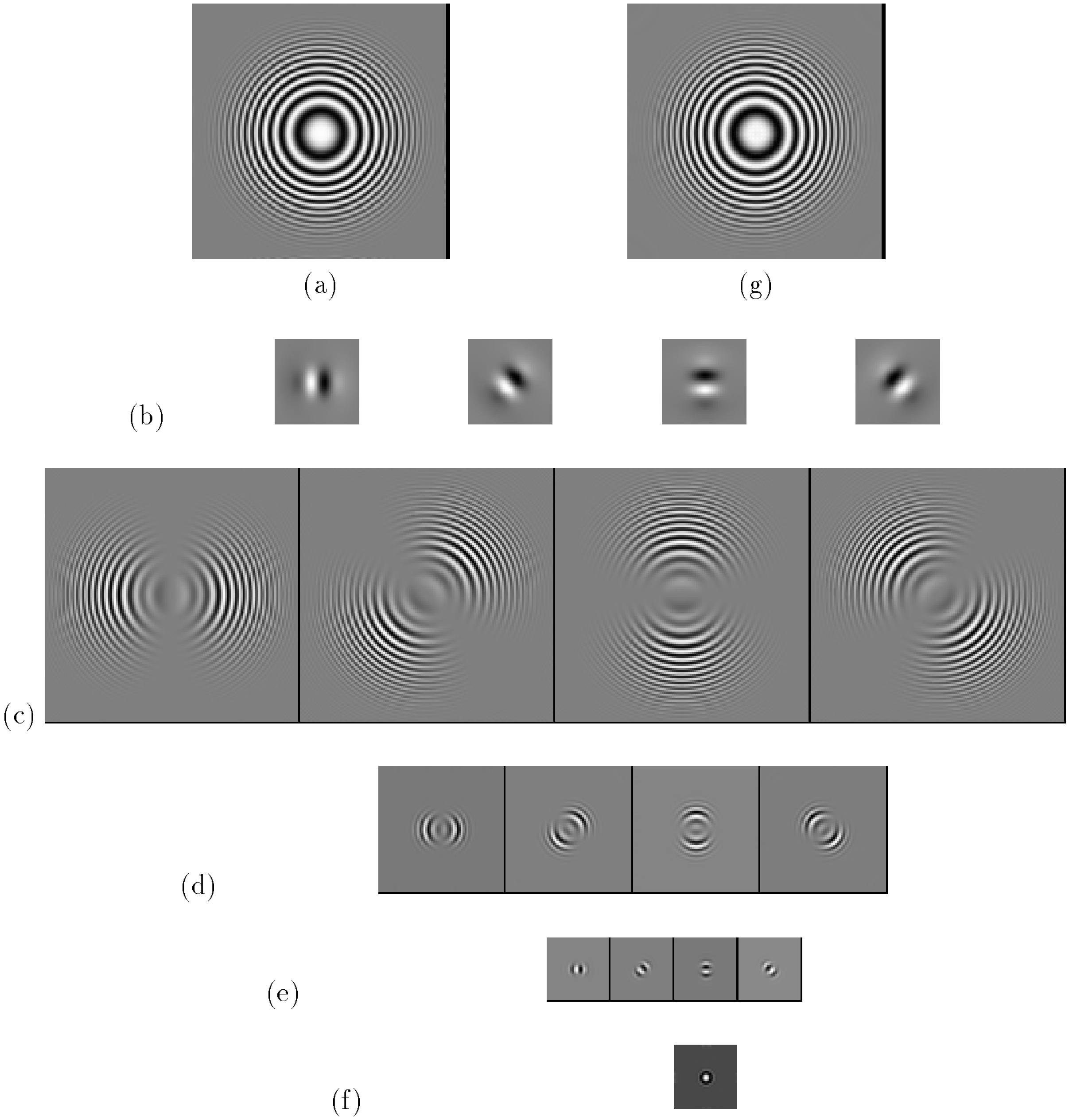


Figure 2-7: Steerable image transform. (a) Low-pass filtered original image. (b) Odd-phase analyzing filters, oriented at 0°, 45°, 90°, 135°. These four filters form a steerable basis set; any orientation of this filter can be written as a linear combination of the basis filters. (c) - (e) Steerable, bandpass coefficients in a multi-scale pyramid representation of (a). A linear combination of these transform coefficients will synthesize the transform coefficient for analyzing filters oriented at any angle. (f) Low-pass image. (g) Image reconstructed from the pyramid representation, showing near-perfect agreement with (a).

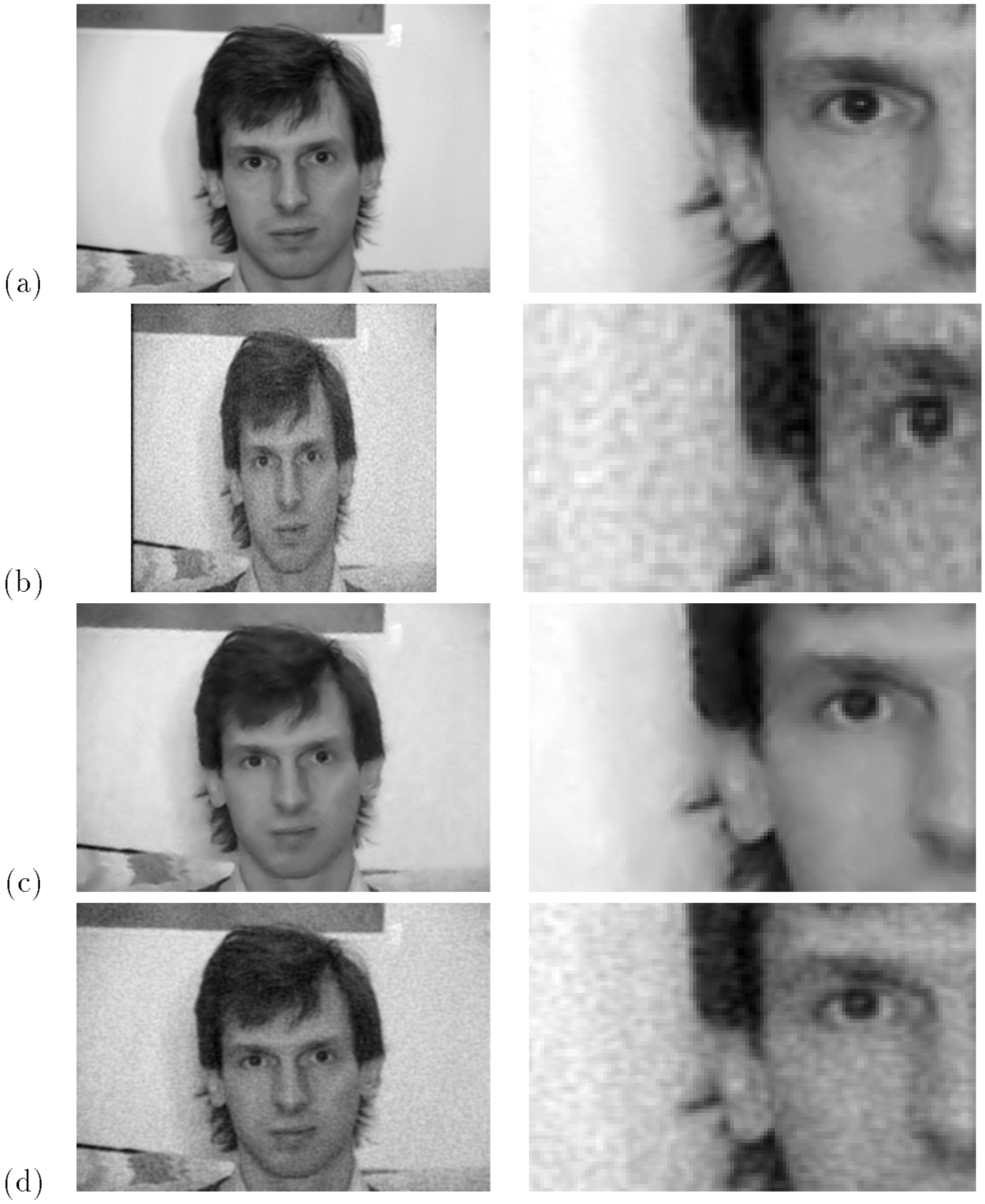


Figure 2-8: Noise removal example using steerable filters. Figures on the right are enlarged portions of those on the left. (a) The original noise-free image. (b) The image corrupted by noise. SNR is 12.42 dB. (c) Results of image restoration using steerable pyramid. The image was decomposed into the multi-resolution oriented sub-bands of the steerable pyramid and processed to remove noise in a way that independent of the image orientation. See [103] for complete description. The processing substantially removes the noise, while leaving important image features intact. The SNR of the processed image is 23.0 dB. For comparison, the results of image restoration using a Wiener filter are shown in (d). The visual appearance of the noise is much worse, while the image structures are more blurred. SNR is 19.24 dB.

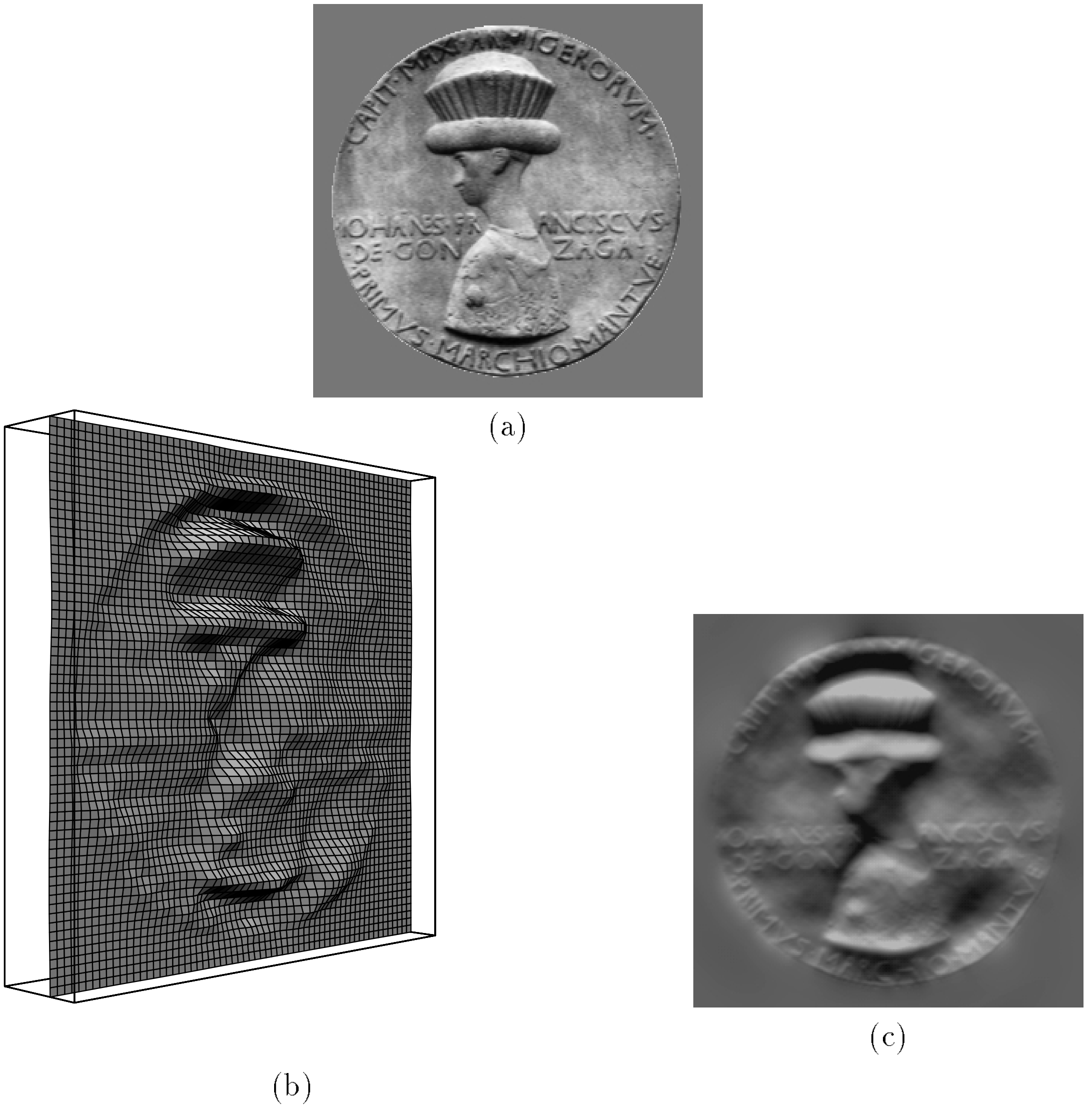


Figure 2-9: Example showing the use of steerable filters in shape-from-shading analysis. (a) Image input for (b) Range map resulting from linear shape-from-shading analysis [86] using steerable pyramid. The approximations used in the linear shading algorithm apply for oblique illumination. The result is displayed as a low-resolution 3-D plot. Steering was used to accommodate different light directions, as described in [33]. (c) Same range map, with pixel intensity showing surface height. This simple mechanism correctly derived the image surface characteristics.

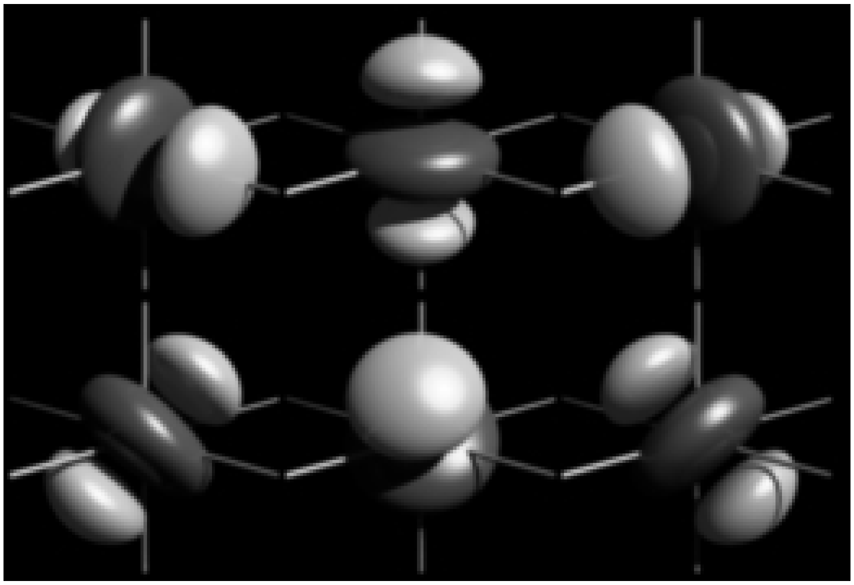


Figure 2-10: Example of a three-dimensional steerable filter. Surfaces of constant value are shown for the six basis filters of a second derivative of a three-dimensional Gaussian. Linear combinations of these six filters can synthesize the filter rotated to any orientation in three-space. Such three-dimensional steerable filters are useful for analysis and enhancement of motion sequences or volumetric image data, such as MRI or CT data. For discussions of steerable filters in three or more dimensions, see [59, 58, 33, 89]. (Martin Friedmann rendered this image with the Thingworld program).

Chapter 3

Analyzing Orientation

3.1 Analyzing the Dominant Orientation

Orientation analysis is an important task in early vision [54, 57, 60, 117, 112]. Knutsson and Granlund [57] devised an elegant method for combining the outputs of quadrature pairs to extract a measure of orientation. We describe a related method which makes optimal use of the filters designed in Section 2.4. We measure the orientation strength along a particular direction, θ, by the squared output of a quadrature pair of bandpass filters steered to the angle θ. We call this spectral power the *oriented energy*, $E(\theta)$.

Using the nth derivative of a Gaussian and its Hilbert transform as our bandpass filters, we have:

$$E_n(\theta) = [G_n^\theta]^2 + [H_n^\theta]^2. \tag{3.1}$$

Writing G_n^θ and H_n^θ as a sum of basis filter outputs times interpolation functions, Eq. (3.1) simplifies to a Fourier series in angle, where only even frequencies are present, because of the squaring operation:

$$E_n(\theta) = C_1 + C_2 \cos(2\theta) + C_3 \sin(2\theta) + [\text{higher order terms} \ldots]. \tag{3.2}$$

We use the lowest frequency term to approximate the direction, θ_d and strength,

S, of the dominant orientation (the orientation which maximizes $E_n(\theta)$),

$$\theta_d = \frac{\arg[C_2, C_3]}{2} \tag{3.3}$$

$$S = \sqrt{C_2^2 + C_3^2}. \tag{3.4}$$

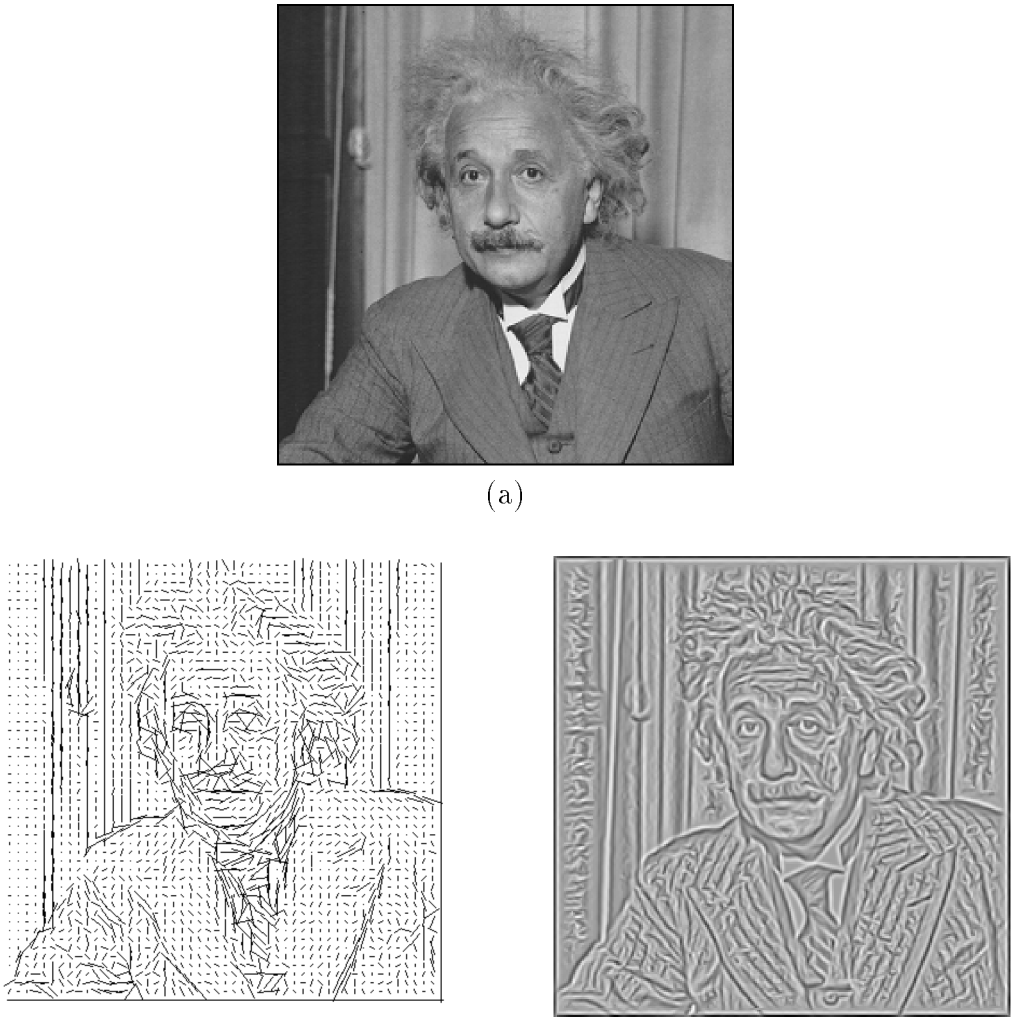
This approximation is exact if there is only one orientation present locally.

Figure 3-1 (b) shows an orientation map derived using this method, using G_2 and H_2 to measure $E_2(\theta)$. The line lengths are proportional to S, the contrast along that orientation. The measured orientations and strengths accurately reflect the oriented structures of the input image. This measurement of orientation angle was made directly from the basis filter outputs, without having to actually perform the steering operation. [33] lists C_2 and C_3 as functions of the basis filter outputs for x-y separable G_2 and H_2 basis filter outputs.

One can remove noise and enhance oriented structures by angularly adaptive filtering [60, 53, 73]. Steerable filters offer an efficient method for such processing. We took the appropriate combinations of the G_2 basis filter outputs for Fig. 3-1 (a) to adaptively steer G_2 along the local direction of dominant orientation. No additional filtering was required for this step. To enhance local contrast, we divided the filtered image by a local average of its absolute value. The result, Fig. 3-1 (c), enhances the oriented structures of Fig. 3-1 (a) which lie within the G_2 passband. The entire process of finding the dominant orientation, steering G_2 along it, and deriving the enhanced image involved only a single pass of the image through the basis filters.

3.2 Analyzing Multiple Local Orientations

Junctions, certain textures, and transparent or overlapping objects all may contain more than one local orientation. The 3-dimensional version of this also occurs in motion analysis [4, 45], for example in the presence of occlusion or transparency. Filters with broad orientation tuning, such as G_2 and H_2, typically give oriented energy responses which do not reflect the orientations at these regions. Most researchers [88, 98, 99, 102, 48, 33] therefore use filters of tight orientation tuning to analyze regions with multiple orientations. The price for that is more basis filters. We will show later an alternate approach which uses the basis filters more efficiently. First,



(a)

(b)

(c)

Figure 3-1: (a) Original image of Einstein, (b) Orientation map of (a) made using the lowest order terms in a Fourier series expansion for the oriented energy as measured with G_2 and H_2. (c) Image of Einstein with oriented structures enhanced. The G_2 basis filter outputs were combined to adaptively steer G_2 so that it lined up with the dominant orientation everywhere. Both operations, finding the orientation map and the adaptive filtering, required only one pass through the steerable basis filters.

let us explore the use of more tightly tuned filters.

3.2.1 Using Narrow Filters

A steerable filter with a narrower frequency tuning, such as the fourth derivative of a Gaussian, G_4, will give a higher resolution analysis of orientation. The filter taps and analytical form for the steerable quadrature filter pair G_4 and H_4 are given in [33]. (H_4 is the least squares fit of a 5th order polynomial times a Gaussian to the Hilbert transform of G_4.)

Figure 3-2 shows two test images, a vertical line, and a cross, and their oriented energy as a function of angle, measured at the center using a G_4, H_4 quadrature pair, plotted in both Cartesian and polar coordinates. Note that the steerable filters adequately describe the multiple orientations of the cross, as seen by the floret shape.

Fig. 3-3 shows a test image, (a), and several measures of its oriented energy, using the G_4, H_4 quadrature pair. Fig. 3-3 (b) shows the DC component of oriented energy, the angular average of Eq. (3.1). Because we are using a quadrature pair, the energy measure responds to both lines and edges. Fig. 3-3 (c) is a measure of orientation where only one orientation is allowed at each point, calculated from the lowest order Fourier terms of Eq. (3.1). No dominant orientation is detected at intersections of oriented structures. Fig. 3-3 (d) shows polar plots of the oriented energy distribution for various points in the image. Note that this measure captures the multiple orientations present at intersections and corners, shown by the florets there. These measures could all be calculated by constructing a different quadrature pair for each orientation observed; however, using the steerable filters greatly reduces the computational load.

Figure 3-4 shows a detail from a texture, and the corresponding polar orientation maps at every pixel in the texture image, offering a rich description of the textural details. Note that florets of one dominant orientation are separated from florets of another dominant orientation by florets where both orientations are present.

3.2.2 Removing Interference Effects

Using filters of sharp orientation tuning to analyze regions of multiple orientations has a drawback: it requires many filters to make a steerable basis set. The approach we

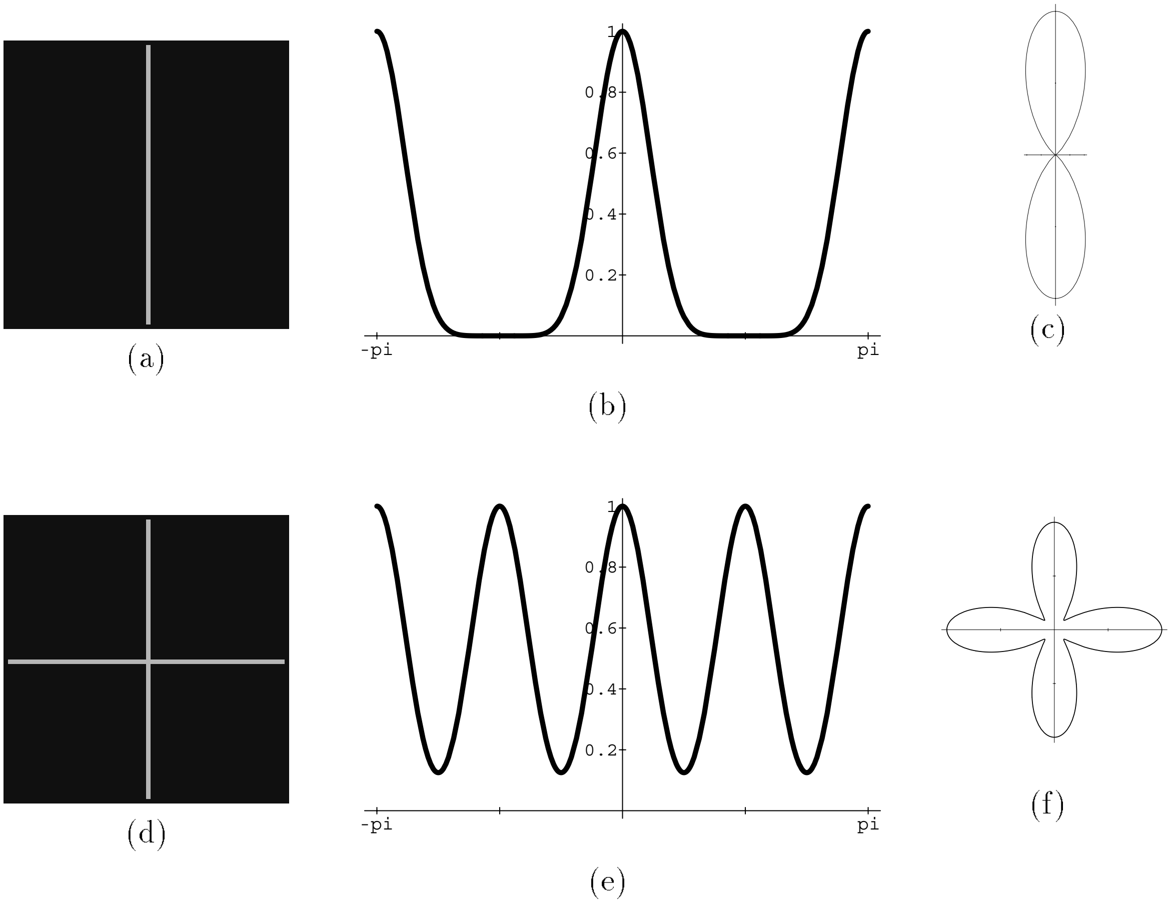


Figure 3-2: Test images of (a) vertical line and (b) intersecting lines. (c) and (d): Oriented energy as a function of angle at the centers of test images (a) and (b). Oriented energy was measured using the G_4, H_4 quadrature steerable pair. (e) and (f): polar plots of (c) and (d).

describe here requires fewer basis filters and is therefore more efficient. Alternatively, one can use this approach to increase the angular resolution of a given set of analyzing filters.

An implicit assumption made when using energy measures to analyze multiple orientations in space or space-time is that the energy of the multiple structures is the sum of the energies of the structures taken individually [88, 98, 99, 48, 33]. Of course, this is not the case in general: linear superposition holds for the filter *amplitude* responses, but not for the sum of their squares.

A frequency domain analysis of the energy measure lets us see the problem and a remedy. First, let us find the Fourier transform of the energy measure. Suppose we have a quadrature pair of oriented, bandpass filters, called G and H. The energy

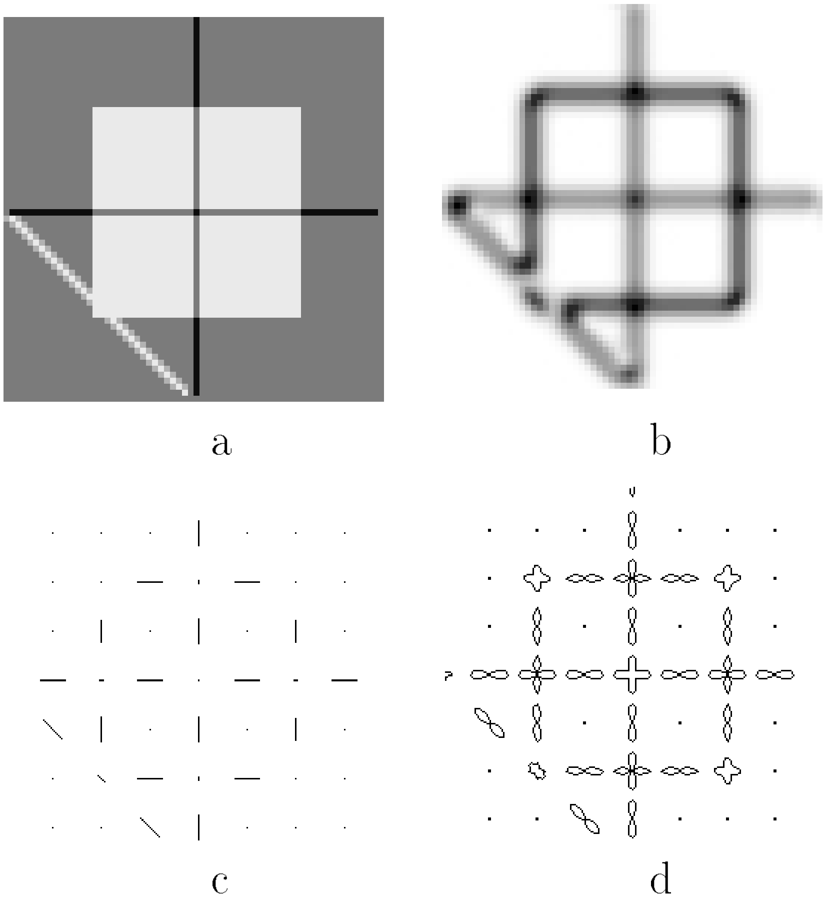


Figure 3-3: Measures of orientation derived from G_4 and H_4 steerable filter outputs. (a) Input image for orientation analysis (b) Angular average of oriented energy as measured by G_4, H_4 quadrature pair. (c) Dominant orientation plotted at each point. No dominant orientation is found at the line intersection or corners. (d) Oriented energy as a function of angle, shown as a polar plot for a sampling of points in the image (a). Note the multiple orientations found at intersection points of lines or edges and at corners, shown by the florets there.

measure derived from the quadrature pair is $G^2 + H^2$. If the filter G has even symmetry, then the transforms, $\hat{G}$ and $\hat{H}$, will be as shown schematically in Fig. 3-5 (a) and (b). $\hat{G}$ is real and even and $\hat{H}$ is imaginary and odd, shown by the lobes being labelled "plus, plus" and "minus, plus", for $\hat{G}$ and $\hat{H}$, respectively.

By the convolution theorem, $\widehat{G^2} = \hat{G} * \hat{G}$, where $*$ represents convolution. That is shown in Fig. 3-6 (a) and (b), along with $\widehat{H^2} = \hat{H} * \hat{H}$. Each has a center lobe, which is the autocorrelation function of a single lobe of the bandpass filter responses, and two side lobes, from the interaction of one bandpass lobe with another. The lobes in $\hat{G} * \hat{G}$ and $\hat{H} * \hat{H}$ are identical, except for the signs shown in the figure. In the transform of the energy measure $G^2 + H^2$, the side lobes cancel exactly, and one is left with the single lobe shown in Fig. 3-6 (c), which is the autocorrelation function of a single lobe of the transform of the bandpass filters G or H. This lobe, centered at DC, has been demodulated down from its original bandpass response.

Having found the transform of the energy measure, let us suppose we apply the energy measure in a region of two orientations, such as that shown in Fig. 3-7 (a). The Fourier transform of the two intersecting lines is as shown in Fig. 3-7 (b), two lines perpendicular to each of the other two. The response of filter G will be as shown in Fig. 3-7 (c). The energy measure $G^2 + H^2$ will be the center lobe of the autocorrelation of Fig. 3-7 (d), shown in Fig. 3-7 (e).

The energy response of Fig. 3-7 (e) has contributions at three different frequencies. The DC term arises when Fig. 3-7 (d) and its copy are superimposed in the autocorrelation. This is a point by point squaring of the power at each frequency in Fig. 3-7 (d). For this term, superposition does hold—the sum of the squared power from each line is equal to the squared power from the two lines. The other two frequency contributions are *interference* terms coming from the interaction of one line with another in the autocorrelation of Fig. 3-7 (d). These are not present in the energy response of either line by itself and they cause the superposition principal to fail for the energy measure of the two lines. Thus, the components for which superposition holds and those for which it does not are at different spatial frequencies. A linear filter can separate one from the other. Low-pass filtering the energy outputs will substantially remove interference effects from the energy measure. Only the DC term of the autocorrelation of Fig. 3-7 (d) will remain, for which the principle of superposition applies.

We confirm the above theoretical analysis experimentally. Figure 3-8 (a) and (c) show a horizontal and a vertical line. (b) and (d) are their respective floret polar plots showing orientation strength as a function of angle, measured using the G_2, H_2 energy measure. The cross in Fig. 3-9 (a) is the sum of the two lines. To analyze this junction, we would like the energy measure for the cross to give the sum of the energy measures for the horizontal and vertical lines of which it is composed. However, that is not the case; the floret polar plot (b) is a complex figure with some florets showing the correct orientations, some the wrong ones, and some showing no preferred orientations at all. For comparison, (c) illustrates what the linear sum of the horizontal and vertical line oriented energies would look like, if we were only able to measure it. However, if we low-pass filter the energy outputs of the cross, we obtain the simple floret plot shown in (d). Each floret shows the orientations of the two lines which make up the cross. This result is virtually identical to the desired sum of the blurred energies of the horizontal and vertical lines, (e).

To blur the floret plots, one could find the energy at each orientation to be plotted in the floret, spatially blur it, and plot the resulting point in the energy floret. However, because we are using steerable filters, that is not necessary. The G_2 filter only has angular Fourier frequencies 0 and ± 2. Its squared energy $G_2^2 + H_2^2$ can therefore only have Fourier frequencies 0, ± 2, and ± 4. Thus, the energy at 5 orientations specifies the energy at all orientations and we only need to spatially blur the energy outputs at those 5 basis filters. Theorem 1 (or the formulas of Table 1 in [33]) let one interpolate in angle between the basis responses.

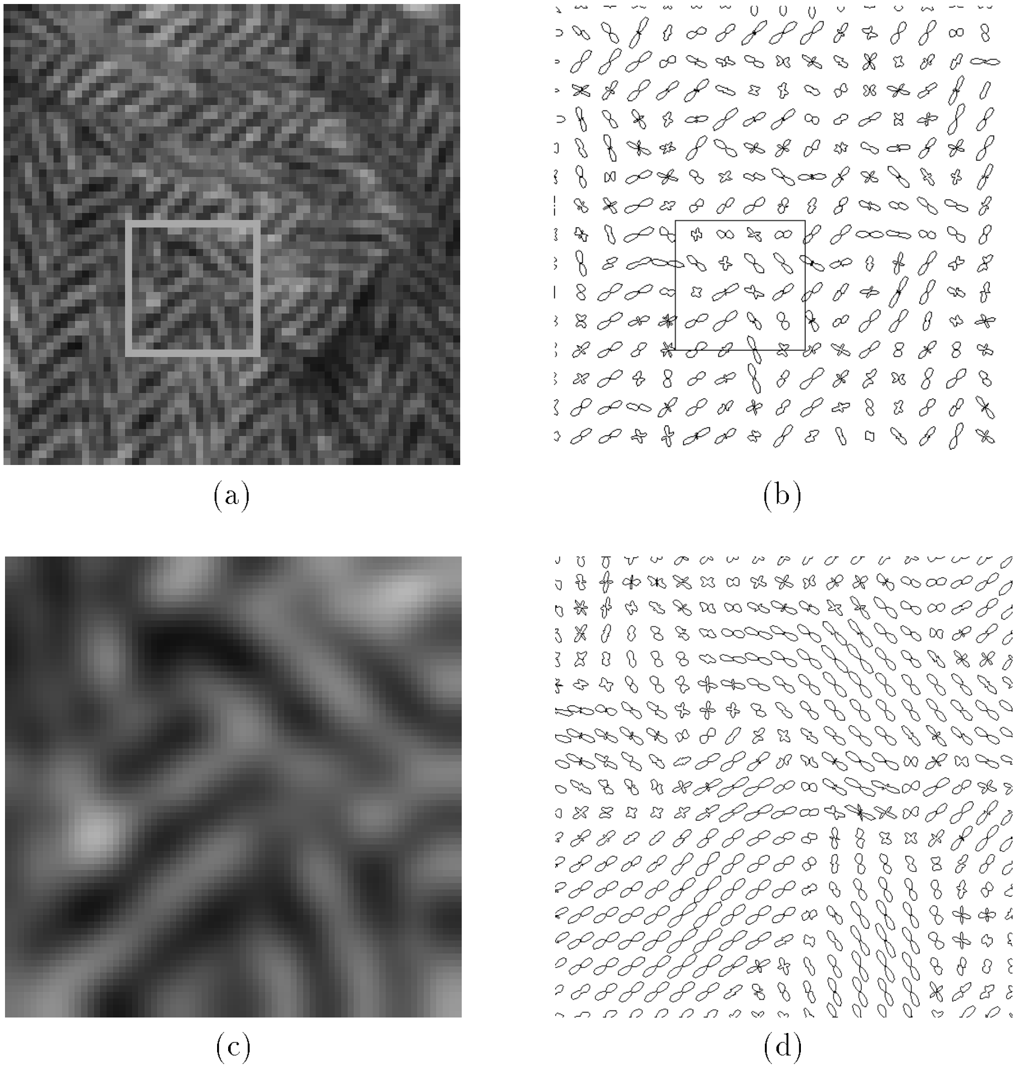
We have found a post-processing step which allows us to treat the oriented energies of overlapping structures as a sum of the energies of the individual parts. Before, we needed to use narrowly tuned filters to do this; now we can use the more efficient, broadly tuned filters. An analysis for which superposition holds is essential for the proper processing of junctions. This analysis also applies to other algorithms which involve squaring bandpass filter outputs (or derivatives) in regions of multiple orientations. An important case includes motion analysis in the presence of occlusion or transparency.

Comparison of the two methods

From Figs. 3-8 and 3-9 one can see how narrowing the angular tuning of the filters allows superposition to approximately hold for the oriented energies: if the bandpass filters cover only one oriented structure in frequency, then there will be no interference effects in the energy term, Fig. 3-9 (b). However, this requires using far more filters than are necessary to represent the two or three oriented structures which may be present at a junction. By blurring the G_2, H_2 energy measures, we could easily represent the two orientations in Fig. 3-9 (a).

One could object that blurring the oriented energies will lower the spatial resolution; however using narrow filters may lower the spatial resolution by at least as much. Let the separation in frequency of the transforms of the two oriented structures at the passband of the analyzing filters be D. To avoid interference effects, the narrow filters must be substantially confined within a length D in frequency. By the uncertainty relation [17], this will require filters of a spatial size $\gtrsim \frac{1}{D}$. Our preferred approach, blurring the more coarsely tuned energy outputs, requires a low-pass filter with a width in frequency of $\approx 2D$, or a spatial extent of $\gtrsim \frac{1}{2D}$. Thus, blurring the coarsely tuned energy outputs could actually give a higher resolution description of the image structure than using the energy measure from the narrowly tuned filters.

We note that with either approach there is a tradeoff between spatial and angular resolution: to remove interference effects from two lines which are close together in angle, one must apply a very severe low-pass filter to the coarsely tuned energies, or alternatively use very narrowly tuned energy filters. Either results in a large positional uncertainty. This agrees with the intuitive notion that it should require a large area to resolve two oriented structures which are close to each other in angle.



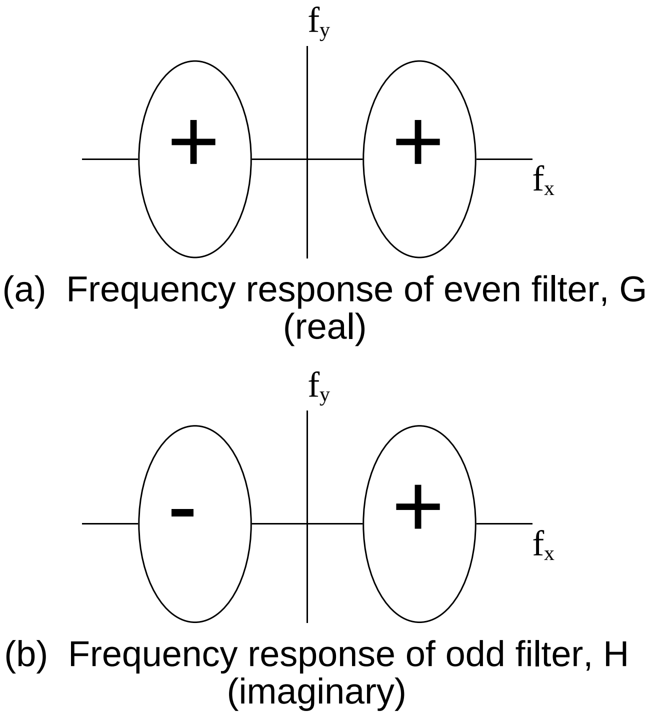
(a)

(b)

(c)

(d)

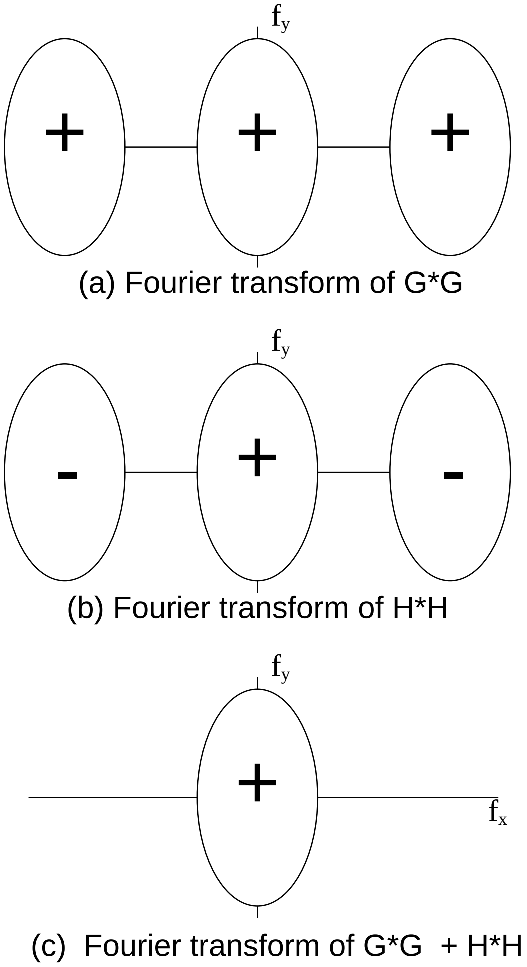
Figure 3-4: (a) Texture image; (b) Polar plots of oriented energy of (a) at every fourth pixel. Each plot is normalized by the average over all angles of the oriented energy. (c) Detail of (a) (zoomed and blurred); (d) Normalized polar plots showing oriented energy of (c) at every pixel.



(a) Frequency response of even filter, G (real)

(b) Frequency response of odd filter, H (imaginary)

Figure 3-5: Frequency content of two bandpass filters in quadrature. (a) even phase filter, called G in text, and (b) odd phase filter, H. Plus and minus sign illustrate relative sign of regions in the frequency domain. See Fig. 3-6 for calculation of the frequency content of the energy measure derived from these two filters.



(a) Fourier transform of G*G

(b) Fourier transform of H*H

(c) Fourier transform of G*G + H*H

Figure 3-6: Derivation of energy measure frequency content for the filters of Fig. 3-5. (a) Fourier transform of $G * G$. (b) Fourier transform of $H * H$. Each squared response has 3 lobes in the frequency domain, arising from convolution of the frequency domain responses. The center lobe is modulated down in frequency while the two outer lobes are modulated up. (There are two sign changes which combine to give the signs shown in (b). To convolve H with itself, we flip it in f_x and f_y, which interchanges the $+$ and $-$ lobes of Fig. 3-5 (b). Then we slide it over an unflipped version of itself, and integrate the product of the two. That operation will give positive outer lobes, and a negative inner lobe. However, H has an imaginary frequency response, so multiplying it by itself gives an extra factor of -1, which yields the signs shown in (b)). (c) Fourier transform of the energy measure, $G * G + H * H$. The high frequency lobes cancel, leaving only the baseband spectrum, which has been demodulated in frequency from the original bandpass response. This spectrum is proportional to the sum of the auto-correlation functions of either lobe of Fig. 3-5 (a) and either lobe of Fig. 3-5 (b).

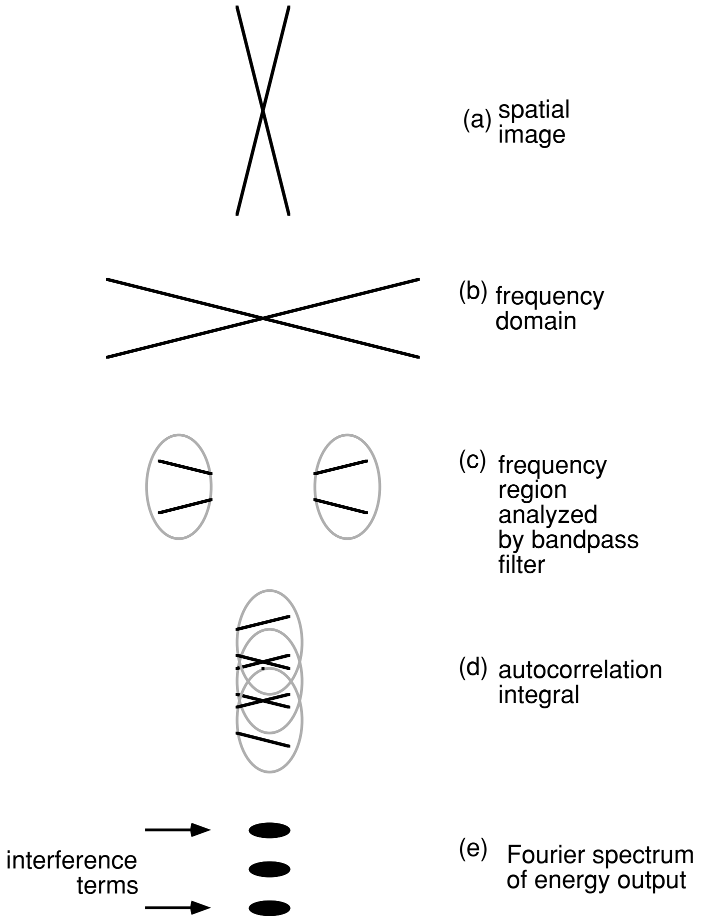

(a) spatial image
(b) frequency domain
(c) frequency region analyzed by bandpass filter
(d) autocorrelation integral
interference terms
(e) Fourier spectrum of energy output

Figure 3-7: Showing the origin of interference effects when using energy measures to analyze regions of multiple orientations. (a) Test image of two intersecting lines. (b) Fourier transform of (a). (c) Part of (b) seen by the bandpass filters. (d) Frequency spectrum of energy measure applied to image (a). This is proportional to the auto-correlation of either one of the two lobes of (b). The result has 3 dominant contributions. The middle blob at DC is the integral of the squared frequency response over the bandpass region. For this term, superposition holds, and the energy of the sum of two images (non-overlapping in the frequency domain) will be the sum of the energies of each individual image. The other two terms are interference terms, arising from interactions between the Fourier transforms of the two images. Low-pass filtering the squared energy output can remove those terms while retaining the term for which superposition holds. Note this is not the same as low-pass filtering the linear filters before taking the energy.

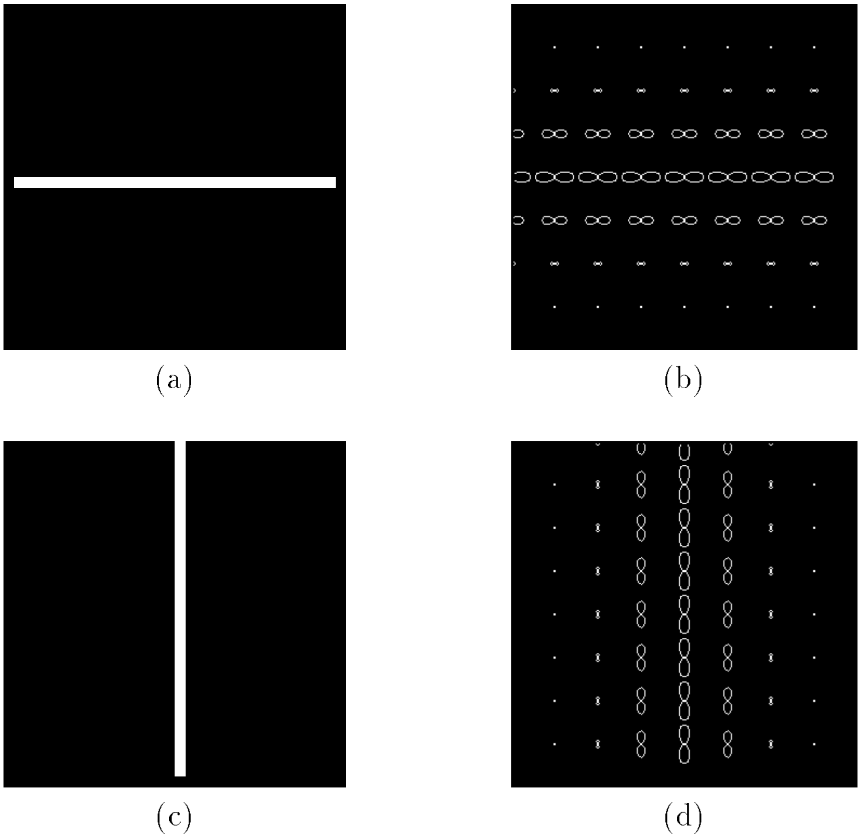
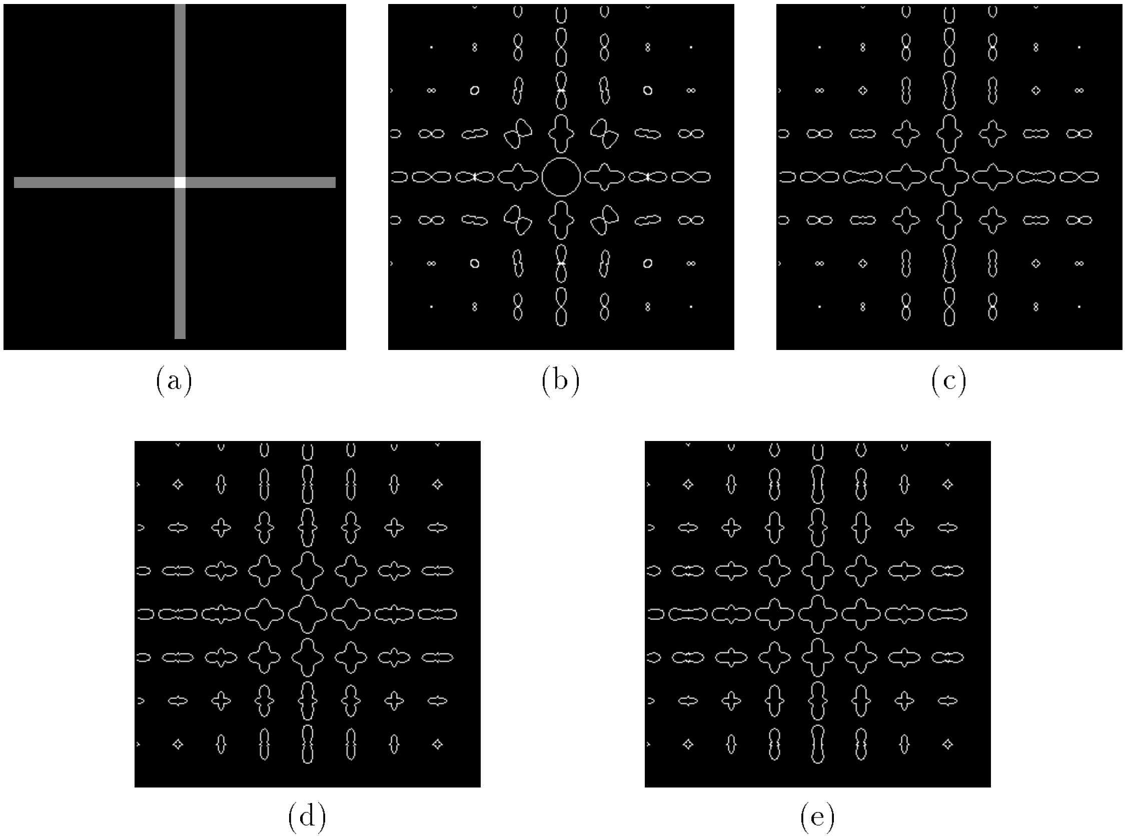


Figure 3-8: The problem with using energy measures to analyze a structure of multiple orientations, and how to solve it (part one). (a) Horizontal line and (b) floret polar plot of G_2 and H_2 quadrature pair oriented energies as a function of angle and position. The same for a vertical line are shown in (c) and (d). Continued in Fig. 3-9



(a) (b) (c)

(d) (e)

Figure 3-9: The problem with using energy measures to analyze a structure of multiple orientations, and how to solve it (part two). (a) Cross image (the sum of Fig. 3-8 (a) and (c)). The oriented energy (b) of the cross is not the sum of the energies of the horizontal and vertical lines, Fig. 3-8 (b) and (d), due to an effect analogous to optical interference. Many of the florets do not show the two orientations which are present; several show angularly uniform responses. For comparison, (c) shows the sum of energies Fig. 3-8 (b) and (d). Floret polar plot of energies after spatial blurring, (d), are predicted to remove interference effects, as described in text. Note that the energy local maxima correspond to image structure orientations. These florets are nearly identical to the sum of blurred energies of the horizontal and vertical lines, (e), showing that superposition nearly holds. (The agreement is not exact because the low-pass filter used for the blurring was not perfect).

Chapter 4

Contours and Phase

Armed with the analytical tools of the previous chapter, we can analyze local image structure. Based on quadrature pairs of oriented filters, we will develop the tools which we will use to analyze junctions. Because contours form junctions, we first study contours and their phase characteristics in this chapter.

4.1 Contour Detection – Energy Maxima

Filters with orientation tuning are often used in the detection of lines and edges [20, 43]. One feature detector that has gained popularity is Canny's edge operator [20], which is optimized to detect step edges; Canny's system can also be used with different filter choices to detect features other than step edges.

A filter that is optimized for use with an edge will give spurious responses when applied to features other than edges. For example, when the Canny edge filter is applied to a line rather than an edge, it produces two extrema in its output rather than one, and each is displaced to the side of the actual line position. On the other hand, if a filter is optimized for detecting lines, it will give spurious responses with edges. Since natural images contain a mixture of lines, edges, and other contours, it is often desirable to find a contour detector that responds appropriately to the various contour types. A linear filter cannot serve this task, but a local energy measure derived from quadrature pairs can serve it quite well. Morrone et al. [77, 76] have shown that local energy measures give peak response at points of constant phase as a function of spatial frequency, and that they correspond to the points where human observers localize contours. Perona and Malik [89] have shown that energy measures are optimal with respect to a variety of edge types. We have already described the extraction of local energy measures with quadrature pairs of steerable filters. We now wish to use steerable energy measures to generate sparse image descriptions, and to compare the results with those of a system such as Canny's.

In making this comparison we must keep in mind that Canny's full scheme involves three stages: a filtering stage, an initial decision stage, and a complex post-processing stage which cleans up the candidate edges. The filters are merely the front end to a considerable battery of post-processing machinery. Therefore to make our comparison we removed Canny's filtering stage and substituted the outputs of our steerable energy measures; we left the post-processing stages intact. We obtained Lisp code for the Canny edge detector from the MIT Artificial Intelligence Laboratory.

For the contour detector, we use the G_2 and H_2 quadrature steerable basis set. We first find at every position the angle of dominant orientation, θ_d, by the angle of maximum response of the steerable quadrature pair, as described in Section 3.1. We then find the squared magnitude of the quadrature pair filter response, steered everywhere in the direction of dominant orientation, $E_2(\theta_d) = [G_2^{\theta_d}]^2 + [H_2^{\theta_d}]^2$. A given point, $(x_0,\ y_0)$, is a potential contour point if $E_2(\theta_d)$ is at a local maximum in the direction perpendicular to the local orientation, θ_d. (Another approach, described by Perona and Malik [89] and which we will use in Chapter 6, is to mark as contour points those points which have maximal energy response with respect to both orientation and position).

The local maxima points are then thresholded with hysteresis as in the Canny method, using the values of $E_2(\theta_d)$ as the basis of thresholding, instead of the gradient magnitude.

Figure 4-1 (a) shows a test image consisting of a filled circle and an open square. The response of the Canny edge detector is shown in Fig. 4-1 (b). It correctly finds the edges of the circle, but signals double edges on either side of the lines defining the square. Figure 4-1 (c) shows the output using the steerable quadrature pair. The new detector responds with a single value correctly centered on both the circle and the square, giving a cleaner, sparser description of the same information.

Because the responses of G_2 and H_2 indicate the local phase, we can use them to further classify contours as edges, dark lines, or light lines. Steering G_2 and H_2 along the dominant orientation gives the phase, φ, of contour points:

$$\varphi = \arg[G_2^{\theta_d}, H_2^{\theta_d}]. \tag{4.1}$$

To preferentially pick-out lines or edges, we scaled the energy magnitude, $E_2(\theta_d)$ by a phase preference factor, $\Lambda(\varphi)$,

$$\Lambda(\varphi) = \begin{cases} \cos^2(\varphi - \varphi_\circ) & \text{if } \frac{-\pi}{2} \leq \varphi - \varphi_\circ \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases}, \tag{4.2}$$

where

$$\varphi_\circ = \begin{cases} 0 & \text{for dark lines} \\ \pi & \text{for light lines} \\ \pm\frac{\pi}{2} & \text{for edges} \end{cases}. \tag{4.3}$$

The thresholding stage proceeds as before. Figure 4-1 shows the result of such processing, selecting for dark lines, (d), and edges, (e). (The blobs on the square are due to multiple orientations at a single point, and could be removed by a post-processing thinning operator.)

4.2 Phase at Energy Maxima

It is often asserted that important image contours are edges. It is natural to ask what the distribution of phases along image contours in natural scenes. Is it really biased towards edges; is it uniformly distributed over all phases? The answer might affect the approach a visual system should use for a variety of tasks.

We can plot a histogram of energy strength and local phase along contours, which we call a *phase-energy histogram.* We steer a quadrature pair of filters (G_2 and H_2) along the dominant orientation everywhere in the image and measure the energy response. We make use of the fact that energy measures response maximally at contours and find the positions of maximal energy response with respect to displacement perpendicular to the dominant orientation. We then add one count to the histogram for each such locally maximal energy and the local phase at that point. In order to avoid shifts in phase due to pixel sampling positions not lying exactly on the energy local maximum, we oversample both the image and the filters by a factor of 8. There is an ambiguity in the dominant orientation vector; a vector in the opposite direction describes the orientation equally well. This introduces an ambiguity in the sign of the phase, since reversing the orientation changes the sign of the odd filter response and hence of the phase. We plot every phase-energy point twice in the histogram, once for each sign of the phase.

Figure 4-2 illustrates the histogram coordinates. Phase angle increases counterclockwise, starting from zero at "three-o'clock". We plot negative phases from three-o'clock to nine-o'clock. Energy increases radially from the center of the clock.

Now let us plot actual phase-energy histograms for some test images (all were 64 x 64 pixels). Figure 4-3 (a) is composed only of black lines, which we define to be 0° phase angle. Since all the lines in the test image are of the same contrast, the phase-energy histogram, (d), shows a single peak at 0° phase angle. Figure 4-3 (b) is the same set of contours, rendered with equal-contrast white lines. As expected, the histogram, (e), shows a single peak at 180°. Figure 4-3 (c) again shows the same contours, rendered as edges. Now the histogram, (f), shows several peak responses at ±90° because there are several edge contrasts. For all three cases, the phase-energy histogram accurately characterizes the contour characteristics of the test images.

From the results shown in Fig. 4-3 we see that the phase-energy histograms measure what we want them to. We now examine the phase and magnitude distributions of contours in some natural images. Figure 4-4 shows an image at several scales of analysis, and the corresponding phase-energy histograms. At a fine scale, the image contours are predominantly edges. At coarser scales, however, contours of other phases become more pronounced. Structures which had been identified as two edges can become a single line-phase contour. For this example, an edge model does not hold over all scales.

Even the boundary of a physical object can appear at many different phases. Figure 4-5 shows an example. The detail of image (a) containing the hat has mostly edge contours, as shown in the phase-energy histogram, (b). Yet, at sample points spaced evenly along the energy peak of the contour of the hat, (c), one finds a variety of phases. As shown by a phase-energy plot of those points (d), the hat begins at the lower left as a white line contour, then becomes an edge, then a white line again, a low-contrast contour, and finally an edge again. An edge-based analysis of this contour would mis-locate the boundary, and mark parts of it twice.

4.2.1 Simple Model

We can use a simple image model to gain intuition about why we observed a statistical bias toward contours of edge phase. Our image model will be isolated rectangles of different shades of grey and sizes on a white background (see Fig. 4-6). This roughly models a textureless world filled with objects of all sizes. To simplify the analysis, we will consider the problem in one dimension.

Let us first assume that the rectangles have a constant contrast against the background. Fig. 4-7 (a) shows a test image of such rectangles over a range of sizes. We will analyze this image world with a quadrature pair of filters.

Figure 4-7 (b) illustrates the three size regimes over which the phase at energy peak will have characteristic behaviors. For the wide bars at the right, the quadrature pair will find two edges, of uniform energy for all the bars. For bars of sizes near that of the filters themselves, the phase at peak energy will be intermediate between lines and edges. Bars at the left will have line phase, *but the energy at that phase will get smaller and smaller as the bar of constant contrast becomes narrower and narrower.* The energy response to the test image for the G_2 and H_2 quadrature pair has this behavior, as shown in Fig. 4-7 (c). Thus, there will be many measurements at the maximal energy at edge phase, few measurements of phase intermediate between line and edge, and many measurements at line phase, but at very small energies. The distribution of contour phases will be biased toward high contrast edge-phase contours. Figure 4-7 (d) shows a plot of the phase as a function of position. Figure 4-8 is a polar plot of the phase and energies at positions of energy local maxima of Fig. 4-7 (a). The dot at exactly edge phase is actually 19 measurements superimposed. Thus the edge phase structure dominates the phase-energy histogram of this simple test image. For rectangles of a range of contrasts against the background, this phase-energy histogram would simply scale radially (in energy magnitude). The result would be a distribution similar to what we observe in Figures 4-4 (f) and (g) and 4-5 (b).

4.3 Summary of Analysis Tools

We have developed useful tools for image analysis. Steerable filters offer computational efficiency, and give an analytic formula for filter response as a function of angle. The analytic formula is useful for further analysis, for example, to calculate the dominant orientation.

By studying the frequency domain characteristics of these energy measures, we found an efficient way to use them to analyze multiple orientations, which will be useful for junction analysis. We designed a contour detector based on local energy measures which marks both lines and edges with a single response and can be used to further categorize the contours as either dark lines, light lines, or edges. Finally, we studied the local phase characteristics of images along the dominant orientation at energy maxima. Our findings show that a simple edge model is not adequate to describe image contours, and validate our energy-based approach.

Now that we can efficiently apply oriented filters, and analyze orientation, contours, and phase, it is time to analyze junctions, which provide visual cues to interpret images.

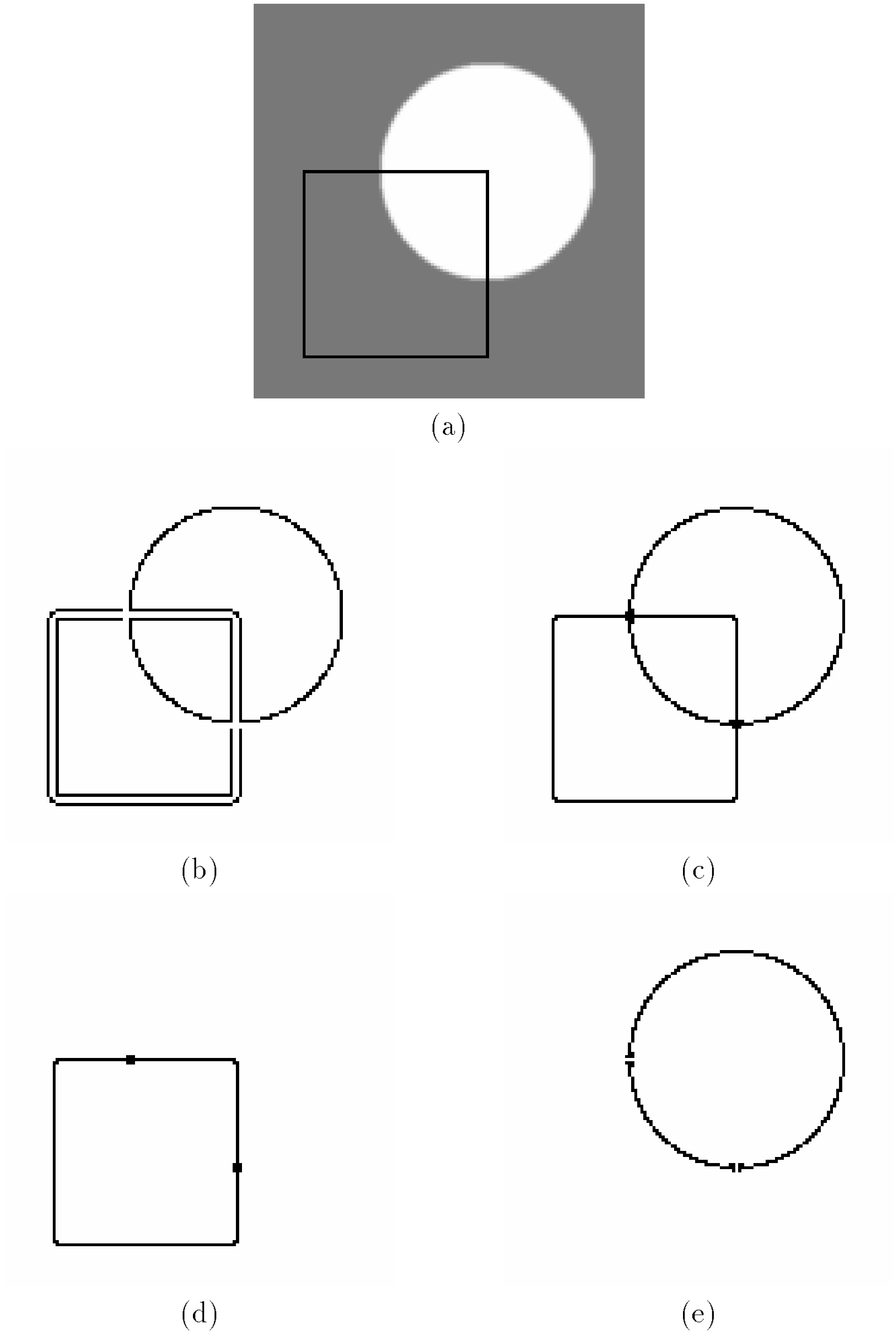


Figure 4-1: (a) Circle and square test image. (b) Output of Canny edge detector. The edges of the circle are accurately tracked, but the lines of the square are marked as two edges, neither at the correct position. (c) Output of steerable filter contour detector. Both edges and lines are marked as single contours, centered on the image feature. (d) Dark lines found by combining the contour detector with a phase estimator. (e) Edges found by combining the contour detector with a phase estimator.

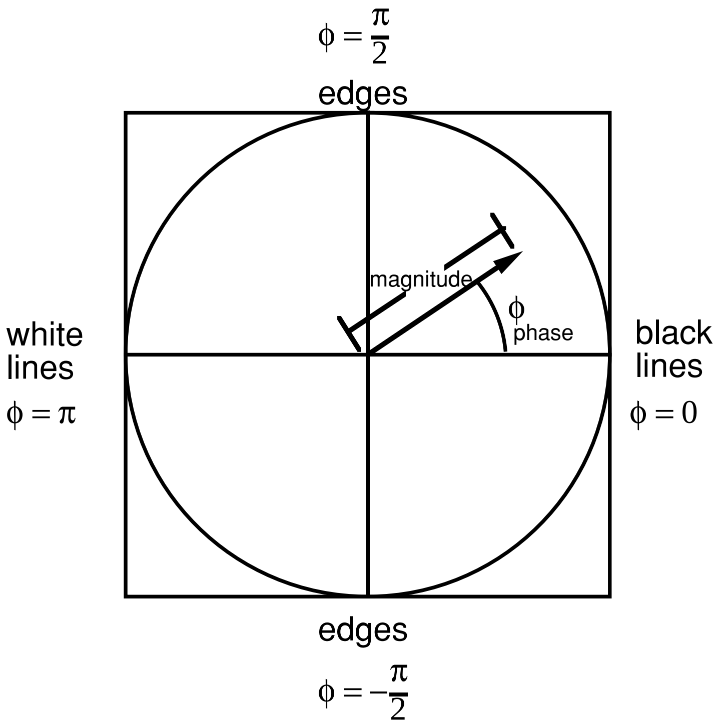


Figure 4-2: Explanation of phase-energy histogram intensities. In the image, a quadrature pair is steered along the locally dominant orientation of the image. From the quadrature pair outputs, magnitude and phase are measured positions of maximal energy response relative to displacements perpendicular to the dominant orientation. The intensity of the phase-energy histogram at a point is proportional to the number of measurements in the image at phase and peak magnitudes near that point.

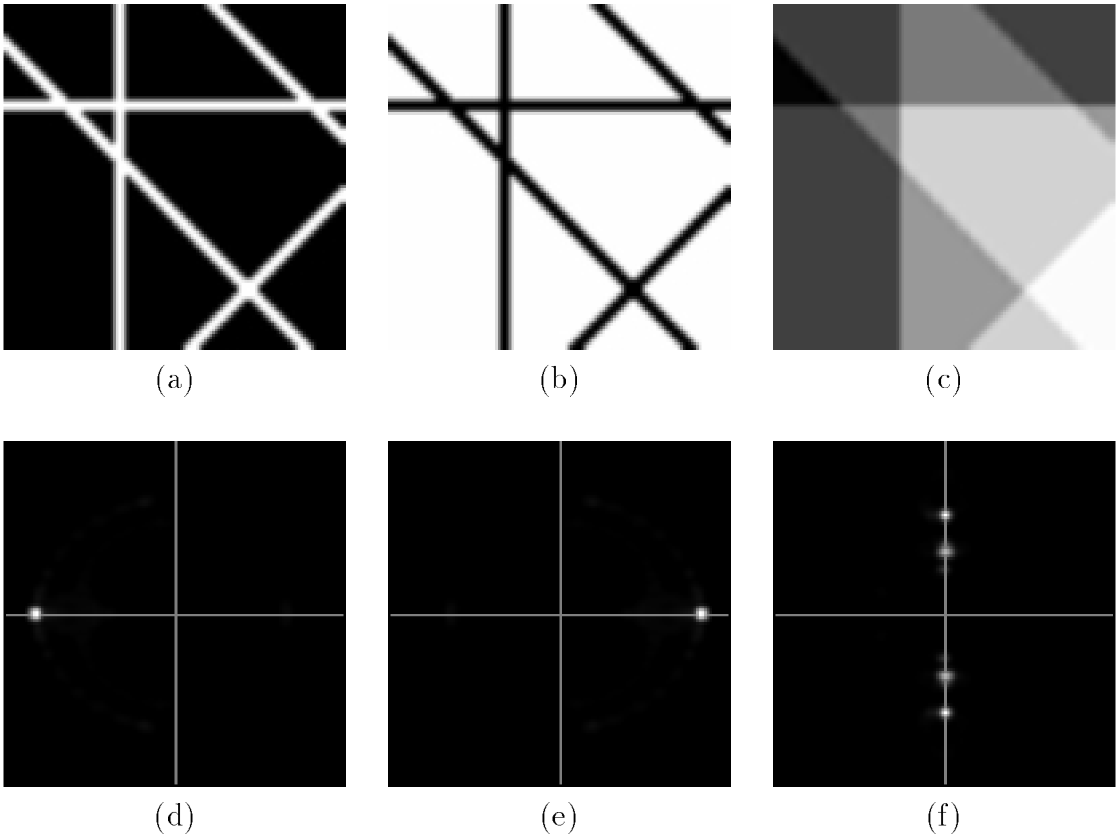


Figure 4-3: Test figures and their corresponding phase-energy histograms. (a), (b), (c) show the same configuration of contours rendered with contours of different phases–black lines, white lines, and edges, respectively. (d), (e), (f) show the corresponding phase-energy histograms. The single dot in histogram (d) indicates contributions from a single contrast of white-line phase. The dot in histogram (e) indicates contributions from a single contrast of black-line phase. The edges in (c) are of more than one contrast, shown by the multiple dots at edge phase. This plot is symmetric about the horizontal axis because white-to-black edges are indistinguishable from black-to-white edges. The phase-energy histograms correctly characterize the phase distributions along the contours of the test images.

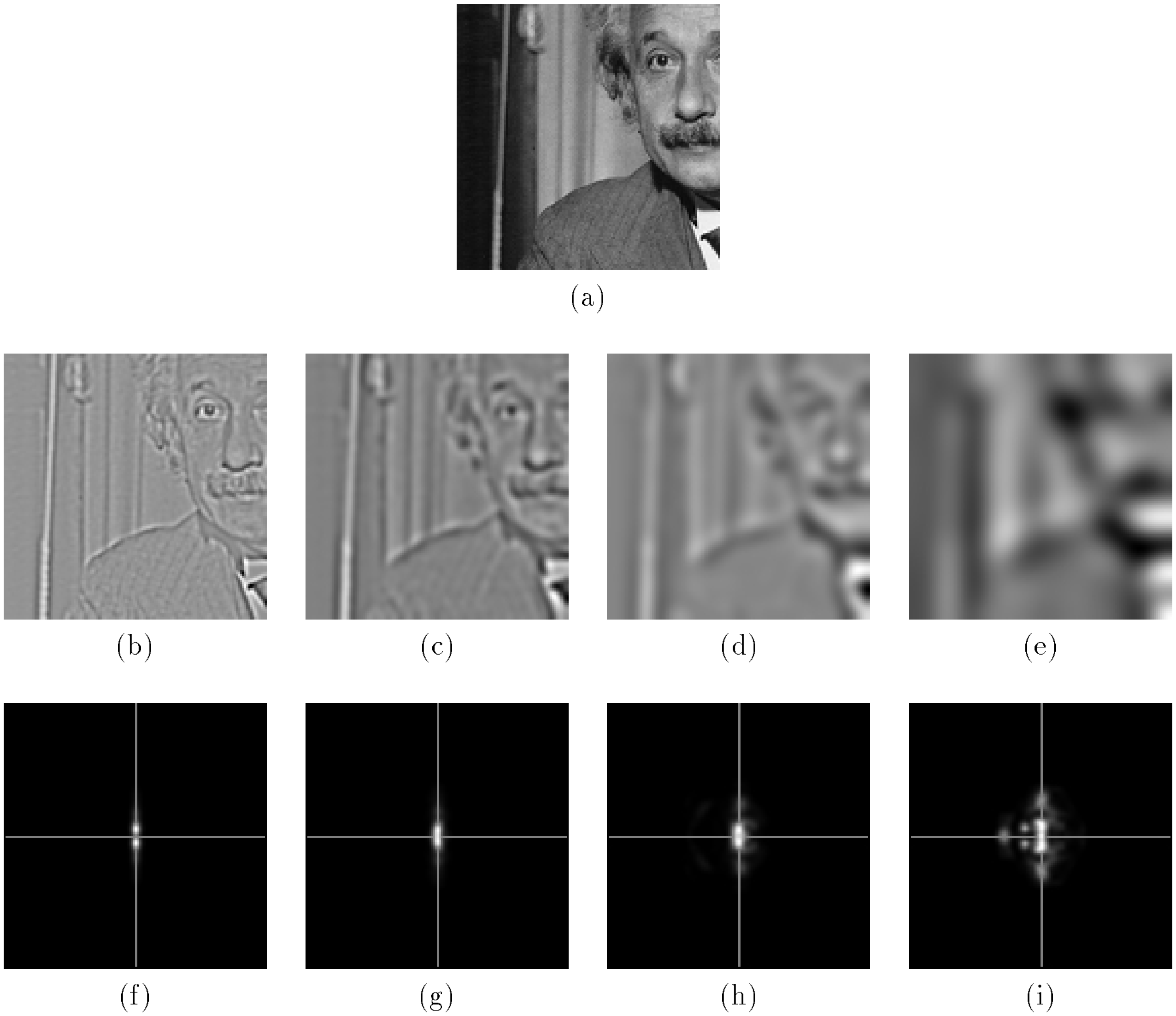
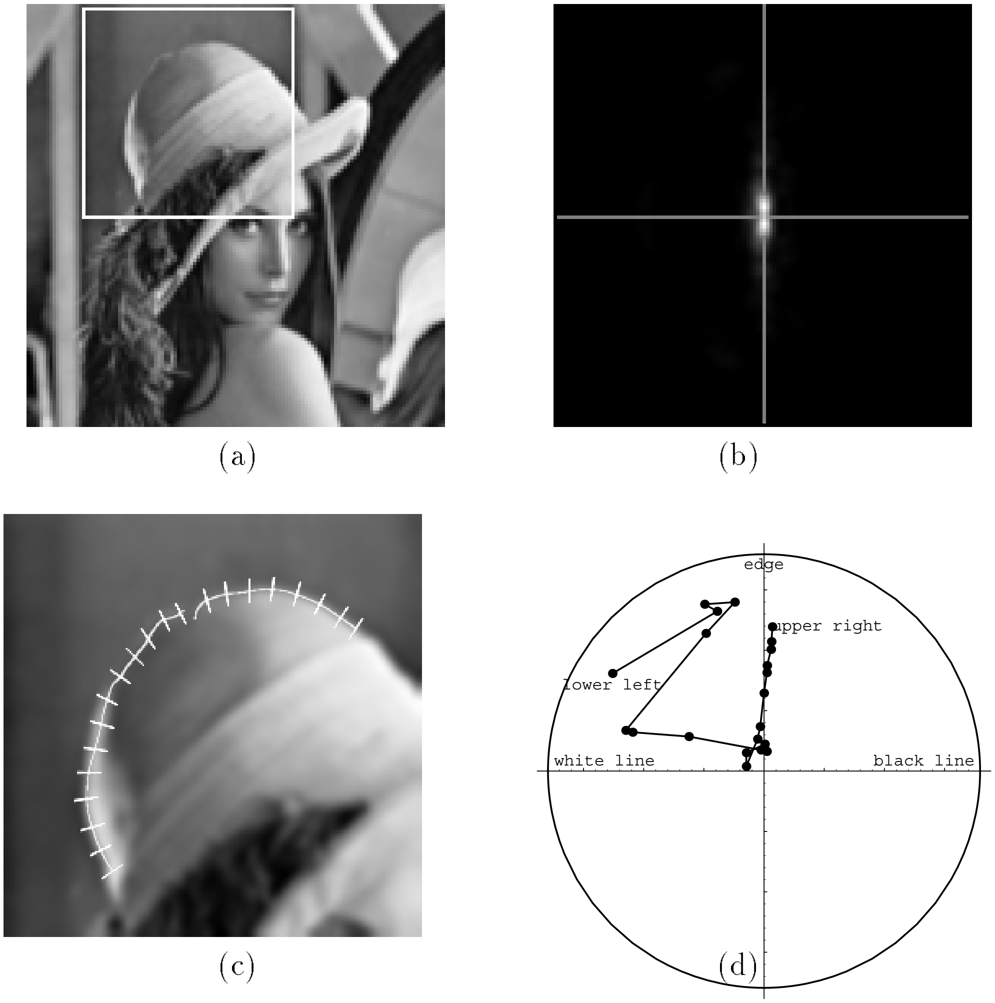


Figure 4-4: Effect of variation in scale on phase-energy histograms. (a) Test image, section of portrait of Einstein. (b) – (e) Test image bandpassed by four different scales of filters. (f) – (i) Corresponding phase-energy histograms. Notice that at scale (b), this image happens to be dominated by edge-phase contours, as seen in (f). At the coarser scales of analysis, many of the contours become of line or intermediate phase, as shown by the histograms (h) and (i).



(a)

(b)

(c)

(d)

Figure 4-5: (a) Image of Lenna, showing region of detail analyzed in phase-energy histogram (b). At the scale of analysis, the image is predominantly edge contours. However, while the statistical properties are dominated by edges, important image features can contain contours of all phases. Phase and magnitude measurements were taken along the energy local maximum which defines the contour of the hat, shown in (c). Measurements were taken every 4 pixels. (d) shows the results. The beginning position of the line in the phase-energy plot shows that the contour of the lower left hand corner of the hat is a strong white line. Then, following the data in the phase-energy plot, we see that the hat contour becomes edge-like, line-like, low-contrast line and edge, and finally edge-like, in agreement with the appearance of the contour in (c). A simple edge model does not fit this contour.

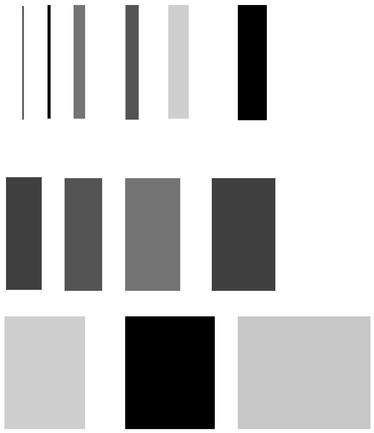


Figure 4-6: Schematic illustration of image model used to analyze expected phase-energy histogram characteristics. We assume an image consists of rectangles of a wide range of sizes and contrasts.

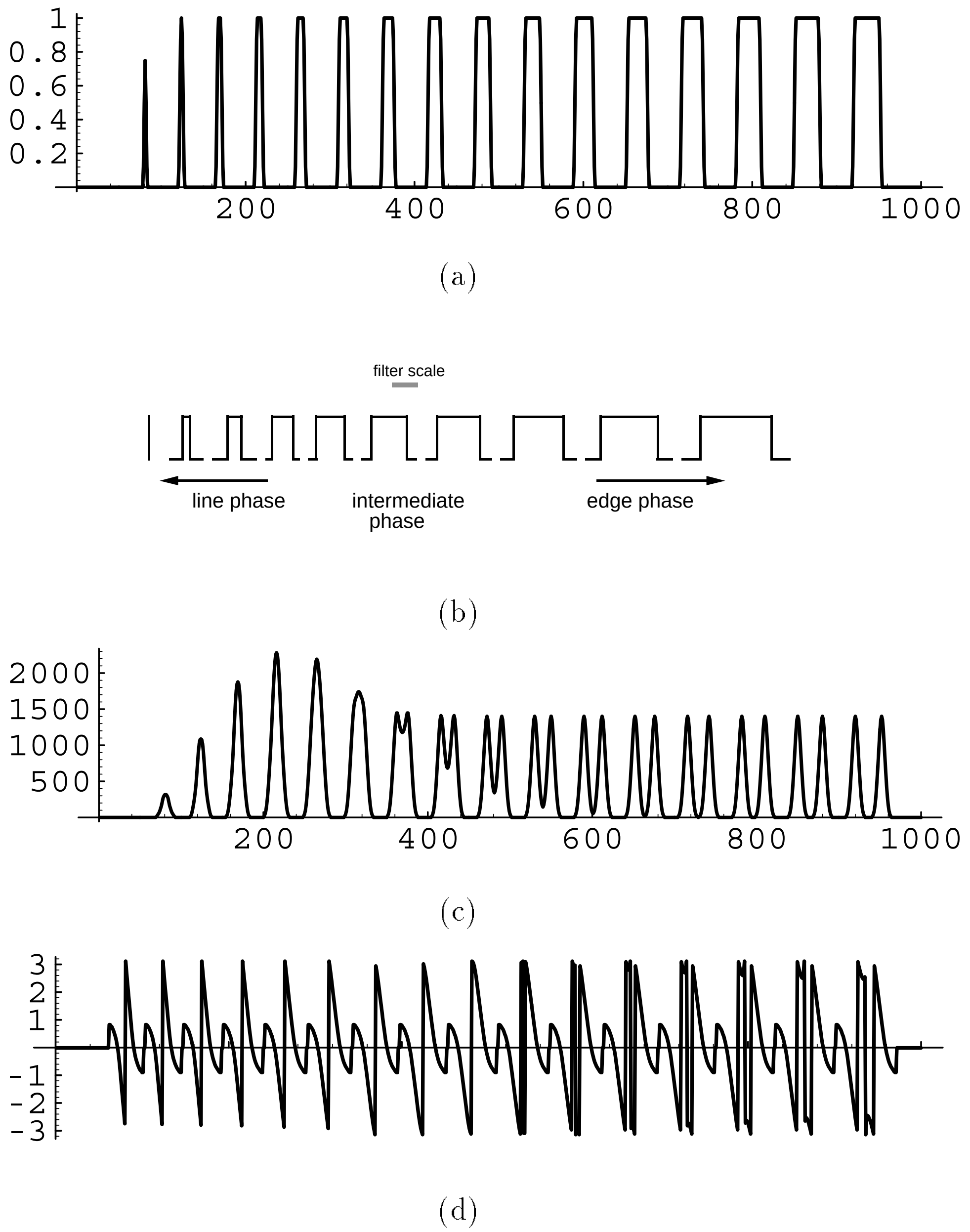


Figure 4-7: Plots showing relationship of energy and phase for a simple image model. (a) Image model consists of rectangular pulses of many widths. (To remove spatial sampling effects, the series of constant amplitude pulses were zoomed and blurred to the resolution shown, which slightly blurs the pulse edges and attenuates the far left pulse). We applied the G_2, H_2 quadrature pair of filters to (a). (b) illustrates the three size regimes of the pulses. (c) Output of energy measure applied to (a). Note that for pulses wider than a certain width, the maximum energy at their edge stays constant. As pulses become narrower, however, the peak energy decreases. This causes a bias in the phase-energy histogram—strong edges are more likely to occur frequently than strong lines are. (d) Phase of (a), measured by the quadrature pair. See also Fig. 4-8.

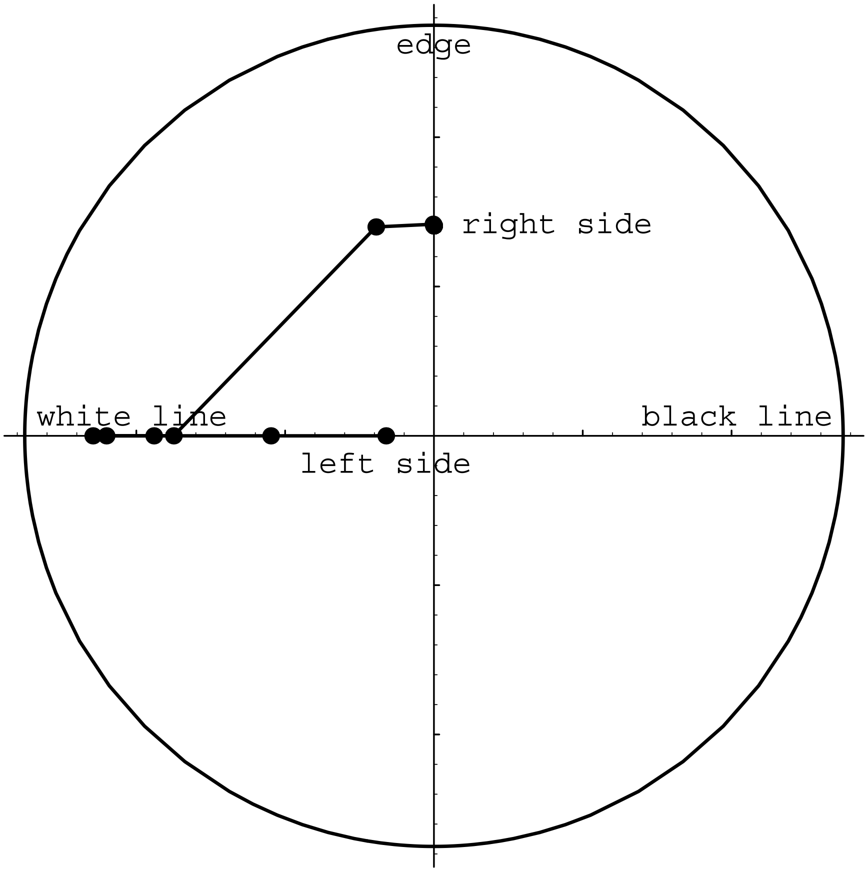


Figure 4-8: Polar plot of the energy and phase at positions of energy local maxima for the test image of Fig. 4-7. The data points corresponding to the far right and left sides of the test image are labeled. There are actually 19 data points superimposed on the exact same dot at edge phase ("right side"), illustrating the bias toward strong edges in the simple image model of Fig. 4-7. This bias is observed in real images.

Chapter 5

Cue Detection I

In the next two chapters, we will use our image analysis tools to analyze local visual cues for scene interpretation. Before proceeding, we briefly review related approaches to the problem of using junction information to interpret scenes.

5.1 Related Work

5.1.1 Blocks World

Vision researchers studying the *blocks world* developed important methods for using local information to interpret scene structure. The blocks world restricts scene objects to be polyhedral blocks. (See [23] for a review).

Guzman [42] made use of vertices and junctions to recognize 3-dimensional objects. He developed heuristics for grouping the elements of a line drawing into objects. Huffman [51] and Clowes [22] systematically labelled each line as corresponding to either a concave edge, a convex edge, or an occluding edge. Only certain labellings are self-consistent at intersections. The researchers made exhaustive searches to find self-consistent line drawing interpretations. Waltz [108] added more possibilities for line interpretations. An exhaustive search for the self-consistent line labellings would have been infeasible. Instead, he compared local junctions and pruned out locally inconsistent labellings, continuing that process until all junctions had been labelled. His system was able to successfully interpret many blocks word scenes.

Recently, Sinha, Adelson and Pentland have identified shading and reflectance in the blocks world, using an approach related to that of Waltz [104, 5]. Figure 5-1 shows a "ψ" junction, which the authors exploit as a visual cue. If two of the line segments are collinear, and the other two are not, then the collinear segments are labelled as a bend, and the other two segments are labelled as reflectance change. (This cue has also been discussed by [14, 97]). In Chapter 6, we will generalize that visual cue for images which are not pre-labelled into line segments and junctions, and we will remove the restriction that the two of the line segments be collinear.

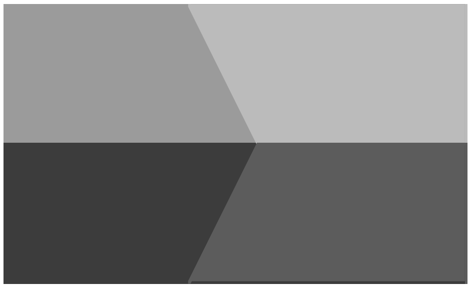


Figure 5-1: Shading cue exploited by various researchers [104, 14, 97]. If four line segments meet at a common point, and two are parallel, then those two segments are presumed to result from a surface bend.

5.1.2 Vision Modules

It is common to hypothesize modules specific to particular visual tasks, the outputs of which are integrated at higher processing levels [8, 10, 28, 72, 90, 105]. Such integration is needed to analyze images such as Fig. 1-1. Both Adelson [1] and Knill and Kersten [56, 55] have described the importance of contours and other contextual information on the perception of lightness and transparency. They demonstrate this with illusions which simultaneous contrast cannot explain. Bulthoff and Mallot [18] have studied what one perceives when different cues for depth conflict. Aloimonos and Shulman [8] discuss how to combine shading with motion information, texture with motion, and other combinations.

5.1.3 Perception Literature

The perception literature asks how humans infer scenes from images. Rock [93] and Hochberg [49] have pointed out that local cues in the image give information about depth and reflectance. We will look for T and ψ junctions, which give evidence for their cues of *interposition* and *shading*.

The *grouping principles* of the Gestalt psychologists (reviewed in [49, 93]) address how to use local cues to choose an interpretation for a set of objects. We will exploit two of their grouping principles, *good continuation* and *proximity*, in the curve finding algorithm we use in Chapter 6.

5.2 Cue Detection with Local, Oriented Filters

As many researchers have observed, junctions are important cues for scene interpretation. They can be perceptual cues for shading, occlusion and transparency. Figure 1-1 illustrates this; the three figures differ only in their junctions, yet give three very different physical interpretations. Figure 5-2 also shows this: the T-junctions inside the boxed region show that the hand goes behind the head.

Given this importance, we might expect biological visual systems to have low-level mechanisms which detect and characterize junctions. We therefore sought to build simple junction detectors using biologically plausible low-level computational machinery [34]. Similar approaches to what we present below were developed independently by Heitger *et. al.* [48] and Perona [88].

The types of junctions we seek to detect in this chapter are L-junctions (corners), X-junctions (crosses), and T-junctions. Figure 5-3 shows some examples of these junctions. X-junctions can indicate transparency. T-junctions can indicate occlusion. (Researchers often assume that side of the stem of the T indicates the “far” side of the occluding contour (e.g., [81, 13, 69]). From observing real images, we feel that as likely as not the stem of the T can occur from a marking or feature on the

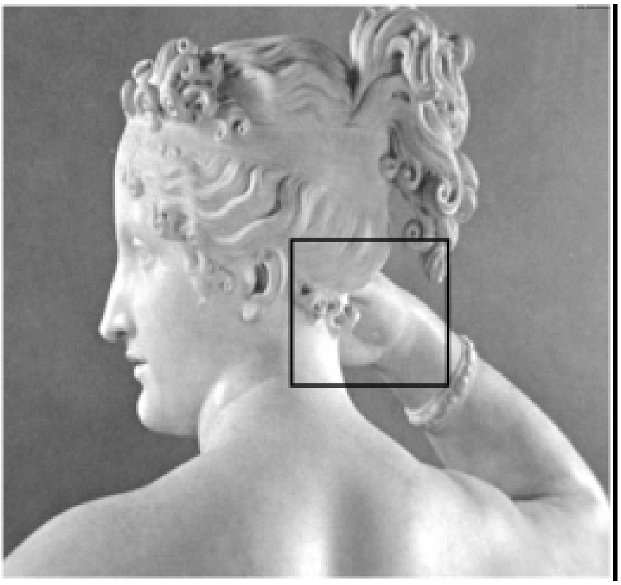


Figure 5-2: Illustration of the usefulness of junctions in scene interpretation. The T-junctions in the boxed region indicate that the hand goes behind the head.

front surface which terminates at limb of the occluding contour. Figure 5-4 shows an example. Therefore, we attach no foreground or background designation to our T-junctions or their contours.)

We want to detect junctions independently of the phase of their contours, since, as we learned in Chapter 4, image contours can come in many different phases. We say that, for example, an X-junction occurs when a line crosses a line, or an edge crosses another edge, or a line crosses an edge, and analogously for the other junction types. Figure 5-3 illustrates prototypes of the different junctions.

Our approach to analyzing junctions is to first use energy measures to analyze orientation strength. We then apply spatial derivatives of the energy measures to study the local image structure. Figure 5-5 shows a T-junction and floret polar plots of the oriented energy as a function of angle and position. Each of the two contours of the T-junction is marked by the position of the local maximum, taken against the contour, of the energy oriented along the contour. In addition, the stem of the T stops at the position of the contour of the bar of the T. The region where the stem stops, at the intersection of the bar and stem contours, defines the T-junction region.

The first step in our procedure is to apply a basis set of steerable quadrature pair of oriented filters; we used the G_2, H_2 pair (see Fig. 5-6). From these filter responses, we can calculate the oriented energy as a function of angle for all angles and positions.

We spatially blurred these responses to remove interference effects, as described in Section 3.2.2. We found the two dominant orientations, which we define to be the angles of the two largest local maxima of oriented energy as a function of angle. (To analyze orientation by searching for local maxima, we found better results using the more tightly tuned G_4, H_4 filters). We assume that the two contours of the junction, if present, are oriented along the two dominant orientations.

With the knowledge of these two dominant local orientations, we are ready to take derivatives of the energy along and against these directions to detect regions which have the expected local structure of a junction. First, however, we must apply an important gain control step.

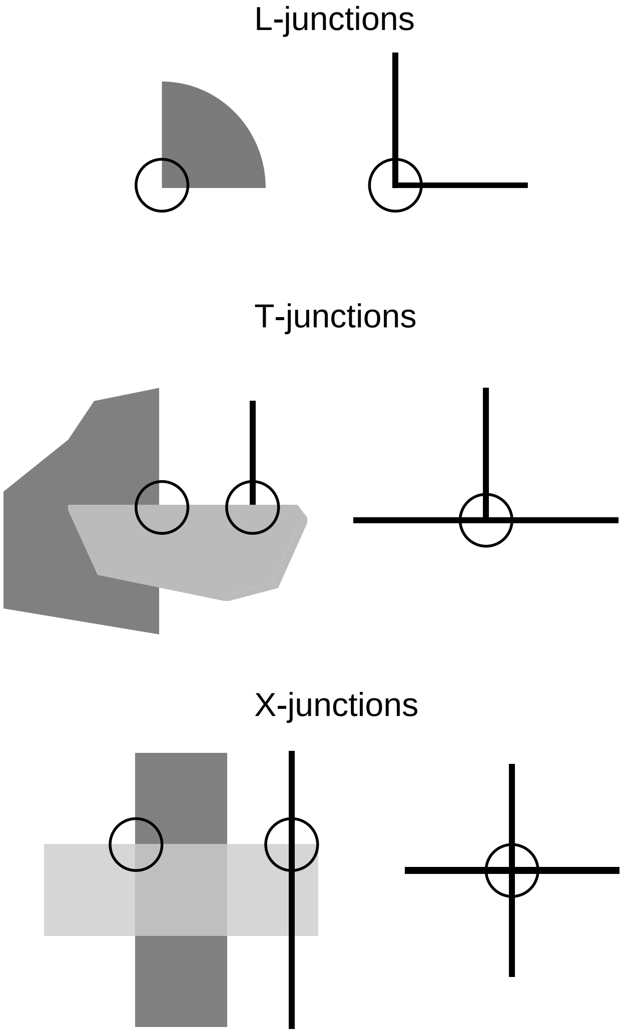


Figure 5-3: Examples of junctions which we would like to classify for image analysis. L, T, and X-junctions in contours of both line and edge phases are circled.

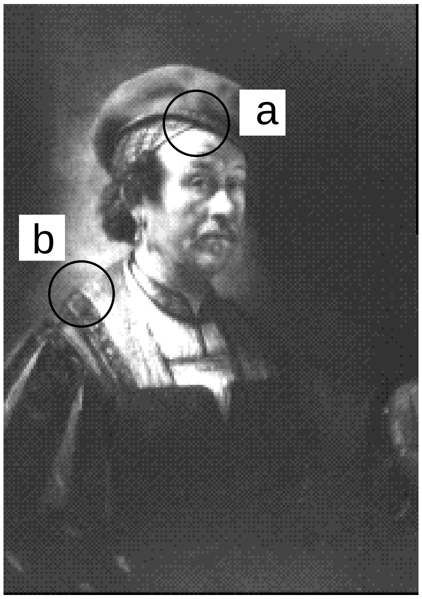


Figure 5-4: Rembrandt self-protrait which illustrates why T-junctions do not indicate the ordering of occluding layers. The T-junction at (a) occurs with the "stem" of the T corresponding to the occluded layer. (The stem is the vertical bar in the capital letter, "T". The horizontal cross stroke is the "bar" of the T.) Many researchers assume the layers follow this relation. However, the T-junction at (b) shows another common configuration. A marking on the occluding layer causes the stem of the T to lie on the occluding layer. Because of this ambiguity, we will not assign a depth ordering to the surfaces on each side of the bar of a T-junction.

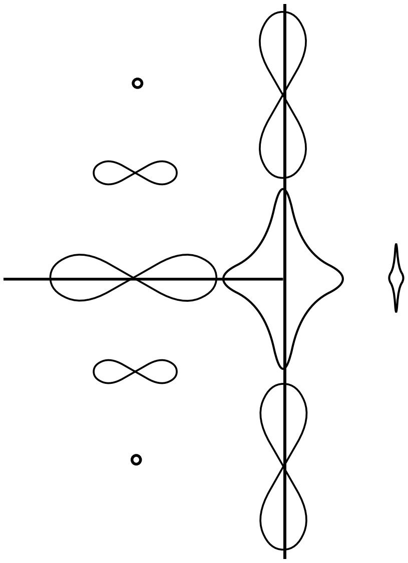


Figure 5-5: Local energy measures can be used to identify T-junctions. Floret polar plots of oriented energy as a function of orientation are shown for various positions near a T-junction. This plot illustrates the local energy characteristics which we require for a T-junction: the energy perpendicular to the two dominant orientations must be at a local maximum; and the energy along the dominant orientations must show end-stopping for exactly one of the two orientations.

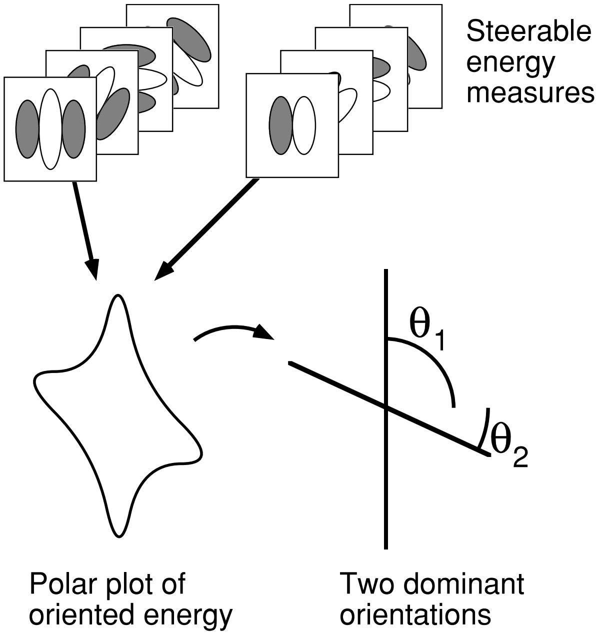


Figure 5-6: Initial processing in junction detection. A bank of steerable quadrature pair filters measures oriented energy as a function of angle. The two dominant orientations are defined to be the angles corresponding to the two strongest energy local maxima.

5.2.1 Gain Control

At an occlusion T-junction, such as that of Fig. 5-7 (a), there can be a strong change in oriented energy due to a change of reflectance in the materials behind the occluding edge. This is illustrated in Fig. 5-7 (b), which is the horizontal energy of (a). The higher contrast of the black to white transition over the grey to white transition causes a sharp change in the horizontal energy at the junction. This change in energy is hard to distinguish from the end of a contour. We would like to have our measurement for the horizontal strength of the continuous contour of the T be constant throughout the T-junction. We need a contrast normalization step.

Local contrast normalization models have been used to account for physiological and psychophysical data on the low-level perception of moving or static images [46, 102, 111]. Typically, in such models, the response of a linear filter is normalized by the sum of the energies measured in filters in a local neighborhood over all orientations. Such normalization treats regions of varying contrasts equivalently and allows image information to be represented within a small dynamic range.

This normalization works well for most contours, but causes problems at junctions. Many junctions of interest, such as Fig. 5-7 (a), have important contours of widely differing contrasts. These contour segments with differing contrasts can be of different orientations, or, as shown in Fig. 5-7 (a) they can have the same orientation, aligned along a single contour. A normalization by the sum of filter response energies at all orientations over a local spatial region would cause the strong contour to overwhelm the weak one, shown in Fig. 5-7 (c). Even normalization by filter response energies from a single orientation, summed over a local region, gives a similar result, shown in Fig. 5-7 (d).

A solution to this problem is to normalize by the filter response energies from a single orientation, but summed over a particular local region. If we average only in the direction perpendicular to the direction of the oriented energy filters, then the strong contour segment will not obliterate the weak one at a junction where a contour undergoes a strong to weak contrast transition. The normalized energy, E_n, is the raw energy, E, normalized by E blurred perpendicularly to the contour direction, $\bar{E}^{\perp}$:

$$E_n = \frac{E}{(\bar{E}^{\perp} + N)}, \tag{5.1}$$

where N is a small constant which eliminates division by zero in the presence of noise.

Figure 5-7 (e) shows the resulting normalized energy, showing smooth continuation along the contour throughout the junction.

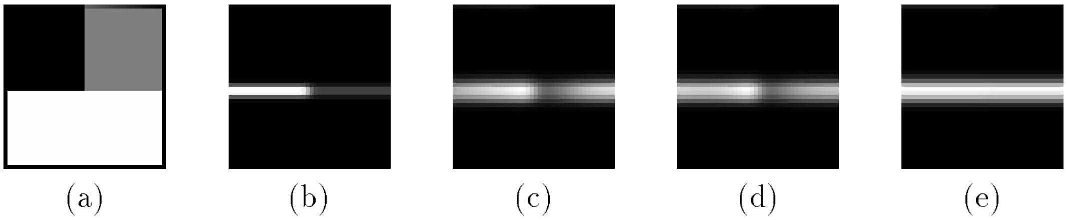


Figure 5-7: Calculation of gain-controlled oriented energies. (a) Example image showing a contour with a high to low contrast transition. Horizontal oriented energy (b) decreases dramatically over the junction. This decrease is difficult to distinguish from the termination of a contour. Some local gain control is needed to make it clear that the contour continues. A common gain control procedure is to normalize by the energy blurred over a local spatial region and over all orientations. This fails to show the continuation of the contour at the junction; the high-contrast segment overwhelms the low-contrast segment (c). Normalization by the energy of a single orientation blurred over a spatially isotropic region also fails at the junction, (d). Instead, we normalize by the energy at a single orientation blurred only perpendicularly to the filter orientation. Then one part of a contour does not influence another part of the same contour, and the gain normalized response is uniform along the contour, even at the T-junction, as shown in (e).

5.2.2 Junction Detection and Classification

Now that we have the normalized oriented energies, we want to compare the spatial structure of theses energies with those of prototype junctions, such as the junctions shown in Fig. 5-3. The contours of the junction may meet at any angle, so we do not require that the energy profiles agree exactly everywhere, since that would require a different junction prototype for every possible angle between the contours. Instead, we make comparisons of slices through the energy taken relative to the measured two dominant orientations at the junction.

A junction occurs where two contours meet. To determine whether there is a contour along each of the dominant orientation directions, we use the approach developed in Section 4.1—we look for the position perpendicular to the contour where the response oriented along the contour is maximal. Fig. 5-8 (a) shows the prototypical energy response as a function of position perpendicular to the contour.

To determine whether or not a contour stops at the junction, we examine the normalized energy oriented along the direction of the contour as a function of position along the contour. The prototype function for that energy tapers from a high, constant value down to zero, as illustrated in Fig. 5-8 (b).

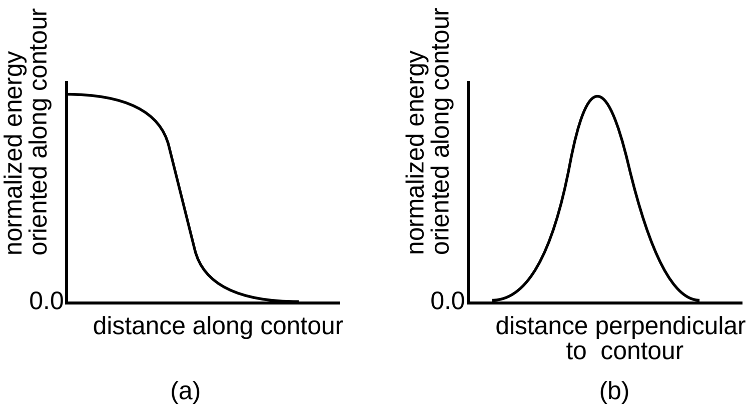


Figure 5-8: Template slices of normalized oriented energy as a function of position. We compare the actual energy measurements against these prototypes in order to determine whether the local region describes a junction and to identify what kind. (a) Prototypical response for energy oriented along the stem of a T-junction as a function of distance along the stem. (The stem is the contour which terminates at the junction). The response is constant away from the junction, then falls to zero as the contour ends. This "stopping" response can also indicate a contour at a corner. (b) Prototypical response of energy oriented along a contour as a function of distance perpendicular to the contour. Utilizing the local definition of a contour described in Section 4.1, this locally maximal response indicates the presence of a contour. If the oriented energy along both dominant orientations indicate a contour, then we say we are at a junction.

The main idea, then, is to find the slices of the normalized oriented energy at the junction and compare those functions with the prototypes of Fig. 5-8. We only want to compare the functions over a local region. A reasonable difference measure is the integral under a local Gaussian window of the squared difference between the prototype functions and the corresponding slices of the actual normalized energy.

Locally, the prototype functions are smooth and relatively slowly varying. Windowed by a Gaussian function, they may be well approximated by a second order polynomial times the windowing Gaussian. Using this approximation, we can find the desired squared difference from the prototype functions by using derivatives of the normalized energies along and against the two locally dominant orientations. It is possible to find this integrated squared difference with an additional level of the local filtering operations which we have been using so far.

Let $I(x)$ be the normalized oriented energy along some direction as a function of x, the distance either along or against a contour orientation. This is the slice through the normalized energy which corresponds to the slices which generated the prototype functions of Fig. 5-8. First, we want to find the coefficients of a second order polynomial expansion of the function f_1 which is $I(x)$ blurred by a Gaussian filter, G_0 (0th derivative of the Gaussian):

$$f_1 = G_0 * I(x) = a_1 x^2 + b_1 x + c_1. \tag{5.2}$$

Differentiating the above equation and evaluating the result at $x = 0$ gives polynomial coefficients as a function of derivative of Gaussian filter outputs:

$$\begin{aligned} a_1 &= \frac{1}{2}(G_2 * I(x))|_{x=0} && (5.3)\\ b_1 &= (G_1 * I(x))|_{x=0} && (5.4)\\ c_1 &= (G_0 * I(x))|_{x=0}. && (5.5) \end{aligned}$$

We want to find the squared deviation, E, of the function f_1 from a prototype function, f_2 (as in Fig. 5-8), over a local region. We introduce a second Gaussian, of standard deviation σ, to define the local region. Then the integral of the squared difference, windowed by the second Gaussian, gives the desired squared deviation:

$$E = \int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} (f_1 - f_2)^2 \mathrm{dx} \tag{5.6}$$

Writing out f_1 and f_2 as polynomials (as in Eq. (5.2) and evaluating the definite integrals of Eq. (5.6), we find,

$$\begin{aligned} E &= \sqrt{\pi}\sigma(c_1 - c_2 + \frac{\sigma^2}{2}(a_1 - a_2))^2 + && (5.7)\\ &\frac{\sqrt{\pi}}{2}\sigma^3(b_1 - b_2)^2 + && (5.8)\\ &\frac{\sqrt{\pi}}{2}\sigma^5(a_1 - a_2)^2, && (5.9) \end{aligned}$$

where a_2, b_2, and c_2 are the corresponding polynomial coefficients of the prototype function, f_2, found as in Eq. (5.5). The dependence of E on σ makes sense. For large σ, we are comparing the functions over a large area, and the difference in the quadratic terms of the polynomial expansions dominates. For small σ, the difference in the constant terms is most important. This expression, with Eq. (5.5) for the polynomial coefficients, lets us find a squared difference measure of how closely I(x) locally resembles a prototype function by taking a sum of the squared difference of local filter outputs. That will tell us how much the local structure of the oriented energy resembles that of a prototypical junction.

We are interested in how the shapes of the oriented energies compare, not their magnitudes. To remove any magnitude variations not taken care of by the gain normalization step, we divide the filter outputs by the G_0 output, and alter the prototype coefficients accordingly.

To count as a junction, the local oriented energy profile must match the prototype shapes along two directions for each of the two different orientations. We would like to simply cascade measures of agreement with the prototype functions. We can do that by multiplication if we first convert each of the squared difference measures, E, to numbers, p, between zero and one, where one corresponds to zero squared difference. We used the function,

$$p = \frac{\beta^n}{(E^n + \beta^n)}, \tag{5.10}$$

where β and n determine the offset and sharpness of the transformation, respectively. (We used $n = 3$, $\beta = 0.3$ to measure contour-ness and $n = 3$, $\beta = 0.1$ for stopped-ness).

Figure 5-9 shows a diagram of the overall system. The G_1 and G_2 derivatives taken against the contour allow us to use Eq. (5.9) to measure "contour-ness"–whether or not we are on top of the contour. The derivatives taken along the contour measure "stopped-ness"–whether or not we are at the end of the contour.

We are at a junction when the contour-ness along each of the dominant orientations is high. The results of the stopped-ness computation classifies the junction type, as shown in Fig. 5-10. If both contours have a high stopped-ness, then the junction is an L-junction. (See also [92] for another approach to L-junction detection). If one orientation shows high stopped-ness, and the other shows low stopped-ness, then the junction is a T-junction. If neither orientation shows stopped-ness, then that shows evidence for an X-junction.

For example, let p_{1c} be the output of Eq. (5.10) for the contour-ness of the contour at orientation 1, p_{1s} be its output for the stopped-ness of that contour, with analogous labelling for the measurements along orientation 2. The T can occur with either orientation 1 being stopped and orientation 2 not, or vice versa. Thus, we have for the T-ness, T:

$$T = p_{1c}p_{2c}\max[p_{1s}(1-p_{2s}), p_{2s}(1-p_{1s})] \tag{5.11}$$

We form the outputs of the other detectors analogously, in accordance with Fig. 5-10.

We have a choice for the representation of the final result: it can be winner-take-all or distributed. A winner-take-all representation is like a digital number, with a bit corresponding the winning junction set to "1", and the bits corresponding to the losing junctions set to "0". A distributed representation stores an analog response for each of the three junction types. A winner-take-all representation is more robust against noise, but carries less information than a distributed representation. Because it is important to present higher visual processing levels with the ambiguities or uncertainties of visual measurements, we chose a distributed representation for our results. We retain the response of each detector (e.g., Eq. 5.11) at each image position.

The system can successfully identify and classify junctions in simple images. Figure 5-11 shows some results. (a) - (d) show simple examples of L, T and X junctions; (e) - (h) show the relative outputs of each of the 3 types of detectors. (i) shows the insert of Fig. 5-2, and (j) shows the T-junctions detected. Note that where the top of the hand meets the head in (j), a T-junction involving a very high-contrast to low-contrast transition excites both the T and corner detectors. That makes sense, since the high-contrast T-junction is somewhere in between a corner and a simple T-junction.

Figure 5-13 shows the junction detection results for the Einstein portrait of Fig. 3-1 (a). A number of the T-junction responses in (a) correctly label occlusion T-junctions, such as at arrow 1. Arrow 2 points to a properly labelled T-junction which occurs not from occlusion, but from a coincidental (well-dressed?) alignment between the shirt lapel and coat border. There are properly few responses of the X-junction detector, (b). Most of the responses are in the hair.

5.3 Discussion

The method presented above identifies and classifies L, T, and X junctions properly in simple images. It does so independently of the phase of the contours which define the junction. All the operations are local. It uses the same image processing mechanisms that are thought to be available in the early stages of processing in the visual cortex.

Because this method of junction detection involves the squaring and blurring of oriented linear filter outputs, it resembles computational models of preattentive texture perception [12, 70]. Such texture perception mechanisms could be linked together to detect junctions. Conversely, stages of the junction detector can discriminate regions of texture. This is illustrated in Fig. 5-14.

We have only implemented this method on a single scale. A more robust implementation would accommodate the variation in scale observed in natural images (e.g., [72, 114, 19]). One could use two different approaches. One could implement the junction detectors with sets of filters spanning a range of spatial scales and then apply a voting or robust estimation procedure to reach a consensus from the detector outputs. Alternatively, one could try to find the best spatial scale or range of scales to describe structure at every region of the image and only use the detector outputs within that range. Stable bandpass filter output zero-crossings across scale may indicate important size regimes [114, 9]. An approach such as [9] may be used to mark the preferred spatial scales.

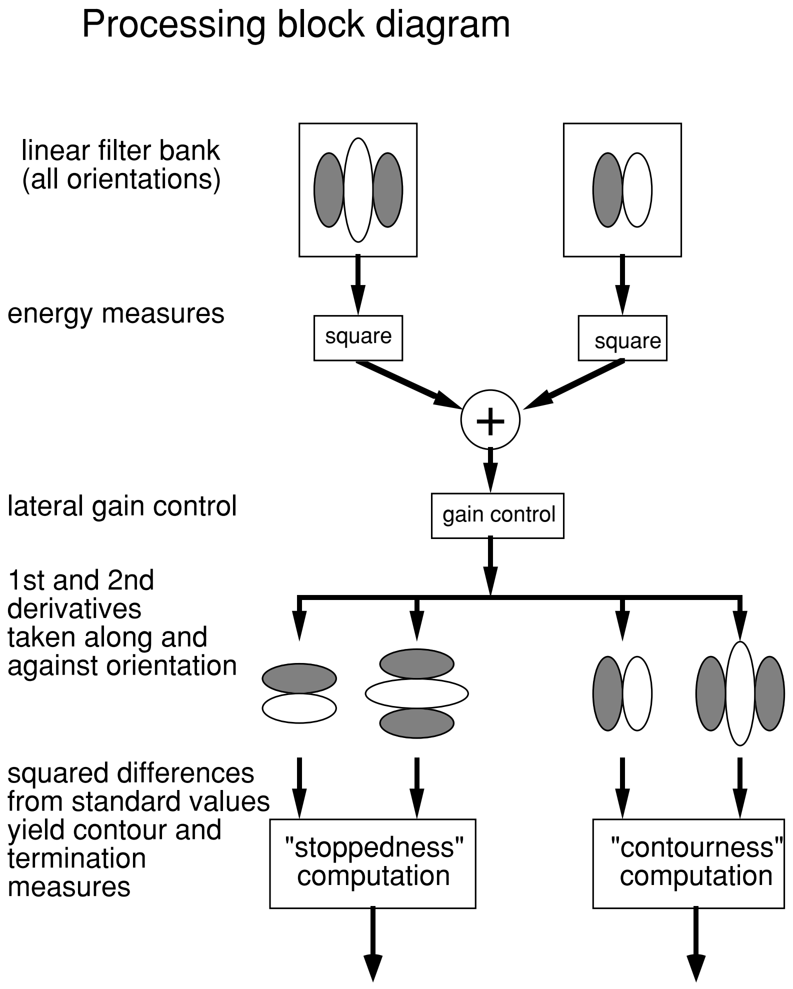


Figure 5-9: Block diagram of the filter-based processing to identify and classify junctions. A bank of linear filters of two phases, covering all orientations, analyzes the image. From those responses, energy measures are formed, and the two dominant orientations are found. A normalization step is applied to the energies. First and second derivatives are taken along and against the two dominant orientations. Squared differences from prototypical responses yield measures of “stopped-ness” and “contour-ness”.

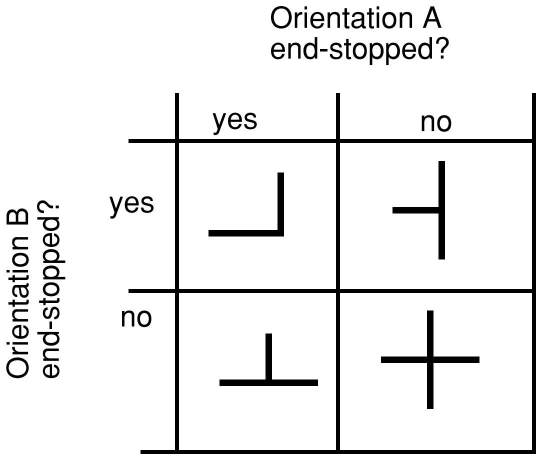


Figure 5-10: Junction classification. A region with high "contour-ness" along each dominant orientation defines a junction. If both orientations show stopped-ness, the junction is classified as an L-junction. If one orientation shows stopped-ness, but not the other, the junction is a T-junction. If neither orientation is stopped, junction is classified as an X-junction.

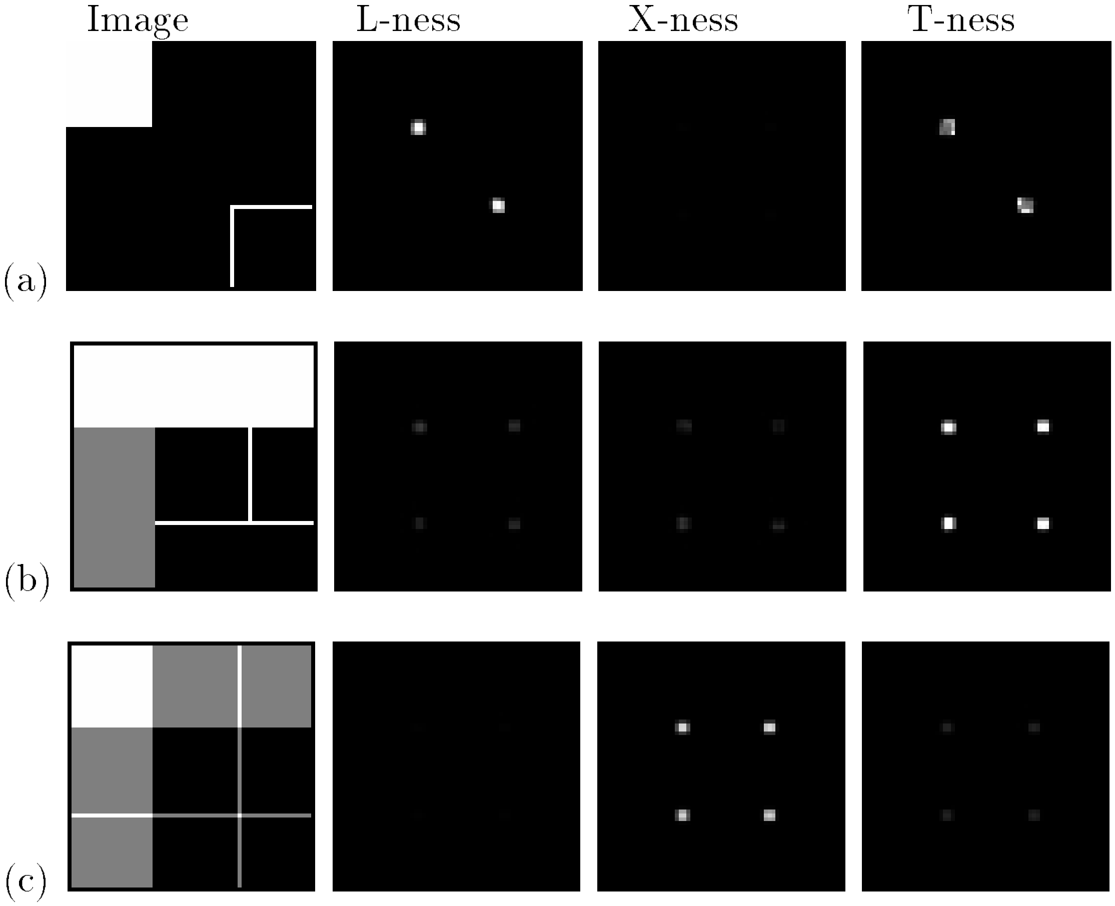


Figure 5-11: Showing outputs of local junction detectors made from oriented filter outputs. Test images are composed of (a) L-junctions, (b) T-junctions, and (c) X-junctions. In each case, the correct filter responds at the desired location, and the incorrect filters do not respond significantly. Note that the detectors operate correctly on junctions composed of either lines or edges.

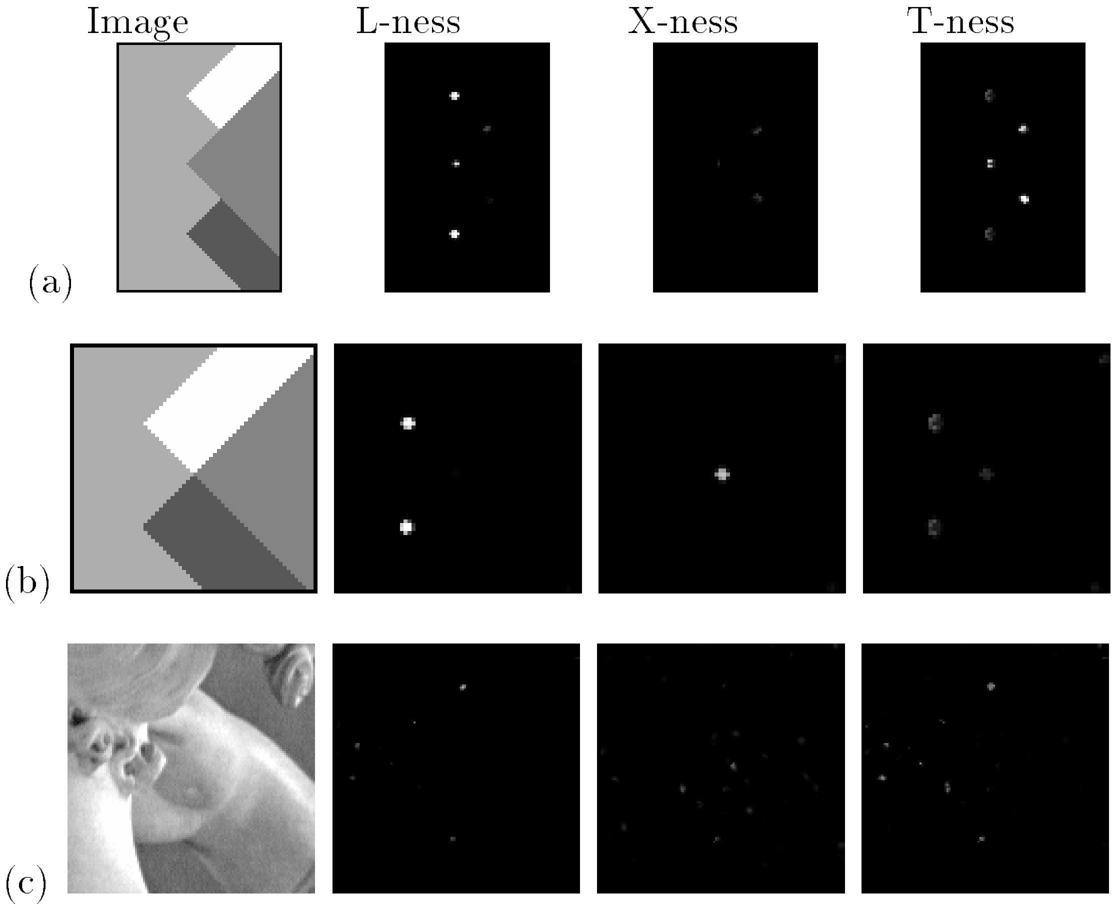
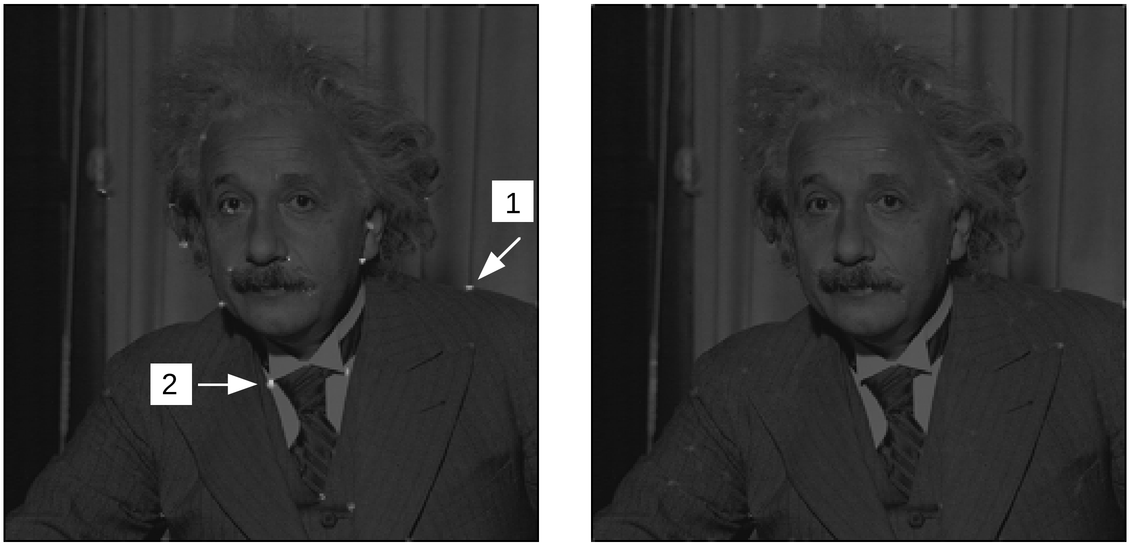


Figure 5-12: More outputs of local junction detectors made from oriented filter outputs. Test images (a) and (b) contain L, T, and X junctions. (c) is a detail of Fig. 5-2. While there is some response at junctions by detectors of different types (for example, a T-ness response at L-junctions), the strongest responding detector is always of the correct junction type. In (c), the higher occlusion point of the hand behind the head is a very high contrast T-junction. It gets classified as having some T-ness and some L-ness, a reasonable classification, based on the local image information.



(a) T-junctions

(b) X-junctions

Figure 5-13: Output of T and X junction detectors for Einstein portrait, overlaid on original image. The T-junction detector, (a), fires at some expected (arrow 1) and unexpected (arrow 2, an accidental alignment) T-junctions. The X-junction detector, (b), is mostly silent, as desired, except in the hair.

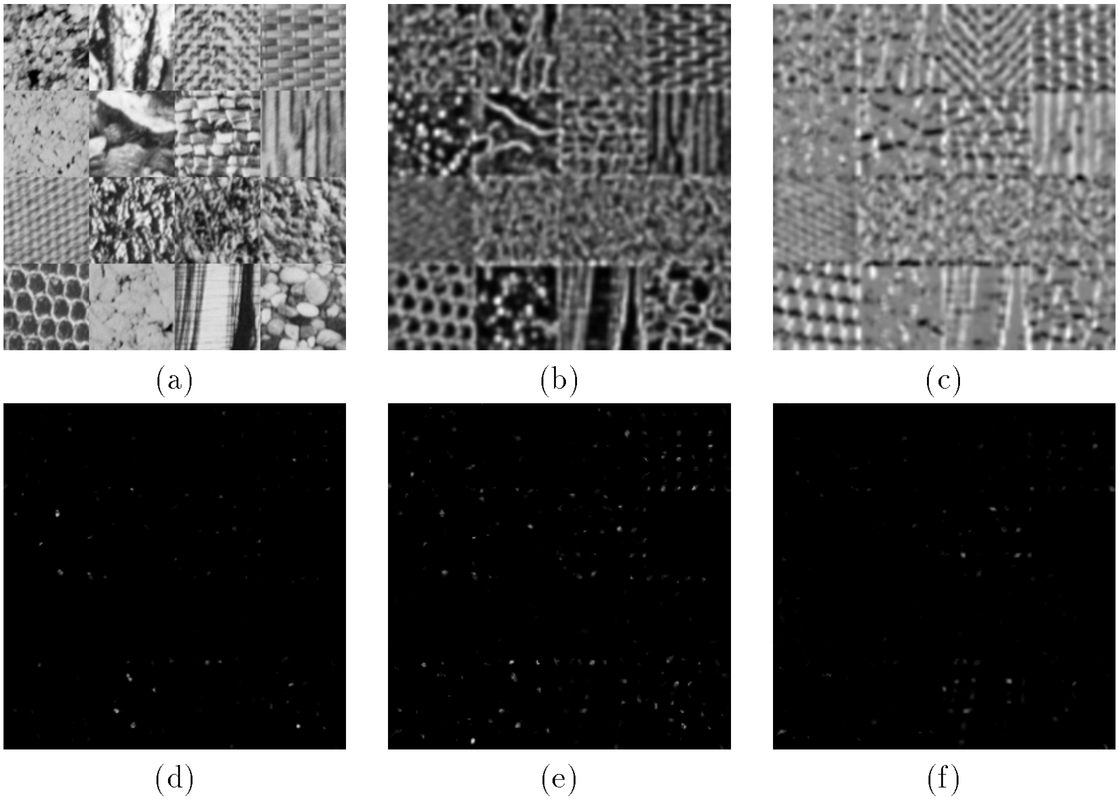


Figure 5-14: Junction detectors have similarities to filter-based computational models for texture discrimination. These detectors, and intermediate steps in their calculation, can give rough discrimination of textures. (a) Texture image. (b) and (c) are intermediate results in the junction calculation (the DC and $\sin(2\theta)$ components, respectively, of the Fourier series for the normalized oriented energy as a function of angle). Their squared, blurred values would discriminate between different texture regions. (d), (e), and (f) are the outputs of the L, T, and X junction detectors, respectively. Again, different texture regions show different characteristic responses.

Chapter 6

Cue Detection II, and Propagation of Local Evidence

6.1 Overview

The methods for junction analysis described in the previous chapter have certain limitations. Figure 6-1 illustrates junctions which pose problems for filter energy based methods. An energy-based method is likely to confuse in impulse with a contour, causing Fig. 6-1 (a) to be incorrectly labelled as two T-junctions. Depending on the scale of the filter, it may have trouble detecting junctions where one contour is of low-contrast or has a gap, as in Fig. 6-1 (b). Fig. 6-1 (c) and (d) show the outputs for these images, which exhibit the expected problems.

To address these limitations, we introduce a junction detector which is based on salient contours. It integrates information over a larger area than the energy based method, which allows it to bridge gaps. Since a dot would not be marked as a salient contour, Fig. 6-1 (a) could be processed correctly. The contour based analysis of junctions also provides a framework in which to spatially propagate local cue evidence.

6.2 Related Work

Researchers have used related contour-based approaches before. Lowe and Binford [14, 68, 69, 65, 66, 67] used relationships between contours to form image interpreta-

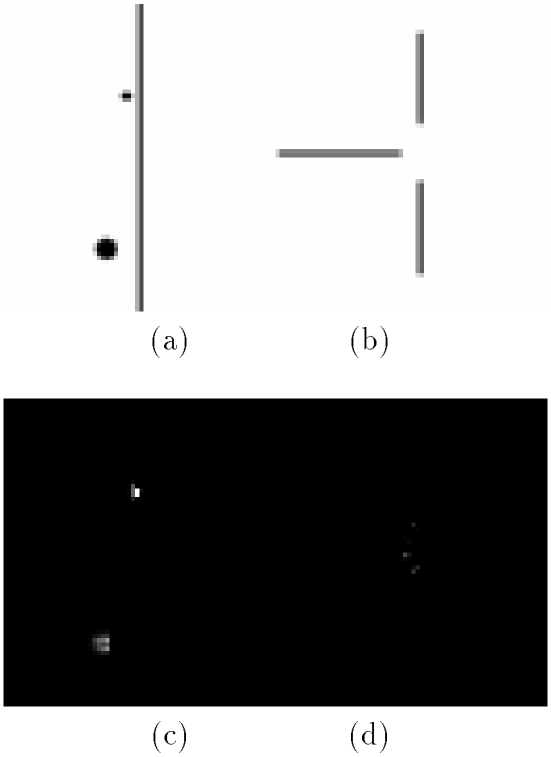


Figure 6-1: Image illustrating problems of a local approach to junction detection. Some structures, such as those in (a), can mimic the local oriented energy structure of a junction, causing false detections of T-junctions. A local approach can not fill gaps in contours, shown in (b). The T-junction detector of Chapter 5 fails for both these images. It incorrectly responds to the spot near the contour, (c), and give a negligible response to the T-junction at the contour with a gap, (d).

tions. In [69], they exploited assumptions of general camera and illumination positions to derive a set of 3-d inferences from observed 2-d relationships among lines. Their computational results, however, began from hand-drawn spline curves. In [66, 67], Lowe combined some of these inferences with a model-based approach and showed robust results for identifying instances of the model in a cluttered scene. Our approach is from a lower level than this. We will use only local calculations, and operate on contours instead of straight line segments. In some sense, we incorporate their grouping based on collinearity in the method we use to find contours.

Witkin [113] suggested that correlations between pixel values along curves parallel to edges could distinguish occlusion from shadow boundaries. Across an occlusion edge, the correlation between pixels would drop, while they would remain high across an illumination edge.

Grossberg and Mingolla [39, 40] used an analysis of contours as an early stage in their visual processing architecture. The method we use to find contours has similarities to their "boundary contour" process, which is sensitive to the orientation and amount of contrast but not to the direction of contrast in edges. Their work adds important insights, among them the need for competitive processes to precisely locate line terminations when using oriented filters.

In a recent thesis, Nitzberg [81] used a contour based approach to find an optimal layered interpretation for simple images, allowing continuation of occluding contours. He found contours using a Canny-like algorithm with extensive post-processing. The global interpretation guided the local interpretation of image cues. Classifying T-junctions was the last step, after a global interpretation has been found. His system assumed that all intensity edges were occluding boundaries, and so could not handle images such as Fig. 1-1 where contours can have different possible interpretations.

Williams [110] also found an optimal continuation and layering. His system, which was limited to working with straight edges, used integer linear programming to find the best set of line segment continuations.

Beymer [13] has extended Canny edges to meet at junctions in order to search for occluding T-junctions, which he noted occur in pairs if edges remain unbroken and if the image boundary is considered an occluding boundary. He paired junctions together along curves to form simple occlusion interpretations.

6.3 Finding Salient Contours

Our new method to analyze junctions has three parts: (1) finding salient contours in the image; (2) finding local evidence for various image cues from the configuration of the contours; (3) propagating the local evidence along the salient contours.

The salient contour measure we want should tell us the likelihood of an image contour at a given image position and orientation, based on the responses of oriented filters. It should favor long, straight curves, and continue over gaps. A number of approaches could be used, including relaxation labelling [24, 52, 44, 83], the approach of Grossberg and Mingolla [39, 40], snakes [53], or dynamic programming [95]. Splines [11], or elastica [81, 80] could be used to interpolate across gaps. We implemented the dynamic programming method of Shaashua and Ullman [95]. It solves an explicit optimization problem, gives good results, and the computation time is linear with curve length.

(A note on terminology: in this chapter, “contour” has the burden of several meanings. We will discuss “image contours”–lines or edges in the image. We build a variation of our contour detector of Section 4.1 which measures “contour strength”. We will use the output of the contour detector in the dynamic programming algorithm to find “salient contours”–long curves which bridge gaps and connect contour fragments. We will use the word “salient” before those contours which result from Shaashua and Ullman’s algorithm, or our modification to it. We will sometimes call them “paths”.)

6.3.1 Post-Processing of Energy Outputs

The representation for Shaashua and Ullman’s scheme is of a set of 16 orientation elements arriving at each pixel position from neighboring pixels. Along each orientation element there is some local image evidence of orientation strength. In their implementation, Shaashua and Ullman used fragments of Canny edges for such local evidence. From Chapter 4, we know that important structures come in all phases, and we will base our evidence for orientation strength on local energy measures. Furthermore, Canny edges, based on first derivatives, give poor performance at junctions ([13] discusses this issue). To ensure adequate representation of junction structure, and to match the sampling resolution in orientation, we will analyze orientation with the fourth derivative of a Gaussian, G_4, and its Hilbert transform, H_4. Having 8 samples in orientation (corresponding to 16 angles) is approximately enough to make a steerable basis for filters with this tuning (9 would be exactly enough).

We found that spurious salient contours were reduced if we post-processed the local energy outputs to keep only the outputs which had locally maximal response in orientation and in position perpendicular to the orientation. This is similar to the contour detector described in [89].

We first blur the oriented energy outputs to remove interference effects, as discussed in Section 3.2.2. To find the spatial local maxima of the blurred energies, we apply a second level of quadrature pair filtering to the energy output at each orientation. Energy local maximum points correspond to positions of white on black phase in the energy image (see Section 4.2). To make a mask, M, which marks the energy local maxima, we form

$$M = \begin{cases} \cos^s(\phi - \pi) & \text{if } \frac{-\pi}{2} \leq \phi - \pi \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} , \tag{6.1}$$

where ϕ is the local phase angle, and s is a selectivity factor, set to 4.0. Figure 6-2 illustrates this energy local maxima marking on a test edge image.

After applying Eq. (6.1) to eliminate non-maximal responses in position, and further eliminating non-maximal responses in angle, we have a mask, M, between 0 and 1 for each position and orientation which identifies image contours. However, some energy local maxima are of very low contrast, and we do not want to consider them contours.

To eliminate these noisy, low-contrast responses, we run the energy through a point non-linearity and multiply it by the mask, M. We introduce a simple noise and signal model, and use a non-linearity which would give the probability that an energy response was not caused by noise. While we do not believe that the oriented energies follow the simple processes assumed in it, using the model gives us a non-linearity function with physically intuitive parameters to adjust.

We assume there are two Gaussian processes which generate linear filter outputs: a mean zero noise process, and a mean t contour process. We assume each has variance σ. Our contrast dependent multiplier is the probability that the observed oriented energy was caused by the contour process.

The oriented energy is the square of the linear filter outputs. The probability, p, of a given oriented filter measurement, x, from a mean t Gaussian process, as described above, is $p(x) = e^{-(x-t)^2/(2\sigma^2)}$. From basic probability theory [50], the probability of the energy measurement, $y = x^2$, is

$$p(y) = \frac{1}{2\sqrt{y}}(e^{-(\sqrt{y}-t)^2/(2\sigma^2)} + e^{-(-\sqrt{y}-t)^2/(2\sigma^2)}) \tag{6.2}$$

Multiplying M by $p(y)$ gives a gain controlled contour measure between 0 and 1 for every orientation and position. Figure 6-3 plots this function for the parameters we used, $t = 100$ and $\sigma = 22$.

Figure 6-4 shows an example of energy normalization using this function. (b) is the blurred horizontal oriented energy for the image in (a) (note the large range of the energies because of the range of contour contrasts). (c) shows the positions for which the horizontal energy response is maximal in angle and perpendicular position. (d) is the union of the local maxima for all orientations.

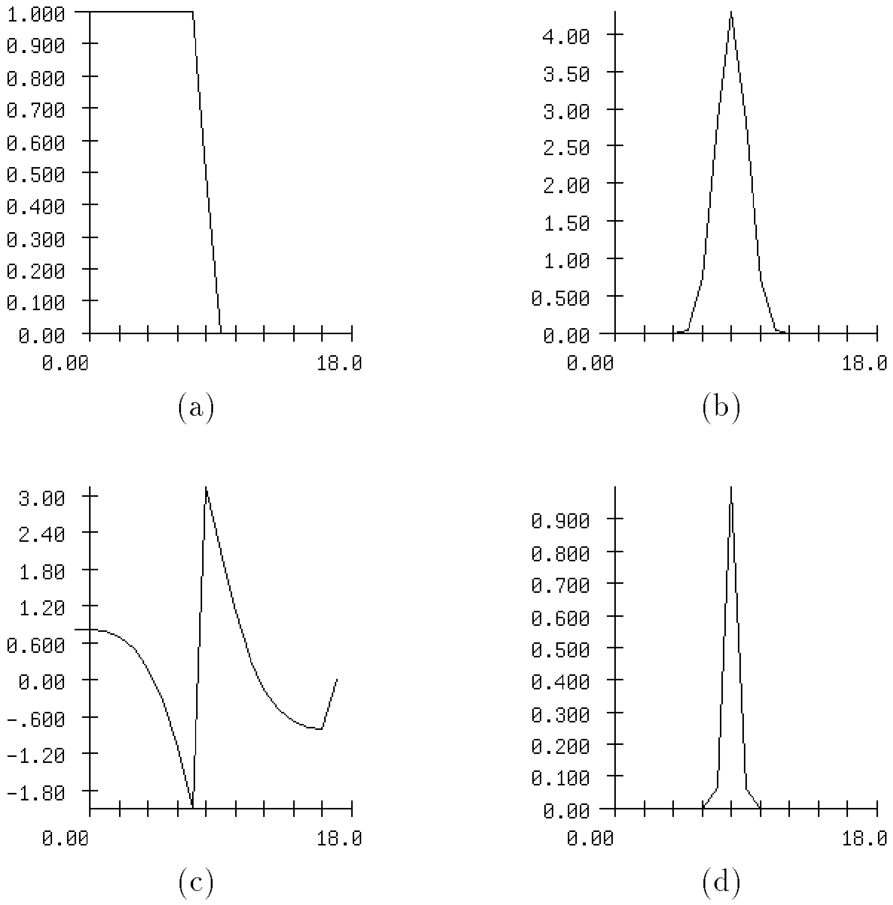


Figure 6-2: 1-d cross sections of images showing method to mark local contour evidence. (a) Input image (an edge). (b) G_2, H_2 quadrature pair energy output. The spatial local maximum of this response marks the position of the (edge) contour. This energy output alone would not serve to mark the contour because it is too wide, and the magnitude is a function of contour contrast. We apply a second stage of G_2, H_2 filtering to measure the local phase, (c), of the energy output. Energy local maxima will appear as positions of white-on-black phase (π radians). The non-linearity of Eq. (6.1) applied to the phase output will then mark those contour positions with output 1.0 in a narrow but smooth spike signal, (d).

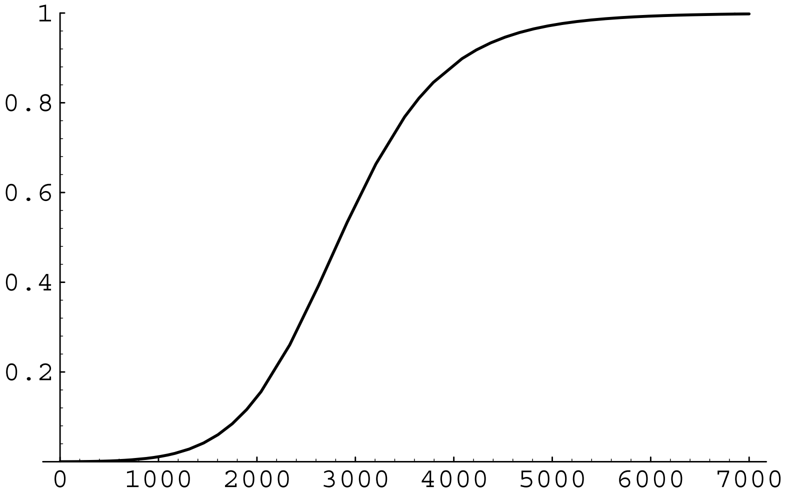


Figure 6-3: Energy normalization function used to suppress low-amplitude noise. See text for derivation.

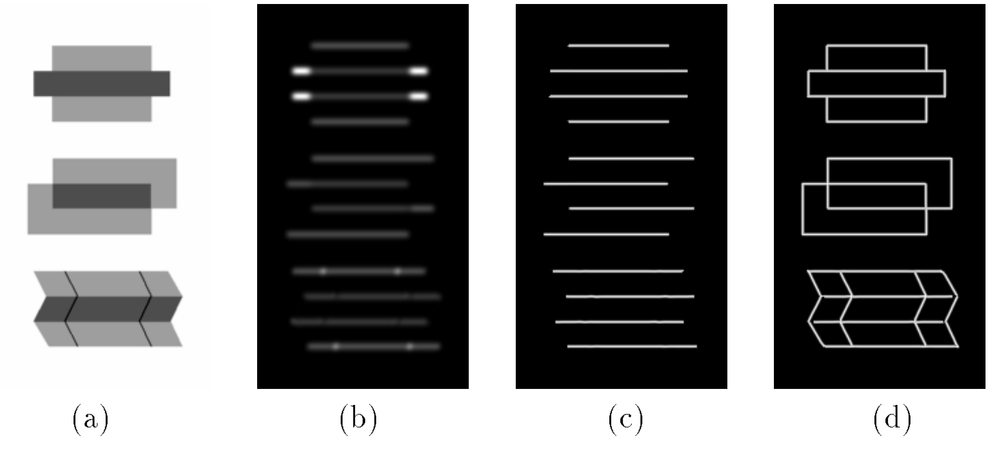


Figure 6-4: From image to oriented contours. (a) Test image. (b) blurred horizontal oriented energy. Note high dynamic range because of the range of image contrasts. (c) Horizontal contour strength, calculated as described in text. (d) Union of contours found for all orientations.

6.3.2 Saliency

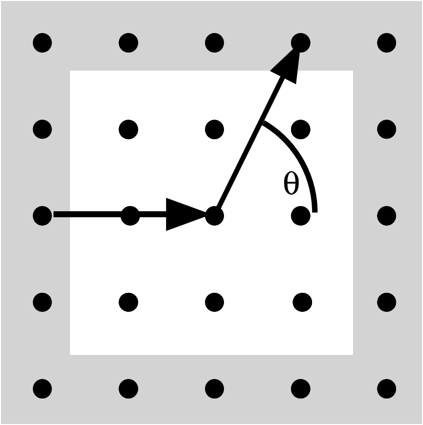


Figure 6-5: Image representation used with structural saliency calculation. Shown are two of the orientation elements which point to and leave from each point. In this implementation, orientation elements span two pixels, as shown. Each black circle represents a pixel. The grey box covers all the pixels to which an orientation element at the center pixel can connect. Each of the 16 elements which arrive at a point is connected with one of the 16 elements which leaves from that point. Following the connections from one element to another through different positions traces a curve. Each element has a saliency value, which depends on the local evidence for each orientation element, as well as on the bending angles, θ, between incoming and outgoing elements in the curve.

Figure 6-5 shows the set of 16 orientation elements which meet at each pixel in the scheme of Shaashua and Ullman. Each orientation element can link its head to the tail of any of the 16 elements which leave from the tail position. The task of the saliency algorithm is to find a set of linkings which traces salient image contours and to find a measure of the strength of the salient contours found, based on the local evidence for contours (or edges).

Shaashua and Ullman devised a local calculation, based on dynamic programming, which guarantees finding the most salient curve starting from a given orientation element. The saliency which the dynamic programming optimizes unfortunately depends on the direction in which the curve is traversed, but it does indeed give long curves of low total curvature a large saliency.

The recursive saliency calculation is as follows:

$$S_i{}^0 = \sigma_i \tag{6.3}$$

$$S_i{}^{n+1} = \sigma_i + \max_j [S_i{}^n f_{i,j}], \tag{6.4}$$

where $S_i{}^k$ is the saliency of the ith orientation element after the kth iteration, σ_i is the local saliency of the ith element, and $f_{i,j}$ is a coupling constant between the ith and jth orientation elements. The maximization is taken over all neighboring orientation elements, j. The coupling constant penalizes sharp bends of the curve and effectively imposes a prior distribution on the expected shapes of the image contours. Shaashua and Ullman showed that after N iterations, the above algorithm will find the saliency of the most salient curve of length N originating from each contour.

After every iteration, each orientation element has a favorite next element. To trace the maximal saliency curve, one has to follow the link after iteration N of the first element, then the link which that next element chose after iteration $N-1$, then the next element of that link after iteration $N-2$, etc. This entails storing a vector of the N linking choices for each orientation element, in order to trace the optimal curves of length N. Shaashua and Ullman implemented an approximation to this, storing only the last choice of each element, and tracing curves by following that set of links. In general, this is not the most salient curve, but in practise, the curves it draws are reasonable.

Figure 6-6 shows (a) a test image, (b) the orientation strength evidence, and (c) the maximum saliency over all orientations. Picking a salient point and following each link gives the curve shown in (d). The curve follows long, straight curves and traces the visually salient circle of (a).

6.3.3 Competition

Unfortunately, the problem which dynamic programming solves so elegantly is not the right problem for us. The algorithm finds the curve of maximal saliency which starts from any position. It will therefore assign high maximal saliency to curves which start near an image contour. After paying a small penalty for curving to join the contour, the path will follow the contour. A cloud of high saliency values tend to surround all image contours (Fig. 6-6 (c)). Furthermore, if two curves cross, the maximally salient contours will not necessarily cross each other, but may well merge onto one of the two contours. This method will not describe contours accurately at junctions.

Shaashua and Ullman addressed this problem [96]. They proposed that all linking pairings could be constrained to be reciprocal (if A chooses B then the vector opposite B must choose the vector opposite A) and that the set of linkings which maximized the sum of the saliencies over the image would create the desired linkings. They described a scheme which approximates this optimal grouping.

We implemented their scheme and found it to be unstable for the measures of orientation strength that we used. Salient contours would follow image curves for some short time, then veer off. We believe this difference from their result is due to the difference in the local evidence for orientation strength. They derived their local saliencies from thinned edge fragments, and as a result their local evidence for contours was always exactly one pixel wide. Ours were in general wider than this, and could vary in width along the contour. Such variations may cause the instabilities.

Seeking a more stable method, we opted to let many curves choose one orientation element but force them to compete for that element based on how strong each curve is relative to the others. The resulting curves are stable, yet delineate the contours of the image well.

The first step of this method is to calculate the salient contours of length N using the dynamic programming method of Shaashua and Ullman. This ensures that curves bridge gaps. Then we add a competition phase. One iteration of the competition phase is a modified version of a dynamic programming iteration. We weigh the saliency of an outgoing element by the "backwards strength", B_i, of the element choosing it relative to the sum of the backwards strengths of all the elements (index k) which choose it. Strong paths thus get first priority for picking the paths they want to connect with. That eliminates the "cloud" of salient paths picking one image contour, and encourages proper behavior at junctions.

The following iterative procedure calculates backwards saliency strength of element i after iteration n, ${B_i}^n$:

$$ {B_i}^0 \quad = \quad \sigma_i \tag{6.5} $$

$$B_i{}^{n+1} \quad = \quad \sigma_i + \sum_j B_j{}^n f_{i,j}, \tag{6.6}$$

where again σ_i is the local orientation strength, and $f_{i,j}$ is the coupling constant between orientation i and orientation j. The sum is over elements j which feed into element i.

The following recursion relation determines the saliency of the ith orientation element after iteration n, $S_i{}^n$:

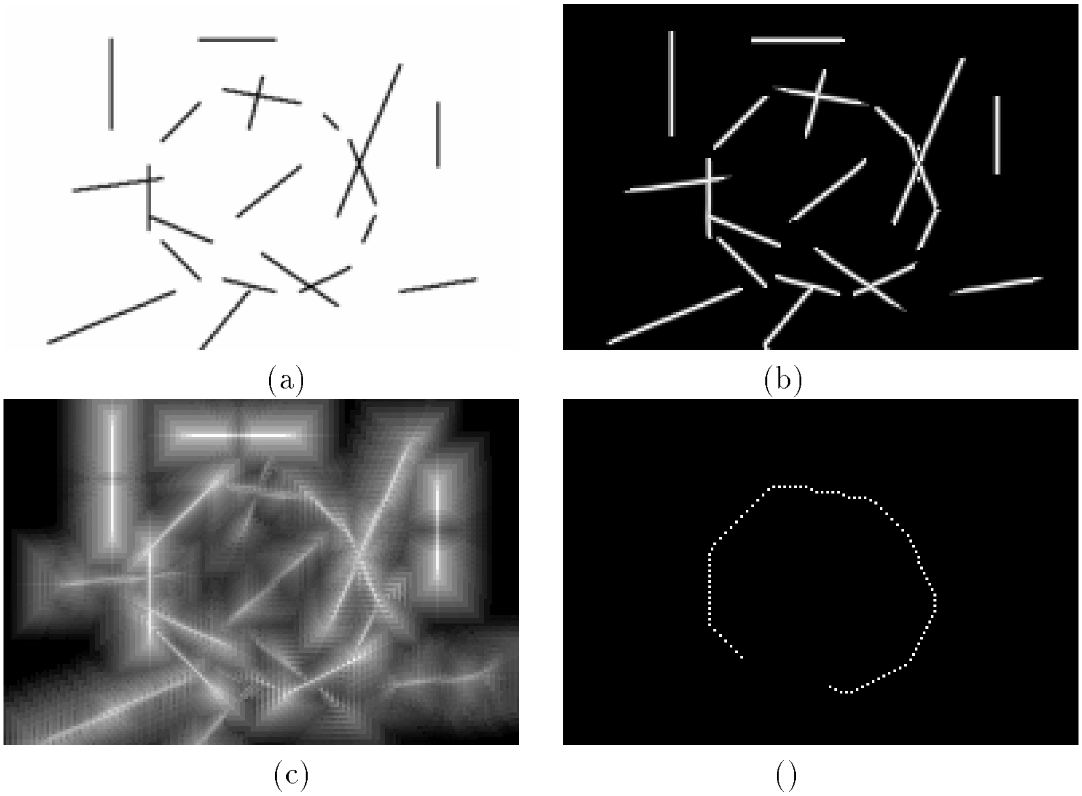
$$S_i{}^{n+1} = \sigma_i + \max_j \frac{S_i{}^n f_{i,j} B_i}{\sum_k B_k}. \tag{6.7}$$

The sum in the denominator is over elements k which chose to connect to element i on iteration n.

Unlike the dynamic programming algorithm, this is not guaranteed to find an optimum path, nor is it guaranteed that the saliencies reflect that paths found. Thus, we add a final step where we repeat the calculations of Eqs. (6.6) and (6.7), but without changing any of the links. While this does not guarantee that the paths are optimal, it does guarantee that saliency values accurately reflect the curve paths.

In practise, the salient contours this procedure finds are stable, and generally behave well at junctions. Figure 6-7 shows the outputs of the dynamic programming algorithm, and our modified algorithm, at a junction. The dynamic programming algorithm shows salient curves everywhere near an image contour, while for our modified algorithm salient curves are generally confined to image contours. With the dynamic programming algorithm, several branches of the ψ junction choose to follow the stem of the ψ. For our modified algorithm, the contours continue more as the eye might follow them.

After this step, at every position and direction in the image we have a saliency number reflecting the strength of the salient path heading in that direction, and a link telling the next step in that curve. In the next section, we will use this local configuration of saliencies to identify junctions.



(a)

(b)

(c)

()

Figure 6-6: Saliency calculation. (a) Original figure, adapted from [95]. (b) Orientation evidence, based on spatial and angular local maxima of oriented filter outputs. (Shaashua and Ullman used Canny edge fragments for this step). Based on the orientation strength evidence in (b), the saliency algorithm was applied for 20 iterations. (c) shows the saliency of most salient contour of the 16 contours leaving each position. Note the "cloud" of salient values surrounding each image contour. (d) Curve traced starting from a position and orientation of high saliency. The curves traced by following the last choice of each orientation element are a reasonable approximation to the maximally salient curves, which would require storing a vector of 20 numbers (one per iteration) at each position.

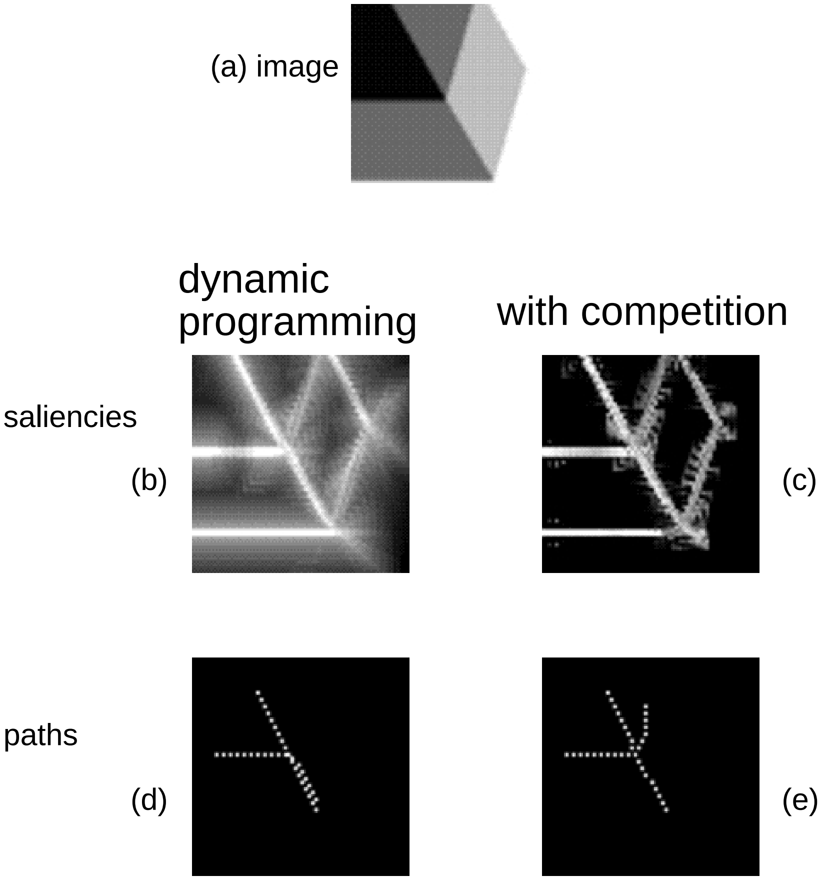


Figure 6-7: Comparison of contours and their strengths between dynamic programming algorithm and the dynamic programming with competition. (a) input image. Saliency images show the maximum over all orientations of the saliencies of elements at a position for (b) the standard dynamic programming algorithm and (c) the algorithm with competition added (no longer dynamic programming). Note the cloud of high saliency values around contours for the dynamic programming case. (d) and (e) show traces of two paths for the two algorithms. Without competition between curves, all paths at a junction may choose the same strong outgoing path. Competition allows a better parsing of the junction. The choppiness of the diagonal lines is related to the quantization in angle, discussed in Section 6.6.2.

6.4 Finding Local Evidence

In the saliency we have a local measure of a more global structure, contour strength. We want to analyze the configuration of contour strengths and find local evidence for visual cues. Examples of the types of cues we could look for include:

- T-junctions. If one image contour stops at another one, it may indicate that the bar of the T is an occluding contour.
- X-junctions. Two contours crossing may provide evidence for transparency.
- ψ-junctions. An image contour which causes others to bend as they cross it provides evidence for being due to a surface bend.
- Shading. Adelson [2] has pointed out that curves which change intensity as they change orientation may be shading cues. One could search for salient curves which satisfy that criterion.
- Reflectance or illumination edges. Edges due to either reflectance or illumination changes cause a multiplicative change in intensity across their boundary.

We will study the first three of these, which all involve junctions.

6.4.1 T and X–Junctions

The bar of a T-junction is a salient contour which meets a salient contour on one side of it, but not on the other. An X-junction contour is a salient contour which sees salient contours off to both sides. We will call X-junctions any junction where one curve crosses another and treat a ψ-junction as a special case of an X-junction.

To evaluate T-ness and X-ness, we find the strength of the strongest salient contour near the left or right side of an orientation element, which we call l and r, respectively. We first blur the saliencies slightly, to avoid misclassifications due to spatial quantization effects. l or r is the maximum saliency at the three orientations most perpendicular to the particular orientation element.

The classifications we want to make are similar to the logical operations NOR, AND, and XOR. If neither l nor r is large, we want to say there is no evidence for a junction. This corresponds to the NOR operation which would output a 1 in the region where $l = 0$ and $r = 0$. If both l and r are large, then there is a contour off to both the left and right sides and we have evidence for an X-junction. This corresponds to the logical operation AND, which outputs 1 where $l = 1$ and $r = 1$. If one of l and r is large, and the other is not, then there is evidence for a T-junction. This corresponds to the exclusive-or operation. These logical functions are plotted in Fig. 6-8.

We seek membership functions which divide up the l and r parameter space in a similar way. We divide the space into three regions: evidence for a T-junction ($T(l,r)$), an X-junction ($X(l,r)$), or no junction ($N(l,r)$). Based on analyzing the l and r values for a number of prototype junctions, and following the logical functions of Fig. 6-8, we used to following functions to divide the parameter space:

$$\begin{aligned} T(l,r) &= R(\sqrt{l^2+r^2}, t_r, s_r)\,(1 - A(\arg(r,l), t_a, s_a)) && (6.8)\\ X(l,r) &= R(\sqrt{l^2+r^2}, t_r, s_r)\,A(\arg(r,l), t_a, s_a) && (6.9)\\ N(l,r) &= 1 - R(\sqrt{l^2+r^2}, t_r, s_r), && (6.10) \end{aligned}$$

where the radial, $R(x,t,s)$, and angular, $A(\theta,t,s)$, functions are:

$$\begin{aligned} R(q,t,s) &= 1/(1+e^{(-s(q-t))}) && (6.11)\\ A(\theta,t,s) &= \begin{cases} R(\theta,t,s) & \text{if } \theta < \pi/4 \\ R(\pi/2-\theta,t,s) & \text{if } \theta > \pi/4 \end{cases} && (6.12) \end{aligned}$$

and t_r, s_r, t_a, s_a are parameters which determine the sharpness and thresholds of the classification transitions. Figure 6-9 illustrates this classification of the l–r parameter space.

We must include two more constraints before we classify junctions. The orientation element at which we measure l and r must itself be a strong salient path, both in the forward direction and in the backward direction. (Without this constraint any image contour would show evidence for X-junction-ness all along it, since orientation elements perpendicular to the contour see strong saliency both to their left and right.) Also, the curve on which the orientation element lies must not have high curvature, since then the strong path it sees to the left or right could then be the continuation of its own curve. We can apply these constraints through local calculations with preset thresholds.

The above method works well to identify and classify T and X junctions for simple images. The "soft" partitioning of the l–r parameter space allows for a graceful change in the classification of T and X junctions, as illustrated in Fig. 6-10.

The junction analysis using salient contours allows us to correctly analyze junctions which the local filter method of Chapter 5 could not. Figure 6-11 shows examples. The left figure of (a) shows dots next to a line, which caused spurious responses in the local energy based T-junction detector. However, the dots have very low saliency, and the salient contour based junction detector does not label them as a junctions, as seen in (d). The right figure of (a) shows a T-junction where the stem of the T terminates at a gap in the bar of the T. Such a junction can occur in natural images where contour contrasts are variable. The local energy based measure (d) did not identify this as a T-junction. The salient contour based junction measure, which can bridge gaps in contours, successfully identifies this as a T-junction.

6.4.2 ψ-Junctions

The third type of junction we want to detect is a ψ–junction, shown in Fig. 5-1. In the blocks world domain it is sufficient to check that two of the contours which meet at the junction are parallel and that two are not. We want to generalize to real images.

Suppose there is a flat surface with straight contours marked on it, as shown in Fig. 6-12. If we bend the surface, and view it under orthographic projection, the contours as viewed in the image will have maximal curvature at the point of maximum bending of the surface. Suppose the shading at the bend in the surface causes a salient contour in the image. Then a detector which responds at salient paths which has other salient paths of high curvature crossing it will respond maximally at the contour caused by the normal change. That is the basis for our ψ junction detector.

For every curve crossing an orientation element, we calculate its local curvature, weighed by its saliency. We restrict that response to X-junctions by multiplying by the local X-junction-ness. The result is a detector which responds maximally to ψ junctions which may indicate normal changes.

Figure 6-13 shows the result of the three junction detectors on the test image of Fig. 1-1. These salient contour based algorithms correctly identify all instances of all three junction types in the image.

It is illustrative to examine the detector outputs for Einstein portrait of Fig. 3-1 (a). Like the results on this relatively complicated image from the energy based method (Fig. 5-13), some of the junctions are labelled correctly and some are not. The T-junction response of Fig. 6-14 (c) at arrow 2 correctly identifies a T-junction where the fold of the tie knot ends in front of the shirt. Arrow 1 in (b) and (d) marks a spurious transparency caused by an incorrect contour continuation. As seen in the saliencies, (b), the lapel contour incorrectly joins to a tie stripe contour. That contour crosses the boundary of the tie, causing the X-junction response in (d).

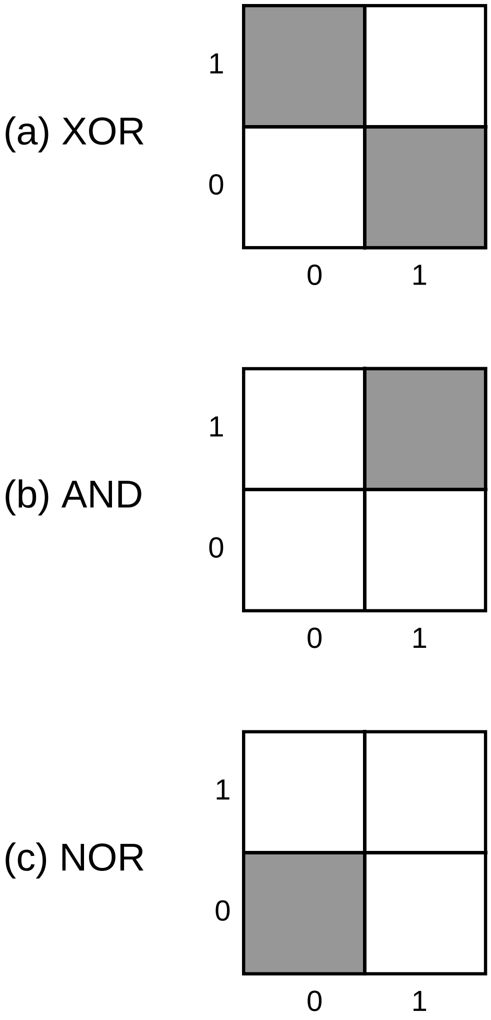
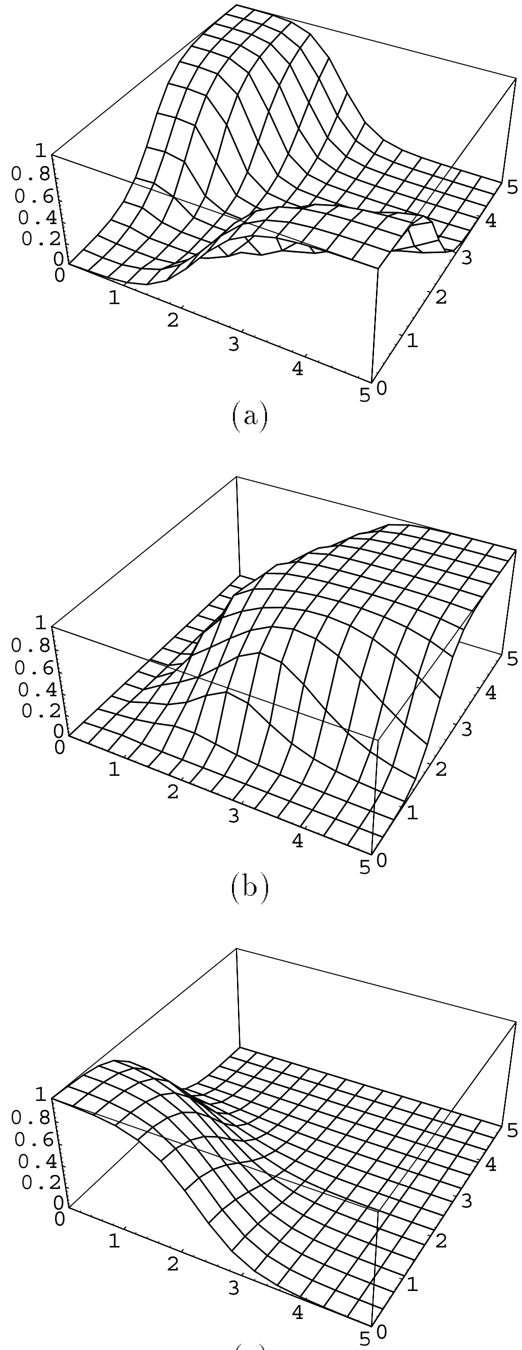


Figure 6-8: Intuition behind the junction classification scheme of Fig. 6-9. The (a) XOR, (b) AND, and (c) NOR functions correspond to the classifications for evidence for T-junctions, X-junctions, and no junction, respectively, in Fig. 6-9.

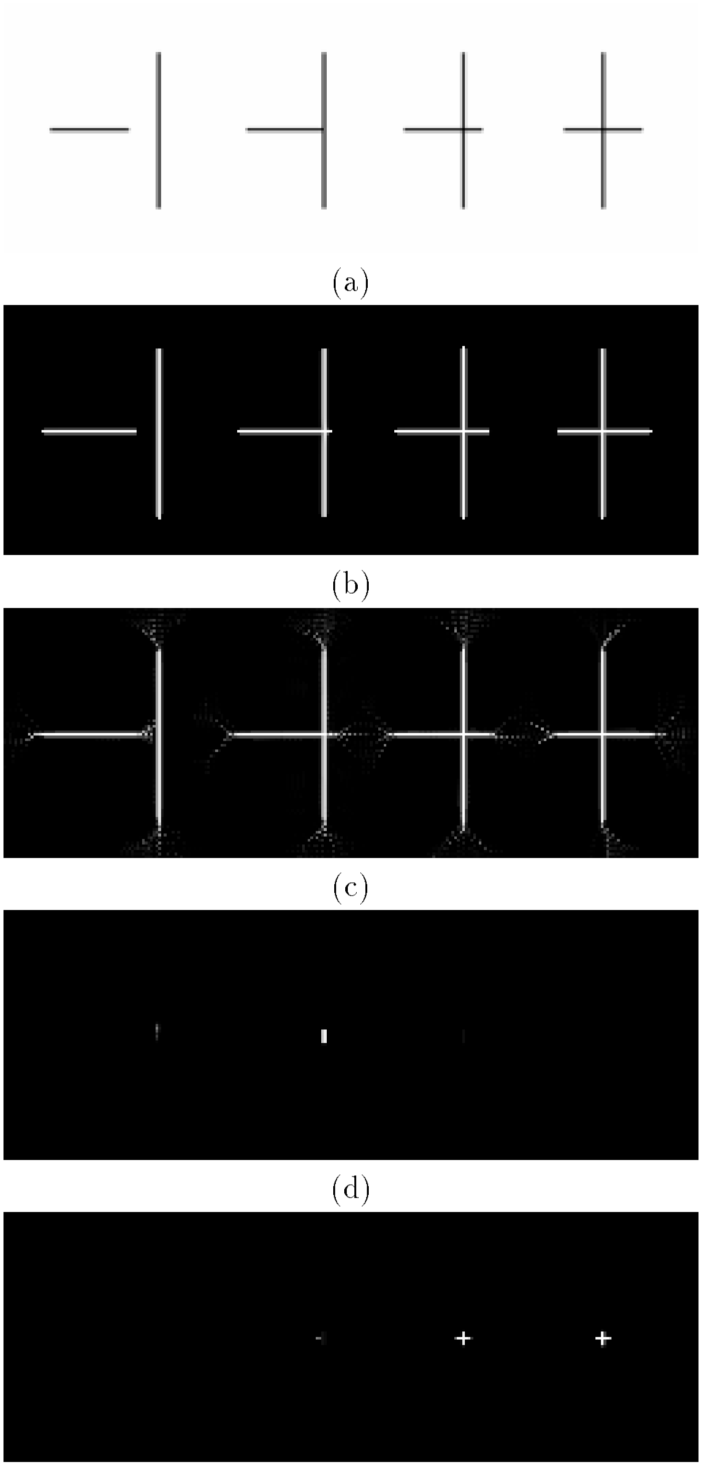


(a)

(b)

(c)

Figure 6-9: Classification of local saliency data. The horizontal axes are the l and r values as described in text. Plots show local evidence for (a) T-junction, (b) X-junction, and (c) no junction. Functions were rough fits to saliency values at test junctions, based on the prototypes of Fig. 6-8.



(a)

(b)

(c)

(d)

(e)

Figure 6-10: Showing system response to junctions which show a gradual change in type from T to X. (a) Image. (b) Contour data input to cooperative network. (c) Maximum over all orientations of modified saliency. (d) and (e) show the local evidence for occlusion and transparency, respectively. Note the smooth transition from T-ness to X-ness.

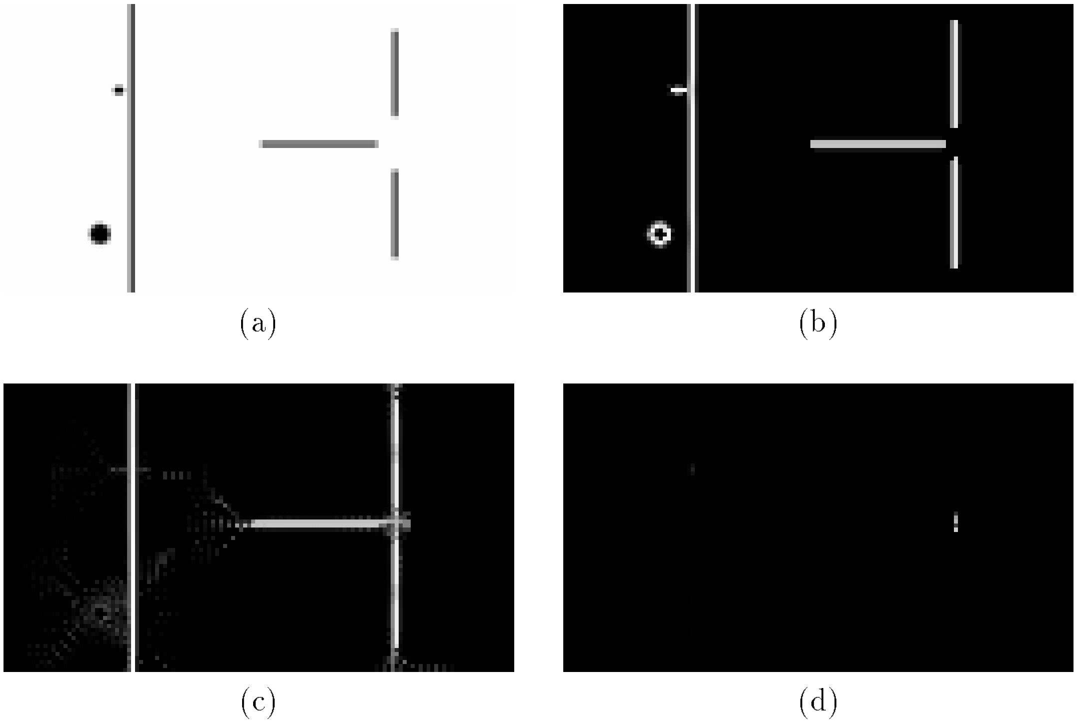


Figure 6-11: Images showing the advantages for junction detection of an approach based on salient contours. (a) Image for which the methods of Chapter 5 fail. (b) Contour data input to cooperative network. (c) Maximum over all orientations of the output of the cooperative network, the modified saliency. (d) shows the local evidence for occlusion. The T-junction detector correctly finds no T-ness in the left figure but does respond to the other figure, despite the gap.

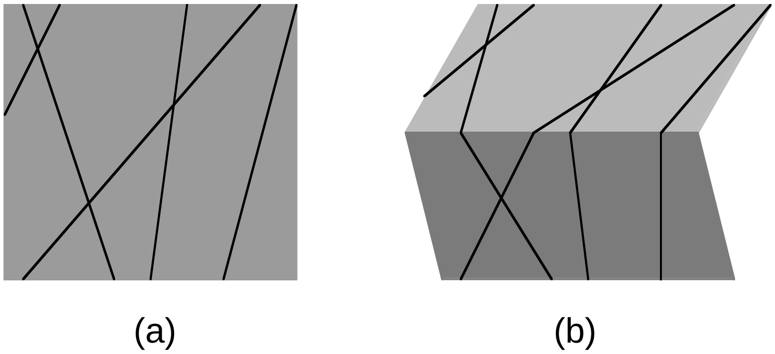


Figure 6-12: The intuition behind identifying ψ junctions with surface bends. Consider a flat surface marked with contours which are at least roughly straight, (a). Suppose we bend the surface, and that that bend introduces a salient contour from the shading, (b). Then the surface bend will cause the projected images of the other contours to curve at the salient contour caused by the shading. An operator which detects points where high curvature paths cross salient contours will respond maximally at the ψ junctions introduced by the surface bend.

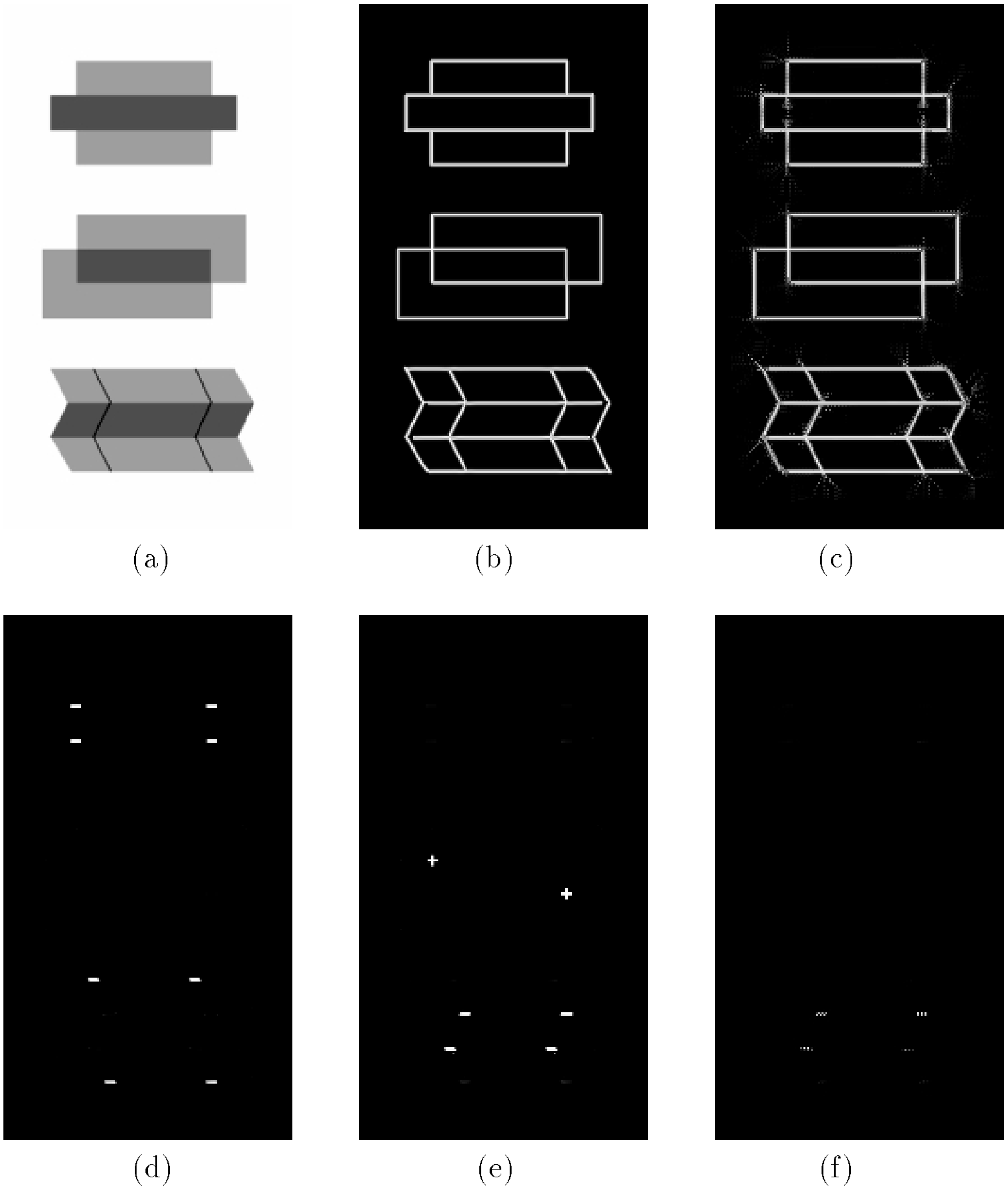
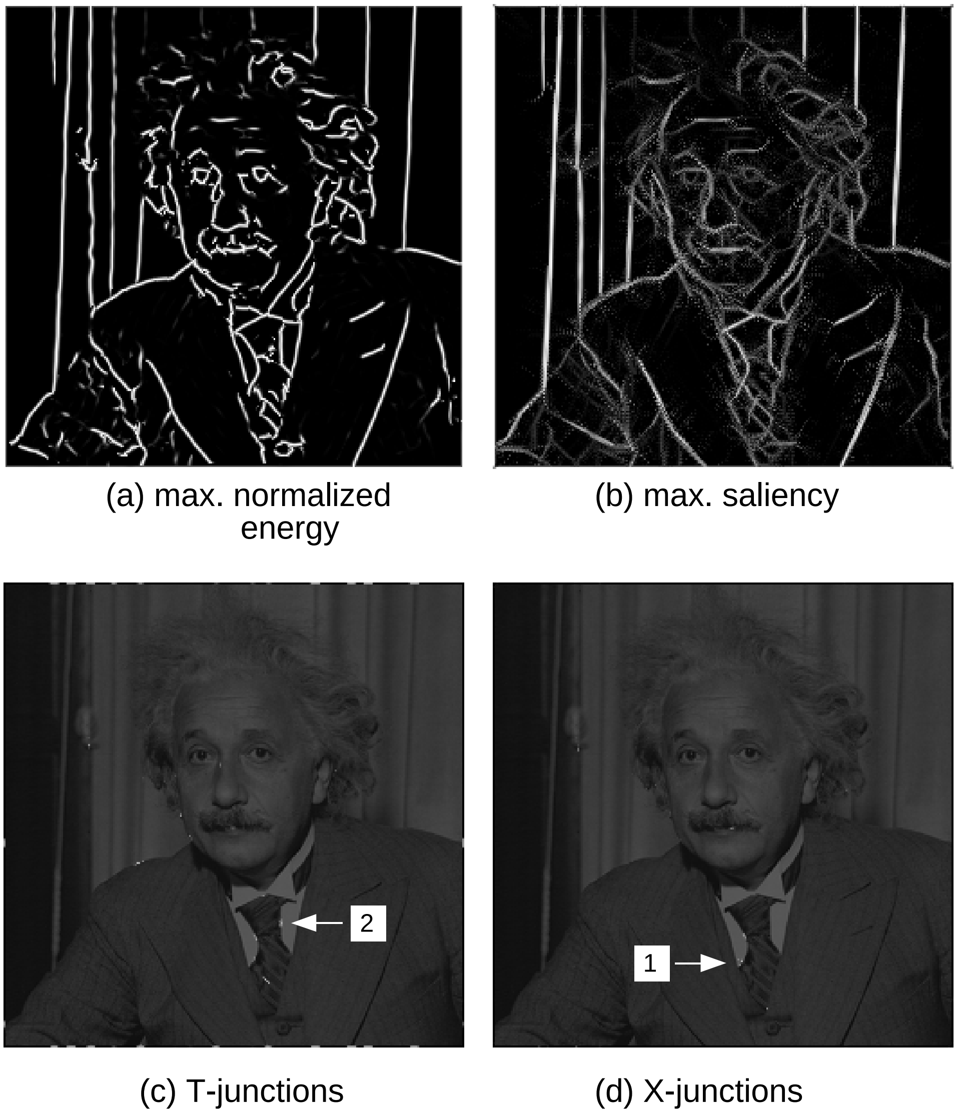


Figure 6-13: Local evidence results. (a) Image (Fig. 1-1) showing image contours due to occlusion, transparency, and surface bends. (b) Contour detection based on oriented energy outputs. This is the input to the saliency calculation stage. (c) shows maximum over all orientations of the modified saliency. Based on the local configuration of saliencies, we calculate evidence for (d) T-junctions, (e) X-junctions, and (f) ψ junctions. The system responds correctly in every case (the brightest false positive response is $\frac{1}{8}$ of the correct responses). The X-junction detector responds to curves crossing each other, and thus responds also to the ψ junctions. The response of the ψ junction detector can be used to distinguish X from ψ.


(a) max. normalized energy

(b) max. saliency

(c) T-junctions

(d) X-junctions

Figure 6-14: Response of contour-based junction detector to Einstein of Fig. 3-1 (a). (a): Maximum over all orientations of normalized energy. (b): Maximum over orientations of salient contour strength. Note incorrect contour completion at arrow 1. (c) and (d) show local evidence for T-junctions and X-junctions, respectively. The spurious contour at position 1 causes a transparency response in (d). Various responses are correct; arrow 2 points to T-junction formed at knot of tie, which correctly reflects the tie boundary covering the white shirt.

6.5 Propagating Local Evidence

The detection of junctions based on salient contours also provides a simple way to propagate local information obtained at the junction along the rest of the contour. We have the saliencies and linking information at every point; we can pass local evidence along the salient contours, weighed by the contour strength.

To convert the saliency values to a 0 to 1 multiplier, we pass the saliency values through the non-linearity $r(x, t, s)$ of Eq. (6.12), where x is the saliency value, and t and s are parameters which we kept fixed for all images. We introduce an extinction factor, α so that local evidence does not propagate too far.

The propagated evidence, ${E_i}^n$, at position i after n propagation iterations is:

$$ {E_i}^n = \max({E_i}^{n-1}, \max({E_i}^0, {E_j}^{n-1}\alpha)). \qquad (6.13) $$

The first maximum operation ensures that the propagated evidence at a point never decreases, and the second maximum ensures that it never falls below the local evidence.

Applying Eq. (6.13) to each of the three types of local evidence of Fig. 6-13 gives the results shown in Fig. 6-15. Image contours are properly labelled as contours containing T-junctions, which may indicate occlusion (a), contours which contain X-junctions, which may indicate transparency (b), and contours containing ψ junctions, which may indicate a surface bend. Thus, even though the pixel values for the central regions of each of the three figures are exactly the same, propagation of the local evidence at junctions correctly yields three different interpretations for the contours of the three regions.

Figure 6-16 shows system output on a photograph of a simple scene. One of the two T–junctions only appears faintly in the system output, due to a mistake by the modified salient curve finder. The other T–junction in the image is detected, and the image contour which shows evidence for occlusion is properly labelled as such.

The contours which the system finds in Fig. 6-15 are still at a relatively early stage of interpretation. The T-junction contours provide evidence for occlusion, but do not indicate which side of the contour is the far side. One must incorporate more global information for that. The X-junction contours indicate a contour which crosses others. Further processing based on image intensity levels [75, 3] is needed to

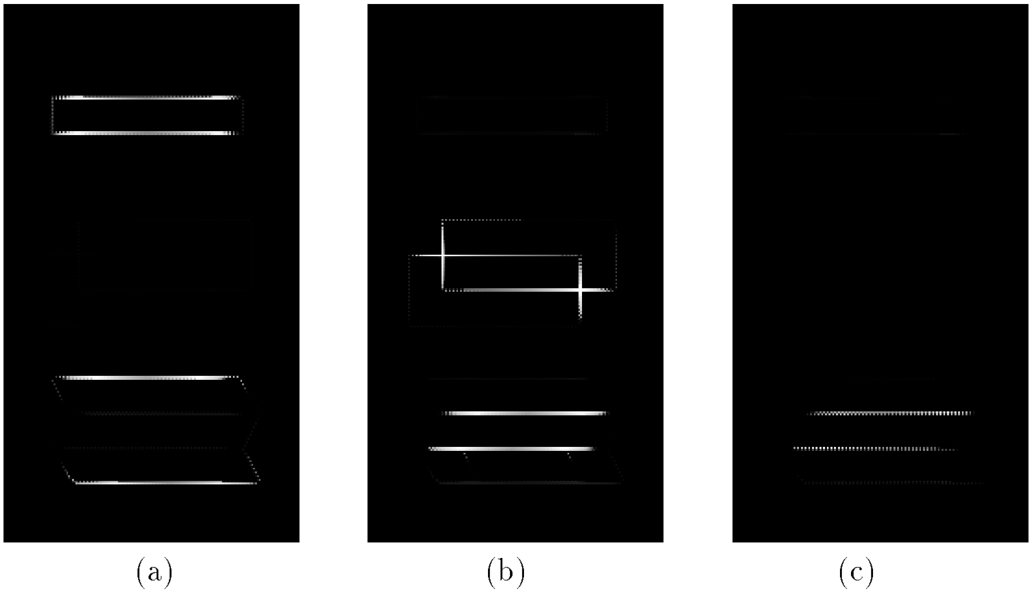


Figure 6-15: Local evidence of Fig. 6-13 propagated spatially, correctly labelling image contours. (a) Contours containing T-junctions, which may indicate occlusion. (b) Contours which contain X junctions, which may indicate transparency. (c) Contours containing ψ junctions, which may indicate bends.

ascertain whether or not the contours represent transparency.

6.6 Discussion

6.6.1 Comparison with the Work of Parent and Zucker

The salient contour analysis of this chapter bears resemblance to the work of Parent and Zucker [83]. Both begin with linear oriented filters. Our method uses pairs of filters in quadrature, and so is not restricted to contours of a particular phase. Both methods follow the linear filtering with cooperative processing stages. Parent and Zucker use relaxation labelling incorporating support for local tangency and local curvature consistency. The structural saliency algorithm of Shaashua and Ullman incorporates tangent and curvature consistency within the dynamic programming algorithm, which favors long curves of low curvature. This essentially imposes a prior statistic on the shape that curves ought to follow. Shaashua and Ullman approximate their optimum curves by storing only the choice of each orientation element at the last iteration, instead of the choice after every iteration. To that we added competition between the orientation elements. The resulting procedure can be considered a type of relaxation labelling. We and Parent and Zucker show different applications of the work. They show useful image processing applications, while we explore the use of these contours for image interpretation. They avoid a problem encountered by the method of this chapter, which has strayed from the "steerable" philosophy used throughout the rest of this thesis: artifacts from orientation quantization. They employ a linear interpolation between pixel positions, which avoids some orientation sampling artifacts which we discuss below. The method of Shaashua and Ullman, and our modification of it, restricts the heads and tails of orientation vectors to pixel sample positions.

6.6.2 Orientation Quantization Effects

A limitation of the dynamic programming approach to curve finding is the quantization of angle. To estimate the effect of this, one can compare the computed saliency of a line at two different orientations. Ideally, the saliency would be independent of line orientation. Let us assume that the first orientation of the line is parallel with one direction of orientation sampling, and that the second orientation is half-way in between two orientation samples (see Fig. 6-17). Assuming unity local saliencies, the first line, with unity coupling constants between all its links, will have a saliency $S_{\theta_1} = N$. The second line, however, has a coupling constant between each link, call it k. Following the algorithm of Eq. (6.4), its saliency will be

$$\begin{aligned} S_{\theta_2} &= 1 + \ldots + k(1 + k(1 + k(1))) && (6.14) \\ &= 1 + k + k^2 + \ldots + k^{N-1} && (6.15) \\ &= \frac{1 - k^N}{1 - k}, && (6.16) \end{aligned}$$

where the last equation is an identity for geometric sums of this form [91]. For these neighboring orientations the coupling constant $k = 0.91$, which gives $\lim_{N \to \infty} S_{\theta_2} = 11.1$. Thus, for one orientation of the line in the image, the saliency equals the number of iteration steps, N, while if we rotate that line slightly, the saliency can be no larger than 11.1! This suggests limiting curves to small values of N. We used $N = 6$.

Even using the small value of N that we did, we observe orientation dependent behavior. (The non-maximal suppression imposed in the orientation domain to avoid spurious salient contours (Sect. 6.3.1) will also contribute to this.) In the experiment of Figs. 6-18 and 6-19, we processed an image and a rotated version of itself. Ideally, the interpretation of the image would be independent of its orientation. The normalized energy responses are very nearly the same before (Fig. 6-18 (c)) and after (Fig. 6-18 (d)) rotation. However, the maximum of the saliencies, Fig. 6-18 (e) and (f), are considerably different. In the original image, some image contours line up with the orientation samples, while in the rotated version, other contours do, giving rise to the orientation dependency of contour strength. Reflecting the differences in contour strengths, the T-junction detector responses, Fig. 6-19 (c) and (d), are different. For comparison, we also show the responses of the energy-based T-junction detector of Chapter 5 in Fig. 6-19 (e) and (f). These are nearly the same in the original and rotated versions, reflecting the fact that the energy derivatives are all made relative to the measured local orientations.

6.6.3 Noise Sensitivity

One would expect the salient contour based approach to have greater robustness to noise than the local energy based method of Chapt. 5, since it integrates and smooths contours over a larger area. As shown in Fig. 6-20, the opposite is true. The energy based method gives a reasonable response at a 7 dB signal to noise ratio (SNR), while the salient contour based method only works reliable at 13 dB SNR.

This is not a fundamental result of the two classes of algorithms; the difference in noise sensitivity can be traced to the different methods used for energy normalization. The energy based approach uses a wide local average of energy to normalize the energy output. If the image is of higher contrast than the noise, the image structures will dominate the energy landscape. The salient contour approach relied on the cooperative processing to smooth energy variations and used a simple point nonlinearity to remove contrast effects. Values of noise higher than that assumed in the energy normalization function will be treated as signal. To improve the noise robustness of the salient contour method, it may be desirable to include a local gain control in the energy normalization.

6.6.4 Contours Versus Regions

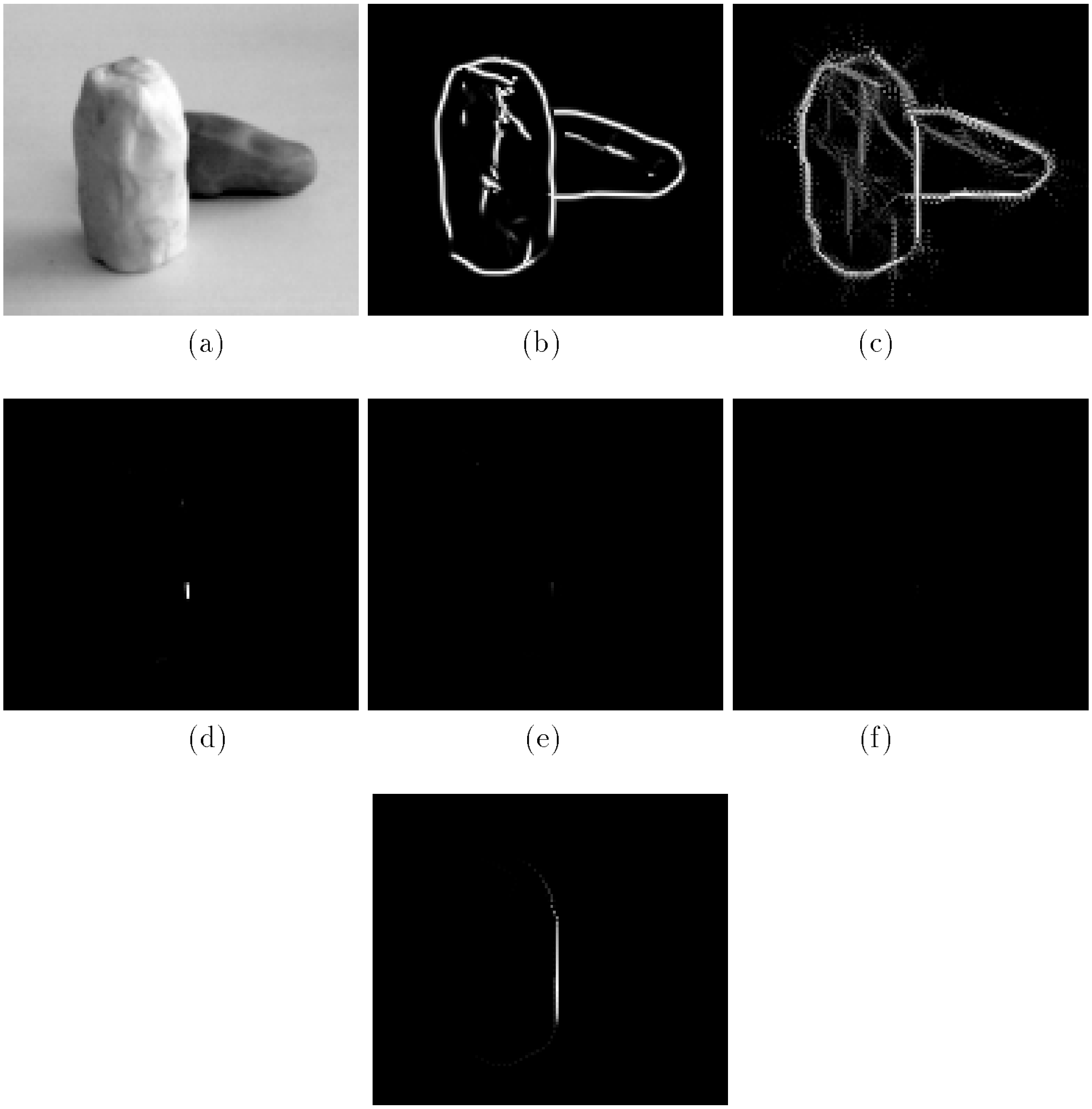
An alternative to the contour based method used in this chapter would be a region based method. Many vision problems can be approached through techniques of either class. Region based methods have been used in segmentation and surface reconstruction [15, 106, 26].

A region based approach might be quite feasible for problems such as the interpretation of Fig. 1-1. Analysis of regional image intensities could make transparency estimates, while our contour based method would have to leave those ambiguous. One could use the local oriented energy measures of Chapter 3 in the surface reconstruction algorithms of Blake and Zisserman [15], or Terzopoulos [106]. In some cases a contour based algorithm might be faster than a region based one, because of the lower dimensionality of a contour than a region. However, our contour finder used a parallel computation over the entire image, and so did not exploit any advantage from manipulating one-dimensional contours. In short, this problem is suited to an analysis based on regions as well as one based on contours. (Many researchers combine the two approaches, introducing line processes into region based schemes, or combining the outputs of independent calculations [84, 21, 15, 106].)

6.6.5 Higher Level Processing

The contour labellings which the salient contour based system produces are tentative. There will be false positive responses. For example, Fig. 6-18 (g) shows spurious evidence for occlusion caused by vertical marks on the risers of the stairs. Higher level processing is needed to confirm tentative identifications made at the low level. Higher level information will also be needed to identify junctions which cannot be identified from low level information alone.

Setting parameters is another issue which may require input from higher level processing. There are a number of parameters related to signal strength, noise level and spatial scale in each of the two systems we presented for junction identification. While all images in the thesis for each algorithm were made at the same parameter settings, it was difficult to find settings which worked well for every image. The best settings for synthetic and natural images often differ markedly. Synthetic images often have long, straight contours, which may line up with the preferred orientation directions. Natural scenes, in general, have neither. Estimates of local noise and signal strengths can be useful to set parameters (see, for example, [20]). However, the improvements obtainable through such techniques are limited. Higher level information about the important image scale, signal, and noise levels may be needed to determine proper parameter settings.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 6-16: Salient-contour-based junction analysis of simple image. (a) Image showing occlusion. (b) Local orientation evidence. (c) Maxima of modified saliency. (d), (e), (f) show the evidence for T, X, and ψ junctions, respectively. The top occluding T–junction has only faint T-ness because the curve-finder erroneously shows weak evidence for a curve with high forward and backward saliency at that point. (g) Propagation of evidence for contour containing a T–junction correctly identifies the occluding image contour.

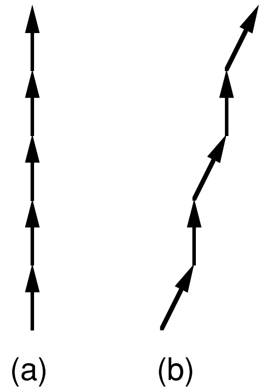


Figure 6-17: Illustrating saliency artifact due to quantization in orientation. The saliency of a straight line should be independent of its orientation. However, a line parallel with one of the orientation axes, as in (a), will have unity coupling constant between links, while one between two orientation axes could have a weaker coupling constant for every link. This worst case is analyzed in the text.

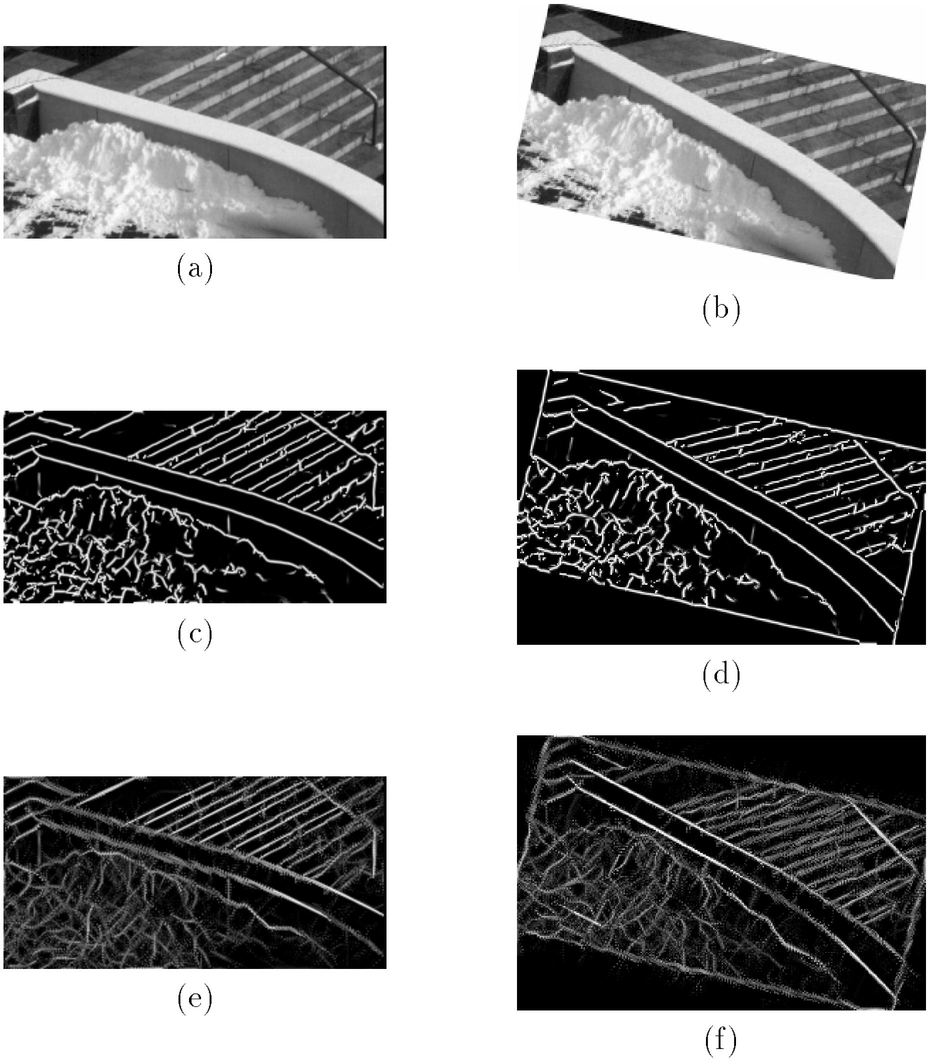


Figure 6-18: Showing dependence of processing result on orientation of image relative to quantized orientations. Input image, (a), and a rotated version of it (b) were processed. The normalized local energies are very nearly rotated versions of each other, as shown by the rotational invariance of the maximum energies at each position (c) and (d). However, the outputs of the salient contour finders, (e) and (f), are noticeably different. Contours which happen to line up with the orientation sampling structure are strong, while contours in between are weaker.

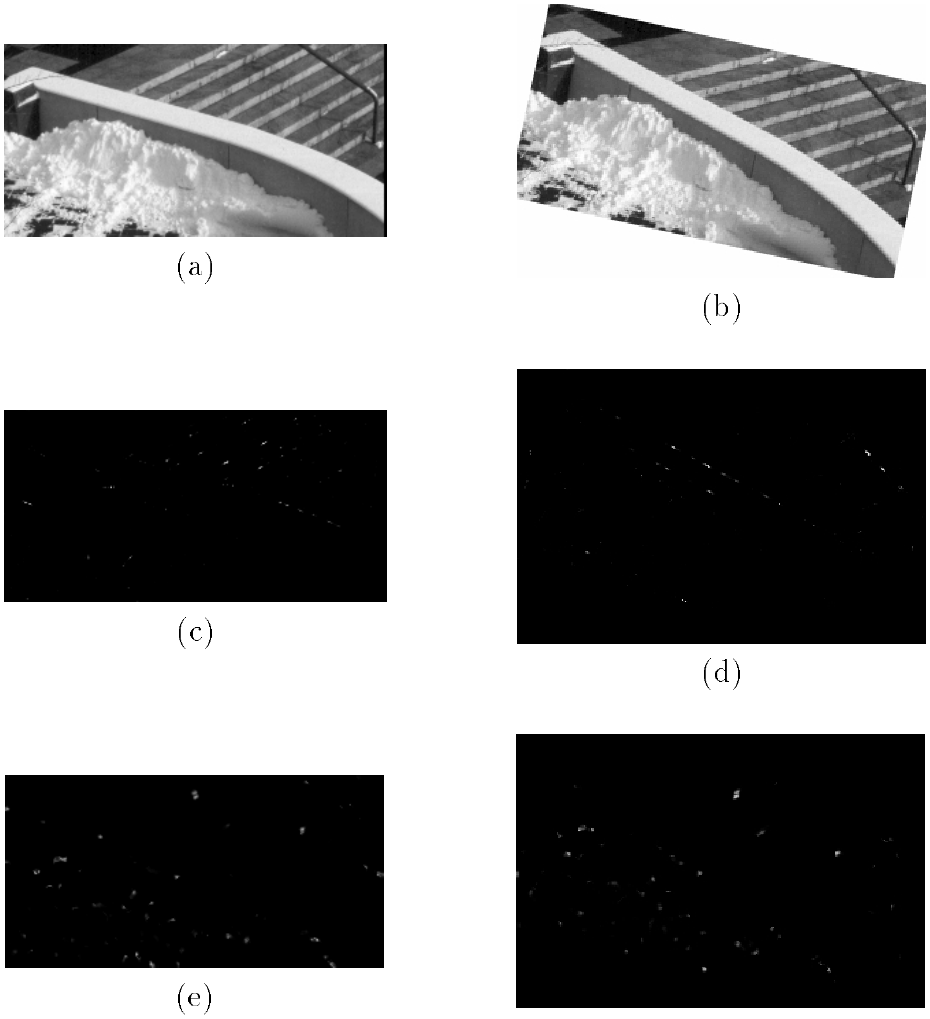
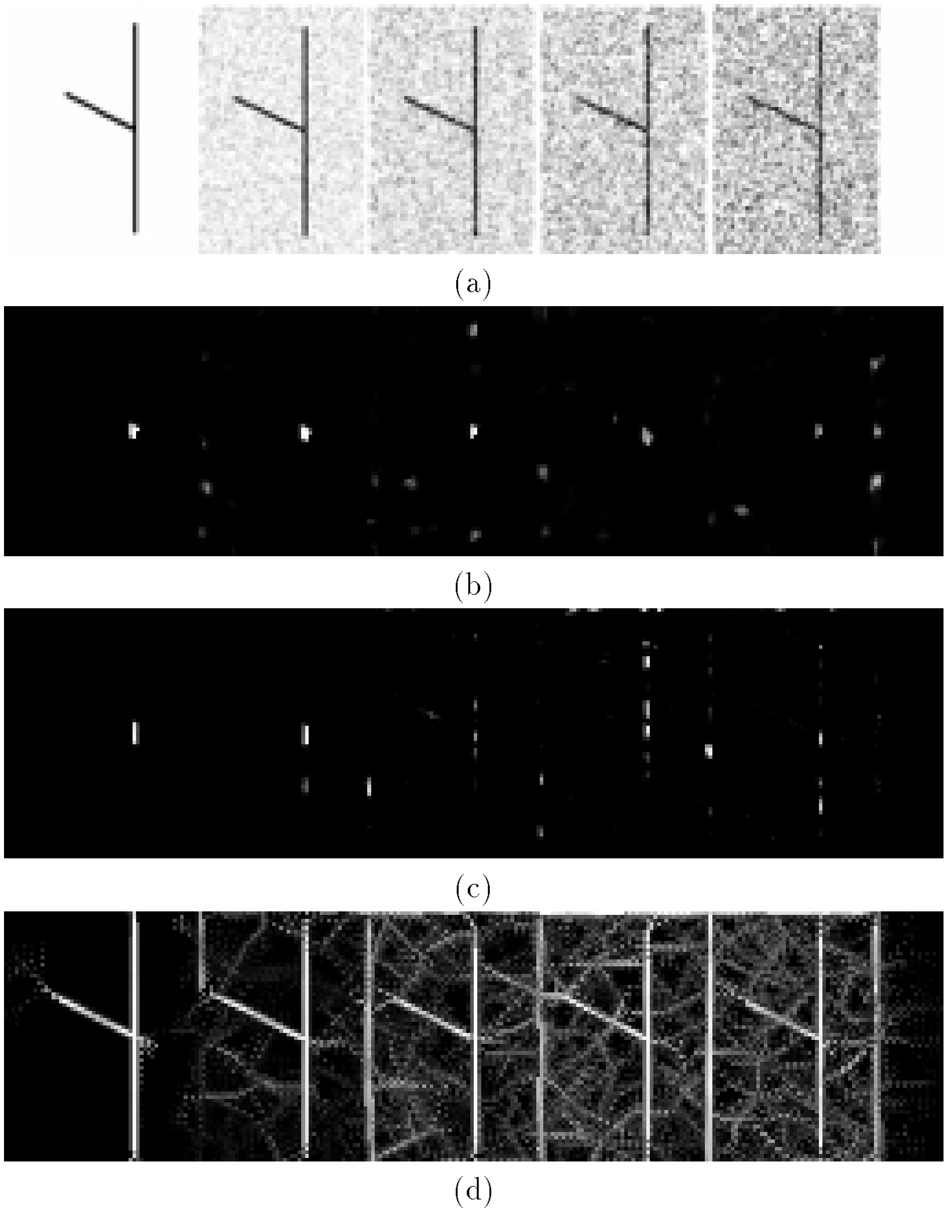


Figure 6-19: Orientation dependent results, continued from Fig. 6-18. Input images (a) and (b) are repeated, for convenience. The orientation dependence of the saliency outputs affects the junction analysis, as shown by the different outputs of the T-junction detectors, (c) and (d). For comparison, the outputs of the energy based T-junction detector of Chapter 5 are shown in (e) and (f). These T-junction measurements are substantially invariant to rotations.



(a)

(b)

(c)

(d)

Figure 6-20: Comparison of noise sensitivity of both cue detection methods. (a) T-junction test image, embedded in various levels of Gaussian random noise. From left to right, the signal to noise ratios in dB for the noisy images are: 13.3, 9.5, 7.0, 5.1 (based on signal and image variances). (b) Output of local energy based T-junction detector of Chapter 5. This shows good robustness up to high levels of noise. (c) Output of salient-contour based T-junction detector of Chapter 6. Relatively low levels of noise affect the result. This difference in results in due to the particular energy normalizations used. (d) shows the normalized oriented energy input to the salient contour calculation. Because the normalization is based on a point non-linearity, rather than on a measure of local activity, noise of a sufficiently high amplitude is treated as signal.

Chapter 7

Conclusions

The goal of this work was to develop a system to form an initial interpretation of the physical origin of observed image structures. The images of Fig. 1-1 show that a purely local interpretation of image intensities cannot correctly account for their physical origin. Different scene properties, such as occlusion, transparency, or shading, can produce identical image intensities. On the other hand, a completely global approach, where everything is analyzed in the context of everything else, is too difficult.

We chose to use a local analysis, but to analyze special local regions which reveal scene structure–image junctions. “T”-junctions can indicate occlusion; “X”-junctions can indicate transparency, and “ψ”-junctions can indicate surface normal changes. We chose a bottom-up approach with no explicit restrictions on what objects we expect to see.

Junctions form where contours meet. The junction classification depends on the relative orientation of the various contours. Therefore, to analyze image junctions we needed to detect contours and analyze orientation.

Our first step was to apply linear oriented filters. In Chapter 2 we developed a new technique, using *steerable filters*, which allows arbitrary oriented filters to be applied over a continuum of orientations. This is a computationally efficient way to apply oriented filters. It is appealing analytically, allowing explicit formulas for the filter response as a function of angle, and other derived measurements. This new technique has many applications in image processing and computer vision. Steerable filters have been used for image enhancement, motion analysis, orientation analysis, shape from shading, and image representation.

Using the steerable filters, we analyze orientation with local energy measures, applying steerable filters to a technique developed in [57]. We examine special problems of orientation analysis which arise in regions of multiple orientations. Analysis of these regions are important for motion analysis as well as static scene analysis. We develop a simple post-filter which increases the accuracy of the orientation analysis in these regions. The post-filter allows for a parsimonious use of filters for orientation analysis.

Having developed tools to analyze orientation at junctions, we studied contours. In Chapter 4 we developed a detector for image contours based on local energy measures. We studied the distribution of the phase of image contours in several images. The wide distribution found lent support to our energy based approach for contour detection.

In Chapter 5 we developed operators which responded selectively to junctions of particular types. These detectors were based on templates for cross-sections, relative to the junction orientations, of outputs of local energy measures. These operators successfully identified T, X, and L junctions in synthetic and simple natural scenes.

Local energy-based measurements of junctions can be fooled by spurious signals near contours, or contour gaps. For a more robust detector, we developed junction detectors based on salient contours. We made use of the elegant algorithm to find salient contours developed in [95]. In doing so, we strayed from the philosophy of the steerable filters and used quantized orientations. To better represent image contours and junctions, we added a competition term to the salient contour algorithm. The result was a local representation of the longer range structure of image contours. This representation was able to bridge contour gaps, and discount spurious signals near contours. The configuration of local saliencies represented more global information than the local energy measures.

We used the local configuration of salient contours to analyze T, X, and ψ junctions such as those in Fig. 1-1. The salient contours offered a simple way to propagate that local junction information along image contours. We were able to label contours as showing possible evidence for occlusion, transparency, and surface normal bend. We showed the results of this algorithm on a variety of synthetic and natural scenes.

One can continue this work in various ways. The orientation quantization of Chapter 6 caused the results of the algorithm to depend on orientation. A cooperative contour detector which treated all orientations identically would improve the junction detection and evidence propagation results. One would like to develop such a contour finder in the spirit of the steerable filters, allowing continuous variation of orientation and position. The steerable pyramid of Section 2.7 might be an appropriate representation for that task.

Bottom-up processing is only half the story. A top-down approach is more robust, and can compensate for noise or clutter which would stifle a purely bottom-up scheme. The contour identifications developed above are tentative, and need to be confirmed or disputed by higher level scene information. An important area to study is the interaction of the bottom-up and the top-down processing. Should higher-level expectations influence low-level measurements? And if so, how? We hope that the bottom-up tools and techniques we developed here will add power and generality to systems which integrate both top-down and bottom-up approaches.

Bibliography

[1] E. H. Adelson. Lightness judgments and perceptual organization. *Investigative Opthalmology and Visual Science Supplement*, page 1304, 1990. (ARVO 1990).

[2] E. H. Adelson, 1992. personal communication.

[3] E. H. Adelson and P. Anandan. Ordinal characteristics of transparency. In *AAAI Workshop on Qualitative Vision*, pages 77–81, Boston, MA, 1990.

[4] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, 1985.

[5] E. H. Adelson and A. P. Pentland. The perception of shading and reflectance. In B. Blum, editor, *Channels in the Visual Nervous System: Neurophysiology, Psychophysics, and Models*, pages 195–207. Freund Publishing, London, 1991.

[6] E. H. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. In *Proc. SPIE — Vis. Comm. and Image Proc. II*, pages 50–58, Cambridge, MA, 1987.

[7] E. H. Adelson, E. P. Simoncelli, and W. T. Freeman. Pyramids and multiscale representations. In *Proc. 13th European Conference on Visual Perception*, Paris, 1990.

[8] J. Aloimonos and D. Shulman. *Integration of Visual Modules.* Academic Press, Inc., 1989.

[9] H. H. Baker. Building surfaces of evolution: the weaving wall. *Intl. J. Comp. Vis.*, 3:51–71, 1989.

[10] D. H. Ballard and C. M. Brown. *Computer Vision*, page 43. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1982.

[11] R. H. Bartels, J. C. Beatty, and B. A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling.* Morgan Kaufmann Publishers, Inc., Los Altos, CA 94022, 1987.

[12] J. R. Bergen and E. H. Adelson. Visual texture segmentation and early vision. *Nature*, 333:363–364, 1988.

[13] D. J. Beymer. Junctions: their detection and use for grouping in images. Master's thesis, Massachusetts Institute of Technology, 1989.

[14] T. O. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17:205–244, 1981.

[15] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[16] A. C. Bovik. Multichannel texture analysis using localized spatial filters. *IEEE Pat. Anal. Mach. Intell.*, 12(1):55–73, 1990.

[17] R. N. Bracewell. *The Fourier Transform and its Applications.* McGraw-Hill, 1978.

[18] H. H. Bulthoff and H. A. Mallot. Interaction of different modules in depth perception. In *Proc. 1st Intl. Conf. Computer Vision*, pages 295 – 305, Washington, DC, 1987. IEEE.

[19] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comm.*, 31(4):532–540, 1983.

[20] J. F. Canny. A computational approach to edge detection. *IEEE Pat. Anal. Mach. Intell.*, 8(6):679–698, 1986.

[21] C. C. Chu and J. K. Aggarwal. The integration of region and edge-based segmentation. In *Proc. 3rd Intl. Conf. Computer Vision*, pages 117–120, Osaka, Japan, 1990. IEEE.

[22] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971.

[23] P. R. Cohen and E. A. Feigenbaum, editors. *The Handbook of Artificial Intelligence*, pages 139–194. Addison Wesley, 1982.

[24] P. R. Cohen and E. A. Feigenbaum, editors. *The Handbook of Artificial Intelligence*, pages 292–300. Addison Wesley, 1982.

[25] P. Danielsson and O. Seger. Rotation invariance in gradient and higher order derivative detectors. *Comp. Vis., Graphics, Image Proc.*, 49:198–221, 1990.

[26] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by minimal description. In *Proc. 3rd Intl. Conf. Computer Vision*, pages 112–116, Osaka, Japan, 1990. IEEE.

[27] J. G. Daugman. Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoust., Speech, Signal Proc.*, 36(7):1169–1179, 1988.

[28] E. A. DeYoe and D. C. V. Essen. Concurrent processing streams in monkey visual cortex. *Trends in Neurosciences*, 11(5):219–226, 1988.

[29] D. Esteban and C. Galand. Application of quadrature mirror filters to split band voice coding schemes. In *Proc. ICASSP*, pages 191 – 195, 1977.

[30] D. Fleet and A. Jepson. Computation of normal velocity from local phase information. In *Proc. IEEE CVPR*, pages 379–386, San Diego, CA, 1989.

[31] W. T. Freeman and E. H. Adelson. Steerable filters. In *Topical Meeting on Image Understanding and Machine Vision.* Optical Society of America, June 1989. Technical Digest Series Volume 14.

[32] W. T. Freeman and E. H. Adelson. Steerable filters for early vision, image analysis, and wavelet decomposition. In *Proc. 3rd Intl. Conf. Computer Vision*, pages 406 – 415, Osaka, Japan, 1990. IEEE.

[33] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Pat. Anal. Mach. Intell.*, 13(9), September 1991.

[34] W. T. Freeman and E. H. Adelson. Junction detection and classification. *Investigative Opthalmology and Visual Science Supplement*, page 1279, 1991. (ARVO 1991).

[35] W. T. Freeman and E. H. Adelson. Steerable filters for image processing and analysis. In *Proc. 7th Workshop on Multidimensional Signal Processing*, page 9.8, Lake Placid, New York, 1991. IEEE Signal Processing Society.

[36] W. T. Freeman, E. H. Adelson, and D. J. Heeger. Motion without movement. *ACM Computer Graphics (SIGGRAPH '91)*, 25(4):27–30, 1991. Las Vegas, Nevada.

[37] W. T. Freeman, E. H. Adelson, and A. P. Pentland. Shape-from-shading analysis with bumplets and shadelets. *Investigative Opthalmology and Visual Science Supplement*, page 410, 1990. (ARVO 1990).

[38] G. H. Granlund. In search of a general picture processing operator. *Comp. Graphics, Image Proc.*, 8:155–173, 1978.

[39] S. Grossberg and E. Mingolla. Neural dynamics of form perception: boundary completion, illusory figures, and neon color spreading. *Psychological Review*, 92(2):173–211, 1985.

[40] S. Grossberg and E. Mingolla. Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. *Perception and Psychophysics*, 38(2):141–171, 1985.

[41] A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math.*, 15:723–736, 1984.

[42] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. In *AFIPS Fall Joint Conferences*, pages 291–304, 1968. volume 33.

[43] R. M. Haralick. The digital step edge from zero crossings of second directional derivatives. *IEEE Pat. Anal. Mach. Intell.*, 6(1):58–68, 1984.

[44] M. Hedlund, G. H. Granlund, and H. Knutsson. A consistency operation for line and curve enhancement. In *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pages 93–96, 1982.

[45] D. J. Heeger. Optical flow using spatiotemporal filters. *Intl. J. Comp. Vis.*, 1(4):279–302, 1988.

[46] D. J. Heeger. Nonlinear model of neural responses in cat visual cortex. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, chapter 9. MIT Press, Cambridge, MA, 1991.

[47] D. J. Heeger, W. T. Freeman, and E. H. Adelson. Nude descending a staircase. In *Physics Art* at the California Museum of Science and Industry (Los Angeles, summer 1989), and in SIGGRAPH '90 Art Show (Dallas, August 1990).

[48] F. Heitger, L. Rosenthaler, R. von der Heydt, E. Peterhans, and O. Kubler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, 32:963–981, 1992.

[49] J. E. Hochberg. *Perception.* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1978.

[50] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to probability theory.* Houghton Mifflin Company, 1971.

[51] D. A. Huffman. Impossible objects as nonsense sentences. In R. Meltzer and D. Michie, editors, *Machine intelligence 6*, pages 295–323. Elsevier, New York, 1971.

[52] R. A. Hummel and S. W. Zucker. On the foundation of relaxation labeling processes. *IEEE Pattern Analysis and Machine Intelligence*, 5(3):267–287, 1983.

[53] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Intl. Conf. Computer Vision*, pages 321 – 331. IEEE, 1987.

[54] M. Kass and A. P. Witkin. Analyzing oriented patterns. In *Proc. Ninth IJCAI*, pages 944–952, Los Angeles, CA, August 1985.

[55] D. Kersten. Transparancy and the cooperative computation of scene attributes. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, chapter 15. MIT Press, Cambridge, MA, 1991.

[56] D. C. Knill and D. Kersten. Apparent surface curvature affects lightness perception. *Nature*, 351:228–230, 1991.

[57] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213, 1983.

[58] H. Knutsson, L. Haglund, and H. Barman. A tensor based approach to structure analysis and enhancement in 2d, 3d, and 4d. In *Proc. 7th Workshop on*

Multidimensional Signal Processing, page 9.10, Lake Placid, New York, 1991. IEEE Signal Processing Society.

[59] H. Knutsson, L. Haglund, and G. H. Granlund. Tensor field controlled image sequence enhancement. In *SSAB Symposium on Image Analysis*, Linkoping, Sweden, March 1990.

[60] H. Knutsson, R. Wilson, and G. H. Granlund. Anisotropic nonstationary image estimation and its applications: Part 1 – Restoration of noisy images. *IEEE Trans. Comm.*, 31(3):388–397, 1983.

[61] J. J. Koenderink. Design for a sensorium. In W. von Seelen, B. Shaw, and U. M. Leinhos, editors, *Organization of Neural Networks*, pages 185–207. Verlagsgesellschaft mbH, 1988.

[62] J. J. Koenderink. Operational significance of receptive field assemblies. *Biol. Cybern.*, 58:163–171, 1988.

[63] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367–375, 1987.

[64] J. Lim. *Two-Dimensional Signal and Image Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[65] D. G. Lowe. *Perceptual organization and visual recognition.* Kluwer Academic Publishers, Boston, MA, 1985.

[66] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[67] D. G. Lowe. Integrated treatment of matching and measurement errors for robust model-based motion tracking. In *Proc. 3rd Intl. Conf. Computer Vision*, pages 436 – 444, Osaka, Japan, 1990. IEEE.

[68] D. G. Lowe and T. O. Binford. The interpretation of three-dimensional structure from image curves. In *Proc. Seventh IJCAI*, pages 613–618, Vancouver, 1981.

[69] D. G. Lowe and T. O. Binford. The recovery of three-dimensional structure from image curves. *IEEE Pat. Anal. Mach. Intell.*, 7(3):320–326, 1985.

[70] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7:923–931, 1990.

[71] S. G. Mallat. A theory for multi-resolution signal decomposition: the wavelet representation. *IEEE Pat. Anal. Mach. Intell.*, 11(47):674–693, 1989.

[72] D. C. Marr. *Vision.* W H Freeman and Company, 1982.

[73] J. B. Martens. Applications of polynomial transforms in image coding and computer vision. In *Proceedings of SPIE*, volume 1199, pages 1279–1290, Cambridge, MA, 1989.

[74] J. B. Martens. The Hermite transform – theory. *IEEE Trans. Acoust., Speech, Signal Proc.*, 38(9):1595–1606, 1990.

[75] F. Metelli. The perception of transparency. *Scientific American*, 1974.

[76] M. C. Morrone and D. C. Burr. Feature detection in human vision: a phase-dependent energy model. *Proc. R. Soc. Lond. B*, 235:221–245, 1988.

[77] M. C. Morrone and R. A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, 1987.

[78] P. M. Morse and H. Feshback. *Methods of Theoretical Physics*, volume 1. McGraw-Hill, 1953.

[79] J. A. Movshon, I. D. Thompson, and D. J. Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J. Physiol.*, 283:53–77, 1978.

[80] D. Mumford. Elastica and computer vision. unpublished manuscript, Harvard University, 1990.

[81] M. Nitzberg. *Depth from overlap.* PhD thesis, Harvard University, 1991.

[82] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[83] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Pat. Anal. Mach. Intell.*, 11(8):823–839, 1989.

[84] T. Pavlidis and Y. T. Liow. Integrating region growing and edge detection. *IEEE Pat. Anal. Mach. Intell.*, 12(3):225–233, 1990.

[85] A. P. Pentland. Local shading analysis. *IEEE Pat. Anal. Mach. Intell.*, 6(2):170–187, 1984.

[86] A. P. Pentland. Linear shape from shading. *Intl. J. Comp. Vis.*, 1(4):153–162, 1990.

[87] P. Perona. Deformable kernels for early vision. Technical Report MIT-LIDS-P-2039, MIT Laboratory for Information and Decision Sciences, MIT, Cambridge, MA 02139, 1991.

[88] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. In *Proceedings of European Conference on Computer Vision*, 1992.

[89] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. 3rd Intl. Conf. Computer Vision*, Osaka, Japan, 1990.

[90] T. Poggio, E. B. Gamble, and J. J. Little. Parallel integration of vision modules. *Science*, 242:436–440, 1989.

[91] M. Protter and J. C. Morrey. *Modern Mathematical Analysis.* Addison-Wesley, Reading, MA, 1964.

[92] K. Rangarajan, M. Shah, and D. V. Brackle. Optimal corner detector. *Comp. Vis., Graphics, Image Proc.*, 48:230–245, 1989.

[93] I. Rock. *Perception.* Scientific American Library, 1984.

[94] T. Sanger. Stereo disparity computation using Gabor filters. *Biol. Cybern.*, 59:405–418, 1988.

[95] A. Shaashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *2nd International Conference on Computer Vision (ICCV)*, pages 321–327, December 1988.

[96] A. Shaashua and S. Ullman. Grouping contours by iterated pairing network. In *Advances in neural information processing systems 3 (NIPS)*, December 1990.

[97] S. A. Shafer. Optical phenomena in computer vision. Computer Science Technical Report 135, Computer Science Department, University of Rochester, Rochester, NY 14627, 1984.

[98] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *Proceedings of the tenth international conference on pattern recognition*, pages 274–278, 1990.

[99] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *Computer vision and pattern recognition '91*, pages 289–295, 1991.

[100] E. P. Simoncelli and E. H. Adelson. Non-separable extensions of quadrature mirror filters to multiple dimensions. *Proc. IEEE*, 78(4):652–664, 1990.

[101] E. P. Simoncelli and E. H. Adelson. Subband transforms. In J. W. Woods, editor, *Subband Image Coding*, chapter 4. Kluwer Academic Publishers, Norwell, MA, 1990.

[102] E. P. Simoncelli and E. H. Adelson. Relationship between gradient, spatio-temporal energy, and regression models for motion perception. *Investigative Opthalmology and Visual Science Supplement*, page 893, 1991. (ARVO 1991).

[103] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans. Information Theory*, 2(38):587–607, 1992.

[104] P. Sinha. The perception of shading and reflectance. Master's thesis, Massachusetts Institute of Technology, 1992. Electrical Engineering and Computer Science.

[105] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-level Vision.* Kluwer Academic Publishers, Boston, 1989.

[106] D. Terzopoulos. Regularization of inverse problems involving discontinuities. *IEEE Pat. Anal. Mach. Intell.*, 8(4):413–424, 1986.

[107] M. Vetterli. Multidimensional subband coding: some theory and algorithms. *Signal Processing*, 6(2):97–112, 1984.

[108] D. Waltz. Generating semantic descriptions from drawings of scenes with shadows. In P. Winston, editor, *The psychology of computer vision*, pages 19–92. McGraw-Hill, New York, 1975.

[109] A. B. Watson. The cortex transform: Rapid computation of simulated neural images. *Comp. Vis., Graphics, Image Proc.*, 39:311–327, 1987.

[110] L. R. Williams. Perceptual organization of occluding contours. In *Proc. 3rd Intl. Conf. Computer Vision*, Osaka, Japan, 1990. IEEE.

[111] H. R. Wilson. A psychophysically motivated model for two-dimensional motion perception. *Investigative Opthalmology and Visual Science Supplement*, page 893, 1991. (ARVO 1991).

[112] R. Wilson and A. H. Bhalerao. Kernel designs for efficient multiresolution edge detection and orientation estimation. *IEEE Pat. Anal. Mach. Intell.*, 14(3):384–390, 1992.

[113] A. P. Witkin. Intensity-based edge classification. In *Proc. National Conf. on AI*, pages 36–41, Pittsburgh, PA, August 1982.

[114] A. P. Witkin. Scale space filtering. In *Proc. 8th Intern. Joint Conf. Artif. Intell.*, pages 1019–1021, Karlsruhe, Germany, August 1983.

[115] J. W. Woods and S. D. O'Neil. Subband coding of images. *IEEE Trans. Acoust., Speech, Signal Proc.*, 34(5):1278–1288, 1986.

[116] R. A. Young. Simulation of human retinal function with the Gaussian derivative model. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 564–569, 1986.

[117] S. W. Zucker. Early orientation selection: Tangent fields and the dimensionality of their support. *Comp. Vis., Graphics, Image Proc.*, 32:74–103, 1985.