THREE-DIMENSIONAL COMPUTER VISION

A GEOMETRIC VIEWPOINT

OLIVIER FAUGERAS

Olivier Faugeras

Three-Dimensional Computer Vision

A Geometric Viewpoint

The MIT Press
Cambridge, Massachusetts
London, England

This book was formatted in ZzTEX by Paul Anagnostopoulos and typeset by Joe Snowden. The typeface is Lucida Bright and Lucida New Math, created by Charles Bigelow and Kris Holmes specifically for scientific and electronic publishing. The Lucida letterforms have the large x-heights and open interiors that aid legibility in modern printing technology, but also echo some of the rhythms and calligraphic details of lively Renaissance handwriting. Developed in the 1980s and 1990s, the extensive Lucida typeface family includes a wide variety of mathematical and technical symbols designed to harmonize with the text faces.

Library of Congress Cataloging-in-Publication Data

Faugeras, Olivier, 1949–
Three-dimensional computer vision : a geometric viewpoint /
Olivier Faugeras.
p. cm. — (Artificial intelligence)
Include bibliographical references and index.
ISBN 0-262-06158-9
1. Computer vision. I. Title. II. Series: Artificial
intelligence (Cambridge, Mass.)
TA1632.F38 1993
006.3'7—dc20 93-9126
CIP

10 9 8 7 6 5

Contents

Series Foreword

Artificial intelligence is the study of intelligence using the ideas and methods of computation. Unfortunately a definition of intelligence seems impossible at the moment because intelligence appears to be an amalgam of so many information-processing and information-representation abilities.

Of course psychology, philosophy, linguistics, and related disciplines offer various perspectives and methodologies for studying intelligence. For the most part, however, the theories proposed in these fields are too incomplete and too vaguely stated to be realized in computational terms. Something more is needed, even though valuable ideas, relationships, and constraints can be gleaned from traditional studies of what are, after all, impressive existence proofs that intelligence is in fact possible.

Artificial intelligence offers a new perspective and a new methodology. Its central goal is to make computers intelligent, both to make them more useful and to understand the principles that make intelligence possible. That intelligent computers will be extremely useful is obvious. The more profound point is that artificial intelligence aims to understand intelligence using the ideas and methods of computation, thus offering a radically new and different basis for theory formation. Most of the people doing work in artificial intelligence believe that these theories will apply to any intelligent information processor, whether biological or solid state.

There are side effects that deserve attention, too. Any program that will successfully model even a small part of intelligence will be inherently massive and complex. Consequently artificial intelligence continually confronts the limits of computer-science technology. The problem

The proposal was accepted, and this was the beginning of Esprit P940. Surprisingly enough, we achieved most of the objectives we had outlined in the proposal, and the final review of the project was successfully held in January 1992. Since many of the ideas that are described in this book have matured and developed during the progress of this effort, it seems natural to publish the book now that the project has come to a successful end.

The book is mostly about my own work and that of my collaborators and students within my research group at INRIA, in particular Nicholas Ayache, Jean-Daniel Boissonnat, Rachid Deriche, Martial Hébert, Elizabeth Lebras-Mehlman, Francis Lustman, Théo Papadopoulo, Luc Robert, Michel Schmitt, Giorgio Toscani, Régis Vaillant, and Zhengyou Zhang. It also includes information on some work that was done jointly with Steve Maybank from GEC.

During the course of the writing, I have benefited from many stimulating discussions with Nassir Navab, Tuan Luong, Peter Sander, and Thierry Vieville from my group at INRIA, as well as with Vincent Torre and Alessandro Verri from the University of Genoa; Giovanni Garibotto; Stefano Masciangiello from Elsag, Thomas Skordas from ITMI; Philippe Isambert and Eric Théron from MSII; Gérard Gaillat, formally from Matra; and Bernard Buxton from GEC.

I would also like to acknowledge the help of Giorgio Musso and Giovanni Garibotto from Elsag and that of the INRIA administration for their role in keeping the administrative maze that is sometimes generated by European projects to a minimum. This effort has been essential in bringing P940 to a successful end and allowing me to spend most of my time on the research part of the project.

Finally, and most important, I am extremely grateful to my wife, Agnès, and our sons Blaise, Clément, Cyrille, and Quentin. Completing this book took much more time than I thought it would, and I very much appreciate their patience and support.

Sophia-Antipolis, June 1992

In this general framework, this book is an attempt to propose solutions to the problems arising from the following scenario: A mobile platform must move about in an unknown indoor environment with capabilities allowing it to do the following:

1. Avoid static and mobile obstacles.
2. Build models of objects and places in order to be able to recognize and locate them.
3. Characterize its own motion and that of moving objects by providing descriptions of the corresponding three-dimensional motions.

We think that these tasks are in many respects generic in the sense that a large number of robotics systems should be able to perform them in order to be able to interact with fairly unconstrained environments. This hypothesized genericity has the consequence that this book is not only a set of solutions to specific problems, but that many of the ideas in it are general and can be used in different settings. The book can therefore also be read as a general book on computer vision.

We have stressed the mathematical soundness of our ideas in all our our approaches at the risk of scaring some readers away. We strongly believe that detailing the mathematics is worth the effort, for it is only through their use that computer vision can be established as a science. But there is a great danger that computer vision will become yet another area of applied mathematics. We are convinced that this danger can be avoided if we keep in mind the original goal of designing and building robotics systems that perceive and act. We believe that the challenge is big enough that computer vision can become, perhaps like physics, a rich source of inspiration and problems for mathematicians.

Another key feature of this book is the importance of geometry, in particular three-dimensional geometry. This is because the world where robots move and act is, like ours, three-dimensional, that we have decided to expend so much effort on describing three-dimensional geometry and its fascinating relationship with the imaging process by which many key three-dimensional features are distorted in an intricate manner.

We have also devoted a great deal of attention to the problem of uncertain data. Even though we think that geometry must play a crucial role in computer vision systems, this geometry has to be built from noisy mea-

is interested in this subject may find it profitable to read the beautifully written book by Semple and Kneebone [SK52].

2.1 How to read this chapter

Section 2.2 is a general introduction to projective spaces and can be read quickly the first time to get an idea of the concepts involved. Propositions 2.1 and 2.2 are fundamental in the sense that they give a practical way of changing coordinate systems in a projective space, an operation that must be done more often than we may wish. Theorem 2.1 is absolutely essential for understanding chapter 5.

Section 2.3 should allow the reader to develop an intuition about projective spaces since we are studying the simplest of them, the projective line. We introduce two fundamental concepts: the point at infinity, which is the key to understanding the relationship between the projective spaces and the usual affine spaces with which we are more familiar, and the cross-ratio, which is one of the most useful invariants.

Section 2.4 takes us one dimension higher to the projective plane. Since we will often model an image as "living" in a projective plane, this section is important to read and understand. We introduce four fundamental concepts. First is the principle of duality by which points and lines are essentially equivalent. This duality affords a systematic way of transferring proofs that have been made for points to lines and vice versa. Second is the line at infinity, which plays the same role as the point at infinity of the projective line in helping us to understand the relationship between the projective plane and the more familiar affine plane. Third is the cross-ratio of four lines intersecting at a point, which, like the cross-ratio of four points on a line, is one of the most useful invariants. Fourth is the idea of a pencil of lines, which is essential to understanding the epipolar geometry we use in the stereo analysis that is done in chapter 6. The material on conics is used both in chapter 3 to interpret the intrinsic parameters of a camera and in chapter 7 to prove the correctness of the five-point algorithm; it can be skipped on the first reading.

The last part of section 2.4 shows that there are strikingly simple relationships between the usual affine and euclidean planes and the projective plane. They arise from the choice of a special line, called the *line*

and we can take $\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}$ and $\rho_i = \frac{\mu_i}{\lambda_i}$. Furthermore, if $\mathbf{P}\mathbf{x}_i = \rho_i \mathbf{y}_i$ and $\mathbf{Q}\mathbf{x}_i = \sigma_i \mathbf{y}_i$, then $\mathbf{P}\mathbf{A}\mathbf{e}_i = \lambda_i \rho_i \mathbf{y}_i$ and $\mathbf{Q}\mathbf{A}\mathbf{e}_i = \mu_i \rho_i \mathbf{y}_i$ and hence, by the previous proposition, $\mathbf{P}\mathbf{A} = \tau\mathbf{Q}\mathbf{A}$, i.e., $\mathbf{P} = \tau\mathbf{Q}$ for some scalar τ. ■

This proposition shows that a collineation is defined by $n + 2$ pairs of corresponding points. We will use this property many times in the next chapters.

2.2.4 The relationship between $\mathcal{P}^m$ and the unit sphere S^m of R^{m+1}

Here we state a theorem that will turn out to be extremely useful in chapter 5.

Theorem 2.1
The space $\mathcal{P}^m$ is topologically equivalent to the unit sphere S^m of R^{m+1} in which we have identified antipodal points.

Proof This proof can be found in all books on algebraic topology, for example in the book by Greenberg and Harper [GH81]. We can develop an intuition about what is going on as follows. A point x of S^m is represented by a vector $\mathbf{x} = [x_1, \ldots, x_{n+1}]^T$ such that $\sum_{i=1}^{n+1} x_i^2 = 1$. This also represents a point of $\mathcal{P}^m$. The vector $-\mathbf{x}$ represents the antipodal point of x, which is also on S^m but represents the *same* point of $\mathcal{P}^m$. ■

In particular, this theorem says that all projective spaces are compact spaces. This comes as a bit of a surprise since we all know, at least vaguely, that projective spaces are about points at infinity and that compact subsets of R^n are bounded. But we should not follow our intuition. Projective spaces are indeed compact, although the attentive reader will have no doubt realized that the kind of compact space we are talking about, a sphere that has been folded upon itself by identifying antipodal points, is by no means easy to picture in the mind's eye.

In chapter 5 we will also study the differential structure of the space $\mathcal{P}^n$ and find it very simple, almost as simple as that of R^{m+1}. The importance of this theorem is due to the fact that the folded unit spheres of R^2 and R^3 very naturally appear in the problems of representing directions; the folded unit sphere of R^4 appears in the problem of using quaternions for representing three-dimensional rotations. Since $\mathcal{P}^n$ is simpler to use

2.4.2 The line at infinity

Among all possible lines, the one whose equation is $x_3 = 0$ is called the *line at infinity* of $\mathcal{P}^2$, denoted by l_∞. The reason for this terminology is that we think of the projective plane as containing the usual affine plane under the correspondence $[X_1, X_2]^T \rightarrow [X_1, X_2, 1]^T$ or $X_1\mathbf{e}_1 + X_2\mathbf{e}_2 + \mathbf{e}_3$. This is a one-to-one correspondence between the affine plane and the projective plane minus the line of equation $x_3 = 0$. For each projective point of coordinates (x_1, x_2, x_3) that is not on that line, we have

$$X_1 = \frac{x_1}{x_3} \quad X_2 = \frac{x_2}{x_3} \tag{2.4}$$

If $X_1 \rightarrow \infty$ while X_2 does not, we obtain $\mathbf{e}_1$, which is on l_∞. Similarly, when $X_2 \rightarrow \infty$ while X_1 does not, we obtain $\mathbf{e}_2$.

Each line in the projective plane of the form of equation (2.3) intersects l_∞ at the point $(-u_2, u_1, 0)$, which is that line's point at infinity. Note that the vector $[-u_2, u_1]^T$ gives the direction of the affine line of equation $u_1X_1 + u_2X_2 + u_3 = 0$. This gives us a neat interpretation of the line at infinity: Each point on that line, with coordinates $(x_1, x_2, 0)$, can be thought of as a direction in the underlying affine plane, the direction parallel to the vector $[x_1, x_2]^T$. Indeed, it does not matter if x_1 and x_2 are defined only up to a scale factor since the direction does not change. We will use this observation in chapter 5 when we discuss the problem of representing two-dimensional directions.

Another useful property is the following:

Proposition 2.4
The representation of the point of intersection of two distinct projective lines is the cross-product of their representations.

Proof To see this, simply apply the principle of duality. We have seen that the representation of the line going through two points is the cross-product of the representation of those points, and this implies that the representation of the point of intersection of two lines is the cross-product of their representations. For an alternate proof see problem 4. ■

Note that this implies that, in projective geometry, two distinct lines always intersect.

it is equal to 2, we can choose the matrix **P** in such a way that the conic equation becomes

$$d_1x_1^2 + d_2x_2^2 = 0$$

which can then be factored as

$$(\sqrt{d_1}x_1 + \sqrt{-d_2}x_2)(\sqrt{d_1}x_1 - \sqrt{-d_2}x_2) = 0$$

The conic is therefore a pair of lines (real or complex) intersecting at the point represented by $\mathbf{e}_3$.

If the rank is equal to 1, the same reasoning shows that the equation can be written

$$d_1x_1^2 = 0$$

The conic is therefore reduced to a single line taken twice.

2.4.7 Affine transformations of the plane

We have seen that there is a one-to-one correspondence between the usual affine plane and the projective plane minus the line at infinity. In the affine plane, we know that an affine transformation defines a correspondence $\mathbf{X} \rightarrow \mathbf{X}'$, which can be expressed in matrix form as

$$\mathbf{X}' = \mathbf{B}\mathbf{X} + \mathbf{b} \tag{2.7}$$

where **B** is a 2×2 matrix of rank 2, and **b** is a 2×1 vector. From this equation it is clear that these transformations form a group called the *affine group*, which is a subgroup of the projective group. This subgroup has the interesting property that it preserves the line at infinity.

Let **A** be the matrix of a collineation that leaves l_∞ invariant. The matrix **A** can be written as

$$\mathbf{A} = \begin{bmatrix} \mathbf{C} & \mathbf{c} \\ \mathbf{0}_2^T & a_{33} \end{bmatrix}$$

where **C** is a 2×2 matrix and **c** is a 2×1 vector. The condition that the rank of **A** is 3 implies that $a_{33} \neq 0$ and the rank of **C** is equal to 2. Using the equations (2.4) we can write equation (2.7) with $\mathbf{B} = \frac{1}{a_{33}}\mathbf{C}$ and $\mathbf{b} = \frac{1}{a_{33}}\mathbf{c}$.

$$l_{ij} = x_i^{(1)} x_j^{(2)} - x_j^{(1)} x_i^{(2)} \qquad i,j = 1,\ldots,4$$

Since $l_{ij} = -l_{ji}$, there are only six of these numbers that are apparently independent, for example the six numbers

$$l_{41}, l_{42}, l_{43}, l_{23}, l_{31}, \text{ and } l_{12}$$

which we can take as the components of the coordinate vector $\mathbf{l}$ of l.

These six numbers are not really independent, though, since their ratio is independent of the choice of the points P_1 and P_2. Indeed, let us define

$$\mathbf{y}^{(1)} = \alpha \mathbf{x}^{(1)} + \beta \mathbf{x}^{(2)}$$

$$\mathbf{y}^{(2)} = \alpha' \mathbf{x}^{(1)} + \beta' \mathbf{x}^{(2)}$$

Then it is easy to show that the line coordinates m_{ij} that they define satisfy

$$m_{ij} = (\alpha\beta' - \alpha'\beta) l_{ij}$$

This shows that the ratios of the l_{ij} are constant.

There is a further relation between the line coordinates that can be obtained by noticing that the 4×4 determinant $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ is identically 0. Therefore, we obtain the identity

$$S(\mathbf{l}) = l_{41} l_{23} + l_{42} l_{31} + l_{43} l_{12} = 0 \tag{2.11}$$

The six numbers l_{ij} that are connected by the relation (2.11) are referred to as the *Grassmann or Plücker coordinates* of the line (see problem 7). If we think of the vector $\mathbf{l}$ as the coordinate vector of a point in $\mathcal{P}^5$, equation (2.11) can be interpreted as the equation of a surface of degree 2 on which all points of $\mathcal{P}^5$ that represent lines of $\mathcal{P}^3$ must lie.

This representation of lines is useful in many respects. As an example, let us consider two lines l and l' represented by their Plücker coordinates $\mathbf{l}$ and $\mathbf{l}'$. The following proposition is true:

Proposition 2.5

A necessary and sufficient condition for the two lines l and l' to intersect is that their Plücker coordinates $\mathbf{l}$ and $\mathbf{l}'$ satisfy the equation

$$S(\mathbf{l}, \mathbf{l}') = (l_{41} l'_{23} + l'_{41} l_{23}) + (l_{42} l'_{31} + l'_{42} l_{31}) + (l_{43} l'_{12} + l'_{43} l_{12}) = 0 \tag{2.12}$$

2.5.6.3 Intersection of a quadric with a plane

Consider a quadric S and a plane π. We assume that we have changed coordinates so that the equation of the plane is $x_4 = 0$. The intersection is a curve given by

$$\sum_{i,j=1}^{4} a_{ij}x_i x_j = 0 = x_4$$

and is therefore a conic in π. If S is proper, this conic is degenerate if and only if the plane is tangent to S.

2.5.6.4 Intersection of a quadric with a line

Let Q and R be two points of the space represented by $\mathbf{y}$ and $\mathbf{z}$, respectively. A variable point on the line $\langle Q, R\rangle$ is represented by $\mathbf{y} + \theta\mathbf{z}$, and this point lies on the quadric S if and only if

$$S(\mathbf{y} + \theta\mathbf{z}) = 0$$

We can expand this and group terms of similar degrees in θ as follows:

$$S(\mathbf{y}) + 2\theta S(\mathbf{y}, \mathbf{z}) + \theta^2 S(\mathbf{z}) = 0 \tag{2.13}$$

where

$$S(\mathbf{y}, \mathbf{z}) = \mathbf{y}^T \mathbf{A}\mathbf{z}$$

Therefore, in general, there are two points of intersection of the line $\langle Q, R\rangle$ with the quadric S. These points can be real or complex, distinct or identical, and are obtained by solving the quadratic equation (2.13).

2.5.6.5 The tangent cone to a quadric from a point

The line $\langle Q, R\rangle$ is tangent to S when the discriminant of the quadratic equation (2.13) is equal to 0:

$$S(\mathbf{y}, \mathbf{z})^2 - S(\mathbf{y})S(\mathbf{z}) = 0 \tag{2.14}$$

This equation has an interesting interpretation. If we keep Q fixed and let R vary while keeping the line $\langle Q, R\rangle$ tangent to S, then equation(2.14), when considered as an equation in $\mathbf{z}$, is the equation of the tangents to S

d. Give the inverse of a transformation defined in problem 8.

e. Conclude that the set of transformations defined in problem 8 and those of question a form a group.

10. Verify the Laguerre formula for planes.

11. Could you suggest (and prove) a projective formula (i.e., with a cross-ratio in it) for the angle of two lines in 3-D space?

12. Prove equation (2.16).

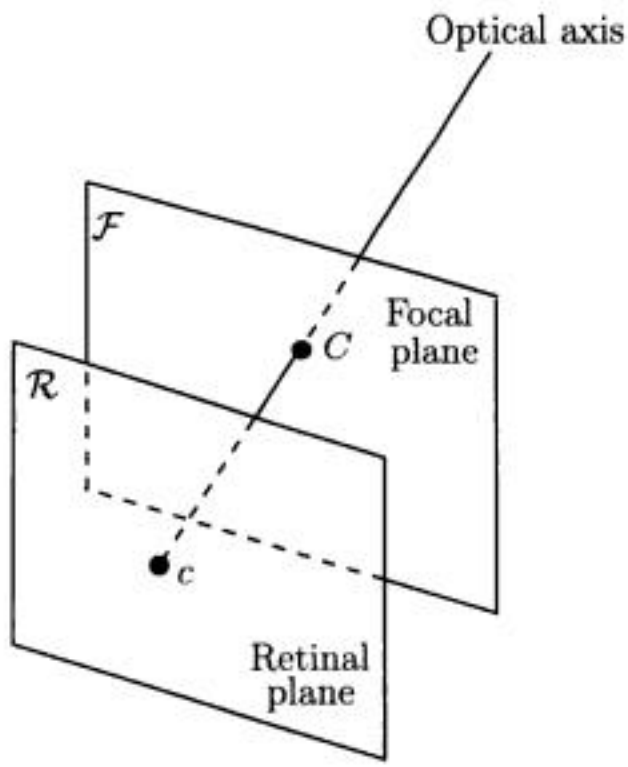


Figure 3.3 The optical axis, focal plane, and retinal plane.

3.2.1.2 A physical model

Next we will relate the amount of light that is reflected and emitted at a point on an object to the brightness of the image of that point in the retinal plane. The amount of light falling on a surface is called the *irradiance*, and it is measured in $W \times m^{-2}$, watts per square meter. The amount of light radiated from a surface is called the *radiance*, and it is measured in $W \times m^{-2} \times sr^{-1}$, watts per square meter per steradian.[1] A simple computation that can be found, for example in the book by Horn [Hor86], shows that the relationship between the image irradiance E and the scene radiance L is a very simple linear relationship:

$$E = L\frac{\pi}{4}(\frac{d}{f})^2 \cos^4 \alpha$$

The parameters involved in this equation are defined in figure 3.4.

1. The steradian is the unit used to measure solid angles.

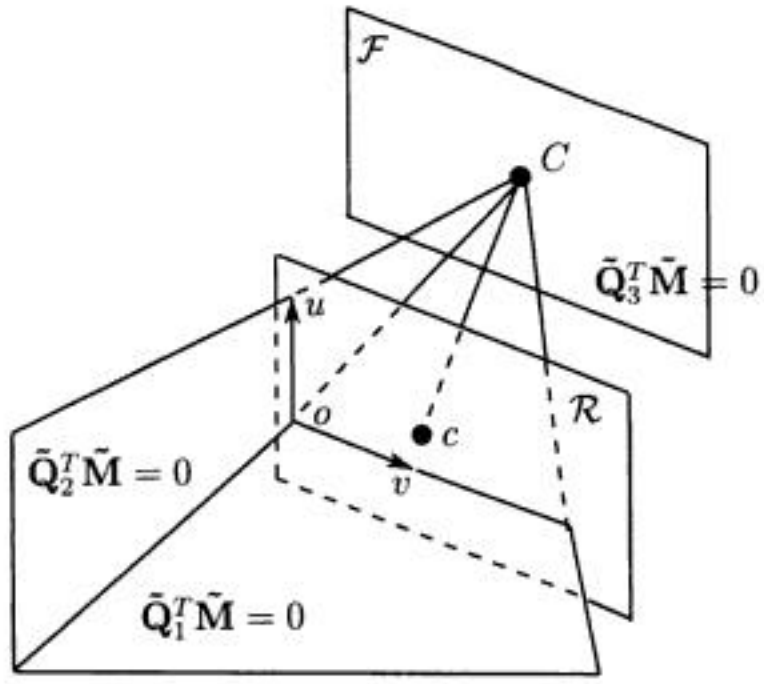


Figure 3.7 The row vectors of matrix $\tilde{\mathbf{P}}$ define the focal plane and the line joining the optical center to the origin of the coordinates in the retinal plane.

plane. Note that this line of intersection is not, in general, the optical axis of the camera. This is shown in figure 3.7.

From the perspective projection matrix $\tilde{\mathbf{P}}$ we can compute some useful information:

- One very useful piece of information in many applications is the optical center C of the camera. According to figure 3.7, C is defined as the intersection of the three planes of equations $\tilde{\mathbf{Q}}_i^T\tilde{\mathbf{M}} = 0$. Therefore it is obtained by solving the system of three linear equations

$$\tilde{\mathbf{P}}\begin{bmatrix}\mathbf{C}\\1\end{bmatrix} = \mathbf{0}$$

If we write the 3×4 matrix $\tilde{\mathbf{P}}$ as $\begin{bmatrix}\mathbf{P} & \tilde{\mathbf{p}}\end{bmatrix}$, where $\mathbf{P}$ is a 3×3 matrix and $\tilde{\mathbf{p}}$ a 3×1 vector, and assume that the rank of $\mathbf{P}$ is 3, this equation can be rewritten as

$$\mathbf{C} = -\mathbf{P}^{-1}\tilde{\mathbf{p}} \tag{3.7}$$

The careful reader may be a bit concerned by the fact that we are extracting from a matrix $\tilde{\mathbf{P}}$, which is defined only up to a scale factor,

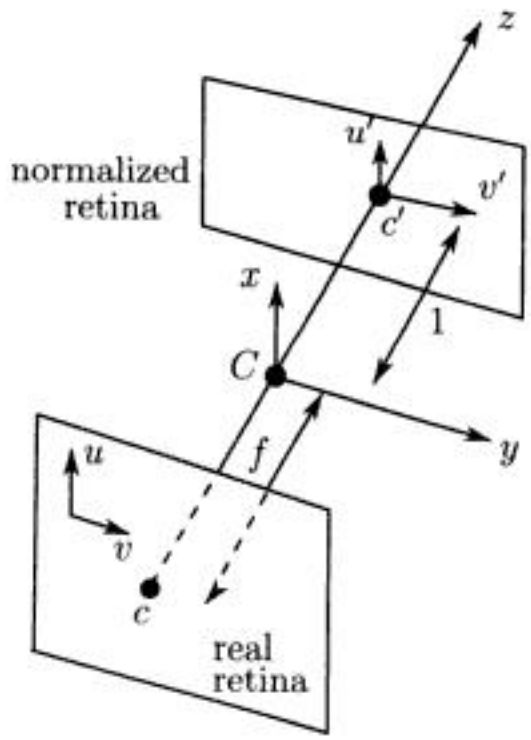


Figure 3.9 Relationship between the real and normalized retinal planes.

$$u' = \frac{u - u_0}{\alpha_u}$$

$$v' = \frac{v - v_0}{\alpha_v} \tag{3.15}$$

There are two important conclusions that we can draw from these algebraic manipulations.

- First, we have developed a geometric interpretation. If we consider the plane parallel to the retinal plane and at a unit distance from the optical center, this plane, together with the optical center, defines a "normalized" camera (see figure 3.9). Note that this plane is on the other side of C with respect to the retinal plane, producing an inverted image compared to the original one.
- Second, we have an interpretation of the dimensions of k_u, k_v, α_u, and α_v. Indeed, let us write the projection equation using equation (3.12):

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\{\tilde{m}, \tilde{n}; \tilde{a}, \tilde{b}\} = \frac{0 - \theta_0}{0 - \overline{\theta_0}} : \frac{\infty - \theta_0}{\infty - \overline{\theta_0}}$$

The ratio containing ∞ is equal to 1 (that is the magic); therefore

$$\{\tilde{m}, \tilde{n}; \tilde{a}, \tilde{b}\} = \frac{\theta_0}{\overline{\theta_0}} = e^{2iArg(\theta_0)}$$

where $Arg(\theta_0)$ is the argument of the complex number θ_0. We will let the reader finish the computation in problem 1.

3.3.2 Changing the world reference frame

Just as it is important to study how the matrix $\tilde{\mathbf{P}}$ changes when we change the image coordinate system, it is likewise important for many applications to study how the matrix $\tilde{\mathbf{P}}$ varies when we change the 3-D coordinate system.

3.3.2.1 Extrinsic parameters

As shown in figure 3.11, we go from the old coordinate system centered at the optical center C to the new coordinate system centered at O by a rotation $\mathbf{R}$ followed by a translation $\mathbf{T} = \mathbf{CO}$. Following the notation of the same figure, and similar to the retinal case, we have

$$\mathbf{CM} = \mathbf{CO} + \mathbf{OM}$$

We express $\mathbf{OM}$ in the new coordinate system as follows:

$$\mathbf{OM} = x_{new}\mathbf{I} + y_{new}\mathbf{J} + z_{new}\mathbf{K}$$

and then introduce the rotation matrix $\mathbf{R}$ from the old coordinate system to the new, yielding

$$\mathbf{I} = \mathbf{Ri} \qquad \mathbf{J} = \mathbf{Rj} \qquad \mathbf{K} = \mathbf{Rk}$$

Finally, denoting the vector $\mathbf{CO}$ in the old coordinate system as $\mathbf{t} = [t_x, t_y, t_z]^T$, we have

$$\mathbf{CM} = \mathbf{t} + \mathbf{R}\begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \end{bmatrix}$$

3.4.1 Estimation of the perspective projection matrix $\tilde{\mathbf{P}}$

Since we wish to estimate matrix $\tilde{\mathbf{P}}$ and then possibly compute from the result the values of the intrinsic and extrinsic parameters, it is important to study the conditions under which a 3×4 matrix $\tilde{\mathbf{P}}$ can be written in the form of equation (3.21). In doing this we will discover some important constraints that have not always been taken into account in the literature.

3.4.1.1 Constraints on $\tilde{\mathbf{P}}$

Clearly, not any 3×4 matrix $\tilde{\mathbf{P}}$ can be written in the form of equation (3.21). Indeed, this matrix depends upon ten parameters, whereas a general projective 3×4 matrix depends upon eleven parameters. In fact, we have the following theorem:

Theorem 3.1
Let $\tilde{\mathbf{P}}$ be a 3×4 matrix defined by equation (3.6) such that $\text{rank}(\mathbf{P}) = 3$. There exist four sets of extrinsic and intrinsic parameters such that $\tilde{\mathbf{P}}$ can be written as equation (3.21) if and only if the following two constraints are satisfied:

$$\|\mathbf{q}_3\| = 1 \tag{3.22}$$

$$(\mathbf{q}_1 \wedge \mathbf{q}_3) \cdot (\mathbf{q}_2 \wedge \mathbf{q}_3) = 0 \tag{3.23}$$

Proof The *if* condition is satisfied if equations (3.22) and (3.23) are true if $\tilde{\mathbf{P}}$ is written in the form of equation (3.21). The proof of the *only if* condition is obtained as follows. Using equation (3.21) and the fact that matrix $\tilde{\mathbf{P}}$ is known up to a scale factor ε that must be ± 1 because of equation (3.22), we can compute a set of intrinsic and extrinsic parameters. The resolution proceeds as follows. By comparing the third rows in equations (3.6) and (3.21), we obtain

$$\begin{aligned} t_z &= \varepsilon q_{34} \\ \mathbf{r}_3 &= \varepsilon \mathbf{q}_3^T \end{aligned} \tag{3.24}$$

Taking the inner products of $\mathbf{q}_3$ with $\mathbf{q}_1$ and $\mathbf{q}_2$ yields u_0 and v_0:

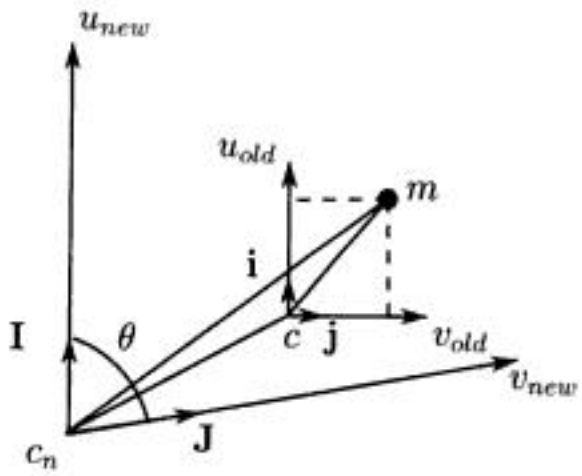


Figure 3.12 The retinal coordinate system may not even be orthogonal.

orthogonal, and that leads us to introduce an extra parameter, the angle θ between the two vectors **I** and **J**, still assuming that **I**, for example, is parallel to **i** (see figure 3.12). We have therefore increased the number of intrinsic parameters by one and can thus decrease the number of constraints on matrix $\tilde{\mathbf{P}}$ by one.[3] The relation between $(\mathbf{i}, \mathbf{j})$ and $(\mathbf{I}, \mathbf{J})$ can be written as

$$\mathbf{i} = \mathbf{sI} \quad \text{and} \quad \mathbf{j} = \mathbf{sJ}$$

From figure 3.12, it is clear that

$$\mathbf{I} = \frac{\mathbf{i}}{k_u} \qquad \mathbf{J} = \frac{\mathbf{i}\cos\theta + \mathbf{j}\sin\theta}{k_v}$$

Therefore

$$\mathbf{s} = \begin{bmatrix} k_u & -\dfrac{k_u}{\tan\theta} \\ 0 & \dfrac{k_v}{\sin\theta} \end{bmatrix}$$

In this case we can work out a set of equations similar to equations (3.15). Indeed, the most general matrix $\tilde{\mathbf{P}}$ can be written as

3. We could also say that, since it is equivalent to know the intrinsic parameters or the image of the absolute conic, and since a general conic depends upon five parameters (see section 2.4.6), five intrinsic parameters are necessary.

Since all z_i are different from 0, we have rank($\mathbf{A}$) = rank($\mathbf{A}'$). By inspection, it is clear that the first column of $\mathbf{A}'$ plus its sixth column plus its eleventh column is equal to 0. Therefore rank($\mathbf{A}'$) ≤ 11. Let us prove that it is equal to 11.

Let $\mathbf{c}_j$, $j = 1, \ldots, 11$, be the first ten and the twelfth column vectors of matrix $\mathbf{A}'$. We want to prove that a relation such as

$$\sum_{j=1}^{11} \lambda_j \mathbf{c}_j = \mathbf{0} \tag{3.38}$$

implies that $\lambda_j = 0$, $j = 1, \ldots, 11$. But equation (3.38) is equivalent to the N relations

$$\begin{aligned} &\lambda_1 x_i z_i + \lambda_2 y_i z_i + \lambda_3 z_i^2 + \lambda_4 z_i \\ &\qquad - \lambda_9 x_i^2 - \lambda_{10} x_i y_i - \lambda_{11} x_i = 0 \qquad i = 1, \ldots, N \end{aligned} \tag{3.39}$$

and the N relations

$$\begin{aligned} &\lambda_5 x_i z_i + \lambda_6 y_i z_i + \lambda_7 z_i^2 + \lambda_8 z_i \\ &\qquad - \lambda_9 x_i y_i - \lambda_{10} y_i^2 - \lambda_{11} y_i = 0 \qquad i = 1, \ldots, N \end{aligned} \tag{3.40}$$

From the fact that the polynomials xz, yz, z, z^2, x^2, and xy in the three variables (x, y, and z) and x are linearly independent, as are the polynomials xz, yz, z, z^2, xy, y^2, and y, we conclude that, if the reference points are in general positions, the equations (3.39) imply that

$$\lambda_1 = \cdots = \lambda_4 = \lambda_9 = \cdots = \lambda_{11} = 0$$

and the equations (3.40) imply that

$$\lambda_5 = \cdots = \lambda_8 = \lambda_9 = \cdots = \lambda_{11} = 0$$

Therefore, rank($\mathbf{A}$) = 11.

We can now define what we mean by *general position*. Here it means that not all of the points fall on either of the quadric surfaces whose equations are given by equations (3.39) and (3.40). Each of these quadrics is defined by seven parameters ($\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_9, \lambda_{10}$, and λ_{11} for the first one, and $\lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}$, and λ_{11} for the second one). Therefore six points are necessary to define each of them. If we choose six points at random, they define two quadrics through the equations (3.39) and (3.40).

To further compare the two methods, we have perturbed the data in the following manner: Noise has been added to the pixel coordinates of the reference points. The noise is gaussian and independent, and its standard deviation varies between 0 (for no noise) and 3 pixels. We have generated a large number of independently perturbed sets of reference points and calibrated from each of those sets. The values of the extrinsic and intrinsic parameters are plotted in the graphs of figures 3.13 and 3.14. The dotted curves represent the performances of the linear method, and the continuous curves represent the performances of the nonlinear method, which clearly appears to be more robust.

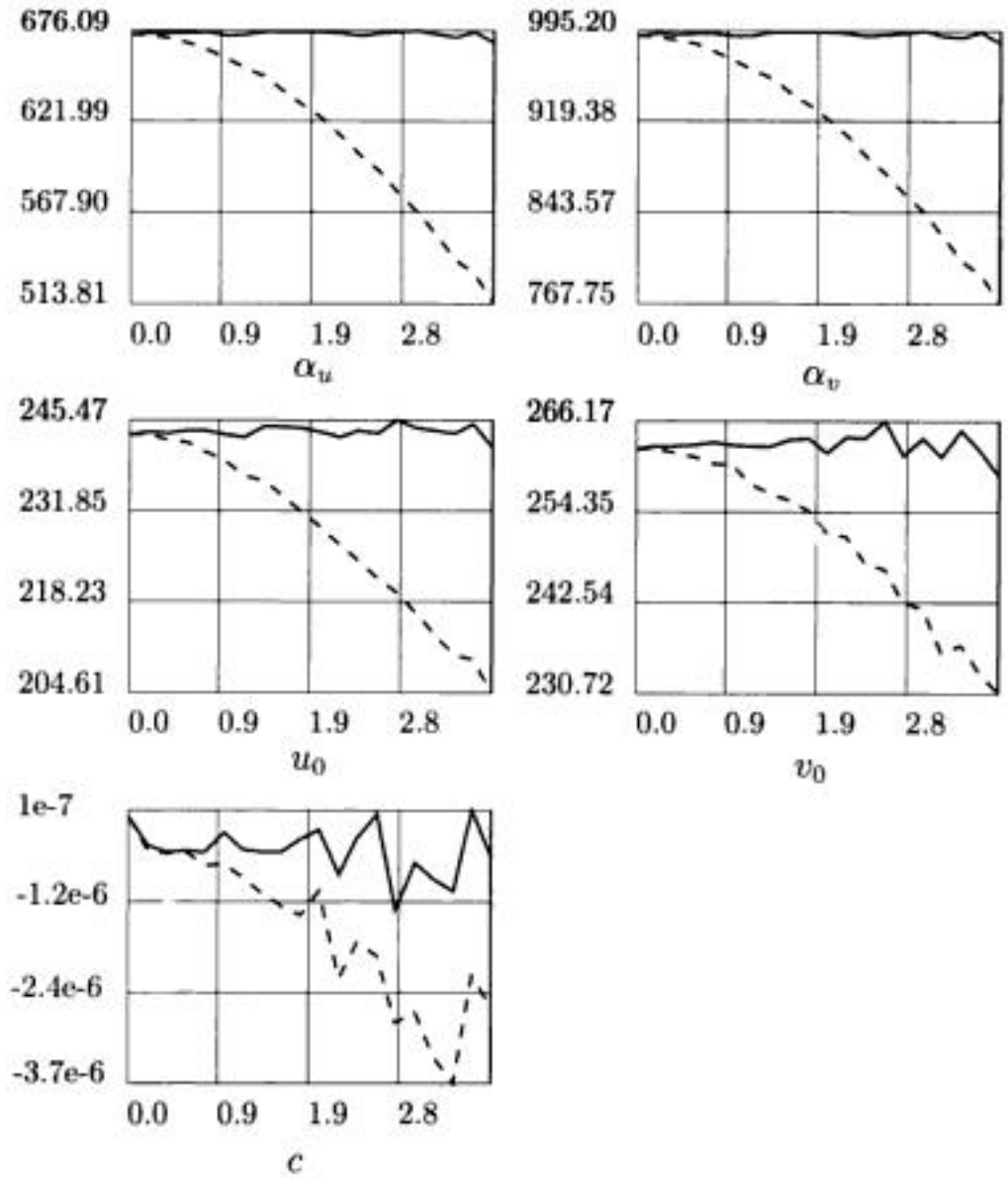


Figure 3.13 How the intrinsic parameters vary with the pixel noise.

4 Edge Detection

4.1 Introduction and precursors

4.1.1 What are edges?

For a person, it is usually very easy to find the contours of objects in a scene. The corresponding operation is extremely difficult for a computer. There are several reasons for this.

1. The first reason has to do with language. When we talk about contours of objects, we assume that the notion of object is well understood, but in fact it is one of the goals of computer vision to identify objects in scenes. Therefore the idea of contour cannot be defined through a definition of the idea of object since that would be creating a vicious circle.
2. Even if we use a more physical definition of an edge as a discontinuity of some sort of the image intensity function, we still have the problem of measurement noise, which is the second reason that the detection of edges in an image is a difficult task. Indeed, detecting discontinuities of image intensity can be achieved mathematically by computing derivatives of this function. The detection of edges is difficult, however, because this intensity is a physical measurement that is subject to noise (see section 4.2); moreover, the operation of derivation is prone to enhancing that noise. This problem is made even worse by the fact that the images that are generally processed have been both sampled and quantized.

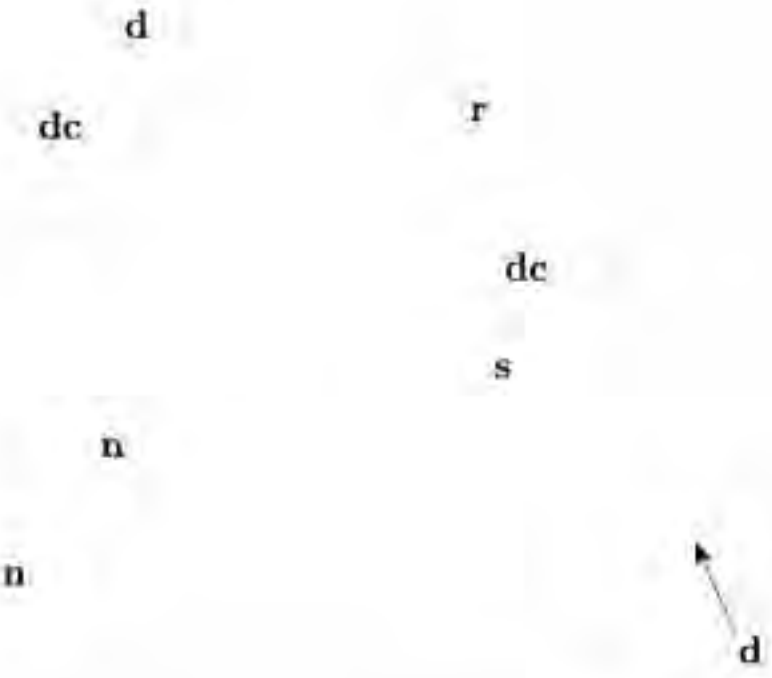


Figure 4.1 Edges in an image have different physical sources.

3. A third difficulty related to edges is that there are, so to speak, different sources of edges in a scene. Some edges come from shadows cast by objects, some from variations in the reflectance of objects, some from variations in the texture of objects, and some from variations in depth. As an example of this, figure 4.1 shows a scene composed of a cylindrical object with a dark belt standing on a polyhedral table. Edges labelled *d* are caused by discontinuities of the distance of the objects to the camera together with a discontinuity of the normal to the object surface. Edges labelled *dc* are caused by discontinuities of the distance of the objects to the camera, but the normal to the object is continuous. Edges labelled *n* are caused by a discontinuity of the normal to the object, while those labelled *r* are caused by a change of the reflectance of the object with no change of its geometrical properties. In a similar fashion, those labelled *s* are caused by the shadow of the cylinder cast on the table. All these distinct physical processes may produce edges. But if we want to relate edges in an image to objects in a scene, we must be capable of identifying these various sorts of edges.

Given these preliminary words of caution, we will look at some of the precursors for detecting edges.

4.1.2 Early edge detectors

All the earlier edge detectors tried to tackle the two problems of computing some derivatives of the image intensity and of being robust to noise. These two requirements, as will be shown later, are contradictory, and various tradeoffs have been proposed to achieve a balance between accurate detection of edges and robustness to noise.

The idea of using derivatives of the image intensity function to enhance the detection of edges can be viewed from several standpoints:

- In the signal domain, an edge in some direction corresponds to a large local variation of the intensity function in a direction perpendicular to the edge. Computing such a derivative therefore replaces the problem of detecting an edge with the problem of detecting a local extremum.
- In the frequency domain, a derivative operator can be viewed as a high-frequency booster ($\sin \omega x$ becomes $\omega \cos \omega x$), and therefore such an operator will enhance edges more than flat areas.

Note that these simple ideas do not take the noise into account.

The general paradigm is a two-step method that has been used in the past and, as we will see, is still being used today:

1. Enhance the presence of edges in the original intensity image $f(x, y)$, thus creating a new image $g(x, y)$ where edges are more conspicuous. Large values of g indicate the likelihood of the presence of an edge.[1]
2. Threshold $g(x, y)$ to make an edge/no-edge decision, yielding a binary edge map $f_e(x, y)$.

This paradigm is described in figure 4.2.

1. This applies also to the case, discussed later, where edges are detected by zero-crossings of the second-order derivative of the image intensity via a simple change of variable.

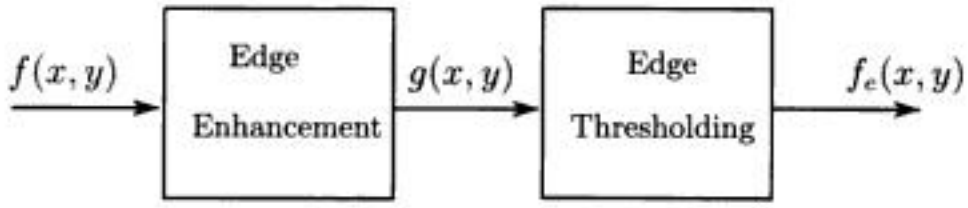


Figure 4.2 A simple paradigm for edge detection.

4.1.2.1 *Discrete approximations of derivatives*

Most of the early edge detectors have used some simple discrete approximations of continuous derivatives. Partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ of the intensity function $f(x,y)$ can be approximated with finite differences:

$$\frac{\partial f}{\partial x}(x,y) \simeq \Delta_x f(x,y) = f(x+1,y) - f(x,y)$$

$$\frac{\partial f}{\partial y}(x,y) \simeq \Delta_y f(x,y) = f(x,y+1) - f(x,y)$$

If we view the intensity function $f(x,y)$ as a surface in R^3 given by its equation $z = f(x,y)$, the three-dimensional vector $\mathbf{G} = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, -1]^T$ is normal to that surface, and its magnitude is related to its local steepness. Since the z-coordinate of $\mathbf{G}$ is constant, its projection $\mathbf{g} = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]^T$ on the (x,y) plane carries exactly the same information and points toward the direction of maximum intensity change, while its norm is an indication of the rate of change in this direction[2] and is called the image gradient. This implies in particular that $\mathbf{g}$ is orthogonal to the direction of edges. In practice, $\mathbf{g}$ is approximated by $[\Delta_x, \Delta_y]^T$. It also allows us to compute the rate of intensity changes in any direction, not only the direction of maximal change, simply by projecting it onto that direction: If $D_\theta f$ denotes the partial derivative in the direction θ, we have $D_\theta f = \nabla f \cdot \mathbf{u}_\theta = \mathbf{g} \cdot \mathbf{u}_\theta$, where $\mathbf{u}_\theta$ is the unit vector in the direction θ[3]. We will see more about the differential properties of the image intensity function in section 4.2.4.2.

---

2. $\mathbf{g}$ is also denoted by ∇f.

3. More generally, the derivative of f in the direction of a vector $\mathbf{u}$ (not necessarily of unit length) is noted $D_\mathbf{u} f$ and is equal to $\nabla f \cdot \mathbf{u}$. In particular, we will consider $D_\mathbf{g} f$, the derivative of f in the direction of the gradient.

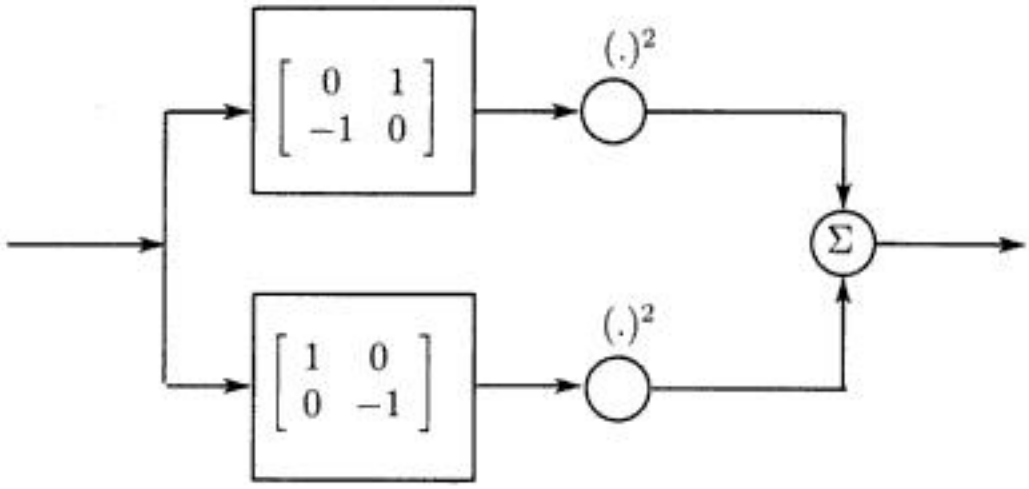


Figure 4.3 The Roberts operator.

Another example is obtained by smoothing the image with the impulse response

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

and then computing Δ_y of the result. The corresponding impulse response is

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

which is one of Sobel's masks, also discussed in the next section.

4.1.2.3 A few "classical operators"

We will now tie some of the previous ideas to some edge operators that have proven to be useful in practice.

1. The Roberts operator [Rob65] is described in figure 4.3, which shows that this operator computes $D_{45}f$ and $D_{135}f$ (see footnote 3), and then the euclidean norm of the discrete gradient.
2. Sobel's and Prewitt's operators [Sob78, Pre70] are described in figure 4.4, which shows that they compute the horizontal and vertical

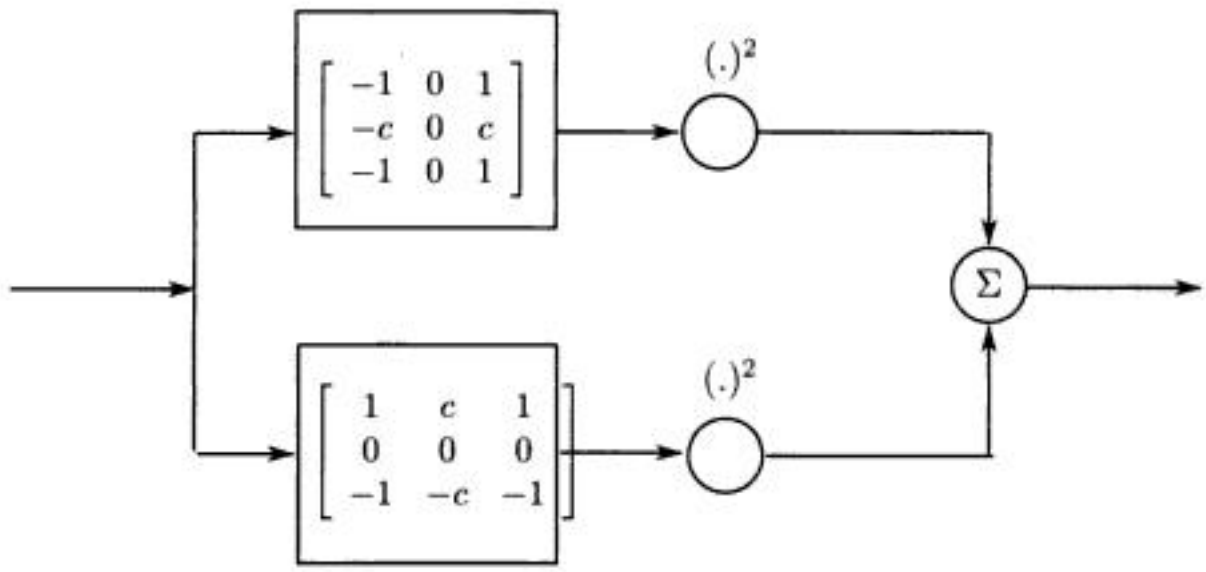


Figure 4.4 Sobel's ($c = 2$) and Prewitt's ($c = 1$) operators.

components of a smooth gradient, and then combine them to yield the euclidean norm of this vector.

4.1.3 Measuring the quality of an edge detector

The idea of measuring the quality of an edge detector is an old one, and Abdou and Pratt [Abd78] have used it to define a figure of merit for comparing edge detectors. The idea is not to use the figure of merit to design the edge detector, as will be done in Section 4.3, but to use it to select the best among existing detectors. Figure 4.5 shows a number of defects from which a detector can suffer. The figure of merit proposed by Abdou and Pratt is

$$F = \frac{1}{max(I_I, I_A)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}$$

where I_I is the ideal number of edge points, I_A is the actual number of edge points, $d(i)$ is the shortest distance of the ith actual edge point to an ideal edge point, and α is a positive constant. The figure of merit F is less than or equal to 1, with equality when $I_I = I_A$ and $d(i) = 0$ for all i. Using a measure of the quality of an edge detector in its design has proven to be extremely powerful, as is shown in section 4.3.

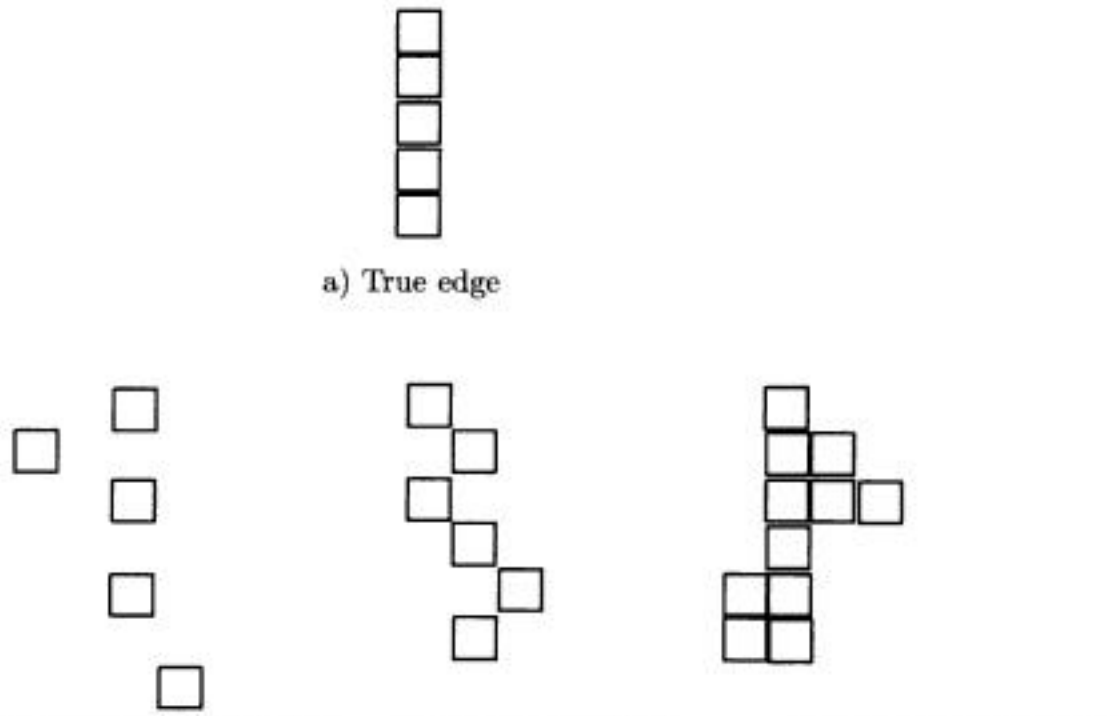


Figure 4.5 Edge defects.

4.2 Computing derivatives and smoothing

In the previous section we have shown the need to compute smoothed derivatives of the image. We will now study this problem in greater detail.

4.2.1 Differentiation as an ill-posed problem

One way of realizing that differentiation is indeed quite sensitive to noise is to consider a signal $s(x)$ perturbed by a sinusoidal noise:

$$S(x) = s(x) + \varepsilon \sin(\omega x)$$

If ε is sufficiently small, $S(x)$ and $s(x)$ are very close to each other (in the mean-square sense, for example[4]). Taking the derivative with respect to x, we obtain

$$S'(x) = s'(x) + \varepsilon\omega \cos(\omega x)$$

4. $\lim_{a\to\infty} \int_{-a}^{a} (S(x) - s(x))^2 \, dx = \frac{\varepsilon^2}{2}$

If ω is large enough, then the sinusoidal term may completely dominate the signal. The fact that high-frequency noise can hurt operations like deconvolution has been known for decades both to mathematicians and to electrical engineers. Powerful tools like the Wiener filter have been heavily used in problems such as image restoration [AH77]. Unfortunately, this formulation models both the signal s and the noise as stochastic processes and assumes some statistical knowledge about both of them. Usually stationary Gaussian processes are assumed, but these are not well suited for modeling images, especially near edges that signify an abrupt change in the statistical properties of the underlying process and therefore imply nonstationarity. Another point is that this formulation is equivalent to maximizing the signal-to-noise ratio; but, as we will see in section 4.3, there are other criteria that are also important and are not taken into account by this approach.

The previous discussion points to the fact that the problem of differentiation is ill-posed. An ill-posed problem can be defined in contrast to a well-posed problem. In 1923, Hadamard [Had23] introduced the definition of a well-posed mathematical problem as one whose solution (1) exists, (2) is unique, and (3) is robust against noise (i.e., depends continuously on the data). Differentiation can be seen as a linear inversion problem of finding $f(x)$, given

$$g(x) = \int Y(x - y) f(y)\, dy \tag{4.1}$$

where $Y(x)$ is the step function, sometimes also known as the Heaviside function, defined as follows:

$$Y(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

It is well known that, if the linear operator applied to the data (here the convolution with $Y(x)$) is not well behaved, then the corresponding inversion problem is not well-posed, but rather is ill-posed.

Methods for transforming ill-posed problems into well-posed ones have recently been developed [TA77]. The idea is, given the problem of finding $\mathbf{f}$ from the data $\mathbf{g}$ such that $\mathbf{g} = \mathbf{Af}$, to set up a mean-square problem involving the sum of two terms:

$$\min_{\mathbf{f}}(\|\mathbf{g} - \mathbf{Af}\| + \lambda\|\mathbf{Pf}\|) \tag{4.2}$$

where **P** is a regularization operator that is intended to impose smoothness on the solution **f**. This technique has been used in the restoration of images [AH77, Pra78]. There are other ways of regularizing an ill-posed problem but, as pointed out by Torre and Poggio [TP86], the most useful one for computer vision is the one above.

The basic result of these methods is that, for convolution equations such as (4.1), regularization is achieved by filtering $g(x)$ with a smooth low-pass filter, and differentiation is performed on the smoothed version of $g(x)$. In order to be stabilizing, the filters must satisfy a number of properties described by Torre and Poggio [TP86] that are not very constraining. Therefore, the theory validates the intuitive ideas used in the design of the early edge detectors.

We have seen that, in order to be more robust to noise, differentiation has to be performed on a smooth version of the signal. We will now discuss two related ways of smoothing, smoothing by filtering and smoothing by approximation. Each case is illustrated by an example of an edge detector.

4.2.2 Smoothing by filtering: the Marr-Hildreth detector

Filtering is commonly used to smooth the data before differentiating it. There are a number of possible filters.

- Band-limited filters can eliminate noise with known frequency content. The corresponding impulse responses are infinite and may therefore pose some problems of implementation (see section 4.4). Examples of such filters are the prolate spheroidal functions [LP61], which satisfy all regularization properties given in the article by Torre and Poggio [TP86], and the Wiener filter in the case of a pink noise image model with an independent additive white noise.
- We can also use support-limited filters or finite impulse response (FIR) filters. They are interesting from the computational standpoint, but in general fail to satisfy the regularization properties because of their infinite frequency support.
- Another class of filters that can be considered as a compromise between the first two classes is the class of filters that minimize uncer-

Figure 4.7 The surface $D^2_{45}G(r)$ (see text).

would greatly reduce the computational burden if their number could be decreased by using just one orientation-independent operator, for example. This immediately points toward the Laplacian. The condition under which it can be used is that the intensity variation in $G_\sigma \otimes f$ is linear along, but not necessarily near, a line of zero-crossings. In that case, the zero values of the Laplacian will detect and accurately locate the zero-crossings. Edges are detected as the pixels satisfying

$$\nabla^2 G_\sigma \otimes f(x, y) = 0 \tag{4.5}$$

This condition of linear variation is approximately satisfied in practice. In principle, however, if the intensity varies along an edge in a very nonlinear way, the Laplacian will see the zero-crossings displaced to one side (see figure 4.8). This figure shows a vertical step edge modulated vertically by a nonlinear function; the intensity variation along the edge is not linear. The results of the computation of the zero-crossings of the Laplacian are shown at the right side of the figure, and a closer view at the bottom shows the horizontal displacement of the edge.

The results of applying this operator to the image of figure 4.9 are shown in figure 4.10. Obviously there are too many zero-crossings that have no obvious perceptual significance; some sort of thresholding must take place in order to obtain the result of figure 4.11. The thresholding that has taken place is explained in more detail in section 4.5.

Figure 4.9 A picture of an office scene.

4.2.3 Some useful results from the calculus of variations

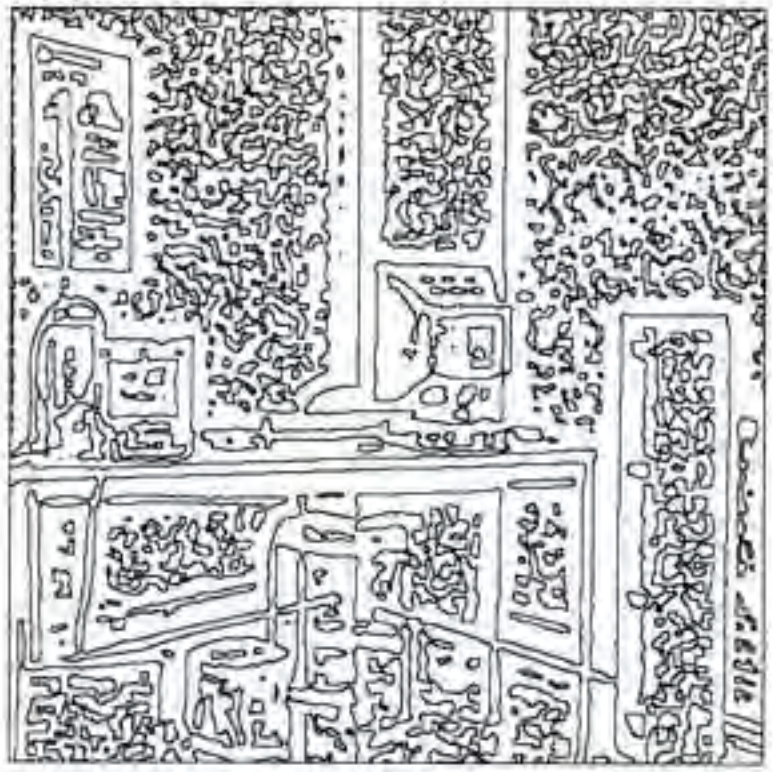
In some of the forthcoming sections we are going to need some simple results from the calculus of variations. Suppose we are given a function Φ from $R \times \underbrace{R \times \cdots \times R}_{n+1}$ into R, which we assume to be of class C^k. For a given function $S : R \to R$ of class C^n, we consider $(x, S(x), S'(x), \ldots, S^{(n)}(x))$ and compute the number

$$\varphi(S) = \int_a^b \Phi(x, S(x), S'(x), \ldots, S^{(n)}(x))\, dx$$

This number depends upon the choice of S. If we consider the set of functions S such that $S(a) = \alpha$ and $S(b) = \beta$ for two given real numbers α and β, we are interested in determining, among all functions S satisfying those two conditions, those for which φ is extremal.

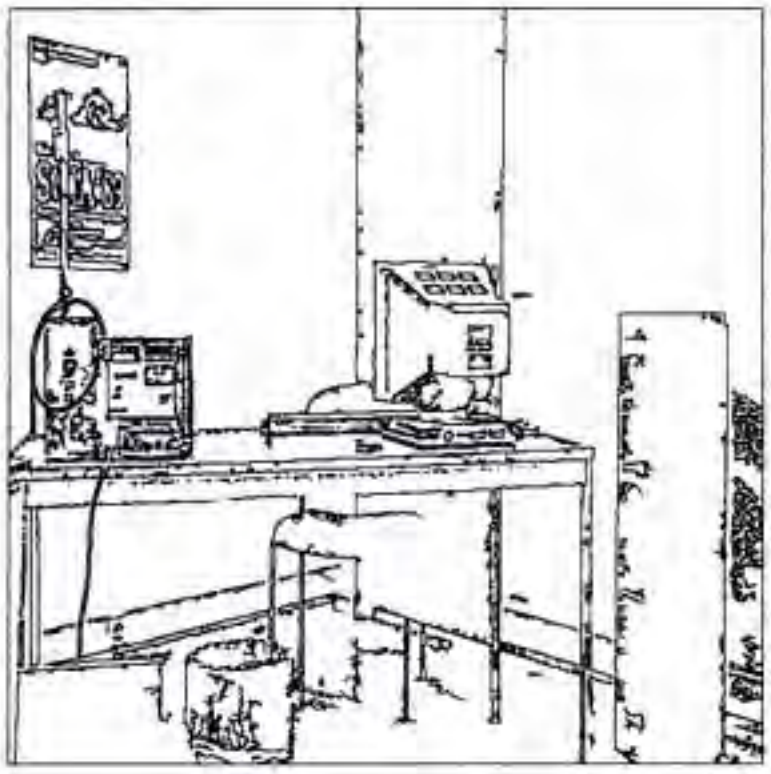
A *necessary* condition for this is that the functions satisfy the Euler-Lagrange equation

Zero crossings of $\nabla^2 G$ with σ=1.

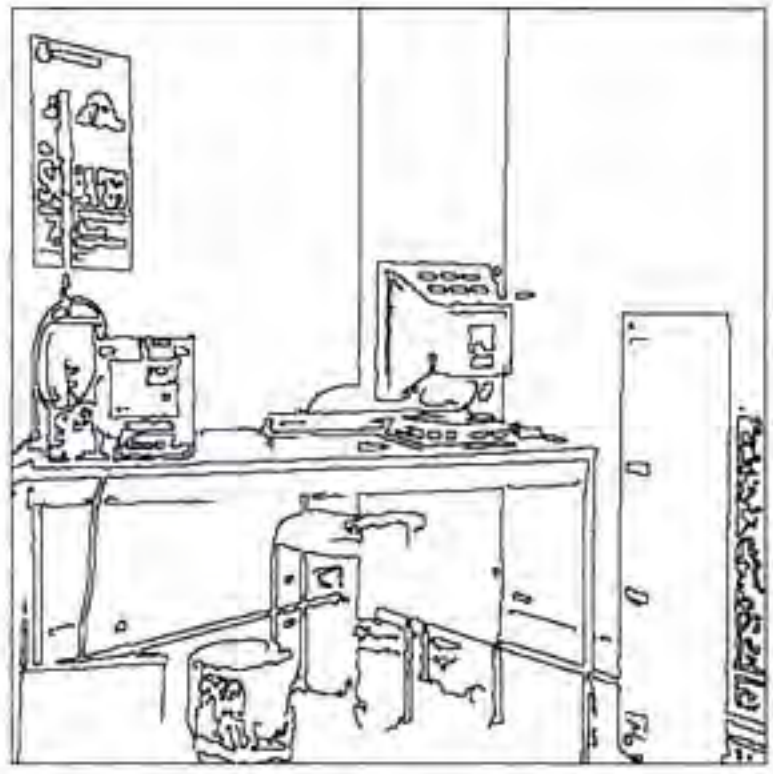


Zero crossings of $\nabla^2 G$ with σ=3.

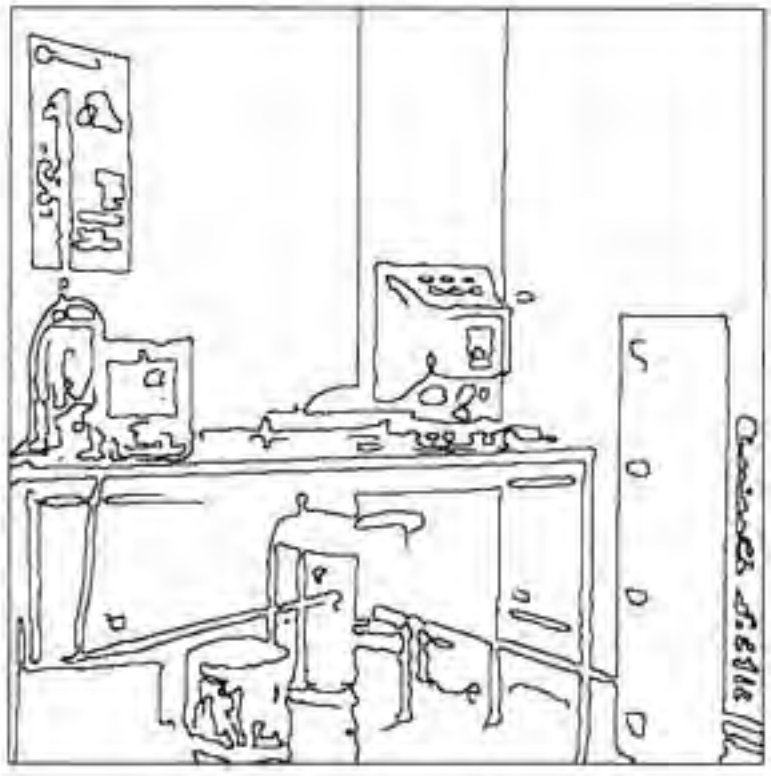
Figure 4.10 The zero-crossings of the Marr-Hildreth operator applied to the image in figure 4.9.



σ=1 $T_h = 15$ and $T_l = 5$.



σ=2 $T_h = 15$ and $T_l = 5$.



σ=3 $T_h = 15$ and $T_l = 5$.

Figure 4.11 Only the zero-crossings of the image in figure 4.9 for which the magnitude of the gradient is above a certain threshold have been retained.

and minimize

$$\sum_{(x,y)\in R} (f(x,y) - S(x,y))^2$$

with respect to $a_1, \ldots, a_n$. The result is given (see problem 2) by

$$a_k = \frac{\sum_{(x,y)\in R} P_k(x,y) f(x,y)}{\sum_{(x,y)\in R} P_k^2(x,y)}$$

This shows that each coefficient a_k is obtained by a linear combination of the image values with the function $Q_k(x,y) = \frac{P_k(x,y)}{\sum_{(x,y)\in R} P_k^2(x,y)}$. Derivatives of f are then approximated with derivatives of S. In particular, Haralick proposed finding the edges as the zero-crossings of $D_{\mathbf{g}}^2 f$, i.e., of $D_{\mathbf{g}}^2 S$. We will see more of this idea in section 4.5.2 [7].

4.2.4.2 *The second derivative in the direction of the gradient and the Laplacian*

Let us study the relationship between the zero-crossings of the second derivative $D_{\mathbf{g}}^2 f$ of the intensity function f in the direction of the gradient $\mathbf{g}$ and those of the Laplacian. We know what a first-order directional derivative is (see section 4.1.2.1), but what is a second directional derivative? If we consider a function $f : R^n \to R$, a point M of R^n represented by the vector $\mathbf{M}$, and an n-dimensional vector $\mathbf{u}$, we can consider the restriction of the function f to the line of R^n defined by the point M and the vector $\mathbf{u}$. This is a function that we call r from R to R defined by $r(x) = f(\mathbf{M} + x\mathbf{u})$. The first-order derivative of r at $x = 0$ is equal to the directional derivative $D_{\mathbf{u}} f$ of f in the direction $\mathbf{u}$, and its second-order derivative at $x = 0$ is equal, by definition, to the second directional derivative $D_{\mathbf{u}}^2 f$ of f in the direction of the vector $\mathbf{u}$. If we do a second-order Taylor series expansion of r in the vicinity of 0, we will get the expression of $D_{\mathbf{u}}^2 f$. Indeed, we have

$$r(x) = r(0) + xr'(0) + \frac{x^2}{2} r''(0) + \varepsilon(x^2)$$

$$= f(\mathbf{M}) + x\nabla f \cdot \mathbf{u} + \frac{x^2}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \varepsilon(x^2)$$

7. $D_{\mathbf{g}} f$ is the derivative in the direction of the gradient $D_{\mathbf{g}} f = \nabla f \cdot \mathbf{g} = \mathbf{g} \cdot \mathbf{g} = \|\mathbf{g}\|^2$.

where **H** is the Hessian of f evaluated at **M**. From this it follows that

$$D^2_{\mathbf{u}} f = \mathbf{u}^T \mathbf{H} \mathbf{u}$$

In the case where f is the image intensity function, we immediately have

$$D^2_{\mathbf{g}} f = \mathbf{g}^T \mathbf{H} \mathbf{g} = f_x^2 f_{xx} + 2 f_x f_y f_{xy} + f_y^2 f_{yy} \tag{4.8}$$

The second derivative in the direction $\mathbf{g}_\top = [-f_y, f_x]^T$ orthogonal to the gradient **g** is given by

$$D^2_{\mathbf{g}_\top} f = \mathbf{g}_\top^T \mathbf{H} \mathbf{g}_\top = f_y^2 f_{xx} - 2 f_x f_y f_{xy} + f_x^2 f_{yy} \tag{4.9}$$

This yields a simple relation between the Laplacian ∇^2, $D^2_{\mathbf{g}}$, and $D^2_{\mathbf{g}_\top}$:

$$\nabla^2 = \frac{D^2_{\mathbf{g}} + D^2_{\mathbf{g}_\top}}{\|\mathbf{g}\|^2}$$

Notice that the operators $D^2_{\mathbf{g}}$ and $D^2_{\mathbf{g}_\top}$ are in general nonlinear. If we consider the intensity surface of section 4.1.2.1, we can express its mean curvature H (see appendix C) as a function of the derivatives of f:

$$H = \frac{(1 + f_x^2) f_{yy} + (1 + f_y^2) f_{xx} - 2 f_x f_y f_{xy}}{2g^3}$$

where $g^2 = 1 + f_x^2 + f_y^2 = 1 + \|\mathbf{g}\|^2$. A simple algebraic manipulation shows that

$$2g^3 H = g^2 \nabla^2 f - D^2_{\mathbf{g}} f$$

From this we immediately conclude that the zeros of $D^2_{\mathbf{g}} f$ coincide with those of $\nabla^2 f$ if and only if the mean curvature H is zero, which in general is not true.

4.3 One-dimensional edge detection by the maxima of the first derivative

We will now present a family of edge detectors based on the detection of extrema in the output of the convolution of the image with an impulse response to be determined. Since this impulse response has no reason to be anisotropic, it must be even or odd. Because we want to detect edges as extrema in the output, the impulse response must be "deriva-

tionlike" and therefore odd. The nice thing about the approach to be described [Can83, Can86] is that it explicitly takes two factors into account in the design of the edge detector: (1) a model of the kind of edges to be detected and (2) a quantitative definition of the performance this edge detector is supposed to have (remember section 4.1.3).

Using these specifications, we derive a criterion that must be satisfied by the unknown impulse response and we minimize that criterion using techniques of variational calculus. The edge detector thus produced can be said to be optimal for the given criterion. We restrict ourselves to step edges, and our edge model is the following:

$$e(x) = AY(x) + n(x) \tag{4.10}$$

where $n(x)$ is a stationary white noise process satisfying

$$E(n(x)) = 0$$

and

$$E(n^2(x)) = \sigma_0^2$$

Several possible criteria can be chosen to characterize the performance of an edge detector. Three such criteria are (1) good detection, i.e., robustness to noise; (2) good localization; and (3) uniqueness of response. The last criterion means that the detector should not produce multiple outputs in response to a single edge. Let us now study in detail the derivation of the filter.

4.3.1 Deriving quantitative criteria

Let $h(x)$ be the unknown impulse response, and $o(x)$ the output signal (see figure 4.12). We have

$$o(x) = \int_{-\infty}^{+\infty} e(x-y)h(y)\,dy = A\int_{-\infty}^{x} h(y)\,dy + \int_{-\infty}^{+\infty} n(x-y)h(y)\,dy$$

At $x = 0$, we have

$$o(0) = A\int_{-\infty}^{0} h(y)\,dy + \int_{-\infty}^{+\infty} n(-y)h(y)\,dy$$

This is the sum of two contributions, the signal contribution

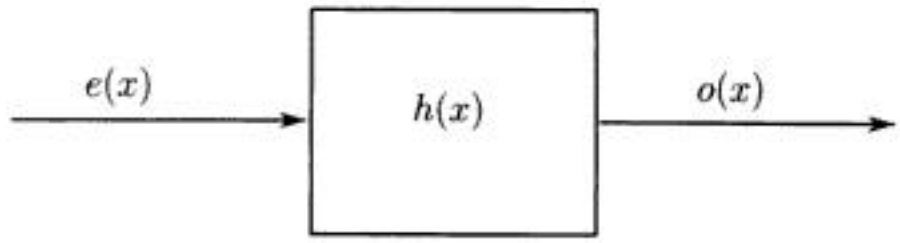


Figure 4.12 Edge detection by convolution with $h(x)$.

$$S = A\int_{-\infty}^{0} h(y)\,dy$$

and the noise contribution

$$N = \int_{-\infty}^{+\infty} h(y)n(-y)\,dy$$

4.3.1.1 Detection criterion

From the definition of the random noise $n(x)$, N is a random variable such that $E(N) = 0$ and

$$E(N^2) = \sigma_0^2 \int_{-\infty}^{+\infty} h^2(y)\,dy$$

We can therefore define the signal-to-noise ratio at $x = 0$ as[8]

$$SNR = \frac{S}{E(N^2)^{1/2}} = \frac{A}{\sigma_0} \qquad \frac{\int_{-\infty}^{0} h(y)\,dy}{(\int_{-\infty}^{+\infty} h^2(y)\,dy)^{1/2}} = \frac{A}{\sigma_0}\Sigma(h)$$

This is an analytical expression for the detection criterion. If we maximize $\Sigma(h)$, the result is the standard matched filter. The derivation of this filter is interesting in its own right, since it will allow us to become more familiar with the calculus of variations.

4.3.1.2 Matched filter

For mathematical commodity, but without loss of generality, we will assume that h is nonzero only in an interval $[-W, W]$. Since h is odd, the denominator of $\Sigma(h)$ can be rewritten as

8. We assume that $h(x) > 0$ for $x < 0$.

$$\sqrt{2}\left(\int_{-W}^{0} h^2(y)\,dy\right)^{1/2}$$

In order to minimize $\Sigma(h)$ we can maximize

$$\int_{-W}^{0} h^2(y)\,dy$$

subject to the constraint

$$\int_{-W}^{0} h(y)\,dy = c_1$$

Applying the Lagrange multiplier idea, this is the same as maximizing

$$C(h) = \int_{-W}^{0} (h^2(y) + \lambda_1 h(y))\,dy = \int_{-W}^{0} \Phi(y,h)\,dy$$

We know from section 4.2.3 that this can be done by solving the corresponding Euler-Lagrange equation[9]

$$\Phi_h = 0$$

considered as a differential equation in h.

In this case, the Euler-Lagrange equation is very simple:

$$2h(y) + \lambda_1 = 0 \tag{4.11}$$

which shows that h is constant over the interval $[-W,0]$. The corresponding odd function is displayed in figure 4.13. This is the well-known difference of boxes used in the Herskowitz-Binford edge detector [HB80].

4.3.1.3 The localization criterion

Let us now look at the localization criterion. One possible way to make this criterion quantitative is to define it as the amount of displacement of the position x_0 of the maximum in the output $o(x)$ with respect to the true position $x = 0$ of the edge. The random variable x_0 depends on both the edge and the noise. We define the localization as the inverse of the standard deviation of x_0. A maximum in the output $o(x)$ at x_0 corresponds to a zero value of the derivative $o'(x_0)$. Let us compute $o'(x)$ as follows:

9. Lower indexes indicate a partial derivative.

$$E(N(x)^2) = \sigma_0^2 \int_{-\infty}^{+\infty} h'^2(y)\, dy$$

Let us look at the signal part. Assuming that x_0 is close to 0, we approximate $S(x_0)$, up to the second order, as

$$S(x_0) \simeq Ah(0) + x_0 Ah'(0)$$

Note that h is odd, and therefore $h(0) = 0$, which is equal to $Ax_0h'(0)$. Since x_0 also satisfies $o'(x_0) = 0$, we can write

$$o'(x_0) = S(x_0) + N(x_0) \approx Ax_0h'(0) + N(x_0) = 0$$

Therefore

$$x_0 \simeq -\frac{N(x_0)}{Ah'(0)}$$

The mean of x_0 is 0, and its variance $E(x_0^2)$ is given by

$$E(x_0^2) = \frac{\sigma_0^2}{A^2} \qquad \frac{(\int_{-\infty}^{+\infty} h'^2(y)\, dy)}{h'^2(0)}$$

The localization is defined as

$$\frac{1}{E(x_0^2)^{1/2}} = \frac{A}{\sigma_0} \qquad \frac{|\, h'(0)\,|}{(\int_{-\infty}^{+\infty} h'^2(y)\, dy)^{1/2}} = \frac{A}{\sigma_0}\Lambda(h)$$

Notice that both the detection and the localization criteria are the product of a term that is a property of the signal and the noise, A/σ_0, and a term that is a property of the operator only.

The product $\Sigma\Lambda$ can be considered as a measure of both criteria that has the nice property of amplitude and scale independence, as can be verified by replacing $h(x)$ by $h_\lambda(x) = \frac{1}{\lambda}h(\frac{x}{\lambda})$. Therefore

$$\Sigma\Lambda = \frac{\int_{-\infty}^{0} h(y)\, dy}{(\int_{-\infty}^{+\infty} h^2(y)\, dy)^{1/2}} \frac{|\, h'(0)\,|}{(\int_{-\infty}^{+\infty} h'^2(y)\, dy)^{1/2}} \tag{4.12}$$

Finding the odd function h that maximizes the product $\Sigma\Lambda$ can be achieved by the calculus of variations in a way that is similar to what was done in the previous section. We assume again that h is 0 outside of the interval $[-W, +W]$, where W is a constant defining the size of the impulse response h. We then maximize

$$\int_{-W}^{0} h^2(y)\,dy$$

subject to the constraints

$\int_{-W}^{0} h(y)\,dy = c_1,$

$\int_{-W}^{0} h'^2(y)\,dy = c_2,$

and

$h'(0) = c_3$

Applying the results of section 4.2.3, this is the same as maximizing

$$C(h) = \int_{-W}^{0} (h^2(y) + \lambda_1 h(y) + \lambda_2 h'^2(y))\,dy = \int_{-W}^{0} \Phi(y, h, h')\,dy$$

The corresponding Euler-Lagrange equation is

$$\Phi_h - \frac{d}{dy}\Phi'_h = 2h(y) + \lambda_1 - 2\lambda_2 h''(y) = 0 \qquad (4.13)$$

This differential equation has the solution

$$h(x) = -\frac{\lambda_1}{2}\left(1 - \frac{\cosh \alpha(x + W/2)}{\cosh \alpha W/2}\right) \quad \text{on } [-W, 0]$$

where $\alpha = \lambda_2^{-1/2}$.

In particular, when α approaches infinity, λ_2 approaches 0 and equation (4.13) reduces to equation (4.11). The value of $h(x)$ approaches a constant over the range $[-W, 0]$, the signal-to-noise ratio approaches 1, and the localization term increases without bound. This function which, as we saw previously, is the optimal matched filter for the step edge model given in equation (4.10), gives the best possible signal-to-noise ratio with arbitrarily good localization. What is wrong then? Well, as shown in Figure 4.14, it tends to exhibit many maxima in its response to noisy step edges. The left part of the figure shows a step edge with noise added to it. The signal-to-noise ratio is equal to 1. The right part of the figure shows the result of applying the optimal matched filter to this noisy edge; the extra maxima are quite obvious. These extra maxima should be considered erroneous. However, we did not consider the interaction of the response at several nearby points when we constructed our crite-

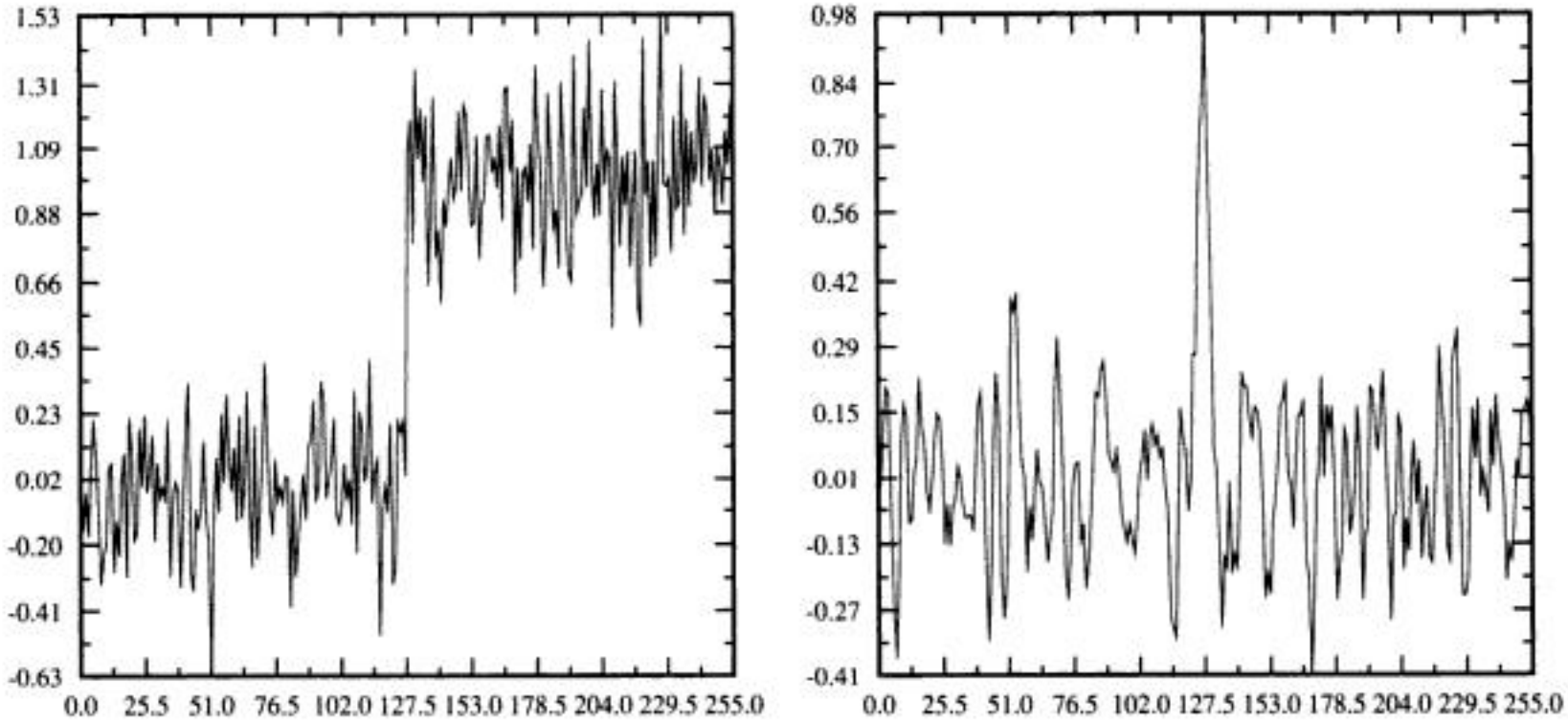


Figure 4.14 Problems with the matched filter (see text).

rion. This needs to be made explicit by adding a further constraint to the solution.

4.3.1.4 *The uniqueness-of-response criterion*

We need to add the requirement that h does not have "too many" responses to a single step edge in the vicinity of the step. In order to make this idea quantitative, we must obtain an expression for the distance between adjacent noise peaks in the output. We note that the mean distance between adjacent maxima in the output is twice the mean distance between adjacent zero-crossings in the derivative of the operator output. Rice [Ric45] tells us that the mean distance x_{ave} between the zero-crossings of a Gaussian random process obtained by filtering a white noise process with the impulse response $g(x)$ is equal to

$$x_{ave} = \pi \sqrt{-\frac{R_g(0)}{R_g''(0)}}$$

where $R_g(\tau) = \int_{-\infty}^{+\infty} g(x+\tau)g(x)\,dx$ is the autocorrelation function of the filtered noise. See problem 4 or the book by Papoulis [Pap65] for a proof of this formula.

Since $R_g''(\tau) = \int_{-\infty}^{+\infty} g''(x+\tau)g(x)\,dx$ and

$$R_g''(0) = \int_{-\infty}^{+\infty} g''(x)g(x)\,dx = \int_{-\infty}^{+\infty} g(x)d[g'(x)]$$

$$= [g(x)g'(x)]_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} g'^2(x)\,dx$$

assuming that g and g' are zero at infinity, we have $R_g''(0) = -\int_{-\infty}^{+\infty} g'^2(x)\,dx$, and

$$x_{ave} = \pi\left(\frac{\int_{-\infty}^{+\infty} g^2(x)\,dx}{\int_{-\infty}^{+\infty} g'^2(x)\,dx}\right)^{1/2}$$

In the case of edge detection, we are looking at maxima of the output of the convolution of the image intensity with the impulse response h or, equivalently, at zero-crossings of the convolution with h'. We thus have $g = h'$, and x_{ave} is the average distance between two extrema of $o(x)$. The average distance between two maxima of $o(x)$ is therefore equal to $2x_{ave}$. Therefore

$$x_{max} = 2x_{ave} = 2\pi\left(\frac{\int_{-\infty}^{+\infty} h'^2(x)\,dx}{\int_{-\infty}^{+\infty} h''^2(x)\,dx}\right)^{1/2} \tag{4.14}$$

4.3.2 Finding the optimal h

The optimal h can be found in several ways, depending on how we combine the three criteria. In all cases, the calculus of variations is used to derive the h that maximizes the chosen criterion. We will first discuss the original idea of Canny [Can83, Can86], which is illuminating, and then we will discuss two related approaches proposed by Deriche [Der87] and Spacek [Spa85].

4.3.2.1 Canny's approach

Canny maximizes the criterion $\Sigma\Lambda$ subject to the constraint $x_{max} = kW$, which states that the average maximum distance between two local maxima has to be some fraction of the spatial extent of the impulse response. From the beginning, therefore, the assumption is that this extent is finite, in marked contrast to Deriche's approach, which will be discussed next.

From the definition of equation (4.14), it can be seen that it adds only one constraint to the previous three:

$$\int_{-W}^{0} h''^2(y)\,dy = c_4$$

Using the same ideas, we find that we have to minimize

$$C(h) = \int_{-W}^{0} (h^2(y) + \lambda_1 h(y) + \lambda_2 h'^2(y) + \lambda_4 h''^2(y))\,dy$$

$$= \int_{-W}^{0} \Phi(y, h, h', h'')\,dy$$

The corresponding Euler-Lagrange equation is

$$\Phi_h - \frac{d}{dy}\Phi_{h'} + \frac{d^2}{dy^2}\Phi_{h''} = 0$$

This yields the following differential equation:

$$2h(x) - 2\lambda_2 h''(x) + 2\lambda_4 h''''(x) + \lambda_1 = 0$$

It can be shown [Can86] that the solution $h(x)$ of the differential equation is

$$h(x) = e^{-\alpha x}(a_1 \sin \omega x + a_2 \cos \omega x) + e^{\alpha x}(a_3 \sin \omega x + a_4 \cos \omega x) - \lambda_1/2$$

for x in $[-W, 0]$.

This is subject to the boundary conditions

$$h(0) = h(-W) = h'(-W) = 0$$

$$h'(0) = c_3$$

We have also added the condition that x_{max} is some fraction k of the operator length W:

$$x_{max} = kW$$

Then a_1, a_2, a_3, and a_4 are easily computed as functions of α, ω, c_3, and λ_1 [Can86]:

$$a_1 = -\frac{\lambda_1}{8(\omega^2 \sinh^2 \alpha - \alpha^2 \sin^2 \omega)} f_1(\alpha, \beta, \omega)$$

$$a_2 = -\frac{\lambda_1}{8(\omega^2 \sinh(\alpha)^2 - \alpha^2 \sin(\omega)^2)} f_2(\alpha, \beta, \omega)$$

with

$$f_1 = \alpha(\beta - \alpha)\sin 2\omega - \alpha\omega\cos 2\omega + (-2\omega^2 \sinh\alpha + 2\alpha^2 e^{-\alpha})\sin\omega + 2\alpha\omega\sinh\alpha\cos\omega + \omega(\alpha+\beta)e^{-2\alpha} - \beta\omega$$

$$f_2 = \alpha(\beta - \alpha)\cos 2\omega + \alpha\omega\sin 2\omega - 2\alpha\omega\cosh\alpha\sin\omega - 2\omega^2\sinh\alpha\cos\omega + 2\omega^2 e^{-\alpha}\sinh\alpha + \alpha(\alpha-\beta)$$

where $\beta = -\frac{2c_3}{\lambda_1}$. The values of a_3 and a_4 are obtained from a_1 and a_2, respectively, by changing the sign of α.

We therefore have a parameterization of h in terms of these four parameters. We still must find the values of the parameters that maximize the ratio of integrals that forms our criterion. It can be shown that this ratio is only a function of α, ω, and β. Therefore, the problem of finding the optimal filter has been reduced from an optimization problem in an infinite-dimensional space (the space of admissible functions h) to a nonlinear optimization problem in three variables α, ω, and β. Using constrained numerical optimization, Canny found that the largest value of k that could be obtained was about 3.64. The performance of the filter was then given by $\Sigma\Lambda = 1.12$. The corresponding values of α, β, and ω are

$$\alpha = 2.05220$$

$$\beta = 2.91540$$

$$\omega = 1.56939$$

which yield the following values for the coefficients a_i, $i = 1, \ldots 4$, and λ_1:

$$a_1 = .1486768717$$

$$a_2 = -.2087553476$$

$$a_3 = -1.244653939$$

$$a_4 = -.7912446531$$

$$\lambda_1 = -2$$

This optimal filter (shown in Figure 4.15) is close to the first derivative of a Gaussian

$$C(h) = \Sigma(h)\Lambda(h)\frac{x_{max}(h)}{W}$$

Or, expressing this as a function of h and its derivatives, we have

$$C(h) = \frac{1}{W} \qquad \frac{\int_{-W}^{0} h(x)\,dx}{(\int_{-W}^{0} h^2(x)\,dx)^{1/2}} \times \frac{|\,h'(0)\,|}{(\int_{-W}^{0} h''^2(x)\,dx)^{1/2}}$$

Using the same techniques as in section 4.3.1.4, we have

$$\Phi(x, h, h'') = h^2 + \lambda_1 h + \lambda_2 h''^2$$

The corresponding Euler-Lagrange equation is

$$2h(x) + \lambda_1 + 2\lambda_2 h''''(x) = 0$$

This yields the general solution

$$h(x) = (a_1 \sin \alpha x + a_2 \cos \alpha x)e^{\alpha x} + (a_3 \sin \alpha x + a_4 \cos \alpha x)e^{-\alpha x} - \lambda_1/2$$

where $a_1, a_2, a_3, a_4, \alpha$, and λ_1 are constants that are determined by the various boundary conditions and constraints that we have on h.

It is interesting to note that this solution for h is Canny's solution for $\alpha = \omega$. If we allow W to grow to infinity, the solution becomes

$$h(x) = ae^{\alpha x} \sin \alpha x$$

This is Deriche's solution for $\alpha = \omega$. For $W = 1$, Spacek found the following values:

$$a_1 = -13.3816 \qquad a_2 = 2.7953$$
$$a_3 = 0.0542 \qquad a_4 = -3.7953$$
$$\alpha = 1 \qquad \lambda_1 = -2$$

4.3.2.4 *Comparing the performances of the operators*

In order to compare the performances of the different operators, we have computed the values of Σ, Λ, and x_{max} for the Canny, Deriche, and Spacek operators and the first derivative of a Gaussian (FDG). Analytic expressions can be obtained for the Deriche and FDG filters as shown in table 4.2. We need to define W for an IIR filter $h(x)$ since, in particular,

Table 4.2 The performance indexes of the four filters described in the text.

	Canny	FDG	Deriche	Spacek
Σ	.6093694443	$\sqrt{\frac{2\sigma}{\sqrt{\pi}}}$	$\sqrt{\frac{2}{\alpha}}$	.6039685433
Λ	1.838076985	$\frac{2}{\sqrt{3\sigma\sqrt{\pi}}}$	$\sqrt{2\alpha}$	1.933738156
$\Sigma\Lambda$	1.120067951	$\sqrt{\frac{8}{3\pi}} = .9213177312$	2	1.167917017
x_{max}	1.200000207	$2\pi\sigma\sqrt{\frac{2}{5}}$	$\frac{2\pi}{\sqrt{5}\alpha}$	1.148754323
W	.4235557861	$\sigma\frac{\sqrt{15}}{2}$	$\frac{\sqrt{3}}{\alpha}$	.4086727170
k	2.833157395	$\frac{4\pi}{5}\sqrt{\frac{2}{3}} = 2.052079727$	$\frac{2\pi}{\sqrt{15}} = 1.622311471$	2.810939599
$\Sigma\Lambda k$	3.173328798	1.890617438	3.244622942	3.282944191

the FDG and Deriche filters are of this type. A possible way is to define W through equation (4.3)[10]:

$$W = \sqrt{\frac{\int_{-\infty}^{0} x^2 h^2(x)\,dx}{\int_{-\infty}^{0} h^2(x)\,dx}}$$

The value of the performance index $k = x_{max}/W$ is thus always well defined. This is the definition that we have used in the table. Notice that, because of that definition, W is not equal to 1 for the Canny and Spacek filters. From the table, we see that if we consider only the criterion $\Sigma\Lambda$, the operators can be ranked from best to worst as Deriche, Spacek, Canny, and FDG. If we consider the product $\Sigma\Lambda k$, then the order becomes Spacek, Deriche, Canny, and FDG.

4.4 Discrete implementations

Until now we have been working in the continuous domain. In practice we work with discrete signals and the convolution operations are discrete convolutions. For good introductions to discrete signal processing see

10. We have $x_m = \int_{-\infty}^{+\infty} x h^2(x)\,dx = 0$ since $h^2(x)$ is an even function.

the books by Oppenheim and Schafer and by Gold and Rabiner [OS75, GR78]. Given an impulse response $h(n)$ and an input sequence $i(n)$, the output sequence $o(n)$ is the discrete convolution $h \otimes i(n)$ of h and i:

$$o(n) = \sum_k i(k)h(n-k) = \sum_k h(k)i(n-k) \tag{4.15}$$

The sequence h can be finite or infinite. In the second case, care has to be taken to insure the convergence of equation (4.15). The world of impulse responses is therefore divided into two classes: finite impulse responses (FIR) and infinite impulse responses (IIR).

When implementing convolutions with FIR filters, one has several possibilities. The first is to directly apply equation (4.15). If the length of h is N, to compute one output point we need to perform N multiplications and $N-1$ additions. Another possibility is to use the Fourier transform. If we assume that the length P of the input sequence is larger than N, this technique requires $k \log_2 P$ operations per output point, where the constant k depends on the specific implementation. From this it looks as if dealing with IIR filters is an impossible task unless we truncate the IIR so that it becomes finite. It is not so if we deal with recursive filtering techniques, as we will show next.

4.4.1 Recursive systems

Suppose we have two finite sequences $a(n)$ and $b(n)$ of lengths p and q, respectively. Given an input sequence $i(n)$, let us compute the output sequence $o(n)$ as

$$o(n) = \sum_{k=0}^{p-1} a(k)i(n-k) - \sum_{l=1}^{q} b(l)o(n-l) \tag{4.16}$$

i.e., the ouput at point n is a linear combination of the previous p input points and the previous q output points.[11] The corresponding system

11. Such a system is called *causal*. Noncausal systems are described by the equation

$$o(n) = \sum_{k=0}^{p-1} a(k)i(n+k) - \sum_{l=1}^{q} b(l)o(n+l)$$

is linear and stationary and therefore it is characterized by an impulse response $h(n)$:

$$o(n) = \sum_p h(k)i(n-k)$$

In the case where the sequence $b(n)$ is zero, equation (4.16) reduces to equation (4.15), with $h(n) = a(n)$. If the sequence b is nonzero, then h is infinite [OS75]. Therefore, equation (4.16) provides a way of computing the convolution of an input sequence $i(n)$ with an IIR using a finite number of operations per output point ($p + q$ multiplications and $p + q - 1$ additions). So, given an IIR $h(n)$, if its effect on an input sequence $i(n)$ can be represented as equation (4.16), and if $p + q$ is less than N (the length of the truncated version of $h(n)$ that yields a good FIR approximation to it), then we have a clear way to implement the convolution with h. The problem is that not all convolutions with a sequence $h(n)$ can be represented as equation (4.16). Fortunately, it is true in the case of Deriche filter.

In order to study recursive systems[12], the reader must be familiar with the notion of the z-transform of a sequence, which is also used in the study of the complexity of algorithms as the notion of generative sequences [Knu68]. For more details, the reader is referred to, for example, chapter 2 of the book by Oppenheim and Schafer [OS75]. If we take the z-transform of both sides of equation (4.16), we obtain

$$O(z) = I(z) \sum_{k=0}^{p-1} a(k)z^{-k} - O(z) \sum_{l=1}^{q} b(l)z^{-l}$$

Therefore

$$O(z) = \frac{\sum_{k=0}^{p-1} a(k)z^{-k}}{1 + \sum_{l=1}^{q} b(l)z^{-l}} I(z) \tag{4.17}$$

The key property of a recursive system, which is apparent from this equation, is that the ratio of the z-transforms of the output and the input is a rational function of z. Note also that, given equation (4.17), it is easy to write the recursive relationship (4.16) between the input and output sequences just by reading the coefficients of the rational function.

12. The word *recursive* stems from the fact that the output in equation (4.16) is computed as a linear combination of some of its previous values.

4.4.2 The recursive implementation of Deriche's filter

Let us consider the recursive implementation of Deriche's filter defined by the impulse response

$$h(n) = sne^{-\alpha|n|}$$

Let us split $h(n)$ into two parts

$$h(n) = h_+(n) + h_-(n)$$

where

$$h_+(n) = \begin{cases} sne^{-\alpha n} & n \geq 0 \\ 0 & n < 0 \end{cases}$$

and

$$h_-(n) = \begin{cases} 0 & n > 0 \\ sne^{\alpha n} & n \leq 0 \end{cases}$$

Because the z-transform is a linear operation, the z-transform $H(z)$ of $h(n)$ is the sum of the z-transform $H_+(z)$ of $h_+(n)$ and $H_-(z)$ of $h_-(n)$. Let us compute both:

$$H_+(z) = s\sum_{n=0}^{+\infty} ne^{-\alpha n}z^{-n} = sz\sum_{n=0}^{+\infty} ne^{-\alpha n}z^{-n-1} = szG_+(z)$$

Clearly

$$G_+(z) = -\frac{d}{dz}K_+(z)$$

where

$$K_+(z) = \sum_{n=0}^{+\infty} e^{-\alpha n}z^{-n} = \frac{1}{1 - e^{-\alpha}z^{-1}}$$

The series is convergent for $|z| > e^{-\alpha}$. Finally

$$H_+(z) = \frac{se^{-\alpha}z^{-1}}{(1 - e^{-\alpha}z^{-1})^2}$$

This describes a causal system with a double pole at $e^{-\alpha}$. Similarly we find that

$$H_-(z) = -\frac{se^{-\alpha}z}{(1 - e^{-\alpha}z)^2}$$

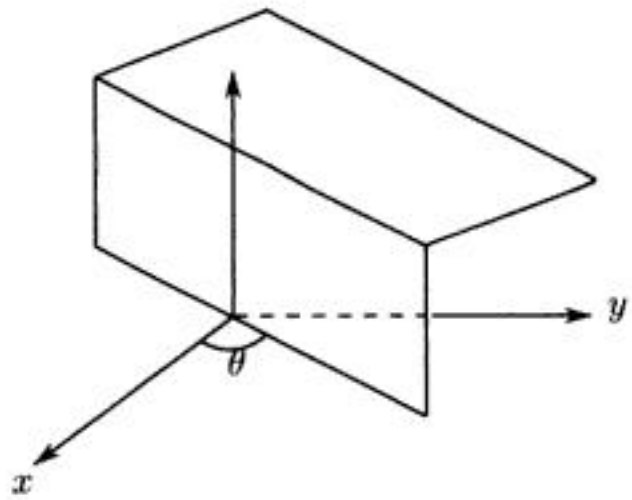


Figure 4.17 A two-dimensional constant-intensity step edge.

$k(x)k(y)$. Since $h(x) = \frac{d}{dx}k(x)$, by applying the convolution theorem to $f \otimes K(x,y)$ we have

$$\frac{\partial}{\partial x}(f \otimes K(x,y)) = f \otimes h(x)k(y)$$

$$\frac{\partial}{\partial y}(f \otimes K(x,y)) = f \otimes k(x)h(y)$$

From this it is clear that the gradient of F can be computed by convolving the rows (the columns, respectively) of f with those of h and k. The advantage of this approach is that the filtering operations required to compute the gradient of the smoothed image are separable (i.e., rows, columns), and therefore one-dimensional, which can be extremely efficient in terms of computation. The disadvantage is that the smoothing is nonisotropic: The directions 45 degrees and 135 degrees undergo more smoothing than 0 degrees and 90 degrees, and it is expected that edges in these directions will be detected more accurately than in others, the worst being the cases of 45 degrees and 135 degrees.

In order to see this better, let us assume that f is one-dimensional, i.e., $f(x,y) \equiv f(x)$. This implies[13] that

13. We assume that k is normalized so that $\int k(x)\,dx = 1$.

$$\begin{cases} \frac{\partial}{\partial x}(f \otimes K(x,y)) & = f \otimes h(x) \\ \frac{\partial}{\partial y}(f \otimes K(x,y)) & = 0 \end{cases}$$

The only nonzero component of the gradient of F is thus obtained by convolving f with h. Let us then suppose that f is rotated by 45 degrees, i.e., $f(x,y) \equiv f(x+y)$. The partial derivative in that direction is the projection of the gradient in that direction, and therefore is proportional to $(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}) f \otimes K(x,y)$:

$$f(x,y) \otimes [h(x)k(y) + k(x)h(y)]$$

Applying the change of variable $u = x + y$ and $v = x$, this is the same as

$$f(u) \otimes [h(v)k(u-v) + h(u-v)k(v)]$$

The equivalent impulse response in u is

$$\int (h(v)k(u-v) + h(u-v)k(v))\, dv = 2h \otimes k(u)$$

The result is proportional to

$$(f \otimes k) \otimes h(u)$$

The only nonzero component of the gradient of F is thus obtained by first smoothing f with k and then convolving the result with h. Therefore f is subjected to more smoothing in the 45 degrees direction than in the 0 degrees or 90 degrees directions (the extra smoothing corresponding to the convolution with k).

In order to alleviate this problem, we can take K to be isotropic: $K(x,y) \equiv k(r)$, $(r = \sqrt{x^2 + y^2})$. The gradient is given by

$$\frac{\partial}{\partial x}(f \otimes k(r)) = f \otimes h(r)\frac{x}{r}$$

$$\frac{\partial}{\partial y}(f \otimes k(r)) = f \otimes h(r)\frac{y}{r}$$

The partial derivative in the direction θ is given by

$$f \otimes \frac{h(r)}{r}(x\cos\theta + y\sin\theta)$$

Suppose now that f is only a function in the direction $\theta : f(x, y) \equiv f(x\cos\theta + y\sin\theta)$. By applying the change of variable $u = x\cos\theta + y\sin\theta$ and $v = -x\sin\theta + y\cos\theta$, we obtain

$$f(u) \otimes \frac{h(\rho)}{\rho} u \quad \text{where } \rho = \sqrt{u^2 + v^2}$$

The equivalent impulse response, a function of u, is

$$u \int \frac{h(\rho)}{\rho} dv$$

which is in general different from $h(u)$. Therefore we still have the same problem as in the separable case: An edge is not detected by convolving the intensity distribution perpendicular to its direction with h. The difference is that the error is uniformly distributed in all directions. From an implementation standpoint, due to the simplicity of separable convolutions, the first solution appears more attractive.

4.5.2 Finding edge pixels

Having computed the smoothed gradient at each pixel in the image, potential edge pixels are selected by choosing those that are local maxima of the gradient magnitude in the direction of the gradient. This technique is called *nonmaxima suppression*. We assume that at an edge, the gradient $\mathbf{g}$ of F, is normal to the edge, and that its magnitude there reaches a local maximum along a cross-section of the smoothed intensity image taken along its direction. This is equivalent to saying that an edge will be detected at zero-crossings of the second derivative $D_{\mathbf{g}}^2 F$ of the smoothed intensity F in the direction of the gradient. $D_{\mathbf{g}}^2 F$ is defined as a function of F by equation (4.8).

It is worthwhile to note that this definition of an edge point does not necessarily imply that $\mathbf{g}$ is parallel to the normal of the edge curve. In order to see this, we notice that the edge curve (c) is defined by the equation

$$e(x, y) = \mathbf{g}^T \mathbf{H} \mathbf{g} = 0 \tag{4.18}$$

Therefore its normal is parallel to the vector ∇e, which is found to be equal to

$$\nabla e = (2\mathbf{H}^2 + \mathbf{R})\mathbf{g} \tag{4.19}$$

where

$$\mathbf{R} = \begin{bmatrix} (\mathbf{H}_x\mathbf{g})^T \\ (\mathbf{H}_y\mathbf{g})^T \end{bmatrix}, \quad \mathbf{H}_x = \begin{bmatrix} F_{x^3} & F_{x^2y} \\ F_{x^2y} & F_{xy^2} \end{bmatrix}$$

is the partial derivative of the Hessian $\mathbf{H}$ with respect to x and $\mathbf{H}_y$ its derivative with respect to y.

There is no reason why the vector defined by equation (4.19) should be parallel to $\mathbf{g}$. In fact it is fairly easy to characterize those image curves that satisfy the property that their normals are parallel to the image gradient at every point. Consider figure 4.18, in which we have represented the image curve (c) in the xy-plane and the image surface (Σ) of equation $z = F(x, y)$. The curve (c) is the parallel projection of a curve (C) on the intensity surface along the z-direction. Let m be a point of (c), and let M be the corresponding point of (C). Let $\mathbf{G}$ (see section 4.1.2.1) be the gradient of the image intensity surface at M, $\mathbf{T}$ the unit vector tangent to (C), $\mathbf{t}$ the unit vector tangent to (c), and $\mathbf{g}$ the projection of $\mathbf{G}$. Since $\mathbf{G}$ is normal to the intensity surface and $\mathbf{T}$ is tangent to a curve on this surface, we have

$$\mathbf{T} \cdot \mathbf{G} = g_x T_x + g_y T_y - T_z = 0 \tag{4.20}$$

But since $\mathbf{t}\sqrt{T_x^2 + T_y^2} = [T_x, T_y]^T$, we also have

$$(\mathbf{t} \cdot \mathbf{g})\sqrt{T_x^2 + T_y^2} = g_x T_x + g_y T_y$$

which is equal to T_z (equation (4.20)). Therefore $\mathbf{t} \cdot \mathbf{g} = 0$ is equivalent to $T_z = 0$, i.e., the curve (C) must be the intersection of the image intensity surface with a plane of constant z. This means that the intensity does not vary along (c), whose equation is therefore $F(x, y) = constant$. In practice, it is often the case that this condition is satisfied even though there may be cases where it is not.

If this is true, then, as shown in figure 4.19, it is easy to deduce a value for the curvature κ of the contour considered. Indeed, if we consider the unit normal $\mathbf{n} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$ and the unit tangent $\mathbf{t} = \frac{\mathbf{g}_T}{\|\mathbf{g}_T\|}$, we have the following relation:

$$\mathbf{t} \cdot \mathbf{g} = 0$$

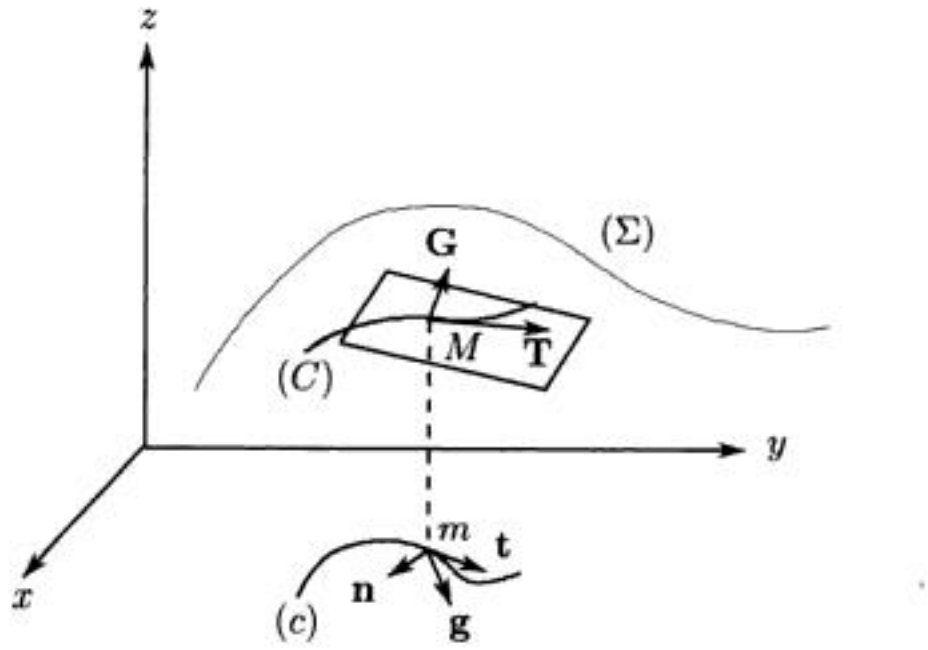


Figure 4.18 In order for **g** to be parallel to the normal **n** to (c), (C) has to be a special curve on (Σ).

Differentiating this expression with respect to the arclength s of the edge curve and using the Frenet formula (see appendix C), we obtain

$$\kappa(\mathbf{n} \cdot \mathbf{g}) + \mathbf{t}^T \mathbf{H} \mathbf{t} = 0$$

since $\frac{d\mathbf{g}}{ds} = \frac{\partial \mathbf{g}}{\partial x}\frac{dx}{ds} + \frac{\partial \mathbf{g}}{\partial y}\frac{dy}{ds} = \mathbf{H}\mathbf{t}$. Using equation (4.9), this yields

$$\kappa = -\frac{\mathbf{g}_\tau^T \mathbf{H} \mathbf{g}_\tau}{\|\mathbf{g}\|^3} = \frac{2F_x F_y F_{xy} - F_{x^2} F_y^2 - F_{y^2} F_x^2}{(F_x^2 + F_y^2)^{3/2}}$$

But because $D_{\mathbf{g}}^2 F = 0$, equation (4.8) yields

$$2F_x F_y F_{xy} = -F_x^2 F_{x^2} - F_y^2 F_{y^2}$$

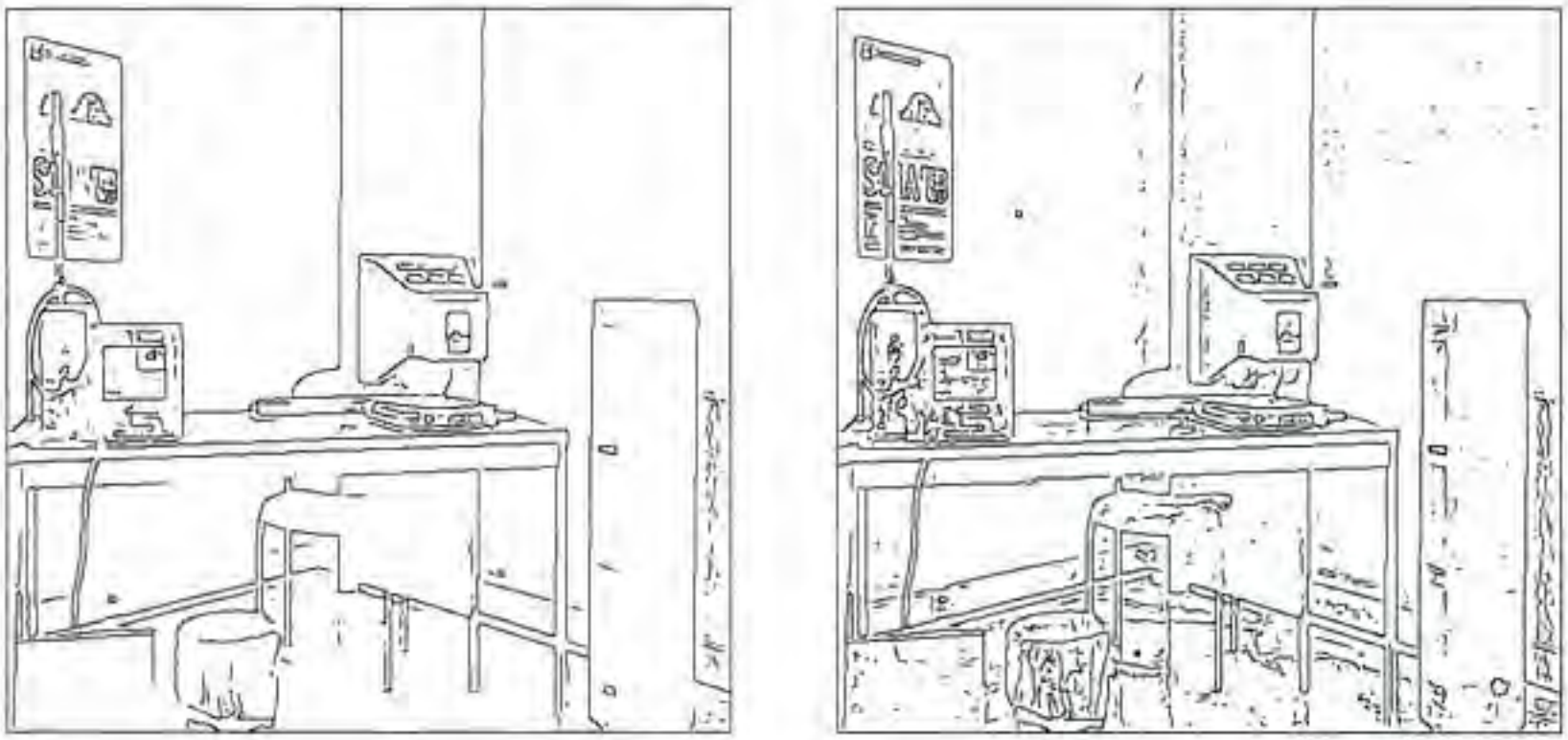
which leads to

$$\kappa = -\frac{F_{x^2} + F_{y^2}}{(F_x^2 + F_y^2)^{1/2}} = -\frac{\nabla^2 F}{\|\mathbf{g}\|} \tag{4.21}$$

Equation (4.21) yields the curvature of the edge curve as a function of the image intensities. If the hypothesis that the smoothed image intensity is constant along the edge curve is not true, then the computation is more involved (see problem 5).

linearly interpolating between the values at pixels A_3 and A_4 for B and between those at pixels A_7 and A_8 for C. Pixel A is marked as an edge pixel only if the value at A is strictly greater than the values at B and C. The result of this operation is a binary image where, for example, black pixels indicate an edge.

In figure 4.21 we show results of the Deriche edge detector applied to a variety of images and with different values of the parameter α controlling the width of the impulse response. It is clear from these results that too many edge points are being detected, especially in regions where the contrast is low. In this respect, the situation is analogous to that of figure 4.11. If we want to cut down on the number of edge pixels, we must add some thresholding operation. Thresholding is a plague that occurs in many areas in engineering, but to our knowledge it is unavoidable and must be tackled with courage. Here the idea is to keep only the edge pixels for which the gradient norm is above some threshold T. If we set T too low, then too many pixels are still present, and if we set it too high, then connected chains of edge pixels start breaking into smaller chains, which is highly undesirable (figure 4.22).



A threshold of 15 and $\alpha = 1$ A threshold of 5 and $\alpha = 1$

Figure 4.22 The difficulty of choosing the right threshold.

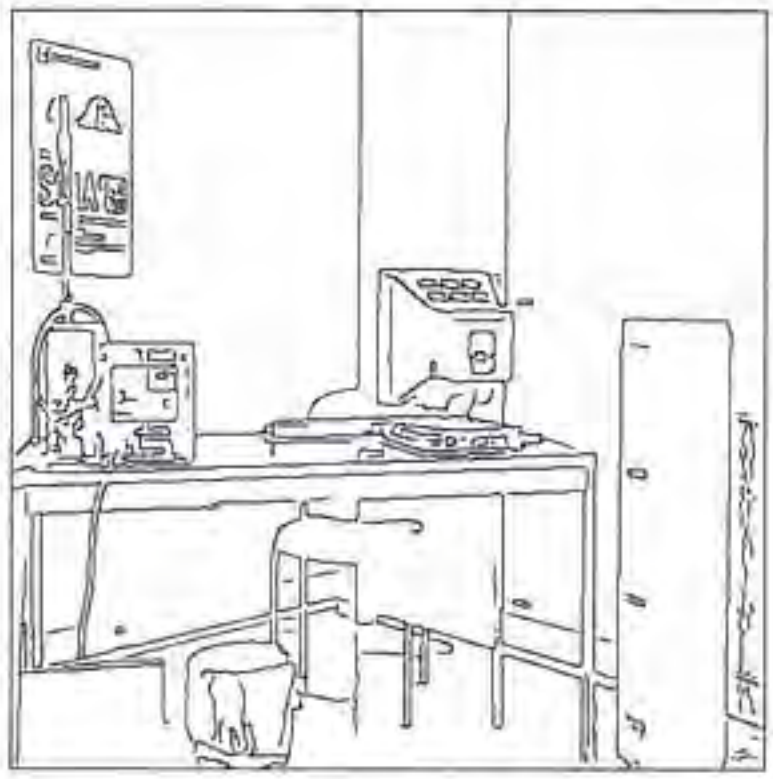


Figure 4.23 Hysteresis thresholding for the Deriche operator ($\alpha = 1$, $T_2 = 5$, $T_1 = 15$).

The basic question is this: How do we choose T? There is no good answer to this question, and the choice of T must be guided by the application and the lighting conditions of the scene. A variant of this thresholding idea has been introduced by Canny [Can83]. It is called *hysteresis thresholding*, and its purpose to keep the edge pixels connected as much as possible. To achieve this goal, we use two thresholds T_1 and T_2 with $T_2 < T_1$. If we think of a chain of connected edge pixels as a whole, then if for one pixel in the chain the gradient norm is higher than T_1, we will keep all pixels in the chain with a gradient norm larger than the lower threshold T_2. The result of this operation is shown in figure 4.23, which is to be compared with results of figure 4.22.

4.6 More references

The approach followed by Canny of using an edge model and of minimizing a criterion in order to find the "best" filter can be tracked further in the past, for example to Modestino and Fries [Mod77], who also proposed

a recursive implementation of their filter, and to Shanmugam, Dickey, and Green [SDG79]. An interesting variation on this theme can be found in the work of Shen and Castan [SC92]. An extension of these ideas to the detection of three-dimensional edges can be found in the article by Monga, Deriche, and Rocchisani [MDR91]. Another approach, very different from the ones presented in this chapter, considers the problem of finding edge curves in an image as one of finding sequences of connected pixels maximizing some criterion defined on the discrete image considered as a graph. These curves are then found using either heuristic search techniques such as the A^* algorithm [Mar72] or dynamic programming (see chapter 6) [Mon71]. These methods can also be extended to take into account sophisticated statistical models as proposed by Basseville [BEG81, Bas81].

Variants of the idea of smoothing the data by approximation, or surface-fitting, which was presented in section 4.2.4.1, can also be found in the original paper by Hueckel [Hue71] and in a paper by Nalwa and Binford [Nal86]. Yet another very different approach is to use the so-called Hough transform [Hou62] to detect edge lines as parameterized curves [Dud72, KBS75] (see also chapter 11). Finally, there are three excellent books, one by Rosenfeld and Kak [RK82], one by Martin Levine [Lev85], and one by Anil Jain [Jai89], which cover the basic issues of image processing, including edge detection.

4.7 Problems

1. In this problem we show that the solution to the discrete smoothing problem (see section 4.2.4) is a cubic spline. Given some measurements f_k, we look for a C^2 function $S(x)$ that minimizes the following criterion:

$$C(S) = \sum_k (f_k - S(x_k))^2 + \lambda \int S''(x)^2 \, dx \tag{4.22}$$

   a. Show that the criterion (4.22) is a convex functional, i.e., that

   $$C(\alpha S_1 + (1-\alpha)S_2) \leq \alpha C(S_1) + (1-\alpha)C(S_2)$$

   for all functions S_1 and S_2, and $0 \leq \alpha \leq 1$.

b. Show that (4.22) can be written as

$$\int [\lambda S''(x)^2 + (S(x) - f(x))^2 \sum_k \delta(x - x_k)]\, dx$$

and that the corresponding Euler-Lagrange equation is

$$\lambda S^{(4)}(x) + S(x) \sum_k \delta(x - x_k) = \sum_k f_k \delta(x - x_k)$$

c. Since the criterion C is convex, the minimum of criterion (4.22) is unique. We will now show that, if the x_k are evenly spaced, S can be determined in terms of the f_k by a convolution for infinite or periodic sequences. Let $S(x) = R \otimes f(x)$. Since we know f only at the points x_k, we can take $f(x) = \sum_k f_k \delta(x - x_k)$. Show that the coefficient of f_k in the Lagrange-Euler equation is

$$\lambda R^{(4)}(x - x_k) + \sum_l R(x_l - x_k)\delta(x - x_l) - \delta(x - x_k) \qquad (4.23)$$

d. Show that if the x_k are not evenly spaced these equations are inconsistent and that if they are evenly spaced they reduce to the following equation:

$$\lambda R^{(4)}(x) + \sum_l R(x_l)\delta(x - x_l) - \delta(x) = 0 \qquad (4.24)$$

e. We will now show that the solutions to equation (4.24) correspond to cubic splines "stitched" together at the points x_k. Let $R_k(x)$ denote the solution in the range $x_k \le x \le x_{k+1}$ and write $R_k(x) = \alpha_k x^3 + \beta_k x^2 + \gamma_k x + \delta_k$. Write three algebraic conditions on the coefficients $\alpha_k, \beta_k, \gamma_k, \delta_k, \alpha_{k-1}, \beta_{k-1}, \gamma_{k-1}$, and δ_{k-1} expressing the fact that $R(x)$ is C^2 at x_k.

f. Show that $R^{(3)}(x)$ has a discontinuity of $-\frac{R(x_k)}{\lambda}$ at x_k. *Hint:* Integrate equation (4.24), and find a simple relationship between α_k and α_{k-1}.

g. Use the results of questions e and f to determine three first-order recursive relationships between β_k and β_{k-1}, γ_k and γ_{k-1}, and δ_k and δ_{k-1}.

2. Work out the minimization problem in the Haralick edge detector.

3. Let $n(x)$ be a stationary process with autocorrelation function $\Lambda_n(\tau) = E(n(x)n(x+\tau))$. If we convolve n with an impulse response $h(x)$, we obtain a new random process $m(x) = n \otimes h(x)$. Show that the autocorrelation $\Lambda_m(\tau)$ of m is equal to

   $$\Lambda_n \otimes g(\tau)$$

   where $g(\tau) = h \otimes h_-(\tau)$, and $h_-(\tau) = h(-\tau)$.

4. In this problem, inspired by Perona and Malik [PM90], we prove the result of Rice about the mean distance between zero-crossings of a stationary Gaussian random process (see section 4.3.1.4). Let $n(x)$ be a white stationary Gaussian process and $f(x)$ be a twice-differentiable impulse response. We will consider the filtered Gaussian process $n_f(x) = f \otimes n(x)$. We are going to compute the average spacing x_{ave} between positive-derivative zero-crossings of $n_f(x)$.

   a. A necessary and sufficient condition for x to be a positive-derivative zero of n_f is that $n_f(x) = 0$ and $n'_f(x) > 0$. Considering the function $H(n_f(x))$ and its derivative (H is the Heaviside function), build a function $s(x)$ that is infinite at the positive-derivative zeros of n_f and zero elsewhere.

   b. The quantity we want to compute is the reciprocal of the average value of $s(x)$. Using standard properties of the Fourier transform, show that

   $$E(s) = \frac{1}{4\pi^2} \int\int \frac{\partial E(e^{2\pi i(pn_f + qn'_f)})}{\partial q} dpdq$$

   c. Taking the density probability of n equal to $e^{-\frac{1}{2}n^2(t)dt}$, and letting $k = 2i\pi$, show that

   $$E(e^{2\pi i(pn_f + qn'_f)})$$
   $$= \int e^{-\frac{1}{2}\int[(n(t) - k[pf(x-t) + qf'(x-t)])^2 - k^2[pf(x-t) + qf'(x-t)]^2]dt} dn$$

   d. Show that this is equal to

   $$e^{\frac{1}{2}k^2(p^2R_f(0) - q^2R''_f(0))}$$

5 Representing Geometric Primitives and Their Uncertainty

In this chapter we will study in detail how to represent some fundamental geometric primitives. A *representation* is a mapping from the set of geometric primitives under study to a set of numerical parameters, a subset of R^n. The variable n is the dimension of the representation. Given the problem of choosing among several representations of a set of geometric primitives, it is important to ask the following questions [MK78, BR78]:

1. **Is the representation unique?** Does every representable geometric primitive have a unique representation? This is equivalent to saying that the previous mapping is one-to-one.
2. **Is the representation complete?** In other words, does every geometric primitive admit a representation?
3. **Is the representation minimal?** This means that the number of parameters used in the representation is minimal. This number is characteristic of the set of geometric primitives and is called its *dimension*. It is important to note that that there exist nonminimal representations whose dimensions are larger than the dimension of the set of geometric primitives.[1]
4. **Is the representation smooth?** This means that, if a geometric primitive varies smoothly, then its representation also varies smoothly.

1. There may also exist representations whose dimensions are smaller than the dimension of the set of geometric primitives. We exclude those from our consideration because they do not allow for the reconstruction of the primitives.

Of course, we do not always insist on having representations that satisfy all these requirements. It depends on the application.

There is a very natural way of thinking about these questions. Mathematically speaking, the relevant notion is that of a *manifold*. Rather than rediscovering this notion for every case, we will give the general definition. Interestingly enough, this notion, which is forced upon us even by examples as simple as projective points in the plane, implies that we abandon in general the simple idea of representing a set of geometric primitives by a single mapping into a set of parameters. We need several mappings and, as a consequence, a single primitive has several representations. The key idea of manifolds is that all those representations are smoothly related.

5.1 How to read this chapter

This chapter is organized into two almost independent sets of sections. In sections 5.2 to 5.5 we discuss the problems of representations, while in section 5.6 we discuss the problem of computing the uncertainty of parameters that are nonlinear functions, either explicit or implicit, of some measurements. This problem occurs often in many applications of computer vision and can be considered of great practical importance. In the case where the functions of the measurements are explicitly known, the result is summarized in equation (5.28), which gives the covariance matrix of the parameters as a function of the Jacobian matrix and the covariance matrix of the measurements. We also give a deterministic interpretation of this result that avoids the need for introducing probabilities, which we believe to be unnatural most of the time. This can be skipped on the first reading. In the frequent case where the parameters are obtained by minimizing some criterion function, the Jacobian matrix is not directly available but can be obtained by applying the implicit functions theorem. Section 5.6.4 gives an example of the application of this result to a very common practical problem and requires only some simple results from section 5.3.3.2.

The first four sections of the chapter fill two different needs. The first need is to establish a solid theoretical basis for the representation of ge-

ometric primitives such as points, lines, planes, orientations, directions, and displacements, which are routinely manipulated in computer vision problems. The second need is to give us tools for chapters 6, 8, and 11. The reader who is interested only in the tools can read the definition of a manifold in section 5.2, section 5.3.3 on the representation of 2-D lines, section 5.4.3 on the representation of planes, section 5.4.4 on the representation of 3-D lines, section 5.5.2 on quaternions, and section 5.5.4 on exponentials of antisymmetric matrices.

5.2 Manifolds

Suppose we are given a set X of primitives (see figure 5.1). It is convenient to think of those primitives as points in R^n forming a subset of that space. We assume that X is such that there exists a family U_i of open sets of R^n covering X and such that, for each U_i, there exists a one-to-one mapping φ_i from $U_i \cap X$ into R^d. The maps φ_i should be thought of as the representations of the set of primitives X. They must satisfy the following *coherence condition:*

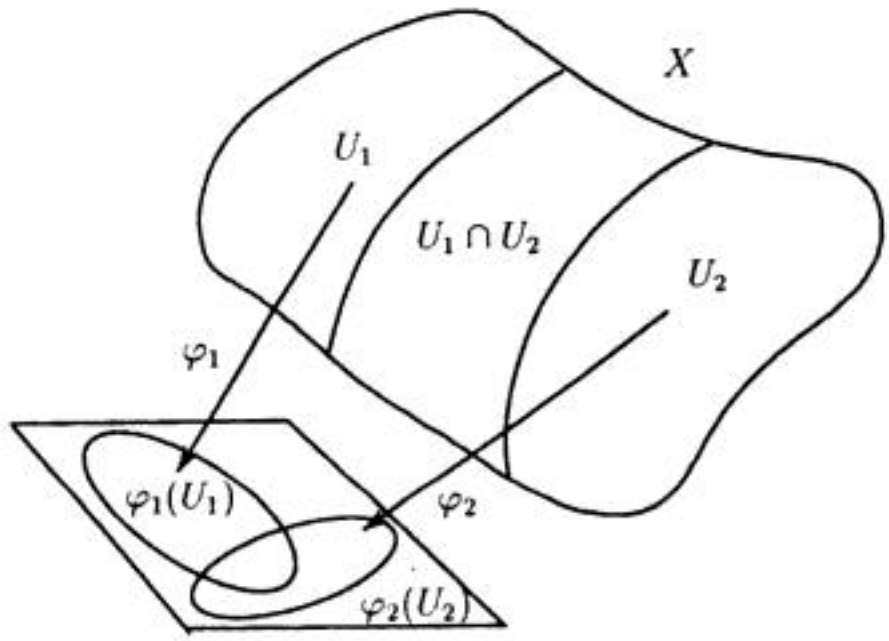


Figure 5.1 A manifold of dimension 2.

THREE-DIMENSIONAL COMPUTER VISION
A Geometric Viewpoint
Olivier Faugeras

This monograph by one of the world's leading vision researchers provides a thorough, mathematically rigorous exposition of a broad and vital area in computer vision: the problems and techniques related to three-dimensional (stereo) vision and motion. The emphasis is on using geometry to solve problems in stereo and motion, with examples from navigation and object recognition.

Faugeras takes up such important problems in computer vision as projective geometry, camera calibration, edge detection, stereo vision (with many examples on real images), different kinds of representations and transformations (especially 3-D rotations), uncertainty and methods of addressing it, and object representation and recognition. His theoretical account is illustrated with the results of actual working programs.

Olivier Faugeras is Research Director of the Computer Vision and Robotics Laboratory at INRIA Sophia-Antipolis and a Professor of Applied Mathematics at the Ecole Polytechnique in Paris. *Three-Dimensional Computer Vision* is included in the Artificial Intelligence series, edited by Michael Brady, Daniel Bobrow, and Randall Davis.

"A magnificent tour de force. *Three-Dimensional Computer Vision* deals with an extremely broad and important chunk of computer vision and covers the area with excellent breadth. It provides examples of the described techniques being applied to real images, and it is built on the kind of solid mathematical underpinnings that are essential if the field is to move from the 'black art' stage to a real science. Anyone who claims to be serious about research in this area absolutely must be aware of this work."
— W. Eric L. Grimson, Artificial Intelligence Laboratory, Massachusetts Institute of Technology

"This is an excellent book that will be required reading for anyone seriously interested in problems involving stereo and/or motion. It is one of the few examples of developing theory using solid mathematics and then demonstrating that the theory is actually useful in practice. The presentation is consistent and the parts fit well together. I expect it to become a classic in the field of computer vision."
— William Thompson, Department of Computer Science, University of Utah

Cover art: panel from Albrecht Dürer, *Draftsman Drawing a Reclining Nude (The Manual of Measurement)*, c.1527. The Samuel Putnam Avery Fund. Reproduced courtesy of the Museum of Fine Arts, Boston.

THE MIT PRESS
Massachusetts Institute of Technology
Cambridge, Massachusetts 02142

0-262-06158-9

780262 061582 90000