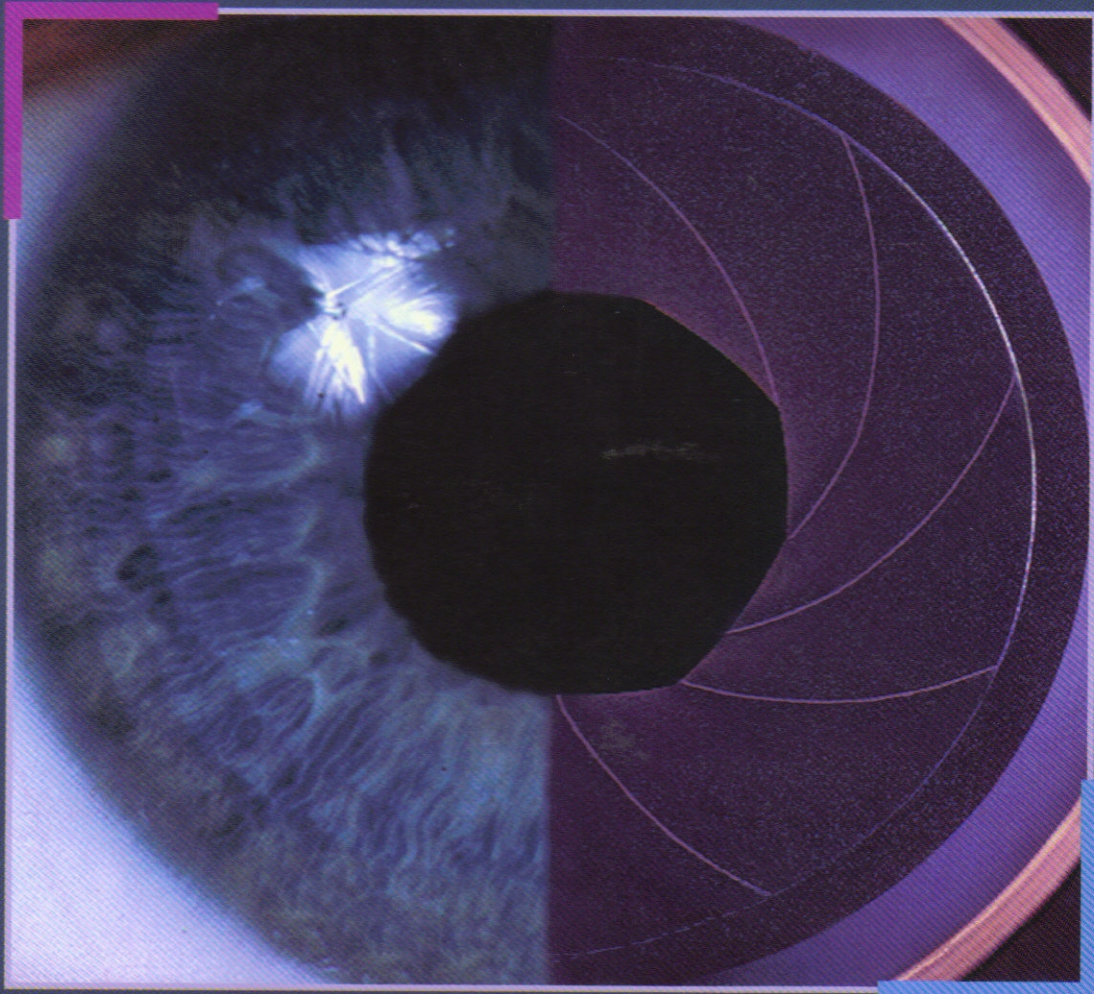


MACHINE VISION



Ramesh Jain • Rangachar Kasturi • Brian G. Schunck

INTEL-1041
10,142,659

MACHINE VISION

Ramesh Jain

University of California, San Diego

Rangachar Kasturi

Pennsylvania State University

Brian G. Schunck

Adept Technology

McGraw-Hill, Inc.

New York	St. Louis	San Francisco	Auckland	Bogotá	
Caracas	Lisbon	London	Madrid	Mexico City	Milan
Montreal	New Delhi	San Juan	Singapore	Sydney	
Tokyo	Toronto				

Machine Vision

Copyright ©1995 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

This book is printed on acid-free paper.

4 5 6 7 8 9 0 FGR FGR 9 0 9 8

ISBN 0-07-032018-7

The editor was Eric M. Munson;
the production supervisor was Kathryn Porzio.
The cover was designed by BC Graphics.
Project supervision was done by Publication Services, Inc.
Quebecor Printing/Fairfield was printer and binder.

Library of Congress Cataloging-in-Publication Data

Jain, Ramesh.

Machine vision / Ramesh Jain, Rangachar Kasturi, Brian G. Schunck.

p. cm.

Includes index.

ISBN 0-07-032018-7

- | | |
|-----------------------|--------------------------------|
| 1. Computer vision. | I. Kasturi, Rangachar, (date). |
| II. Schunck, Brian G. | III. Title. |
| TA1634.J35 | 1995 |
| 006.4'2—dc20 | |

95-3771

Contents

Preface	xvii
Acknowledgments	xix
1 Introduction	1
1.1 Machine Vision	1
1.2 Relationships to Other Fields	4
1.3 Role of Knowledge	5
1.4 Image Geometry	6
1.4.1 Perspective Projection	8
1.4.2 Coordinate Systems	9
1.5 Sampling and Quantization	10
1.6 Image Definitions	12
1.7 Levels of Computation	13
1.7.1 Point Level	14
1.7.2 Local Level	15
1.7.3 Global Level	16
1.7.4 Object Level	17
1.8 Road Map	18
2 Binary Image Processing	25
2.1 Thresholding	28
2.2 Geometric Properties	31
2.2.1 Size	31

2.2.2	Position	32
2.2.3	Orientation	33
2.3	Projections	35
2.4	Run-Length Encoding	38
2.5	Binary Algorithms	39
2.5.1	Definitions	40
2.5.2	Component Labeling	44
2.5.3	Size Filter	47
2.5.4	Euler Number	48
2.5.5	Region Boundary	50
2.5.6	Area and Perimeter	50
2.5.7	Compactness	51
2.5.8	Distance Measures	52
2.5.9	Distance Transforms	53
2.5.10	Medial Axis	55
2.5.11	Thinning	57
2.5.12	Expanding and Shrinking	60
2.6	Morphological Operators	61
2.7	Optical Character Recognition	70
3	Regions	73
3.1	Regions and Edges	73
3.2	Region Segmentation	76
3.2.1	Automatic Thresholding	76
3.2.2	Limitations of Histogram Methods	86
3.3	Region Representation	86
3.3.1	Array Representation	88
3.3.2	Hierarchical Representations	88
3.3.3	Symbolic Representations	90
3.3.4	Data Structures for Segmentation	92
3.4	Split and Merge	96
3.4.1	Region Merging	97
3.4.2	Removing Weak Edges	100
3.4.3	Region Splitting	103
3.4.4	Split and Merge	104
3.5	Region Growing	105

4	Image Filtering	112
4.1	Histogram Modification	112
4.2	Linear Systems	115
4.3	Linear Filters	118
4.4	Median Filter	122
4.5	Gaussian Smoothing	123
4.5.1	Rotational Symmetry	127
4.5.2	Fourier Transform Property	128
4.5.3	Gaussian Separability	129
4.5.4	Cascading Gaussians	129
4.5.5	Designing Gaussian Filters	132
4.5.6	Discrete Gaussian Filters	134
5	Edge Detection	140
5.1	Gradient	143
5.2	Steps in Edge Detection	145
5.2.1	Roberts Operator	146
5.2.2	Sobel Operator	147
5.2.3	Prewitt Operator	148
5.2.4	Comparison	148
5.3	Second Derivative Operators	149
5.3.1	Laplacian Operator	149
5.3.2	Second Directional Derivative	156
5.4	Laplacian of Gaussian	157
5.5	Image Approximation	162
5.6	Gaussian Edge Detection	168
5.6.1	Canny Edge Detector	169
5.7	Subpixel Location Estimation	173
5.8	Edge Detector Performance	176
5.8.1	Methods for Evaluating Performance	177
5.8.2	Figure of Merit	178
5.9	Sequential Methods	179
5.10	Line Detection	180
6	Contours	186
6.1	Geometry of Curves	188
6.2	Digital Curves	188
6.2.1	Chain Codes	189

6.2.2	Slope Representation	191
6.2.3	Slope Density Function	191
6.3	Curve Fitting	192
6.4	Polyline Representation	194
6.4.1	Polyline Splitting	196
6.4.2	Segment Merging	196
6.4.3	Split and Merge	198
6.4.4	Hop-Along Algorithm	199
6.5	Circular Arcs	200
6.6	Conic Sections	203
6.7	Spline Curves	207
6.8	Curve Approximation	210
6.8.1	Total Regression	212
6.8.2	Estimating Corners	214
6.8.3	Robust Regression	214
6.8.4	Hough Transform	218
6.9	Fourier Descriptors	223
7	Texture	234
7.1	Introduction	234
7.2	Statistical Methods of Texture Analysis	236
7.3	Structural Analysis of Ordered Texture	239
7.4	Model-Based Methods for Texture Analysis	240
7.5	Shape from Texture	241
8	Optics	249
8.1	Lens Equation	250
8.2	Image Resolution	250
8.3	Depth of Field	251
8.4	View Volume	253
8.5	Exposure	254
9	Shading	257
9.1	Image Irradiance	257
9.1.1	Illumination	259
9.1.2	Reflectance	261
9.2	Surface Orientation	264

9.3	The Reflectance Map	267
9.3.1	Diffuse Reflectance	267
9.3.2	Scanning Electron Microscopy	268
9.4	Shape from Shading	269
9.5	Photometric Stereo	271
10	Color	276
10.1	Color Physics	276
10.2	Color Terminology	277
10.3	Color Perception	278
10.4	Color Processing	280
10.5	Color Constancy	284
10.6	Discussion	286
11	Depth	289
11.1	Stereo Imaging	289
11.1.1	Cameras in Arbitrary Position and Orientation	291
11.2	Stereo Matching	293
11.2.1	Edge Matching	294
11.2.2	Region Correlation	295
11.3	Shape from X	298
11.4	Range Imaging	300
11.4.1	Structured Lighting	301
11.4.2	Imaging Radar	305
11.5	Active Vision	305
12	Calibration	309
12.1	Coordinate Systems	311
12.2	Rigid Body Transformations	313
12.2.1	Rotation Matrices	316
12.2.2	Axis of Rotation	318
12.2.3	Unit Quaternions	318
12.3	Absolute Orientation	320
12.4	Relative Orientation	325
12.5	Rectification	331
12.6	Depth from Binocular Stereo	332
12.7	Absolute Orientation with Scale	334

12.8	Exterior Orientation	336
12.8.1	Calibration Example	340
12.9	Interior Orientation	341
12.10	Camera Calibration	346
12.10.1	Simple Method for Camera Calibration	347
12.10.2	Affine Method for Camera Calibration	352
12.10.3	Nonlinear Method for Camera Calibration	355
12.11	Binocular Stereo Calibration	357
12.12	Active Triangulation	359
12.13	Robust Methods	361
12.14	Conclusions	361
13	Curves and Surfaces	365
13.1	Fields	366
13.2	Geometry of Curves	367
13.3	Geometry of Surfaces	369
13.3.1	Planes	369
13.3.2	Differential Geometry	370
13.4	Curve Representations	373
13.4.1	Cubic Spline Curves	373
13.5	Surface Representations	374
13.5.1	Polygonal Meshes	374
13.5.2	Surface Patches	378
13.5.3	Tensor-Product Surfaces	380
13.6	Surface Interpolation	381
13.6.1	Triangular Mesh Interpolation	381
13.6.2	Bilinear Interpolation	382
13.6.3	Robust Interpolation	384
13.7	Surface Approximation	385
13.7.1	Regression Splines	387
13.7.2	Variational Methods	395
13.7.3	Weighted Spline Approximation	395
13.8	Surface Segmentation	397
13.8.1	Initial Segmentation	398
13.8.2	Extending Surface Patches	399
13.9	Surface Registration	400

14	Dynamic Vision	406
14.1	Change Detection	408
14.1.1	Difference Pictures	409
14.1.2	Static Segmentation and Matching	415
14.2	Segmentation Using Motion	416
14.2.1	Time-Varying Edge Detection	416
14.2.2	Stationary Camera	418
14.3	Motion Correspondence	420
14.4	Image Flow	428
14.4.1	Computing Image Flow	428
14.4.2	Feature-Based Methods	429
14.4.3	Gradient-Based Methods	429
14.4.4	Variational Methods for Image Flow	431
14.4.5	Robust Computation of Image Flow	432
14.4.6	Information in Image Flow	435
14.5	Segmentation Using a Moving Camera	436
14.5.1	Ego-Motion Complex Log Mapping	437
14.5.2	Depth Determination	439
14.6	Tracking	443
14.6.1	Deviation Function for Path Coherence	444
14.6.2	Path Coherence Function	445
14.6.3	Path Coherence in the Presence of Occlusion	447
14.6.4	Modified Greedy Exchange Algorithm	448
14.7	Shape from Motion	451
15	Object Recognition	459
15.1	System Components	460
15.2	Complexity of Object Recognition	462
15.3	Object Representation	465
15.3.1	Observer-Centered Representations	466
15.3.2	Object-Centered Representations	467
15.4	Feature Detection	472
15.5	Recognition Strategies	473
15.5.1	Classification	474
15.5.2	Matching	479
15.5.3	Feature Indexing	481

15.6	Verification	481
15.6.1	Template Matching	482
15.6.2	Morphological Approach	483
15.6.3	Symbolic	483
15.6.4	Analogical Methods	486
A	Mathematical Concepts	492
A.1	Analytic Geometry	492
A.2	Linear Algebra	494
A.3	Variational Calculus	498
A.4	Numerical Methods	500
B	Statistical Methods	502
B.1	Measurement Errors	502
B.2	Error Distributions	504
B.3	Linear Regression	506
B.4	Nonlinear Regression	510
C	Programming Techniques	511
C.1	Image Descriptors	511
C.2	Mapping Operators	516
C.3	Image File Formats	517
	Bibliography	519
	Index	542

Chapter 5

Edge Detection

The early stages of vision processing identify features in images that are relevant to estimating the structure and properties of objects in a scene. Edges are one such feature. Edges are significant local changes in the image and are important features for analyzing images. Edges typically occur on the boundary between two different regions in an image. Edge detection is frequently the first step in recovering information from images. Due to its importance, edge detection continues to be an active research area. This chapter covers only the detection and localization of edges. Basic concepts in edge detection will be discussed. Several common edge detectors will be used to illustrate the basic issues in edge detection. Algorithms for combining edges into contours are discussed in Chapter 6.

An edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either (1) *step* discontinuities, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or (2) *line* discontinuities, where the image intensity abruptly changes value but then returns to the starting value within some short distance. However, step and line edges are rare in real images. Because of low-frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become *ramp* edges and line edges become *roof* edges, where intensity changes are not instantaneous but occur over a finite distance. Illustrations of these edge profiles are shown in Figure 5.1.

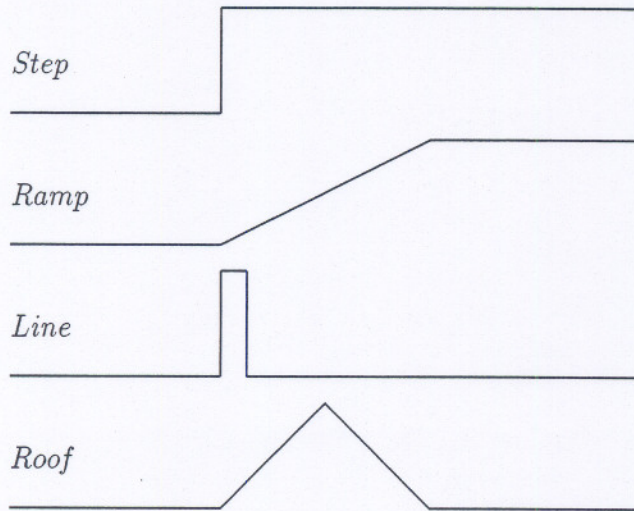


Figure 5.1: One-dimensional edge profiles.

It is also possible for an edge to have both step and line characteristics. For example, a surface that changes orientation from one flat surface to another will produce a step edge; but if the surface has a specular component of reflectance and if the surface corner is rounded, there can be a highlight due to the specular component as the surface orientation of the rounded corner passes the precise angle for specular reflection. The edge profile generated by such a situation looks like a step edge with a superimposed line edge. There are also edges associated with changes in the first derivative of the image intensity. For example, mutual reflection from the sides of a concave corner generate roof edges. Edges are important image features since they may correspond to significant features of objects in the scene. For example, the boundary of an object usually produces step edges because the image intensity of the object is different from the image intensity of the background.

This chapter will deal almost exclusively with step edges, although many of the ideas can be adapted to other types of image intensity changes. The profile of an ideal step edge and the profile of a real image that provides some examples of step edges are displayed in Figure 5.2. By definition, an edge is a significant local change in the image intensity. The plot shows step

previous chapter. The threshold values used for detection are given in the captions.

Figure 5.4 shows results of all the edge detection methods discussed so far, from the simple 1×2 gradient approximation up to the Prewitt operator. Figure 5.5 shows the results of the edge detectors when the filtering step is omitted. The next set of images (Figure 5.6) shows the results of edge detection on the same image now with additive Gaussian noise, $\sigma = 12$. The filter used was the same Gaussian filter as used in the previous figure. The final series of images (Figure 5.7) shows the results of edge detection on the same noisy image. However, for these images the filtering step was again omitted. Note the many false edges detected as a result of the noise.

5.3 Second Derivative Operators

The edge detectors discussed earlier computed the first derivative and, if it was above a threshold, the presence of an edge point was assumed. This results in detection of too many edge points. (Notice the thick lines after thresholding in Figures 5.4 to 5.7.) A better approach would be to find only the points that have local maxima in gradient values and consider them edge points, as shown in Figure 5.8. This means that at edge points, there will be a peak in the first derivative and, equivalently, there will be a zero crossing in the second derivative. Thus, edge points may be detected by finding the zero crossings of the second derivative of the image intensity.

There are two operators in two dimensions that correspond to the second derivative: the Laplacian and second directional derivative.

5.3.1 Laplacian Operator

The second derivative of a smoothed step edge is a function that crosses zero at the location of the edge (see Figure 5.8). The Laplacian is the two-dimensional equivalent of the second derivative. The formula for the Laplacian of a function $f(x, y)$ is

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \quad (5.18)$$

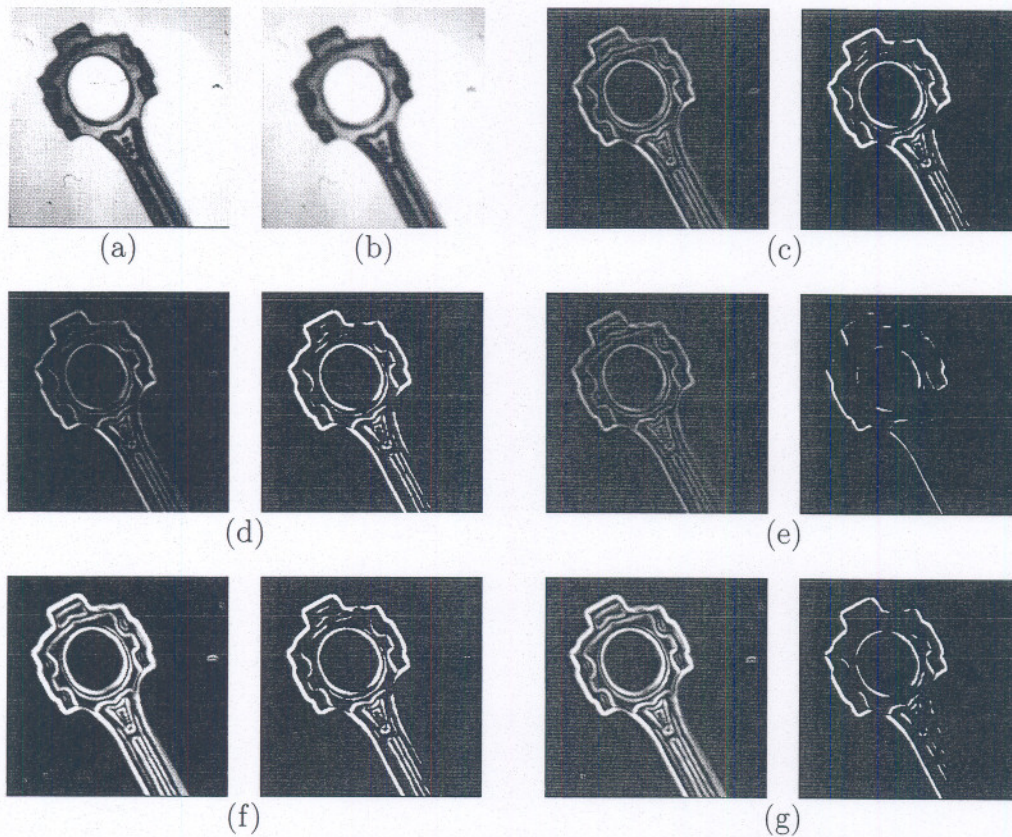


Figure 5.4: A comparison of various edge detectors. (a) Original image. (b) Filtered image. (c) Simple gradient using 1×2 and 2×1 masks, $T = 32$. (d) Gradient using 2×2 masks, $T = 64$. (e) Roberts cross operator, $T = 64$. (f) Sobel operator, $T = 225$. (g) Prewitt operator, $T = 225$.

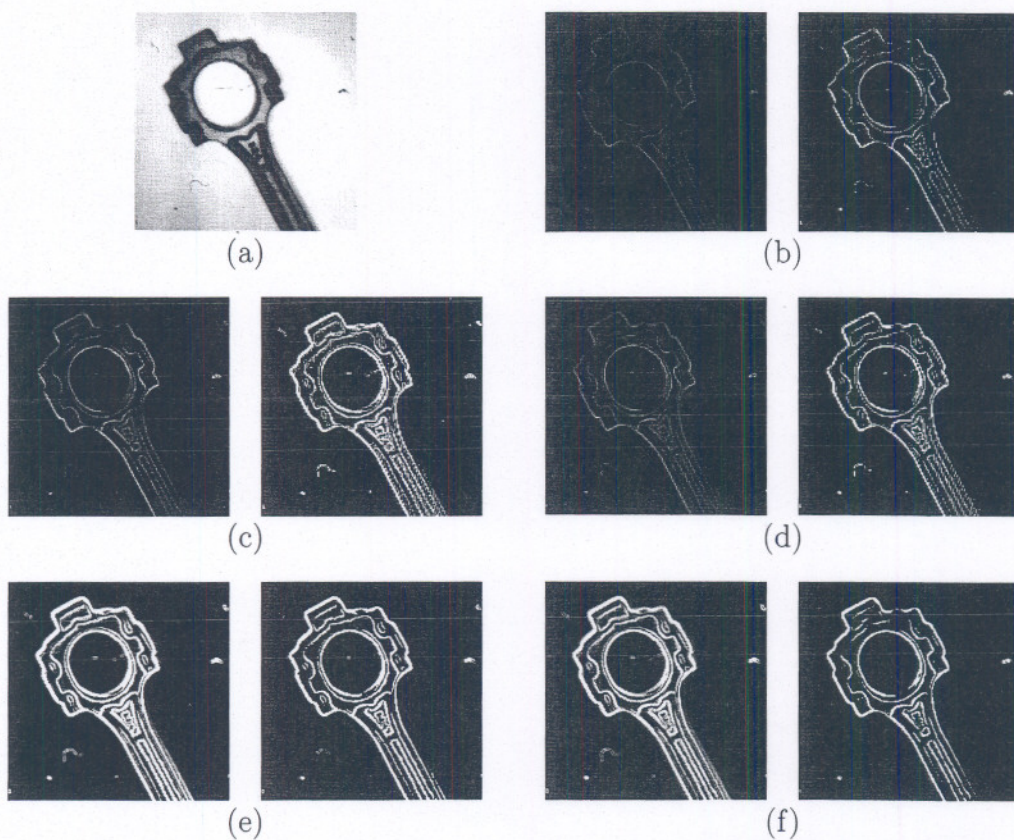


Figure 5.5: A comparison of various edge detectors without filtering. (a) Original image. (b) Simple gradient using 1×2 and 2×1 masks, $T = 64$. (c) Gradient using 2×2 masks, $T = 64$. (d) Roberts cross operator, $T = 64$. (e) Sobel operator, $T = 225$. (f) Prewitt operator, $T = 225$.

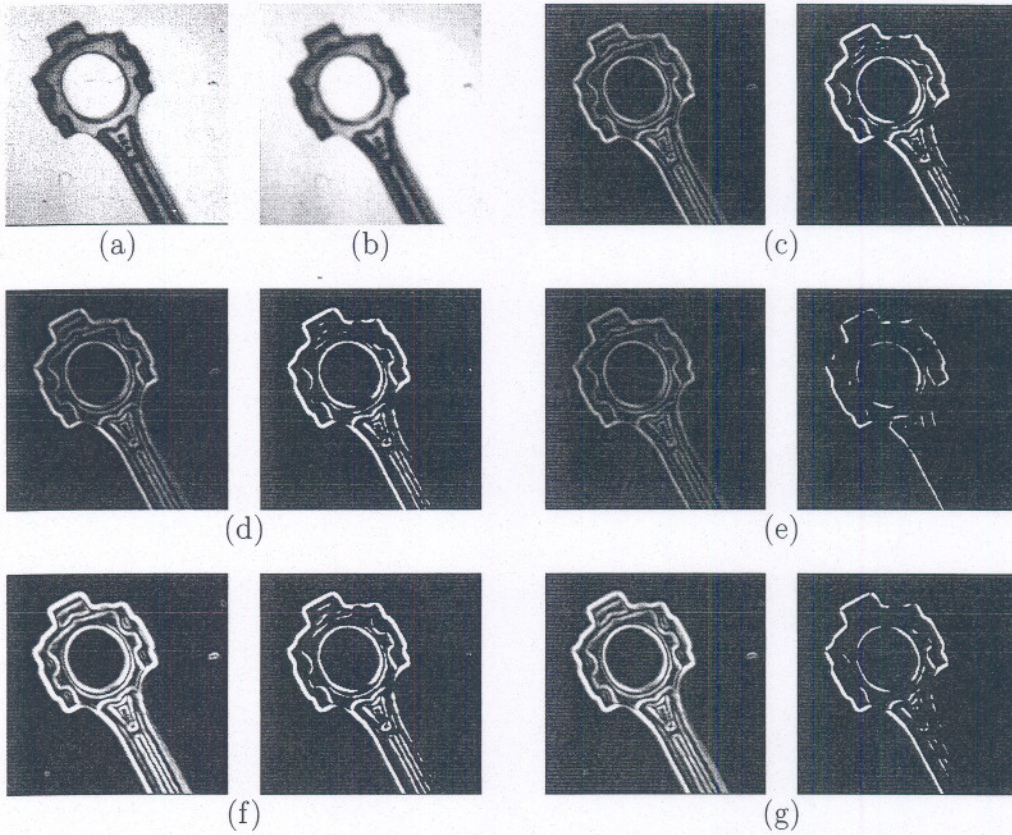


Figure 5.6: A comparison of various edge detectors on a noisy image. (a) Noisy image. (b) Filtered image. (c) Simple gradient using 1×2 and 2×1 masks, $T = 32$. (d) Gradient using 2×2 masks, $T = 64$. (e) Roberts cross operator, $T = 64$. (f) Sobel operator, $T = 225$. (g) Prewitt operator, $T = 225$.

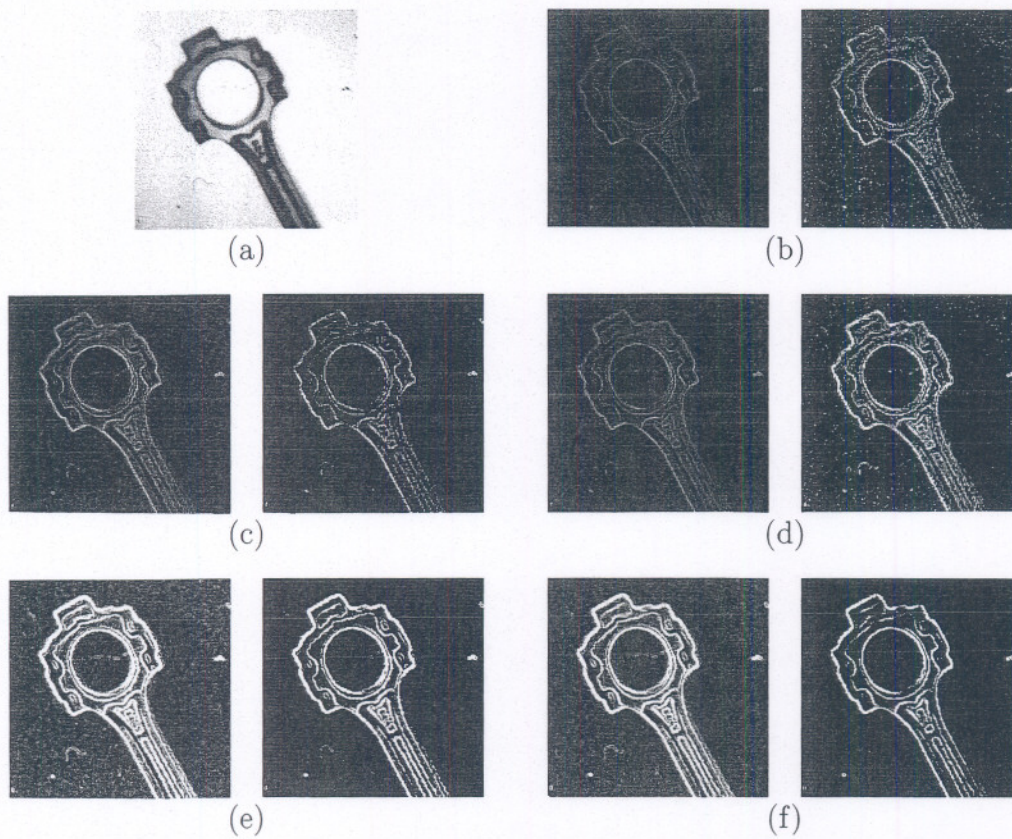


Figure 5.7: A comparison of various edge detectors on a noisy image without filtering. (a) Noisy image. (b) Simple gradient using 1×2 and 2×1 masks, $T = 64$. (c) Gradient using 2×2 masks, $T = 128$. (d) Roberts cross operator, $T = 64$. (e) Sobel operator, $T = 225$. (f) Prewitt operator, $T = 225$.

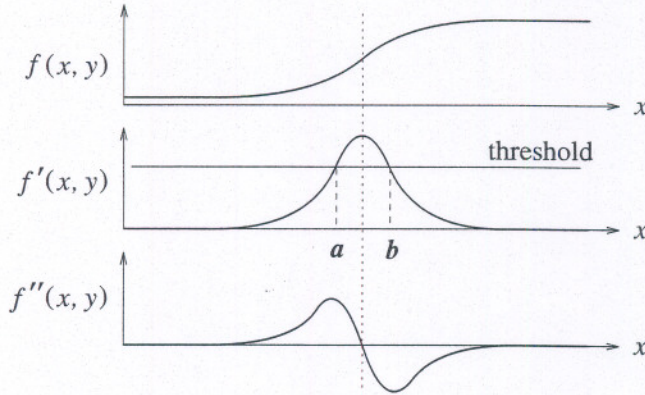


Figure 5.8: If a threshold is used for detection of edges, all points between a and b will be marked as edge pixels. However, by removing points that are *not* a local maximum in the first derivative, edges can be detected more accurately. This local maximum in the first derivative corresponds to a zero crossing in the second derivative.

The second derivatives along the x and y directions are approximated using difference equations:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial G_x}{\partial x} \quad (5.19)$$

$$= \frac{\partial (f[i, j+1] - f[i, j])}{\partial x} \quad (5.20)$$

$$= \frac{\partial f[i, j+1]}{\partial x} - \frac{\partial f[i, j]}{\partial x} \quad (5.21)$$

$$= (f[i, j+2] - f[i, j+1]) - (f[i, j+1] - f[i, j]) \quad (5.22)$$

$$= f[i, j+2] - 2f[i, j+1] + f[i, j]. \quad (5.23)$$

However, this approximation is centered about the pixel $[i, j+1]$. Therefore, by replacing j with $j-1$, we obtain

$$\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1], \quad (5.24)$$

which is the desired approximation to the second partial derivative centered about $[i, j]$. Similarly,

$$\frac{\partial^2 f}{\partial y^2} = f[i, j+1] - 2f[i, j] + f[i, j-1]. \quad (5.25)$$

By combining these two equations into a single operator, the following mask can be used to approximate the Laplacian:

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad (5.26)$$

Sometimes it is desired to give more weight to the center pixels in the neighborhood. An approximation to the Laplacian which does this is

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array} \quad (5.27)$$

The Laplacian operator signals the presence of an edge when the output of the operator makes a transition through zero. Trivial zeros (uniform zero regions) are ignored. In principle, the zero crossing location can be estimated to subpixel resolution using linear interpolation, but the result may be inaccurate due to noise.

Consider the example shown in Figure 5.9. This figure shows the result of the Laplacian on an image with a simple step edge. A single row of the resulting image is:

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 6 & -6 & 0 & 0 & 0 \\ \hline \end{array}$$

In this example, the zero crossing, corresponding to the edge in the original image, lies halfway between the two center pixels. The edge should be marked at either the pixel to the left or the pixel to the right of the edge, as long as it is marked consistently throughout the image. In most cases, however, the zero crossing rarely lies exactly between two pixels, and the actual edge

2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8

A sample image containing a vertical step edge.

0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0

Figure 5.9: The response of the Laplacian to a vertical step edge.

location must be determined by interpolating the pixel values on either side of the zero crossing.

Now consider the example in Figure 5.10. This figure shows the response of the Laplacian to a ramp edge. A single row of the output of the Laplacian is

0	0	0	3	0	-3	0	0
---	---	---	---	---	----	---	---

The zero crossing directly corresponds to a pixel in the image. Again, this is an ideal situation, and the actual edge location should be determined by interpolation.

5.3.2 Second Directional Derivative

The second directional derivative is the second derivative computed in the direction of the gradient. The operator is implemented using the formula

$$\frac{\partial^2}{\partial n^2} = \frac{f_x^2 f_{xx} + 2f_x f_y f_{xy} + f_y^2 f_{yy}}{f_x^2 + f_y^2} \quad (5.28)$$