Chapter 5 Edge Detection

Edge; significant local feature

Def. 5.1 Edge point; a point in an image at [i,j]
Def. 5.2 Edge fragment; [i,j] coord's and edge orientation Θ
Def. 5.3 Edege detector; an algorithm which produce a set of edges
Def. 5.4 Contour; list of edges or mathematical curve of the edges
Def. 5.5 Edge linking; forming process of an ordered set of edges , clockwise
Def. 5.6 Edge following; serching process to determine contours

edge set;
correct edges, false edges (flase positive), missing edges (false negative)

5.1 Gradient

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial x} \end{bmatrix}$$

magnitude; $G[f(x,y)] = \sqrt{G_x^2 + G_y^2}$

approximately; $G[f(x,y)] \simeq |G_x| + |G_y|$
$\simeq \max(|G_x|, |G_y|)$

direction; $\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$

isotropic operator; magnitude of the gradient is independent of the edge direction

Numerical approximation;

$$G_x \simeq f[i, j+1] - f[i, j]$$
$$G_y \simeq f[i, j] - f[i+1, j]$$

Gx=

-1	1
-1	1

Gy=

1	1
-1	-1

5.2 Steps in Edge detection

Filtering ; noise reducing

edge strength ?

Enhancement ;

Detection; Gradient > threshold

Localization; location of edges are estimated

5.2.1 Roberts operator

$$G[f[i,j]] = |f[i,j] - f[i+1, j+1]| + |f[i+1, j] - f[i, j+1]|$$

Gx=

1	0
0	-1

Gy=

0	-1
1	0

5.2.2 Sobel operator

$$M = \sqrt{s_x^2 + s_y^2}$$

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

c=2

sx =

-1	0	1
-2	0	2
-1	0	1

, sy =

-1	-2	1
0	0	0
1	2	1

emphasis on pixels closer to the center

5.2.3 Prewitt operator

c=1

sx =

-1	0	1
-1	0	1
-1	0	1

, sy =

-1	-1	1
0	0	0
1	1	1

no emphasis

5.3 Second Derivative Operators

5.3.1 Laplacian operator

Laplacian of a function f(x,y)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial G_x}{\partial x} \\ &\simeq \frac{\partial(f[i,j+1]-f[i,j])}{\partial x} \\ &= \frac{\partial f[i,j+1]}{\partial x} - \frac{\partial f[i,j]}{\partial x} \\ &= f[i,j+2]-2f[i,j+1]+f[i,j] \\ &\simeq f[i,j+1]-2f[i,j]+f[i,j-1]\end{aligned}$$

$$\frac{\partial^2 f}{\partial y^2} \simeq f[i+1,j]-2f[i,j]+f[i-1,j]$$

$\nabla^2 \approx$

0	1	0
1	-4	1
0	1	0

or

1	4	1
4	-20	4
1	4	1

* trivial zero (uniform) ignored

5.3.2 Second Directional Derivative

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

* Laplacian, 2nd directional derivatice ; noise sensitive

5.4 Laplacian of Gaussian

filter out the noise before edge enhancement

Laplacian of Gaussian (LoG); (Marr and Hildereth)

Gaussianl filtering with the Laplacian for edge detection

- smoothing filter is a Gaussian
- enhancement; 2nd derivative or Laplacian
- detection criterion; zero crossing in 2nd derivative and large peak in first derivative
- edge location; linear interpolation

$$h(x,y) = \nabla^2[(g(x,y) * f(x,y)]$$
$$= [\nabla^2 g(x,y)] * f(x,y)$$

where $\nabla^2 g(x,y) = (\frac{x^2+y^2-2\sigma^2}{\sigma^4})e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

operator;

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Method;
(1) Convolve the image with a Gaussian smoothing filter and compute the Laplacian of the result
(2) Convolve the image with the linear filter that is the Laplacian of the Gaussian filter

Scale Space

5.5 Image approximation

facet model; only in the local neighborhood of a pixel

$$z = f(x,y)$$

piecewise constant, piecewise bilinear, biquadratic, bicubic, etc.

bicubic polynomial;

$$f(x,y) = k_1 + k_2x + k_3y + k_4x^2 + k_5xy + k_6y^2 + k_7x^3 + k_8x^2y + k_9xy^2 + k_{10}y^3$$

The first derivative in the direction Θ

$$f_\Theta{}'(x,y) = \frac{\partial f}{\partial x}\cos\Theta + \frac{\partial f}{\partial y}\sin\Theta$$

$$f_\Theta{}''(x,y) = \frac{\partial^2 f}{\partial x^2}\cos^2\Theta + 2\frac{\partial^2 f}{\partial x \partial y}\cos\Theta\sin\Theta + \frac{\partial^2 f}{\partial y^2}\sin^2\Theta$$

$$\sin\Theta = \frac{k_2}{\sqrt{k_2^2 + k_3^2}}$$

$$\cos\Theta = \frac{k_3}{\sqrt{k_2^2 + k_3^2}}$$

At (x_0, y_0), and $x_0 = \rho\cos\Theta$, $y_0 = \rho\sin\Theta$

$$f_\theta''(x_0, y_0) = 6(k_7\sin^3\theta + k_8\sin^2\theta\cos\theta + k_9\sin\theta\cos^2\theta + k_{10}\cos^3\theta)\rho + 2(k_4\sin^2\theta + k_5\sin\theta\cos\theta + k_6\cos^2\theta)$$
$$= A\rho + B$$

if $\rho \le \rho_0$ exists s.t.

$$f_\Theta''(x_0, y_0; \rho) = 0$$
$$f_\Theta'(x_0, y_0; \rho) \neq 0$$

there is an edge point in the pixel

5.6 Gaussian Edge Detection

edges; locally large gradient magnitude

not really perfect sharp; low pass filtering, camera optics, bandwidth limit

noise

edge approximation; suppress the effect of noise

locate the edges as accurately as possible

First derivative of Gaussian ; best compromise

5.6.1 Cany Edge Detector

$$S[i,j] = G[i,j;\sigma] * I[i,j]$$

$$P[i,j] \sim (S[i,j+1] - S[i,j] + S[i+1,j+1] - S[i+1,j])/2$$
$$Q[i,j] \sim (S[i,j] - S[i+1,j] + S[i,j+1] - S[i+1,j+1])/2$$

Magnitude, Orientation

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2}$$
$$\Theta[i,j] = \arctan(Q[i,j], P[i,j])$$

floating point calculation; look table, integer and fixed point

Nonmaxima Suppression;

smooth ridges -> thinned edges

along the gradient line, not peak point -> 0, leave only peak gradient point

$$\zeta[i,j] = Sector[\Theta[i,j])$$

$$N[i,j] = \mathrm{nms}(M[i,j], \zeta[i,j])$$

flase edge fragments due to noise and fine texture

Thresholding;

Single threshold; selecting a proper one is difficult

Double threshold;

apply twoo thresholds $\tau_2 \sim 2\tau_1$ to the N[i,j]

$T_2[i,j]$; using high threshold -> fewer false edge or missing edge

$T_1[i,j]$; using low threshold -> many edge

From T2[i,j] follow contour, if missing, look at T1[i,j] at the 8 neighbor, link

5.7 Subpixel Location Estimation

- to noisy to allow any simple interpolation
- obtaining subpixel resolution after gradient based scheme

* procedure

i) take samples of the magnitude of the Gaussian edge detector output along the gradient direction

ii) first moment(weighted average)

$$\delta d = \frac{\sum_{i=1}^{n} g_i d_i}{\sum_{i=1}^{n} g_i}$$

d_i; distance of a pixel along the gradient

g_i; gradient magnitude