

Timage Enhancement

April 2, 2008

unsharp masking + high boost filtering

- unsharp masking

$$f_s(x,y) = f(x,y) - \underbrace{f}_{\text{original}} \xrightarrow{\text{processed}}$$

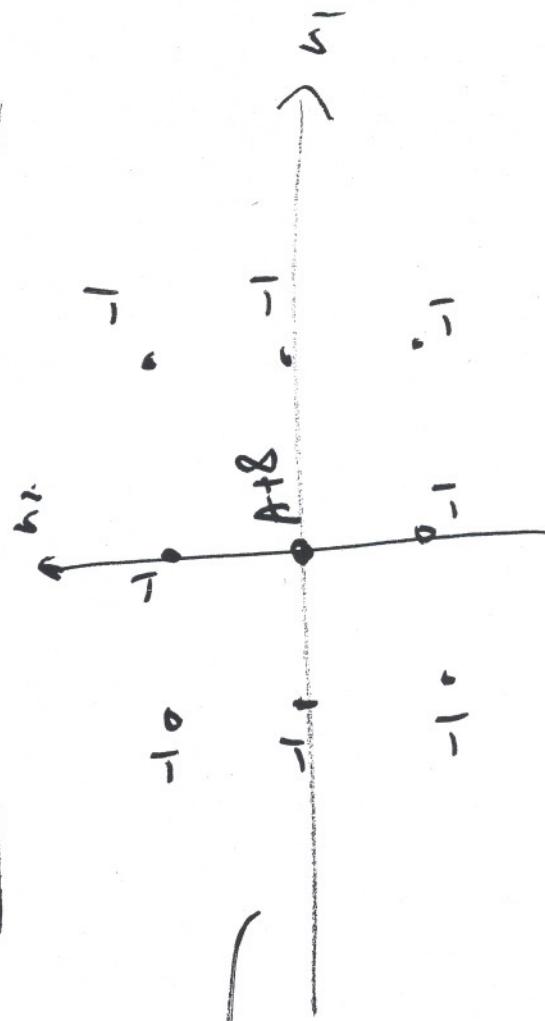
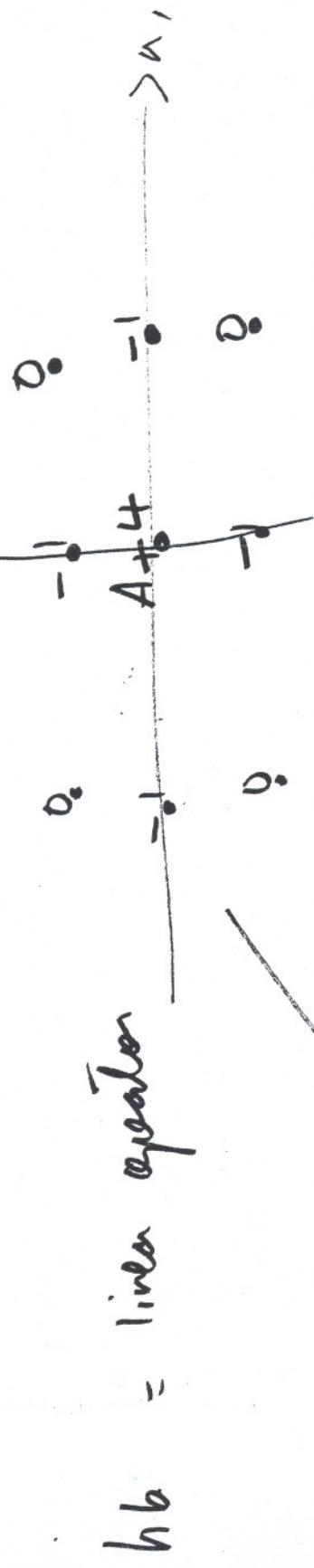
$\hat{f}(x,y)$
 $\hat{f} = \text{blurred}$

high boost filtering

$$f_{hb}(x,y) = A f(x,y) - \hat{f}(x,y)$$

$A > 1$
boost factor

$$f_{hb}(x,y) = (A_{-1}) f(x,y) + f_s(x,y).$$



Possible response
impulse response
in training.
HB

Homomorphic Processing

$$f(x,y) = i(x,y) \cdot r(x,y)$$

illumination reflection

$$\text{Define } z(x,y) = l_n(f(x,y) + i(x,y))$$

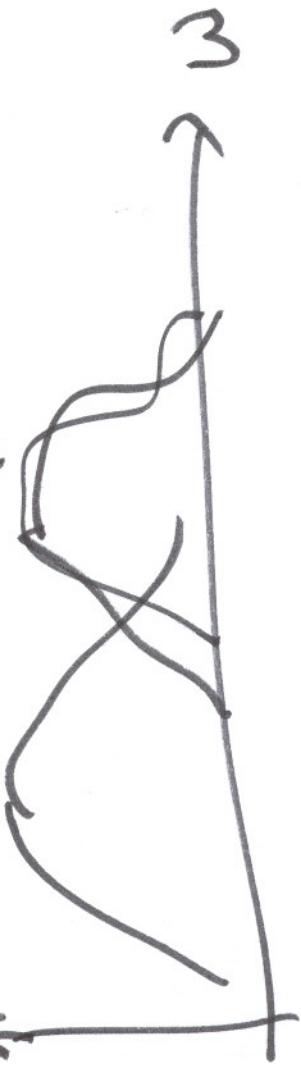
$z(x,y) = l_n(i(x,y)) + l_n(r(x,y))$

the last

exploit the fact

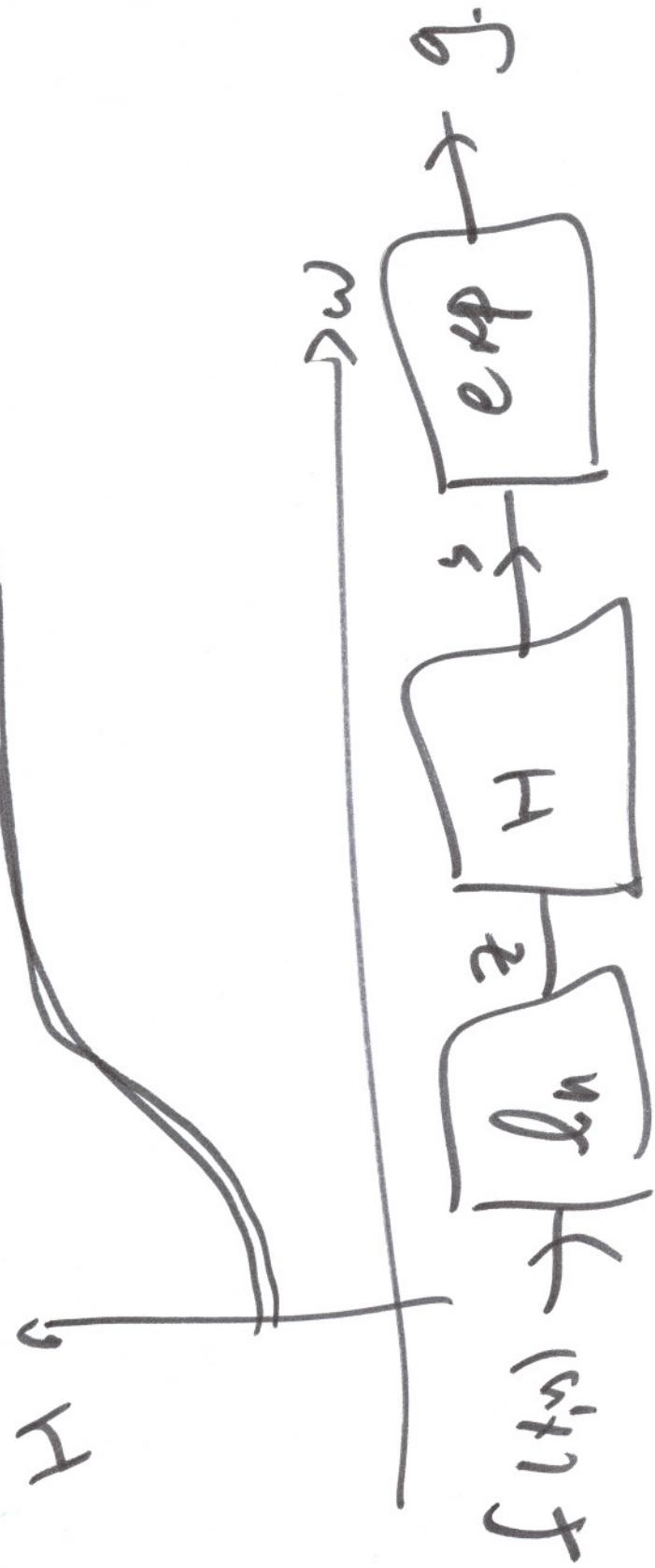
occurring different bands

of the spectra.



Apply lines 6.1/6.2.

$$Z(t, u) \rightarrow L^{\text{ST}} \rightarrow S(t, u)$$



Using backprop for Euclidean

First derivative:

$$\vec{\nabla}f = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

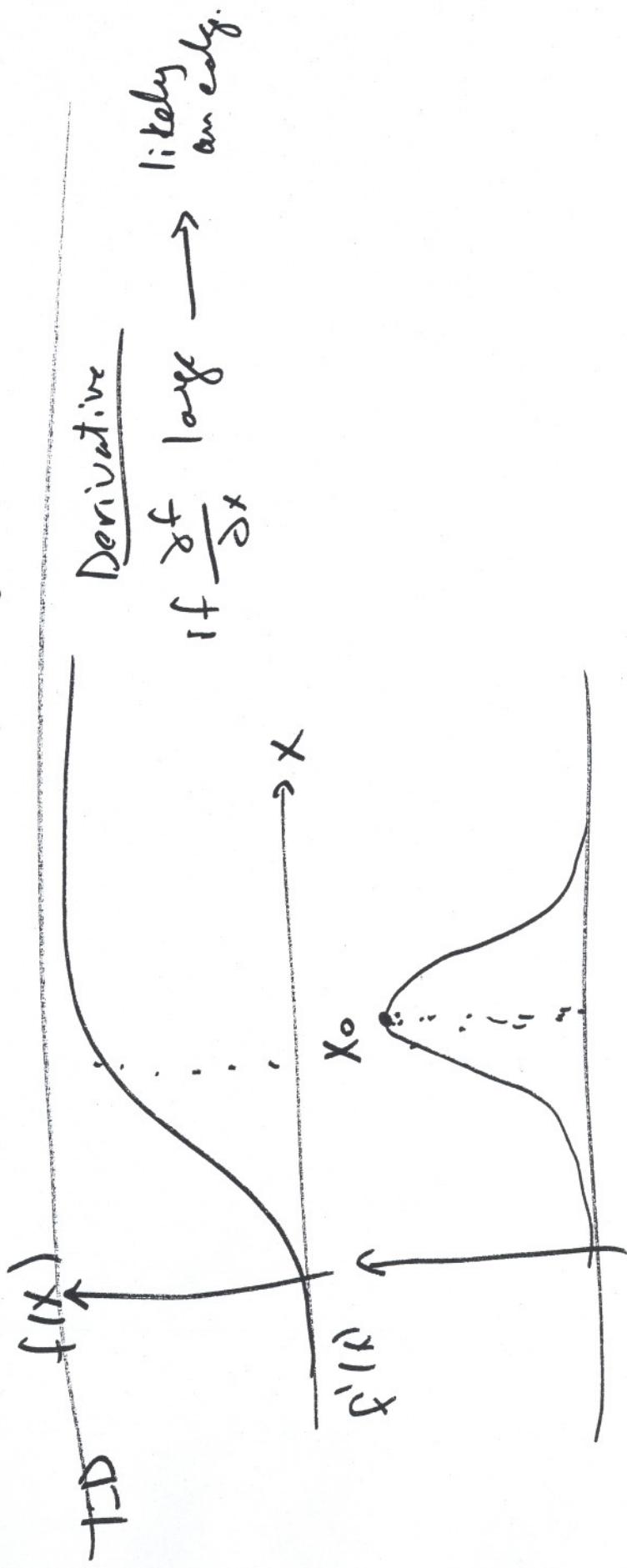
$$|\vec{\nabla}f| = \sqrt{G_x^2 + G_y^2} \rightarrow \text{Edge Detection.}$$

To simplify computation

$$|\vec{\nabla}f| \approx |G_x| + |G_y|$$

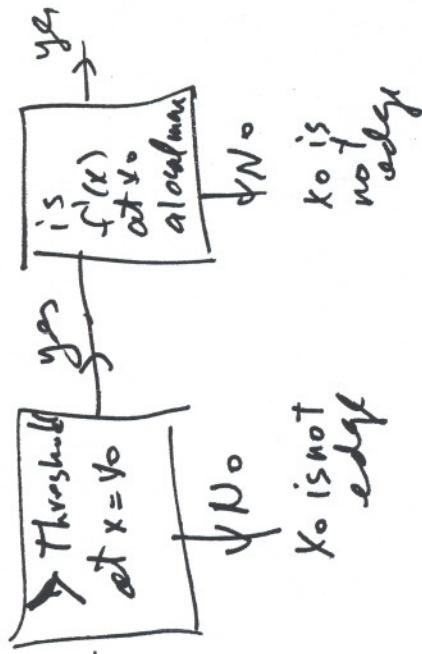
Edge Detection

- Gradient Method.
- replacement $\nabla^2 f$
- Lot G = replace of Gauss: $H; \text{Gauss/Marr}$.



$$f(x) \rightarrow \left[\frac{\partial f}{\partial x} \quad f'(x) \right]$$

1D Edge Detect:

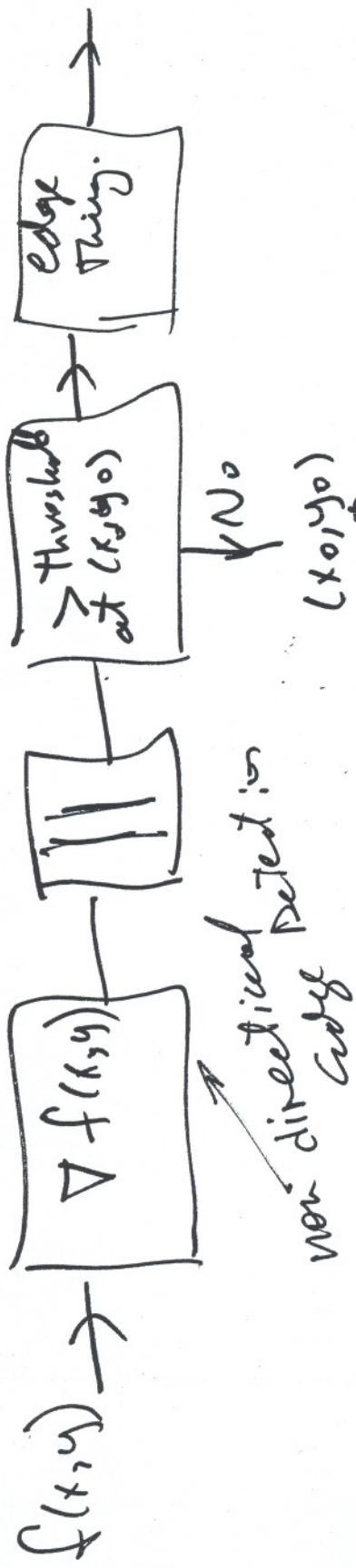


Show 10.7 of G/W

Extension To 2D

$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x, y) \\ \frac{\partial f}{\partial y}(x, y) \end{bmatrix}$

$f(x, y)$



edge thing:
 (a) If $|\nabla f(x,y)|$ has a local max at
 (x_0, y_0) in horizontal direction, but not
 vertical, we declare (x_0, y_0) an edge
 when
 $\frac{\delta f(x,y)}{\delta x} > \frac{\delta f(x,y)}{\delta y}$

$$K \approx 2$$

(b) If $|Vf(x,y)|$ has a local max at (x_0, y_0) in vertical direction but not horizontal, place (x_0, y_0) on edge. when

$$\left| \begin{array}{c} \frac{\partial f}{\partial y}(x,y) \\ \frac{\partial f}{\partial x}(x,y) \end{array} \right| \rightarrow K \quad \left| \begin{array}{c} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{array} \right| (x_0, y_0)$$

Directional Gradient Edge Detectors

Towards a particular direction.

- Bias

Complete vertical edge:

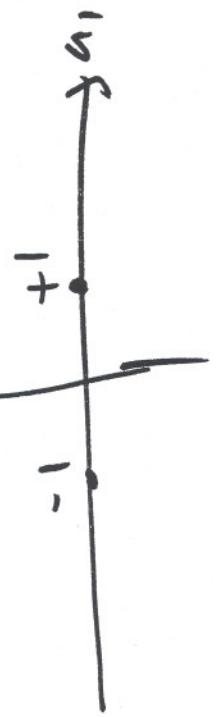
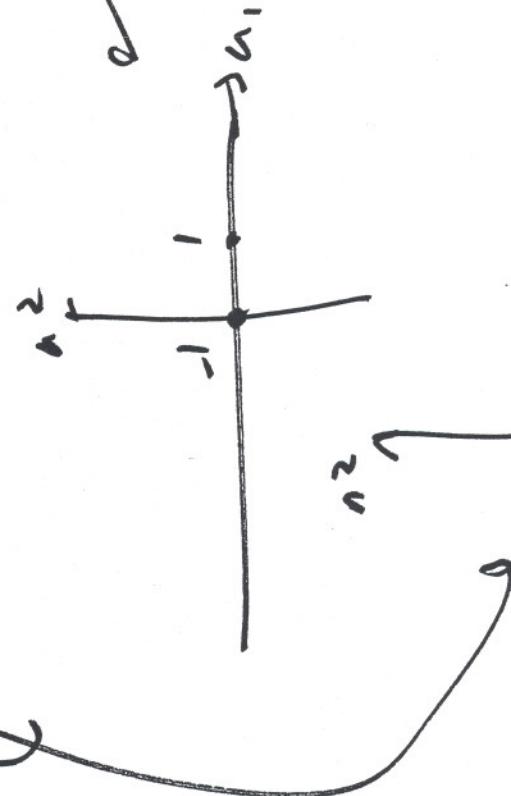
$$\frac{\partial f(x,y)}{\partial x}$$

Horizontal edge

$$\frac{\partial f(x,y)}{\partial y}$$

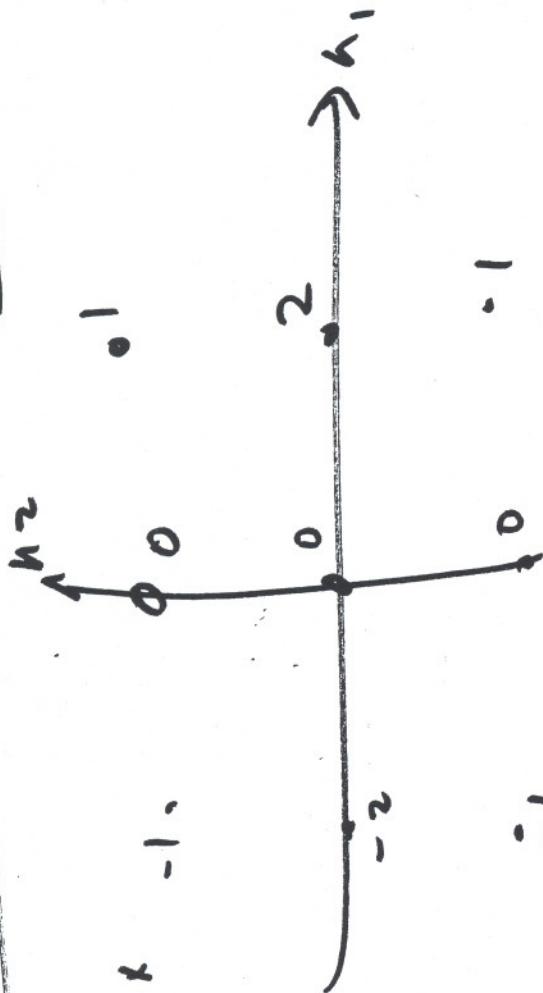
$$\frac{\partial f(x,y)}{\partial x} = \sum \frac{f(x+1,y) - f(x-1,y)}{2T}$$

$$f(x+1,y) - f(x-1,y)$$

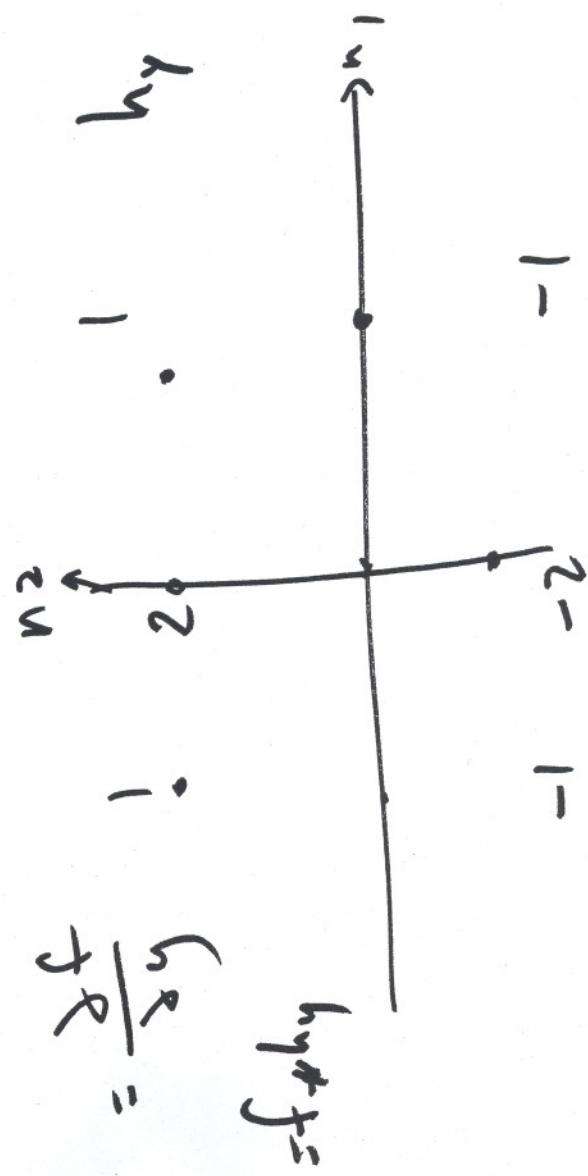


Favours ∇ instead of Edge Detection

Sobel



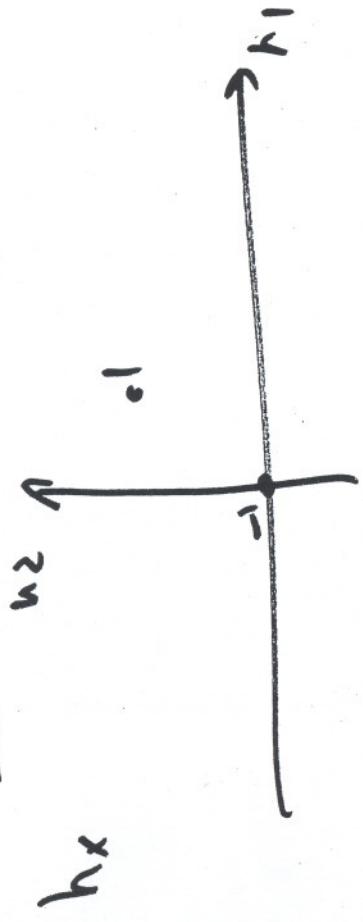
$$\frac{\partial f}{\partial x}$$



$$= \frac{\partial f}{\partial y}$$

$$= f * h_y$$

Dobert's even Detektor



u_2, h_2

h_x



h_1

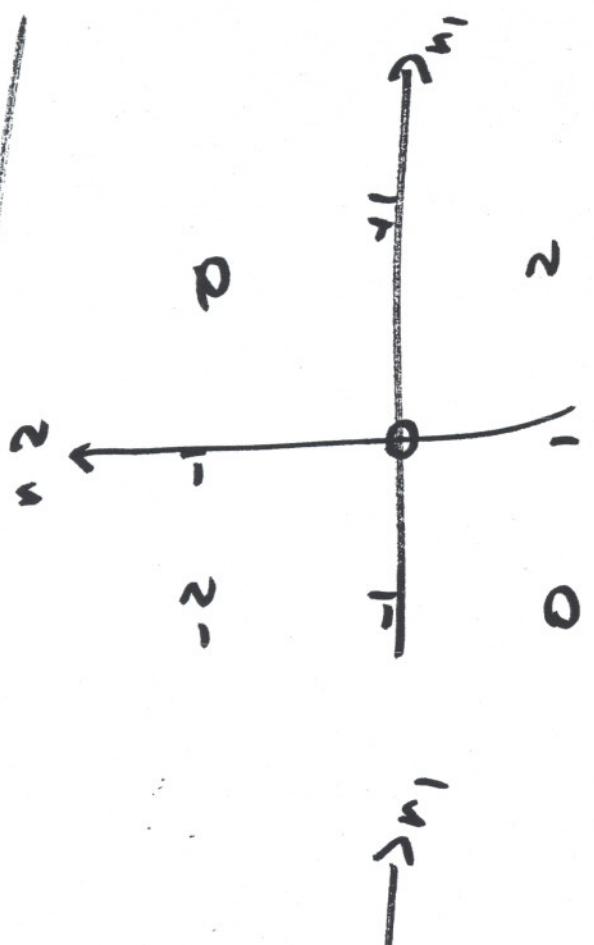
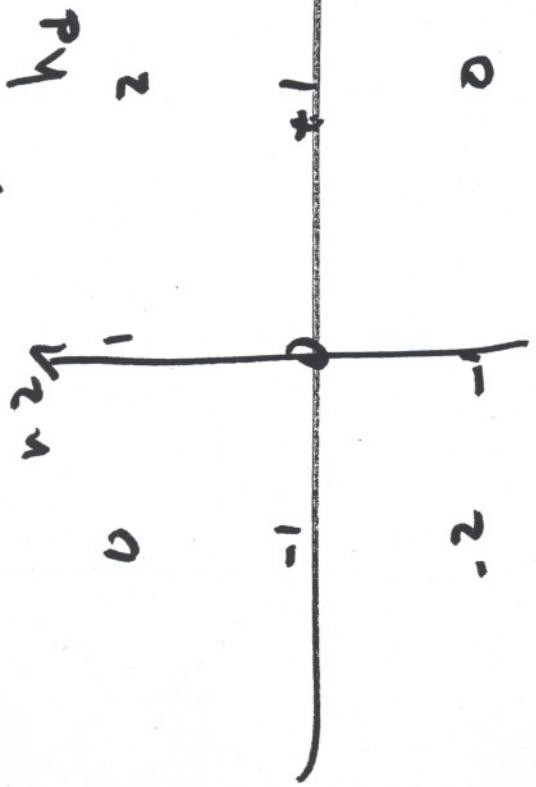
(1)

(-1)

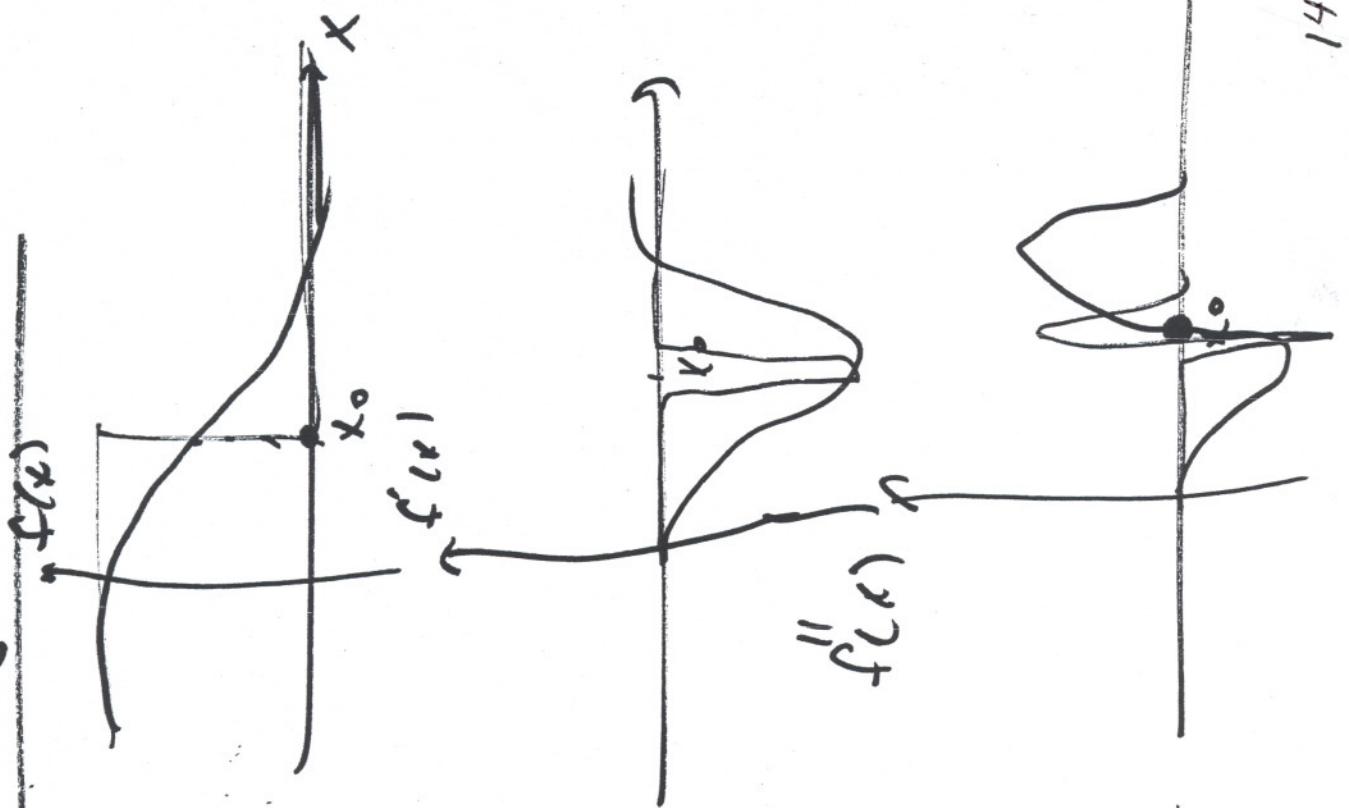
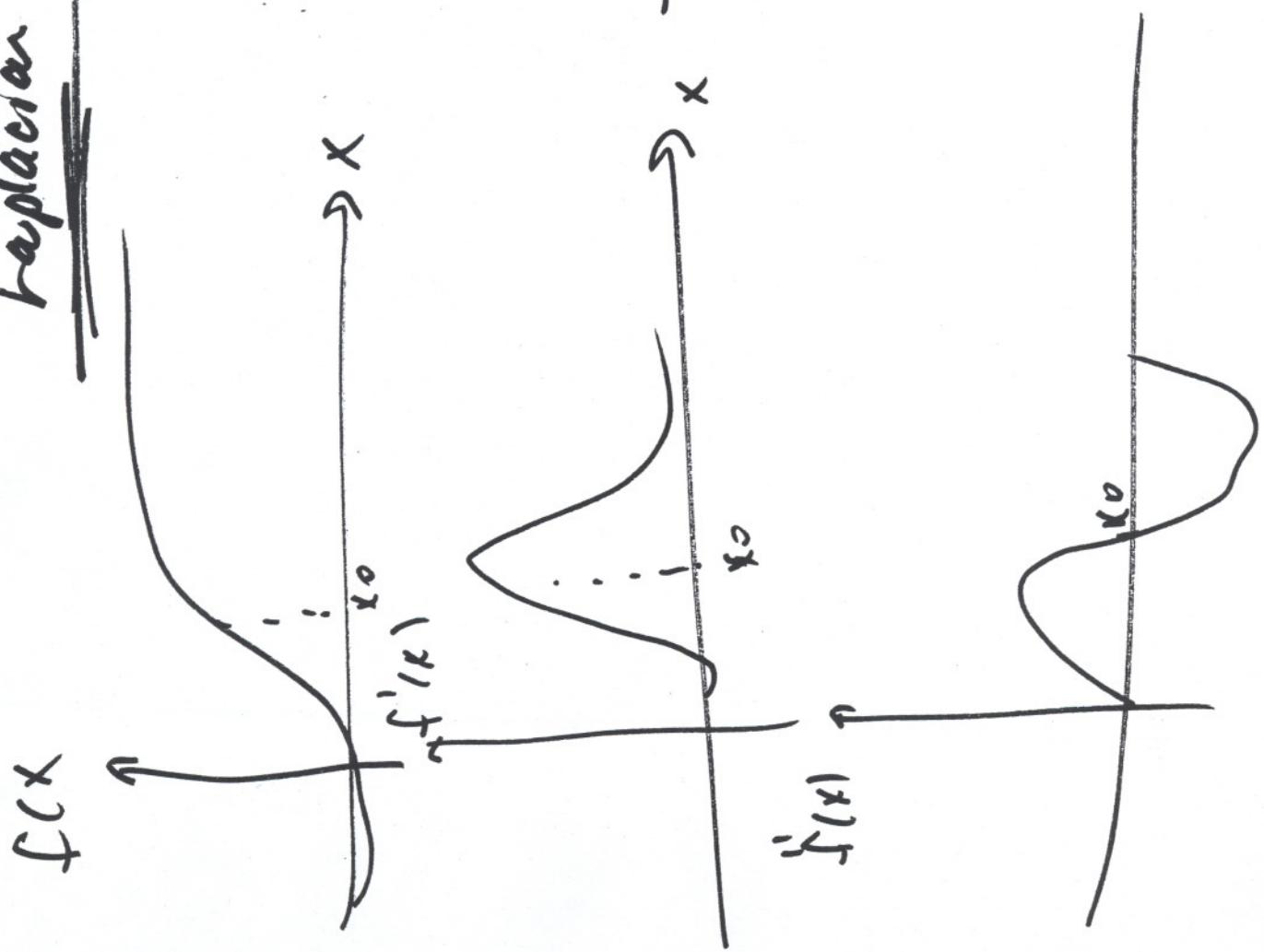
12

Diagonal Directional Gradient f. Iter

Global \rightarrow Diagnal.



hipótesis sobre derivadas



$$\begin{aligned}
 \text{Extension } &= 20 \\
 \Delta^2 f(x, y) &= \Delta \left(\Delta f(x, y) \right) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \\
 &= \frac{\partial^2 f(x, y)}{\partial x^2} - f_x(n_1, n_2) + f_x(n_{i-1}, n_2) \\
 &\quad + f_x(n_i + 1, n_2) - f_x(n_i, n_2) \\
 &\quad + \frac{\partial^2 f(x, y)}{\partial y^2} - f_y(n_1, n_2) + f_y(n_{i-1}, n_2) \\
 &\quad + f_y(n_i + 1, n_2) - f_y(n_i, n_2) \\
 &\quad + \frac{\partial^2 f(x, y)}{\partial x \partial y} - f_{xy}(n_1, n_2) + f_{xy}(n_{i-1}, n_2) \\
 &\quad + f_{xy}(n_i + 1, n_2) - f_{xy}(n_i, n_2) \\
 &\quad + f_{xy}(n_i, n_{i+1}) - f_{xy}(n_{i-1}, n_{i+1})
 \end{aligned}$$

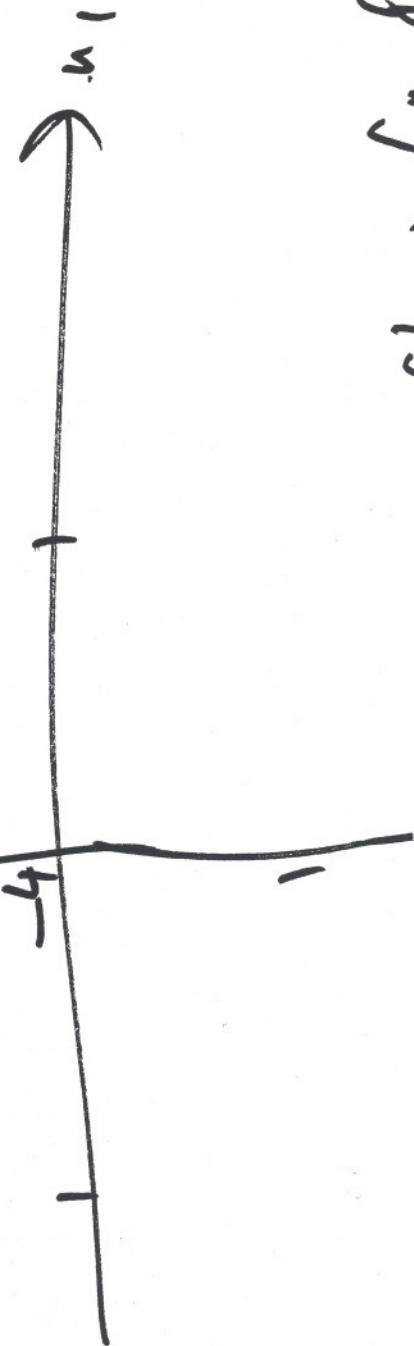
For linear and -
 $\frac{d}{dt} \left(x(t) \right)$ edge.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \rightarrow f(u_1+1, u_2) + f(u_1-1, u_2) + f(u_1, u_2+1) \\ + f(u_1, u_2-1) - 4f(u_1, u_2)$$

$$= f$$

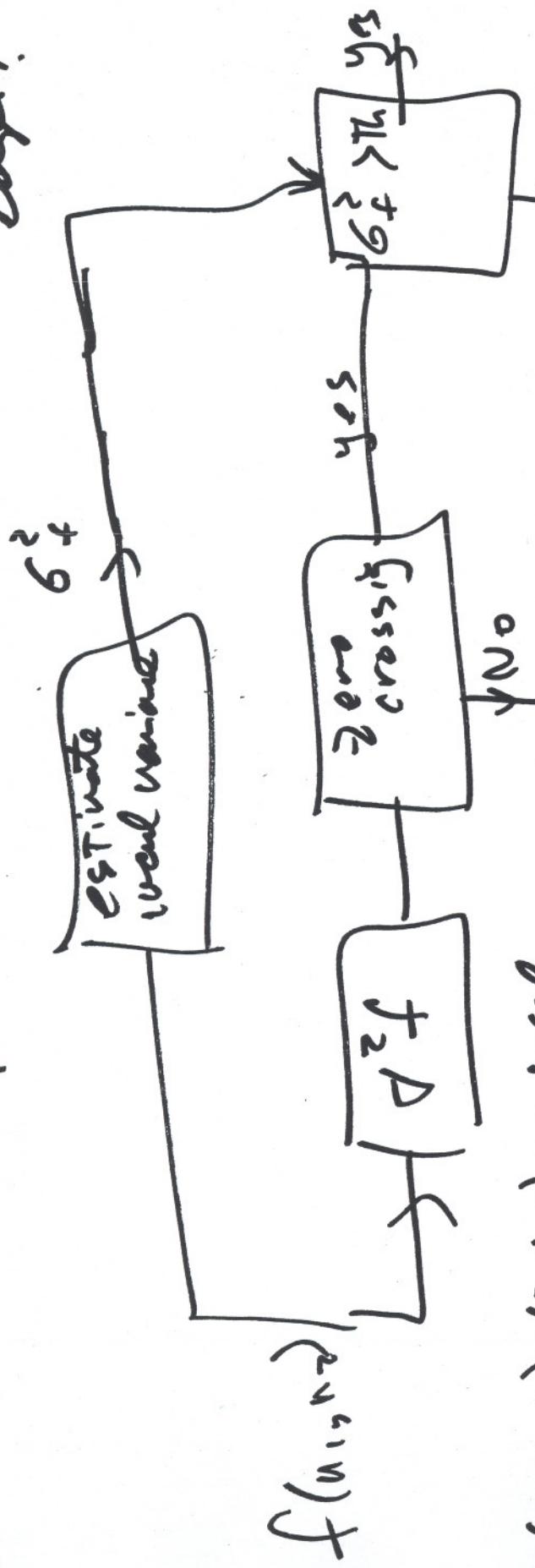
u_1

u_2



Show f_j $\delta_{j,j}$
J Lin

How To use ∇^2_f operator to detect edges?



$$(2N+1)\chi(2^{M+1}) \rightarrow 1.0 \text{ cal.}$$

三

edge
not

not an
edge

Licet magis

local mean

$$m_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=-M}^{M} \sum_{k_2=-M}^{M} f(k_1, k_2)$$

local variance.

$$\sigma_f^2(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=-M}^{M} \sum_{k_2=-M}^{M} \left(f(k_1, k_2) - m_f(n_1, n_2) \right)^2$$

Window $(2M+1) \times (2M+1)$
centered about (n_1, n_2)

Edge Detection Laplacian of Gaussian

$$\text{Gaussian} \quad L_a(x, y) = e^{-\frac{(x^2+y^2)}{2\pi\sigma^2}}$$

$$H(\sigma_{xx}, \sigma_{yy}) = 2\pi^2 \sigma^2 e$$

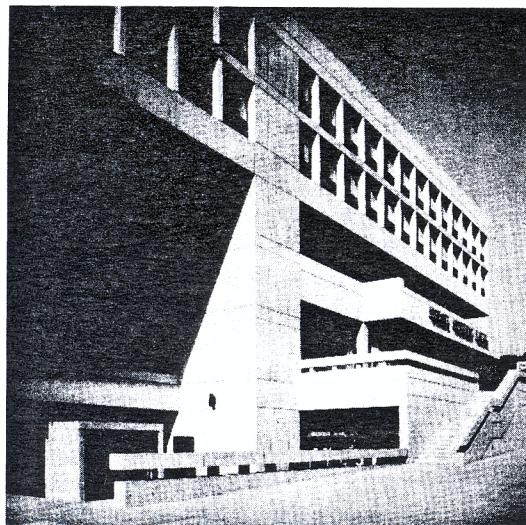
$$\text{"Laplacian of Gaussian"} = L \circ G.$$

$$\Delta^2 \left[f(x, y) * h(x, y) \right] =$$

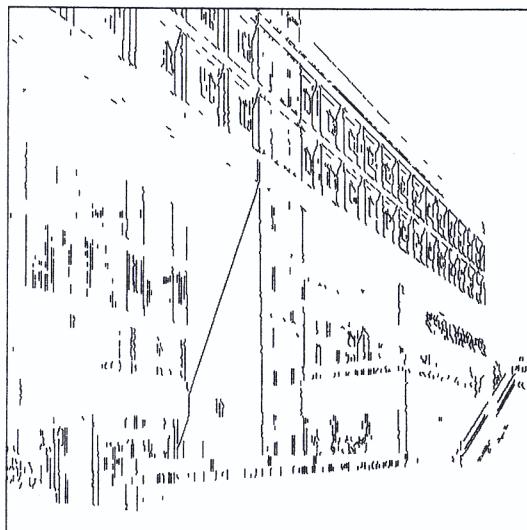
$$= f(x, y) * \left[\sigma^2 h(x, y) + \frac{\partial^2 h(x, y)}{\partial x^2} + \frac{\partial^2 h(x, y)}{\partial y^2} \right]$$

$$\begin{aligned}
 H^2 h(x, y) &= \\
 &\frac{e^{-(x^2+y^2)/(2\pi\sigma^2)}}{(2\pi\sigma^2)^2} \left(x^2 + y^2 - 2\pi\sigma^2 \right) \\
 F \left\{ \begin{array}{c} \uparrow \\ \end{array} \right\} &= -2\pi\sigma^2 \theta \\
 &- \pi\sigma^2 (R_x^2 + R_y^2)/2 \\
 &+ (R_x^2 + R_y^2)/2
 \end{aligned}$$

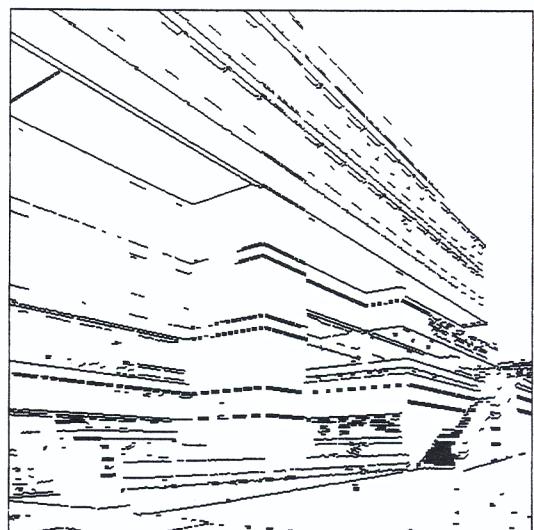
Fig. 8.36 T.Lim



(a)

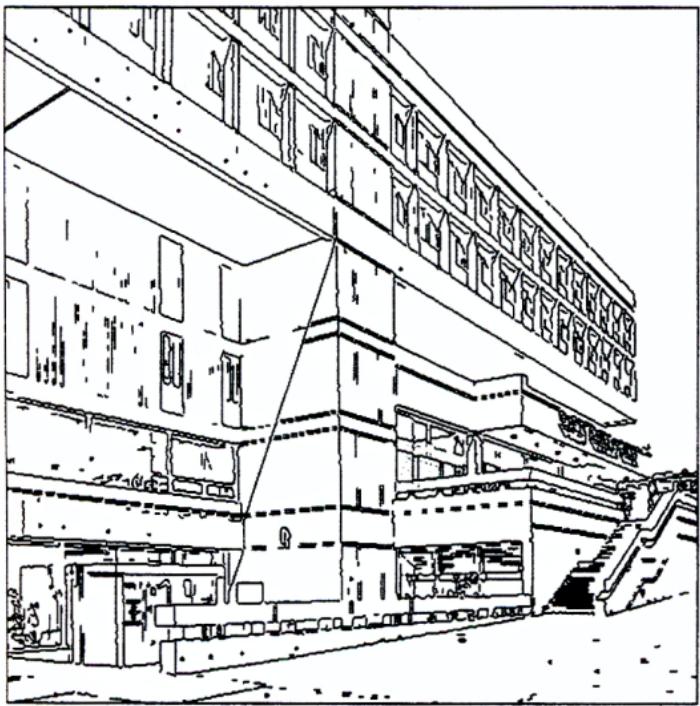


(b)

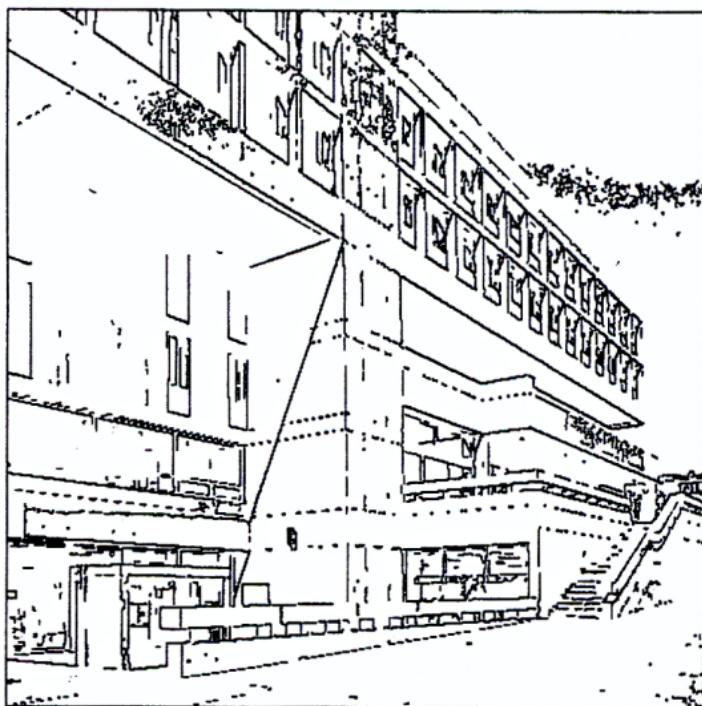


(c)

Figure 8.30 Edge maps obtained by directional edge detectors. (a) Image of 512×512 pixels; (b) result of applying a vertical edge detector; (c) result of applying a horizontal edge detector.



(a)

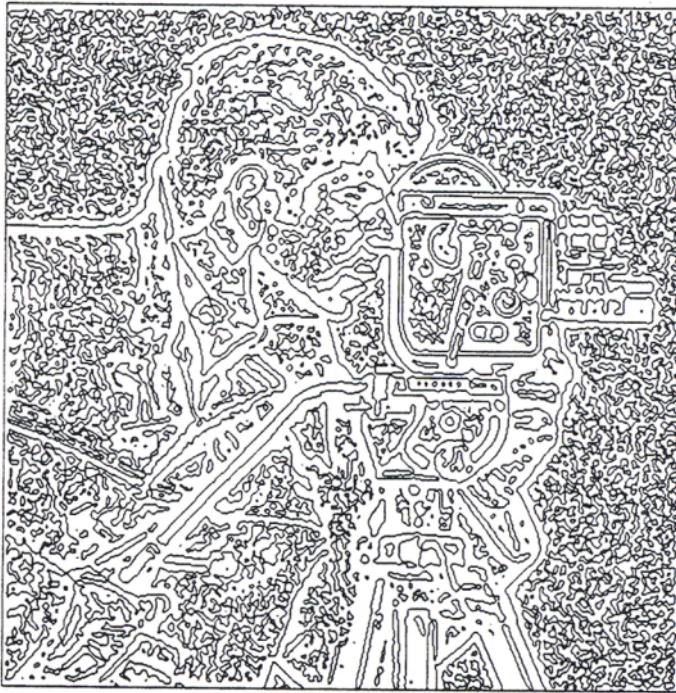


(b)

Figure 8.31 Result of applying (a) Sobel edge detector and (b) Roberts's edge detector to the image in Figure 8.30(a).



(a)



(b)

Figure 8.33 Edge map obtained by a Laplacian-based edge detector. (a) Image of 512×512 pixels; (b) result of convolving the image in (a) with $h(n_1, n_2)$ in Figure 8.32(a) and then finding zero-crossing points.

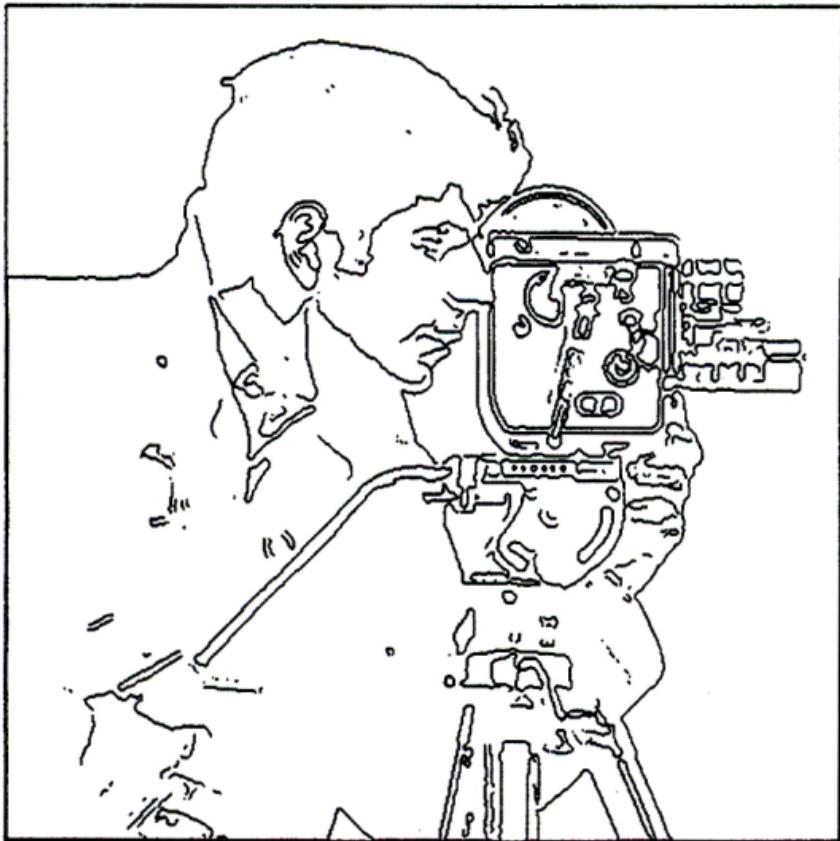


Figure 8.35 Edge map obtained by applying the system in Figure 8.34 to the image in Figure 8.33(a).



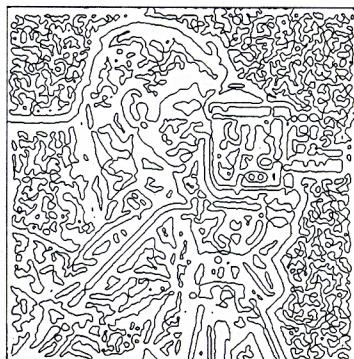
(a)



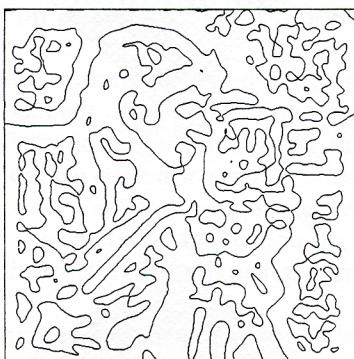
(b)



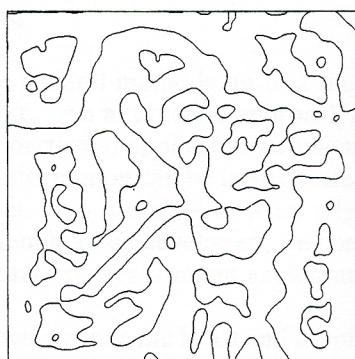
(c)



(d)



(e)



(f)

Figure 8.37 Edge maps obtained from lowpass filtered image. Blurred image with (a) $\sigma^2 = 4$; (b) $\sigma^2 = 16$; (c) $\sigma^2 = 36$. Result of applying Laplacian-based algorithm to the blurred image; (d) $\sigma^2 = 4$; (e) $\sigma^2 = 16$; (f) $\sigma^2 = 36$.