

Frei and Chen's orthogonal masks

Basis of edge subspace

$$w_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad w_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$w_3 = \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \quad w_4 = \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

Basis of line subspace

$$w_5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad w_6 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$w_7 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad w_8 = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

"Average" subspace

$$w_9 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Magnitudes of projections of x onto the edge, line, and average

$$p_e = \left[\sum_{i=1}^4 (\mathbf{w}_i^T \mathbf{x})^2 \right]^{\frac{1}{2}} \quad (\text{A})$$

$$p_l = \left[\sum_{i=5}^8 (\mathbf{w}_i^T \mathbf{x})^2 \right]^{\frac{1}{2}} \quad (\text{B})$$

$$p_a = |\mathbf{w}_9^T \mathbf{x}|$$

Angles between x and its projections onto edge, line and average subspaces

$$\theta_e = \cos^{-1} \left\{ \frac{1}{\|\mathbf{x}\|} \left[\sum_{i=1}^4 (\mathbf{w}_i^T \mathbf{x})^2 \right]^{\frac{1}{2}} \right\} \quad (C)$$

$$\theta_l = \cos^{-1} \left\{ \frac{1}{\|\mathbf{x}\|} \left[\sum_{i=51}^{84} (\mathbf{w}_i^T \mathbf{x})^2 \right]^{\frac{1}{2}} \right\} \quad (D)$$

$$\theta_a = \cos^{-1} \left\{ \frac{|\mathbf{w}_9^T \mathbf{x}|}{\|\mathbf{x}\|} \right\} \quad (E)$$

The first four masks are suitable for detecting edges; the second four masks represent templates suitable for detecting lines; and the last four mask is proportional to the average of the pixels in the region at which the mask is located in the image. Given an image, Frei and Chen's masks can be used to generate 11 processed images. The choice of the specific mask depends on the specific application.

2.3.7 Frei-Chen Masks

The Frei-Chen masks are unique in that they form a complete set of basis vectors (see Section 2.5). This means that we can represent any 3×3 subimage as a weighted sum of the nine Frei-Chen masks (Figure 2.3-2). These weights are found by projecting a 3×3 subimage onto each of these masks. This projection process is similar to the convolution process in that both overlay the mask on the image, multiply coincident terms, and sum the results (a vector inner product). This is best illustrated by example.

EXAMPLE

2 - 8

Suppose that we have the following subimage, I_s :

$$I_s = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

To project this subimage onto the Frei-Chen masks, start by finding the projection onto f_1 . Overlay the subimage on the mask and consider the first row. The 1 in the upper-left corner of the subimage coincides with the 1 in the upper-left corner of the mask, the 0 is over the $\sqrt{2}$, and the 1 on the upper-right corner of the subimage coincides with the 1 in the mask. Note that all these must be summed and then multiplied by the $1/(2\sqrt{2})$ factor to normalize the masks. The projection of I_s onto f_1 is equal to

$$\frac{1}{2\sqrt{2}}[1(1) + 0(\sqrt{2}) + 1(1) + 1(0) + 0(0) + 1(0) + 1(-1) + 0(-\sqrt{2}) + 1(-1)] = 0$$

If we follow this process and project the subimage I_s onto each of the Frei-Chen masks, we get the following:

$$f_1 \rightarrow 0, f_2 \rightarrow 0, f_3 \rightarrow 0, f_4 \rightarrow 0, f_5 \rightarrow -1, f_6 \rightarrow 0, f_7 \rightarrow 0, f_8 \rightarrow -1, f_9 \rightarrow 2$$

We can now see what is meant by a complete set of basis vectors allowing us to represent a subimage by a weighted sum. The basis vectors in this case are the Frei-Chen masks, and the weights are the projection values. Take the weights and multiply them by each mask; then sum the corresponding values. For this example the only nonzero terms correspond to masks f_5, f_8 , and f_9 , and we find the following:

$$(-1)\left(\frac{1}{2}\right) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + (-1)\left(\frac{1}{6}\right) \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} + (2)\left(\frac{1}{3}\right) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} = I_s$$

FreiChen Masks

$$I_s = \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$$

$$F_1 \longrightarrow \frac{1}{2\sqrt{2}} [1 + (-1) + 1 + (-1)] = 0$$

$$F_1 \longrightarrow 0$$

$$F_6 \longrightarrow 0$$

$$F_2 \longrightarrow 0$$

$$F_7 \longrightarrow 0$$

$$F_3 \longrightarrow 0$$

$$F_8 \longrightarrow -1$$

$$F_4 \longrightarrow 0$$

$$F_9 \longrightarrow 2$$

$$F_5 \longrightarrow -1$$

FreiChen Masks

Frei Chen mask represent a subimage by a weighted sum

$$(-1)\frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + (-1)\frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} + (2)\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = I_s$$

Four mask for edge subspace

Four mask for line subspace

One mask for an average subspace

Point Detection

point mask

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Line Detection

A line is defined as a sequence (row/column) of black points within a white background (or vice versa)

Width = 1 pixel

Line masks

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

Horizontal

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

+45°

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

vertical

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

-45°

Edge Detection

An edge is defined as a sequence (row/column) of black points separating regions of black and white

Purpose:

1. Segmentation
2. Recognition – patterns
3. Motion

After applying various edge detectors one can use histogram equalization to strengthen them.

General 3x3 Mask

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

Roberts

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

Prewitt

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

Sobel

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

Isotropic (Frei-Chen)

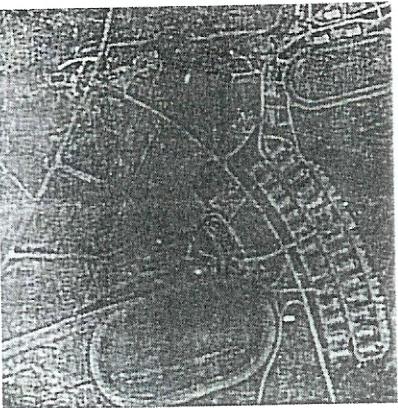
$$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Extended Prewitt

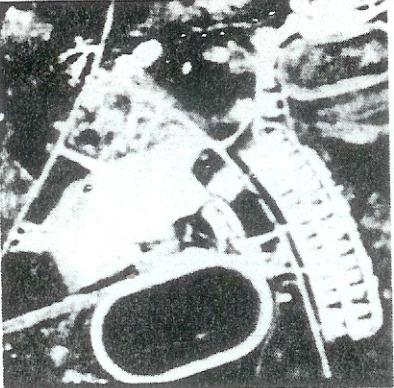
$$\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$

Pyramid

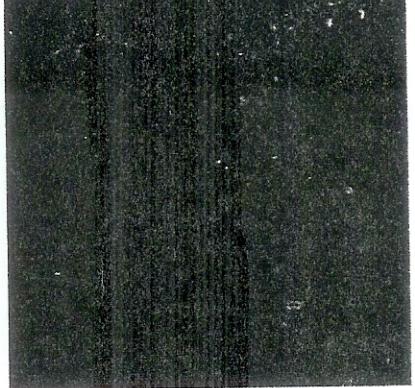
$$\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$



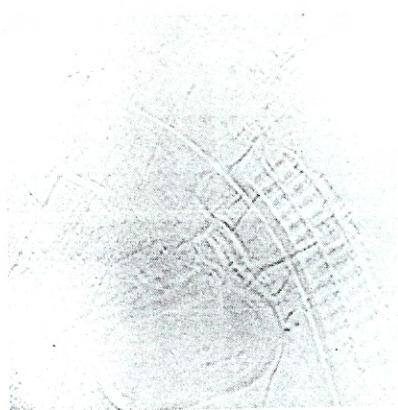
(a)



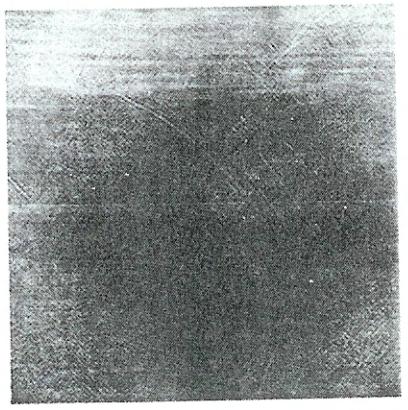
(b)



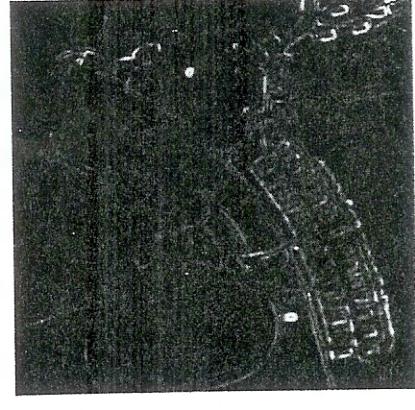
(c)



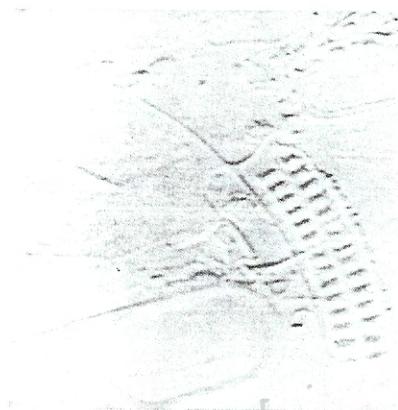
(d)



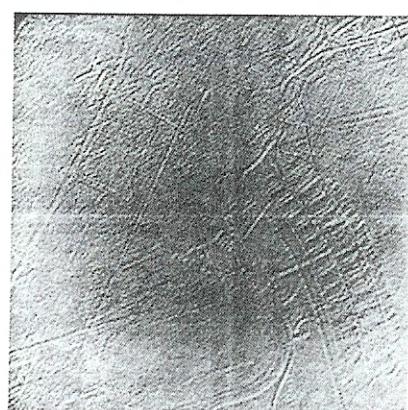
(e)



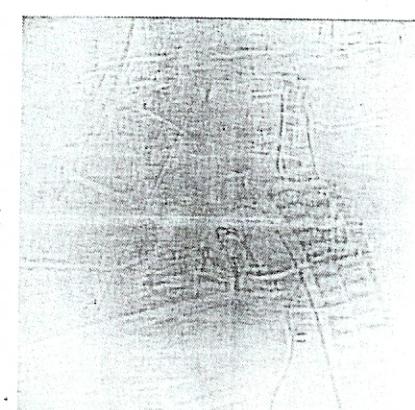
(f)



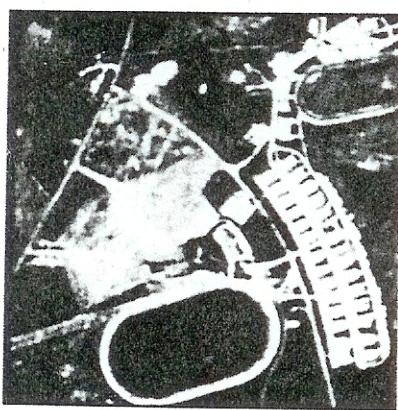
(g)



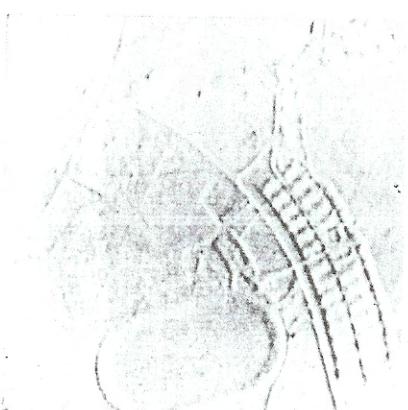
(h)



(i)



(j)



(k)

Figure 7.13 (Continued) (a)–(f) projections onto w_1, w_2, w_3, w_4 , and w_5 subspaces, (g)–(l) magnitude of projection onto edge subspace; (m) magnitude of projection onto line subspace. (From Hall and Frei [1976].)

Figure 7.13 (Continued) (a) Original image; (b)–(f) projections onto w_1, w_2, w_3, w_4 , and w_5 subspaces; (g)–(l) magnitude of projection onto edge subspace; (m) magnitude of projection onto line subspace. (From Hall and Frei [1976].)



Compass operators

(N)

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

(NW)

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

(W)

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

(SW)

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

(S)

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

(SE)

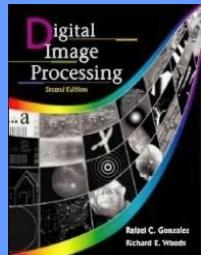
$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(E)

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

(NE)

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



Robinson Compass Mask

- Similar to Kirsch mask
- Easier with 0,1,2 values
- Symmetrical by the axis
- Compute only 4 mask
- The other mask are found by negating the results
- Edge magnitude
 - Maximum value found
- Edge direction
 - Defined by the mask that produces the maximum magnitude
 - K_0 vertical edge
 - K_5 Northwest / Southeast

$$R_0 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$R_4 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$R_6 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$R_7 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Kirsch Compass Mask

- Take a single marks and rotate it to 8 major compass orientation
- Edge magnitude
 - Maximum value found
- Edge direction
 - Defined by the mask that produces the maximum magnitude
 - K_0 vertical edge
 - K_5 Northwest / Southeast

$$K_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad K_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad K_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad K_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$
$$K_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad K_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad K_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad K_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Result of Marksing



Original image



Sobel Operator



Laplacian Operator



Kirsh Operator



Prewitt Operator



Frei-Chen Operator



Roberts Operator



Robinson Operator

10.9.5 Spline-Based Edge Detection

The cubic B-spline transform discussed in Sect. 8.6.5 for interpolation yields a continuous representation of a discrete image that is also continuous in its first and second derivative:

$$g_3(x) = \sum_n c_n \beta_3(x - n), \quad (10.84)$$

where $\beta_3(x)$ is the cubic B-spline function defined in (8.55). From this continuous representation, it is easy to compute the spatial derivatives of $g_3(x)$:

$$\frac{\partial g_3(x)}{\partial x} = \sum_n c_n \frac{\partial \beta_3(x - n)}{\partial x}. \quad (10.85)$$

For a discrete derivative filter, we only need the derivatives at the grid points. From Fig. 8.19a it can be seen that the cubic B-spline function covers at most 5 grid points. The maximum of the spline function occurs at the central grid point. Therefore, the derivative at this point is zero. It is also zero at the two outer grid points. Thus, the derivative is only unequal to zero at the direct left and right neighbors of the central point. Therefore, the derivative at the grid point x_m reduces to

$$\left. \frac{\partial g_3(x)}{\partial x} \right|_{x_m} = (c_{m+1} - c_{m-1})/2. \quad (10.86)$$

Thus the computation of the first-order derivative based on the cubic B-spline transformation is indeed an efficient solution. We apply first the cubic B-spline transform in the direction of the derivative to be computed (Sect. 8.6.5) and then the \mathcal{D}_{2x} operator. Therefore, the transfer function is given by

$$\hat{D}_x = i \frac{\sin(\pi \tilde{k}_x)}{2/3 + 1/3 \cos(\pi \tilde{k}_x)} = i\pi \tilde{k}_x - i \frac{\pi^5 \tilde{k}_x^5}{180} + O(\tilde{k}_x^7). \quad (10.87)$$

The errors in the magnitude and direction of a gradient vector based on the B-spline derivative filter are shown in Fig. 10.25. They are considerably less than for the simple difference filters (Fig. 10.21). This can be seen more quantitatively from Taylor expansions for the relative errors in the magnitude of the gradient

$$\frac{\Delta |\nabla g|}{|\nabla g|} \approx -\frac{(\pi \tilde{k})^4}{1440} (5 + 3 \cos 4\phi) \quad (10.88)$$

10.9.6 Least Squares Optimized Gradient

In this section first-order derivative filters are discussed that have been optimized using the least squares technique introduced in Sect. 3.4.2 and already used in Sect. 8.6.6 to optimize interpolation filters. Two classes of filters are presented, non-recursive filters and filters with a recursive correction filter. For the non-recursive family of filters, we use the following formulation for the transfer function:

$${}^R \hat{D} = i \sin(\pi \tilde{k}) + i \sum_{u=2}^R d_u [\sin(u\pi \tilde{k}) - u \sin(\pi \tilde{k})]. \quad (10.90)$$

This equation yields the additional constraint that the transfer function must be equal to $i\pi \tilde{k}$ for small wave numbers. This constraint reduces the degree of freedom by one and for a filter with R coefficients, so only $R - 1$ can be varied. The first coefficient d_1 is then given by

$$d_1 = 1 - \sum_{u=2}^R u d_u. \quad (10.91)$$

These filters are considerably better than those designed with the cubic B-spline interpolation (Fig. 10.25). The relative deviations from the ideal derivative filter are less than 1% for wave number up to 0.6 for the least squares optimized filters. Even better results are obtained if a recursive correction filter is applied in the same way as it was for the recursive implementation of the optimized interpolation filters (Sect. 8.6.6).

Finally, the errors in the magnitude and direction of the gradient computed from a filter with four coefficients for both filter families are shown in Figs. 10.26 and 10.27. Compare these results with Figs. 10.21-10.25. A more detailed discussion on the design of optimal derivative filters including tables with filter coefficients can be found in Jähne [1997].

Edge - Median

I Sobel

$$E_h = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \cdot x$$

$$= x_{c+1,j-1} + 2x_{c+1,j} + x_{c+1,j+1} - (x_{c-1,j-1} + 2x_{c-1,j} + x_{c-1,j+1})$$

Combine E_h and E_v

$$E^{\text{total}} = \sqrt{E_h^2 + E_v^2}$$

II W. Median with weights as in Sobel

e.g. order with gray levels $-1 < G < 1$

$$\text{med}\{x_{c+1,j-1}, x_{c+1,j}, x_{c+1,j+1}, x_{c+1,j+1}\}$$

$$\text{med}\{x_{c-1,j-1}, x_{c-1,j}, x_{c-1,j+1}, x_{c-1,j+1}\}$$

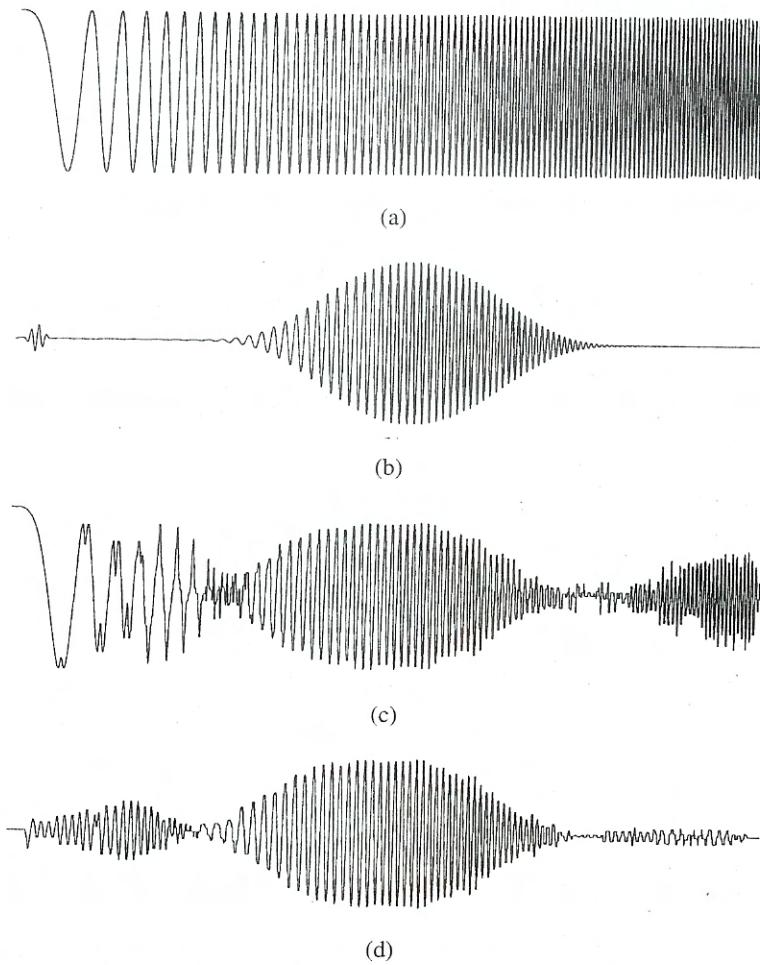


Fig. 3. Frequency selective filter outputs. (a) Chirp test signal. (b) Linear FIR filter output. (c) Weighted median smoother output. (d) Weighted median filter output with real-valued weights.

are constrained to positive values, is severely constrained. Finally, we test the new adaptive WM filter algorithm in the design of an optimal robust high-pass filter.

A. Frequency Selective WM Filters

The major limitation of WM filters with positive weights is that these filter structures are smoothers in nature. A large number of important problems in signal processing require the design of *bandpass* and *highpass* filters. Channel equalization, beamformers, and predictors are example applications where flexible frequency-selective processing is needed where weighted median filter smoothers were precluded right from the start. The new WM filter structure opens the possibility of utilizing WM filters as signal processing elements in these applications.

Since linear FIR filters output the mean of a set of weighted samples, the median of an equivalently weighted sample set ought to provide a similar output. Notably, this is the case when the same set of weights as those designed for a linear FIR filter is used in the smoother structure. As we will see shortly, the frequency response characteristics of the attained WM filter follows that of the equivalent linear FIR filter, but more importantly, these are significantly more robust in the processing of signals embedded in noise.

Fig. 3(a) depicts a linearly swept-frequency cosine signal spanning instantaneous frequencies ranging from 0 to 400 Hz. Fig. 3(b) shows the chirp signal filtered by a 120-tap linear

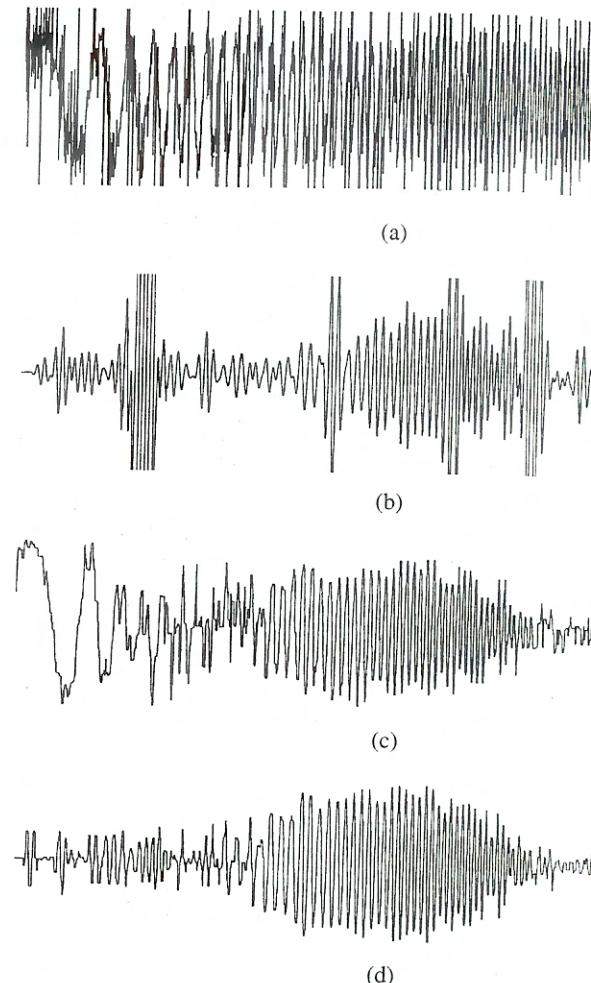
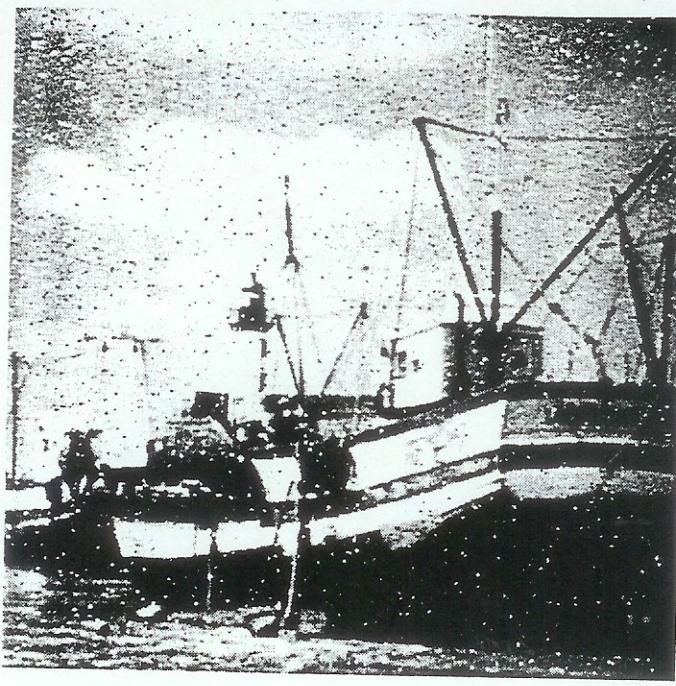


Fig. 4. Frequency selective filter outputs in noise. (a) Chirp test signal with stable noise. (b) Linear FIR filter output. (c) Weighted median smoother output. (d) Weighted median filter output with real-valued weights.

FIR filter designed by Matlab's `firl` function with $0.075 \leq \omega \leq 0.125$ (normalized frequency ω) and $M = 120$. Fig. 3(c) shows the best WM filter (smoother) when the coefficients are constrained to positive values. The positive coefficients are found by the method described in [12]. The WM smoother clearly fails to deliver the desired frequency components, and it also introduces artifacts at the low frequencies. Fig. 3(d) depicts the output of the WM filter where real-valued weights are allowed. The WM filter weights are set identically to that of the equivalent FIR filter. Fig. 3(d) shows the significant attenuation of the low-frequency components. The high-frequency components are cancelled almost completely as well. The small artifacts exhibited at low frequencies are due to the fact that the output of the WM filter is constrained to be the median of the input samples.

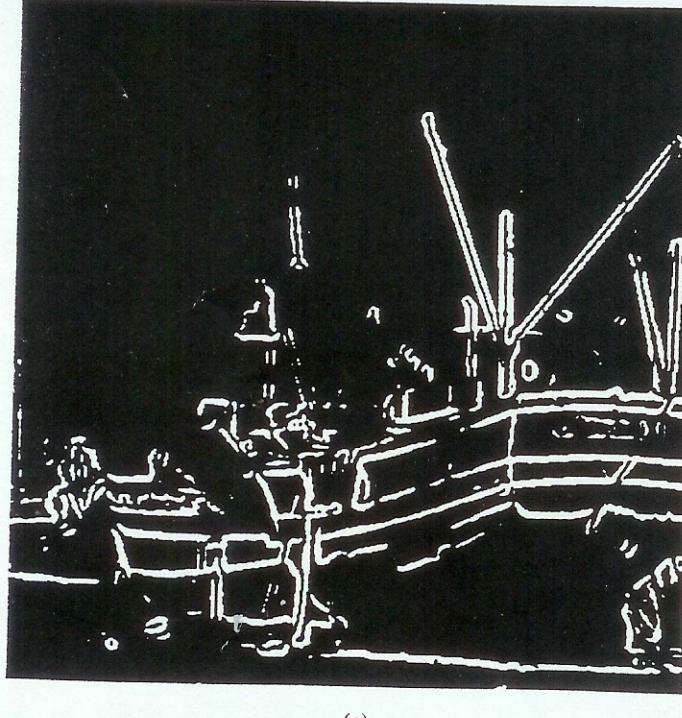
Fig. 4(a) depicts the chirp test signal with added white Gaussian noise. The parameter $\alpha = 1.4$ was used, simulating the impulsive characteristics [13]. Fig. 4(a) is truncated to the same scale as Fig. 3. Fig. 4(b) shows the signal filtered by the 120-tap linear FIR filter. The output is severely affected by the noise components. Ringing artifacts emerge with each impulse fed into the filter. Fig. 4(c) shows the WM filter output when the coefficients are constrained to positive values only. In this case, the noise does not affect the response significantly, but the response is not as smooth as the FIR filter output. Fig. 4(d) shows the WM filter output with real-valued weights, which shows a more balanced response compared to (c) and (b).



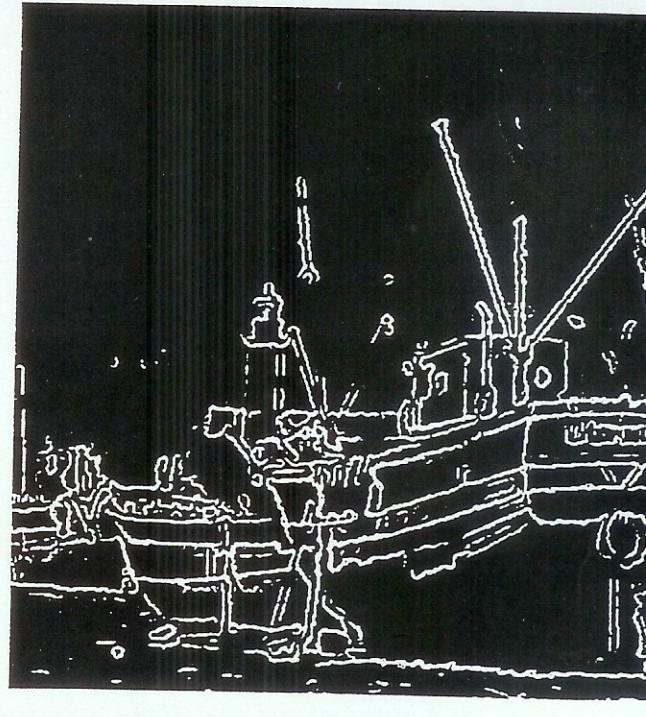
(a)



(b)

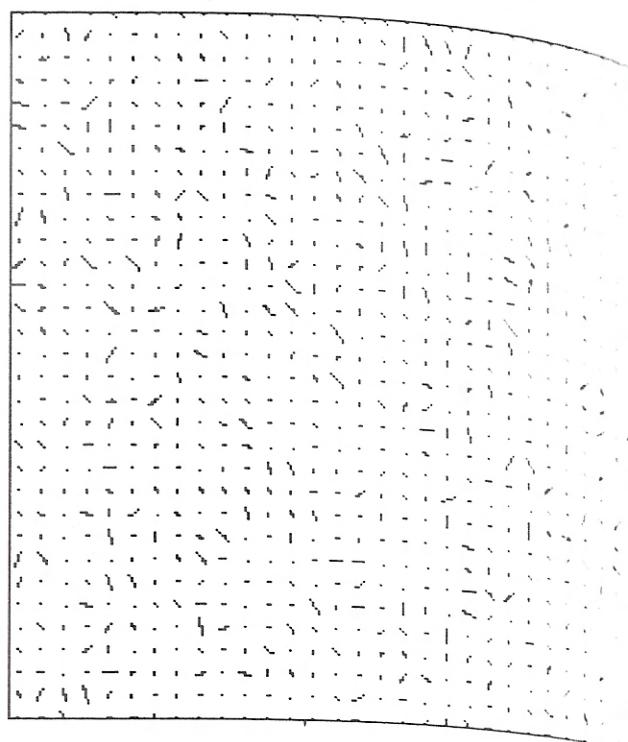
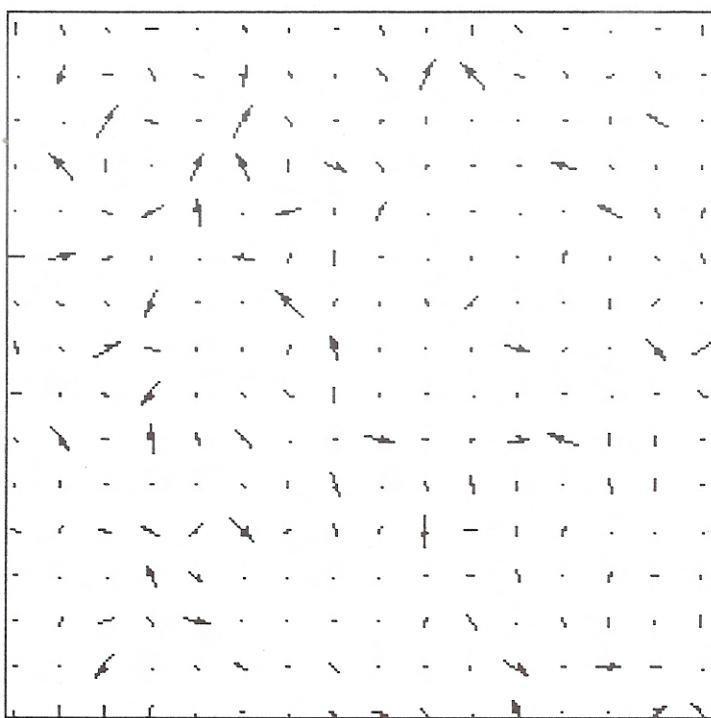


(c)



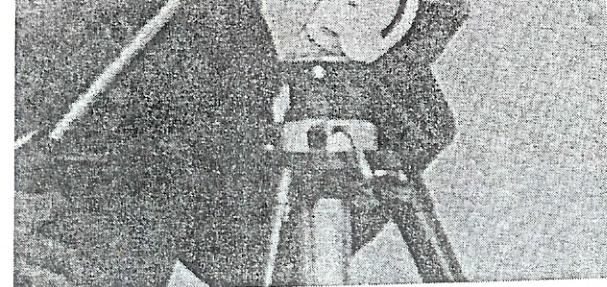
(d)

Fig. 6. (a) "Boat" blurred with a 5×5 mean filter and corrupted with impulsive noise ($p = 0.02$); (b) Sobel edge map of the corrupted image prefiltered with (c) 3×3 median, and (d) 5×5 LUM filter ($k = l = 5$). The thresholds



progression of values around each more-or-less circular hole evident. The use of a pseudo-color scale for this display is particularly suitable, since a rainbow of hues can show the progression without the arbitrary discontinuity required by the grayscale (in this example, at an angle of zero degrees). Unfortunately, the pixels within relatively homogeneous areas of the features also have colors assigned, because at every point there is some direction to the gradient, and these colors tend to overwhelm the visual impression of the image. A solution is to combine the magnitude of the gradient with the direction, as shown in **Figure 30**. This is the dendrite image from **Figure 2** processed to show the magnitude and the direction of the Sobel edge gradient, the latter in color. These two images are then multiplied together so that the intensity of the color is proportional to the magnitude of the gradient. The result clearly shows the orientation of boundaries.

Another way to improve the resulting image is to show the



2 Original cameraman image, 512×512 pixels.

of the threshold value T is a tradeoff between capturing the actual edges in the image and the noise. Increasing T decreases sensitivity to noise at the cost of missing the weakest edges, forcing the edge segments to be broken and fragmented. By decreasing T , one can obtain more connected and richer edge contours, but the sensitivity is likely to produce more false edges. If no thresholding is used, as in Eq. (3) and Fig. 3(a), the edge width increases as T increases and widen as it decreases. Figures 3(a) and 3(b) show edge maps obtained from several different threshold values.

A directional edge detector is useful. One can be constructed by decomposing the gradient into horizontal and vertical components and applying them separately. Expressed in the

$$\frac{\partial f_c}{\partial n} = \nabla f_c(x, y) \cdot n,$$

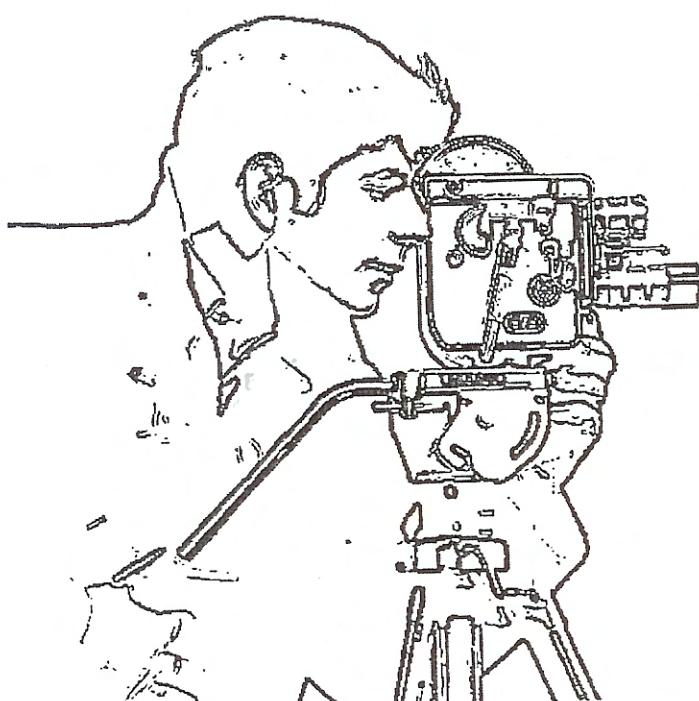
$$= \frac{\partial f_c(x, y)}{\partial x} \cos \theta + \frac{\partial f_c(x, y)}{\partial y} \sin \theta$$

where θ is the angle of n relative to the positive x axis. The directional derivative is most sensitive to edges perpendicular to n .

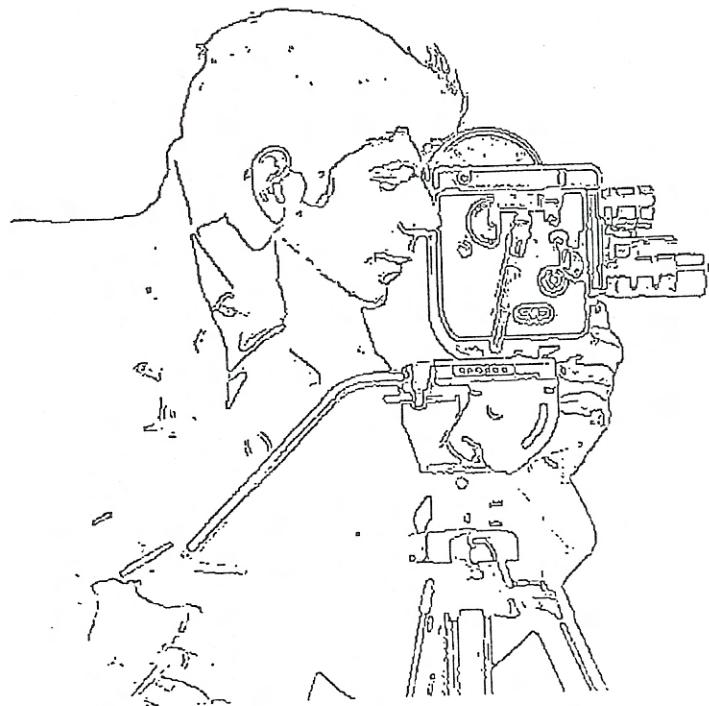
The continuous-space gradient magnitude provides an isotropic or rotationally symmetric edge detector, equivalent to edges in any direction [12]. It is easy to show that the gradient is isotropic. In addition to the original $X-Y$ coordinate system, let us introduce a new system, $X'-Y'$, which is rotated by an angle ϕ relative to $X-Y$. Let $n_{x'}$ and $n_{y'}$ be the unit vectors in the x' and y' directions, respectively. For the gradient magnitude to be isotropic, the same result must be produced in both coordinate systems, regardless of ϕ . Using Eq. (4) along with the notation, we find the partial derivatives with respect to the coordinate axes are

$$f_{x'} = \nabla f \cdot n_{x'} = f_x \cos \phi + f_y \sin \phi$$

$$f_{y'} = \nabla f \cdot n_{y'} = -f_x \sin \phi + f_y \cos \phi$$



(a)



(b)

FIGURE 3 Gradient edge detection steps, using the Sobel operator: (a) After thresholding $|\nabla f|$; (b) after thinning the edges by finding the local maximum of $|\nabla f|$ along the gradient direction.

Figure 2.3-6 (Continued)

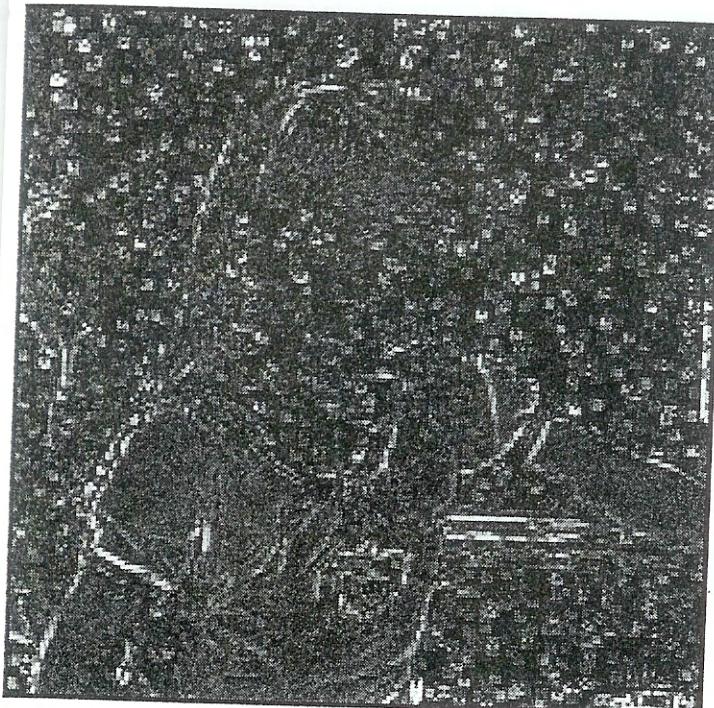
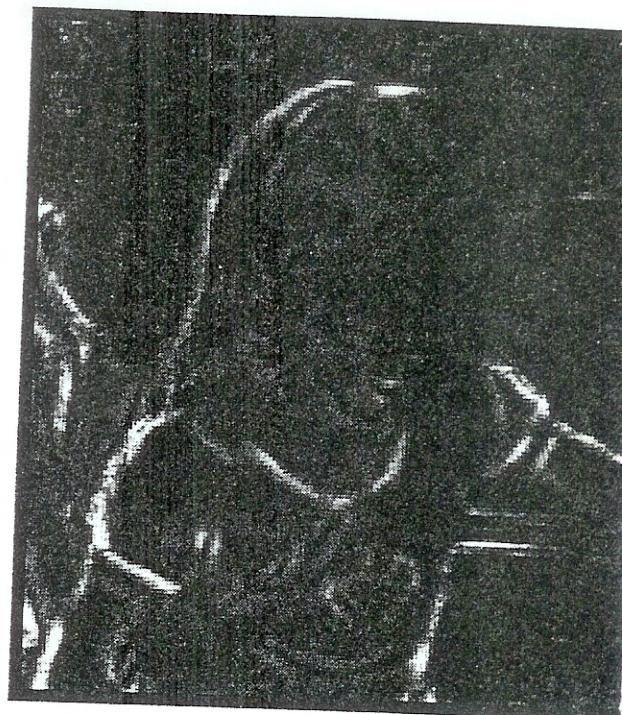
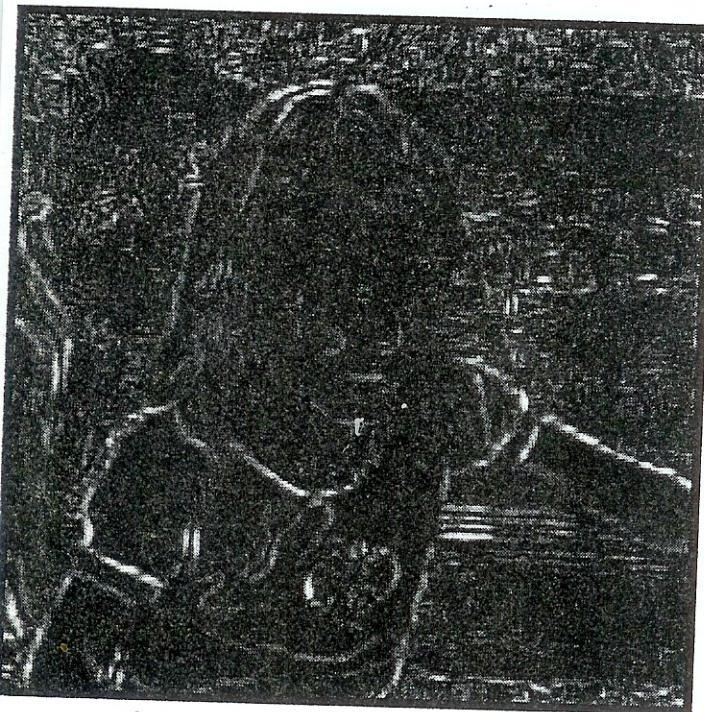
e. Prewitt with 3×3 mask.f. Prewitt with 7×7 mask.g. Truncated pyramid with 7×7 mask.

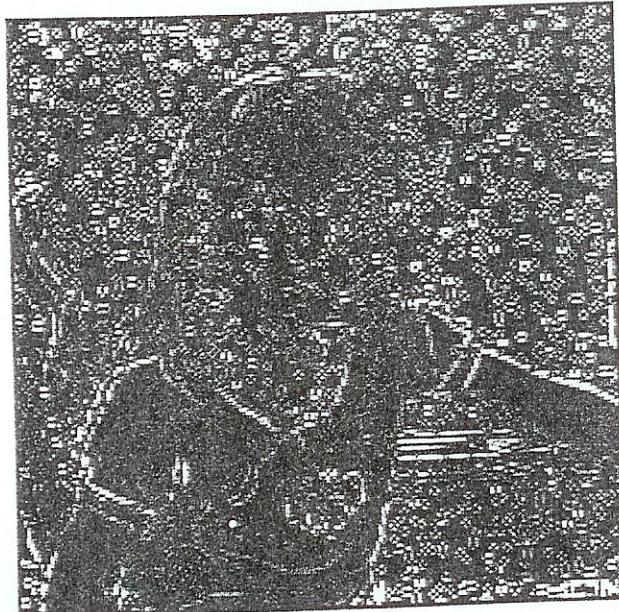
Figure 2.3-6 Edge Detection Examples—Noise



a. Original image.



b. Image with added gaussian and salt-and-pepper noise.



c. Sobel with 3×3 mask.



d. Sobel with 7×7 mask.