Spatial Degradation

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

where $g(x,y)$ is the degraded image

$h(x,y)$ is a degradation function

$f(x,y)$ is the original input image

$n(x,y)$ is additive noise

Spatial Degradation

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

where $g(x,y)$ is the degraded image

$h(x,y)$ is a degradation function

$f(x,y)$ is the original input image

$n(x,y)$ is additive noise

frequency $$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

frequency $$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

space $\quad g(x,y) = h(x,y) * f(x,y) + n(x,y)$

frequency $\quad G(u,v) = H(u,v)F(u,v) + N(u,v)$

If input image $f(x,y)$ is impulse $A\delta(x,y)$ then

$$h(x,y) * f(x,y) = Ah(x,y)$$

space $g(x,y) = h(x,y) * f(x,y) + n(x,y)$

frequency $G(u,v) = H(u,v)F(u,v) + N(u,v)$

If input image $f(x,y)$ is impulse $A\delta(x,y)$ then

$$h(x,y) * f(x,y) = Ah(x,y)$$

If $f(x,y) = A\delta(x,y)$ then $G(u,v) = AH(u,v) + N(u,v)$

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

frequency $$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If input image $f(x,y)$ is impulse $A\delta(x,y)$ then

$$h(x,y) * f(x,y) = Ah(x,y)$$

If $f(x,y) = A\delta(x,y)$ then $G(u,v) = AH(u,v) + N(u,v)$

We can measure $H(u,v)$ for a system by using the Known input $f(x,y) = A\delta(x,y)$ and making the input bright enough so that the noise is negligible. Then

$$H(u,v) = \frac{G(u,v)}{A}$$

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

frequency $$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If input image $f(x,y)$ is impulse $A\delta(x,y)$ then

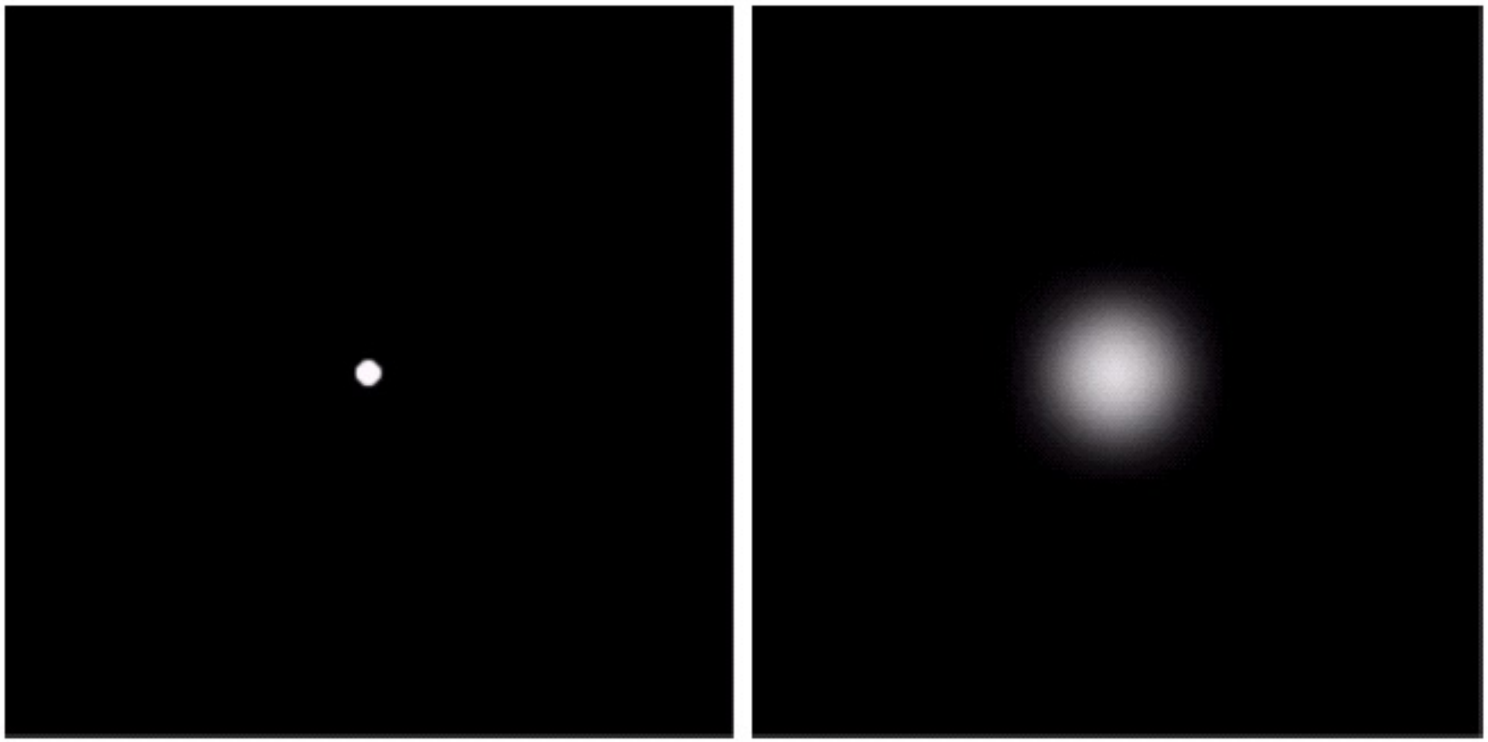
$$h(x,y) * f(x,y) = Ah(x,y)$$

If $f(x,y) = A\delta(x,y)$ then $G(u,v) = AH(u,v) + N(u,v)$

We can measure $H(u,v)$ for a system by using the Known input $f(x,y) = A\delta(x,y)$ and making the input bright enough so that the noise is negligible. Then

$$H(u,v) = \frac{G(u,v)}{A}$$

$h(x,y)$ is called the impulse response or point spread function of the system.



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

Ex For blurring due to atmospheric turbulence

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Inverse Filtering

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Inverse Filtering

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If the spatial degradation $H(u,v)$ is known, we can compute

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Inverse Filter

Inverse Filtering

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If the spatial degradation $H(u,v)$ is known, we can compute

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)} \qquad \text{Inverse Filter}$$

If $H(u,v)$ is small at frequency (u,v), then $|\hat{F}(u,v) - F(u,v)|$ can be large at (u,v) causing the restoration to be inaccurate.

Inverse Filtering

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If the spatial degradation $H(u,v)$ is known, we can compute

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$ Inverse Filter

If $H(u,v)$ is small at frequency (u,v), then $|\hat{F}(u,v) - F(u,v)|$ can be large at (u,v) causing the restoration to be inaccurate.

$H(u,v) =$

$e^{-0.0025[u^2+v^2]^{5/6}}$

Known exactly

Inverse Filtering

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

If the spatial degradation $H(u,v)$ is known, we can compute

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$ Inverse Filter

If $H(u,v)$ is small at frequency (u,v), then $|\hat{F}(u,v) - F(u,v)|$ can be large at (u,v) causing the restoration to be inaccurate.

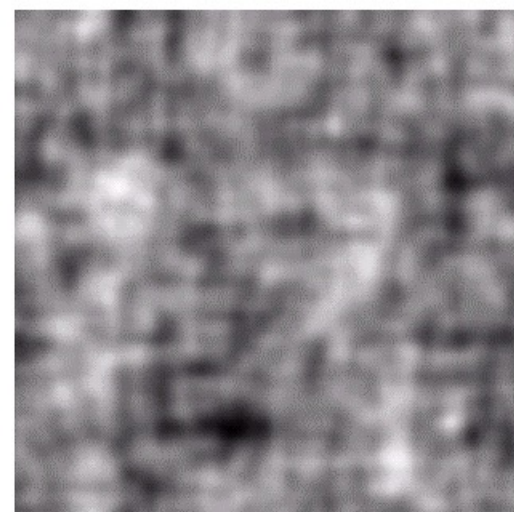
$H(u,v) =$

$e^{-0.0025[u^2+v^2]^{5/6}}$

Known exactly

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with *H* cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.



$H(u,v)$ is often small for high frequencies causing $\hat{F}(u,v)$ to be inaccurate for high frequencies.

$H(u,v)$ is often small for high frequencies causing $\hat{F}(u,v)$ to be inaccurate for high frequencies.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Inverse Filter

$H(u,v)$ is often small for high frequencies causing $\hat{F}(u,v)$ to be inaccurate for high frequencies.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)} \qquad \text{Inverse Filter}$$

We can use

$$\text{Lowpass}\left[\hat{F}(u,v)\right] = \text{Lowpass}\left[\frac{G(u,v)}{H(u,v)}\right]$$

$$= \text{Lowpass}\left[F(u,v)\right] + \text{Lowpass}\left[\frac{N(u,v)}{H(u,v)}\right],$$

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

Wiener Filtering

$$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

Find restored image $\hat{f}$ that minimizes $E\left[|f-\hat{f}|^2\right]$

Wiener Filtering

$$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

Find restored image $\hat{f}$ that minimizes $E\left[|f-\hat{f}|^2\right]$

Minimizing filter is given by

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\left(\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}\right)\right] G(u,v)$$

Wiener Filter

where $S_n(u,v)$ = power spectrum of noise

$S_f(u,v)$ = power spectrum of f

Wiener Filtering

$$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

Find restored image $\hat{f}$ that minimizes $E\left[|f-\hat{f}|^2\right]$

Minimizing filter is given by

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\left(\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}\right)\right] G(u,v)$$

Wiener Filter

where $S_n(u,v)$ = power spectrum of noise

$S_f(u,v)$ = power spectrum of f

If $n(x,y)=0$, then $S_n(u,v)=0$ and $\hat{F}(u,v) = \dfrac{G(u,v)}{H(u,v)}$

so Wiener filter becomes inverse filter.

If $S_n(u,v)$ and $S_f(u,v)$ are not known, we can approximate the Wiener filter by letting

$$\frac{S_n(u,v)}{S_f(u,v)} = K \qquad \text{constant}$$

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Restoration by Frequency Domain Filtering

Restoration by Frequency Domain Filtering

Band reject filters (BRF)

Restoration by Frequency Domain Filtering

Band reject filters (BRF)

Ideal BRF

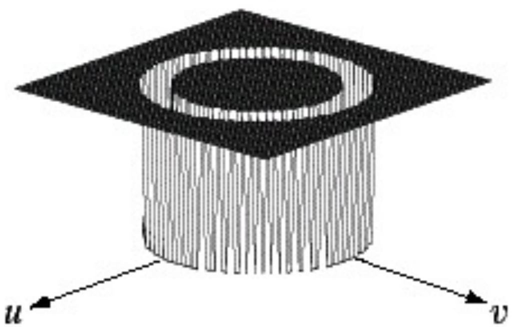


FIGURE 5.15

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

$D(u,v) = \sqrt{u^2+v^2}$, W is width of rejection band,

D_0 is distance from origin $(0,0)$ to center of rejection band

Restoration by Frequency Domain Filtering

Band reject filters (BRF)

Ideal BRF

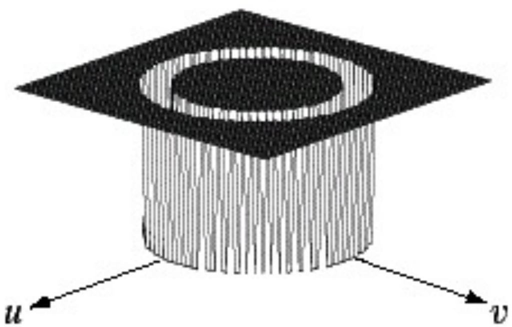


FIGURE 5.15

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

$D(u,v) = \sqrt{u^2+v^2}$, W is width of rejection band,

D_0 is distance from origin $(0,0)$ to center of rejection band

Butterworth BRF

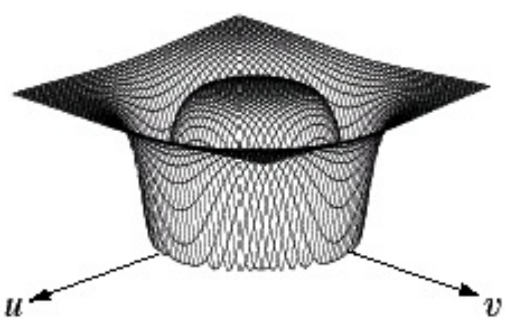


FIGURE 5.15

$$H(u,v) = \frac{1}{1 + \left[\dfrac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}$$

n is order of filter and controls sharpness.

Gaussian BRF

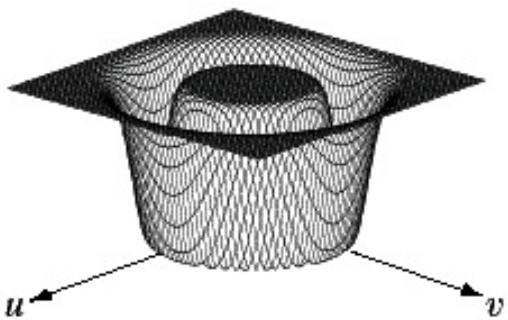
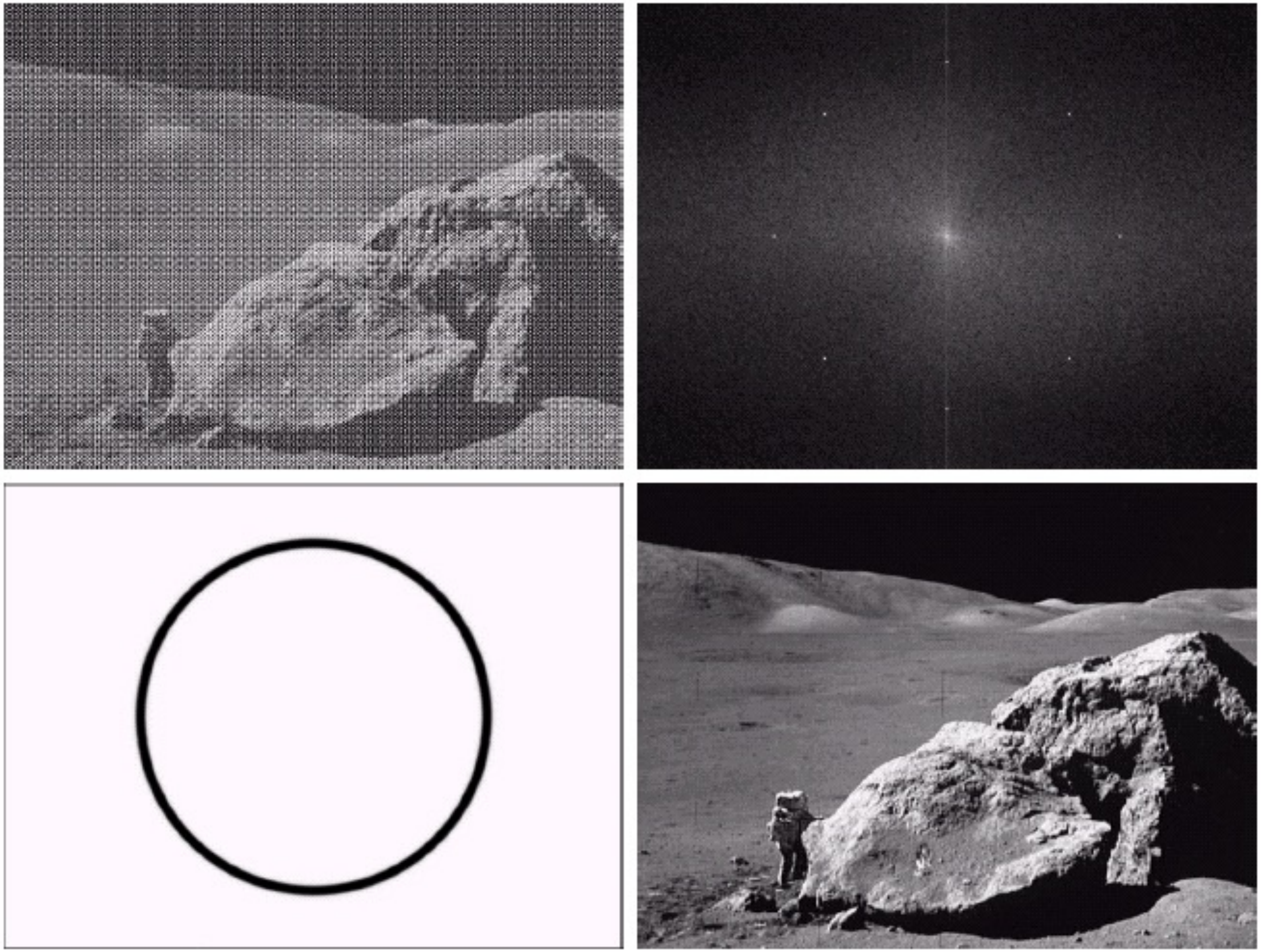


FIGURE 5.15

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]^2}$$

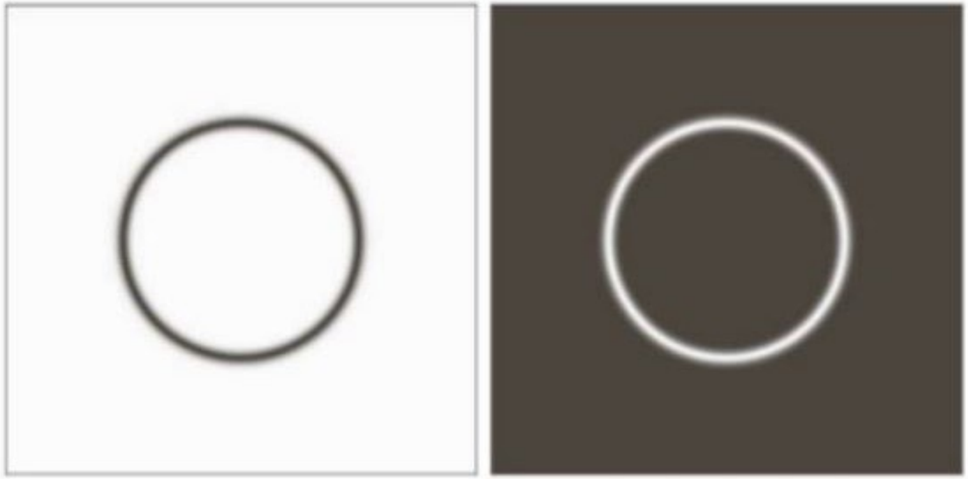


a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Bandpass Filters

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

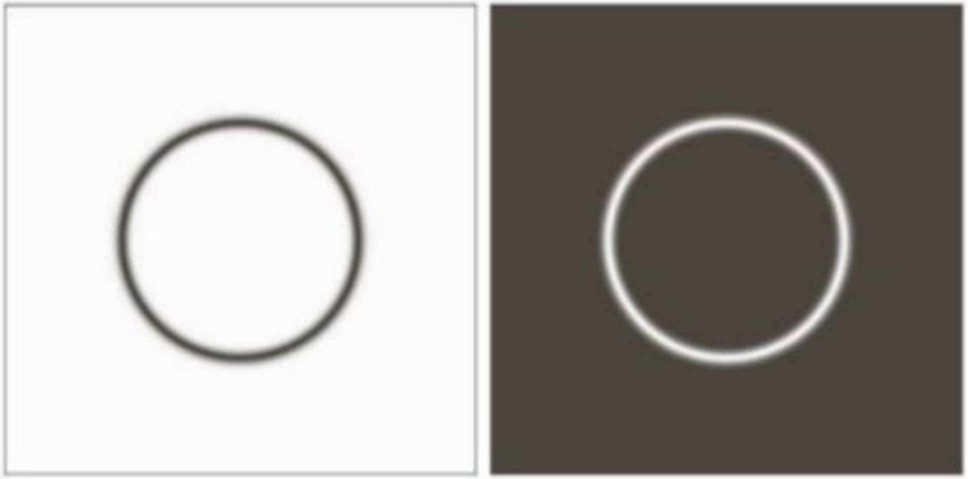


a b

FIGURE 4.63 (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

Bandpass Filters

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$



a b

FIGURE 4.63 (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

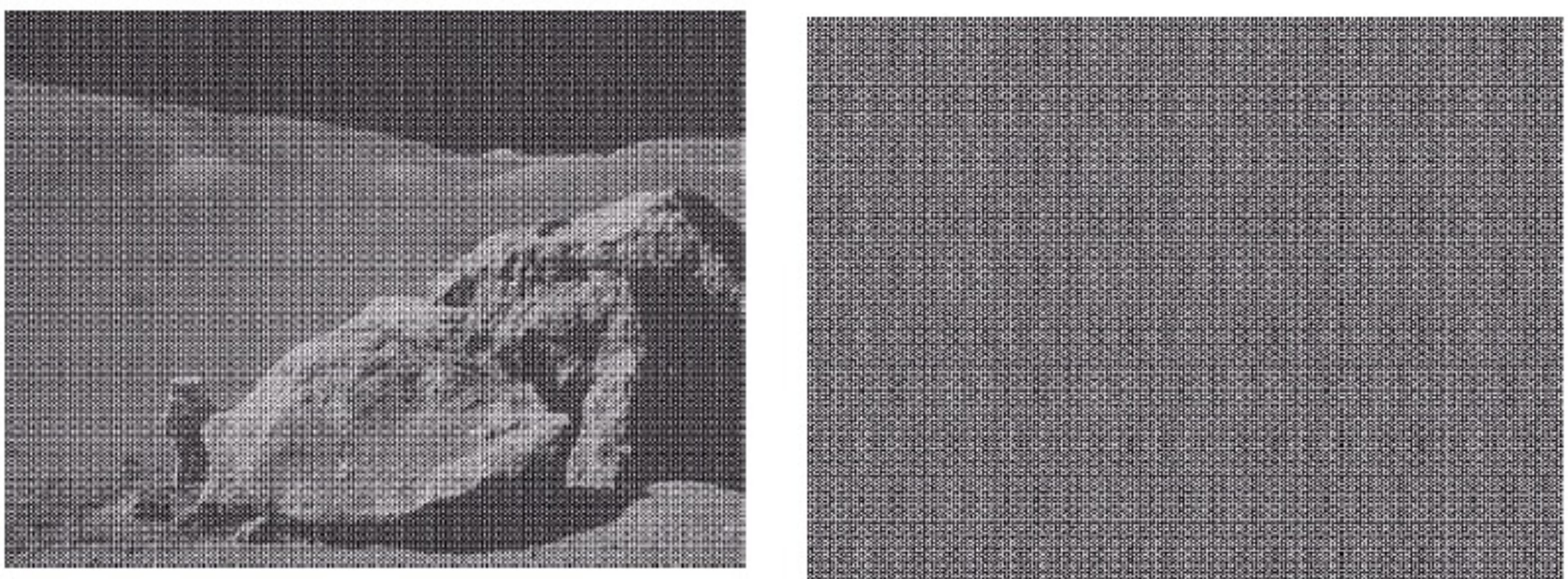


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

Notch Filters

Ideal Notch Reject Filter

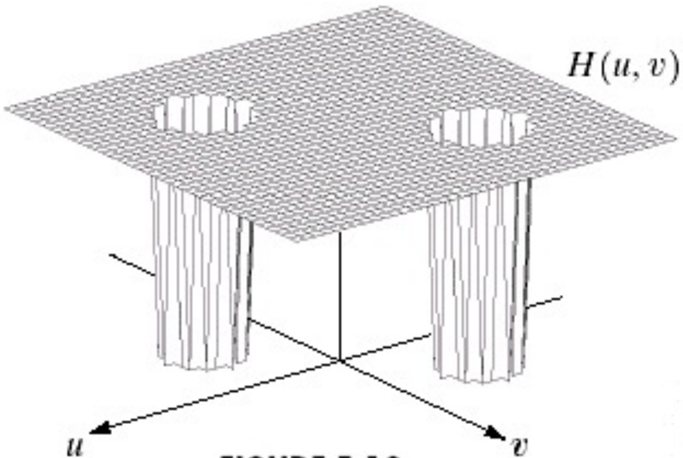


FIGURE 5.18

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \text{ or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u,v) = \left[(u-u_0)^2 + (v-v_0)^2\right]^{1/2}$$

$$D_2(u,v) = \left[(u+u_0)^2 + (v+v_0)^2\right]^{1/2}$$

Notch Filters

Ideal Notch Reject Filter

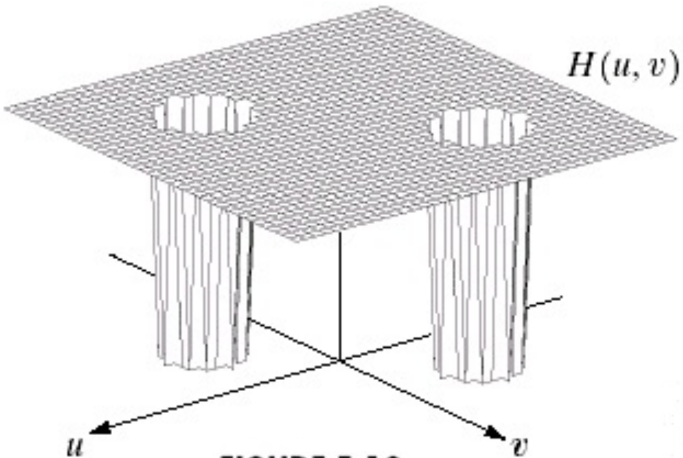


FIGURE 5.18

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \text{ or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u,v) = \left[(u-u_0)^2 + (v-v_0)^2 \right]^{1/2}$$

$$D_2(u,v) = \left[(u+u_0)^2 + (v+v_0)^2 \right]^{1/2}$$

Butterworth Notch Reject Filter

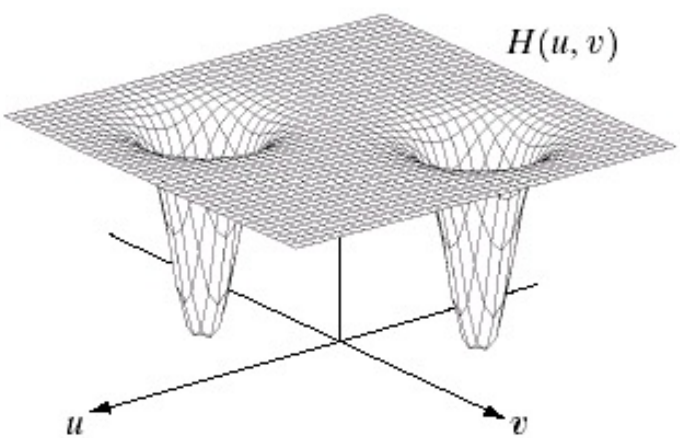


FIGURE 5.18

$$H(u,v) = \frac{1}{1 + \left[\dfrac{D_0^2}{D_1(u,v) D_2(u,v)} \right]^n}$$

order n

Gaussian Notch Reject Filter

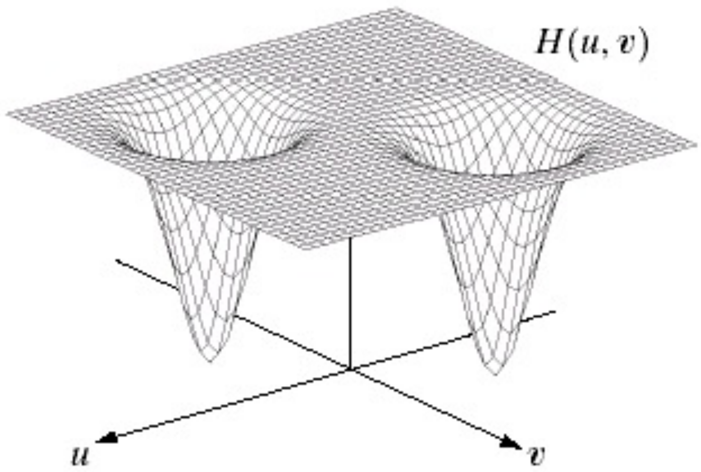


FIGURE 5.18

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$

Gaussian Notch Reject Filter

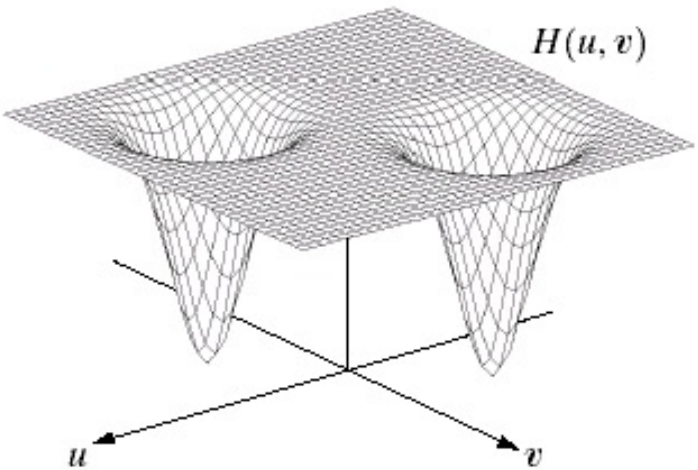


FIGURE 5.18

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$

If $(u_0, v_0) = (0,0)$ then notch reject filter rejects low frequencies and is a high pass filter.

Notchpass Filters

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

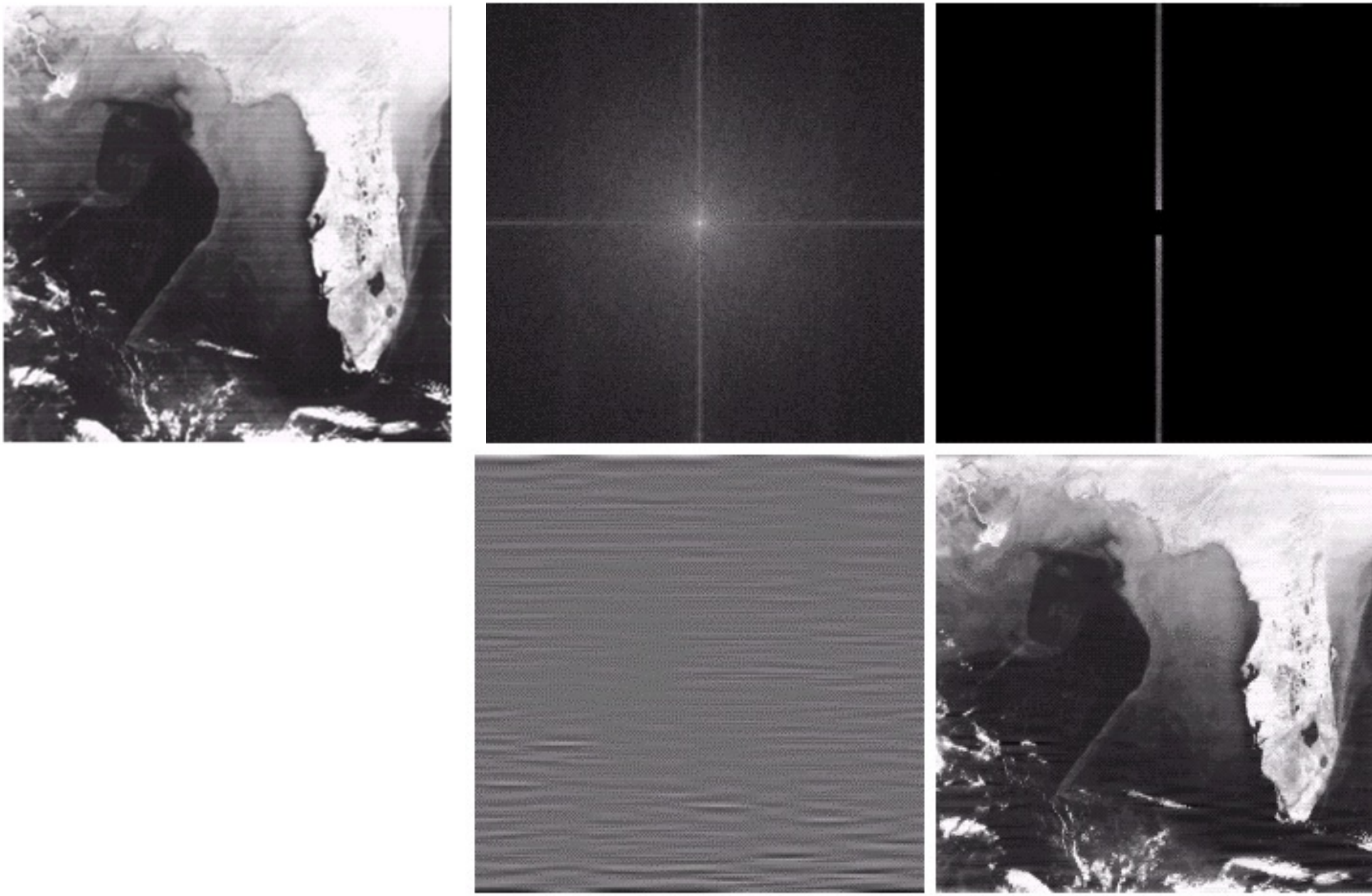
where $H_{nr}(u,v)$ is the transfer function for a notch reject filter

Notchpass Filters

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

where $H_{nr}(u,v)$ is the transfer function for a notch reject filter

If $(u_0, v_0) = (0,0)$ then notchpass filter is a lowpass filter.



sor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Restoring Geometric Distortion

ideal image $f(x,y)$

distorted image $g(r(x,y), s(x,y)) = f(x,y)$

Pixels in image f can move to new locations in image g.

Restoring Geometric Distortion

ideal image $f(x,y)$

distorted image $g(r(x,y), s(x,y)) = f(x,y)$

Pixels in image f can move to new locations in image g.

Ex $r(x,y) = x \quad s(x,y) = y \longrightarrow g(x,y) = f(x,y)$

Restoring Geometric Distortion

ideal image $f(x,y)$

distorted image $g(r(x,y), s(x,y)) = f(x,y)$

Pixels in image f can move to new locations in image g.

Ex $r(x,y) = x$ $\quad s(x,y) = y \longrightarrow g(x,y) = f(x,y)$

Ex $r(x,y) = \frac{x}{2}$ $\quad s(x,y) = \frac{y}{2} \longrightarrow g(\frac{x}{2}, \frac{y}{2}) = f(x,y)$

$g(3,3) = f(6,6)$, g is half the size of f

Restoring Geometric Distortion

ideal image $f(x,y)$

distorted image $g(r(x,y), s(x,y)) = f(x,y)$

Pixels in image f can move to new locations in image g.

Ex $r(x,y) = x \quad s(x,y) = y \longrightarrow g(x,y) = f(x,y)$

Ex $r(x,y) = \frac{x}{2} \quad s(x,y) = \frac{y}{2} \longrightarrow g(\frac{x}{2}, \frac{y}{2}) = f(x,y)$

$g(3,3) = f(6,6)$, g is half the size of f

$r(x,y)$ and $s(x,y)$ are usually specified by tiepoints

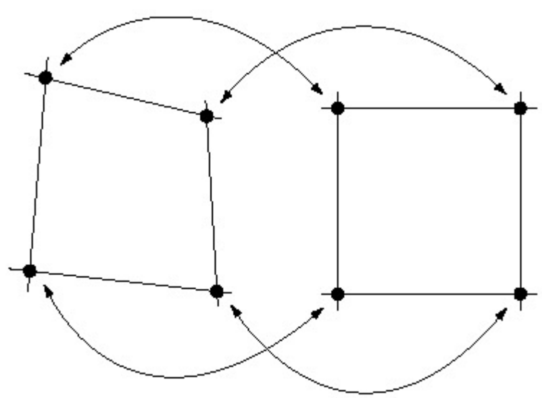
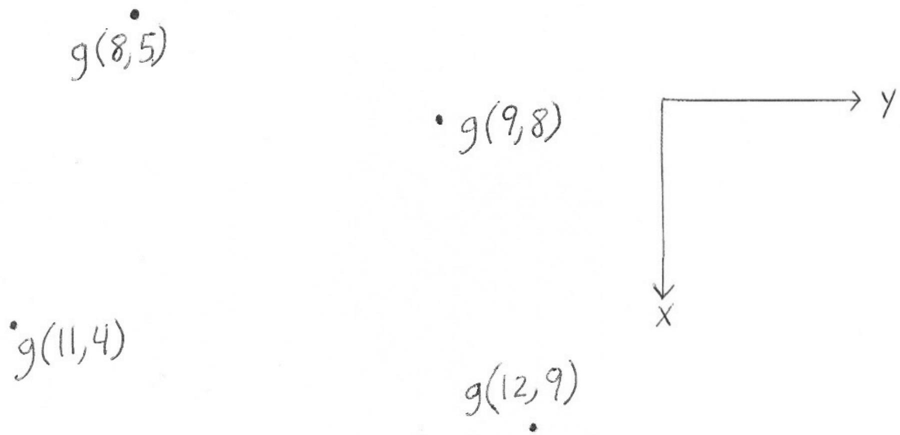
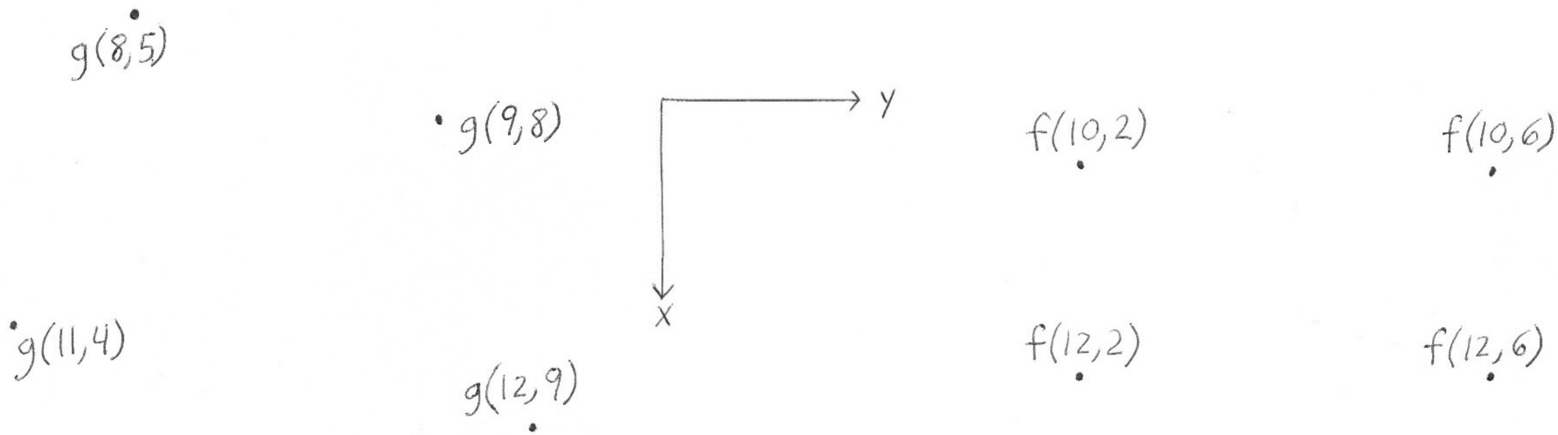


FIGURE 5.32 Corresponding tiepoints in two image segments.

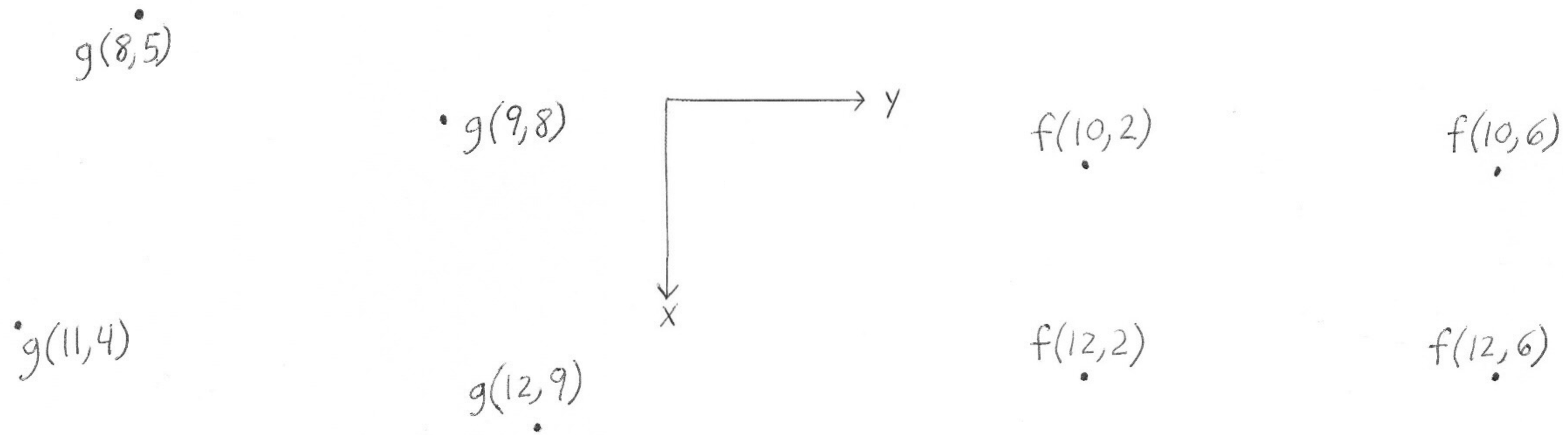
Ex



Ex



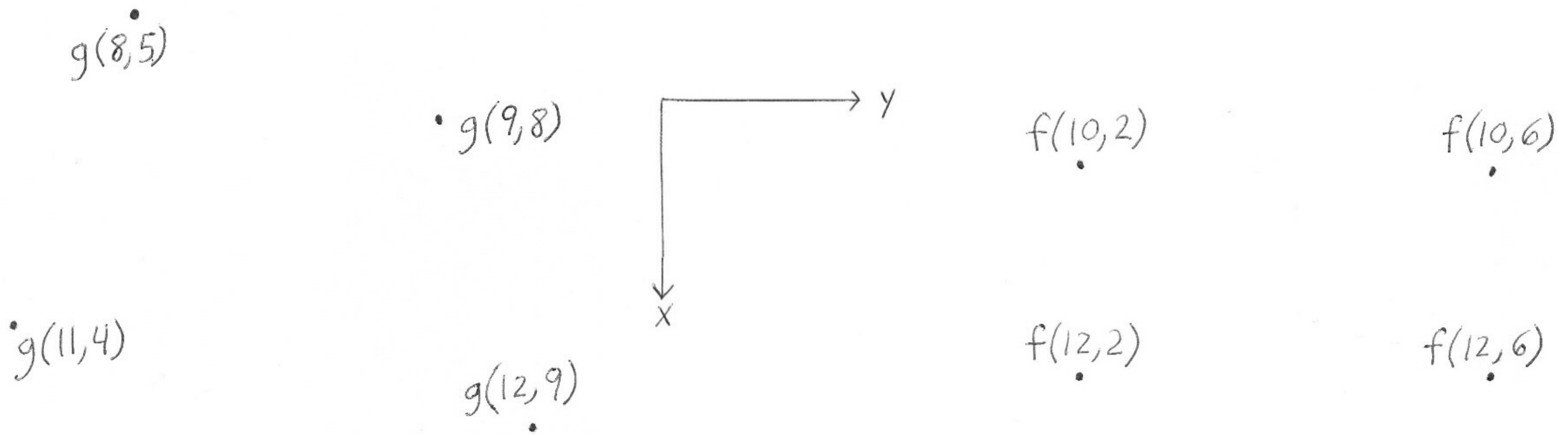
Ex



$$g(8,5) = f(10,2) \qquad g(9,8) = f(10,6)$$

$$g(11,4) = f(12,2) \qquad g(12,9) = f(12,6)$$

Ex

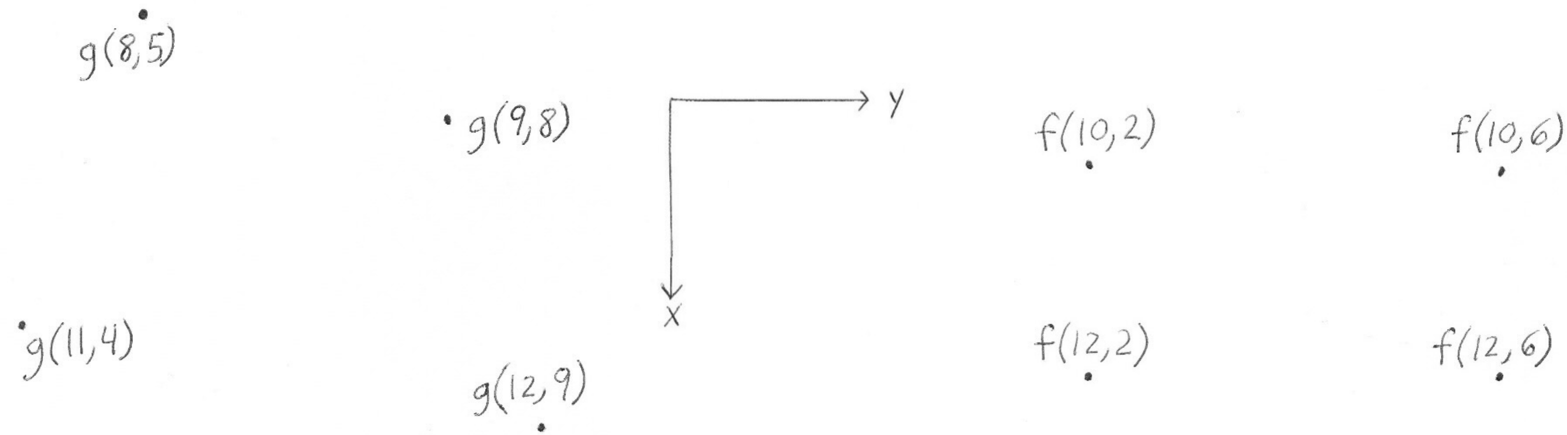


$$g(8,5) = f(10,2) \qquad g(9,8) = f(10,6)$$

$$g(11,4) = f(12,2) \qquad g(12,9) = f(12,6)$$

$$g(r(x,y), s(x,y)) = f(x,y)$$

Ex



$g(8,5) = f(10,2)$ $\quad$ $g(9,8) = f(10,6)$

$g(11,4) = f(12,2)$ $\quad$ $g(12,9) = f(12,6)$

$$g(r(x,y), s(x,y)) = f(x,y)$$

$r(10,2) = 8$ $\quad$ $r(10,6) = 9$

$s(10,2) = 5$ $\quad$ $s(10,6) = 8$

$r(12,2) = 11$ $\quad$ $r(12,6) = 12$

$s(12,2) = 4$ $\quad$ $s(12,6) = 9$

Can use a bilinear model for the geometric distortion to map between the quadrilateral in f and the quadrilateral in g. Model is given by

$$r(x,y) = c_1 x + c_2 y + c_3 xy + c_4$$
$$s(x,y) = c_5 x + c_6 y + c_7 xy + c_8$$

4 equations in 4 unknowns for c_1, c_2, c_3, c_4
4 equations in 4 unknowns for c_5, c_6, c_7, c_8

For previous example, for $r(x,y)$ we have

at $(10,2)$ $\quad 8 = c_1 10 + c_2 2 + c_3 (10)(2) + c_4$

at $(12,2)$ $\quad 11 = c_1 12 + c_2 2 + c_3 (12)(2) + c_4$

at $(10,6)$ $\quad 9 = c_1 10 + c_2 6 + c_3 (10)(6) + c_4$

at $(12,6)$ $\quad 12 = c_1 12 + c_2 6 + c_3 (12)(6) + c_4$

Similar equations can be solved for c_5, c_6, c_7, c_8

For previous example, for $r(x,y)$ we have

at $(10,2)$ $\quad 8 = c_1 10 + c_2 2 + c_3 (10)(2) + c_4$

at $(12,2)$ $\quad 11 = c_1 12 + c_2 2 + c_3 (12)(2) + c_4$

at $(10,6)$ $\quad 9 = c_1 10 + c_2 6 + c_3 (10)(6) + c_4$

at $(12,6)$ $\quad 12 = c_1 12 + c_2 6 + c_3 (12)(6) + c_4$

Similar equations can be solved for c_5, c_6, c_7, c_8

Model each image as the union of a set of quadrilaterals defined by tiepoints.

Generate pixel values in the restored image using

$$\hat{f}(x_0, y_0) = g(r(x_0, y_0), s(x_0, y_0))$$

for $(x_0, y_0) = (0,0), (0,1), \cdots, (M-1, N-1)$

where $r(x_0, y_0)$, $s(x_0, y_0)$ are the bilinear model equations evaluated at $(x, y) = (x_0, y_0)$ for the quadrilateral in f that contains (x_0, y_0)

Generate pixel values in the restored image using

$$\hat{f}(x_0, y_0) = g(r(x_0, y_0), s(x_0, y_0))$$

for $(x_0, y_0) = (0,0), (0,1), \cdots, (M-1, N-1)$

where $r(x_0, y_0)$, $s(x_0, y_0)$ are the bilinear model equations evaluated at $(x, y) = (x_0, y_0)$ for the quadrilateral in f that contains (x_0, y_0)

For a given coordinate (x_0, y_0), the transformed coordinate $(r(x_0, y_0), s(x_0, y_0))$ may not be a pair of integers.

Generate pixel values in the restored image using

$$\hat{f}(x_0, y_0) = g(r(x_0, y_0), s(x_0, y_0))$$

for $(x_0, y_0) = (0,0), (0,1), \cdots, (M-1, N-1)$

where $r(x_0, y_0)$, $s(x_0, y_0)$ are the bilinear model equations evaluated at $(x,y) = (x_0, y_0)$ for the quadrilateral in f that contains (x_0, y_0)

For a given coordinate (x_0, y_0), the transformed coordinate $(r(x_0, y_0), s(x_0, y_0))$ may not be a pair of integers.

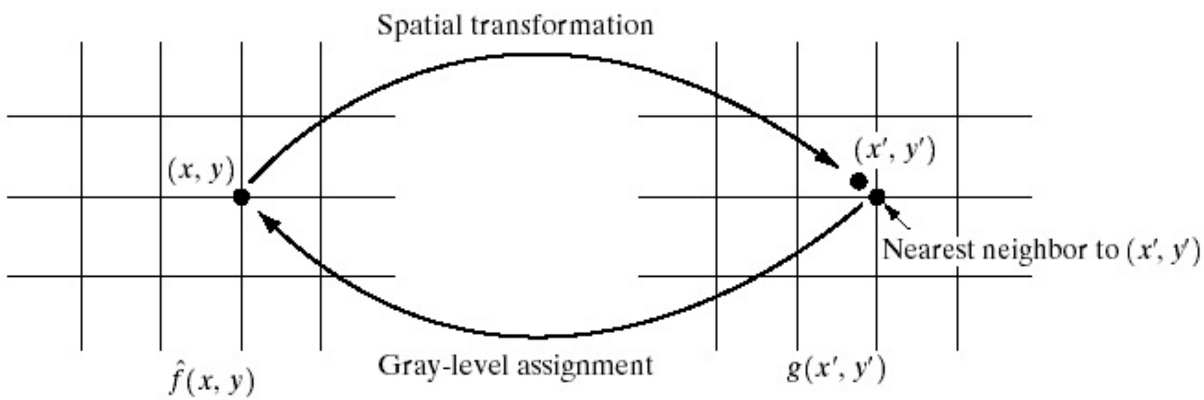


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

Determining $g(r(x_0, y_0), s(x_0, y_0))$ requires gray level interpolation.

Determining $g(r(x_0,y_0), s(x_0,y_0))$ requires gray level interpolation.

Nearest neighbor interpolation – find the closest integer coordinate $(\hat{r}, \hat{s})$ to $(r(x_0,y_0), s(x_0,y_0))$ and use $\hat{f}(x_0,y_0) = g(\hat{r}, \hat{s})$ for the restored image

Determining $g(r(x_0,y_0), s(x_0,y_0))$ requires gray level interpolation.

Nearest neighbor interpolation – find the closest integer coordinate $(\hat{r}, \hat{s})$ to $(r(x_0,y_0), s(x_0,y_0))$ and use $\hat{f}(x_0,y_0) = g(\hat{r}, \hat{s})$ for the restored image

Bilinear interpolation – find the four closest integer coordinates to $(r(x_0,y_0), s(x_0,y_0))$. Let these four pixels have values $g(x_1,y_1), g(x_2,y_2), g(x_3,y_3), g(x_4,y_4)$

Assume a continuous model for $g(x,y)$ near $(r(x_0,y_0), s(x_0,y_0))$ of the form

$$g(x,y) \doteq ax + by + cxy + d$$

Assume a continuous model for $g(x,y)$ near $(r(x_0,y_0), s(x_0,y_0))$ of the form

$$g(x,y) \doteq ax + by + cxy + d$$

Solve for a,b,c,d using

$$g(x_1,y_1) = ax_1 + by_1 + cx_1y_1 + d$$

$$g(x_2,y_2) = ax_2 + by_2 + cx_2y_2 + d$$

$$g(x_3,y_3) = ax_3 + by_3 + cx_3y_3 + d$$

$$g(x_4,y_4) = ax_4 + by_4 + cx_4y_4 + d$$

Assume a continuous model for $g(x,y)$ near $(r(x_0,y_0), s(x_0,y_0))$ of the form

$$g(x,y) \doteq ax + by + cxy + d$$

Solve for a,b,c,d using

$$g(x_1,y_1) = ax_1 + by_1 + cx_1y_1 + d$$
$$g(x_2,y_2) = ax_2 + by_2 + cx_2y_2 + d$$
$$g(x_3,y_3) = ax_3 + by_3 + cx_3y_3 + d$$
$$g(x_4,y_4) = ax_4 + by_4 + cx_4y_4 + d$$

Use $\hat{f}(x_0,y_0) = ar(x_0,y_0) + bs(x_0,y_0) + cr(x_0,y_0)s(x_0,y_0) + d$ for the restored image.

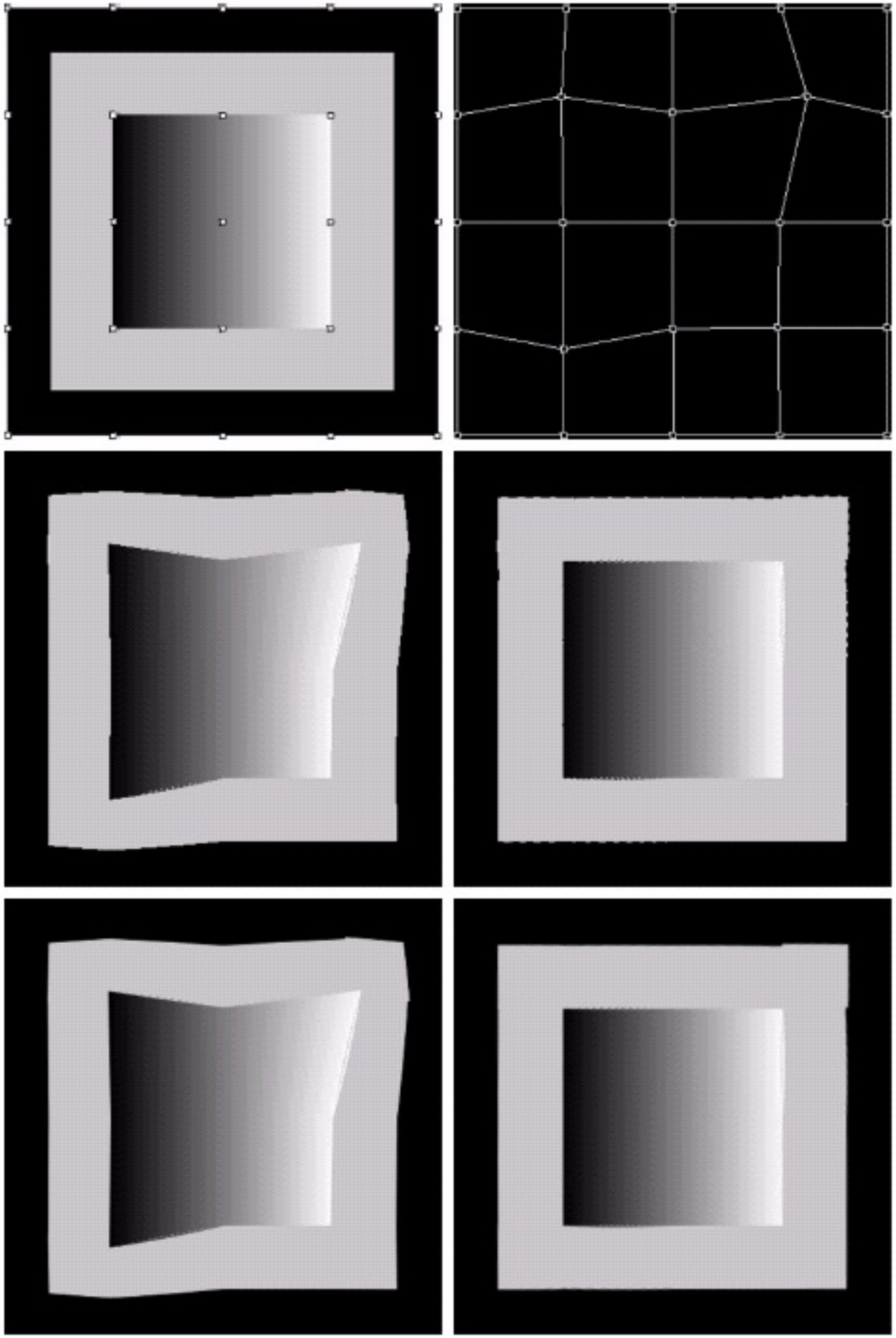
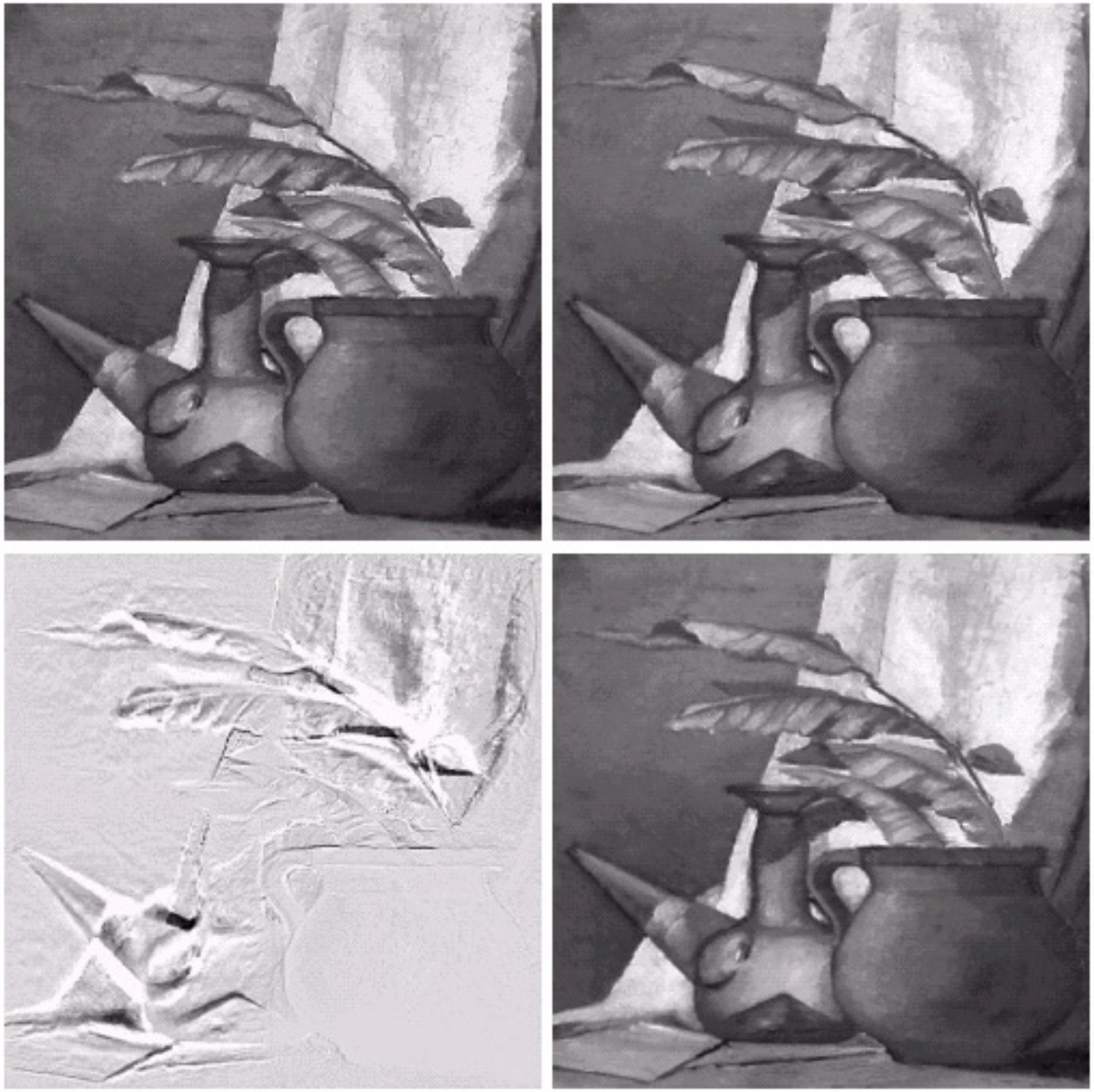


FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.



a b
c d

FIGURE 5.35 (a) An image before geometric distortion. (b) Image geometrically distorted using the same parameters as in Fig. 5.34(e). (c) Difference between (a) and (b). (d) Geometrically restored image.