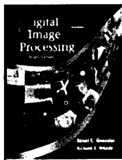


Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

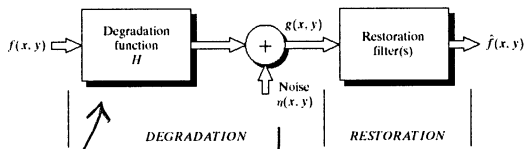


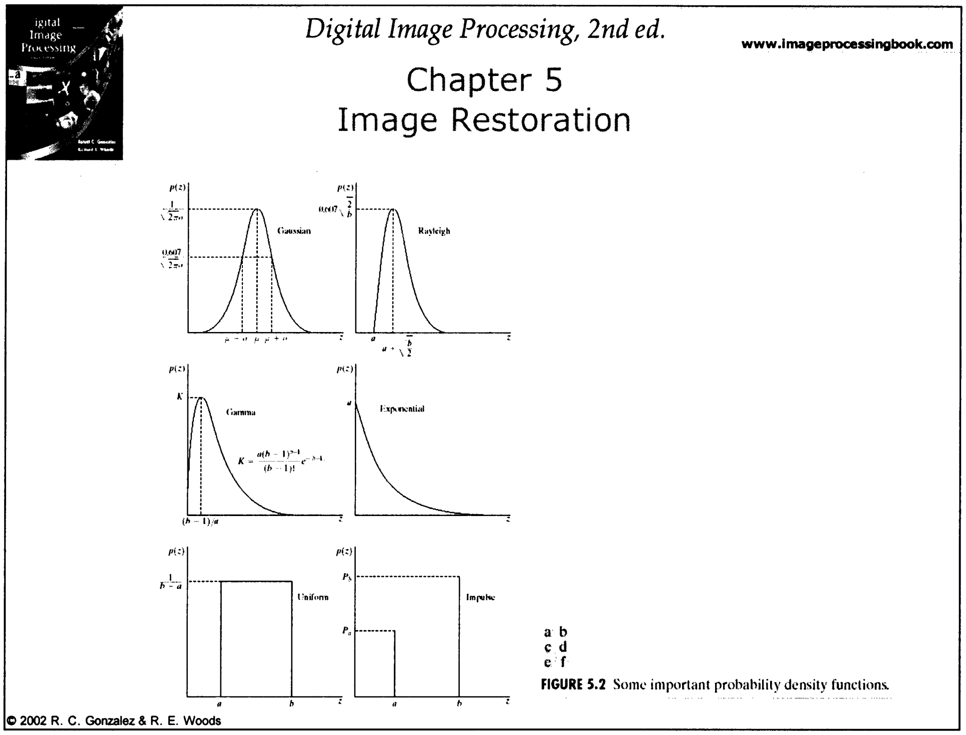
FIGURE 5.1 A model of the image degradation/restoration process.

limits us to certain types of degradation

additive noise

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

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



Gaussian

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

Rayleigh

$$p(z) = \begin{cases} \frac{2}{b}(z-a) e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}$$

used for skewed histograms

Gamma

$$p(z) = \begin{cases} \frac{a^b z^{b+1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Exponential

$$p(z) = \begin{cases} a e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Uniform (white)

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

Impulse

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & z = b \\ 0 & \text{otherwise} \end{cases}$$

Chapter 5
Image Restoration

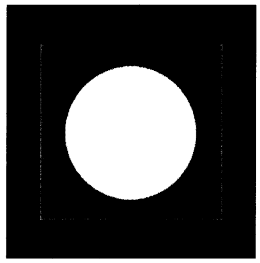
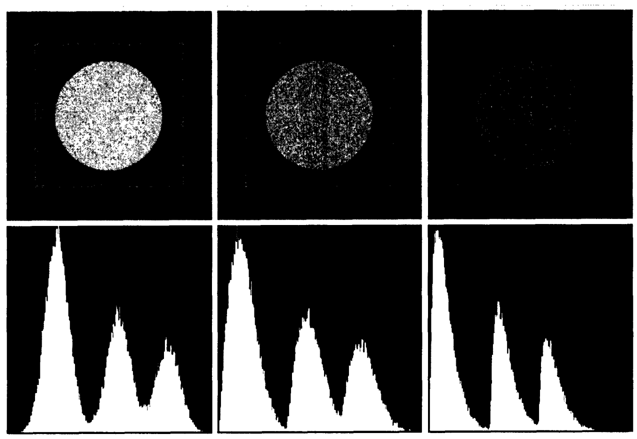


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

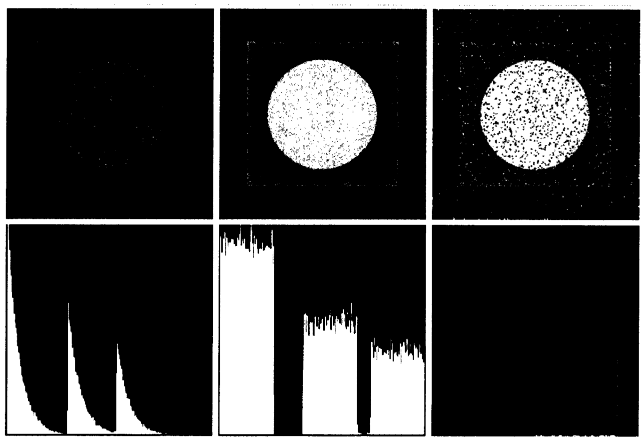


distributions in each area of image.

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Hard to visually tell (spatially) the effects of different noise sources apart.

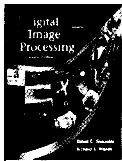
Chapter 5
Image Restoration



Exponential Uniform Salt & Pepper

g h i
j k l

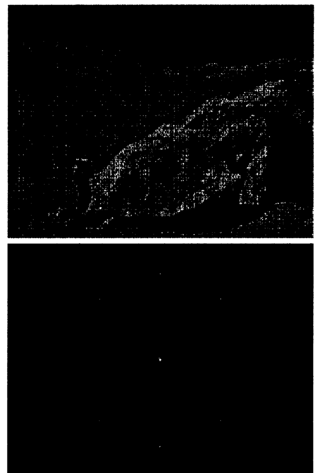
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration



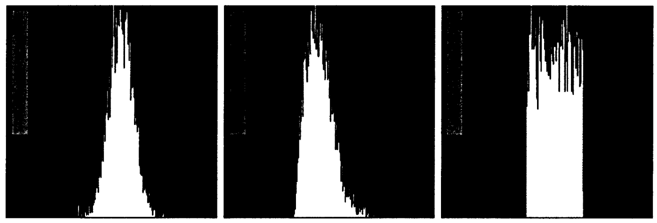
a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

note pattern of noise sources in image.

This is an example of periodic noise.

Chapter 5
Image Restoration



a b c

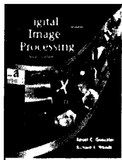
FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Gaussian Rayleigh uniform

Compute mean and variance and relate them to the distribution parameters.

$$\mu = \sum_{z_i} z_i p(z_i)$$

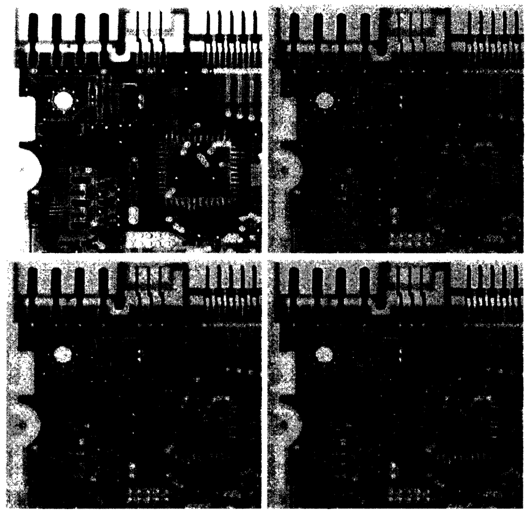
$$\sigma^2 = \sum_{z_i} (z_i - \mu)^2 p(z_i)$$



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

arithmetic mean filter

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

value of restored pixel at (x,y)

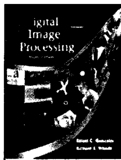
$$\frac{1}{mn}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

geometric mean filter

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}}$$

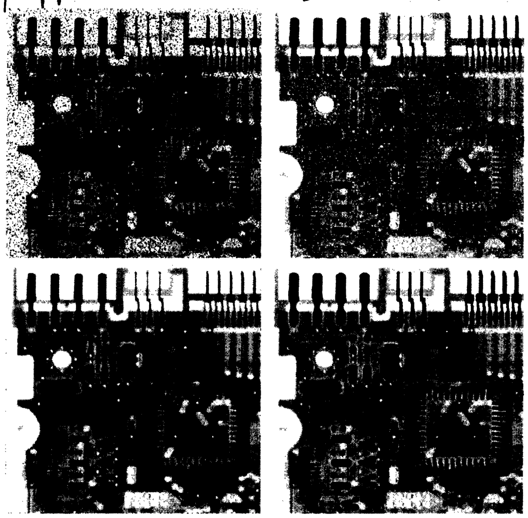
just multiply pixels in the subimage window and raise to the power $\frac{1}{mn}$

smoothing comparable to arithmetic mean filter but without losing as much image detail



Chapter 5
Image Restoration

pepper noise salt noise



contraharmonic $Q = 1.5$

contraharmonic $Q = -1.5$

a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

salt noise random 1's

pepper noise random 0's

<u>contraharmonic mean filter</u>

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

$\sum g^1 = \sum g$

reduces to arithmetic mean filter if $Q=0$

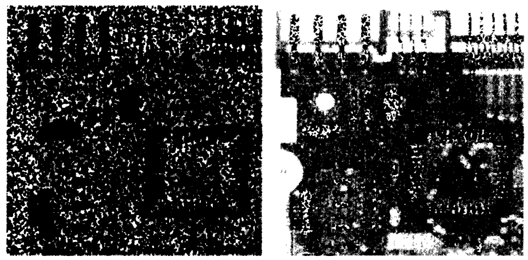
$\sum g^0 \rightarrow$ # g pixels

use $Q>0$ for eliminating pepper noise

$Q<0$ for eliminating salt noise

but cannot eliminate both simultaneously

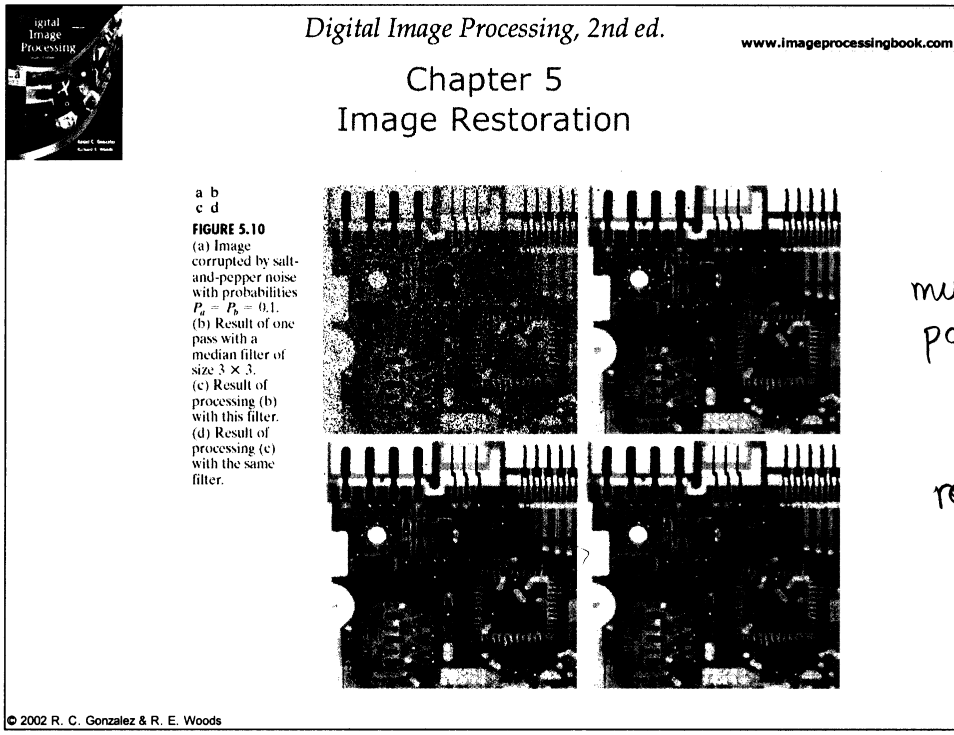
Chapter 5
Image Restoration



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

trying to filter pepper noise with wrong sign of Q	trying to filter salt noise with wrong sign of Q



multiple passes of median filter

repeated passes tend to blur

Order statistics filters – based on ordering (ranking) pixels in a neighborhood

median — $\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\}$

remember median is the middle element in the list

Other order statistics filters

max $\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\max} \{g(s,t)\}$

best for finding bright points

min $\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\min} \{g(s,t)\}$

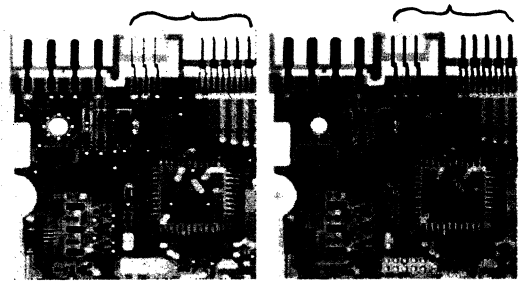
best for finding dark points

midpoint $\hat{f}(x,y) = \frac{1}{2}\left[\underset{(s,t)\in S_{xy}}{\max} \{g(s,t)\} + \underset{(s,t)\in S_{xy}}{\min} \{g(s,t)\}\right]$

works best for Gaussian or uniform noise

Chapter 5
Image Restoration

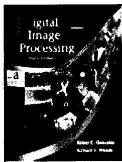
Notice difference in finger sizes



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

The min and max filters do a reasonable job on removing impulsive noise but also remove
dark pixels from the borders of dark objects
light pixels from the borders of light objects



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

additive uniform noise $\mu=0, \sigma=800$ →

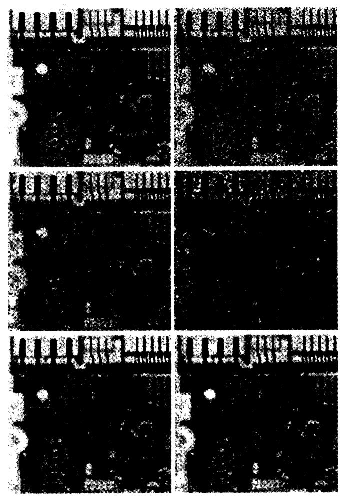
additive uniform PLUS salt-and-pepper noise

5x5 arithmetic mean filter

geometric mean filter

median

alpha-trimmed mean with d = 5
approaches median filter as d increases but also smooths.



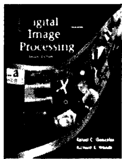
a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5 × 5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

Alpha trimmed mean filter

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

delete $\frac{d}{2}$ lowest and $\frac{d}{2}$ highest values of $g(s,t)$ giving remainder $g_r(s,t)$



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

additive Gaussian $\sigma_\eta^2 = 1000,\ \mu = 0$ →

a b
c d

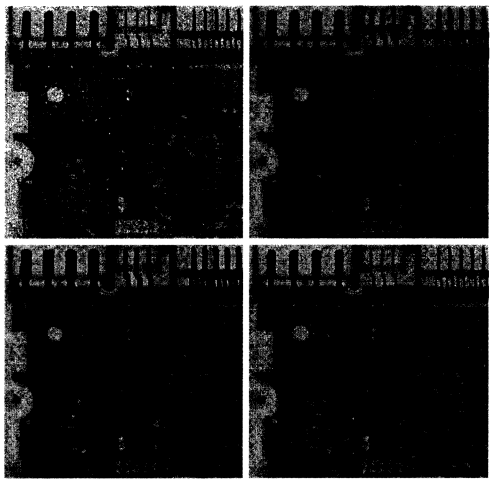


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

7×7 arithmetic mean (visible blurring)

geometric mean

adaptive noise reduction

The results are worse when you estimate σ_η^2

Adaptive noise reduction filter:

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

σ_η^2 = noise variance over the entire image (estimate)

m_L = local mean
σ_L^2 = local variance
} computed locally

The idea is that

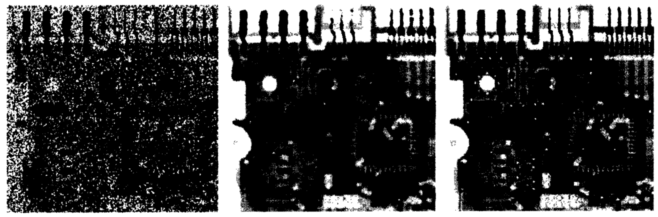
when $\sigma_\eta^2 = \sigma_L^2$ it returns the local mean, i.e., averaging out the noise

$\sigma_\eta^2 \ll \sigma_L^2$ this is probably the location of an edge and we should return the edge value, i.e., $g(x,y)$

$\sigma_\eta^2 = 0$ no noise we return $g(x,y)$

if $\sigma_\eta^2 < \sigma_L^2$ we get problems; i.e., producing negative gray levels

Chapter 5
Image Restoration



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Very high impulse noise. | 7×7 median filtering lots of loss of detail | adaptive median filtering with $S_{max} = 7$ much better detail

Adaptive median filter: (varies S_{xy})

z_{min} = min. gray value in S_{xy}

z_{max} = max. gray value in S_{xy}

z_{med} = median gray value in S_{xy}

z_{xy} = gray level value at (x,y)

S_{max} = max allowed size of S_{xy}

level A:

$A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

IF $A1 > 0$ AND $A2 < 0$ THEN go to level B

ELSE increase the window size S_{xy}.

IF window size $\le S_{max}$ repeat level A

ELSE output z_{xy}.

If $z_{max} > z_{med} > z_{min}$ then z_{med} is NOT impulsive. Go to B. Loop continues to increase S_{xy} untill z_{med} is not impulsive.

level B:

$B1 = z_{xy} - z_{min}$

$B2 = z_{xy} - z_{max}$

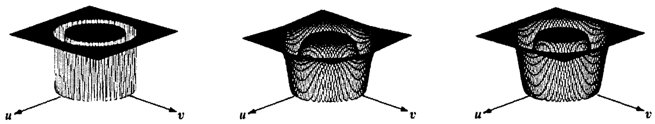
IF $B1 > 0$ AND $B2 < 0$, output z_{xy}

ELSE output z_{med}.

If $z_{max} > z_{xy} > z_{min}$ then z_{xy} is not impulsive and we output z_{xy} otherwise output the median

The idea is we want to avoid outputting impulsive outputs unless they are real. As S_{xy} increases z_{min} and z_{max} should increase to include anything but "real" noise impulses.

Chapter 5
Image Restoration



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

D_0 is center of stop band

W is full width of stop band

Ideal:

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

Butterworth

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

Gaussian

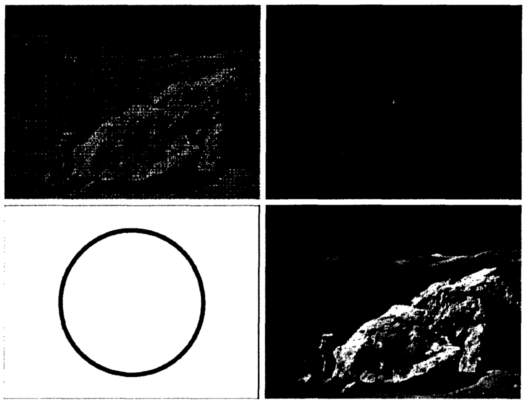
$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)\,W} \right]^2}$$

Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

notice strong frequency components in a ring



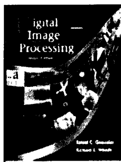
a b
c d

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.)

very impressive improvement.

You can not get such results using a spatial domain approach with small filter masks

Usually don't do band pass because it can remove too much image detail.

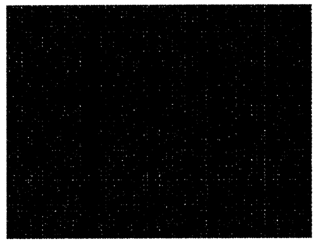


Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

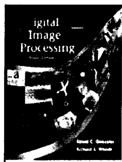
FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Bandpass is the opposite of band reject. This is the image of the noise found in 5.16(a).

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

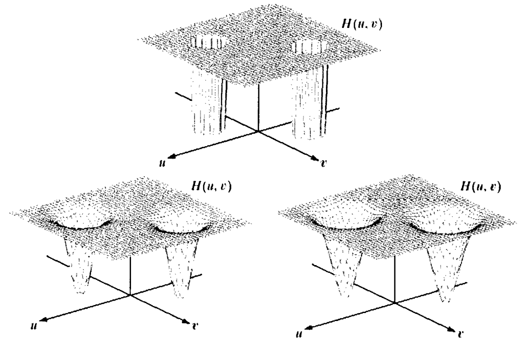
Bandpass filtering is often used to identify noise patterns.



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration



a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Notch filters:

More complicated formulas:

Ideal:

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

where $D_1(u,v) = \sqrt{\left(u - \frac{M}{2} - u_0\right)^2 + \left(v - \frac{N}{2} - v_0\right)^2}$

$$D_2(u,v) = \sqrt{\left(u - \frac{M}{2} + u_0\right)^2 + \left(v - \frac{N}{2} + v_0\right)^2}$$

Note: frequency response centered at $\frac{M}{2}, \frac{N}{2}$

Butterworth

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

Gaussian

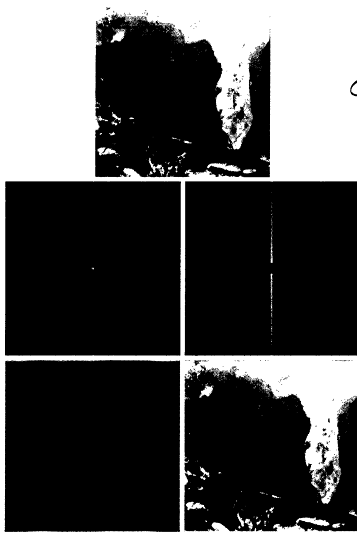
$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)\, D_2(u,v)}{D_0^2}\right]}$$

Another class of notch filters (pass) can be constructed as

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

where $H_{nr}(u,v)$ are the above notch reject filters.

Chapter 5
Image Restoration



original noise corrupted image (scan lines)

noise source not obvious in frequency domain

construct a simple notch filter and apply this filter to the frequency spectra

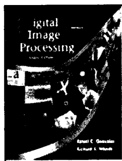
spatial image of noise resulting from applying (c) to (b)

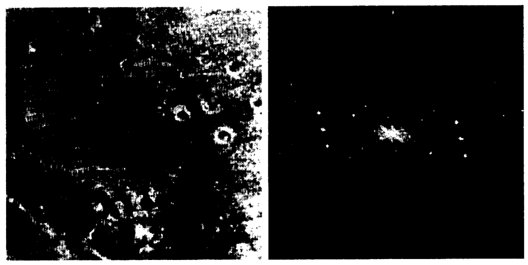
cleaned up image.

a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor 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.)

Noise in this case is very regular and caused by the scanning process.





a b
FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

This is an example of complex periodic noise.
Several noise sources are present.
These are hard to detect and filter.

To handle this we need to develop an optimum method of eliminating the noise, i.e., estimating $f(x,y)$

Consider a corrupted image with several interference components

Use a notch filter $H(u,v)$ to isolate the noise in the frequency domain

$$N(u,v) = \underbrace{H(u,v)}_{\substack{\text{construct notch filter}\\ \text{by observing spectrum } G(u,v)\\ \text{on a display}}} \underbrace{G(u,v)}_{\text{Fourier transform of corrupted image}}$$

In the spatial domain,

$$\eta(x,y) = \mathfrak{F}^{-1}\{H(u,v)\,G(u,v)\}$$

write

$$\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y) \qquad (1)$$

in principal this shoued yield the actual $f(x,y)$
in practice it is an approximation.

$\Rightarrow$ Vary $w(x,y)$ to get the "best" estimate $\hat{f}(x,y)$

One best estimate is to minimize $\sigma^2_{\hat{f}}$ over a specified neighborhood of every point (x,y) in the image.

For a neighborhood $(2a+1) \times (2b+1)$ about (x,y) we can write The variance

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \Big[\hat{f}(x+s, y+t) - \underbrace{\overline{\hat{f}}(x,y)}_{\substack{\text{average of } \hat{f} \text{ in}\\ \text{the neighborhood of } (x,y)}}\Big]^2 \qquad (2)$$

$$\overline{\hat{f}}(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{+a} \sum_{t=-b}^{b} \hat{f}(x+s, y+t)$$

Substituting (1) into (2) gives

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{+a} \sum_{t=-b}^{+b} \Big\{ g(x+s,y+t) - \omega(x+s,y+t)\eta(x+s,y+t) - \overline{g(x,y)} + \overline{\omega(x,y)\eta(x,y)} \Big\}^2$$

We assume $\omega(x,y)$ changes slowly so

$$\omega(x+s,y+t) \approx \omega(x,y)$$

and we can write

$$\overline{\omega(x,y)\eta(x,y)} = \omega(x,y)\overline{\eta(x,y)}$$

$\therefore$

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{+a} \sum_{t=-b}^{+b} \Big\{ g(x+s,y+t) - \omega(x,y)\eta(x+s,y+t) - \overline{g(x,y)} + \omega(x,y)\overline{\eta(x+s,y+t)} \Big\}^2$$

minimize by computing $\dfrac{\partial \sigma^2(x,y)}{\partial \omega(x,y)} = 0$

Extra credit if you prove this result

$$\omega(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \overline{g(x,y)}\,\overline{\eta(x,y)}}{\overline{\eta^2(x,y)} - \left[\overline{\eta(x,y)}\right]^2}$$

Compute $\omega(x,y)$ for one point in each non-overlapping neighborhood.

Chapter 5
Image Restoration

Not shifted
(0,0)

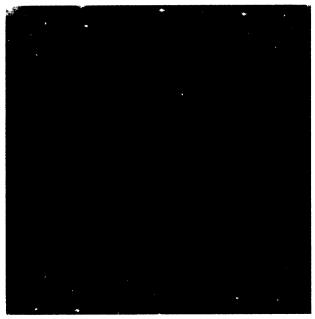
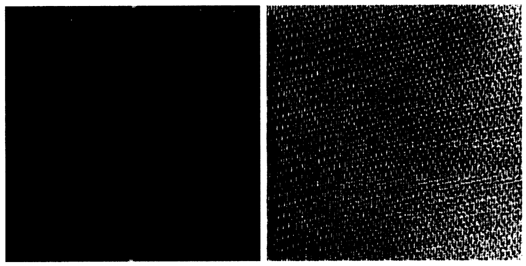


FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)

Selected neighborhood of $a=b=15$

Chapter 5
Image Restoration



a b

FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

This is the noise spectrum and the corresponding spatial noise $\eta(x, y)$ obtained by inverse transforming (a).

Chapter 5
Image Restoration

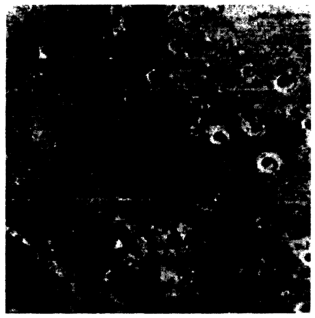


FIGURE 5.23 Processed image. (Courtesy of NASA.)

This is the image AFTER using

$$\hat{f}(x,y) = g(x,y) - w(x,y)\underbrace{\eta(x,y)}_{\text{noise from previous page}}$$

computed as

$$w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \overline{g(x,y)}\;\overline{\eta(x,y)}}{\overline{\eta^2(x,y)} - \left[\overline{\eta(x,y)}\right]^2}$$

In the presence of noise we have

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\, h(x,\alpha,y,\beta)\, d\alpha\, d\beta + \eta(x,y)$$

or if H is position invariant

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \underbrace{f(\alpha,\beta)\, h(x-\alpha,y-\beta)\, d\alpha\, d\beta}_{\text{convolution integral}} + \eta(x,y)$$

Using the book's notation

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

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

because degradations are modeled as convolution
image restoration is often called image deconvolution
the filters used in the restoration process are called deconvolution filters

Using this definition we can re-write (1) as

$$g(x,y) = H[f(x,y)] = H\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\,\delta(x-\alpha, y-\beta)\,d\alpha\, d\beta\right.$$

for $\eta(x,y) = 0$.

Assuming H is linear and using the linearity of integrals we can reverse the order to get

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} H\left[f(\alpha,\beta)\,\delta(x-\alpha, y-\beta)\right] d\alpha\, d\beta$$

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\,\underbrace{H\left[\delta(x-\alpha, y-\beta\right]}_{h(x,\alpha,y,\beta)}\, d\alpha\, d\beta$$

$h(x,\alpha,y,\beta)$ is called the impulse response

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\, h(x,\alpha,y,\beta)]\, d\alpha\, d\beta$$

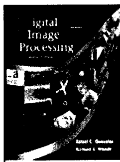
superposition (Fredholm) integral of the first kind

$\Rightarrow$ If the response to an impulse is known, the response to <u>any</u> input $f(\alpha,\beta)$ can be calculated by this equation.

If H is position invariant then

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\, h(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

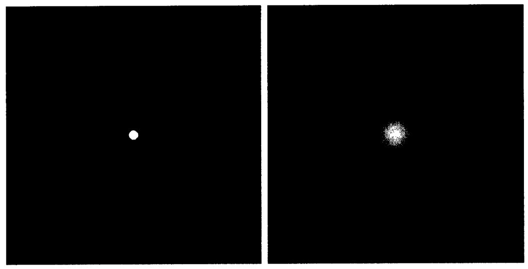
the convolution integral



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration



a b

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

observed image

fourier transform of observed image

Estimate the impulse response by imaging a bright spot of light. To minimize noise. Then

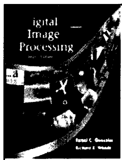
$G(u,v) = H(u,v)F(u,v) + N(u,v) \cong H(u,v)F(u,v)$

Since the fourier transform of $A\delta(x,y)$ is A we have $H(u,v) \cong \dfrac{G(u,v)}{A}$

Estimating the degradation function H from observation

$$\underbrace{H_s(u,v)}_{\text{assume position invariance and extend to complete image}} = \frac{G_s(u,v)}{\hat{F}_s(u,v)} \quad \begin{array}{l} \longleftarrow \text{observed subimage} \\ \longleftarrow \text{reconstructed subimage} \end{array}$$

reconstructed subimage: look for a subimage which has known detail, is relatively noise free, etc.



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

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.)

Image degraded by varying levels of atmospheric turbulence

Some degradation models have a physical basis
Such a model for atmospheric turbulence is

$$H(u,v) = \underbrace{e^{-k(u^2+v^2)^{5/6}}}_{\text{same form as a Gaussian Low pass filter}}$$

Modeling linear motion as an image degradation

Assume "shutter" opening and closing takes place instantaneously

Let $x_0(t), y_0(t)$ be the time varying x & y motions

For a period T of exposure

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\, dt$$

moving image

blurred image

Fourier transforming

$$G(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)\, e^{-j2\pi(ux+vy)}\, dx\,dy$$

$$G(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_0^T f[x - x_0(t), y - y_0(t)]\, dt\; e^{-j2\pi(ux+vy)}\, dx\,dy$$

$$G(u,v) = \int_0^T \left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f[x - x_0(t), y - y_0(t)]\, e^{-j2\pi(ux+vy)}\, dx\,dy\right] dt$$

$$G(u,v) = \int_0^T F(u,v)\, \underbrace{e^{-j2\pi[u x_0(t) + v y_0(t)]}}_{\text{phase shift due to shift of } f(x,y)}\, dt$$

$$G(u,v) = F(u,v) \underbrace{\int_0^T e^{-j2\pi[u x_0(t) + v y_0(t)]}\, dt}_{\text{call this } H(u,v)}$$

$$G(u,v) = H(u,v)\, F(u,v)$$

For simple linear motion $x_0(t) = \frac{at}{T}$, $y_0(t) = 0$

we can derive

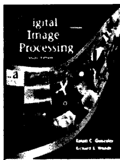
$$H(u,v) = \int_0^T e^{-j2\pi u x_0(t)} dt$$

$$H(u,v) = \int_0^T e^{-j\frac{2\pi u a t}{T}} dt$$

$$H(u,v) = \frac{T}{\pi u a} \sin(\pi u a)\, e^{-j\pi u a}$$

If we allow $y_0 = \frac{bt}{T}$ as well the degradation function becomes

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin\left[\pi(ua+vb)\right] e^{-j\pi(ua+vb)}$$



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

spatial inverse transform of degraded image $G(u,v)$

This is simulated motion degradation

688 × 688 pixel image of Gonzalez & Woods, 1/e

motion is given by

$$\left.\begin{aligned} x_0(t) &= 0.1t \\ y_0(t) &= 0.1t \end{aligned}\right\} \quad \begin{aligned} a &= b = 0.1 \\ T &= 1 \end{aligned}$$

$$H(u,v) = \frac{1}{0.1\pi(u+v)} \sin[0.1\pi(u+v)]\, e^{-j0.1\pi(u+v)}$$

5.7 Inverse Filtering

Degraded image is given by $G(u,v) = \underbrace{H(u,v)}_{\text{degradation function}} F(u,v) + \underbrace{N(u,v)}_{\text{noise}}$

Estimate $\tilde{F}(u,v)$ by simply dividing $G(u,v)$ by $H(u,v)$ $\quad \tilde{F}(u,v) = \dfrac{G(u,v)}{H(u,v)}$

Then
$$\tilde{F}(u,v) = \frac{H(u,v)\,F(u,v) + N(u,v)}{H(u,v)}$$

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

can never recover $F(u,v)$ exactly

1. $N(u,v)$ is not known since $n(x,y)$ is a random variable
2. If $H(u,v) \rightarrow 0$ then $\dfrac{N(u,v)}{H(u,v)}$ will dominate

   This can be somewhat overcome by restricting the analysis to values near the origin.

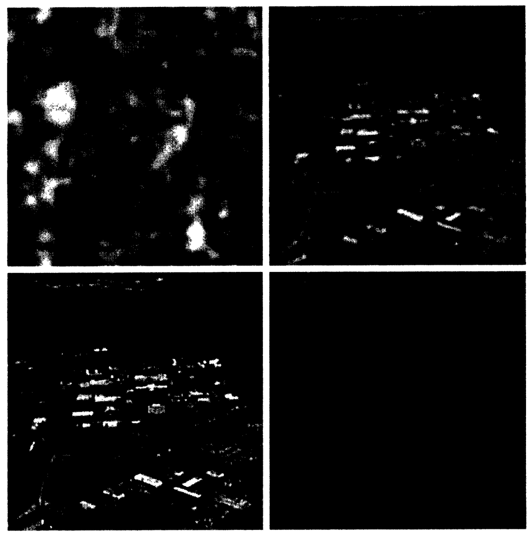
Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

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.



$D_0 = 40$

$D_0 = 70$

$D_0 = 85$

anything above this resembled (a)

This example shows the problems of small values of $H(u,v)$ in the inversion process.

For $H(u,v) = e^{-k\left[\left(u-\frac{M}{2}\right)^2+\left(v-\frac{N}{2}\right)^2\right]^{5/6}}$

This is never zero but can get small.

Using $\hat{F}(u,v) = \dfrac{G(u,v)}{H(u,v)}$ gives (a) above.

We can improve the result by cutting off values of $\dfrac{G(u,v)}{H(u,v)}$ outside a radius D_0.

The cutoff shown above was done using a Butterworth low-pass filter of order 10.

5.5 Linear, position-invariant degradations

Based upon our model degradation is modeled as an $H(x,y)$
Note: H is <u>NOT</u> in the frequency domain.

$$g(x,y) = H\left[f(x,y)\right] + \eta(x,y) \qquad (1)$$

$$\Rightarrow g(x,y) = H\left[f(x,y)\right] \qquad \text{let } \eta(x,y) = 0$$

<u>Properties</u>

<u>linear</u> if $H\left[a f_1(x,y) + b f_2(x,y)\right] = a H\left[f_1(x,y)\right] + b H\left[f_2(x,y)\right]$

This simply says that if H is a linear operator the response to a sum of two inputs is the sum of the two responses

<u>homogeneous</u> if $H\left[a f_1(x,y)\right] = a H\left[f_1(x,y)\right]$

The response to a constant multiple of the input is that same constant multiplied by the response to that input

<u>position (space) invariant</u> if $H\left[f(x-\alpha, y-\beta)\right] = g(x-\alpha, y-\beta)$
for any function $g(x,y) = H\left[f(x,y)\right]$.

This says that the response is only dependent on the value of the input and <u>NOT</u> its position

The impulse function $A\delta(x-x_0, y-y_0)$ is defined by

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} s(x,y) A\delta(x-x_0, y-y_0) = A s(x_0, y_0) \qquad \text{discrete}$$

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha,\beta)\, \delta(x-\alpha, y-\beta)\, d\alpha\, d\beta = f(x,y) \qquad \text{continuous}$$

5.8 Minimum Mean Square Error (Wiener) Filtering

To generate the best estimate $\hat{f}$ of f we minimize

$$e^2 = \underbrace{E}_{\text{expected value}}\left\{ (f-\hat{f})^2 \right\}$$

Assumptions

1. f and n are uncorrelated
2. f and/or n is zero mean
3. gray levels in $\hat{f}$ are a linear function of gray levels in f

Then the best estimate $\hat{F}(u,v)$ is given by

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

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_\eta(u,v)}{S_f(u,v)}} \right] G(u,v)$$

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

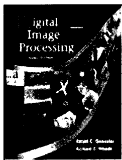
where $H(u,v)$ = degradation function

$H^*(u,v)$ = complex conjugate of $H(u,v)$

$|H(u,v)|^2 = H^*(u,v)\,H(u,v)$

$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of noise

$S_f(u,v) = |F(u,v)|^2$ = power spectrum of undegraded image



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration

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.

just computing $\dfrac{G(u,v)}{H(u,v)}$

$D_0 = 75$ radially limited $\dfrac{G(u,v)}{H(u,v)}$

Wiener filtering using interactive values of K

In practice $S_f(u,v) = |F(u,v)|^2$ of the undegraded image is not usually known.

So we simply replace $\dfrac{S_\eta(u,v)}{S_f(u,v)}$ by a constant K

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)| + K} G(u,v)$$

Digital Image Processing, 2nd ed.

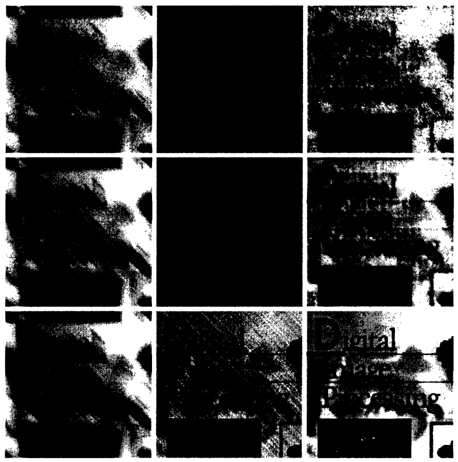
www.imageprocessingbook.com

Chapter 5
Image Restoration

$\sigma^2 = 650$

$\sigma^2 = 65$

$\sigma^2 = 0.065$



a b c
d e f
g h i

inverse filtering Wiener filtering

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Another example of Wiener filtering

Degradation = motion + noise

$x_0 = 0.1t$ Gaussian noise

$y_0 = 0.1t$ $\mu = 0$

$\sigma^2 = 650$

Inverse filtering (g) still shows a "curtain" of noise but reasonable at removing noise

5.9 Constrained Least Squares Filtering

There is an alternative to the Wiener statistical least squared error approach. It relies upon expressing the images and the degration in matrix form.

$$\underline{g} = \underline{H}\,\underline{f} + \underline{\eta} \qquad (1)$$

where

$$\underline{g} = [\; g^{(x,y)}_{row1} \quad g^{(x,y)}_{row2} \quad g^{(x,y)}_{row3} \quad \cdots \quad g^{(x,y)}_{row\,N} \;]$$

$\underline{f}$ & $\underline{\eta}$ have the same form, and dimensions $MN+1$

$\underline{H}$ has dimensions $MN \times MN$ which is VERY big

Pose the restoration as finding the minimum of $\nabla^2 f$, i.e., smoothness, constrained by (1),

$$\text{minimize} \quad C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x,y)\right]^2$$

subject to the constraint

$$\| \underline{g} - \underline{H}\,\underline{\hat{f}} \| = \| \underline{\eta} \|^2$$

where $\| \underline{\eta} \|^2 = \underline{\eta}^T \underline{\eta}$, $\hat{f}$ is the estimate of the degraded image

See Castleman [1996]

In the frequency domain

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma\, |P(u,v)|^2} \right] G(u,v)$$

$$P(u,v) = \Im\left[p(x,y)\right] \quad \text{where} \quad p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

γ is adjusted to satisfy the constraint.

Chapter 5
Image Restoration

a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

$\sigma^2 = 650$ $\sigma^2 = 65$ $\sigma^2 = 0.065$

This shows the result of processing Fig. 5.29 using constrained least squares filter with γ manually adjusted.

γ can be computed. Define

$$\underline{r} = \underline{g} - \underline{H}\,\underline{f}$$

We want to find γ such that

$$\|\underline{r}\|^2 = \|\underline{\eta}\|^2 \pm a$$

$\uparrow$ accuracy factor

It can be shown that $\|\underline{r}\|^2$ is a monotonically increasing function of γ

Find γ by

1. Specifying an initial value of γ
2. Compute $\|\underline{r}\|^2$
3. Stop if $\|\underline{r}\|^2 = \|\underline{\eta}\|^2 \pm a$. Otherwise

[increase γ if $\|\underline{r}\|^2 < \|\underline{\eta}\|^2 + a$
[decrease γ if $\|\underline{r}\|^2 > \|\underline{\eta}\|^2 + a$

Recompute $\hat{F}(u,v)$ using this new value of γ

Go to 2.

$$\sigma_n^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\eta(x,y) - m_\eta\right]^2$$

where $m_\eta = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \eta(x,y)$

But $\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\eta(x,y) - m_\eta\right]^2 = \|\eta\|^2$

$$\Rightarrow \|\eta\|^2 = MN\left[\sigma_\eta^2 - m_\eta\right]$$

so we only need the mean and variance of the noise to compute $\|\eta\|^2$

Chapter 5
Image Restoration

a b

FIGURE 5.31
(a) Iteratively determined constrained least squares restoration of Fig. 5.16(b), using correct noise parameters. (b) Result obtained with wrong noise parameters.

Result of restoring image using

γ based on correct noise parameters

γ based on incorrect noise parameters

Chapter 5
Image Restoration

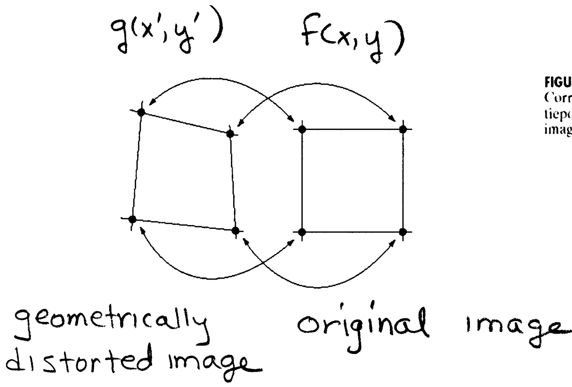


FIGURE 5.32 Corresponding tiepoints in two image segments.

$$x' = r(x,y) = c_1 x + c_2 y + c_3 xy + c_4$$

$$y' = s(x,y) = c_5 x + c_6 y + c_7 xy + c_8$$

for all points in the distorted rectangle

restored image is $\hat{f}(x_0, y_0) = g(x_0', y_0')$ $\quad x_0, y_0$ undistorted point

need gray scale transform

geometric transform

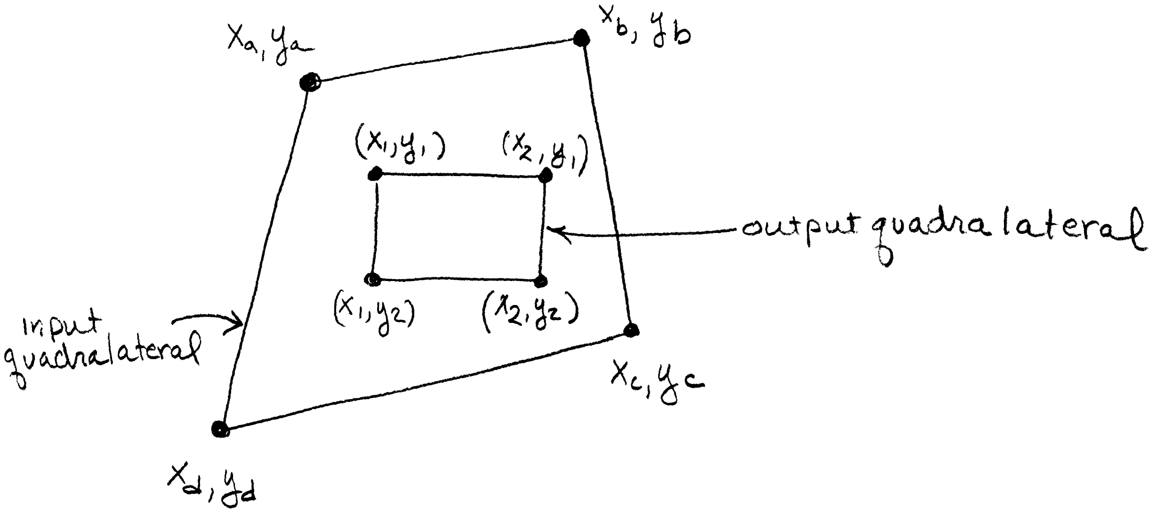
value of point in undistorted image

How do we get transform?

Find four control points in each image. $x,y \rightarrow x',y'$

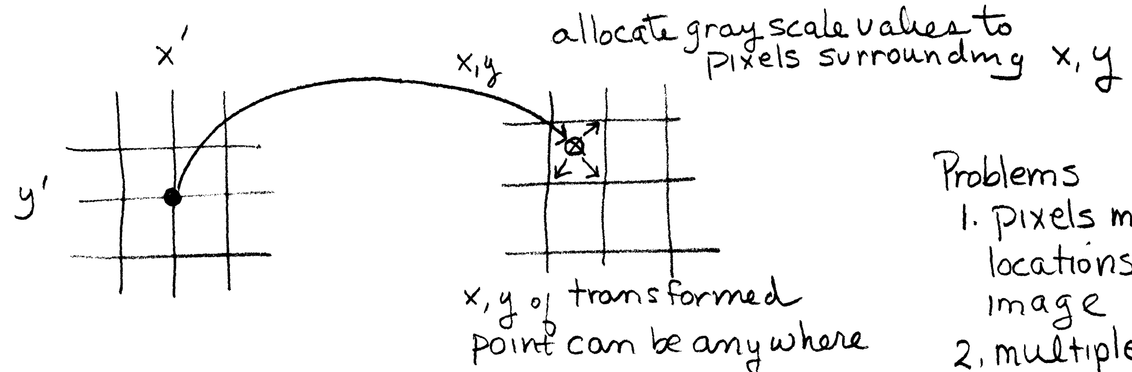
This gives eight equations in eight unknowns.

Solve for $c_1, \dots, c_8$



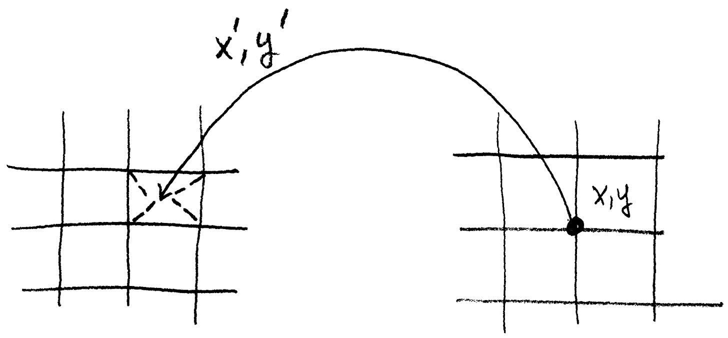
In general, $G(x,y) = F(x', y') = F(ax + by + cxy + d,\ ex + fy + gxy + h)$

You can do forward mapping (pixel carry over) or backward mapping (pixel filling)



pixel-filling (forward) approach

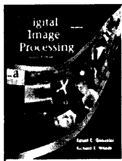
Problems
1. pixels mapping to locations outside image
2. multiple addressing of output pixels
3. missing output pixels



Each output pixel is determined.

Find gray scale value by interpolation.

Interpolation: nearest neighbor
bi-linear interpolation



Chapter 5
Image Restoration

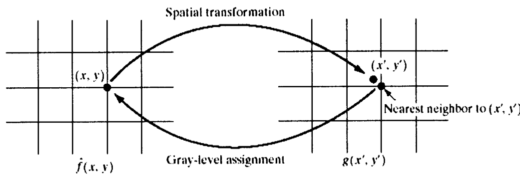
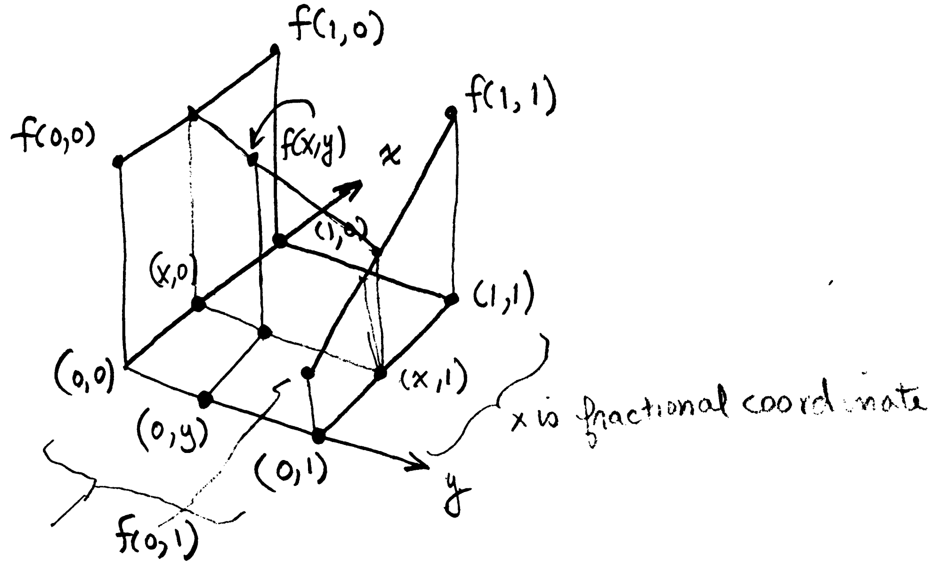


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

Most common gray scale interpolation

Bi-linear interpolation is more accurate.

8.2.2. Bilinear interpolation



can't fit plane through four points

fit hyperbolic paraboloid $\quad f(x,y) = ax + by + cxy + d$

fit to values at each corner by simple algorithm

1) linearly interpolate between upper two points

$$f(x,0) = f(0,0) + x\left[f(1,0) - f(0,0)\right] \qquad (1)$$

2) linearly interpolate between lower two points

$$f(x,1) = f(0,1) + x\left[f(1,1) - f(1,0)\right] \qquad (2)$$

3) interpolate vertically

$$f(x,y) = f(x,0) + y\left[f(x,1) - f(x,0)\right], \qquad (3)$$

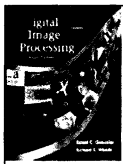
Combine all 3 equations

$$f(x,y) = \left[f(1,0) - f(0,0)\right]x + \left[f(0,1) - f(0,0)\right]y + \left[f(1,1) + f(0,0) - f(0,1) - f(1,0)\right]xy + f(0,0)$$

5 additions
+4 multiplications
+3 additions

8 additions plus 4 ~~additions~~ multiplications efficient

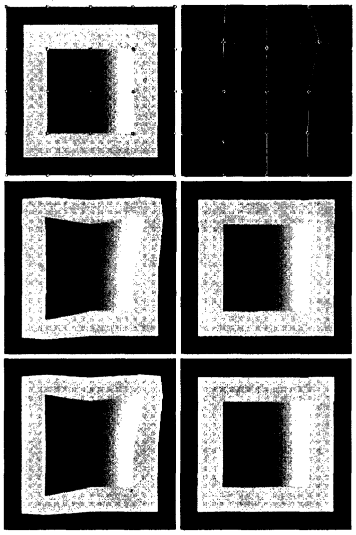
surfaces produced by bilinear interpolation match in amplitude at neighborhood boundaries, but do not match in slope, $\Rightarrow$ generated surface is continuous but derivatives are discontinuous at boundaries



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 5
Image Restoration



geometric distortion

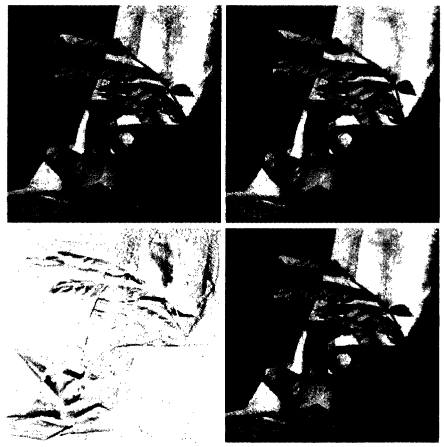
corrected using
nearest neighbor gray scale interpolation

corrected using
bilinear gray scale interpolation

a b
c d
e f

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.

Chapter 5
Image Restoration



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.

Geometric distortion is less noticible in complex images.

(b) is distorted the same as in previous figure

(c) difference image showing distortion

(d) geometrically corrected image.

8.3 Spatial Transformations (Castleman)

$$g(x,y) = f(x',y') = f\left[a(x,y),\ b(x,y)\right]$$

if $a(x,y) = x \quad b(x,y) = y \quad \Rightarrow$ Identity operator

if $a(x,y) = x + x_0 \quad b(x,y) = y + y_0$

this transformation translates (x_0, y_0) to origin by amount $\sqrt{x_0^2 + y_0^2}$

This can be written in matrix form as

$$\text{new coordinates}\left\{\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix}\right. = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \left.\begin{aligned} a(x,y) &= x + x_0 \\ b(x,y) &= y + y_0 \end{aligned}\right\}\text{translation}$$

$\leftarrow$ extra coordinate

<u>homogeneous coordinates</u>

$$\text{new coordinates}\left\{\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix}\right. = \begin{bmatrix} \frac{1}{c} & 0 & 0 \\ 0 & \frac{1}{d} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \left.\begin{aligned} a(x,y) &= \frac{x}{c} \\ b(x,y) &= \frac{y}{d} \end{aligned}\right\}\text{scaling}$$

in general we do this in 3-D

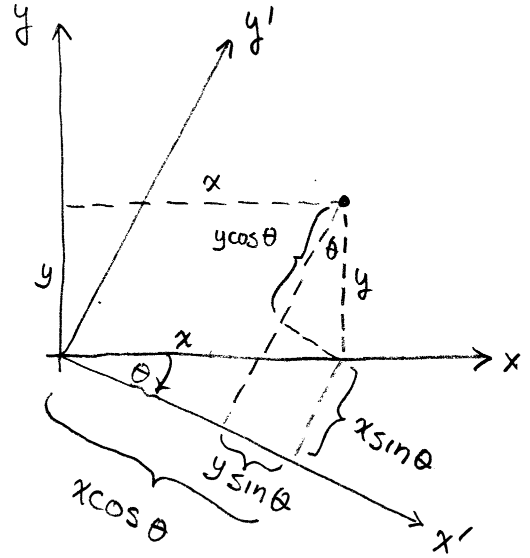
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & u \\ 0 & 0 & 1 & v \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + T \\ y + u \\ z + v \\ 1 \end{bmatrix}\Bigg\}\ \text{this is a general translation}$$

The general form of the transformation

$$\left[\begin{array}{c|c} \text{rotation} & \leftarrow \text{translation} \\ \hline \text{perspective} & \leftarrow \text{scale} \end{array}\right]$$

Rotations about the origin are of this form

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$



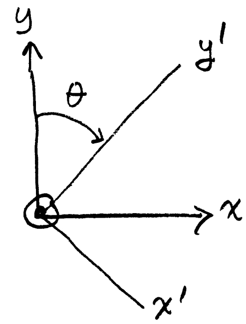
This notation is opposite that normally found in the robotics community, i.e., this would be a rotation in $-\theta$ about the z-axis in robotics.

Homogeneous coordinates allow for transformations about the x and y axes although we rarely need them in basic image processing.

The full three-dimensional rotations are:

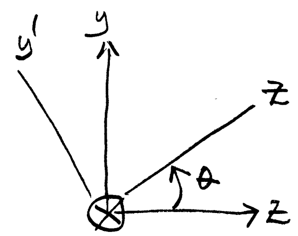
about the z-axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



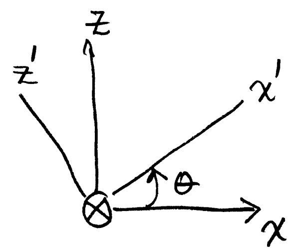
about the x-axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



about the y-axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & +\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

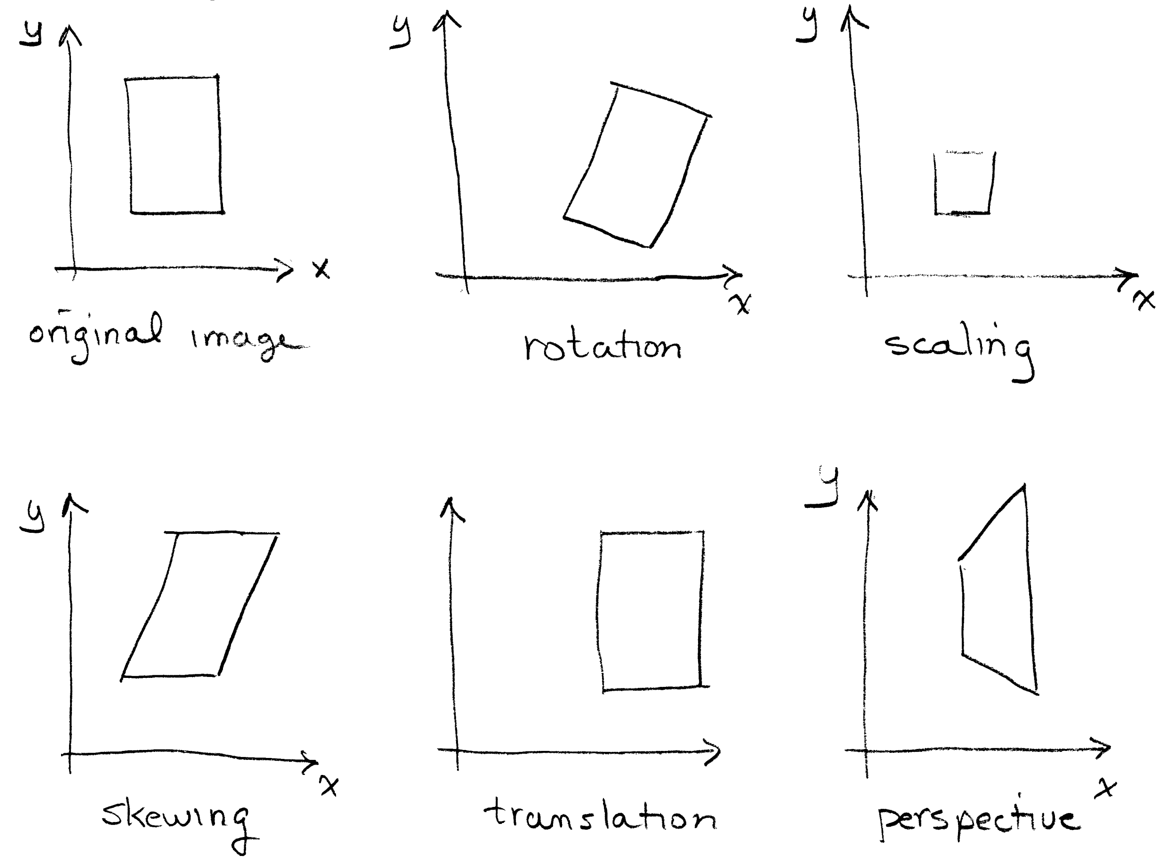


Note: these are left-handed rotations as compared to robotics.

The z-dimension has also been added for full 3-D operations.

Reference: Balland and Brown, Ch. 2 p. 17-22 optics
p. 467 homogeneous coords,
p. 477 geometric transformations

different types of effects can be achieved using homogeneous coordinate transformations.

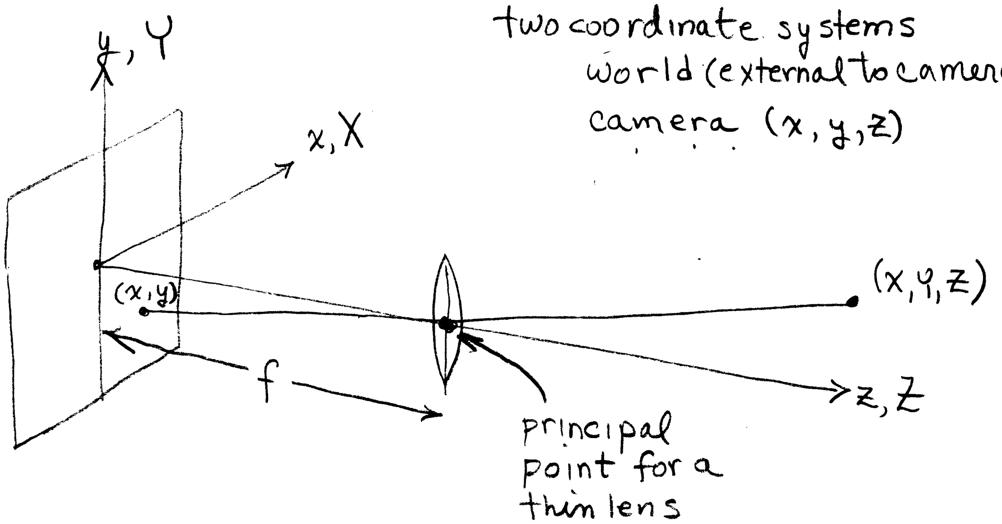


We have seen all of these but the perspective transformation.

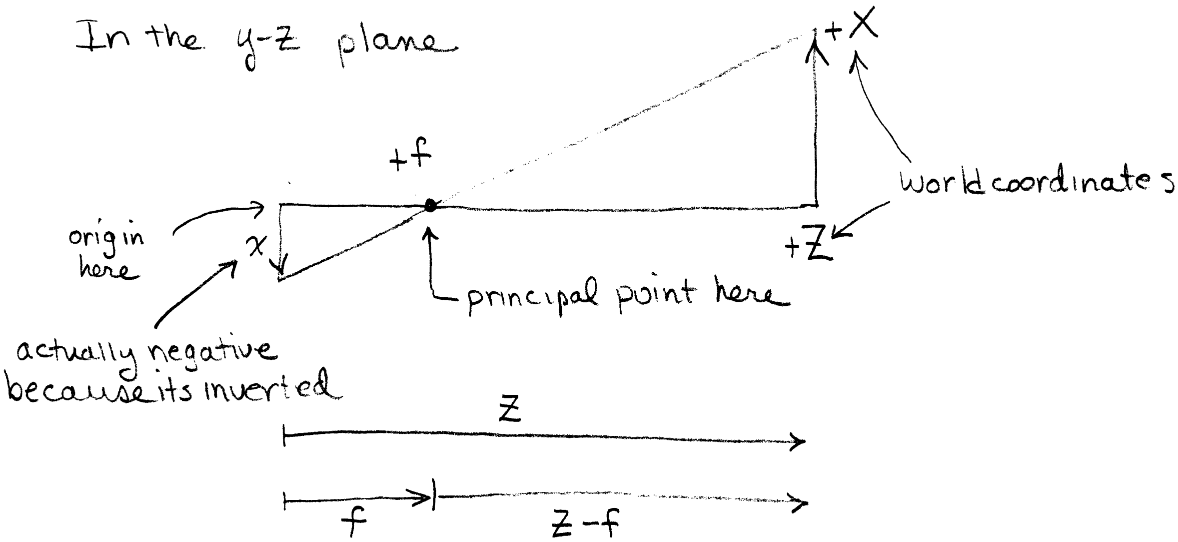
A perspective transformation is <u>exactly</u> an imaging transformation such as that obtained with a lens.

How does a lens work?

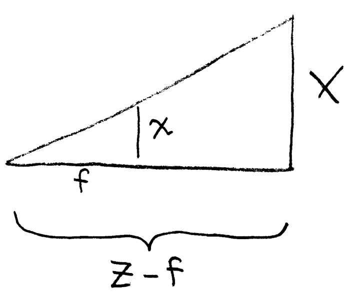
two coordinate systems
- world (external to camera) X, Y, Z
- camera (x, y, z)



In the y-z plane



Form a single triangle



Using similar triangles

$$\frac{X}{Z-f} = \frac{-x}{f}$$

minus sign accounts for inversion

re-arranging

$$\frac{x}{f} = -\frac{X}{Z-f} = \frac{X}{f-Z}$$

similarly

$$\frac{y}{f} = -\frac{Y}{Z-f} = \frac{Y}{f-Z}$$

The focal plane coordinates are then

$$x = \frac{f}{f-Z} X$$

$$y = \frac{f}{f-Z} Y$$

This is called the perspective transformation, i.e.,

$$x' = \frac{f}{f-Z} X = \frac{X}{1-\frac{Z}{f}}$$

Note: mapping a 3-D scene onto the image plane is a many-to-one transformation.

The parametric equation of the line given by image plane coordinates (x_0, y_0) can be written by rearranging

$$x_0 = \frac{f}{f-Z} X$$

as

$$X = \frac{f x_0 - x_0 Z}{f} = x_0 - \frac{x_0}{f} Z$$

Similarly $$Y = y_0 - \frac{y_0}{f} Z$$

You <u>cannot</u> recover 3-D point information by means of this inverse perspective unless you know at least one of the word coordinates of the point, or use more than one camera

The perspective transformation is <u>exactly</u> an imaging transform and can be written as a homogeneous coordinate transformation.

For example,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1-\frac{z}{f} \end{bmatrix}$$

What this last element does is scale all the other elements by $1-\frac{z}{f}$.

Note that you can have perspective transforms in z but this makes no physical sense.

Homogeneous coordinates make it easy to do compound transformations

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{translate} \\ \text{origin to } (x_0, y_0)}} \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ +\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{rotate } \theta \\ \text{about } (x_0, y_0)}} \underbrace{\begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{translate} \\ \text{origin back} \\ \text{to original origin}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\longrightarrow$

sequence of operations