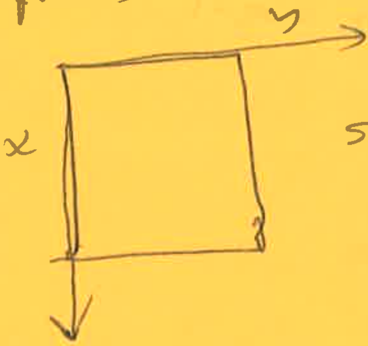
3/1

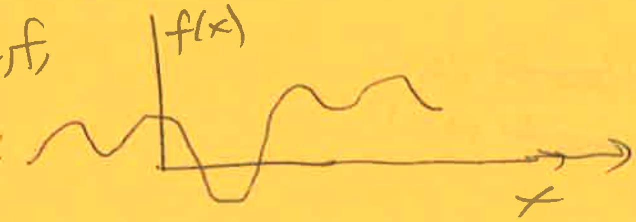
Spatial Filtering (chapter 3)

spatial domain:



space

consider a 1D function f, in usual layout:



can represent it as a sum of cosine functions

equals

basis functions

can say that f is the weighted sum of a set cosine functions at different frequencies

frequency domain

Can use other basis functions, e.g., polynomials

intensity transformation

change individual gray levels

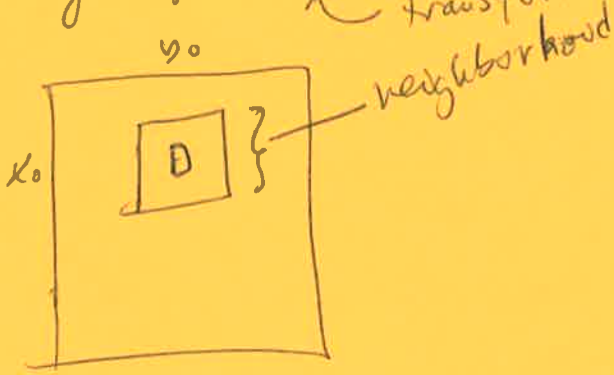
spatial transformation

change gray level ~~as~~ a function of its neighborhood primary example: enhancement

more suitable image

$$g(x,y) = T(f(x,y))$$

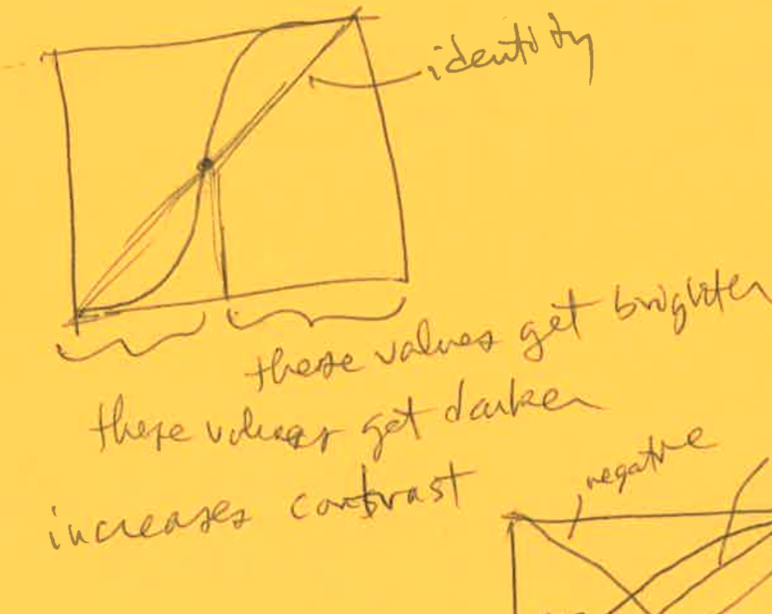
transform



<u>intensity transform</u> neighborhood is 1×1

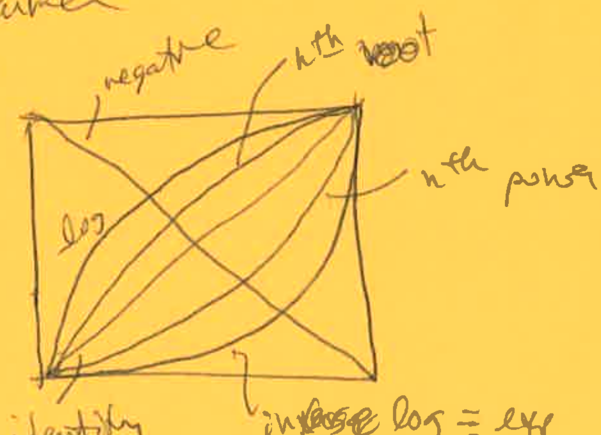
$$s = T(r)$$

transform shape



threshold function

increases contrast



Negatives if $f(x,y) \in [0, L-1]$

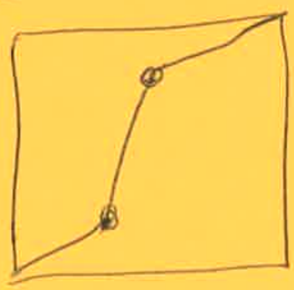
$s = L-1-r$ enhance hard to see things

Log $s = c \log(1+r)$ expands visible pixels (reduces dynamic range)

Power-Law $s = c r^{\gamma}$ gamma correction for display

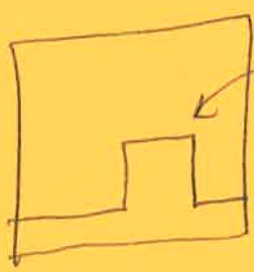
see examples in text to bring out details

Piecewise linear



e.g., can use all gray levels more effectively

Intensity-level Slicing



only shows these

leaves all the same except these

Bit-plane slicing lower order bits may have no info.

Histogram Processing Let r_k $k=0, \ldots L-1$, denote intensities of f

unnormalized histogram: $h(r_k) = n_k$ n_k = # pixels with intensity r_k

normalize $p(r_k) = \dfrac{n_k}{MN}$

good images tend to have uniform distribution of gray levels

Histogram equalization

Given 2 gray level distributions

$p_r(r)$ + $p_s(s)$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$ e.g., if $p_r \equiv p_s$ then easy to see

A useful transform:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

Have then

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$
$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$
$$= (L-1)\,p_r(r)$$

So use $p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$

$$= p_r(r)\frac{1}{(L-1)\,p_r(r)}$$
$$= \frac{1}{(L-1)}$$ uniform distribution

$\Rightarrow$ equal histogram

Discrete form $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$

p.152

p.156 Histogram Matching

consider discrete case pp ~~158~~- 159

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = \cancel{s_k}$$

$= s_k$ where $p_z(z_i)$ is ith value of specified histogram

$$z_q = G^{-1}(s_k)$$

Alg. 1. Compute $p_r(r)$ for input image + get s_k

2. Compute $G(z_q)$

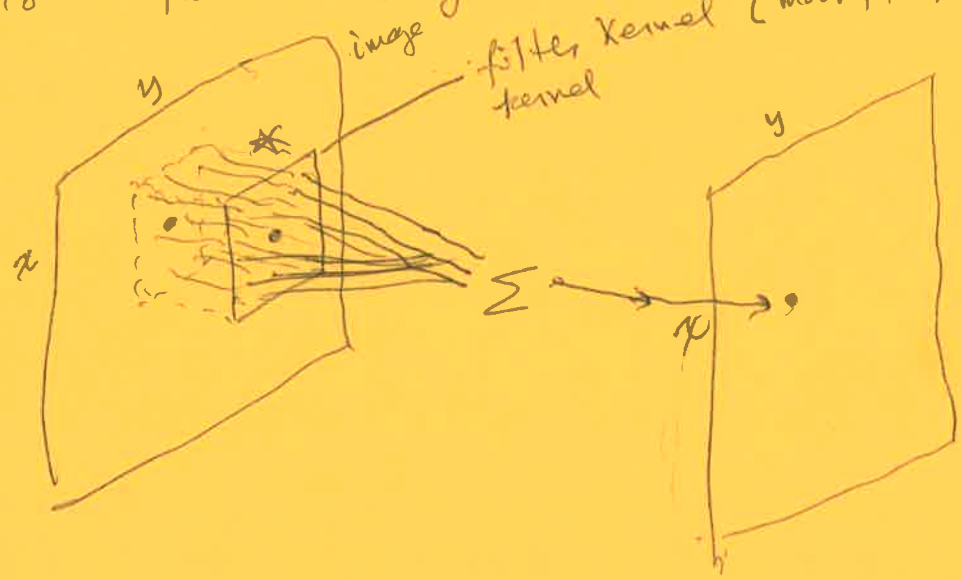
3. $\forall s_k$ use stored values of G to find corresponding value of z_q so $G(z_q)$ is closest to s_k

4. Form histogram-specified image by mapping every equalized pixel with value s_k to the corresponding value z_q using mappings from step 3.

---

p. 178 Spatial Filtering

filter kernel (mask, template, window)



Spatial correlation:

2D $$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

1D $$g(x) = \sum_{s=-a}^{a} w(s)\, f(x+s)$$

3/6

See p. 180

pad $\quad$ f $\quad$ pad $\qquad$ w $\qquad$ convolution

00 | 0 0 0 1 0 0 0 0 | 0 0 $\qquad$ 1 2 4 2 8 $\qquad$ ~~w~~ 8 2 4 2 1

1 2 4 2 8

  1 2 4 2 8 ...

  ↓ ↓

  0 8

rotate 180

similar ~~same~~ for 2D

discrete impulse of amplitude A

$$\delta(x-x_0, y-y_0) = \begin{cases} A & x=x_0 \quad y=y_0 \\ 0 & \text{otherwise} \end{cases}$$

correlation

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

convolution

$$(w \bigstar f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

if convolve every pixel of kernel, then resulting image is

$$S_v \times S_h = (M+m-1) \times (N+n-1)$$

p. 185 separable kernels $\qquad$ get 2D from 2 1D's

$$G(x,y) = G_1(x) G_2(y)$$

e.g. $\underbrace{w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{m \times n} = \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{m \times 1} \underbrace{\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}}_{(n \times 1)^T}$

2D filter ⇒ $MNmn$ $\quad$ ×'s & +'s

separable ⇒ $MNm + MNn$

consider $C = \dfrac{MNmn}{MN(m+n)} = \dfrac{mn}{m+n}$ $\quad$ e.g., 11×11 ⇒ 5.2 times more p.186 (3|7)

If the rank of w is 1, then:

1. Find any non-zero value in kernel (call it E at $w(r,c)$)
2. Form $\bar{c}$ & $\bar{r}$

$$\bar{c} = w(:,c) \qquad \bar{r} = w(r,:)$$

3. Let $\bar{v} = \bar{c}$ & $\tilde{w}^T = \bar{r}/E$

Some frequency domain ideas

1. convolution in spatial domain is multiplication in freq. domain
2. An impulse of amplitude A in spatial domain is constant A in freq. domain

smoothing (averaging) reduce noise
p.188

Box kernels

$$w = \frac{1}{mn}\begin{array}{c|c} & \begin{array}{ccccc} & & n & & \end{array} \\ m & \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & & & & \vdots \\ 1 & & & & \\ 1 & 1 & \cdots & & 1 \end{bmatrix} \end{array}$$

p.190 Gaussian filter

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}} = Ke^{-\frac{r^2}{2\sigma^2}}$$

where $r = \sqrt{s^2+t^2}$

p.192 3x3

$$\frac{1}{4.8976}\begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

circularly symmetric

? sum of

3/8

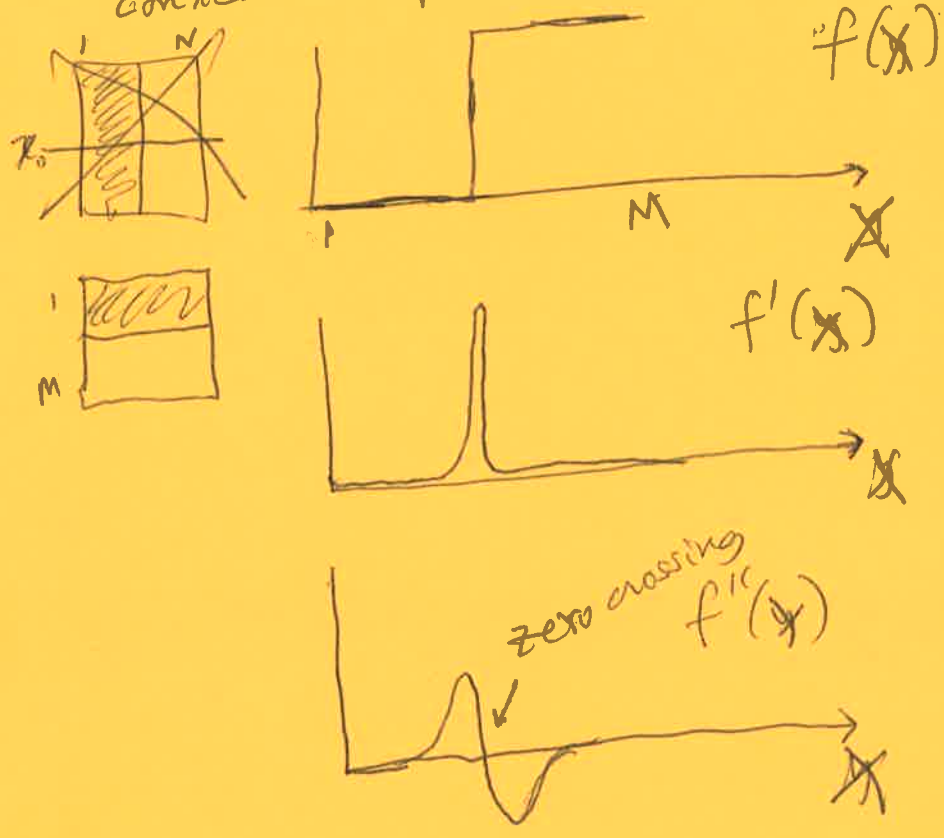
pp. 198-199 order statistics

median
min
max

Sharpening Highpass filters

Consider the profile of an edge

differentiation emphasizes discontinuities



estimate (approx): 1st deriv.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

filter $\boxed{-1 \mid 1}$

2nd deriv:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

$\boxed{1 \mid -2 \mid 1}$

f: | | | | |

f': $f(2)-f(1)$ | $f(3)-f(2)$ | $f(4)-f(3)$

f'': $\big(f(3)-f(2)\big)-\big(f(2)-f(1)\big)$

$= f(3)-2f(2)+f(1)$

pp. 202-204 Laplacian : isotropic derivative operator

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

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

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

filter:

	1	
1	-4	1
	1	

im

imG 71, 25

[dx, dy] = gradient(im)

[dxx,

[dyx

L = dxx + dyy

imG − L

or

1	1	1
1	-8	1
1	1	1

To emphasize high-frequency content:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

$c = -1$ for negative center of filter

3/20

p.206 Highboost filtering

1. Blur image
2. Set mask = image − blurred image
3. hb = image + mask

imG : image

imGG blur $H = 71, 25$ (N, σ)

mask = imG − imGG

N = imG + 10 * mask

p.208 Gradient

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

matlab

gradient(f)

Edge (noise) detector:

```
[dx, dy] = gradient(im)
Mag = sqrt(dx.^2 + dy.^2);
ori = atan2(dy, dx)
ori = posori(ori)
```

atan2: $[-\frac{\pi}{2}, \frac{\pi}{2}]$

posori: $[0, 2\pi]$

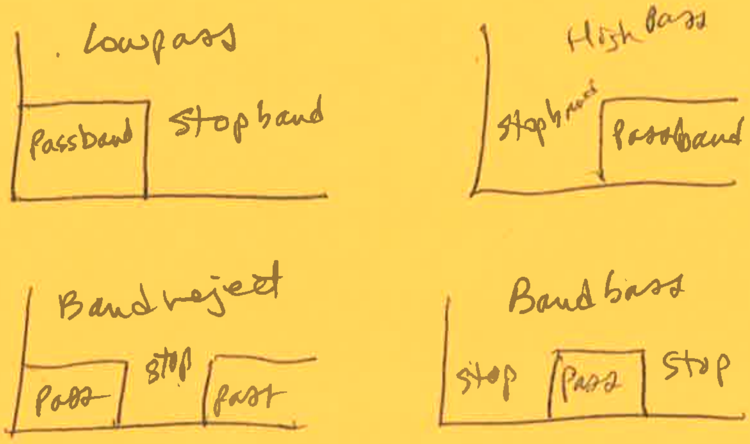
image 3.57b

says: Sobel gradient

but gradient is 2D at each pixel

3\11

p. 212 Lowpass Filters → others



lowpass $lp(x,y)$

highpass $\delta(x,y) - lp(x,y)$

band reject $lp_1(x,y) + hp_2(x,y)$

bandpass $\delta(x,y) - br(x,y)$

test image : zone plate

create_im('cs4640_zone_plate', 597, 597, 256, -8.2, 8.2, -8.2, 8.2)

Fig. 3.60 approximate with

```
S = sinc([-4:0.1:4])/10;
BP = S'*S;
surf(BP)
zp1 = conv2(zp, S1);
zp2 = conv2(zp1, S1');
show zp2
```

```
BP3 = 0.1*BP2/256 - 0.02;
zp3 = conv2(zp, BP3);
```

shift to -0.02 + scale to 1

use cs4640_sinc