Introduction

Wednesday, January 9, 2019 7:57 AM

Why in EE?

Why images?

images	words
sight	hearing
parallel	serial
spatial	?
universal	language-specific
more info	
vague	specific
concrete	abstract

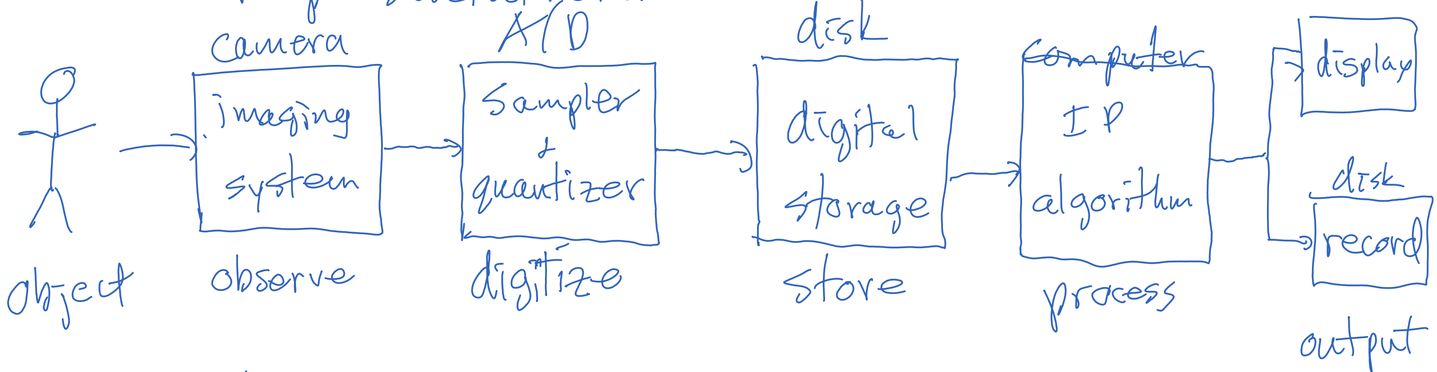
What is an image?

- luminance of a scene (standard picture)
- absorption characteristics of an x-ray
- radar cross-section of a target
- temperature profile of a scene (infrared)

general ⟹ any 2-D function that contains information

What is digital image processing?

- using a computer to process 2-D data. Generally, the input is an image, and the output is another image or image representation.



typical image processing sequence

Image representation

- continuous vs. discrete
- digital processing requires sampling + quantization
  - must sample finely enough to preserve relevant information
  - must quantize so that quantization noise (roundoff error) is acceptable
- an image sample is called a picture element (or pixel or pel). An image is represented as a large array of #'s.

- a monochrome image is usually quantized to 256 gray levels, or 8 bits.
- a color (red, green, blue) image is 8 bits/color.

Applications

enhancement — accentuating certain features or improving the look of the image for analysis or display

- edge sharpening
- contrast enhancement
- smoothing
- color manipulation

geometric transformations — changing coordinate system or warping image

restoration — removal or minimization of known degradations

Ch. 2.1 - 2.2

correcting sensor nonlinearities

- correcting sensor nonlinearities
- removing blurring effects of optical systems
- correcting geometric distortion

⇒ images contain an enormous amount of information

- reduction in storage (consumer photos, databases, medical records, etc.)

video, video chat,

Friday, January 25, 2019 2:19 PM

Two-dimensional signals & systems

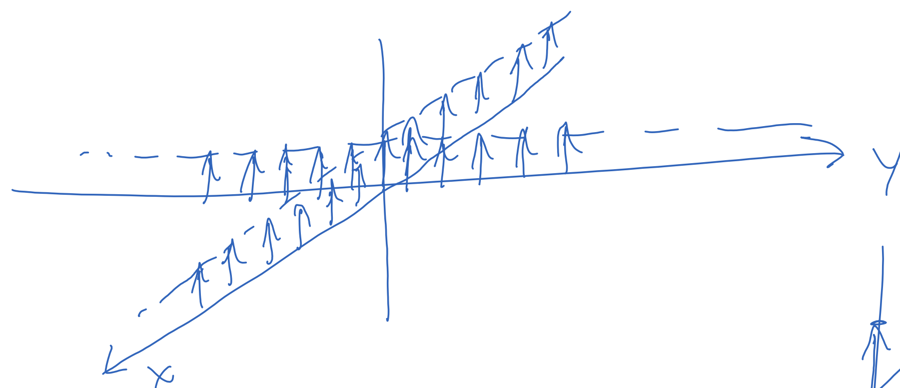
Continuous signals & systems: basic signals

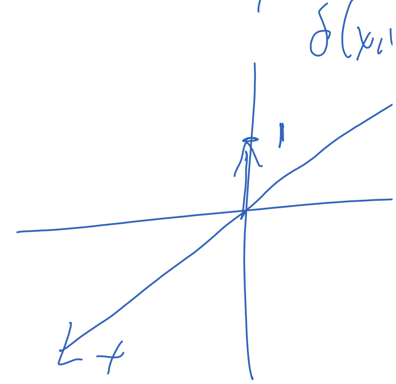
Dirac delta: $\delta(x,y) = 0 \,, \quad x, y \neq 0$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x-x_0, y-y_0)\,dx\,dy = f(x_0, y$$

(sifting property)

$$\delta(x,y) = \delta(x)\,\delta(y)$$





step function:

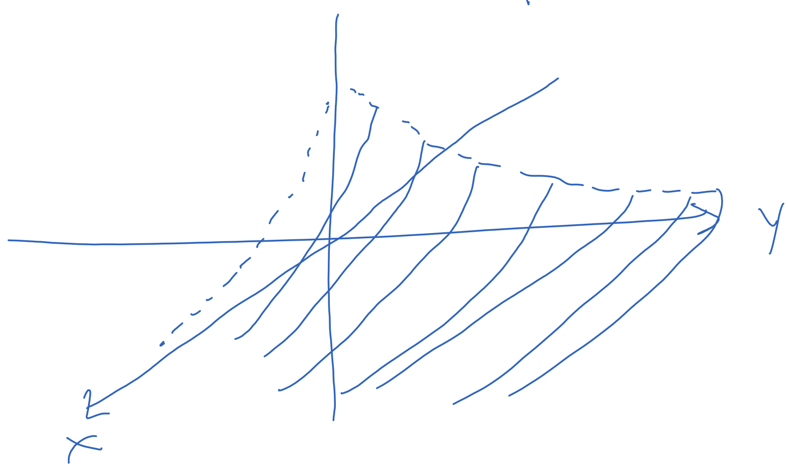
$$u(x,y) = \begin{cases} 1 \,, & x, \\ 0 \,, & \text{ot} \end{cases}$$

$= u(x)\,u(y)$

decaying exponential:

$$f(x,y) = \begin{cases} \exp\{-\alpha x - \beta y \\ 0 \end{cases}$$

$\alpha, \beta > 0$ $\doteq \{ e$



Sinusoid:

$$f(x,y) = \sin(\omega_{0x} x + \omega_{0y} y)$$

ω_{0x} = frequency along x in

ω_{0y} = " " y i

A <u>separable</u> signal is one that

factored into a product of 1-D s

$$\delta(x,y) = \delta(x)\,\delta(y)$$

$$u(x,y) = u(x)\,u(y)]$$

$$\exp\{-\alpha x - \beta y\}\,u(x,y) = [e^{-\alpha x}u(x)][[$$

$$e^{j(\omega_{0x}x + \omega_{0y}y)} = [e^{j\omega_{0x}x}$$

Linear systems

If $T[\cdot]$ is a system, it is linear if

$$T[af(x,y) + bg(x,y)] = aT[f(x,y)] + bT[g(x,y)]$$

for all $a, b, f(x,y), g(x,y)$

– superposition

Ex: Film with $T[f(x,y)] = g(x,y) = \log(1+I)$

– not linear

$$T[(1)(1)+(1)(1)] = \log(1+2) = \log 3$$

$$(1)T[1] + (1)T[1] = 2\log(1+1) = \log 4$$

$\neq$

$\Rightarrow$ not linear

Ex: optical system that blurs horizontally

$$T[f(x,y)] = \int_0^1 f(x-x', y)\,dx'$$

show that system $\int_0^1$ linear

Must show that superposition holds for all inputs.

$$T[af(x,y) + b\,g(x,y)] = \quad af(x-x',y) + b\,g(x-x',y)$$

$$= a\int_0^1 f(x-x',y)\,dx' + b\int_0^1 g(x-x',y)\,dx'$$

$$= a\,T[f(x,y)] + b\,T[g(x,y)]$$

$\Rightarrow$ linear

shift-invariant systems

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

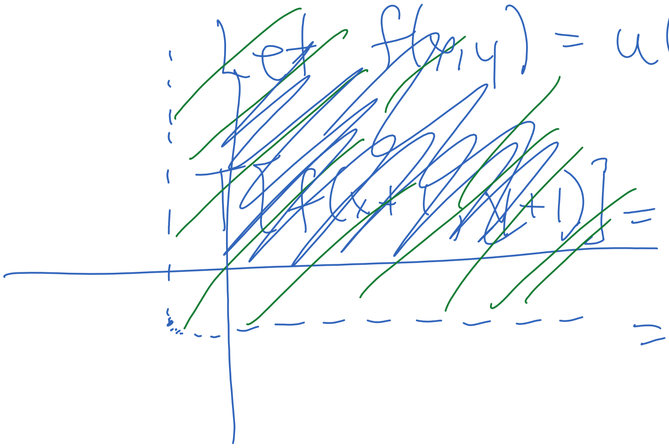
The system is SI iff

$$g(x-x_0, y-y_0) = T[f(x-x_0, y-y_0)]$$

for all shifts (x_0, y_0) and any $f(x,y)$.

Ex: $g(x,y) = T[f(x,y)] = f(x,y)\,u(x,y)$

~~Let $f(x,y) = u(x,y)$~~ Let $(x_0, y_0) = (-1,$



$T[f(x+1, y+1)] = f(x+1, y+1)\,u(x,y)$

$= u(x+1, y+1)\,\cancel{f(x,y)}\,u(x,y)$

$= u(x+1, y+1)\,u(x,y)$

$= u(x,y)$

$g(x+1, y+1) = u(x+1, y+1)$

$\neq u(x,y) \Rightarrow$ not SI

<u>Linear shift-invariant systems</u>

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x', y')\,\delta(x-x', y-y')\,dx'dy'$$

For linear system,

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

$$= T\left[\iint f(x', y')\,\delta(x-x', y-y')\,dx'dy'\right]$$

$$= \iint f(x', y')\,T[\delta(x-x', y-y')]\,dx'dy'$$

If we define $h(x,y) = T[\delta(x,y)]$, we have

$$h(x-x', y-y') = T[\delta(x-x', y-y')] \text{ for all } x', y'$$

for a shift-invariant system

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')\, h(x-x', y-y')\, dx'\, dy'$$

$$= f(x,y) * h(x,y) \quad : \text{2-D convolution}$$

$h(x,y)$ is the impulse response

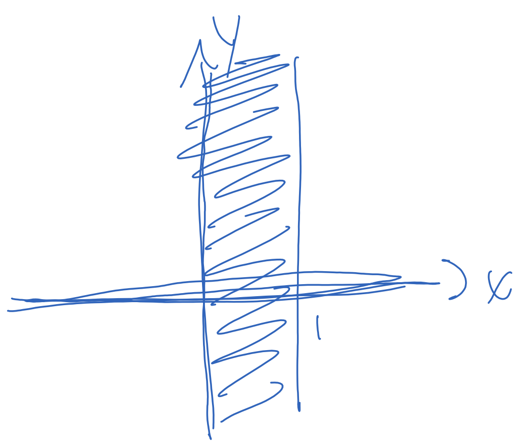
Ex: horizontal blur

$$h(x,y) = \int_0^1 \delta(x-x', y)\, dx'$$

$$= \int_0^1 \delta(x-x')\,\delta(y)\, dx'$$

$$= \delta(y') \int_0^1 \delta(x-x')\, dx'$$

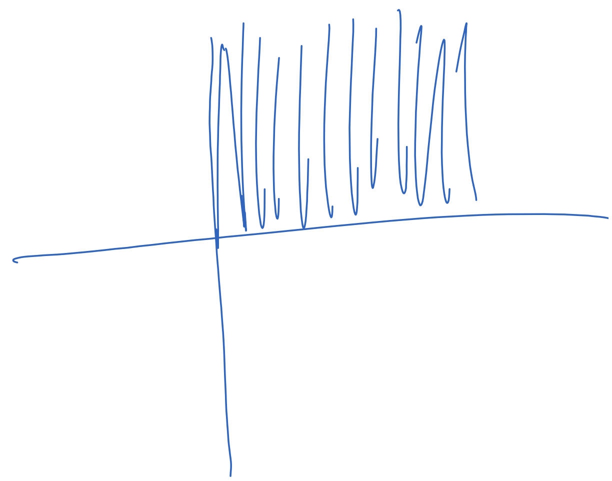
$$= \delta(y)\,[u(x) - u(x-1)]$$

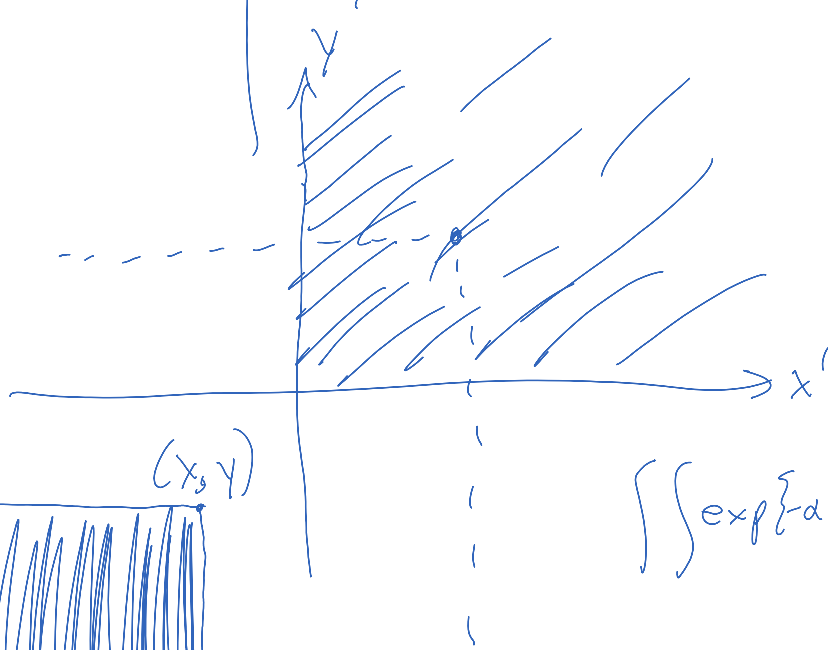


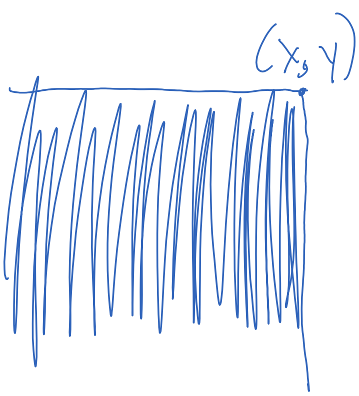
Read 4.3 - 4.5

HW #1 - to be posted

Ex: $\exp\{-\alpha x-\beta y\}u(x,y) * u(x,y)$







$\iint \exp\{-\alpha x'-\beta y'\}u(x',y')\, u(x-x', y-$

$g(x,y)=0$, $x<0$ or $y<0$

$$g(x,y)=\int_0^y\int_0^x \exp\{-\alpha x'-\beta y'\}\,dx'\,dy'$$

$$= \int^y e^{-\beta y'}dy' \int^x e^{-\alpha x'}dx'$$

$$= -\frac{1}{\beta} e^{-\beta y'}\Big|_0^y \cdot -\frac{1}{\alpha} e^{-\alpha x'}\Big|_0^x$$

$$= \frac{1}{\alpha\beta}\left[e^{-\beta y}-1\right]\left[e^{-\alpha x}-1\right]$$

$$g(x,y) = \frac{1}{\alpha\beta}\left[1-e^{-\alpha x}\right]\left[1-e^{-\beta y}\right]u(x,y)$$

Fourier transform

$$F(\omega_x,\omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp\{-j(\omega_x x+\omega_y y)\}\,dx\,dy$$

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\omega_x,\omega_y)\exp\{j(\omega_x x+\omega_y y)\}\,d\omega_x\,d\omega_y$$

$$f(x,y) \longleftrightarrow F(\omega_x,\omega_y)$$

$$\mathcal{F}\{f(x,y)\} = F(\omega_x,\omega_y)$$

Fourier transform properties

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x-x_0,\, y-y_0)\exp\{-j(\omega_x x+\omega_y y)\}\,dx\,dy$$

$$= \exp\{-j(\omega_x x_0 + \omega_y y_0)\}$$

Linearity

$$a f(x,y) + b g(x,y) \longleftrightarrow a F(\omega_x,\omega_y) + b G(\omega_x,\omega_y)$$

Convolution

$$f(x,y) * g(x,y) \longleftrightarrow F(\omega_x,\omega_y)\, G(\omega_x,\omega_y)$$

Multiplication

$$f(x,y)\, g(x,y) \longleftrightarrow \frac{1}{4\pi^2} F(\omega_x,\omega_y) * G(\omega_x,\omega_y)$$

Separability

$$F_y(\omega_x, y) = \int_{-\infty}^{\infty} f(x,y)\, e^{-j\omega_x x}\, dx$$

$$F(\omega_x,\omega_y) = \int_{-\infty}^{\infty} F_y(\omega_x, y)\, e^{-j\omega_y y}\, dy$$

$$= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x,y)\, e^{-j\omega_x x}\, dx \right] e^{-j\omega_y y}\, dy$$

$$= \int\int f(x,y) \exp\{-j(\omega_x x + \omega_y y)\}\, dx\, dy$$

FT of separable signals

$$f(x,y) = f_X(x)\, f_Y(y)$$

$$F(\omega_x,\omega_y) = F_X(\omega_x)\, F_Y(\omega_y)$$

- FT is also separable

Shifts

$$f(x-x_0, y-y_0) \longleftrightarrow \exp\{-j(\omega_x x_0 + \omega_y y_0)\} F(\omega_x,\omega_y)$$

$$\exp\{j(\omega_{x_0} x + \omega_{y_0} y)\} f(x,y) \longleftrightarrow F(\omega_x - \omega_{x_0}, \omega_y - \omega_{y_0})$$

(modulation)

Scaling

$$f(ax, by) \longleftrightarrow \frac{1}{|ab|} F\left(\frac{\omega_x}{a}, \frac{\omega_y}{b}\right)$$

Parseval's theorem

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2\, dx\, dy = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |F(\omega_x,\omega_y)|^2\, d\omega_x\, d\omega_y$$

~~Spatial derivatives~~

$$\cancel{\frac{\partial f(x,y)}{\partial x}} \longleftrightarrow -j\omega_x F(\omega_x,\omega_y)$$

$$\frac{\partial f(x,y)}{\partial y} \longleftrightarrow -j\omega_y F(\omega_x,\omega_y)$$

Laplacian:

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \longleftrightarrow -(\omega_x^2+\omega_y^2)F(\omega_x,\omega_y)$$

Ex: $f(x,y) = \begin{cases} 1, & 0 \le x \le a,\ 0 \le y \le b \\ 0, & \text{otherwise} \end{cases}$

$$F(\omega_x,\omega_y) = \int_0^b \int_0^a \exp\{-j(\omega_x x + \omega_y y)\}\, dx\, dy$$

$$= \left[\int_0^b e^{-j\omega_y y}\, dy\right]\left[\int_0^a e^{-j\omega_x x}\, dx\right]$$

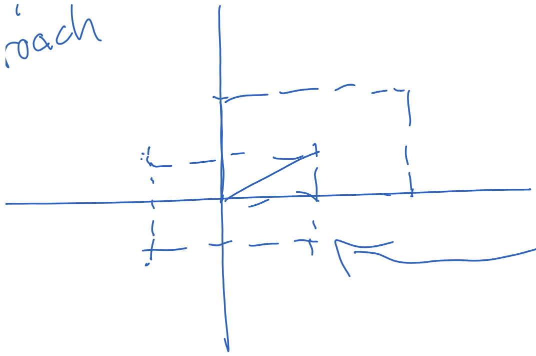
$$= -\frac{1}{j\omega_y} e^{-j\omega_y y}\Big|_0^b \cdot -\frac{1}{j\omega_x} e^{-j\omega_x x}\Big|_0^a$$

$$= -\frac{1}{\omega_x\omega_y}\left(1-e^{-j\omega_x a}\right)\left(1-e^{-j\omega_y b}\right)$$

$$= -\frac{1}{\omega_x\omega_y} e^{-j\frac{\omega_x a}{2}}\left(e^{j\frac{\omega_x a}{2}} - e^{-j\frac{\omega_x a}{2}}\right) \cdots$$

$$= ab \exp\left\{-j\left(\frac{a\omega_x}{2} + \frac{b\omega_y}{2}\right)\right\} \frac{\sin\frac{a\omega_x}{2}}{\frac{a\omega_x}{2}} \cdot \frac{\sin\frac{b\omega_y}{2}}{\frac{b\omega_y}{2}}$$

roach



2-D centered pulse

$\Rightarrow$ separable

$\Rightarrow$ FT is prod. of sinc functions plus linear phase

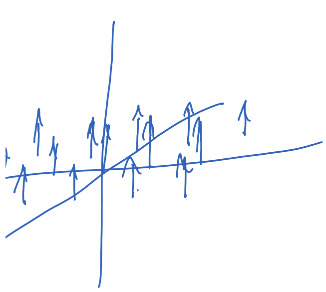
; IFT of $\delta(\omega_x - \omega_{x0}, \omega_y - \omega_{y0})$

$$x,y) = \frac{1}{4\pi^2} \int\int \delta(\omega_x - \omega_{x0}, \omega_y - \omega_{y0}) \exp\{j(\omega_x x + \omega_y y)\}\, d\omega_x\, d\omega_y$$

$$= \frac{1}{4\pi^2} \exp\{j(\omega_{x0} x + \omega_{y0} y)\}$$

$$1 \longleftrightarrow 4\pi^2 \delta(\omega_x, \omega_y)$$

$$f(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\Delta x, y - n\Delta y)$$



$$= \sum_m \sum_n \delta(x - m\Delta x)\, \delta(y - n\Delta y)$$

$$= \left[\sum_m \delta(x - m\Delta x)\right]\left[\sum_n \delta(y - n\Delta y)\right]$$

$$= \left[\sum_m a_m \exp\left\{j\frac{2\pi m}{\Delta x}x\right\}\right]\left[\sum_n b_n \exp\left\{j\frac{2\pi n}{\Delta y}y\right\}\right]$$

$$a_m = \frac{1}{\Delta x}\int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} \delta(x)\exp\left\{-j\frac{2\pi m}{\Delta x}x\right\}dx$$

$$= \frac{1}{\Delta x} \quad \text{for all } m$$

$$b_n = \frac{1}{\Delta y} \quad \text{for all } n$$

$$(x,y) = \left[\sum_m \frac{1}{\Delta x}\exp\left\{j\frac{2\pi m}{\Delta x}x\right\}\right]\left[\sum_n \frac{1}{\Delta y}\exp\left\{j\frac{2\pi n}{\Delta y}y\right\}\right]$$

$$= \frac{1}{\Delta x\Delta y}\sum_m\sum_n \exp\left\{j2\pi\left(\frac{m}{\Delta x}x + \frac{n}{\Delta y}y\right)\right\}$$

$$(\omega_x,\omega_y) = \frac{4\pi^2}{\Delta x\Delta y}\sum_m\sum_n \delta\left(\omega_x - m\frac{2\pi}{\Delta x}, \omega_y - n\frac{2\pi}{\Delta y}\right)$$

$$: \quad f(x,y) = \cos(\omega_{x_0}x + \omega_{y_0}y)$$

$$= \frac{1}{2}\exp\{j(\omega_{x_0}x + \omega_{y_0}y)\}$$

$$+ \frac{1}{2}\exp(-j(\omega_{x_0}x + \omega_{y_0}y))$$

$$F(\omega_x,\omega_y) = \frac{4\pi^2}{2}\delta(\omega_x - \omega_{x_0}, \omega_y - \omega_{y_0})$$

$$+ \frac{4\pi^2}{2}\delta(\omega_x + \omega_{x_0}, \omega_y + \omega_{y_0})$$

$$f(x,y) = \cos(\omega_{x_0} x + \omega_{y_0} y) * [\delta(x - x_0, y - y_0) + \delta(x + x_0, y + y_0)]$$

$$= \cos(\omega_{x_0}(x - x_0) + \omega_{y_0}(\cancel{x}y - y_0)) + \cos(\omega_{x_0}(x + x_0) + \omega_{y_0}(y + y_0))$$

ing conv. thm.:

$$g(x,y) = \delta(x - x_0, y - y_0) + \delta(x + x_0, y + y_0)$$

$$\omega_x, \omega_y) = \exp\{-j(\omega_x x_0 + \omega_y y_0)\} + \exp\{j(\omega_x x_0 + \omega_y y_0)\}$$

$$= 2\cos(\omega_x x_0 + \omega_y y_0)$$

$$x, \omega_y) = \left[\tfrac{1}{2}\delta(\omega_x - \omega_{x_0}, \omega_y - \omega_{y_0}) + \tfrac{1}{2}\delta(\omega_x + \omega_{x_0}, \omega_y + \omega_{y_0})\right] \times 2\cos(\omega_x x_0 + \omega_y y_0)$$

<u>impling</u>

$$s(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\Delta x, y - n\Delta y)$$

sampled signal:

$$f_s(x,y) = f(x,y)\, s(x,y)$$

$$= f(x,y) \sum_m \sum_n \delta(x - m\Delta x, y - n\Delta y)$$

$$= \sum_m \sum_n f(x,y)\,\delta(x - m\Delta x,\, y - n\Delta y)$$

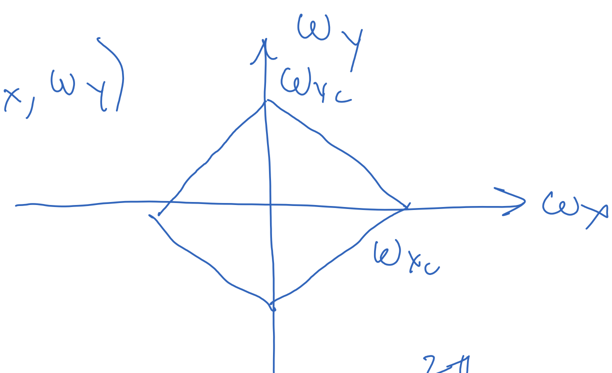
$$= \sum_m \sum_n f(m\Delta x, n\Delta y)\,\delta(x - m\Delta x,\, y - n\Delta y)$$

$$\omega_x, \omega_y) = \frac{1}{4\pi^2} F(\omega_x, \omega_y) * S(\omega_x, \omega_y)$$

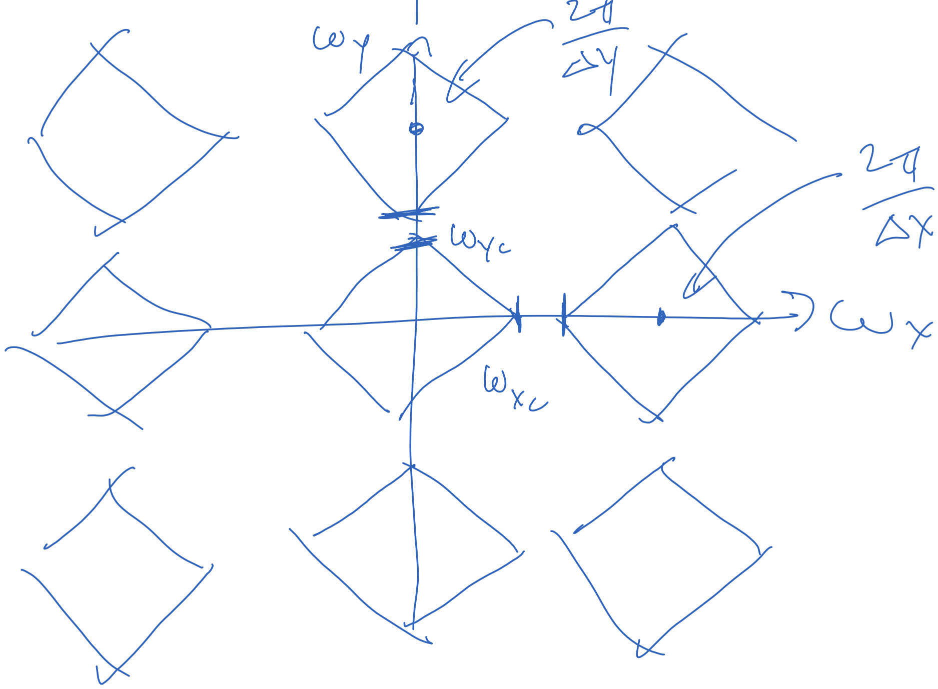
$$= \frac{1}{4\pi^2} F(\omega_x, \omega_y) * \frac{4\pi^2}{\Delta x \Delta y} \sum_m \sum_n \delta\left(\omega_x - m\frac{2\pi}{\Delta x},\, \omega_y - n\frac{2\pi}{\Delta y}\right)$$

$$= \frac{1}{\Delta x \Delta y} \sum_m \sum_n F\left(\omega_x - m\frac{2\pi}{\Delta x},\, \omega_y - n\frac{2\pi}{\Delta y}\right)$$

$F(\omega_x, \omega_y)$



$\omega_y)$



$$\frac{2\pi}{\Delta x} - \omega_{xc} > \omega_{xc}$$

$$\boxed{\begin{aligned} \frac{2\pi}{\Delta x} &> 2\omega_{xc} \\ \frac{2\pi}{\Delta y} &> 2\omega_{yc} \end{aligned}}$$

can recover $F(\omega_x, \omega_y)$ if the copies
't overlap. If Δx and Δy become too
ge, then copies overlap. The original copy
then no longer be recovered.

$\Rightarrow$ this is called <u>spatial aliasing</u>

<u>mpling theorem</u>

A bandlimited image $f(x, y)$ sampled on a uniform rectangular grid with spacing $\Delta x, \Delta y$ can be recovered from the sample values $f(m\Delta x, n\Delta y)$ if

$$\frac{2\pi}{\Delta x} > 2\omega_{xc}$$

and

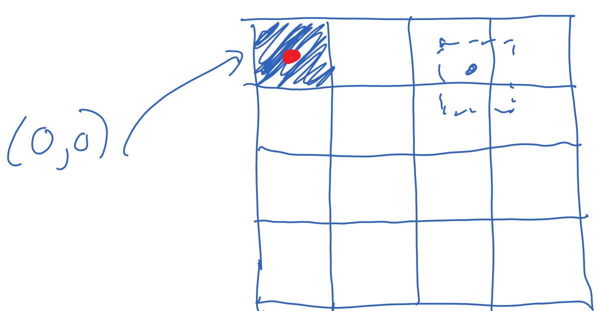
$$\frac{2\pi}{\Delta y} > 2\omega_{yc}$$

ℓ 4.6 - 4.7

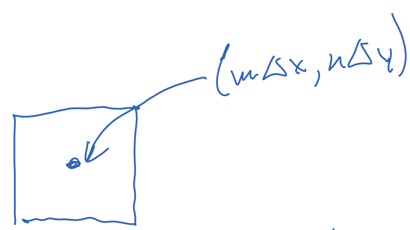
ct 2 due Fri.

deal sampling

more realistic sampling model
presents the sampling process as
tegrating intensity over rectangular
tches.



$$(m\Delta x, n\Delta y) = \int_{-\frac{\Delta y}{2}}^{\frac{\Delta y}{2}} \int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} f(m\Delta x - x', n\Delta y - y')\, dx' dy'$$



$$\Pi(x,y) = \begin{cases} 1, & -\frac{\Delta x}{2} < x \le \frac{\Delta x}{2}, \quad -\frac{\Delta y}{2} < y \le \frac{\Delta y}{2} \\ 0, & \text{otherwise} \end{cases}$$

$$\Delta x, n\Delta y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pi(x', y')\, f(m\Delta x - x', n\Delta y - y')\, dx' dy'$$

$\int^{\infty}$

$$x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Pi(x', y')\, f(x - x', y - y')\, dx'\, dy'$$

$$= f(x, y) * \Pi(x, y)$$

original scene

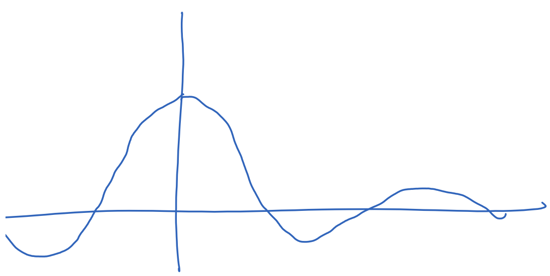
$$_i(x, y) = f_i(x, y)\, s(x, y)$$

$$= \sum_m \sum_n f_i(m\Delta x, n\Delta y)\, \delta(x - m\Delta x, y - n\Delta y)$$

$\Rightarrow$ filtering followed by ideal sampling

$$_i(\omega_x, \omega_y) = F(\omega_x, \omega_y)\, \mathcal{F}\{\Pi(x, y)\}$$

$$= F(\omega_x, \omega_y)\, \frac{\sin \frac{\Delta x}{2}\omega_x}{\frac{\Delta x}{2}\omega_x} \cdot \frac{\sin \frac{\Delta y}{2}\omega_y}{\frac{\Delta y}{2}\omega_y}$$



* we're sampling the filtered image instead of the original

* filtered signal has higher frequencies suppressed

– aliasing will be reduced but not eliminated

– image will be slightly blurred

…lay/reconstruction

…DSP, reconstruction is done in concept by …eating an impulse train from a sequence, …en lowpass filtering. This is implemented …ing electronics.

… image processing, the display is the filter. (The HVS is also a filter.) Samples are …jected as tiny rectangular patches (LCD display), …aussian spots (CRT), etc.

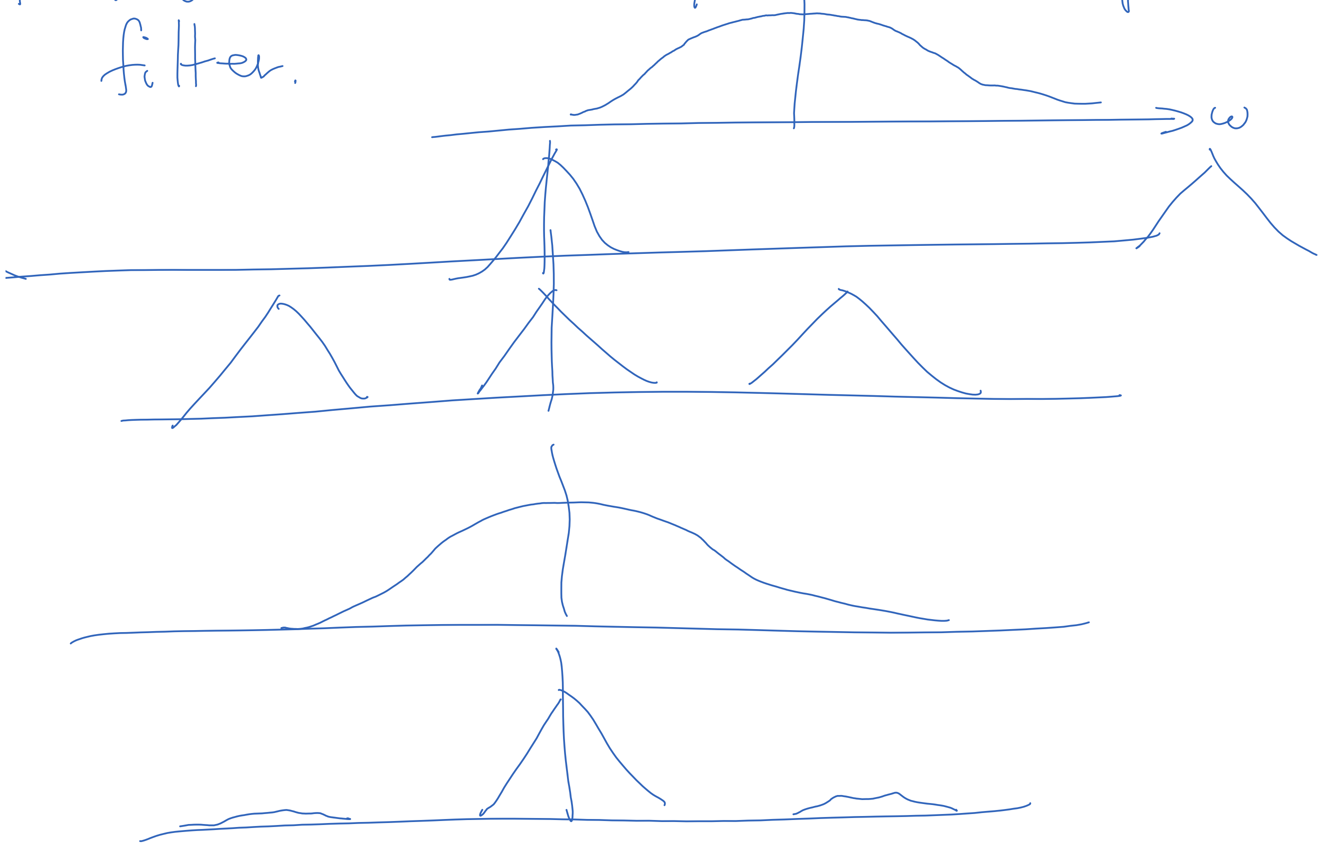
optical display "filter" can be modeled as —

$f_s(x,y) \to \boxed{p(x,y)} \to f_s(x,y) ** p(x,y)$ reduces/removes spectral replicas

$F_s(\omega_x, \omega_y)\, P(\omega_x, \omega_y)$

periodically replicated

Think of human visual system as a lowpass filter.



sic signals

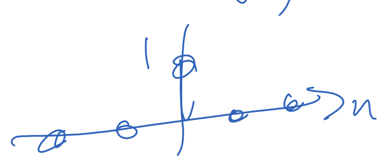
delta (impulse, unit sample)

Kronecker delta, not Dirac

$$\delta(m,n) = \begin{cases} 1, & m=n=0 \\ 0, & \text{otherwise} \end{cases}$$

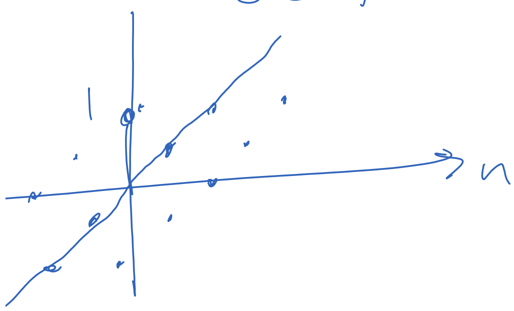
$$= \delta(m)\,\delta(n)$$

$\delta(n)$



S

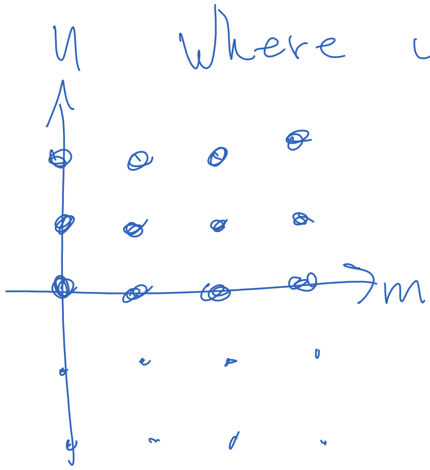
where $\delta(n) = \begin{cases} 1, & n=0 \\ 0, & n\neq 0 \end{cases}$

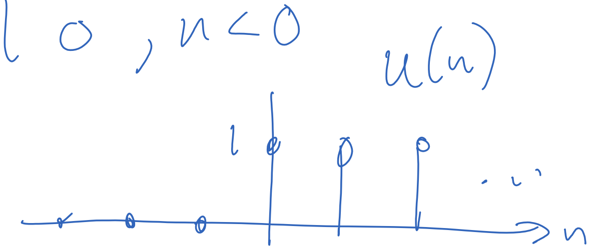


ep $u(m,n) = \begin{cases} 1, & m,n \geq 0 \\ 0, & \text{otherwise} \end{cases}$

$= u(m)\,u(n)$

where $u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$





d 4.8 - 4.9

posted

onential

$f(m,n) = \exp\{-\alpha m - \beta n\}\, u(m,n)$

$-\alpha m$ $-\beta n$

$$= \left[e \quad u(m) \right] \left[e \quad u(n) \right]$$

$\alpha, \beta \geq 0$ to prevent blowing up

.soid

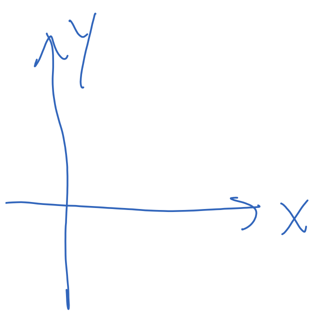
$$f(m,n) = f(x,y) \Big|_{(x,y) = (m\Delta x,\, n\Delta y)}$$

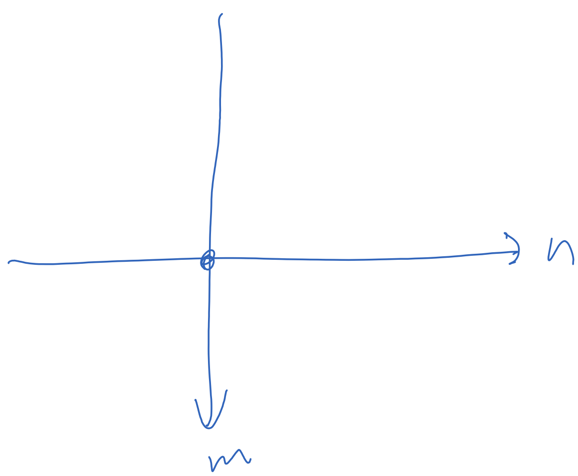
$$= \sin\left(\underbrace{\omega_{0x} \Delta x}\, m + \underbrace{\omega_{0y} \Delta y}\, n \right)$$

$$= \sin\left(\omega_{0m} m + \omega_{0n} n \right)$$

e

~s





l systems

$$f(m,n) \longrightarrow \boxed{\begin{array}{c}\text{2-D} \\ \text{system} \\ T[\cdot]\end{array}} \longrightarrow g(m,n)$$

Integer
index values

y

$f(m,n) + b\,g(m,n)] = aT[f(m,n)] + bT[g(m,n)]$

for all a, b, $f(m,n)$, $g(m,n)$

nvariance

$[f(m-k, n-\ell)] = g(m-k, n-\ell)$

for all k, ℓ, $f(m,n)$

(k, ℓ must be integers)

– can be represented by a convolution sum

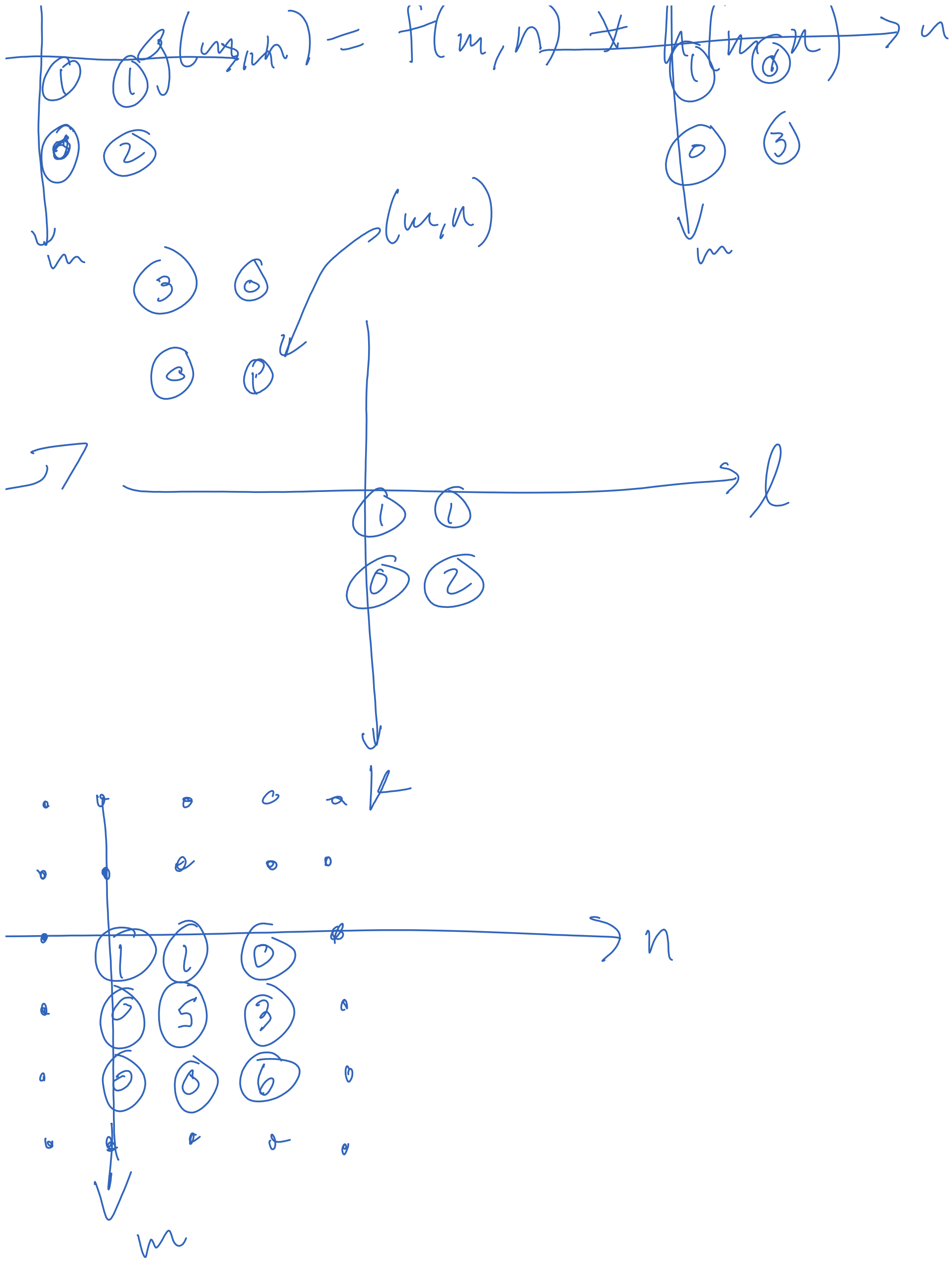
$$f(m,n) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} f(k,\ell)\,\delta(m-k, n-\ell)$$

$$g(m,n) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} f(k,\ell)\,h(m-k, n-\ell)$$

where $h(m,n) = T[\delta(m,n)]$

is the impulse response $h(m,n)$

$g(m,n) = f(m,n) * h(m,n)$



Transform of 2-D sequence

$$x(n) = X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

ω is in rad/sample

$$\omega_n) = \sum_{n=-\infty}^{\infty} x(m,n)\, e^{-j\omega_n n}$$

: 1-D FT of $x(m,n)$ along row m

$$n) = \sum_{m=-\infty}^{\infty} X_u(m; \omega_n)\, e^{-j\omega_m m}$$

$$-j(\omega_m m + \omega_n n)$$

$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(m,n)\, e$$

IFT of 2-D seq

$$\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_m, \omega_n)\, e^{j(\omega_m m + \omega_n n)}\, d\omega_m\, d\omega_n$$

$$= X(\omega_m - 2\pi, \omega_n) = X(\omega_m, \omega_n - 2\pi) = X(\omega_m - 2\pi, \omega_n - 2\pi)$$

impulse response gives ~~frequency~~

: of system

$$H(\omega_m, \omega_n) = \mathcal{F}\{h(m,n)\}$$

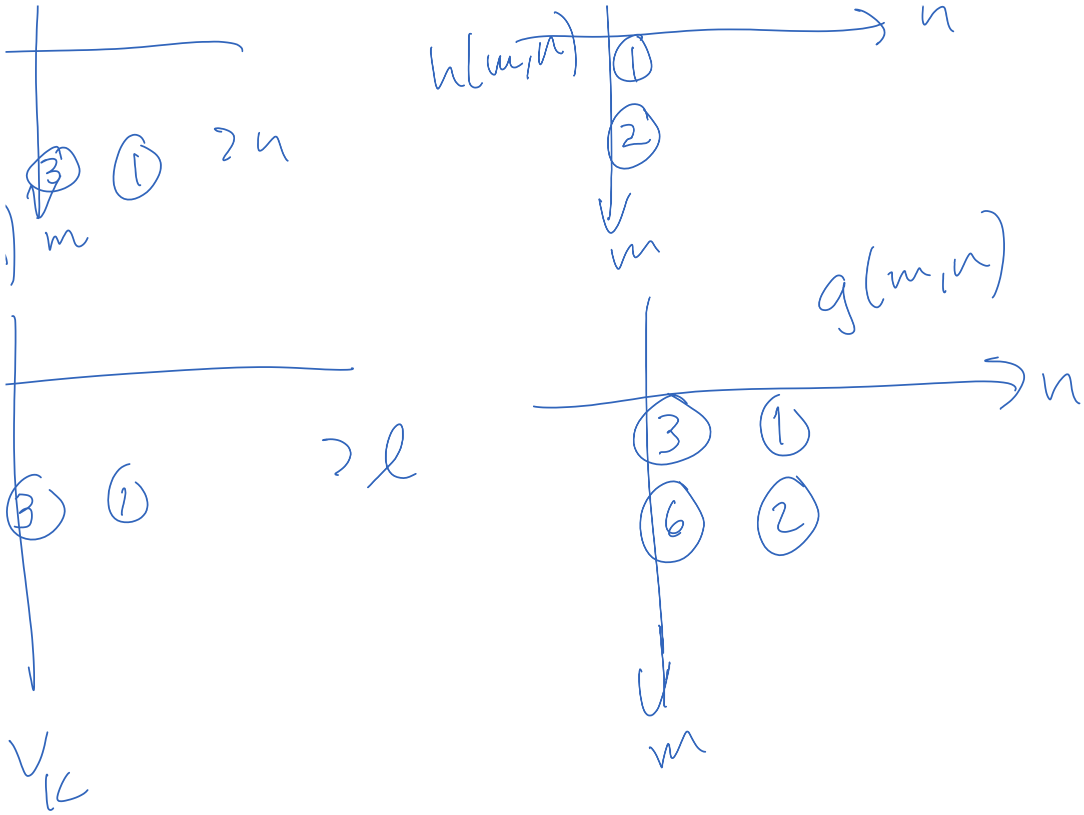
$$G(\omega_m, \omega_n) = F(\omega_m, \omega_n)\, H(\omega_m, \omega_n)$$

- convolution ~~th~~eorem

$$) * h(m,n) \longrightarrow F(\omega_m, \omega_n)\, H(\omega_m, \omega_n)$$

$$f(m,n) = 3\delta(m,n) + \delta(m, n-1)$$

$$h(m,n) = \delta(m,n) + 2\delta(m-1, n)$$



$f(m-k, n-l)\, e^{-j(\omega_m m + \omega_n n)}$

$(\omega_m k + \omega_n l)$

$$F(\omega_m, \omega_n) = 3 + e^{-j\omega_n}$$

$$H(\omega_m, \omega_n) = 1 + 2e^{-j\omega_m}$$

$$_m, \omega_n) = F(\omega_m, \omega_n)\, H(\omega_m, \omega_n)$$

$$= \left(3 + e^{-j\omega_n}\right)\left(1 + 2e^{-j\omega_m}\right)$$

$$= 3 + e^{-j\omega_n} + 6e^{-j\omega_m} + 2e^{-j(\omega_m + \omega_n)}$$

$$3\delta(m,n) + \delta(m, n-1) + 6\delta(m-1, n) + 2\delta(m-1, n-1)$$

: of discrete-space FT:

e – can decompose into two 1-D FTs

$f(m-k, n-l) \longleftrightarrow F(\omega_m, \omega_n)\, e^{-j(\omega_m k + \omega_n l)}$

ion $f(m,n)\,g(m,n) \longleftrightarrow \frac{1}{4\pi^2} F(\omega_m, \omega_n) * G(\omega_m, \omega_n)$

ll

ution thm.

al's theorem

$$\sum_{=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |f(m,n)|^2 = \frac{1}{4\pi^2} \int^{\pi} \int^{\pi} |F(\omega_m, \omega_n)|^2 \, d\omega_m \, d\omega_n$$

i function and can't be stored in a

. Integrals can't be calculated perfectly

omputer.

---

ete Fourier transform (DFT)

l image,

$) = F(\omega_m, \omega_n)$

$(\omega_m, \omega_n) =$

$$\left(\frac{2\pi k}{M}, \frac{2\pi l}{N}\right)$$

$$0 \le k \le M-1 \quad -j\left(\frac{2\pi km}{M} \quad 0 \le l \le N-1 \quad \frac{2\pi ln}{N}\right)\}$$

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp \qquad \text{- DFT}$$

$$(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k,l) \exp\left\{j\left(\frac{2\pi km}{M} + \frac{2\pi ln}{N}\right)\right\}$$

s:

r

val's theorem

$$\sum \sum |f(m,n)|^2 = \frac{1}{MN} \sum \sum |F(k,l)|^2$$

$$\sum \sum |f(m,n) - g(m,n)|^2 = \frac{1}{MN} \sum \sum |F(k,l) - G(k,l)|^2$$

rability - Can take 1-D DFT of rows

d then 1-D DFTs of columns (or reverse)

seq. is separable, then DFT is separable.

$(m,n) = f_m(m) f_n(n))$

iodicity

$$F(k+M, l+N) = F(k, l+N) = F(k+M, l) = F(k,l)$$

$\ldots n+M, n+N) = f(m, n+N) = f(m+M, n) = f(m,n)$

cular shift

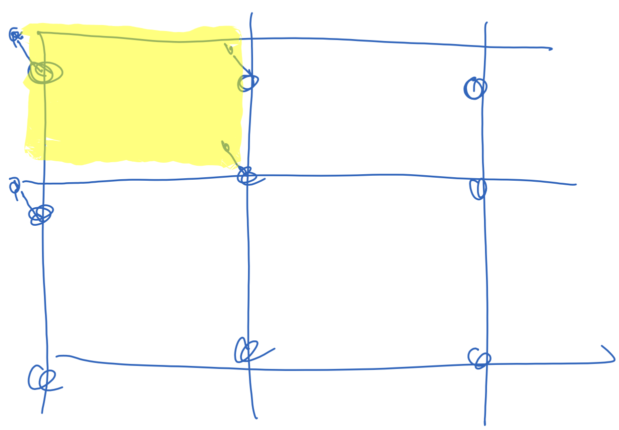
$\ldots_0 n - n_0) \overset{?}{\longleftrightarrow} F(k,\ell)\exp\left\{-j\left(\frac{2\pi k m_0}{M} + \frac{2\pi \ell n_0}{N}\right)\right\}$

$, ((n-n_0))_N)$

$((m))_M = m \bmod M$

…t of $\exp\left\{-j\left(\frac{2\pi k m_0}{M} + \frac{2\pi \ell n_0}{N}\right)\right\}$

$, n-n_0)$



$((n-n_0))_N)$

convolution

$\ldots(m,n) \overset{?}{\longleftrightarrow} \boxed{F(k,\ell)\,H(k,\ell)}$

$\ldots) = \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} f(k,\ell)\, h(((m-k))_M, ((n-\ell))_N)$

…e a linear convolution of $f(m,n)$

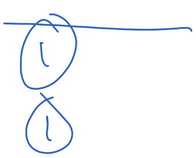
…ith a periodic version of $h(m,n)$

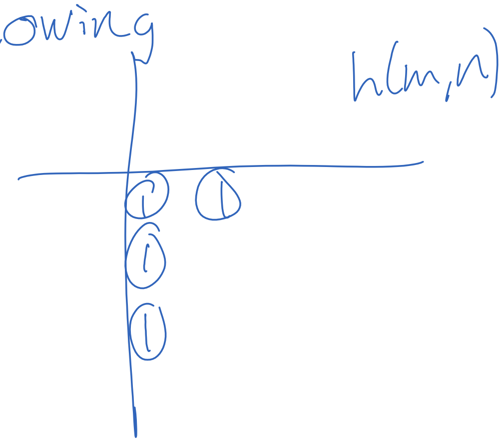
Size is $g(m,n)$? $\quad M \times N$

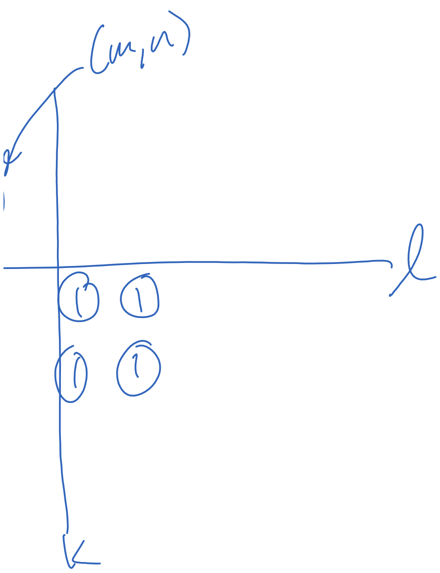
size is $f * h$? $\quad (M_f + M_h - 1) \times (N_f + N_h - 1)$

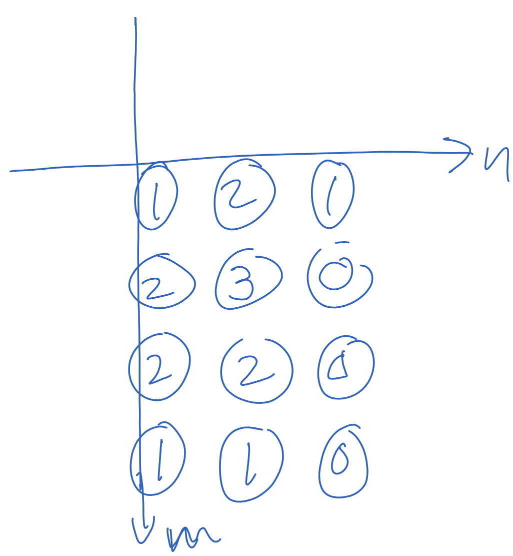
nplement circular convolution by
tic extension of one of the signals

nplement by linear convolution
ed by periodic extension of the
t, then windowing

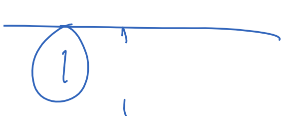








linear conv.



$) \circledast$ |

$) \circledast$!   Circ. conv.

- convolution w/ DFTs:

-et $h(m,n)$ be $M_h \times N_h$

$f(m,n)$ be $M_f \times N_f$

pad $h(m,n)$ and $f(m,n)$ to be $\geq (M_f + M_h - 1) \times (N_f + N_h - 1)$

DFTs of padded seq's.

-iply DFTs pointwise

$$H .* F$$

-e IDFT of result

Fourier transform (FFT)

-ient DFT implementation

ructed by decomposing DFT into
um of small DFTs

- requires M^2N^2 multiplies

- requires $MN\log_2 MN$ multiplies

24 x 1024,

DFT = 10^{12} mults.

FFT = 10^7

(1 day vs. 1 sec)

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\exp\left\{-j\left(\frac{2\pi km}{M}+\frac{2\pi ln}{N}\right)\right\}$$

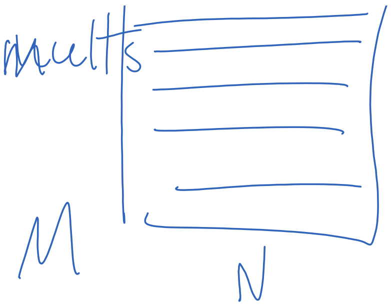
$$\sum_{m=0}^{M-1} \exp\left\{-j\frac{2\pi km}{M}\right\} \underbrace{\left\{\sum_{n=0}^{N-1} f(m,n)\exp\left\{-j\frac{2\pi ln}{N}\right\}\right\}}_{\text{1-D DFT}}$$

⇒ row-column decomposition

1-D FFT of all rows

Then 1-D FFT of columns of result

th DFT requires N^2 mults



es

row-col w/ DFTs

$+ \; N \cdot M^2 = NM(M+N)$

compare to $M^2 N^2$

1-D FFTs

(each requires $\frac{1}{2} N \log_2 N$ mults)

or R-C w/ FFTs

$$\frac{1}{2} MN \log_2 N + \frac{1}{2} NM \log_2 M$$

$$= \frac{1}{2} MN \log_2 MN$$

l × 1024,

direct DFT $= 2^{40} \approx 10^{12}$ mults

R-C DFT $= 2^{31} \approx 2 \times 10^{9}$ "

R-C FFT $= 10 \times 2^{20} \approx 10^{7}$

<u>radix FFT</u>

) divide-and-conquer strategy)

DFT is divided into successively
aller 2-D DFTs

ultiplies is $\frac{3}{8} N^2 \log_2 N^2$

ompared to $\frac{1}{2} N^2 \log_2 N^2$ for R-C FFT

* more complex than R-C FFT

Monday, February 11, 2019 8:12 AM

Image enhancement

Def: accentuating important image features or suppressing unwanted features to make visual information more accessible

Examples are edges, contrast, or texture.

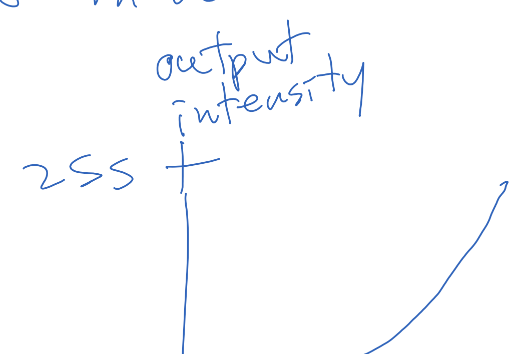
Enhancement methods are motivated by a wide variety of goals:

Ex: - contrast enhancement
- noise enhancement
- edge sharpening
- magnification

Tools:
- pointwise operations
- algebraic "
- spatial "
- combinations of these

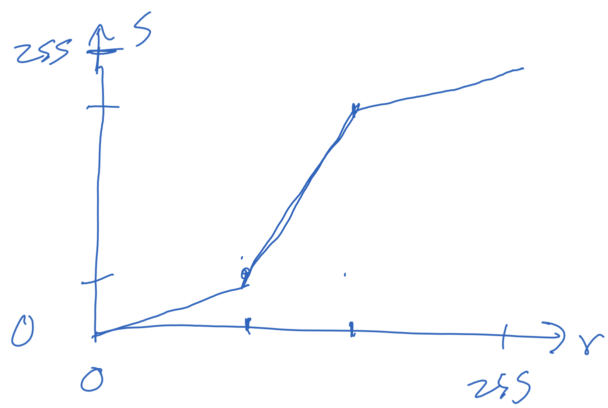
Pointwise operations

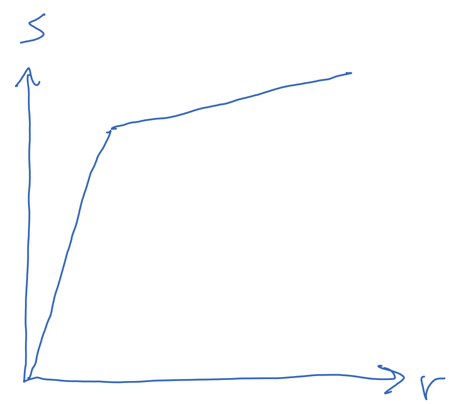
$s = T(r)$



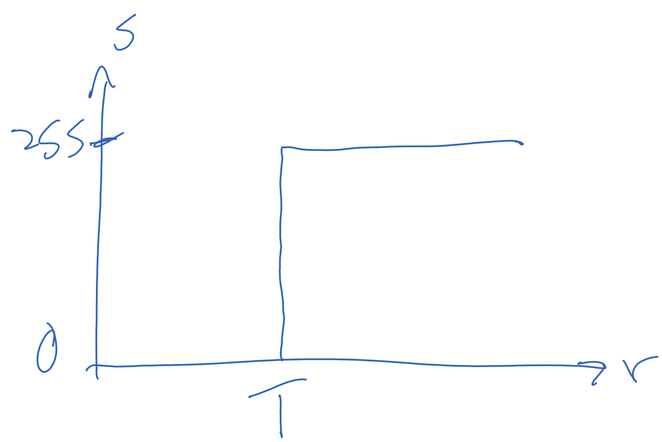
- pointwise transformation of intensity values

- contrast stretching – interval of pixels where intensity values are concentrated is stretched over larger intensity range to improve contrast



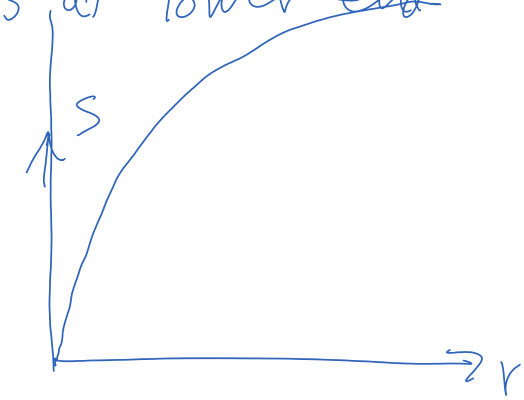


- thresholding



- useful for images known to be binary (e.g. faxes or digitized forms)
- useful for segmentation

• range compression — if dynamic range is too large, we lose details at lower ~~end~~



e.g. DFT magnitude

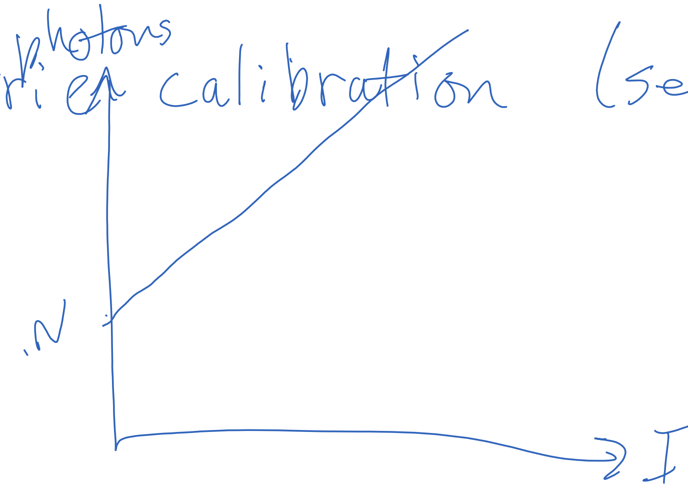
$$s = c\log(|r| + \varepsilon)$$

adjustable

---

photometric calibration (sensor or display)

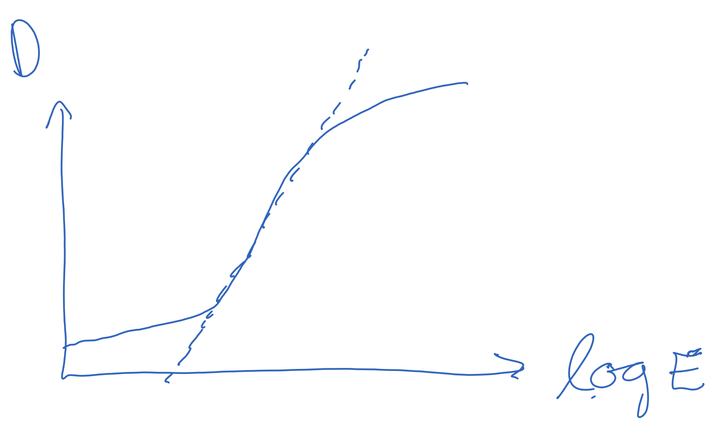
CCD



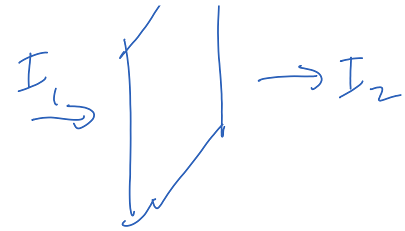
$$f(I) = \alpha I + N$$

$$\Rightarrow s(r) = \frac{r-N}{\alpha}$$

film negative



density $D = \log \frac{I_1}{I_2}$



exposure $E = T I$,

where T = exposure time, I = intensity of light

Read 3.4, Skim 3.5–3.6

$D \approx \gamma \log E + K$ in linear region

How do we correct for this response if measure light intensity (in the lab) passing through the negative?

$$D = \gamma \log T I_0 + \log e^K$$ where I_0 = object intensity

$$= \log (T I_0)^\gamma e^K$$

fixed (property of negative)

In lab,

$$D = \log \frac{I_s}{I_m}$$

where I_s = source lamp intensity (constant over image)

I_m = measured intensity on other side of negative (function of spatial coordinates)

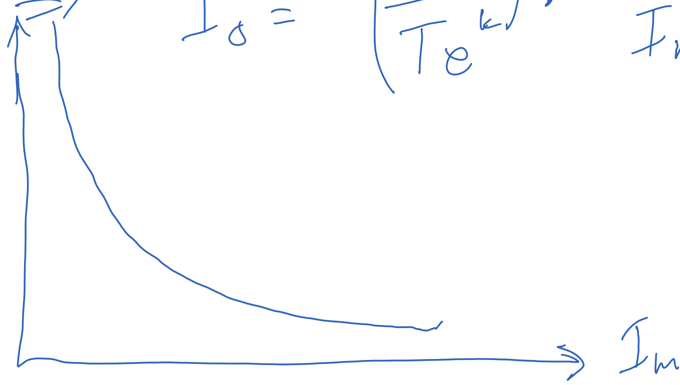
$$\log \frac{I_s}{I_m} = \log \frac{(T I_0)^\gamma e^k}{I_s}$$

$$I_m = (T I_0)^\gamma e^k$$

To correct, solve for I_0 in terms of I_m!

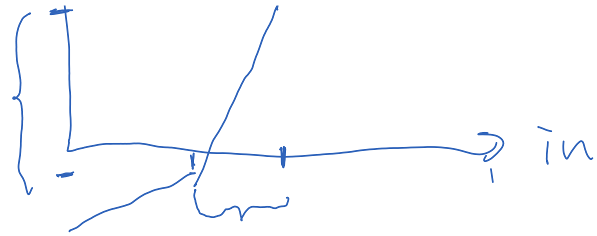
$$I_0 = \frac{1}{T}\left(\frac{I_s}{I_m e^k}\right)^{\frac{1}{\gamma}}$$

Let $\frac{1}{\gamma} = 1$ $I_0 \Rightarrow$ $I_0 = \left(\frac{I_s}{T e^k}\right) \cdot \frac{1}{I_m}$



Noise effects in pointwise operations

Note that noise variance changes when a pointwise operation is applied.



If the transformation can be locally approximated by a straight line

$$s = \alpha r + \beta$$

If $r = \bar{r} + u$, where u is zero-mean noise, σ_u^2 variance

then

$$s = \alpha(\bar{r} + u) + \beta$$

$$= \underbrace{\alpha \bar{r} + \beta} + \alpha u$$

αu is the new noise term with

variance $\alpha^2 \sigma_u^2$, std. dev. $= |\alpha| \sigma_u$

Algebraic operations

* Enhancement is sometimes performed by combining images

$f(m,n) + g(m,n)$ $\qquad$ $f+g$

$f(m,n) - g(m,n)$ $\qquad$ $f-g$

$f(m,n)g(m,n)$ $\qquad$ $f.*g$

$f(m,n)/g(m,n)$ $\qquad$ $f./g$

- image averaging

Let $f_i(m,n) = \bar{f}(m,n) + u_i(m,n)$

where $u_i(m,n)$ is independent, identically distributed, zero-mean, with σ_u^2

$$f_{AVE}(m,n) = \frac{1}{N}\sum_{i=1}^{N} f_i(m,n) = \frac{1}{N}\sum\left(\bar{f}(m,n) + u_i(m,n)\right)$$

$$= \bar{f}(m,n) + \frac{1}{N}\sum_{i=1}^{N} u_i(m,n)$$

$$\operatorname{var}\left[f_{AVE}(m,n)\right] = E\left\{\left[f_{AVE}(m,n) - \bar{f}(m,n)\right]^2\right\}$$

$$= E\left\{\left[\frac{1}{N}\sum_{i=1}^{N} u_i(m,n)\right]^2\right\}$$

$$= \frac{1}{N}\sigma_u^2$$

- image subtraction
  - ideal for highlighting subtle differences between similar images

* motion detection
* change detection in medical images

$$g(m,n) = f_2(m,n) - f_1(m,n)$$

- image multiplication / division
  - multiplication by binary image can mask out parts of image. 0's mask, and 1's retain
  - division can correct for nonuniform sensor response

Read 3.7
HW will be posted
Project 3 due Fri.

histograms

- a histogram is a plot of relative freq. of all gray levels
- pointwise operations modify the histogram
- histograms can be used to define

transformations

<u>histogram</u> <u>equalization</u> defines a pointwise transformation that tries to level out the histogram.

Let $h(x_i)$ = # of pixels with intensity x_i

L = # of gray levels

Then $\sum_{i=0}^{L-1} h(x_i)$ = total # of pixels = MN

Ideally, we would like $\hat{h}(x_i) = \frac{MN}{L}$ at each x_i

or $\sum_{i=0}^{k} \hat{h}(x_i) = \frac{MN}{L}(k+1)$

Can define a transformation between j + k such that

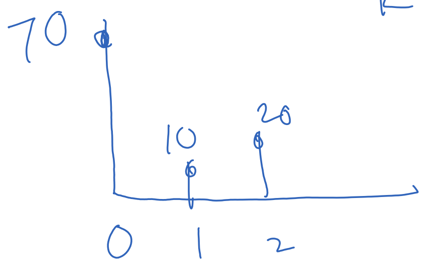
$$\sum_{i=0}^{j} h(x_i) = \sum_{i=0}^{k=\alpha(j)} \hat{h}(x_i) = \frac{MN}{L}(k+1)$$

Solving for k:

$$\sum^{j} h(x_i) - 1)$$

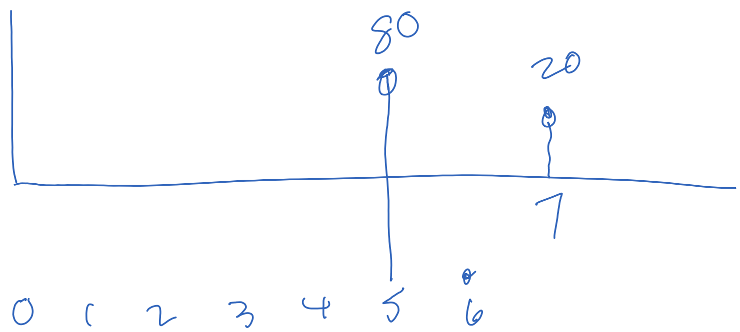
$T=8$

$$k = \text{round}\left(\frac{L}{MN} \cdot \right)$$



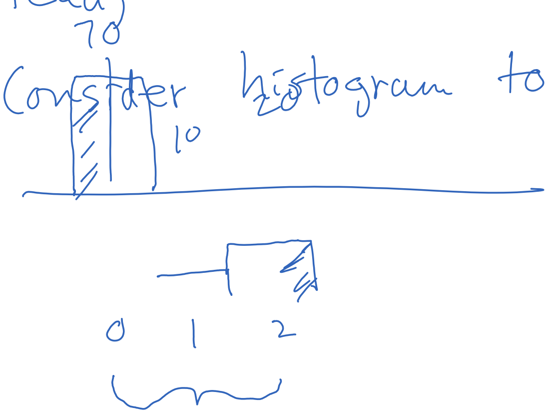
$L=8$

j	formula	k
0	$\frac{8}{100}(70)-1$	5
1	$\frac{8}{100}(80)-1$	5
2	$\frac{8}{100}(100)-1$	7



Instead,

1) Consider histogram to be pulses



2) Force histogram to fill range

$\Rightarrow$ only the right half of the left pulse & left half of the right pulse affect the spread

3) Spread depends on total area between

the pulse midpoints

4) total region will fill $L_d - 1$

where L_d = desired # of gray levels

sum right half of i-1 pulse and left half of $\sum_{i=1}$ i pulse

begin with $\sum_{i=1}$ to leave out left half of $h(x_0)$

$\frac{1}{2}\left(h(x_i) + \frac{1}{2}h(x_{i-1}\right)$

$$\frac{L_d - 1}{\underbrace{\left[\sum_{i=1}^{L-1} \frac{1}{2}\left(h(x_i) + h(x_{i-1})\right)\right]}_{MN - \frac{1}{2}h(x_0) - \frac{1}{2}h(x_{L-1})}} K$$

$$K = \text{round}\left[\frac{(L_d - 1)\left[\sum_{i=1}^{j} \frac{1}{2}\left(h(x_i) + h(x_{i-1})\right)\right]}{MN - \frac{1}{2}h(x_0) - \frac{1}{2}h(x_{L-1})}\right]$$

$L_d = 8$

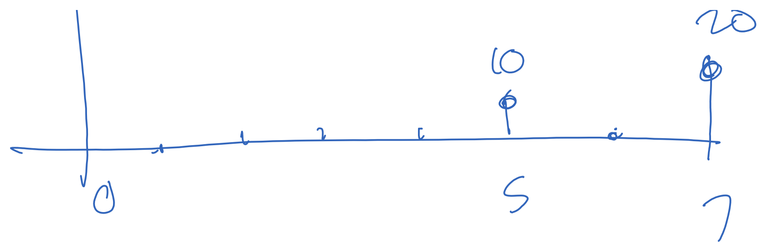
$N - \frac{1}{2}h(x_0) - \frac{1}{2}h(x_{L-1})$

$100 - \frac{1}{2}70 - \frac{1}{2}(20)$

55

j	formula	K
0	-	0
1	$\frac{7}{55}(40) = 5.1$	5
2	$\frac{7}{55}(55) = 7$	7

70



Spatial filtering

* can be linear or nonlinear

* spatial averaging (lowpass filter)
  - can be performed by convolution with uniform $(2M+1) \times (2M+1)$ kernel

$$g(m,n) = \sum\sum h(k,l)\, f(m-k, n-l)$$

$$= \frac{1}{(2M+1)^2} \sum_{k=-M}^{M} \sum_{l=-M}^{M} f(m-k, n-l)$$

- linear
- freq. response: $G(\omega_m, \omega_n) = H(\omega_m, \omega_n)\, F(\omega_m, \omega_n)$

$$H(\omega_m, \omega_n) = \frac{\sin\left(\frac{2M+1}{2}\omega_m\right)}{\sin \frac{1}{2}\omega_m} \cdot \frac{\sin\left(\frac{2M+1}{2}\omega_n\right)}{\sin \frac{1}{2}\omega_n}$$

Read 3.7

Effect: reduces corruption due to

noise. If noise is white with variance σ_n^2, Then $(2M+1)\times(2M+1)$ local averaging decreases variance to

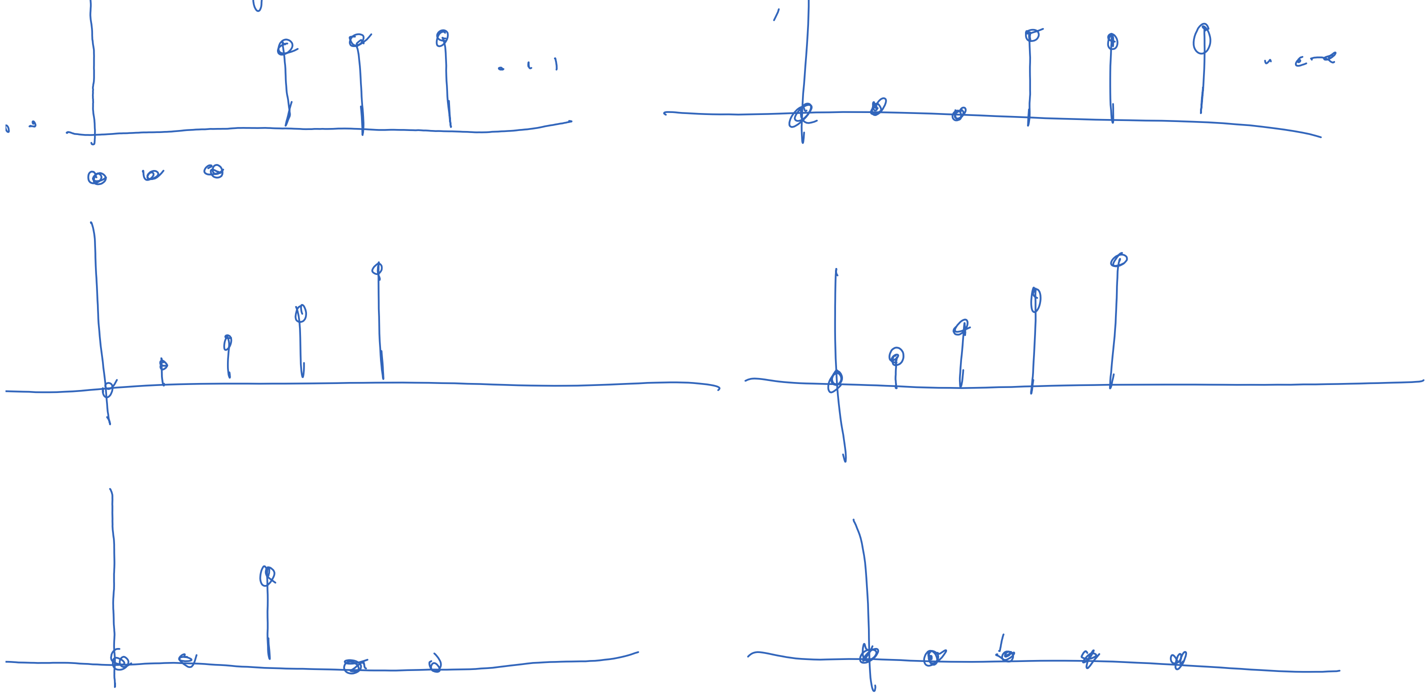
$$\frac{\sigma_n^2}{(2M+1)^2}$$

However, it also produces blurring of underlying image.

- median filtering

$$g(m,n) = \text{median}\{f(m-k, n-\ell),\ (k,\ell)\in W\}$$

3-point median



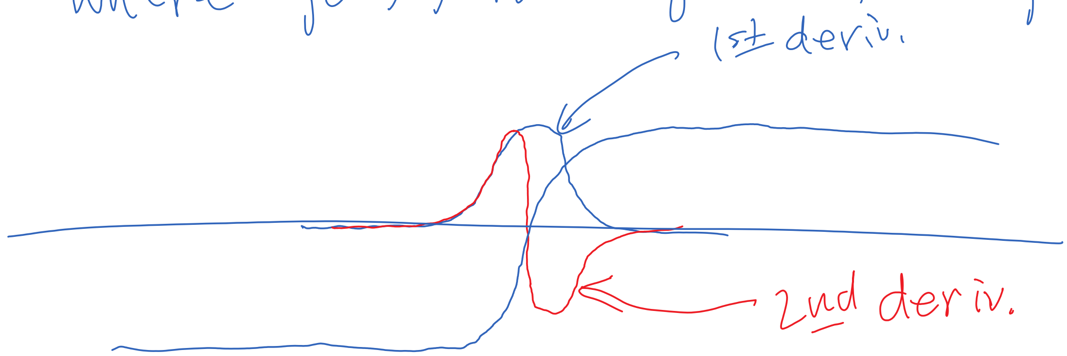
* sorting to choose middle value requires many computations

* sliding window can reduce comparisons required

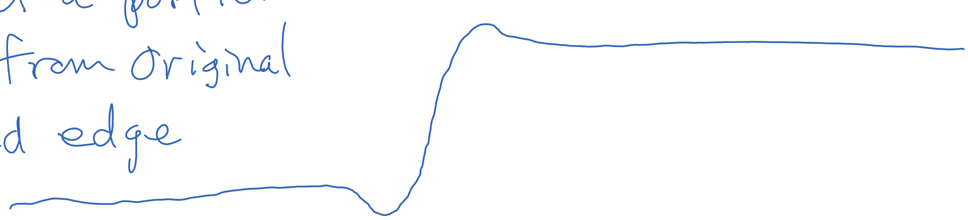
- unsharp masking
  - subtract a smoothed version of the image from the original. This will accentuate sharp changes in the image.
  - equivalently, we can add a gradient or highpass image

$$v(m,n) = u(m,n) + \lambda g(m,n)$$

where $g(m,n)$ is a gradient image



t: subtract a portion
nd deriv. from original
ccentuated edge



$$g(x,y) = \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2}$$: Laplacian

$$\frac{du(x)}{dx} \approx \frac{u(x+\Delta)-u(x)}{\Delta}$$

In discrete case: $\approx \dfrac{u(m+1)-u(m)}{1}$

Approx. 2nd deriv.: $u(m+1)-u(m)-\big(u(m)-u(m-1)\big)$

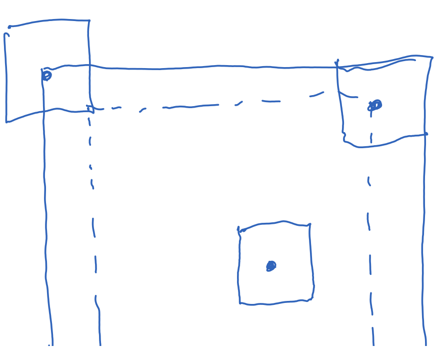
$$= u(m+1) - 2u(m) + u(m-1)$$

or 2-D,

$$g(m,n) = u(m,n+1) - 2u(m,n) + u(m,n-1) + u(m+1,n) - 2u(m,n) + u(m-1,n)$$

$$g(m,n) = u(m,n) * \underbrace{\begin{array}{|c|c|c|}\hline & 1 & \\ \hline 1 & -4 & 1 \\ \hline & 1 & \\ \hline\end{array}}_{\text{discrete Laplacian}}$$

Boundaries in spatial operations

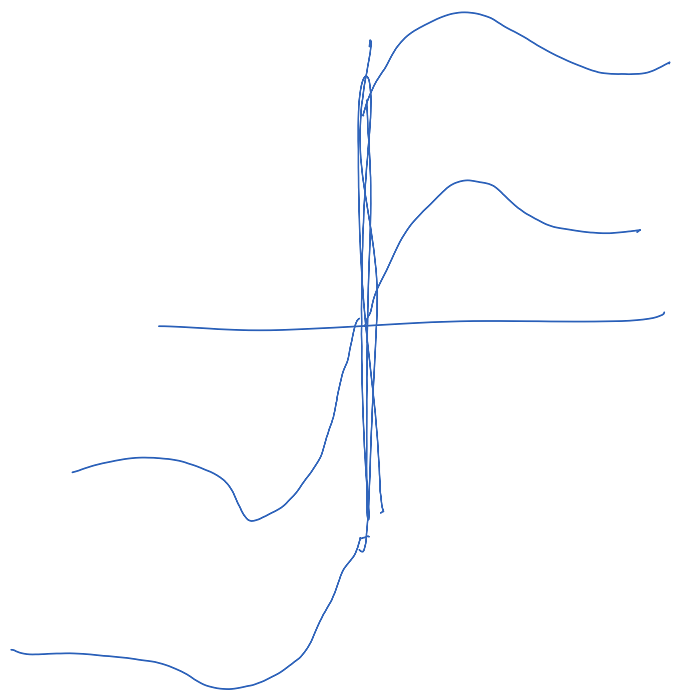


- neighborhood will hang off the image near the edges of the image

-ions:

- irror the image over he boundaries (symmetric extension)
- eplicate boundary pixels
- se average value as onstant background
- eplicate first difference
- hange algorithm at oundaries to avoid ising pixels outside ROS

- If pixels outside region of support (ROS) are assumed to be zero, this creates a false edge around the image that can lead to artifacts in processing



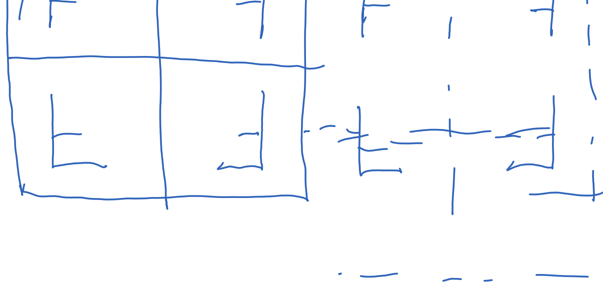
set posted

1, 3.48, 3.49

undaries in FFT-based processing

replicate boundaries, then zeropad to obtain a linear convolution and/or to get an image up to power of 2 size.

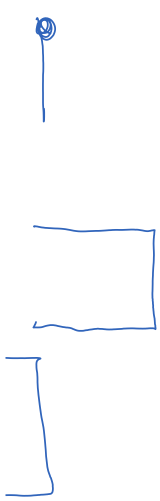
use symmetric extension before FFT



- after adding boundaries + processing,
only keep part that is size of original

in of neighborhood

vs.



Usually, we want to treat the center of the neighborhood as the origin. Otherwise, it will shift the output image

$$\sum \sum f(m-k, n-l)\, h(k,l)$$

ination examples

.ge detection

ıges characterize object boundaries and are
.erefore useful for segmentation, identification,
nd image registration (lining up two different
nages).

ixel values where abrupt grayscale changes
ccur in one direction are considered edges

ree steps:

Compute approximate gradi~~ents in~~ both directions

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

Sobel operators

$$\Downarrow \qquad\qquad \Downarrow$$

$$g_n(m,n) \qquad\qquad g_m(m,n)$$

compute gradient magnitude

$$g(m,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)}$$

strong gradients $\Rightarrow$ edges. So, threshold.

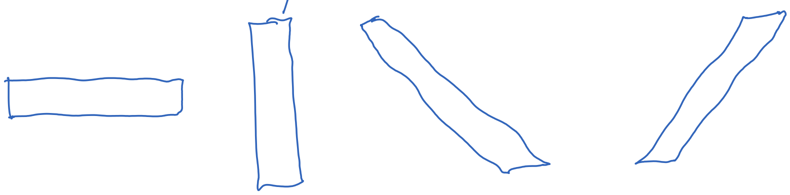
Anything over threshold is an edge.

.4) remove isolated points + link edges

rectional averaging

- smoothing can be done in directions that don't cross over edges (prevents blurring of edges)

  - image is smoothed in several directions independently



  - select pixel from one of these directional images for which the value is closest to original noisy pixel —
    selection is made point by point.

nomorphic filtering

$$u(m,n) = i(m,n)\,r(m,n)$$

where $i(m,n)$ = illumination of scene

$r(m,n)$ = reflectance of scene

$$(m,n) = \log u(m,n)$$

$$= \log i(m,n) + \log r(m,n)$$

$$= \hat{i}(m,n) + \hat{r}(m,n)$$

mptions:

illumination varies slowly across a scene

reflectance varies rapidly across a scene

=> possible to partially separate the two by filtering

Would like a filter such that

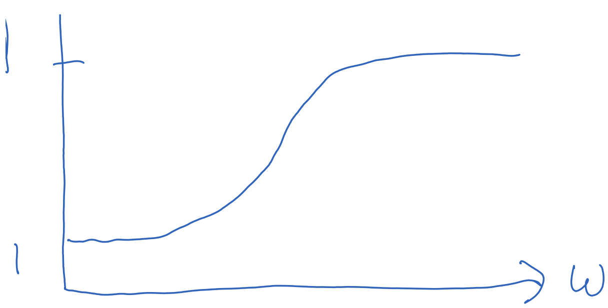
$$\hat{y}(m,n) = \alpha\, \hat{i}(m,n) + \beta\, \hat{r}(m,n)$$

$\alpha < 1$ dynamic range compression

$\beta > 1$ contrast enhancement

$\hat{i}(m,n)$ is lowpass

$\hat{r}(m,n)$ is highpass



ilter $\hat{u}(m,n)$ to get $\hat{y}(m,n)$

$$y(m,n) = \exp\left[\hat{y}(m,n)\right]$$

- undoes log operation

pace-variant sharpening

- edge strength calculation $g(m,n)$
- let $s(m,n)$ be sharpened version of $f(m,n)$

$$y(m,n) = g(m,n)\,s(m,n) + (k - g(m,n))\,f(m,n)$$

nistogram equalization followed by median filtering

Friday, February 22, 2019 8:02 AM

geometric transformations

- Images often need to be stretched, shrunk, shifted, rotated, magnified, or geometrically transformed in some other way.

Applications:

- correction of lens distortion
- correction for viewing angle
- correction of nonlinear field in MRI
- image registration (lining up for comparison)
- projection onto nonplanar surfaces (or reverse)
- correction of lens designed for hi-res middle for digital zoom

Two algorithms required:

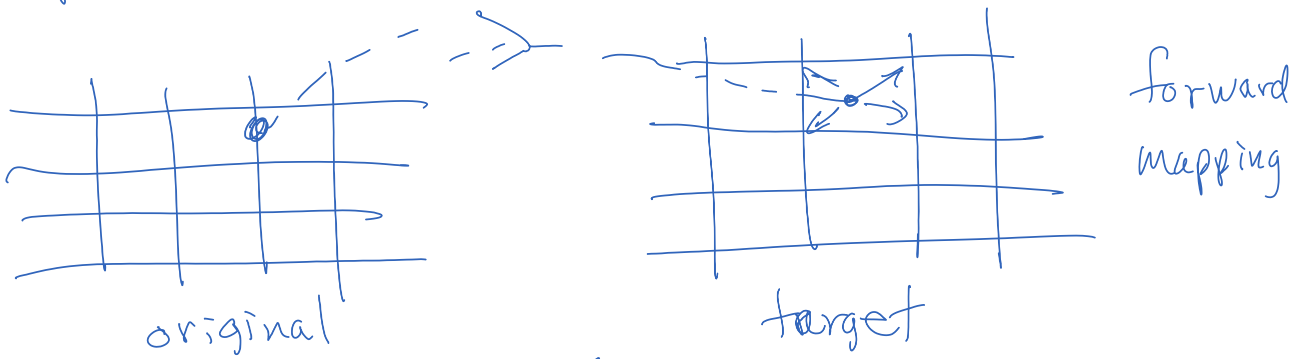
1) mapping that defines transformation from original target coordinates

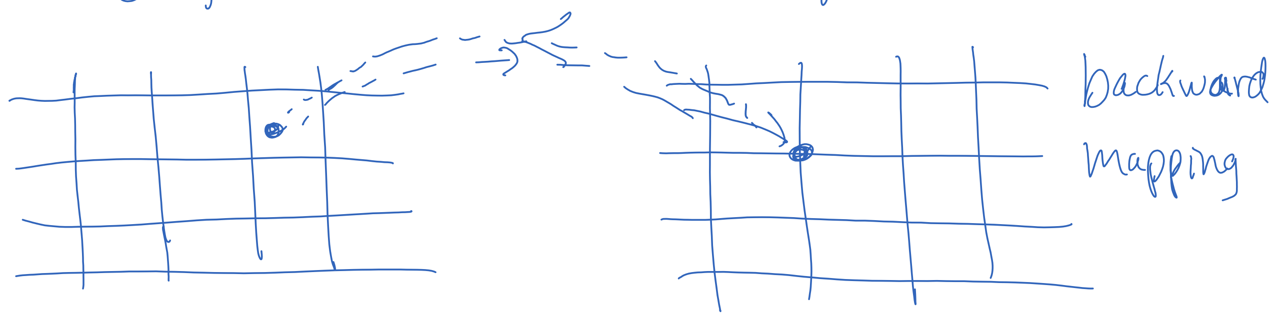
2) method of interpolating one set of sample values to another set

Interpolation

- integer grid points may not map to integer grid points in transformed image

two options:





Backward mapping is preferable:

- each output pixel is addressed exactly once, in line-by-line fashion
- forward mapping is wasteful — many pixel values may map outside the target image

$\Rightarrow$ in practice, we must define the mapping that takes us from the <u>target</u> pixel locations back to <u>original</u> pixel locations

Since original image location is generally between samples, we must interpolate.

Read 5.1, 5.2, 5.5, 5.7
HW to be posted

~~Control point specification~~

A set of corresponding points between two images can be chosen to estimate a mapping from one image to another.
The transformation is chosen to map the points in one image to the corresponding points in the other.
These points can be used to estimate specific parameters of an affine map, polynomial warp, etc.

Let $x_i = a_1 + a_2 x_0 + a_3 y_0 + a_4 x_0 y_0$

$$y_i = b_1 + b_2 x_0 + b_3 y_0 + b_4 x_0 y_0$$

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{bmatrix} = \begin{bmatrix} 1 & x_{01} & y_{01} & x_{01}y_{01} \\ 1 & x_{02} & y_{02} & x_{02}y_{02} \\ & & \vdots & \\ 1 & x_{0n} & y_{0n} & x_{0n}y_{0n} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

x_{ij} = input x coordinate for j^{th} point

x_{oj} = output x " " " "

y_{ij} = input y " " " "

y_{oj} = output y " " " "

$X = Ua$, a is unknown vector

* In general, U is not square.
  $\Rightarrow$ can't take simple inverse

* U must have at least as many rows as columns to obtain a unique solution.
  $\Rightarrow$ need at least as many point pairs as coefficients

$\Rightarrow$ choose $\hat{a}$ such that $\| x - Ua \|^2$ x
is minimized

- called least-squares solution

$$\hat{a} = (U^T U)^{-1} U^T x$$

$$y = Ub \quad \Rightarrow \quad \hat{b} = (U^T U)^{-1} U^T y$$

- control points should be well distributed and not in a straight line

the more the better

* Zooming in and out has the advantage that all points maintain a uniform spacing across image

$\Rightarrow$ we can use standard filtering/ resampling techniques

* we can change sample spacing by integer factors by relating sampled image to underlying continuous image.

Zooming in

* increasing sampling density

spread out samples by factor of L

in both directions + fill in zeros.

FT becomes

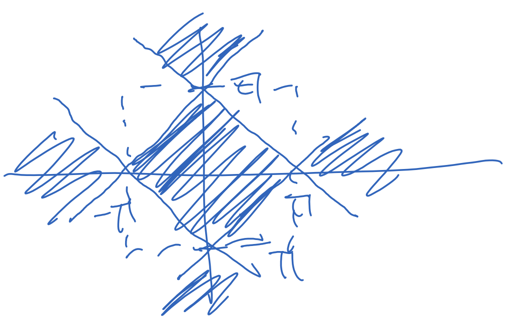
$$F_L(\omega_m, \omega_n) = \sum_{m',n'=\ldots,-L,0,L,\ldots}\sum f\left(\frac{m'}{L}, \frac{n'}{L}\right) \exp\{-j(\omega_m m' + \omega_n n')\}$$

$m' = Lm$

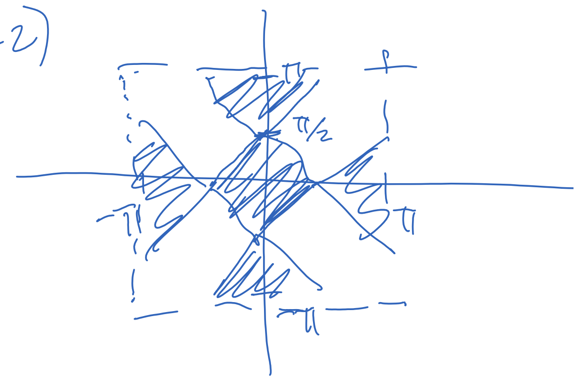
$n' = Ln$

$$= \sum_{m,n=\ldots,-1,0,1,\ldots}\sum f(m,n) \exp\{-j(\underbrace{L\omega_m}m + \underbrace{L\omega_n}n)\}$$

$$= F(L\omega_m, L\omega_n)$$



expand by L $\Longrightarrow$ $(L=2)$



Lowpass filter to eliminate higher spectral copies.

$\Longrightarrow$ creates factor of L interpolation

1) expand by factor of L

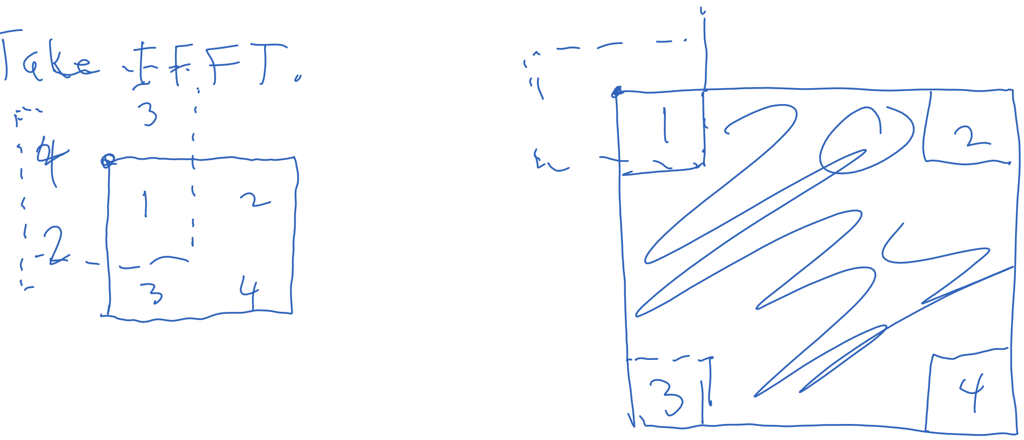
2) filter w/ LPF w/ cutoff of $\frac{\pi}{L}$.

Using FFTs:

1) Take FFT of original.

2) Zeropad to L times original size

3) Take IFFT.



zooming out

- decreasing density
  - same as if sampling at a lower density

    $\Rightarrow$ may cause aliasing

- First, filter with lowpass filter w/ cutoff $\frac{\pi}{M}$ to prevent aliasing

- Then downsample by factor of M.
  (Keep every M samples) $y = x(1:M:end, 1:M:end);$

  Using FFTs: