Lecture #9 Digital Image Processing.

March 31 '05

Last week : Image Restoration

Chap 5

Image u — noise ?

$\xrightarrow{u} \boxed{h} \rightarrow \oplus \rightarrow$ observed v - - - - $\boxed{g^*} \rightarrow \hat{u}$

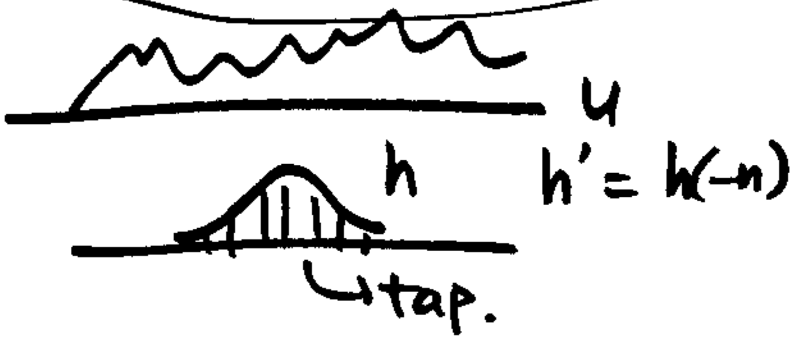
h: filter of degration process

Blur : motion. BW Limitation in Sensor

g : restoration filter

$u(n)$. $h(n)$

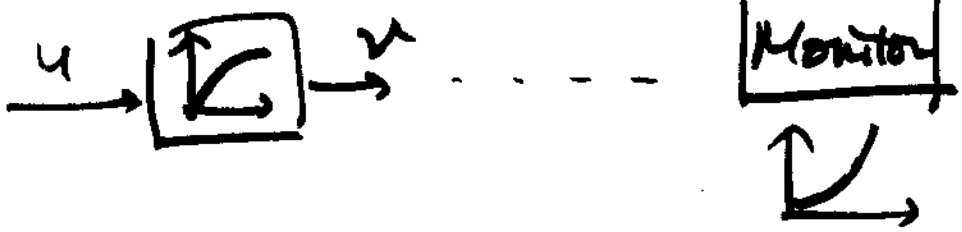
$\xrightarrow{u} \boxed{h} \rightarrow$ $u * h = \sum_{m=-\infty}^{\infty} u(m)\, \underline{h(n-m)}$



Linear System w/ Additive Noise.

⇓ Nonliear.

⇓ Multiplicative



Find the best linear estimator g^* s.t. $\hat{u}$
restoration filter

- $MSE = E\left(|u(m,n) - \hat{u}(m,n)|^2\right)$ minimized.

given $\underline{h}$, Autocorr. of $\underline{u}$, $\underline{\eta}$

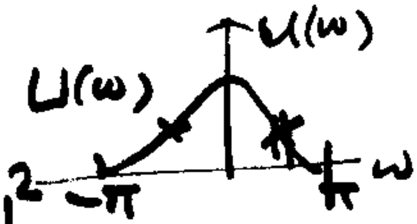
$$G^*(\omega) = \frac{H^*(\omega)\, S_{uu}(\omega)}{|H|^2 S_u(\omega) + S_\eta(\omega)}$$

Wiener Filter.

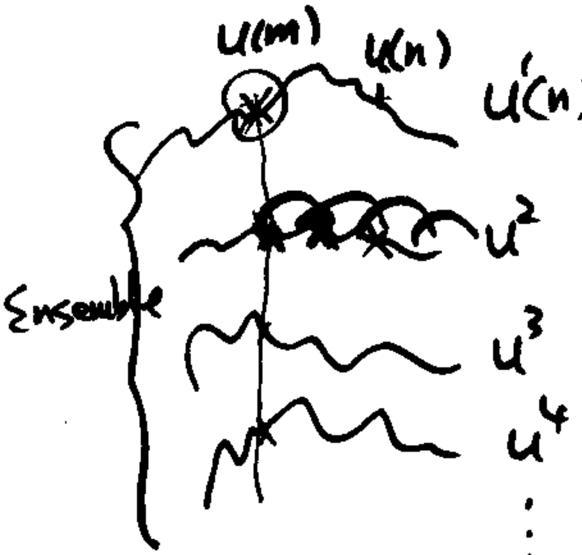
S_{uu} : Power Spectral Density of u
$S_{\eta\eta}$: " of η
$H(\omega)$: F.T. of $h(n)$
$G(\omega)$: F.T. of $g(n)$

$\xrightarrow{\text{F.T.}}$

$S_{uu} : \simeq |U(\omega)|^2$



Covered in Random ~~Sequence~~ signals & Noise.
EE 472 ?



Autocorrelation Function of $r(m,n)$

$= \underline{E[u(m)u^*(n)]}$ Stationary

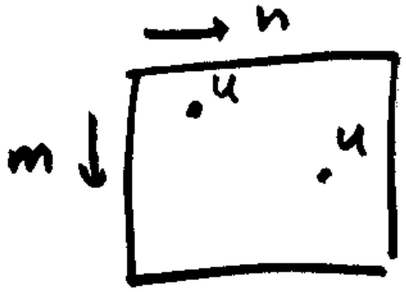
$= E[u(m-n)u^*(0)]$

$\underline{r(m)} = E[u(m)u^*(0)]$

$= E[u(m+i)u^*(i)]$

$r(0) = E[u(i)u^*(i)] = E(|u|^2)$

Avg. Power.



$r_{Image}(m,n)$

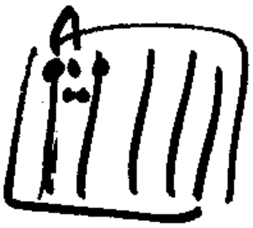


Image: Stationary X

Communication: V

U(n) AutoCorr. $r_u(n)$ $r(n)$

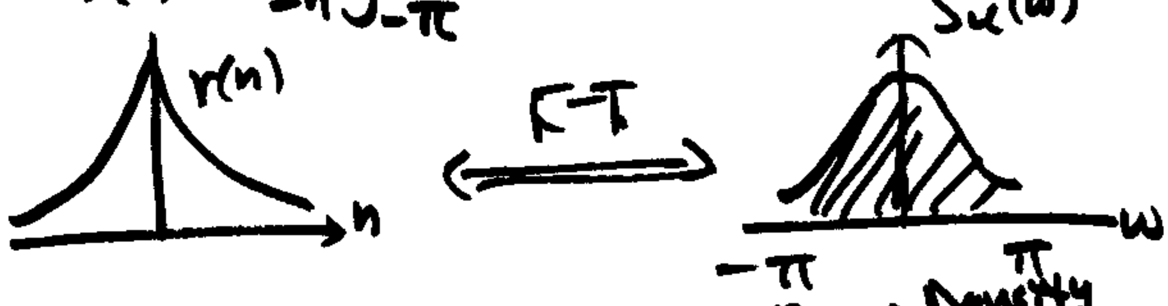
Random ⟶ Fixed

⇓ F.T.(r(n))

P.S.D $\underline{S_u(\omega)}$

$$S_u(\omega) = F.T.\left(r(n)\right)$$
$$= \sum_{n=-\infty}^{+\infty} r(n)\, e^{-j\omega n}$$

I.F.T.
$$r(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_u(\omega)\, e^{j\omega n}\, d\omega$$



Power. Density.

Set $n=0$
$$\underbrace{r(0)}_{= E\left[|u|^2\right]\ \text{Avg. Power}} = \underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_u(\omega)\, d\omega}_{\text{Integral}}$$

Matlab Example

256 × 256 $\boxed{\text{Image } u}$ $\xrightarrow{\text{fftn}}$ 256×256×3

```
up = abs(fftn(im2double(u))).^2
up(:,:,1)
up(:,:,2)
      3
```

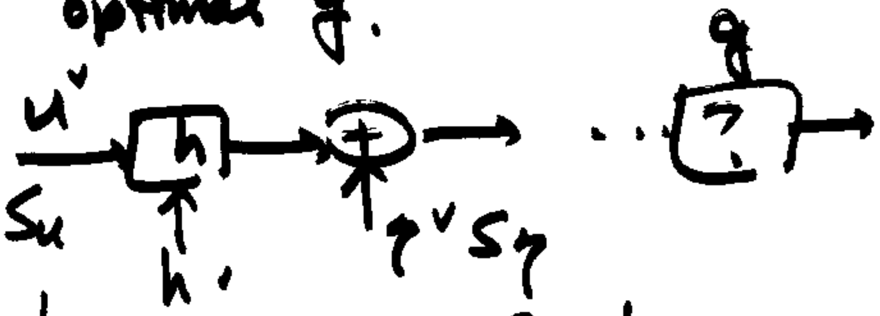
$|U(\omega)|^2 \approx S_u(\omega)$

$r_u(n) \xleftarrow{\text{I.F.T.}} S_u(\omega)$

```
ru = real(ifftn(up)))
     ↳ fftshift
```

If we know h, Su, Sη

optimal g.

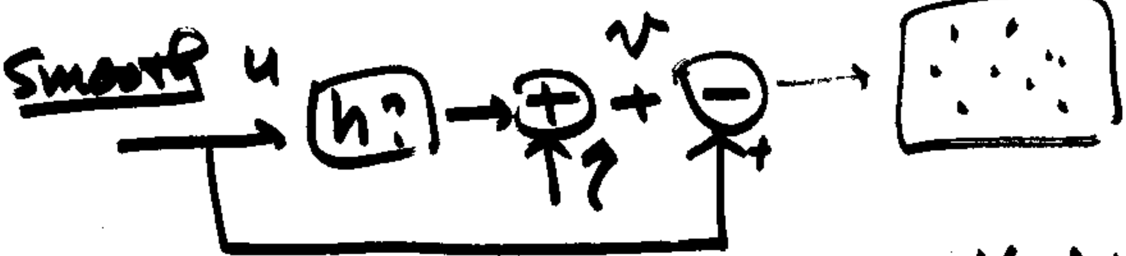


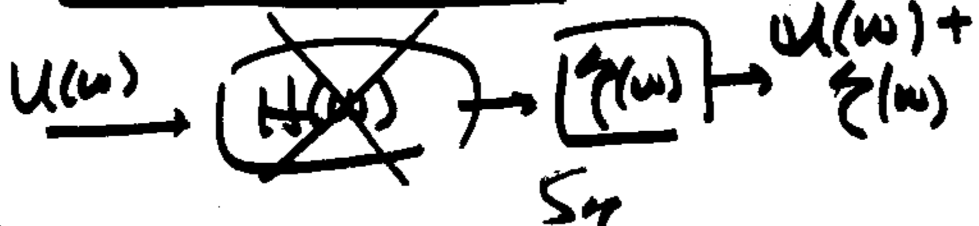
g* can be found

Su : ? F.T.(U)

AutoCorr of U.

<u>Phantom Signal</u> : smooth

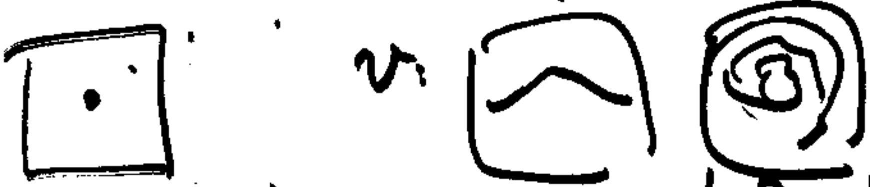




<u>Blind Deconvolution</u>

$v = \underline{u} * h = h$

if $u = \delta(n)$



h : point spread function

<u>LSI</u> Impulse response

Linear Shift Invariant. (LSI)

Image Analysis :

G & W Chap 10. Chap 9. Anil Jain.

Im → Pre-Process → Feature Extraction → Segmentation → Classification / Interpre. / Measurement

enhancement
restoration

- color Distribution
  Histogram
- Moments
- Edges
- Texture

region
curves.
lines.
shapes (2D)

Edge Detection & Linking

Edge : points ~~that~~ show rapid transition

Derivative : 1st order
2nd order

$\boxed{f(x,y)}$ → at point (x, y)
① rate of change
② the angle w/ max. rate of change

gradient

$\vec{v} = (f_x, f_y)$

$\theta = \tan^{-1}(f_y / f_x)$

$\neq 0$

$\boxed{B \mid \to W}$

$\to f_x$ 4

f_y 0

Operators

$f(m,n)$

f_x Prewitt h_1

f_y $h_1 \begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix}$ → Avg. Smoothing Reducing noise Less sensitive to noise.

$h_2 \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Sobel

$\begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \Longrightarrow$ Better performance in terms of location

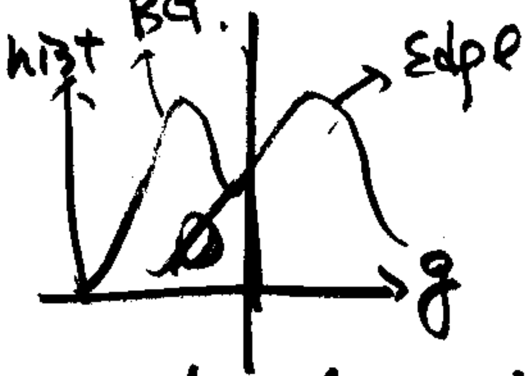
---

Gradient

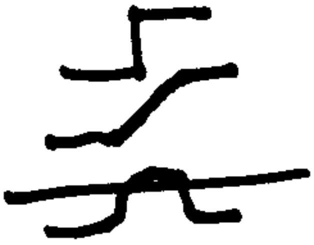
$I(m,n)$ $N\times N$ → $h_1 \rightarrow g_x$ ($N\times N$) → $\boxed{g = \sqrt{g_x^2 + g_y^2}}$ → $g > Th \Rightarrow$ Edge.

→ $h_2 \rightarrow g_y$ ($N\times N$) → $\boxed{\theta = \tan^{-1} g_y/g_x}$

(g, θ)



Edge : ✓Sobel. → + thinning.
Prewitt.
DOG → 2nd order.
LOG
Canny : 1986 → thinning.



Edge.

 $\left|\frac{\partial f}{\partial x}\right|$  $\Rightarrow$ Smoothing.

 $\frac{\partial f}{\partial x}$  Threshold

$\frac{\partial^2}{\partial x^2}$  $\rightarrow$ Threshold + zero crossing

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

Discrete Laplacian Op.

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}}_{\frac{\partial^2}{\partial x^2}} + \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}}_{\frac{\partial^2}{\partial y^2}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \boxed{4} & -1 \\ 0 & -1 & 0 \end{bmatrix}_{3\times 3}$$

Larger Mask: Smoothing $\uparrow$. Less sensitive to noise

Less precise

More computation

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}_{3\times 3}$$ Lap. Op.

$\rightarrow \boxed{\text{Smoothing}} \rightarrow \boxed{\text{Lap}} \rightarrow$

Gaussian. ∇^2

$\equiv \rightarrow \boxed{\nabla^2 \text{Gaussian}} \rightarrow$

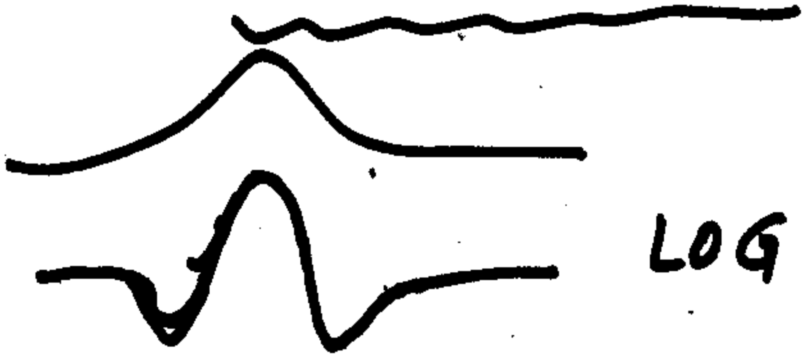
LOG

Gaussian $h(x) = c \cdot e^{-x^2/2\sigma^2}$

$$\nabla^2 h(x) = \frac{c}{\sigma^2} e^{-x^2/2\sigma^2}\left(\frac{x^2}{\sigma^2} - 1\right)$$

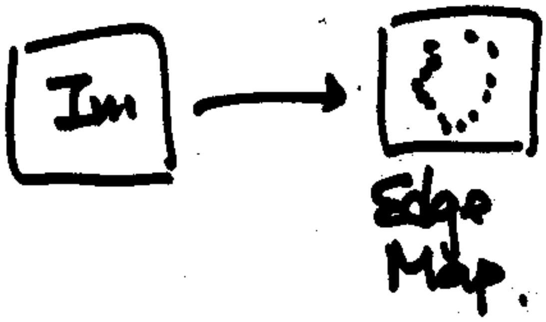
h

$\nabla^2 h$

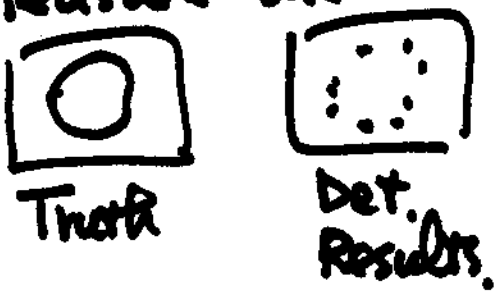


LOG

How to measure the accuracy of edge detection



① Measure the overlap w/ Ground Truth



Miss Rate = # Missed / # True

False Alarm Rate = # False / # Detected

② 

d_i : shortest dist to true points

$$\frac{\sum_{i=1}^{N} \frac{1}{1+\alpha d_i^2}}{\mathrm{Max}(N_D, N_T)}$$