Image Size: The number B bits required to store Digital (gray-scale) image is $B = MNk$.

Sampling Digitize coordinate value

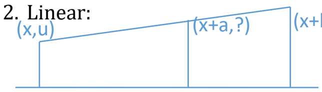
Quantization Digitize amplitude value

Resampling & Interpolation:

First step: align the grid.

Second step: Set the values to new grid.

1. Nearest Neighbor: closest in original
2. Linear:



$y = [(b-a)u + av]/b$

Adjacency useful for establishing object Boundary and defining image component

$N_4(p) = \{(x+1,y),(x-1,y),(x,y+1),(x,y-1)\}$

$N_D(p) = \{(x+1,y-1),(x-1,y+1),(x+1,y+1),(x-1,y-1)\}$

$N_8(p) = N_D(p) \cup N_4(p)$

m-adjacency

$q \in N_4(p) \vee q \in N_D(p) \wedge N_4(p) \cap N_D(p) \neq \emptyset$

Path $\forall i \in [1,n], p_i$ is adjacent to p_{i-1}

Intensity Transformation $s = T(r)$

Image Negatives $s = (L-1) - r$

Log Transform $s = c\log(1+r)$

Power Law $s = cr^{\gamma}$

flexible than log transformation.

$\gamma > 1$, lighter; $\gamma < 1$, darker

Piecewise-Linear Find a, b

$ar_1 + b = s_1, ar_2 + b = s_2$

pros: customize

cons: not automatic(*n photo » n function)*

Image Histogram $h(r_k) = n_k$

Normalized $p(r_k) = n_k/n$

mapping $s_k = ((L-1)/MN)\sum_{j=0}^{k} n_j$

Linear Function $r = T(x)$

Linear if $T(ax_1 + bx_2) = aT(x_1) + bT(x_2)$

w1 w2 w3 w4 w5 w6 w7 w8 w9	Correlation $s'_5 = \sum_{i=1}^{9} w_i s_i$ Convolution $s'_5 = \sum_{i=1}^{9} w_i s_{9-i}$

Smooth(low-pass) vs Sharp(high-pass)

0.1	0.1	0.1		1	2	1
0.1	0.1	0.1		2	4	2
0.1	0.1	0.1		1	2	1

Average Gaussian

	Time	Effect
Average	Fast	Box effect
Median	Slow	remove salt, pepper

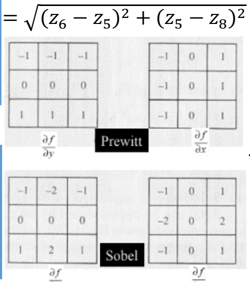
High-Boost = (A-1)Original + Highpass

Gradient of image

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

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

z1 z2 z3 z4 z5 z6 z7 z8 z9	For z_5, $\lvert\nabla \cdot f\rvert = \sqrt{(z_6-z_5)^2 + (z_5-z_8)^2}$



Laplacian:

0	1	0
1	-4	1
0	1	0

Gradient of image

$\nabla^2 f = -4f(x,y) + f(x,y+1) + f(x,y-1) + f(x+1,y) + +f(x-1,y)$

Complex number $e^{j\theta} = \cos\theta + j\sin\theta$

Unit Impulse $\delta(t) = \begin{cases} 1, if\ t = 0 \\ 0, if\ t \neq 0 \end{cases}$

Fourier Transfrom

$FT[f(t)] = F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt$

$FT^{-1}[F(\mu)] = f(t) = \int_{-\infty}^{\infty} f(t)e^{j2\pi\mu t}dt$

$f(t) \bigstar h(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)dt$

Sampling $\boldsymbol{f(t)}s_{\Delta T}(t) = \sum_{-\infty}^{\infty} f(t)\delta(t - n\Delta T)$

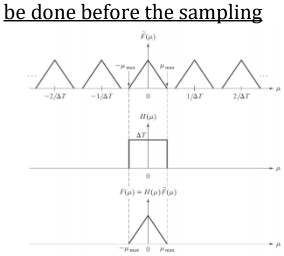
Nyquist rate

Sufficient separation guaranteed if:

$\frac{1}{\Delta T} > 2\mu_{max}$ ("=" is Nyquist rate)

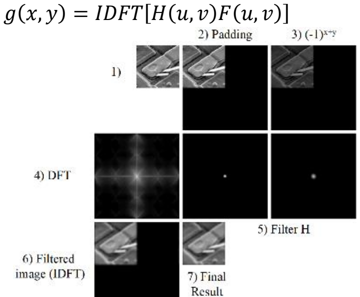
Aliasing Transform corrupted by frequencies from adjacent periods

be done before the sampling



Frequency Domain Filtering

$g(x,y) = IDFT[H(u,v)F(u,v)]$



Ideal Lowpass Filter

$H(u,v) = \begin{cases} 1, if D(u,v) \leq D_0 \\ 0, if D(u,v) > D_0 \end{cases}$

$D(u,v) = \sqrt{(u-P/2)^2 + (v-Q/2)^2}$

R is small will cause ringing and bluring

Butterworth Lowpass Filter

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

Gaussian Lowpass Filter

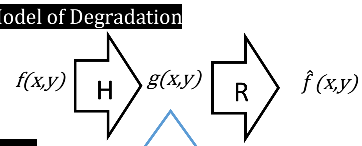
$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$

LowPass(u,v) = 1 - HighPass(u,v)

Objective of Restoration

$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2$

Model of Degradation



Noise

Image acquisition: light level, heat cause noise to the result.

Image transmission: wireless network might corrupted

Property

-Noise are independent of spatial loc.

-No correlation with pixel and noise

-Fourier spectrum is constant, the noise is usually called **white noise**

Mean filter

Arithmetic (noise-reducing by blurring)

Geometric (better than arit., loss detail)

$\hat{f}(x,y) = [\prod_{s,t \in W} g(s,t)]^{1/mn}$

Harmonic for salt and Guass noise

$\hat{f}(x,y) = mn/\sum_{(s,t)\in W} g(s,t)^{-1}$

Contraharmonic mean for salt and pepper but not both

p->$Q \in R^+$, s ->$Q \in R^-$

$\hat{f}(x,y) = \sum_W g(s,t)^{Q+1} / \sum_W g(s,t)^Q$

Order-Statistic Filters

Median, Max, Min, Mid-pt, α trimmed

d = mn – 1, median, d = 0, mean

[~~0~~, 3, 55, 65, 70, ..., ~~250~~] take mean

Adaptive Filters

- change behavior base on statistical characteristics of image in window
- superior to that no-adaptive filter

Ada. Local noise reduction

$\hat{f}(x,y) = (1 - \frac{\sigma_\eta^2}{\sigma_L^2})g(x,y) + \frac{\sigma_\eta^2}{\sigma_L^2}[m_L]$

Adaptive Median Filter

Wiener Filtering

Based on aforementioned conditions, the image estimate in the frequency domain is given by

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v) \quad \text{(Eq. 5.8-2)}$$

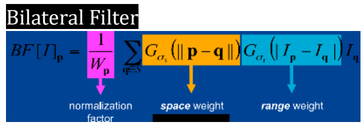
$H^*(u,v)$ = complex conjugate of $H(u,v)$

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

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

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

Bilateral Filter



Edge

Detection of short linear edge segments

Aggregation of edgels into extended edges

Image Gradient $\nabla f, \theta = \tan^{-1} f_y/f_x$

discrete:$f'_x[x,y] = f[x+1,y] - f[x,y]$

Sobel

Effect onf noise smooth then take edge

Canny Edge detection, localization (relievability)

Linear filtering, addition iid Guass-noise

response to edge, no noise

reduce the precision of localization

detect edge near true edge

Hysteresis

large thres. detects large scale edges

large thres. detects fine feature

Hough Transform

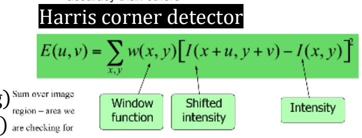
1. Initialize H[d, θ]=0
2. for each edge point I[x,y] in the image
   for θ = 0 to 180
   $d = x\cos\theta + y\sin\theta$
   H[d, θ] += 1
3. Find the value(s) of (d, θ) where H[d, θ] is maximum
4. The detected line in the image is given by $d = x\cos\theta + y\sin\theta$

Interest Point

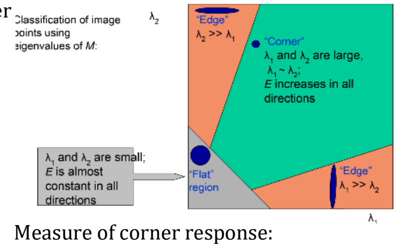
Distinctiveness

- We want to be able to reliably determine which point goes with which
- Some points can be localised or matched with higher accuracy than others

Harris corner detector



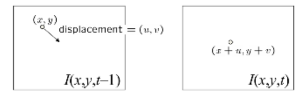
$E(u,v) \approx (u\ \ v)\sum_{x,y} w(x,y)\begin{bmatrix} I_xI_x & I_xI_y \\ I_xI_y & I_yI_y \end{bmatrix}\begin{pmatrix} u \\ v \end{pmatrix}$



Measure of corner response:

$R = \det M - k(\operatorname{trace} M)^2$

The brightness constancy constraint



- Brightness Constancy Equation:

$I(x,y,t-1) = I(x + u(x,y), y + v(x,y), t)$

Can be written as: shorthand: $I_x = \frac{\partial I}{\partial x}$

$I(x,y,t-1) \approx I(x,y,t) + I_x \cdot u(x,y) + I_y \cdot v(x,y)$

So, $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Lucas-Kanade:

How to get more equations for a pixel?

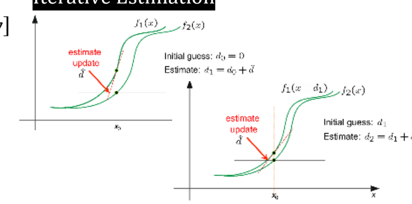
Spatial coherence constraint: pretend the pixel's neighbors have the same (u,v)

- If we use a 5x5 window, that gives us 25 equations per pixel
  $0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u\ v]$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b}$

Iterative Estimation



Conditions for solvability

Look Familiar?

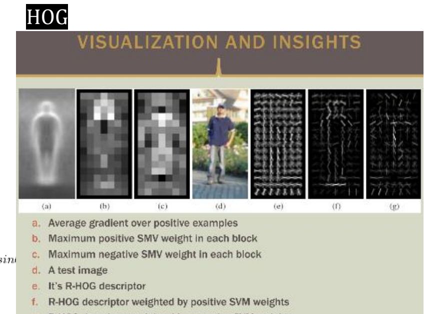
$$\begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum I_xI_t \\ \sum I_yI_t \end{bmatrix}$$

$A^TA \qquad A^Tb$

When is this solvable?

- $\mathbf{A^TA}$ should be invertible
- $\mathbf{A^TA}$ should not be too small
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^TA}$ should not be too small
- $\mathbf{A^TA}$ should be well-conditioned
  - λ_1/λ_2 should not be too large (λ_1 = larger eigenvalue)

HOG



Cell struct