1 Overview

Image acquisition model: Based on illumination (light) & reflectance (material/object properties):

$$f(x, y) = i(x, y) \cdot r(x, y)$$

2 main types of image processing:

1. **Spatial domain**: Operates on pixels directly
   (a) Intensity transformation: Operates on single pixels (pointwise), $T : \mathbb{N} \to \mathbb{N}$
   (b) Spatial filtering: Operates on $(m \times n)$ neighborhoods of pixels, $H : \mathbb{N}^{m\times n} \to \mathbb{N}$
2. **Frequency domain**: Transform to & process in frequency domain

PDF Properties:

1. $p_r(r) \geq 0 \; \forall \, r \in \mathbb{R}$; $\int_{-\infty}^{\infty} p_r(r)dr = 1$

2 Intensity Transformations

Basic transformations: Image negatives $[s = (L-1) - r]$, piecewise linear

- Log/exp $[s = c \cdot \log(1 - r)]$: Expands range of dark/light pixels, respectively
- Power law/gamma $[s = c \cdot r^\gamma]$:$\gamma \in (0, 1)$ expands dark pixels; $\gamma > 1$ compresses dark pixels

Histogram Processing: For $r_k = 0, 1, \ldots, L-1$:

$$\textbf{Histogram: } h(r_k) = \sum_{x=0}^{M-1}\sum_{y=0}^{M-1} \mathcal{I}(f(x, y) = r_k)$$

$$\textbf{Normalized: } p(r_k) = \frac{h(r_k)}{MN}$$

Histogram Equalization:

$$s = T(r) = (L-1) \cdot \int_0^r p_r(w)dw \quad \left[(L-1) \cdot \sum_{j=0}^{k} p(r_j)\right]$$

- Stipulations: T is monotonically increasing, $0 \leq T(r) \leq L-1$
- $p_s(s) \sim \text{Uniform}(0, L-1)$ for p_r continuous
- Proof: Via that $|p_r(r)dr| = |p_s(s)ds| \implies p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$

Histogram Matching:

1. Apply histogram equalization to the source r to get $s = T(r) \sim \text{Uniform}(0, L-1)$
2. Apply histogram equalization to the target z to get $s' = G(z) \sim \text{Uniform}(0, L-1)$
3. Final transformation: $z = G^{-1}(T(r))$

Histogram Statistics:

$$\text{Mean: } m = \sum_{i=0}^{L-1} r_i p(r_i) \; ; \quad n^{th} \text{ moment: } \mu_n(r) = \sum_{i=0}^{L-1}(r_i - m)^n p(r_i) \qquad [\text{Variance: } \mu_2(r)]$$

3 Spatial Filtering

Linear Spatial Filtering: On neighborhoods $\{f(x + s, y + t) : -a \leq s \leq a, -b \leq s \leq b\}$:

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

- $(2a + 1) \times (2b + 1)$ kernel $\to$ a rows of padding on top & bottom, b columns on left & right

Smoothing Filters: Blur images for noise reduction; larger mask $\to$ more blurring, typically

1. Weighted average filter: $g(x, y) = \frac{\sum_s\sum_t v(s,t)f(x+s,y+t)}{\sum_s\sum_t v(s,t)}$
2. Order-statistic filters (nonlinear): Median/max/min filters (good for salt & pepper)

Sharpening Filters: Highlight transitions in intensity; compute using 2D discrete Laplacian [limit definition of the derivative evaluated at $h = 1$, $\delta = 1$]:

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1, y) + f(x, y+1) + f(x-1, y) + f(x, y-1) - 4f(x, y)$$

$\to$ **Composite Laplacian** (sharpened image): $g(x, y) = f(x, y) - \Delta f(x, y)$

Unsharp Masking: Use blurring to sharpen images

1. Blur the original image f to obtain a blurred image $\bar{f}$
2. Take $g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$
3. For $k \geq 0$ [scaling factor], take $g(x, y) = f(x, y) + k \cdot g_{mask}(x, y)$

($*$) Gradients for image sharpening: use $g(x, y) = |\nabla f|$ (various formulas; ex: central difference)

4 Frequency-Domain Filtering

4.1 Background

Euler's formula: $e^{i\theta} = \cos(\theta) + i\sin(\theta)$

Complex Functions: $\bar{F(u)} = R(u) - iI(u)$; $|F(u)|^2 = R(u)^2 + I(u)^2$

The Impulse Function:

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}$$

Properties:

- Integral: $\int_{-\infty}^{\infty} \delta(t)dt = 1$
- ***Sifting property***: $\int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = f(t_0)$
- Variations: continuous/discrete, 1D/2D

Fourier Series:

$$f_T(t) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi nt/T}, \text{ where } c_n = \frac{1}{T}\int_{-T/2}^{T/2} f_t(t)e^{-i2\pi nt/T}dt \; [\text{for } n = 0, \pm1, \pm2, \ldots]$$

Fourier Transform $[t \to \omega]$:

$$F(\mu) = \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\mu t}dt \quad \left[F(\mu) = \sum_{x=0}^{M-1} f(x)e^{-i2\pi\mu x/M}\right]$$

Inverse Fourier Transform $[\omega \to t]$:

$$f(t) = \mathcal{F}^{-1}[F(\mu)] = \int_{-\infty}^{\infty} F(\mu)e^{i2\pi\mu t}d\mu \quad \left[f(x) = \frac{1}{M}\sum_{\mu=0}^{M-1} F(\mu)e^{-2\pi\mu x/M}\right]$$

Fourier/frequency spectrum: $|F(\mu)| = \sqrt{R(\mu)^2 + I(\mu)^2}$

2D Fourier transform (continuous & discrete):

$$F(\mu, \nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t, z)e^{-i2\pi(\mu t + \nu z)}dtdz \quad F(\mu, \nu) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)e^{-i2\pi(\mu x/M + \nu y/N)}$$

2D IFT:

$$f(t, z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu, \nu)e^{i2\pi(\mu t + \nu z)}d\mu d\nu \quad f(x, y) = \frac{1}{MN}\sum_{\mu=0}^{M-1}\sum_{\nu=0}^{N-1} F(\mu, \nu)e^{i2\pi(\mu x/M + \nu y/N)}$$

Periodicity of the DFT: $F(\mu, \nu) = F(\mu + k_1M, \nu + k_2N)$; $f(x, y) = f(x + k_1M, y + k_2N)$

- Periodic with periods M, N

Convolution (associative, commutative, distributive):

$$(f * h)(t) = \int_{-\infty}^{\infty} f(\tau)h(t - \tau)d\tau \quad \left[\sum_{m=0}^{M-1} f(m) \cdot h(x - m)\right]$$

$\longrightarrow$ **Convolution Theorem**: $\mathcal{F}[f * h] = \mathcal{F}[f] \cdot \mathcal{F}[h]$, $\mathcal{F}[f \cdot h] = \mathcal{F}[f] * \mathcal{F}[h]$

- 2D convolution:

$$[f * h](t, z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t, \xi)h(t - \tau, z - \xi)d\tau d\xi \quad [f * h](x, y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$$

Fourier Transform - Additional Properties

- $\mathcal{F}[\delta(t, z)] = 1$
- $F(0, 0) = MNf_{\text{mean}}$
- **Translation Properties**:
  1. $f(x, y)e^{i2\pi(\mu_0 x/M + \nu_0 y/M)} \Longleftrightarrow F(\mu - \mu_0, \nu - \nu_0)$
  2. $f(x - x_0, y - y_0) \Longleftrightarrow F(\mu, \nu)e^{-i2\pi(\mu x_0/M + \nu y_0/N)}$
- **Conjugate Symmetric Property**: If $f(x, y)$ is real, then $\overline{F(\mu, \nu)} = F(-\mu, -\nu)$
- Polar coordinates: $f(r, \theta + \theta_0) = F(w, \phi + \theta_0)$

($*$) Separability of the DFT: $F(\mu, \nu) = \sum_{x=0}^{M-1} F(x, \nu)e^{-2\pi i\mu x/M}$ [sum of 1D DFTs]

4.2 Frequency-Domain Filtering

Motivations:

- Low-frequency components correspond to slowly-varying regions (i.e. smooth regions); high-frequency components correspond to fast-varying components (e.g. noise, edges)
- Filtering in frequency domain can be faster than in spatial domain
- Can express symmetric linear spatial filters as frequency-domain filters

Frequency Filtering:

$$\boxed{g(x,y) = (-1)^{x+y} \cdot \text{Real}\left\{\text{IDFT}\left[H(\mu,\nu)\cdot\mathcal{F}\left[(-1)^{x+y}f(x,y)\right]\right]\right\}}$$

- $\mathcal{F}\left[f(x,y)(-1)^{x+y}\right] = F(\mu - M/2, \nu - N/2)$ [same for H], shifts $(0,0)$ from corner to center
- Lowpass (smoothing)/highpass (sharpening): Bandwidth D_0 [larger $\rightarrow$ less filtering]
- $H_{\text{highpass}} = 1 - H_{\text{lowpass}}$; $H_{BR} = 1 - H_{BP}$, $H_{NR} = 1 - H_{NP}$
- Notch reject filters: $H_{NR} = \prod_{k=1}^{N} H_{NR}(k)(\mu,\nu) H_{NR}(-k)(\mu,\nu)$, for spatially dependent noise

2D continuous Laplacian filter: $\mathcal{F}[\Delta f] = -4\pi^2(\mu^2+\nu^2)F(\mu,\nu)$

- Using that: $f(x,y) = \int\int F(\mu,\nu)e^{2\pi i(\mu x+\nu y)}d\mu d\nu \implies \frac{\partial^n}{\partial x^n} f = (2\pi i\mu)^n \int\int F(\mu,\nu)e^{2\pi i(\mu x+\nu y)}d\mu d\nu$

Homomorphic filtering: $f(x,y) = i(x,y)\cdot r(x,y) \implies z(x,y) := \ln[f(x,y)] = \ln[i(x,y)] + \ln[r(x,y)] \implies \mathcal{F}[z(x,y)] = \mathcal{F}[\ln[i(x,y)]] + \mathcal{F}[\ln[r(x,y)]]$

- $S = H\cdot Z = H\cdot F_i + H\cdot F_r \implies s = i' + r' = \ln(g(x,y)) \implies g = e^s = e^{i'}e^{r'}$

5 Image Restoration

Image Degradation Model:

$$\boxed{g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \Longleftrightarrow G(\mu,\nu) = H(\mu,\nu)F(\mu,\nu) + N(\mu,\nu)}$$

- g [output] $= h$ [degradation] $*f$ [original] $+\eta$ [noise]

Estimation of Noise Parameters:

1. Find a subregion S of the image with relatively constant background intensity
2. Generate the histogram of S and compare it with known probability density functions
   - For impulse noise: instead, look at probabilities of black & white pixels, respectively
3. Estimate the mean and variance $\longrightarrow$ use to estimate the noise parameters

Image Restoration with Spatial Filtering:

For spatially-invariant noise (noise is indep. of coords (x,y), uncorrelated with image $f(x,y)$):

- Arithmetic, geometric mean filters
- Order-statistic: Median, min/max, midpoint (avg of min/max), alpha-trimmed (average of subset of neighborhood; delete $d/2$ highest & lowest intensities; extremes: arith mean, median)
- *Harmonic mean filter*: $\hat{f}(x,y) = mn/\left[\sum_{s,t} 1/g(s,t)\right]$
  - Good for salt noise ($1/g(s,t)$ small), but not pepper noise ($1/g(s,t)$ large)
- *Contraharmonic mean filter*: $\hat{f}(x,y) = \left[\sum_{s,t} g(s,t)^{Q+1}\right] / \left[\sum_{s,t} g(s,t)^{Q}\right]$
  - $Q > 0$ for pepper noise; $Q < 1$ for salt noise

Image Restoration with Frequency-Domain Filtering Spatially-dependent periodic noise:

$$\mathcal{F}\left[\sin(2\pi\mu_0 x + 2\pi\nu_0 y)\right] = \frac{iMN}{2}\left[\delta(\mu+\mu_0,\nu+\nu_0) - \delta(\mu-\mu_0,\nu-\nu_0)\right]$$

- Shifted: impulses at $(\mu/2-\mu_0, \nu/2-\nu_0), (\mu/2+\mu_0, \nu/2+\nu_0) \rightarrow$ use notch reject filter to remove

Estimating the Degradation Function

1. By observation: Find subimage s where $\eta \approx 0$, use sharpening filter to obtain unblurred subimage $\hat{f}$ (within s), use $H_s = G_s/\hat{f}$ to approximate H for entire image
2. By experimentation: Given source used to acquire degraded image, obtain degradation g of a small dot of light $f = A\delta(x,y)$; use F, G to compute degradation f'n H
3. By modeling: based on physical parameters

   Blurring from linear motion:

   $$g(x,y) = \int_0^T f(x - x_0(t), y - y_0(t))dt \implies G(\mu,\nu) = F(\mu,\nu)\int_0^T e^{2\pi i(\mu x_0(t)+\nu y_0(t))}dt$$

   For uniform linear motion $[x_0(t) = at/T, y_0(t) = bt/T]$:

   $$H = \frac{T}{\pi(\mu a + \nu b)}\sin\left(\pi(\mu a+\nu b)\right)e^{-i\pi(\mu a+\nu b)}$$

Frequency-Domain Filtering

- Inverse filtering: $\hat{F}(\mu,\nu) = G(\mu,\nu)/H(\mu,\nu)$, assumes η negligible
  - Approximation suffers when noise is significant (divide by small H) $\rightarrow$ use a lowpass filter to suppress high-frequency noise
- Wiener/min-MSE filtering: Minimizes $\left|f - \hat{f}\right|$ MSE [average over all (x,y)]
  - Assumes $\hat{F}_{opt} = WG$ for some W linear (want to find), either F or N mean 0, we know H, noise η & image f are uncorrelated
  - Plancherel's theorem:

    $$\int_{\mathbb{R}}\int_{\mathbb{R}} |f|^2\,dxdy = \int_{\mathbb{R}}\int_{\mathbb{R}} |F|^2\,d\mu d\nu \qquad \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|f|^2 = \frac{1}{MN}\sum\sum|F|^2$$

  - Rewrite filters $W(\mu,\nu) = R_W(\mu,\nu) + iI_W(\mu,\nu)$, take partials of MSE w.r.t. R_W, I_W
  - Final result:

    $$W = \frac{1}{H}\cdot\frac{|H|}{|H|^2 + S_\eta/S_F} \text{ [approx. } S_\eta/S_f \text{ by } k;\ k=0 \text{ gives direct inverse filter]}$$

- Constrained least-squares: find $\hat{f} = \min_f \gamma\,||\Delta f||_2^2 + ||g - h*f||_2^2$
  - Unconstrained reformulation of orig. constrained optimization:

    $$\min_f\ ||\Delta f||_2^2 \text{ subject to } \left|\left|g - h*\hat{f}\right|\right|_2^2 = ||\eta||_2^2$$

    - Have a constraint; "best solution" is $\hat{f}$ maximally smooth [min Laplacian 2-norm]
  - Same method as prev. (Plancherel's + minimize $\hat{F}$ over real/imaginary separately)
  - Final result (P: FT of Laplacian):

    $$\hat{F}(\mu,\nu) = \left[\frac{1}{H}\cdot\frac{|H|}{|H|^2 + \gamma|P|^2}\right]G$$

  - Connection with mean/variance:

    $$||\eta||^2 = \sum_{x=1}^{M}\sum_{y=1}^{N}\underbrace{\left[(\eta(x,y)-\bar{\eta})^2 + 2\eta(x,y)\bar{\eta} - \bar{\eta}^2\right]}_{\eta(x,y)^2} = MN(\sigma_\eta^2 + \bar{\eta}^2)$$

6 Color Image Processing

RGB: $f(x,y) = \vec{c}(x,y) = [c_R(x,y), c_G(x,y), c_B(x,y)]$, point in Cartesian space; pixel depth: total # bits/pixel

- Per-component vs vector-based processing

HSI:

1. Intensity: $I = \frac{1}{3}(c_R + c_G + c_B)$, $\in [0,1]$
2. Saturation: $S = 1 - \min\{c_r, c_g, c_B\}/I$, $\in [0,1]$ (0=gray, 1=pure)
3. Hue (angle $\in [0,360]$):

$$\theta = \arccos\left(\frac{(c_R - c_G) + (c_R - c_B)}{2\sqrt{(c_R-c_B)^2 + (c_R-c_B)(c_G-c_B)}}\right)$$

Can use grayscale processing techniques (histogram eq., smoothing) on intensity channel only for HSI $\rightarrow$ more efficient, less unwanted color alterations compared to RGB per-component

- Complementary color: $H' = 180 + H$, liket grayscale negative, use for color image completion