Computational Sensing and Imaging Notebook

Junchi Wang

May 2025

1 Computational Sensing & Imaging

1.1 Computational Imaging Overview

Computational imaging is the joint design of **optics capture** and **computational algorithms** to create novel systems.

1.2 Imaging Model

Let $Ax = b$, where:

- x: scene
- A: CCDs and lens
- b: captured image

To reconstruct the scene:

$$x = A^{-1}b$$

1.3 Wrapped Sample

Example:

$$\begin{aligned} \text{threshold} &= 64 \\ \text{raw\_sample} &= 70 \\ \text{wrapped\_sample} &= 70 - 128 = -58 \end{aligned}$$

The wrapped sample causes folding at the threshold, leading to ambiguity which needs to be resolved in post-processing.

2 Modulo Sampling and Recovery

2.1 Key Theorem

Let $f(t)$ be a bandlimited function with no frequencies higher than B_0 (rad/s). A sufficient condition for perfect recovery of $f(t)$ from its modulo samples $y_k = \mod(f(t_k))$, with $t = kT,\ k \in \mathbb{Z}$, is:

$$T \leq \frac{1}{2B_0}$$

In other words, the sampling still satisfies the Nyquist theorem, even if the samples are wrapped.

2.2 Quantization and Resolution

- N: quantization levels (resolution N)
- Modulo samples reduce resolution
- After recovery: resolution becomes $2N$

2.3 Matrix Shapes: Fat and Tall

- **Tall matrix**: A with more rows than columns, so knowns > unknowns
- **Fat matrix**: A with more columns than rows, so unknowns > knowns

$$\text{Tall: } Ax = b, \quad \text{Fat: } Ax = b$$

3 Sampling: 1D, 2D, 3D Voxel Acquisition

3.1 Uniform Sampling

Sample every T_s seconds:

$$f[m] = f(mT_s)$$

Fourier Domain:

$$\hat{f}_{\text{us}}(\omega) = \sum_{n \in \mathbb{Z}} f(\omega + n\Omega_s), \quad \Omega_s = \frac{2\pi}{T_s}$$

3.2 Definition: Bandlimitedness

A function is Ω-bandlimited if the largest frequency component in its Fourier transform is Ω, i.e.,

$$\hat{f}(\omega) = 0 \quad \text{for } |\omega| > \Omega$$

To avoid aliasing: $\Omega_s \gg 2\Omega$

3.3 Nyquist-Shannon Sampling Theorem

If a function $f(t)$ contains no frequency higher than Ω (rad/sec), it can be perfectly reconstructed from its samples spaced less than:

$$T < \frac{\pi}{\Omega}$$

3.4 Sampling Dimensions

- 1D Sampling: Audio
- 2D Sampling: Image
- 3D Sampling: Video (voxels)

3.5 Analog-to-Digital Converters (ADC)

$$f(t) \xrightarrow{\text{sampling}} f[m] \xrightarrow{\text{quantization}} \hat{f}[m]$$

3.6 Sampling Methods

- Point-wise sampling (theoretical)
- Average sampling (practical):

$$f_\varphi(mT) = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)\varphi(t - mT)dt = (f * \varphi)(mT)$$

- One-bit sampling
- Modulo sampling (new approach)

Denoising: Often necessary after sampling.

One-bit Sampling:

- Higher sample rate $\Rightarrow$ more power
- Trade-off between sample rate, resolution, and quantization noise

4 Optics Refresher

4.1 Wave Nature of Light

Light exhibits both wave and particle behavior.

$$f = \frac{c}{\lambda}$$

where:

- f: frequency
- λ: wavelength
- c: speed of light

4.2 Photon Energy

$$E = hf = \frac{hc}{\lambda}$$

where h is Planck's constant.

4.3 Snell's Law (Refraction)

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$

where n is the refractive index.

4.4 Thin Lens Equation

$$\frac{1}{f} = \frac{1}{s_1} + \frac{1}{s_2}$$

where:

- f: focal length
- s_1: distance from object to lens
- s_2: distance from lens to sensor

5 Pipeline of Digital Image Formation

- **Camera Irradiance (Physics)**

$$\text{Optics} \rightarrow \text{Aperture} \rightarrow \text{Shutter}$$

- **Sensor (CMOS/CCD)**

$$\text{Sensor} \rightarrow \text{Gain (ISO)} \rightarrow \text{ADC} \rightarrow \text{RAW}$$

- **DSP**

$$\text{Demosaic} \rightarrow \text{Sharpen}$$

$$\text{White Balance} \rightarrow \text{Gamma Correction} \rightarrow \text{Compression} \rightarrow \text{JPEG}$$

6 Modelling: Forward vs Inverse Problems

6.1 Forward Process

The forward model is typically easy:

$$y(t) = T[f(t)] + n(t)$$

Where:

- $f(t)$: scene (true signal)
- $y(t)$: observation
- $T[\cdot]$: transformation/operator
- $n(t)$: noise

6.2 Inverse Process

This involves recovering the original signal $f(t)$ from the observed $y(t)$, which is often ill-posed or difficult.

6.3 LTI Systems

Consider a Linear Time-Invariant (LTI) system:

$$y(t) = h(t) * f(t)$$

6.3.1 Discrete Representation

$$y(t) = \sum_i h(i) \cdot f(t-i)$$

or in matrix form:

$$\vec{y} = H\vec{f}$$

Where H is the transformation matrix.

6.4 Applications

- Deconvolution
- Denoising
- Superresolution
- Phase retrieval
- Inpainting

7 Deconvolution

7.1 Problem Formulation

$$g(t) = f(t) * h(t) + n(t)$$

7.2 Fourier Domain Representation

Apply the Fourier transform:

$$G(\omega) = F(\omega) \cdot H(\omega)$$

To recover $F(\omega)$, one possible solution:

$$F(\omega) = \frac{G(\omega)}{H(\omega)}$$

7.3 Issue with Direct Inversion

If $H(\omega)$ has low values at high frequencies, then $F(\omega)$ will have large components in those frequencies, amplifying noise.

7.4 Wiener Filter

To address this, the Wiener filter is used:

$$H'(\omega) = \frac{S_{ff}(\omega)H(\omega)}{S_{ff}(\omega)|H(\omega)|^2 + S_{nn}(\omega)}$$

Where:

- $H'(\omega)$: recovery transfer function
- $S_{ff}(\omega)$: power spectral density of the signal
- $S_{nn}(\omega)$: power spectral density of the noise

7.5 Recovery Process

$$F(\omega) \xrightarrow{\cdot H(\omega)} G(\omega) \xrightarrow{\cdot H'(\omega)} \hat{F}(\omega) \Rightarrow \text{Recovery}$$