Review of Sampling & Linear Systems

EE367/CS448I: Computational Imaging and Display
stanford.edu/class/ee367
Lecture 5

Gordon Wetzstein
Stanford University
What’s a Discrete Image?

- continuous 2D visual signal on sensor: \( i(x,y) \)

- integration over pixels:
  \[
  \tilde{i}(x,y) = i(x,y) \ast \left( \text{rect} \left[ \frac{x}{w} \right] \cdot \text{rect} \left[ \frac{y}{h} \right] \right)
  \]

(detector footprint modulation transfer function, Boreman 2001)
What’s a Discrete Image?

- continuous 2D visual signal on sensor: \( i(x,y) \)

- integration over pixels: 
  \[
  \tilde{i}(x,y) = i(x,y) \ast \left( \text{rect} \left[ \frac{x}{w} \right] \cdot \text{rect} \left[ \frac{y}{h} \right] \right)
  \]
  (detector footprint modulation transfer function, Boreman 2001)

- discrete sampling: 
  \[
  E[i,j] = \text{sample} \left( \tilde{f}(x,y) \right) = \tilde{f}(x,y) \cdot \sum_m \sum_n \delta(i,j)
  \]
  (in irradiance \( \frac{W}{m^2} \))
Fourier Transform

- any continuous, integrable, periodic function can be represented as an infinite sum of sines and cosines:

\[ f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)e^{2\pi i \xi x} \, d\xi \quad \leftrightarrow \quad \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i \xi x} \, dx \]

- most important for us: discrete Fourier transform

\[ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}[k]e^{2\pi i kn/N} \quad \leftrightarrow \quad \hat{x}[k] = \sum_{n=0}^{N-1} x[n]e^{-2\pi i kn/N} \]

- convolution theorem (critical):

\[ x * g = F^{-1}\left\{ F\{x\} \cdot F\{g\} \right\} \]
Discrete Fourier Transform

- What is this?
Discrete Fourier Transform

- What is this?
Filtering – Low-pass Filter

- low-pass filter: convolution in primal domain

\[ b = x \ast c \]
Filtering – Low-pass Filter

- low-pass filter: multiplication in frequency domain

\[ F\{b\} = F\{x\} \cdot F\{c\} \]
Filtering – Low-pass Filter

- low-pass filter: hard cutoff

\[ F\{b\} = F\{x\} \cdot F\{c\} \]
Filtering – Low-pass Filter

- Bessel function of the first kind or “jinc”
Filtering – Low-pass Filter

- hard frequency filters often introduce ringing
Filtering – High-pass Filter

- sharpening (possibly with ringing, but don’t see any here)
Filtering – Unsharp Masking

- sharpening (without ringing): unsharp masking, e.g. in Photoshop

\[ b = x \ast (\delta - c_{lowpass\_gauss}) = x - x \ast c_{lowpass\_gauss} \]

or

\[ b = x \ast (\delta + c_{highpass}) = x + x \ast c_{highpass} \]
Filtering – Unsharp Masking

- sharpening (without ringing): unsharp masking, e.g. in Photoshop
Filtering – Band-pass Filter
Filtering – Oriented Band-pass Filter

- edges with specific orientation (e.g., hat) are gone!
Optical Filtering with Fourier Optics

- can do all of this optically (with coherent light)!
- Fourier optics – not part of this course

"Input" plane, containing one of the two functions to be cross-correlated, $f(x,y)$, say.

Multiplicative transmission mask, containing FT $G(k_x,k_y)$ of 2nd function, $g(x,y)$.

FT of $f(x,y)$ i.e. $F(k_x,k_y)$ is formed in this plane.

Correlation of $f(x,y)$ and $g(x,y)$ appears in this plane.

http://en.wikipedia.org/wiki/Fourier_optics
Image Downsampling (& Upsampling)

- best demonstrated with “high-frequency” image
- that’s just resampling, right?
re-sample image: I(1:4:end,1:4:end) in Matlab
something is wrong - aliasing!
need to low-pass filter image first!
need to low-pass filter image first!
first: filter out high frequencies ("anti-aliasing")
then: then re-sample image: I(1:4:end,1:4:end)
Image Downsampling (& Upsampling)

- “anti-aliasing” \(\rightarrow\) **before** re-sampling, apply appropriate filter!

- how much filtering? Shannon-Nyquist sampling theorem:

\[
f_s \geq 2 f_{max}
\]
no anti-aliasing

with anti-aliasing
Examples of Aliasing: Temporal Aliasing

- wagon wheel effect (temporal aliasing)
- sampling frequency was lower than $2f_{\text{max}}$
Examples of Aliasing: Temporal Aliasing

- wagon wheel effect (temporal aliasing):

youtube.com/watch?v=jH9uGkEOmA
Examples of Aliasing: Sampling on Sensor

- point source on focal plane maps to PSF
Examples of Aliasing: Sampling on Sensor

- PSF must be larger than 2*pixel size!
Other Forms of Aliasing

- photography – optical AA filter removed ("hot rodding" camera)
Other Forms of Aliasing

• photography – optical AA filter removed ("hot rodding" camera)

without AA filter

with AA filter (standard)
Other Forms of Aliasing

- photography – optical AA filter removed (“hot rodding” camera)

without AA filter

with AA filter (standard)
Lens as Optical Low-pass Filter

- away from focal plane: out of focus blur
Lens as Optical Low-pass Filter

- shift-invariant convolution

focal plane
Lens as Optical Low-pass Filter

diffraction-limited PSF of circular aperture (aka “Airy” pattern):

\[ b = c \ast x \]

sharp image

measured, blurred image

point spread function (PSF): c
Deconvolution – Next Class!

- given measurements $b$ and convolution kernel $c$, what is $x$?

$$x \ast c = b$$
Overview of Terms

- point spread function (PSF) = blur kernel
- optical transfer function (OTF) – Fourier transform of PSF
- modulation transfer function (MTF) – magnitude of OTF

\[
F \{ PSF \} = OTF = MTF \cdot e^{i\phi}
\]
Sampling – Quick Summary

• Shannon-Nyquist theorem: always sample signal at a sampling rate $\geq 2$*highest frequency of signal!

• if Shannon-Nyquist is violated, aliasing occurs

• aliasing cannot be corrected digitally in post-processing (see optical anti-aliasing filter)

• PSF is usually a low-pass filter 😞
Matrices and Linear Systems - Review

• basic linear algebra, review if necessary!

• stanford.edu/ee263 – lecture slides and recorded lectures online

• quick summary now
most computational imaging problems are linear
geometric optics approximation of light is linear in intensity
not necessarily true for wave-based models (e.g. interference, phase retrieval, ...), but we don’t cover these
Matrices and Linear Systems - Review

- most computational imaging problems are linear

\[ b = Ax \]

- blurry, noisy, or otherwise corrupted measurements
- latent (unknown) image
- system matrix
Matrix Properties

• common problem: given $b$, what can I hope to recover?

• answer: analyze matrix properties: condition number, rank, characterize range space somehow!

\[ b = Ax \]
Matrix Properties

- singular value decomposition (SVD): \( A = U \Sigma V^* \)

- matrix condition number: \( \kappa(A) = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A)} \) (1 is the best)

- rank(\( A \)): number of independent columns = number of nonzero singular values
Matrix Properties

- singular value decomposition (SVD): \[ A = U \Sigma V^* \]

- if \( A \) square, eigen decomposition: \[ A = U D U^* \]

- in general: \[ A^* A = (V \Sigma^* U^*) (U \Sigma V^*) = V \Sigma^* \Sigma V^* \]

→ so eigen values of \( A^* A \) are singular values of \( A \) squared
Matrix Properties: Useful Example

• given $b$, what can I hope to recover?

• example: convolution with PSF $c$

\[ b = c \ast x = Cx \]
Useful Example

- Matrix form of convolution: $b = c \ast x \rightarrow b = Cx$
- $C$ is circulant matrix (for circular boundary conditions)
  
  $C = \begin{bmatrix}
  c_0 & c_{n-1} & \cdots & c_2 & c_1 \\
  c_1 & c_0 & c_{n-1} & \cdots & c_2 \\
  \vdots & c_1 & c_0 & \ddots & \vdots \\
  c_{n-2} & \cdots & \cdots & c_{n-1} \\
  c_{n-1} & c_{n-2} & \cdots & c_1 & c_0
  \end{bmatrix}$

- Eigen-decomposition of circulant matrix: $C = UDU^* = F^{-1}DF$
- Matrix of eigenvectors is discrete Fourier transform!
- Eigenvalues: $\text{diag}(D) = \hat{c}$
  
  $b = c \ast x = Cx = F^{-1}\text{diag}(\hat{c})Fx = F^{-1}\left\{F \{c\} \cdot F \{x\}\right\}$
Useful Example

- matrix $C$ is rank-deficient
- i.e. when convolution kernel is low-pass filter!

$\text{abs}(F\{c\}) \rightarrow$ modulation transfer function (MTF)
Useful Example

- matrix $C$ is rank-deficient
- i.e. when convolution kernel is low-pass filter!

$$\text{abs}(F\{c\}) \rightarrow \text{modulation transfer function (MTF)}$$

$$\text{sort}(\text{MTF}) = \text{eigenvalues of } C^T C$$
Useful Example

- matrix $C$ is rank-deficient
- i.e. when convolution kernel is low-pass filter!

$$\text{abs}(F\{c\}) \rightarrow \text{modulation transfer function (MTF)}$$

$$\text{sort}(\text{MTF}) = \text{eigenvalues of } C^T C$$

**Noise floor of camera**

signal-to-noise-ratio (SNR) is below threshold
Useful Example

- matrix $C$ is rank-deficient
- i.e. when convolution kernel is low-pass filter!

$$\text{abs}(F\{c\}) \rightarrow \text{modulation transfer function (MTF)}$$

$\text{sort}(\text{MTF}) = \text{eigenvalues of } C^T C$

noise floor of camera (e.g. high ISO)

signal-to-noise-ratio (SNR) is below threshold
Useful Example

- matrix $C$ is rank-deficient
- i.e. when convolution kernel is low-pass filter!

$\text{abs}(F\{c\}) \rightarrow \text{modulation transfer function (MTF)}$

$\text{sort}(\text{MTF}) = \text{eigenvalues of } C^T C$

- signal-to-noise-ratio (SNR) is below threshold
- noise floor of camera (cooled, scientific CCD)
Linear Systems

• other common problem: given $b$, what is $x$?

• answer: invert matrix?

\[ b = Ax \quad \rightarrow \quad x_{est} = A^{-1}b \]
Linear Systems

• other common problem: given $b$, what is $x$?

• answer: invert matrix – generally not!

$$b = Ax \quad \Rightarrow \quad x_{est} = A^{-1}b$$
Linear Systems

• problem 1: matrix inverse only defined for square, full-rank matrices – most imaging problems are NOT!

• problem 2: most imaging problems deal with really big matrices – couldn’t compute inverse, even if there was one!

• solution: iterative (convex) optimization
Linear Systems

• case 1: over-determined system = more measurements than unknowns

\[ A \in \mathbb{R}^{m \times n}, \quad m > n \]

• formulate least-squared error objective function:

\[
\minimize_x \frac{1}{2} \left\| b - Ax \right\|_2^2 \quad \left\| r \right\|_2^2 = \sum_i r_i^2, \quad r = b - Ax
\]
Linear Systems

• least squares solution: gradient of objective = 0

• gradient:

\[
\nabla_x \frac{1}{2} \|b - Ax\|^2 = \nabla_x \frac{1}{2} (b^Tb - 2b^TAx + x^TA^TAx) = A^TAx - A^Tb
\]

• equate to zero – normal equations:

\[
A^TAx = A^Tb
\]
Linear Systems

• closed-form solution to normal equations:

\[ A^T Ax = A^T b \quad \rightarrow \quad x_{est} = \left( A^T A \right)^{-1} A^T b \]

• rarely applicable, because again A is big and usually not full rank

• regularized solution

\[ x_{est} = \left( A^T A + \lambda I \right)^{-1} A^T b \]

(always full rank, but still too big to directly invert)
Linear Systems – Gradient Descent

• solve with iterative method, easiest one: gradient descent

\[
\left( A^T A + \lambda I \right) x = A^T b
\]

• use the negative gradient of objective as descent direction at iteration \( k \), with step length \( \alpha \)

\[
x^{(k+1)} = x^{(k)} - \alpha \nabla_x = x^{(k)} - \alpha \tilde{A}^T \left( \tilde{A} x^{(k)} - \tilde{b} \right)
\]
Linear Systems – Gradient Descent

- use the negative gradient of objective as descent direction at iteration $k$, with step length $\alpha$

\[
x^{(k+1)} = x^{(k)} - \alpha \nabla_x = x^{(k)} - \alpha \tilde{A}^T (\tilde{A} x^{(k)} - \tilde{b})
\]

- for large-scale problems, implement as function handles!
Linear Systems – Gradient Descent

- back to convolution example (here just $C$, not $C^T C$)

$$x^{(k+1)} = x^{(k)} - \alpha C^T (Cx^{(k)} - b)$$

- efficient implementation using convolution theorem:

$$x^{(k+1)} = x^{(k)} - \alpha F^{-1} \left\{ \hat{c}^* \cdot F \left\{ F^{-1} \left\{ \hat{c} \cdot F \{ x^{(k)} \} \right\} - b \right\} \right\}$$
Next: How to do it right – Deconvolution!

- inverse filtering
- Wiener filtering
- sparse gradients
- ADMM
References and Further Reading

• http://www.imagemagick.org/Usage/fourier/
• wikipedia