Image Processing

Computational Photography

CSE 291

Lecture 3
Announcements

• Assignment 1 is due Apr 8, 11:59 PM
• Assignment 2 will be released Apr 8
  – Due Apr 15, 11:59 PM
Image sensing pipeline
Image processing

• A discipline in which both the input and output of a process are images
  – There are usually other input parameters to the process
Demosaicing

Color filter array (CFA)

Image sensor

Bayer pattern

CFA

Interpolated (lower case) pixel values

Original Image

CCD array with Bayer pattern showing location of white/black transition

Aliased Image
Image processing

• Color spaces
• Gamut mapping
• White balancing and color balancing
Image processing

- Dehazing
- Denoising
  - Bilateral filtering
- Deconvolution

Motion blur and additive noise

Degraded image
Inverse filtering
Wiener filtering
Constrained least squares filtering
Spatial filtering: correlation and convolution (1D)

**Correlation**

(a) 0 0 0 1 0 0 0 0 1 2 4 2 8

(b) 0 0 0 1 0 0 0 0 1 2 4 2 8

(c) 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8

(d) 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8

(e) 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8

(f) 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8

**Convolution**

(i) 0 0 0 1 0 0 0 0 8 2 4 2 1

(j) 0 0 0 1 0 0 0 0 8 2 4 2 1

(k) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(l) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(m) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(n) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

**Correlation result**

(g) 0 8 2 4 2 1 0 0

**Convolution result**

(h) 0 0 0 8 2 4 2 1 0 0 0 0

**Extended (full) correlation result**

(i) 0 0 0 1 0 0 0 0 8 2 4 2 1

(j) 0 0 0 1 0 0 0 0 8 2 4 2 1

(k) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(l) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(m) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

(n) 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1

**Extended (full) convolution result**

(o) 0 1 2 4 2 8 0 0

(p) 0 0 0 1 2 4 2 8 0 0 0 0

---

Illustration of 1-D correlation and convolution of a kernel, \( w \), with a function \( f \) consisting of a discrete unit impulse. Note that correlation and convolution are functions of the variable \( x \), which acts to displace one function with respect to the other. For the extended correlation and convolution results, the starting configuration places the rightmost element of the kernel to be coincident with the origin of \( f \). Additional padding must be used.
Spatial filtering (2D)

2D correlation

\[ w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \]

2D convolution

\[ w(x, y) \circledast f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t) \]
Correlation and convolution (2D)

**FIGURE 3.36**
Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0’s are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of $x$ and $y$. As these variable change, they displace one function with respect to the other. See the discussion of Eqs. (3-45) and (3-46) regarding full correlation and convolution.
Correlation and convolution

- Convolution is commutative and associative, correlation is not

<table>
<thead>
<tr>
<th>Property</th>
<th>Convolution</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutative</td>
<td>$f \ast g = g \ast f$</td>
<td>—</td>
</tr>
<tr>
<td>Associative</td>
<td>$f \ast (g \ast h) = (f \ast g) \ast h$</td>
<td>—</td>
</tr>
<tr>
<td>Distributive</td>
<td>$f \ast (g + h) = (f \ast g) + (f \ast h)$</td>
<td>$f \bullet (g + h) = (f \bullet g) + (f \bullet h)$</td>
</tr>
</tbody>
</table>
Smoothing kernels

\[
\frac{1}{9} \times \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

Average (box kernel)

\[
\frac{1}{4.8976} \times \begin{bmatrix}
0.3679 & 0.6065 & 0.3679 \\
0.6065 & 1.0000 & 0.6065 \\
0.3679 & 0.6065 & 0.3679 \\
\end{bmatrix}
\]

Weighted average (Gaussian kernel)
Smoothing with box kernel

Input image

3x3
11x11
21x21
Smoothing with Gaussian kernel

<table>
<thead>
<tr>
<th>Standard deviation $\sigma$</th>
<th>Percent of total volume under surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39.35</td>
</tr>
<tr>
<td>2</td>
<td>86.47</td>
</tr>
<tr>
<td>3</td>
<td>98.89</td>
</tr>
</tbody>
</table>

Volume under surface greater than $3\sigma$ is negligible
Smoothing with Gaussian kernel

\[ \sigma = 7 \]
\[ 43 \times 43 \]
\[ \sigma = 7 \]
\[ 85 \times 85 \]

Difference
Smoothing with Gaussian kernel

Input image  \( \sigma = 3.5 \)  21x21  \( \sigma = 7 \)  43x43
Border padding

Zero padding when $v = 0$

Constant padding

Replicate padding

Mirror padding
Border padding

- Zero padding
- Mirror padding
- Replicate padding
Derivatives

**Figure 3.50**
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b) Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.
Sharpening filters

**FIGURE 3.52**
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.51(a).
(c) Image sharpened using Eq. (3-63) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.51(b). (Original image courtesy of NASA.)

**FIGURE 3.53**
The Laplacian image from Fig. 3.52(b), scaled to the full [0, 255] range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels correspond to the highest positive value.
Gradient (first derivatives)
Dehazing

- “Single Image Haze Removal Using Dark Channel Prior”, Kaiming He, Jian Sun, and Xiaoou Tang
  – Best paper, CVPR 2009
Dehazing

- “Single Image Dehazing via Multi-Scale Convolutional Neural Networks”, Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang

ECCV 2016
Model of image degradation

• Spatial domain

\[ g(x, y) = h(x, y) \ast f(x, y) + \eta(x, y) \]

Degraded image  Degradation function  Original image  Noise image
Model of image degradation, then restoration

\[ g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \]
Image restoration

1. Remove noise
2. Estimate original image
   - Deconvolution
Noise modeled as different probability density functions

- **Gaussian**: \( p(z) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(z-\bar{z})^2}{2\sigma^2}\right) \)

- **Rayleigh**: \( p(z) = 0.607 \frac{z}{b} \exp\left(-\frac{z^2}{2b^2}\right) \)

- **Erlang (Gamma)**: \( K = \frac{a(b-1)^{b-1}}{(b-1)!} e^{-(b-1)/z} \)

- **Exponential**: \( p(z) = \frac{1}{b-a} \exp\left(-\frac{z-a}{b-a}\right) \)

- **Uniform**: \( p(z) = \begin{cases} 1, & a \leq z \leq b \\ 0, & \text{otherwise} \end{cases} \)

- **Salt-and-pepper**: \( p(z) = \begin{cases} P_s, & 0 < z < V \\ P_p, & V < z < 2^k - 1 \\ 0, & \text{otherwise} \end{cases} \)
Adding noise from different models

Free of noise

Gaussian
Rayleigh
Gamma
Adding noise from different models

Free of noise

Exponential

Uniform

Salt and pepper
Histograms of sample patches

Sample “flat” patches from images with noise

Identify closest probability density function (pdf) match:

- Gaussian
- Rayleigh
- Uniform
Mean filters

- X-ray image
- Additive Gaussian noise
- Arithmetic mean filtered
- Geometric mean filtered
Order-statistic filters

Additive salt and pepper noise

1x median filtered

2x median filtered

3x median filtered
Order-statistic filters

Max filtered

Min filtered
Comparing filters

Additive uniform + salt and pepper noise

Arithmetic mean filtered

Median filtered

Geometric mean filtered

Alpha-trimmed mean filtered
Adaptive filters

Additive Gaussian noise

Geometric mean filtered

Arithmetic mean filtered

Adaptive noise reduction filtered
Adaptive filters

Additive salt and pepper noise

Median filtered

Adaptive median filtered
Bilateral filter

- An edge-preserving low pass filter

\[ g(i, j) = \frac{\sum_{k,l} f(k, l) w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)} \]

- Bilateral weight function

\[ w(i, j, k, l) = \exp \left( -\frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2} \right) \]

Domain kernel  Range kernel
Bilateral filter

Figure 3.20  Bilateral filtering (Durand and Dorsey 2002) © 2002 ACM: (a) noisy step edge input; (b) domain filter (Gaussian); (c) range filter (similarity to center pixel value); (d) bilateral filter; (e) filtered step edge output; (f) 3D distance between pixels.
Model of image degradation

• Spatial domain

\[ g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \]

Degraded image  Degradation function  Original image  Noise image

• Frequency domain

\[ G(u, v) = H(u, v)F(u, v) + N(u, v) \]

Degraded image  Degradation function  Original image  Noise image
Image processing in the frequency domain

Image in spatial domain \( f(x, y) \) \[ \rightarrow \] Fourier transform \[ \rightarrow \] Image in frequency domain \( F(u, v) \)

Frequency domain processing

Image in frequency domain \( G(u, v) \) \[ \rightarrow \] Inverse Fourier transform \[ \rightarrow \] Image in spatial domain \( g(x, y) \)

Jean-Baptiste Joseph Fourier
1768-1830
Continuous Fourier transform

1D

2D

CSE 291, Spring 2020
Unit discrete impulse

1D

2D
Impulse train

1D

\[ s_{\Delta x}(x) \]

2D

\[ s_{\Delta t \Delta z}(t, z) \]
Fourier transform of sampled function and extracting one period

Over-sampled

Under-sampled

1D

Recovered

Imperfect recovery due to interference

2D
Aliasing

1D

2D

Original

Aliasing
Aliasing in real images

Original  Aliasing  No aliasing

Figure 4.19 Illustration of aliasing on resampled natural images. (a) A digital image of size 772 × 548 pixels with visually negligible aliasing. (b) Result of resizing the image to 33% of its original size by pixel deletion and then restoring it to its original size by pixel replication. Aliasing is clearly visible. (c) Result of blurring the image in (a) with an averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)
2D Discrete Fourier Transform
Centering the DFT

In MATLAB, use `fftshift` and `ifftshift`
Centering the DFT

Original

Shifted DFT

DFT (look at corners)

Log of shifted DFT
DFT magnitude of geometrically transformed images

Translated

Rotated about center

Same magnitude as original (invariant to translation)
DFT phase of geometrically transformed images

Original  Translated  Rotated about center
Contributions of magnitude and phase to image formation

Phase

IDFT: Magnitude only (zero phase)

IDFT: Phase only (zero magnitude)

IDFT: Rectangle magnitude and boy phase

IDFT: Boy magnitude and rectangle phase
Filtering using convolution theorem

Filtering in spatial domain using convolution

Expected result

Filtering in frequency domain using product without zero-padding

Wraparound error
Filtering using convolution theorem

Filtering in frequency domain using product with zero-padding

No wraparound error

Zero padding

Fourier transform

Product

Gaussian lowpass filter in frequency domain

Inverse Fourier transform
Filtering using convolution theorem

Filtering in spatial domain using convolution

Filtering in frequency domain using product

Identical results
Filtering in the frequency domain

• Ideal lowpass filter (LPF)
  – Frequency domain
Filtering in the frequency domain

- Ideal lowpass filter (LPF)
  - Spatial domain

\[
H(u,v) \quad h(x,y)
\]
Filtering in the frequency domain

- Gaussian lowpass filter (LPF)
Filtering in the frequency domain

- Butterworth lowpass filter (LPF)
Filtering in the frequency domain

- Ideal LPF
- Gaussian LPF
- Butterworth LPF
Highpass filter (HPF)  
Frequency domain

Ideal HPF

Gaussian HPF

Butterworth HPF
Highpass filter (HPF)
Spatial domain

Ideal HPF  Gaussian HPF  Butterworth HPF
Filtering in the frequency domain

Ideal HPF  Gaussian HPF  Butterworth HPF
Filtering in the frequency domain

1D Lowpass filter
1D Sharpening filter

Frequency domain
Spatial domain

CSE 291, Spring 2020
Filtering in the frequency domain

Lowpass filter  Highpass filter  Offset highpass filter

2D
Bandreject filters

Ideal

Gaussian

Butterworth
Filtering in the frequency domain

- Sharpening filter
Image processing in the frequency domain

Image in spatial domain $f(x,y)$ → Fourier transform $F(u,v)$ → Image in frequency domain $G(u,v)$

Image in spatial domain $g(x,y)$ ← Inverse Fourier transform $G(u,v)$ ← Image in frequency domain $F(u,v)$

Jean-Baptiste Joseph Fourier
1768-1830
Periodic noise

Additive sinusoidal noise

Conjugate impulses

DFT magnitude
Notch reject filters
Notch reject filter

Degraded image

Filter in frequency domain

Conjugate impulses

DFT magnitude

Estimate of original image

Conjugate impulses

Conjugate impulses
Notch reject filter

Degraded image

Filter in frequency domain

DFT magnitude

Estimate of original image
Estimating the degradation function

• Methods
  – Observation
  – Experimentation
  – Mathematical modeling
Estimation of degradation function by experimentation

Impulse of light  Imaged (degraded) impulse
Estimation of degradation function by mathematical modeling
Estimation of degradation function by mathematical modeling

Motion blur model
Image restoration, inverse filtering

Full

Limited to radius of 40

Limited to radius of 70

Limited to radius of 85
Image restoration, Wiener filtering

Inverse filtering

Wiener filtering

Full

Radially limited
Image restoration, constrained least squares filtering

Degraded image  Inverse filtering  Wiener filtering  Constrained least squares filtering

Motion blur and additive noise