Image Filtering

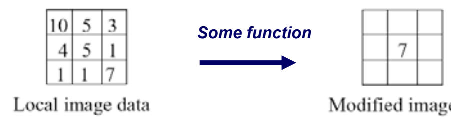
Reading:
–Chapter 7, F&P

Due: Problem Set 1

February 14, 2008

What is image filtering?

- Modify the pixels in an image based on some function of a local neighborhood of the pixels.



Linear Functions

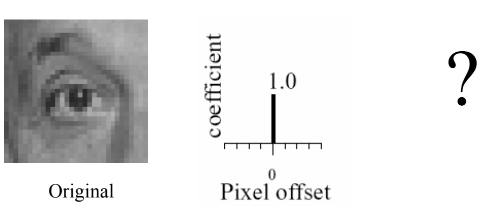
- **Simplest: linear filtering.**

  Replace each pixel by a linear combination of its neighbors.
- **The prescription for the linear combination is called the "convolution kernel".**

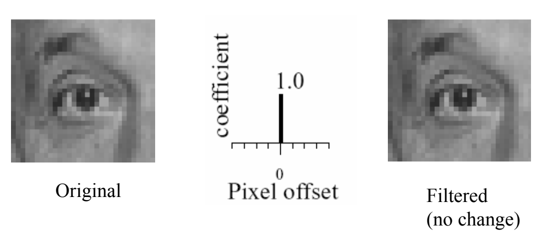
Convolution

$$f[x,y] = I \otimes g = \sum_{k,l} I[x-k,y-l]\,g[k,l]$$

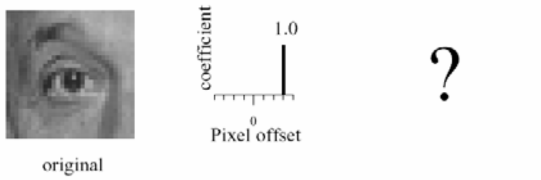
Linear Filtering (warm-up slide)



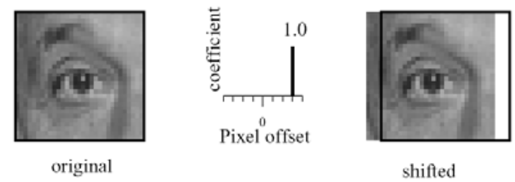
Linear Filtering (warm-up slide)



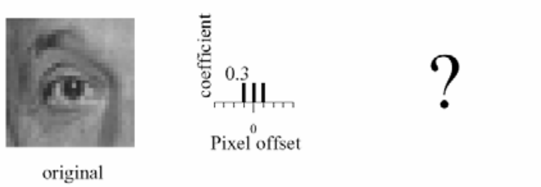
Linear Filtering



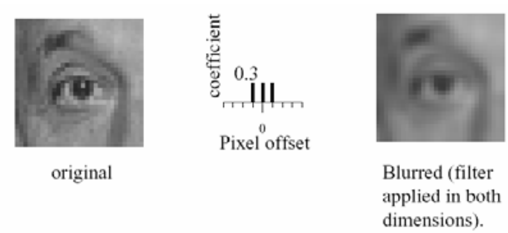
Shifted



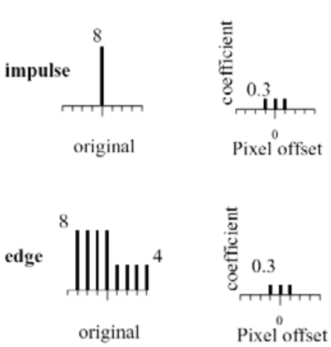
Linear Filtering



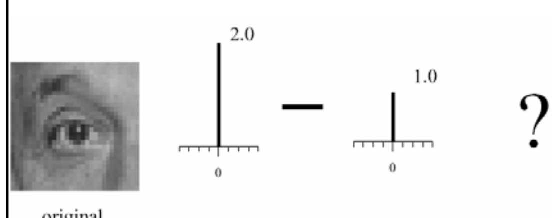
Blurring



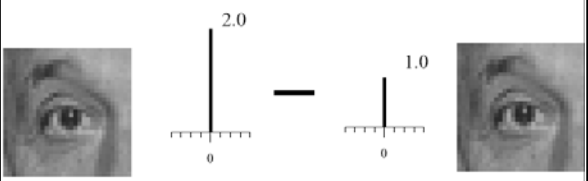
Blur Example



Linear Filtering (warm-up slide)



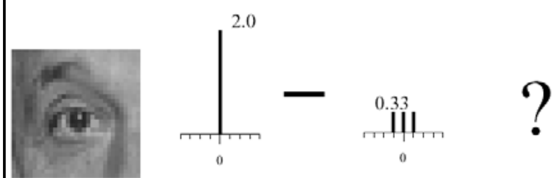
Linear Filtering (no change)



original

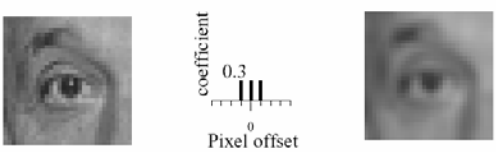
Filtered
(no change)

Linear Filtering



original

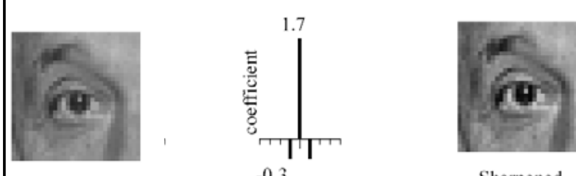
Remember Blurring



original

Blurred (filter
applied in both
dimensions).

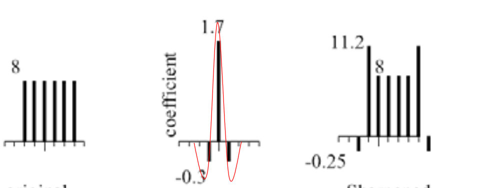
Sharpening



original

Sharpened
original

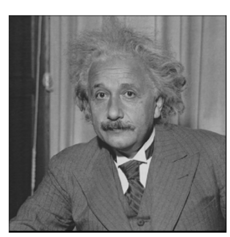
Sharpening Example



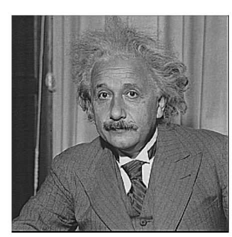
original

Sharpened
(differences are
accentuated; constant
areas are left untouched).

Sharpening



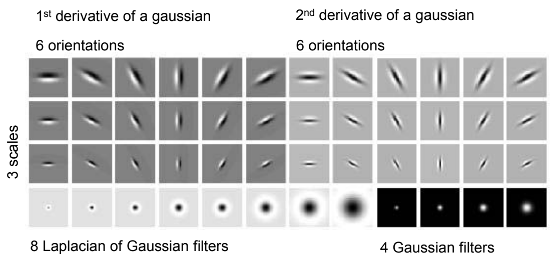
before



after

Oriented Filters

- **Filter bank:**
  - Mix of edge, bar, spot filters at multiple scales and orientations



Linear Image Transformation

- **In analyzing images, it's often useful to make a change of basis.**

$$\vec{F} = U\vec{f}$$

Vectorized image

Transformed image

Fourier Transform, or
Wavelet Transform, or
Steerable Pyramid Transform

Self-inverting Transforms

- Same basis functions are used for the inverse transform

$$\vec{f} = U^{-}\vec{F}$$

$$= U^{+}\vec{F}$$

Transpose and complex conjugate

Example: Fourier Transform

- **Forward Transform**

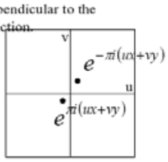
$$F[u,v] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f[x,y]\, e^{-\pi i\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

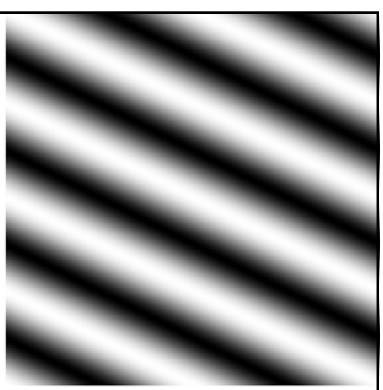
- **Inverse Transform**

$$f[x,y] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F[u,v]\, e^{+\pi i\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

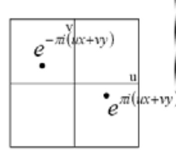
FFT on-line book: http://ccrma.stanford.edu/%7Ejos/mdft/mdft.html

To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

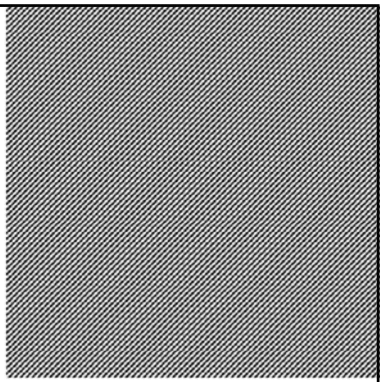


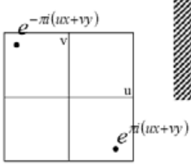


Here u and v are larger than in the previous slide.



And larger still...

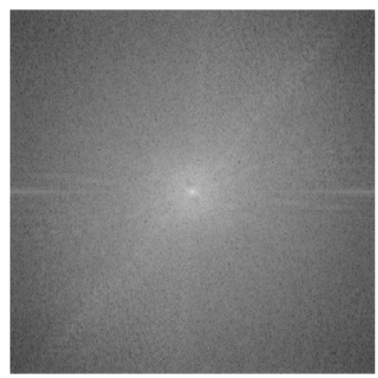




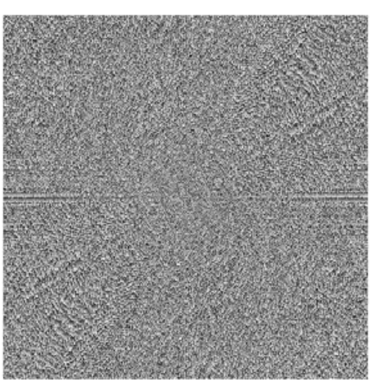
Phase and Magnitude

- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform
- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn't
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

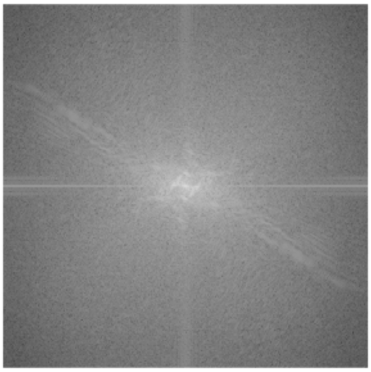
This is the magnitude transform of the cheetah pic



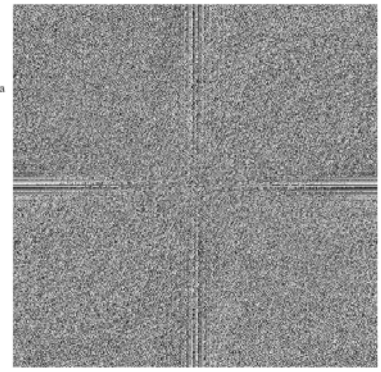
This is the phase transform of the cheetah pic



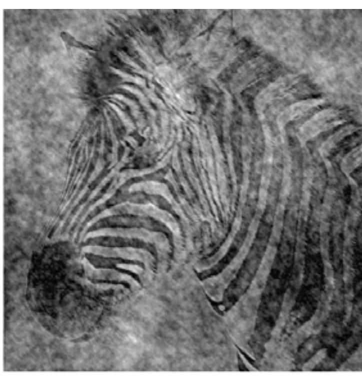
This is the magnitude transform of the zebra pic



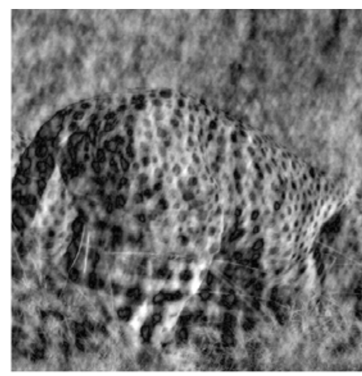
This is the phase transform of the zebra pic



Reconstruction with zebra phase, cheetah magnitude



Reconstruction with cheetah phase, zebra magnitude



Discrete-time, continuous frequency Fourier transform

Many sequences can be represented by a Fourier integral of the form

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}\,d\omega, \qquad (2.133)$$

where

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \qquad (2.134)$$

Oppenheim, Schafer and Buck, Discrete-time signal processing, Prentice Hall, 1999

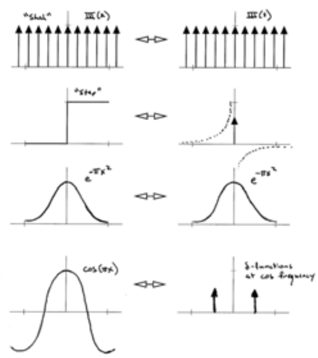
Discrete-time, continuous frequency Fourier transform pairs

TABLE 2.3 FOURIER TRANSFORM PAIRS

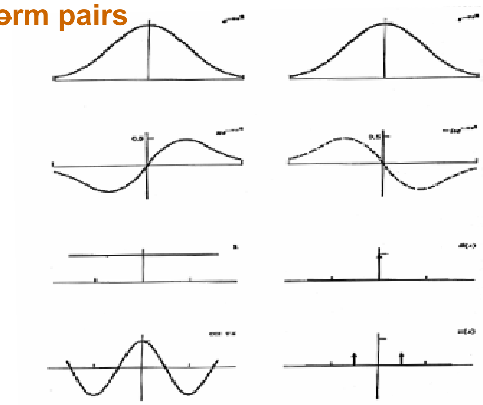
Sequence	Fourier Transform
1. $\delta[n]$	1
2. $\delta[n-n_0]$	$e^{-j\omega n_0}$
3. $1 \quad (-\infty < n < \infty)$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega + 2\pi k)$
4. $a^n u[n] \quad (\lvert a\rvert < 1)$	$\frac{1}{1-ae^{-j\omega}}$
5. $u[n]$	$\frac{1}{1-e^{-j\omega}} + \sum_{k=-\infty}^{\infty} \pi\delta(\omega + 2\pi k)$
6. $(n+1)a^n u[n] \quad (\lvert a\rvert < 1)$	$\frac{1}{(1-ae^{-j\omega})^2}$
7. $\frac{r^n \sin\omega_p(n+1)}{\sin\omega_p} u[n] \quad (\lvert r\rvert < 1)$	$\frac{1}{1-2r\cos\omega_p e^{-j\omega} + r^2 e^{-j2\omega}}$
8. $\frac{\sin\omega_c n}{\pi n}$	$X(e^{j\omega}) = \begin{cases} 1, & \lvert\omega\rvert < \omega_c, \\ 0, & \omega_c < \lvert\omega\rvert \le \pi \end{cases}$
9. $x[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$	$\frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}$
10. $e^{j\omega_0 n}$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega - \omega_0 + 2\pi k)$
11. $\cos(\omega_0 n + \phi)$	$\sum_{k=-\infty}^{\infty} [\pi e^{j\phi}\delta(\omega - \omega_0 + 2\pi k) + \pi e^{-j\phi}\delta(\omega + \omega_0 + 2\pi k)]$

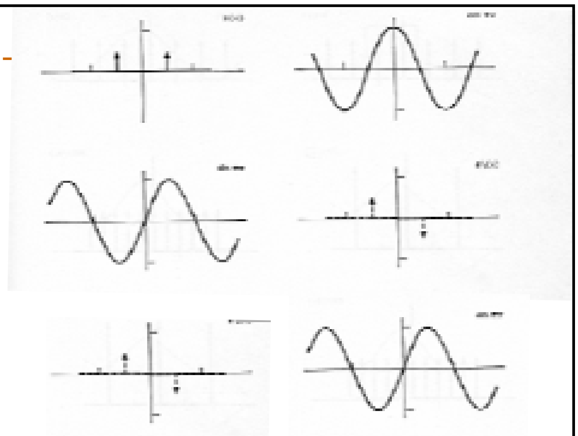
Oppenheim, Schafer and Buck, Discrete-time signal processing, Prentice Hall, 1999

Bracewell's dictionary of Fourier transform pairs



Bracewell's dictionary of Fourier transform pairs





Why is the Fourier domain useful?

- **It tells us the effect of linear convolutions.**
- **There is a fast algorithm for performing the DFT, allowing for efficient signal filtering.**
- **The Fourier domain offers an alternative domain for understanding and manipulating the image.**

Why is the Fourier transform useful?

- ***Convolution theorem*:**
  - **the Fourier transform of the convolution of two functions is the product of their individual Fourier transforms**
- ***Addition Theorem*:**
  - The Fourier transform of the addition of two functions f(x) and g(x) is the **addition of their Fourier transforms** F(s) and G(s).
- ***Shift Theorem*:**
  - A function f(x) shifted along the x-axis by a to become f(x-a) has the Fourier transform $e^{-2\pi ias}F(s)$. **The magnitude of the transform is the same, only the phases change.**
- ***Similarity Theorem*:**
  - For a function f(x) with a Fourier tranform F(s), if the x-axis is scaled by a constant *a* so that we have f(ax), the Fourier transform becomes (1/a)F(s/a). In other words, a "wide" function in the time-domain is a "narrow" function in the frequency-domain.
- ***Modulation Theorem*:**
  - The Fourier transform of a function f(x) multiplied by $\cos(2\pi fx)$ is

    $\frac{1}{2}F(s-f)+\frac{1}{2}F(s+f)$

Fourier transform of convolution

Consider a (circular) convolution of g and h

$$f = g \otimes h$$

Fourier transform of convolution

$f = g \otimes h$

Take DFT of both sides

$$F[m,n] = DFT(g \otimes h)$$

Fourier transform of convolution

$f = g \otimes h$

$F[m,n] = DFT(g \otimes h)$

Write the DFT and convolution explicitly

$$F[m,n] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$$

Fourier transform of convolution

$f = g \otimes h$

$F[m,n] = DFT(g \otimes h)$

$F[m,n] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$

Move the exponent in

$$= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}h[k,l]$$

Fourier transform of convolution

$f = g \otimes h$

$F[m,n] = DFT(g \otimes h)$

$F[m,n] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$

$= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}h[k,l]$

Change variables in the sum

$$= \sum_{\mu=-k}^{M-k-1}\sum_{\upsilon=-l}^{N-l-1}\sum_{k,l} g[\mu,\upsilon]e^{-\pi i\left(\frac{(k+\mu)m}{M}+\frac{(l+\upsilon)n}{N}\right)}h[k,l]$$

Fourier transform of convolution

$f = g \otimes h$

$F[m,n] = DFT(g \otimes h)$

$F[m,n] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$

$= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}h[k,l]$

$= \sum_{\mu=-k}^{M-k-1}\sum_{\upsilon=-l}^{N-l-1}\sum_{k,l} g[\mu,\upsilon]e^{-\pi i\left(\frac{(k+\mu)m}{M}+\frac{(l+\upsilon)n}{N}\right)}h[k,l]$

Perform the DFT (circular boundary conditions)

$$= \sum_{k,l} G[m,n]e^{-\pi i\left(\frac{km}{M}+\frac{\mathrm{ln}}{N}\right)}h[k,l]$$

Fourier transform of convolution

$f = g \otimes h$

$F[m,n] = DFT(g \otimes h)$

$F[m,n] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]h[k,l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}$

$= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\sum_{k,l} g[u-k, v-l]e^{-\pi i\left(\frac{um}{M}+\frac{vn}{N}\right)}h[k,l]$

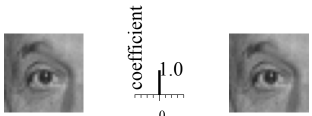
$= \sum_{\mu=-k}^{M-k-1}\sum_{\upsilon=-l}^{N-l-1}\sum_{k,l} g[\mu,\upsilon]e^{-\pi i\left(\frac{(k+\mu)m}{M}+\frac{(l+\upsilon)n}{N}\right)}h[k,l]$

$= \sum_{k,l} G[m,n]e^{-\pi i\left(\frac{km}{M}+\frac{\mathrm{ln}}{N}\right)}h[k,l]$

Perform the other DFT (circular boundary conditions)

$$= G[m,n]H[m,n]$$

Analysis of our simple filters

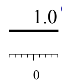


original

Filtered
(no change)

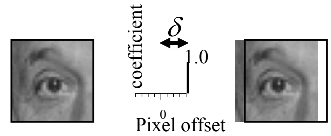
$$F[m] = \sum_{k=0}^{M-1} f[k] e^{-\pi i \left(\frac{km}{M}\right)}$$

$$= 1$$



constant

Analysis of our simple filters

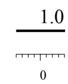


original

shifted

$$F[m] = \sum_{k=0}^{M-1} f[k] e^{-\pi i \left(\frac{km}{M}\right)}$$

$$= e^{-\pi i \frac{\delta m}{M}}$$



Constant magnitude, linearly shifted phase

Convolution versus FFT

- **1D FFT: O(NlogN) computation time, where N is number of samples.**
- **2D FFT: 2N(NlogN), where N is number of pixels on a side**
- **Convolution: K N^2, where K is number of samples in kernel**
- **Say $N=2^{10}$, K=100. 2D FFT: 20 2^{20}, while convolution gives 100 2^{20}**

The END