

Introduction to Computer Vision

CS / ECE 181B

Today

- Handout #4 : Available this afternoon
- Midterm: May 6, 2004
- HW #2 due tomorrow
- Ack: Prof. Matthew Turk for the lecture slides.

Additional Pointers

- See my ECE 178 class web page
<http://www.ece.ucsb.edu/Faculty/Manjunath/ece178>
- See the review chapters from Gonzalez and Woods (available on the 181b web)
- A good understanding of linear filtering and convolution is essential in developing computer vision algorithms.
- Topics I recommend for additional study (that I will not be able to discuss in detail during lectures)--> sampling of signals, Fourier transform, quantization of signals.

April 2004

2

Area operations: Linear filtering

- Point, local, and global operations
 - Each kind has its purposes
- Much of computer vision analysis starts with local area operations and then builds from there
 - Texture, edges, contours, shape, etc.
 - Perhaps at multiple scales
- Linear filtering is an important class of local operators
 - Convolution
 - Correlation
 - Fourier (and other) transforms
 - Sampling and aliasing issues

April 2004

3

Convolution

- The response of a linear shift-invariant system can be described by the **convolution** operation

$$R_{ij} = \sum_{u,v} H_{i-u, j-v} F_{uv}$$

Output image Convolution filter kernel Input image

$$R = H * F = H \otimes F$$

Convolution notations

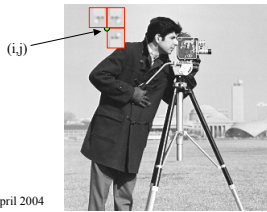
$$R_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} H_{mn} F_{i-m, j-n}$$

April 2004

4

Convolution

- Think of 2D convolution as the following procedure
- For every pixel (i,j) :
 - Line up the image at (i,j) with the filter kernel
 - Flip the kernel in both directions (vertical and horizontal)
 - Multiply and sum (dot product) to get output value $R(i,j)$



April 2004

5

Convolution

$$R_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} H_{mn} F_{i-m,j-n}$$

- For every (i,j) location in the output image R , there is a summation over the local area

$$R_{4,4} = H_{0,0}F_{4,4} + H_{0,1}F_{4,3} + H_{0,2}F_{4,2} + H_{1,0}F_{3,4} + H_{1,1}F_{3,3} + H_{1,2}F_{3,2} + H_{2,0}F_{2,4} + H_{2,1}F_{2,3} + H_{2,2}F_{2,2}$$

$$= -1*222 + 0*170 + 1*149 + -2*173 + 0*147 + 2*205 + -1*149 + 0*198 + 1*221 = 63$$

April 2004

6

Convolution: example

$$y(1,0) = \sum_{k,l} x(k,l)h(1-k,-l) = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & -2 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} = 3$$

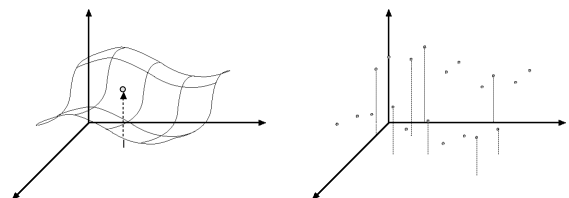
$$y(m,n) = \begin{vmatrix} 1 & 5 & 5 & 1 \\ 3 & 10 & 5 & 2 \\ 2 & 3 & -2 & -3 \end{vmatrix} \quad \text{verify!}$$

April 2004

7

Spatial frequency and Fourier transforms

- A discrete image can be thought of as a regular sampling of a 2D continuous function
 - The basis function used in sampling is, conceptually, an impulse function, shifted to various image locations
 - Can be implemented as a convolution

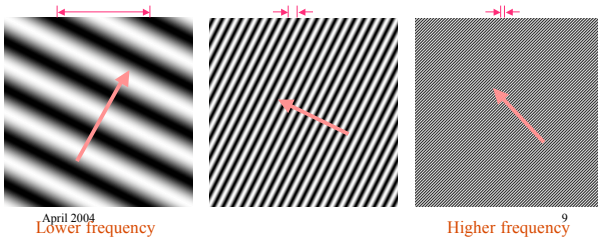


April 2004

8

Spatial frequency and Fourier transforms

- We could use a different basis function (or basis set) to sample the image
- Let's instead use 2D sinusoid functions at various frequencies (scales) and orientations
 - Can also be thought of as a convolution (or dot product)



April 2004

Lower frequency

Higher frequency

Fourier transform

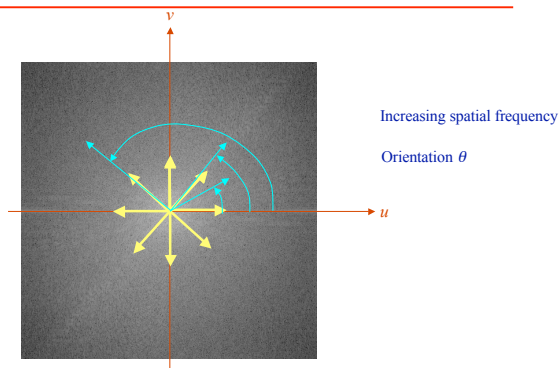
$$F(u, v) = \iint_{\mathbb{R}^2} g(x, y) e^{-i2\pi(ux+vy)} dx dy$$

- For a given (u, v) , this is a dot product between the whole image $g(x, y)$ and the complex sinusoid $\exp(-i2\pi(ux+vy))$
 - $\exp(i\theta) = \cos\theta + i \sin\theta$
- $F(u, v)$ is a complete description of the image $g(x, y)$
- Spatial frequency components (u, v) define the scale and orientation of the sinusoidal “basis filters”
 - **Frequency** of the sinusoid: $(u^2 + v^2)^{1/2}$
 - **Orientation** of the sinusoid: $\theta = \tan^{-1}(v/u)$

April 2004

10

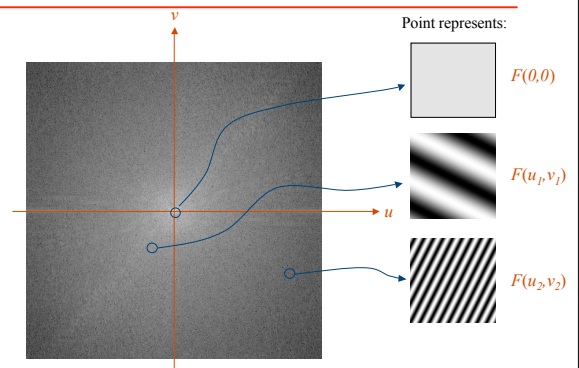
(u, v) – Frequency and orientation



April 2004

11

(u, v) – Frequency and orientation

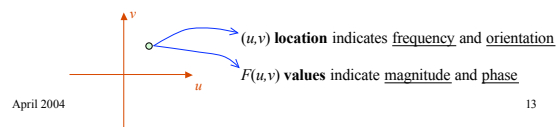


April 2004

12

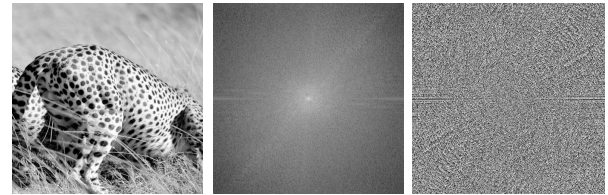
Fourier transform

- The output $F(u,v)$ is a complex image (real and imaginary components)
 - $F(u,v) = F_R(u,v) + i F_I(u,v)$
- It can also be considered to comprise a *phase* and *magnitude*
 - Magnitude: $|F(u,v)| = [(F_R(u,v))^2 + (F_I(u,v))^2]^{1/2}$
 - Phase: $\phi(F(u,v)) = \tan^{-1}(F_I(u,v) / F_R(u,v))$



April 2004

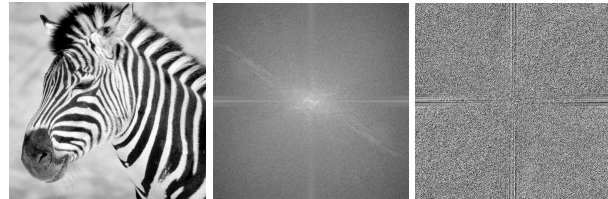
13



Original

Magnitude

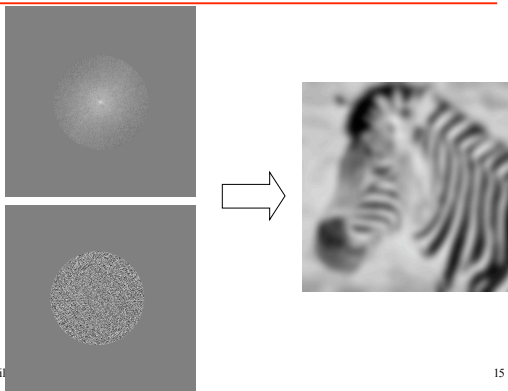
Phase



April 2004

14

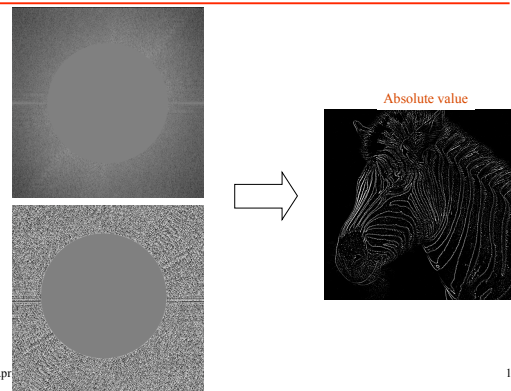
Low-pass filtering via FT



April

15

High-pass filtering via FT



Apr

16

Fourier transform facts

- The FT is linear and invertible (inverse FT)
- A fast method for computing the FT exists (the FFT)
- The FT of a Gaussian is a Gaussian
- $F(f * g) = F(f) F(g)$
- $F(fg) = k F(f) * F(g)$
- $F(\delta(x,y)) = 1$
- (See Table 7.1)

April 2004

17

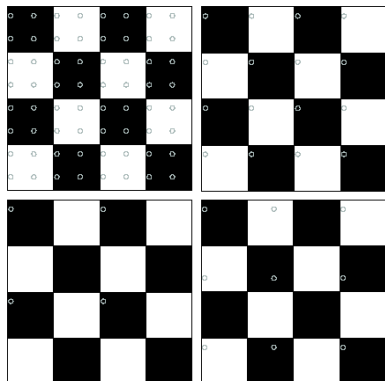
Sampling and aliasing

- Analog signals (images) can be represented accurately and perfectly reconstructed if the sampling rate is high enough
 - ≥ 2 samples per cycle of the highest frequency component in the signal (image)
- If the sampling rate is not high enough (i.g., the image has components over the *Nyquist frequency*)
 - Bad things happen!
- This is called *aliasing*
 - Smooth things can look jagged
 - Patterns can look very different
 - Colors can go astray
 - Wagon wheels can move backwards (temporal sampling)

April 2004

18

Examples



April 2004

19



April 2004

Original

20

Filtering and subsampling



Filtered then Subsampled

Subsampled

April 2004

21

Filtering and sub-sampling



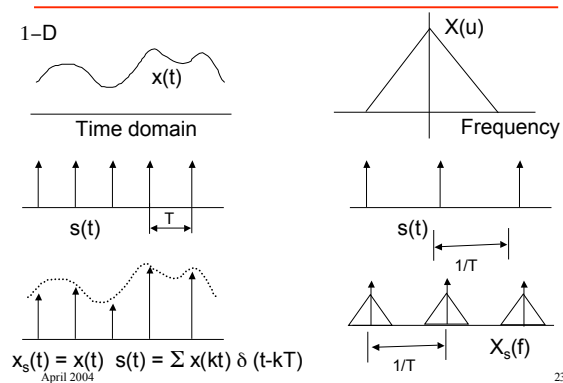
Filtered then Subsampled

Subsampled

April 2004

22

Sampling in 1-D



23

The bottom line

- High frequencies lead to trouble with sampling
- Solution: suppress high frequencies before sampling
 - Multiply the FT of the image with a mask that filters out high frequency, or...
 - Convolve with a low-pass filter (commonly a Gaussian)

April 2004

24

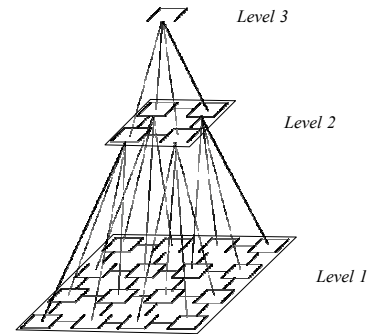
Filter and subsample

- So if you want to sample an image at a certain rate (e.g., resample a 640x480 image to make it 160x120), but the image has high frequency components over the *Nyquist frequency*, what can you do?
 - Get rid of those high frequencies by low-pass filtering!
- This is a common operation in imaging and graphics:
 - “Filter and subsample”
- **Image pyramid**: Shows an image at multiple scales
 - Each one a *filtered and subsampled* version of the previous
 - Complete pyramid has $(1 + \log_2 N)$ levels (where N is image height or width)

April 2004

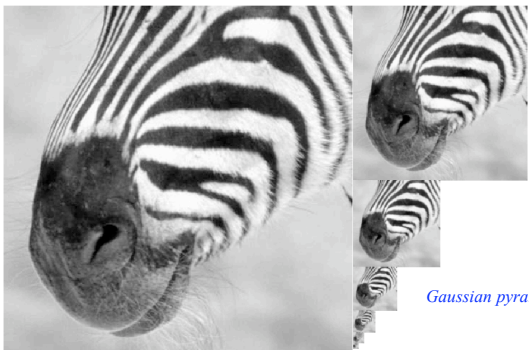
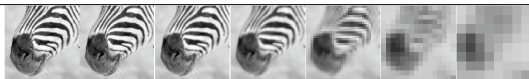
25

Image pyramid



April 2004

26



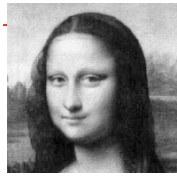
1

Image pyramids

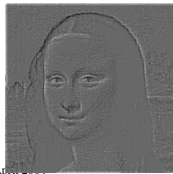
- Image pyramids are useful in object detection/recognition, image compression, signal processing, etc.
- Gaussian pyramid
 - Filter with a Gaussian
 - Low-pass pyramid
- Laplacian pyramid
 - Filter with the difference of Gaussians (at different scales)
 - Band-pass pyramid
- Wavelet pyramid
 - Filter with wavelets

April 2004

28



Gaussian pyramid



Laplacian pyramid

April 2004

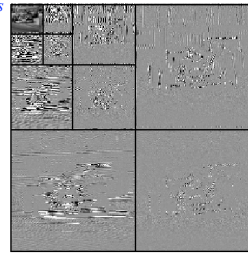
29

Wavelet Transform Example



Original

Low pass



High pass - horizontal

High pass - vertical

High pass - both

April 2004

30

Pyramid filters (1D view)

Gaussian Pyramid



$G(x)$

Laplacian Pyramid



$G_1(x) - G_2(x)$

Wavelet Pyramid



$G(x) \sin(x)$

April 2004

31

Spatial frequency

- The Fourier transform gives us a precise way to define, represent, and measure spatial frequency in images
- Other transforms give similar descriptions:
 - Discrete Cosine Transform (DCT) – used in JPEG
 - Wavelet transforms – very popular
- Because of the FT/convolution relationship
 - $F(f * g) = F(f) F(g)$
 - convolutions can be implemented via Fourier transforms!
 - $f * g = F^{-1} \{ F(f) F(g) \}$
 - ♦ For large kernels, this can be much more efficient

April 2004

32

Convolution and correlation

- Back to convolution/correlation
- Convolution (or FT/IFT pair) is equivalent to linear filtering
 - Think of the filter *kernel* as a pattern, and convolution checks the response of the pattern at every point in the image
 - At each point, it is a dot product of the local image area with the filter kernel

$$R_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} H_{mn} F_{i-m, j-n}$$

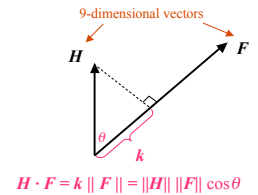
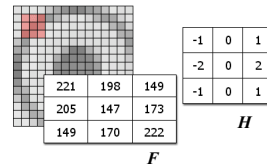
- Conceptually, the image responds best to the pattern of the filter kernel (*similarity*)
 - An edge kernel will produce high responses at edges, a face kernel will produce high responses at faces, etc.

April 2004

33

Convolution and correlation

- For a given filter kernel, what image values really do give the largest output value?
 - All “white” – maximum pixel values
- What image values will give a zero output?
 - All zeros – or, any local “vector” of values that is perpendicular to the kernel “vector”

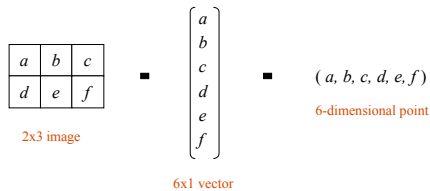


April 2004

34

Image = vector = point

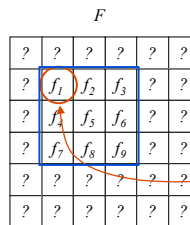
- An m by n image (or image patch) can be reorganized as a mn by 1 vector, or as a point in mn -dimensional space



April 2

35

Correlation as a dot product



At this location, $F \cdot H$ equals the dot product of two 9-dimensional vectors

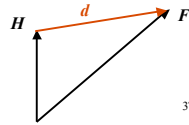
$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} \text{ dot } \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = f^T h = \sum f_i h_i$$

April 2004

36

Finding patterns in images via correlation

- Correlation gives us a way to find patterns in images
 - Task: Find the pattern H in the image F
 - Approach:
 - Convolve (correlate) H and F
 - Find the maximum value of the output image
 - That location is the “best match”
 - H is called a “matched filter”
- Another way: Calculate the distance d between the image patch F and the pattern H
 - $d^2 = \sum (F_i - H_i)^2$
 - Approach:
 - The location with minimum d^2 defines the best match
 - This is quite expensive



April 2004

37

Minimize $d^2 = (F - H)^2 = \sum_i (F_i - H_i)^2$

$$= \sum_i (F_i^2 - 2F_i H_i + H_i^2)$$

$$= \sum_i F_i^2 - 2 \sum_i F_i H_i + \sum_i H_i^2$$

Assume fixed (more or less) Correlation Fixed

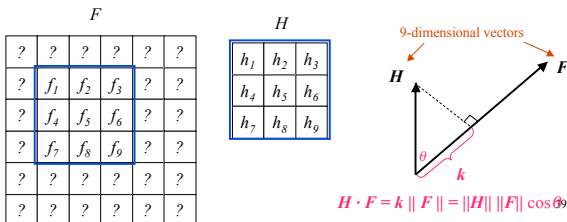
So minimizing d^2 is approximately equivalent to maximizing the correlation

April 2004

38

Normalized correlation

- Problems with these two approaches:
 - Correlation responds “best” to an all “white” patch (maximum pixel values)
 - Both techniques are sensitive to scaling of the image
- Normalized correlation solves these problems



Normalized correlation

- We don't really want *white* to give the maximum output, we want the maximum output to be when $H = F$
 - Or when the angle θ is zero
- Normalized correlation measures the angle θ between H and F
 - What if the image values are doubled? Halved?
 - It is independent of the magnitude (brightness) of the image

$$R = \cos \theta = \frac{H \cdot F}{\|H\| \|F\|}$$

- What if the image values are doubled? Halved?
 - R is independent of the magnitude (brightness) of the image

April 2004

40

Normalized correlation

- Normalized correlation measures the angle θ between H and F
 - What if the image values are doubled? Halved?
 - What if the template values are doubled? Halved?
 - Normalized correlation output is independent of the magnitude (brightness) of the image

$$R = \cos \theta = \frac{H \cdot F}{\|H\| \|F\|}$$

- Drawback: More expensive than correlation
 - Specialized hardware implementations...