

CS 534: Computer Vision
Linear Filters

Spring 2004
Ahmed Elgammal
Dept of Computer Science
Rutgers University

CS 534 – Ahmed Elgammal 1

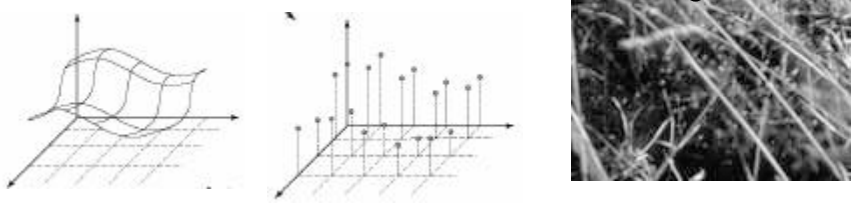
Outlines

- What are linear filters
- Smoothing and convolution
- Image noise
- Smoothing with a Gaussian
- Differentiation and convolution
- Fourier Transform, Discrete Fourier Transform
- Sampling and Aliasing

CS 534 – Ahmed Elgammal 2

Digital image

- Assume we use a gray-level image
- Digital image: a two-dimensional light intensity function $f(x,y)$ where x and y denote spatial coordinates, the value of f at any point is proportional to the brightness (gray level) of the image at that point.
- A digital image:
 - is discretized in the spatial domain
 - Is discretized in the brightness domain.



CS 534 – Ahmed Elgammal 3

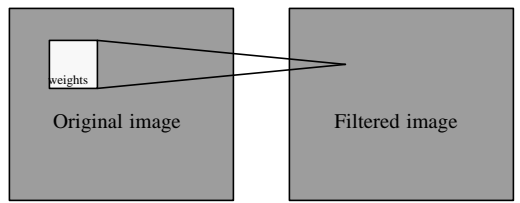
Linear Filters

General process:

- Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.

Properties

- Output is a linear function of the input
- Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)
- This operation is called convolution



CS 534 – Ahmed Elgammal 4

Smoothing and convolution

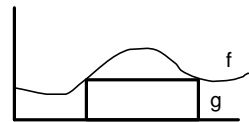
- The convolution of two functions, $f(x)$ and $g(x)$ is defined as

$$h(x) = \int_{-\infty}^{\infty} g(x')f(x-x')dx' = g(x) * f(x)$$

- When the functions f and g are discrete and when g is nonzero only over a finite range $[-n,n]$ then this integral is replaced by the following summation:

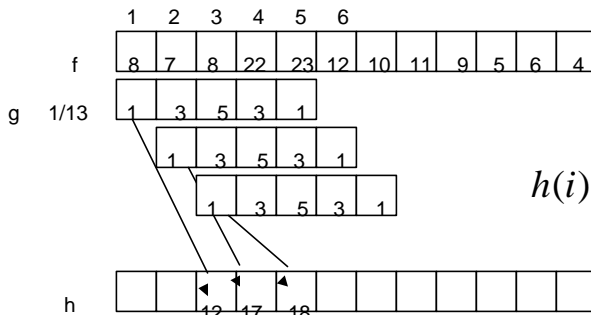
$$h(i) = \sum_{j=-n}^n g(j)f(i+j)$$

- g is called kernel function



CS 534 – Ahmed Elgammal 5

Example of 1-d convolution



$$h(i) = \sum_{j=-n}^n g(j)f(i+j)$$

$$h(4) = \sum_{j=-2}^2 g(j)f(4+j)$$

$$= g(-2)f(2) + g(-1)f(3) + g(0)f(4) + g(1)f(5) + g(2)f(6)$$

CS 534 – Ahmed Elgammal 6

Smoothing and convolution

- These integrals and summations extend simply to functions of two variables:

$$h(i, j) = f(i, j) * g = \sum_{k=-n}^n \sum_{l=-n}^n g(k, l) f(i + k, j + l)$$

- Convolution computes the weighted sum of the gray levels in each $n \times n$ neighborhood of the image, f , using the matrix of weights g .
- Convolution is a so-called linear operator because
 - $g * (af_1 + bf_2) = a(g * f_1) + b(g * f_2)$
- Convolution is shift invariant.

CS 534 – Ahmed Elgammal 7

2-D convolution

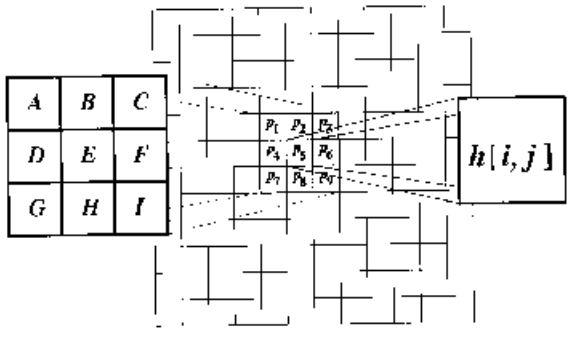
$$\begin{aligned} h(5, 5) &= \sum_{k=-1}^1 \sum_{l=-1}^1 g(k, l) f(5 + k, 5 + l) \\ &= g(-1, -1) f(4, 4) + g(-1, 0) f(4, 5) + g(-1, 1) f(4, 6) \\ &\quad + g(0, -1) f(5, 4) + g(0, 0) f(5, 5) + g(0, 1) f(5, 6) \\ &\quad + g(1, -1) f(6, 4) + g(1, 0) f(6, 5) + g(1, 1) f(6, 6) \end{aligned}$$

CS 534 – Ahmed Elgammal 8

Smoothing and convolution

4.2. LINEAR SYSTEMS

117

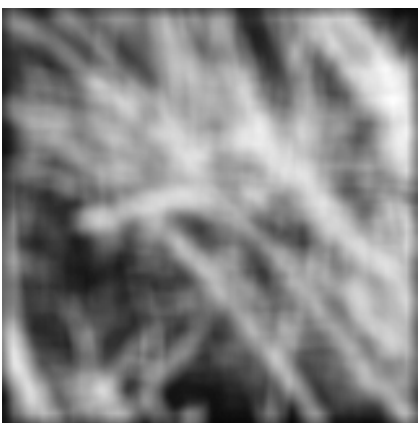


$$h[i, j] = A p_1 + B p_2 + C p_3 + D p_4 + E p_5 + F p_6 + G p_7 + H p_8 + I p_9$$

CS 534 – Ahmed Elgammal 9

Example: Smoothing by Averaging

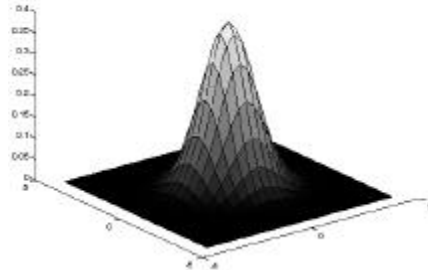
Flat kernel: all weights equal 1/N



CS 534 – Ahmed Elgammal 10

Smoothing with a Gaussian

- Smoothing with an average actually doesn't compare at all well with a defocussed lens
 - Most obvious difference is that a single point of light viewed in a defocussed lens looks like a fuzzy blob; but the averaging process would give a little square.



- A Gaussian gives a good model of a fuzzy blob

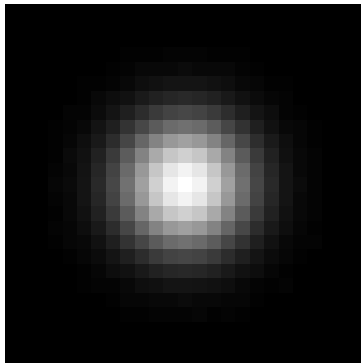
CS 534 – Ahmed Elgammal 11

An Isotropic Gaussian

- The picture shows a smoothing kernel proportional to

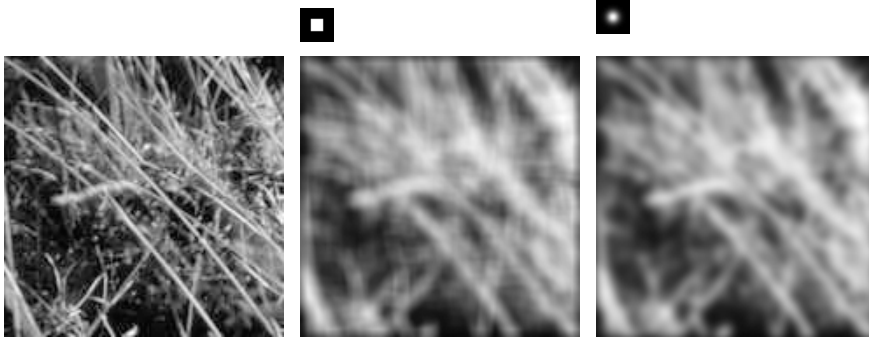
$$\exp\left(-\left(\frac{x^2 + y^2}{2s^2}\right)\right)$$

- (which is a reasonable model of a circularly symmetric fuzzy blob)



CS 534 – Ahmed Elgammal 12

Smoothing with a Gaussian



CS 534 – Ahmed Elgammal 13

Gaussian smoothing

- Advantages of Gaussian filtering
 - rotationally symmetric (for large filters)
 - filter weights decrease monotonically from central peak, giving most weight to central pixels
 - Simple and intuitive relationship between size of σ and the smoothing.
 - The Gaussian is separable:

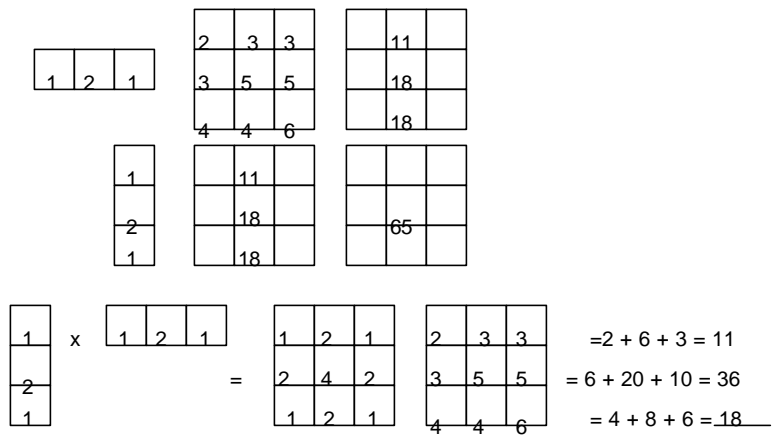
$$e^{-\frac{(x^2+y^2)}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} * e^{-\frac{y^2}{2\sigma^2}}$$

CS 534 – Ahmed Elgammal 14

Advantage of seperability

- First convolve the image with a one dimensional horizontal filter
- Then convolve the result of the first convolution with a one dimensional vertical filter
- For a $k \times k$ Gaussian filter, 2D convolution requires k^2 operations per pixel
- But using the separable filters, we reduce this to $2k$ operations per pixel.

Separability



Advantages of Gaussians

- Convolution of a Gaussian with itself is another Gaussian
 - so we can first smooth an image with a small Gaussian
 - then, we convolve that smoothed image with another small Gaussian and the result is equivalent to smoother the original image with a larger Gaussian.
 - If we smooth an image with a Gaussian having sd σ twice, then we get the same result as smoothing the image with a Gaussian having standard deviation $(2\sigma)^{1/2}$

CS 534 – Ahmed Elgammal 17

Noise

- Simplest noise model
 - independent stationary additive Gaussian noise
 - the noise value at each pixel is given by an independent draw from the same normal probability distribution

$$f_{observed}(x, y) = f(x, y) + N(0, \mathbf{s}^2)$$

- Issues
 - this model allows noise values that could be greater than maximum camera output or less than zero
 - for small standard deviations, this isn't too much of a problem - it's a fairly good model
 - independence may not be justified (e.g. damage to lens)
 - may not be stationary (e.g. thermal gradients in the ccd)

CS 534 – Ahmed Elgammal 18

$\sigma=1$



CS 534 – Ahmed Elgammal 19

$\sigma=16$



CS 534 – Ahmed Elgammal 20

The response of a linear filter to noise

- Do only stationary independent additive Gaussian noise with zero mean (non-zero mean is easily dealt with)
- **Mean:**
 - output is a weighted sum of inputs
 - so we want mean of a weighted sum of zero mean normal random variables
 - must be zero
- **Variance:**
 - recall
 - variance of a sum of random variables is sum of their variances
 - variance of constant times random variable is constant² times variance
 - then if σ^2 is noise variance and kernel is K , variance of response is

$$f_{observed}(x, y) = f(x, y) + N(0, \mathbf{s}^2)$$

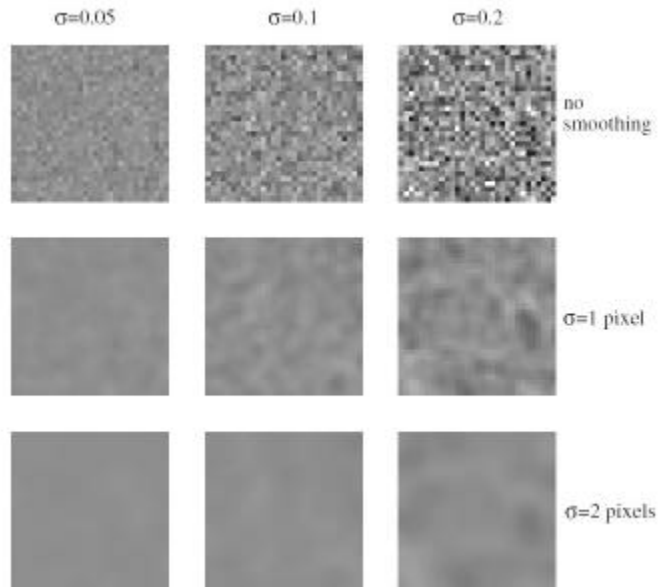
$$g * f_{observed} = g * f + g * N(0, \mathbf{s}^2)$$

$$\Downarrow$$

$$N'(0, \mathbf{s}'^2)$$

$$\mathbf{s}^2 \sum_{u,v} K_{u,v}^2$$

CS 534 – Ahmed Elgammal 21



CS 534 – Ahmed Elgammal 22

Differentiation and convolution

- Recall

$$\frac{\partial f}{\partial x} = \lim_{e \rightarrow 0} \left(\frac{f(x + \mathbf{e}, y) - f(x, y)}{\mathbf{e}} \right)$$

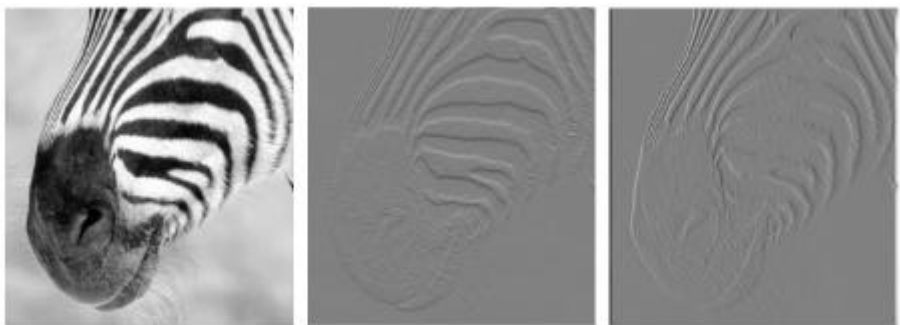
- Now this is linear and shift invariant, so must be the result of a convolution.
- We could approximate this as

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x} \quad G = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

(which is obviously a convolution; it's not a very good way to do things, as we shall see)

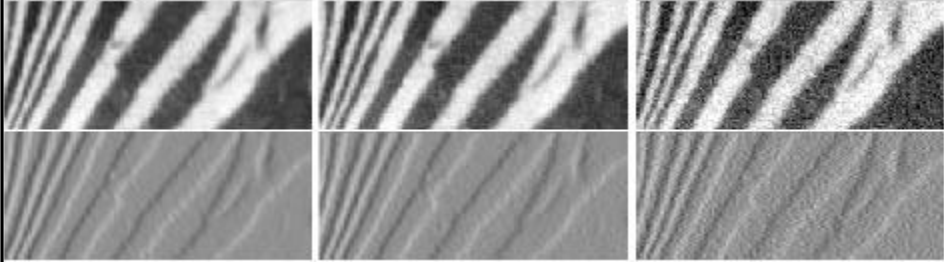
CS 534 – Ahmed Elgammal 23

Finite differences



CS 534 – Ahmed Elgammal 24

Finite differences responding to noise



Increasing noise ->
(this is zero mean additive gaussian noise)

CS 534 – Ahmed Elgammal 25

Finite differences and noise

- Finite difference filters respond strongly to noise
 - obvious reason: image noise results in pixels that look very different from their neighbours
- Generally, the larger the noise the stronger the response
- What is to be done?
 - intuitively, most pixels in images look quite a lot like their neighbors
 - this is true even at an edge; along the edge they're similar, across the edge they're not
 - suggests that smoothing the image should help, by forcing pixels different to their neighbors (=noise pixels?) to look more like neighbors

CS 534 – Ahmed Elgammal 26

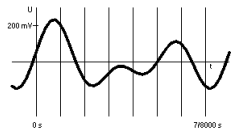
Fourier Transform

- Fourier Transform as a change of basis

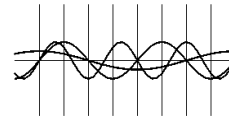
$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$e^{-j2\pi ux} = \cos(2\pi ux) - j \sin(2\pi ux)$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$



Temporal or spatial domain



Frequency domain

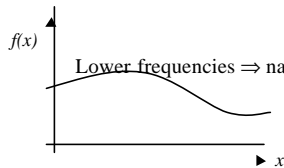
- $F(u)$ is composed of an infinite sum of sine and cosine terms
- Inverse transform: reconstruction of the signal as infinite sum of sine and cosine terms

CS 534 – Ahmed Elgammal 27

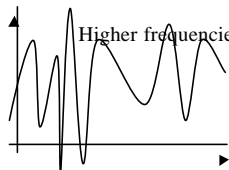
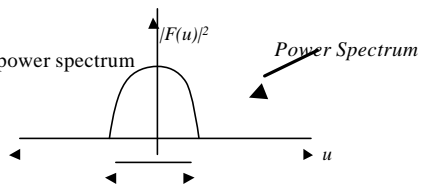
- Since of a real function is generally complex, we use magnitude and phase

$$F(u) = R(u) + jI(u) \quad \Rightarrow \quad |F(u)| = (R^2(u) + I^2(u))^{1/2}$$

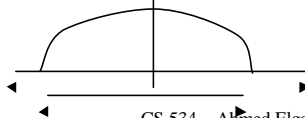
$$f(u) = \tan^{-1} I(u) / R(u)$$



Lower frequencies \Rightarrow narrower power spectrum



Higher frequencies \Rightarrow wider power spectrum



CS 534 – Ahmed Elgammal 28

Important Properties:

- Fourier transform is linear

$$F(g(x) + h(x)) = F(g(x)) + F(h(x))$$

$$F(kg(x)) = kF(g(x))$$

- FT and Convolution

$$f(x) * g(x) \Leftrightarrow F(u)G(u)$$

$$f(x)g(x) \Leftrightarrow F(u) * G(u)$$

- FT of a Gaussian is a Gaussian

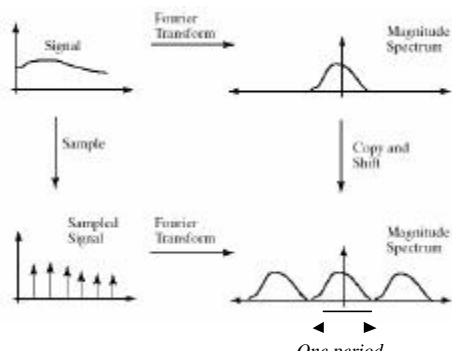
Discrete Fourier Transform

- If we discretize $f(x)$ using uniformly spaced samples $f(0), f(1), \dots, f(N-1)$, we can obtain FT of the sampled function

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N}$$

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{-j2\pi ux/N}$$

- Important Property:
Periodicity $F(u) = F(u+N)$



2-dimension

In two dimension

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\mathbf{p}(ux+vy)} dx dy$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\mathbf{p}(ux+vy)} du dv$$

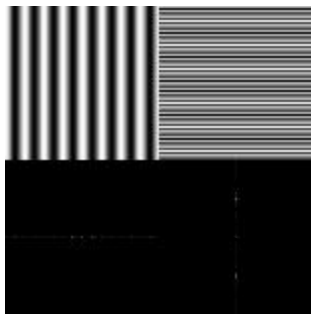
$$e^{-j2\mathbf{p}(ux+vy)} = \cos(2\mathbf{p}(ux+vy)) + j \sin(2\mathbf{p}(ux+vy))$$

- These terms are sinusoids on the x,y plane whose orientation and frequency are defined by u,v



CS 534 – Ahmed Elgammal 31

- The FT tries to represent all images as a summation of cosine-like images



Images of pure cosines

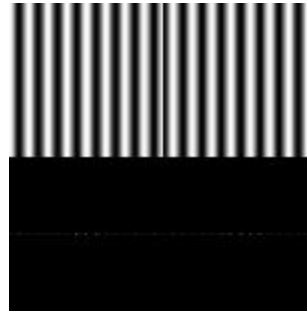
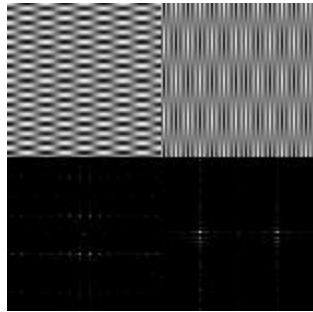
- *Center of the image*: the origin of the frequency coordinate system
- *u-axis*: (left to right) the horizontal component of frequency
- *v-axis*: (bottom-top) the vertical component of frequency
- Center dot (0,0) frequency : image average

FT

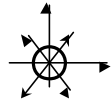
- high frequencies in the vertical direction will cause bright dots away from the center in the vertical direction.
- high frequencies in the horizontal direction will cause bright dots away from the center in the horizontal direction.

CS 534 – Ahmed Elgammal 32

- Since images are real numbers (not complex) FT image is symmetric around the origin.



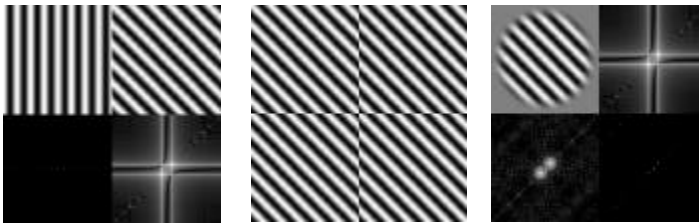
FT: symmetry



FT is shift invariant

CS 534 – Ahmed Elgammal 33

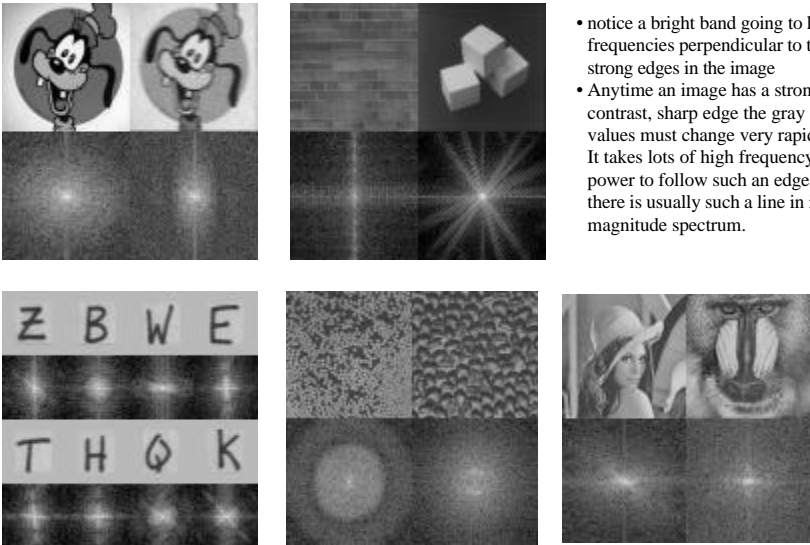
- In general, rotation of the image results in equivalent rotation of its FT



Why it is not the case ?

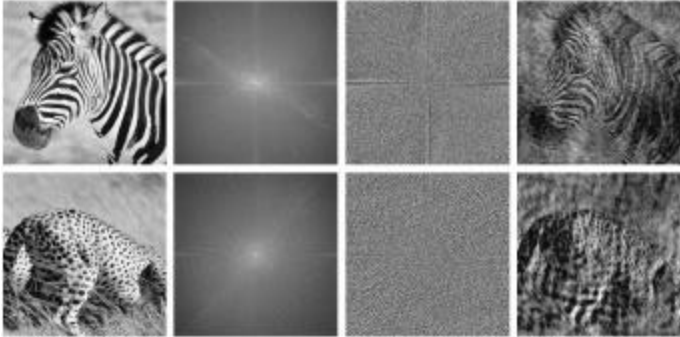
- Edge effect !
- FT always treats an image as if it were part of a periodically replicated array of identical images extending horizontally and vertically to infinity
- Solution: “windowing” the image

CS 534 – Ahmed Elgammal 34



- notice a bright band going to high frequencies perpendicular to the strong edges in the image
- Anytime an image has a strong-contrast, sharp edge the gray values must change very rapidly. It takes lots of high frequency power to follow such an edge so there is usually such a line in its magnitude spectrum.

CS 534 – Ahmed Elgammal 35

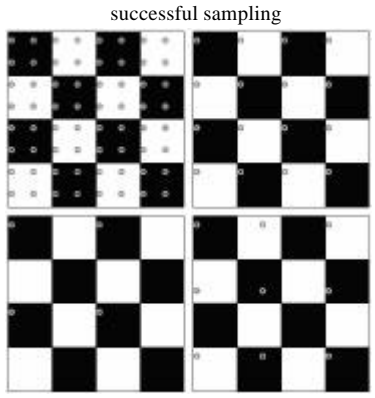
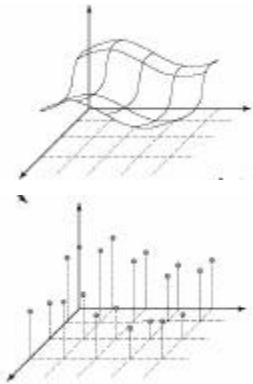


- What happens if we swap the magnitude spectra ?
- Phase spectrum holds the spatial information (where things are),
- Phase spectrum is more important for perception than magnitude spectrum.

CS 534 – Ahmed Elgammal 36

Sampling and Aliasing

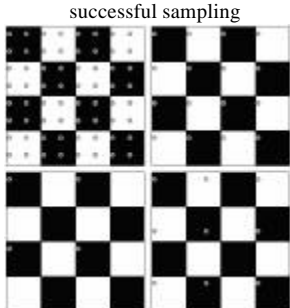
- Differences between continuous and discrete images
- Images are sampled version of a continuous brightness function.



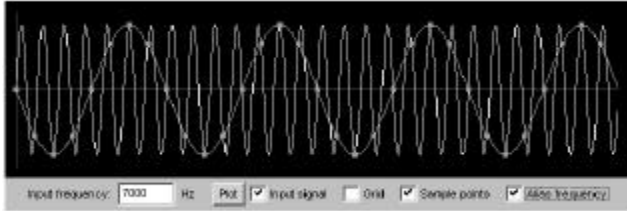
unsuccessful sampling

Sampling and Aliasing

- Sampling involves loss of information
- Aliasing: high spatial frequency components appear as low spatial frequency components in the sampled signal



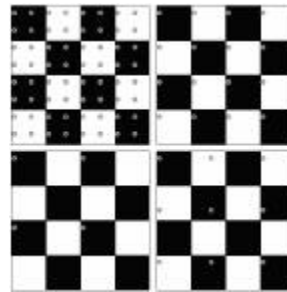
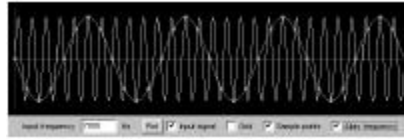
unsuccessful sampling



Java applet from: <http://www.dsptutor.freeuk.com/aliasing/AD102.html>

Aliasing

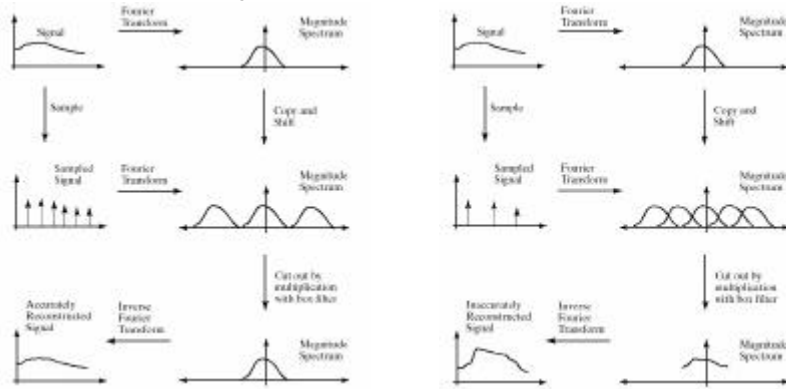
- Nyquist theorem:** The sampling frequency must be at least twice the highest frequency present for a signal to be reconstructed from a sampled version. (*Nyquist frequency*)



CS 534 – Ahmed Elgammal 39

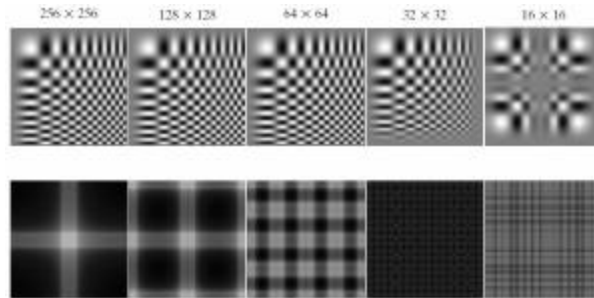
Sampling, aliasing, and DFT

- DFT consists of a sum of copies of the FT of the original signal shifted by the sampling frequency:
 - If shifted copies do not intersect: reconstruction is possible.
 - If shifted copies do intersect: incorrect reconstruction, high frequencies are lost (Aliasing)



CS 534 – Ahmed Elgammal 40

Sampling, aliasing, and DFT

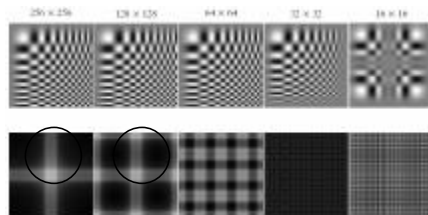


Notices: Interferences between FT copies

CS 534 – Ahmed Elgammal 41

Smoothing and Resampling

- In many vision application we want to reduce image size by half by resampling
- We need to reduce the aliasing effect
- Spatial frequencies above the new sampling frequency should be removed.
- Multiply the DFT by a Gaussian will do
- Equivalently: convolve the original image with a Gaussian.
- Choice of σ is critical:
 - Large σ : Less aliasing, more information lost
 - Small σ : More aliasing, less information lost



CS 534 – Ahmed Elgammal 42

Sources

- Forsyth and Ponce, Computer Vision a Modern approach: chapter 7.
- R. Gonzalez and P. Wintz, Digital Image Processing, second edition.
- Slides by:
 - D. Forsyth
- Fourier transform images from Prof. John M. Brayer @ UNM

<http://www.cs.unm.edu/~brayer/vision/fourier.html>

- Aliasing java applets available at
http://klebanov.homeip.net/~pavel/fb/java/la_applets/Aliasing/
<http://www.dsptutor.freeuk.com/aliasing/AliasingDemo.html>