HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Computer Vision

Chapter 3: Image Processing

Computer Vision

Chapter 3. Image Processing

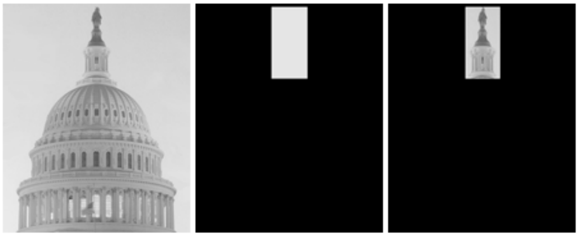
Content

- Rappel: digital image representation
- Point Processing
- Convolution and Linear filtering
- More neighborhood operators
  - Median/max/min filters
  - Arithmetical/Logical operations
  - Binary image and morphological operations
- Image transforms

Arithmetical/Logical Operations

- AND operation
- OR operation
- Image subtraction
- Image addition

AND operation



Original | And mask | Output image

OR operation

Original | OR mask | Output image

Image Addition

- If f and g are two images, the pixelwise addition R is defined as:
  - R(x,y) = Min(f(x,y)+g(x,y) ; 255)
- Image addition is used to
  - lower the noise in a serie of images
  - increase the luminance by adding the image to itself

Source : Eric Favier. L'analyse et le traitement des images. ENISE.

Average Images

- g(x,y) is the addition of f(x,y) and noise $\eta(x,y)$

$$g(x, y) = f(x, y) + \eta(x, y)$$

- If we have several images {g(x,y)}, we can compute the average one

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

Average Images

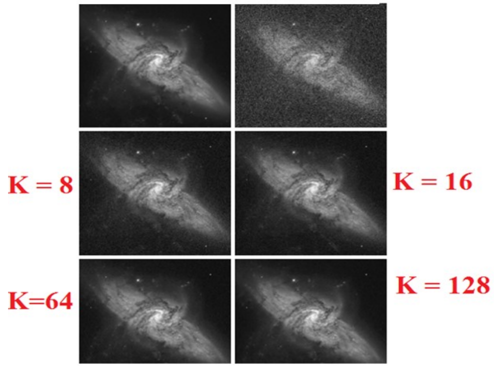
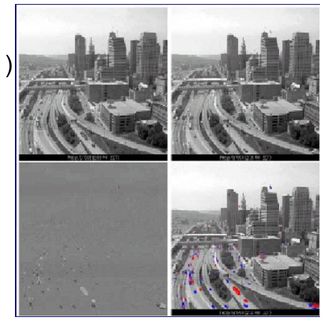


Image subtraction

- The pixelwise substraction of two images f and g is:
  $S(x,y) = Max(f(x,y)-g(x,y) ; 0)$
- Image substraction is used to
  - detect defaults, detect difference between images
  - detect motion in images



Source : Eric Favier. L'analyse et le traitement des images. ENISE.

Image subtraction

a b
c d

FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

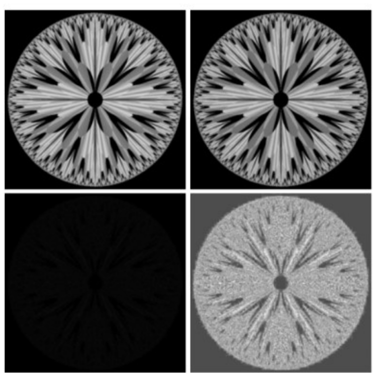


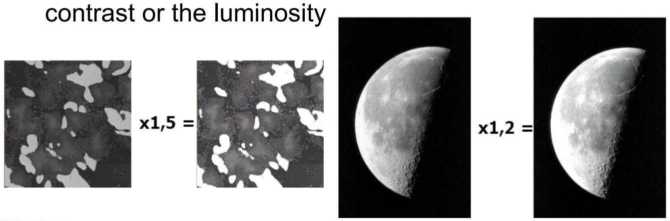
Image subtraction

Image *Background image* *Detection*

After detection, we still have some noise, that we can clean to keep only the object of interest

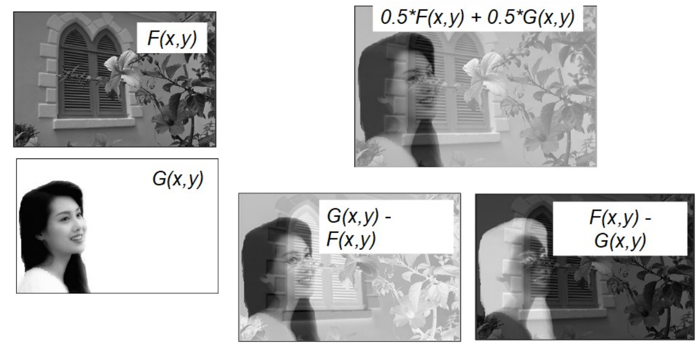
Image multiplication

- The multiplication S of an image f by a ratio (factor) is defined as:
  - S(x,y) = Max(f(x,y)*ratio ; 255)
- Image multiplication can be used to increase the contrast or the luminosity



Source : Eric Favier. L'analyse et le traitement des images. ENISE.

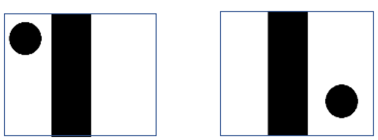
Operations on images



Source : www.nte.montaigne.u-bordeaux.fr/SuppCours/5314/Dai/TraitImage01-02.ppt

Exercise

- Given two images as bellow



1. Transform images to negative ones
2. Process to have an image which has only the "ball"

Binary images

Binary images

- Two pixel values: foreground (object, 1) and background (0)
- Be used
  - To mark region(s) of interest
  - As results of thresholding method

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: edge detection

Gradient magnitude

`fg_pix = find(gradient_mag > t);`

Looking for pixels where gradient is strong.

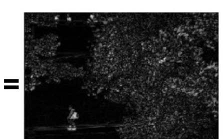
Thresholding

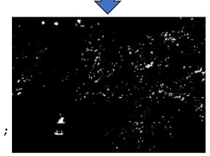
- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: background subtraction

-

=





Looking for pixels that differ significantly from the "empty" background.

`fg_pix = find(diff > t);`

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: intensity-based detection

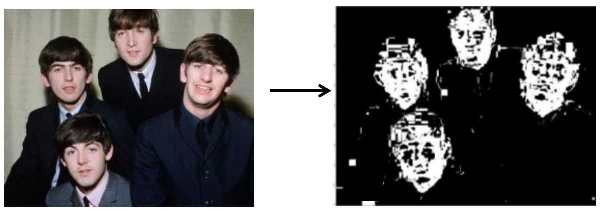
`fg_pix = find(im < 65);`

Looking for dark pixels

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: color-based detection



`fg_pix = find(hue > t1 & hue < t2);`

Looking for pixels within a certain hue range.

Slide credit: Kristen Grauman

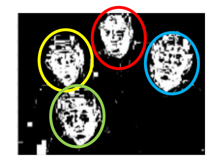
SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

22

Issues

- What to do with “noisy” binary outputs?
  - Holes
  - Extra small fragments

- How to demarcate multiple regions of interest?
  - Count objects
  - Compute further features per object



Slide credit: Kristen Grauman

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

23

Morphological operators

- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image.
- Useful to clean up result from thresholding
- Main components
  - Structuring element
  - Operators:
    - Basic operators: Dilation, Erosion
    - Others: Opening, Closing, …

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

24

Dilation

- Expands connected components
- Grow features
- Fill holes

Before dilation

After dilation

Slide credit: Kristen Grauman

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

25

Erosion

- Erode connected components
- Shrink features
- Remove bridges, branches, noise

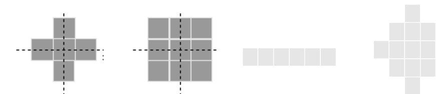
Before erosion

After erosion

Slide credit: Kristen Grauman

Structuring elements

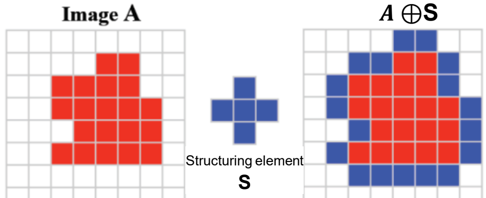
- **Masks** of varying shapes and sizes used to perform morphology, for example:



- Scan mask (structuring element) over the object (foreground) borders (inside and outside) and transform the binary image

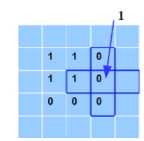
Dilation

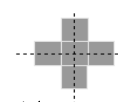
- Moving S on each pixel of A
  - check if the intersection (pixels belonging to object) is not empty
  - If yes, the center of B belongs to the result image
- If a pixel of S is onto object pixels (A), then the central pixel belongs to object
  - Otherwise (i.e. all pixels of are background), set to background (no change)



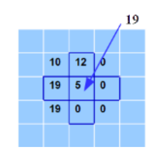
Dilation

- As max filter
- Can be applied both on
  - binary images
  - or grayscale images



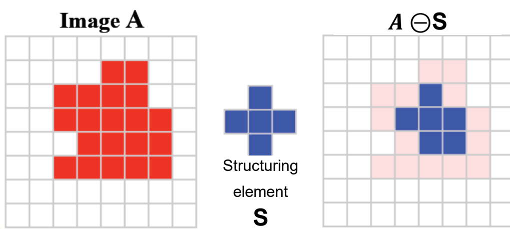


Mask / Structuring element



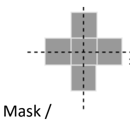
Erosion

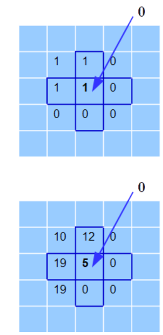
- We put the element S on each pixel x of A
  - like convolution
- If all pixels of S are onto object pixels (A), then the central pixel belongs to object
  - Otherwise (i.e. a mask pixel is background), set to background



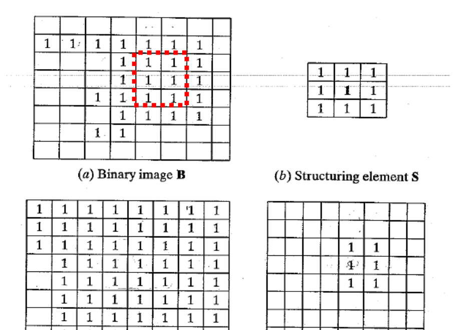
Erosion

- As min filter
- Can be applied both on
  - binary images
  - or grayscale images





2D example



(a) Binary image B

(b) Structuring element S

(c) Dilation B ⊕ S

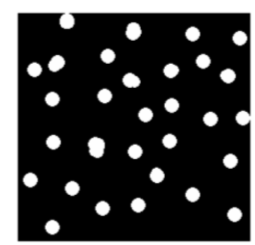
(d) Erosion B ⊖ S

Shapiro & Stockman

Opening

- Erode, then dilate
- Remove small objects, keep original shape

Before opening



After opening

Closing

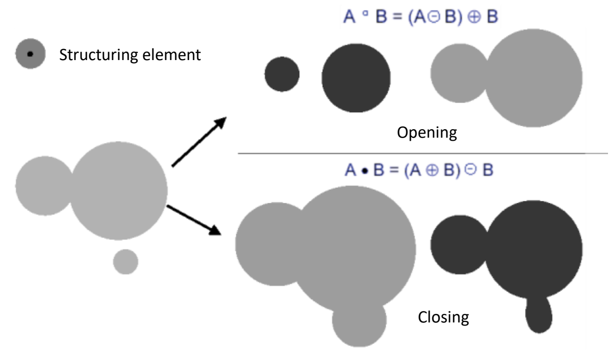
- Dilate, then erode
- Fill holes, but keep original shape

Before closing

After closing

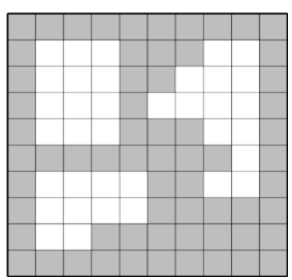
demo: http://bigwww.epfl.ch/demo/jmorpho/start.php

Opening vs Closing



Connected component labeling

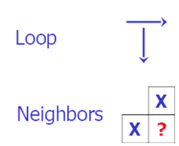
- We loop over all the image to give a unique number (label) for each region
- All pixels from the same region must have the same number (label)
- Objectifs:
  - Counting objects
  - Separating objets
  - Creating a mask for each object
  - …

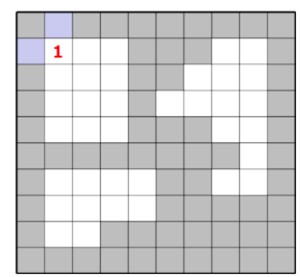


Connected component labeling

First loop over the image

- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label

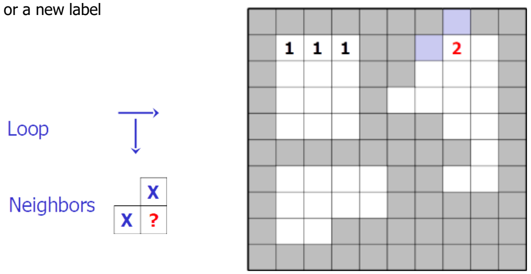




Connected component labeling

First loop over the image

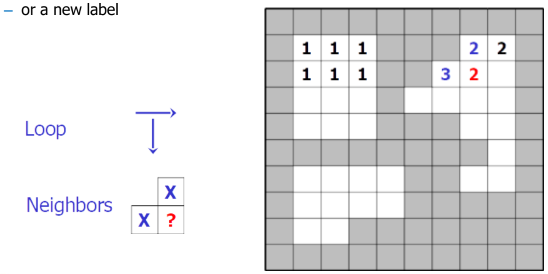
- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label



Connected component labeling

First loop over the image

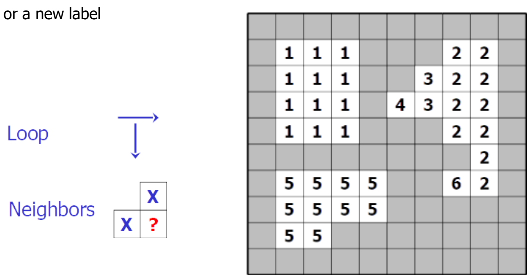
- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label



Connected component labeling

First loop over the image

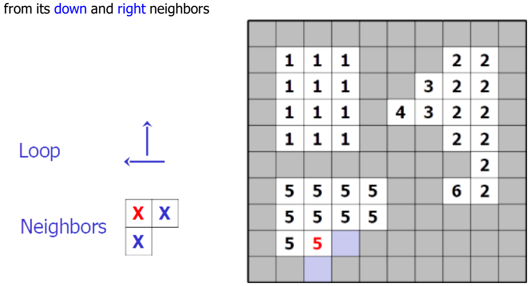
- For each pixel in a region, we set
  - or the smallest label from its top or left neighbors
  - or a new label



Connected component labeling

Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors



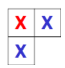
Connected component labeling

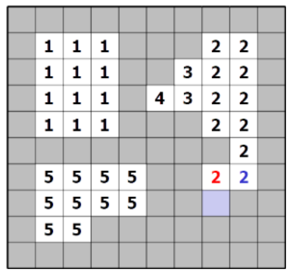
Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors

Loop

Neighbors





SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

42

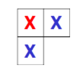
Connected component labeling

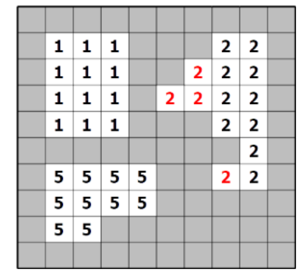
Second loop over the image

- For each pixel in a region, we set
  - the smallest from its own label and the labels from its down and right neighbors

Loop

Neighbors



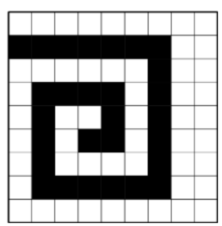


SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

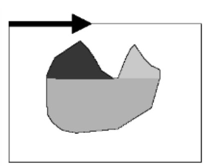
43

Connected component labeling

- Two loops are enough?
  - *example: spiral region !*



- Solutions
  - We continue, go and back two ways, until no new change in labels
  - It is possible to do only one loop: manage a table of equivalences when 2 different labels are neighbors

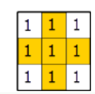


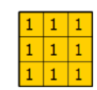
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

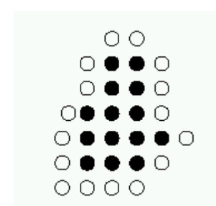
44

CC labeling: how many neighbors?

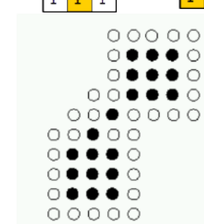
- Advice: Use different connexities for edges and regions
  - *4-Connexity for regions*
  - *8-Connexity for edges*







Region : 4-connected
Edge : 8-connected



Region : 8-connected
Edge : 4-connected

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

45

CC labeling: how many neighbors?

- Regions labeling
  - We use 4-connexity
  - Each loop, we compare 2 neighbors

	1	
2	?	

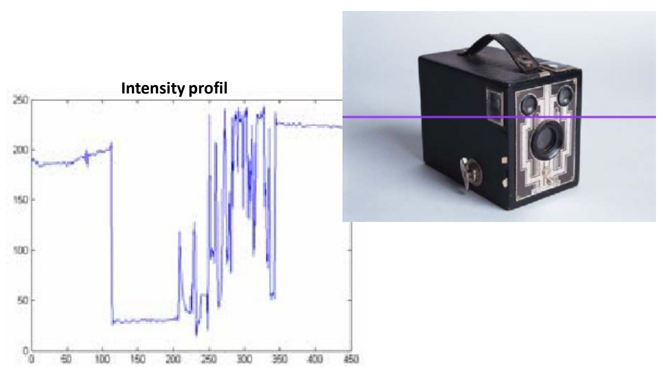
- Edge labeling
  - 8-connexity
  - Each loop, we compare 4 neighbors

1	2	3
4	?	

Content

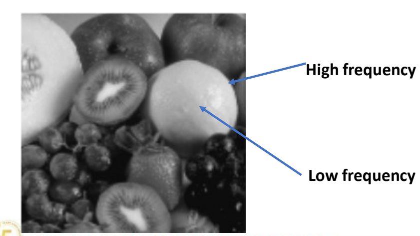
- Rappel: digital image representation
- Point Processing
- Convolution and Linear filtering
- More neighborhood operators
- Image transforms
  - Frequency domain
    - Frequencies in images
    - Fourrier transform
    - Frequential Processing (frequential filters)
  - PCA (additional reading)

Frequencies in images



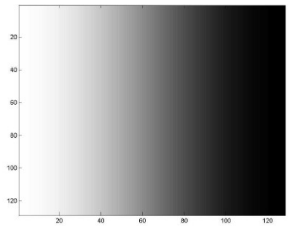
Frequencies in images

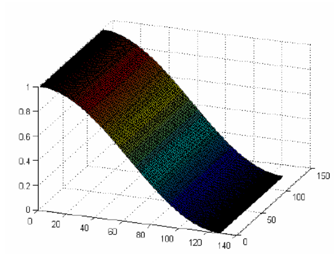
- What are the (low/high) frequencies in an image?
  - Frequency **= intensity change**
  - Slow changes (homogeneous /blur regions): low frequency
  - fast/abrupt changes (egde, contour, noise): high frequency



Most of energy concentrated in low frequencies

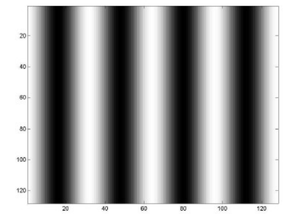
Low frequencies

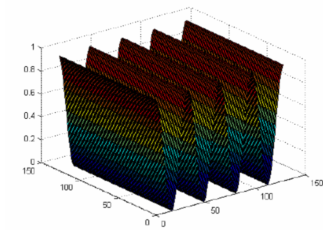




50

High frequencies





51

Image spectral analysis

- An image is a visual signal
  - We can analyse the frequencies of the signal
- How?
  - we will create a new « image » which will contains all frequencies of the image
    - Like a 2D frequency graphic
  - The basic tool for it is the **Fourier Transform**
- We talk about the **frequency domain**, opposing to the **spatial domain** (image)

52

Frequencies in a signal

High frequency signal

...

Low frequency signal

Sum of all the above signals

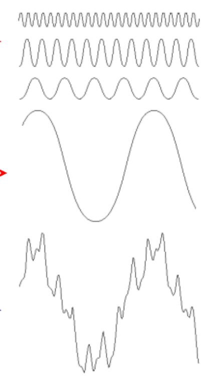


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Source : *Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.*

53

Fourier series

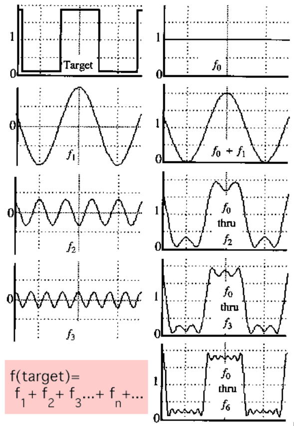
A bold idea (1807) - Jean Baptiste Joseph Fourier (1768-1830):

***Any** univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

Our building block:

$$A\sin(\omega t)+B\cos(\omega t)$$

Add enough of them to get any signal *g(t)* you want!

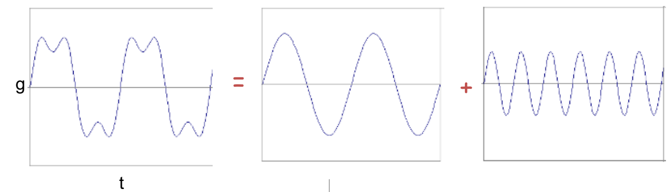


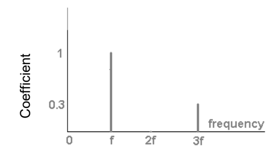
Hays

Example

$t = [0,2], f = 1$

$g(t) \quad = \quad \sin(2\pi f\, t) \quad + (1/3)\sin(2\pi(3f)\, t)$

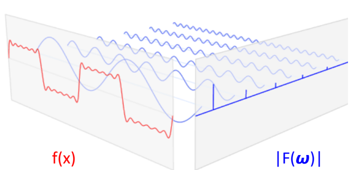




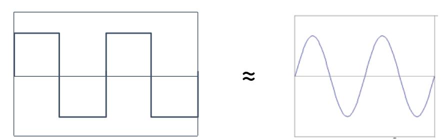
Slides: Efros

Fourier Transform

- Fourier transform is a mathematical transform that
  - Decomposes functions depending on space or time into functions depending on spatial or temporal frequency



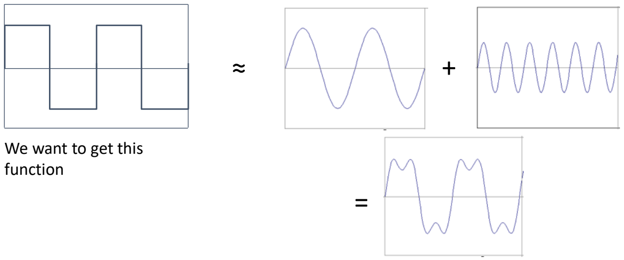
Fourier Series



We want to get this function

Slide by Alexei A. Efros

Fourier Series

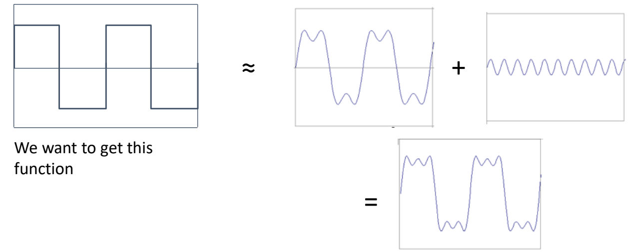


We want to get this function

Slide by Alexei A. Efros

58

Fourier Series

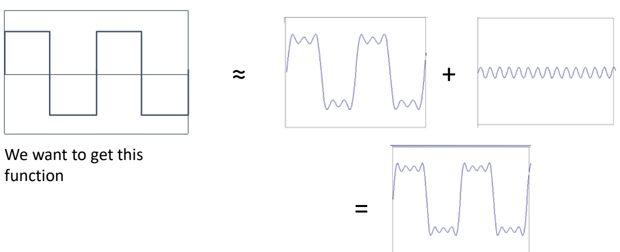


We want to get this function

Slide by Alexei A. Efros

59

Fourier Series

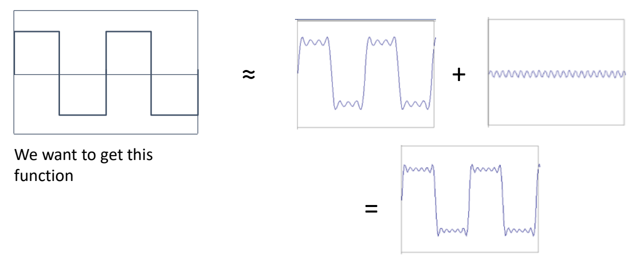


We want to get this function

Slide by Alexei A. Efros

60

Fourier Series

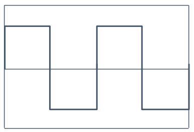


We want to get this function

Slide by Alexei A. Efros

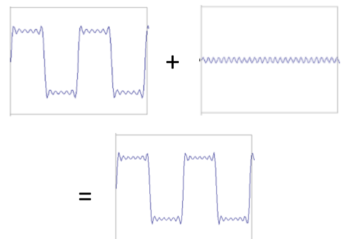
61

Fourier Series



We want to get this function

$\approx$

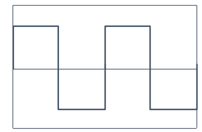


Slide by Alexei A. Efros

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

62

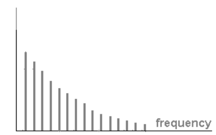
Fourier Series



We want to get this function

$$= A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi\, kt)$$

We'll get there in the limit



Slide by Alexei A. Efros

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

63

The math

$$\text{Fourier Transform}: \quad F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx$$

$$\text{Inverse Fourier Transform}: \quad f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega) e^{i\omega x} d\omega$$

- Where are the sines and cosines? $e^{i\omega x} = \cos(\omega x) + i\sin(\omega x)$
- The result is a complex function $F(\omega) = R(\omega) + iI(\omega)$
- We've been showing only the **amplitude A (spectre)** so far: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$
- Phase is also encoded:

$$\phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$$

Slide by Steve Seitz

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

64

Magnitude and phase

- Fourier transform stores the **magnitude** and **phase** at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of real and complex numbers

Amplitude:

$$A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$$

Phase:

$$\phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$$

Slide by Rober Pless

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

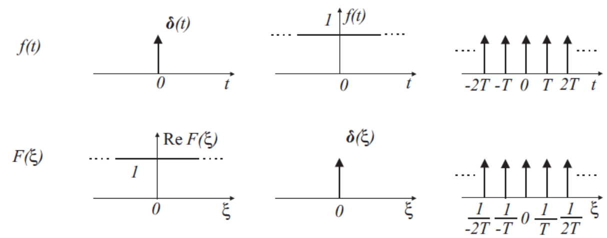
65

Discrete Fourier transform

$$H_{f_j} = \frac{1}{N} \sum_k h_{t_k} e^{2\pi i f_j t_k}$$
$$h_{t_j} = \frac{1}{N} \sum_k H_{f_k} e^{-2\pi i f_k t_j}$$

where the t_k are the time corresponding to my signal in the time domain h_{t_k}. f_k are the corresponding frequency to my signal in the frequency domain, and N is the number of points of the signal data.

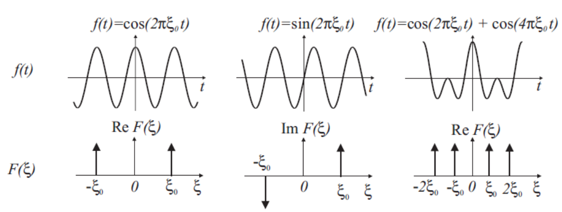
Basic Fourier Transform pairs



Dirac constant ∞ sequence of Diracs

Source: Václav Hlaváč - **Fourier transform, in 1D and in 2D**

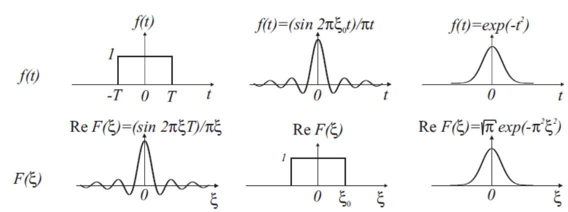
Basic Fourier Transform pairs



cosine sine two cosines mixture

Source: Václav Hlaváč - **Fourier transform, in 1D and in 2D**

Basic Fourier Transform pairs

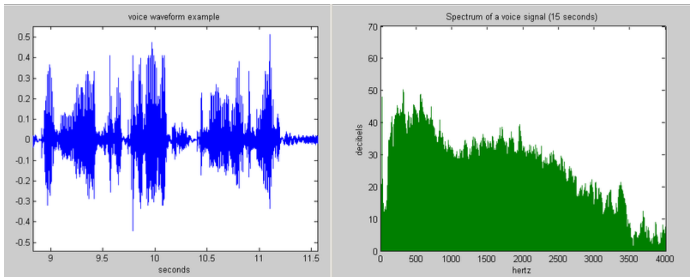


rectangle in t rectangle in ξ Gaussian

Source: Václav Hlaváč - **Fourier transform, in 1D and in 2D**

Example: Music

- We think of music in terms of frequencies at different magnitudes



Slide: Hoiem

2D FFT

- Continuous FFT:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i(xu+yv)}\, \mathrm{d}x\, \mathrm{d}y$$

- Inverse FFT:

$$f(x,y) = \int^{\infty}\int^{\infty} F(u,v)\, e^{2\pi i(xu+yv)}\, \mathrm{d}u\, \mathrm{d}v$$

2D FFT - discrete

Direct transform

$$F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, \exp\left[-2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)\right],$$

$$u = 0,1,\ldots,M-1\,, \qquad v = 0,1,\ldots,N-1\,,$$

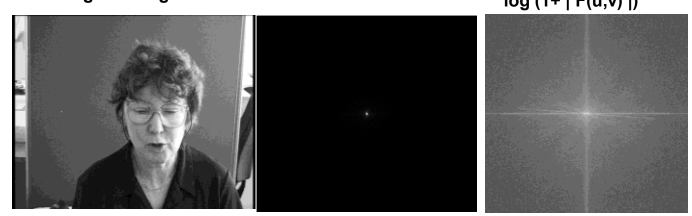
Inverse transform

$$f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\, \exp\left[2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)\right],$$

$$m = 0,1,\ldots,M-1\,, \qquad n = 0,1,\ldots,N-1\,.$$

Image Fourrier transform

Original image **Spectra | F(u,v) |** **Enhanced Spectra log (1+ | F(u,v) |)**



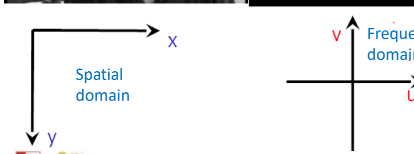


Image Fourrier transform

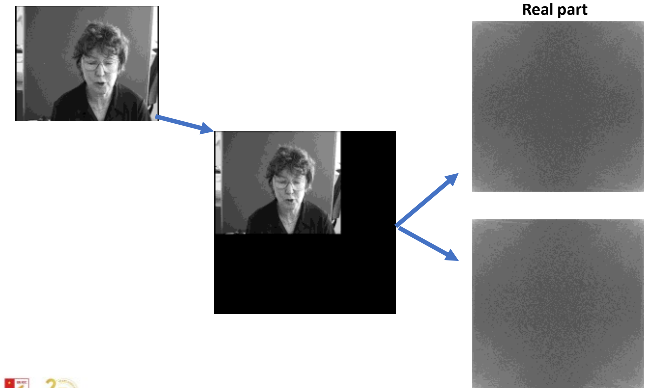
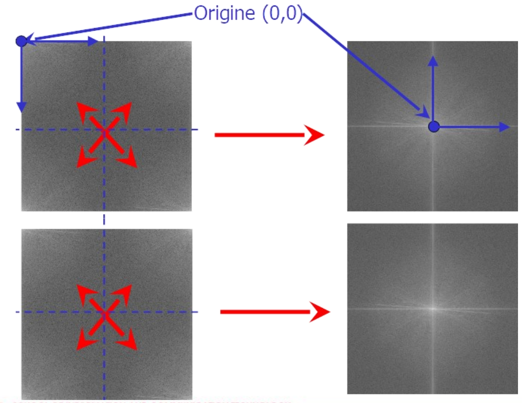
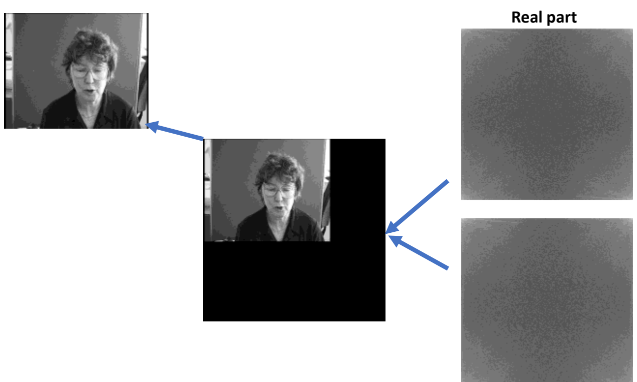


Image Fourrier transform

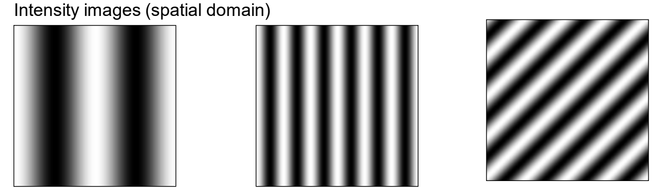


Inverse Fourrier transform

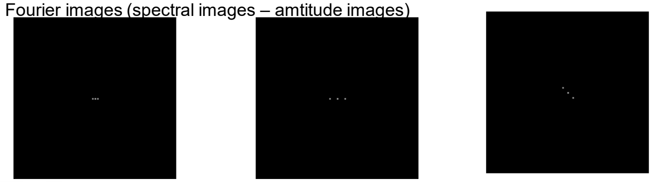


Fourier analysis in images

Intensity images (spatial domain)



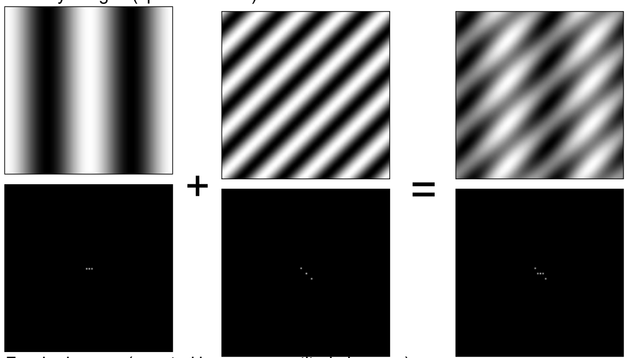
Fourier images (spectral images – amtitude images)



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering More: http://www.cs.unm.edu/~brayer/vision/fourier.html

Signals can be composed

Intensity images (spatial domain)



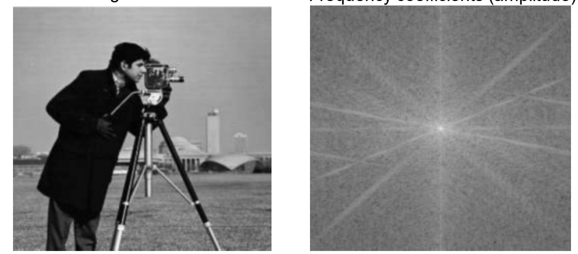
Fourier images (spectral images – amtitude images)

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Fourier Transform of an image

Natural image

Fourier decomposition
Frequency coefficients (amplitude)



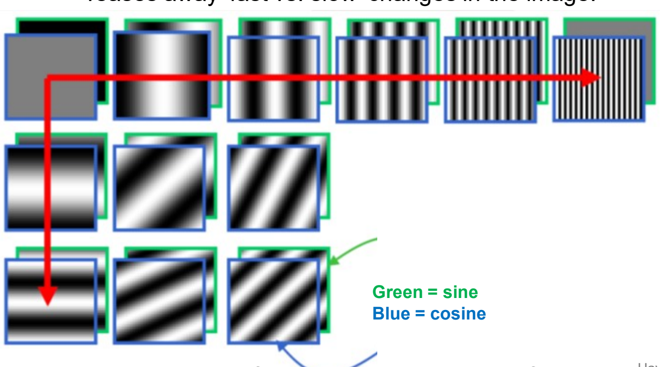
f(x,y) |F(ω)|

What does it mean to be at pixel x,y?
What does it mean to be more or less bright in the Fourier decomposition image?

Slide by Steve Seitz

Fourier Bases

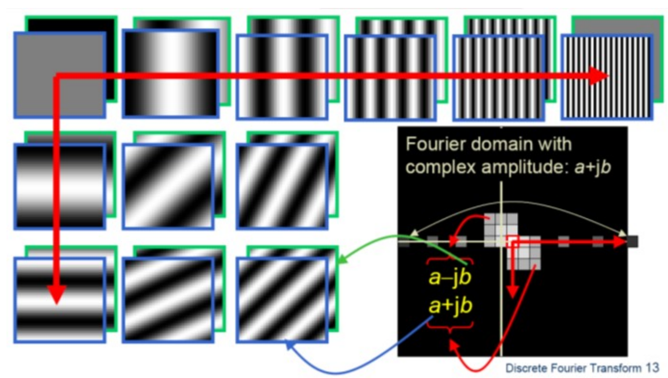
Teases away 'fast vs. slow' changes in the image.



This change of basis is the Fourier Transform

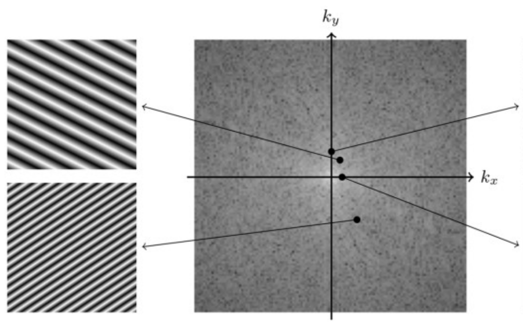
Hays

Fourier Bases



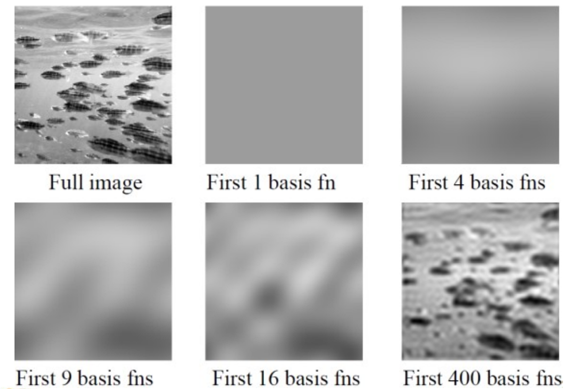
Hays

2D Fourier Transform



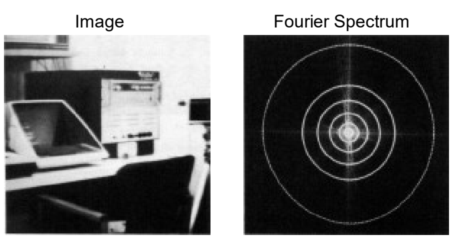
Slide by Steve Seitz

Basis reconstruction



Danny Alexander

2D Fourier transform



Percentage of image power enclosed in circles (small to large) :
90, 95, 98, 99, 99.5, 99.9

Most of energy concentrated in low frequencies

Image filtering in the frequential domain

- We will be dealing only with functions (images) of finite duration so we will be interested only in Fourier Transform

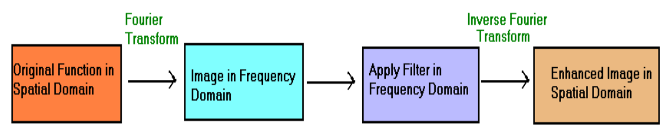
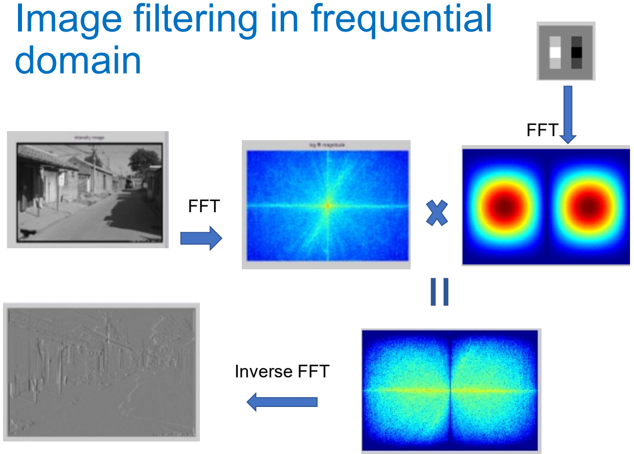


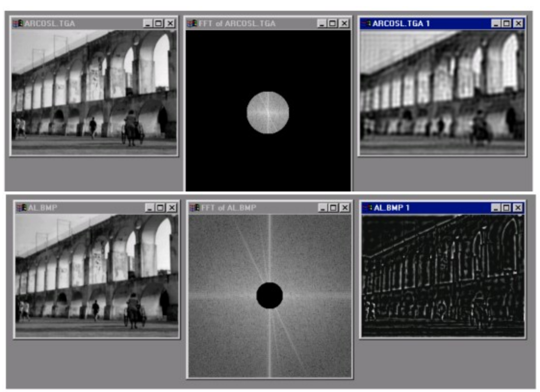
Image filtering in frequential domain



Slide by Derek Hoiem

Now we can edit frequencies!

Low-pass and high-pass filtering



Slide by Alexei Efros

Low-pass filter

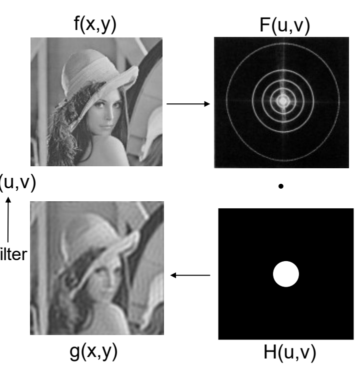
spatial domain — frequency domain

f(x,y) ⟶ F(u,v)

G(u,v) = F(u,v) • H(u,v)

filter

g(x,y) ⟵ G(u,v)

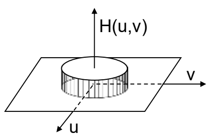


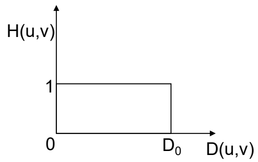
Purpose: smoothing noise

H(u,v) - Ideal low-pass filter

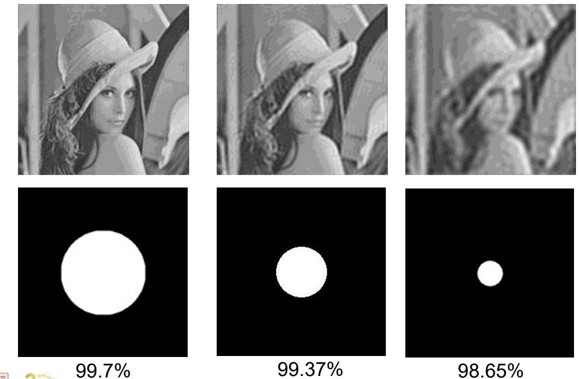
$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases} \qquad D(u,v) = \sqrt{u^2 + v^2}$$

D_0 = cut off frequency





Blurring - Ideal low-pass filters

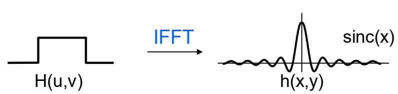


The ringing problem

$$G(u,v) = F(u,v) \cdot H(u,v)$$

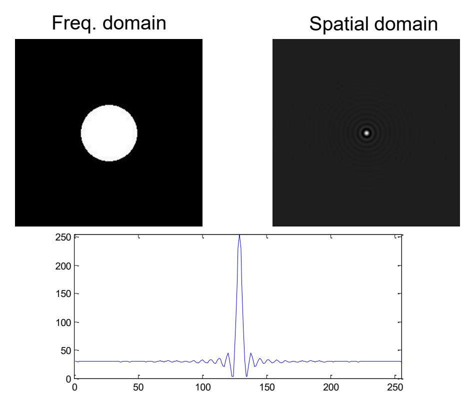
Convolution Theorem

$$g(x,y) = f(x,y) * h(x,y)$$

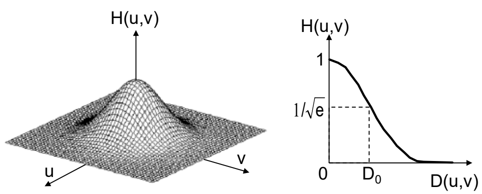


$\uparrow D_0 \longrightarrow \downarrow$ Ringing radius + $\downarrow$ blur

The ringing problem



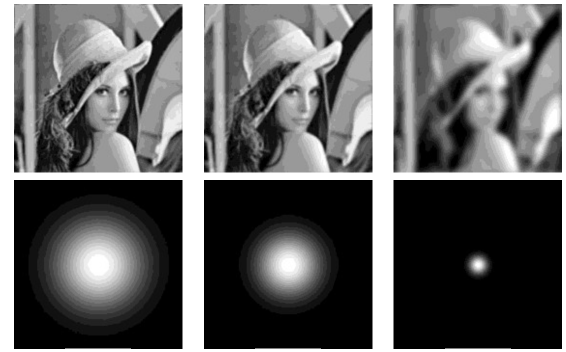
H(u,v) - Gaussian filter



$H(u,v) = e^{-D^2(u,v)/(2D^2_0)}$ $D(u,v) = \sqrt{u_2 + v_2}$

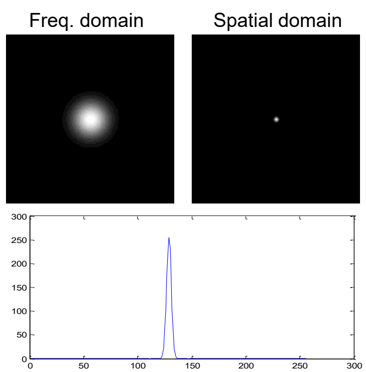
Softer Blurring + no Ringing

Blurring - Gaussain lowpass filter



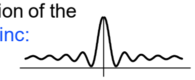
99.11% 98.74% 96.44%

The Gaussian lowpass filter



Blurring in the Spatial Domain

Averaging = convolution with $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ = point multiplication of the transform with sinc:



Gaussian Averaging = convolution with $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

= point multiplication of the transform with a gaussian.

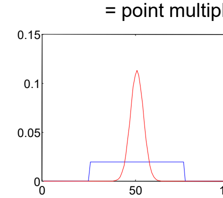
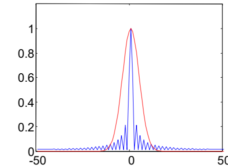


Image Domain



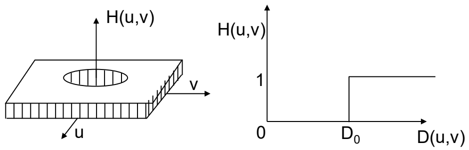
Frequency Domain

High-pass filter

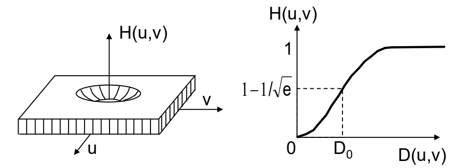
H(u,v) - Ideal Filter

$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 \\ 1 & D(u,v) > D_0 \end{cases} \qquad D(u,v) = \sqrt{u^2 + v^2}$$

D_0 = cut off frequency



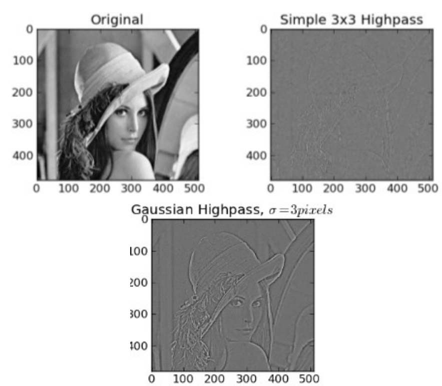
High-pass gaussian filter



$$H(u,v) = 1 - e^{-D^2(u,v)/(2D^2_0)}$$

$$D(u,v) = \sqrt{u_2 + v_2}$$

High-pass filtering

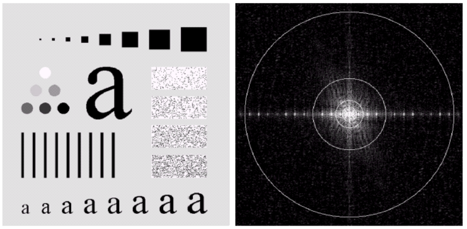


$$h_1(3x3) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

High pass filtering

Original image

Spectre with filters

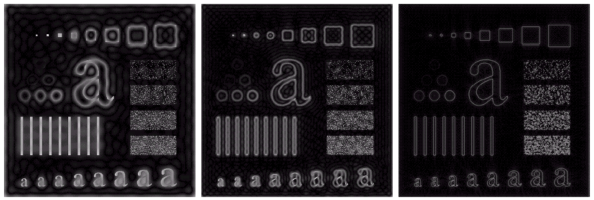


a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

High pass filtering



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

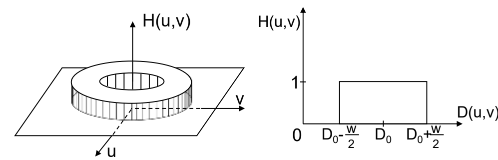
Band-pass filtering

$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 - \frac{w}{2} \\ 1 & D_0 - \frac{w}{2} \le D(u,v) \le D_0 + \frac{w}{2} \\ 0 & D(u,v) > D_0 + \frac{w}{2} \end{cases}$$

$D(u,v) = \sqrt{u^2 + v^2}$

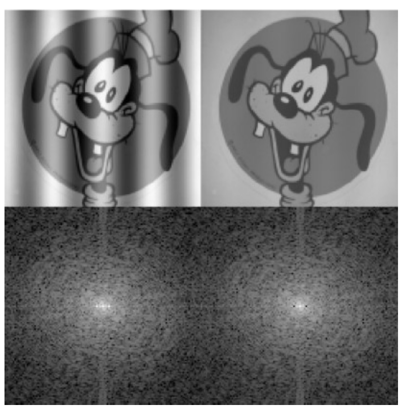
D_0 = cut off frequency

w = band-width



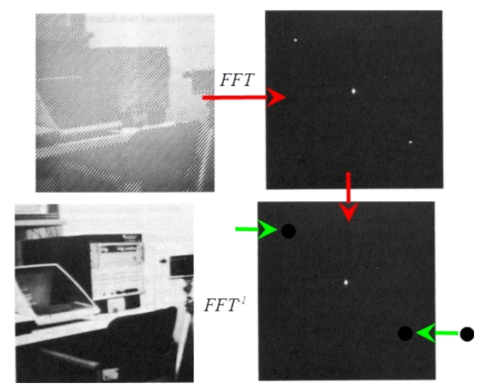
Can be obtained by multiplying the filter functions of **a low-pass and of a high-pass** in the frequency domain

Removing sinus noise

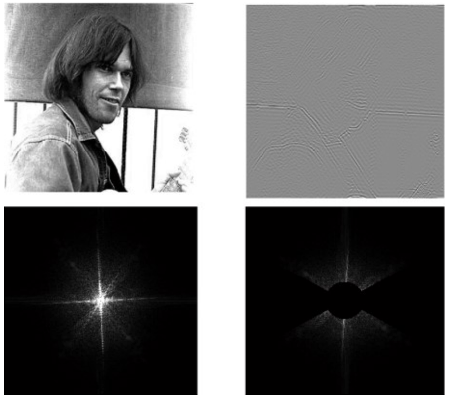


Brayer

Removing sinus noise



High-pass filtering + orientation



Hybrid Images

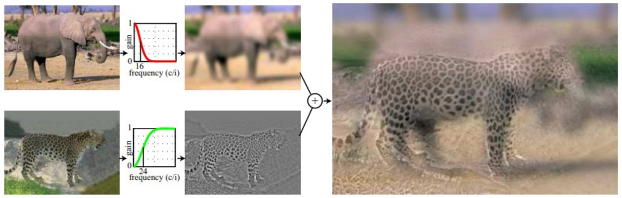


Figure 2: hybrid images are generated by superimposing two images at two different spatial scales: the low-spatial scale is obtained by filtering one image with a low-pass filter, and the high spatial scale is obtained by filtering a second image with a high-pass filter. The final hybrid image is composed by adding these two filtered images.

A. Oliva, A. Torralba, P.G. Schyns, SIGGRAPH 2006

Content

- Rappel: digital image representation
- Point Processing
- Convolution and Linear filtering
- More neighborhood operators
- Image transforms
  - Frequency domain
  - PCA (additional reading)
    - PCA
    - Example of using PCA for face recognition