Frequency

Vinay P. Namboodiri

January 2, 2014

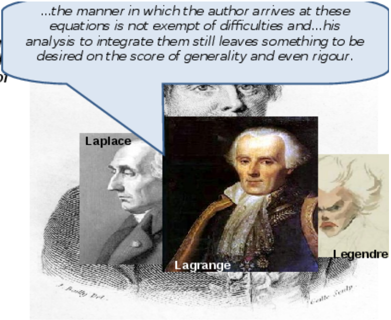
Input for slides includes content by Steve Seitz, Trevor Darrell, James Hays, Kristen Grauman, Antonio Torralba, Li Fei Fei, David Jacobs, Derek Hoiem

Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

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

- Don't believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it's (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions

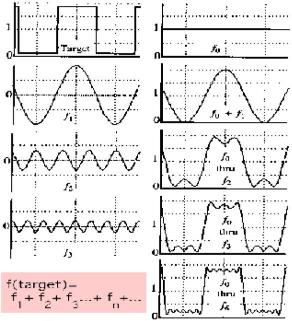


Example

Our building block:

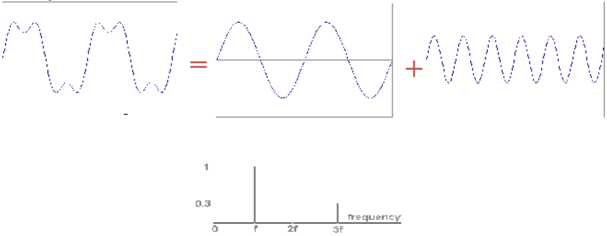
$$A\sin(\omega x + \phi)$$

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



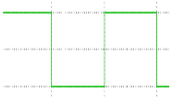
Example

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

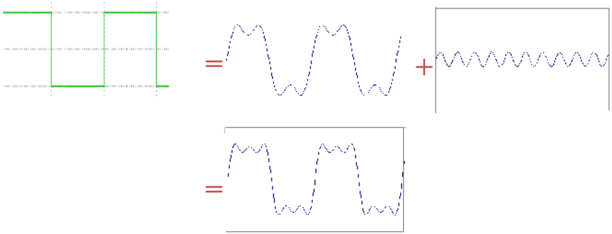


Slides: Efros

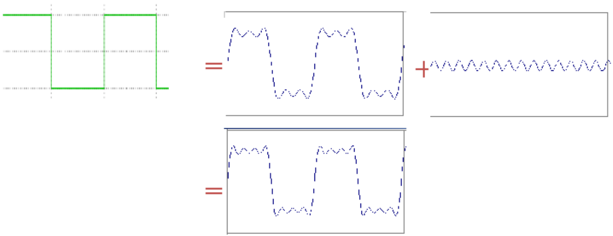
Example



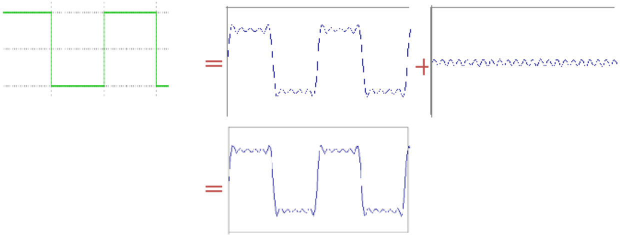
Example



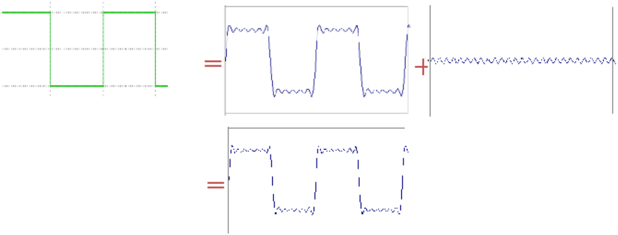
Example



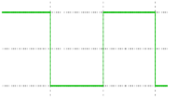
Example



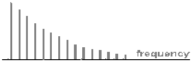
Example



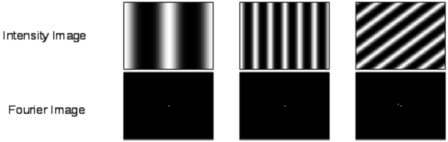
Example



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

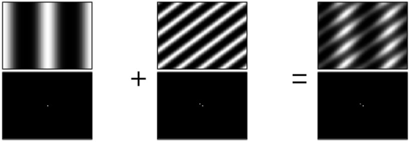


Example in image



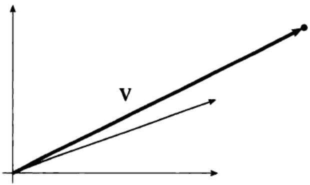
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Example in image

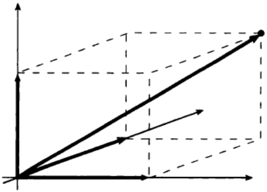


http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

Projection of points in Space

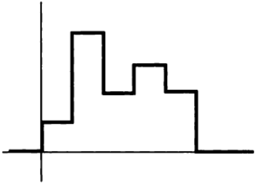


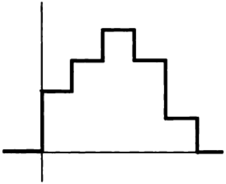
(a)



(b)

Projection of Functions

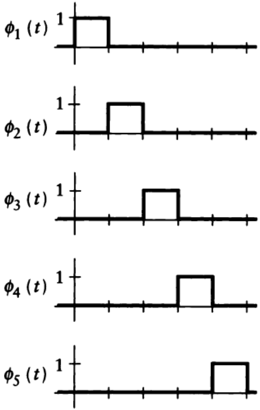

(1, 4, 2, 3, 2)


(2, 3, 4, 3, 1)

How to form the function in terms of basis functions?

$$f(t) = c_1\phi_1(t) + c_2\phi_2(t) + ... + c_n\phi_n(t) \tag{1}$$

Projection of Functions



Fourier transform

In the same way we obtain Fourier transform as projection of a signal $h(x)$ on to a sinusoidal basis function
In the continuous domain it is given by

$$H(\omega) = \int_{-\infty}^{\infty} h(x)e^{(-j\omega x)}dx \tag{2}$$

Illustration of Fourier transform

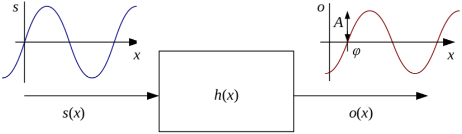


Figure 3.24 The Fourier Transform as the response of a filter $h(x)$ to an input sinusoid $s(x) = e^{j\omega x}$ yielding an output sinusoid $o(x) = h(x) * s(x) = Ae^{j\omega x + \phi}$.

The Fourier transform is a tabulation of the magnitude and phase response at each frequency

$$H(\omega) = F\{h(x)\} = Ae^{j\phi} \tag{3}$$

where A is the magnitude and ϕ is the phase response
The magnitude encodes how much signal there is at a particular frequency. The phase encodes spatial information indirectly

Discrete Fourier transform

Now, in the discrete domain we have samples only at discrete intervals, i.e. $h(x) = h[1], h[2], ..., h[n]$
Therefore, for discrete signals, we have

$$H\omega = \int_0^{(N-1)T} h(x)e^{-j\omega x}dx \quad (4)$$

$$H\omega = h[0]e^{-j0} + h[1]e^{-j\omega T} + ... + f[k]e^{-j\omega kT} \quad (5)$$

And therefore the Fourier transform in discrete domain is given by

$$H(\omega) = \sum_0^{N-1} h(x)e^{-j\omega kT} \quad (6)$$

Convolution property of Fourier Transform

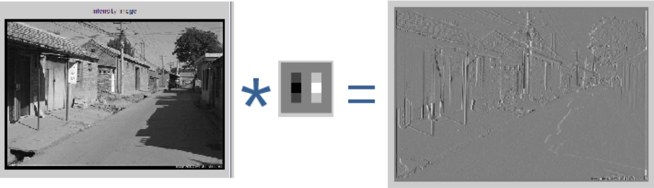
- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$F[g * h] = F[g]F[h] \tag{7}$$

- Convolution in spatial domain is equivalent to multiplication in frequency domain

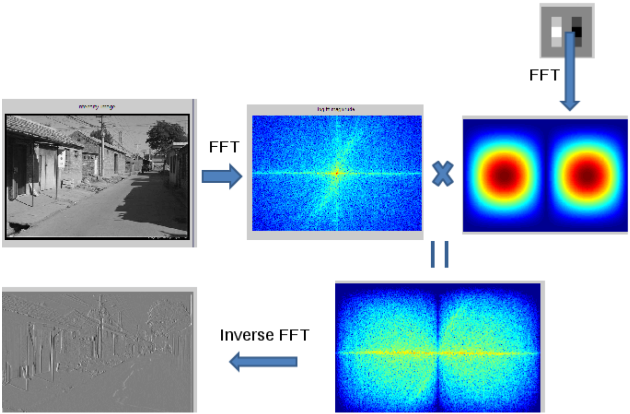
$$g * h = F^{-1}[F[g]F[h]] \tag{8}$$

Filtering in spatial domain



1	0	-1
2	0	-2
1	0	-1

Filtering in frequency domain



FFT in Matlab

- Filtering with fft

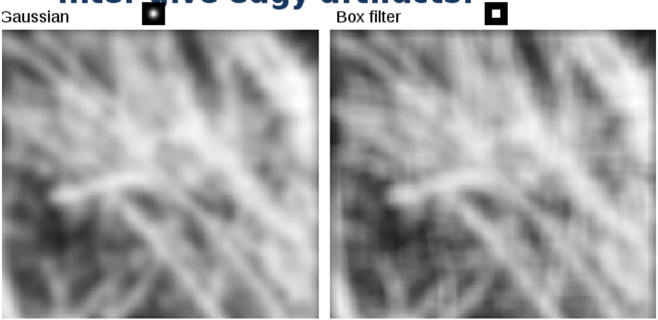
```
im = double(imread('…'))/255;
im = rgb2gray(im); % "im" should be a gray-scale floating point image
[imh, imw] = size(im);

hs = 50; % filter half-size
fil = fspecial('gaussian', hs*2+1, 10);

fftsize = 1024; % should be order of 2 (for speed) and include padding
im_fft  = fft2(im,  fftsize, fftsize);                       % 1) fft im with padding
fil_fft = fft2(fil, fftsize, fftsize);                       % 2) fft fil, pad to same size as
image
im_fil_fft = im_fft .* fil_fft;                              % 3) multiply fft images
im_fil = ifft2(im_fil_fft);                                  % 4) inverse fft2
im_fil = im_fil(1+hs:size(im,1)+hs, 1+hs:size(im, 2)+hs); % 5) remove padding
```

- Displaying with fft

```
figure(1), imagesc(log(abs(fftshift(im_fft)))), axis image, colormap jet
```

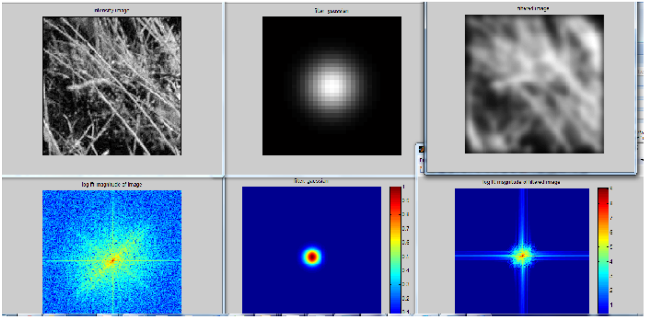
Filtering

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?



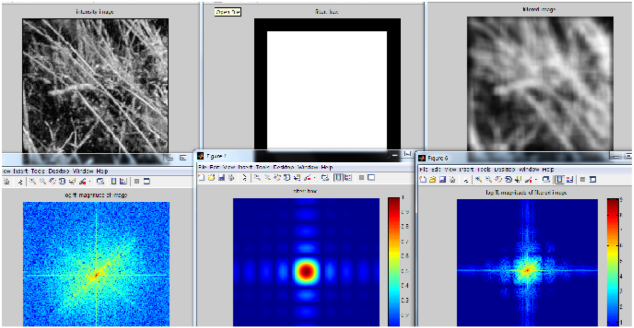
Gaussian Filter

Gaussian



Box Filter

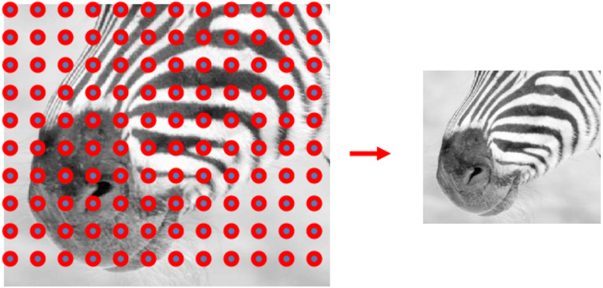
Box Filter



Sampling

Why does a lower resolution image still make sense to us? What do we lose?

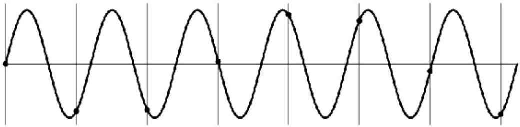
Subsampling by a factor of 2



Throw away every other row and
column to create a 1/2 size image

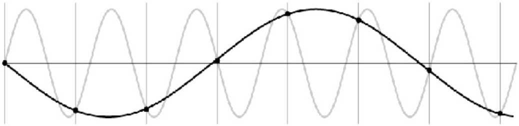
Aliasing Problem

- 1D example (sinewave):



Aliasing Problem

- 1D example (sinewave):



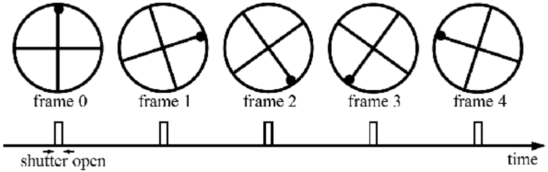
Aliasing Problem

- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - "Wagon wheels rolling the wrong way in movies"
  - "Checkerboards disintegrate in ray tracing"
  - "Striped shirts look funny on color television"

Aliasing in Video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

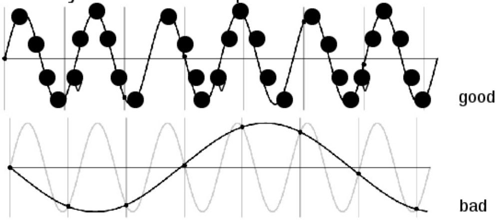
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Nyquist Shannon Sampling theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



Anti-aliasing

Solutions:

- Sample more often
- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it's better than aliasing
  - Apply a smoothing filter

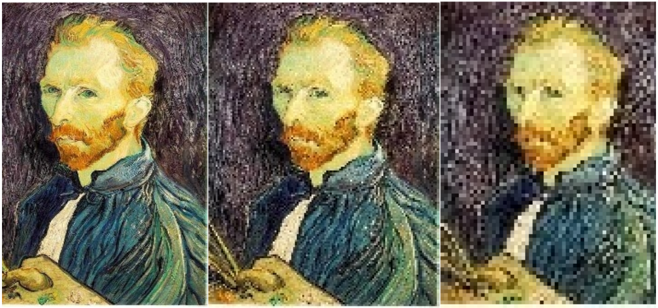
Algorithm for downsampling

1. Start with image(h, w)
2. Apply low-pass filter

   im_blur = imfilter(image, fspecial('gaussian', 7, 1))
3. Sample every other pixel

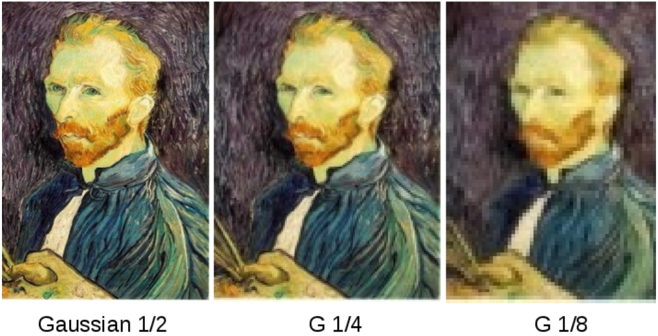
   im_small = im_blur(1:2:end, 1:2:end);

Sub-sampling without pre-filtering



1/2   1/4 (2x zoom)   1/8 (4x zoom)

Sub-sampling with pre-filtering

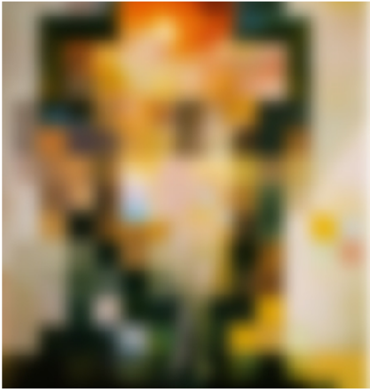


Hybrid Images

Salvador Dali invented Hybrid Images?

Salvador Dali

Hybrid Images

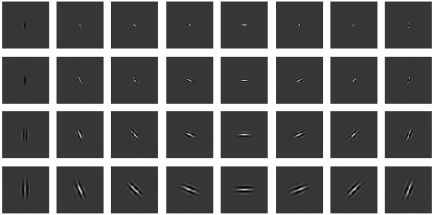


Hybrid Images

- A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images," SIGGRAPH 2006

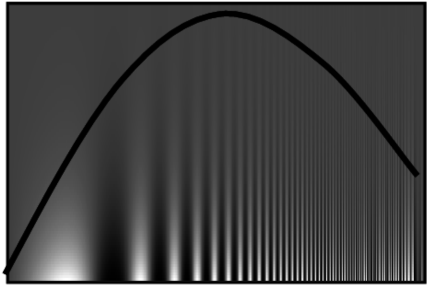
Clues from Human perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-high frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it

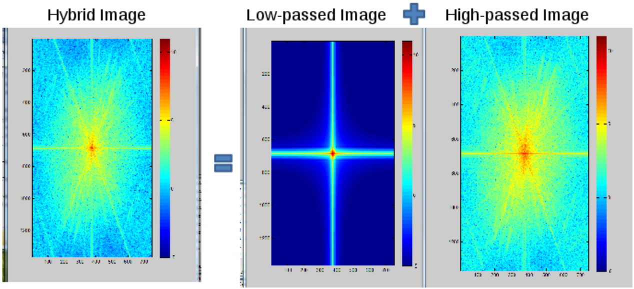


Early Visual Processing: Multi-scale edge and blob filters

Campbell-Robson contrast sensitivity curve



Hybrid Images



Example of Hybrid Images

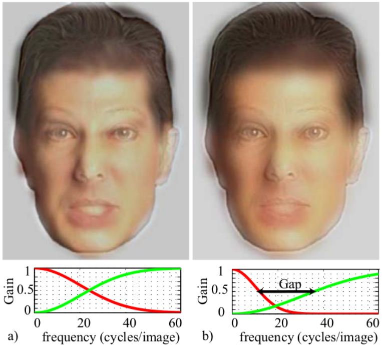


Figure 5: An angry man or a thoughtful woman? Both hybrid images are produced by combining the faces of an angry man (low spatial frequencies) and a stern woman (high spatial frequencies). You can switch the percept by watching the picture from a few meters. a) Bad hybrid image. The image looks ambiguous from up close due to the filter overlap. b) Good Hybrid image.

Example of Hybrid Images

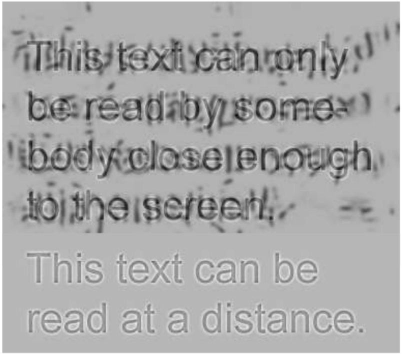


Figure 8: The hybrid font becomes invisible at few meters. The bottom text remains easy to read at relatively long distances.

Recap

- Fourier transform obtained by projection of signal onto sinusoidal signals
- Sampling of images may result in aliasing and anti-aliasing by following Nyquist theorem for sampling frequency
- Application of frequency domain transform for creating Hybrid images