Fourier vs Wavelets

Researchlab 4 Presentation

Maurice Samulski

June 27th, 2005

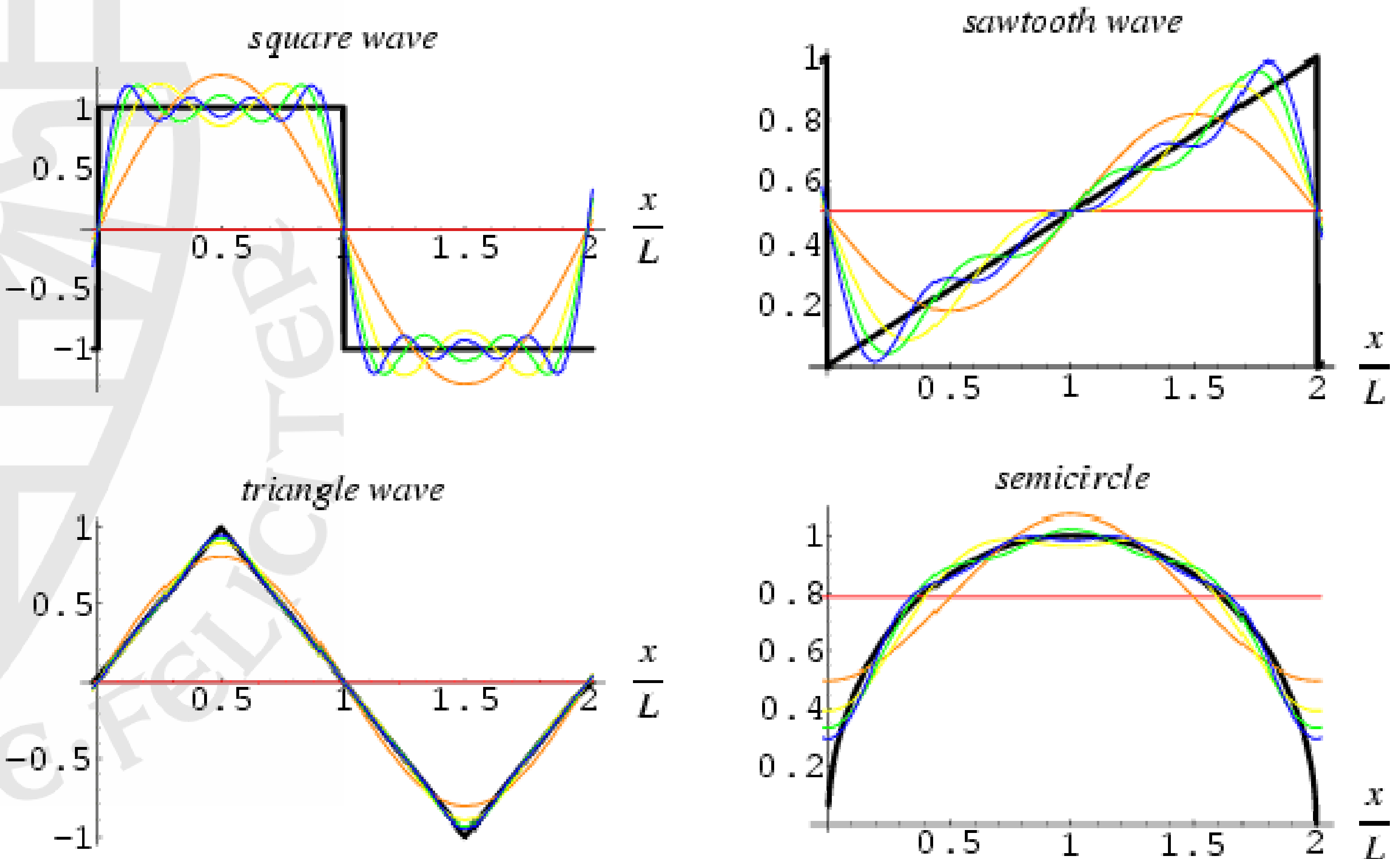
Contents

- Introduction
- Discrete Fourier Transform
- Discrete Cosine Transform
- Wavelet Transform
- Comparison between DCT and WT
- Conclusions

Fourier analysis

- Joseph Fourier 1807
- Represent functions by superposing sines and cosines with different frequencies and amplitudes
- $s(t) = 3 \sin(t) - 100 \sin(4t) - 20 \sin(200t)$

Fourier analysis



Discrete Fourier Transform (DFT)

- DFT of image f(x,y) with size m x n

$$F(u,v) = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x,y) \cdot e^{-2\pi i (\frac{ux}{m} + \frac{vy}{n})}$$

$$\texttt{for}\, u = 0, \ldots, m-1 \,\texttt{and for}\, v = 0, \ldots, n-1$$

$$e^{-2\pi i f} = cos(2\pi f) + i \cdot sin(2\pi f)$$

Discrete Fourier Transform (DFT)

- Inverse DFT of F(u,v)

$$f(x,y) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} F(u,v) \cdot e^{2\pi i(\frac{ux}{m} + \frac{vy}{n})}$$

$$\texttt{for}\, x = 0, \ldots, m-1 \; \texttt{and} \;\; \texttt{for}\, y = 0, \ldots, n-1$$

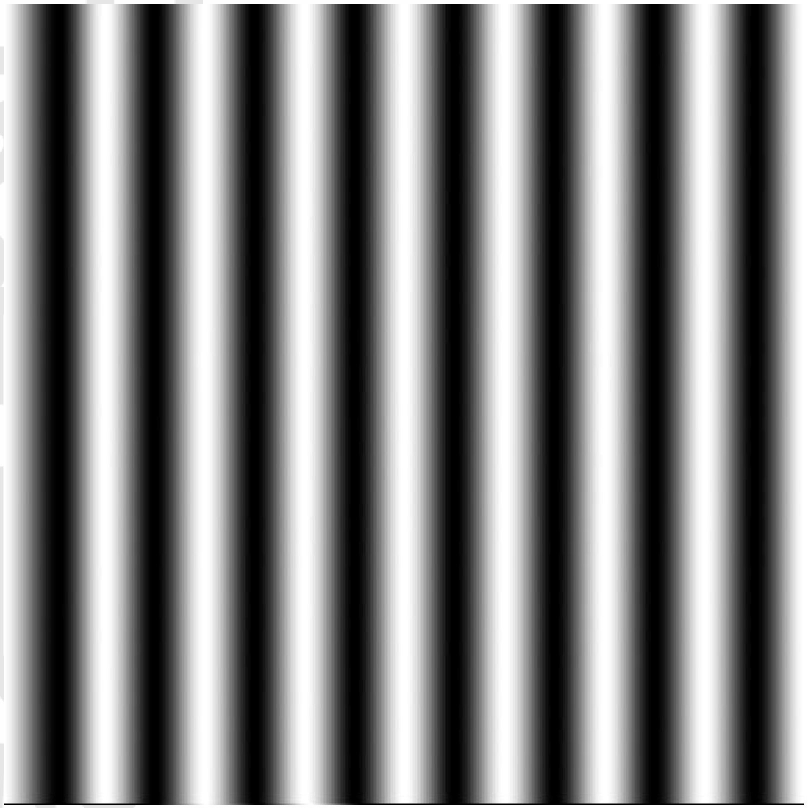
Discrete Fourier Transform

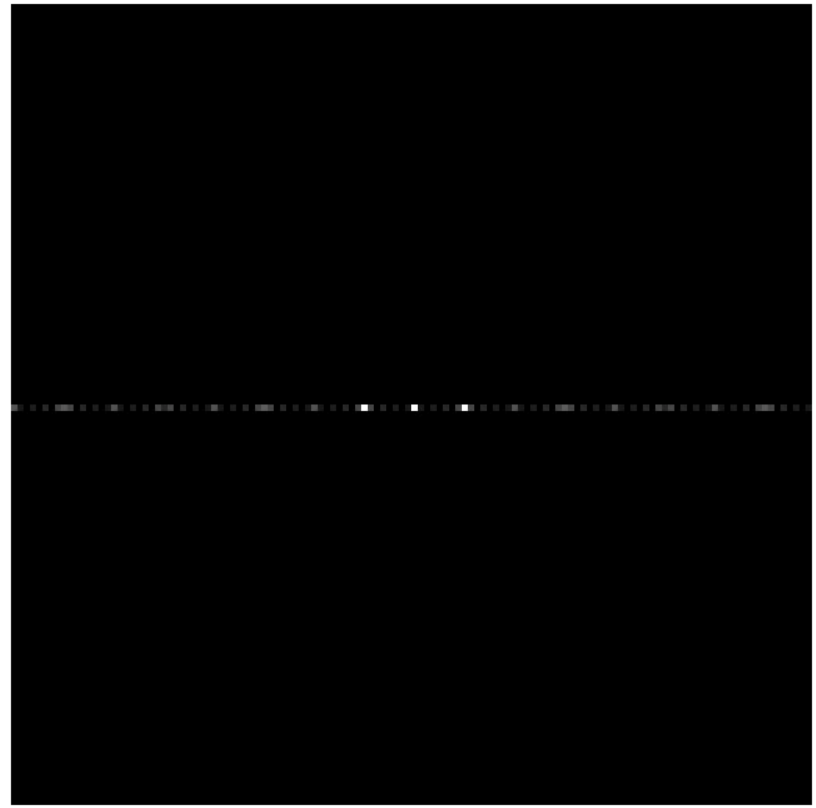
- Image f(x,y) is real
- Fourier transform F(u,v) is complex
- F(u,v) often represented as

$$magnitude(F(u,v)) = \sqrt{R^2(u,v) + I^2(u,v)}$$

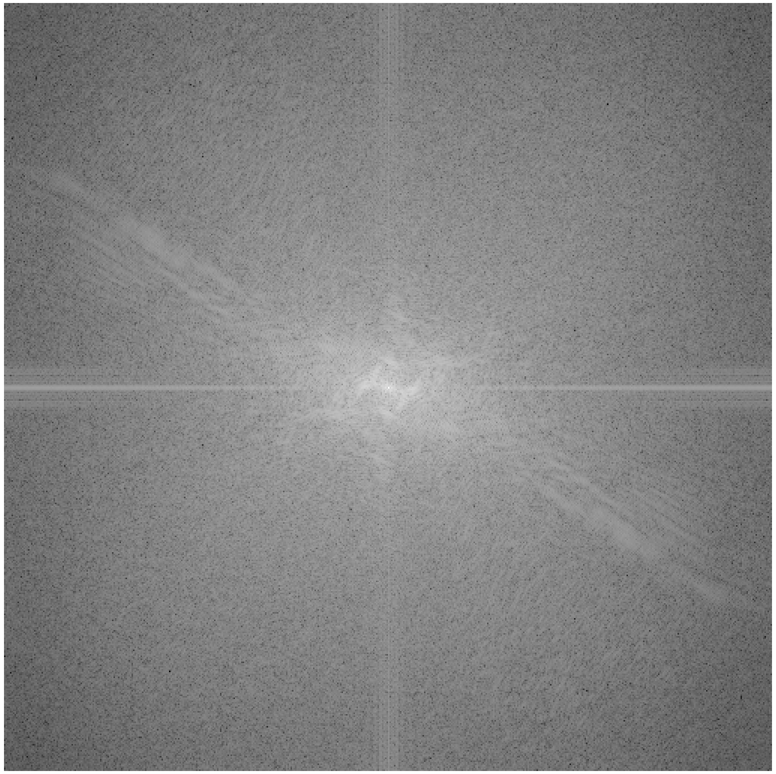
$$phase(F(u,v)) = \tan^{-1}[\frac{I(u,v)}{R(u,v)}]$$

Discrete Fourier Transform

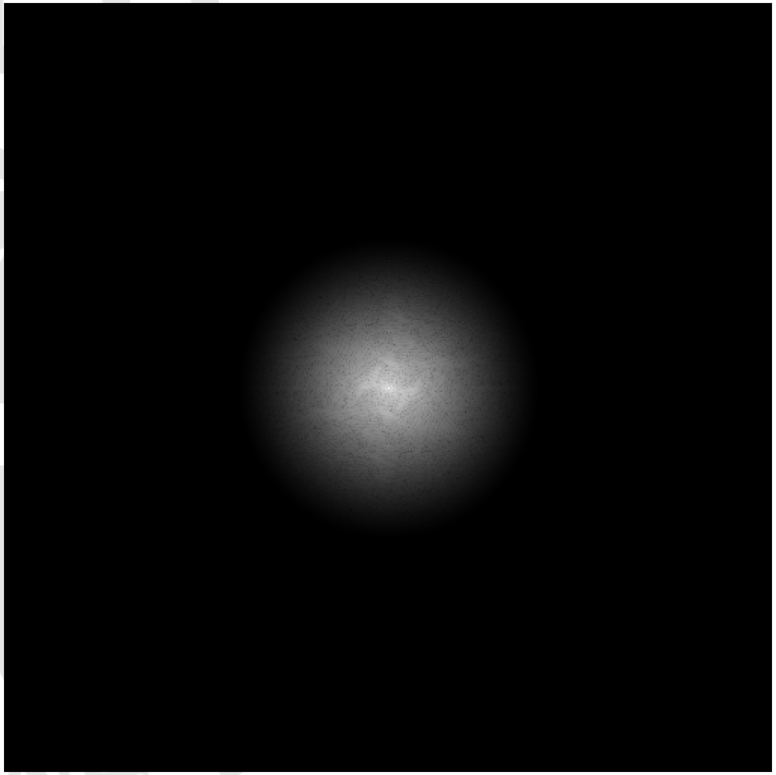




Discrete Fourier Transform



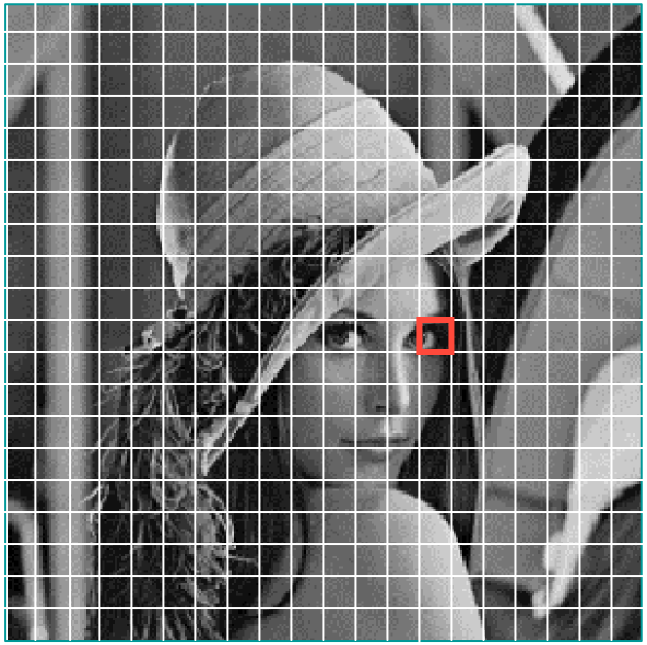
Discrete Fourier Transform



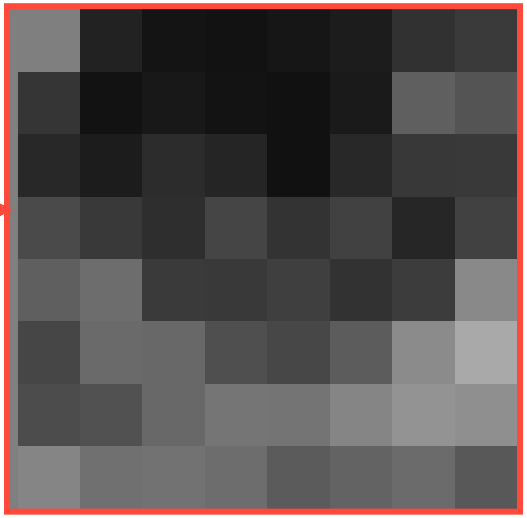
Discrete Cosine Transform (DCT)

- Very similar to the discrete Fourier transform, but
  - Uses only real numbers
  - Decomposes a function into a series of even cosine components only
  - Different ordering of coefficients
- Computationally cheaper than DFT and therefore very commonly used in image processing, eg JPEG and MPEG

(1) Divide image into 8x8 blocks

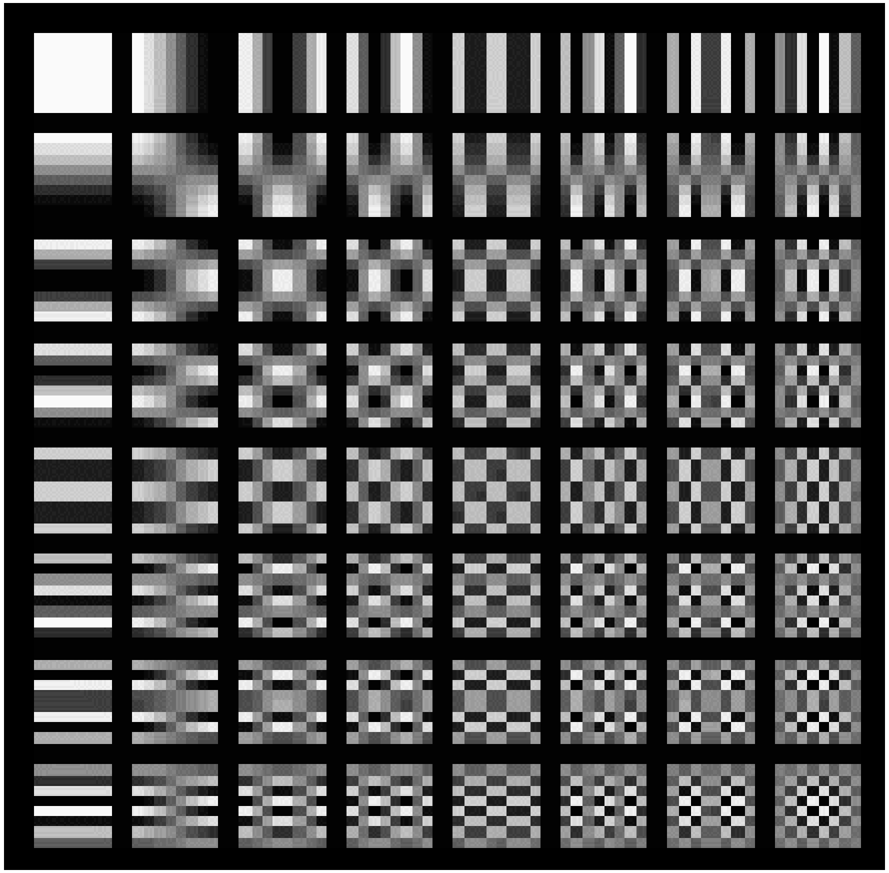


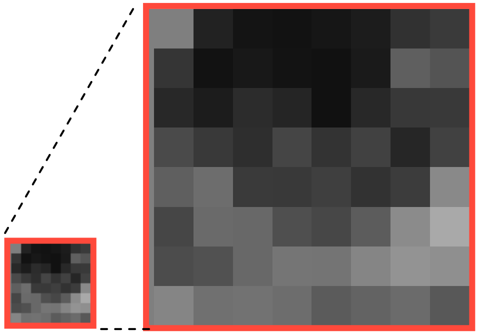
Input image



8x8 block

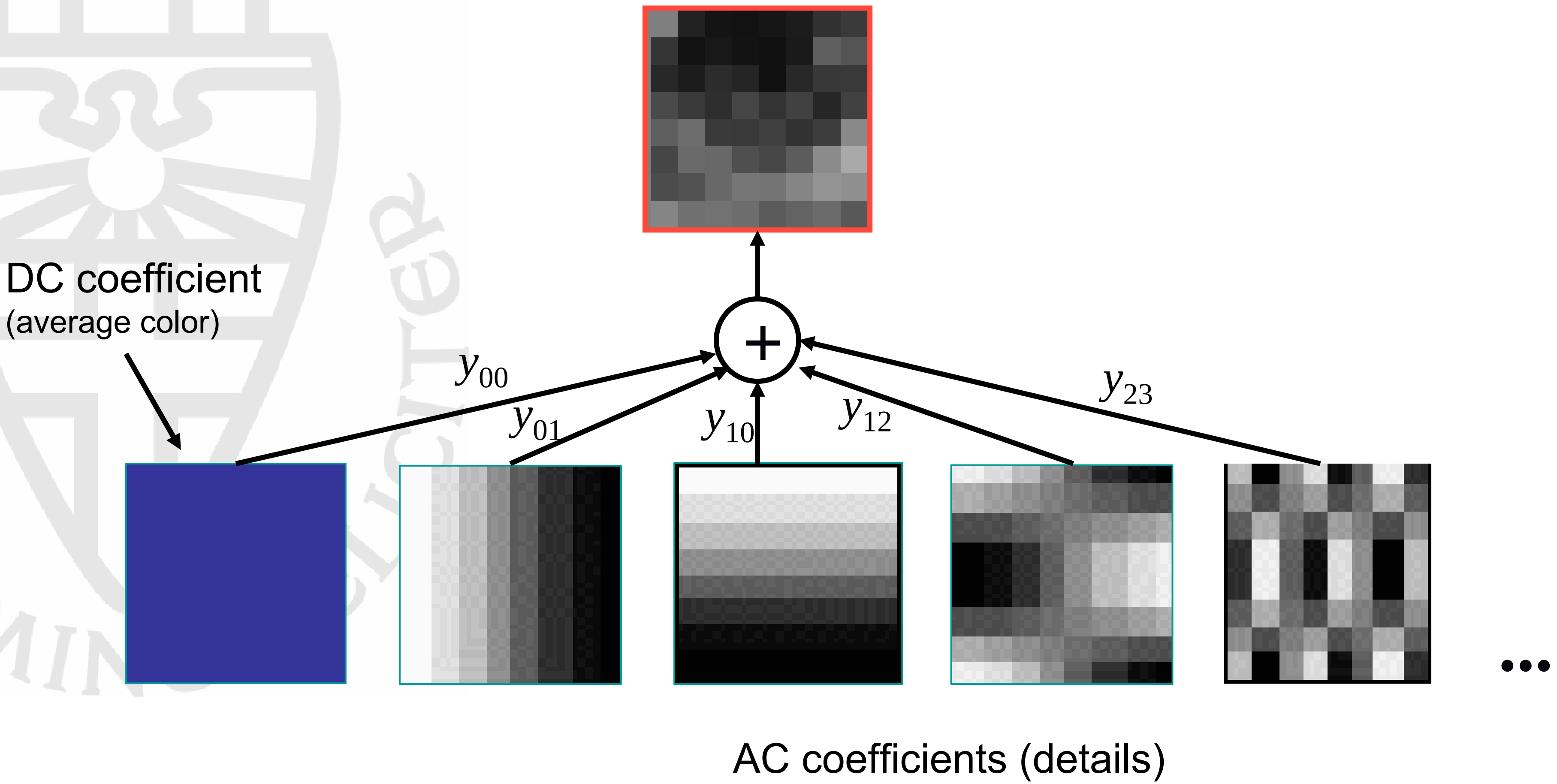
(2a) 2-D DCT basis functions





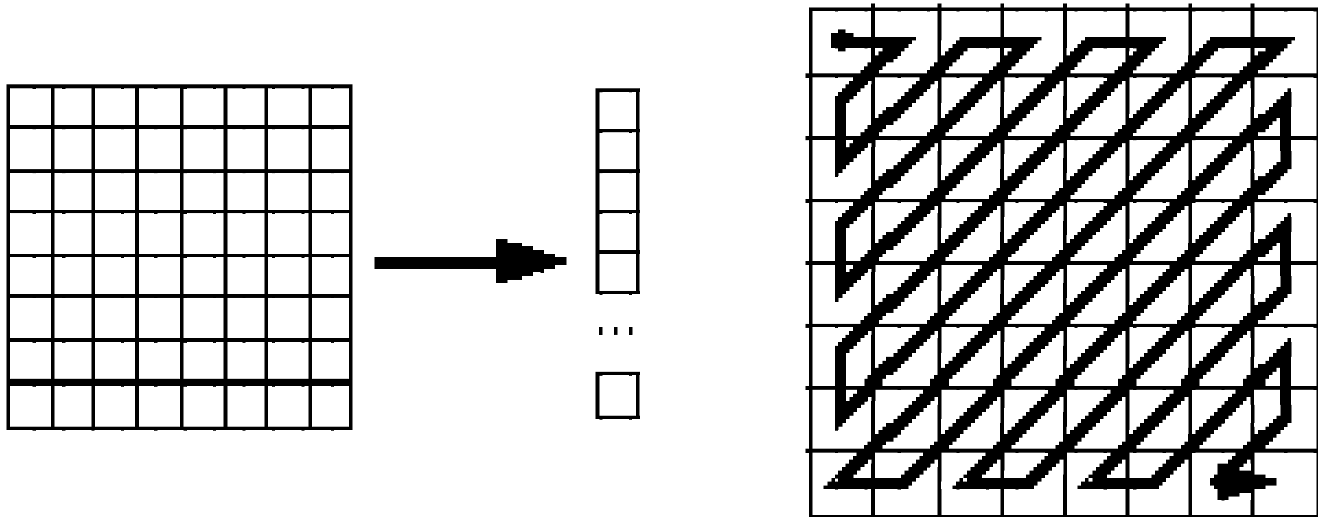
8x8 block

(2b) 2-D Transform Coding



(3) Zig-zag ordering DCT blocks

- Why? To group low frequency coefficients in top of vector.
- Maps 8 x 8 to a 1 x 64 vector.



DCT compression

- Because human eye is most sensitive to low frequencies, less sensitive to high frequencies, we can truncate the coefficients which represent these high frequencies
- The lower quality setting, the more coefficients are truncated
- Lesser coefficients mean less detail of the block which leads to the famous blocking artifact

Wavelets

- The major advantage of using wavelets is that they can be used for analyzing functions at various scales
- It stores versions of an image at various resolutions, which is very similar how the human eye works.
- As you zoom in at smaller and smaller scales, you can find details that you did not see before.

Haar wavelet example (1D)

- Suppose we have a one-dimensional data set containing eight pixels:

  [10 8 6 8 1 5 8 2]

- We can represent this image in the Haar basis by computing a wavelet transform, by averaging the pixels together pairwise:

  [9 7 3 5]

- Clearly, some information has been lost in this averaging process, we need to store detail coefficients:

  [1 -1 -2 1]

Haar wavelet example (1D)

- The full decomposition will look like

Resolution	Averages	Detail Coefficients
8	[10 8 6 8 1 5 6 4]	
4	[9 7 3 5]	[1 −1 −2 1]
2	[8 4]	[1 −1]
1	[6]	[2]

Table 1: Decomposition of 8-pixel image

- We will store this as follows: [6 2 1 −1 1 −1 −2 1]
- No information has been gained or lost by this process

Haar wavelet example (1D)

- The full decomposition will look like

Resolution	Averages	Detail Coefficients
8	[10 8 6 8 1 5 6 4]	
4	[9 7 3 5]	[1 −1 −2 1]
2	[8 4]	[1 −1]
1	[6]	[2]

Table 1: Decomposition of 8-pixel image

- This transform will be stored as:

[6 2 1 −1 1 −1 −2 1]

- No information has been gained or lost by this process

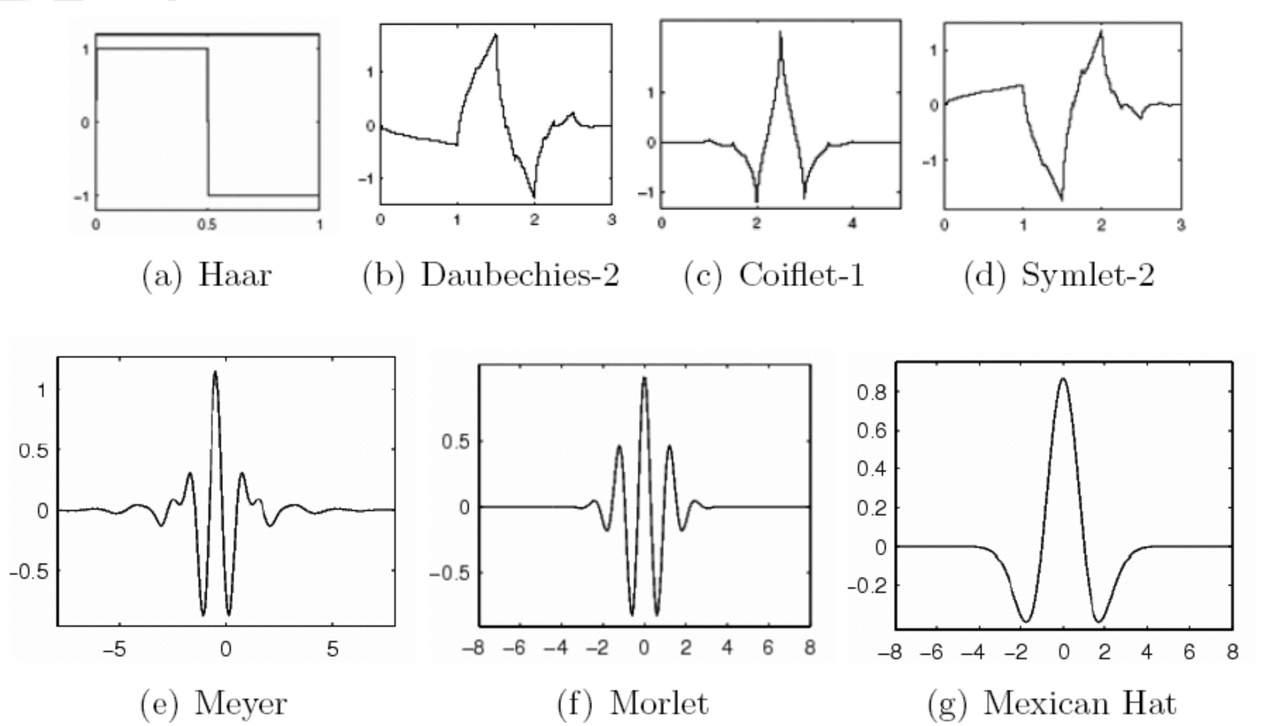
Haar wavelet

- This may look wonderful and all, but what good is compression that takes eight values and compresses it to eight values?
- Pixel values are similar to their neighbors
- The image can be compressed by removing small coefficients from this transform

- The one-dimensional Haar Transform can be easily extended to two-dimensional
- Input matrix instead of an input vector
  - apply the one-dimensional Haar transform on each row
  - apply the one-dimensional Haar transform on each column

Other wavelets

- The Haar wavelet uses simple basis functions (discontinuous) for scaling and determining detail coefficients
- Not suitable for smooth functions



JPEG vs JPEG2000

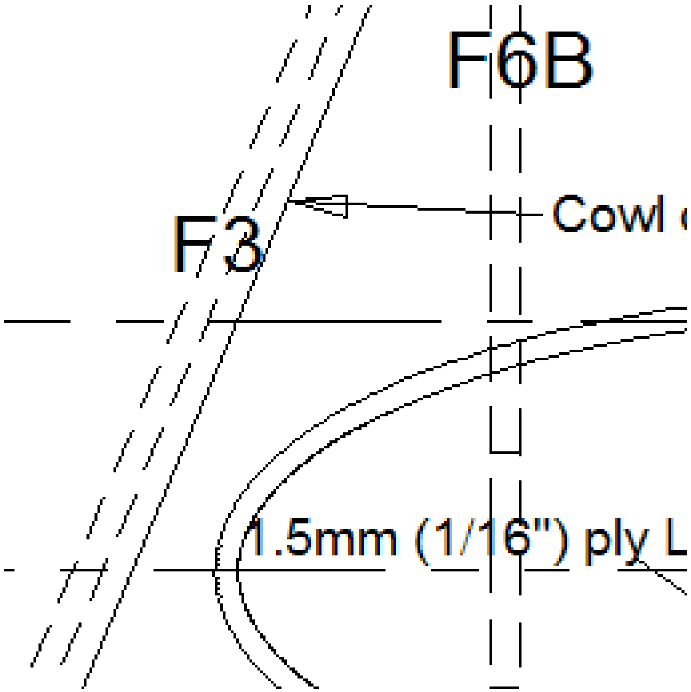
- Generally, there are two visible damages caused by image compression:
  - Blocking artifacts: artificial horizontal and vertical borders between blocks
  - Blur: loss of fine detail and the smearing of edges

Test: Image quality

- Test results are subjective
- With ‘normal’ compression (2+ bits/pixel), quality advantage of JPEG2000 is negligible
- Real quality advantage will only become clear by using very high compression ratios (0.5 or less b/p)
- At 0.25 b/p, JPEG images begin to look like a mosaic while with JPEG2000 it gets a elegant blur across the image
- JPEG2000 image files tend to be 20 to 60% smaller than their JPEG counterparts for the same subjective image quality

Test: Image quality (Original)

Lena Original (512x512x24b)



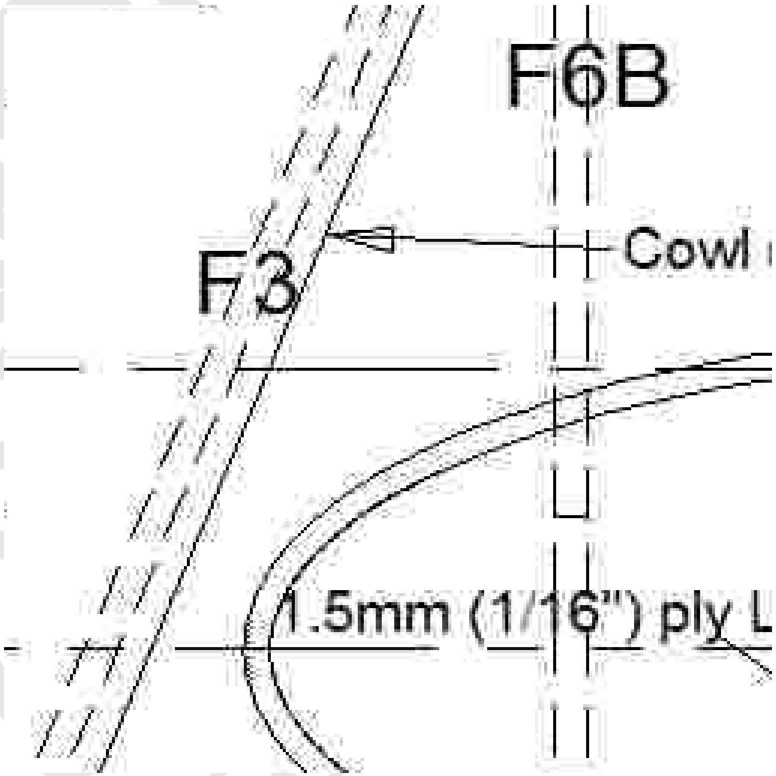
Building Plan (small piece)

Results: Image quality (Lena)

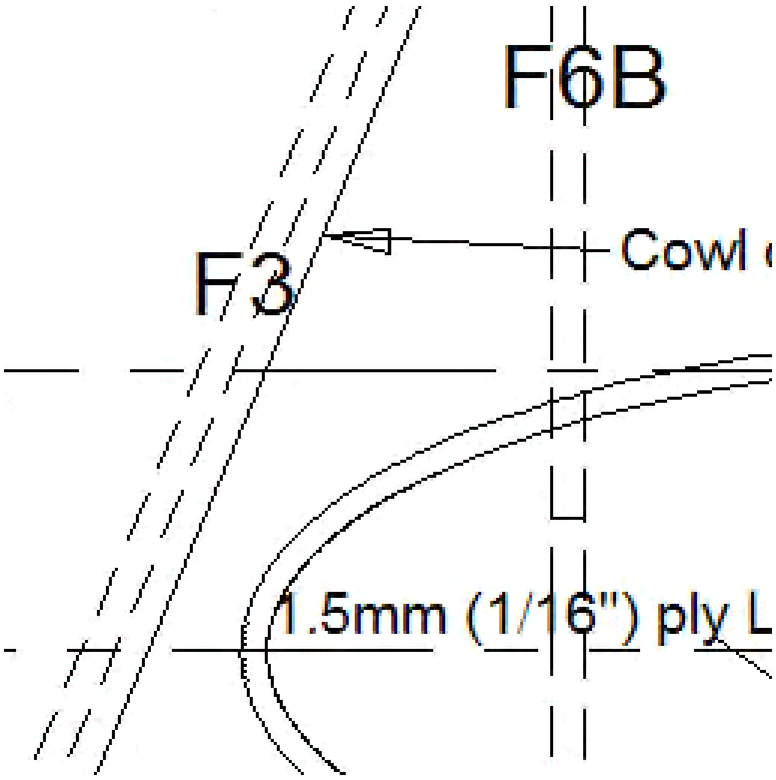
JPEG (0.2 b/p)

JPEG2000 (0.2 b/p)

Results: Image quality (Building plan)



JPEG (0.2 b/p)



JPEG2000 (0.2 b/p)

Results: Performance

- Price to pay: considerable increase in computational complexity and memory usage

Test image	Uncompressed size	Resolution	Color depth	Quality	JPEG2000 time	JPEG time
Construction plan	34 MB	5000x3477	16bit	0.75bpp	13.49 sec	1.98 sec
Lena	786 KB	512x512	24bit	0.75bpp	0.94 sec	0.37 sec
Tulips	1.2 MB	768x512	24bit	0.75bpp	0.78 sec	0.33 sec
Monarch	1.2 MB	768x512	24bit	0.75bpp	0.77 sec	0.36 sec

Table 2: Performance table JPEG vs JPEG2000

Conclusions

- JPEG2000 works better with sharp spikes in images
- Quality advantages are really visible when compressing with very high compression ratios
- Only to be used with very large datasets like fingerprints, MRI scans, building plans, etc.
- You can choose between different wavelet basis functions to get the optimal result for a specific application
- Blur isn't experienced as bad as blocking artifacts
- Time needed to compress high resolution images takes a lot of time with JPEG2000

Questions?