Lecture notes of Image Compression and Video Compression series 2005

2. Transform Coding

Prof. Amar Mukherjee

Weifeng Sun

Topics

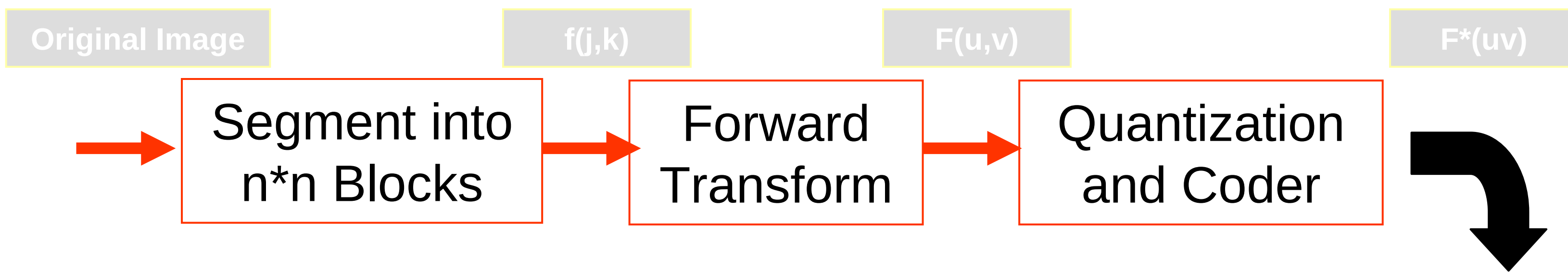
- Introduction to Image Compression
- Transform Coding
- Subband Coding, Filter Banks
- Haar Wavelet Transform
- SPIHT, EZW, JPEG 2000
- Motion Compensation
- Wireless Video Compression

Transform Coding

- Why transform Coding?
  - Purpose of transformation is to convert the data into a form where compression is easier. This transformation will transform the pixels which are correlated into a representation where they are decorrelated. The new values are usually smaller on average than the original values. The net effect is to reduce the redundancy of representation.
- For lossy compression, the transform coefficients can now be quantized according to their statistical properties, producing a much compressed representation of the original image data.

Transform Coding Block Diagram

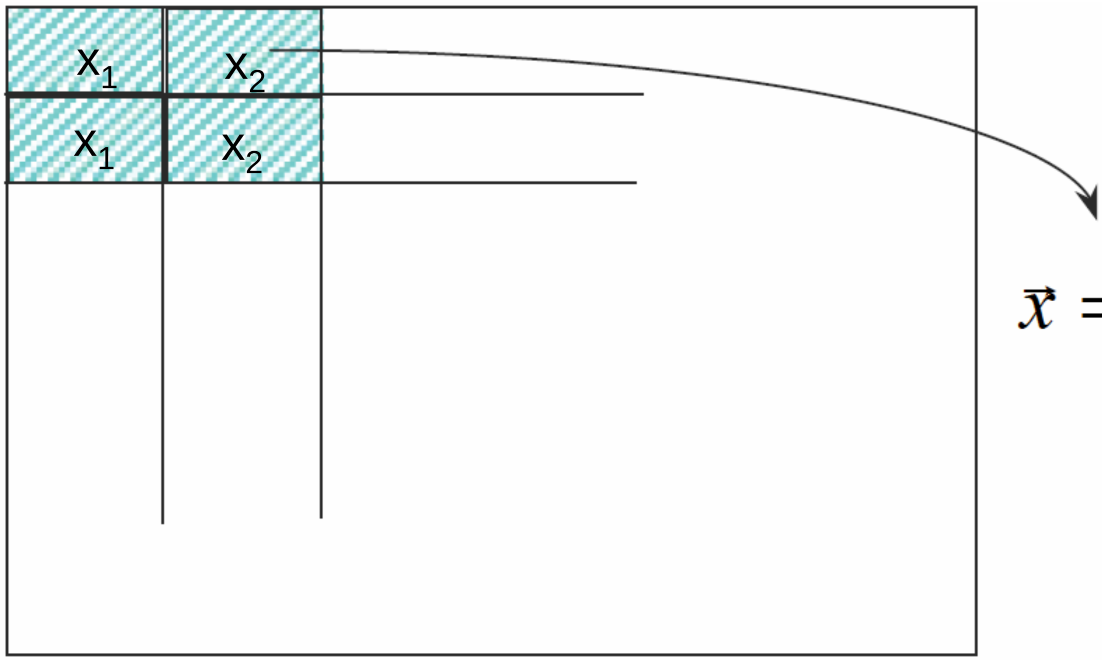
- **Transmitter**



- **Receiver**

How Transform Coders Work

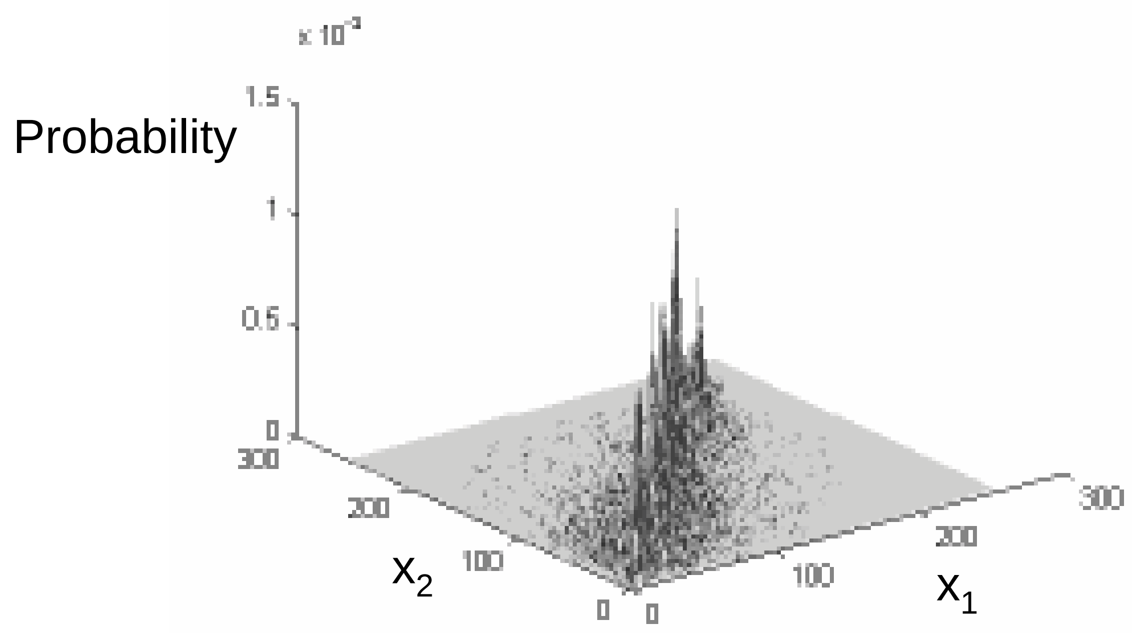
- Divide the image into 1x2 blocks
  - Typical transforms are 8x8 or 16x16



$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Joint Probability Distribution

- Observe the Joint Probability Distribution or the Joint Histogram.

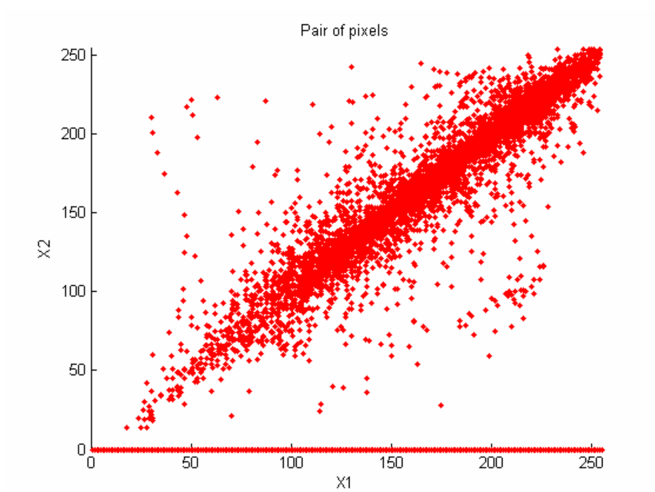


Pixel Correlation in Image[Amar]

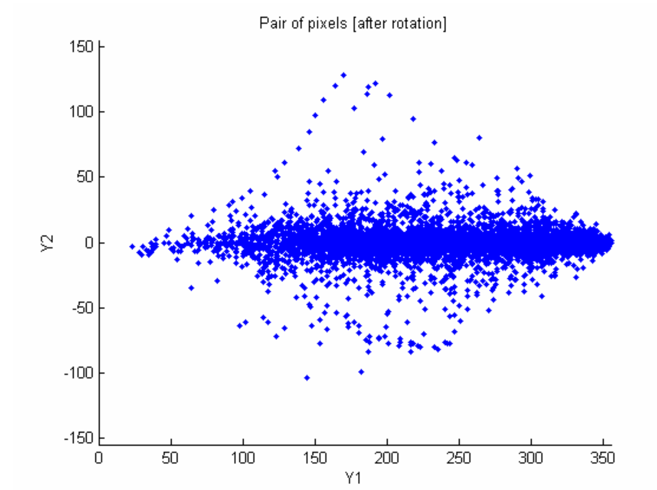
- Rotate 45^o clockwise

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Source Image: Amar



Before Rotation

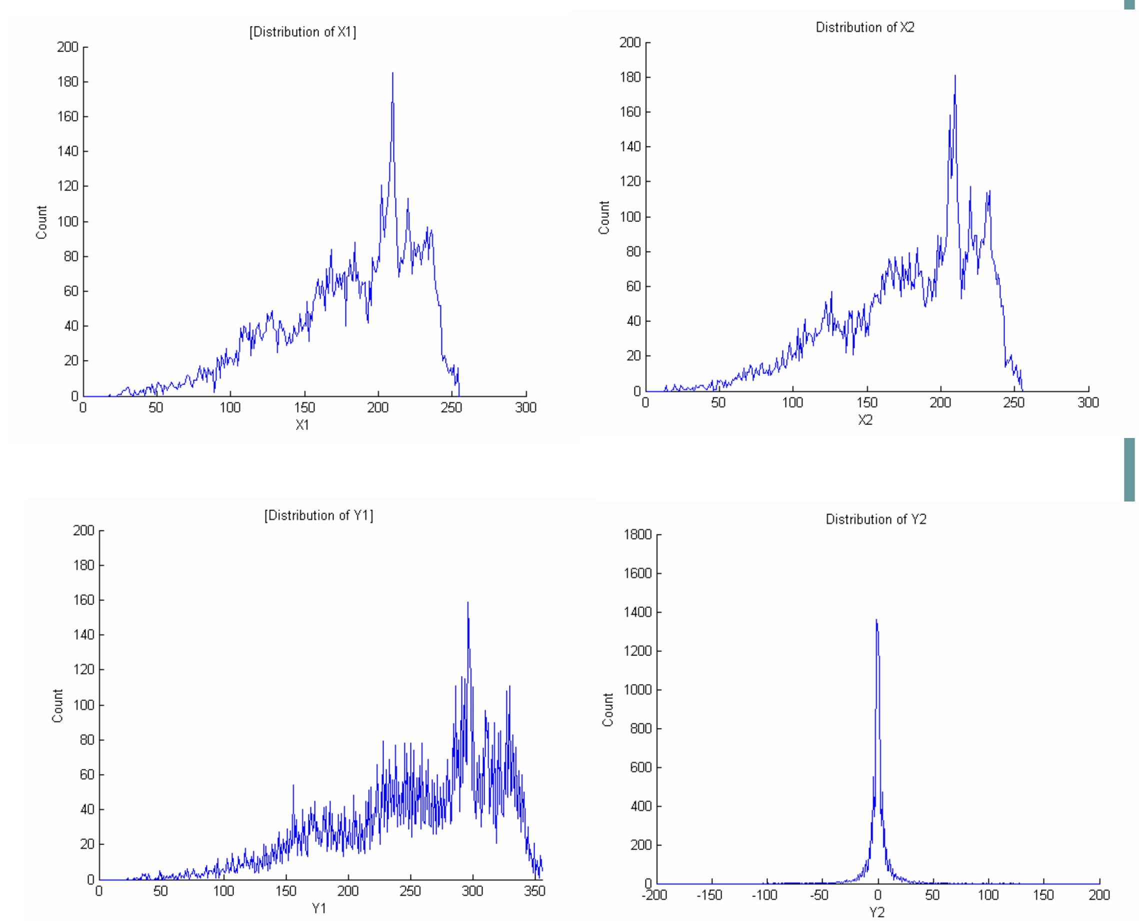


After Rotation

Pixel Correlation Map in [Amar] -- coordinate distribution

- Upper:
  Before Rotation
- Lower:
  After Rotation

- Notice the variance of Y_2 is smaller than the variance of X_2.
- Compression: apply entropy coder on Y_2.

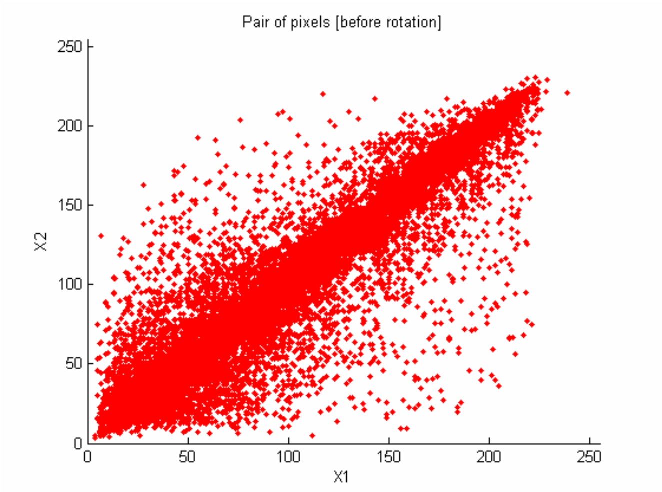


Pixel Correlation in Image[Lenna]

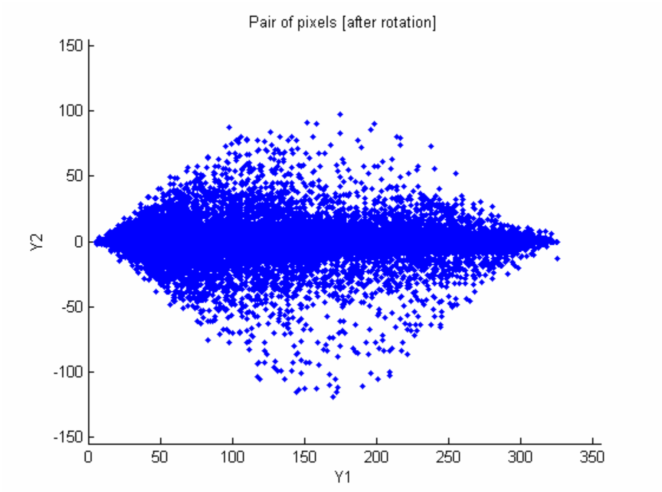
- Let's look at another example

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Source Image: Lenna



Before Rotation

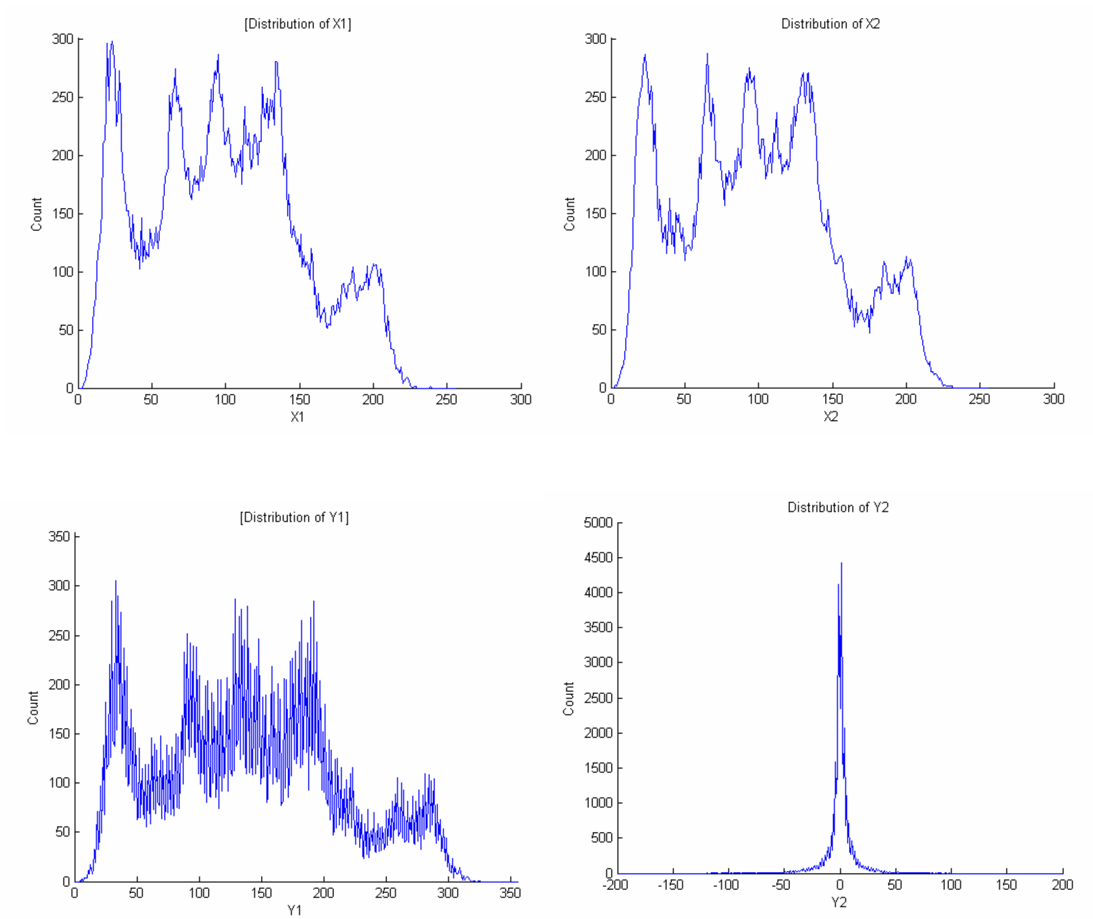


After Rotation

Pixel Correlation Map in [Lenna] -- coordinate distribution

- Upper:
  Before Rotation
- Lower:
  After Rotation

- Notice the variance of Y_2 is smaller than the variance of X_2.
- Compression: apply entropy coder on Y_2.



Rotation Matrix

- Rotated 45 degrees clockwise

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = AX = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

- Rotation matrix A

$$A = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Orthogonal/orthonormal Matrix

- Rotation matrix is orthogonal.
  - The dot product of a row with itself is nonzero.
  - The dot product of different rows is 0.

$$A_i \bullet A_j = \begin{cases} \neq 0 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

- Futhermore, the rotation matrix is orthonormal.
  - The dot product of a row with itself is **1**.

$$A_i \bullet A_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

- Example: $A = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

Reconstruct the Image

- Goal: recover X from Y.
- Since Y = AX, so $X = A^{-1}Y$
- Because the inverse of an orthonormal matrix is its transpose, we have $A^{-1} = A^T$
- So, $Y = A^{-1}X = A^TX$
- We have inverse matrix

$$A^{-1} = A^T = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Energy Compaction

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = A \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad A = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

- Rotation matrix [A] compacted the energy into Y_1.
- Energy is the variance (http://davidmlane.com/hyperstat/A16252.html)

$$\delta^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)^2 \qquad \text{where } \mu \text{ is the mean}$$

- Given: $X = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$, then $Y = AX = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \frac{\sqrt{2}}{2} \begin{bmatrix} 9 \\ 1 \end{bmatrix} = \begin{bmatrix} 6.364 \\ 0.707 \end{bmatrix}$
- The total variance of X equals to that of Y. It is 41.
- Transformation makes Y_2 (0.707) very small.
  - If we discard min{X}, we have error $4^2/41 = 0.39$
  - If we discard min{Y}, we have error $0.707^2/41 = 0.012$
  - Conclusion: we are more confident to discard min{Y}.

Idea of Transform Coding

- Transform the input pixels $X_0,X_1,X_2,...,X_{n-1}$ into coefficients $Y_0,Y_1,...,Y_{n-1}$ (real values)
  - The coefficients have the property that most of them are near zero.
  - Most of the "energy" is compacted into a few coefficients.
- Scalar quantize the coefficient
  - This is bit allocation.
  - Important coefficients should have more quantization levels.
- Entropy encode the quantization symbols.

Forward transform (1D)

- Get the sequence Y from the sequence X.
- Each element of Y is a linear combination of elements in X.

$$Y_j = \sum_{i=0}^{n-1} a_{j,i} X_i \qquad j = 0,1,\cdots n-1$$

Basis Vectors

$$\begin{bmatrix} Y_0 \\ \vdots \\ Y_{n-1} \end{bmatrix} = \begin{bmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & \ddots & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,n-1} \end{bmatrix} \begin{bmatrix} X_0 \\ \vdots \\ X_{n-1} \end{bmatrix} \qquad Y = AX$$

The element of the matrix are also called the weight of the linear transform, and they should be independent of the data (except for the KLT transform).

Choosing the Weights of the Basis Vector

- The general guideline to determine the values of A is to make Y_0 large, while remaining $Y_1, ..., Y_{n-1}$ to be small.
- The value of the coefficient will be large if weights a_{ij} reinforce the corresponding data items X_i. This requires the weights and the data values to have similar signs. The converse is also true: Y_i will be small if the weights and the data values to have dissimilar signs.

Extracting Features of Data

- Thus, the basis vectors should extract distinct features of the data vectors and must be independent orthogonal). Note the pattern of distribution of +1 and -1 in the matrix. They are intended to pick up the low and high “frequency” components of data.
- Normally, the coefficients decrease in the order of $Y_0, Y_1, \ldots, Y_{n-1}$.
- So, Y is more amenable to compression than X.

$$Y = AX$$

Energy Preserving (1D)

- Another consideration to choose rotation matrix is to conserve energy.
- For example, we have orthogonal matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad X = \begin{bmatrix} 4 \\ 6 \\ 5 \\ 2 \end{bmatrix} \qquad Y = AX = \begin{bmatrix} 17 \\ 3 \\ -5 \\ 2 \end{bmatrix}$$

- Energy before rotation: $4^2+6^2+5^2+2^2=81$
- Energy after rotation: $17^2+3^2+(-5)^2+1^2=324$
- Energy changed!
- **<u>Solution</u>: scale W by scale factor. The scaling does not change the fact that most of the energy is concentrated at the low frequency components.**

Energy Preserving, Formal Proof

- The sum of the squares of the transformed sequence is the same as the sum of the squares of the original sequence.
- Most of the energy are concentrated in the low frequency coefficients.

Energy

$$\sum_{i=1}^{n-1} Y_i^2 = Y^T Y$$

$$= (AX)^T (AX)$$

$$= X^T A^T AX$$

$$= X^T (A^T A) X$$

$$= X^T X$$

$$= \sum_{i=1}^{n-1} X_i^2$$

Orthonormal Matrix;
See page #12

Why we are interested in the orthonormal matrix?

- Normally, it is computationally difficult to get the inverse matrix.
- The inverse of the transformation matrix is simply its transpose.

$$A^{-1}=A^T$$

$$A=\frac{1}{\sqrt{8}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&1&-1&1&-1&1&-1\end{bmatrix}$$

$$B=A^{-1}=A^T=\frac{1}{\sqrt{8}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&1&-1&1&-1&1&-1\end{bmatrix}$$

Two Dimensional Transform

- From input Image I, we get D.
- Given Transform matrix A

4	6	7	8
5	2	4	4
6	3	9	6
7	5	6	9

$$A = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad D = \begin{bmatrix} 4 & 7 & 6 & 9 \\ 6 & 8 & 3 & 6 \\ 5 & 4 & 7 & 6 \\ 2 & 4 & 5 & 9 \end{bmatrix}$$

- 2D transformation goes as:

$$Y = AXA^T = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 4 & 7 & 6 & 9 \\ 6 & 8 & 3 & 6 \\ 5 & 4 & 7 & 6 \\ 2 & 4 & 5 & 9 \end{bmatrix}\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 22.75 & -2.75 & 0.75 & -3.75 \\ 1.75 & 3.25 & -0.25 & -1.75 \\ 0.25 & -3.25 & 0.25 & -2.25 \\ 1.25 & -1.25 & 0.75 & 1.75 \end{bmatrix}$$

- Notice the energy compaction.

Two Dimensional Transform

- Because transformation matrix is orthonormal, $A^T = A^{-1}$
- So, we have
  - Forward transform

$$Y = AXA^{-1} = AXA^T$$

  - Backward transform

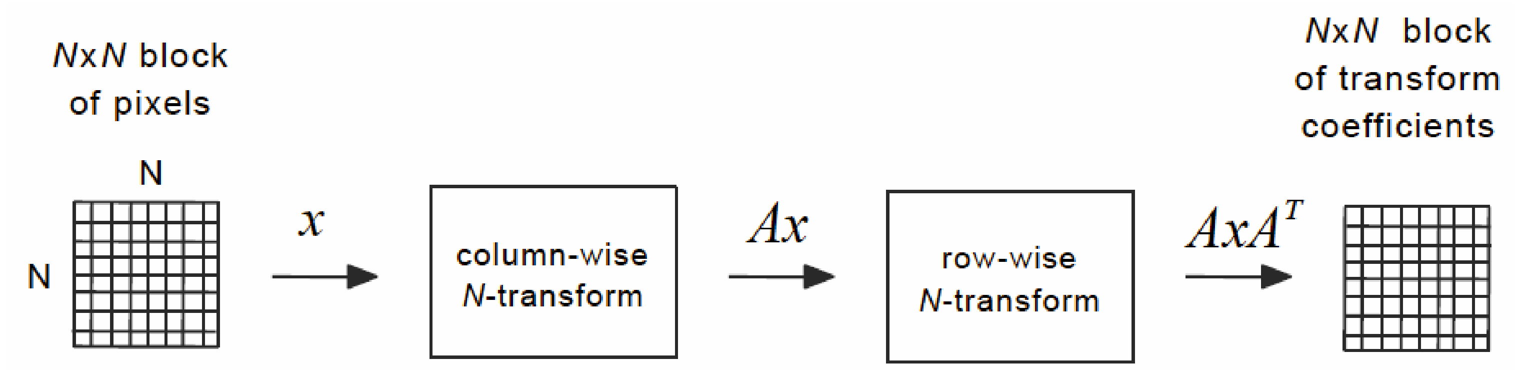
$$X = A^{-1}YA = A^TYA$$

Linear Separable transform

- Two dimensional transform is simplified as two iterations of one-dimensional transform.

$$Y_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{k,i} X_{i,j} a_{l,j}$$

- Column-wise transform and row-wise transform.



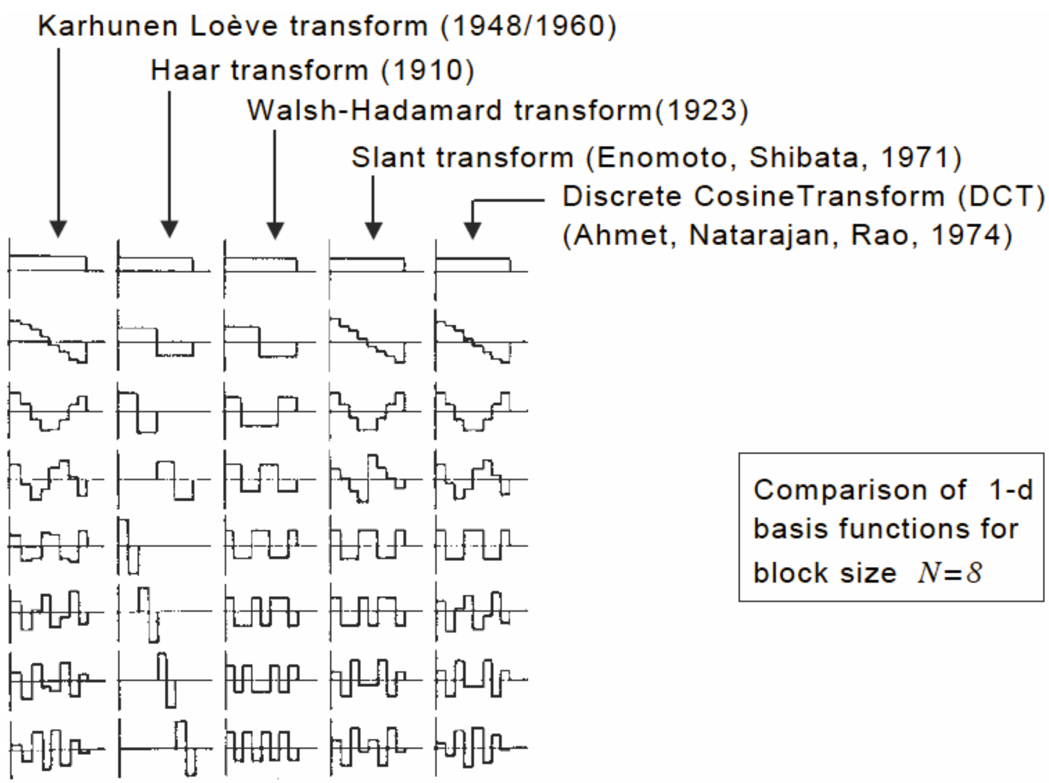
Transform and Filtering

- Consider the orthonormal transform. $$A = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$
- If A is used to transform a vector of 2 identical elements x = $[x,x]^T$, the transformed sequence will $(\sqrt{2}x, 0)^T$ indicating the "low frequency" or the "average" value is $\sqrt{2}x$ and the "high frequency" component is 0 because the signal value do not vary.
- If x = $[3,1]^T$ or $[3,-1]^T$, the output sequence will be $(2\sqrt{2}, \sqrt{2})^T$ and $(\sqrt{2}, 2\sqrt{2})^T$ respectively. Now, the high frequency component has positive value and it is bigger for $[3,-1]^T$,indicating a much large variation. Thus, the two coefficients behave like output of a "low-pass" and a "high-pass" filters, respectively.

Transform and Functional Approximation

- Transform is a kind of function approximation.
- Image is a data set. Any data set is a function.
- Transform is to approximate the image function by a combination of simpler, well defined "waveforms" (basis functions).
- Not all basis sets are equal in terms of compression.
- DCT and Wavelets are computationally easier than Fourier.

Comparison of various transforms



Comparison of 1-d basis functions for block size $N=8$

- The KLT is optimal in the sense of decorrelating and energy-packing.
- Walsh-Hadamard Transform is especially easy for implementation.
  - basis functions are either -1 or +1, only add/sub is necessary.

Two-Dimensional Basis Matrix

- The outer product of two vectors V_1 and V_2 is defined as $V_1^T V_2$. For example,

$$V_1^T V_2 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} d & e & f \end{bmatrix} = \begin{bmatrix} ad & ae & af \\ bd & be & bf \\ cd & ce & cf \end{bmatrix}$$

- For a matrix A of size n*n, the outer product of i^{th} row and j^{th} column is defined as

$$\alpha_{ij} = \begin{bmatrix} a_{i,0} \\ a_{i,1} \\ \vdots \\ a_{i,n-1} \end{bmatrix} \begin{bmatrix} a_{j,0} & a_{j,1} & \cdots & a_{j,n-1} \end{bmatrix} = \begin{bmatrix} a_{i,0}a_{j,0} & a_{i,0}a_{j,1} & \cdots & a_{i,0}a_{j,n-1} \\ a_{i,1}a_{j,0} & a_{i,1}a_{j,1} & \cdots & a_{i,1}a_{j,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,n-1}a_{j,0} & a_{i,n-1}a_{j,1} & \cdots & a_{i,n-1}a_{j,n-1} \end{bmatrix}$$

Outer Product

- For example, if $A = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- We have:

$$\alpha_{00} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\alpha_{01} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$\alpha_{10} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$\alpha_{11} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Outer Product (2)

- We have shown earlier that $X = A^T Y A$
- Consider X to be a 2*2 matrix:

$$\begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} y_{00}+y_{10} & y_{01}+y_{11} \\ y_{00}-y_{10} & y_{01}-y_{11} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} y_{00}+y_{10}+y_{01}+y_{11} & y_{00}+y_{10}-y_{01}-y_{11} \\ y_{00}-y_{10}+y_{01}-y_{11} & y_{00}-y_{10}-y_{01}+y_{11} \end{bmatrix}$$

$$= \frac{1}{2}\left\{ y_{00}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + y_{10}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} + y_{01}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} + y_{11}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right\}$$

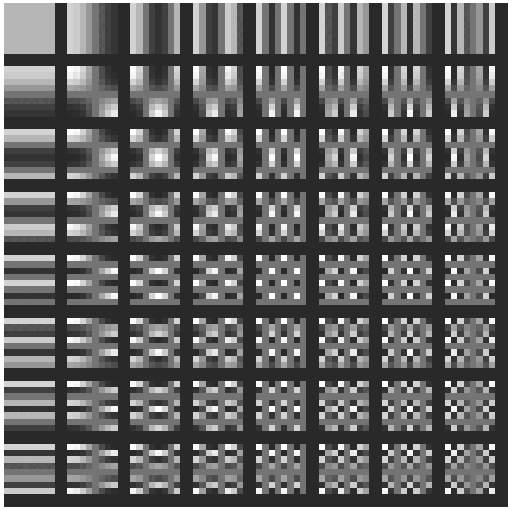
$$= y_{00}\alpha_{00} + y_{01}\alpha_{01} + y_{10}\alpha_{10} + y_{11}\alpha_{11}$$

Basis Matrix

- The quantities $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}$ are called the **basis matrices** in 2-D space.
- In general, the outer products of a *n*n* orhtonormal matrix form a basis matrix set in 2 dimension. The quantity α_{00} is called the DC coefficient (note all the elements for the DC coeeficient are 1, indicating an average operation), and other coefficients have alternating values and are called AC coefficients.

Fast Cosine Transform

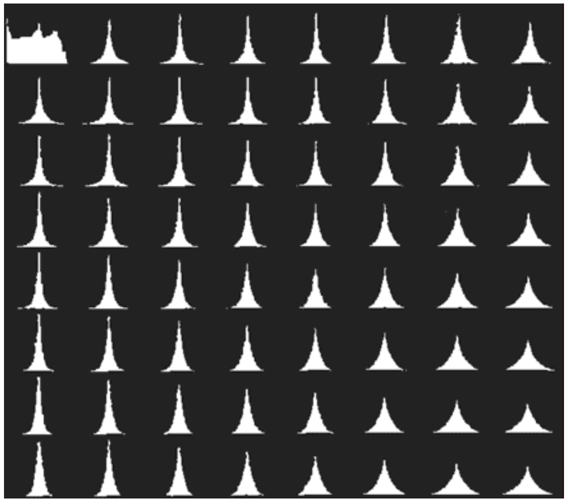
- 2D 8X8 basis functions of the DCT:
- The horizontal frequency of the basis functions increases from left to right and the vertical frequency of the basis functions increases from top to bottom.



.

Amplitude distribution of the DCT coefficients

- Histograms for 8x8 DCT coefficient amplitudes measured for natural images
  - DC coefficient is typically uniformly distributed.
  - The distribution of the AC coefficients have a Laplacian distribution with zero-mean.



Discrete Cosine Transform (DCT)

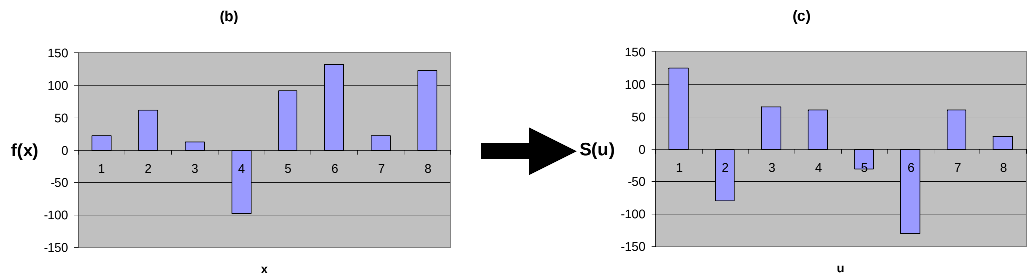
- Conventional image data have reasonably high inter-element correlation.
- DCT avoids the generation of the spurious spectral components which is a problem with DFT and has a fast implementation which avoids complex algebra.

One-dimensional DCT

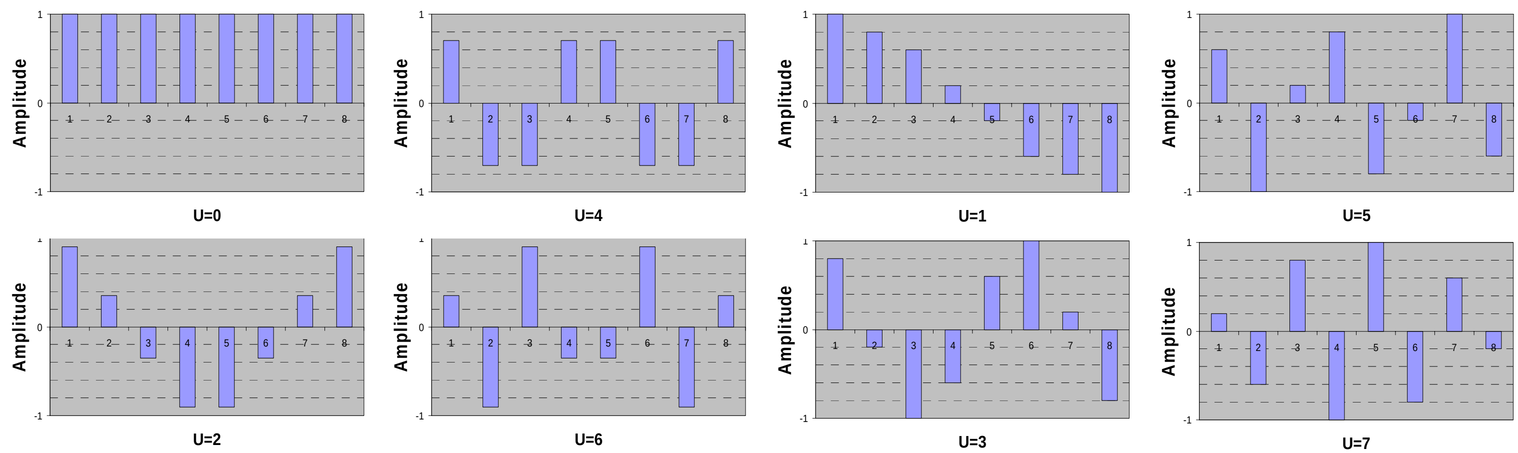
- The basis idea is to decompose the image into a set of “waveforms”, each with a particular “special” frequency.
- To human eyes, high spatial frequencies are imperceptible and a good approximation of the image can be created by keeping only the lower frequencies.
- Consider the one-dimensional case first. The 8 arbitrary grayscale values (with range 0 to 255, shown in the next slide) are level shifted by 128 (as is done by JPEG).

One-dimensional DCT

An example of 1-D DCT decomposition



Before DCT (image data)　　After DCT (coefficients



The 8 basis functions for 1-D DCT

One-dimensional DCT

- The waveforms cam be denoted as $w(f) = \cos(f\theta)$, with $0 \le \theta \le \pi$ with frequencies *f* = 0, 1, …, 7. Each wave is sampled at 8 points $\theta = \frac{\pi}{16}, \frac{3\pi}{16}, \frac{5\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}, \frac{15\pi}{16}$ to form a basis vector.
- The eight basis vector constitutes a matrix A:

1	1	1	1	1	1	1	1
0.981	0.831	0.556	0.195	-0.195	-0.556	-0.831	-0.981
0.924	0.383	-0.383	-0.924	-0.924	-0.383	0.383	0.924
0.831	-0.195	-0.981	-0.556	0.556	0.981	0.195	-0.831
0.707	-0.707	-0.707	0.707	0.707	-0.707	-0.707	0.707
0.556	-0.981	0.195	0.831	-0.831	-0.195	0.981	-0.556
0.383	-0.924	0.924	-0.383	-0.383	0.924	-0.924	0.383
-0.195	-0.556	0.831	-0.981	0.981	-0.831	0.556	-0.195

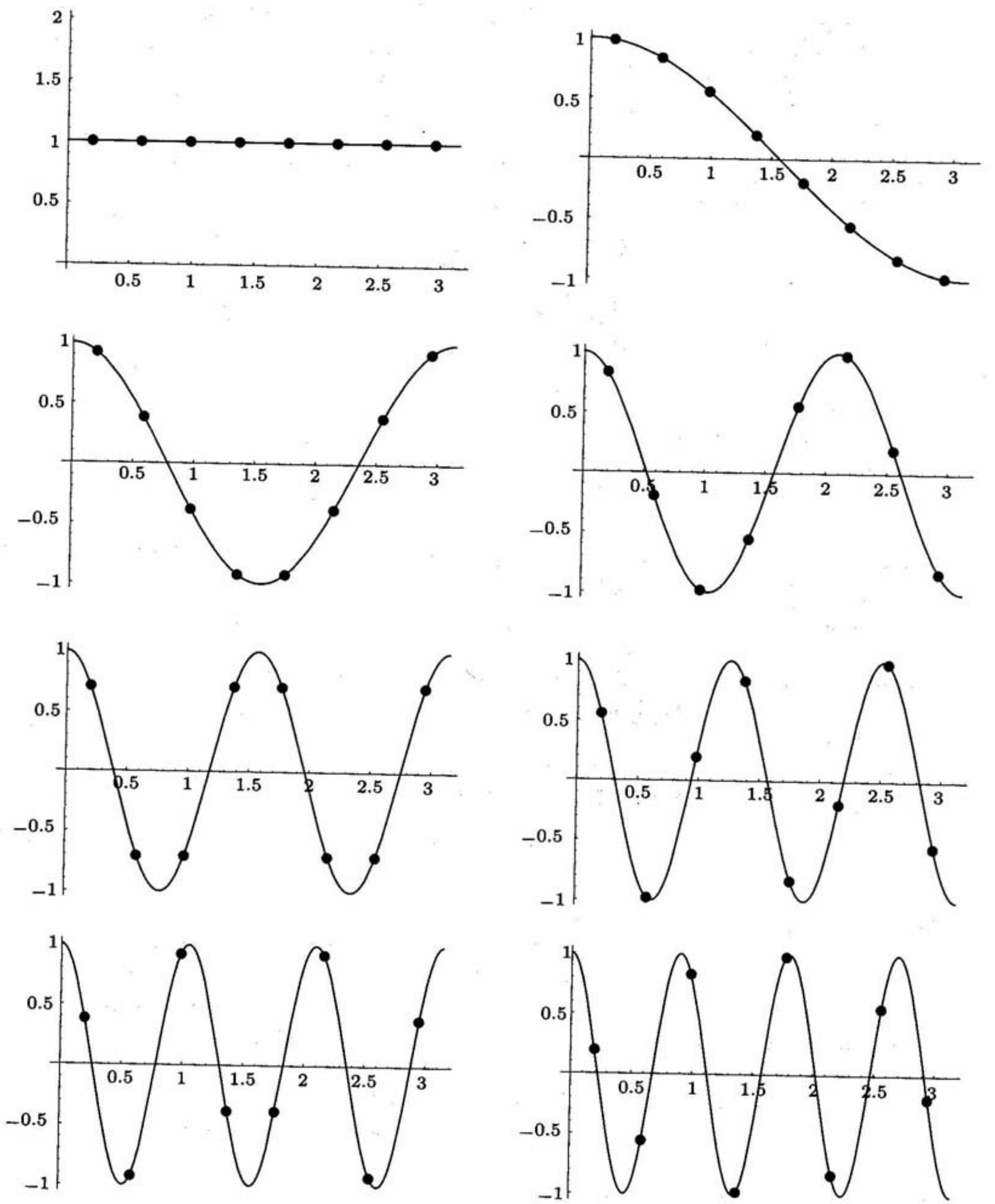


Figure 4.17: Calculating A One-Dimensional DCT.

One-dimensional DCT

- The output of the DCT transform is:

$$S(u) = A\left[I(x)\right]^T$$

  where A is the 8*8 transformation matrix defined in the previous slide, and I(x) is the input signal.
- S(u) are called the coefficients for the DCT transform for input signal I(x).

One-dimensional FDCT and IDCT with N=8 Sample points

- The 1-D DCT in JPEG is defined as:
  - FDCT $$S(u) = \frac{C(u)}{2}\sum_{x=0}^{7} I(x)\cos[\frac{(2x+1)u\pi}{16}] \quad \text{for } u = 0,1,..7$$
  - IDCT $$I(x) = \sum_{u=0}^{7}\frac{C(u)}{2} S(u)\cos[\frac{(2x+1)u\pi}{16}] \quad \text{for } x = 0,1,..,7$$
  - Where (*u* is frequency)
    - I(x) is the 1-D sample
    - S(u) is the 1-D DCT coefficient
    - And $$C(u) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0 \end{cases}$$

One-dimensional FDCT and IDCT with N Sample points

- The 1-D DCT in JPEG is defined as:
  - FDCT

$$S(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} I(x) \cos[\frac{(2x+1)u\pi}{2N}] \text{ for } u = 1,2,..\text{N-1}$$

  - IDCT

$$I(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{n-1} C(u) S(u) \cos[\frac{(2x+1)u\pi}{2N}] \text{ for } x = 0,1,..\text{N-1}$$

  - Where (*u* is frequency)
    - I(x) is the 1-D sample
    - S(u) is the 1-D DCT coefficient and

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0 \end{cases}$$

One-dimensional FDCT and IDCT

- As an example, let I(x)=[12,10,8,10,12,10,8,11]

$$S(u) = A[I(x)]^T = [28.6375, 0.5712, 0.4619, 1.757, 3.182, -1.729, 0.191, -0.309].$$

- If we now apply IDCT, we will get back I(x). We can quantize the coefficient S(u) and still obtain a very good approximation of I(x).
- For example,

$$IDCT(28.6, 0.6, 0.5, 1.8, 3.2, -1.8, 0.2, -0.3)$$

$$= (12.0254, 10.0233, 7.96054, 9.93097, 12.0164, 9.9932, 7.99354, 10.9989)$$

- While

$$IDCT\ (28, 0, 0, 2, 3, -2, 0, 0)$$

$$= (11.236, 9.6244, 7.6628, 9.573, 12.347, 10.014, 8.053, 10.684)$$

Two-dimensional FDCT and IDCT for 8X8 block

- The 2-D DCT in JPEG is defined as:
  - FDCT

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} I(y,x)\cos[\frac{(2y+1)v\pi}{16}]\cos[\frac{(2x+1)u\pi}{16}]$$

  - IDCT

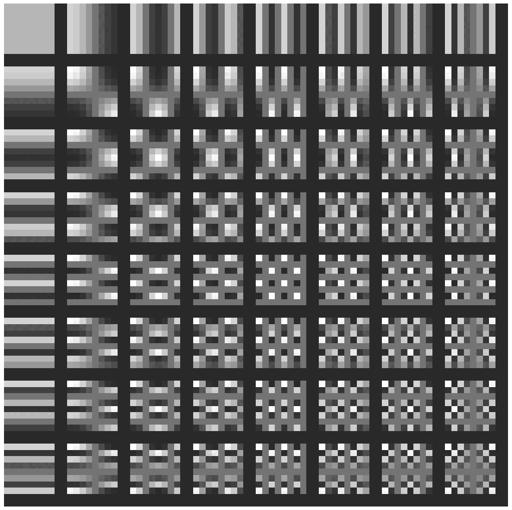
$$I(y,x) = \sum_{v=0}^{7}\frac{C(v)}{2}\sum_{u=0}^{7}\frac{C(u)}{2} S(v,u)\cos[\frac{(2y+1)v\pi}{16}]\cos[\frac{(2x+1)u\pi}{16}]$$

  - Where
    - I(y,x) is the 2-D sample
    - S(v,u) is the 2-D DCT coefficient
    - And

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0 \end{cases} \qquad C(v) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } v = 0 \\ 1 & \text{for } v > 0 \end{cases}$$

Fast Cosine Transform

- 2D 8X8 basis functions of the DCT:
- The horizontal frequency of the basis functions increases from left to right and the vertical frequency of the basis functions increases from top to bottom.



.

Two-dimensional FDCT for NXM block

FDCT

$$S(v,u) = \frac{2}{\sqrt{NM}} \frac{C(v)}{2} \frac{C(u)}{2} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(y,x) \cos[\frac{(2y+1)v\pi}{2M}] \cos[\frac{(2x+1)u\pi}{2N}]$$

for $u = 0,1,\ldots, N\text{-}1$ and $v = 0,1,\ldots, M-1$

- Where I(y,x) is the 2-D sample, S(v,u) is the 2-D DCT coefficient and

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0 \end{cases} \qquad C(v) = \begin{cases} \frac{\sqrt{2}}{2} & \text{for } v = 0 \\ 1 & \text{for } v > 0 \end{cases}$$

Separable Function

- The 2-D FDCT is a separable function because we can express the formula as

$$S(v,u)=\sqrt{\frac{2}{M}}C(v)\sum_{y=0}^{M-1}\left\{\sqrt{\frac{2}{N}}C(u)\sum_{x=0}^{N-1} I(y,x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\right\}\cos\left[\frac{(2y+1)\pi u}{2M}\right]$$

$$\text{for } u=0,1,\ldots,N-1 \text{ and } v=0,1,\ldots,M-1$$

This means that we can compute a 2-D FDCT by first computing a row-wise 1-D FDCT and taking the output and perform on it a column-wise 1-D FDCT transform. This is possible, as we explained earlier, because the basis matrices are orthonormal.

Two-dimensional IDCT for NXM block

$$I(y,x)=\frac{2}{\sqrt{NM}}\sum_{v=0}^{M-1}\sum_{u=0}^{N-1}C(u)\,C(v)\,S(v,u)\cos[\frac{(2y+1)v\pi}{2M}]\cos[\frac{(2x+1)u\pi}{2N}]$$

$$\text{for } x=0,1,\ldots,\ N-1 \text{ and } y=0,1,\ldots,\ M-1.$$

This function is again separable and the computation can be done in two steps: a row-wise 1-D IDCT followed by a column-wise 1-D IDCT. A straightforward algorithm to compute both FDCT and IDCT will need (for a block of NXN pixels) an order of *O*(N^3) *multiplication/ addition operations. After the ground-breaking discovery of O*(*N logN*) algorithm for Fast Fourier Transform, several researchers proposed efficient algorithms for DCT computation.

JPEG Introduction - The background

- JPEG stands for Joint Photographic Expert Group
- A standard image compression method is needed to enable interoperability of equipment from different manufacturer
- It is the first international digital image compression standard for continuous-tone images (grayscale or color)
- The history of JPEG – the selection process

JPEG Introduction – what's the objective?

- "very good" or "excellent" compression rate, reconstructed image quality, transmission rate
- be applicable to practically any kind of continuous-tone digital source image
- good complexity
- have the following modes of operations:
  - sequential encoding
  - progressive encoding
  - lossless encoding
  - hierarchical encoding

JPEG Architecture Standard

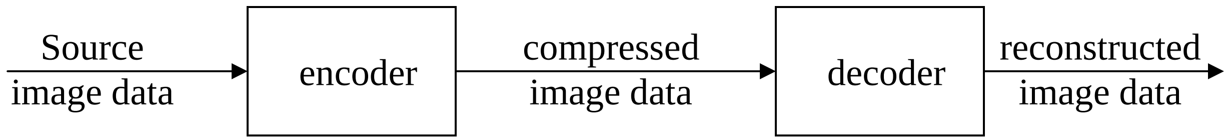
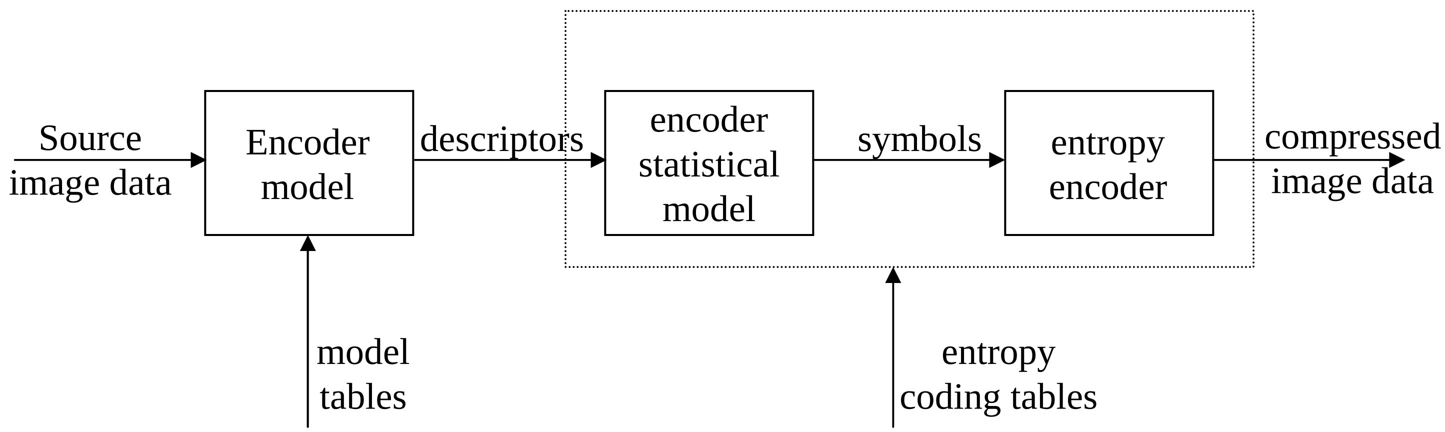


Image compression system



The basic parts of an JPEG encoder

JPEG Overview (cont.)

JPEG has the following Operation Modes:

- Sequential DCT-based mode
- Progressive DCT-based mode
- Sequential lossless mode
- Hierarchical mode

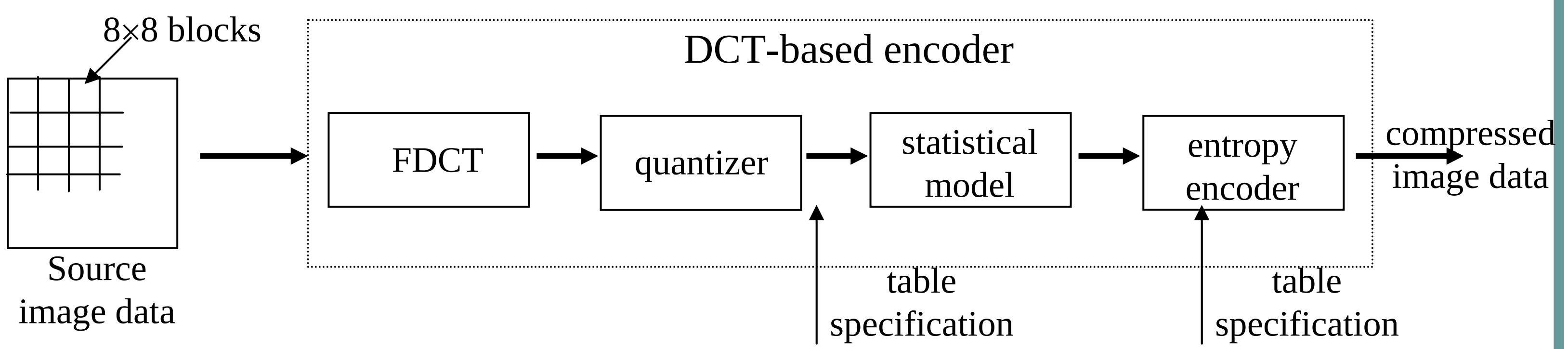
JPEG entropy coding supports:

- Huffman encoding
- Arithmetic encoding

JPEG Baseline System

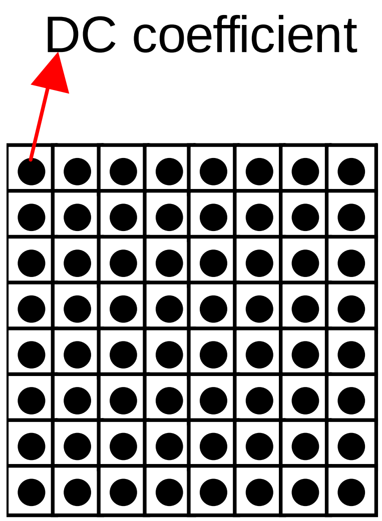
JPEG Baseline system is composed of:

- Sequential DCT-based mode
- Huffman coding



The basic architecture of JPEG Baseline system

The Baseline System



- The DCT coefficient values can be regarded as the relative amounts of the 2-D spatial frequencies contained in the 8×8 block
- the upper-left corner coefficient is called the **DC coefficient**, which is a measure of the average of the energy of the block
- Other coefficients are called **AC coefficients**, coefficients correspond to high frequencies tend to be zero or near zero for most natural images

Quantization Tables in DCT

- Human eyes are less sensitive to high frequencies.
- We use different quantization value for different frequencies.
  - Higher frequency, bigger quantization value.
  - Lower frequency, smaller quantization value.
- Each DCT coefficient corresponds to a certain frequency.
- High-level HVS is much more sensitive to the variations in the achromatic channel than in the chromatic channels.

The Baseline System – Quantization

- Why quantization? .
  - **to achieve further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality**
- Generally, the "high frequency coefficients" has larger quantization values
- Quantization makes most coefficients to be zero, it makes the compression system efficient, but it's the main source that make the system "lossy" and introduces distortion in the reconstructed image.
- The quantization step-size parameters for JPEG are given by two Quantization Matrices, each element of the matrix being an integer of size between 1 to 255. The DCT coeeficients are divided by the corresponding quantization parameters and rounded to nearest integer.

$$F'(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

F(u,v): original DCT coefficient
F'(u,v): DCT coefficient after quantization
Q(u,v): quantization value

There are two quantization tables: one for the luminance component and the other for the Chrominance component. JPEG standard does not prescribe the values of the table; it is left upto the user, but they recommend the following two tables under normal circumstances. These tables have been created by analysing data of human perception and a lot of trial and error.

Quantization Tables in DCT

- So, we have two quantization tables.
  - Measured for an "average" person.
  - Higher frequency, bigger quantization value.
  - Lower frequency, smaller quantization value.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Luminance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

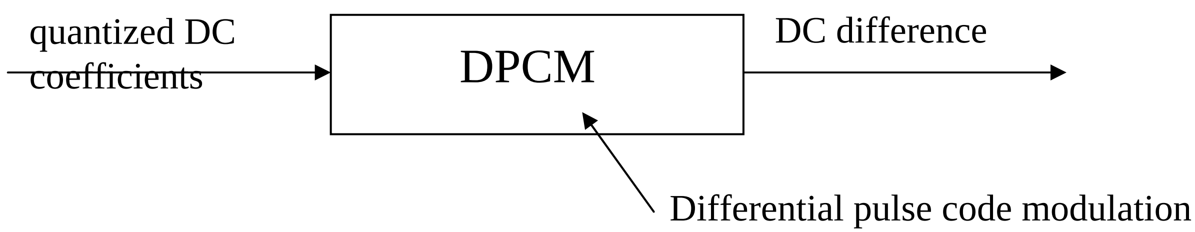
Chrominance quantization table

Two-dimensional DCT

- The image samples are shifted from unsigned integer with range [$0, 2^{N-1}$] to signed integers with range [$-2^{N-1}, 2^{N-1}-1$]. Thus samples in the range 0-255 are converted in the range -128 to 127 and those in the range 0 to 4095 are converted in the range -2048 to 2047. This zero-shift done for JPEG to reduce the internal precision requirements in the DCT calculations.
- How to interpret the DCT coefficients?
  - The DCT coefficient values can be regarded as the relative amounts of the 2-D spatial frequencies contained in the 8×8 block.
  - F(0,0) is called DC coefficient, which is a measure of the average of the energy of the block.
  - Other coefficients are called AC coefficients, coefficients correspond to high frequencies tend to be zero or near zero for most images.
- The energy is concentrated in the upper-left corner.

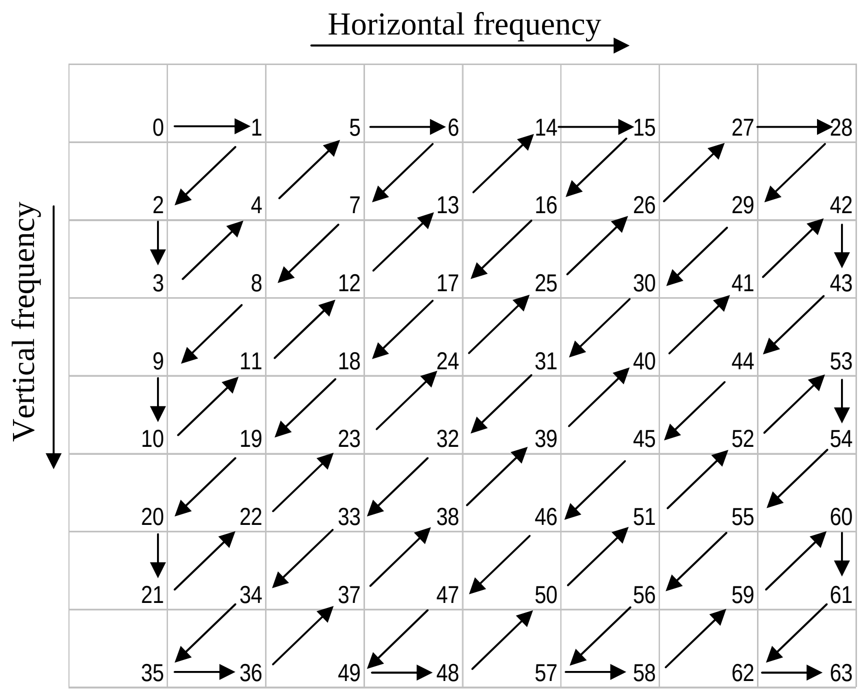
Baseline System - DC coefficient coding

- After quantization step, the DC coefficient are encoded separately by using differential encoding. The DC difference sequence is constituted by concatenating the DC coefficient of the first block followed by the difference values of DC coefficients of the succeeding blocks. The average values of the succeeding blocks are usually correlated.



Baseline System - AC coefficient coding

- AC coefficients are arranged into a zig-zag sequence. This is because most of the significant coefficients are located in the upper left corner of the matrix. The high frequency components are mostly 0's and can be efficiently coded by run-length encoding.



An Example (Ref:T. Acharya, P. Tsai,"JPEG 2000 Standard for Image Compression), Wiley-Iterscience, 2005

The 8X8 Image Data Block

110	110	118	118	121	126	131	131
108	111	125	122	120	125	134	135
106	119	129	127	125	127	138	144
110	126	130	133	133	131	141	148
115	116	119	120	122	125	137	139
115	106	99	110	107	116	130	127
110	91	82	101	99	104	120	118
103	76	70	95	92	91	107	106

The 8 × 8 Data Block After Level Shifting

−18	−18	−10	−10	−7	−2	3	3
−20	−17	−3	−6	−8	−3	6	7
−22	−9	1	−1	−3	−1	10	16
−18	−2	2	5	5	3	13	20
−13	−12	−9	−8	−6	−3	9	11
−13	−22	−29	−18	−21	−12	2	−1
−18	−37	−46	−27	29	−24	−8	−10
−25	−52	−58	−33	−36	−37	−21	−22

DCT Coefficients of the Above 8 × 8 Block

−89.00	−63.47	18.21	−6.85	7.50	13.45	−7.00	0.13
74.14	−2.90	−19.93	−21.04	−17.88	−10.81	8.29	5.26
−63.65	3.10	5.08	14.82	10.12	9.33	1.31	−0.62
3.73	2.85	6.67	8.99	−3.38	1.54	1.04	−0.62
2.50	0.57	−4.46	0.52	3.00	−2.89	−0.32	1.33
7.52	−1.80	−0.63	−0.10	0.41	−3.21	−2.74	−2.07
−3.40	0.43	0.81	0.28	−0.40	−0.19	−0.58	−1.09
−2.26	−0.88	1.73	0.23	−0.21	−0.12	1.23	1.61

Results of DCT Coefficients Quantized by Luminance Quantization Matrix

−6	−6	2	0	0	0	0	0
6	0	−1	−1	−1	0	0	0
−5	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

After the DC coefficient is differentially encoded, the AC coefficients are ordered in the zig-zag sequence and the sequence is subsequently broken into

Baseline System - Statistical modeling

- Statistical modeling translate the inputs to a sequence of “symbols” for Huffman coding to use
- Statistical modeling on DC coefficients:
  - Category: different size (SSSS)
  - Magnitude: amplitude of difference (additional bits)

Coding the DC coeeficients

SSSS	DPCM difference	Additional bits (binary)
0	0	–
1	–1,1	0,1
2	–3,–2,2,3	00,01,10,11
3	–7,...,–4,4,...,7	000,...,011,100,...,111
4	–15,...,–8,8,...,15	0000,...,0111,1000,...,1111
⋮	⋮	⋮
16	32768	–

tables.jpg

The prediction errors are classified into 16 categories in modulo 2^{16}. Category *C* contains the range of integers $[-(2^c-1), +(2^c-1)]$ excluding the numbers in the range $[-(2^{c-1}-1), +(2^{c-1}-1)]$ which falls in the previous category. The category number is encoded using either a fixed code (*SSSS*), unary code (0, 10, 110, …) or a variable length Huffman code.

Coding the DC/AC coefficients

If the magnitude of the error is positive, it is encoded using 'category' number of bits with a leading '1'. If it is negative, I's complement of the positive number is used. Thus, if error is -6 and if the Huffman code for category 3 is 1110, it will get a code (1110001). The JPEG Standard recommends Huffman two codes for the category numbers: one for Luminance DC and the other for Chrominance DC (See Salomon, p.289; it is also described in Annex K (Table k.3 and K.4 of the Standard)

- Statistical modeling on AC coefficients:
  - Run-length: RUN-SIZE=16*RRRR+SSSS
  - Amplitude: amplitude of difference (additional bits)

The AC coefficients contains just a few nonzero numbers, with runs of zeros followed by a long run of trailing zeros. For each nonzero number, two numbers are generated: 1)16*RRRR which gives the number of consecutive zeros preceding the nonzero coefficient being encoded, 2)SSSS gives the category corresponding to number of bits required to encode the AC coefficient using variable length Huffman code. This is specified in the form of a table (see next slide). The second attribute of the AC coefficient is its amplitude which is encoded exactly the same way as the DC values. Note the code (0,0) is reserved for EOB (signifying that the remaining AC coefficients are just a single trailing run of 0's). The code (15,0) is reserved for a maximum run of 16 zeros. If it is more than 16, it is broken down into runs of 16 plus possibly a single run less than 16. JPEG recommends two tables for Huffman codes for AC coefficients, Tables K5 and K6 for luminance and Chrominance values.

Encoding the AC Coefficients

RRRR \ SSSS	0	1	2	3	4	5	6	7	8	9	10	11 *	12 *	13 *	14 *	15 *
0	EOB	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1	N/A	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2	N/A	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
3	N/A	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
4	N/A	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
5	N/A	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
6	N/A	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
7	N/A	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
8	N/A	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
9	N/A	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
10	N/A	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
11	N/A	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
12	N/A	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
13	N/A	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
14	N/A	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
15	ZRL	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

* Not used in sequential mode including baseline with 8 bit input
N/A Not applicable for sequential mode

tables.jpg

Huffman AC statistical model
run-length/amplitude combinations

SSSS	AC coefficients	Precision	
1	−1,1	8	12
2	−3,−2,2,3	8	12
3	−7,...,−4,4,...,7	8	12
4	−15,...,−8,8,...,15	8	12
5	−31,...,−16,16,...,31	8	12
6	−63,...,−32,32,...,63	8	12
7	−127,...,−64,64,...,127	8	12
8	−255,...,−128,128,...,255	8	12
9	−511,...,−256,256,...,511	8	12
10	−1023,...,−512,512,...,1023	8	12
11	−2047,...,−1024,1024,...,2047	8	12
12	−4095,...,−2048,2048,...,4095	8*	12
13	−8191,...,−4096,4096,...,8191		12
14	−16383,...,−8192,8192,...,16383		12
15	−32767,...,−16384,16384,...,32767		12*

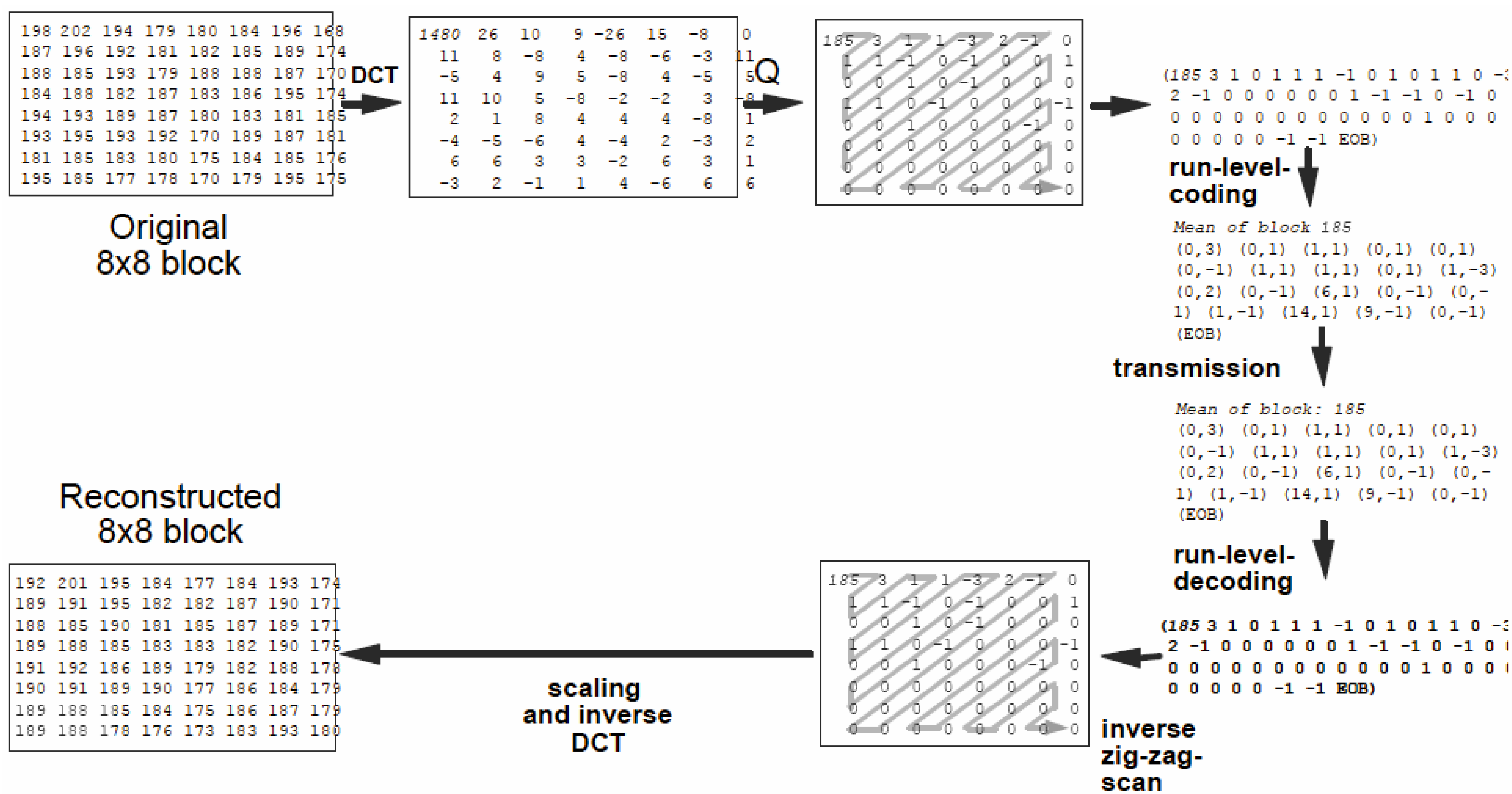
tables.jpg

Huffman coding of AC coefficients

Example (Contd.)

- Let us finish our example. The DC coeeficient is -6 has a code o 1110001. The AC coeeficients will be encoded as:
- (0,3)(-6), (0,3)(6), (0,3)(-5), (1,2)(2), (1,1)(-1), (5,1)(-1),(2,1)(-1), (0,1)(1),(0,0). The Table K.5 gives the codes
- 1010, 00, 100, 1100, 11011, 11100 and 11110101 for the pairs of symbols (0,0), (0,1), (0,3), (1,1), (1,2), (2,1) and (5,1), respectively. The variable length code for AC coefficients 1, -1, 2, -5, 6 and -6 are 1, 0, 10, 010,110 and 001, respectively. Thus JPEG will give a compressed representation of the example image using only 55 bits as:
- 1110001 100001 100110 100010 1101110 11000 11110100 111000 001 1010
- The uncompressed representation would have required 512 bits!

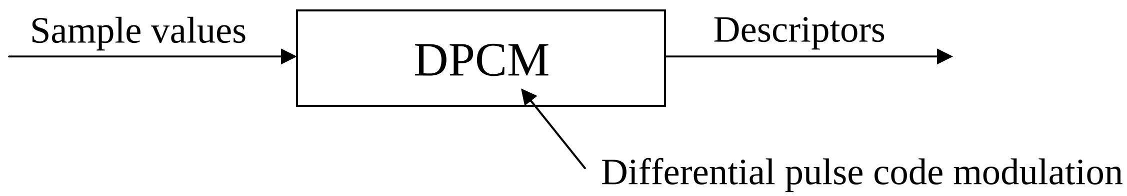
Example to illustrate encoding/decoding



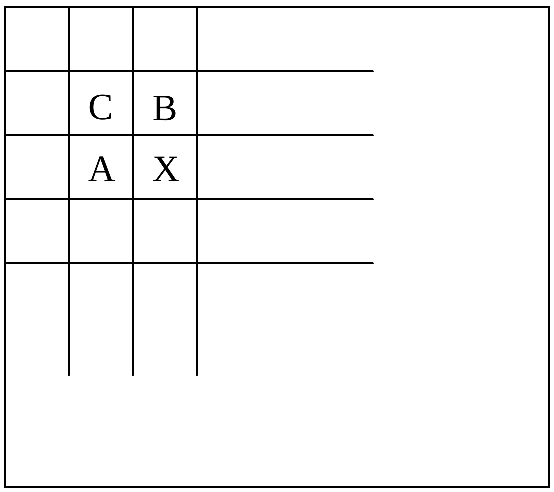
JPEG Progressive Model

- Why progressive model?
  - Quick transmission
  - Image built up in a coarse-to-fine passes
- First stage: encode a rough but recognizable version of the image
- Later stage(s): the image refined by successive scans till get the final image
- Two ways to do this:
  - Spectral selection – send DC, AC coefficients separately
  - Successive approximation – send the most significant bits first and then the least significant bits

JPEG Lossless Model



Predictors for lossless coding	
selection value	prediction strategy
0	no prediction
1	A
2	B
3	C
4	A+B-C
5	A+(B-C)/2
6	B+(A-C)/2
7	(A+B)/2



JPEG Hierarchical Model

- Hierarchical model is an alternative of progressive model (pyramid)
- Steps:
  - filter and down-sample the original images by the desired number of multiplies of 2 in each dimension
  - Encode the reduced-size image using one of the above coding model
  - Use the up-sampled image as a prediction of the origin at this resolution, encode the difference
  - Repeat till the full resolution image has been encode

The Effect of Segmentation

- The image samples are grouped into 8×8 blocks. 2-D DCT is applied on each 8×8 blocks.

- Because of blocking, the spatial frequencies in the image and the spatial frequencies of the cosine basis functions are not precisely equivalent. According to Fourier's theorem, all the harmonics of the fundamental frequencies must be present in the basis functions to be precise. Nonetheless, the relationship between the DCT frequency and the spatial frequency is a close approximation if we take into account the sensitivity of human eye for detecting contrast in frequency.
- The segmentation also introduces what is called the "blocking artifacts". This becomes very pronounced if the DC coefficients from block to block vary considerably. These artifacts appear as edges in the image, and abrupt edges imply high frequency. The effect can be minimized if the non-zero AC coefficients are kept.

Some other transforms

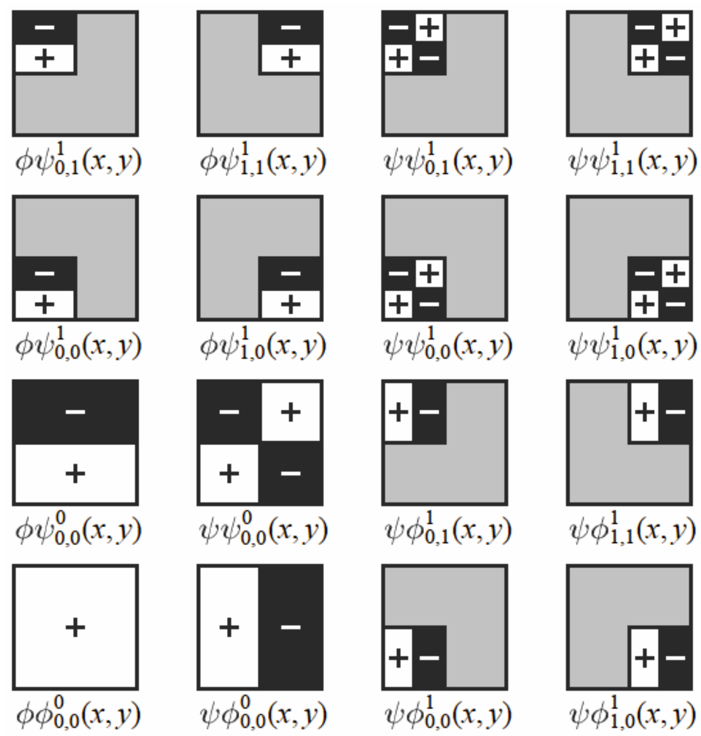
- Discrete Fourier Transform (DFT)
- Haar Transform
- Karhunen Loève Transform (KLT)
- Walsh-Hadamard Transform (WHT)

Discrete Fourier Transform (DFT)

- Well-known for its connection to spectral analysis and filtering.
- Extensive study done on its fast implementation ($O(N\log_2 N)$ for N-point DFT).
- Has the disadvantage of storage and manipulation of complex quantities and creation of spurious spectral components due to the assumed periodicity of image blocks.

Haar Transform

- Very fast transform.
- The easiest wavelet transform.
- Useful in edge detection, image coding, and image analysis problems.
- Energy Compaction is fair, not the best compression algorithms.



2D basis function of Haar transform

Karhunen Loève Transform (KLT)

- Karhunen Loève Transform (KLT) yields decorrelated transform coefficients.
- Basis functions are eigenvectors of the covariance matrix of the input signal.
- KLT achieves <u>optimum</u> energy concentration.
- Disadvantages:
  - KLT dependent on signal statistics
  - KLT not separable for image blocks
  - Transform matrix cannot be factored into sparse matrices

Walsh-Hadamard Transform (WHT)

- Although far from optimum in an energy packing sense for typical imagery, its simple implementation (basis functions are either -1 or +1) has made it widely popular.

Transformation matrix:

$$A = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Walsh-Hadamard Transform (WHT)

- Walsh-Hadamard transform requires adds and subtracts
- Use of high speed signal processing has reduced its use due to improvements with DCT.

Transform Coding: Summary

- Purpose of transform
  - de-correlation
  - energy concentration
- KLT is optimum, but signal dependent and, hence, without a fast algorithm
- DCT reduces blocking artifacts appeared in DFT
- Threshold coding + zig-zag-scan + 8x8 block size is widely used today
  - JPEG, MPEG, ITU-T H.263.
- Fast algorithm for scaled 8-DCT
  - 5 multiplications, 29 additions
- Audio Coding
  - MP3 = MPEG 1- Layer 3 uses DCT