CSE 490 GZ
Introduction to Data Compression
Winter 2004

Transform Coding

JPEG

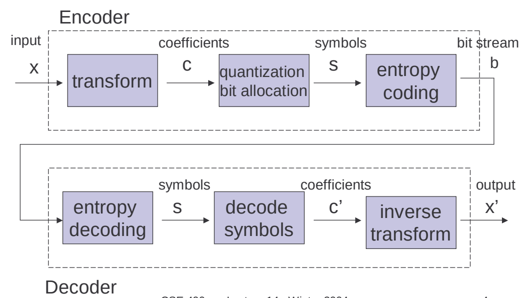
Idea of Transform Coding

- Transform the input pixels $x_0,x_2,\ldots,x_{N-1}$ into coefficients $c_0,c_1,\ldots,c_{N-1}$ (real values)
  - The coefficients are 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

Decoding

- Entropy decode the quantized symbols
- Compute approximate coefficients $c'_0,c'_1,\ldots,c'_{N-1}$from the symbols.
- Inverse transform $c'_0,c'_1,\ldots,c'_{N-1}$ to $x'_0,x'_1,\ldots,x'_{N-1}$ which is a good approximation of the original $x_0,x_2,\ldots,x_{N-1}$.

Block Diagram of Transform Coding



Mathematical Properties of Transforms

- Linear Transformation - Defined by a real nxn matrix A = (a_{ij})

$$\begin{bmatrix} a_{00} & \cdots & a_{0,N-1} \\ \vdots & & \vdots \\ a_{N-1,0} & \cdots & a_{N-1,N-1} \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} c_0 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

- Orthonormality $A^{-1} = A^T$ (transpose)

Why Coefficients

$$A^T c = x$$

$$\begin{bmatrix} a_{00} & \cdots & a_{N-1,0} \\ \vdots & & \vdots \\ a_{0,N-1} & \cdots & a_{N-1,N-1} \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{N-1} \end{bmatrix} = \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

$$\begin{bmatrix} a_{00} \\ \vdots \\ a_{0,N-1} \end{bmatrix} c_0 + \cdots + \begin{bmatrix} a_{N-1,0} \\ \vdots \\ a_{N-1,N-1} \end{bmatrix} c_{N-1} = \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

basis vectors

coefficients

Why Orthonomality

- The energy of the data equals the energy of the coefficients

$$\sum_{i=0}^{N-1} c_i^{\,2} = c^T c = (Ax)^T (Ax)$$

$$= (x^T A^T)(Ax) = x^T (A^T A)x = x^T x = \sum_{i=0}^{N-1} x_i^{\,2}$$

Compaction Example

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

$$A^2 = A \Rightarrow A^{-1} = A$$

$$A^T = A = A^{-1} \qquad \text{orthonormal}$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} b \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{2}b \\ 0 \end{bmatrix} \qquad \text{compaction}$$

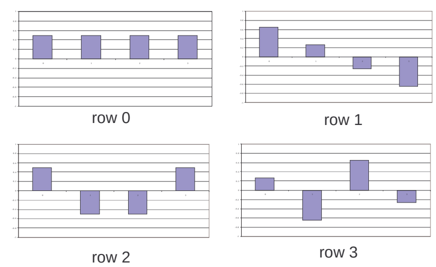
Discrete Cosine Transform

$$d_{ij} = \begin{cases} \sqrt{\dfrac{1}{N}} & i = 0 \\ \sqrt{\dfrac{2}{N}}\cos\dfrac{(2j+1)i\pi}{2N} & i > 0 \end{cases}$$

N = 4

$$D = \begin{bmatrix} .5 & .5 & .5 & .5 \\ .65328 & .270598 & -.270598 & -.65328 \\ .5 & -.5 & -.5 & .5 \\ .270598 & -.65328 & .65328 & -.270598 \end{bmatrix}$$

Basis Vectors

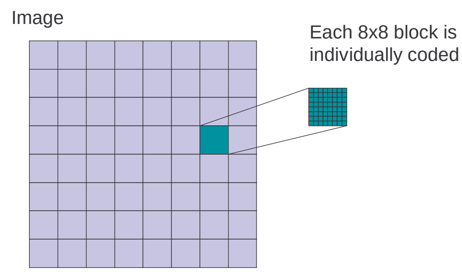


Decomposition in Terms of Basis Vectors

$$\begin{bmatrix} .5 \\ .5 \\ .5 \\ .5 \end{bmatrix} c_0 + \begin{bmatrix} .653281 \\ .270598 \\ -.270598 \\ -.653281 \end{bmatrix} c_1 + \begin{bmatrix} .5 \\ -.5 \\ -.5 \\ .5 \end{bmatrix} c_2 + \begin{bmatrix} .270598 \\ -.653281 \\ .653281 \\ -.270598 \end{bmatrix} c_3 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

DC coefficient — AC coefficients

Block Transform



2-Dimensional Block Transform

Block of pixels X

x_{00}	x_{01}	x_{02}	x_{03}
x_{10}	x_{11}	x_{12}	x_{13}
x_{20}	x_{21}	x_{22}	x_{23}
x_{30}	x_{31}	x_{32}	x_{33}

Transform

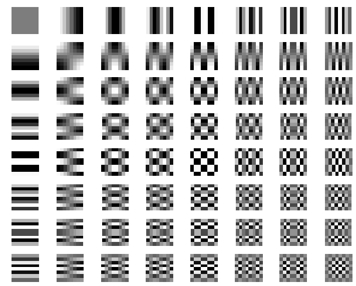
$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Transform rows $r_{ij} = \sum_{k=0}^{N-1} a_{kj} x_{ik}$

Transform columns $c_{ij} = \sum_{m=0}^{N-1} a_{im} r_{mj} = \sum_{m=0}^{N-1} a_{im} \sum_{k=0}^{N-1} a_{kj} x_{mk} = \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} a_{im} a_{kj} x_{mk}$

Summary $C = AXA^T$

8x8 DCT Basis



Importance of Coefficients

- The DC coefficient is the most important.
- The AC coefficients become less important as they are farther from the DC coefficient.
- Example Bit Allocation

8	7	5	3	2	1	0	0
7	5	3	2	1	0	0	0
5	3	2	1	0	0	0	0
3	2	1	0	0	0	0	0
2	1	0	0	0	0	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

compression
55 bits for 64 pixels = .86 bpp

Quantization Tables

- For a nxn block we construct a nxn matrix Q such that Q_{ij} indicates how many quantization levels to use for coefficient c_{ij}.
- Encode c_{ij} with the label

$$s_{ij} = \left\lfloor \frac{c_{ij}}{Q_{ij}} + 0.5 \right\rfloor$$

Larger Q_{ij} indicates fewer levels.

- Decode s_{ij} to

$$c'_{ij} = s_{ij} Q_{ij}$$

Example Quantization

- c = 54.2, Q = 24 $s = \left\lfloor \frac{54.2}{24} + 0.5 \right\rfloor = 2$

$c' = 2 \cdot 24 = 48$

- c = 54.2, Q = 12 $s = \left\lfloor \frac{54.2}{12} + 0.5 \right\rfloor = 5$

$c' = 5 \cdot 12 = 60$

- c= 54.2, Q = 6 $s = \left\lfloor \frac{54.2}{6} + 0.5 \right\rfloor = 9$

$c' = 9 \cdot 6 = 54$

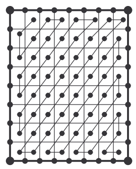
Example Quantization Table

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	33	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

Increase the bit rate = halve the table
Decrease the bit rate = double the table

Zig-Zag Coding

- DC label is coded separately.
- AC labels are usually coded in zig-zag order using a special entropy coding to take advantage the ordering of the bit allocation (quantization).



JPEG (1987)

- Let P = [p_{ij}], 0 < i,j < N be an image with 0 < p_{ij} < 256.
- Center the pixels around zero
  - $x_{ij} = p_{ij} - 128$
- Code 8x8 blocks of P using DCT
- Choose a quantization table.
  - The table depends on the desired quality and is built into JPEG
- Quantize the coefficients according to the quantization table.
  - The quantization symbols can be positive or negative.
- Transmit the labels (in a coded way) for each block.

Block Transmission

- DC coefficient
  - DC coefficients don't change much from block to neighboring block. Hence, their labels change even less.
  - Predictive coding using differences is used to code the DC label.
- AC coefficients
  - Do a zig-zag coding.

Example Block of Labels

5	2	0	0	0	0	0	0
-8	0	0	0	0	0	0	0
3	1	1	0	0	0	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

Coding order of AC labels
2 –8 3 0 0 0 0 1 1 0 0 1 0 0 .....

Coding Labels

- Categories of labels
  - 1 {0}
  - 2 {-1, 1}
  - 3 {-3,-2,2,3}
  - 4 {-7,-6,-5,-4,4,5 6 7}
- Label is indicated by two numbers C,B
- Examples

label	C,B
0	1
2	3, 2
-4	4, 3

Coding AC Label Sequence

- A symbol has three parts (Z,C,B)
  - Z for number of zeros preceding a label 0 ≤ Z ≤ 15
  - C for the category of the of the label
  - B for a C-1 bit number for the actual label
- End of Block symbol (EOB) means the rest of the block is zeros. EOB = (0,0,-)
- Example: 2 –8 3 0 0 0 0 1 1 0 0 1 0 0 .....

(0,3,2)(0,5,7)(0,3,3)(4,2,1)(0,2,1)(2,2,1)(0,0,-)

Coding AC Label Sequence

- Z,C have a prefix code
- B is a C-1 bit number

Partial prefix code table

Z \ C	0	1	2	3
0	1010	00	01	100
1		1100	11011	11110001
2		1110	11111001	1111110111
3		111010	111110111	111111110101

(0,3,2) (0,5,7) (0,3,3) (4,2,1) (0,2,1) (2,2,1) (0,0,-)

100 10 11010 0111 100 11 1111111000 1 01 1 11111001 1 1010

46 bits representing 64 pixels = .72 bpp

Notes on Transform Coding

- Video Coding
  - MPEG – uses DCT
  - H.263, H.264 – uses DCT
- Audio Coding
  - MP3 = MPEG 1- Layer 3 uses DCT
- Alternative Transforms
  - Lapped transforms remove some of the blocking artifacts.
  - Wavelet transforms do not need to use blocks at all.