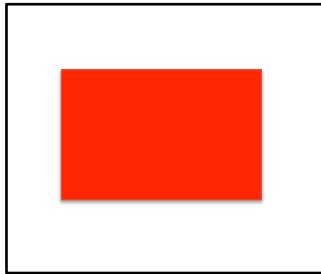

Lecture 5: Compression I

Reading: book chapter 6, section 3 & 5
chapter 7, section 1, 2, 3, 4, 8

This Week's Schedule

- Today:
 - The concept behind compression
 - Rate distortion theory
 - Image compression via DCT
- Wed.:
 - Speech compression via Prediction
 - Video compression via IPB and motion estimation/
compensation

Motivation



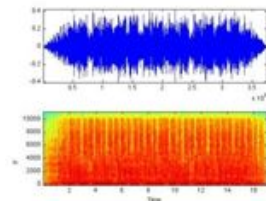
- A simple example: a 10x10 colored image, how many bits are required to represent this image:

- Without compression
- With compression

The concept behind compression is to extract and remove redundancy that naturally exists in media data;
In many cases, we can remove information without affecting the visual/audio effect, because human eyes/ears have limited sensitivity

Redundancy in Media Data

- Medias (speech, audio, image, video) are not random collection of signals, but exhibit a similar structure in local neighborhood
 - **Temporal redundancy**: current and next signals are very similar (smooth media: speech, audio, video)
 - **Spatial redundancy**: the pixels' intensities and colors in local regions are very similar (image, video)
 - **Spectral redundancy**: When the data is mapped into the frequency domain, a few frequencies dominate over the others



Lossless Compression

- Lossless compression
 - Compress the signal but can reproduce **the exact** original signal
 - Used for archival purposes and often medical imaging, technical drawings
 - Example 1: **Run Length Encoding** (BMP, PCX)
BBBBEEEEEEEECCCCDAAAAA → 4B8E4C1D5A
 - Example 2: **Lempel-Ziv-Welch (LZW)**: adaptive dictionary, dynamically create a dictionary of strings to efficiently represent messages, used in GIF & TIFF
 - Example 3: **Huffman coding**: the length of the codeword to present a symbol (or a value) scales inversely with the probability of the symbol's appearance, used in PNG, MNG, TIFF

Huffman Coding

Fixed-length coding

Symbol	Probability	Binary Code	Code length
A	0.28	000	3
B	0.2	001	3
C	0.17	010	3
D	0.17	011	3
E	0.1	100	3
F	0.05	101	3
G	0.02	110	3
H	0.01	111	3

Average symbol length=3



Symbol	Probability	Binary Code	Code length
A	0.28	00	1
B	0.2	10	3
C	0.17	010	3
D	0.17	011	3
E	0.1	110	3
F	0.05	1110	4
G	0.02	11110	5
H	0.01	11111	5

Average symbol length=2.63

The length of the codeword to present a symbol (or a value) scales inversely with the probability of the symbol's appearance.

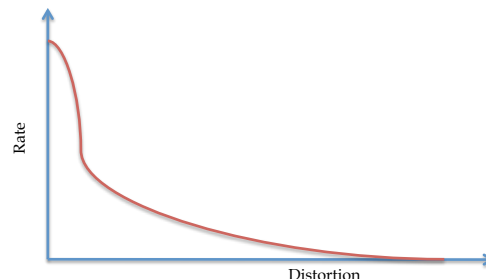
Lossy Compression

- The compressed signal after de-compressed, does not match the original signal
 - Compression leads to some signal **distortion**
 - Suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a **substantial reduction** in bit rate.
- Types
 - **Color space reduction**: reduce 24→8bits via color lookup table
 - **Chrominance subsampling**: from 4:4:4 to 4:2:2, 4:1:1, 4:2:0
 - eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information
 - **Transform coding** (or perceptual coding): Fourier transform (DCT, wavelet) followed by quantization and entropy coding

Today's focus

Rate Distortion Theory

- As the degree of compression increases, the number of bits used to represent the image reduces, and this increases the distortion



Distortion Measures

The three most commonly used distortion measures in image are:

- **mean square error** (MSE) σ^2 ,

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - y_n)^2$$

where x_n , y_n , and N are the input data sequence, reconstructed data sequence, and length of the data sequence respectively.

- **signal to noise ratio** (SNR), in decibel units (dB),

$$SNR = 10 \cdot \log_{10} \left(\frac{\frac{1}{N} \sum_{n=1}^N x_n^2}{\sigma^2} \right)$$

- **peak signal to noise ratio** (PSNR), in decibel units (dB),

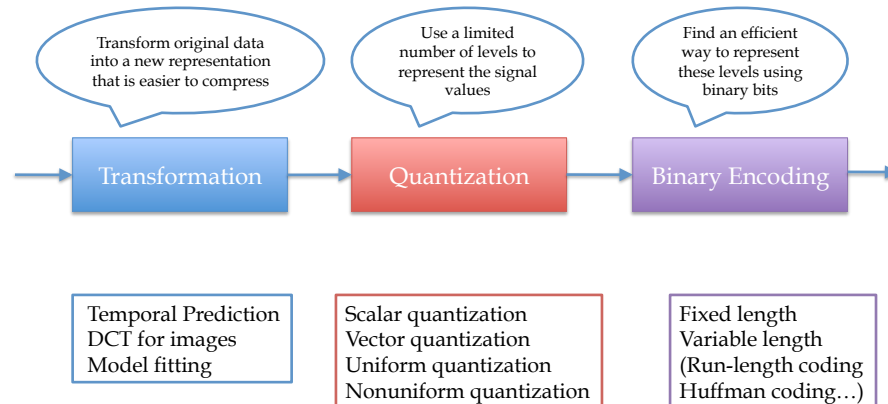
$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{\sigma^2} \right) = 20 \cdot \log_{10} \left(\frac{255}{\sigma} \right)$$

Lossy Compression

- The compressed signal after de-compressed, does not match the original signal
 - Compression leads to some signal distortion
 - Suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a **substantial reduction** in bit rate.
- Types
 - **Color space reduction**: reduce 24→8bits via color lookup table
 - **Chrominance subsampling**: from 4:4:4 to 4:2:2, 4:1:1, 4:2:0, eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information
 - **Transform coding** (or perceptual coding): Fourier transform (DCT, wavelet) followed by quantization and entropy coding

Today's focus

A Typical Compression System



Why Transformation?

- Goal of transformation:
 - To yield a more efficient representation of the original samples.
 - The transformed parameters should require fewer bits to code.
- Types of transformation used:
 - For speech coding: prediction
 - Code predictor and prediction error sample
 - For audio coding: subband decomposition
 - Code subband samples
 - **For image coding: DCT and wavelet transforms**
 - Code DCT/wavelet coefficients



Image: Transformation

- Represent an image as the linear combination of some basis images and specify the linear coefficients
- **Instead of storing the original image, now store the linear coefficients $\{t_k\}$**

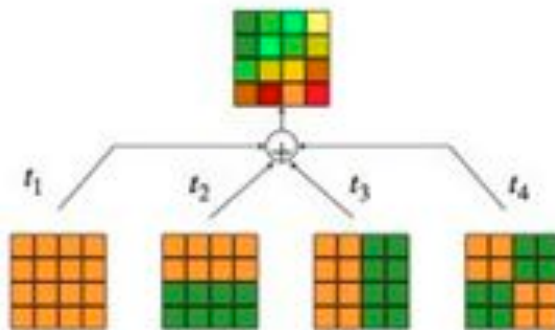


Image: Transformation II

- Optimality Criteria:
 - **Energy compaction:** a few basis images are sufficient to represent a typical image.
 - **Decorrelation:** coefficients for separate basis images are uncorrelated.
- Karhunen Loeve Transform (KLT) is the Optimal transform for a given covariance matrix of the underlying signal.
- **Discrete Cosine Transform** (DCT) is close to KLT for images that can be modeled by a first order Markov process (i.e., a pixel only depends on its previous pixel).

1D Transformation

$$s = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{N-1} \end{bmatrix}$$



The inverse transform says that s can be represented as the sum of N basis vectors

$$s = t_0 u_0 + t_1 u_1 + \dots + t_{N-1} u_{N-1}$$

$$u_k = \begin{bmatrix} u_{k,0} \\ u_{k,1} \\ \vdots \\ u_{k,N-1} \end{bmatrix}$$

where u_k corresponds to the k -th transform kernel:



$$t_k = [s_0 \quad s_1 \quad \dots \quad s_{N-1}] \cdot \begin{bmatrix} u_{k,0} \\ u_{k,1} \\ \vdots \\ u_{k,N-1} \end{bmatrix}^* = \sum_{n=1}^{N-1} u_{k,n}^* \cdot s_n$$

The forward transform says that the expansion coefficient t_k can be determined by the inner product of s with u_k

1D Discrete Cosine Transform

$$u_{k,n} = \alpha(k) \cos\left(\frac{\pi k}{2N} (2n+1)\right)$$

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}}, \quad k = 1, 2, \dots, N-1$$

$$t_k = \sum_{n=1}^{N-1} u_{k,n}^* \cdot s_n$$

$$u_k = \alpha(k) \begin{bmatrix} \cos\left(\frac{\pi k}{2N}\right) \\ \cos\left(\frac{\pi k}{2N} \cdot 3\right) \\ \vdots \\ \cos\left(\frac{\pi k}{2N} (2N+1)\right) \end{bmatrix}$$

Example 4-Point DCT

Using $u_{k,n} = \cos(k) \cos\left(\frac{k\pi}{2n+1}\right)$, $u(0) = \frac{1}{4} = \frac{1}{2} \cdot \cos(k) = \frac{1}{4} = \frac{1}{2}$, $k \neq 0$.

$$\text{1D DCT basis are: } \mathbf{u}_0 = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{u}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{\pi}{8}\right) \\ \cos\left(\frac{3\pi}{8}\right) \\ \cos\left(\frac{5\pi}{8}\right) \\ \cos\left(\frac{7\pi}{8}\right) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0.9239 \\ 0.3827 \\ -0.3827 \\ -0.9239 \end{bmatrix}, \mathbf{u}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{2\pi}{8}\right) \\ \cos\left(\frac{6\pi}{8}\right) \\ \cos\left(\frac{4\pi}{8}\right) \\ \cos\left(\frac{7\pi}{8}\right) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \mathbf{u}_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos\left(\frac{3\pi}{8}\right) \\ \cos\left(\frac{9\pi}{8}\right) \\ \cos\left(\frac{15\pi}{8}\right) \\ \cos\left(\frac{21\pi}{8}\right) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0.3827 \\ -0.9239 \\ 0.9239 \\ -0.3827 \end{bmatrix}$$

For $\mathbf{s} = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 3 \end{bmatrix}$ determine the transform coefficients t_i . Also determine the reconstructed vector from all coefficients and two largest coefficients.

Group Exercise !

2D Transformation

$$\mathbf{S} = \begin{bmatrix} S_{0,0} & S_{0,1} & \dots & S_{0,N-1} \\ S_{1,0} & S_{1,1} & \dots & S_{1,N-1} \\ \dots & \dots & \dots & \dots \\ S_{M-1,0} & S_{M-1,1} & \dots & S_{M-1,N-1} \end{bmatrix}$$

$$\mathbf{S} = T_{0,0} \mathbf{U}_{0,0} + T_{0,1} \mathbf{U}_{0,1} + \dots + T_{M-1,N-1} \mathbf{U}_{M-1,N-1}$$

$$\mathbf{U}_{k,l} = \mathbf{u}_k (\mathbf{u}_l)^T = \begin{bmatrix} u_{k,0} \\ u_{k,1} \\ \dots \\ u_{k,N-1} \end{bmatrix} \begin{bmatrix} u_{l,0}^* & u_{l,1}^* & \dots & u_{l,N-1}^* \end{bmatrix} = \begin{bmatrix} u_{k,0} u_{l,0}^* & u_{k,0} u_{l,1}^* & \dots & u_{k,0} u_{l,N-1}^* \\ u_{k,1} u_{l,0}^* & u_{k,1} u_{l,1}^* & \dots & u_{k,1} u_{l,N-1}^* \\ \dots & \dots & \dots & \dots \\ u_{k,N-1} u_{l,0}^* & u_{k,N-1} u_{l,1}^* & \dots & u_{k,N-1} u_{l,N-1}^* \end{bmatrix}$$



$$T_{k,l} = (\mathbf{U}_{k,l}, \mathbf{S}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} U_{k,l,m,n}^* S_{m,n}$$

2D DCT

MATLAB © dct2

$$u_{k,n} = \cos\left(k\pi\left(n + \frac{1}{2}\right)\right) \quad \text{for } k = 0, 1, \dots, N-1$$

$$U_{k,l} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u_{k,n} \cos\left(l\pi\left(n + \frac{1}{2}\right)\right)$$

4x4 DCT

$$\text{Using } u_{k,n} = \cos\left(k\pi\left(n + \frac{1}{2}\right)\right) \quad \text{for } k = 0, 1, \dots, N-1$$

$$\text{1D DCT basis are: } u_k = \frac{1}{\sqrt{4}} \begin{bmatrix} \cos\left(\frac{k\pi}{4}\right) \\ \cos\left(\frac{3k\pi}{4}\right) \\ \cos\left(\frac{5k\pi}{4}\right) \\ \cos\left(\frac{7k\pi}{4}\right) \end{bmatrix}$$

using $U_{k,n} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u_{k,n} \cos\left(l\pi\left(n + \frac{1}{2}\right)\right)$ yields:

$$U_{k,l} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \text{matrix } E_1$$

$$\text{Using } E_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \text{matrix } E_2$$

$$E_2 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \text{matrix } E_3$$

$$E_3 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \text{matrix } E_4$$

Computing eigenvalues using Matlab's built-in function "eig":

$$E = \begin{bmatrix} 0.71 & -0.0000 & 0.00 & 0.0000 \\ 0.0000 & -0.00 & 0.0000 & 0.00 \\ 0.00 & 0.0000 & -0.00 & -0.0000 \\ -0.0000 & 0.00 & 0.0000 & -0.00 \end{bmatrix}$$

Determining the eigenvectors using Matlab's built-in function "eig":

$$V = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

The reconstructed image is shown below the original because we can't tell any high frequency coefficients.

Matlab Demo

```

x=imread('avatar1.jpg');
y=x(:,:,1)*0.299 + x(:,:,2)*0.587 + x(:,:,3)*0.114; %convert RGB → Y
subplot(2,2,1); imshow(y);
[m,n]=size(y);
m=floor(m/8); n=floor(n/8);
for i=1:m,
    for j=1:n,
        z=y((i-1)*8+1:i*8, (j-1)*8+1:j*8);
        d(i,j,:)=dct2(z);
        zz((i-1)*8+1:i*8, (j-1)*8+1:j*8)=d(i,j,:);
    end;
end;
y(9*8+1:80,9*8+1:80)
squeeze(d(10,10,:,:))
surf(squeeze(d(10,10,:,:)))
subplot(2,2,2); imshow(zz);
  
```

14	12	8	8	8	9	8	8
13	12	8	7	8	8	8	7
11	10	8	7	7	8	7	7
10	8	8	7	7	7	7	7
8	8	8	7	7	7	7	7
7	7	8	8	7	7	7	8
7	7	8	9	7	7	7	8
7	7	10	9	8	7	7	10

↓

64.3750	5.2042	3.1427	0.4215	0.6250	-0.1329	0.7277	-0.3977
4.0089	5.2874	4.1819	5.1303	-0.9152	-0.8085	-1.2858	-0.4849
3.7400	0.4698	0.5366	-0.2701	0.7209	-0.3769	-0.0884	-0.5441
-0.1462	-0.3201	-0.0846	0.4658	-0.0402	0.1594	-0.5902	-0.5669
0.6250	-0.1584	0.1237	-0.3610	0.3750	-0.0028	0.6253	0.5311
-0.4242	-0.1350	-0.1379	0.2629	0.5487	0.7664	0.2623	-0.6434
0.2097	-0.5612	-0.0884	-0.5091	0.1073	-0.1914	0.7134	-0.1806
-0.2003	0.1186	-0.1791	0.1318	-0.4151	-0.1678	-0.0892	-0.5197

Recover via IDCT

% No compression

```
for i=1:m,
    for j=1:n,
        u=squeeze(d(i,j,:));
        w((i-1)*8+1:i*8, (j-1)*8+1:j*8)=idct2(u);
    end;
end;
subplot(2,2,3); imshow(w,[0,255]);
```

% Some level of compression

```
for i=1:m,
    for j=1:n,
        u=squeeze(d(i,j,:));
        f= zeros(8,8);
        k=1;
        f(1:k,1:k)=u(1:k,1:k); % only consider the first kxk coefficient
        v((i-1)*8+1:i*8, (j-1)*8+1:j*8)=idct2(f);
    end;
end;
subplot(2,2,4); imshow(v,[0,255]);
```

Matlab Results

an original image block

ans =

```
14 12 8 8 8 9 8 8
13 12 8 7 8 8 8 7
11 10 8 7 7 8 7 7
10 8 8 7 7 7 7 7
8 8 8 7 7 7 7 7
7 7 8 8 7 7 7 8
7 7 8 9 7 7 7 8
7 7 10 9 8 7 7 10
```

its DCT coefficients

ans =

```
64.3750 5.2042 3.1427 0.4215 0.6250 -0.1329 0.7277 -0.3977
4.0089 5.2874 4.1819 5.1303 -0.9152 -0.8085 -1.2858 -0.4849
3.7400 0.4698 0.5366 -0.2701 0.7209 -0.3769 -0.0884 -0.5441
-0.1462 -0.3201 -0.0846 0.4658 -0.0402 0.1594 -0.5902 -0.5669
0.6250 -0.1584 0.1237 -0.3610 0.3750 -0.0028 0.6253 0.5311
-0.4242 -0.1350 -0.1379 0.2629 0.5487 0.7664 0.2623 -0.6434
0.2097 -0.5612 -0.0884 -0.5091 0.1073 -0.1914 0.7134 -0.1806
-0.2003 0.1186 -0.1791 0.1318 -0.4151 -0.1678 -0.0892 -0.5197
```

recovered image block

ans =

```
14 12 8 8 8 9 8 8
13 12 8 7 8 8 8 7
11 10 8 7 7 8 7 7
10 8 8 7 7 7 7 7
8 8 8 7 7 7 7 7
7 7 8 8 7 7 7 8
7 7 8 9 7 7 7 8
7 7 10 9 8 7 7 10
```

recovered with first 4x4 coefficient

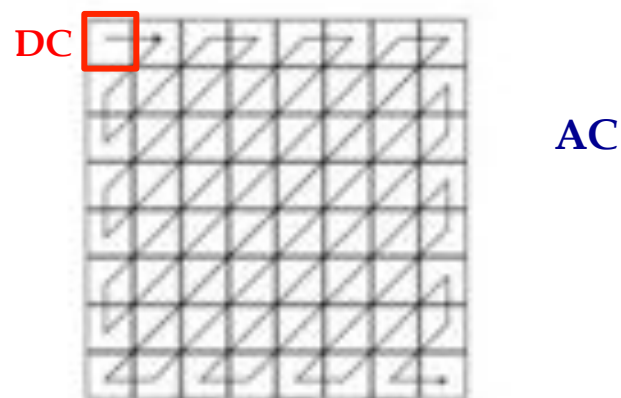
ans =

```
14 12 9 7 8 9 8 8
13 11 8 7 8 8 8 7
11 10 8 7 7 8 7 7
10 9 8 7 7 7 7 7
8 8 8 7 7 7 7 7
7 7 8 8 7 7 7 8
6 8 9 8 7 7 8 8
6 8 9 9 8 7 8 9
```

Compression via DCT



Zig-Zag Ordering of DCT Coefficients



Zig-Zag ordering: converting a 2D matrix into a 1D array, so that the frequency (horizontal+vertical) increases in this order, and the coefficient variance (average of magnitude square) decreases in this order.

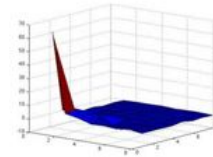
After Zig-Zag Ordering

Original
block

14	12	8	8	8	9	8	8
13	12	8	7	8	8	8	7
11	10	8	7	7	8	7	7
10	8	8	7	7	7	7	7
8	8	8	7	7	7	7	7
7	7	8	8	7	7	7	8
7	7	8	9	7	7	7	8
7	7	10	9	8	7	7	10

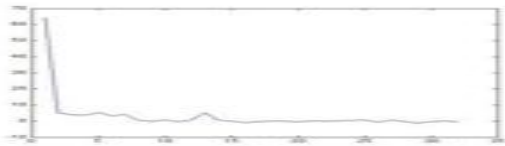
DCT
results

64.3750	5.2042	3.1427	0.4215	0.6250	-0.1329	0.7277	-0.3977
4.0089	5.2874	4.1819	5.1303	-0.9152	-0.8085	-1.2858	-0.4849
3.7400	0.4698	0.5366	-0.2701	0.7209	-0.3769	-0.0884	-0.5441
-0.1462	-0.3201	-0.0846	0.4658	-0.0402	0.1594	-0.5902	-0.5669
0.6250	-0.1584	0.1237	-0.3610	0.3750	-0.0028	0.6253	0.5311
-0.4242	-0.1350	-0.1379	0.2629	0.5487	0.7664	0.2623	-0.6434
0.2097	-0.5612	-0.0884	-0.5091	0.1073	-0.1914	0.7134	-0.1806
-0.2003	0.1186	-0.1791	0.1318	-0.4151	-0.1678	-0.0892	-0.5197

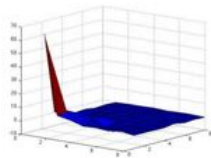
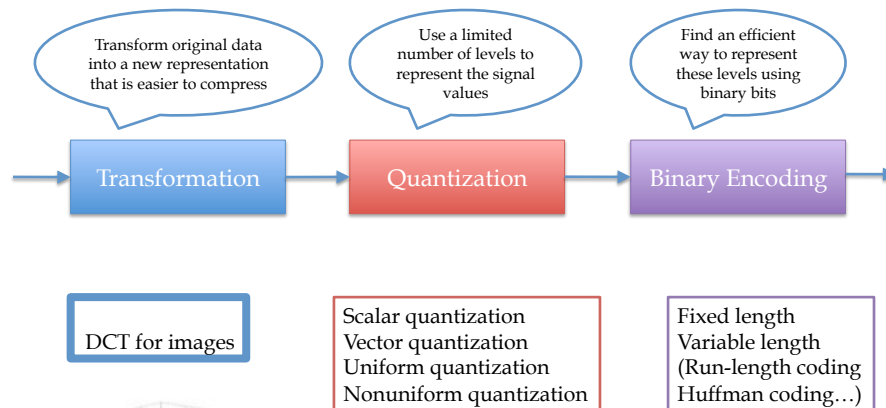


Zig-Zag
ordering

64.3750 5.2042 4.0089 3.7400 5.2874 3.1427 4.1819 0.4698 -0.1462 0.6250 -0.3201 0.5366 5.1303 0.6250 -0.1329 -0.9152 -0.2701 -0.0846 -0.1584 -0.4242 0.2097 -0.1350 0.1237 0.4658 0.7209 -0.8085 0.7277 -0.3977 -1.2858 -0.3769 -0.0402 -0.3610



Recap: A Typical Compression System

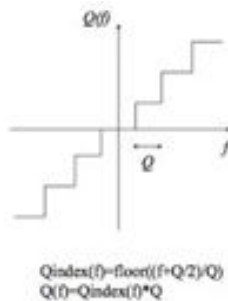


Quantization

- Reduce the number of distinct output values to a much smaller set.
- Main source of the "loss" in lossy compression.
- Three different forms of quantization.
 - **Uniform**: midrise and midtread quantizers.
 - **Nonuniform**: companded quantizer.
 - **Vector Quantization**.

Uniform Quantization

- A uniform scalar quantizer partitions the domain of input values into equally spaced intervals, except possibly at the two outer intervals.
 - The output or reconstruction value corresponding to each interval is taken to be the midpoint of the interval.
 - The length of each interval is referred to as the *step size* Q



Quantizing DCT Coefficients

- Use **uniform quantizer** on each coefficient
- Different coefficient is quantized with different step-size (Q):
 - Human eye is more sensitive to low frequency components
 - Low frequency coefficients with a smaller Q
 - High frequency coefficients with a larger Q
 - Specified in a normalization matrix
 - Normalization matrix can then be scaled by a scale factor (QP), i.e. Actual quantization = $QP * Q$, $QP=1, 2, 3, 4, \dots$

JPEG Quantization

- For Luminance component

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	68	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

Q =

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

Original
block

14	12	8	8	8	9	8	8
13	12	8	7	8	8	8	7
11	10	8	7	7	8	7	7
10	8	8	7	7	7	7	7
8	8	8	7	7	7	7	7
7	7	8	8	7	7	7	8
7	7	8	9	7	7	7	8
7	7	10	9	8	7	7	10

DCT
results

64.3750	5.2042	3.1427	0.4215	0.6250	-0.1329	0.7277	-0.3977
4.0089	5.2874	4.1819	5.1303	-0.9152	-0.8085	-1.2858	-0.4849
3.7400	0.4698	0.5366	-0.2701	0.7209	-0.3769	-0.0884	-0.5441
-0.1462	-0.3201	-0.0846	0.4658	-0.0402	0.1594	-0.5902	-0.5669
0.6250	-0.1584	0.1237	-0.3610	0.3750	-0.0028	0.6253	0.5311
-0.4242	-0.1350	-0.1379	0.2629	0.5487	0.7664	0.2623	-0.6434
0.2097	-0.5612	-0.0884	-0.5091	0.1073	-0.1914	0.7134	-0.1806
-0.2003	0.1186	-0.1791	0.1318	-0.4151	-0.1678	-0.0892	-0.5197

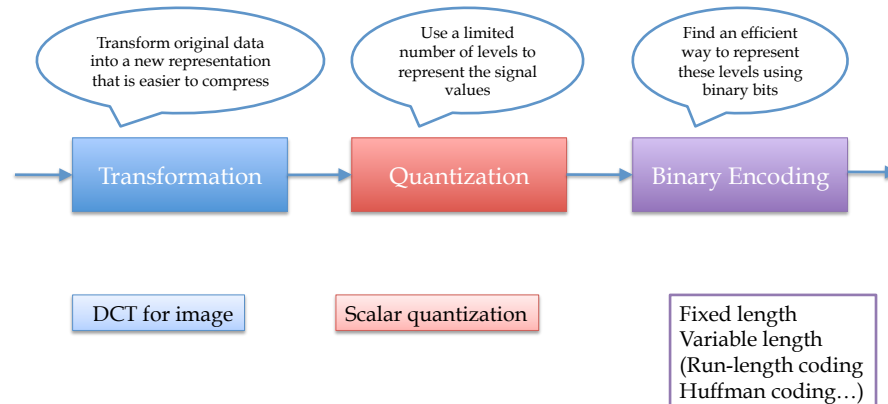
Quantized
& dequantized
DCT results

4	0	0	0	0	0	0	0	64	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	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	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

Recovered
block

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8

A Typical Compression System



Coding Quantized DCT coefficient

- **DC coefficient:** Predictive coding
 - The DC value of the current block is predicted from that of the **previous block**, and the error is coded using Huffman coding
- **AC Coefficients:** Run length coding
 - Many high frequency AC coefficients are zero after first few low-frequency coefficients
 - Runlength Representation:
 - Ordering coefficients in the **zig-zag** order
 - Specify how many zeros before a non-zero value
 - Each **symbol=(length-of-zero, non-zero-value)**
 - Code all possible symbols using **Huffman coding**
 - More frequently appearing symbols are given shorter codewords
 - One can use default Huffman tables or specify its own tables

An Illustrative Example

quant =

DC	2	5	0	-2	0	-1	0	0
	9	1	-1	2	0	1	0	0
	14	1	-1	0	-1	0	0	0
	3	-1	-1	-1	0	0	0	0
	2	-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



Zig-zag + run length encoding

DC {2,(0,5),(0,9),(0,14),(0,1),(1,-2),(0,-1),(0,1),(0,3),(0,2),(0,-1),(0,-1),(0,2),(1,-1),(2,-1),(0,-1),(4,-1),(0,-1),(0,1),EOB}

EOB: End of block, one of the symbol that is assigned a short Huffman codeword

Coding DC component

- Current quantized DC index: 2
- Previous block DC index: 4
- Prediction error: -2
 - The prediction error is coded in two parts:
 - **Which category it belongs to** (in the Table of JPEG Coefficient Coding Categories), and code using a Huffman code (JPEG Default DC Code)
 - DC= -2 is in category “2”, with a codeword “100”
 - **Which position it is in that category**, using a fixed length code, length=category number
 - “-2” in category 2, has a fixed length code of “10”.
 - The overall codeword is “10010”

JPEG Tables for Coding DC

Range	DC Difference Category	AC Category
0	0	N/A
-1, -2, 1, 2	1	1
-3, -4, 3, 4	2	2
-5, -6, 5, 6	3	3
-7, -8, 7, 8	4	4
-9, -10, 9, 10	5	5
-11, -12, 11, 12	6	6
-13, -14, 13, 14	7	7
-15, -16, 15, 16	8	8
-17, -18, 17, 18	9	9
-19, -20, 19, 20	10	10
-21, -22, 21, 22	11	11
-23, -24, 23, 24	12	12
-25, -26, 25, 26	13	13
-27, -28, 27, 28	14	14
-29, -30, 29, 30	15	15
-31, -32, 31, 32	16	16
-33, -34, 33, 34	17	17
-35, -36, 35, 36	18	18
-37, -38, 37, 38	19	19
-39, -40, 39, 40	20	20
-41, -42, 41, 42	21	21
-43, -44, 43, 44	22	22
-45, -46, 45, 46	23	23
-47, -48, 47, 48	24	24
-49, -50, 49, 50	25	25
-51, -52, 51, 52	26	26
-53, -54, 53, 54	27	27
-55, -56, 55, 56	28	28
-57, -58, 57, 58	29	29
-59, -60, 59, 60	30	30
-61, -62, 61, 62	31	31
-63, -64, 63, 64	32	32
-65, -66, 65, 66	33	33
-67, -68, 67, 68	34	34
-69, -70, 69, 70	35	35
-71, -72, 71, 72	36	36
-73, -74, 73, 74	37	37
-75, -76, 75, 76	38	38
-77, -78, 77, 78	39	39
-79, -80, 79, 80	40	40
-81, -82, 81, 82	41	41
-83, -84, 83, 84	42	42
-85, -86, 85, 86	43	43
-87, -88, 87, 88	44	44
-89, -90, 89, 90	45	45
-91, -92, 91, 92	46	46
-93, -94, 93, 94	47	47
-95, -96, 95, 96	48	48
-97, -98, 97, 98	49	49
-99, -100, 99, 100	50	50
-101, -102, 101, 102	51	51
-103, -104, 103, 104	52	52
-105, -106, 105, 106	53	53
-107, -108, 107, 108	54	54
-109, -110, 109, 110	55	55
-111, -112, 111, 112	56	56
-113, -114, 113, 114	57	57
-115, -116, 115, 116	58	58
-117, -118, 117, 118	59	59
-119, -120, 119, 120	60	60
-121, -122, 121, 122	61	61
-123, -124, 123, 124	62	62
-125, -126, 125, 126	63	63
-127, -128, 127, 128	64	64
-129, -130, 129, 130	65	65
-131, -132, 131, 132	66	66
-133, -134, 133, 134	67	67
-135, -136, 135, 136	68	68
-137, -138, 137, 138	69	69
-139, -140, 139, 140	70	70
-141, -142, 141, 142	71	71
-143, -144, 143, 144	72	72
-145, -146, 145, 146	73	73
-147, -148, 147, 148	74	74
-149, -150, 149, 150	75	75
-151, -152, 151, 152	76	76
-153, -154, 153, 154	77	77
-155, -156, 155, 156	78	78
-157, -158, 157, 158	79	79
-159, -160, 159, 160	80	80
-161, -162, 161, 162	81	81
-163, -164, 163, 164	82	82
-165, -166, 165, 166	83	83
-167, -168, 167, 168	84	84
-169, -170, 169, 170	85	85
-171, -172, 171, 172	86	86
-173, -174, 173, 174	87	87
-175, -176, 175, 176	88	88
-177, -178, 177, 178	89	89
-179, -180, 179, 180	90	90
-181, -182, 181, 182	91	91
-183, -184, 183, 184	92	92
-185, -186, 185, 186	93	93
-187, -188, 187, 188	94	94
-189, -190, 189, 190	95	95
-191, -192, 191, 192	96	96
-193, -194, 193, 194	97	97
-195, -196, 195, 196	98	98
-197, -198, 197, 198	99	99
-199, -200, 199, 200	100	100
-201, -202, 201, 202	101	101
-203, -204, 203, 204	102	102
-205, -206, 205, 206	103	103
-207, -208, 207, 208	104	104
-209, -210, 209, 210	105	105
-211, -212, 211, 212	106	106
-213, -214, 213, 214	107	107
-215, -216, 215, 216	108	108
-217, -218, 217, 218	109	109
-219, -220, 219, 220	110	110
-221, -222, 221, 222	111	111
-223, -224, 223, 224	112	112
-225, -226, 225, 226	113	113
-227, -228, 227, 228	114	114
-229, -230, 229, 230	115	115
-231, -232, 231, 232	116	116
-233, -234, 233, 234	117	117
-235, -236, 235, 236	118	118
-237, -238, 237, 238	119	119
-239, -240, 239, 240	120	120
-241, -242, 241, 242	121	121
-243, -244, 243, 244	122	122
-245, -246, 245, 246	123	123
-247, -248, 247, 248	124	124
-249, -250, 249, 250	125	125
-251, -252, 251, 252	126	126
-253, -254, 253, 254	127	127
-255, -256, 255, 256	128	128
-257, -258, 257, 258	129	129
-259, -260, 259, 260	130	130
-261, -262, 261, 262	131	131
-263, -264, 263, 264	132	132
-265, -266, 265, 266	133	133
-267, -268, 267, 268	134	134
-269, -270, 269, 270	135	135
-271, -272, 271, 272	136	136
-273, -274, 273, 274	137	137
-275, -276, 275, 276	138	138
-277, -278, 277, 278	139	139
-279, -280, 279, 280	140	140
-281, -282, 281, 282	141	141
-283, -284, 283, 284	142	142
-285, -286, 285, 286	143	143
-287, -288, 287, 288	144	144
-289, -290, 289, 290	145	145
-291, -292, 291, 292	146	146
-293, -294, 293, 294	147	147
-295, -296, 295, 296	148	148
-297, -298, 297, 298	149	149
-299, -300, 299, 300	150	150
-301, -302, 301, 302	151	151
-303, -304, 303, 304	152	152
-305, -306, 305, 306	153	153
-307, -308, 307, 308	154	154
-309, -310, 309, 310	155	155
-311, -312, 311, 312	156	156
-313, -314, 313, 314	157	157
-315, -316, 315, 316	158	158
-317, -318, 317, 318	159	159
-319, -320, 319, 320	160	160
-321, -322, 321, 322	161	161
-323, -324, 323, 324	162	162
-325, -326, 325, 326	163	163
-327, -328, 327, 328	164	164
-329, -330, 329, 330	165	165
-331, -332, 331, 332	166	166
-333, -334, 333, 334	167	167
-335, -336, 335, 336	168	168
-337, -338, 337, 338	169	169
-339, -340, 339, 340	170	170
-341, -342, 341, 342	171	171
-343, -344, 343, 344	172	172
-345, -346, 345, 346	173	173
-347, -348, 347, 348	174	174
-349, -350, 349, 350	175	175
-351, -352, 351, 352	176	176
-353, -354, 353, 354	177	177
-355, -356, 355, 356	178	178
-357, -358, 357, 358	179	179
-359, -360, 359, 360	180	180
-361, -362, 361, 362	181	181
-363, -364, 363, 364	182	182
-365, -366, 365, 366	183	183
-367, -368, 367, 368	184	184
-369, -370, 369, 370	185	185
-371, -372, 371, 372	186	186
-373, -374, 373, 374	187	187
-375, -376, 375, 376	188	188
-377, -378, 377, 378	189	189
-379, -380, 379, 380	190	190
-381, -382, 381, 382	191	191
-383, -384, 383, 384	192	192
-385, -386, 385, 386	193	193
-387, -388, 387, 388	194	194
-389, -390, 389, 390	195	195
-391, -392, 391, 392	196	196
-393, -394, 393, 394	197	197
-395, -396, 395, 396	198	198
-397, -398, 397, 398	199	199
-399, -400, 399, 400	200	200
-401, -402, 401, 402	201	201
-403, -404, 403, 404	202	202
-405, -406, 405, 406	203	203
-407, -408, 407, 408	204	204
-409, -410, 409, 410	205	205
-411, -412, 411, 412	206	206
-413, -414, 413, 414	207	207
-415, -416, 415, 416	208	208
-417, -418, 417, 418	209	209
-419, -420, 419, 420	210	210
-421, -422, 421, 422	211	211
-423, -424, 423, 424	212	212
-425, -426, 425, 426	213	213
-427, -428, 427, 428	214	214
-429, -430, 429, 430	215	215
-431, -432, 431, 432	216	216
-433, -434, 433, 434	217	217
-435, -436, 435, 436	218	218
-437, -438, 437, 438	219	219
-439, -440, 439, 440	220	220
-441, -442, 441, 442	221	221
-443, -444, 443, 444	222	222
-445, -446, 445, 446	223	223
-447, -448, 447, 448	224	224
-449, -450, 449, 450	225	225
-451, -452, 451, 452	226	226
-453, -454, 453, 454	227	227
-455, -456, 455, 456	228	228
-457, -458, 457, 458	229	229
-459, -460, 459, 460	230	230
-461, -462, 461, 462	231	231
-463, -464, 463, 464	232	232
-465, -466, 465, 466	233	233
-467, -468, 467, 468	234	234
-469, -470, 469, 470	235	235
-471, -472, 471, 472	236	236
-473, -474, 473, 474	237	237
-475, -476, 475, 476	238	238
-477, -478, 477, 478	239	239
-479, -480, 479, 480	240	240
-481, -482, 481, 482	241	241
-483, -484, 483, 484	242	242
-485, -486, 485, 486	243	243
-487, -488, 487, 488	244	244
-489, -490, 489, 490	245	245
-491, -492, 491, 492	246	246
-493, -494, 493, 494	247	247
-495, -496, 495, 496	248	248
-497, -498, 497, 498	249	249
-499, -500, 499, 500	250	250
-501, -502, 501, 502	251	251
-503, -504, 503, 504	252	252
-505, -506, 505, 506	253	253
-507, -508, 507, 508	254	254
-509, -510, 509, 510	255	255
-511, -512, 511, 512	256	256
-513, -514, 513, 514	257	257
-515, -516, 515, 516	258	258
-517, -518, 517, 518	259	259
-519, -520, 519, 520	260	260
-521, -522, 521, 522	261	261
-523, -524, 523, 524	262	262
-525, -526, 525, 526	263	263
-527, -528, 527, 528	264	264
-529, -530, 529, 530	265	265
-531, -532, 531, 532	266	266
-533, -534, 533, 534	267	267
-535, -536, 535, 536	268	268
-537, -538, 537, 538	269	269
-539, -540, 539, 540	270	270
-541, -542, 541, 542	271	271
-543, -544, 543, 544	272	272
-545, -546, 545, 546	273	273
-547, -548, 547, 548	274	274
-549, -550, 549, 550	275	275
-551, -552, 551, 552	276	276
-553, -554, 553, 554	277	277
-555, -556, 555, 556	278	278
-557, -558, 557, 558	279	279
-559, -560, 559, 560	280	280
-561, -562, 561, 562	281	281
-56		

JPEG Tables for Coding DC

Representation inside the category
11 for -3, 10 for -2, 01 for 2, 00 for 3

DC Difference	Category	AC Category
0	0	N/A
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30
31	31	31
32	32	32
33	33	33
34	34	34
35	35	35
36	36	36
37	37	37
38	38	38
39	39	39
40	40	40
41	41	41
42	42	42
43	43	43
44	44	44
45	45	45
46	46	46
47	47	47
48	48	48
49	49	49
50	50	50
51	51	51
52	52	52
53	53	53
54	54	54
55	55	55
56	56	56
57	57	57
58	58	58
59	59	59
60	60	60
61	61	61
62	62	62
63	63	63
64	64	64
65	65	65
66	66	66
67	67	67
68	68	68
69	69	69
70	70	70
71	71	71
72	72	72
73	73	73
74	74	74
75	75	75
76	76	76
77	77	77
78	78	78
79	79	79
80	80	80
81	81	81
82	82	82
83	83	83
84	84	84
85	85	85
86	86	86
87	87	87
88	88	88
89	89	89
90	90	90
91	91	91
92	92	92
93	93	93
94	94	94
95	95	95
96	96	96
97	97	97
98	98	98
99	99	99
100	100	100
101	101	101
102	102	102
103	103	103
104	104	104
105	105	105
106	106	106
107	107	107
108	108	108
109	109	109
110	110	110
111	111	111
112	112	112
113	113	113
114	114	114
115	115	115
116	116	116
117	117	117
118	118	118
119	119	119
120	120	120
121	121	121
122	122	122
123	123	123
124	124	124
125	125	125
126	126	126
127	127	127
128	128	128
129	129	129
130	130	130
131	131	131
132	132	132
133	133	133
134	134	134
135	135	135
136	136	136
137	137	137
138	138	138
139	139	139
140	140	140
141	141	141
142	142	142
143	143	143
144	144	144
145	145	145
146	146	146
147	147	147
148	148	148
149	149	149
150	150	150
151	151	151
152	152	152
153	153	153
154	154	154
155	155	155
156	156	156
157	157	157
158	158	158
159	159	159
160	160	160
161	161	161
162	162	162
163	163	163
164	164	164
165	165	165
166	166	166
167	167	167
168	168	168
169	169	169
170	170	170
171	171	171
172	172	172
173	173	173
174	174	174
175	175	175
176	176	176
177	177	177
178	178	178
179	179	179
180	180	180
181	181	181
182	182	182
183	183	183
184	184	184
185	185	185
186	186	186
187	187	187
188	188	188
189	189	189
190	190	190
191	191	191
192	192	192
193	193	193
194	194	194
195	195	195
196	196	196
197	197	197
198	198	198
199	199	199
200	200	200
201	201	201
202	202	202
203	203	203
204	204	204
205	205	205
206	206	206
207	207	207
208	208	208
209	209	209
210	210	210
211	211	211
212	212	212
213	213	213
214	214	214
215	215	215
216	216	216
217	217	217
218	218	218
219	219	219
220	220	220
221	221	221
222	222	222
223	223	223
224	224	224
225	225	225
226	226	226
227	227	227
228	228	228
229	229	229
230	230	230
231	231	231
232	232	232
233	233	233
234	234	234
235	235	235
236	236	236
237	237	237
238	238	238
239	239	239
240	240	240
241	241	241
242	242	242
243	243	243
244	244	244
245	245	245
246	246	246
247	247	247
248	248	248
249	249	249
250	250	250
251	251	251
252	252	252
253	253	253
254	254	254
255	255	255

Category	Base Code	Length	Category	Base Code	Length
0	000	3	6	1110	10
1	001	4	7	11100	12
2	100	5	8	111100	14
3	00	2	9	1111100	16
4	101	7	A	11111100	18
5	110	8	B	111111100	20

10010

Coding AC components

- First symbol (0,5) is represented in two parts:
 - Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code the “(runlength, category)” using a Huffman code (JPEG Default AC Code)
 - AC=5 is in category “3”,
 - Symbol (0,3) has codeword “100”
 - Which position it is in that category, using a fixed length code, length=category number
 - “5” is the number 5 (starting from 0) in category 3, with a fixed length code of “101”
 - The overall codeword for (0,5) is “100101”
- Second symbol (0,9)
 - ‘9’ in category ‘4’, (0,4) has codeword ‘1011’
 - ‘9’ is number 9 in category 4 with codeword ‘1001’
 - overall codeword for (0,9) is ‘10111001’

(runlength, category) Coding Table for AC symbol (0,5)

JPEG Coefficient Coding Categories

Range	DC Difference Category
0	0
-1, 1	1
-2, -3, 2, 3	2
-4, -5, -6, -7, 4, 5, 6, 7	3
-8, -9, -10, -11, 8, 9, 10, 11	4
-12, -13, -14, -15, 12, 13, 14, 15	5
-16, -17, -18, -19, 16, 17, 18, 19	6
-20, -21, -22, -23, 20, 21, 22, 23	7
-24, -25, -26, -27, 24, 25, 26, 27	8
-28, -29, -30, -31, 28, 29, 30, 31	9
-32, -33, -34, -35, 32, 33, 34, 35	A
-36, -37, -38, -39, 36, 37, 38, 39	B
-40, -41, -42, -43, 40, 41, 42, 43	C
-44, -45, -46, -47, 44, 45, 46, 47	D
-48, -49, -50, -51, 48, 49, 50, 51	E
-52, -53, -54, -55, 52, 53, 54, 55	F

Runlength	Category	Runlength	Category	Runlength	Category
1	0	1	0	1	0
2	0	2	0	2	0
3	0	3	0	3	0
4	0	4	0	4	0
5	0	5	0	5	0
6	0	6	0	6	0
7	0	7	0	7	0
8	0	8	0	8	0
9	0	9	0	9	0
10	0	10	0	10	0
11	0	11	0	11	0
12	0	12	0	12	0
13	0	13	0	13	0
14	0	14	0	14	0
15	0	15	0	15	0
16	0	16	0	16	0
17	0	17	0	17	0
18	0	18	0	18	0
19	0	19	0	19	0
20	0	20	0	20	0
21	0	21	0	21	0
22	0	22	0	22	0
23	0	23	0	23	0
24	0	24	0	24	0
25	0	25	0	25	0
26	0	26	0	26	0
27	0	27	0	27	0
28	0	28	0	28	0
29	0	29	0	29	0
30	0	30	0	30	0
31	0	31	0	31	0
32	0	32	0	32	0
33	0	33	0	33	0
34	0	34	0	34	0
35	0	35	0	35	0
36	0	36	0	36	0
37	0	37	0	37	0
38	0	38	0	38	0
39	0	39	0	39	0
40	0	40	0	40	0
41	0	41	0	41	0
42	0	42	0	42	0
43	0	43	0	43	0
44	0	44	0	44	0
45	0	45	0	45	0
46	0	46	0	46	0
47	0	47	0	47	0
48	0	48	0	48	0
49	0	49	0	49	0
50	0	50	0	50	0
51	0	51	0	51	0
52	0	52	0	52	0
53	0	53	0	53	0
54	0	54	0	54	0
55	0	55	0	55	0
56	0	56	0	56	0
57	0	57	0	57	0
58	0	58	0	58	0
59	0	59	0	59	0
60	0	60	0	60	0
61	0	61	0	61	0
62	0	62	0	62	0
63	0	63	0	63	0
64	0	64	0	64	0
65	0	65	0	65	0
66	0	66	0	66	0
67	0	67	0	67	0
68	0	68	0	68	0
69	0	69	0	69	0
70	0	70	0	70	0
71	0	71	0	71	0
72	0	72	0	72	0
73	0	73	0	73	0
74	0	74	0	74	0
75	0	75	0	75	0
76	0	76	0	76	0
77	0	77	0	77	0
78	0	78	0	78	0
79	0	79	0	79	0
80	0	80	0	80	0
81	0	81	0	81	0
82	0	82	0	82	0
83	0	83	0	83	0
84	0	84	0	84	0
85	0	85	0	85	0
86	0	86	0	86	0
87	0	87	0	87	0
88	0	88	0	88	0
89	0	89	0	89	0
90	0	90	0	90	0
91	0	91	0	91	0
92	0	92	0	92	0
93	0	93	0	93	0
94	0	94	0	94	0
95	0	95	0	95	0
96	0	96	0	96	0
97	0	97	0	97	0
98	0	98	0	98	0
99	0	99	0	99	0
100	0	100	0	100	0
101	0	101	0	101	0
102	0	102	0	102	0
103	0	103	0	103	0
104	0	104	0	104	0
105	0	105	0	105	0
106	0	106	0	106	0
107	0	107	0	107	0
108	0	108	0	108	0
109	0	109	0	109	0
110	0	110	0	110	0
111	0	111	0	111	0
112	0	112	0	112	0
113	0	113	0	113	0
114	0	114	0	114	0
115	0	115	0	115	0
116	0	116	0	116	0
117	0	117	0	117	0
118	0	118	0	118	0
119	0	119	0	119	0
120	0	120	0	120	0
121	0	121	0	121	0
122	0	122	0	122	0
123	0	123	0	123	0
124	0	124	0	124	0
125	0	125	0	125	0
126	0	126	0	126	0
127	0	127	0	127	0
128	0	128	0	128	0
129	0	129	0	129	0
130	0	130	0	130	0
131	0	131	0	131	0
132	0	132	0	132	0
133	0	133	0	133	0
134	0	134	0	134	0
135	0	135	0	135	0
136	0	136	0	136	0
137	0	137	0	137	0
138	0	138	0	138	0
139	0	139	0	139	0
140	0	140	0	140	0
141	0	141	0	141	0
142	0	142	0	142	0
143	0	143	0	143	0
144	0	144	0	144	0
145	0	145	0	145	0
146	0	146	0	146	0
147	0	147	0	147	0
148	0	148	0	148	0
149	0	149	0	149	0
150	0	150	0	150	0
151	0	151	0	151	0
152	0	152	0	152	0
153	0	153	0	153	0
154	0	154	0	154	0
155	0	155	0	155	0
156	0	156	0	156	0
157	0	157	0	157	0
158	0	158	0	158	0
159	0	159	0	159	0
160	0	160	0	160	0
161	0	161	0	161	0
162	0	162	0	162	0
163	0	163	0	163	0
164	0	164	0	164	0
165	0	165	0	165	0
166	0	166	0	166	0
167	0	167	0	167	0
168	0	168	0	168	0
169	0	169	0	169	0
170	0	170	0	170	0
171	0	171	0	171	0
172	0	172	0	172	0
173	0	173	0	173	0
174	0	174	0	174	0
175	0	175	0	175	0
176	0	176	0	176	0
177	0	177	0	177	0
178	0	178	0	178	0
179	0	179	0	179	0
180	0	180	0	180	0
181	0	181	0	181	0
182	0	182	0	182	0
183	0	183	0	183	0
184	0	184	0	184	0
185	0	185	0	185	0
186	0	186	0	186	0
187	0	187	0	187	0
188	0	188	0	188	0
189	0	189	0	189	0
190	0	190	0	190	0
191	0	191	0	191	0
192	0	192	0	192	0
193	0	193	0	193	0
194	0	194	0	194	0
195	0	195	0	195	0
196	0	196	0	196	0
197	0	197	0	197	0
198	0	198	0	198	0
199	0	199	0	199	0
200	0	200	0	200	0
201	0	201	0	201	0
202	0	202	0	202	0
203	0	203	0	203	0
204	0	204	0	204	0
205	0	205	0	205	0
206	0	206	0	206	0
207	0	207	0	207	0
208	0	208	0	208	0
209	0	209	0	209	0
210	0	210	0	210	0
211	0	211	0	211	0
212	0	212	0	212	0
213	0	213	0	213	0
214	0	214	0	214	0
215	0	215	0	215	0
216	0	216	0	216	0
217	0	217	0	217	0
218	0	218	0	218	0
219	0	219	0	219	0
220	0	220	0	220	0
221	0	221	0	221	0
222	0	222	0	222	0
223	0	223	0	223	0
224	0	224	0	224	0
225	0	225	0	225	0
226	0	226	0	226	0
227	0	227	0	227	0
228	0	228	0	228	0
229	0	229	0	229	0
230	0	230	0	230	0
231	0	231	0	231	0
232	0	232	0	232	0
233	0	233	0	233	0
234	0	234	0	234	0
235	0	235	0	235	0
236	0	236	0	236	0
237	0	237	0	237	0
238	0	238	0	238	0
239	0	239	0	239	0
240	0	240	0	240	0
241	0	241	0	241	0
242	0	242	0	242	0
243	0	243	0	243	0
244	0	244	0	244	0
245	0	245	0	245	0
246	0	246	0	246	0
247	0	247	0	247	0
248	0	248	0	248	0
249	0	249	0	249	0
250	0	250	0	250	0

JPEG Default AC Codes

JPEG

- The Joint Photographic Expert Group (JPEG), under both the International Standards Organization (ISO) and the International Telecommunications Union Telecommunication Sector (ITU-T)
 - www.jpeg.org
- Has published several standards
 - JPEG: lossy coding of continuous tone still images
 - Based on DCT
 - JPEG-LS: lossless and near lossless coding of continuous tone still images
 - Based on predictive coding and entropy coding
 - JPEG2000: scalable coding of continuous tone still images (from lossy to lossless)
 - Based on wavelet transform

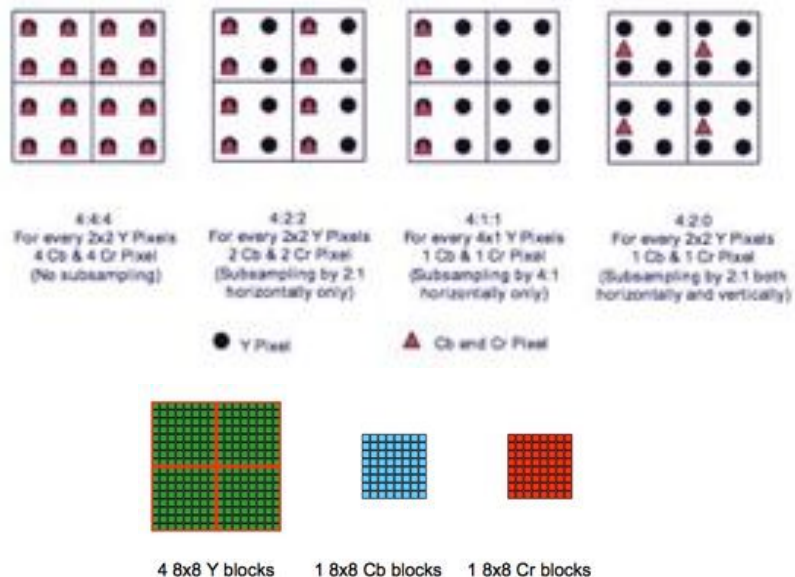
1992 JPEG

- Support several modes
 - Baseline system (what is commonly known as JPEG!): lossy
 - Can handle grayscale or color images, with 8bits per color component
 - Extended system:
 - can handle higher precision (12 bit) images, providing progressive streams, etc.
 - Lossless version
- Baseline version
 - Each color component is divided into 8x8 blocks
 - For each 8x8 block, three steps are involved:
 - **Block DCT**
 - **Perceptual-based quantization**
 - **Variable length coding: Run length and Huffman coding**

Coding Colored Images

- Color images are typically stored in (R,G,B) format
 - JPEG standard can be applied to each component separately
 - Does not make use of the correlation between color components
 - Does not make use of the lower sensitivity of the human eye to chrominance samples
- Alternate approach
 - Convert (R,G,B) representation to a YCbCr representation
 - Y: luminance, Cb, Cr: chrominance
 - Down-sample the two chrominance components
 - Because the peak response of the eye to the luminance component occurs at a higher frequency than to the chrominance components

Chrominance Subsampling



Quantization Tables for Y, Cr Cb

For luminance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	60	56
14	17	22	29	51	87	80	62
18	22	37	56	68	100	103	77
24	35	55	64	82	104	113	92
40	64	78	87	100	121	120	101
72	92	95	98	112	100	103	99

For chrominance

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

Summary

- The concept behind compression and transformation
- How to perform 2D DCT: forward and inverse transform
 - Manual calculation for small sizes, using inner product notation
 - Using Matlab: dct2, idct2
- Why DCT is good for image coding
 - Real transform, easier than DFT
 - Most high frequency coefficients are nearly zero and can be ignored
 - Different coefficients can be quantized with different accuracy based on human sensitivity
- How to quantize & code DCT coefficients
 - Varying step sizes for different DCT coefficients based on visual sensitivity to different frequencies; A quantization matrix specifies the default quantization stepsize for each coefficient; The matrix can be scaled using a user chosen parameter (QP) to obtain different trade-offs between quality and size
 - DC: prediction + huffman; AC: run-length + huffman

Next Lecture

- Speech/ Audio/Video Compression!