Lecture-7

Video Compression

Mubarak Shah

What is Compression?

- Compression is a process of converting data into a form requiring less space to store or less time to transmit, which permits the original data to be reconstructed with acceptable precision at a later time.

Orange Juice Analogy!

- Freshly squeezed orange juice (uncompressed)
- Remove water (redundancy), convert it to concentrate (encoding)
- Shipped, stored, and sold.
- Add water to concentrate (decoding), tastes like freshly squeezed!!!

Why is compression necessary?

- Storage space limitations
- Transmission bandwidth limitations.

Resolution

- QCIF: 180 x 144
- MPEG: 352 x 288
- VGA: 640 x 480
- NTSC 720x486
- Workstation 1280x1024
- HDTV: 1920 x 1080
- 35mm slide: 3072 x 2048

Floppy Disk

- Floppy disk capacity = 1.44 MB
- A single 1280x1024x24 image= 3.9 MB
- A single 640x480x24=922kB
- Floppy disk holds only one VGA image!

CD-ROM

- Capacity=600 MB
- A 1280x1024x24 @30 fps=118MB/s
- CD-ROM would hold only about 5 sec of video!
- A 160x120x16 image @30 fps=1.15MB/sec
- CD-ROM now holds 8.7 minutes of video

DVD-ROM

- Capacity 2.4 GB to 15.9 GB
- Single side/single layer→Double side/dual layers
- 4.4 to 25 times capacity of CD ROM
- 20 sec to 2 minutes of 1280x1024x24 @30 fps
- 3 hours of 160x120x16 image @30 fps

Bandwidth

- 160X120=1.15 MB/sec
- Quad-speed CD-ROM drive delivers 600 KB/sec (half of the required speed)
- DVD ROM delivers from 4Mbs to 9.8Mbs
- "T1" line delivers 1.54 Mb/sec (192KB/sec)
- Ethernet delivers 10Mb/sec (1.25 MB/sec) (barely fast enough, will use up entire bandwidth, 2-way video not possible)

Digital TV

- Networks started broadcasting limited DTV programs in Nov 98.
- All commercial stations are supposed to switch to DTV by 2002
- All stations are supposed to switch to DTV by 2003
- Govt wants broadcasters' NTSC channels returned by 2006 for auctioning!

Digital TV

- CBS and NBC use 1080i (1920X1080), which is 995Mb/s at 30 fps
- ABC and Fox use 720p (1280X720), which is 424Mb/s at 30 fps
- 6 MHz channel assigned to each network can carry 19.4Mb/s
- Need 50:1 compression ratio!

Why is compression acceptable?

- Limitations of visual perception
  - Number of shades (colors, gray levels) we can perceive
  - Reduced sensitivity to noise in high-frequencies (e.g. edges of objects)
  - Reduced sensitivity to noise in brighter areas
- Ability of visual perception
  - Ability of the eye to integrate spatially
  - Ability of the mind to interpolate temporally

Why is compression acceptable?

- Some type of visual information is less important than others
- Goal is to throw away bits in psycho-visually lossless manner
- We have been conditioned to accept imperfect reproduction
- Limitations of intended output devices

Why is compression possible?

- Some sample values (gray levels, colors) are more likely to occur at a particular pixel than others.
  - Remove spatial and temporal redundancy that exist in natural video
    - Correlation itself can be removed in a lossless fashion
    - Important to medical applications
    - Only realizes about 2:1 compression

Why is compression possible?

- No single algorithm can compress all possible data
- Random data cannot be compressed

Lossless Compression

- Needed when loss is unacceptable or highly undesirable
- Fixed compression ratio is hard to achieve
- Compression/decompression time varies with image

Lossy Compression

- Used when loss is acceptable or inevitable
- Permits fixed compression ratios
- Better suited for fixed time decompression

Compression Techniques

- Subsampling
- Quantization
- Delta Coding
- Prediction
- Color space conversion
- Huffman coding
- Run-length encoding
- De-correlation
- Motion Compensation
- Model-based compression

Subsampling

- Selecting one single value to represent several values in a part of the image.
  - For example, use top left corner of 2X2 block to represent the block
  - Compression ratio 75%

11	15	19	55
13	14	21	32
39	17	24	76
43	34	27	80

→

11	11	19	19
11	11	19	19
39	39	24	24
39	39	24	24

Subsampling

- A better way- averaging
- Compression ratio 75%

11	15	19	55
13	14	21	32
39	17	24	76
43	34	27	80

→

13	13	32	32
13	13	32	32
33	33	51	51
33	33	51	51

Quantization

- Mapping of a large range of possible sample values into a smaller range of values or codes.
- Fewer bits are required to encode the quantized sample.
- Examples
  - -Letter grades (A, B, C, D, F)
  - Rounding of person's age, height, or weight

Quantization

- Truncation and Rounding
- Quantized levels need not be evenly spaced
- Can be used for relative as well as absolute information
- Information is lost in quantiztion, but the error can be recovered

Truncation

- Discard lower-order bits
  – average error 1/2 LSB of target resolution
- Example

9	11	17	21		0	10	10	20
19	51	33	14	→	10	50	30	10
19	23	18	15		10	20	10	10
53	47	12	43		50	40	10	40

Rounding

- Add 5 and then truncate the result.
  - One more LSB participate than in truncation
  - average error 1/4 LSB

13	19	9	5		10	20	10	10
14	17	8	15		10	20	10	20
52	49	53	47	→	50	50	50	50
50	58	51	42		50	60	50	40

Delta Coding

- Code the difference between adjacent pixels.
- Since adjacent pixels are similar, the difference is normally small, and requires fewer bits to code.
- A typical pixel value requires 8 bits.
- The difference between any 8 bit pixels is in the range [-255,255], which needs 9 bits!

Delta Coding

- But most deltas will be small.
  - Smaller deltas can be assigned shorter codes
  - Smaller deltas can be ignored completely
  - smaller deltas can be quantized more finally for better quality
- Complementary delta values can share a code; e.g., +1 and -255 yield same result in 8 bit positive value.
- 9 bits are not required!

Encoding with quantization loss

- Encoder must calculate incorrect pixel value that the decoder will decode, and use that value in computing the next delta, to minimize the quantization loss.

Prediction

- Prediction further reduces delta values.
- In delta coding prediction is the last pixel
- Better prediction algorithm means better compression ratio.
- It can improve picture quality

Prediction

- Use left pixel (delta coding)
- Use linear interpolation (left+(left-previous))
- Use 2d interpolation (left+above-corner)

Color Spaces

- R, G, B
- Y, Cb, Cr
- Y, I, Q
- C, M, Y
- I, H, S
- Y, U, V

Luma & Chroma

$$Y = .3R + .6G + .1B$$

$$C_b = \frac{R - Y}{1.6} + .5$$

$$C_r = \frac{B - Y}{2} + .5$$

Y, I, Q

$$Y = .3R + .59G + .11B$$

$$I = .6R + .28G - .32B$$

$$Q = .21R - .52G + .31B$$

I=Red-Cyan
Q=magenta-green
Y=white-black

C, M, Y

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

Cyan, Magenta and Yellow: Primary colors of pigments.

Intensity, Hue and Saturation

$$I = R + G + B$$

$$S = 1 - 3\frac{\min(R,G,B)}{I}$$

$$h = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}$$

Saturation measures lack of whiteness in the color. Hue is proportional to the average wavelength of the color. (A "deep", "bright" "orange".) (245,110,20)

Y, U, V

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ -.169 & -.331 & .5 \\ .5 & -.419 & -.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.4002 \\ 1 & -.344 & -.714 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ Y \end{bmatrix}$$

Y represents the brightness of a pixel.
U, V represent how far blue and red are from white.

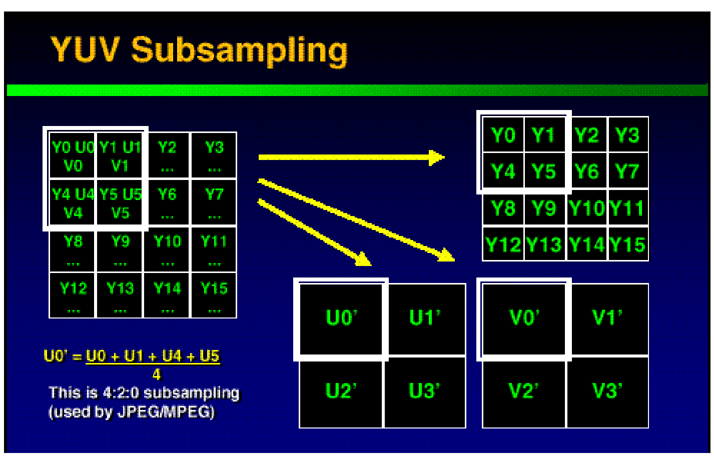
Average Delta Values for Adjacent Pixels

Y=13
U=1
V=1
YUV=13

R=13
G=13.2
B=12.7
RGB=13

We can sub-sample U & V over a number of pixels without loss of picture quality.

YUV Subsampling



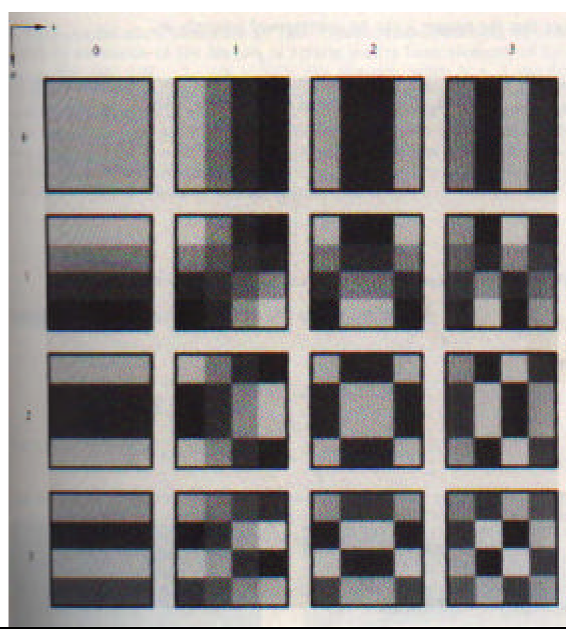
Discrete Cosine Transform

$$C(u,v) = \boldsymbol{a}(u)\boldsymbol{a}(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\boldsymbol{p}}{2N}\right]\cos\left[\frac{(2y+1)v\boldsymbol{p}}{2N}\right]$$

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \boldsymbol{a}(u)\boldsymbol{a}(v)C(u,v)\cos\left[\frac{(2x+1)u\boldsymbol{p}}{2N}\right]\cos\left[\frac{(2y+1)v\boldsymbol{p}}{2N}\right]$$

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1,2,\ldots N-1 \end{cases}$$

DCT Bases Functions



Example

$$I_{i,j} = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

image

Example

$$F_{u,v} = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & 1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

DCT

Other Techniques

- Fractals
- Wavelets
- Vector Quantization
- K-L Transform
- ...

Compression using original source

- For best compression, get the original source material and try to *understand* its properties.
  - Email messages are far smaller than fax, voice mail or video mail.
  - A musical score is far more compact than a digitized recording

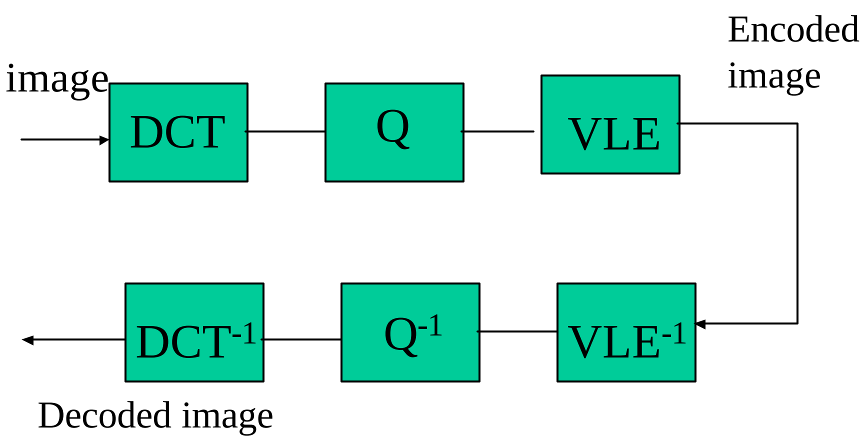
Compression of Synthesized Image or Video

- For synthesized image or video clip it is far more efficient to transmit original source material and re-synthesized the image or clip at the receiver than to transmit the compressed image or video clip.

How to Select Compression Scheme?

- High quality reproduction?
- Very high compression ratio?
- Fixed compression ratio?
- Real-time compression?
- Real-time decompression?
- Limited de-compression computer power?

JPEG BLOCK DIAGRAM



RLE: Example

8
0, 4, 4
1, 2, 5
1, 5, 2
1, 3, 2, 1, 1
2, 1, 2, 2, 1
0, 4, 1, 1, 2
8

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

JPEG Baseline Coding

- Divide image into blocks of size 8X8.
- Level shift all 64 pixels values in each block by subtracting 2^{n-1}, (where 2^n is the maximum number of gray levels).
- Compute 2D DCT of a block.
- Quantize DCT coefficients using quantization table.

JPEG Baseline Coding

- Zig-zag scan the quantized DCT coefficients to form 1-D sequence.
- Code 1-D sequence (AC and DC) using JPEG Huffman variable length codes.

JPEG ZIG-ZAG SCAN

$$
\begin{bmatrix}
1 & 2 & 6 & 7 & 15 & 16 & 28 & 29 \\
3 & 5 & 8 & 14 & 17 & 27 & 30 & 43 \\
4 & 9 & 13 & 18 & 26 & 31 & 42 & 44 \\
10 & 12 & 19 & 25 & 32 & 41 & 45 & 54 \\
11 & 20 & 24 & 33 & 40 & 46 & 53 & 55 \\
21 & 23 & 34 & 39 & 47 & 52 & 56 & 61 \\
22 & 35 & 38 & 48 & 51 & 57 & 60 & 62 \\
36 & 37 & 49 & 50 & 58 & 59 & 63 & 64
\end{bmatrix}
$$

JPEG Coefficient Coding Categories

Range	DC	AC
0	0	N/A
-1,1	1	1
-3,-2,2,3	2	2
-7,..,-4,4,..,7	3	3
-15,..,-8,8,..,15	4	4
….	..	..
-32767,..,32767	F	N/A

JPEG DC Code

Cat	Base Code	Length
0	010	3
1	011	4
2	100	5
3	00	5
4	101	7
5	110	8
6	1110	10
7	11110	12
8	111110	14
9	1111110	16
A	11111110	18
B	111111110	20

JPEG AC Code

Run/Cat	Base Code	Length
(0,0)	1010(EOB)	4
(0,1)	00	3
(0,2)	01	4
(0,3)	100	6
(0,4)	1011	8
(0,5)	11010	10
(0,6)	111000	12
(0,7)	1111000	14
(0,8)	1111110110	18
(0,9)	1111111110000010	25
…	…	...

Construction of JPEG Code

- Compute difference between the current DC coefficient and that of previously encoded block.
- Determine DC category of difference, and use the base code.
- Generate remaining bits of code from the LSB (Least Significant Bits) of the difference.

Example (Encoding)

$$I = \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 66 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

$$I' = \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -62 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -65 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

$$DCT = \begin{bmatrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ -11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{bmatrix}$$

$$Q' = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Example (Encoding)

1-D coefficeints
[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5
0 -1 2 0 0 0 0 0 -1 -1 EOB]

Coded array
1010110 0100 001 0100 0101 100001
0110 100011 001 100011 001 001
100101 11100110 110110 0110 11110100
000 1010 92 bits, 512/92 =5.6:1

Determining Code (DC)

- The DC DCT is "-26".
- The DC DCT of previous block was "-17".
- The difference is: -26-(-17)=-9
- DC category for "-9" is "4", with base code "101", and code length "7 bits".
- The difference (-9)=$(0110)_2$.
- The code for (-26) is 1010110.

Determining Code (AC)

- “-3” is AC category 2, preceded by “0” zeros
- Base code for 0/2 is “01”, length is “4” bits
- Two LSB of (-3)=$(100)_2$ are “00”
- The code of “-3” is “0100”

Example (Decoding)

$$P = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P' = \begin{bmatrix} -416 & -33 & -60 & 32 & 48 & 0 & 0 & 0 \\ 12 & -24 & -56 & 0 & 0 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P'' = \begin{bmatrix} -70 & -64 & -61 & -64 & -69 & -66 & -58 & -50 \\ -72 & -73 & -61 & -39 & -30 & -40 & -54 & -59 \\ -68 & -78 & -58 & -9 & 13 & -12 & -48 & -64 \\ -59 & -77 & -57 & 0 & 22 & -13 & -51 & -60 \\ -52 & -71 & -72 & -54 & -54 & -71 & -71 & -54 \\ -42 & -50 & -70 & -68 & -67 & -67 & -61 & -50 \\ -45 & -59 & -70 & -68 & -67 & -67 & -61 & -50 \\ -35 & 47 & -61 & -66 & -60 & -48 & -44 & -44 \end{bmatrix}$$

$$P''' = \begin{bmatrix} 58 & 64 & 67 & 64 & 59 & 62 & 70 & 78 \\ 56 & 55 & 67 & 89 & 98 & 88 & 74 & 69 \\ 60 & 50 & 70 & 119 & 141 & 116 & 80 & 64 \\ 69 & 51 & 71 & 128 & 149 & 115 & 77 & 68 \\ 74 & 53 & 64 & 105 & 115 & 84 & 65 & 72 \\ 76 & 57 & 56 & 74 & 75 & 57 & 57 & 74 \\ 83 & 69 & 59 & 60 & 61 & 61 & 67 & 78 \\ 93 & 81 & 67 & 62 & 69 & 80 & 84 & 84 \end{bmatrix}$$

Comparison

$$I = \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 66 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

$$P'' = \begin{bmatrix} 58 & 64 & 67 & 64 & 59 & 62 & 70 & 78 \\ 56 & 55 & 67 & 89 & 98 & 88 & 74 & 69 \\ 60 & 50 & 70 & 119 & 141 & 116 & 80 & 64 \\ 69 & 51 & 71 & 128 & 149 & 115 & 77 & 68 \\ 74 & 53 & 64 & 105 & 115 & 84 & 65 & 72 \\ 76 & 57 & 56 & 74 & 75 & 57 & 57 & 74 \\ 83 & 69 & 59 & 60 & 61 & 61 & 67 & 78 \\ 93 & 81 & 67 & 62 & 69 & 80 & 84 & 84 \end{bmatrix}$$

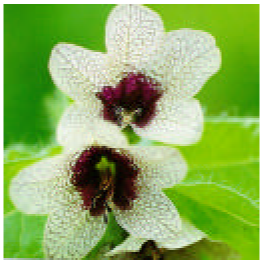
Original Image

Decoded Image

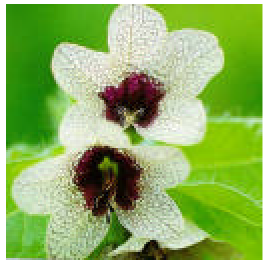
Difference

$$Diff = \begin{bmatrix} -6 & -9 & -6 & 2 & 11 & -1 & -6 & -5 \\ 7 & 4 & -1 & 1 & 11 & -3 & -5 & 3 \\ 2 & 9 & -2 & -6 & -3 & -12 & -14 & 9 \\ -6 & 7 & 0 & -4 & -5 & -9 & -7 & 1 \\ -7 & 8 & 4 & -1 & 11 & 4 & 3 & 2 \\ 3 & 8 & 4 & -4 & 2 & 11 & 1 & 1 \\ 2 & 2 & 5 & -1 & -6 & 0 & -2 & 5 \\ -6 & -2 & 2 & 6 & -4 & -4 & -6 & 10 \end{bmatrix}$$

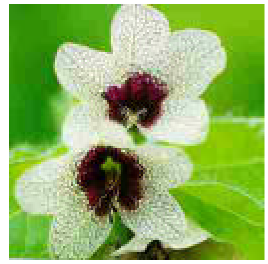
JPEG



Original 64K

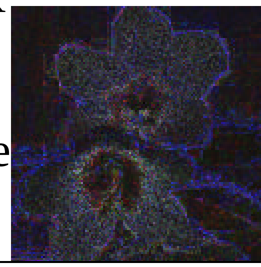


13K



5K

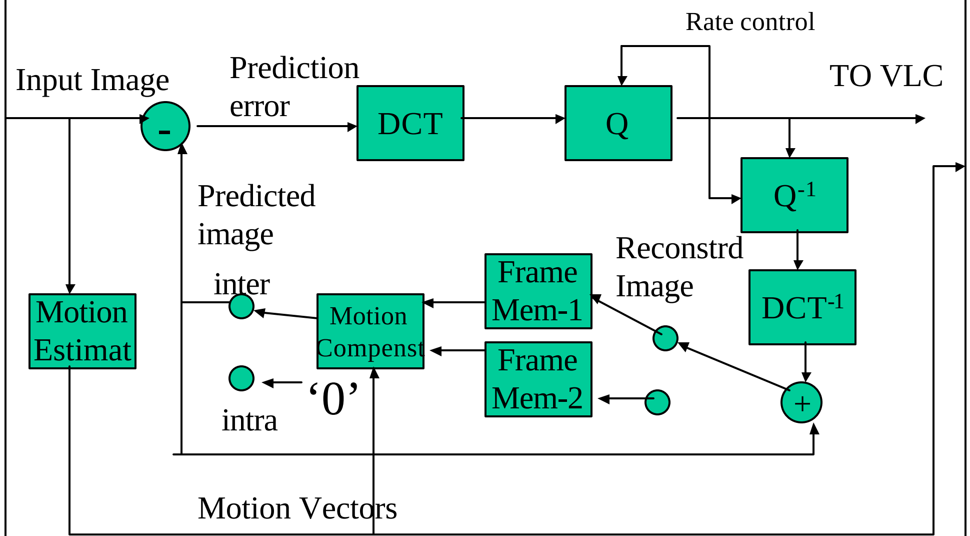
Difference



Video Compression Standards

- H.261
- H.263
- MPEG-1
- MPEG-2
- MPEG-4
- MPEG-7 (Multimedia Content Description Interface)

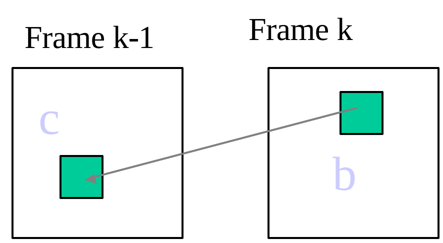
MPEG-1 Encoder



Motion Prediction

$$b' = c'$$

$$Error = b - b'$$



MPEG-1 & MPEG -2 Artifacts

- Blockiness
  - poor motion estimation
  - seen during dissolves and fades
- Mosquito Noises
  - edges of objects (high frequency DCT terms)
- Dirty Window
  - streaks or noise remain stationary while objects move

MPEG-1 & MPEG -2 Artifacts

- Wavy Noise
  - seen during pans across crowds
  - coarsely quantized high frequency terms cause errors

Where MPEG-2 will fail?

- Motions which are not translation
  - zooms
  - rotations
  - non-rigid (smoke)
  - dissolves
- Others
  - shadows
  - scene cuts
  - changes in brightness

Video Compression At Low Bitrate

- The quality of block-based coding video (MPEG-1 & MPEG-2) at low bitrate, e.g., 10 kbps is very poor.
  - Decompressed images suffer from blockiness artifacts
  - Block matching does not account for rotation, scaling and shear

Model-Based Video Coding

Model-Based Compression

- Object-based
- Knowledge-based
- Semantic-based

Model-Based Compression

- Analysis
- Synthesis
- Coding

Video Compression

- MC/DCT (MPEG-1 & 2)
  - Source Model: translation motion only
  - Encoded Information: Motion vectors and color of blocks
- Object-Based
  - Source Model: moving unknown objects
    - translation only
    - affine
    - affine with triangular mesh
  - Encoded Information: Shape, motion, color of each moving object

Video Compression

- Knowledge-Based
  - Source Model: Moving known objects
  - Encoded Information: Shape, motion and color of known objects
- Semantic
  - Source Model: Facial Expressions
  - Encoded Information: Action units