

ĐA PHƯƠNG TIỆN

Click to add text

Người trình bày: TS. Vương Hoàng Nam

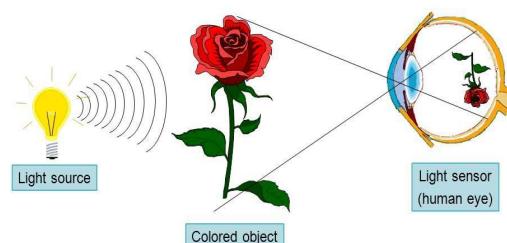
Mở cơ hội học tập cho mọi người

PHẦN 1. LÝ THUYẾT CƠ SỞ NÉN ẢNH

I.1 Sự cảm thụ màu sắc của người

Khi một người quan sát và cảm nhận màu sắc của một vật thể (đối tượng), sẽ có 3 yếu tố cơ bản sau:

- **Nguồn ánh sáng**
- **Vật thể được quan sát**
- **Mắt người quan sát**



Mở cơ hội học tập cho mọi người

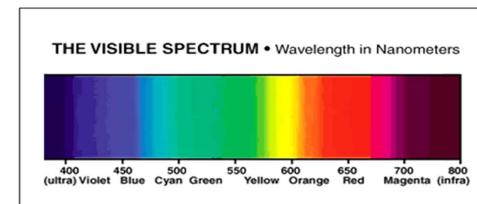
I.1 Sự cảm thụ màu sắc của người

- *Nguồn ánh sáng*

Là các ánh sáng mà mắt con người cảm nhận được. Đó là các sóng điện từ có bước sóng từ 380 đến 780 nm (nanometre, 10^{-9} m).

Nguồn ánh sáng có thể là tự nhiên (ví dụ: mặt trời) hoặc nhân tạo (ví dụ:bóng đèn).

Ánh sáng thực chất là chùm hạt photon. Các photon này có tính chất vật lý (kích thước, khối lượng...) giống nhau nhưng lại mang các năng lượng khác nhau.



I.1 Sự cảm thụ màu sắc của người

- Nguồn ánh sáng

Năng lượng của một photon được xác định bởi: $E = \frac{hc}{\lambda}$

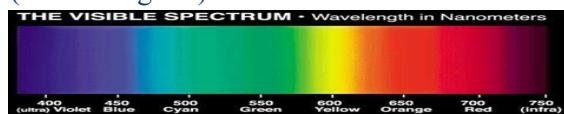
E: năng lượng của photon

h: hằng số Planck

c: vận tốc ánh sáng

λ : bước sóng của photon

Các photon mang năng lượng khác nhau sẽ tương ứng với bước sóng (tần số) khác nhau. Photon có năng lượng càng lớn tương ứng với bước sóng càng nhỏ (tần số càng lớn).

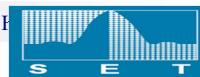
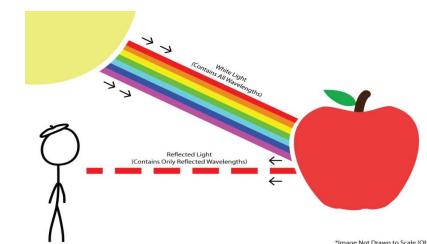
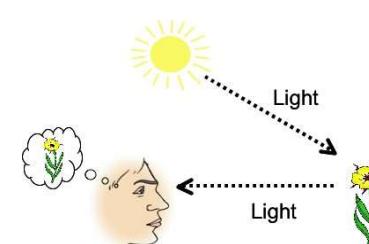


Các photon tương ứng các bước sóng khác nhau sẽ được mắt người cảm thụ thành các màu sắc khác nhau

I.1 Sự cảm thụ màu sắc của người

- Vật thể (đối tượng) được quan sát

Ánh sáng từ nguồn sáng được chiếu tới vật thể được quan sát. Phụ thuộc vào tính chất của vật thể (như hình dạng, kết cấu, chất liệu ...) mà vật thể này có thể hấp thụ một số bước sóng và phản xạ các bước sóng còn lại. Chính nhờ ánh sáng phản xạ trên vật thể đi tới mắt chúng ta mà ta có thể cảm thụ về màu sắc và hình dạng của vật thể được quan sát.



I.1 Sự cảm thụ màu sắc của người

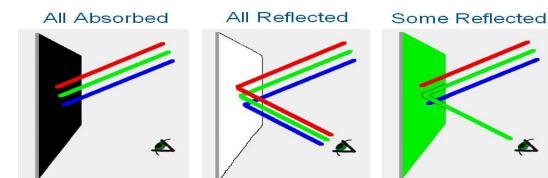
- Vật thể quan sát

Màu trắng: phản xạ và hấp thụ những bước sóng ánh sáng nào?

Trả lời: Vật có màu trắng sẽ phản xạ tất cả các bước sóng

Màu đen: phản xạ và hấp thụ những bước sóng ánh sáng nào?

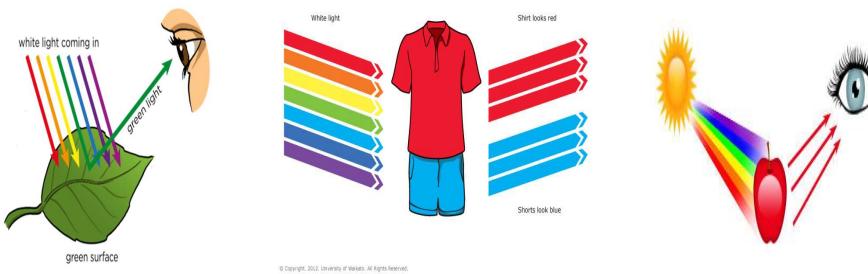
Trả lời: Vật có màu đen sẽ hấp thụ tất cả các bước sóng



I.1 Sự cảm thụ màu sắc của người

- Vật thể quan sát

Một số ví dụ về màu sắc chúng ta quan sát được



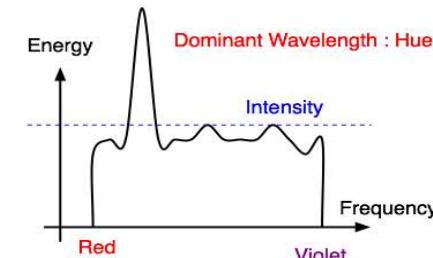
I.2 Đặc trưng của màu sắc

- Màu sắc quang học

Một màu sắc (ánh sáng) quang học bất kỳ có thể biểu diễn trong một mặt phẳng tọa độ 2 chiều.

Trục hoành là trục biểu diễn tần số (hoặc bước sóng)

Trục tung là trục biểu diễn năng lượng ánh sáng (cường độ photon)



Ánh sáng quang học trên có màu gì? Trả lời: **Màu đỏ (hoặc cam)**

I.2 Đặc trưng của màu sắc

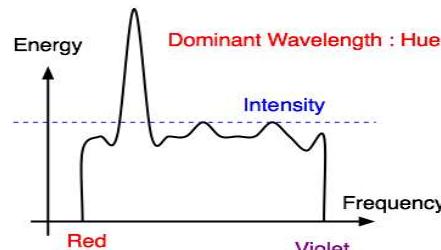
- Màu sắc quang học

Màu sắc (colour) có hai đặc trưng cơ bản: *Sắc màu* (Hue) và *độ sáng* (Brightness).

Sắc màu phụ thuộc các bước sóng trội (dominant), là các bước sóng mang năng lượng vượt trội hơn mức năng lượng trung bình.

Hình vẽ: các bước sóng chính chủ yếu ở vùng phổ màu đỏ.

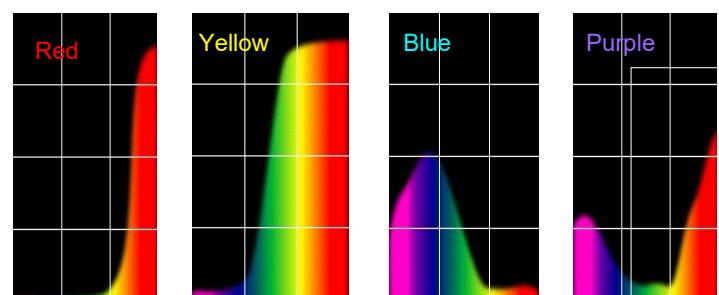
Độ sáng phụ thuộc giá trị năng lượng (càng lớn càng sáng)



I.2 Đặc trưng của màu sắc

- Màu sắc quang học

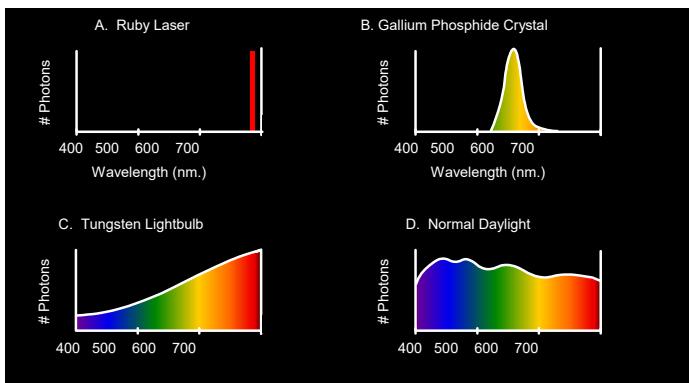
Một số ví dụ về phổ quang học một số màu trong tự nhiên: đỏ (Red), vàng (Yellow), lam (Blue), tím (Purple)



I.2 Đặc trưng của màu sắc

- Màu sắc quang học

Một số ví dụ về phổ quang học của một số nguồn ánh sáng: A-Laser (Ruby), B-Led (Gallium), C-Sợi đốt (Tungsten) và D-Ánh sáng ban ngày
Ánh sáng ban ngày còn gọi là ánh sáng trắng.



Learning Opportunity for All

13

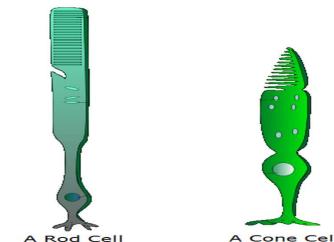
I.3 Đặc trưng mắt người

- Mắt người quan sát

Sở dĩ mắt chúng ta cảm nhận các hình ảnh, màu sắc quang học do trong mắt chúng ta có rất nhiều tế bào nhạy (cảm thụ) ánh sáng.

Có hai loại tế bào cảm thụ ánh sáng: tế bào hình que (rod cell) và tế bào hình nón (cone cell).

Mắt người bình thường có khoảng 15 triệu cell hình que và 5 triệu cell hình nón



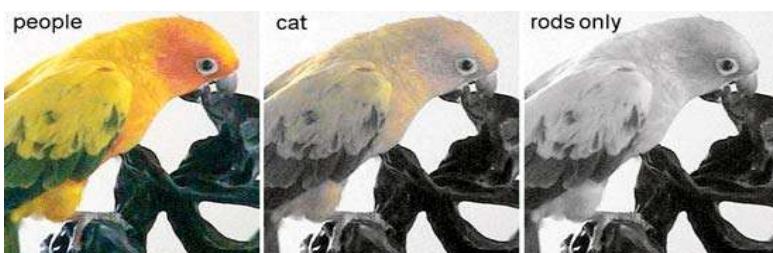
Learning Opportunity for All

14

- Mắt người quan sát

Các cell hình que cho cảm nhận về độ sáng trong khi các cell hình nón cho phép cảm nhận về màu sắc.

Thông thường mắt người có nhiều hơn các cell hình nón và thường cảm nhận màu sắc tốt hơn nhiều so với các loài động vật khác.



Learning Opportunity for All

15

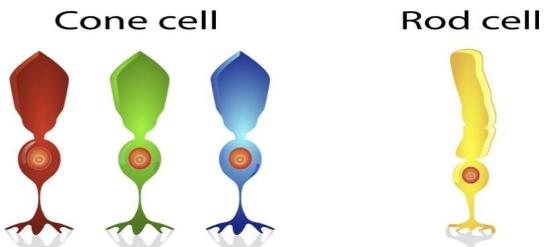
- Mắt người quan sát

Có ba loại cell hình nón được ký hiệu S, M, L (Short, Medium, Long wavelength). Các loại cell này nhạy cảm với các vùng phổ màu sắc khác nhau.

S nhạy với các bước sóng ngắn vùng phổ lam (Blue-cone)

M nhạy với các bước sóng trung bình vùng phổ lục (Green-cone)

L nhạy với các bước sóng dài vùng phổ đỏ (Red-cone)



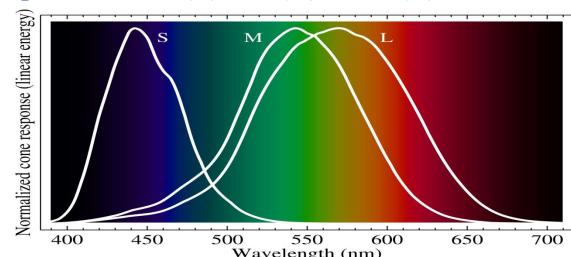
Learning Opportunity for All

16

- Mắt người quan sát

Hình vẽ minh họa độ nhạy (tương đối) của các cell hình nón loại S, M và L theo phổ bước sóng ánh sáng.

Về cơ bản, ta có thể thấy mắt người có khả năng cảm nhận tất cả các bước sóng (màu sắc) trong vùng phổ ánh sáng nhìn thấy và đặc biệt nhạy cảm với các phổ màu lam (B), lục (G) và đỏ (R).



Learning Opportunity for All

17

- Mắt người quan sát

Tuy nhiên trên thực tế do một số khuyết tật của mắt, con người không có khả năng cảm nhận được một số màu sắc nào đó (hiện tượng mù màu sắc - colour blindness). Hình vẽ minh họa hiện tượng mù màu (mù đỏ, mù lục, mù lam).



Learning Opportunity for All

18

I.4 Lý thuyết ba màu

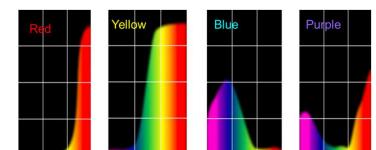
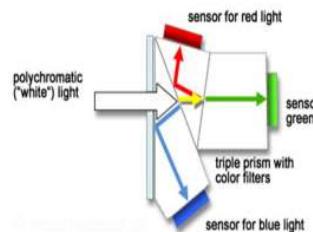
$$S = aR + bG + cB$$

Mọi màu sắc trong tự nhiên đều có thể tạo lại từ tổ hợp tuyến tính 3 màu cơ bản R,G,B

Learning Opportunity for All

19

I.3 Lý thuyết ba màu



Màu (R,G,B). R,G,B là các điện áp tỷ lệ với năng lượng ánh sáng (photon) của 3 vùng phổ R,G,B.

$$S = aR + bG + cB$$

Tỷ lệ a:b:c quyết định sắc màu (Hue)

Độ lớn a,b,c quyết định độ sáng (Brightness)

Learning Opportunity for All

20

I.3 Lý thuyết ba màu

$$S = aR + bG + cB$$

Mọi màu sắc trong tự nhiên đều có thể tạo lại từ tổ hợp tuyến tính 3 màu cơ bản R,G,B

Red: 700nm Green: 546nm Blue: 435nm

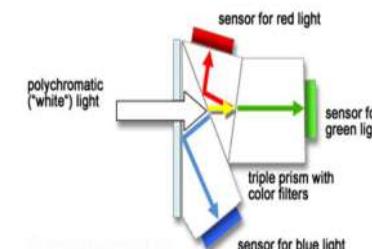
$$S_1 = 0,01 R + 0,2 G$$

$$S_2 = 0,1 R + 0,2 G$$

$$S_3 = 0,2 R + 0,3 G$$

$$S_4 = 0,2 R + 0,4 G$$

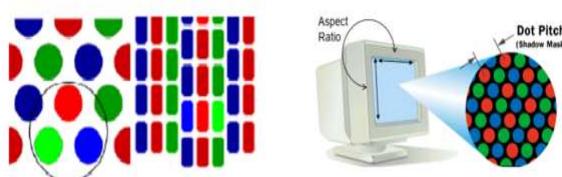
Màu S_1, S_2, S_3, S_4 có màu ntn?
Hai màu S_2, S_4 có điểm gì giống nhau



Màu (R,G,B). R,G,B là các điện áp tỷ lệ với năng lượng ánh sáng (photon) của 3 vùng phổ R,G,B.

Tỷ lệ a:b:c quyết định sắc màu (Hue)

Độ lớn a,b,c quyết định độ sáng (Brightness)

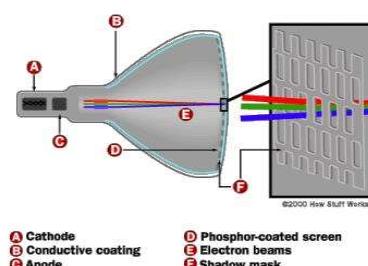


Mỗi pixel gồm 3 sub-pixel màu R,G,B.

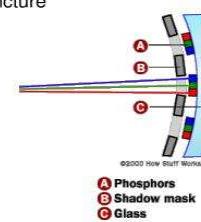
Màu trong tự nhiên thường có bao nhiêu bước sóng?

Màu trên màn hình có bao nhiêu bước sóng?

How a Color CRT TV Works



- Beam of electrons are sent from the cathode/anode setup
- Directed beam sweeps the screen many times per second
- The eye sees it as continually changing picture

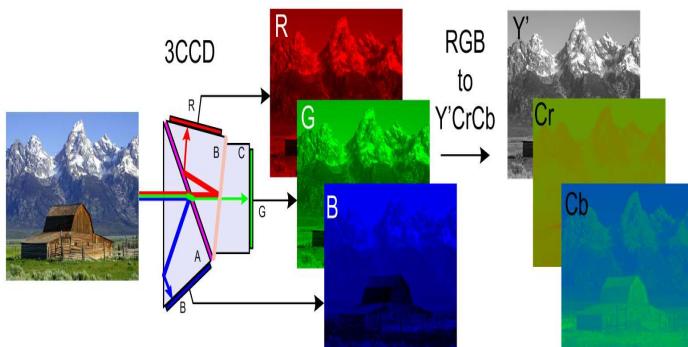
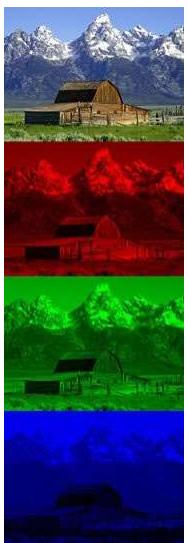


- Each "dot" on the screen one of three types of phosphor – chemicals that emit red, green, or blue light when hit with an electron
- These combine to make any possible color

$$Y=0,3R+0,59G+0,11B$$

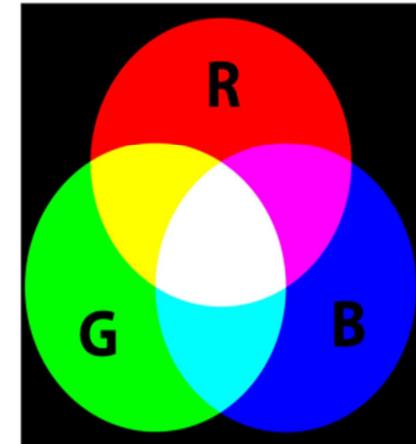
$$Cb=B-Y$$

$$Cr=R-Y$$



Learning Opportunity for
All

25

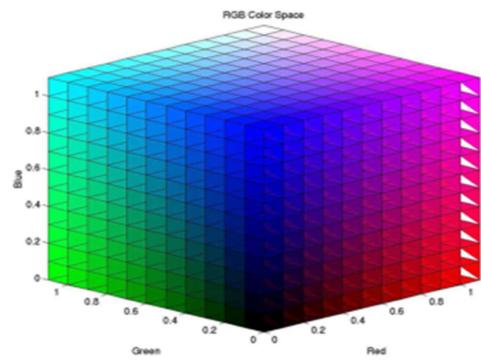


Black (0,0,0) White(1,1,1) Red (1,0,0) Green (0,1,0)
Blue (0,0,1) Yellow (1,1,0) Cyan (0,1,1) Magneta (1,0,1)

Learning Opportunity for
All

26

Không gian màu RGB

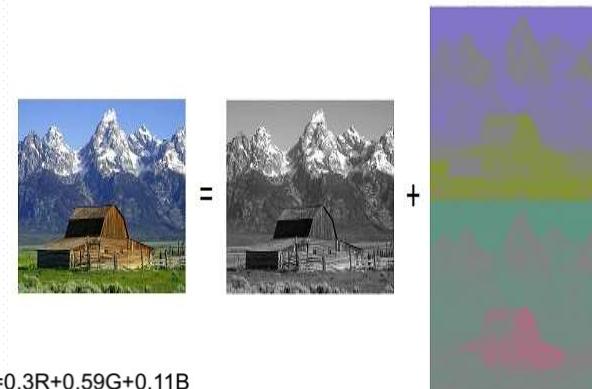


Black (0,0,0) White (255,255,255)
 $256^3 = 16$ triệu màu

Learning Opportunity for
All

27

KHÔNG GIAN MÀU YCbCr



$$Y=0,3R+0,59G+0,11B$$

$$Cb=B-Y$$

$$Cr=R-Y$$

Learning Opportunity for
All

28



$$Y=0,3R+0,59G+0,11B$$

$$Cb=B-Y$$

$$Cr=R-Y$$



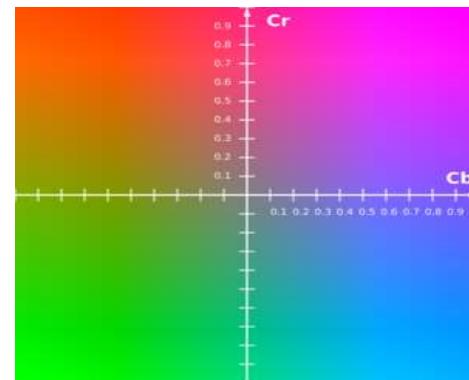
YCbCr

Tính (Y,Cr,Cb) nếu (R,G,B) bằng:

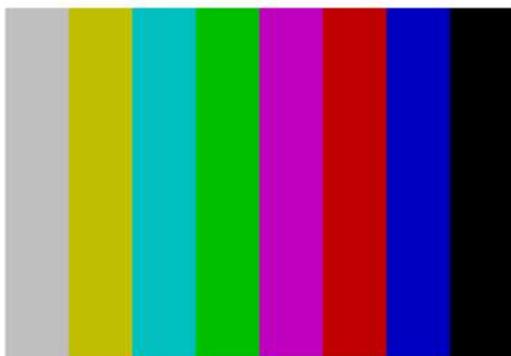
a/ (1,1,1)

b/(1,0,0)

c/(0,0,1)



KHÔNG GIAN MÀU YCbCr



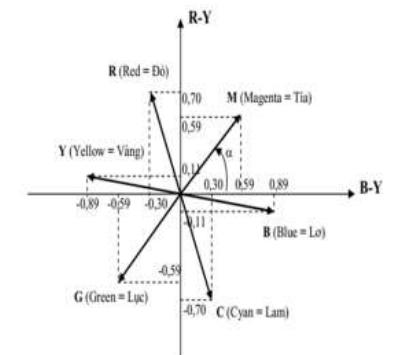
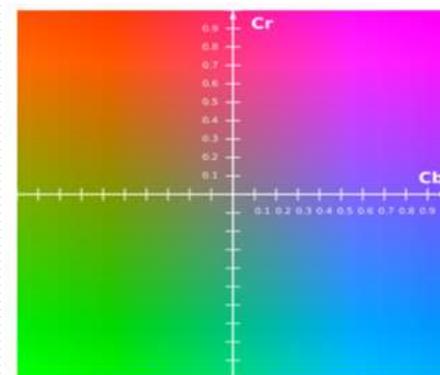
$$Y=0,3R+0,59G+0,11B$$

$$Cb=B-Y$$

$$Cr=R-Y$$



YCbCr



$$Y=0,3R+0,59G+0,11B$$

$$Cb=B-Y$$

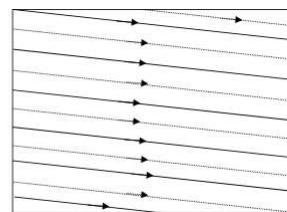
$$Cr=R-Y$$

NTSC Video

- 525 scan lines per frame, 30 fps (33.37 msec/frame).
- Interlaced, each frame is divided into 2 fields, 262.5 lines/field
- 20 lines reserved for control information at the beginning of each field
- So a maximum of 485 lines of visible data
 - Laserdisc and S-VHS have actual resolution of ~420 lines
 - Ordinary TV -- ~320 lines
 - Each line takes 63.5 microseconds to scan.
- Color representation:
 - Uses YIQ color model.

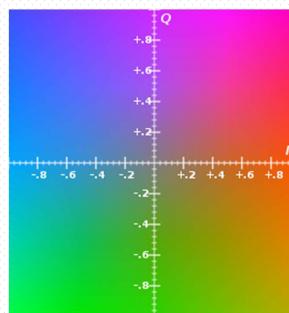
PAL (SECAM) Video

- 625 scan lines per frame, 25 frames per second (40 msec/frame)
- Interlaced, each frame is divided into 2 fields, 312.5 lines/field
- Color representation:
 - Uses YUV color model



Learning Opportunity for All

33



RGB to YIQ

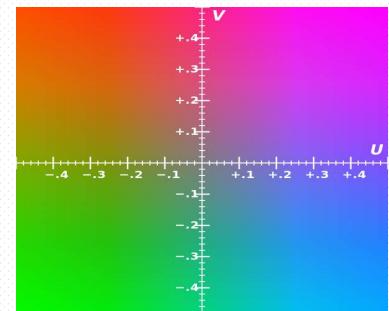
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

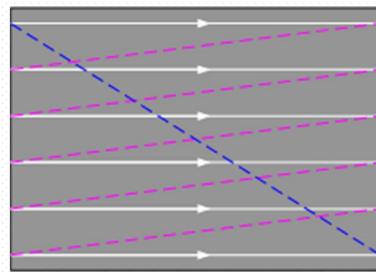
Learning Opportunity for All

34



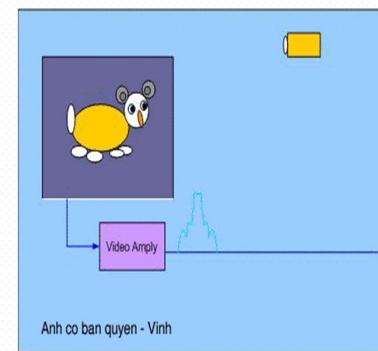
$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ U \\ V \end{bmatrix}.$$

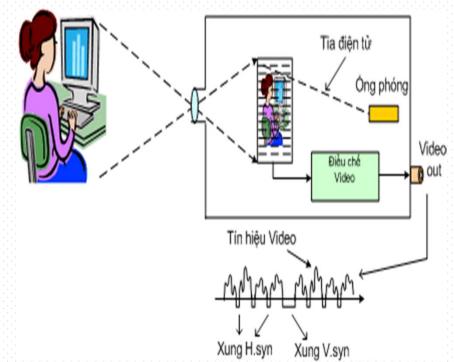
Nguyên lý quét hình

Learning Opportunity for All

35

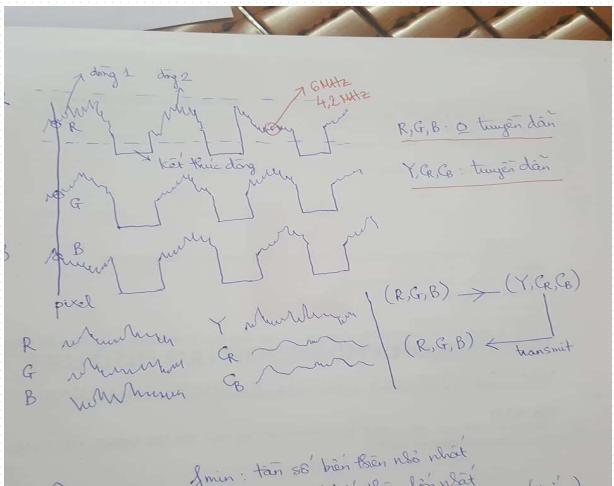


Learning Opportunity for All



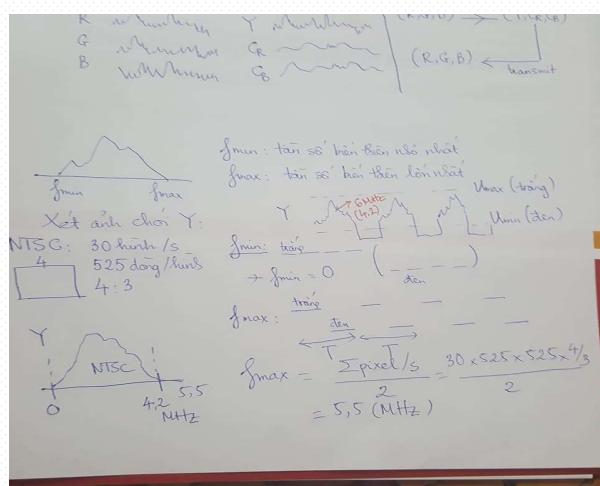
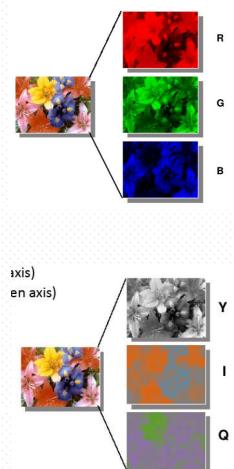
36

Phô tín hiệu Video



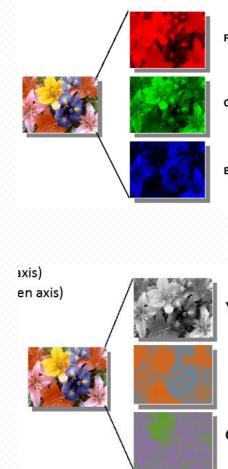
Learning Opportunity for All

37

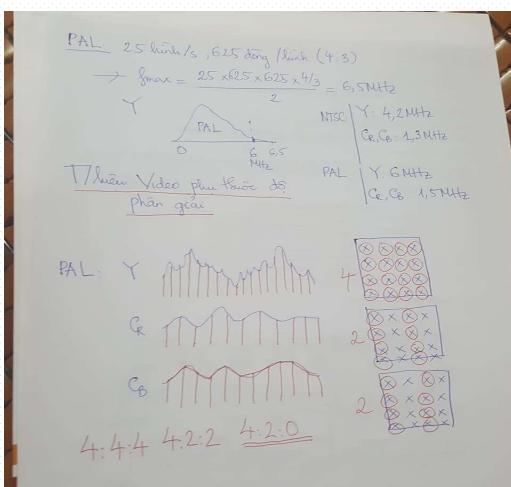


Learning Opportunity for All

38

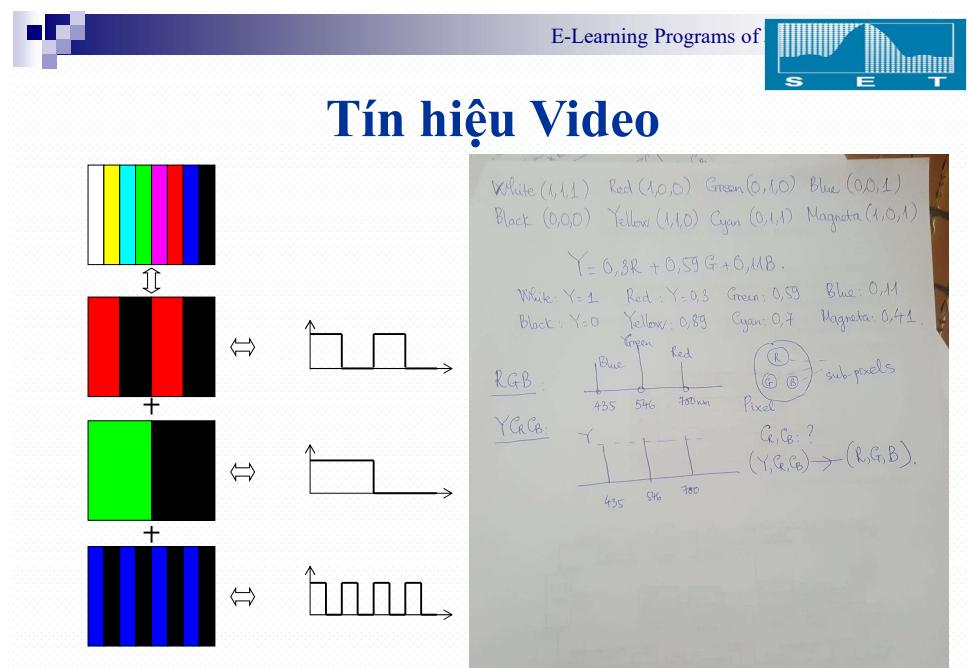
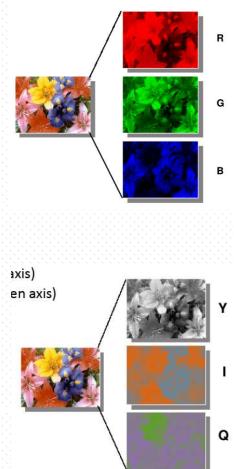


Phô tín hiệu Video



Learning Opportunity for All

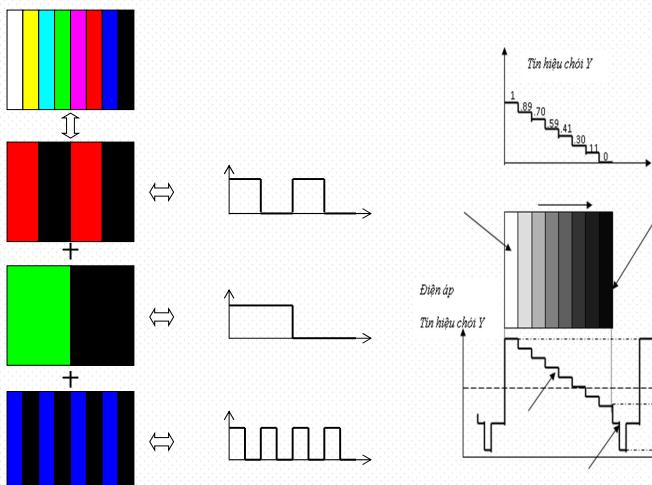
39



Learning Opportunity for All

40

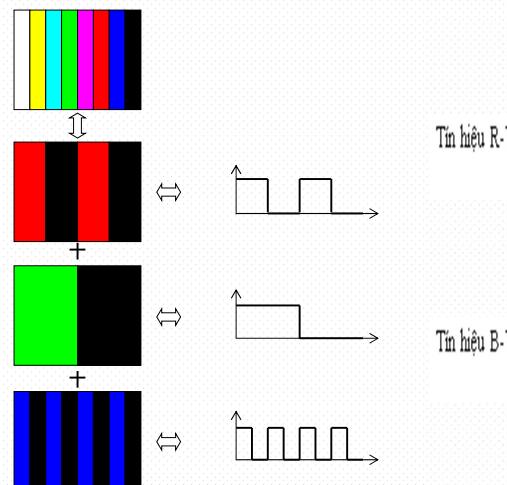
Tín hiệu Video



Learning Opportunity for All

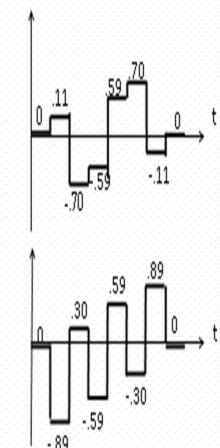
41

Tín hiệu Video



Tín hiệu R-Y

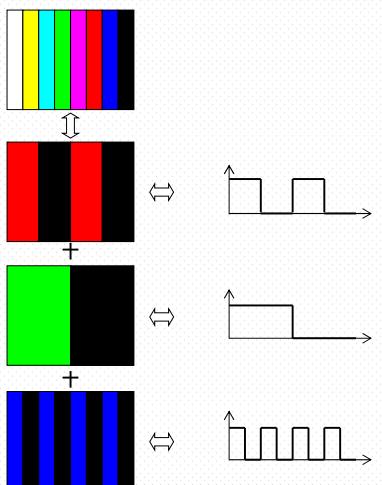
Tín hiệu B-Y



Learning Opportunity for All

42

Tín hiệu Video



Learning Opportunity for All

43

Vẽ dạng tín hiệu YCrCb



44

I.4 Mục đích nén ảnh

■ Khái niệm nén (Compression)

- Định nghĩa
- Mã hóa nguồn
- Tỷ số nén

Mục đích nén ảnh (tiếp)

■ Khái niệm nén (Compression)

Nguyên tắc chính của các kỹ thuật nén hiện nay là giảm thông tin dư thừa và không cần thiết trong các tín hiệu nhưng vẫn đảm bảo được chất lượng của tín hiệu (hình ảnh, âm thanh...).

Mục đích nén ảnh (tiếp)

■ Khái niệm nén (Compression)

- Khái niệm nén được đề cập là phương pháp mã hóa nguồn. Mục đích chủ yếu của nó là làm giảm kích thước dữ liệu.



Mục đích nén ảnh (tiếp)

- Tỷ số nén: Được xác định bằng tỷ số kích thước của dữ liệu trước nén và sau nén.
- $1600 \times 1200 \times 3\text{Byte (24bit)} = 5,7\text{MB}$

1600×1200



5,7MB

Né
n

1600×1200



406KB

Nén ảnh



49

Mục đích nén ảnh (tiếp)

- Tỷ số nén: $5,7\text{MB}/406\text{KB}=12,4$ (lần)
- Tỷ số nén càng cao thì kích thước sau nén sẽ càng nhỏ nhưng đồng thời (thông thường) chất lượng tín hiệu cũng giảm đi.

Một số chuẩn nén ảnh tĩnh

- **JPEG** Joint Photographic Experts Group
- **GIF** Graphics Interchange Format
- **PNG** Portable Network Graphics

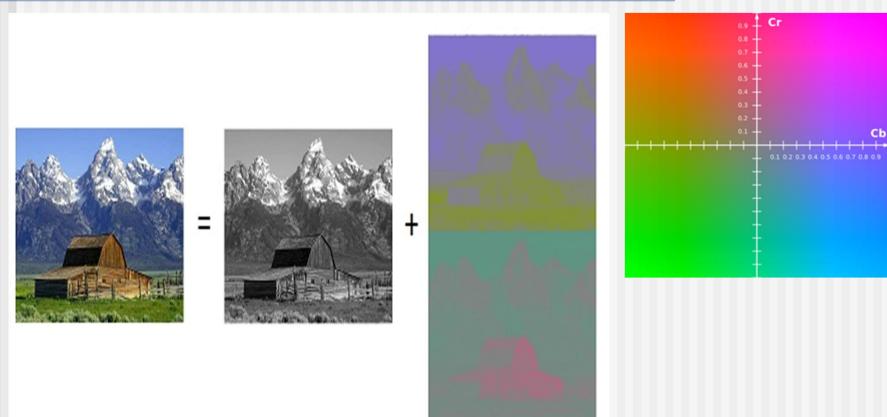
Đây là ba loại định dạng ảnh số phổ biến nhất.

Câu hỏi:

So sánh các định dạng ảnh JPEG, GIF và PNG?

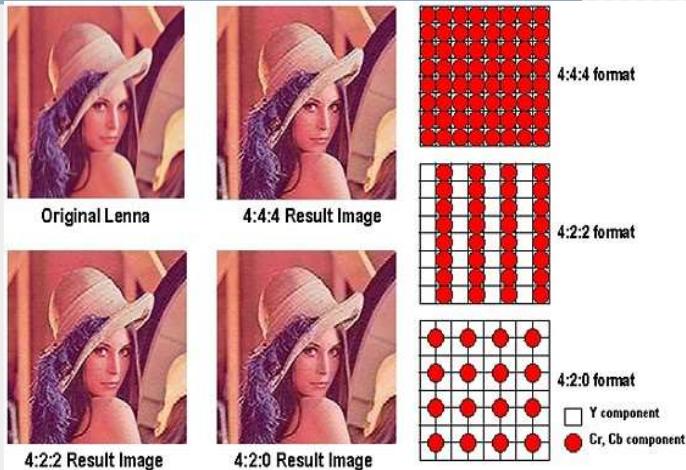
51

Chuẩn lấy mẫu



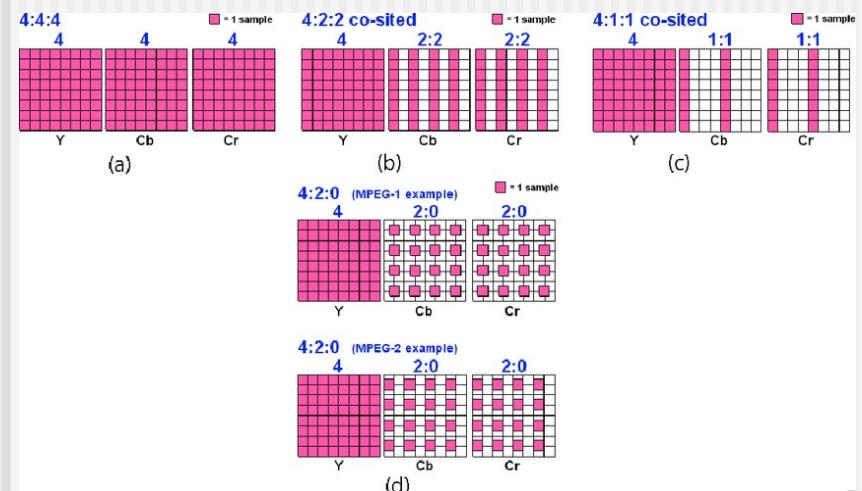
52

Chuẩn lấy mẫu 4:4:4, 4:2:2, 4:2:0



53

MPEG



54

Chuẩn lấy mẫu 4:4:4, 4:2:2, 4:2:0

■ BT1:

Cho một ảnh màu kích thước 288x352 pixel, lấy mẫu 4:2:2, mỗi điểm ảnh cần 8 bit biểu diễn

a/ Tính dung lượng của ảnh

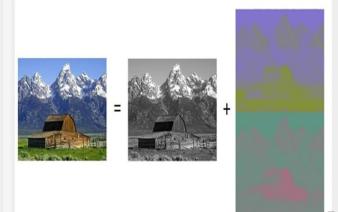
b/ Nén JPEG cho ảnh trên, biết dung lượng ảnh sau nén là 50KB. Tính tỷ số nén của ảnh?

Kích thước ảnh chưa mã hóa:

$$288 \times 352 \times 8\text{bit} \times (1+0,5+0,5) = 1.622 \text{ Kbit} = 198 \text{ KB}$$

Tỷ số nén = 3,96 (lần)

1 KB=1024 Byte



55

Chuẩn lấy mẫu 4:4:4, 4:2:2, 4:2:0

■ BT2:

Cho một ảnh màu kích thước 288x352 pixel, lấy mẫu 4:2:2, giá trị mỗi điểm ảnh cần 8bit biểu diễn.

Mã hóa JPEG cho ảnh trên, biết tỷ số nén cho ảnh chói Y là 10 lần, tỷ số nén cho các tín hiệu màu Cb,Cr là 20 lần.

Tính tỷ số nén của ảnh?

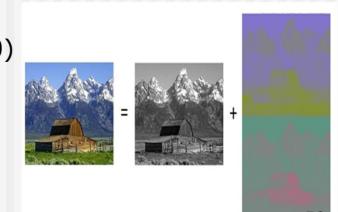
Kích thước ảnh chưa mã hóa:

$$288 \times 352 \times 8\text{bit} \times (1+0,5+0,5)$$

Kích thước ảnh sau mã hóa:

$$288 \times 352 \times 8\text{bit} \times (1/10+0,5/20+0,5/20)$$

Tỷ số nén: 40/3 (lần)



56

Nén ảnh JPEG

- Nén tổn hao (Lossless Compression)
 - Nén không tổn hao (Lossy Compression)
- JPEG : Joint Photographic Experts Group (1986-1992) là chuẩn nén ảnh phổ dụng nhất
- RLC (Run Length Code)
 - VLC (Variable Length Code)
 - DPCM (Differential Pulse Code Modulation)
 - DCT (Discrete Cosine Transform)

57

RLC

- aaaaaaabbbbcccddef
RLC: 7a4b4c2d1e1f

58

VLC -Variable Length Code

- aabcdefaaaaabbbbbcccddefc
a: 7 b: 6 c:5 d: 3 e,f:2
- Fixed Length Code: 3 bit
- a: 000 b:001 c:011 d:010 e: 100 f: 101
- 0000000101101010010100000000... (75 bit)
- aabcdefaaa...
- Mẫu xuất hiện nhiều – từ mã ngắn càng tốt

59

VLC -Variable Length Code

- aabcdefaaaaabbbbbcccddefc
a: 7 b: 6 c:5 d: 3 e,f:2
- VLC
- a: 0 b:1 c:01 d:10 e: 100 f: 101
0010110100101000001111...
aabcdefaaabbbb...
- Quy tắc: Trong bộ từ mã không có từ mã nào làm phần đầu của một từ mã khác.
- Minh họa mã Huffman

60

VLC -Variable Length Code

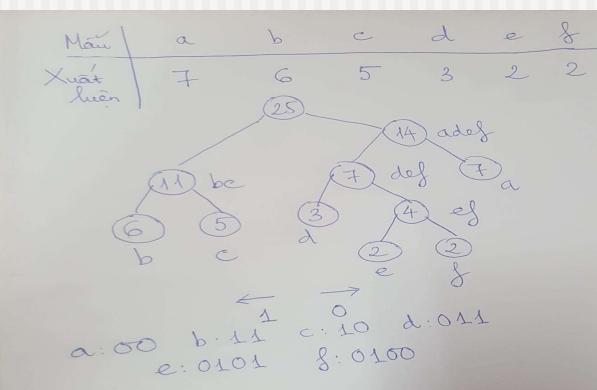
- aabedefaaaaabbbbcccddefc
a: 7 b: 6 c:5 d: 3 e,f:2
a:00 b:11 c:10 d:011 e:0101 f:0100

000011100110101010000000000001111... (53 bit)
2,12 bit/ ký tự

61

VLC -Variable Length Code

- aabedefaaaaabbbbcccddefc
a: 7 b: 6 c:5 d: 3 e,f:2



62

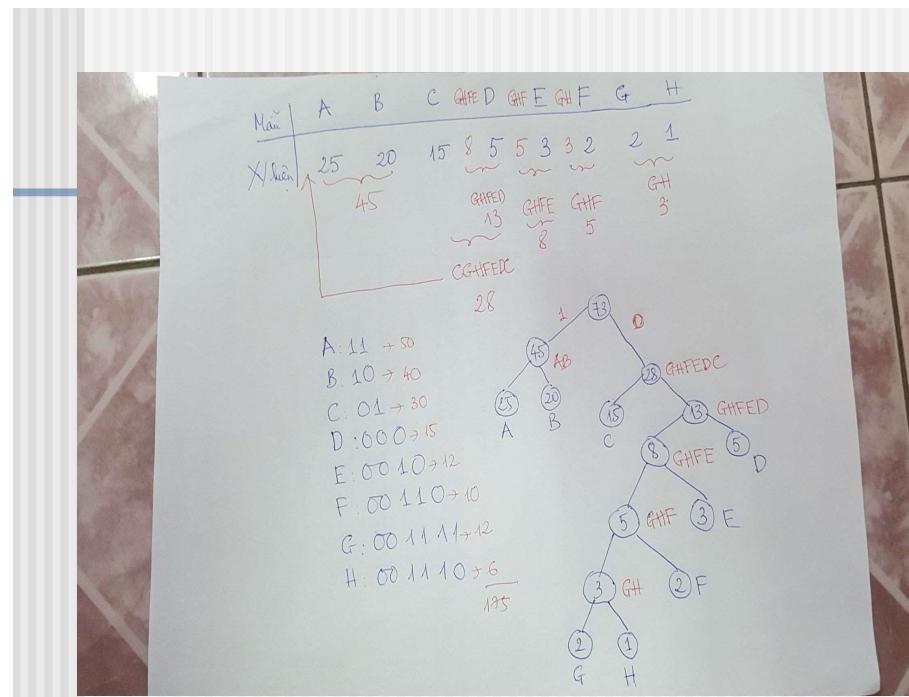
VLC -Variable Length Code

- Vẽ cây nhị phân Huffman cho tập mẫu sau đây:
A: 25 B:20 C:15 D:5 E: 3 F:2 G:2 H:1
Xác định tổng số bit cần thiết để mã hóa cho tập ký tự trên (73 ký tự)
- Trả lời: 175 bit (2,4bit/ký tự)

ABCDAADH....ACE....
111001000111000001110.....11010010.....

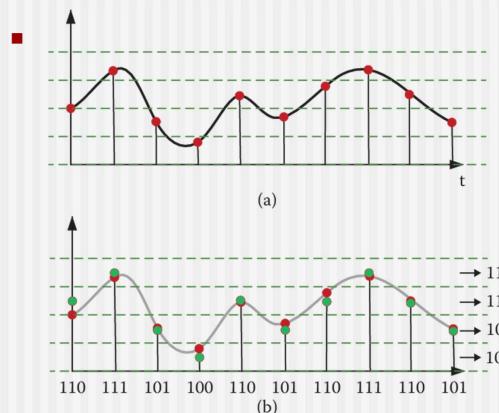
A:11 B:10 C:01 D:000 E:0010 F: 00110 G:001111
H: 001110

63



64

DPCM (Differential Pulse Code Modulation)



127 7

65

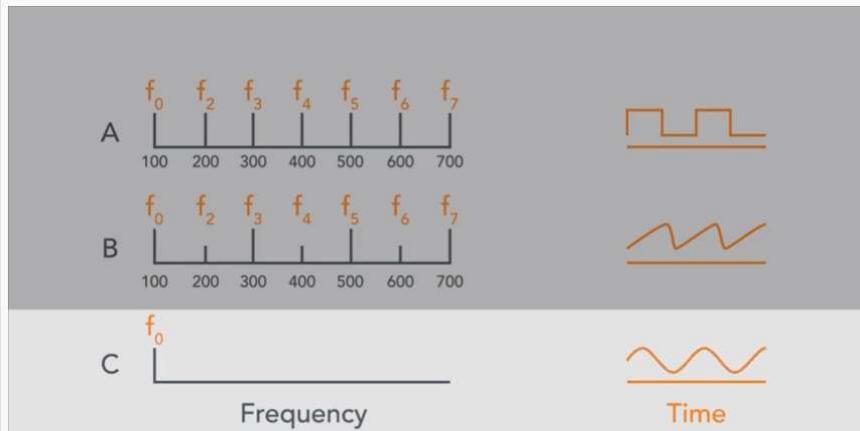
DCT (Discrete Cosine Transform)

- Biến đổi miền thời – miền tần số
- Bắt đầu phép biến đổi cơ bản Fourier Transform (biến đổi Fourier liên tục)

Fourier đã chứng minh một tín hiệu ngẫu nhiên bất kỳ bao giờ cũng phân tích được thành tổng các hàm dao động điều hòa dạng sin, cosin có biên độ, tần số và pha khác nhau.
Tập hợp các tần số này gọi là phổ tín hiệu

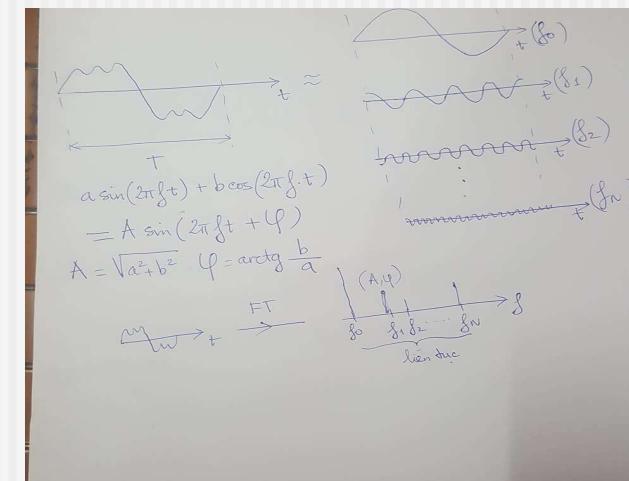
66

MIỀN TẦN SỐ & MIỀN THỜI GIAN



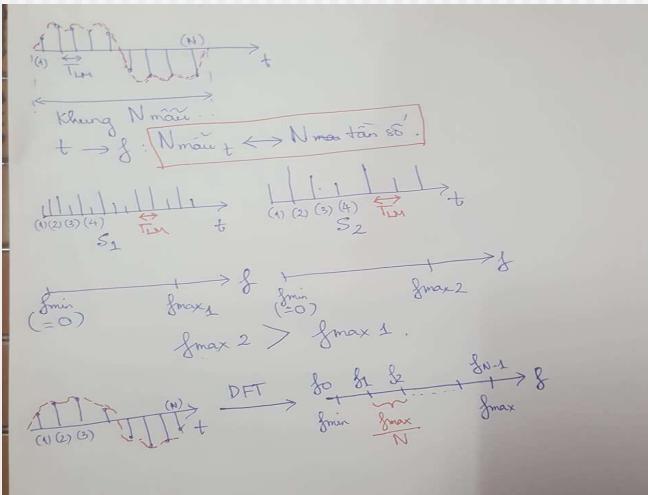
67

FT (Continuos FourierTransform)



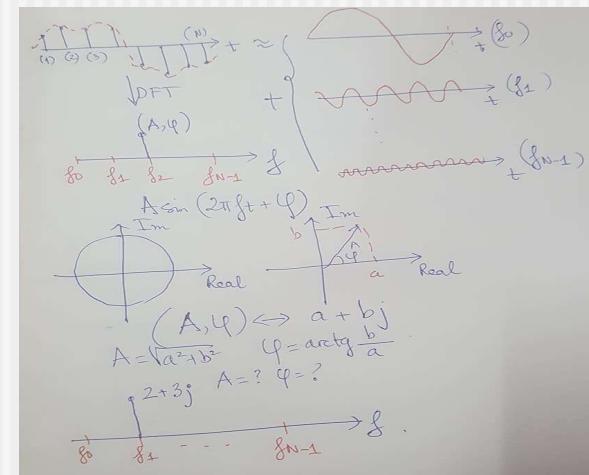
68

DFT (Discrete Fourier Transform)



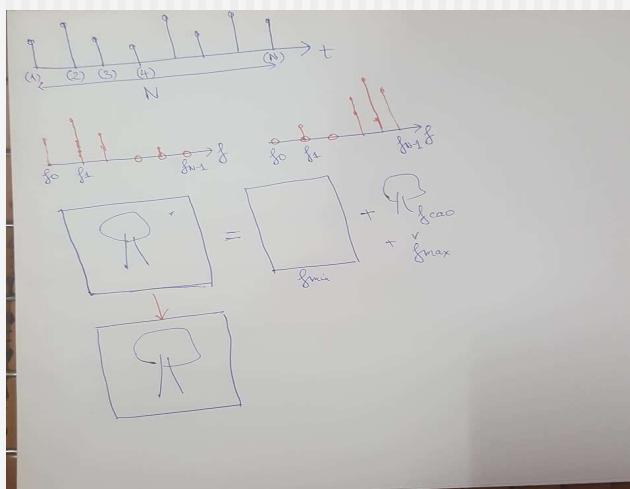
69

DFT (Discrete FourierTransform)



70

Xử lý tín hiệu trong miền tần số



71

Biến đổi DCT 1 chiều

■ N=8

1-D DISCRETE COSINE TRANSFORM DCT

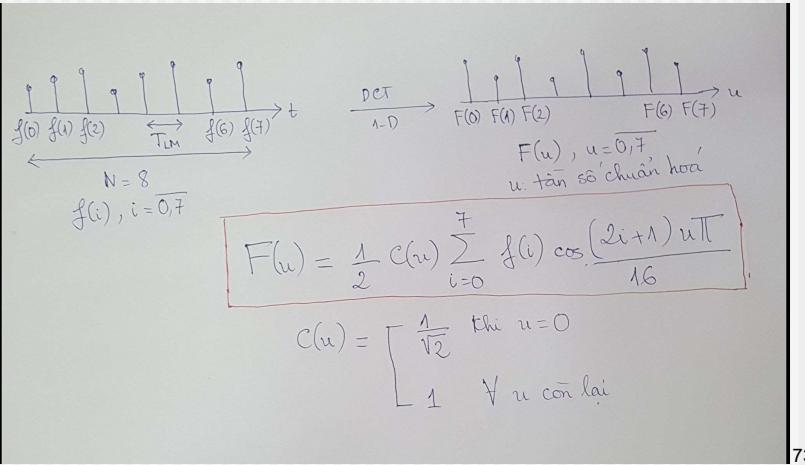
$$C(u) = a(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

$$u = 0, 1, \dots, N-1$$

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

72

Biến đổi DCT 1 chiều



73

Biến đổi DCT 1 chiều

Ví dụ 1:

Cho $f(i) = 20$ với $i=0, 1, \dots, 7$. Thực hiện biến đổi 1D-DCT cho khung tín hiệu trên.

$F(0)=?$

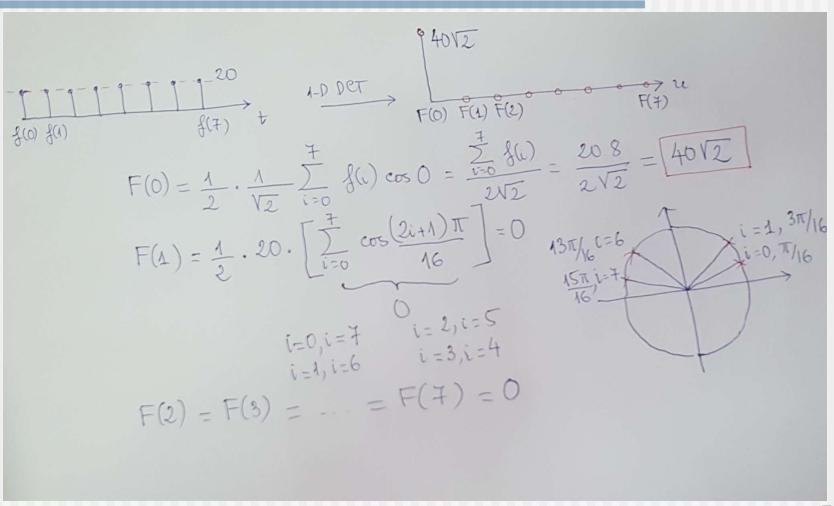
$F(1)=?$

$F(2)=?$

$F(7)=?$

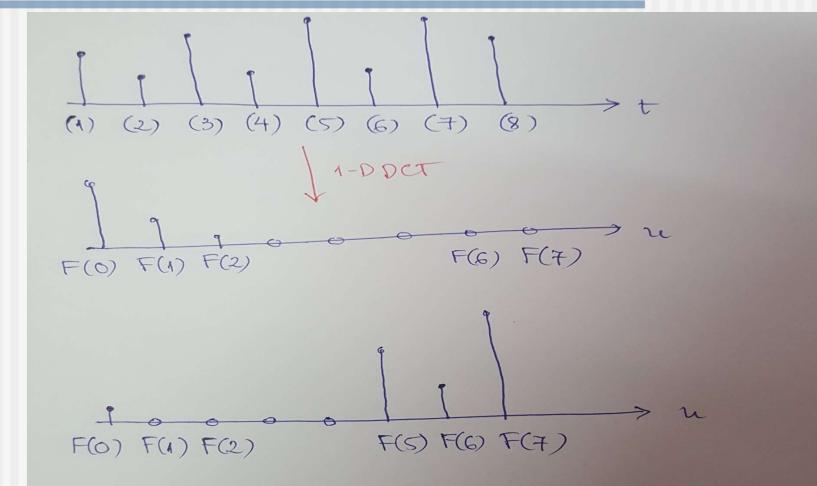
74

Biến đổi DCT 1 chiều



75

Biến đổi DCT 1 chiều



76

Biến đổi DCT 1 chiều

1-D DCT:

$$F(\omega) = \frac{a(u)}{2} \sum_{n=0}^{N-1} f(n) \cos \frac{(2n+1)\omega\pi}{16}$$

1-D Inverse DCT:

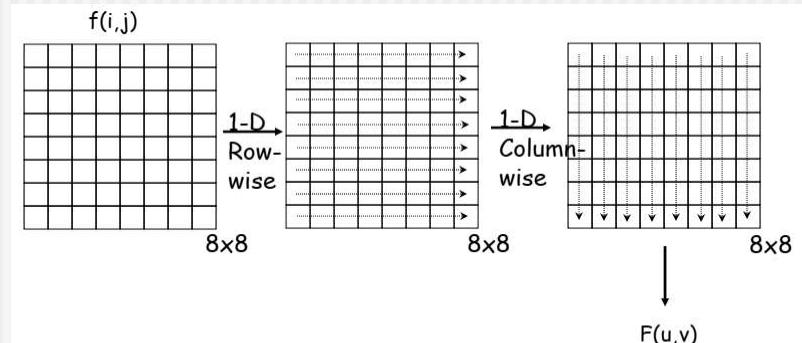
$$f'(n) = \frac{a(u)}{2} \sum_{\omega=0}^{N-1} F(\omega) \cos \frac{(2n+1)\omega\pi}{16}$$

$$a(0) = \frac{1}{\sqrt{2}}$$

$$a(p) = 1 \quad [p \neq 0]$$

77

Biến đổi DCT 2 chiều



r $F(0,0)$ is called the DC component and the rest of $F(i,j)$ are called AC components

78

Biến đổi DCT 2 chiều

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i,j) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16}$$

$$C(u) = C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } u=v=0 \\ 1 & \forall (u,v) \text{ còn lại} \end{cases}$$

79

Biến đổi DCT 2 chiều

Ví dụ 2: $f(i,j)=50$ $i=0,1,\dots,7$; $j=0,1,\dots,7$

Tính

Hệ số DC (Direct Current)

$F(0,0)=?$

63 hệ số AC (Alternating Current)

$F(0,1)=?$

$F(1,0)=?$

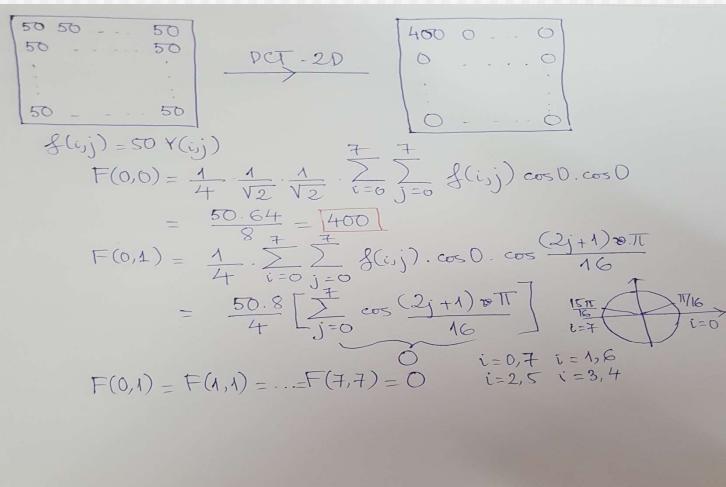
$F(1,1)=?$

....

$F(7,7)=?$

80

Biến đổi DCT 2 chiều



81

Biến đổi DCT 2 chiều

Ví dụ 3:

- If an 8×8 grayscale image is in the range 0 .. 255, what is the largest value a DCT coefficient could be, and for what input image? (Also, state *all* the DCT coefficient values for that image.)
- If we first subtract the value 128 from the whole image and then carry out the DCT, what is the exact effect on the DCT value $F[2, 3]$?
- Why would we carry out that subtraction? Does the subtraction affect the number of bits we need to code the image?
- Would it be possible to invert that subtraction, in the IDCT? If so, how?

82

Biến đổi DCT 2 chiều

Ví dụ 3:

- When the image is all WHITE, i.e., all pixels have $I = 255$. The largest coefficient is the DC value which is $8 \times 255 = 2,040$. All others (AC values) are zero.
- There is no effect on $F[2, 3]$. In fact, no effect on any AC values.
- The idea here is to turn it into a zero mean image, so we do not waste any bits in coding the mean value. (Think of an 8×8 block with intensity values ranging from 120 to 135.)
- After decoding, simply add 128 back to all pixel values.

83

Biến đổi DCT 2 chiều

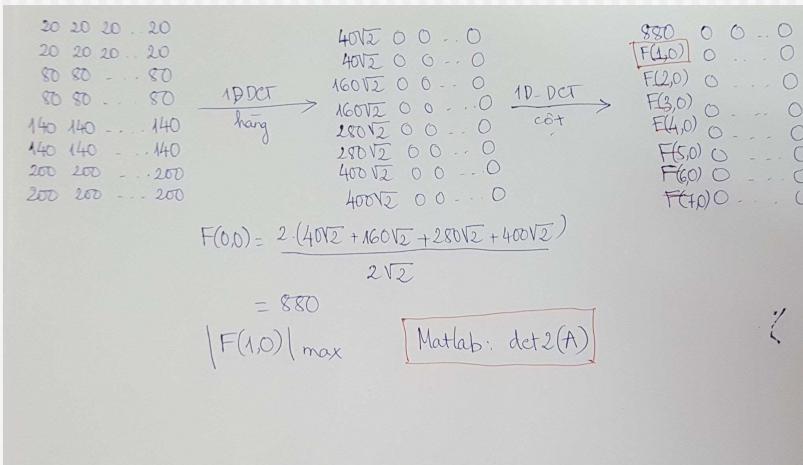
Ví dụ 4:

- What is the value of $F(0, 0)$ if the image $f(i, j)$ is as below?
- Which AC coefficient $|F(u, v)|$ is the largest for this $f(i, j)$? Why? Is this $F(u, v)$ positive or negative? Why?

20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
80	80	80	80	80	80	80	80
80	80	80	80	80	80	80	80
140	140	140	140	140	140	140	140
140	140	140	140	140	140	140	140
200	200	200	200	200	200	200	200
200	200	200	200	200	200	200	200

84

Biến đổi DCT 2 chiều



85

Biến đổi DCT 2 chiều –nén JPEG



An 8×8 block from the Y image of 'Lena'

200 202 189 188 189 175 175 175	515 65 -12 4 1 2 -8 5
200 203 198 188 189 182 178 175	-16 3 2 0 0 -11 -2 3
203 200 200 195 200 187 185 175	-12 6 11 -1 3 0 1 -2
200 200 200 197 187 187 187	-8 3 -4 2 -2 -3 -5 -2
200 205 200 200 195 188 187 175	0 -2 7 -5 4 0 -1 -4
200 200 200 200 190 187 175	0 -3 -1 0 4 1 -1 0
205 200 199 200 191 187 187 175	3 -2 -3 3 3 -1 -1 3
210 200 200 200 188 185 187 186	-2 5 -2 4 -2 2 -3 0

$f(i,j)$

$F(u,v)$

Fig. 9.2: JPEG compression for a smooth image block.

86

Biến đổi DCT 2 chiều –nén JPEG



Another 8×8 block from the Y image of 'Lena'

70 70 100 70 87 87 150 187	-80 -40 89 -73 44 32 53 -3
85 100 96 79 87 154 87 113	-135 -59 -26 6 14 -3 -13 -28
100 85 116 79 70 87 86 196	47 -76 66 -3 -108 -78 33 59
136 69 87 200 79 71 117 96	-2 10 -18 0 33 11 -21 1
161 70 87 200 103 71 96 113	-1 -9 -22 8 32 65 -36 -1
161 123 147 133 113 113 85 161	5 -20 28 -46 3 24 -30 24
146 147 175 100 103 103 163 187	6 -20 37 -28 12 -35 33 17
156 146 189 70 113 161 163 197	-5 -23 33 -30 17 -5 -4 20

$f(i,j)$

$F(u,v)$

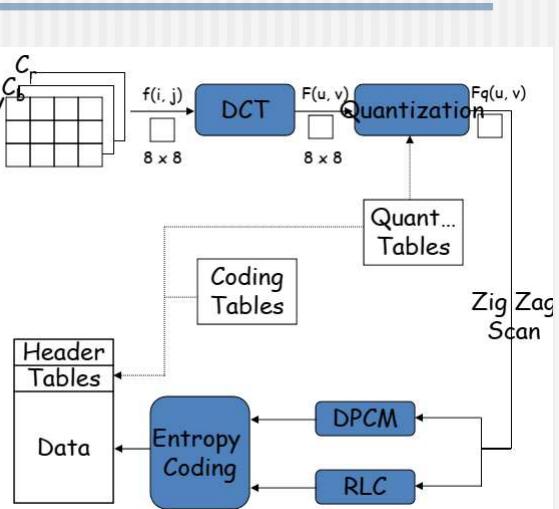
Fig. 9.3: JPEG compression for a textured image block.

87



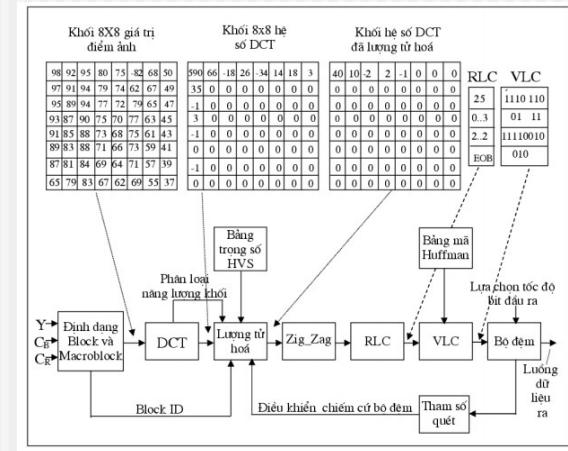
88

Mã hóa JPEG



89

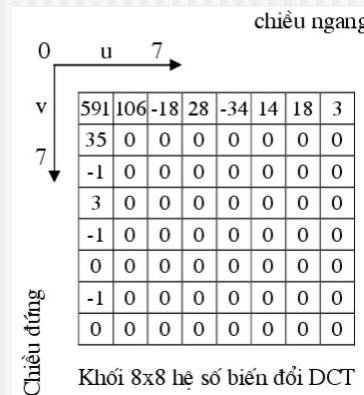
Mã hóa JPEG



90

Mã hóa JPEG – Biến đổi DCT

Khối 8x8 giá trị điểm ảnh							
98	92	95	80	75	82	68	50
97	91	94	78	74	81	67	49
95	89	92	77	72	79	65	47
93	87	91	75	70	77	63	45
91	85	88	73	68	75	61	43
89	83	86	71	66	73	59	41
87	81	84	69	64	71	57	39
85	79	82	67	62	69	55	37



Khối 8x8 hệ số biến đổi DCT

91

Mã hóa JPEG- Bảng trọng số

j	0	1	2	3	4	5	6	7	
k	7	16	11	10	16	24	40	51	61
7	12	12	14	19	26	58	60	55	
7	14	13	16	24	40	57	69	56	
7	14	17	22	29	51	87	80	62	
7	18	22	37	56	68	109	103	77	
7	24	35	55	64	81	104	113	92	
7	49	64	78	87	103	121	120	101	
7	72	92	95	98	112	100	103	99	

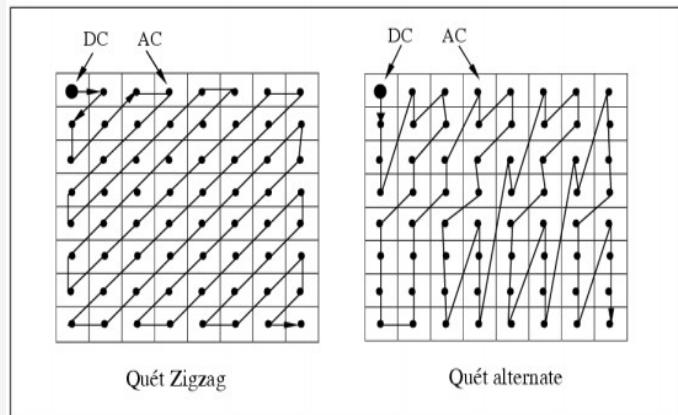
Bảng trọng số (theo chuẩn JPEG cho mẫu tín hiệu màu)

j	0	1	2	3	4	5	6	7	
k	7	17	18	24	47	99	99	99	99
7	18	21	26	66	99	99	99	99	
7	24	26	56	99	99	99	99	99	
7	47	66	99	99	99	99	99	99	
7	99	99	99	99	99	99	99	99	
7	99	99	99	99	99	99	99	99	
7	99	99	99	99	99	99	99	99	
7	99	99	99	99	99	99	99	99	

Bảng trọng số (theo chuẩn JPEG cho mẫu tín hiệu màu)

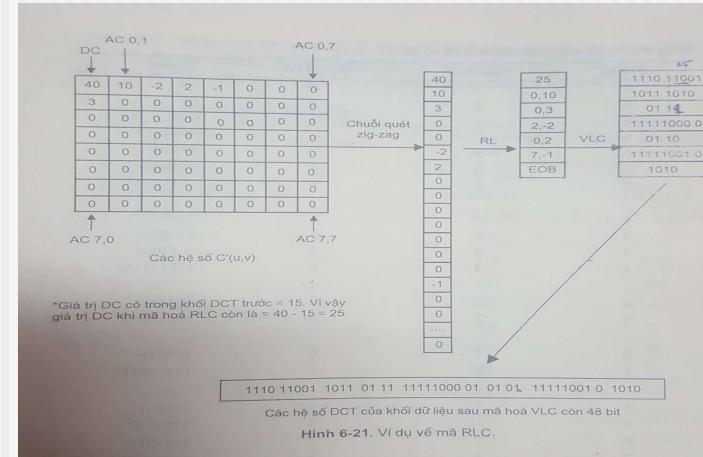
92

Mã hóa JPEG- Quét zig-zac



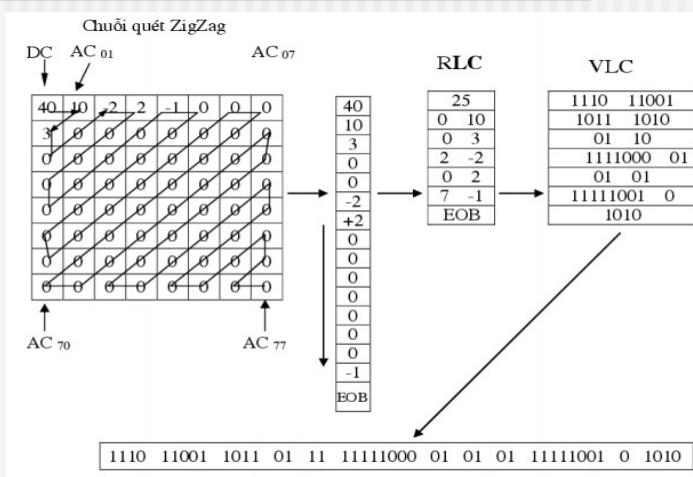
93

Mã hóa JPEG-



94

Mã hóa JPEG-



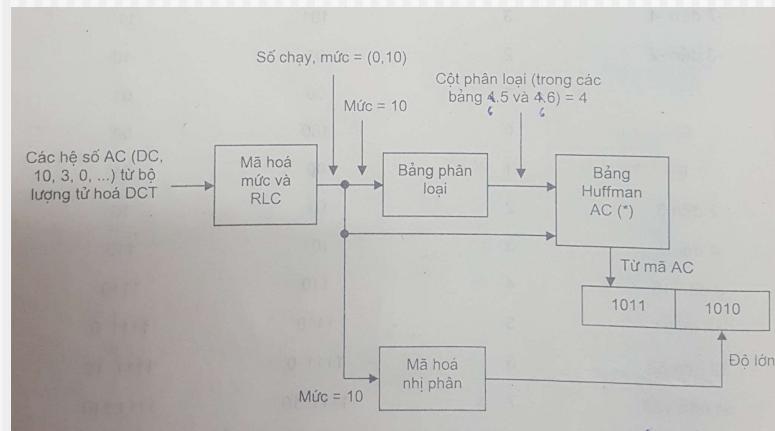
95

Mã hóa JPEG- Mã hóa hệ số xoay chiều AC

- 0,10 : 1011 1010
- 0,3 : 01 11
- 2,-2 : 11110000 01
- 2: 10 → -2: 01
- 0,2 : 01 10
- 7,-1 : 11111001 0
- EOB : End of Block 1010

96

Mã hóa JPEG- Mã hóa hệ số xoay chiều AC



97

Bảng phân loại hệ số AC

Bảng 2.2. Giá trị hệ số AC và phân hạng

Hạng	Giá trị hệ số		
	NA	0	
1	-1	1	
2	-3..-2	2..3	
3	-7,-5,-4	4,5,6,7	
4	-15.....-8	8.....15	
5	-36.....-31	31.....36	
6	-63.....-32	32.....63	
7	-127.....-68	68.....127	
8	-255.....-128	128.....255	
....

98

Bảng mã hóa Huffman hệ số AC

Bảng 2.4. Bảng mã Huffman cho hệ số AC mặc định trong tiêu chuẩn JPEG

Chuỗi 0	Hạng	Dộ dài mã	Tùy mã
0	1	2	00
0	2	2	01
0	3	3	100
0	4	4	1011
0	5	5	11010
0	6	6	111000
0	7	7	1111000
...
1	1	4	1100
1	2	6	111001
1	3	7	1111001
1	4	9	111110110
...
2	1	5	11011
2	2	8	11111000
...
3	1	6	111010
3	2	9	111110111
...
4	1	9	111011
5	1	7	1111010
6	1	7	1111011
7	1	8	11111001
8	1	8	11111010
9	1	9	111111000
10	1	9	111111001
11	1	9	1111111010
...
End of Block	4	1010	

99

Mã hóa JPEG- Mã hóa hệ số xoay chiều AC

1,4 : 1111001 100

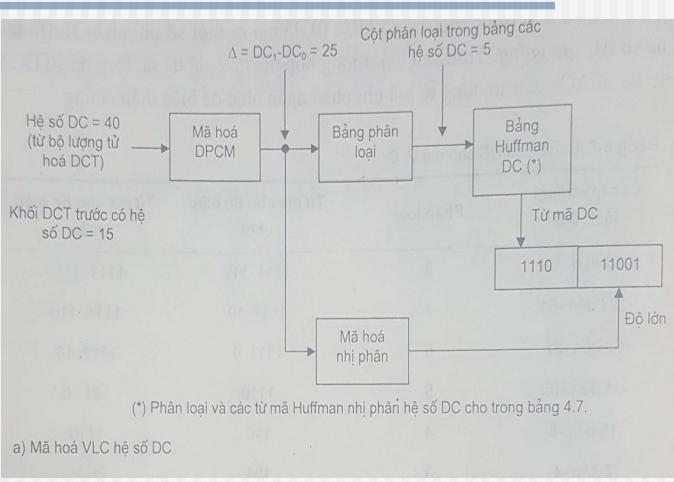
0,-3 : 01 00

2,-2 : 11110000 01

2: 10 → -2: 01

100

Mã hóa JPEG- Mã hóa hệ số xoay DC



a) Mã hóa VLC hệ số DC

101

Bảng phân loại hệ số DC

-255.....-128	7	1111	110
-127.....-64	7	1111	10
-63.....-32	6	1111	0
-31.....-16	5		1110
-15.....-8	4		110
-7.....-4	3		101
-3.....-2	2		01
-1	1		00
0	0		100
1	1		00
2.....3	2		01
4.....7	3		101
8.....15	4		110
16.....31	5		1110
32.....63	6		1111 0
64.....127	7		1111 10
128.....255	8		1111 110

102

Mã hóa JPEG- Mã hóa hệ số xoay chiều DC

40 \rightarrow 25

$$\text{Delta} = DC_N - DC_{N-1} = 40 - 15 = 25$$

25: 1110 11001

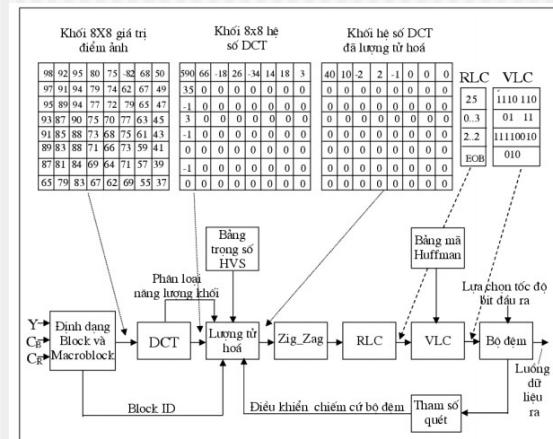
$$\text{Delta} = 128$$

128: 1111110 10000000

103

Mã hóa JPEG

$$\text{Tỷ số nén} = 512/48=10,5 \text{ (lần)}$$



104

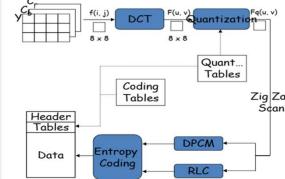
Mã hóa JPEG-Ví dụ 1

Cho khối ảnh chói 8×8 pixel, giá trị mọi điểm ảnh đều bằng 50. Thực hiện mã hóa JPEG cho khối ảnh trên biết giá trị DC (đã lượng tử hóa) khối ảnh trước đó bằng:

- a/ 30 (Delta=-5)
- b/ 25 (Delta=0)

Tính tỷ số nén trong 2 trường hợp trên

- a/ 101 010 1010 (51,2 lần)
- b/ 100 1010 (73 lần)



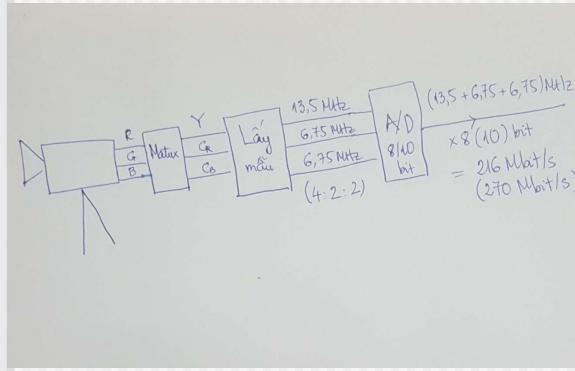
105

Mã hóa MPEG

Moving Picture Experts Group, 1988. Chuẩn nén Video (MPEG-1, MPEG-2, MPEG-4)

Video: chuỗi ảnh

VD: Truyền hình 25 ảnh/s hoặc 30 ảnh/s

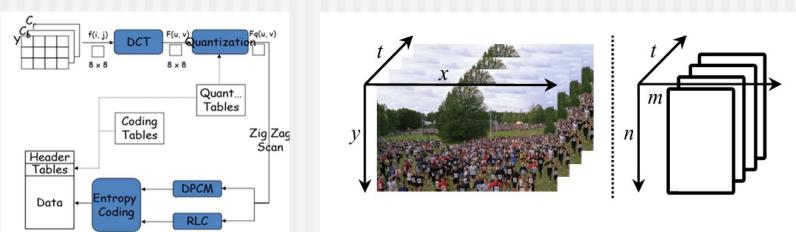


106

Mã hóa MPEG

Moving JPEG (M-JPEG): chuẩn JPEG động

- Chất lượng tốt nhưng đòi hỏi tốc độ cao (~50Mbps), không thích hợp truyền dẫn.
- Lưu trữ, studio....



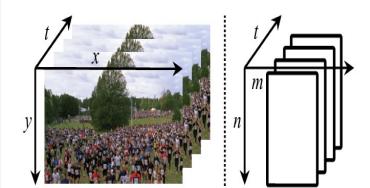
107

Mã hóa MPEG

Chuẩn MPEG ra đời, dùng cho truyền dẫn.

Loại bỏ các thông tin dư thừa: Dư thừa không gian, dư thừa thời gian.

- **MPEG-1 coding 1,15 Mbit/second (quality: VHS)**
(25 pict/s, 288 x 360 pixels, non-interlaced)
- **MPEG-2 coding**
News: 2.3 Mbit/s
Movies: 2...4 Mbit/s, (24 pict/s)
Sports: 4...8 Mbit/s
HDTV: 15...20 Mbit/s
- **MPEG-4 coding**
VHS quality video: 500...700 kbit/s
Movies: 1...2 Mbit/s



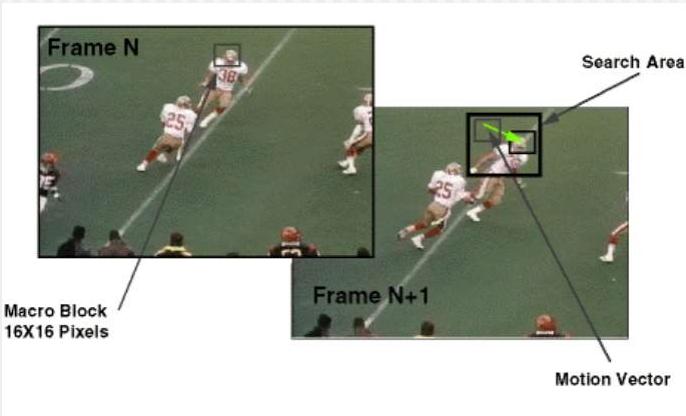
108

Mã hóa MPEG

Dư thừa không gian (spatial redundancy) loại bỏ bằng DCT, dư thừa thời gian (temporal redundancy).

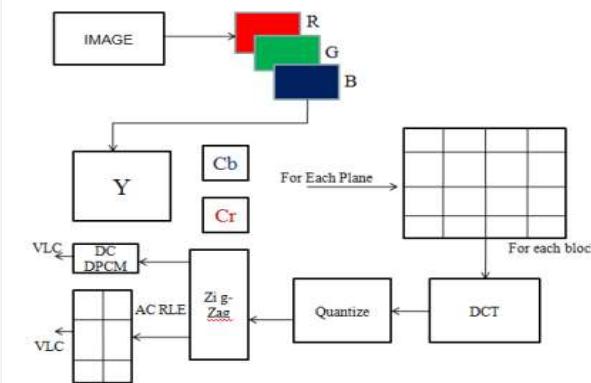


Mã hóa MPEG



111

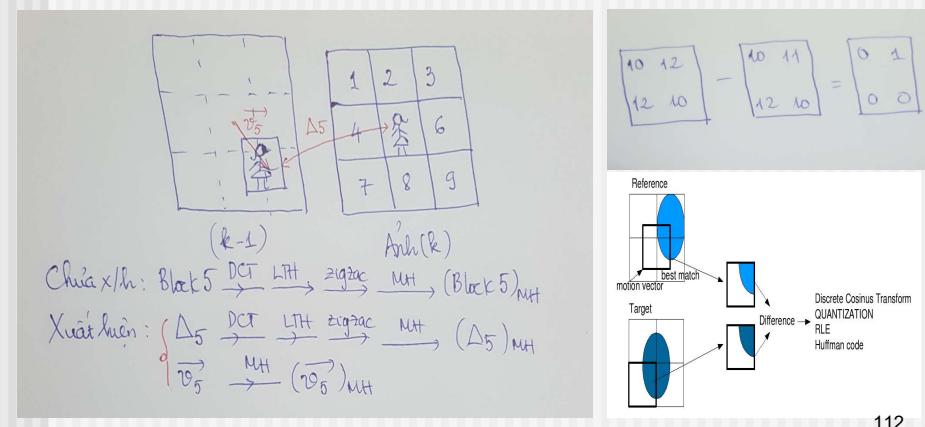
Mã hóa MPEG – dùng 4:2:0



110

Mã hóa MPEG

Loại bỏ dư thừa thời gian thủ tục kiểm tra. Kiểm tra liệu thông tin khối 5 trên ảnh (k) xuất hiện trên ảnh (k-1)?



112

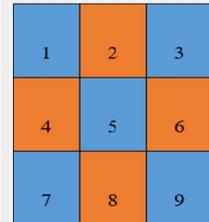
Mã hóa MPEG

Mã hóa MPEG cho ảnh thứ (k) gồm 9 khối 8x8: 1 đến 9.

Biết thông tin các khối 2,4,6,8 có xuất hiện trên ảnh (k-1), thông tin các khối 1,3,5,7,9 không xuất hiện trên ảnh (k-1).

Bộ mã hóa MPEG mã hóa ảnh (k) ntn?

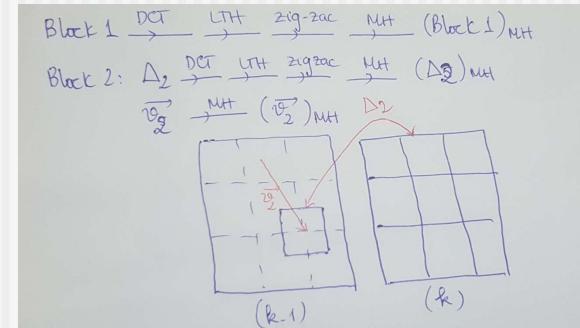
Bộ giải mã MPEG giải mã ảnh (k) ntn?



113

Mã hóa MPEG

Bộ mã hóa MPEG



Tương tự

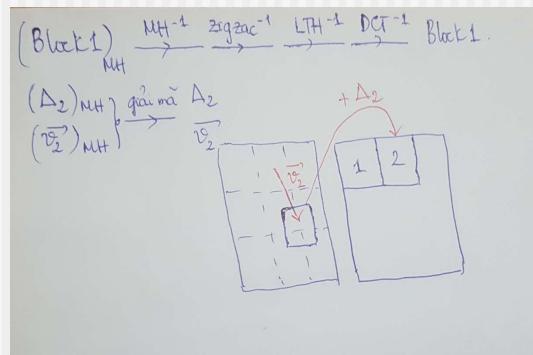
Block 3,5,7,9: giống block 1

Block 4,6,8: giống block 2 (truyền ảnh sai số và vector chuyển động)

114

Mã hóa MPEG

Bộ giải mã hóa MPEG



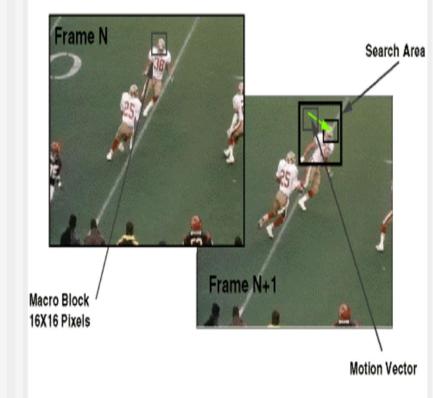
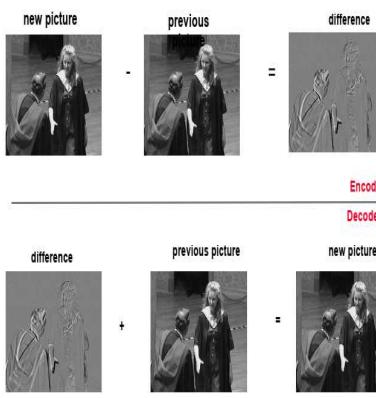
Tương tự

Block 3,5,7,9: giống block 1

Block 4,6,8: giống block 2

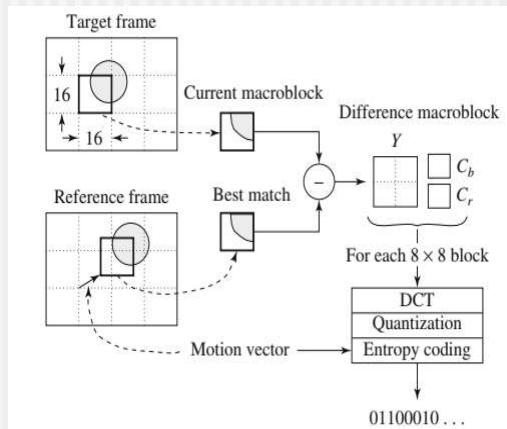
115

Mã hóa MPEG



116

Mã hóa MPEG



117

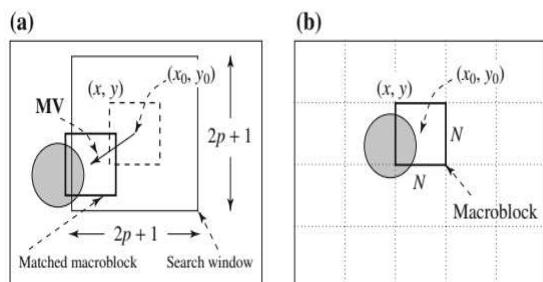
Thuật toán tìm Motion Vector

- Sequential search
- 2D Logarithmic search
- Hierarchical search

118

Thuật toán tìm Motion Vector

-Mean Absoblute Difference



Macroblocks and motion vector in video compression: **a** reference frame; **b** target frame

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

119

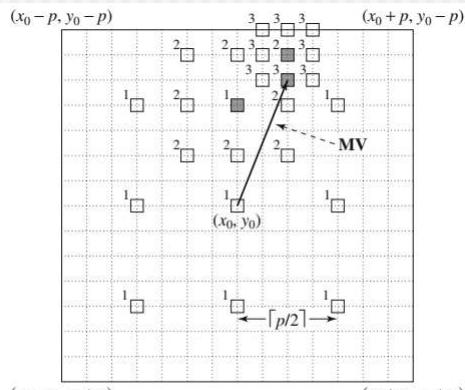
Thuật toán tìm Motion Vector

```

BEGIN
  min_MAD = LARGE_NUMBER; /* Initialization */
  for i = -p to p
    for j = -p to p
      {
        cur_MAD = MAD(i, j);
        if cur_MAD < min_MAD
          {
            min_MAD = cur_MAD;
            u = i;           /* Get the coordinates for MV. */
            v = j;
          }
      }
END
  
```

120

Thuật toán tìm Motion Vector



2D Logarithmic search for motion vectors

121

Thuật toán tìm Motion Vector

Procedure 10.2 (Motion vector: 2D-Logarithmic search)

BEGIN

$$\text{offset} = \lceil \frac{p}{2} \rceil;$$

Specify nine macroblocks within the search window in the Reference frame, they are centered at (x_0, y_0) and separated by offset horizontally and/or vertically;
WHILE last \neq TRUE

{

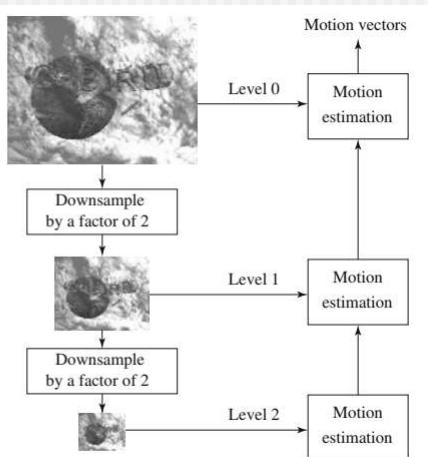
Find one of the nine specified macroblocks that yields the minimum MAD ;
if $\text{offset} = 1$ then last = TRUE;
 $\text{offset} = \lceil \text{offset}/2 \rceil$;
Form a search region with the new offset and new center found;

}

END

Thuật toán tìm Motion Vector

Fig.10.3 A three-level hierarchical search for motion vectors



123

Thuật toán tìm Motion Vector

Procedure 10.3 (Motion vector: Hierarchical search)

BEGIN

// Get macroblock center position at the lowest resolution level k , e.g., level 2.
 $x_0^k = x_0^0/2^k$; $y_0^k = y_0^0/2^k$;

Use Sequential (or 2D Logarithmic) search method to get initial estimated

$\text{MV}(u^k, v^k)$ at level k ;

WHILE last \neq TRUE

{

Find one of the nine macroblocks that yields minimum MAD
at level $k - 1$ centered at
 $(2(x_0^k + u^k) - 1 \leq x \leq 2(x_0^k + u^k) + 1, 2(y_0^k + v^k) - 1 \leq y \leq 2(y_0^k + v^k) + 1)$;
if $k = 1$ then last = TRUE;

$k = k - 1$;

Assign (x_0^k, y_0^k) and (u^k, v^k) with the new center location and motion vector;

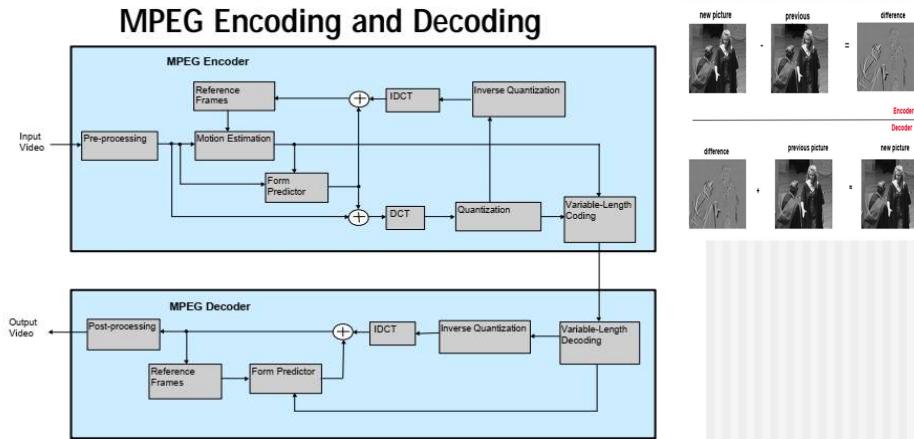
}

END

124

Mã hóa MPEG

MPEG Encoding and Decoding



125

Group of Pictures (GOP)



126

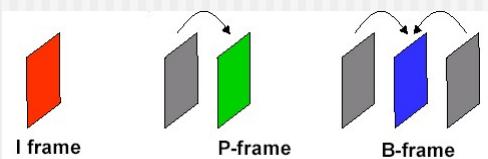
Group of Pictures (GOP)

Group of Pictures (GOP)

I-frames: (Intra) Ảnh I- ảnh mã hóa nguyên

P-frames: (Predicted) Ảnh P-ảnh dự đoán, ảnh được mã hóa dựa vào thông tin của ảnh trước nó.

B-frames: (Bidirectional Predicted) Ảnh B- ảnh dự đoán 2 chiều, ảnh được mã hóa dựa vào thông tin của ảnh trước và sau nó.



127

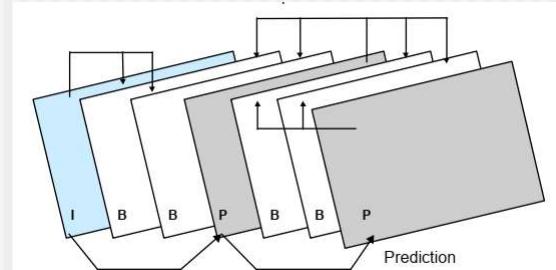
Group of Pictures (GOP)

Thứ tự truyền ảnh ntn?

Tại sao dùng ảnh B?

Tỷ số nén ảnh I,B,P ntn?

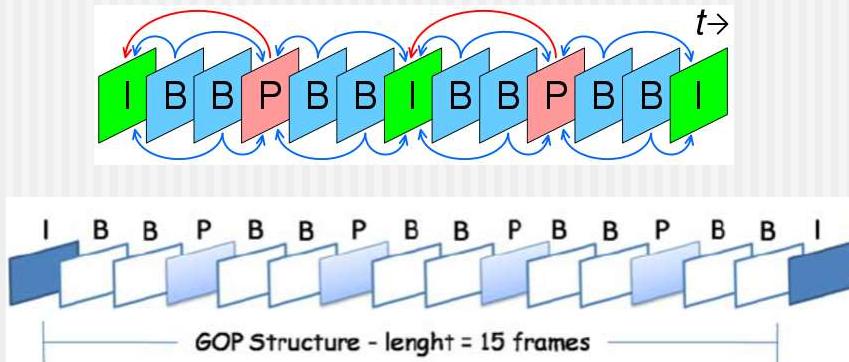
Phân biệt khái niệm ảnh (I,B,P) và (Y,Cr,Cb)



128

Group of Pictures (GOP)

Giải thích GOP sau
Thứ tự truyền: 1 4 2 3 7 5 6 10 8 9... (GOP =6)



129

MPEG

- Bài tập: Đoạn video độ dài 60 phút được mã hóa theo chuẩn MPEG-2, với các tham số: cấu trúc truyền IBBPBB, kích thước ảnh CIF (352x288 pixel, 4:2:0, 8 bit/mẫu), tốc độ 25 hình/s. Biết các tham số nén sau đây:
Tỷ lệ nén trung bình của ảnh I,P và B lần lượt là 8, 16 và 32 lần.(Giả thiết tham số nén đã bao gồm các header cần thiết)
Tính kích thước (dung lượng) đoạn dữ liệu video sau khi mã hóa.

Gợi ý:

Kích thước ảnh chưa nén: $352 \times 288 \times 8\text{bit} \times (1+0,25+0,25)=X$
Trung bình 1s có: 2,5 ảnh I ;7,5 ảnh P và 15 ảnh B.
Tốc độ Video sau mã hóa: $2,5 \times X/8 + 7,5 \times X/16 + 15 \times X/32$
Dung lượng đoạn Video là: 652MB

130

MPEG

- Bài tập: Đoạn video độ dài 60 phút được mã hóa theo chuẩn MPEG-2, với các tham số: cấu trúc truyền IBBPBB, kích thước ảnh CIF (352x288 pixel, 4:2:0, 8 bit/mẫu), tốc độ 25 hình/s. Biết các tham số nén sau đây:
Tỷ lệ nén trung bình của ảnh I,P và B lần lượt là 10, 20 và 40 lần.(Giả thiết tham số nén đã bao gồm các header cần thiết)
Tính kích thước (dung lượng) đoạn dữ liệu video sau khi mã hóa.

Gợi ý:

Kích thước ảnh chưa nén: $352 \times 288 \times 8\text{bit} \times (1+0,25+0,25)=X$

Trung bình 1s có:

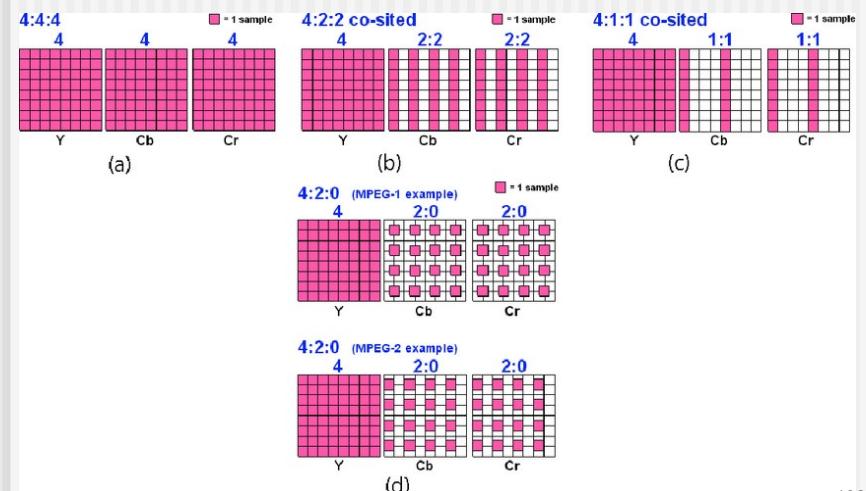
Tốc độ Video sau mã hóa:

Dung lượng đoạn Video là:



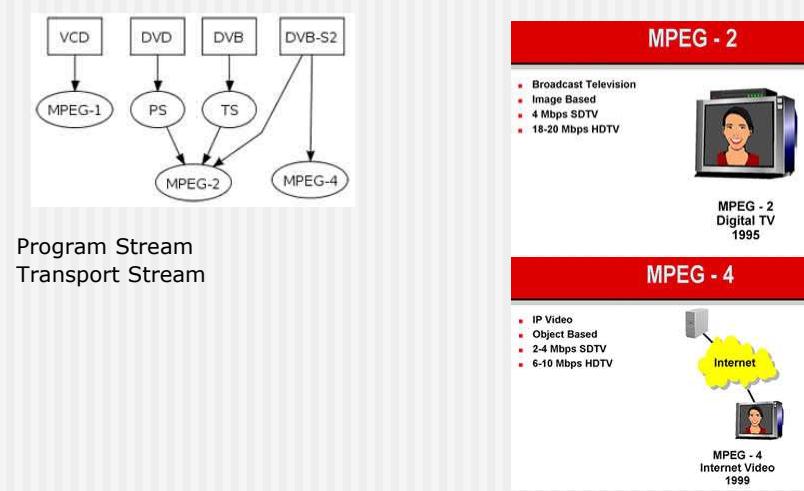
131

MPEG



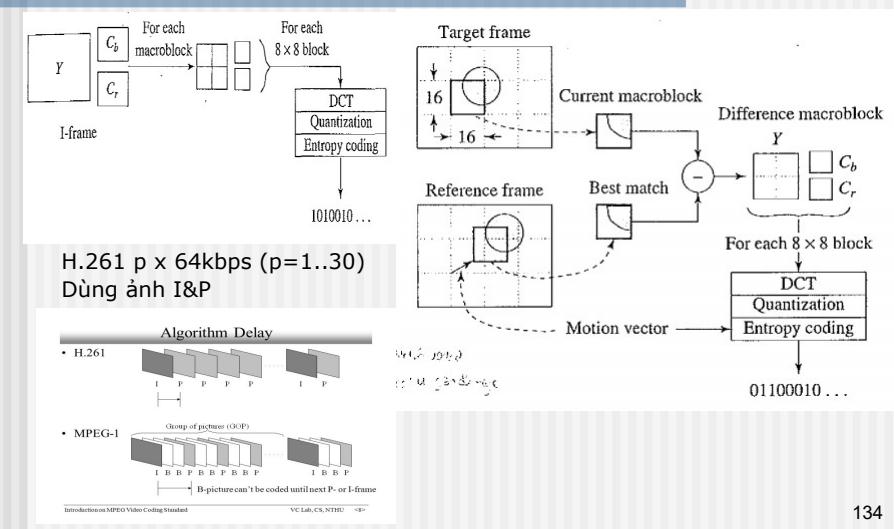
132

MPEG-1/ MPEG-2/MPEG-4



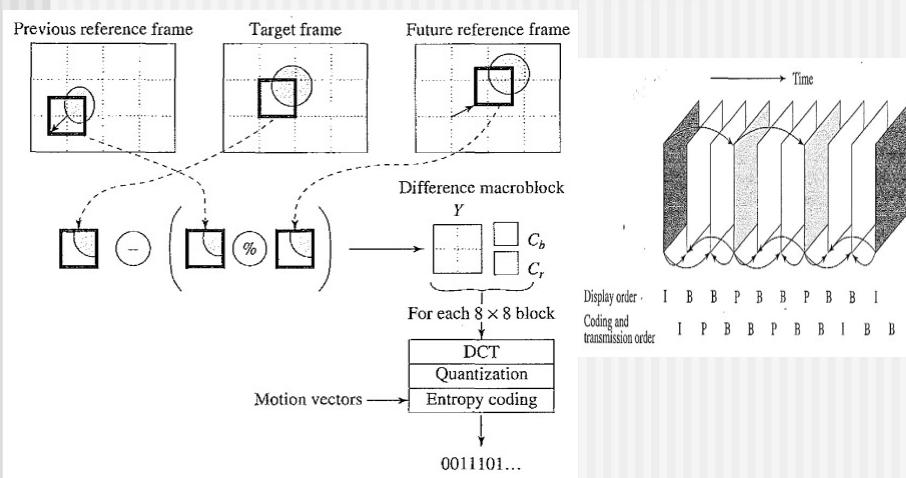
133

MPEG-1/ MPEG-2/MPEG-4



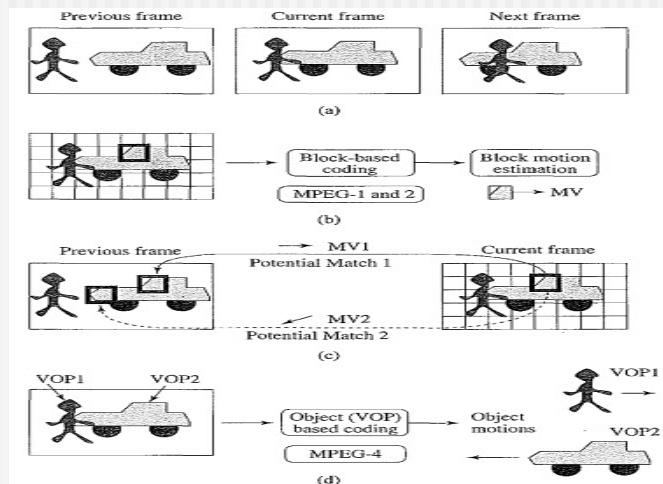
134

MPEG-1/ MPEG-2/MPEG-4



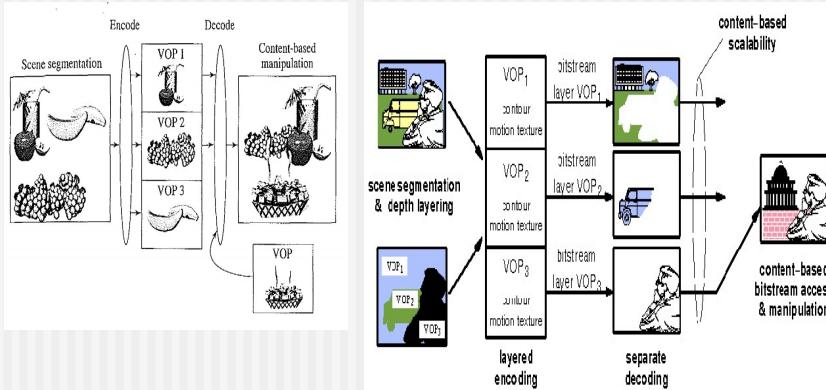
135

MPEG-1/ MPEG-2/MPEG-4



136

MPEG-1/ MPEG-2/MPEG-4

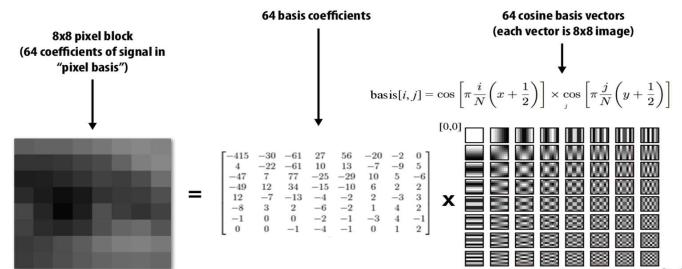


137

Bài tập

BT1: JPEG

Image transform coding via discrete cosine transform (DCT)



In practice: DCT applied to 8x8 pixel blocks of Y' channel, 16x16 pixel blocks of Cb, Cr (assuming 4:2:0)

Stanford CS348K, Fall 2018

138

Bài tập

Quantization

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} / \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Result of DCT
(representation of image in cosine basis)

Changing JPEG quality setting in your favorite photo app modifies this matrix ("lower quality" = higher values for elements in quantization matrix)



Quantization produces small values for coefficients (only few bits needed per coefficient)
Quantization zeros out many coefficients

[Credit: Pat Hanrahan]

Slide credit: Wikipedia, Pat Hanrahan
Stanford CS348K, Fall 2018

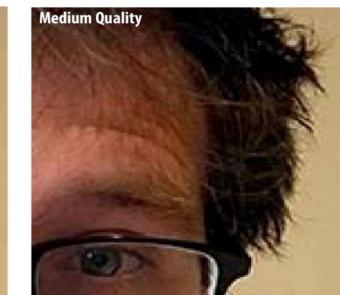
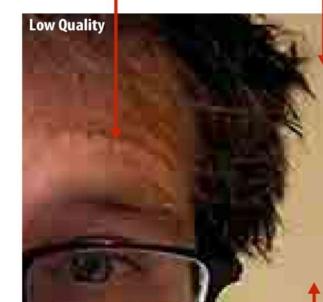
139

Bài tập

JPEG compression artifacts

Noticeable 8x8 pixel block boundaries

Noticeable error near high gradients



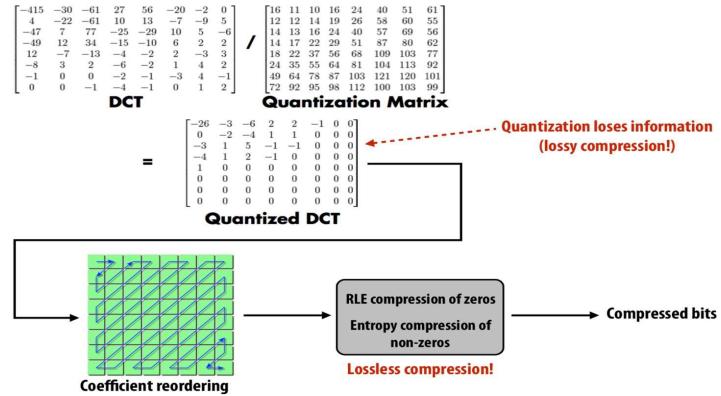
Low-fidelity regions of image represented accurately even under high compression

Stanford CS348K, Fall 2018

140

Bài tập

JPEG compression summary



Credit: Pat Hanrahan

Stanford CS348K, Fall 2018 141

Bài tập

2. In JPEG, the Discrete Cosine Transform is applied to 8×8 blocks in an image. For now, let's call it DCT-8. Generally, we can define a DCT- N to be applied to $N \times N$ blocks in an image. DCT- N is defined as:

$$F_N(u, v) = \frac{2C(u)C(v)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} f(i, j)$$

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

Given $f(i, j)$ as below, show your work for deriving all pixel values of $F_2(u, v)$. (That is, show the result of applying DCT-2 to the image below.)

100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100
100	-100	100	-100	100	-100	100	-100

143

Bài tập

-26:
 (0,-3):
 (1,-3):
 (0,-2):
 (0,-6):
 (0,2):
 (0,-4):
 (0,1):
 (0,-4):
 (0,1):
 (0,1):
 (0,5):
 (0,1):
 (0,2):
 (0,-1):
 (0,1):
 (0,-1):
 (0,2):
 (5,-1):
 (0,-1):
 EOB: 1010

142

Bài tập

Xây dựng cây nhị phân Huffman và tính số lượng bit cần thiết để mã hóa tập ký tự có tần suất xuất hiện như sau:

A: 6 lần B: 4 C: 12 D,E: 5 F:2

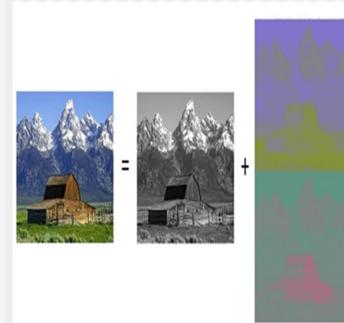
144

Bài tập

Cho một ảnh màu lấy mẫu 4:4:4, giá trị mỗi điểm ảnh nhận 0...255 (8bit/mẫu).

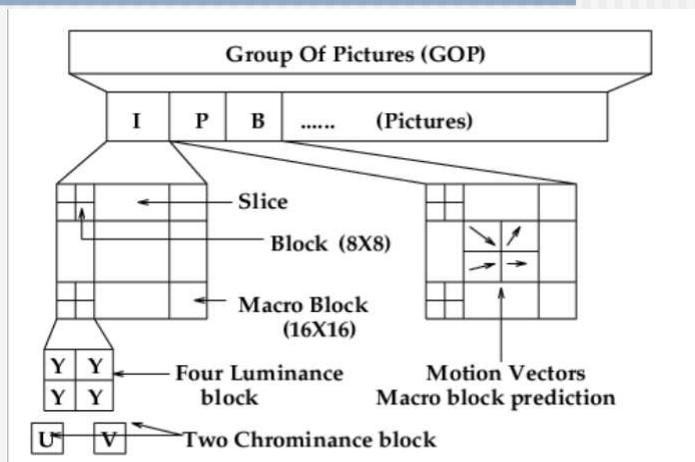
Mã hóa JPEG cho ảnh trên bằng cách chuyển sang chuẩn lấy mẫu 4:2:0 rồi nén JPEG. Biết tỷ số nén cho ảnh chói Y là 10 lần, tỷ số nén cho các tín hiệu màu Cb,Cr là 20 lần.

- Tính tỷ số nén JPEG của ảnh?
- Tỷ số nén cực đại trong JPEG?



145

MPEG1

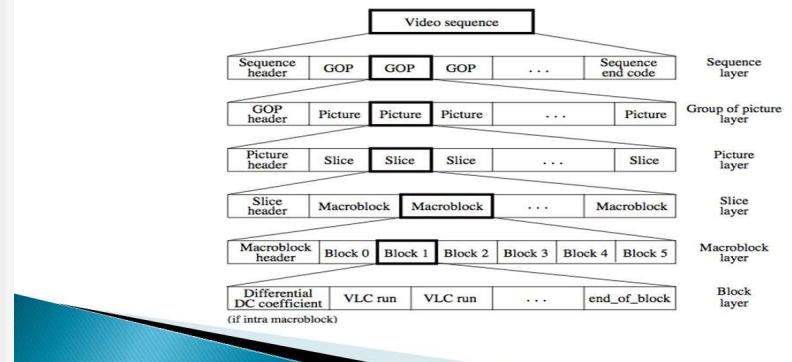


The layered structure of MPEG encoding.

147

MPEG1

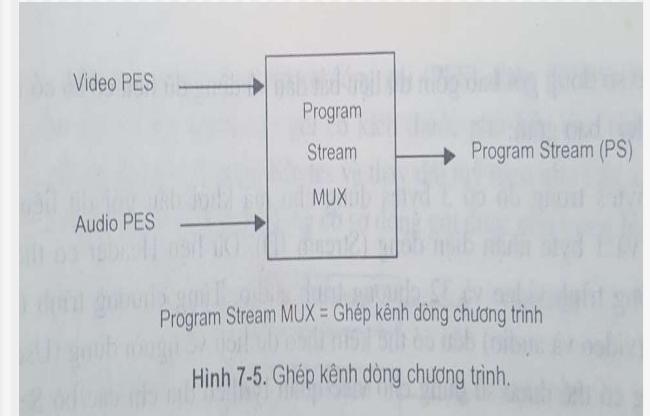
Layers of MPEG-1 Video Bitstream



146

MPEG1

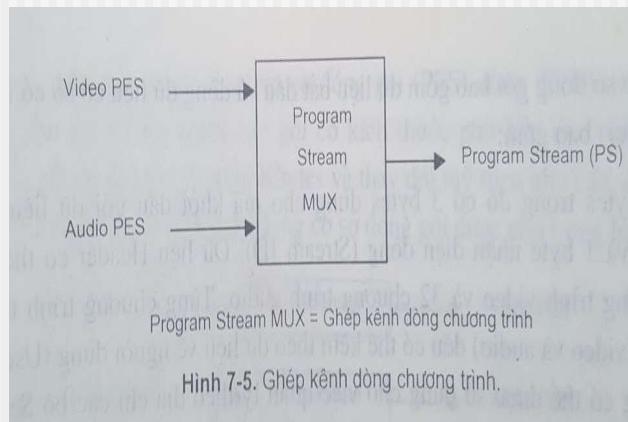
Packet Elementary Stream



148

MPEG1

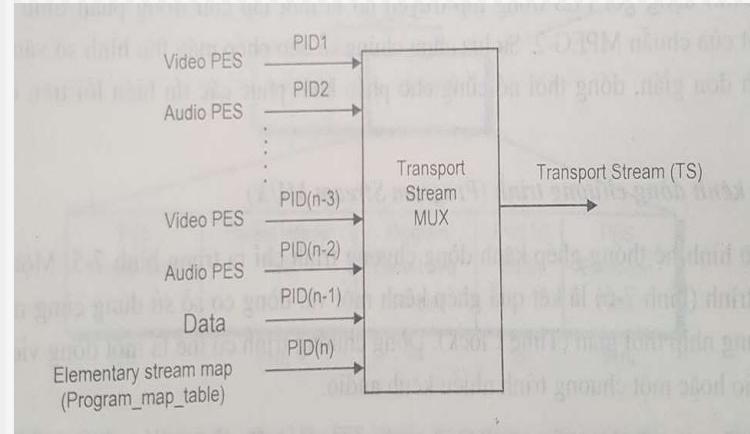
Packet Elementary Stream



149

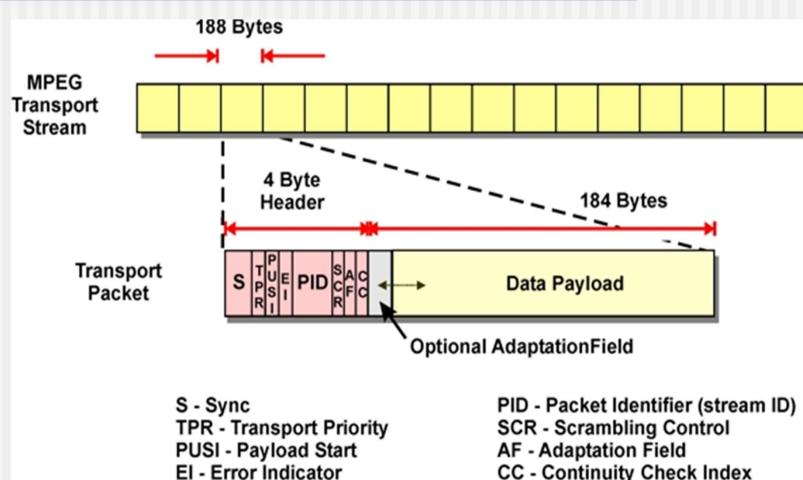
MPEG2

Packet Elementary Stream, Packet Identification



150

MPEG1/MPEG2



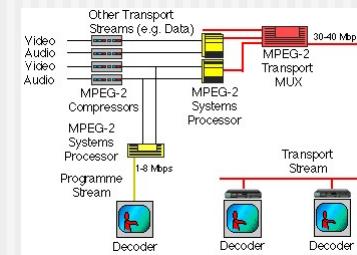
151

MPEG1/2

PS: Program Stream



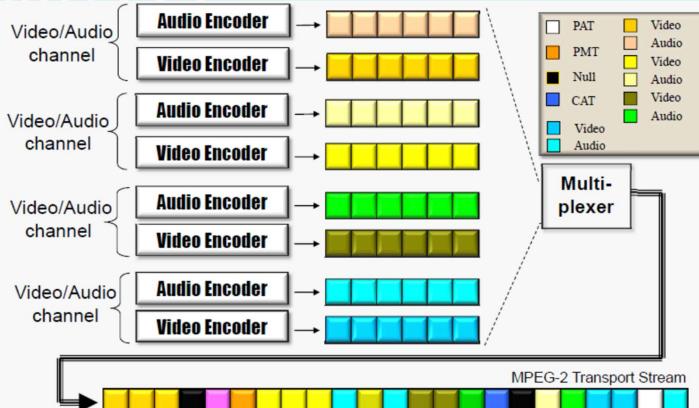
TS: Transport Stream



152

MPEG1/MPEG2

PIDs Defined - Digital Cable Bitstream

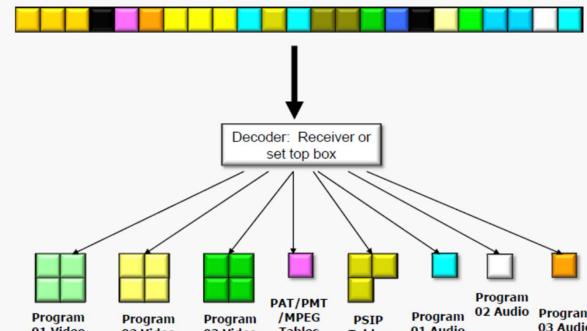


153

MPEG2

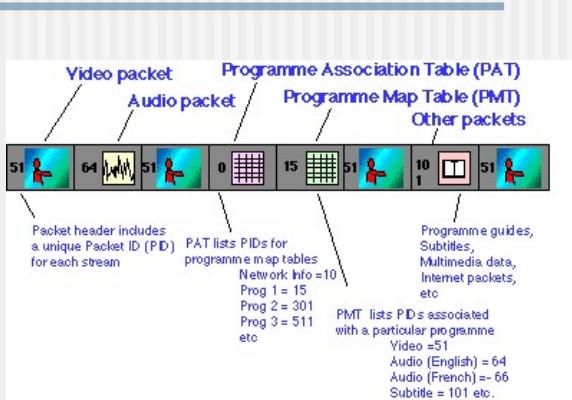
Transport Stream Demultiplexing

- When a set top box first receives a Transport Stream, it demultiplexes that stream based on PID



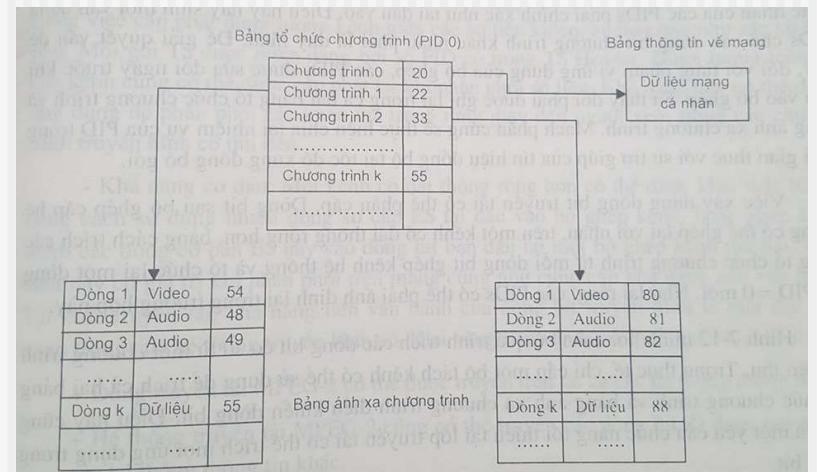
154

MPEG2



155

MPEG2

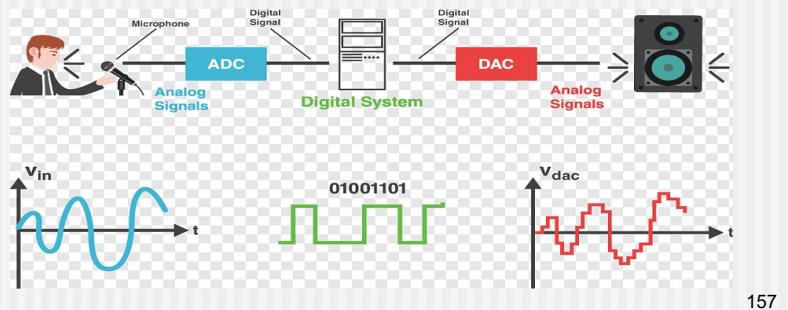


156

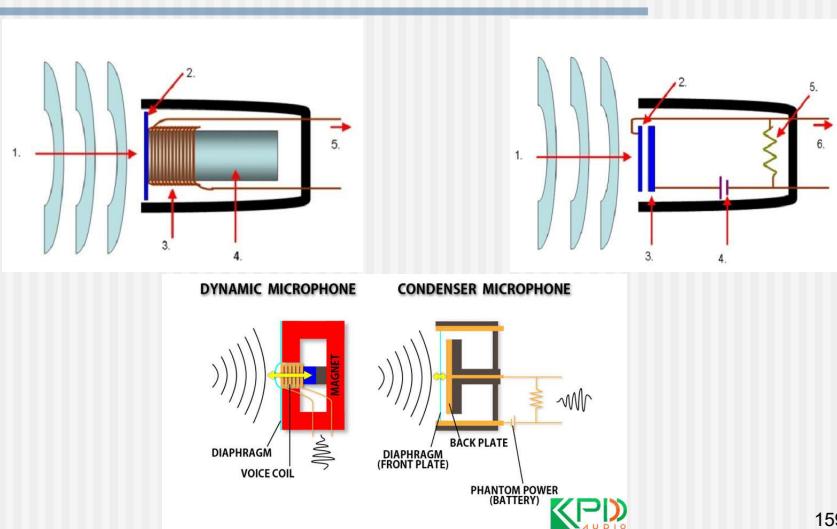
AUDIO

Âm thanh: Dao động cơ học $f=20-20.000$ Hz lan truyền trong môi trường đàn hồi (không khí).

- Tiếng nói (Speech) phổ 300Hz- 4KHz
- Âm thanh (Audio, Music) phổ 20-20KHz



AUDIO



AUDIO

Micro

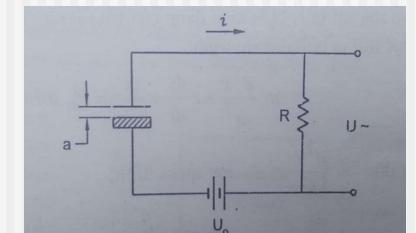
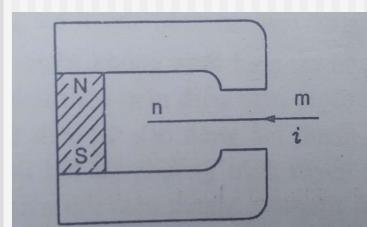
Hai loại phổ biến: Điện động (Dynamic) và điện dung (tụ điện, Condenser)

- **Loại microphone Dynamic** có độ nhạy thấp, dải tần có hạn (từ 50Hz đến 16 KHz). Microphone Dynamic thường dùng cho ca hát sân khấu, karaoke, hát solo, thuyết giảng, hội họp...
- **Loại microphone Condenser** có độ nhạy rất cao và bắt âm thanh rất chính xác, nên thường phục vụ cho các lĩnh vực như thu âm phòng thu, thu âm nhạc cụ, phỏng vấn truyền hình, làm phim, hát hợp xướng, hát opera... các lĩnh vực **cần thu âm thanh ở khoảng cách xa người nói**.

158

AUDIO

Hệ điện động & Hệ tĩnh điện



Tử cảm B, chiều dài dây l

$$\mathbf{F} = \mathbf{B} \cdot \mathbf{l} \cdot i \quad F/i = B \cdot l = \text{const} \quad (\text{tạo loa})$$

$$\mathbf{E} = \mathbf{B} \cdot \mathbf{l} \cdot \mathbf{v} \quad E/v = B \cdot l = \text{const} \quad (\text{tạo micro})$$

160

AUDIO

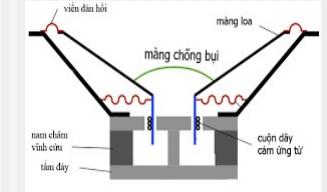
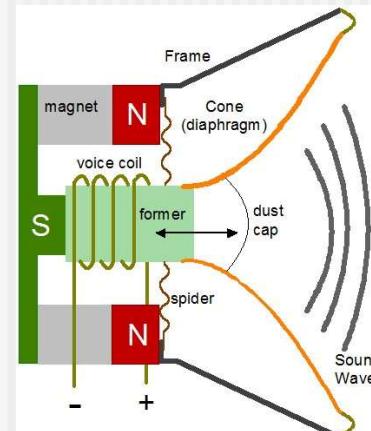
Loa điện động & Loa tĩnh điện (Dynamic Loudspeaker & Electrostatic loudspeaker)



161

AUDIO

Loa điện động

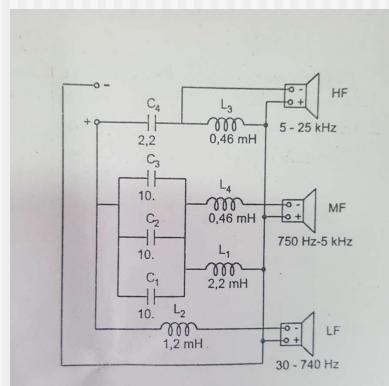


techz

162

AUDIO

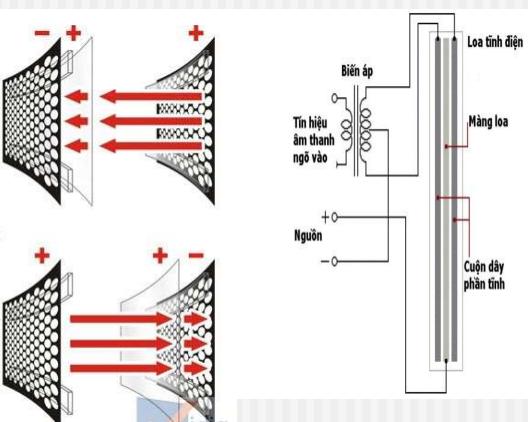
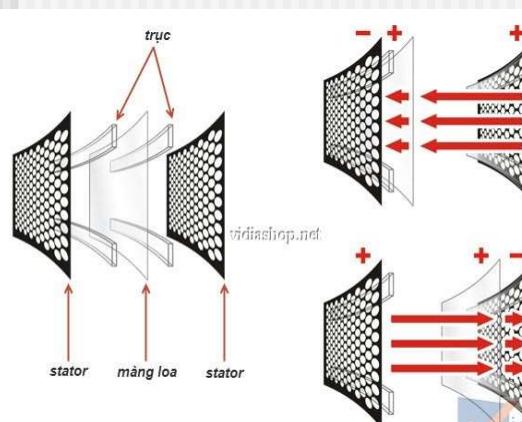
- Loa bass
- Loa trung (midle)
- Loa cao (treble, tép)



163

AUDIO

Loa tĩnh điện

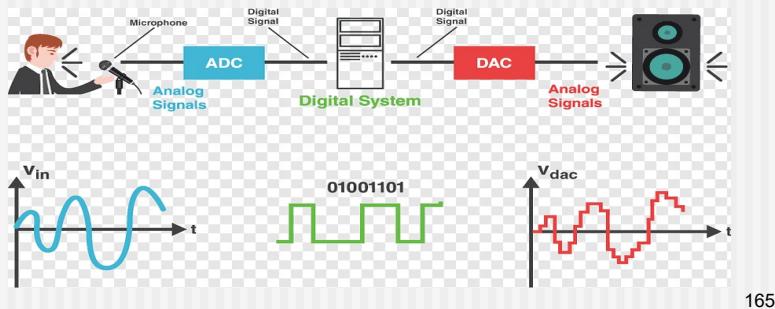


164

SỐ HÓA AUDIO

Nyquist: $f_{LM} \geq 2 f_{max}$

- Tiếng nói (Speech) phổ 300Hz- 4KHz ($f_{LM}=8\text{KHz}$)
- Âm thanh (Audio, Music) phổ 20-20KHz ($f_{LM}=32; 44,1; 48\text{ KHz}$)



SỐ HÓA AUDIO

- Tiếng nói: PCM, mono, $f_{LM}=8\text{KHz}$, 8bit/mẫu, $R=64\text{ kbps}$ (bit per second)
Pulse Code Modulation
- Âm thanh (CD standard): $f_{LM}=44,1\text{ KHz}$, 16bit/mẫu
 $R=2 \times 44.100\text{ mẫu/s} \times 16\text{bit/mẫu}=1,41\text{ Mbit/s}$

BT: Đĩa CD dung lượng 600MB, ghi âm thanh theo chuẩn CD được thời lượng bao nhiêu?
(Trả lời: $t=600\text{MB}/1,41\text{Mbps}=3569\text{s}$)

166

SỐ HÓA TIẾNG NÓI

PCM, mono, $f_{LM}=8\text{KHz}$, 8bit/mẫu, $R=64\text{ kbps}$ (G.711)
(Pulse Code Modulation)

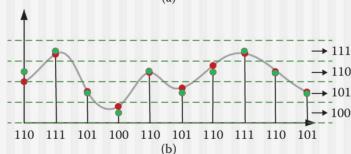
DPCM: Differential PCM, mã hóa vi sai. $R=48\text{kbps}$

ADPCM: Adaptive DPCM, mã hóa vi sai thích nghi, $R=16-32\text{kbps}$ (G.721,G.726)



$$\Delta(n) = s(n) - s(n-1)$$

127 7



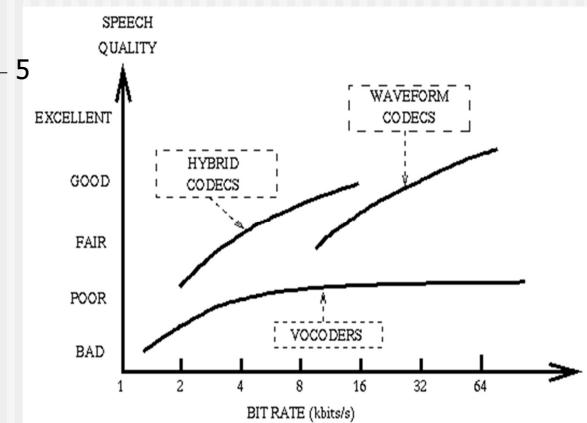
$$s'(n) = \sum_{k=1}^P a_k s(n-k)$$

$$e'(n) = s(n) - s'(n)$$

167

SỐ HÓA TIẾNG NÓI

- Excellent – 5
- Good – 4
- Fair – 3
- Poor – 2
- Bad – 1



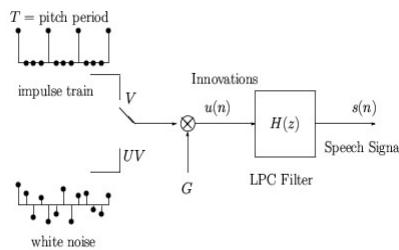
168

Linear Prediction Code

Frame 20ms (160 mẫu): quá trình tĩnh (stationary process)

Âm hữu thanh (Voiced,V): dây thanh quắn rung với tần số (chu kỳ T, độ cao pitch)

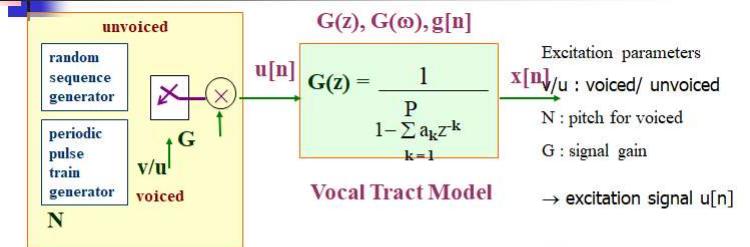
Âm vô thanh (Unvoiced,UV): dây thanh quắn không rung



169

Linear Prediction Code

$(V, UV, T, G, a_1, \dots, a_k)$



A good approximation,
though not precise enough

170

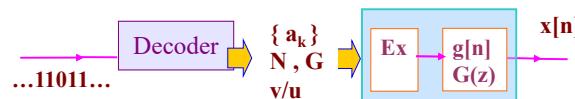
LPC Vocoder(Voice Coder)



N by pitch detection

v/u by voicing detection

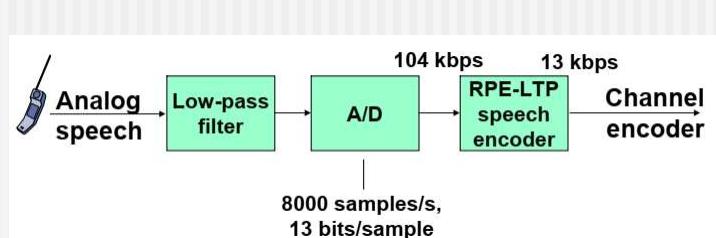
receiver



$\{a_k\}$ can be non-uniform or vector quantized to reduce bit rate further

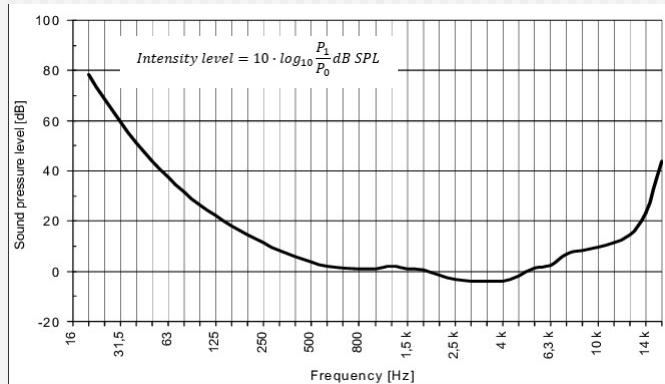
171

SỐ HÓA TIẾNG NÓI



172

Threshold of Hearing



$$T_q(f) = 3.64 \left(\frac{f}{1000} \right)^{-0.8} - 6.5e^{-0.6 \left(\frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left(\frac{f}{1000} \right)^4 \text{ dB SPL}$$

173

Sound Pressure Level (dB)

$$\text{Intensity level} = 10 \cdot \log_{10} \frac{P_1}{P_0} \text{ dB SPL}$$

Sound pressure p (RMS) as Sound field quantity:

The **auditory threshold** is used as the **reference sound pressure $p_0 = 20 \mu\text{Pa} = 2 \times 10^{-5} \text{ Pa}$** . The **threshold of hearing** corresponds to the **sound pressure level $L_p = 0 \text{ dB}$ at $f = 1 \text{ kHz}$** .

Sound intensity I as Sound energy quantity:

The **auditory threshold** is calculated as the **reference sound intensity $I_0 = 10^{-12} \text{ W/m}^2$** . The **threshold of hearing** corresponds also to the **sound intensity level $L_W = 0 \text{ dB}$ at $f = 1 \text{ kHz}$** .

Sound pressure $p = \sqrt{(I \cdot Z_0)}$ Sound intensity $I = p^2 / Z_0$ acoustic impedance $Z_0 = 400 \text{ N}\cdot\text{s}/\text{m}^3$

$$T_q(f) = 3.64 \left(\frac{f}{1000} \right)^{-0.8} - 6.5e^{-0.6 \left(\frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left(\frac{f}{1000} \right)^4 \text{ dB SPL}$$

174

Sound Pressure Level (dB)

$$\text{Intensity level} = 10 \cdot \log_{10} \frac{P_1}{P_0} \text{ dB SPL}$$

- 160 dB Jet engine
- 130 dB Large orchestra at fortissimo
- 100 dB Car on highway
- 70 dB Voice conversation
- 50 dB Quiet residential areas
- 30 dB Very soft whisper
- 20 dB Sound studio

Sound sources (noise) Examples with distance	Sound pressure Level L_p dB SPL
Jet aircraft, 50 m away	140
Threshold of pain	130
Threshold of discomfort	120
Chainsaw, 1 m distance	110
Disco, 1 m from speaker	100
Diesel truck, 10 m away	90
Kerb-side of busy road, 5 m	80
Vacuum cleaner, distance 1 m	70
Conversational speech, 1 m	60
Average home	50
Quiet library	40
Quiet bedroom at night	30
Background in TV studio	20
Rustling leaves in the distance	10
Hearing threshold	0

Sound pressure p $\text{N/m}^2 = \text{Pa}$ Sound field quantity	Sound intensity I W/m^2 Sound energy quantity
200	100
63.2	10
20	1
6.3	0.1
2	0.01
0.63	0.001
0.2	0.0001
0.063	0.00001
0.02	0.000001
0.0063	0.0000001
0.002	0.0000001
0.00063	0.00000001
0.0002	0.000000001
0.000063	0.000000001
0.00002	0.0000000001

175

Loudness Equal Curve

(70Hz,70dB) (1000Hz,40dB) (3500Hz,35dB) Phôn

Loudness versus Amplitude. Which is louder: a 1,000 Hz sound at 60 dB or a 100 Hz sound at 60 dB?

Answer:

The 1000 Hz sound at 60 dB will have loudness at a phon level of 60dB.

The 100 Hz sound will be perceived as a phon level of approximately 50 dB so the 1000 Hz sound will be much louder.

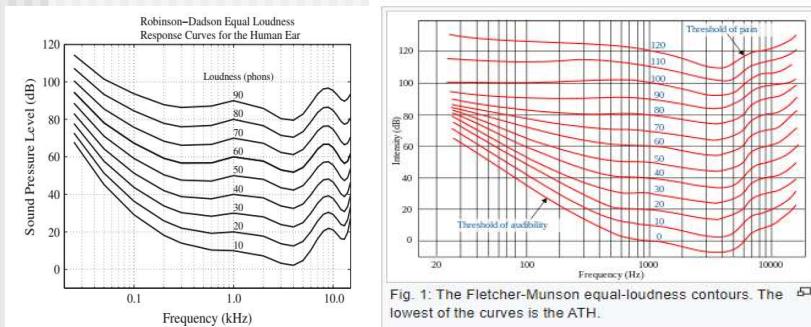


Fig. 1: The Fletcher-Munson equal-loudness contours. The lowest of the curves is the ATH.

176

MÃ HÓA ÂM THANH (ÂM NHẠC)

CD quality – Near CD quality

$$44100 \cdot 16 \cdot 2 = 1411200 \text{ bits per second}$$

Bit depth	Possible values
8	-128 ... 127
16	-32768 ... 32767
24	-8388608 ... 8388607
32	-2147483648 ... 2147483647

- ❑ Example:
Calculate the file size for 1 minute, 44.1 KHz, 16 bits, stereo sound

$$\text{rate} \times \text{duration} \times \text{resolution} \times \text{number of channels}$$

8

- ❑ Where:
 - sampling rate is 44,100 Hz
 - Duration/time is 60 seconds
 - resolution is 16 bits
 - number of channels for stereo is 2

177

MÃ HÓA ÂM THANH (ÂM NHẠC)

Southwestern University
Forward Error Correction (FEC)

Hamming Distance

$$d(101101, 110100) = 3$$

XX X

Error Correcting Code (ECC)

encoded bit sequence	corresponding codeword
00	00000
01	11100
10	10011
11	01111

178

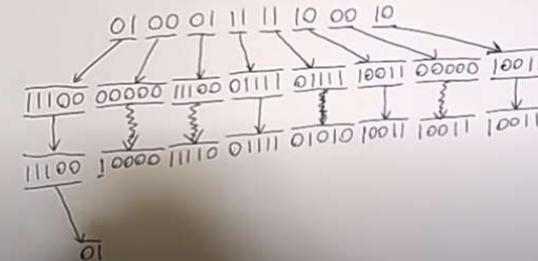
Forward Error Correction

encoded bit sequence	corresponding codeword
00	00000
01	11100
10	10011
11	01111

179

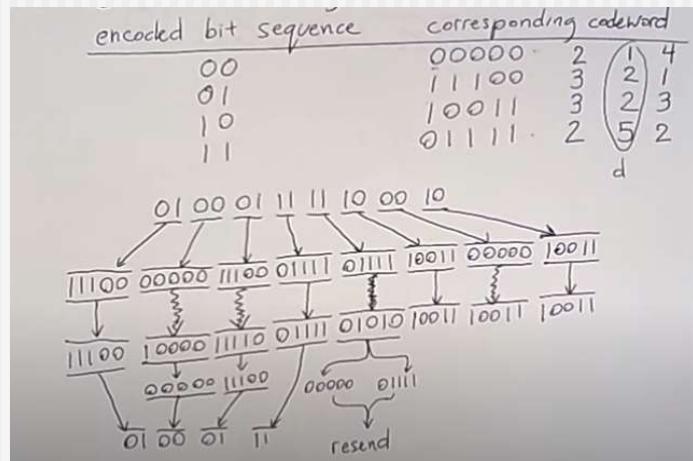
Forward Error Correction

encoded bit sequence	corresponding codeword
00	00000
01	11100
10	10011
11	01111



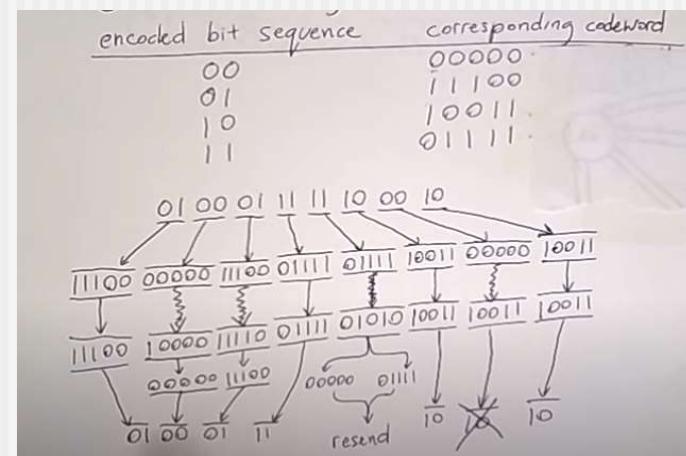
180

Forward Error Correction



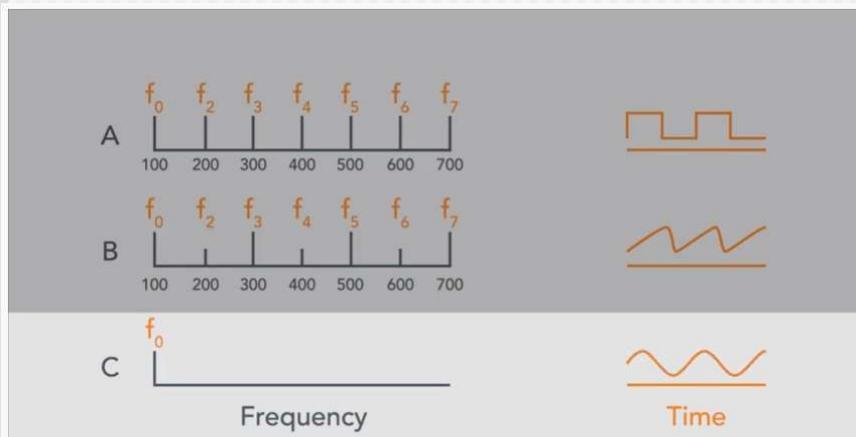
181

Forward Error Correction



182

MÃ HÓA ÂM THANH MIỀN TẦN SỐ & MIỀN THỜI GIAN



183

MÃ HÓA ÂM THANH MIỀN TẦN SỐ & MIỀN THỜI GIAN

	MP3	MPEG-4 HE-AAC	AC-3	Vorbis I	WMA 9
Typical container	MPEG	MPEG	MPEG-2	Ogg	ASF
Sampling rates available (kHz)	32; 44.1; 48	8 - 96	32; 44.1; 48	8 - 192	44.1; 48
Maximum channels	2	48	5 + LFE	255	8
Channel configurations	Left/Right; Mid/Side	many	Up to 5.1	many	
CD transparency bitrate (approx.)	128 kbps	64 kbps	160 kbps	96 kbps	64 - 96 kbps
Maximum bitrate (constant)	320 kbps		640 kbps	500 kbps	384 kbps
Filterbank type	Hybrid (PF+MDCT)	MDCT	MDCT	MDCT	MDCT
Block sizes	384; 1152	256; 2048	512; 256	64 - 8192	128; 256; 512; 1024; 2048
Window shapes	Sine	Sine / KBD	KBD	Vorbis	

184

MÃ HÓA ÂM THANH MIỀN TẦN SỐ



Psychoacoustic Model



Human hearing and voice

- Frequency range is about 20 Hz to 20 kHz, most sensitive at 1 to 5 kHz.
- Dynamic range (quietest to loudest) is about 96 dB
- Normal voice range is about 500 Hz to 2 kHz
 - Low frequencies are vowels and bass
 - High frequencies are consonants

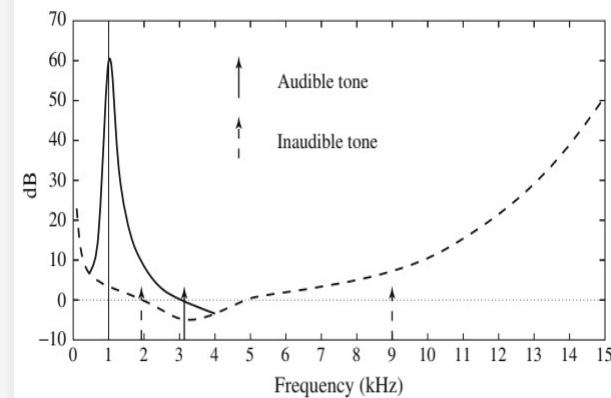
How sensitive is human hearing?

To answer this question we look at the following concepts:

- Threshold of hearing
Describes the notion of "quietness"
- Frequency Masking
A component (at a particular frequency) masks components at neighboring frequencies. Such masking may be partial.
- Temporal Masking
When two tones (samples) are played closed together in time, one can mask the other.

185

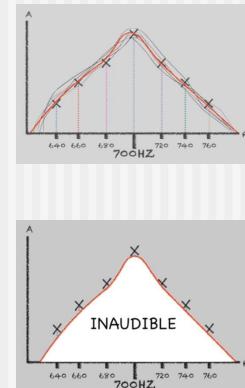
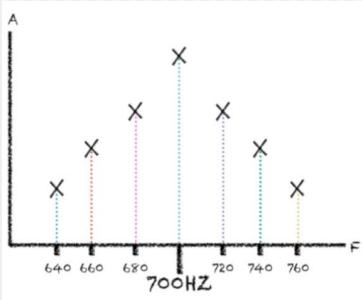
MÃ HÓA ÂM THANH MIỀN TẦN SỐ



$$\text{Threshold}(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4$$

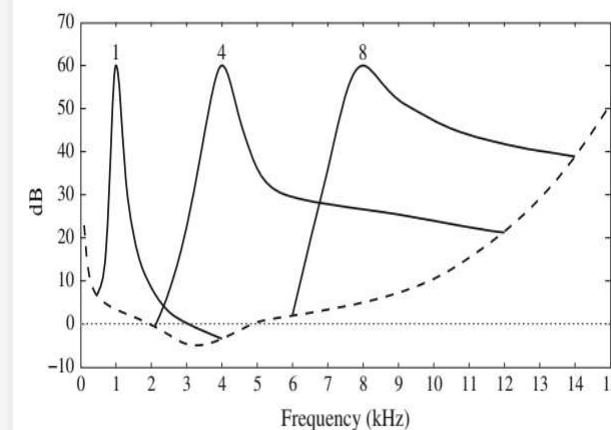
186

Frequency masking



187

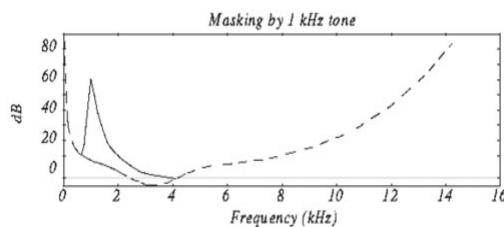
Frequency masking



188

Frequency masking

Experiment: Play 1 kHz tone (masking tone) at fixed level (60 dB). Play test tone at a different level (e.g., 1.1 kHz), and raise level until just distinguishable. Vary the frequency of the test tone and plot the threshold when it becomes audible:



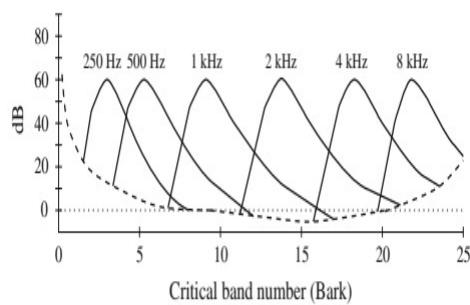
189

Critical Bands

Band #	Lower bound (Hz)	Center (Hz)	Upper bound (Hz)	Bandwidth (Hz)
1	—	50	100	—
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500
25	15500	18775	22050	6550

190

Critical Bands

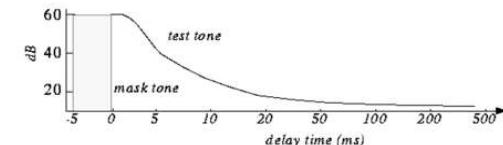


191

Temporal masking

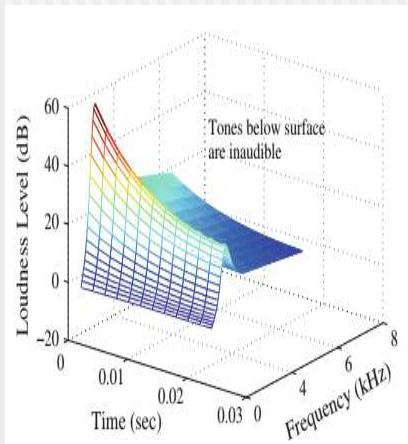
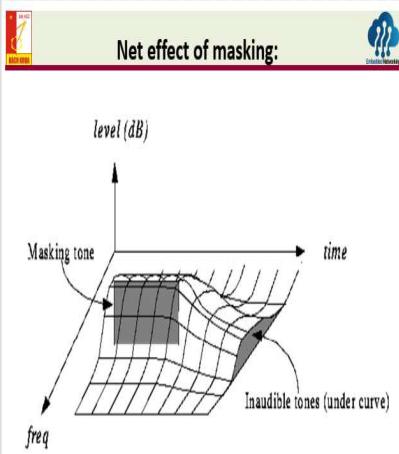


- If we hear a loud sound, and then it stops, it takes a little while until we can hear a soft tone nearby (in frequency).
- Experiment:
 - Play 1 kHz masking tone at 60 dB, plus a test tone at 1.1 kHz at 40 dB. Test tone can't be heard (it's masked).
 - Stop masking tone, then stop test tone after a short delay.
 - Adjust delay time to the shortest time when test tone can be heard (e.g., 5 ms).
 - Repeat with different level of the test tone and plot:



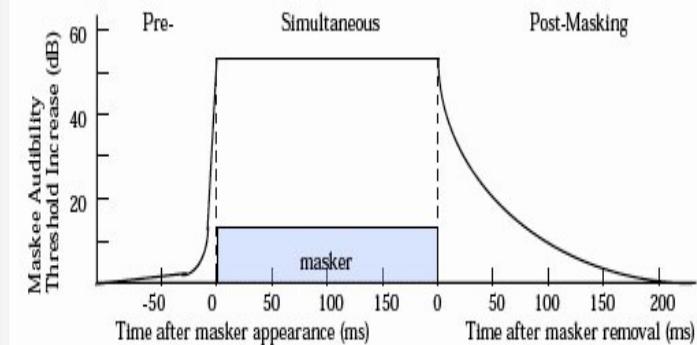
192

Temporal masking



193

Temporal masking



194

Critical bands

The critical bands are of uniform width on the Bark scale.

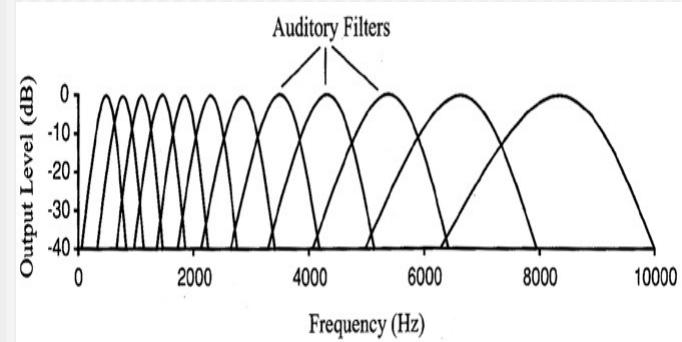
Critical band #	Center freq. (Hz)	Passband (Hz)	Critical band #	Center freq. (Hz)	Passband (Hz)
1	50	-100	14	2 150	2 000 - 2 320
2	150	100 - 200	15	2 500	2 320 - 2 700
3	250	200 - 300	16	1 900	2 700 - 3 150
4	350	300 - 400	17	3 400	3 150 - 3 700
5	450	400 - 510	18	4 000	3 700 - 4 400
6	570	510 - 630	19	4 800	4 400 - 5 300
7	700	630 - 770	20	5 800	5 300 - 6 400
8	840	770 - 920	21	6 400	6 400 - 7 700
9	1 000	920 - 1 080	22	8 500	7 700 - 9 500
10	1 175	1 080 - 1 270	23	10 500	9 500 - 12 000
11	1 370	1 270 - 1 480	24	13 500	12 000 - 15 500
12	1 600	1 480 - 1 720	25	19 500	15 500 -

Table 2 Idealized Critical Band Filterbank

$$BW_c = 25 + 75 \cdot \left(1 + 1.4 \cdot \left(\frac{f}{1000}\right)^2\right)^{0.69} \text{ Hz}$$

195

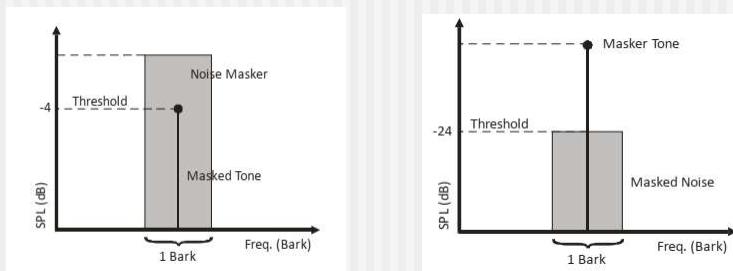
Critical bands



196

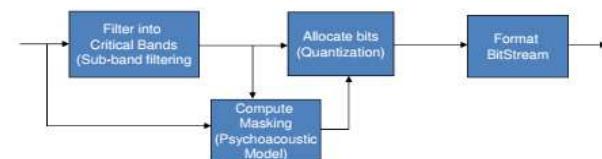
Critical bands

- Tone masking Noise
- Noise masking Tone
- Noise masking Noise

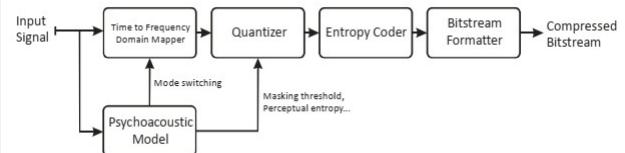


197

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

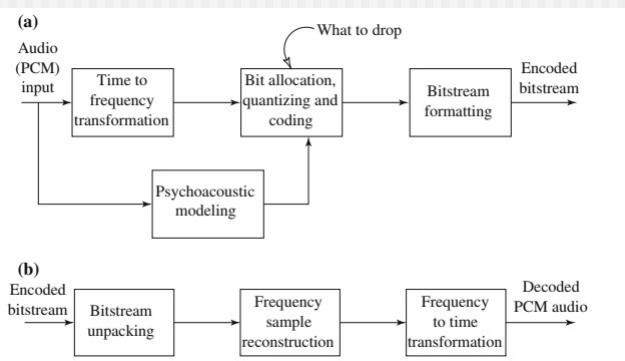


1. Use convolution filters to divide the audio signal (e.g., 48 kHz sound) into 32 frequency sub-bands. (*sub-band filtering*)
2. Determine amount of masking for each band caused by nearby band using the *psychoacoustic model*.
3. If the power in a band is below the masking threshold, don't encode it.



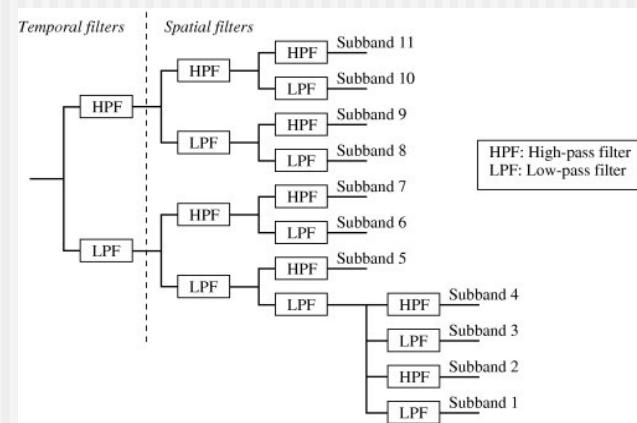
198

MÃ HÓA ÂM THANH MIỀN TẦN SỐ



199

MÃ HÓA ÂM THANH MIỀN TẦN SỐ



200

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

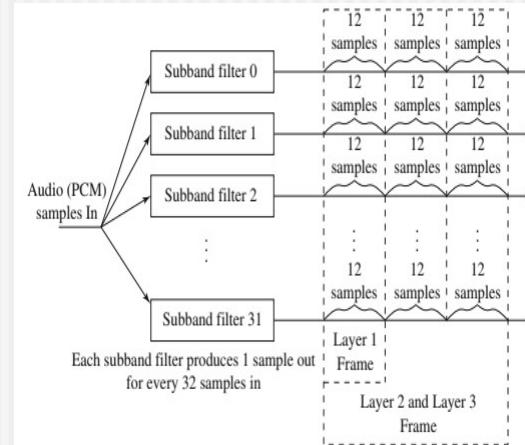
- Say, performing the sub-band filtering step on the input results in the following values (for demonstration, we are only looking at the first 16 of the 32 bands):

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- The 60dB level of the 8th band gives a masking of 12 dB in the 7th band, 15dB in the 9th. (according to the Psychoacoustic model)
- The level in 7th band is 10 dB (< 12 dB), so ignore it.
- The level in 9th band is 35 dB (> 15 dB), so send it.

201

MÃ HÓA ÂM THANH MIỀN TẦN SỐ



202

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

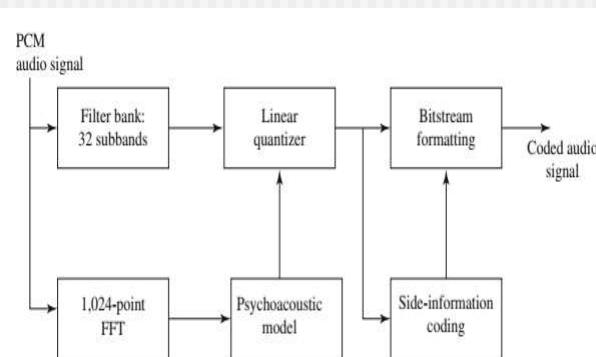


Fig.14.13 MPEG-1 Audio Layers 1 and 2

203

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

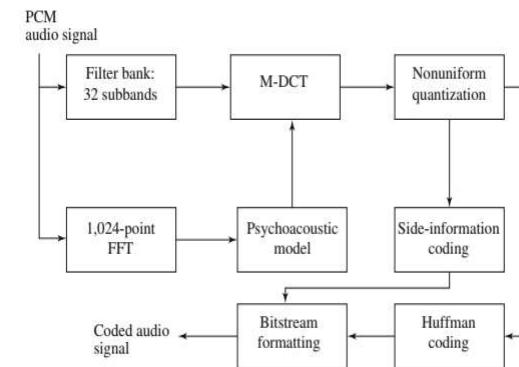


Fig.14.14 MPEG-1 Audio Layer 3

204

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

Table 14.2 MP3 compression performance

Sound quality	Bitrate	Mode	Compression ratio
Telephony	8 kbps	Mono	96:1
Better than shortwave	16 kbps	Mono	48:1
Better than AM radio	32 kbps	Mono	24:1
Similar to FM radio	56–64 kbps	Stereo	26:1–24:1
Near-CD	96 kbps	Stereo	16:1
CD	112–128 kbps	Stereo	14:1–12:1

205

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

MPEG Coding Specifics

- **MPEG Layer I**
 - Filter is applied one frame (12x32 = 384 samples) at a time. At 48 kHz, each frame carries 8ms of sound.
 - Uses a 512-point FFT to get detailed spectral information about the signal. (sub-band filter). Uses equal frequency spread per band.
 - Psychoacoustic model only uses frequency masking.
 - Typical applications: Digital recording on tapes, hard disks, or magneto-optical disks, which can tolerate the high bit rate.
 - Highest quality is achieved with a bit rate of 384k bps.
- **MPEG Layer II**
 - Use three frames in filter (before, current, next, a total of 1152 samples). At 48 kHz, each frame carries 24 ms of sound.
 - Models a little bit of the temporal masking.
 - Uses a 1024-point FFT for greater frequency resolution. Uses equal frequency spread per band.
 - Highest quality is achieved with a bit rate of 256k bps.
 - Typical applications: Audio Broadcasting, Television, Consumer and Professional Recording, and Multimedia.

206

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

MPEG Coding Specifics

- **MPEG Layer III**
 - Better critical band filter is used
 - Uses non-equal frequency bands
 - Psychoacoustic model includes temporal masking effects, takes into account stereo redundancy, and uses Huffman coder.

Stereo Redundancy Coding:

- Intensity stereo coding -- at upper-frequency sub-bands, encode summed signals instead of independent signals from left and right channels.
- Middle/Side (MS) stereo coding -- encode middle (sum of left and right) and side (difference of left and right) channels.

207

MÃ HÓA ÂM THANH MIỀN TẦN SỐ

Effectiveness of MPEG Audio

Layer	Target bit-rate	Ratio	Quality* at 64 kbps	Quality at 128 kbps
Layer I	192 kbps	4:1	--	--
Layer II	128 kbps	6:1	2.1 to 2.6	4+
Layer III	64 kbps	12:1	3.6 to 3.8	4+

*Quality factor:

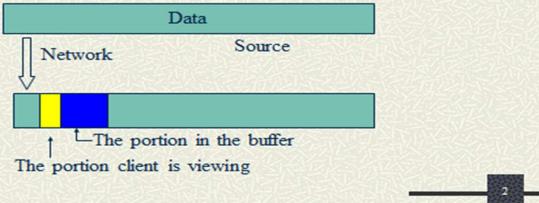
- 5 – perfect
- 4 - just noticeable
- 3 - slightly annoying
- 2 - annoying
- 1 - very annoying

208

Video/Audio Streaming

What is Streaming Technology

A technique for transferring data such that it can be processed as a steady and continuous stream, client does not have to download the entire file to view it.



209

Video/Audio Streaming

OSI

Application
Presentation
Session
Transport
Network
Data link
Physical

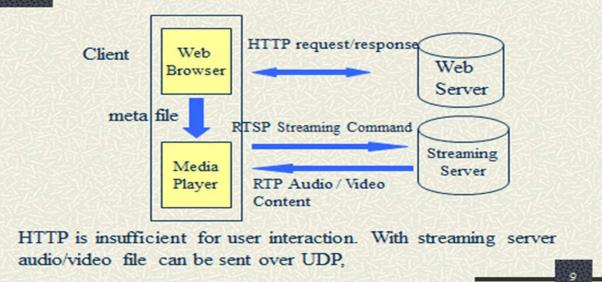
TCP / IP

Application	FTP, Telnet, SMTP/MIME HTTP, SNMP, etc.
Transport	TCP (connection-oriented) UDP (connectionless)
Internet	IPv4, IPv6, RSVP
Network access (LLC and MAC)	X.25, Ethernet, Token ring, FDDI, PPP/SLIP, etc.
Physical	10/100Base-T, 1000Base-T, Fibre Channel, etc.

210

Video/Audio Streaming

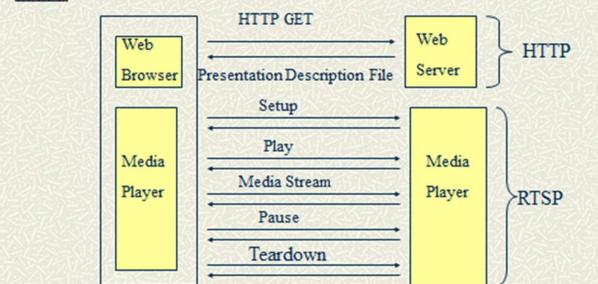
RTSP Diagram



211

Video/Audio Streaming

RTSP Operation



212

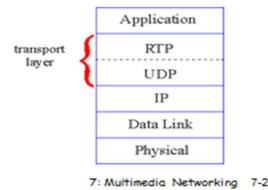
Real Time Streaming Protocol

■ RTP-Realtime Protocol

RTP runs on top of UDP

RTP libraries provide transport-layer interface that extends UDP:

- payload type identification
- packet sequence numbering
- time-stamping



7: Multimedia Networking 7-22

213

TCP Segment Header Format

Bit #	0	7	8	15	16	23	24	31
0								Destination Port
32								Sequence Number
64								Acknowledgment Number
96	Data Offset	Res		Flags				Window Size
128								Header and Data Checksum
160...								Urgent Pointer
								Options

UDP Datagram Header Format

Bit #	0	7	8	15	16	23	24	31
0								Destination Port
32								Length

214

(a)

Bit	0	4	8	12	16	20	24	28	31
0	Version	IHL	DSCP	ECN		Total Length			
32			Identification		Flags		Fragment Offset		
64	Time To Live		Protocol		Header Checksum				
96				Source IP Address					
128				Destination IP Address					
160				Options (if IHL > 5)					

Bit	0	4	8	12	16	20	24	28	31
0	Version	IHL	DSCP	ECN		Total Length			
32			Identification		Flags		Fragment Offset		
64	Time To Live		Protocol		Header Checksum				
96				Source IP Address					
128				Destination IP Address					
160				Options (if IHL > 5)					

215

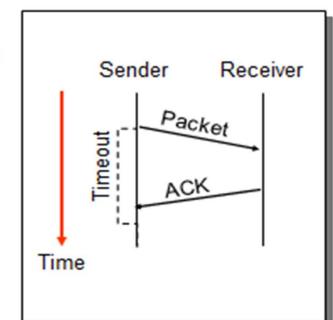
Automatic Repeat reQuest (ARQ)

• Automatic Repeat Request

- Receiver sends acknowledgment (ACK) when it receives packet
- Sender waits for ACK and timeouts if it does not arrive within some time period

• Simplest ARQ protocol

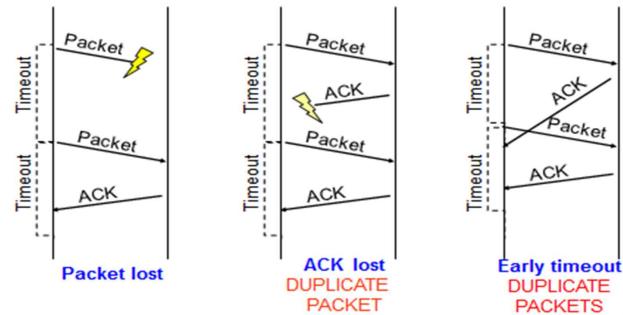
- Stop and wait
- Send a packet, stop and wait until ACK arrives



30

216

Reasons for Retransmission



217

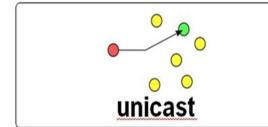


TCP



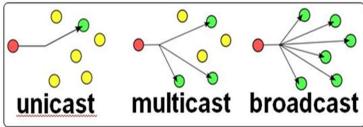
UDP

- Slower but reliable transfers
- Typical applications:
 - Email
 - Web browsing

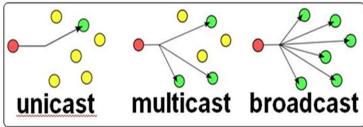


unicast

- Fast but non-guaranteed transfers ("best effort")
- Typical applications:
 - VoIP
 - Music streaming



unicast



multicast broadcast

218

Application	Speed requirement
Telephone	16 kbps
Audio conferencing	32 kbps
CD-quality audio	128–192 kbps
Digital music (QoS)	64–640 kbps
H. 261	64–2 Mbps
H. 263	<64 kbps
H. 264	1–12 Mbps
MPEG-1 video	1.2–1.5 Mbps
MPEG-2 video	4–60 Mbps
MPEG-4 video	1–20 Mbps
HDTV (compressed)	>20 Mbps
HDTV (uncompressed)	>1 Gbps
MPEG-4 video-on-demand (QoS)	250–750 kbps
Videoconferencing (QoS)	384 kbps–2 Mbps

219

Table 16.2 Comparison of YouTube media encoding options

itag value	Default container	Video resolution	Video encoding	Video profile	Video bitrate (Mbit/s)	Audio encoding	Audio bitrate (kbit/s)
5	FLV	240p	Sorenson H.263	N/A	0.25	MP3	64
6	FLV	270p	Sorenson H.263	N/A	0.8	MP3	64
13	3GP	N/A	MPEG-4 Visual	N/A	0.5	AAC	N/A
17	3GP	144p	MPEG-4 Visual	Simple	0.05	AAC	24
18	MP4	270p/360p	H.264	Baseline	0.5	AAC	96
22	MP4	720p	H.264	High	2–2.9	AAC	192
34	FLV	360p	H.264	Main	0.5	AAC	128
35	FLV	480p	H.264	Main	0.8–1	AAC	128
36	3GP	240p	MPEG-4 Visual	Simple	0.17	AAC	38
37	MP4	1080p	H.264	High	3–5.9	AAC	192
38	MP4	3072p	H.264	High	3.5–5	AAC	192
43	WebM	360p	VP8	N/A	0.5	Vorbis	128
44	WebM	480p	VP8	N/A	1	Vorbis	128
45	WebM	720p	VP8	N/A	2	Vorbis	192
46	WebM	1080p	VP8	N/A	N/A	Vorbis	192
82	MP4	360p	H.264	3D	0.5	AAC	96
83	MP4	240p	H.264	3D	0.5	AAC	96
84	MP4	720p	H.264	3D	2–2.9	AAC	152
85	FLV	520p	H.264	3D	2–2.9	AAC	152
100	WebM	360p	VP8	3D	N/A	Vorbis	128
101	WebM	360p	VP8	3D	N/A	Vorbis	192
102	WebM	720p	VP8	3D	N/A	Vorbis	192
120	FLV	720p	AVC	Main@L3.1	2	AAC	128

220

QoS

- Best Effort
- Integrated Service (IntServ) & RSVP (Resource ReSerVation Protocol)
- Differentiated Service (Diffserve)

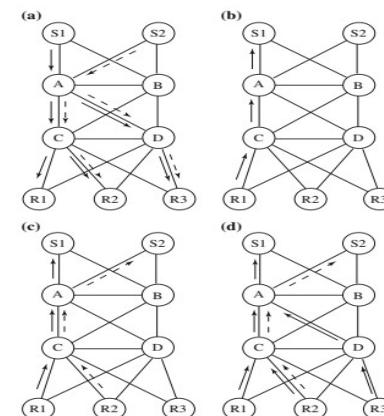
E.g: Audio PCM 44,1KHz

JPEG

MPEG-2 IPB

QoS

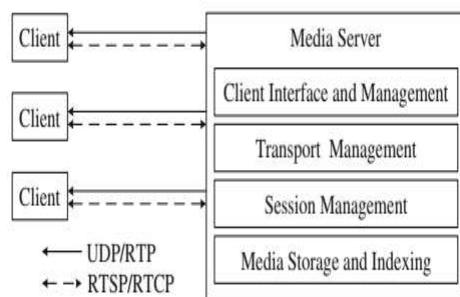
- RSVP



221

222

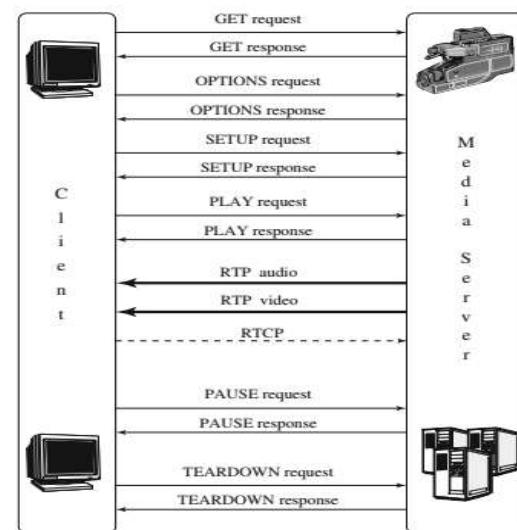
RTSP/RTP/RTCP



| A basic client/server-based media streaming system

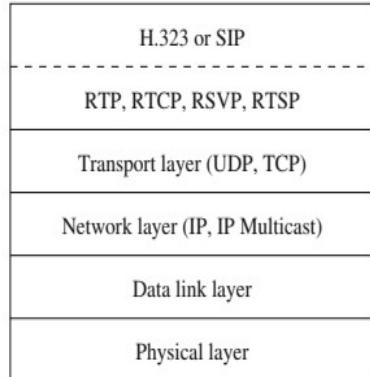
223

RTSP



224

Voice over IP



225

RTP Header

Payload Type	Sequence Number	Timestamp	Synchronization Source Identifier	Miscellaneous Fields
--------------	-----------------	-----------	-----------------------------------	----------------------

RTP Header

Payload Type (7 bits): Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs receiver via payload type field.

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3, GSM, 13 kbps
- Payload type 7, LPC, 2.4 kbps
- Payload type 26, Motion JPEG
- Payload type 31, H.261
- Payload type 33, MPEG2 video

Sequence Number (16 bits): Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence.

7-226

RTP Header (2)

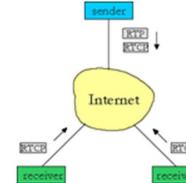


- r **Timestamp field (32 bytes long):** sampling instant of first byte in this RTP data packet
 - m for audio, timestamp clock typically increments by one for each sampling period
 - for example, each 125 usecs for 8 KHz sampling clock
 - m if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active.
 - m Timestamp clock continues to increase at constant rate when source is inactive.
- r **SSRC field (32 bits long):** identifies source of RTP stream
 - m Each stream in RTP session should have distinct SSRC

7-227

Real-time Control Protocol

RTCP - Continued



- each RTP session: typically a single multicast address:
 - all RTP /RTCP packets belonging to session use multicast address.
- RTP, RTCP packets distinguished from each other via distinct port numbers.
- to limit traffic, each participant reduces RTCP traffic as number of conference participants increases

7: Multimedia Networking 7-28

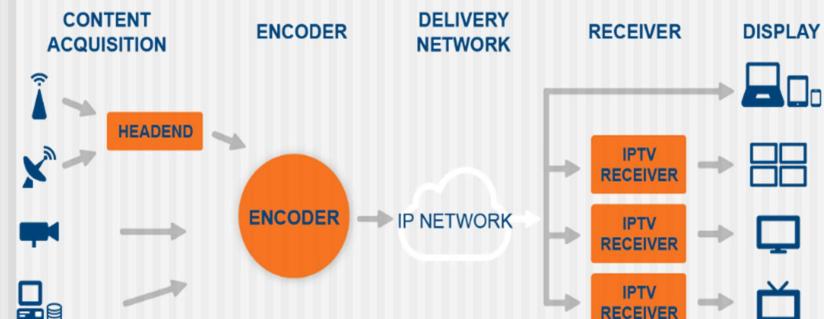
7-228

Tổng quan về công nghệ IPTV

- Khái niệm
- Đặc điểm của IPTV
- Các dịch vụ của IPTV

229

Khái niệm IPTV (Internet Protocol TV)



230

Đặc điểm IPTV

- Hỗ trợ truyền hình tương tác
- Không phụ thuộc thời gian
- Tăng tính cá nhân
- Yêu cầu về băng thông thấp
- Khả năng truy nhập trên nhiều loại thiết bị

231

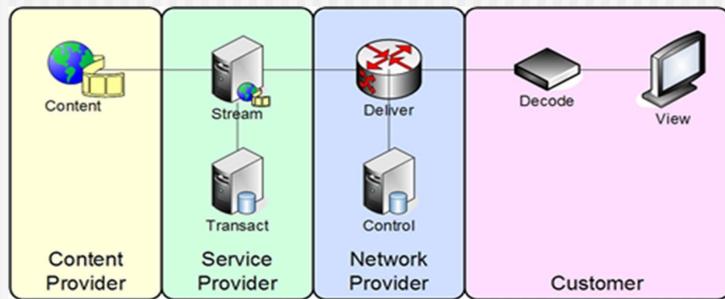
Dịch vụ của IPTV

- Dịch vụ Video quảng bá/ Video theo yêu cầu
- Dịch vụ Phát thanh quảng bá
- Dịch vụ âm nhạc quảng bá/ theo yêu cầu
- Dịch vụ giải trí (karaoke, game..) theo yêu cầu
- Tiếp thị /quảng cáo...

232

Kiến trúc mạng IPTV

■ Mô hình kiến trúc IPTV

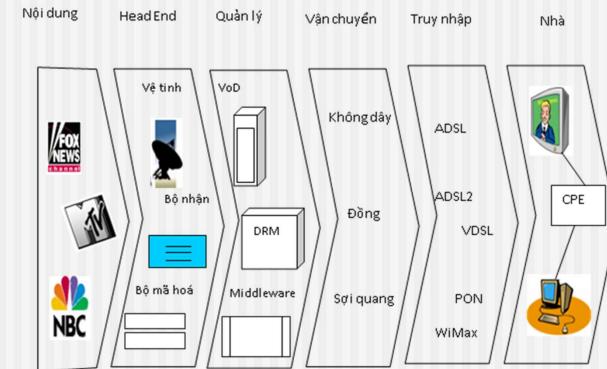


* Ref: ITU TSB IPTV Consultation meeting (Doc. Iptv018e and 20e)

233

Kiến trúc mạng IPTV

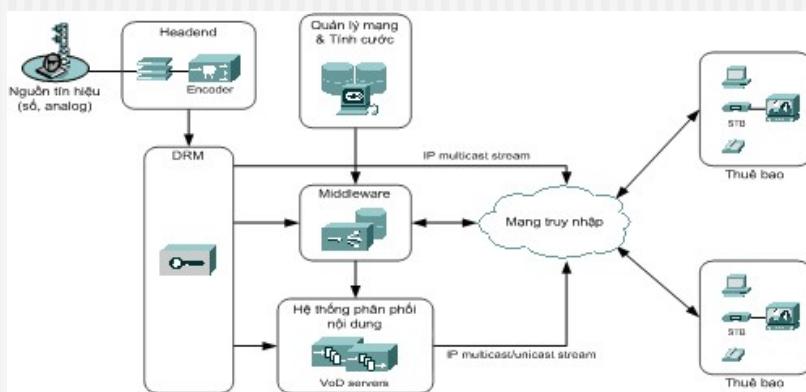
■ Mô hình kiến trúc IPTV



234

Kiến trúc mạng IPTV

■ Mô hình kiến trúc IPTV



235

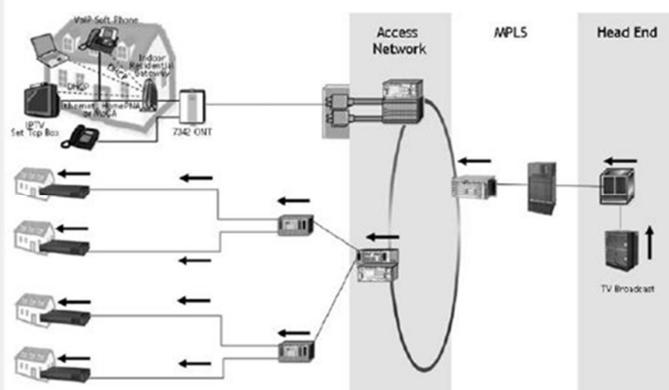
Các giao thức truyền dẫn trong IPTV

- Cơ chế Multicast
- Cơ chế Unicast
- Cơ chế Broadcast
- Giao thức UDP và RTP/RTCP

236

Các giao thức truyền dẫn trong IPTV

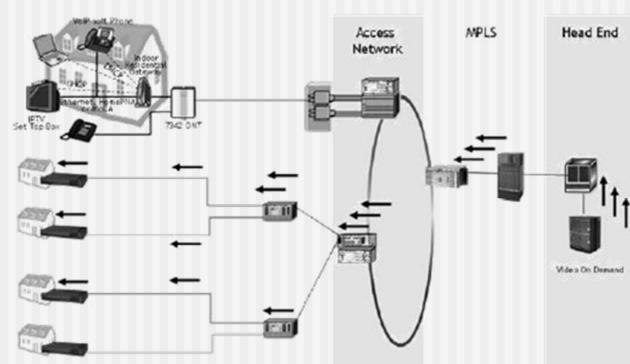
■ Cơ chế Multicast



237

Các giao thức truyền dẫn trong IPTV

■ Cơ chế Unicast



238

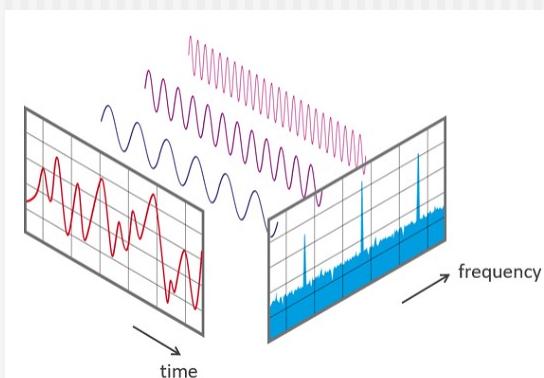
Ôn tập

1. Biến đổi DCT
2. DCT vs DFT)
3. Mã hóa JPEG phân cấp.
4. Khái niệm mã hóa âm thanh trong miền tần số, miền thời gian
5. Thuật toán tìm vector chuyển động

239

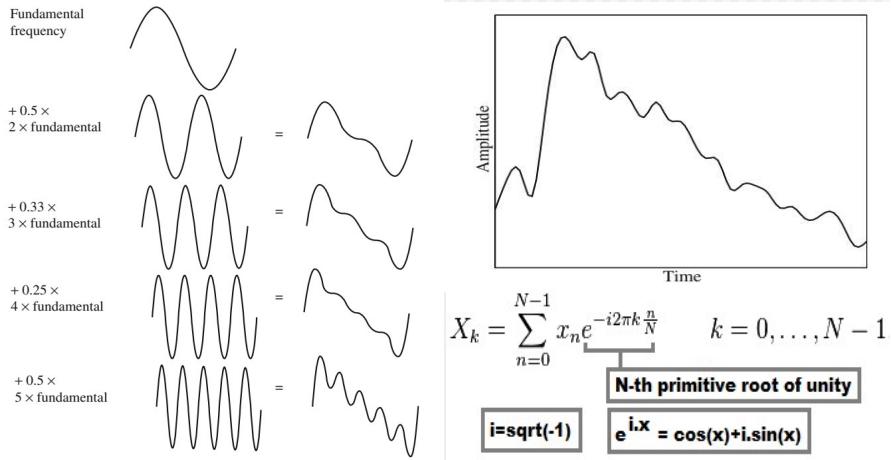
Ôn tập

■ DFT vs DCT



240

DFT



241

DCT

1D Discrete Cosine Transform (1D DCT)

$$F(u) = \frac{C(u)}{2} \sum_{i=0}^7 \cos \frac{(2i+1)u\pi}{16} f(i),$$

where $i = 0, 1, \dots, 7$, $u = 0, 1, \dots, 7$, and the constant $C(u)$

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

242

DCT

1D Inverse Discrete Cosine Transform (1D-IDCT)

$$\tilde{f}(i) = \sum_{u=0}^7 \frac{C(u)}{2} \cos \frac{(2i+1)u\pi}{16} F(u),$$

where $i = 0, 1, \dots, 7$, $u = 0, 1, \dots, 7$, and the constant $C(u)$

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

243

DCT

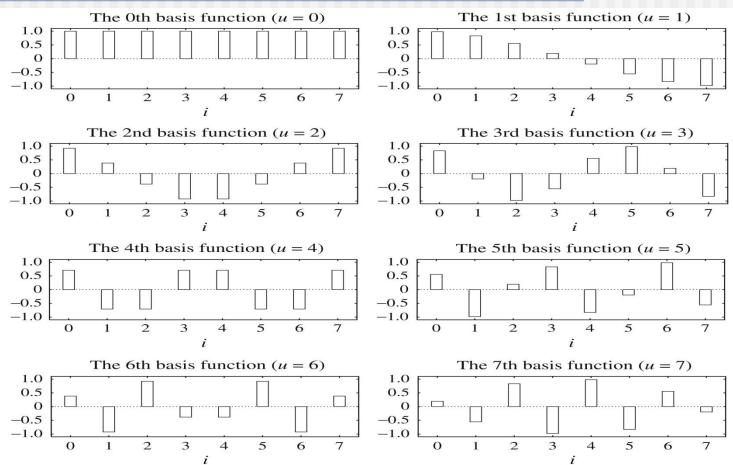


Fig. 8.6 The 1D DCT basis functions

244

DCT

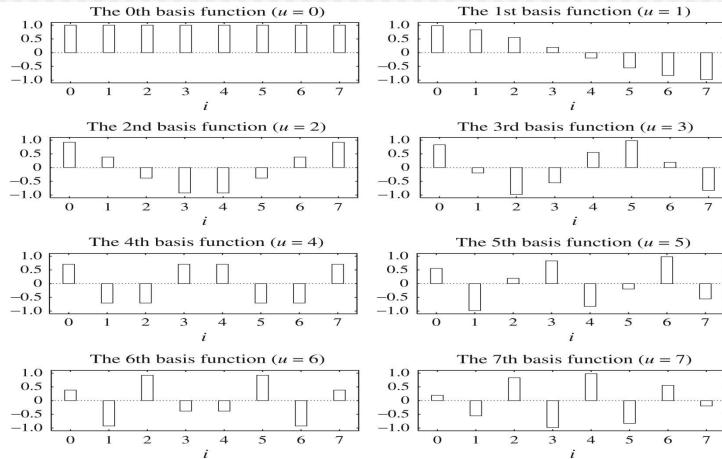


Fig. 8.6 The 1D DCT basis functions

245

DCT

$$\begin{aligned} F_1(0) &= \frac{\sqrt{2}}{2 \cdot 2} \cdot (1 \cdot 100 + 1 \cdot 100 + 1 \cdot 100 + 1 \cdot 100 \\ &\quad + 1 \cdot 100 + 1 \cdot 100 + 1 \cdot 100 + 1 \cdot 100) \\ &\approx 283 \end{aligned}$$

$$\begin{aligned} F_1(1) &= \frac{1}{2} \cdot (\cos \frac{\pi}{16} \cdot 100 + \cos \frac{3\pi}{16} \cdot 100 + \cos \frac{5\pi}{16} \cdot 100 + \cos \frac{7\pi}{16} \cdot 100 \\ &\quad + \cos \frac{9\pi}{16} \cdot 100 + \cos \frac{11\pi}{16} \cdot 100 + \cos \frac{13\pi}{16} \cdot 100 + \cos \frac{15\pi}{16} \cdot 100) \\ &= 0. \end{aligned}$$

246

DCT

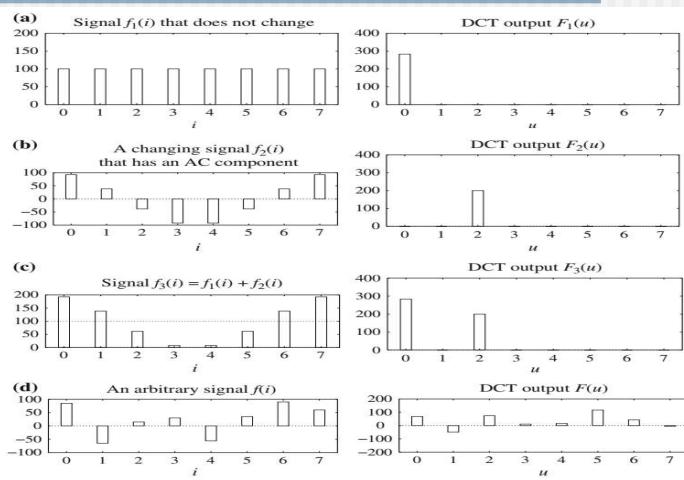
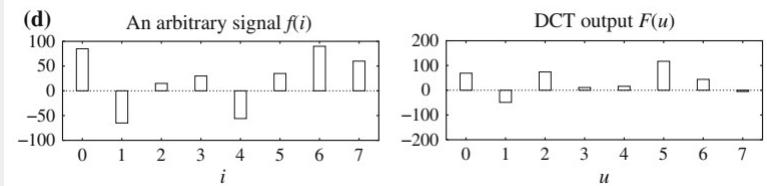


Fig. 8.7 Examples of 1D Discrete Cosine Transform: **a** a DC signal $f_1(i)$; **b** an AC signal $f_2(i)$; **c** $f_3(i) = f_1(i) + f_2(i)$; and **d** an arbitrary signal $f(i)$

247

I- DCT

$$\begin{array}{ll} f(i)(i = 0..7) : & 85 \quad -65 \quad 15 \quad 30 \quad -56 \quad 35 \quad 90 \quad 60 \\ F(u)(u = 0..7) : & 69 \quad -49 \quad 74 \quad 11 \quad 16 \quad 117 \quad 44 \quad -5. \end{array}$$



248

2D-DCT

2D Discrete Cosine Transform (2D DCT)

$$F(u, v) = \frac{C(u) C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i, j),$$

where $i, j, u, v = 0, 1, \dots, 7$, and the constants $C(u)$ and $C(v)$ are

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

249

2D-IDCT

$$\tilde{f}(i, j) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u) C(v)}{4} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} F(u, v)$$

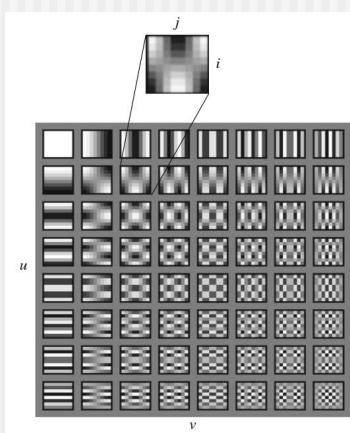
where $i, j, u, v = 0, 1, \dots, 7$, and the constants $C(u)$ and $C(v)$ are

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

250

2D-DCT

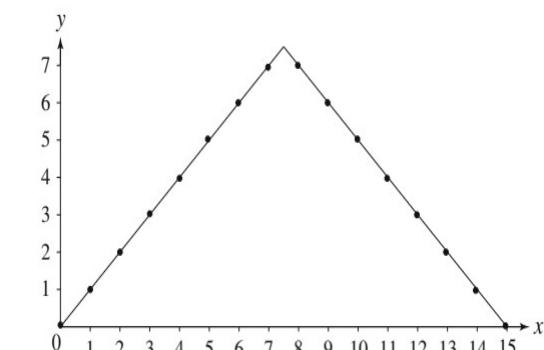
$$\cos \frac{(2i+1) \cdot u\pi}{16} \cdot \cos \frac{(2j+1) \cdot v\pi}{16},$$



251

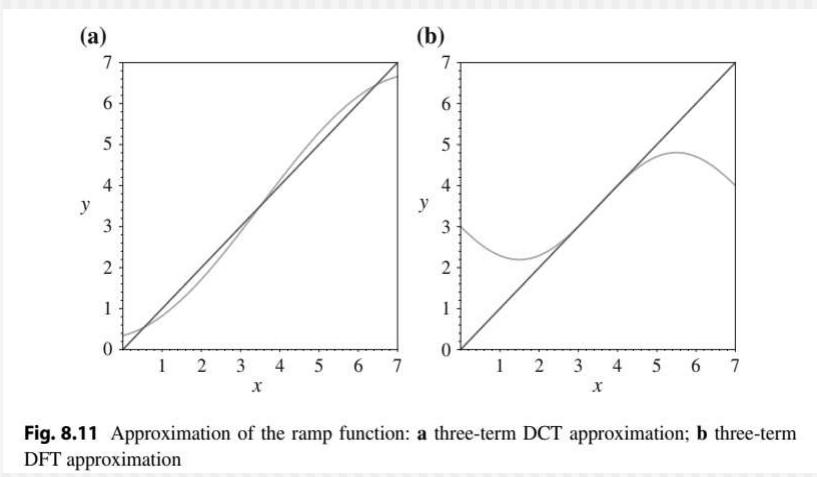
DFT vs DCT

Ramp	DCT	DFT
0	9.90	28.00
1	-6.44	-4.00
2	0.00	9.66
3	-0.67	-4.00
4	0.00	4.00
5	-0.20	-4.00
6	0.00	1.66
7	-0.51	-4.00



252

DFT vs DCT



253

JPEG vs DCT



An 8×8 block from the Y image of 'Lena'

$f(i,j)$	$F(u,v)$
32 -6 -1 0 0 0 0 0	512 66 -10 0 0 0 0 0
-1 0 1 0 0 0 0 0	-12 0 0 0 0 0 0 0
-1 0 0 0 0 0 0 0	-14 0 16 0 0 0 0 0
0 0 0 0 0 0 0 0	-14 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
$\hat{F}(u,v)$	$\tilde{F}(u,v)$
199 190 191 186 182 178 177 176	1 6 -2 2 7 -3 -2 -1
201 198 196 192 188 183 180 178	-1 4 2 -4 1 -1 -2 -3
203 203 202 200 195 189 183 180	0 -3 -2 -5 5 -2 2 -5
202 203 204 203 198 191 183 179	-2 -3 -4 -3 -1 -4 4 8
200 201 202 201 196 189 182 177	0 4 -2 -1 -1 -1 5 -2
200 200 199 197 192 186 181 177	0 0 1 3 8 4 6 -2
204 202 199 195 190 186 183 181	1 -2 0 5 1 1 4 -6
207 204 200 194 190 187 185 184	3 -4 0 6 -2 -2 2 2
$\tilde{f}(i,j)$	$e(i,j) = f(i,j) - \tilde{f}(i,j)$

254

JPEG vs DCT



Another 8×8 block from the Y image of 'Lena'

70 70 100 70 87 87 150 187	-80 -40 89 -73 44 32 53 -3
85 100 96 79 87 154 87 113	-135 -59 -26 6 14 -3 -13 -28
100 85 116 79 70 87 86 196	47 -76 66 -3 -108 -78 33 59
136 69 87 200 79 71 117 96	-2 10 -18 0 33 11 -21 1
161 70 87 200 103 71 96 113	-1 -9 -22 8 32 65 -36 -1
161 123 147 133 113 113 85 161	5 -20 28 -46 3 24 -30 24
146 147 175 100 103 103 163 187	6 -20 37 -28 12 -35 33 17
156 146 189 70 113 161 163 197	-5 -23 33 -30 17 -5 -4 20
$f(i,j)$	$F(u,v)$
-5 -4 9 -5 2 1 1 0	-80 -44 90 -80 48 40 51 0
-11 -5 -2 0 1 0 0 -1	-132 -60 -28 0 26 0 0 -55
3 -6 4 0 -3 -1 0 1	42 -78 64 0 -120 -57 0 56
0 1 -1 0 1 0 0 0	0 17 -22 0 51 0 0 0
0 0 -1 0 0 1 0 0	0 0 -37 0 0 109 0 0
0 -1 1 -1 0 0 0 0	0 -35 55 -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
$\hat{F}(u,v)$	$\tilde{F}(u,v)$
70 60 106 94 62 103 146 176	0 10 -6 -24 25 -16 4 11
85 101 85 75 102 127 93 144	0 -1 11 4 -15 27 -6 -31
98 99 92 102 74 98 89 167	2 -14 24 -23 -4 -11 -3 29
132 53 111 180 55 70 106 145	4 16 -24 20 24 1 11 -49
173 57 114 207 111 89 84 90	-12 13 -27 -7 -8 -18 12 23
164 123 131 135 133 92 85 162	-3 0 16 -2 -20 21 0 -1
141 159 169 73 106 101 149 224	5 -12 6 27 -3 2 14 -37
150 141 195 79 107 147 210 153	6 5 -6 -9 6 14 -47 44
$\tilde{f}(i,j)$	$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

255

JPEG

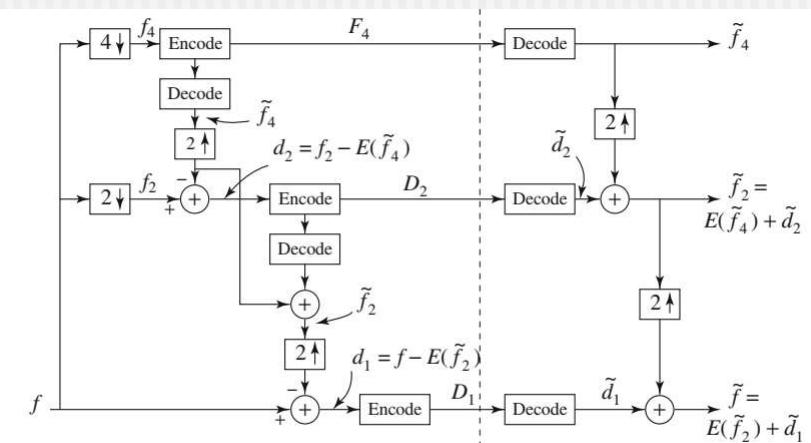


Fig. 9.5 Block diagram for Hierarchical JPEG

256

JPEG

Algorithm 9.1 (Three-Level Hierarchical JPEG Encoder).

1. **Reduction of image resolution.** Reduce resolution of the input image f (e.g., 512×512) by a factor of 2 in each dimension to obtain f_2 (e.g., 256×256). Repeat this to obtain f_4 (e.g., 128×128).
2. **Compress low-resolution image f_4 .** Encode f_4 using any other JPEG method (e.g., Sequential, Progressive) to obtain F_4 .
3. **Compress difference image d_2 .** (a) Decode F_4 to obtain \tilde{f}_4 . Use any interpolation method to expand \tilde{f}_4 to be of the same resolution as f_2 and call it $E(\tilde{f}_4)$. (b) Encode difference $d_2 = f_2 - E(\tilde{f}_4)$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_2 .
4. **Compress difference image d_1 .** (a) Decode D_2 to obtain \tilde{d}_2 ; add it to $E(\tilde{f}_4)$ to get $\tilde{f}_2 = E(\tilde{f}_4) + \tilde{d}_2$, which is a version of f_2 after compression and decompression. (b) Encode difference $d_1 = f_2 - E(\tilde{f}_2)$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_1 .

257

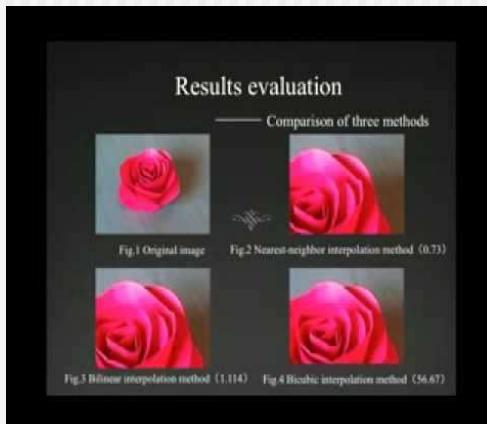
JPEG

Algorithm 9.2 (Three-Level Hierarchical JPEG Decoder).

1. **Decompress the encoded low-resolution image F_4 .** Decode F_4 using the same JPEG method as in the encoder, to obtain \tilde{f}_4 .
2. **Restore image \tilde{f}_2 at the intermediate resolution.** Use $E(\tilde{f}_4) + \tilde{d}_2$ to obtain \tilde{f}_2 .
3. **Restore image \tilde{f} at the original resolution.** Use $E(\tilde{f}_2) + \tilde{d}_1$ to obtain \tilde{f} .

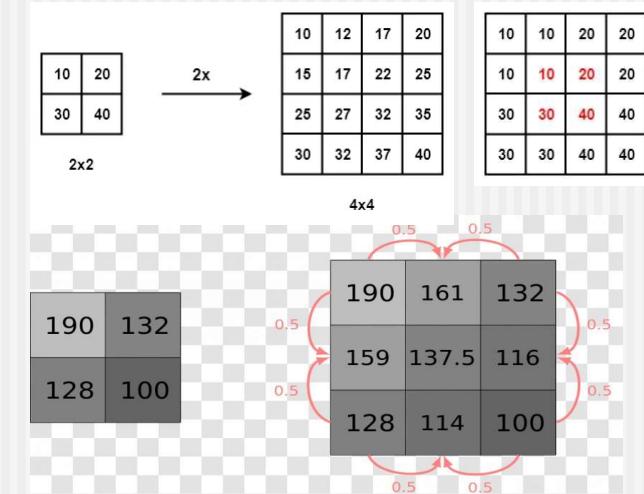
258

JPEG



259

JPEG



260

Video



Fig. 5.7 Connectors for typical analog display interfaces. From left to right: Component video, Composite video, S-video, and VGA



Fig. 5.8 Connectors of different digital display interfaces. From left to right: DVI, HDMI, DisplayPort

261

Motion Vector Searching

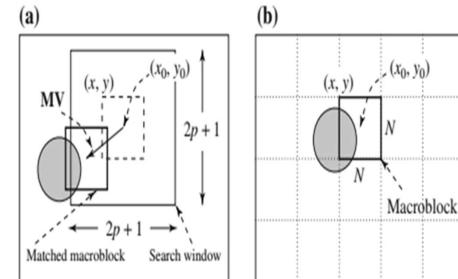


Fig. 10.1 Macroblocks and motion vector in video compression: **a** reference frame; **b** target frame

Motion Vector Searching

Procedure 10.1 (Motion vector: Sequential search)

```

BEGIN
  min_MAD = LARGE_NUMBER; /* Initialization */
  for i = -p to p
    for j = -p to p
      {
        cur_MAD = MAD(i, j);
        if cur_MAD < min_MAD
          {
            min_MAD = cur_MAD;
            u = i; /* Get the coordinates for MV. */
            v = j;
          }
      }
END
  
```

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|, \quad (10.1)$$

where N is the size of the macroblock.

263

Motion Vector Searching

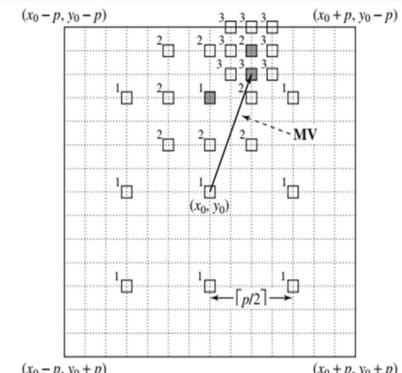


Fig. 10.2 2D Logarithmic search for motion vectors

264

Motion Vector Searching

Procedure 10.2 (Motion vector: 2D-Logarithmic search)

```

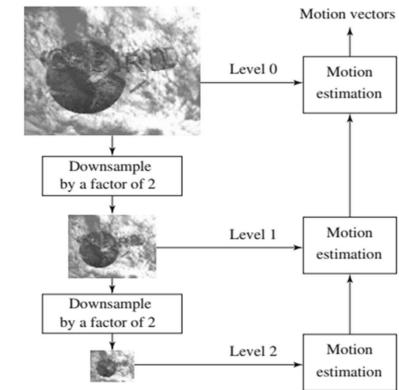
BEGIN
    offset =  $\lceil \frac{p}{2} \rceil$ ;
    Specify nine macroblocks within the search window in the Reference frame,
    they are centered at  $(x_0, y_0)$  and separated by offset horizontally and/or vertically;
    WHILE last ≠ TRUE
    {
        Find one of the nine specified macroblocks that yields the minimum MAD;
        if offset = 1 then last = TRUE;
        offset =  $\lceil \text{offset}/2 \rceil$ ;
        Form a search region with the new offset and new center found;
    }
END

```

265

Motion Vector Searching

Fig. 10.3 A three-level hierarchical search for motion vectors



266

Motion Vector Searching

Procedure 10.3 (Motion vector: Hierarchical search)

```

BEGIN
    // Get macroblock center position at the lowest resolution level k, e.g., level 2.
     $x_0^k = x_0^0/2^k$ ;  $y_0^k = y_0^0/2^k$ ;
    Use Sequential (or 2D Logarithmic) search method to get initial estimated
     $\mathbf{MV}(u^k, v^k)$  at level k;
    WHILE last ≠ TRUE
    {
        Find one of the nine macroblocks that yields minimum MAD
        at level  $k - 1$  centered at
         $(2(x_0^k + u^k) - 1 \leq x \leq 2(x_0^k + u^k) + 1, 2(y_0^k + v^k) - 1 \leq y \leq 2(y_0^k + v^k) + 1)$ ;
        if  $k = 1$  then last = TRUE;
         $k = k - 1$ ;
        Assign  $(x_0^k, y_0^k)$  and  $(u^k, v^k)$  with the new center location and motion
        vector;
    }
END

```

267

Motion Vector Searching

As an example, let's assume the video has a resolution of 720×480 and a frame rate of 30 fps; also, assume $p = 15$ and $N = 16$. The number of operations needed for each motion vector search is thus

$$(2p+1)^2 \cdot N^2 \cdot 3 = 31^2 \times 16^2 \times 3.$$

Considering that a single image frame has $\frac{720 \times 480}{N \cdot N}$ macroblocks, and 30 frames each second, the total operations needed per second is

$$\begin{aligned} \text{OPS_per_second} &= (2p+1)^2 \cdot N^2 \cdot 3 \cdot \frac{720 \times 480}{N \cdot N} \cdot 30 \\ &= 31^2 \times 16^2 \times 3 \times \frac{720 \times 480}{16 \times 16} \times 30 \approx 29.89 \times 10^9. \end{aligned}$$

This would certainly make real-time encoding of this video difficult.

268

Motion Vector Searching

Using the same example as in the previous subsection, the total operations per second drop to

$$\begin{aligned}\text{OPS_per_second} &= (8 \cdot (\lceil \log_2 p \rceil + 1) + 1) \cdot N^2 \cdot 3 \cdot \frac{720 \times 480}{N \cdot N} \cdot 30 \\ &= (8 \cdot \lceil \log_2 15 \rceil + 9) \times 16^2 \times 3 \times \frac{720 \times 480}{16 \times 16} \times 30 \\ &\approx 1.25 \times 10^9.\end{aligned}$$

The total number of macroblocks processed each second is still $\frac{720 \times 480}{N \cdot N} \times 30$. However, the operations needed for each macroblock are reduced to

$$\left[\left(2 \lceil \frac{p}{4} \rceil + 1 \right)^2 \left(\frac{N}{4} \right)^2 + 9 \left(\frac{N}{2} \right)^2 + 9N^2 \right] \times 3.$$

Hence,

$$\begin{aligned}\text{OPS_per_second} &= \left[\left(2 \lceil \frac{p}{4} \rceil + 1 \right)^2 \left(\frac{N}{4} \right)^2 + 9 \left(\frac{N}{2} \right)^2 + 9N^2 \right] \\ &\quad \times 3 \times \frac{720 \times 480}{N \cdot N} \times 30 \\ &= \left[\left(\frac{9}{4} \right)^2 + \frac{9}{4} + 9 \right] \times 16^2 \times 3 \times \frac{720 \times 480}{16 \times 16} \times 30 \\ &\approx 0.51 \times 10^9.\end{aligned}$$

269

Motion Vector Searching

Table 10.1 Comparison of computational cost of motion vector search methods according to the examples

Search method	OPS_per_second for 720 × 480 at 30 fps	
	p = 15	p = 7
Sequential search	29.89×10^9	7.00×10^9
2D logarithmic search	1.25×10^9	0.78×10^9
Three-level hierarchical search	0.51×10^9	0.40×10^9

270