


PHƯƠNG PHÁP NÉN ẢNH

Nén ảnh

- Nén Ảnh là gì?
- Mục đích của nén ảnh
- Các khái niệm
- Nén Ảnh
- Các chuẩn ảnh nén hiện nay

NÉN ẢNH LÀ GÌ?



Tại sao
cần phải
nén ?

Tại sao
ta
có thể
nén ảnh
?

Các **phương**
pháp thường
dùng để nén
ảnh ?

NÉN ẢNH LÀ GÌ?

- Nén là quá trình làm giảm thông tin dư thừa trong dữ liệu.

1600x1200



5,7MB

Nén



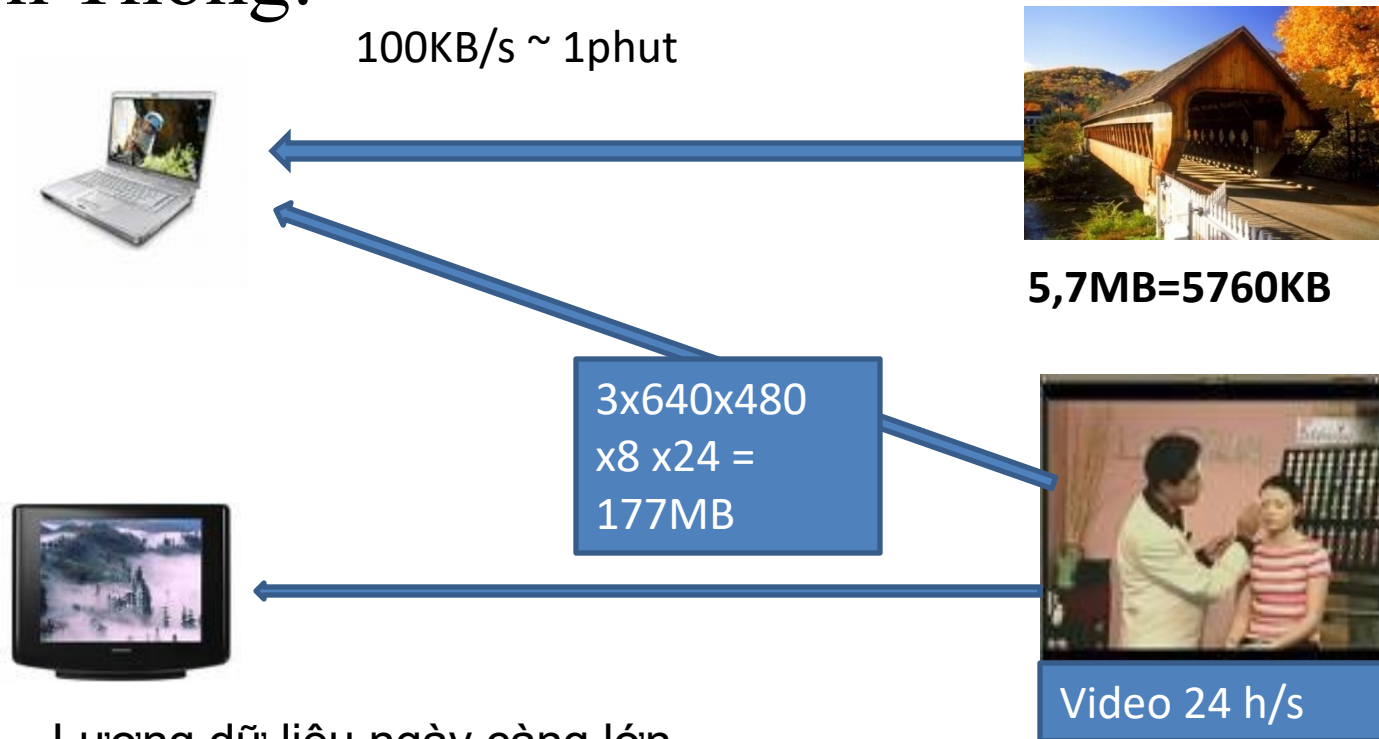
1600x1200



406KB

MỤC ĐÍCH CỦA VIỆC NÉN ẢNH

- Truyền Thông:



Lượng dữ liệu ngày càng lớn

Các yêu cầu về lưu trữ và truyền thông:

- DVD + Video conference + Printer

Tốc độ truyền dữ liệu cinema không nén: 1Gbps

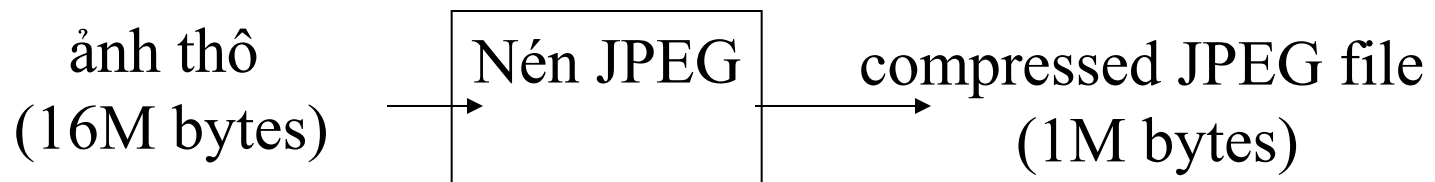
Nén ảnh

- Tại sao lại cần nén ảnh?

-Ví dụ: **camera số (4Mpixel)**

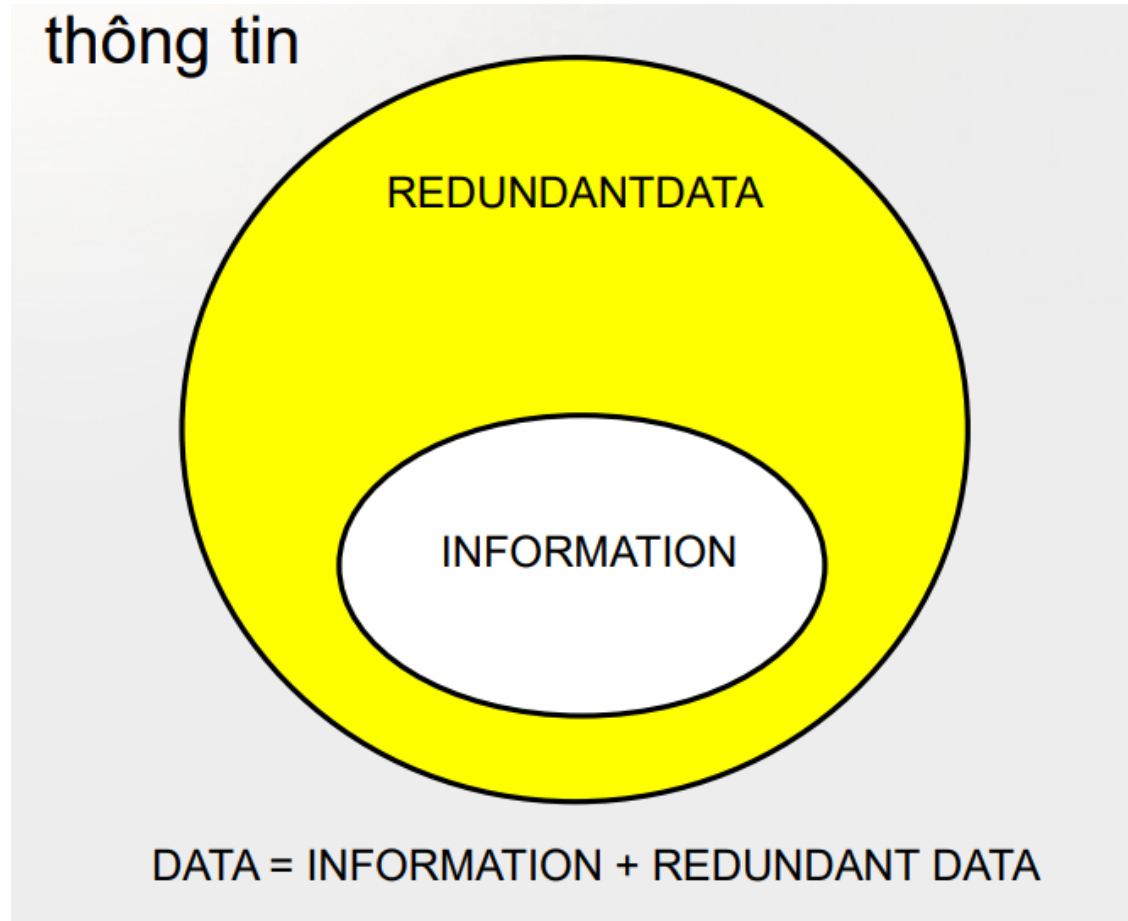
Dữ liệu ban đầu – 24bits, 5.3M pixels → 16M bytes

256M memory card (\$30-50) → 16 pictures



compression ratio=16 → 256 pictures

Nén ảnh



Nguyên lý của nén ảnh

Ví dụ

High-definition television (HDTV) generates images with a resolution of 1125 horizontal TV lines interlaced (where every other line is painted on the tube face in each of two fields, each field being $1/60$ th of a second in duration). The width-to-height aspect ratio of the images is 16:9. The fact that the horizontal lines are distinct fixes the vertical resolution of the images. A company has designed an image capture system that generates digital images from HDTV images. The resolution of each TV (horizontal) line in their system is in proportion to vertical resolution, with the proportion being the width-to-height ratio of the images. Each pixel in the color image has 24 bits of intensity resolution, 8 pixels each for a red, a green, and a blue image. These three “primary” images form a color image. How many bits would it take to store a 2-hour HDTV program?

Ví dụ

The width-to-height ratio is 16/9 and the resolution in the vertical direction is 1125 lines (or, what is the same thing, 1125 pixels in the vertical direction). It is given that the resolution in the horizontal direction is in the 16/9 proportion, so the resolution in the vertical direction is $(1125) \times (16/9) = 2000$ pixels per line. The system “paints” a full 1125×2000 , 8-bit image every 1/30 sec for each of the red, green, and blue component images. There are 7200 sec in two hours, so the total digital data generated in this time interval is $(1125)(2000)(8)(30)(3)(7200) = 1.166 \times 10^{13}$ bits, or 1.458×10^{12} bytes (i.e., about 1.5 terrabytes).



Nén để giảm không gian lưu trữ, tiết kiệm băng thông

CÁC KHÁI NIỆM

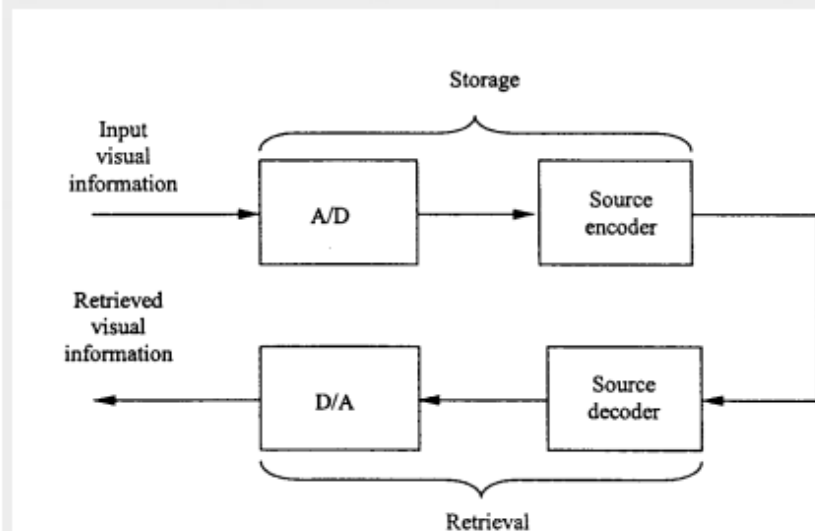
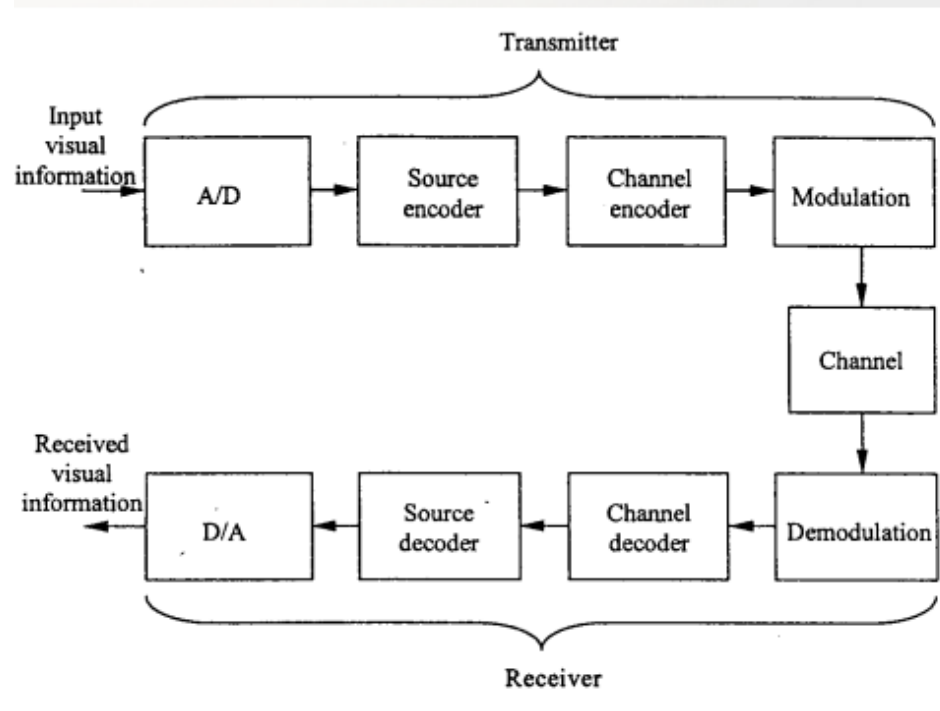
Một số tham số được dùng để đánh giá chất lượng của ảnh nén

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i, j) - K(i, j)||^2$$

Peak Signal-to-Noise Ratio(PSNR):

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

Mô hình chung của nén ảnh trong các hệ thống truyền và lưu trữ dữ liệu



Coder: (en)coder + decoder = codec

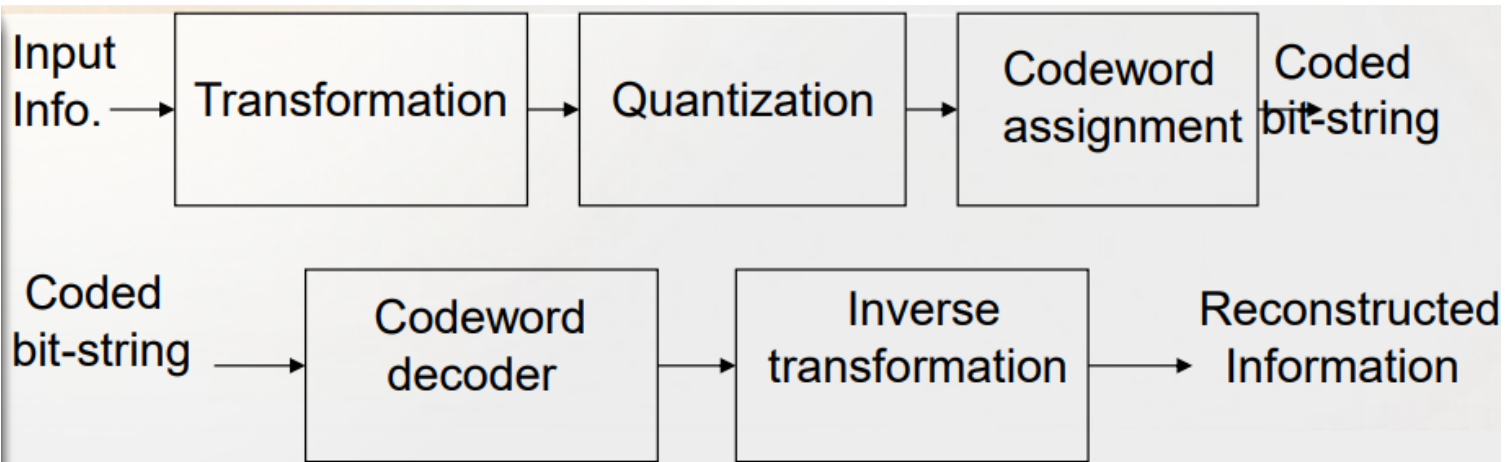
Source encoder: removes redundancy

Channel encoder: adds redundancy

A/D, D/A, en/decryption optional

Only deal with the source coder

Bộ mã hóa nguồn tin



- Transformation: new representation of data
Differential coding, transform coding (MM2)
- Quantization: In-reversible process => lossy coding
- Codeword assignment (entropy coding): Info. Theory: Huffman, run length, arithmetic, dictionary coding

CÁC KHÁI NIỆM

- Tỷ số nén: Là đặc trưng của mọi phương pháp nén.

$$C_R = \frac{n_1}{n_2}.$$

- Dư thừa dữ liệu:

- Sự phân bố kí tự

$$R_D = 1 - \frac{1}{C_R}$$

- Sự lặp lại ký tự.

- Sự phân bố của các chuỗi ký tự .

- Dư thừa trong pixel (Interpixel Redundancy)

- Dư thừa tâm lý thị giác

DƯ THỪA DỮ LIỆU

- Sự phân bố ký tự
 - Một số ký tự (pixel) xuất hiện với tần suất lớn hơn so với các ký tự khác trong dữ liệu gốc (ảnh) .Ta có thể thay thế các ký tự này bằng từ mã nhị phân ít bit hơn và các ký tự xuất hiện nhiều hơn bằng từ mã nhị phân có nhiều bit hơn.
 - Dùng mã hóa Huffman để mã hóa loại dư thừa này.

DƯ THỪA DỮ LIỆU

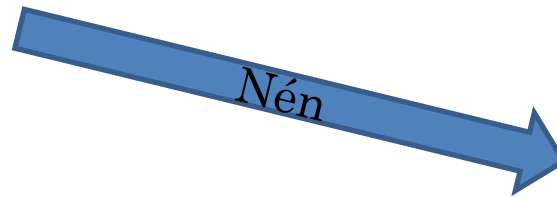
- Sự lặp lại các ký tự
 - Một chuỗi các ký tự (bit 1 hoặc 0) được lặp lại nhiều lần. Ta có thể mã hóa chuỗi lặp đó bằng ít bit hơn. Đây chính là nguyên tắc hoạt động của mã hóa RLC (Run-Length Coding)

DƯ THỪA DỮ LIỆU

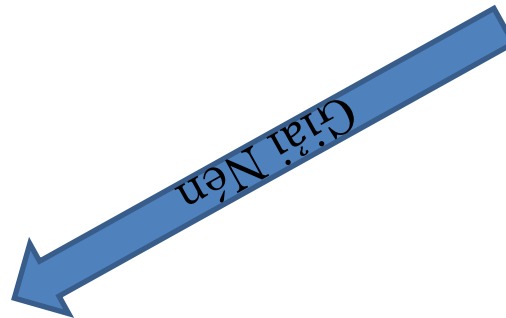
- Sự lặp lại của các chuỗi ký tự:
 - Một số chuỗi ký tự (pixel) có tần suất xuất hiện tương đối cao.
 - Có thể mã hóa các chuỗi đó bằng từ mã ít bit hơn .
 - Để xử lý loại dư thừa này ta sử dụng phương pháp mã hóa LWZ (mã hóa kiểu từ điển) .

Nén tổn thất và nén không tổn thất

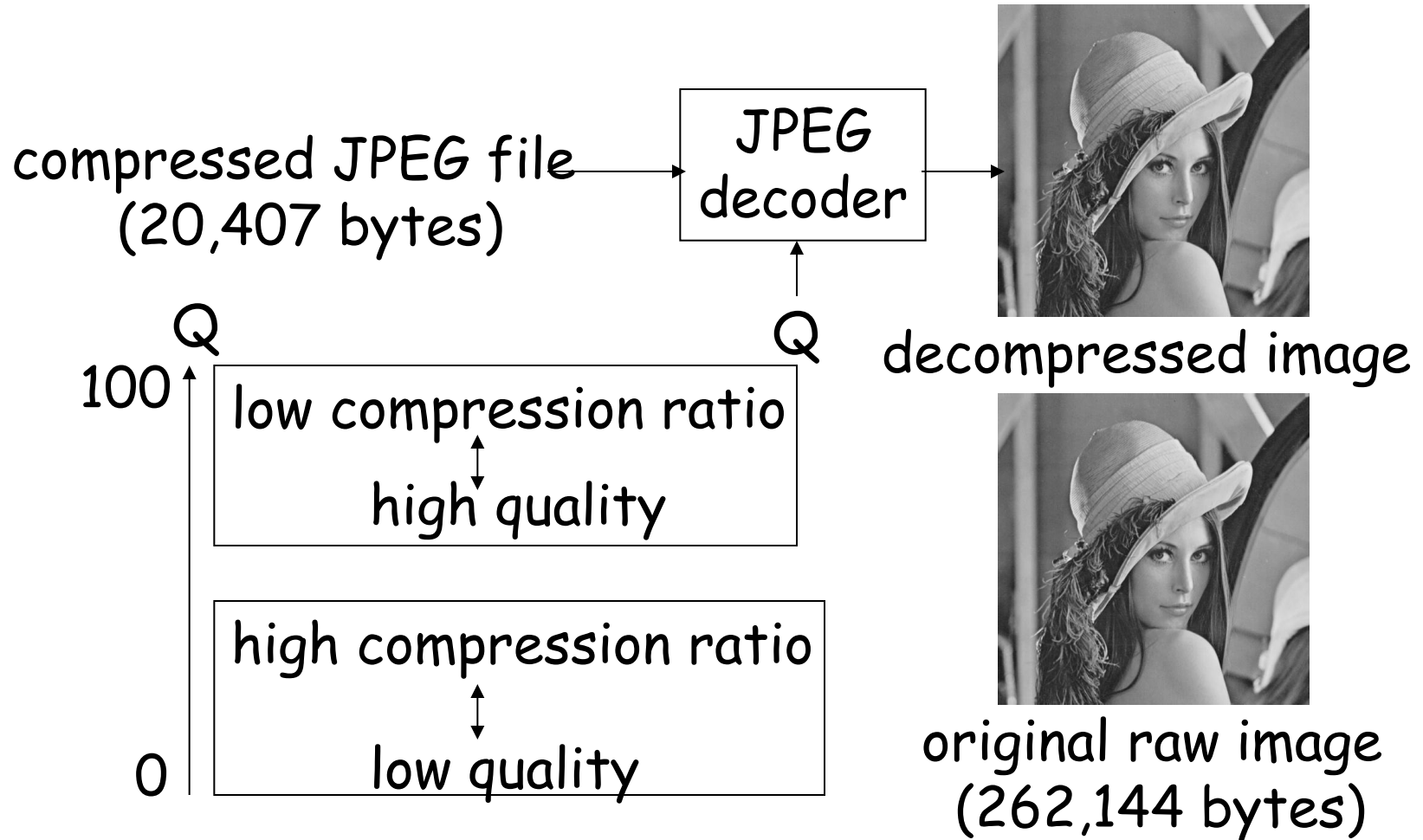
- Nén mất thông tin và không mất thông tin
 - Nén mất thông tin:



ẢNH NÉN

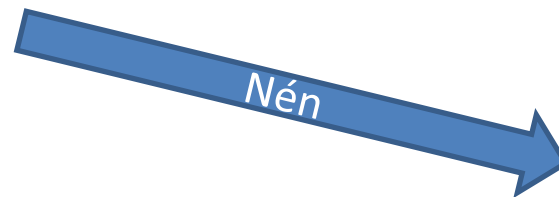


Nén ảnh có tổn thất

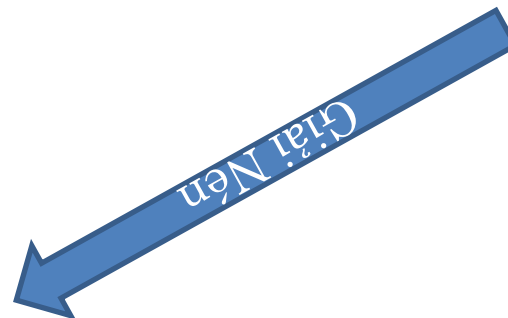


Nén tổn thất và nén không tổn thất

- Nén mất thông tin và không mất thông tin
 - Nén không mất thông tin:



ẢNH NÉN



Nén ảnh không tổn thất

- Định nghĩa
 - Ảnh sau khi giải nén giống hoàn toàn với ảnh gốc (zero error)
- Hệ số nén (Compression ratio)
 - Phụ thuộc lớn vào kiểu ảnh và nội dung ảnh

ảnh nhân tạo	ảnh tự nhiên
>10	1~3
- Ứng dụng
 - Lưu trữ và truyền các ảnh y học

Nén ảnh không tổn thất

Nén dữ liệu không tổn thất

- Trong nén dữ liệu không tổn thất, tính toàn vẹn của dữ liệu được bảo toàn.
- Dữ liệu gốc và dữ liệu sau khi nén và giải nén giống nhau hoàn toàn vì trong các phương pháp này, thuật toán nén và giải nén chính xác là biến đổi ngược của nhau.
- Dữ liệu dư thừa được loại bỏ trong quá trình nén và được thêm vào trong quá trình giải nén.
- Nén dữ liệu không tổn thất thường được sử dụng khi chúng ta không muốn mất mát thông tin.

Các kỹ thuật nén ảnh không tổn thất phổ biến

- WinZip
 - Dựa trên thuật toán Lempel-Ziv được phát minh cách đây 30 năm
- GIF (Graphic Interchange Format)
 - Dựa trên thuật toán LZ nâng cao, tạo ra bởi Welch năm 1983
- PNG (Portable Network Graphics)

Sơ đồ khối hệ thống nén ảnh tiêu biểu



- **Bộ chuyển đổi:** thường dùng phép biến đổi Cosin rời rạc để tập trung năng lượng tín hiệu vào một số lượng nhỏ các hệ số khai triển để thực hiện phép nén hiệu quả hơn là dùng tín hiệu nguyên thủy.
- **Bộ lượng tử hoá:** tạo ra một lượng ký hiệu giới hạn cho ảnh nén với hai kỹ thuật: lượng tử vô hướng (thực hiện lượng tử hoá cho từng phần dữ liệu) và lượng tử vectơ (thực hiện lượng tử hoá một lần một khối dữ liệu). Quá trình này không thuận nghịch.
- **Bộ mã hoá:** gán một từ mã, một dòng bit nhị phân cho mỗi ký hiệu.

Các kỹ thuật mã hoá entropy (mã hóa không tổn thất)

- Mã hoá loạt dài chạy (**RLC-Run Length Coding**): các chuỗi điểm ảnh có cùng độ chói (mức màu) sẽ được mã hoá bằng cặp thông tin (độ chói, chiều dài chuỗi).
- Mã hoá bằng cách **loại bỏ trùng lặp**: các chuỗi đặc biệt được thay thế bằng cờ và số đếm lặp.
- Mã hoá dùng **mẫu thay thế**: đây là dạng mã hoá thống kê mà nó thay thế các mẫu hay lặp lại bằng một mã.
- Mã hóa với độ dài (của từ mã) thay đổi (**VLC-Variable-Length Coding**)

Các kỹ thuật mã hoá nguồn (mã hóa tổn thất)

- **Mã hoá chuyển đổi:** dùng phép biến đổi Fourier hay Cosin để chuyển từ miền thời gian hay miền không gian sang miền tần số.
- **Mã hoá sai phân:** cũng được gọi là mã hoá ước đoán do chỉ mã hoá sự khác biệt giữa giá trị mẫu thực và giá trị ước đoán, mã hoá sai phân thường dùng cho video hình ảnh động. Lớp kỹ thuật này bao gồm: điều mã xung sai phân, điều chế delta, điều mã xung thích nghi.
- **Lượng tử hoá vector:** mã hoá từng khối hai chiều kích thước cố định (gọi là vector) và tra bảng tìm mã phù hợp nhất. Kỹ thuật chỉ thích hợp cho dữ liệu có cấu trúc biết trước.

Các tiêu chuẩn nén ảnh

- **JPEG** (The Joint Photographic Expert Group): dùng cho nén ảnh tĩnh, phát triển bởi sự kết hợp giữa ITU-TS (the International Telecommunications Union-Telecommunication Sector) và ISO (International Standards Organization)
- **MPEG-1, MPEG-2, MPEG-4, MPEG-7**: do Ủy ban ISO IEC/JTC1/SC29- /WG11 phát triển cho mã hoá kết hợp giữa video và audio.
- **H.261**: do Nhóm nghiên cứu XI phát triển và được biết rộng rãi như tiêu chuẩn mã hoá video cho các dịch vụ nghe nhìn tốc độ nx 64Kbps.
- **ITU-TS H.263** cho các ứng dụng điện thoại thấy hình tốc độ dưới 64Kbps.

Các kỹ thuật mã hoá Entropy (mã hóa không tổn thất)

- Run length Coding (RLC)
- Huffman
- Lempel Ziv – Wench (LZW)

Run-length coding (RLC)

- Mã hoá chiều dài chạy (RLC) là kỹ thuật mã hoá đơn giản nhất và được sử dụng rộng rãi.
- Sử dụng mã này không cần biết tần suất xuất hiện các ký tự và rất hiệu quả trong trường hợp dữ liệu được hiển thị bằng các bit 0 và 1.
- Ta thay thế một chuỗi các ký tự lặp đi lặp lại nhiều lần bằng một cặp (*chiều dài chạy, ký tự*).
- Đối với ảnh, chiều dài chạy cực đại chính là kích thước của một hàng.

RUN LENGTH CODING- RLC

- Tư tưởng của phương pháp này là dựa trên sự lặp lại các bit.
- Thay thế các bit đó bởi chiều dài chuỗi và bit lặp .
- Để phân biệt với các ký tự khác ta có thể thêm 1 từ mã đặc biệt trước 2 thông tin chiều dài chuỗi và bit lặp .
- **Ví Dụ: Ta có 1 dãy các giá trị mức xám như sau**

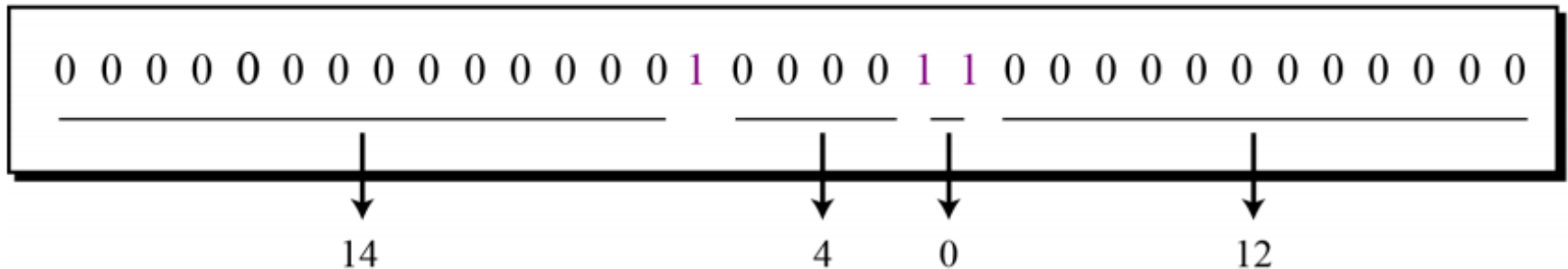
55 22 22 22 22 22 22 22 22 51 52 52 52 60 ...

Ta có thể thay đoạn mã trên bằng

- 55 E 8 22 51 E 3 52 60 ...
- Với E là ký tự đặc biệt , giá trị sau E là chiều dài ký tự lặp và ký tự lặp .

Ví dụ mã RLC

a. Original data



1110 0100 0000 1100

b. Compressed data

HUFFMAN

- Dựa vào mô hình thống kê tính tần suất xuất hiện các ký tự.
- Gán cho các ký tự có tần suất cao bằng một từ mã ngắn, các ký tự có tần suất thấp bằng một từ mã dài.
- **Thuật toán:**

- ***Bước 1:***

- Tính tần suất xuất hiện các ký tự trong dữ liệu gốc
- Xây dựng bảng mã
- Sắp xếp lại bảng mã theo thứ tự tần suất giảm dần

- ***Bước 2:***

- Tạo cây huffman

HUFFMAN

Bảng tần xuất sắp xếp theo thứ tự giảm dần

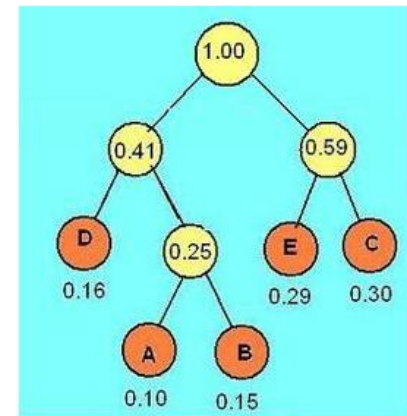
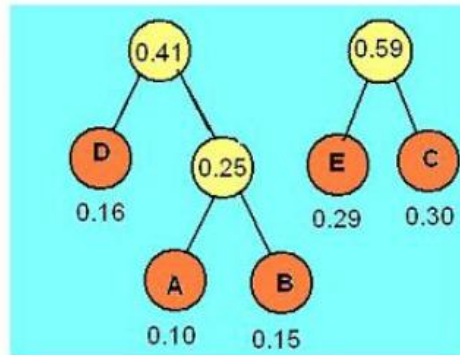
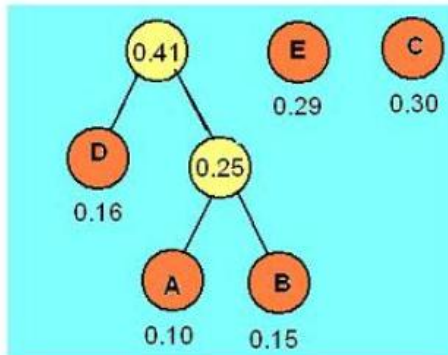
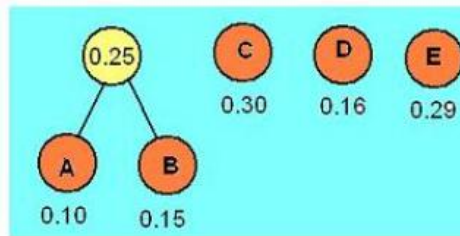
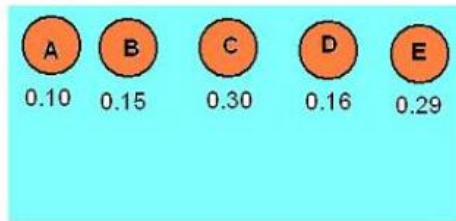
<i>Ký tự</i>	<i>Tần suất</i>	<i>Ký tự</i>	<i>Tần suất</i>	<i>xác suất</i>
"1"	152	"0"	1532	0.2770
"2"	323	"6"	602	0.1088
"3"	412	". "	536	0.0969
"4"	226	" "	535	0.0967
"5"	385	"3"	112	0.0746
"6"	602	"5 "	385	0.0696
"7"	92	"2"	323	0.0585
"8"	112	" _ "	315	0.0569
"9"	87	"4"	226	0.0409
"0"	1532	"+"	220	0.0396
". "	536	"1"	152	0.0275
"+"	220	"8"	112	0.0203
" _ "	315	"7"	92	0.0167
" "	535	"9"	87	0.0158

HUFFMAN

Cho bảng tần suất của 5 chữ cái A, B, C, D, E như sau tương ứng là 0.10; 0.15; 0.30; 0.16; 0.29

A	B	C	D	E
0.10	0.15	0.30	0.16	0.29

Quá trình xây dựng cây Huffman diễn ra như sau:



Như vậy bộ mã tối ưu tương ứng là:

A	B	C	D	E
010	011	11	00	10

Mã LZW

- Được Jacob Braham Ziv đưa ra lần đầu tiên năm 1977, sau đó phát triển thành một họ giải thuật nén từ điển là LZ.
- Năm 1984, Terry Welch cải tiến giải thuật LZ thành một giải thuật tốt hơn: LZW
- Dùng để giảm dư thừa trong pixel
- Không cần biết trước xác suất phân bố của các pixel
- Thường được dùng để nén các loại văn bản, ảnh đen trắng, ảnh màu, ảnh đa mức xám... Và là chuẩn nén cho các dạng ảnh GIF và TIFF.

Mã LZW

- Phương pháp :
- Xây dựng 1 từ điển

Cấu trúc
từ điển

0	0
1	1
...	...
...	...
255	255
256	256
257	257
258	Chuỗi
259	Chuỗi
...	...
...	...
4095	Chuỗi

(Clear Code)

(End Of Information)

Mã LZW

- Từ điển được xây dựng đồng thời với quá trình đọc dữ liệu. Sự có mặt của một chuỗi con trong từ điển khẳng định rằng chuỗi đó đã từng xuất hiện trong phần dữ liệu đã đọc.
- Thuật toán liên tục “tra cứu ” và cập nhật từ điển sau mỗi lần đọc một kí tự ở dữ liệu đầu vào.
- Do kích thước bộ nhớ không phải vô hạn và để đảm bảo tốc độ tìm kiếm, từ điển chỉ giới hạn 4096 ở phần tử dùng để lưu lớn nhất là 4096 giá trị của các từ mã.
- Như vậy độ dài lớn nhất của mã là 12 bít($4096 = 2^{12}$).

Ví dụ

Cho ma trận ảnh 4x4 8bit

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

LEMPER ZIV – WENCH (LZW)

Ví dụ: bảng mã hóa và xây dựng tự điển

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Kết quả: input: $16 \times 8 = 128 \text{ bit}$
output : $5 \times 8 + 5 \times 9 = 85 \text{ bit}$
tỷ số nén $128/85 = 1,5$

Mã LZW

Ví dụ cơ chế nén LZW

Cho chuỗi **ban đầu** là

“**ABCBCABCABCD**”

(Mã ASCII của A là 65, B là 66, C là 67).

Từ điển ban đầu đã gồm **256** kí tự cơ bản.

Chuỗi **đầu ra** sẽ là:

65 - 66 - 67 - 259 - 258 - 67 - 262

Đầu vào có kích thước: $12 \times 8 = 96$ bits.

Đầu ra có kích thước là: $4 \times 8 + 3 \times 9 = 59$ bits

Tỉ số nén là: $96:59 \approx 1,63$.

$$C_R = \frac{n_1}{n_2}.$$

Đầu vào	Đầu ra	Thực hiện
A (65)		A đã có trong từ điển \Rightarrow Đọc tiếp
B (66)	65	Thêm vào từ điển mã 258 đại diện cho mã AB
C (67)	66	Thêm vào từ điển mã 259 đại diện cho mã BC
B	67	Thêm vào từ điển mã 260 đại diện cho mã CB
C		BC đã có trong từ điển \Rightarrow Đọc tiếp
A	259	Thêm vào từ điển mã 261 đại diện cho mã BCA
B		AB đã có trong từ điển \Rightarrow Đọc tiếp
C	258	Thêm vào từ điển mã 262 đại diện cho mã ABC
A	67	Thêm vào từ điển mã 263 đại diện cho mã CA
B		AB đã có trong từ điển \Rightarrow Đọc tiếp
C		ABC đã có trong từ điển \Rightarrow Đọc tiếp
D	262	Thêm vào từ điển mã 263 đại diện cho mã ABCD

Bài tập

- Dùng mã hóa LZ để mã hóa chuỗi **ACCBCABCABACD** tính tỉ lệ nén và hiệu suất nén
- Tỉ lệ nén $\frac{1}{c_r} \times \%$ với $C_R = \frac{n_1}{n_2}$.
- Hiệu suất nén $(1 - \frac{1}{c_r}) \times \%$

NÉN ẢNH THEO CHUẨN JPEG

Chuẩn JPEG

- **JPEG** (Joint Photographic Expert Group) là tên của một tổ chức nghiên cứu về các chuẩn nén ảnh (trước đây là ISO) được thành lập vào năm 1982.
- Năm 1986, JPEG chính thức được thiết lập nhờ sự kết hợp giữa nhóm ISO/IEC và ITU. Tiêu chuẩn này có thể được ứng dụng trong nhiều lĩnh vực: lưu trữ ảnh, Fax màu, truyền ảnh báo chí, ảnh cho y học, camera số v.v...
- Tiêu chuẩn JPEG được định ra cho nén ảnh tĩnh đơn sắc và màu. Tuy nhiên cũng được sử dụng cho nhiều ứng dụng với ảnh động bởi vì nó cho chất lượng ảnh khôi phục khá tốt và ít tính toán hơn so với nén MPEG.

NÉN ẢNH JPEG

- ❖ Là một trong những phương pháp nén ảnh hiệu quả, có tỷ lệ nén ảnh tới **vài chục lần**.
- ❖ Tuy nhiên ảnh sau khi giải nén sẽ khác với ảnh ban đầu. **Chất lượng ảnh bị suy giảm sau** khi giải nén. Sự suy giảm này tăng dần theo hệ số nén.
- ❖ Sự mất mát thông tin này là có thể chấp nhận được vì việc **loại bỏ những thông tin không cần thiết** được dựa trên những nghiên cứu về hệ nhãn thị của mắt người.
- ❖ Phần mở rộng của các file JPEG thường có dạng **.jpeg, .jfif, .jpg, .JPG, hay .JPE; dạng .jpg** là dạng được dùng phổ biến nhất.
- ❖ Hiện nay dạng nén ảnh JPEG rất được phổ biến trong **ĐTDD** cũng như những trang thiết bị lưu giữ có **dung lượng nhỏ**.

NÉN ẢNH JPEG

- ✚ Các **thông tin tần số cao** có thể bị loại bỏ mà không làm mất mát thông tin quan sát vì mắt người **không cảm nhận** được những hiệu ứng do các thành phần tần số cao mang lại **một cách chính xác**
- ✚ Ảnh được chuyển sang **miền tần số** sử dụng phép biến đổi Cosin rời rạc - Discrete Cosine Transform (**DCT**).
- ✚ Phép biến đổi DCT thường được áp dụng cho các **khối pixel kích thước 8×8** .
- ✚ Việc áp dụng DCT **không làm giảm kích thước của dữ liệu**, vì số các hệ số của DCT cũng bằng tổng số pixel của khối (64).
- ✚ Tuy nhiên, các hệ số của DCT được **lượng tử hóa**, vì thế số bit cần thiết để biểu diễn các hệ số DCT sẽ giảm đi. Việc lượng tử hóa sẽ làm biến mất một số thông tin.

NÉN ẢNH JPEG

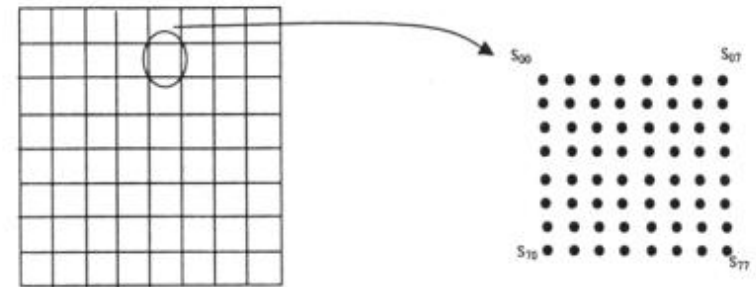
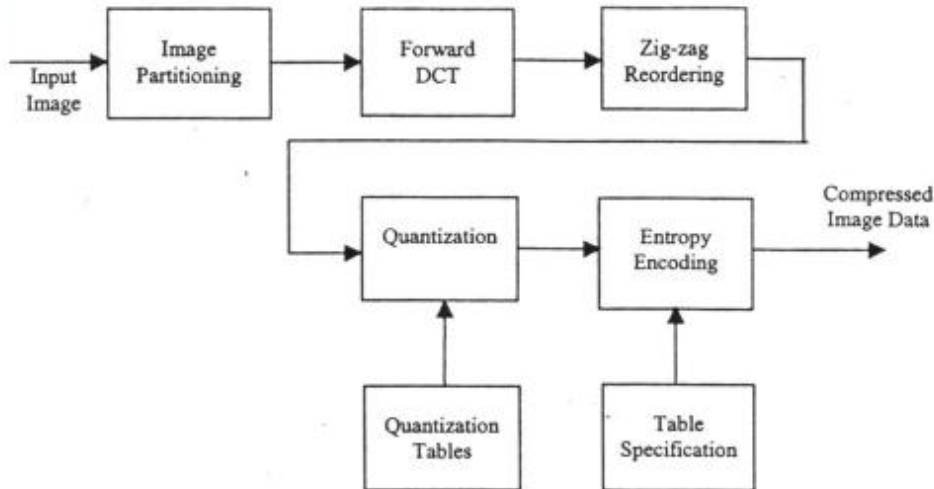


FIGURE 7.4 Partitioning to 8 × 8 blocks.

FIGURE 7.3 Block diagram of a sequential DCT-based encoding process.

$$\text{FDCT: } S_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^7 \sum_{j=0}^7 s_{ij} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16}$$

$$\text{IDCT: } s_{ij} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C_u C_v S_{uv} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16}$$

$$C_u C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

DC

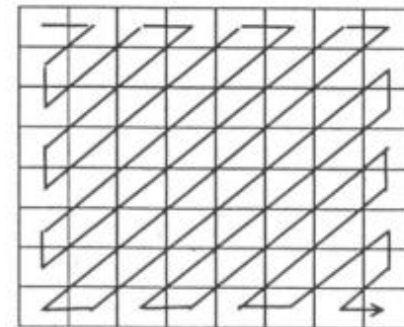


FIGURE 7.5 Zigzag scanning order of DCT coefficients.

Biến đổi Cosin và chuẩn JPEG

$$2\text{-D IDCT} = x_{m,n} = \sum_{k=0}^7 \sum_{l=0}^7 \frac{c(k)c(l)}{4} X_{k,l} \cos\left(\frac{(2m+1)k\pi}{16}\right) \cos\left(\frac{(2n+1)l\pi}{16}\right)$$

Trong đó $m, n = 0, 1, \dots, 7$

$$\text{và } c(k), c(l) = \begin{cases} 1/\sqrt{2}, & k \text{ \& } l = 0 \\ 1, & k^2 + l^2 \neq 0 \end{cases}$$

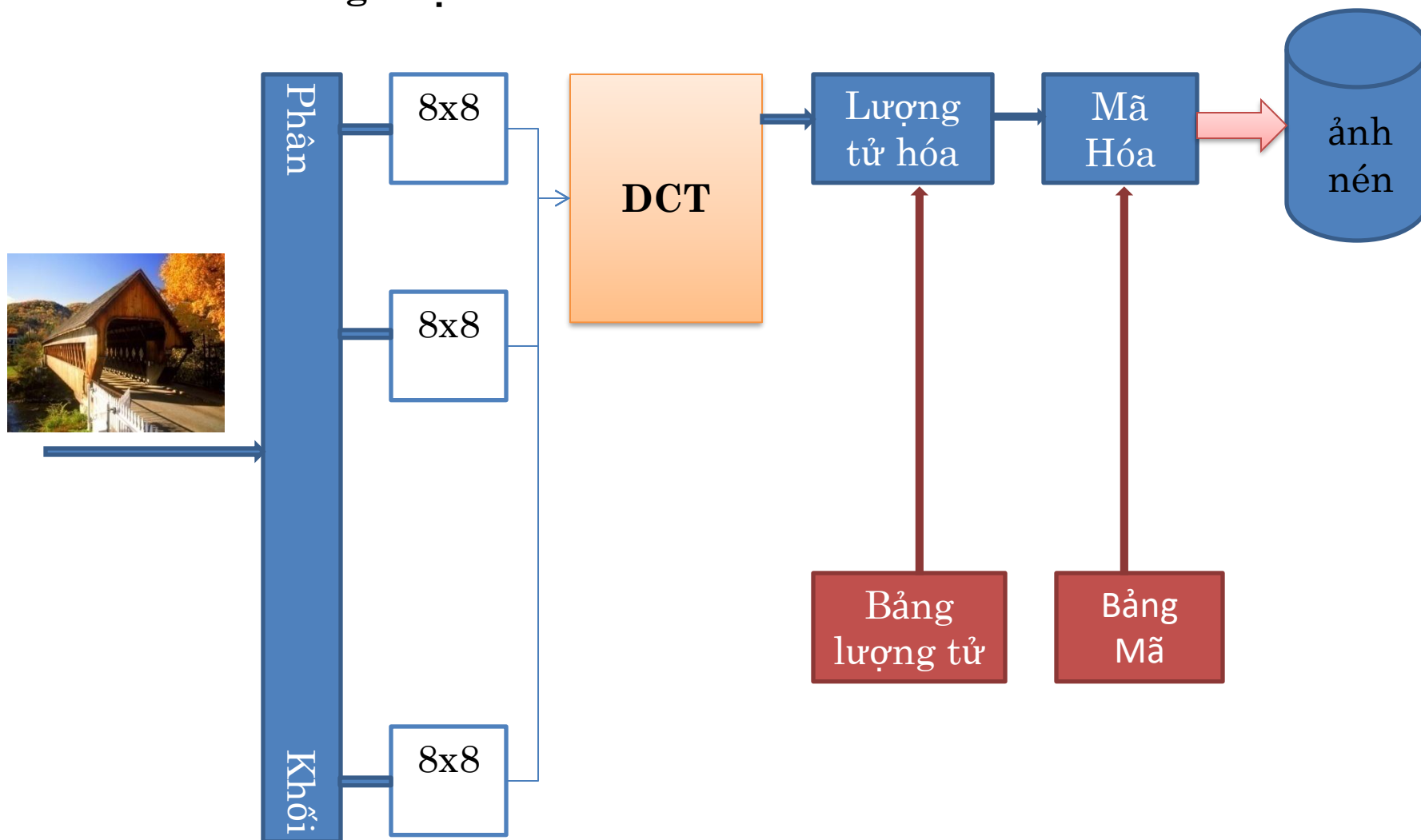
Thuật toán để tính 2-D DCT và IDCT là: thực hiện phép biến đổi 1-D lần lượt cho hàng rồi đến cột của ma trận.

Trong đó
$$X(0,0) = \frac{1}{8} \sum_{m=0}^7 \sum_{n=0}^7 x(m,n)$$

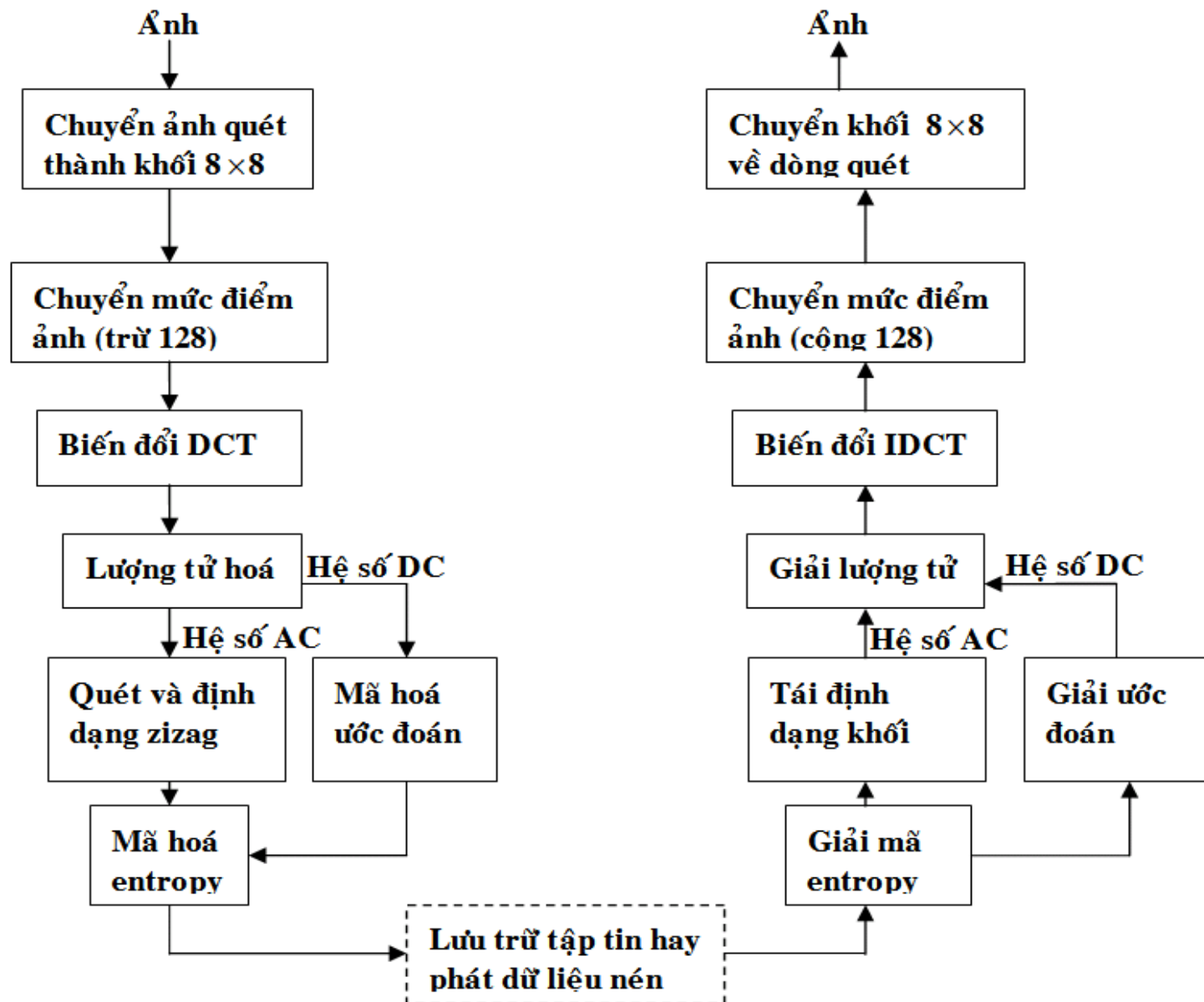
$$X_{(k,l)} = \frac{2c(k)c(l)}{N} \sum_{m=0}^7 \sum_{n=0}^7 x_{m,n} \cos\left(\frac{(2m+1)k\pi}{2N}\right) \cos\left(\frac{(2n+1)l\pi}{2N}\right)$$

Chuẩn JPEG

Các công đoạn Nén ảnh JPEG



Chuẩn JPEG

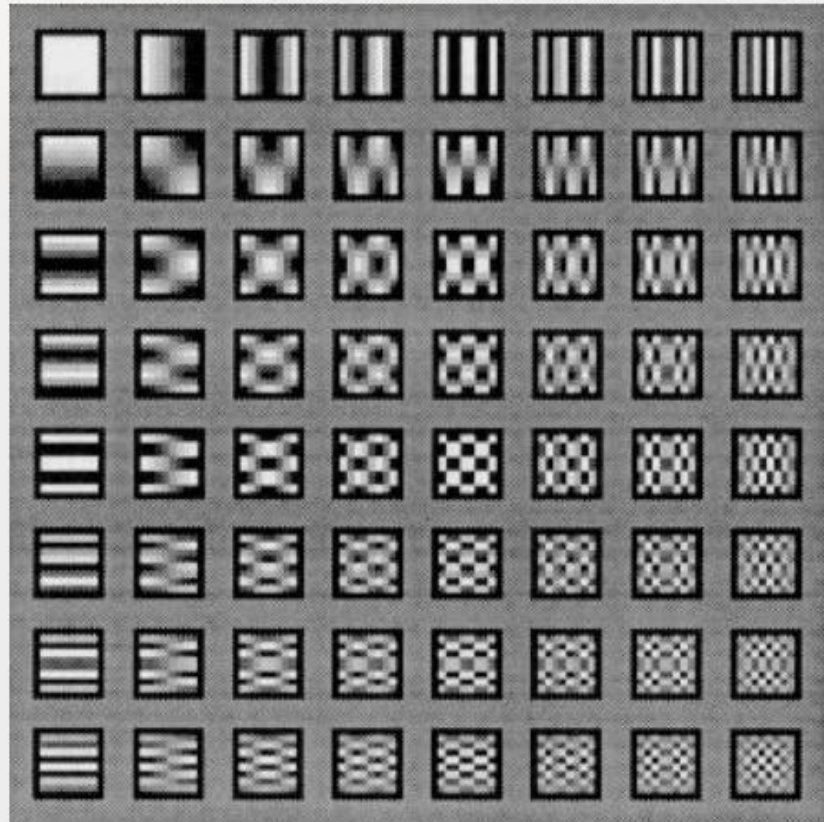


Chuyển ảnh thành các MB

- Tất cả các block có cùng kích thước và mỗi block là một ma trận điểm ảnh 8×8 pixel được lấy từ một ảnh màn hình theo chiều từ trái sang phải, từ trên xuống dưới.
- Kích thước MB là 8×8 được chọn bởi hai lý do sau:
 1. Qua việc nghiên cứu cho thấy hàm tương quan suy giảm rất nhanh khi khoảng cách giữa các pixel vượt quá 8.
 2. Tiện lợi cho việc tính toán và thiết kế phần cứng. Nói chung, độ phức tạp về tính toán sẽ tăng nếu kích thước block tăng.

Chuyển ảnh thành các MB

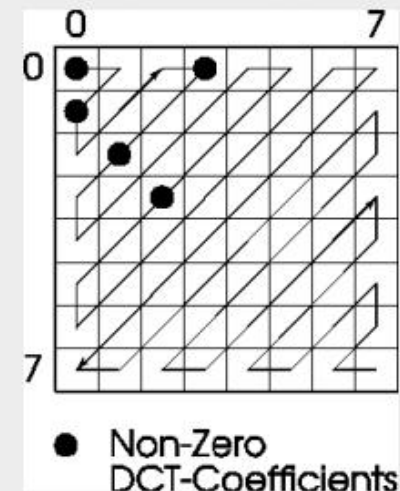
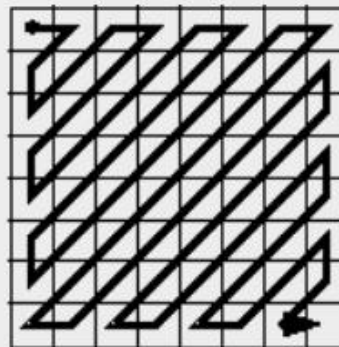
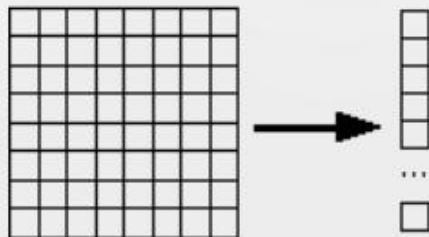
- Each 8x8 block can be looked at as a weighted sum of these basis functions.
- The process of 2D DCT is also the process of finding those weights.



Chuyển ảnh thành các MB

Zig-zag Scan DCT Blocks

- Why? -- To group low frequency coefficients in top of vector.
- Maps 8 x 8 to a 1 x 64 vector.



Chuyển mức điểm ảnh (trừ 128)


Ví Dụ:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix} \xrightarrow{-128} \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Biến đổi DCT

Ví Dụ:

Biến đổi DCT và làm tròn các hệ số


$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Lượng tử hóa

Ví dụ:

Sử dụng Ma trận lượng tử hóa (Q)

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

Lượng tử hóa

Ví Dụ

(tt):

Chia các phần tử của ma trận DCT với các phần tử tương ứng của ma trận lượng tử hóa ở trên theo công thức:

$$G(u, v) = \text{round} \left[\frac{T(u, v)}{Q(u, v)} \right]$$

DC

AC

Kết Quả

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$G(0, 0) = \text{round} \left[\frac{T(0, 0)}{Q(0, 0)} \right] = \text{round} \left[\frac{-415}{16} \right] = -26$$

Mã hóa AC

Sử Dụng Mã hóa Huffman kết thúc sớm chuỗi khi các kí tự còn lại là 0

Ký tự đặc biệt là EOB



Kết Quả

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

Giả sử nếu DC của khối trước là -17

Mã hóa nhị phân

```
1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
      001 100101 11100110 110110 0110 11110100 000 1010
```

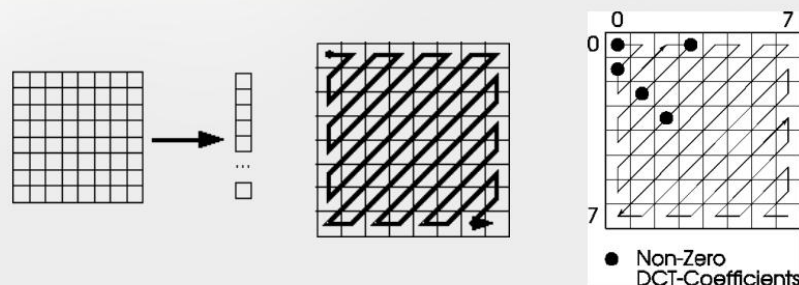

Ví dụ

qdet =

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

Zig-zag Scan DCT Blocks

- Why? -- To group low frequency coefficients in top of vector.
- Maps 8 x 8 to a 1 x 64 vector.



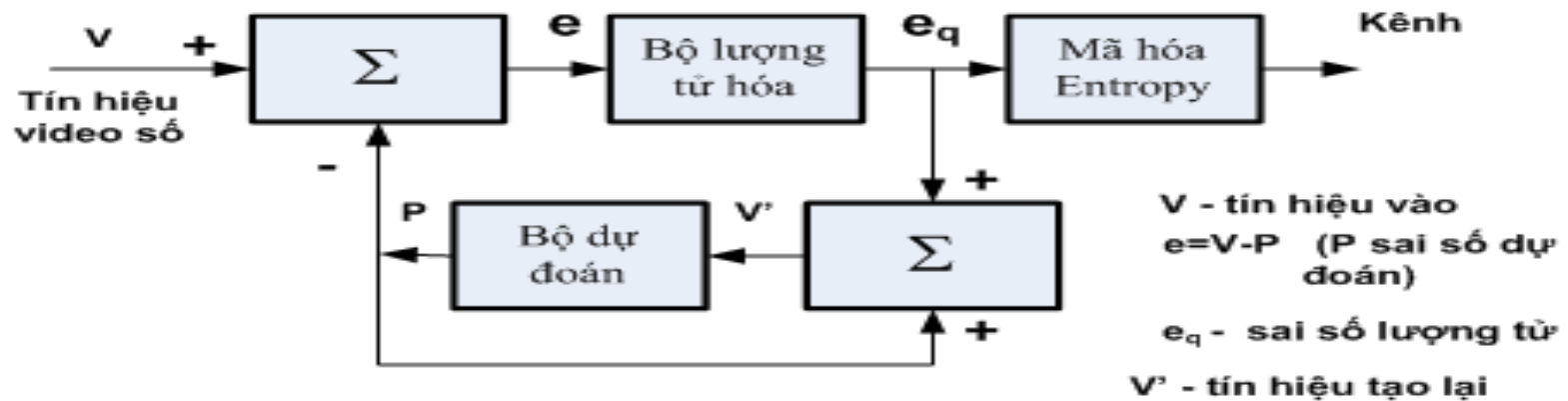
Run-length symbol representation:

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

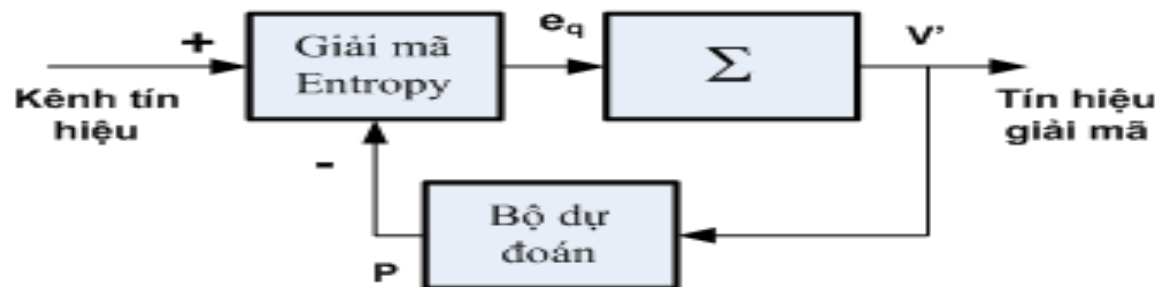
EOB: End of block, one of the symbol that is assigned a short Huffman codeword

Mã hóa DC

Mã hóa DPCM

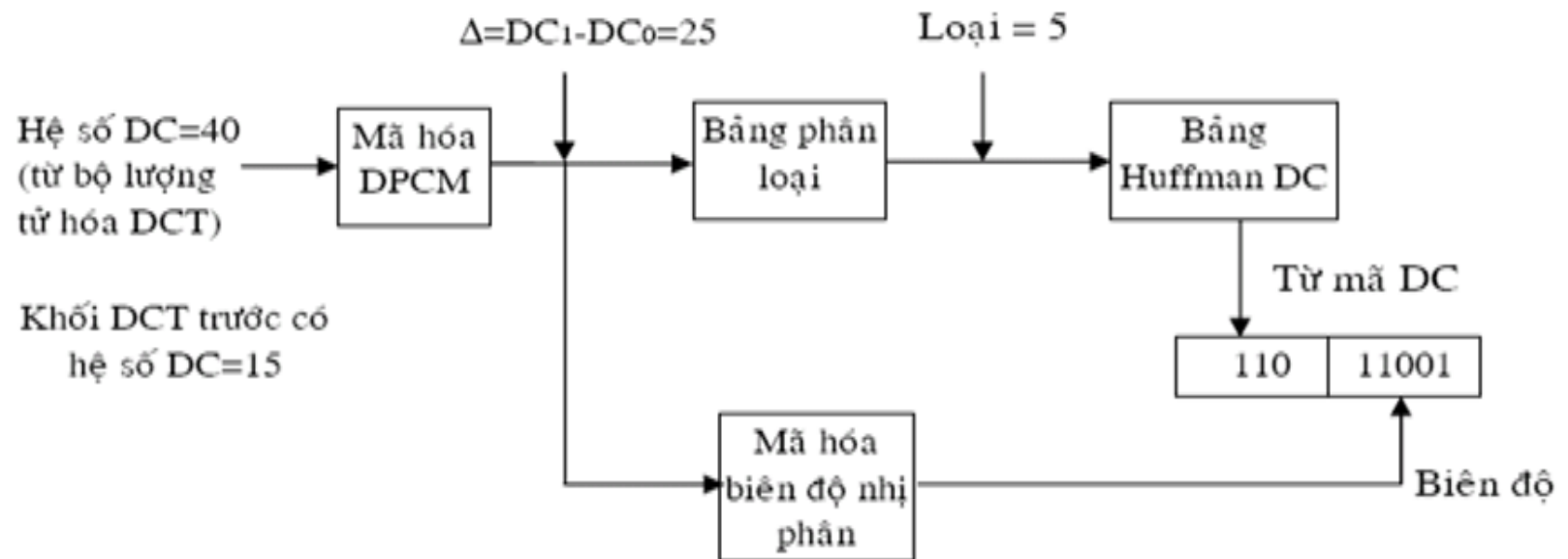


Giải mã DPCM



Mã hóa DC

a) Mã hóa VLC các hệ số DC



Mã hóa DC

Example:

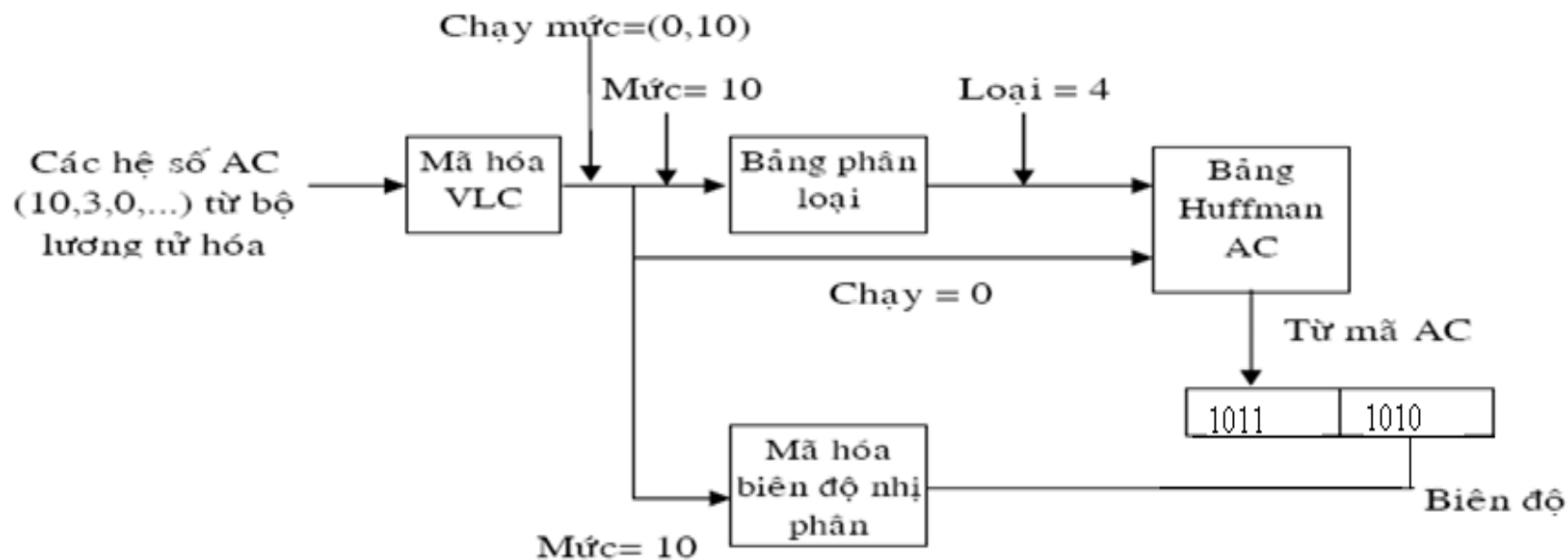
- Current quantized DC index: 2
- Previous block DC index: 4
- Prediction error: -2
- The prediction error is coded in two parts:
 - Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code using a Huffman code (JPEG Default DC Code)
 - DC= -2 is in category “2”, with a codeword “100”
 - Which position it is in that category, using a fixed length code, length=category number
 - “-2” is the number 1 (starting from 0) in category 2, with a fixed length code of “01”
 - Mã tổng thể 10001

Bảng mã DC

Range	DC Difference Category	AC Category
0	0	N/A
-1, 1	1	1
-3, -2, 2, 3	2	2
-7, ..., -4, 4, ..., 7	3	3
-15, ..., -8, 8, ..., 15	4	4
-31, ..., -16, 16, ..., 31	5	5
-63, ..., -32, 32, ..., 63	6	6
-127, ..., -64, 64, ..., 127	7	7
-255, ..., -128, 128, ..., 255	8	8
-511, ..., -256, 256, ..., 511	9	9
-1023, ..., -512, 512, ..., 1023	A	A
-2047, ..., -1024, 1024, ..., 2047	B	B
-4095, ..., -2048, 2048, ..., 4095	C	C
-8191, ..., -4096, 4096, ..., 8191	D	D
-16383, ..., -8192, 8192, ..., 16383	E	E
-32767, ..., -16384, 16384, ..., 32767	F	N/A

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

Mã hóa AC



Mã hóa AC

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

Bảng mã AC

Range	DC Difference Category	AC Category
0	0	N/A
-1, 1	1	1
-3, -2, 2, 3	2	2
-7, ..., -4, 4, ..., 7	3	3
-15, ..., -8, 8, ..., 15	4	4
-31, ..., -16, 16, ..., 31	5	5
-63, ..., -32, 32, ..., 63	6	6
-127, ..., -64, 64, ..., 127	7	7
-255, ..., -128, 128, ..., 255	8	8
-511, ..., -256, 256, ..., 511	9	9
-1023, ..., -512, 512, ..., 1023	A	A
-2047, ..., -1024, 1024, ..., 2047	B	B
-4095, ..., -2048, 2048, ..., 4095	C	C
-8191, ..., -4096, 4096, ..., 8191	D	D
-16383, ..., -8192, 8192, ..., 16383	E	E
-32767, ..., -16384, 16384, ..., 32767	F	N/A

Mã hóa AC

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	1010 (= EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

Ví dụ: DC coefficient of the previous block is 60

Cho ma trận ảnh lượng tử:

$$QDCTindex = \begin{bmatrix} 85 & -5 & -2 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

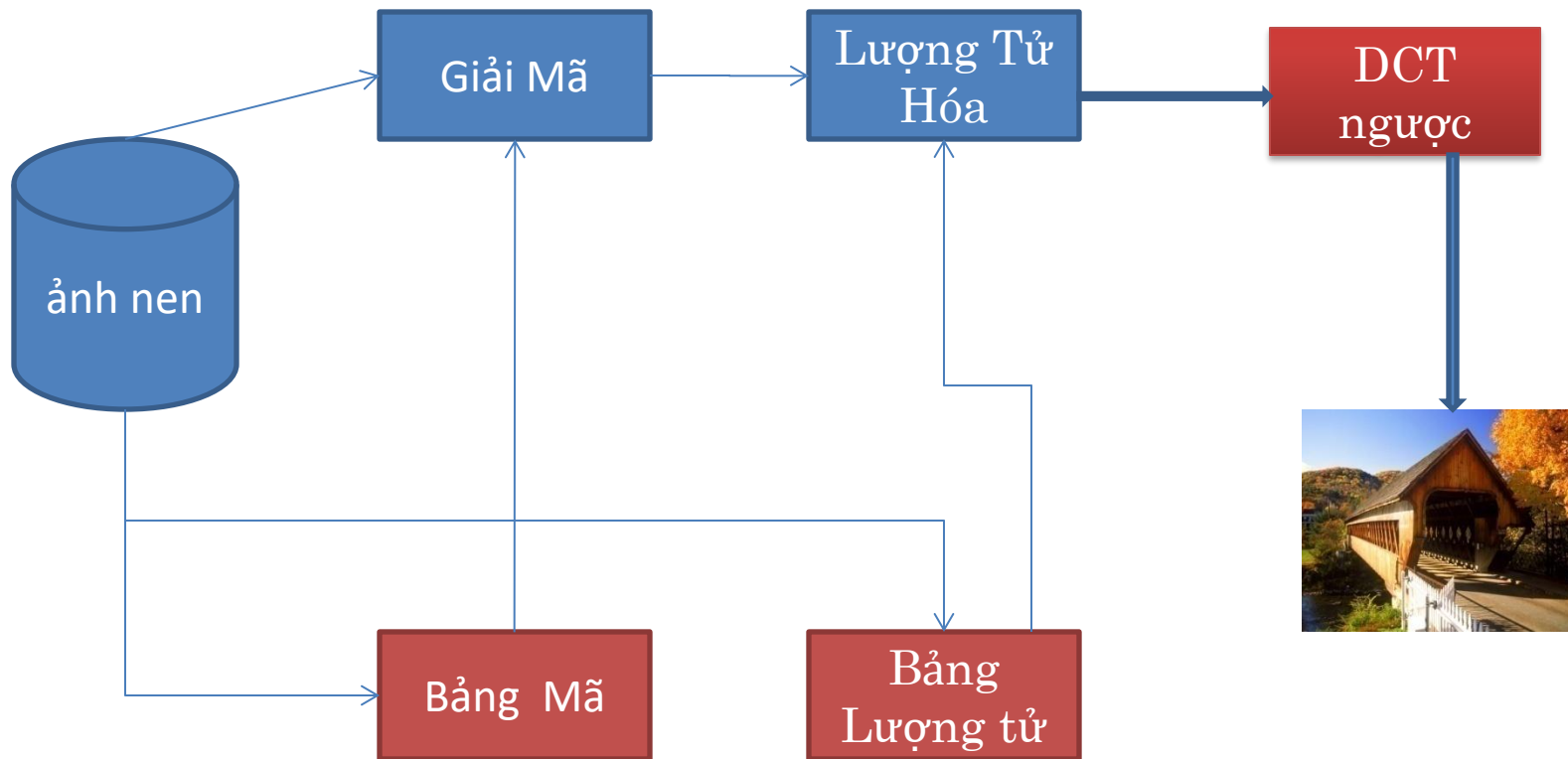
Biết rằng hệ số DC của khối trước là 60 . Hãy mã hóa ma trận ảnh thành mã nhị phân. Tính tỉ số nén

Kết quả

- “11011001, 100010, 001, 1111100001, 0110, 0110, 000,1010”.

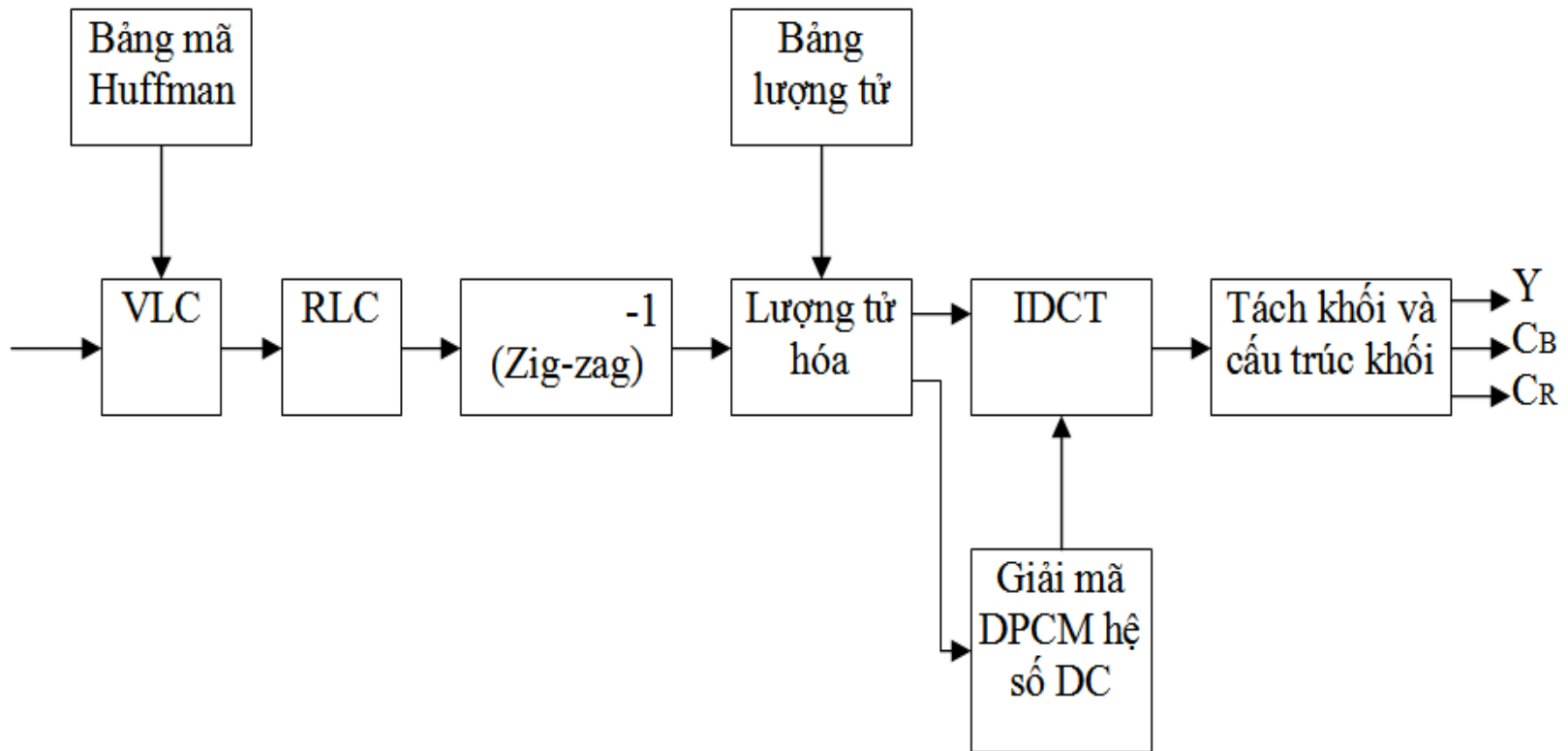
Khôi phục ảnh JPEG

Các công đoạn khôi phục ảnh:



Khôi phục ảnh JPEG

$$R(u,v) = Fq(u,v)Q(u,v)$$



Khôi phục các điểm ảnh trong khối 8x8

0	u	7						
v								
7								

a) Các hệ số DCT lượng tử hóa và chuẩn hóa $F_q(u,v)$

Biến đổi
IDCT

0	u	7						
v								
7								

b) Bảng trọng số $Q(u,v)$

0	i	7						
k								
7								

d) Khối 8x8 các điểm chổi $f^*(j,k)$

0	u	7						
v								
7								

c) Khối các hệ số DCT tương ứng sau khi giải lượng tử

Quá trình biến đổi DCT ngược (IDCT)

Tạo lại khối giá trị các điểm ban đầu theo biểu thức:

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

Sai số giữa các giá trị khôi phục và giá trị gốc được tính như sau:

$$e(j,k) = f(j,k) - f^*(j,k)$$

Biến đổi cosin và chuẩn JPEG

Ảnh Gốc



Ảnh nén và giải nén



Biến đổi cosin và chuẩn JPEG

Ảnh sau khi nén và giải nén bằng phép biến đổi Cosin cho chất lượng không tốt như cũ.

Khắc phục bằng việc làm trơn ảnh sau khi giải nén .

Chuẩn JPEG

Chuẩn JPEG cho nhiều tùy chọn khác nhau để nén ảnh, tùy vào những tùy chọn mà cho hệ số nén và chất lượng ảnh khác nhau



31KB



46KB



60KB



100KB

Ảnh gốc có kích thước $400 \times 300 = 120\text{KB}$

Bài tập

Cho ma trận ảnh S là 2×2

$$S = \begin{pmatrix} 9 & 1 \\ 1 & 9 \end{pmatrix}$$

Tính các hệ số DCT, và viết chuỗi nhị phân sau khi sử dụng phương pháp nén JPEG, tìm ảnh khôi phục sau khi giải nén, biết rằng ma trận lượng tử

$$Q = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}$$

Lời giải

- Áp dụng công thức DCT để tính hệ số DC và AC

$$2-DCT = X_{(k,l)} = \frac{2}{N} c(k)c(l) \sum_{m=0}^7 \sum_{n=0}^7 x_{(m,n)} \cos\left(\frac{((2m+1)k\pi)}{2N}\right) \cos\left(\frac{((2n+1)l\pi)}{2N}\right)$$

Trong đó $k, l = 0, 1, \dots, 7$

$$c(k), c(l) = \begin{cases} \frac{1}{\sqrt{2}} & , k \& l = 0 \\ 1, k^2 + l^2 \neq 0 \end{cases}$$

Hệ số DCT



$$T_{0,0} = (\mathbf{S}, \mathbf{U}_{0,0}) = (9+1+1+9)/2 = 10, T_{0,1} = (\mathbf{S}, \mathbf{U}_{0,1}) = (9-1+1-9)/2 = 0,$$

$$T_{1,0} = (\mathbf{S}, \mathbf{U}_{1,0}) = (9+1-1-9)/2 = 0, T_{1,1} = (\mathbf{S}, \mathbf{U}_{1,1}) = (9-1-1+9)/2 = 8.$$

Ảnh khôi phục sau khi giải nén



$$\hat{S} = \hat{T}_{0,0} \mathbf{U}_{0,0} + \hat{T}_{0,1} \mathbf{U}_{0,1} + \hat{T}_{1,0} \mathbf{U}_{1,0} + \hat{T}_{1,1} \mathbf{U}_{1,1} = \frac{9}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \frac{10}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 9.5 & -0.5 \\ -0.5 & 9.5 \end{bmatrix}$$

TỔNG QUAN JPEG 2000

- Ảnh số hóa ngày càng phổ biến và yêu cầu chất lượng ngày càng cao, ngoài ra còn có các nhu cầu về xử lý ảnh kéo theo
- Nén hình ảnh không chỉ làm giảm dung lượng mà còn phải cho phép tách ghép để sắp xếp xử lý và đáp ứng các nhu cầu trên các thiết bị cụ thể
- Yêu cầu về hiệu suất nén với tỉ số nén cao



SỰ PHÁT TRIỂN CỦA JPEG 2000 LÀ TẤT YẾU

TỔNG QUAN JPEG 2000

The logo consists of the text "JPEG2000" in a bold, yellow, sans-serif font, set against a dark blue rectangular background.

JPEG-2000 có tỉ lệ
xuyên âm thấp hơn
hẳn công nghệ
JPEG truyền thống

The logo consists of the text "JPEG2000" in a bold, yellow, sans-serif font, set against a dark blue rectangular background.

Cho phép tách các
phân giải khác
nhau, các miền
quan tâm và đưa ra
được một dòng bit
nén đơn

The logo consists of the text "JPEG2000" in a bold, yellow, sans-serif font, set against a dark blue rectangular background.

JPEG-2000 sử
dụng kĩ thuật mã
hóa dạng sóng rời
rạc DWT-Discrete
Wavelet
Transform

ƯU ĐIỂM CỦA JPEG2000 SO VỚI JPEG

JPEG2000

Sử dụng được với truyền dẫn và hiển thị lũy tiến về chất lượng, độ phân giải, các thành phần màu có tính định vị không gian

JPEG2000

Sử dụng một cơ chế nén ảnh cho cả 2 dạng thụt nén (có tổn thất và không tổn thất)

JPEG2000

JPEG2000 có thể đưa ra tỷ lệ nén cao hơn nhiều so với JPEG. Tỷ lệ nén có thể lên tới 200:1

JPEG2000

Có khả năng mã hóa ảnh với tỉ lệ nén theo từng vùng khác nhau (ROI)

Chuẩn JPEG2000 và nén ảnh màu

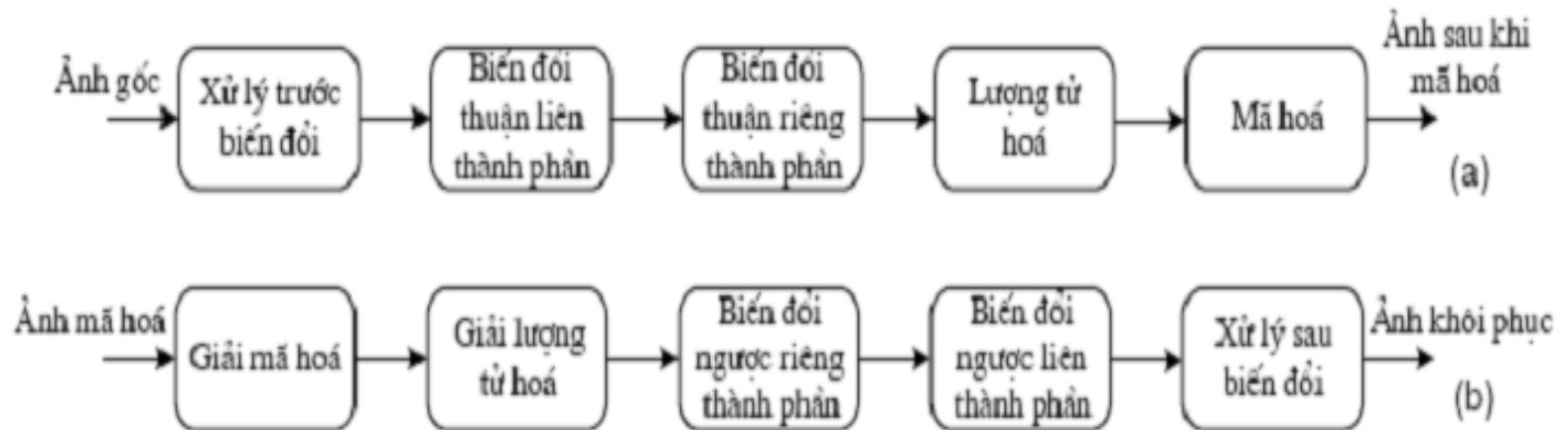
- *JPEG2K sử dụng phép biến đổi wavelet và các phương pháp đặc biệt để có được ảnh nén tối ưu nhất*
- Cho chất lượng tốt nhất khi sử dụng nén ảnh tĩnh
- Sử dụng được với truyền dẫn và hiển thị lũy tiến về chất lượng, độ phân giải.
- Truy cập và giải nén tại mọi thời điểm trong khi nhận dữ liệu
- Giải nén từng vùng ảnh mà không giải nén toàn bộ ảnh

CHUẨN JPEG2K

- Có khả năng mã hoá ảnh với tỷ lệ nén theo từng vùng khác nhau
- Nén một lần nhưng có thể giải nén với nhiều cấp chất lượng tùy theo yêu cầu của người sử dụng

Hiện tại, *ISO* và uỷ ban *JPEG* đã đưa ra khuyến nghị thay thế *JPEG* bằng *JPEG2000*.

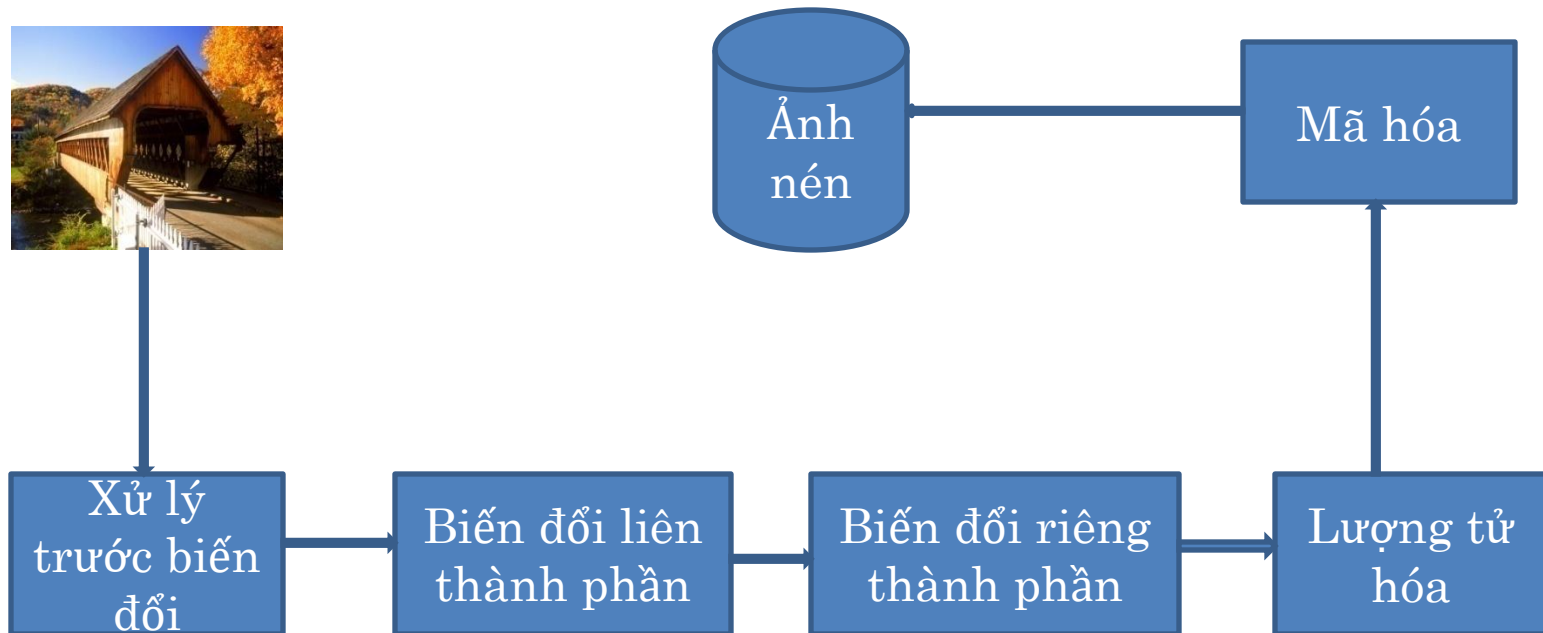
Các bước nén ảnh trong JPEG 2000



Sơ đồ nén và giải nén trong jpeg 2000

CHUẨN JPEG 2000 VÀ NÉN ẢNH MÀU

- Các bước thực hiện nén ảnh theo chuẩn JPEG2K



Các bước nén ảnh trong JPEG 2000

BUỐC 1: Xử lý trước biến đổi

Xử lý trước biến đổi chính là giai đoạn đảm bảo dữ liệu đưa vào nén ảnh có dạng đối xứng qua 0

Các bước nén ảnh trong JPEG 2000

Bước 2: biến đổi liên thành phần

Loại bỏ tính tương quan giữa các thành phần của ảnh

Chuyển từ RGB sang YCrBr sử dụng ICT và RCT

Làm tăng hiệu quả nén

Các bước nén ảnh trong JPEG 2000

Bước 3: Biến đổi riêng thành phần (biến đổi Wavelet)

Sử dụng biến đổi wavelet.

Chia tín hiệu thành các băng con và mỗi băng sẽ được mã hóa riêng rẽ

Thường sử dụng hàm wavelet Daubechies

Các bước nén ảnh trong JPEG 2000

BUỐC 4 : Lượng tử hóa

Cho phép đạt tỉ lệ nén cao hơn bằng cách thể hiện các giá trị biến đổi với độ chính xác tương ứng cần thiết với mức chi tiết của ảnh cần nén.

Các hệ số biến đổi sẽ được lượng tử hoá theo phép lượng tử hoá vô hướng

Các bước nén ảnh trong JPEG 2000

Bước 5 : Mã hóa

JPEG-2000 cho phép sử dụng
nhiều phương pháp mã hóa
khác nhau

Trong thực tế sử dụng 2 phương
pháp phổ biến là SPIHT và
EZW

Các bước nén ảnh trong JPEG 2000

SPIHT

Là viết tắt của
**Set Partitioning in
Hierarchical Trees**

-phương pháp mã
hoá phân cấp theo
phân vùng

SPIHT được thiết kế
tối ưu cho truyền
dẫn luỹ tiến

Sử dụng kỹ thuật
embedded coding

- Có nghĩa là: một ảnh sau nén
với kích cỡ (lưu trữ) lớn (tỷ lệ
nén thấp) sẽ chứa chính dữ liệu
sau nén của ảnh có kích cỡ (lưu
trữ) nhỏ (tỷ lệ nén cao)

Các bước nén ảnh trong JPEG 2000

EZW

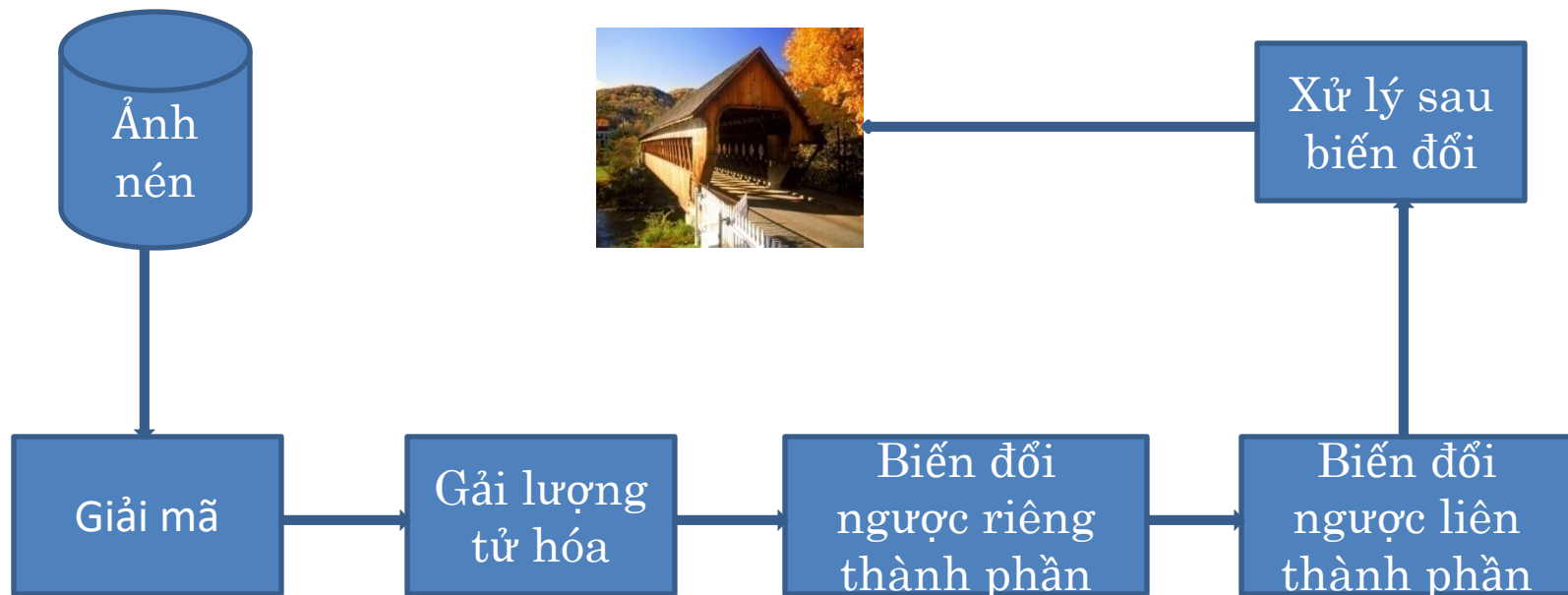
Viết tắt của
**Embedded
Zerotree Wavelet
Encoder**

Cũng dựa trên
phương pháp mã
hóa lũy tiến

Phương pháp này
chủ yếu dựa trên
khái niệm về cây
zero (zerotree).

CHUẨN JPEG2000 VÀ NÉN ẢNH MÀU

- Giải nén: Làm ngược lại các bước trên



Các chuẩn nén ảnh hiện nay

Các định dạng ảnh và tỉ lệ nén:

1600x1200	5760KB
BMP	5626KB
TGA	5626KB
PGM	5626KB
PBM	5626KB
TIF	5565KB
PNP	4405KB
XPM	3759KB
GIF	1802KB
JPEG	754KB

Ưu điểm của chuẩn nén JPEG2k

Một tính năng ưu việt nữa của *JPEG2000* là tính năng mã hoá ảnh quan trọng theo vùng (*ROI - Region of Interest*) mà chúng ta đã đề cập trong phần mã hoá ảnh theo *JPEG2000*.

Tính năng ưu việt thứ 3 của *JPEG2000* so với *JPEG* là chuẩn nén ảnh này có thể hiển thị được các ảnh với độ phân giải và kích thước khác nhau từ cùng một ảnh nén. Với *JPEG* thì điều này là không thể thực hiện được. Sở dĩ có điều này là do *JPEG2000* sử dụng kỹ thuật phân giải ảnh và mã hoá đỉnh kèm

JPEG và JPEG2k

Bit per pixel	0. 125	0. 50	2.00
Ảnh 1 theo <i>JPEG</i>	24.42	31.17	35. 15
Ảnh 1 theo <i>JPEG2000</i>	28. 12	32.95	37. 35
Ảnh 2 theo <i>JPEG</i>	22.6	28. 92	35. 99
Ảnh 2 theo <i>JPEG2000</i>	24.85	31.13	38. 80

JPEG và JPEG2k



JPEG



JPEG2000



JPEG



JPEG2000

Bài tập

RGB <-> YCbCr

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

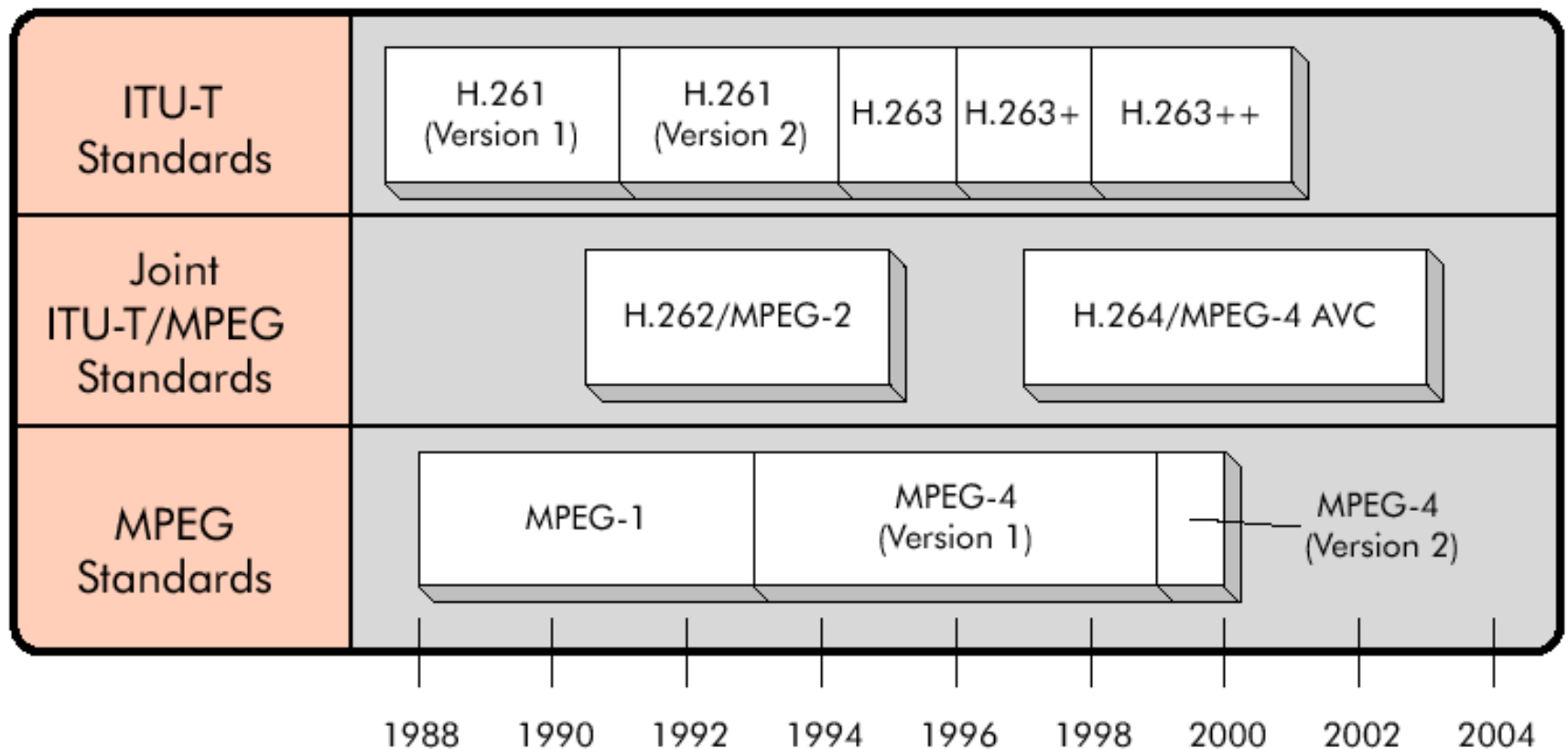
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & -0.001 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.772 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix}$$

Bài tập

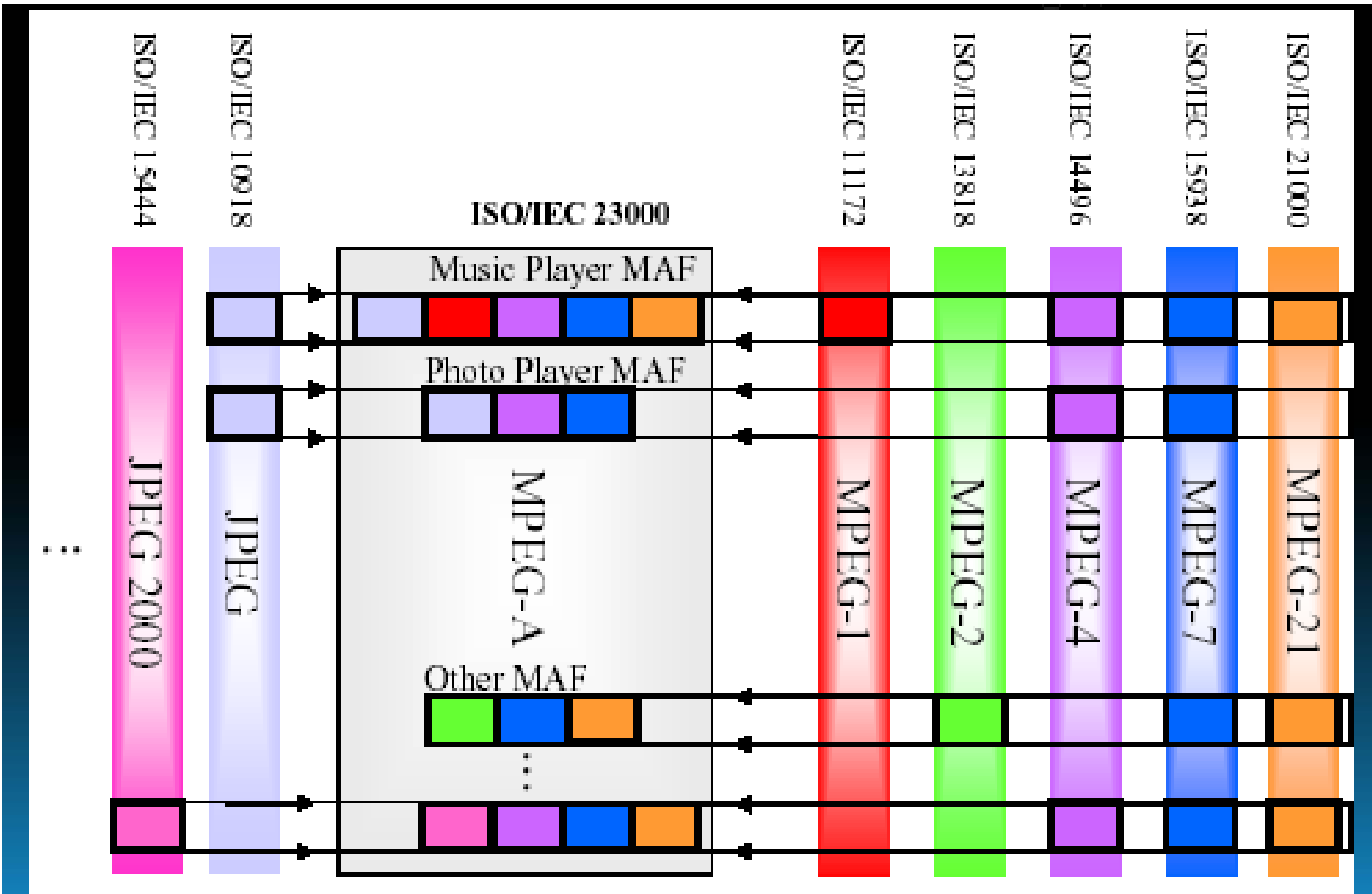
- What is the (Y,Cb,Cr) value for (R,G,B)=(255,128,64)?
- What is the (R,G,B) value for (Y,Cb,Cr)=(128,25,43)?

KỸ THUẬT NÉN VIDEO

Các chuẩn nén video



MPEG



MPEG

- **MPEG-1 ~ 1-1.5Mbps (early 90s)**
 - mã hoá tín hiệu Audio-Video với tốc độ khoảng 1.5Mb/
 - lưu trữ trong đĩa CD với chất lượng tương đương VHS
- **MPEG-2 ~ 2-80Mbps (mid 90s)**
 - Độ phân giải cao
 - Hỗ trợ nhiều chuẩn Video số và truyền hình HD
 - Address scalable video coding
 - Sử dụng cho DVD
- **MPEG-4 ~ 9-40kbps (later 90s)**
 - Mã hóa audio và video tốc độ thấp
 - Applications: Ứng dụng trong truyền thông đa phương tiện và video số
- **MPEG-21 ~ ongoing**

MPEG

Format	Video Parameters	Compressed bit rate	
SIF	352x240 at 30Hz	1.2-3 Mbps	→ MPEG-1
CCIR 601	720x486 at 30Hz	5-10 Mbps	→ MPEG-2
EDTV	960x486 at 30Hz	7-15 Mbps	
HDTV	1920x1080 at 30Hz	20-40 Mbps	

MPEG

- Tiêu chuẩn MPEG là sự kết hợp giữa nén trong ảnh và nén liên ảnh.
- Phương pháp nén có tổn hao dựa trên sự biến đổi DCT
- Bù chuyển động.
- MPEG dùng biểu diễn màu bằng YC_rC_b .

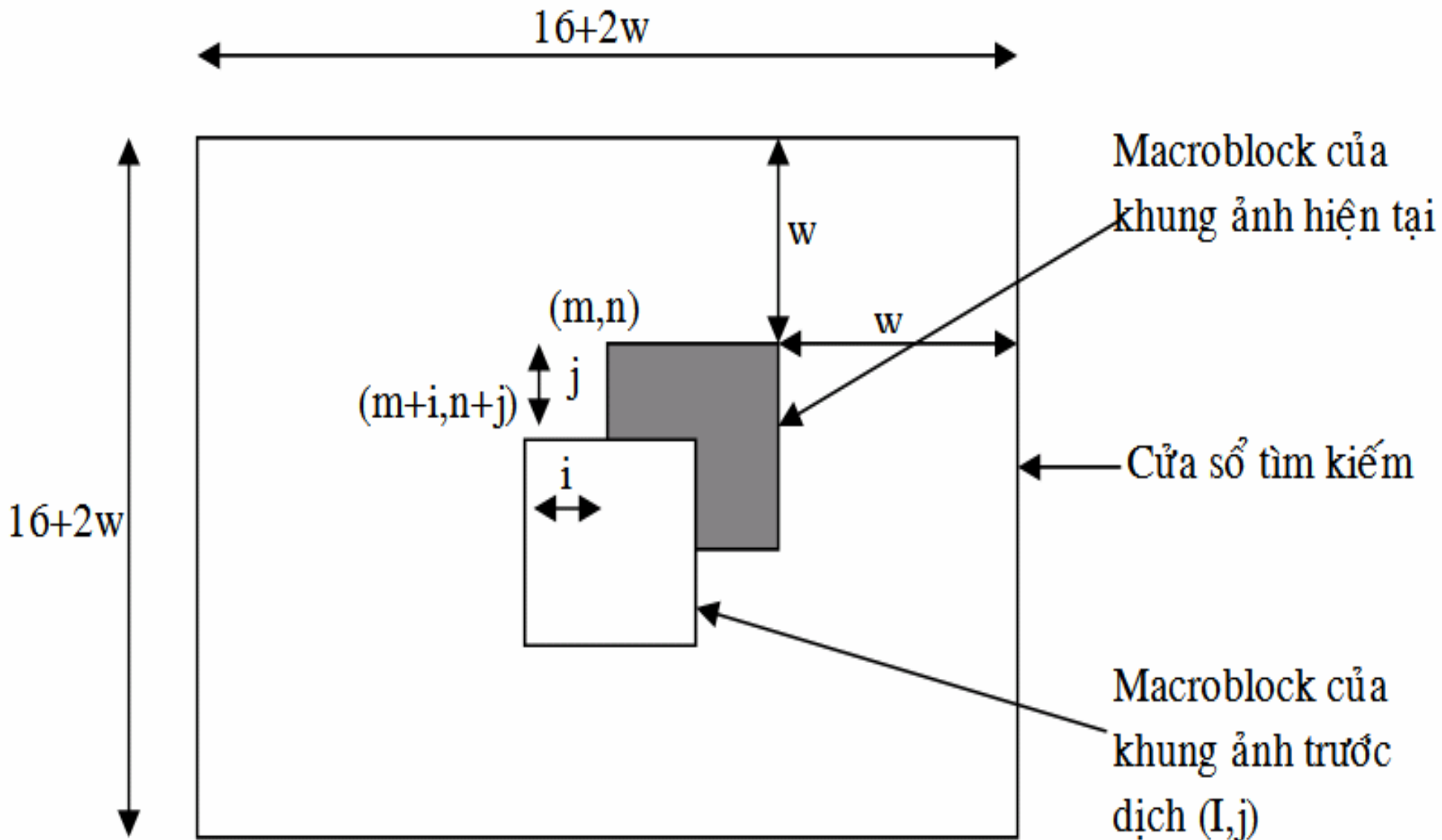
Các phương pháp bù chuyển động

- Phương pháp ước đoán bù chuyển động: Giả thiết ảnh hiện tại là một phép biến đổi từ ảnh trước đó, nghĩa là biên độ và hướng dịch chuyển không cần thiết phải giống ảnh trước đó.
- Phương pháp nội suy bù chuyển động là kỹ thuật nhiều độ phân giải: chỉ mã hoá một tín hiệu phụ với độ phân giải thấp (khoảng $1/2$ đến $1/3$ tốc độ khung). Ảnh có độ phân giải đầy đủ sẽ được xây dựng lại qua nội suy ảnh có độ phân giải thấp cộng thêm thành phần sửa sai. Đơn vị xử lý ảnh mà MPEG sử dụng là macroblock (MB) 16×16 điểm ảnh.

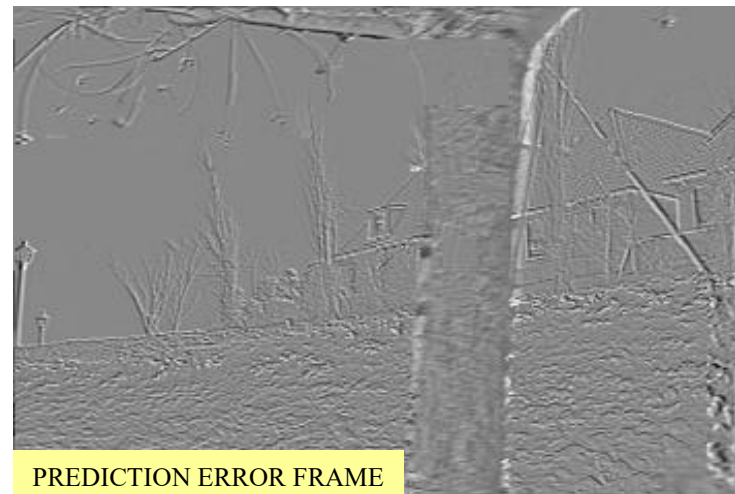
Các phương pháp bù chuyển động

- Trong ảnh mã hoá nội suy, các MB có thể là loại nén trong khung hay nén liên khung. Trong kỹ thuật ước đoán chuyển động, nếu sử dụng kỹ thuật so sánh khối (BMA - Block Matching Algorithm) thì sẽ thu được các vectơ chuyển động theo tiêu chí tối thiểu hoá sai số giữa khối cần tìm vectơ chuyển động và mỗi khối ứng cử.

Các phương pháp bù chuyển động



Các phương pháp bù chuyển động



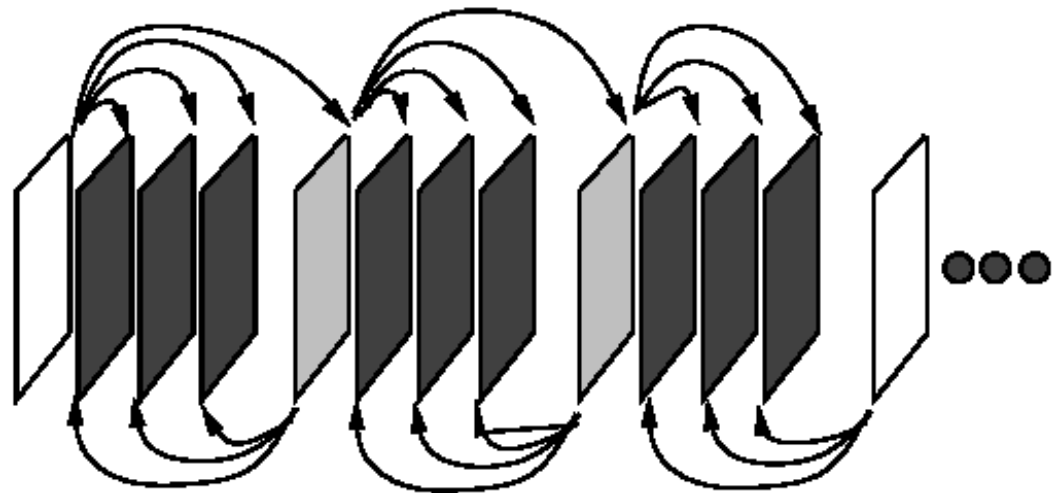
MPEG-1 Picture Types and Group-of-Pictures

- A Group-of-Picture (GOP) contains 3 types of frames (I/P/B)
- Frame order

$\underline{I_1} \underline{BBB} \underline{P_1} \underline{BBB} \underline{P_2} \underline{BBB} I_2 \dots$

- Coding order

$I_1 P_1 BBB P_2 BBB I_2 BBB \dots$



Intra-frame



Forward predicted frame



Bidirectionally predicted frame

(Priority: Intra > Forward > Bidirectional)

Motion Compensation

Các cấu trúc ảnh

- Ảnh loại I (Intra-picture)
- Ảnh loại P (Predicted-picture)
- Ảnh loại B (Bi-directional predicted picture)

Các cấu trúc ảnh

- **Ảnh loại I (Intra-picture):** Là ảnh được mã hóa riêng, tương tự như việc mã hóa ảnh tĩnh trong JPEG. Ảnh I chứa đựng dữ liệu để tái tạo lại toàn bộ hình ảnh vì chúng được tạo thành bằng thông tin của chỉ một ảnh và để dự báo cho ảnh B,P. Ảnh I cho phép truy cập ngẫu nhiên, tuy nhiên cho tỷ lệ nén thấp nhất.

Các cấu trúc ảnh

- **Ảnh loại P (Predicted-picture):** Là ảnh được mã hóa có bù chuyển động từ ảnh I hoặc ảnh P phía trước. Ảnh P cung cấp cho hệ số nén cao hơn ảnh I và có thể sử dụng làm một ảnh so sánh cho việc bù chuyển động cho các ảnh P và B khác.

Các cấu trúc ảnh

- **Ảnh loại B (Bi-directional predicted picture):**
Là ảnh được mã hóa sử dụng bù chuyển động từ các ảnh I hoặc P ở phía trước và ở phía sau. Ảnh B cho tỷ lệ nén cao nhất.

Các cấu trúc ảnh

- **Ảnh loại D (Dc-coded picture):** Là ảnh được sử dụng trong MPEG-1 và MPEG-4 nhưng không được sử dụng trong MPEG-2. Nó giống như ảnh I, tuy nhiên chỉ có thành phần một chiều ở đầu ra DCT được thể hiện. Nó cho phép dò tìm nhanh nhưng chất lượng ảnh thấp.

Các cấu trúc ảnh

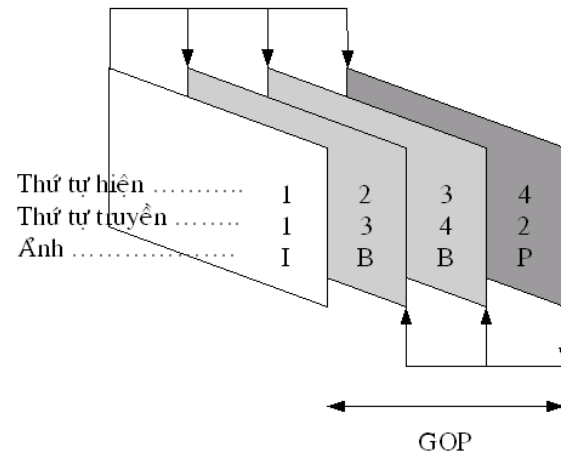
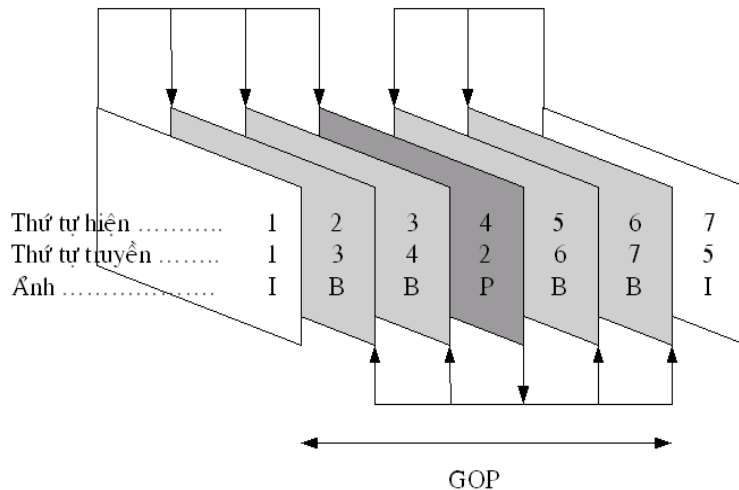
- **Nhóm ảnh (GOP):** Đối với chuẩn MPEG, chất lượng ảnh không những phụ thuộc vào tỷ lệ nén trong từng khuôn hình mà còn phụ thuộc vào độ dài của nhóm ảnh. Nhóm ảnh (GOP-Group of picture) là khái niệm cơ bản của MPEG. Nhóm ảnh là đơn vị mang thông tin độc lập của MPEG

Các cấu trúc ảnh

- **MPEG sử dụng ba loại ảnh I, B, P.**
 - Trong đó, ảnh P, B không phải là một ảnh hoàn chỉnh mà chỉ chứa sự khác biệt giữa ảnh đó và ảnh xuất hiện trước nó (đối với ảnh P) hay sự khác biệt đối với cả khuôn hình xuất hiện trước và sau nó (đối với ảnh B).
 - Để có một khuôn hình hoàn chỉnh ảnh P và B cần có dữ liệu từ các ảnh lân cận, chính vì vậy đối với MPEG một khái niệm mới là GOP (nhóm ảnh) được sử dụng. Mỗi GOP bắt buộc phải bắt đầu bằng một ảnh hoàn chỉnh I và tiếp sau nó là một loạt các ảnh P và B. Nhóm ảnh có thể mở (Open) hoặc đóng (Closed).

Các cấu trúc ảnh

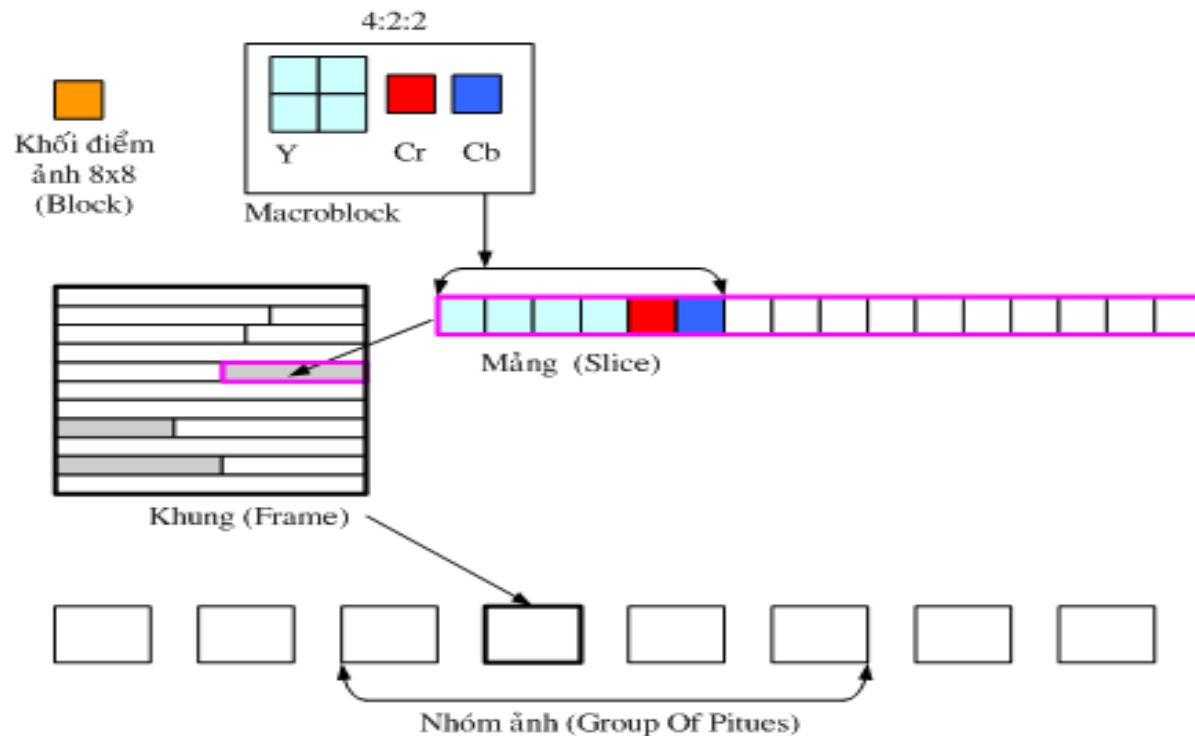
- Ảnh loại I (Intra-picture)
- Ảnh loại P (Predicted-picture)
- Ảnh loại B (Bi-directional predicted picture)



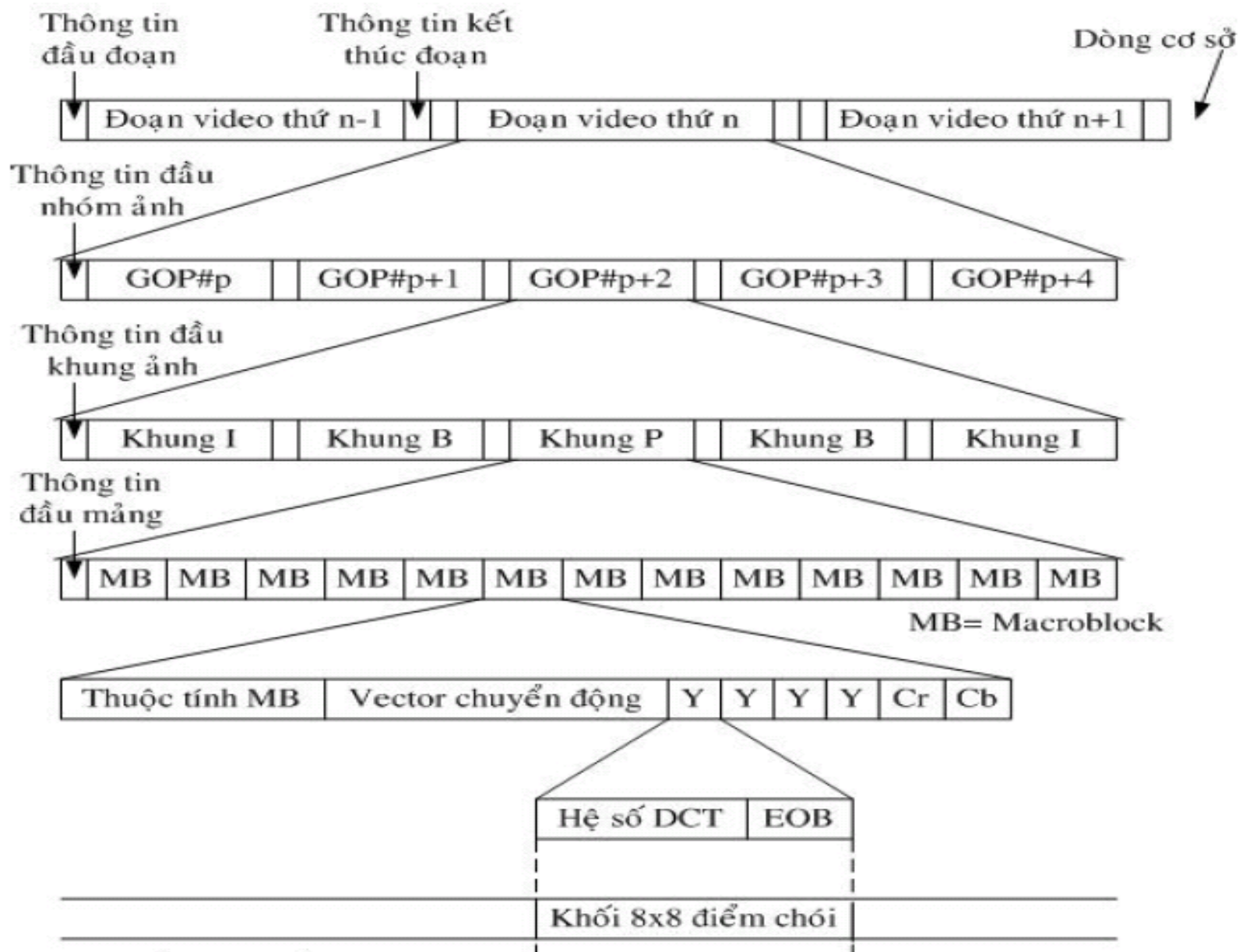
Các cấu trúc ảnh

- Trong Hình trên, ảnh P (ảnh 4) được dự báo trước trên cơ sở ảnh I (ảnh 1). Ảnh B được dự đoán từ hai hướng, ảnh B (ảnh 2) và ảnh B (ảnh 3) được dự đoán từ hai ảnh I (ảnh 1) và ảnh P (ảnh 4). Ảnh B (ảnh 5,6) được dự đoán từ ảnh P (ảnh 4) và ảnh I tiếp theo (ảnh 6). Một điều chú ý là thứ tự truyền ảnh và hiện ảnh trên màn hình là không giống nhau.

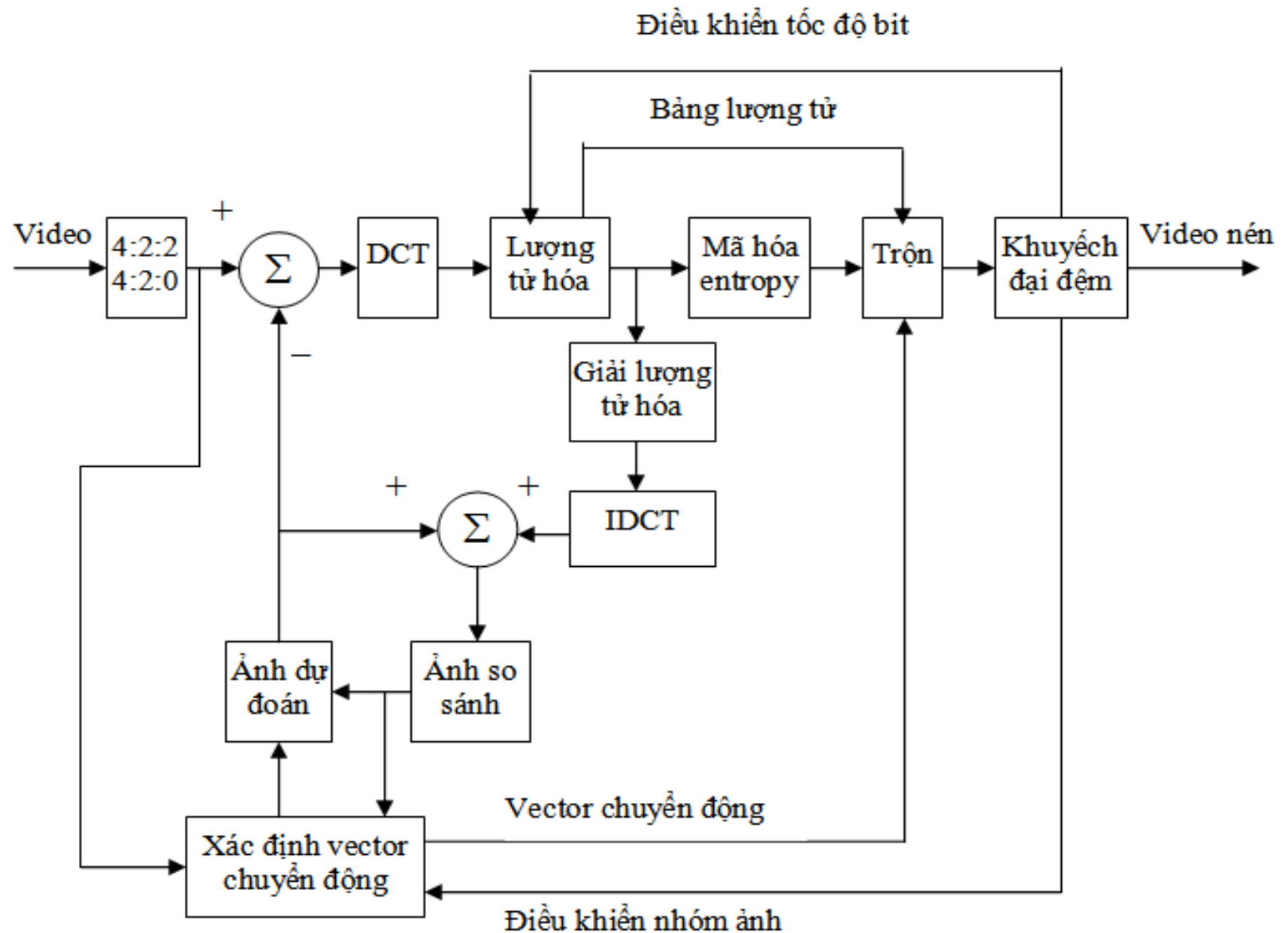
Cấu trúc dòng bit MPEG



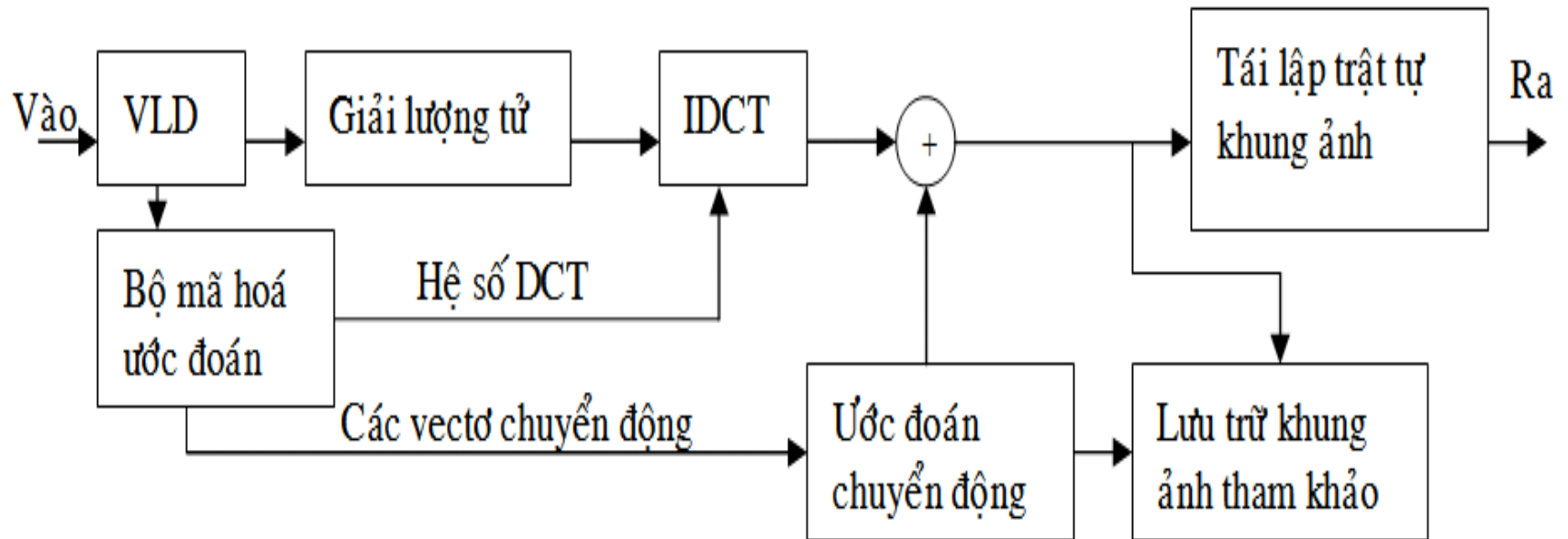
Kiến trúc dòng dữ liệu MPEG



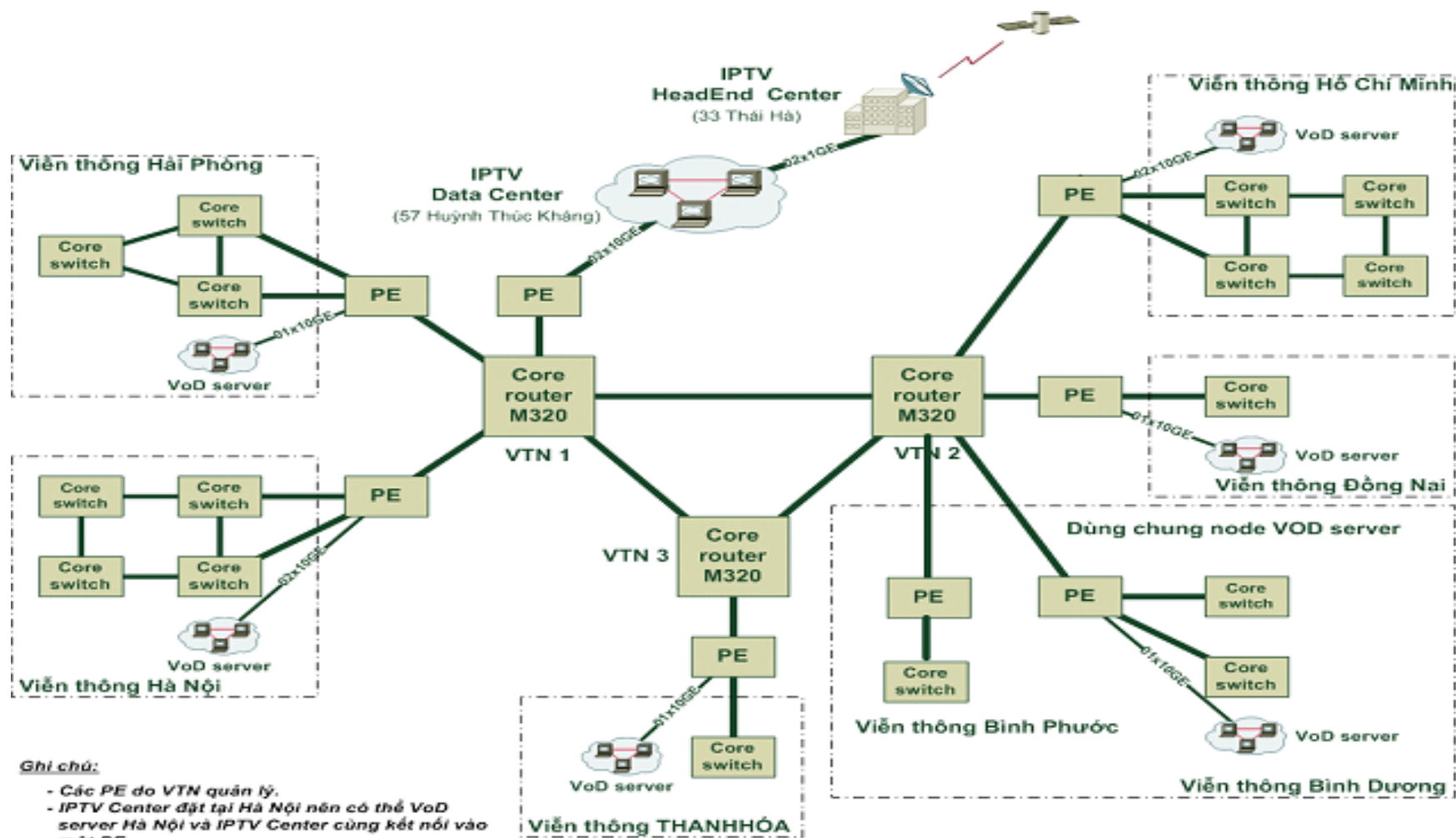
Bộ mã hóa MPEG tiêu biểu



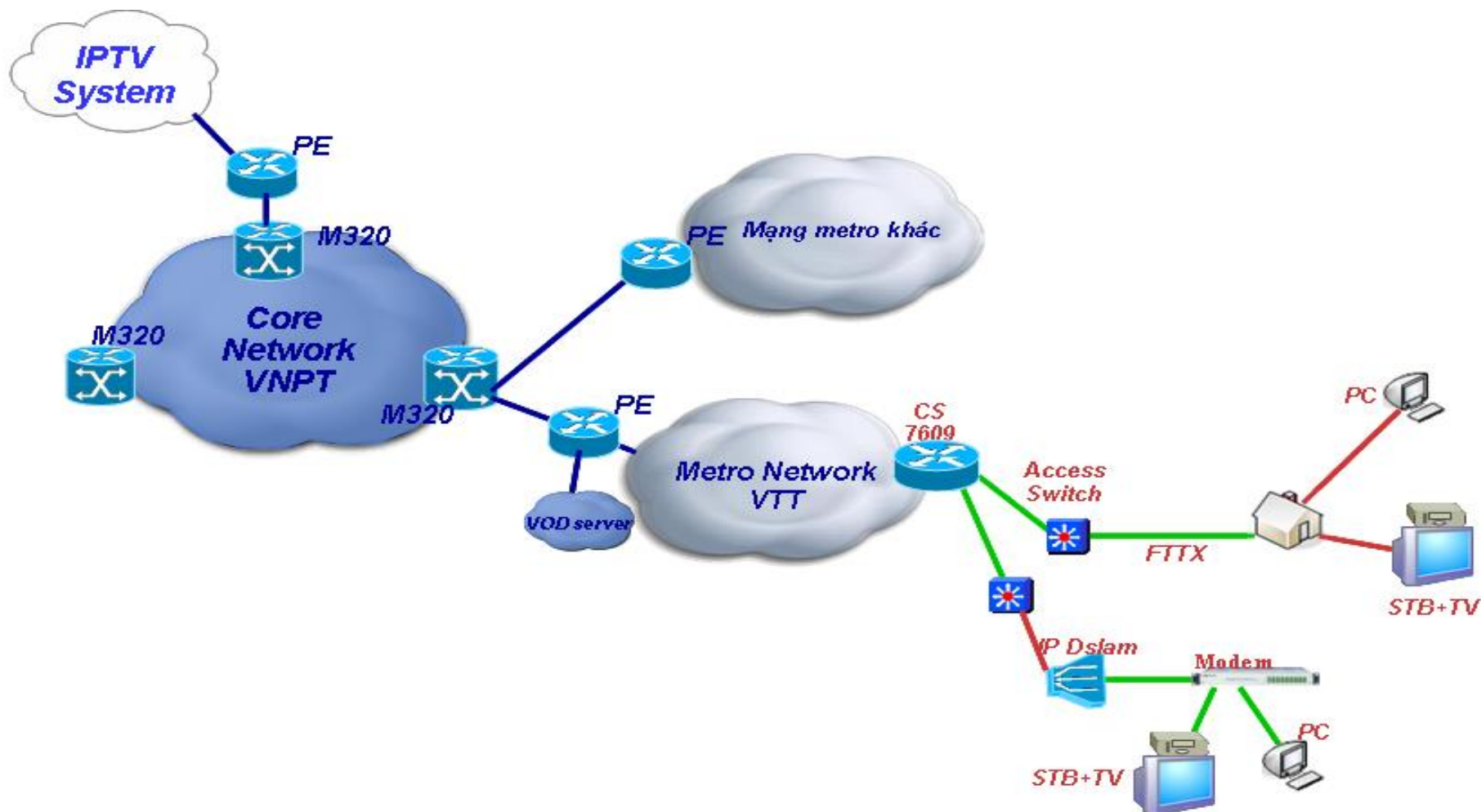
Bộ mã hóa MPEG tiêu biểu



Ứng dụng MPEG trong truyền thông đa phương tiện



Ứng dụng MPEG trong truyền thông đa phương tiện



Ứng dụng MPEG trong truyền thông đa phương tiện

