

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



**NGUYỄN VIỆT QUÂN**

**NGHIÊN CỨU VÀ ĐÁNH GIÁ HIỆU NĂNG CỦA  
CHUẨN NÉN VIDEO H.265/HEVC VỚI CÁC CHUỖI  
VIDEO ĐẦU VÀO CÓ ĐẶC TÍNH THAY ĐỔI**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

HÀ NỘI – 2020

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



**NGUYỄN VIỆT QUÂN**

**NGHIÊN CỨU VÀ ĐÁNH GIÁ HIỆU NĂNG CỦA  
CHUẨN NÉN VIDEO H.265/HEVC VỚI CÁC CHUỖI  
VIDEO ĐẦU VÀO CÓ ĐẶC TÍNH THAY ĐỔI**

**Chuyên ngành: Kỹ thuật viễn thông**

**Mã số: 8.52.02.08**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**  
*(Theo định hướng ứng dụng)*

**NGƯỜI HƯỚNG DẪN KHOA HỌC:**  
**TS. VŨ HỮU TIẾN**

**HÀ NỘI - 2020**

## LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả mô phỏng nêu trong luận văn là trung thực và chưa từng được công bố trong bất kỳ công trình nào khác.

*Hà Nội, tháng 05 năm 2020*

Tác giả luận văn

**Nguyễn Việt Quân**

## **LỜI CẢM ƠN**

Để hoàn thành luận văn này lời đầu tiên tôi xin tỏ lòng biết ơn sâu sắc đến TS.Vũ Hữu Tiến đã tận tình hướng dẫn và chỉ bảo trong suốt quá trình thực hiện.

Tôi chân thành cảm ơn các Thầy, Cô trong khoa Đào Tạo Sau Đại Học, Học viện Công nghệ Bưu chính Viễn thông Hà Nội đã tận tình giúp đỡ tôi trong quá trình hai năm tôi học tập và nghiên cứu.

*Hà Nội, ngày 15 tháng 05 năm 2020*

**Nguyễn Viết Quân**

## MỤC LỤC

<i>CHƯƠNG 1: TỔNG QUAN VỀ MÃ HÓA VIDEO</i>	10
1.1. Giới thiệu chung	10
1.2. Nguyên tắc mã hóa video	11
1.2.1. Nguyên tắc chính của mã hóa video	11
1.2.2 Kỹ thuật giảm dư thừa thông tin trong miền không gian	12
1.2.3 Kỹ thuật giảm dư thừa thông tin trong miền thời gian	15
1.2.4 Sơ đồ tổng quát của mã hóa video	18
1.3 Giải mã hóa video	21
1.4 Kết luận chương	21
<i>CHƯƠNG 2: TỔNG QUAN VỀ MÃ HÓA VIDEO, GIẢI MÃ VIDEO ĐỘ PHÂN GIẢI CAO H.265/HEVC</i>	23
2.1 Tổng quan về mã hóa phân giải cao	23
2.1.1 Giới thiệu chuẩn nén H.265	23
2.1.2 Cấu trúc mã hóa	23
2.2 So sánh hiệu năng nén của h.265 với một số chuẩn nén trước	42
2.3 Kết luận chương	46
<i>CHƯƠNG 3: PHÂN TÍCH ĐÁNH GIÁ HIỆU NĂNG CỦA CHUẨN NÉN VIDEO HEVC TRÊN PHẦN MỀM HEVC TEST MODEL</i>	47
3.1 Giới thiệu phần mềm mô phỏng HM	47
3.2 Các chuỗi video dùng để đánh giá	47
3.3 Các tham số để đánh giá hiệu năng nén	49
3.4 Phân tích kết quả mô phỏng	50
3.4.1 Mã hóa video trong chế độ nội ảnh (Intra mode)	50
3.4.2 Mã hóa video trong chế độ liên ảnh (Inter mode)	51
3.5 So sánh hai chế độ liên ảnh.	53
3.6 Chế độ mã hóa Inter với kích thước chuỗi khác nhau	55
3.7 Kết luận chương	56

## DANH MỤC HÌNH ẢNH

Hình 1.1: Sơ đồ khối của bộ codec DPCM trong xử lý video	12
Hình 1.2: Biểu diễn lượng tử	14
Hình 1.3: Biểu diễn 2 bộ lượng tử UTQ (a) và UTQ-DZ(b)	15
Hình 1.4: (a) Sự khác biệt giữa khung hình hiện thời và trước đó; (b) Ảnh sau khi được bù chuyển động	16
Hình 1.5: Vùng tìm vector chuyển động của macro block hiện thời	17
Hình 1.6: Sơ đồ nguyên lý tổng quát của bộ mã hóa video	19
Hình 1.7: Sơ đồ giải mã hóa tín hiệu video	21
Hình 2.1: Sơ đồ khối bộ mã hóa HEVC	24
Hình 2.2 Cấu trúc CTU	25
Hình 2.3 Cấu trúc CTB	25
Hình 2.4 Cấu trúc CTB chia nhỏ	26
Hình 2.5 Cấu trúc PB	27
Hình 2.6 Cấu trúc TB	27
Hình 2.7 Chế độ chia tách một CB thành PBs	29
Hình 2.8 Chia nhỏ một CTB thành các CB, TB	31
Hình 2.9 Chia nhỏ của một ảnh thành mảng (a) và tile (b) Minh họa quá trình xử lý song song wavefront (c)	32
Hình 2.10 Các chế độ và các hướng cho dự đoán hình trong ảnh	32
Hình 2.11 Vị trí lấy mẫu số nguyên và phân đoạn cho phép nội suy thành phần chói	36
Hình 2.12 Ba chế độ quét hệ số trong HEVC (a) Quét chéo lên bên phải. (b) Quét theo chiều ngang. (c) Quét theo chiều dọc.	39
Hình 2.13 Bốn dạng gradient sử dụng trong SAO	40
Hình 2.14 So sánh về hiệu năng nén của H.265 với một số chuẩn nén trước	42
Hình 2.15 So sánh H.264/AVC và H.265/HEVC khung hình video ở độ phân giải HD 720p	43
Hình 2.16 Thời gian tải video H.265 so với video H.264 với cùng một tốc độ mạng	43
Hình 2.17 Kích thước file nén qua các thời kỳ	44
Hình 2.18 Kết quả nén của H264 với video Akiyo	44
Hình 2.19 Kết quả nén của H265/HEVC với video Akiyo	45
Hình 2.20 Kết quả nén của H264 với video Big Buck Bunny	45

<i>Hình 2.21 Kết quả của H265/HEVC với video Big Buck Bunny Ba</i>	46
<i>Hình 3.1 Thực hiện lệnh HEVC Encoder</i>	47
<i>Hình 3.2 Các frame đầu tiên của chuỗi Akiyo, Claire, Mother-daughter</i>	48
<i>Hình 3.3 Các frame đầu tiên của chuỗi News, Hall, Coastguard</i>	48
<i>Hình 3.4 Các frame đầu tiên của chuỗi Foreman, Carphone, Soccer</i>	48
<i>Hình 3.5 So sánh hiệu năng nén của HEVC cho các chuỗi khác nhau</i>	50
<i>Hình 3.6 So sánh hiệu năng nén của HEVC cho các chuỗi khác nhau</i>	52
<i>Hình 3.7 So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Akiyo</i>	53
<i>Hình 3.8 So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Mobile</i>	54
<i>Hình 3.9 So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Foreman</i>	54
<i>Hình 3.10 Frame đầu tiên của các chuỗi video thử nghiệm</i>	55

## DANH MỤC CÁC TỪ VIẾT TẮT

<b>Từ/Cụm từ viết tắt</b>	<b>Tiếng Anh</b>	<b>Tiếng Việt</b>
AVC	Advanced Video Coding	Nâng cao mã hóa video
BDPSNR	Bjontegaard delta PSNR	Chênh lệch trung bình PSNR
CABAC	Context-Adaptive Binary Arithmetic Coding	Mã hóa số học nhị phân thích nghi dựa trên ngữ cảnh.
CAVLC	Context – Adaptive Variable Length Coding	Mã hóa độ dài biến đổi thích nghi ngữ cảnh.
CODEC	COde - DECode	Bộ nén và giải nén video
IDCT	Inverse Discrete Cosine Transform	Biến đổi cosin rời rạc ngược
MB	MacroBlock	Macroblock – khối ảnh nén cơ bản của H.264
MPEG	The Moving Picture Experts Group	Nhóm chuyên gia ảnh động
MSE	Mean Squared Error	Sai số bình phương trung bình
NAL	Network Abstraction Layer	Lớp trừu tượng hóa mạng
PSNR	Peak Signal-to-Noise Ratio	Tỉ số tín hiệu trên nhiễu đỉnh
QP	Quantisation Paramater	Tham số lượng tử
RDO	Rate-Distortion Optimization	Tối ưu hóa lệ nén và độ méo
VCL	Video Coding Layer	Lớp mã hóa video



## DANH MỤC BẢNG BIỂU

<i>Bảng 2.1 Các hệ số bộ lọc cho phép nội suy lấy mẫu phân đoạn chói</i>	36
<i>Bảng 2.2 Các hệ số bộ lọc cho phép nội suy lấy mẫu phân đoạn chroma</i>	38
<i>Bảng 2.3 Các loại mẫu Edgeldx trong các lớp cạnh biên SAO</i>	42
<i>Bảng 3.1 Các tham số của môi trường mô phỏng</i>	47
<i>Bảng 3.2 So sánh chất lượng ảnh (PSNR-dB) sau giải nén của các chuỗi với các tham số lượng tử (QP) khác nhau</i>	50
<i>Bảng 3.3 So sánh Bitrate (Kbps) của các chuỗi với các tham số lượng tử (QP) khác nhau</i>	519
<i>Bảng 3.4 So sánh chất lượng ảnh (PSNR-dB) sau giải nén của các chuỗi với các tham số lượng tử (QP) khác nhau</i>	52
<i>Bảng 3.5 So sánh Bitrate (Kbps) của các chuỗi với các tham số lượng tử (QP) khác nhau</i>	53
<i>Bảng 3.6 Các tham số của chuỗi video thử nghiệm</i>	55
<i>Bảng 3.7 Hiệu năng của các chuỗi video thử nghiệm trong thuật toán đề xuất</i>	56

## MỞ ĐẦU

Trong những năm gần đây khi mà lĩnh vực công nghệ và truyền thông ngày càng phát triển cùng với sự phát triển mạnh của internet toàn cầu thì việc truyền tải và lưu trữ video cũng ngày càng trở nên phổ biến hơn. Cùng với đó, chất lượng của phần cứng ngày càng được nâng cao và giá thành lại ngày càng hợp lý giúp cho việc tải hay trải nghiệm một chương trình, bộ phim ngày càng trở nên dễ dàng nhanh chóng và mượt mà hơn.

Tuy nhiên chỉ xem thôi thì chưa đủ, khi chất lượng cuộc sống ngày càng nâng cao kéo theo nhu cầu nghe nhìn vì đó cũng ngày càng khắt khe hơn. Nếu như vài năm trước việc thưởng thức một bộ phim ở độ phân giải 480p hay HD 720p đã đủ sức làm thỏa mãn người xem, thì giờ đây phim với độ phân giải Full HD 1080p đã trở nên quá đổi bình thường. Và khi những màn hình 4K được cho ra đời với độ phân giải siêu cao gấp 4 lần Full HD thì nhu cầu trải nghiệm của người dùng cũng ngày càng thể hiện mạnh mẽ hơn và cũng ngày càng khắt khe hơn. Cùng với màn hình 4K, phim 4K cũng được cho ra đời mang đến cho người dùng những trải nghiệm cực kì ấn tượng, cực kì kinh ngạc với chất lượng hình ảnh siêu thực làm cho người xem như lạc vào một thế giới khác, thế giới 4K.

Nhu cầu sử dụng nội dung video chất lượng của con người ngày càng cao dẫn đến sự ra đời của các chuẩn video mới có chất lượng cao như 2K, 4K, 8K. Bên cạnh đó, nhu cầu sử dụng video hiện nay còn phổ biến trong điều kiện truyền tải online, không chỉ qua các kênh truyền hữu tuyến chất lượng cao như cáp quang, mà cả kênh truyền vô tuyến WLAN, mạng di động 3G, 4G cũng được khai thác triệt để mọi lúc mọi nơi. Những yêu cầu này dẫn đến việc phải xử lý dữ liệu video trước khi truyền tải nhằm nâng cao chất lượng hình ảnh đồng thời giảm dung lượng truyền dẫn là một yêu cầu cấp thiết cần giải quyết.

Tuy nhiên, để có được những trải nghiệm tuyệt vời đó thì người dùng cũng sẽ phải đầu tư thêm chi phí cho việc nâng cấp các thiết bị lưu trữ cũng như đường truyền internet. Vì thế nhu cầu một chuẩn nén video mới hiệu quả hơn nhằm giảm tải áp lực cho hệ thống ngày càng bức thiết hơn. Và H.265/HEVC đã được cho ra đời nhằm đáp ứng những nhu cầu trên.

H.265/HEVC là một chuẩn nén video hoàn toàn mới với tên gọi đầy đủ là High Efficiency Video Coding (HEVC) được Ủy ban Viễn thông Quốc tế ITU-T thông qua và bắt đầu được các nhà phát triển đưa vào sản phẩm thương mại. H.265 hứa hẹn mang lại khả năng nén cao gấp đôi so với người tiền nhiệm H.264/AVC (Advance Video Coding) hiện đang được dùng phổ biến khi chiếm tới 80% các nội dung video lưu trữ hiện nay.

HEVC có tỷ lệ nén dữ liệu gấp đôi so với H.264/MPEG-4 AVC ở cùng một mức độ chất lượng video. Nói cách khác nó có thể được sử dụng để cung cấp chất lượng hình ảnh được cải thiện đáng kể khi xét cùng một bitrate. H.265/HEVC cũng mang lại những cải tiến về âm thanh, không gian, màu sắc và quan trọng nhất nó có khả năng giám sát một phạm vi hoạt động nâng cao.

H.265/HEVC cũng mang lại những cải tiến về âm thanh, không gian, màu sắc và quan trọng nhất nó có khả năng giám sát một phạm vi hoạt động nâng cao. Với chuẩn mã hóa này trên các camera thì việc xuất hiện camera 4K-UltraHD và 8k-UltraHD là điều có thể xảy ra trong tương lai, khi đó video chất lượng cao không chỉ còn là một loại hình phương tiện giải trí, mà nó còn đóng góp giá trị khai thác thực tiễn rất lớn cho lĩnh vực giám sát, giao thông và quốc phòng an ninh.

Chính vì những lý do trên, học viên quyết định chọn đề tài: ***“Nghiên cứu và đánh giá hiệu năng của chuẩn nén video H.265/HEVC với các chuỗi video đầu vào có đặc tính thay đổi”*** làm luận văn thạc sỹ. Trong suốt quá trình nghiên cứu và triển khai đề tài, học viên nhận thấy hiệu quả nén của chuẩn mới tăng đáng kể so với những chuẩn cũ, trong nội dung chương 3 của luận văn học viên đã đi sâu và phân tích đánh giá hiệu năng của H.265/HEVC dựa trên phần mềm mã nguồn mở HEVC Test Model.

Nội dung nghiên cứu của luận văn bao gồm các phần:

*CHƯƠNG I: Tổng quan về mã hóa Video.*

*CHƯƠNG II: Tổng quan về mã hóa, giải mã video độ phân giải cao H.265/HEVC.*

*CHƯƠNG III: Phân tích đánh giá hiệu năng của chuẩn nén Video HEVC*

# CHƯƠNG 1. TỔNG QUAN VỀ MÃ HÓA VIDEO

## 1.1 Giới thiệu chung

Nguyên tắc cơ bản để tạo ra video từ trước đến nay là phát liên tục nhiều hình ảnh trong một giây. Khi những hình ảnh chuyển động liên tục nhau (motion picture) được phát liên tục với tốc độ trình chiếu từ 18 hình/giây trở lên, khi đó mắt của chúng ta ghi nhận đó là một chuỗi hình ảnh liên tục, tạo thành một đoạn phim.

Người ta đã chuẩn hóa video phổ biến ở ba chuẩn khung hình gồm 24 fps, 25 fps và 30 fps (fps - frame per second - số khung hình/giây). Tại Việt Nam và châu Âu các nhà làm phim và truyền hình thường dùng chuẩn 25 hình/giây (gọi là hệ PAL). Tại Mỹ thường dùng chuẩn là 30 hình/giây (gọi là hệ NTSC).

Không chỉ trong lĩnh vực điện ảnh video còn cung cấp cho ta thông tin trên toàn thế giới một cách trực quan như thể chính mình đang trải qua những điều đó qua các bản tin, chương trình hàng ngày trên truyền hình,... Mọi người thường ưa thích cách thu thập thông tin qua video hơn là qua sách báo, lý do đơn giản vì nó giúp ta tiết kiệm thời gian đồng thời thông tin được đưa đến dưới dạng hình ảnh sẽ dễ tiếp thu và tránh được sự nhàm chán. Cũng vì lý do đó mà hiện nay các phương pháp giáo dục thông qua video ngày càng xuất hiện nhiều hơn.

Video cũng mang lại những lợi ích to lớn trong việc đảm bảo an ninh. Sử dụng camera giám sát ngày càng phổ biến hơn không chỉ ở các công ty, tòa nhà lớn mà còn cả các cửa hàng nhỏ, hộ gia đình... Đây là một biện pháp hữu hiệu để đảm bảo an ninh, theo dõi giám sát và ngăn chặn các sự cố. Về lĩnh vực nghiên cứu khoa học, video về quá trình phát triển của một sinh vật trong khoảng thời gian lớn hoặc những phản ứng xảy ra ở những nơi mà con người không thể trực tiếp quan sát... là một tư liệu quý giá. Trong kinh doanh việc giới thiệu sản phẩm với người tiêu dùng là một việc vô cùng quan trọng và các video quảng cáo đang làm tốt công việc này. Việc khám và mổ nội soi là một thành tựu lớn của y tế đó là sự kết hợp hoàn hảo giữa công nghệ video và lĩnh vực y khoa, giúp các bác sĩ có thể quan sát bên trong cơ thể người bệnh phát hiện các tác nhân gây bệnh.

Bên cạnh đó các ứng dụng như video call, video conference là những giải pháp đàm thoại trên mạng IP hữu ích. Các cuộc họp có thể diễn ra mà các thành viên không cần phải ở chung một địa điểm điều đó tiết kiệm được một lượng lớn thời gian và kinh phí. Video xuất hiện trong hầu hết mọi lĩnh vực của đời sống và tầm quan trọng của nó là không thể phủ nhận.

Kỹ thuật nén ảnh số đang đóng một vai trò cực kỳ quan trọng trong các hệ thống viễn thông và đa phương tiện để giải quyết vấn đề băng thông của đường truyền. Các kỹ thuật mã hóa video đều cố gắng làm giảm lượng thông tin không cần thiết cho một chuỗi các bức ảnh mà không làm giảm chất lượng của nó đối với người xem. Nói

chung, tín hiệu video thường chứa đựng một lượng lớn các thông tin dư thừa. Chúng thường được chia thành hai loại: dư thừa về mặt không gian (bên trong từng frame) và dư thừa về mặt thời gian (giữa các frame trong chuỗi video). Mục đích của mã hóa video là nhằm làm giảm số bit khi lưu trữ và khi truyền bằng cách phát hiện để loại bỏ các lượng thông tin dư thừa này và dùng các kỹ thuật mã hoá entropy để tối thiểu hoá lượng tin quan trọng cần giữ lại.

Mã hóa video cho phép nén video có dung lượng hàng chục GB giảm xuống chỉ còn vài trăm MB thuận tiện cho việc quản lý và lưu trữ trên các phương tiện giải trí đa dạng theo tỷ lệ mã hóa. Một vấn đề khác cần quan tâm đó là bản thân quá trình tính toán cũng cần tài nguyên nên điều này cần được xem xét trong các tình huống cụ thể, ví dụ trong trường hợp một ứng dụng thời gian thực với yêu cầu độ trễ thấp thì một thuật toán có tỷ số mã hóa rất cao nhưng phức tạp và tốn thời gian giải mã sẽ làm hỏng mục đích của việc mã hóa. Tỷ lệ mã hóa cao về kích thước dữ liệu sẽ đồng nghĩa với việc phải trả giá bằng chất lượng suy giảm và thời gian thực hiện mã hóa lâu hơn.

## **1.2 Nguyên tắc mã hóa video.**

### ***1.2.1. Nguyên tắc chính của mã hóa video.***

Các nghiên cứu trước đây đã chỉ ra rằng giữa các frame trong chuỗi video và giữa các pixel trong cùng một frame có một mối tương quan nhất định. Dựa vào các mối tương quan này chúng ta có thể thực hiện việc mã hóa tín hiệu video mà không làm ảnh hưởng tới độ phân giải của ảnh. Ngoài ra, khai thác đặc điểm của mắt người là kém nhạy cảm với một số thông tin hình ảnh theo không gian và thời gian nên có thể loại bỏ thông tin này trong quá trình mã hóa. Đây chính là kỹ thuật mã hóa tổn hao để tiết kiệm băng thông trong khi vẫn đảm bảo chất lượng video ở mức có thể chấp nhận được. Chương này được tham khảo tại tài liệu [1]

Trong quá trình mã hóa ảnh tĩnh, kỹ thuật mã hóa sử dụng mối tương quan theo không gian giữa các pixel trong ảnh. Kỹ thuật này gọi là mã hóa “nội ảnh” (Intraframe). Thuật ngữ này có ý nghĩa rằng trong quá trình mã hóa ảnh, thông tin được sử dụng chỉ trong phạm vi nội tại bức ảnh đó. Đây là kỹ thuật cơ bản của chuẩn mã hóa JPEG. Trường hợp nếu mối tương quan theo thời gian được khai thác thì kỹ thuật mã hóa được gọi là mã hóa “liên ảnh” (Interframe). Khi đó thông tin được sử dụng để mã hóa ảnh có thể nằm trên một bức ảnh trước hoặc sau trong chuỗi video. Đây là kỹ thuật được sử dụng trong các chuẩn mã hóa video như H.261, H.263, MPEG-1, 2 và 4. Nguyên lý của việc mã hóa video dựa trên các kỹ thuật giảm các dư thừa thông tin sau:

- Dư thừa thông tin trong miền không gian (Spatial redundancy): Dư thừa thông tin trong miền không gian xuất hiện giữa các pixel trong cùng một khung

hình (ví dụ sự tương đồng giữa các pixel). Thông tin dư thừa được loại bỏ bằng kỹ thuật mã hóa biến đổi (transform coding).

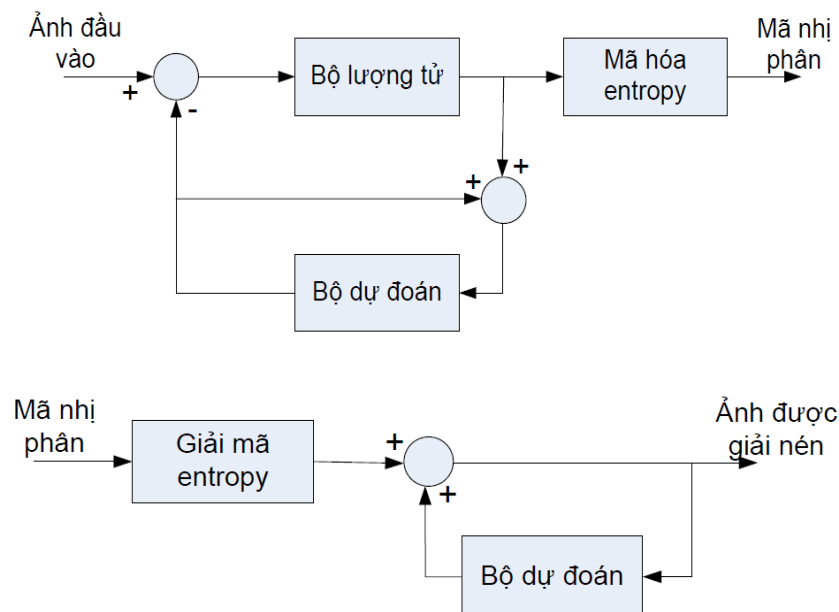
- Dư thừa thông tin trong miền thời gian (Temporal redundancy): Loại thông tin dư thừa này xuất hiện khi giữa các khung ảnh liên tiếp có những thông tin tương đồng. Để giảm dư thừa này người ta dùng kỹ thuật mã hóa sự khác biệt giữa các frame.

- Dư thừa thông tin trong dữ liệu ảnh sau khi mã hóa: Để loại bỏ dư thừa này người ta dùng mã entropy, cụ thể là mã có độ dài thay đổi (Variable Length Coding).

### 1.2.2 Kỹ thuật giảm dư thừa thông tin trong miền không gian.

#### a) Mã hóa dự báo

Tại thời điểm ban đầu, phương pháp giảm dư thừa không gian được đưa ra dựa trên việc dự báo giá trị của các pixel hiện tại dựa vào giá trị của pixel đã được mã hóa trước đó. Phương pháp này được gọi là “Điều chế xung mã sai phân” (Differential Pulse Code Modulation – DPCM). hình 1.1 mô tả sơ đồ khối của bộ mã hóa này. Theo sơ đồ, sự sai khác giữa giá trị của pixel hiện tại và giá trị dự báo từ bộ dự báo được lượng tử và mã hóa trước khi truyền đi. Tại phía giải mã, sự sai khác này được cộng với giá trị dự báo từ bộ dự báo để khôi phục lại giá trị đúng của pixel hiện tại. Trong trường hợp bộ lượng tử không được sử dụng thì kỹ thuật này được gọi là mã hóa không tổn hao (loss-less coding).



**Hình 1.1: Sơ đồ khối của bộ codec DPCM trong xử lý video**

Bộ dự báo cho kết quả tốt nhất nếu quá trình dự báo được dựa trên những giá trị của các pixel liên kế đã được mã hóa trước đó. Các pixel liên kế có thể nằm trong cùng frame (mã hóa nội ảnh) hoặc có thể nằm trong frame trước (mã hóa liên ảnh).

Nếu bộ dự báo sử dụng cả hai kỹ thuật trên thì được gọi là “mã hóa dự báo lai” (Hybrid predictive coding).

**b) Mã hóa biến đổi**

Mục đích của việc sử dụng mã hóa biến đổi là nhằm loại bỏ dư thừa thông tin theo không gian. Quá trình mã hóa được thực hiện bằng cách biến đổi giá trị của các pixel sang miền không gian khác trước khi loại bỏ những dữ liệu không cần thiết. Ý tưởng chính trong việc áp dụng mã hóa biến đổi vào việc mã hóa ảnh là năng lượng của hầu hết các bức ảnh tập trung chủ yếu tại miền tần số thấp và được biểu diễn bởi các hệ số sau biến đổi. Quá trình lượng tử hóa được sử dụng với mục đích giữ lại các hệ số trong miền tần số thấp và loại bỏ các hệ số trong miền tần số cao. Quá trình này sẽ không làm ảnh hưởng nhiều tới chất lượng của ảnh vì các hệ số trong miền tần số cao lưu giữ ít thông tin của ảnh hơn. Tuy nhiên quá trình này ít nhiều vẫn làm mất thông tin của ảnh gốc.

Hiện nay có nhiều phép biến đổi được đưa ra và đã thể hiện được các ưu điểm của nó như phép biến đổi Wavelet, biến đổi Karhune-Loève, biến đổi Cosin rời rạc (Discrete Cosine Transform – DCT). Trong các chuỗi ảnh video, các pixel có mối tương quan theo không gian (các pixel trong một ảnh) và thời gian (các pixel trong các ảnh liên tục). Do vậy, việc lựa chọn biến đổi DCT 3 chiều là phù hợp nhất. Tuy nhiên, trong thực tế, do phải lưu trữ các frame ảnh trong bộ nhớ đệm khi thực hiện biến đổi DCT 3 chiều nên sẽ không phù hợp trong truyền thông thời gian thực. Vì vậy phép biến đổi DCT 2 chiều thường được sử dụng hơn.

Phép biến đổi DCT 2 chiều có thể được tách ra thành hai phép biến đổi 1 chiều: theo chiều ngang và sau đó theo chiều dọc. Với một khối có kích thước  $M \times N$  pixel, phép biến đổi DCT được triển khai như sau:

$$F(u) = \frac{1}{\sqrt{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi(2x+1)u}{2N}\right), \quad u = 0, 1, \dots, N-1 \quad (1.1)$$

$$\text{Trong đó: } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{với } u = 0 \\ 0 & \text{với } u > 0 \end{cases}$$

$f(x)$  là giá trị cường độ của pixel thứ  $x$  và  $F(u)$  là hệ số biến đổi DCT 1 chiều.

Biến đổi DCT ngược như sau:

$$f(x) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos\left(\frac{\pi(2x+1)u}{2N}\right), \quad x = 0, 1, \dots, N-1 \quad (1.2)$$

Để thu được các hệ số biến đổi 2 chiều, ta thực hiện  $N$  phép biến đổi 1 chiều ( $N$  cột), mỗi phép biến đổi thực hiện cho  $M$  điểm ( $M$  hàng):

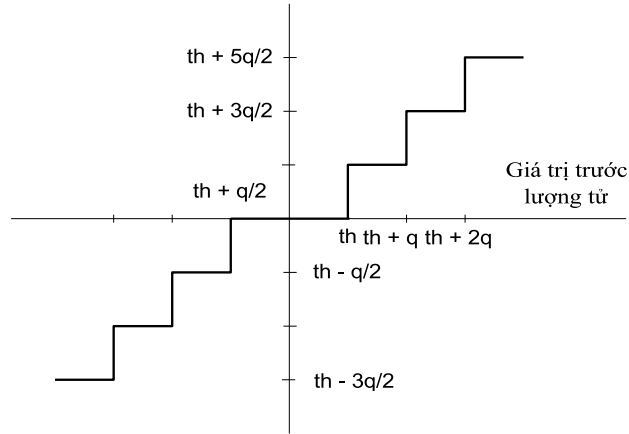
$$F(u, v) = \frac{1}{\sqrt{M}} C(v) \sum_{y=0}^{M-1} F(u, y) \cos\left(\frac{\pi(2y+1)}{2M}\right), \quad v = 0, 1, \dots, M-1 \quad (1.3)$$

Trong đó  $C(v)$  được tính tương tự như  $C(u)$  ở trên.

Theo cách tính trên, một khối có kích thước  $M \times N$  pixel sau khi được biến đổi sẽ có  $MN$  hệ số DCT. Hệ số  $F(0,0)$  được gọi là hệ số 1 chiều (DC). Các hệ số còn lại được gọi là hệ số xoay chiều (AC).

### c) Lượng tử hóa các hệ số DCT

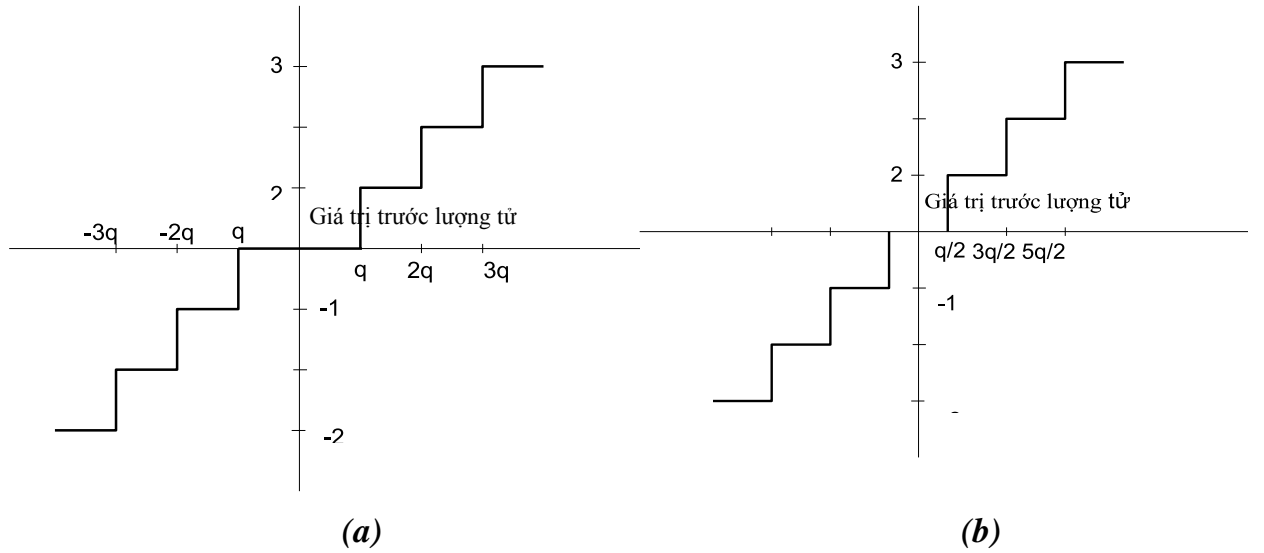
Chúng ta lưu ý rằng phép biến đổi các giá trị từ miền không gian pixel sang miền không gian khác chưa phải là mã hóa tín hiệu. Một khối 64 pixel qua phép biến đổi DCT ta thu được 64 hệ số DCT. Do tính trực giao của phép biến đổi, năng lượng trên cả hai miền pixel và miền không gian sau biến đổi đều bằng nhau. Tuy nhiên, phép biến đổi làm cho năng lượng của ảnh được tập trung chủ yếu trong vùng của các hệ số DCT gần với hệ số 1 chiều (vùng tần số thấp) trong khi phần lớn các hệ số khác mang năng lượng ít hơn. Hệ số DC có giá trị cao nhất. Các hệ số AC càng xa hệ số DC thì giá trị càng bé. Dựa vào đặc điểm của mắt người là ít nhạy cảm với các hình ảnh bị méo ở tần số cao nên bộ lượng tử được sử dụng nhằm loại bỏ đi các hệ số AC tại vùng tần số này. hình 1.2 biểu diễn các giá trị đầu vào và đầu ra của bộ lượng tử. Như ta thấy, với các hệ số có giá trị nhỏ sau khi qua bộ lượng tử sẽ bị mã hóa về mức 0. Đây chính là quá trình loại bỏ những hệ số DCT ở vùng tần số cao.



**Hình 1.2 Biểu diễn lượng tử**

Hai tham số quan trọng của bộ lượng tử là giá trị ngưỡng  $th$ , và bước lượng tử  $q$ . Căn cứ vào giá trị ngưỡng, người ta chia bộ lượng tử ra làm hai loại: *bộ lượng tử đồng nhất* (Uniform quantiser – UTQ) và *bộ lượng tử đồng nhất có dead zone* (UTQ-DZ). Các hệ số DC của chế độ mã hóa nội ảnh được lượng tử bởi UTQ trong khi các hệ số AC và hệ số DC của chế độ mã hóa liên ảnh được lượng tử bởi UTQ-DZ. Lý do là UTQ-DZ làm cho nhiều hệ số AC trở thành giá trị 0 nên hệ số mã hóa sẽ cao hơn. hình 1.3 biểu diễn giá trị đầu vào và đầu ra của hai bộ lượng tử.





**Hình 1.3 Biểu diễn 2 bộ lượng tử UTQ (a) và UTQ-DZ(b)**

Trong bộ lượng tử UTQ, các hệ số  $F(u,v)$  được lượng tử bởi công thức:

$$I(u, v) = \left\lfloor \frac{F(u, v) \pm q}{2q} \right\rfloor \quad (1.4)$$

Giá trị  $I(u,v)$  gọi là các chỉ số lượng tử (*quantization index*). Tại phía giải mã, các hệ số được giải lượng tử theo công thức:

$$F^q(u, v) = I(u, v) \times 2q \quad (1.5)$$

Trong bộ lượng tử UTQ-DZ, các chỉ số lượng tử được tính bởi công thức:

$$I(u, v) = \left\lfloor \frac{F(u, v)}{2q} \right\rfloor \quad (1.6)$$

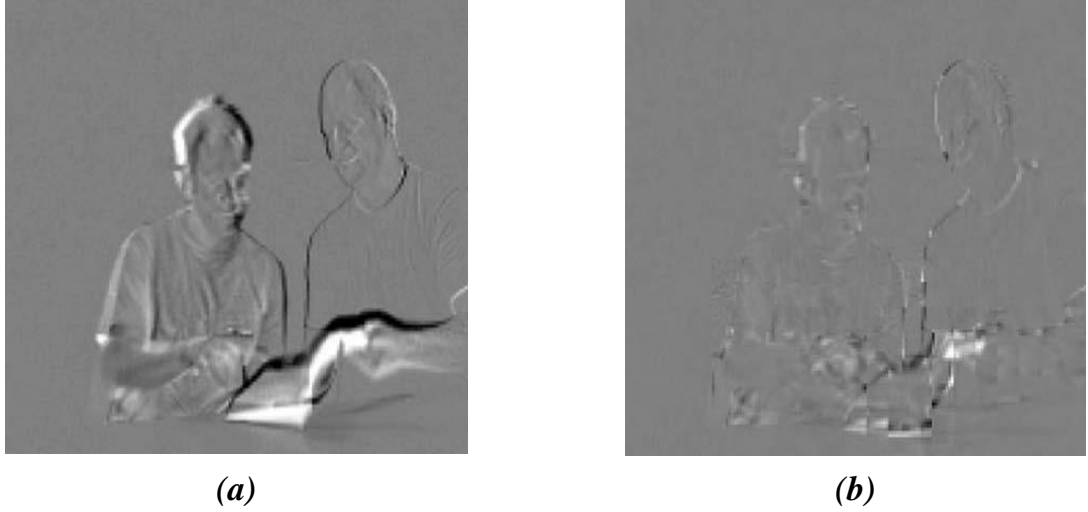
Giá trị giải lượng tử phía giải mã hóa được tính bởi công thức:

$$F^q(u, v) = \{2I(u, v) \pm 1\} \times q \quad (1.7)$$

### 1.2.3 Kỹ thuật giảm dư thừa thông tin trong miền thời gian.

Kỹ thuật giảm dư thừa thông tin trong miền thời gian được thực hiện dựa trên việc tìm ra sự khác nhau giữa các khung hình liên tiếp. Đây chính là thuật toán mã hóa liên ảnh. Đối với các đối tượng tĩnh trong ảnh, sự khác biệt gần như bằng 0. Do vậy những đối tượng này không cần nhiều thông tin để mã hóa. Ngược lại, đối với các đối tượng chuyển động nhiều, sự khác biệt giữa các khung hình là rất lớn. Điều này đồng nghĩa với việc chúng ta cần nhiều thông tin để mã hóa. Để làm giảm lượng thông tin này, người ta tiến hành thêm một bước trung gian gọi là *ước lượng chuyển động* (*motion estimation*) cho các đối tượng trong hình. Quá trình ước lượng chuyển động sẽ cho kết quả là các vector chuyển động. Dựa vào các vector này và khung hình trước đó, khung hình hiện tại sẽ được dự đoán. Quá trình này được gọi là “*bù chuyển động*” (*motion compensated*). Như vậy, sự khác biệt giữa khung hình hiện tại và khung hình dự đoán sẽ được giảm đi so với sự khác biệt giữa khung hình hiện tại và khung hình trước đó.

Hình 1.4 biểu diễn sự khác biệt của khung hình hiện thời với khung hình trước đó (a) và với khung hình sau khi được bù chuyển động (b).



**Hình 1.4 (a) Sự khác biệt giữa khung hình hiện thời và trước đó; (b) Ảnh sau khi được bù chuyển động**

**a) Ước lượng chuyển động**

Trong các tiêu chuẩn mã hóa video, thuật toán *BMA* (*Block Matching Algorithm*) thường được sử dụng để ước lượng chuyển động. Trong thuật toán này, một khung hình được chia thành các khối có kích thước  $N \times N$  pixel. Mỗi khối pixel này sẽ được di chuyển quanh vị trí ban đầu một khoảng  $w$  pixel để tìm ra vị trí của khối trong khoảng  $(N+2w) \times (N+2w)$  có cùng tọa độ nhưng ở khung hình trước đó giống với nó nhất. Khoảng cách từ tâm hình vuông  $(N+2w) \times (N+2w)$  tới tọa độ khối tìm được chính là khoảng chuyển động của khối pixel  $N \times N$ .

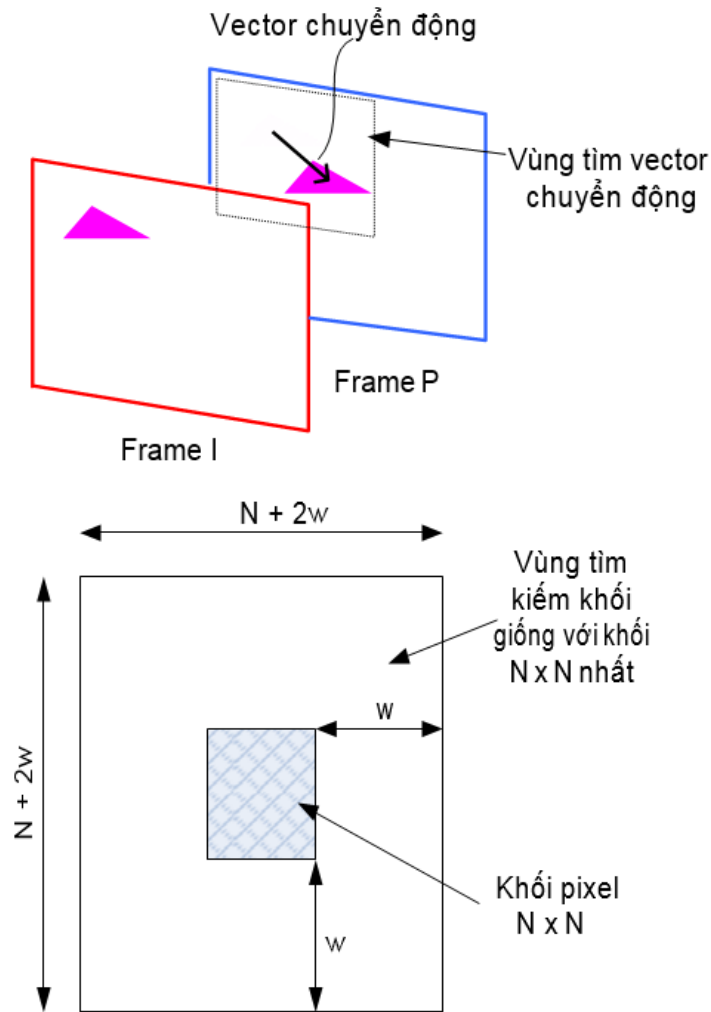
Để tìm và đánh giá mức độ giống nhau giữa hai khối pixel, chúng ta sử dụng một số phương pháp như sử dụng hàm tương quan chéo (*Crosscorrelation function - CCF*), hàm trung bình bình phương lỗi (*Mean Square Error - MSE*) và hàm trung bình tuyệt đối của lỗi (*Mean absolute error - MAE*). Khối pixel giống với khối ban đầu sẽ có CCF lớn nhất hoặc có MSE và MAE nhỏ nhất. Trong thực tế, các chuẩn mã hóa video thường sử dụng phương pháp MSE hoặc MAE:

Phương pháp MSE:

$$M(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N (f(m, n) - g(m + i, n + j))^2, \quad -w \leq i, j \leq w \quad (1.8)$$

Phương pháp MAE:

$$M(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N |f(m, n) - g(m + i, n + j)|, \quad -w \leq i, j \leq w \quad (1.9)$$



**Hình 1.5 Vùng tìm vector chuyển động của macro block hiện thời**

Trong đó  $f(m,n)$  là giá trị của pixel có tọa độ  $(m,n)$  trong khối pixel  $N \times N$  của frame hiện tại,  $g(m+i,n+j)$  là giá trị của pixel có tọa độ  $(m+i,n+j)$  trong khối pixel  $N \times N$  của frame trước đó.

Để xác định được khối pixel giống nhất, chúng ta cần thực hiện  $(2w+1)^2$  phép so sánh. Do vậy, phương pháp MAE thường hay được sử dụng hơn MSE để giảm sự phức tạp trong tính toán.

#### **b) Bù chuyển động**

Sau quá trình ước lượng chuyển động ta có được vector chuyển động của các đối tượng trong khối pixel  $N \times N$ . Dựa vào các vector chuyển động, các khối pixel  $N \times N$  trong khung hình trước đó được dịch chuyển theo hướng và độ lớn của vector chuyển động đã chỉ ra. Quá trình này được gọi là bù chuyển động. Kết quả của quá trình này là một khung hình mới được cho là giống với khung hình hiện thời. Tuy nhiên, đây vẫn chỉ là quá trình dự đoán nên khung hình dự đoán sẽ không thể giống hoàn toàn với khung hình hiện thời.

#### **c) Mã có chiều dài thay đổi**

Trong trường hợp chúng ta muốn giảm số bit mã hóa tại đầu ra của bộ mã hóa, các hệ số DCT và vector chuyển động sẽ được mã hóa bằng mã có chiều dài thay đổi (*Variable Length Coding – VLC*). Nguyên lý của việc mã hóa này là các từ mã ngắn sẽ được gán cho các thông tin có xác suất xuất hiện lớn, các từ mã dài được gán cho các thông tin có xác suất xuất hiện bé. Như vậy, độ dài của các từ mã sẽ biến đổi tỷ lệ nghịch với xác suất xuất hiện của ký hiệu cần được mã hóa. Theo định luật Shannon, số bit tối thiểu cần thiết để mã hóa một ký hiệu có xác suất xuất hiện  $p$  là  $\log_2 p$ . Như vậy, số lượng bit trung bình tối thiểu cần để mã hóa  $I$  trong  $n$  ký hiệu của nguồn thông tin  $S$  được tính như sau:

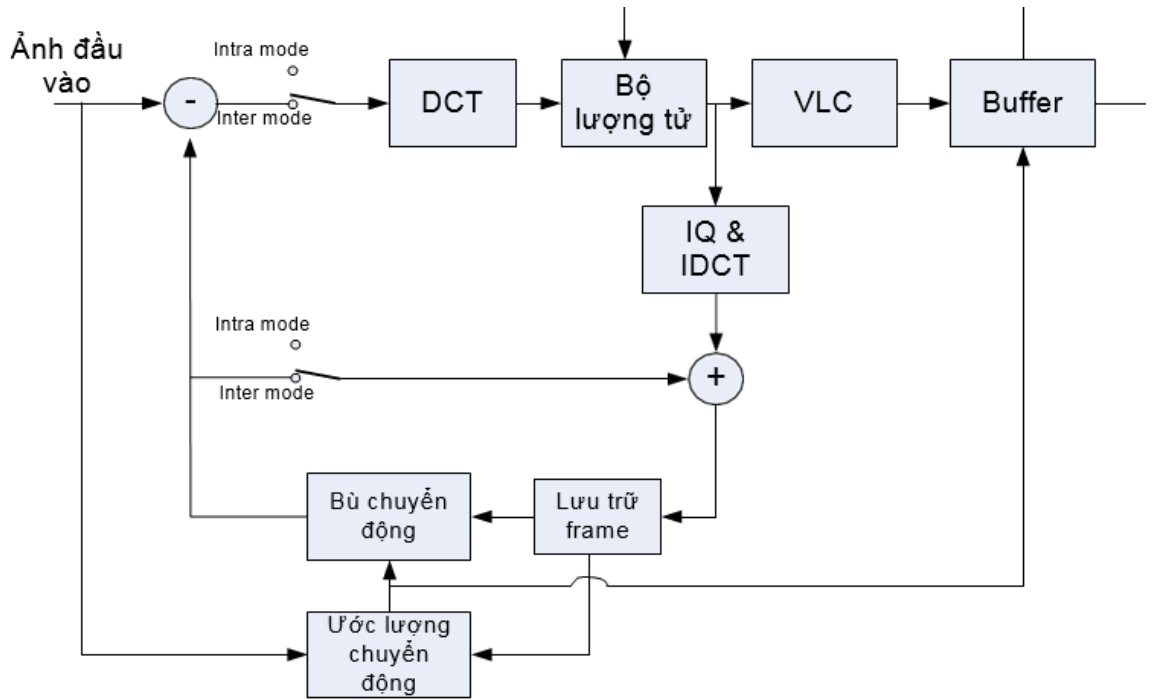
$$H(x) = - \sum_{i=1}^n p_i \log_2 p_i \text{ (bit)} \quad (1.10)$$

$H(x)$  được gọi là *entropy* của nguồn tin  $S$ . Đây cũng chính là *giới hạn (Shannon limit)* người ta mong muốn đạt được khi xây dựng bộ mã cho các nguồn tin. Khi giới hạn này đạt được đồng nghĩa với việc quá trình mã hóa hoặc quá trình mã hóa là tối ưu.

Trong chuẩn mã hóa video có hai loại mã VLC thường được sử dụng là mã *Huffman* và mã *số học (Arithmetic)*. Mã Huffman thường được sử dụng nhiều hơn nhưng phương pháp này được cho là không tối ưu vì mã bị ràng buộc là phải gán cho mỗi ký hiệu của nguồn một số nguyên các bit. Ví dụ, với ký hiệu có xác suất xuất hiện là 0.9, theo giới hạn Shannon ta chỉ cần 0.15 bit để mã hóa ký hiệu này. Tuy nhiên mã Huffman bắt buộc phải sử dụng 1 bit để mã hóa. Điều này dẫn đến việc lãng phí tài nguyên (ví dụ băng thông). Khác với mã Huffman, mã số học có thể đạt được giới hạn Shannon vì các ký hiệu không cần phải mã hóa riêng biệt. Mã Huffman thường được sử dụng để mã hóa các hệ số DCT và vector chuyển động. Trong trường hợp cần mã hóa các thông tin khác thì mã số học được sử dụng.

#### **1.2.4 Sơ đồ tổng quát của mã hóa video.**

Hình 1.6 mô tả sơ đồ tổng quát của bộ mã hóa video được sử dụng trong các chuẩn mã hóa như *H.261*, *H.264*, *MPEG-1*, *MPEG-2* và *H.264/MPEG-4 part 10*.



**Hình 1.6 Sơ đồ nguyên lý tổng quát của bộ mã hóa video**

**a) Intraframe/Interframe loop**

Trong chế độ mã hóa liên ảnh, các giá trị khác biệt giữa các pixel của khung hình hiện tại và khung hình dự đoán dựa vào khung hình trước đó được mã hóa và truyền đi. Tại phía thu, sau khi giải mã, các giá trị khác biệt này sẽ được cộng với khung hình dự đoán do bên thu đưa ra để xây dựng lên khung hình hiện tại. Như vậy quá trình dự đoán đóng vai trò rất quan trọng vì nếu dự đoán càng chính xác thì giá trị khác biệt sẽ càng nhỏ và do đó cần ít thông tin để truyền đi. Đối với trường hợp chuỗi ảnh tĩnh, giá trị dự đoán của các pixel trong khung hình hiện tại sẽ là các pixel cùng tọa độ trong khung hình trước đó.

Trong chế độ mã hóa nội ảnh, giá trị dự đoán được dựa trên các khối hoặc các pixel liền kề trước đó trong cùng một khung hình. Chế độ này được áp dụng cho khung hình đầu tiên của một nhóm các khung hình hoặc áp dụng trong việc mã hóa ảnh tĩnh JPEG sẽ được đề cập chi tiết trong phần sau.

**b) Ước lượng chuyển động**

Lượng dữ liệu truyền đi sẽ rất lớn nếu với mỗi pixel đều có một vector chuyển động. Thay vào đó, người ta sẽ gán cho mỗi nhóm pixel (block) một vector chuyển động. Trong các chuẩn mã hóa, mỗi nhóm pixel là một hình vuông có kích thước 16x16 (được gọi là một *Macroblock* - MB) được ước lượng chuyển động và được bù chuyển động. Thông thường, việc ước lượng chuyển động được thực hiện trên kênh chói (kênh Y) của các khung hình.

**c) Inter/Intra switch**

Inter/Intra switch có tác dụng chuyển đổi giữa hai chế độ liên ảnh và nội ảnh. Trong chuẩn mã hóa ảnh JPEG, tất cả các MB trong ảnh được mã hóa ở chế độ nội ảnh. Với các chuẩn mã hóa video như H.26x và MPEG, một số loại frame được mã hóa ở chế độ nội ảnh, một số loại frame được mã hóa ở chế độ liên ảnh. Ngoài ra, trong cùng một frame liên ảnh, một số MB được mã hóa ở chế độ nội ảnh để tối ưu hóa tỷ lệ mã hóa.

#### ***d) DCT***

Trên kênh Y và hai kênh màu U, V, mỗi MB được chia thành các khối nhỏ hơn có kích thước  $8 \times 8$ . Như vậy, sẽ có 4 khối  $8 \times 8$  thuộc kênh Y và một số khối thuộc kênh màu tùy theo độ phân giải của ảnh. Mỗi khối  $8 \times 8$  này sẽ được biến đổi DCT để có được một ma trận  $8 \times 8$  các hệ số của biến đổi DCT.

#### ***d) Lượng tử hóa***

Có hai loại lượng tử hóa. Một loại có dead zone được sử dụng cho các hệ số AC và hệ số DC của các MB mã hóa liên ảnh. Loại còn lại không có dead zone được sử dụng cho các hệ số DC của MB nội ảnh. Khoảng giá trị của các hệ số được lượng tử có thể từ -2047 đến +2047. Với bộ lượng tử có dead zone, nếu trị tuyệt đối của các hệ số nhỏ hơn bước lượng tử  $q$  thì sẽ được gán bằng 0. Ngược lại, giá trị của các hệ số sau lượng tử sẽ trong khoảng từ 1 đến 31.

#### ***e) Variable Length Coding***

Các hệ số sau khi được lượng tử sẽ được mã hóa bằng mã có độ dài thay đổi. Ngoài ra, giá trị của vector chuyển động cũng được mã hóa bằng mã này cùng với các hệ số lượng tử.

#### ***f) Giải lượng tử và biến đổi DCT ngược***

Để tái tạo khung hình hiện thời, các hệ số DCT sau lượng tử được giải lượng tử và biến đổi DCT ngược. Sau đó, các giá trị này được cộng với khung hình trước đó đang được bộ mã hóa lưu giữ để tái tạo lại khung hình hiện thời. Khung hình hiện thời này sẽ lại được lưu giữ để dùng cho quá trình dự đoán khung hình tiếp theo.

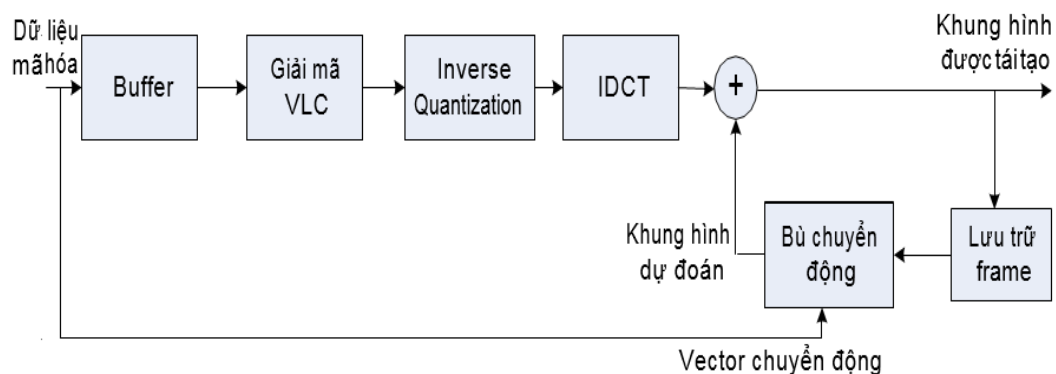
#### ***g) Bộ đệm***

Tốc độ bit được tạo ra bởi bộ mã hóa sẽ liên tục thay đổi vì tốc độ bit phụ thuộc vào mức độ chuyển động của các đối tượng trong video. Vì vậy, trước khi truyền tín hiệu video trên các kênh truyền có băng thông cố định thì các bit được lưu trong bộ đệm để điều tiết việc truyền đi. Bộ đệm trong trường hợp này chính là bộ nhớ có hai cổng ghi và đọc. Cổng ghi có nhiệm vụ nhận dữ liệu là các bit sau khi được mã hóa. Cổng đọc có nhiệm vụ đọc ra dữ liệu với một tốc độ ổn định. Mỗi khi dữ liệu được đọc thì bộ đệm sẽ giải phóng dữ liệu đó trong bộ đệm. Trong trường hợp có dữ liệu ghi vào nhiều hơn dữ liệu đọc ra (với những video có nhiều chuyển động) thì bộ đệm gửi thông tin phản hồi tới bộ lượng tử để tăng mức lượng tử. Khi đó dữ liệu ra của bộ mã

hóa sẽ ít đi vì các hệ số DCT bị giảm đi. Ngược lại, với những video có ít chuyển động thì bộ lượng tử sẽ giảm mức lượng tử để cải thiện chất lượng video. Quá trình này được gọi là ổn định tốc độ bit (*Constant Bit Rate*). Với những bộ mã hóa có tốc độ bit thay đổi (*Variable Bit Rate – VBR*) thì bộ đệm không được sử dụng và mức lượng tử được giữ nguyên trong suốt quá trình mã hóa.

### 1.3 Giải mã hóa video.

Về cơ bản, quá trình giải mã hóa bao gồm các bước giống như quá trình mã hóa nhưng thứ tự ngược lại. hình 1.7 mô tả quá trình giải mã hóa tín hiệu video. Ban đầu dữ liệu nhận được là các tín hiệu mã hóa được tách thành hai phần: dữ liệu mã hóa giá trị khác biệt và dữ liệu mã hóa giá trị vector chuyển động. Vector chuyển động sẽ được đưa vào bộ dự đoán để dự đoán khung hình hiện tại. Dữ liệu mã hóa giá trị khác biệt giữa khung hình hiện tại và khung hình trước đó lần lượt được giải mã entropy, giải lượng tử, biến đổi DCT ngược. Sau khi biến đổi DCT ngược ta sẽ được khung hình là hiệu số của khung hình hiện tại và khung hình dự đoán. Sau khi được cộng với khung hình dự đoán, ta sẽ thu được khung hình tái tạo của khung hình hiện tại.



**Hình 1.7 Sơ đồ giải mã hóa tín hiệu video**

### 1.4 Kết luận chương.

Chương đầu luận văn đã tìm hiểu nguyên tắc chung của các bộ mã hóa video và các nguyên tắc được sử dụng trong các chuẩn mã hóa video như H.261, H.263, MPEG -1, 2 và 4

Sau khi mã hóa video, các dòng bit hoặc sẽ được đóng gói vào các gói tin để truyền qua mạng hoặc được đóng gói vào các file để lưu trữ. Quá trình giải mã hóa bao gồm các bước giống như quá trình mã hóa nhưng thứ tự ngược lại

Công nghệ không ngừng phát triển. Điều này cũng không ngoại lệ khi nói đến quá trình mã hóa và phát video. Các kỹ sư công nghệ luôn cố gắng không ngừng nghỉ nhằm cải thiện cách thức thu, mã hóa và phát hình ảnh nhằm tối đa hiệu quả lưu trữ và thời gian truyền tải. Vào năm 2013, Liên hiệp truyền thông quốc tế ITU đã giới thiệu chuẩn mã hóa mang tên H.265 hay còn gọi High Efficiency Video Coding – HEVC.

Cũng giống như trước đó, H.265 là thành quả của sự kết hợp giữa ITU-T VCEG và ISO/IEC MPEG.

Chuẩn H.265 hứa hẹn là chuẩn mã hóa thay thế xứng đáng cho chuẩn H.264 vốn đang rất phổ biến cho đến thời điểm hiện tại. Với chuẩn H.265 này khi xem video online sẽ giảm được  $\frac{1}{2}$  băng thông tải và không phải chi trả quá nhiều tiền cho việc xem video với kết nối 3G/4G mà vẫn thưởng thức được video chất lượng cao, thời gian tải video giảm đáng kể cũng giúp chúng ta tiết kiệm được rất nhiều thứ. Nội chương tiếp theo của đề tài sẽ giới thiệu đầy đủ hơn về chuẩn mã hóa tiên tiến này, đồng thời trình bày thuật toán mã hóa video với cấu trúc nhóm ảnh tùy biến theo nội dung của video có thể ứng dụng và tối ưu hiệu quả của chuẩn mã hóa H.265.



## **CHƯƠNG 2. TỔNG QUAN VỀ MÃ HÓA VIDEO, GIẢI MÃ VIDEO ĐỘ PHÂN GIẢI CAO H.265/HEVC**

### **2.1 Tổng quan về mã hóa video độ phân giải cao.**

#### **2.1.1 Giới thiệu chuẩn nén H.265.**

Chuẩn nén H.265 là một chuẩn mã hóa/giải mã video và định dạng video tốt nhất hiện nay. Chuẩn nén H.265 có khả năng nén gấp đôi so với chuẩn trước (chuẩn H.264) đó mà chất lượng không đổi. Chuẩn này có thể hỗ trợ độ phân giải lên đến  $8192 \times 4320$ , bao gồm cả chất lượng 8k Ultra HD.

Các ưu điểm của chuẩn nén H.265:

- Được trang bị tính năng hỗ trợ xử lý song song hiệu quả.
- Phân vùng linh hoạt hơn, với kích cỡ các vùng từ lớn đến nhỏ tùy theo nội dung của khung hình.
- Chế độ dò tìm/dự báo và phát tín hiệu, chuyển động của các vector phức tạp hơn.
- Bộ lọc nội suy và bộ lọc khử nhiễu cho các khối ảnh tinh vi hơn.
- Linh hoạt hơn trong các chế độ dự báo và chuyển đổi kích thước khối.

#### **2.1.2 Cấu trúc mã hoá.**

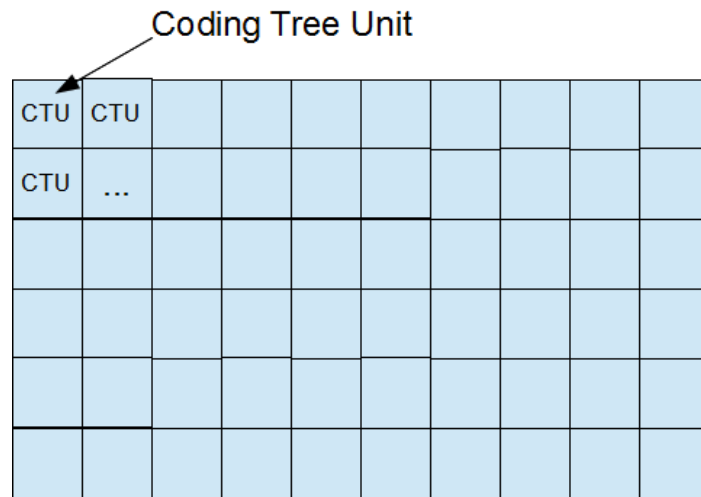
Lớp mã hóa video của HEVC thực hiện cách tiếp cận lại (dự đoán trong ảnh/liên ảnh và mã hóa biến đổi 2D) như được sử dụng trong tất cả các tiêu chuẩn nén video kể từ H.261. Sơ đồ khối của bộ mã hóa video HEVC được thể hiện như ở Hình 2.3.

Mỗi ảnh đầu vào được chia thành các khối ảnh, sau đó được mã hóa và được truyền tải đến các bộ giải mã. Ảnh đầu tiên của một chuỗi video được mã hóa chỉ sử dụng dự đoán trong ảnh. Đối với các ảnh còn lại của chuỗi sử dụng các chế độ mã hóa dự đoán liên ảnh theo thời gian. Quá trình mã hóa cho dự đoán liên ảnh sẽ lựa chọn dữ liệu chuyển động bao gồm các ảnh tham chiếu và vector chuyển động (MV) dùng để dự đoán các mẫu của mỗi khối ảnh. Các bộ mã hóa và giải mã tạo ra tín hiệu dự đoán liên ảnh giống nhau bằng cách sử dụng kỹ thuật bù chuyển động (MC) và thông tin phụ (side information) là dữ liệu quyết định chế độ.

Các tín hiệu dư thừa của dự đoán trong ảnh và liên ảnh sẽ được biến đổi bởi phép biến đổi không gian tuyến tính. Sau đó các hệ số biến đổi được định cỡ, lượng tử hóa, mã hóa Entropy, và được truyền cùng với các thông tin dự đoán.

Đồng thời, bộ mã hóa cũng sao chép lại mạch vòng xử lý giải mã sao cho cả hai phía mã hóa và giải mã cùng tạo ra các dự đoán giống nhau đối với ảnh kế tiếp. Do đó, các hệ số biến đổi lượng tử hóa sẽ được tạo lại qua định cỡ ngược và sau đó là biến đổi ngược để sao lại gần đúng tín hiệu dư thừa. Sau đó các tín hiệu dư thừa này cộng với





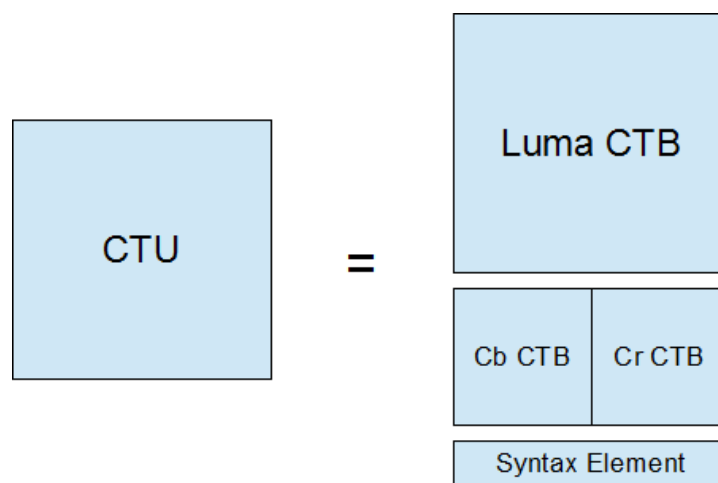
**Hình 2.2 Cấu trúc CTU**

Vậy làm cách nào chúng ta có thể hỗ trợ nhiều loại kích thước theo khối như vậy một cách hiệu quả. Đó là một thách thức mà HEVC đã cố gắng giải quyết bằng cách định nghĩa ra các đơn vị cấu hình mới, trong đó có CTU (*Coding Tree Unit – đơn vị cây mã hóa*) là đơn vị ở cấp cao nhất. Giả sử chúng ta có một hình ảnh để mã hóa, HEVC sẽ chia hình ảnh thành các CTU như Hình 2.4.

Chiều rộng và chiều cao của CTU được cấu hình trong bộ tham số, nghĩa là tất cả các CTU trong chuỗi video sẽ có cùng kích thước: 64x64, 32x32, 16x16 và các kích thước này được hiểu là kích thước logic, không chứa đơn vị đo lường cụ thể.

CTU là một đơn vị logic và nó thường bao gồm ba khối: một mẫu chói (Y), hai mẫu màu (Cb và Cr) và các phần tử cú pháp liên quan. Mỗi khối được gọi là CTB (*Coding Tree Block – Khối cây mã hóa*).

Mỗi CTB cũng có cùng kích thước với CTU là: 64x64, 32x32, 16x16, trong đó các kích thước lớn hơn sẽ cho phép nén tốt hơn.

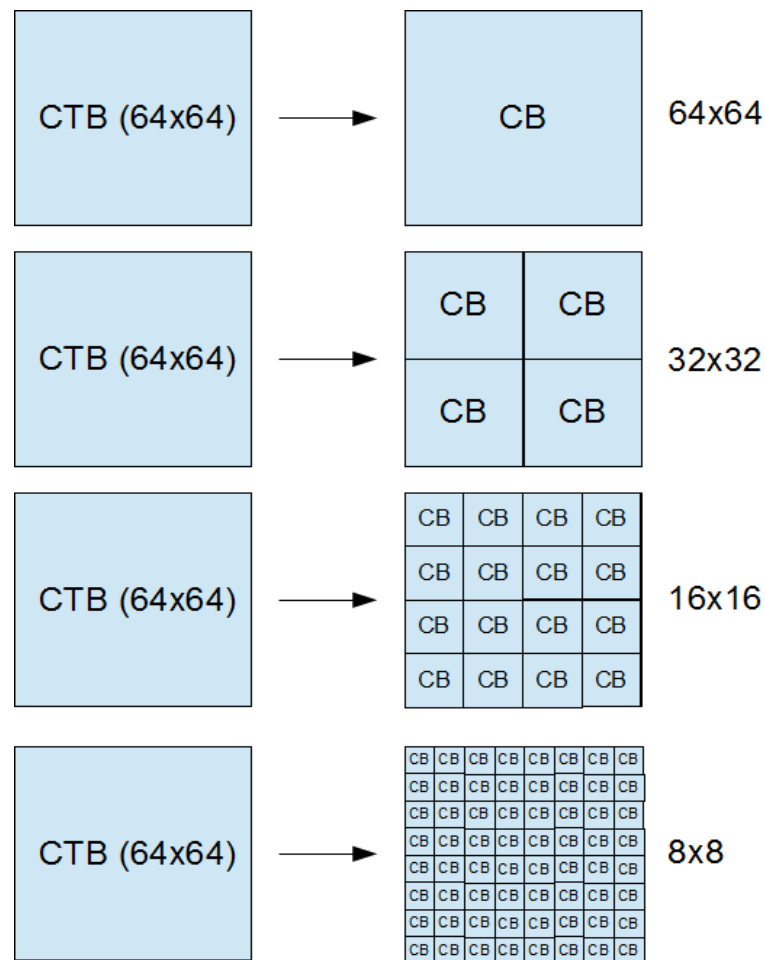


**Hình 2.3 Cấu trúc CTB**

**b) Đơn vị mã hóa (CU) và khối mã hóa (CB)**

Tuy nhiên, tùy thuộc vào một phần của khung hình video, CTB có thể quá lớn để quyết định xem có nên thực hiện dự đoán liên ảnh hay dự đoán nội ảnh hay không. Do đó, mỗi CTB có thể được chia thành nhiều CB (*Coding Block – Khối mã hóa*) khác nhau. Ví dụ: một số CTB được chia thành 16x16 CB trong khi các CTB khác được chia thành 8x8 CB. Hình 2.6 minh họa cách chia nhỏ thành cấu trúc CB của các CTB 64x64.

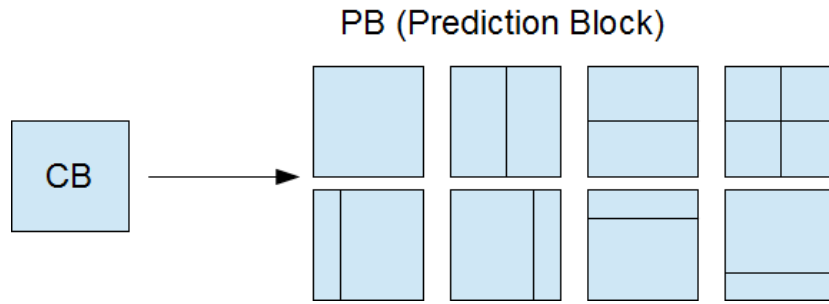
Trong đó, CB là điểm quyết định xem có thực hiện dự đoán liên ảnh hoặc nội ảnh hay không, và kiểu dự đoán sẽ được mã hóa trong CU (*Coding Unit – Đơn vị mã hóa*). Cú pháp cây tứ phân (*Quadtree Syntax*) chứa trong CTU cho phép chia các CB có kích cỡ và vị trí thích hợp dựa trên đặc tính tín hiệu của vùng tạo bởi CTB. Do đó gốc của cây tứ phân là CTU. Thông thường, kích cỡ của CB được xác định là:  $8 \times 8 < \text{kích cỡ CB size} < \text{kích cỡ CTB}$ . Khi đó, đơn vị mã hóa CU sẽ gồm 1 CB chói và 2 CB màu, cùng cú pháp đi kèm. Một CTB có thể chứa chỉ một CU hoặc được phân chia thành nhiều CU, và mỗi CU có một phân vùng liên quan đến các đơn vị dự báo (PUs) và một cây các đơn vị biến đổi (TUs).



**Hình 2.4 Cấu trúc CTB chia nhỏ**

**c) Đơn vị dự đoán (PU) và khối dự đoán (PB)**

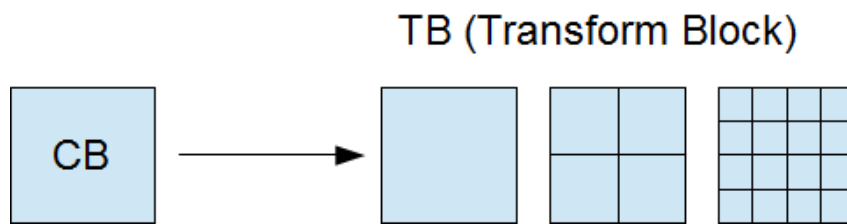
Đơn vị CB là đủ tốt cho việc đưa ra quyết định sử dụng phương pháp dự đoán liên ảnh hay nội ảnh, nhưng nó vẫn có thể quá lớn để lưu trữ các vector chuyển động. Ví dụ, một đối tượng rất nhỏ như tuyết rơi có thể di chuyển ở giữa CB kích thước 8x8 sẽ không thể được xử lý chính xác. Vì vậy, HEVC cần một đơn vị với cấu trúc nhỏ hơn gọi là PB như hình 2.7. Mỗi CB có thể được chia thành các PB (*Prediction Block – Khối dự đoán*) khác nhau tùy thuộc vào phương pháp dự đoán theo thời gian hay không gian. HEVC hỗ trợ các kích cỡ PB khác nhau từ 4x4 cho đến 64x64.



**Hình 2.5 Cấu trúc PB**

**d) Đơn vị biến đổi (TU) và khối biến đổi (TB)**

Một khi dự đoán được thực hiện, chúng ta cần phải mã hóa phần dư (là sự khác biệt giữa hình ảnh dự đoán và hình ảnh thực tế) với phép biến đổi DCT. Trong trường hợp này, cấu trúc CB vẫn có thể là quá lớn cho việc thực hiện thủ tục này bởi một CB có thể chứa cả một phần ảnh với mức độ chi tiết cao (tín hiệu cao tần) và một phần ảnh trơn phẳng (tín hiệu thấp tần). Do đó HEVC chia mỗi CB thành các TB (*Transform Block – Khối biến đổi*) khác nhau và các TB không cần thiết phải tương ứng với mỗi PB trong cấu trúc ảnh. TB có thể thực hiện phép biến đổi đơn trên phần dư của nhiều PB khác nhau và ngược lại.



**Hình 2.6 Cấu trúc TB**

**e) Tín hiệu hóa vector chuyển động**

Dự đoán vector chuyển động tiên tiến (AMVP) được sử dụng, bao gồm cả dẫn xuất của nhiều ứng viên có khả năng nhất dựa trên dữ liệu từ các PB liền kề và hình ảnh tham chiếu. Một chế độ kết hợp cho mã hóa MV cũng có thể được sử dụng, cho phép thừa kế của các MV từ các PBs lân cận theo thời gian hoặc không gian. Hơn nữa, so với H.264 / MPEG-4 AVC, suy luận bỏ qua được cải thiện và suy luận chuyển động trực tiếp cũng được quy định.

**f) Bù chuyển động**

Độ chính xác  $\frac{1}{4}$ -mẫu được sử dụng cho các MV, và các bộ lọc 7-tap hoặc 8-tap được sử dụng cho phép nội suy của các vị trí mẫu-phân đoạn (được so sánh với việc lọc 6-tap của vị trí  $\frac{1}{2}$ -mẫu theo sau bởi phép nội suy tuyến tính cho vị trí  $\frac{1}{4}$ -mẫu trong H.264 / MPEG-4 AVC). Tương tự như H.264 / MPEG-4 AVC, nhiều ảnh tham chiếu được sử dụng. Đối với mỗi PB, một hoặc hai vectơ chuyển động có thể được truyền đi với mã hóa dự đoán đơn hoặc mã hóa dự đoán đôi tương ứng. Như trong H.264 / MPEG-4 AVC, một hoạt động định cỡ (scale) và bù đắp được áp dụng cho các tín hiệu dự đoán gọi là dự đoán có trọng số.

#### ***g) Dự đoán trong ảnh***

Các mẫu biên được giải mã của các khối liền kề là dữ liệu tham chiếu cho dự đoán theo không gian trong các vùng dự đoán liên ảnh không được thực hiện. Dự đoán trong ảnh hỗ trợ 33 chế độ có hướng (so với 8 chế độ trong H.264 / MPEG-4 AVC), cộng với chế độ dự đoán mặt phẳng và DC. Các chế độ dự đoán trong ảnh được chọn sẽ được mã hóa bằng dẫn xuất chế độ có thể xảy ra nhất (ví dụ, các hướng dự đoán) dựa trên những PB lân cận giải mã trước đây đó.

#### ***h) Điều khiển lượng tử hóa***

Như trong H.264 / MPEG-4 AVC, lượng tử hóa tái thiết đồng nhất (URQ) được dùng trong HEVC, với các ma trận định cỡ (scale) lượng tử hóa được hỗ trợ cho các kích thước khối biến đổi khác nhau.

#### ***i) Mã hóa Entropy***

Mã hóa số học nhị phân thích ứng ngữ cảnh (CABAC) được sử dụng để mã hóa Entropy. Nó tương tự như các sơ đồ CABAC trong H.264 / MPEG-4 AVC, nhưng có thêm một số cải tiến để cải thiện tốc độ lưu lượng (đặc biệt là cho các kiến trúc xử lý song song), hiệu suất nén, và để giảm bộ nhớ ngữ cảnh của nó.

#### ***j) Bộ lọc bỏ khối trong-vòng***

Một bộ lọc bỏ khối giống như bộ lọc trong H.264 / MPEG-4 AVC được dùng trong vòng lặp dự đoán liên ảnh. Tuy nhiên, thiết kế được đơn giản hóa về các quá trình đưa ra quyết định và lọc, và nó được làm thuận lợi hơn để xử lý song song.

#### ***k) Bù thích ứng mẫu (SAO)***

Một ánh xạ biên độ phi tuyến tính được đưa vào trong vòng lặp dự đoán liên ảnh đằng sau bộ lọc bỏ khối. Mục đích là để tái tạo tốt hơn các biên độ tín hiệu ban đầu bằng cách sử dụng một bảng look-up, bảng này được mô tả bằng một vài thông số bổ sung mà chúng có thể được xác định bằng cách phân tích biểu đồ ở phía bộ mã hóa.

### ***2.1.2 Các kỹ thuật mã hóa video HEVC***

#### ***a) Biểu diễn lấy mẫu của ảnh***

Để biểu diễn các tín hiệu video màu, HEVC sử dụng một không gian 3 màu cơ bản YCbCr với lấy mẫu 4:2:0. Các ảnh video lấy mẫu tăng lên với kích thước ảnh hình

chữ nhật  $W \times H$ , trong đó  $W$  là chiều rộng và  $H$  là chiều cao của ảnh theo quan điểm lấy mẫu chói, còn với lấy mẫu màu 4:2:0 sẽ là  $W/2 \times H/2$ .

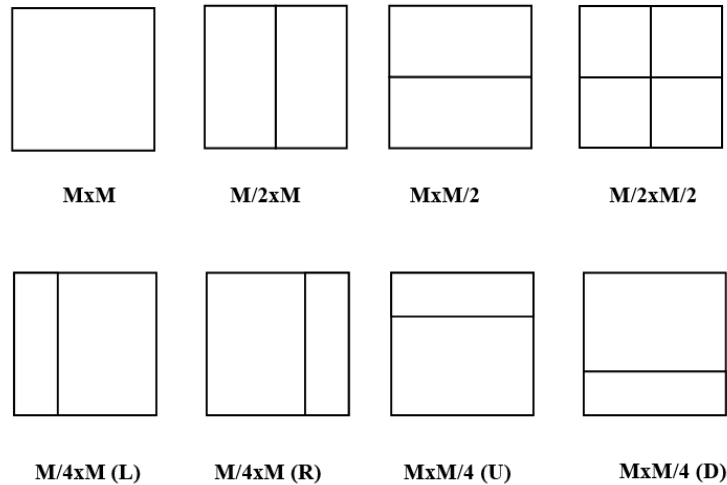
**b) Phân chia hình ảnh thành đơn vị cây mã hóa**

Một ảnh được phân chia thành các đơn vị mã hóa cây (CTU), mỗi CTU chứa các CTB chói và CTB màu. Một CTB chói lấy mẫu  $L \times L$  cho các thành phần chói và lấy mẫu  $L/2 \times L/2$  cho mỗi một trong hai thành phần màu.  $L=16,32,64$  được xác định bởi một yếu tố cú pháp mã hóa được quy định trong SPS.

**c) Phân chia CTB thành CB**

Các khối CTB chói và màu có thể được sử dụng trực tiếp như các CB hoặc có thể phân chia tiếp thành nhiều CB. Việc phân chia được thực hiện bằng cách sử dụng cấu trúc cây. Việc phân chia cấu trúc cây trong HEVC thường được áp dụng đồng thời cho cả khối chói và màu, mặc dù các ngoại lệ áp dụng khi tiến tới kích thước tối thiểu nhất định.

CTU chứa cú pháp cây tứ phân cho phép chia các CB đến một kích thước phù hợp dựa trên các đặc điểm tín hiệu của vùng mà nó được bao phủ bởi các CTB. Quá trình chia tách cây tứ phân có thể được lặp đến khi một CB chói đạt đến một kích thước tối thiểu cho phép. Kích thước này được chọn bởi bộ mã hóa sử dụng cú pháp trong SPS và luôn là  $8 \times 8$  hoặc lớn hơn.



**Hình 2.7 Chế độ chia tách một CB thành PBs.**

Đối với những dự đoán trong ảnh chỉ có  $M \times M$  và  $M/2 \times M/2$  được hỗ trợ.

**d) Khối dự đoán (PB) và đơn vị dự đoán (PU)**

Chế độ dự đoán cho CU là được báo hiệu là trong ảnh hay liên ảnh, tùy vào việc nó sử dụng dự đoán trong ảnh (theo không gian) hoặc dự đoán liên ảnh (theo thời gian).

Khi chế độ dự đoán được báo hiệu là trong ảnh, kích thước PB là kích thước khối tại nơi dự đoán trong ảnh được tạo sẵn có cùng kích thước với CB cho tất cả các kích

thước khối ngoại trừ kích thước CB nhỏ nhất. Đối với các trường hợp sau đó, một cờ xuất hiện cho biết CB được chia thành bốn phần PB mà mỗi phần đều có dự đoán trong ảnh riêng. Lý do của việc cho phép chia tách này là tạo khả năng lựa chọn dự đoán trong ảnh cho các khối nhỏ  $4 \times 4$ . Khi dự đoán trong ảnh thành phần chói hoạt động với các khối  $4 \times 4$ , dự đoán trong ảnh thành phần màu cũng sử dụng các khối  $4 \times 4$  (4 khối thành phần chói  $4 \times 4$ ). Kích thước vùng hiện tại có dự đoán trong ảnh (nó là khác so với kích thước PB có dự đoán trong ảnh được tạo sẵn) phụ thuộc vào việc phân chia mã hóa dư thừa được mô tả như sau.

Khi dự đoán là liên ảnh, nó xác định liệu các CB thành phần chói và màu được chia thành một, hai hoặc bốn PB. Việc chia thành bốn PB chỉ được phép khi kích thước CB bằng với kích thước CB cho phép tối thiểu. Khi một CB được chia thành bốn PB, mỗi PB là một góc phần tư của CB và khi một CB được chia thành hai PB, sẽ có 6 kiểu chia tách. Các khả năng phân chia cho CB dự đoán liên ảnh được mô tả trong Hình 2.9. Bốn loại phân chia cao hơn minh họa cho trường hợp chia tách CB thành kích thước  $M \times M$ , chia tách CB thành hai PB kích thước  $M \times M/2$  hoặc  $M/2 \times M$  hoặc chia tách nó thành bốn PB có kích thước  $M/2 \times M/2$ . 4 loại phân chia thấp hơn được gọi là phân chia chuyển động bất đối xứng (*AMP - asymmetric motion partitioning*), và chỉ được phép khi  $M \geq 16$ , đối với thành phần chói. Một PB của phân chia bất đối xứng có chiều cao hoặc chiều rộng là  $M/4$  và chiều rộng hoặc chiều cao là  $M$  và các PB khác lấp đầy phần còn lại của CB bởi 1 chiều cao hoặc chiều rộng của  $3M/4$  và chiều rộng hoặc chiều cao  $M$ . Mỗi PB dự đoán liên ảnh được gán một hoặc hai vectơ chuyển động và các chỉ số ảnh tham chiếu. Để tối thiểu băng thông bộ nhớ trường hợp xấu nhất, các PB của thành phần chói  $4 \times 4$  là không được phép cho dự đoán liên ảnh, và các PB của thành phần chói  $4 \times 8$  và  $8 \times 4$  được giới hạn cho mã hóa dự đoán đơn. Các PB thành phần chói và màu kết hợp với cú pháp dự đoán tạo thành PU.

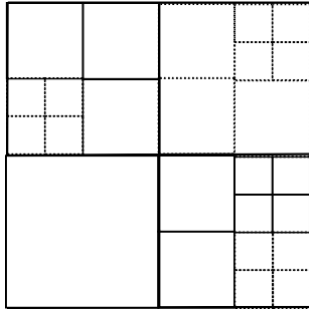
***e) Phân chia cây cấu trúc thành các đơn vị biến đổi và các khối biến đổi***

Đối với mã hóa dư thừa, một CB có thể được phân vùng 1 cách đệ quy thành các khối biến đổi (TB). Việc phân chia được báo hiệu bằng một cây tứ phân dư thừa.

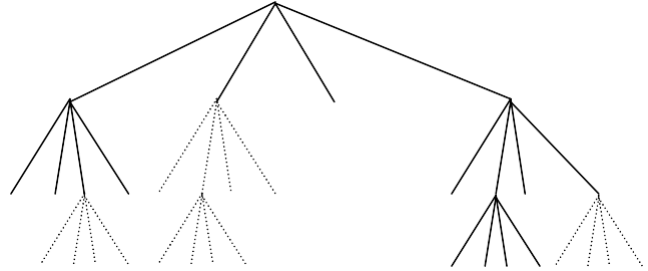
Phân chia CB và TB, nơi mà một khối có thể được chia 1 cách đệ quy thành các góc phần tư, như minh họa trong Hình 2.10. Đối với một CB thành phần chói  $M \times M$ , một cờ báo hiệu có được chia thành bốn khối có kích thước  $M/2 \times M/2$  không. Độ sâu tối đa của cây tứ phân dư thừa trong SPS sẽ báo hiệu cho việc có chia tách thêm nữa không. Mỗi góc phần tư có một cờ chỉ định việc nó có được chia thành bốn góc phần tư không. Các khối nút lá từ cây tứ phân dư thừa là các khối biến đổi được xử lý tiếp bằng mã hóa biến đổi. Các bộ mã hóa chỉ ra kích thước TB thành phần chói tối thiểu và tối đa mà nó sẽ sử dụng. Việc chia tách là ngầm mặc định khi kích thước CB lớn hơn so với kích thước TB tối đa. Và việc không chia tách là ngầm mặc định khi chia



tách sẽ cho kết quả là kích thước TB thành phần chọi nhỏ hơn kích thước tối thiểu. Kích thước TB thành phần màu bằng một nửa kích thước TB thành phần chọi ở mỗi chiều, ngoại trừ trường hợp kích thước TB chọi là 4x4, trong đó trường hợp đó, một TB màu 4x4 duy nhất được bao phủ bởi bốn TB chọi 4x4. Trong trường hợp các CU dự đoán trong ảnh, các mẫu được giải mã của các TB lân cận gần nhất (trong hoặc ngoài CB) được sử dụng như là dữ liệu tham chiếu cho dự đoán trong ảnh.



(a) Phân chia CTB.



(b) Cây tứ phân tương ứng

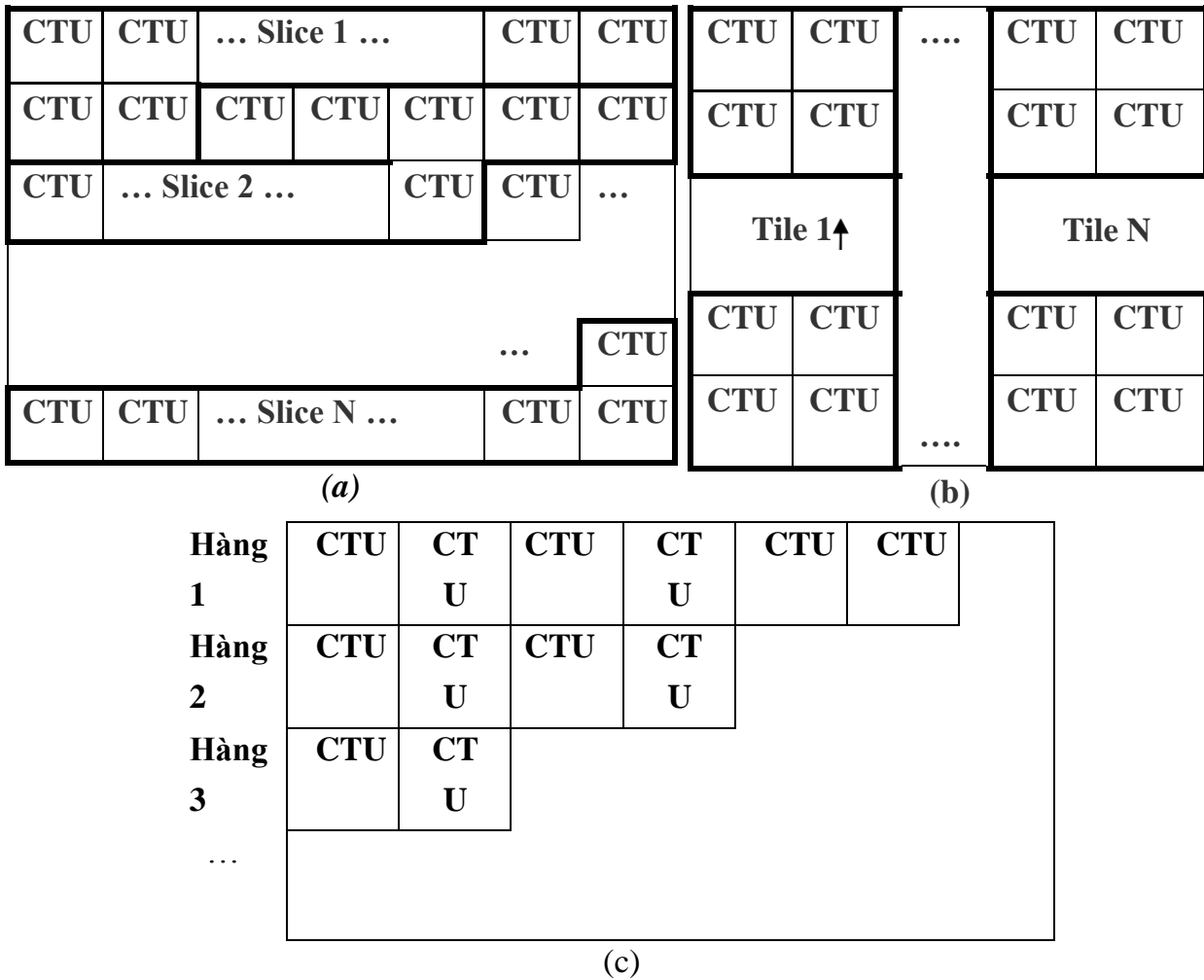
**Hình 2.8 Chia nhỏ một CTB thành các CB, TB.**

Đường liền chỉ ra các biên CB và đường chấm chỉ ra các biên TB.

#### f) Mảng (slice) và Tile

Mảng là một chuỗi các CTU được xử lý theo thứ tự của quét mảnh. Một ảnh được chia thành một hoặc một vài mảng như thể hiện trong Hình 2.11(a) vì thế một ảnh là một tập hợp của một hoặc nhiều mảng. Mỗi mảng có thể được mã hóa sử dụng các loại mã hóa khác nhau như sau.

- *Mảng I*: tất cả CU của mảng được mã hóa chỉ sử dụng dự đoán trong ảnh.
- *Mảng P*: Ngoài các loại mã hóa của mảng I, một số CU của mảng P cũng có thể được mã hóa bằng dự đoán liên ảnh với nhiều nhất một tín hiệu dự đoán bù chuyển động cho mỗi PB (dự đoán đơn). Mảng P chỉ sử dụng danh sách 0 ảnh tham chiếu.
- *Mảng B*: Bên cạnh các loại mã hóa có sẵn trong một mảng P, một số CU của mảng B cũng có thể được mã hóa bằng dự đoán liên ảnh với nhiều nhất hai tín hiệu dự đoán bù chuyển động cho mỗi PB (dự đoán đôi). Mảng B sử dụng cả hai danh sách 0 và 1 ảnh tham chiếu.



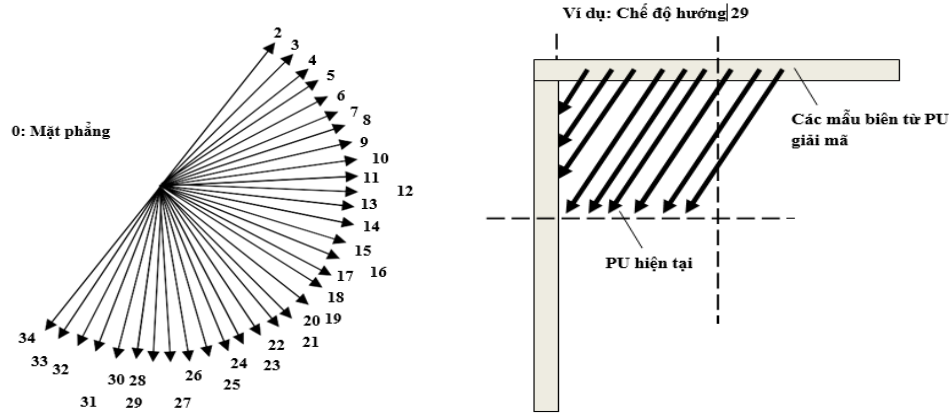
**Hình 2.9 Chia nhỏ của một ảnh thành mảng (a) và tile (b)**

**Minh họa quá trình xử lý song song wavefront (c)**

Mục đích chính của các mảng là đồng bộ lại sau khi dữ liệu mất. Hơn nữa, các mảng thường được hạn chế để sử dụng số bit tối đa. Vì vậy, các mảng thường có thể chứa số CTU thay đổi mức độ cao/mảng.

Tile là các vùng hình chữ nhật của ảnh có khả năng giải mã độc lập. Mục đích chính của tile là cho phép sử dụng các kiến trúc xử lý song song đối với mã hóa và giải mã. Nhiều tile có thể chia sẻ thông tin tiêu đề được chứa trong cùng mảng. Nói cách khác, một tile đơn lẻ có thể chứa nhiều mảng. Tile gồm một nhóm các CTU xếp hình chữ nhật như Hình 2.11(b).

Để hỗ trợ việc đóng gói dữ liệu, các mảng phụ thuộc được bổ sung. Cuối cùng, với WPP, một mảng được chia thành các hàng CTU. Việc giải mã của mỗi hàng có thể được bắt đầu ngay sau một vài quyết định dự báo và thích ứng của các bộ mã hóa Entropy được tạo ra ở hàng trước. Điều này hỗ trợ xử lý song song các hàng của các CTU bằng cách sử dụng một số luồng xử lý trong bộ mã hóa hoặc giải mã (hoặc cả hai). Một ví dụ được trình bày trong Hình 2.11(c). Đối với thiết kế đơn giản, WPP không được sử dụng kết hợp với các Tile.



**Hình 2.10 Các chế độ và các hướng cho dự đoán hình trong ảnh.**

### **g) Dự đoán trong ảnh**

Dự đoán trong ảnh hoạt động theo kích thước TB, và các mẫu biên giải mã trước đó từ những TB lân cận về mặt không gian được sử dụng để tạo ra tín hiệu dự đoán. Dự đoán trong ảnh hỗ trợ 33 hướng khác nhau với các kích thước TB từ 4x4 đến 32x32. Các hướng dự đoán được thể hiện trong Hình 2.12. Ngoài ra dự đoán mặt phẳng và dự đoán DC cũng có thể được sử dụng. Đối với thành phần màu, chế độ dự đoán ngang, dọc, mặt phẳng và DC có thể được báo hiệu một cách rõ ràng, hoặc chế độ dự đoán thành phần màu được chỉ định giống các chế độ dự đoán thành phần chói.

Mỗi CB được mã hóa bằng một trong những loại mã hóa, tùy thuộc vào loại mảng. Tương tự như H.264/MPEG-4/AVC, mã hóa dự đoán trong ảnh được hỗ trợ trong tất cả các loại mảng. HEVC hỗ trợ các phương pháp mã hóa dự đoán trong ảnh khác nhau gọi là *Intra\_Angular*, *Intra\_Planar*, và *Intra-DC*. Sau đây trình bày một lời giải thích ngắn gọn về những điều này và một số kỹ thuật được áp dụng chung.

### **Phân chia khối dự đoán (PB)**

Một CB dự đoán trong ảnh có kích thước  $M \times M$  có thể có một trong hai loại phân chia PB gọi là  $PART\_2N \times 2N$  và  $PART\_N \times N$ , loại đầu tiên chỉ ra rằng các CB không phân chia và loại thứ hai chỉ ra rằng CB được chia thành bốn PB kích thước bằng nhau. ( $N=M/2$ ). Tuy nhiên, nó có khả năng biểu diễn các vùng tương tự mà có thể được xác định bởi bốn PB bằng cách sử dụng bốn CB nhỏ hơn khi kích thước của các CB hiện tại lớn hơn so với kích thước CU tối thiểu. Vì vậy, HEVC chỉ cho phép kiểu phân chia  $PART\_N \times N$  được sử dụng khi kích thước CB hiện tại bằng với kích thước CU tối thiểu. Điều này có nghĩa rằng kích thước PB luôn bằng với kích thước CB khi CB được mã hóa bằng chế độ dự đoán trong ảnh và kích thước CB là không bằng với kích thước CU tối thiểu. Dù chế độ dự đoán trong ảnh ở mức PB, nhưng các quá trình dự đoán thực tại lại hoạt động riêng biệt cho từng TB.

### **Dự đoán *Intra\_Angular***

Dự đoán trong ảnh miền không gian trước đây đã được sử dụng trong H.264/MPEG-4/AVC. Dự đoán trong ảnh của HEVC hoạt động tương tự trong miền không gian, nhưng được mở rộng đáng kể, do kích thước của TB tăng và số lượng các hướng dự đoán tăng. So với tám hướng dự đoán của H.264/MPEG-4 AVC, HEVC hỗ trợ 33 hướng dự báo, ký hiệu là Intra\_Angular[k], k là số chế độ,  $k=2\ldots34$ . Các góc được thiết kế để cung cấp phạm vi dày hơn cho góc gần chiều ngang và gần thẳng đứng và phạm vi kém hơn cho góc gần đường chéo để phản ánh sự phổ biến thống kê quan sát được các góc và tính hiệu quả của việc xử lý dự đoán tín hiệu.

Khi sử dụng một chế độ Intra\_Angular, mỗi TB được dự đoán là có hướng từ các mẫu lân cận theo không gian đã được tái tạo lại (nhưng chưa được lọc bởi bộ lọc trong vòng) trước khi được sử dụng cho dự đoán này. Đối với một TB có kích thước  $N \times N$ , tổng  $4N+1$  mẫu lân cận theo không gian được sử dụng để dự đoán, như trong Hình 2.5. Khi hoạt động giải mã là có trước, các mẫu từ các TB phía dưới bên trái được sử dụng để dự đoán trong HEVC bên cạnh các mẫu từ các TB ở bên trái, phía trên, và phía trên bên phải của TB hiện tại.

Quá trình dự đoán của chế độ Intra\_Angular có thể liên quan đến mẫu ngoại suy từ vị trí mẫu tham chiếu dự đoán theo một hướng đã cho. Để loại bỏ sự chuyển đổi mẫu-theo-mẫu giữa các bộ đệm hàng và cột tham chiếu, cho Intra\_Angular [k] với  $k=2\ldots17$ , các mẫu nằm ở hàng trên được dự tính như mẫu bổ sung nằm ở cột bên trái và với  $k = 18\ldots34$ , các mẫu ở cột bên trái được dự tính nằm cùng với các mẫu hàng trên.

Để cải thiện tính chính xác dự đoán trong ảnh, vị trí mẫu tham chiếu dự tính được tính toán với độ chính xác  $1/32$  mẫu. Phép nội suy song tuyến tính được sử dụng để có giá trị của mẫu tham chiếu dự tính sử dụng hai mẫu tham chiếu gần nhất nằm ở vị trí số nguyên.

Quá trình dự đoán của các chế độ Intra\_Angular là nhất quán trên tất cả các kích thước khối và hướng dự đoán, trong khi H.264/MPEG-4/AVC sử dụng các phương pháp khác nhau cho các kích thước  $4 \times 4$ ,  $8 \times 8$  và  $16 \times 16$ . Thiết kế này là đặc biệt, vì HEVC hỗ trợ nhiều hơn về kích cỡ TB và tăng đáng kể các hướng dự đoán so với H.264/MPEG-4 AVC.

#### ***Dự đoán Intra-Planar và Intra-DC***

Bên cạnh dự đoán Intra\_Angular nhằm tới các vùng với các biên có hướng mạnh mẽ, HEVC hỗ trợ hai phương pháp dự đoán khác, Intra\_Planar và Intra\_DC, 2 chế độ tương tự được quy định trong H.264/MPEG-4/AVC. Trong khi dự đoán Intra\_DC sử dụng giá trị trung bình của các mẫu tham chiếu cho các dự đoán, giá trị trung bình của hai dự đoán tuyến tính sử dụng bốn mẫu tham chiếu góc được sử dụng trong dự đoán Intra\_Planar để ngăn chặn sự gián đoạn dọc theo biên khối. Các chế độ

dự đoán Intra\_Planar được hỗ trợ ở tất cả các kích thước khối trong HEVC, trong khi H.264/MPEG-4 AVC hỗ trợ dự đoán mặt phẳng chỉ khi PB thành phần chói là 16x16, và dự đoán mặt phẳng hoạt động hơi khác so với dự đoán mặt phẳng trong HEVC.

#### ***h) Dự đoán liên ảnh***

##### ***Phân chia PB***

So với các CB dự đoán trong ảnh, HEVC hỗ trợ nhiều hình dạng phân chia PB cho các CB dự đoán liên ảnh. Các chế độ phân chia của PART\_2Nx2N, PART\_2NxN, và PART\_Nx2N chỉ ra các trường hợp khi CB không được chia, chia tách thành hai PB kích thước bằng nhau theo chiều ngang, và chia thành hai PBs kích thước bằng nhau theo chiều dọc. PART\_NxN xác định rằng các CB được chia thành bốn PB kích thước bằng nhau, nhưng chế độ này chỉ được hỗ trợ khi kích thước CB bằng với kích thước CB cho phép nhỏ nhất. Ngoài ra, có bốn loại phân chia có hỗ trợ chia tách CB thành hai PB có kích cỡ khác nhau: PART\_2NxnU, PART\_2NxnD, PART\_nLx2N, và PART\_nRx2N. Những loại này được gọi là phân chia chuyển động không đối xứng.

##### ***Phép nội suy mẫu phân đoạn***

Các mẫu của PB cho một CB dự đoán trong ảnh đạt được từ một vùng khối tương ứng trong ảnh tham chiếu được xác định bởi một chỉ mục ảnh tham chiếu, mà nó ở một vị trí được thay thế bởi các thành phần ngang và dọc của vector chuyển động. Ngoại trừ trường hợp khi các vector chuyển động có một giá trị số nguyên, phép nội suy mẫu phân đoạn được sử dụng để tạo ra các mẫu dự đoán cho các vị trí lấy mẫu không nguyên. Như trong H.264/MPEG-4/AVC, HEVC hỗ trợ vector chuyển động với các đơn vị của một phần tư của khoảng cách giữa các mẫu thành phần chói. Đối với mẫu thành phần màu, tính chính xác vector chuyển động được xác định theo lấy mẫu 4:2:0 của 1/8 khoảng cách giữa các mẫu thành phần màu.

Các phép nội suy mẫu phân đoạn cho mẫu chói trong HEVC sử dụng ứng dụng riêng biệt: một bộ lọc tám-chạm cho các vị trí  $\frac{1}{2}$ -mẫu và một bộ lọc bảy-chạm cho các vị trí  $\frac{1}{4}$ -mẫu. Điều này trái ngược với quá trình được sử dụng trong H.264/MPEG-4 AVC, ở đó áp dụng một quá trình nội suy hai giai đoạn, đầu tiên tạo ra các giá trị của một hoặc hai mẫu lân cận ở vị trí  $\frac{1}{2}$ -mẫu sử dụng bộ lọc sáu-chạm, làm tròn kết quả trung gian, và sau đó tính trung bình hai giá trị tại vị trí số nguyên hoặc  $\frac{1}{2}$ -mẫu. HEVC sử dụng một phép nội suy phân tách phù hợp duy nhất tạo ra tất cả các vị trí phân đoạn mà không có hoạt động làm tròn trung gian, trong đó cải thiện độ chính xác và đơn giản hóa kiến trúc của phép nội suy mẫu phân đoạn. Độ chính xác nội suy cũng được cải thiện trong HEVC bằng cách sử dụng các bộ lọc dài hơn, tức là, bộ lọc bảy-chạm hoặc tám-chạm chứ không phải là bộ lọc sáu-chạm được sử dụng trong H.264/MPEG-4/AVC. Bằng cách chỉ sử dụng bảy-chạm chứ không phải tám-chạm cho các vị trí  $\frac{1}{2}$ -mẫu là đủ cho các vị trí phép nội suy  $\frac{1}{4}$ -mẫu từ các vị trí  $\frac{1}{4}$  mẫu là

tương đối gần vị trí số nguyên-mẫu, vì vậy các mẫu xa nhất trong một phép nội suy tám-chạm sẽ hiệu quả xa hơn trong trường hợp  $\frac{1}{2}$  - mẫu. Các giá trị chạm bộ lọc của bộ lọc phép nội suy được bắt nguồn một phần từ phương trình hàm cơ sở DCT.

<b>A<sub>-1,-1</sub></b>				<b>A<sub>0,-1</sub></b>	<b>a<sub>0,-1</sub></b>	<b>b<sub>0,-1</sub></b>	<b>c<sub>0,-1</sub></b>	<b>A<sub>1,-1</sub></b>				<b>A<sub>2,-1</sub></b>
<b>A<sub>-1,0</sub></b>				<b>A<sub>0,0</sub></b>	<b>a<sub>0,0</sub></b>	<b>b<sub>0,0</sub></b>	<b>c<sub>0,0</sub></b>	<b>A<sub>1,0</sub></b>				<b>A<sub>2,0</sub></b>
<b>d<sub>-1,0</sub></b>				<b>d<sub>0,0</sub></b>	<b>e<sub>0,0</sub></b>	<b>f<sub>0,0</sub></b>	<b>g<sub>0,0</sub></b>	<b>d<sub>1,0</sub></b>				<b>d<sub>2,0</sub></b>
<b>h<sub>-1,0</sub></b>				<b>h<sub>0,0</sub></b>	<b>i<sub>0,0</sub></b>	<b>j<sub>0,0</sub></b>	<b>k<sub>0,0</sub></b>	<b>h<sub>1,0</sub></b>				<b>h<sub>2,0</sub></b>
<b>n<sub>-1,0</sub></b>				<b>n<sub>0,0</sub></b>	<b>p<sub>0,0</sub></b>	<b>q<sub>0,0</sub></b>	<b>r<sub>0,0</sub></b>	<b>n<sub>1,0</sub></b>				<b>n<sub>2,0</sub></b>
<b>A<sub>-1,1</sub></b>				<b>A<sub>0,1</sub></b>	<b>a<sub>0,1</sub></b>	<b>b<sub>0,1</sub></b>	<b>c<sub>0,1</sub></b>	<b>A<sub>1,1</sub></b>				<b>A<sub>2,1</sub></b>
<b>A<sub>-1,2</sub></b>				<b>A<sub>0,2</sub></b>	<b>a<sub>0,2</sub></b>	<b>b<sub>0,2</sub></b>	<b>c<sub>0,2</sub></b>	<b>A<sub>1,2</sub></b>				<b>A<sub>2,2</sub></b>

**Hình 2.11 Vị trí lấy mẫu số nguyên và phân đoạn cho phép nội suy thành phần chói**

Trong Hình 2.13, vị trí được dán nhãn với chữ cái hoa  $A_{i,j}$ , biểu diễn các mẫu chói có sẵn tại các vị trí mẫu số nguyên, trong khi các vị trí khác có nhãn với chữ cái thường biểu diễn các mẫu tại các vị trí mẫu không nguyên, được tạo ra bằng phép nội suy.

Các mẫu dán nhãn  $a_{0,j}, b_{0,j}, c_{0,j}, d_{0,0}, h_{0,0}$  và  $n_{0,0}$  được dẫn xuất từ các mẫu  $A_{i,j}$  bằng cách áp dụng các bộ lọc tám-chạm cho các vị trí nửa-mẫu và bộ lọc bảy-chạm cho các vị trí phần tư - mẫu như sau:

$$\begin{aligned}
 a_{0,j} & \left( \sum_{i=-3..3} A_{i,j} qfilter[i] \right) \gg (B-8) & d_{0,0} & \left( \sum_{i=-3..3} A_{0,j} qfilter[j] \right) \gg (B-8) \\
 b_{0,j} & \left( \sum_{i=-3..4} A_{i,j} hfilter[i] \right) \gg (B-8) & h_{0,0} & \left( \sum_{i=-3..4} A_{0,j} qfilter[j] \right) \gg (B-8) \\
 c_{0,j} & \left( \sum_{i=-2..4} A_{i,j} qfilter[1-i] \right) \gg (B-8) & n_{0,0} & \left( \sum_{i=-2..4} A_{00,j} qfilter[1-j] \right) \gg (B-8)
 \end{aligned}$$

Trong đó: hằng số  $B \geq 8$  là độ sâu bit của mẫu tham chiếu (và thường là  $B=8$  cho hầu hết các ứng dụng) và các giá trị hệ số bộ lọc được đưa ra trong Bảng 2.1. Trong các công thức này,  $\gg$  biểu thị một việc dịch phải số học.

**Bảng 2.1 Các hệ số bộ lọc cho phép nội suy lấy mẫu phân đoạn chọi**

Index $i$	-3	-2	-1	0	1	2	3	4
$hfilter[i]$	-1	4	-11	40	40	-11	4	1
$qfilter[i]$	-1	4	-10	58	17	-5	1	

Các mẫu dán nhãn  $e_{0,0}$ ,  $f_{0,0}$ ,  $g_{0,0}$ ,  $i_{0,0}$ ,  $j_{0,0}$ ,  $k_{0,0}$ ,  $p_{0,0}$ ,  $q_{0,0}$  và  $r_{0,0}$  được dẫn xuất bằng cách áp dụng các bộ lọc tương ứng với các mẫu nằm ở vị trí  $a_{0,j}$ ,  $b_{0,j}$ , và  $c_{0,j}$  theo chiều dọc liên kế như sau:

$$\begin{aligned}
 e_{0,0} & \left( \sum_{v=-3..3} a_{0,v} qfilter[v] \right) \gg 6 & k_{0,0} & \left( \sum_{v=-3..4} c_{0,v} hfilter[v] \right) \gg 6 \\
 f_{0,0} & \left( \sum_{v=-3..3} b_{0,v} qfilter[v] \right) \gg 6 & p_{0,0} & \left( \sum_{v=-2..4} a_{0,v} qfilter[1-v] \right) \gg 6 \\
 g_{0,0} & \left( \sum_{v=-3..3} c_{0,v} qfilter[v] \right) \gg 6 & q_{0,0} & \left( \sum_{v=-2..4} b_{0,v} qfilter[1-v] \right) \gg 6 \\
 i_{0,0} & \left( \sum_{v=-3..4} a_{0,v} hfilter[v] \right) \gg 6 & r_{0,0} & \left( \sum_{v=-2..4} c_{0,v} qfilter[1-v] \right) \gg 6 \\
 j_{0,0} & \left( \sum_{v=-3..4} b_{0,v} hfilter[v] \right) \gg 6 & & 
 \end{aligned}$$

Việc lọc nội suy là có thể tách rời khi B bằng 8, vì vậy các giá trị tương tự có thể được tính toán trong trường hợp này bằng cách áp dụng lọc theo chiều dọc trước khi lọc theo chiều ngang. Khi được thực hiện một cách phù hợp, quá trình bù chuyển động của HEVC có thể được thực hiện chỉ sử dụng phần tử lưu trữ 16-b.

Tại điểm này trong quá trình dự đoán trọng số được áp dụng được chọn bởi các bộ mã hóa. Trong khi H.264/MPEG-4/AVC hỗ trợ cả hai dự đoán trọng số ẩn và rõ ràng theo thời gian, thì HEVC chỉ áp dụng dự đoán trọng số rõ ràng, bằng cách định cỡ (scale) và xê dịch những dự đoán với các giá trị được gửi một cách rõ ràng bằng các bộ mã hóa. Độ sâu bit của dự đoán sau đó được điều chỉnh đến độ sâu bit ban đầu của mẫu tham chiếu. Trong trường hợp dự đoán đơn, các giá trị dự đoán được nội suy (và có thể được trọng số) được làm tròn, dịch-phải, và cắt bớt để có độ sâu bit ban đầu. Trong trường hợp dự đoán đôi, giá trị dự đoán được nội suy (và có thể trọng số) từ hai PB được thêm vào đầu tiên, và sau đó được làm tròn, dịch phải, và cắt bớt.

Trong H.264/MPEG-4/AVC, tăng đến ba giai đoạn của hoạt động làm tròn để đạt được mỗi mẫu dự đoán (cho mẫu nằm tại các vị trí  $\frac{1}{4}$  - mẫu). Nếu dự đoán đôi được sử dụng, tổng các hoạt động làm tròn là bảy trong trường hợp xấu nhất. Trong HEVC, nhiều nhất hai hoạt động làm tròn để đạt được mỗi mẫu nằm tại các vị trí  $\frac{1}{4}$  mẫu, năm hoạt động làm tròn là đủ trong trường hợp xấu nhất khi dự đoán đôi được sử dụng. Hơn nữa, trong sử dụng phổ biến nhất, nơi mà độ sâu bit  $B=8b$ , tổng của các hoạt động làm tròn trong trường hợp xấu nhất là tiếp tục giảm xuống còn 3. Do ít các hoạt động làm tròn, lỗi làm tròn số bị tích lũy là giảm và có tính linh hoạt hơn đến cách thức thực hiện các hoạt động cần thiết trong các bộ giải mã.

**Bảng 2.2 Các hệ số bộ lọc cho phép nội suy lấy mẫu phân đoạn chroma**

Index	-1	0	1	2
filter1[i]	-2	58	10	-2
filter2[i]	-4	54	16	-2
filter3[i]	-6	46	28	-4
filter4[i]	-4	36	36	-4

Quá trình phép nội suy mẫu phân đoạn cho các thành phần màu là tương tự như đối với các thành phần chói, ngoại trừ số lượng các chạm bộ lọc là 4 và độ chính xác phân đoạn là  $\frac{1}{8}$  cho trường hợp định dạng màu lấy mẫu 4:2:0. HEVC định nghĩa một tập các bộ lọc bốn-chạm cho vị trí  $\frac{1}{8}$ -mẫu, như được đưa ra trong Bảng 2.2 đối với trường hợp định dạng màu 4:2:0 (trong H.264/MPEG-4/AVC, chỉ có lọc song tuyến tính 2-chạm được áp dụng).

Các giá trị hệ số bộ lọc ký hiệu là filter1[i], filter2[i], filter3[i], và filter4[i] với  $i=-1...2$  được sử dụng cho nội suy vị trí phân đoạn  $\frac{1}{8}$ th,  $\frac{2}{8}$ th,  $\frac{3}{8}$ th, và  $\frac{4}{8}$ th cho các



mẫu thành phần màu. Bằng cách sử dụng tính đối xứng cho vị trí phân đoạn 5/8th, 6/8th và 7/8th, các giá trị nhân đôi của  $\text{filter3}[1-i]$ ,  $\text{filter2}[1-i]$ , và  $\text{filter1}[1-i]$  with  $i=-1 \dots 2$  tương ứng được sử dụng.

### ***g) Mã hóa Entropy***

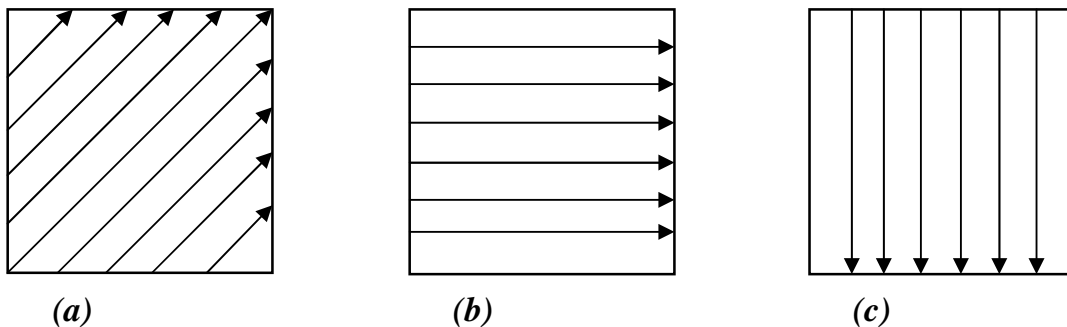
HEVC chỉ rõ chỉ có một phương thức mã hóa Entropy, CABAC [2] không phải là hai như trong H.264 / MPEG-4 AVC. Các thuật toán cốt lõi của CABAC là không thay đổi, và các phần dưới đây trình diễn một số khía cạnh của nó trong HEVC.

### ***Mô hình hóa ngữ cảnh***

Lựa chọn phù hợp của mô hình hóa ngữ cảnh là một yếu tố quan trọng để nâng cao hiệu quả của mã hóa CABAC. Trong HEVC, độ sâu chia tách của cây mã hóa hoặc cây biến đổi được khai thác để dẫn xuất các chỉ số mô hình ngữ cảnh của các yếu tố cú pháp khác nhau, thêm vào những lân cận không gian sử dụng trong H.264/AVC. Mặc dù số lượng các ngữ cảnh sử dụng trong HEVC là ít hơn so với H.264/MPEG-4/AVC, thiết kế mã hóa Entropy thực sự cung cấp độ nén tốt hơn sẽ là một phần mở rộng thẳng của sơ đồ H.264/MPEG-4/AVC. Hơn nữa, việc sử dụng rộng rãi hơn được tạo ra trong HEVC của chế độ đường vòng của các hoạt động CABAC để tăng thông lượng bằng cách giảm lượng dữ liệu được mã hóa sử dụng ngữ cảnh CABAC. Phụ thuộc giữa các dữ liệu được mã hóa cũng được xem xét để cho phép tối đa hóa thông lượng hơn nữa.

### ***Quét hệ số thích ứng***

Quét hệ số được thực hiện trong các khối con 4x4 cho tất cả các kích cỡ TB. Ba phương pháp quét hệ số, đường chéo lên-bên phải, theo chiều ngang, và theo chiều dọc như Hình 2.14, được lựa chọn mặc nhiên để mã hóa các biến đổi hệ số của TB 4x4 và 8x8 trong vùng dự đoán trong ảnh. Việc lựa chọn các trình tự quét hệ số phụ thuộc vào các hướng của dự đoán trong ảnh. Quét theo chiều dọc được sử dụng khi hướng dự đoán gần với chiều ngang và quét chiều ngang được sử dụng khi các hướng dự đoán gần thẳng đứng. Quét đường chéo lên-bên phải được sử dụng cho các hướng dự đoán khác.



**Hình 2.12 Ba chế phương pháp quét hệ số trong HEVC**

**(a) Quét chéo lên bên phải. (b) Quét theo chiều ngang. (c) Quét theo**

### **chiều dọc.**

Đối với các hệ số biến đổi trong chế độ dự đoán liên ảnh của tất cả các kích thước khối và các hệ số biến đổi của dự đoán trong ảnh 16x16 hoặc 32x32, quét đường chéo lên-bên phải 4x4 là được áp dụng dành riêng cho các khối con của các hệ số biến đổi.

#### **Mã hóa hệ số**

Tương tự như H.264/MPEG-4/AVC, HEVC truyền đi vị trí của hệ số biến đổi khác-0 cuối cùng, 1 ánh xạ quan trọng, các bit dấu và các mức của các hệ số biến đổi. Tuy nhiên, những thay đổi khác nhau cho từng phần đã được tạo ra, đặc biệt là xử lý tốt hơn việc tăng kích thước đáng kể các TB.

#### **h) Bộ lọc trong vòng**

Trong HEVC, hai bước xử lý là bộ lọc bỏ khối (bung khối) (DBF) tiếp theo sau là một bộ lọc bù thích ứng mẫu (SAO), được áp dụng cho các mẫu được khôi phục trước khi ghi chúng vào bộ đệm ảnh giải mã trong vòng lặp bộ giải mã. DBF làm giảm những thành phần khối do mã hóa dạng khối tạo ra. Các DBF tương tự như DBF của chuẩn H.264/MPEG-4/AVC, trong khi SAO mới được giới thiệu trong HEVC. Trong khi DBF chỉ được áp dụng cho các mẫu nằm ở biên khối, bộ lọc SAO được áp dụng thích nghi với tất cả các mẫu đáp ứng các điều kiện nhất định, ví dụ, dựa trên gradient. Trong sự phát triển của HEVC, nó cũng đã được xem xét để vận hành một bước xử lý thứ ba gọi là bộ lọc vòng lặp thích ứng (ALF) sau khi lọc SAO; Tuy nhiên, tính năng ALF chưa có trong thiết kế cuối cùng của HEVC.

#### **Bộ lọc bỏ khối**

Bộ lọc bỏ khối được áp dụng cho tất cả các mẫu liền kề với biên PU hoặc biên TU trừ trường hợp khi các biên cũng là một biên ảnh, hoặc khi bỏ khối bị vô hiệu hóa ngang qua các biên mảng hoặc Tile. Cần lưu ý rằng cả hai biên PU và TU cần được xem xét vì các biên PU không phải luôn luôn bằng nhau với biên TU trong một số trường hợp của các CB dự đoán liên ảnh. Các yếu tố cú pháp trong SPS và các điều khiển phần đầu mảng có hay chẳng bộ lọc bỏ khối được áp dụng qua các biên mảng và Tile.

Sức mạnh của bộ lọc bỏ khối được điều khiển bởi các giá trị của một số yếu tố cú pháp tương tự như trong H.264/MPEG-4AVC, nhưng chỉ có ba cái mạnh được sử dụng hơn là năm. P và Q cho trước là hai khối liền kề với một biên lưới 8x8 chung, sức mạnh bộ lọc bằng 2 được gán vào khi một trong những khối là dự đoán trong ảnh. Mặt khác, sức mạnh bộ lọc bằng 1 được gán vào nếu bất kỳ điều kiện sau đây được thỏa mãn:

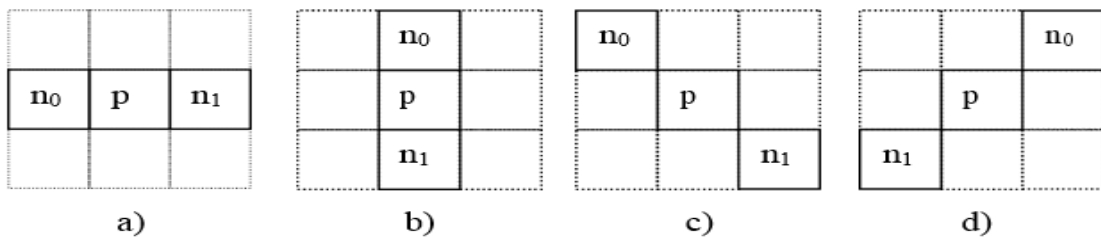
- P hay Q có ít nhất một hệ số biến đổi khác-0
- Các chỉ số tham chiếu của P và Q không bằng nhau.
- Các vectơ chuyển động của P và Q không bằng nhau.

- Sự khác biệt giữa một thành phần vector chuyển động của P và Q là lớn hơn hoặc bằng một mẫu-số nguyên.

Nếu không có điều kiện trên được đáp ứng, sức mạnh độ bộ lọc là 0, có nghĩa là quá trình bỏ khối không được áp dụng.

Theo như sức mạnh bộ lọc và tham số lượng tử hóa trung bình của P và Q, hai ngưỡng  $t_c$  và  $\beta$ , được xác định từ các bảng cho trước. Đối với mẫu thành phần chói, một trong ba trường hợp, không lọc, lọc mạnh mẽ, và lọc yếu, được lựa chọn dựa trên  $\beta$ . Lưu ý rằng quyết định này được chia sẻ qua bốn hàng hoặc cột chói sử dụng hàng cột đầu tiên và cuối cùng để giảm độ phức tạp tính toán.

Chỉ có hai trường hợp, không có lọc và lọc thường, đối với mẫu thành phần màu. Lọc thường chỉ được áp dụng khi các cường độ bộ lọc là lớn hơn 1. Sau đó quá trình lọc được thực hiện bằng cách sử dụng các biến kiểm soát  $t_c$  và  $\beta$ .



**Hình 2.13 Bốn dạng gradient sử dụng trong SAO**

Mẫu dán nhãn "p" cho biết một mẫu trung tâm để xem xét. Hai mẫu dán nhãn " $n_0$ " và " $n_1$ " chỉ rõ hai mẫu lân cận (a) theo chiều ngang ( $\text{sao\_eo\_class} = 0$ ), (b) theo chiều dọc ( $\text{sao\_eo\_class} = 1$ ), (c) theo đường chéo  $135^\circ$  ( $\text{sao\_eo\_class} = 2$ ), và (d) theo dạng gradient  $45^\circ$  ( $\text{sao\_eo\_class} = 3$ ).

### SAO

SAO là một quá trình sửa đổi các mẫu được giải mã bằng cách thêm một giá trị bù đắp vào mỗi mẫu sau ứng dụng các bộ lọc bỏ khối, dựa trên các giá trị trong bảng look-up truyền bởi bộ mã hóa. Lọc SAO được thực hiện trên vùng, dựa trên một loại lọc lựa chọn mỗi CTB bởi một yếu tố cú pháp  $\text{sao\_type\_idx}$ . Giá trị 0 cho  $\text{sao\_type\_idx}$  chỉ ra các bộ lọc SAO không áp dụng cho CTB, và các giá trị 1 và 2 báo hiệu sử dụng lọc bù bằng tần và bù cạnh.

Trong chế độ bù bằng tần đã xác định bởi  $\text{sao\_type\_idx}$  bằng 1, các giá trị bù được chọn phụ thuộc trực tiếp vào biên độ mẫu. Trong chế độ này, dải biên độ mẫu đầy đủ được thống nhất chia thành 32 phân đoạn được gọi là băng tần và các giá trị mẫu thuộc bốn trong những băng tần này được sửa đổi bằng cách thêm vào các giá trị truyền đi biểu hiện bao hàm bù băng tần, nó có thể là dương hay âm. Lý do chính để sử dụng bốn băng tần liên tiếp là trong các khu vực trơn nơi thành phần băng tần có thể xuất hiện, các biên độ mẫu trong một CTB hướng tập trung vào một số băng tần. Ngoài ra,

chọn sử dụng bốn giá trị bù là thống nhất với chế độ bù cạnh của hoạt động mà nó cũng sử dụng bốn giá trị bù.

Trong chế độ bù cạnh (sao\_type\_idx=2), một cú pháp sao\_eo\_class với giá trị từ 0 đến 3 báo hiệu có hay không hướng gradient theo chiều dọc, ngang hoặc 1 trong 2 hướng gradient chéo được sử dụng cho sự phân loại bù cạnh trong CTB. Hình 2.15 mô tả bốn dạng gradient sử dụng cho các sao\_eo\_class tương ứng trong chế độ này. Mỗi mẫu trong CTB được phân loại thành một trong năm loại EdgeMax bằng cách so sánh giá trị p mẫu đặt tại một số vị trí với các giá trị n<sub>0</sub> và n<sub>1</sub> của hai mẫu nằm tại các vị trí lân cận như thể hiện trong bảng 2.3. Việc phân loại này được thực hiện đối với từng mẫu dựa trên các giá trị mẫu được giải mã, vì vậy không có tín hiệu hóa bổ sung cho việc phân loại Edgeldx. Tùy thuộc vào loại Edgeldx ở vị trí mẫu, cho loại EdgeIdx từ 1 tới 4, một giá trị bù từ một bảng look-up đc truyền đi được thêm vào các giá trị mẫu. Các giá trị bù luôn dương cho loại 1 và 2 và âm đối với loại 3 và 4, do đó bộ lọc thường có 1 hiệu ứng làm trơn ở chế độ bù cạnh.

**Bảng 2.3 Các loại mẫu EdgeIdx trong các lớp cạnh biên SAO**

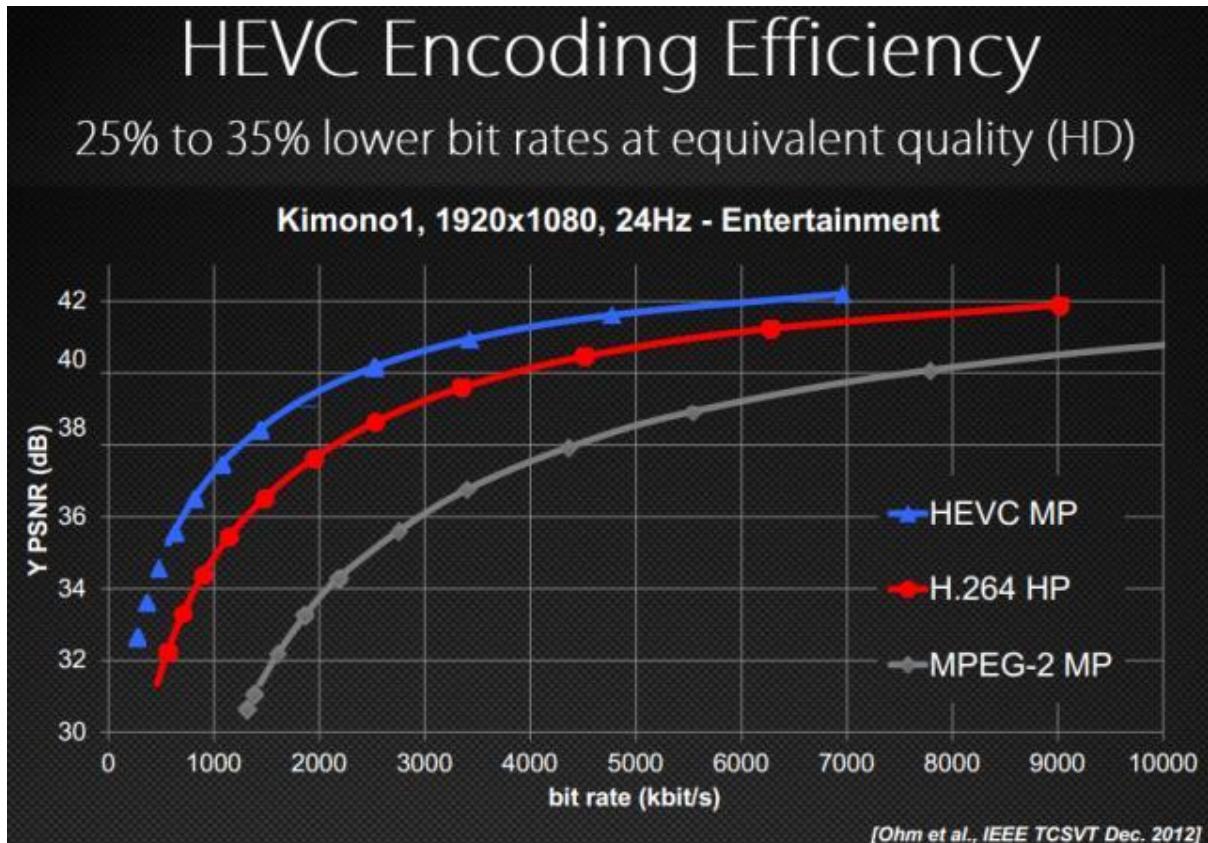
EdgeIdx	Điều kiện	Ý nghĩa
0	Các TH k đc liệt kê	Khu vực đơn điệu
1	$p < n_0$ và $p < n_1$	Tối thiểu cục bộ
2	$p < n_0$ và $p = n_1$ hoặc $p < n_1$ và $p = n_0$	cạnh biên
3	$p > n_0$ và $p = n_1$ hoặc $p > n_1$ và $p = n_0$	cạnh biên
4	$p > n_0$ và $p > n_1$	tối đa cục bộ

Như vậy, với SAO loại 1 và 2, tổng của 4 giá trị bù biên độ được truyền đến các bộ giải mã cho mỗi CTB. Đối với loại 1, các báo hiệu cũng được mã hóa. Các giá trị bù và các yếu tố cú pháp liên quan như sao\_type\_idx và sao\_eo\_class được xác định bởi bộ mã hóa thường tối ưu hóa hiệu suất tỷ lệ - biến dạng. Các tham số SAO được chỉ dẫn để được thừa kế từ CTB bên trái hoặc bên trên sử dụng một cờ kết hợp để tạo ra hiệu quả phát tín hiệu. Tóm lại, SAO là một hoạt động lọc phi tuyến tính cho phép làm trơn bổ sung của các tín hiệu tái tạo, và nó có thể cải thiện sự biểu diễn tín hiệu trong cả hai khu vực trơn và xung quanh các cạnh biên.

## 2.2 So sánh hiệu năng nén của h.265 với một số chuẩn nén trước.

Về mặt lý thuyết H.265/HEVC được đánh giá là hiệu quả hơn H.264/AVC khoảng 25-35% . Ở độ phân giải càng cao, thì hiệu suất nén của H.265/HEVC càng

cao. H.265/HEVC cơ bản có cấu trúc tương tự như các tiêu chuẩn trước đó như MPEG-2 và H.264/AVC.



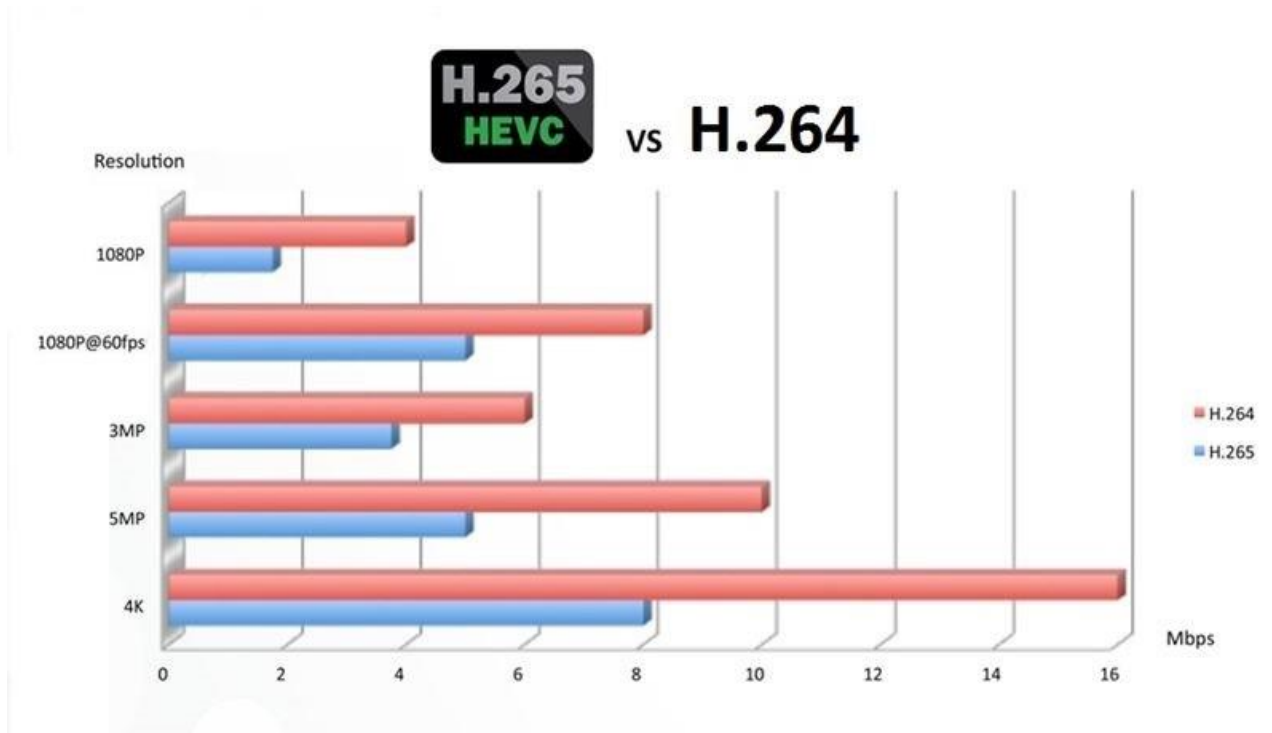
**Hình 2.14: So sánh về hiệu năng nén của H.265 với một số chuẩn nén trước đây**

Chất lượng H.265/HEVC so với H.264/AVC, hình bên dưới là một ví dụ so sánh H.265/HEVC và H.264/AVC đây là hai khung hình video ở độ phân giải HD 720p. Bên trái là sử dụng chuẩn nén H.264, bên phải sử dụng chuẩn nén H.265 với bitrate 420kbps. Có thể thấy chất lượng của video H.265 tốt hơn video H.264 khi các vùng màu thể hiện rõ ràng và sắc nét hơn, vùng tóc cũng thể hiện mượt mà hơn.

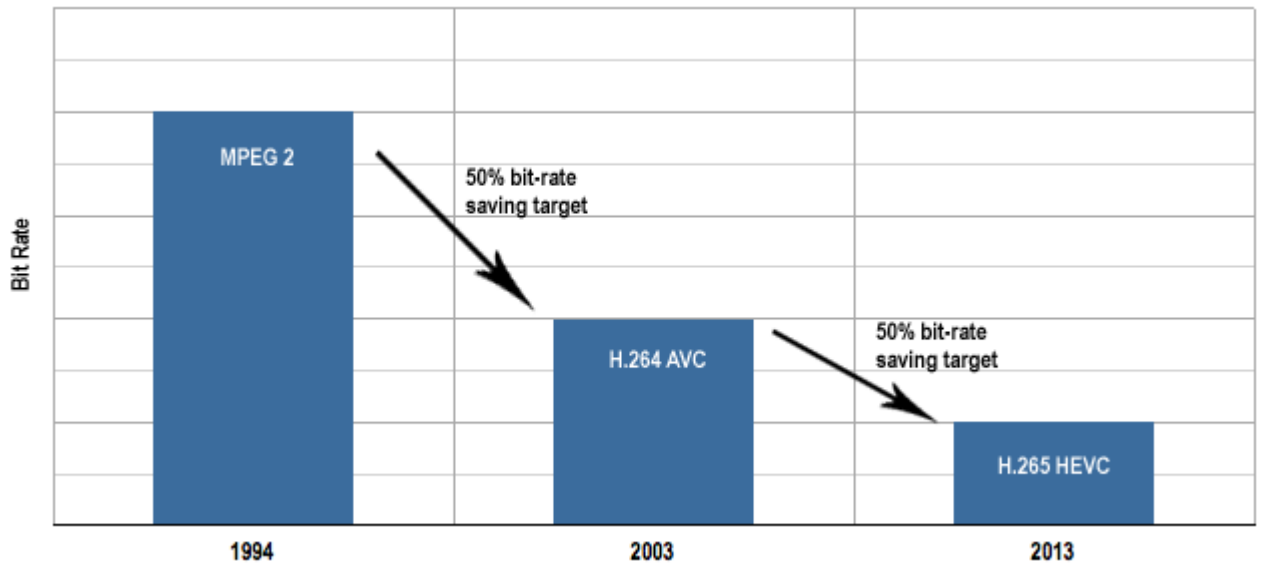


**Hình 2.15 So sánh H.264/AVC và H.265/HEVC khung hình video ở độ phân giải HD 720p**

Thời gian tải video H.265 so với video H.264 với cùng một tốc độ mạng. Chúng ta có thể thấy rõ được video chuẩn nén H.265 lợi như nào.



**Hình 2.16 Thời gian tải video H.265 so với video H.264 với cùng một tốc độ mạng**



**Hình 2.17: Kích thước file nén qua các thời kì**

Kết quả trên hình chỉ ra rằng so với các chuẩn trước đây, H.265 cần số lượng bit ít hơn để mã hóa nhưng đồng thời vẫn đảm bảo chất lượng hình ảnh tốt hơn. Ví dụ, với cùng tốc độ bit là 1000kb/s, chất lượng đạt được của H.265 là 37.5dB trong khi của H.264 chỉ là 35dB.

❖ So sánh hiệu quả nén H264 và H265 với chuỗi Akiyo.

```

pi@raspberrypi: ~
File Edit Tabs Help
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~ $ ./x264/x264 --input-res 352x288 --fps 25 -o akiyo264.flv /home
/pi/Downloads/akiyo_cif.yuv
yuv [info]: 352x288p 0:0 @ 25/1 fps (cfr)
x264 [info]: using cpu capabilities: ARMv6 NEON
x264 [info]: profile High, level 1.3
flv [info]: initial delay 80 ms
x264 [info]: frame I:2 Avg QP:17.59 size: 14526
x264 [info]: frame P:76 Avg QP:22.24 size: 1204
x264 [info]: frame B:222 Avg QP:28.77 size: 115
x264 [info]: consecutive B-frames: 1.0% 0.7% 1.0% 97.3%
x264 [info]: mb I I16..4: 5.6% 66.3% 28.2%
x264 [info]: mb P I16..4: 0.0% 0.0% 0.0% P16..4: 15.8% 10.1% 7.8% 0.0% 0
.0% skip:66.3%
x264 [info]: mb B I16..4: 0.0% 0.0% 0.0% B16..8: 13.6% 1.3% 0.2% direct:
0.3% skip:84.7% L0:37.0% L1:55.0% BI: 8.0%
x264 [info]: 8x8 transform intra:65.6% inter:51.1%
x264 [info]: coded y,uvDC,uvAC intra: 93.1% 91.9% 80.5% inter: 3.4% 2.3% 0.5%
x264 [info]: i16 v,h,dc,p: 52% 11% 2% 34%
x264 [info]: i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 23% 27% 18% 4% 4% 5% 6% 6% 6%
x264 [info]: i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 29% 9% 12% 6% 7% 13% 5% 12% 6%
x264 [info]: i8c dc,h,v,p: 40% 24% 26% 10%
x264 [info]: Weighted P-Frames: Y:0.0% UV:0.0%
x264 [info]: ref P L0: 60.6% 16.0% 14.1% 9.3%
x264 [info]: ref B L0: 89.4% 7.9% 2.8%
x264 [info]: ref B L1: 95.9% 4.1%
x264 [info]: kb/s:97.48

encoded 300 frames, 82.92 fps, 97.97 kb/s
pi@raspberrypi:~ $

```

**Hình 2.18: Kết quả nén của H264 với video Akiyo**



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ./x265/build/arm-linux/x265 --input-res 352x288 --fps 24 -o a
kyo265.flv /home/pi/Downloads/akiyo_cif.yuv
yuv [info]: 352x288 fps 24000/1000 i420p8 frames 0 - 299 of 300
raw [info]: output file: akyo265.flv
x265 [info]: HEVC encoder version 2.6+11-94dc146c5f67
x265 [info]: build info [Linux][GCC 6.3.0][32 bit] 8bit
x265 [info]: using cpu capabilities: ARMv6 NEON
x265 [info]: Main profile, Level-2 (Main tier)
x265 [info]: Thread pool created using 4 threads
x265 [info]: Slices : 1
x265 [info]: frame threads / pool features : 2 / wpp(5 rows)
x265 [warning]: Source height < 720p; disabling lookahead-slices
x265 [info]: Coding QT: max CU size, min CU size : 64 / 8
x265 [info]: Residual QT: max TU size, max depth : 32 / 1 inter / 1 intra
x265 [info]: ME / range / subpel / merge : hex / 57 / 2 / 2
x265 [info]: Keyframe min / max / scenecut / bias: 24 / 250 / 40 / 5.00
x265 [info]: Lookahead / bframes / badapt : 20 / 4 / 2
x265 [info]: b-pyramid / weightp / weightb : 1 / 1 / 0
x265 [info]: References / ref-limit cu / depth : 3 / on / on
x265 [info]: AQ: mode / str / qg-size / cu-tree : 1 / 1.0 / 32 / 1
x265 [info]: Rate Control / qCompress : CRF-28.0 / 0.60
x265 [info]: tools: rd=3 psy-rd=2.00 rskip signhide tmvp strong-intra-smoothing
x265 [info]: tools: deblock sao
x265 [info]: frame I: 2, Avg QP:24.86 kb/s: 1229.57
x265 [info]: frame P: 70, Avg QP:29.71 kb/s: 103.89
x265 [info]: frame B: 228, Avg QP:35.23 kb/s: 12.15
x265 [info]: Weighted P-Frames: Y:0.0% UV:0.0%
x265 [info]: consecutive B-frames: 1.4% 4.2% 5.6% 54.2% 34.7%
encoded 300 frames in 21.19s (14.16 fps), 41.67 kb/s, Avg QP:33.87
pi@raspberrypi:~ $

```

*Hình 2.19: Kết quả nén của H265/HEVC với video Akiyo*

❖ So sánh hiệu quả nén H264 và H265 với chuỗi Big Buck Bunny.

```

pi@raspberrypi: ~
File Edit Tabs Help
encoded 300 frames, 82.92 fps, 97.97 kb/s
pi@raspberrypi:~ $ ./x264/x264 --input-res 352x288 --fps 24 -o bbb264.flv /home/
pi/Downloads/BBB_CIF_24fps
avs [error]: failed to load avisynth
raw [info]: 352x288p 0:0 @ 24/1 fps (cfr)
x264 [info]: using cpu capabilities: ARMv6 NEON
x264 [info]: profile High, level 1.3
flv [info]: initial delay 83 ms
x264 [info]: frame I:151 Avg QP:18.69 size: 21002
x264 [info]: frame P:4688 Avg QP:23.19 size: 2462
x264 [info]: frame B:9476 Avg QP:27.96 size: 485
x264 [info]: consecutive B-frames: 9.1% 4.2% 11.4% 75.3%
x264 [info]: mb I I16..4: 8.4% 56.3% 35.2%
x264 [info]: mb P I16..4: 0.6% 3.1% 1.1% P16..4: 27.7% 11.8% 9.4% 0.0% 0
.0% skip:46.3%
x264 [info]: mb B I16..4: 0.1% 0.5% 0.2% B16..8: 20.6% 3.3% 1.2% direct:
1.2% skip:72.8% L0:39.5% L1:50.3% BI:10.2%
x264 [info]: 8x8 transform intra:61.1% inter:59.7%
x264 [info]: coded y,uvDC,uvAC intra: 75.2% 80.8% 60.0% inter: 10.5% 8.8% 2.6%
x264 [info]: i16 v,h,dc,p: 29% 21% 7% 42%
x264 [info]: i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 22% 13% 16% 5% 8% 11% 6% 10% 8%
x264 [info]: i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 21% 12% 14% 7% 10% 12% 7% 10% 7%
x264 [info]: i8c dc,h,v,p: 49% 17% 21% 12%
x264 [info]: Weighted P-Frames: Y:4.1% UV:2.5%
x264 [info]: ref P L0: 65.9% 16.7% 10.4% 6.7% 0.2%
x264 [info]: ref B L0: 80.1% 14.6% 5.3%
x264 [info]: ref B L1: 94.4% 5.6%
x264 [info]: kb/s:258.93
encoded 14315 frames, 53.31 fps, 258.94 kb/s
pi@raspberrypi:~ $

```

*Hình 2.20: Kết quả nén của H264 với video Big Buck Bunny*



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ./x265/build/arm-linux/x265 --input-res 352x288 --fps 24 -o b
bb265.flv /home/pi/Downloads/BBB_CIF_24fps
yuv [info]: 352x288 fps 24000/1000 i420p8 frames 0 - 14314 of 14315
raw [info]: output file: bbb265.flv
x265 [info]: HEVC encoder version 2.6+11-94dc146c5f67
x265 [info]: build info [Linux][GCC 6.3.0][32 bit] 8bit
x265 [info]: using cpu capabilities: ARMv6 NEON
x265 [info]: Main profile, Level-2 (Main tier)
x265 [info]: Thread pool created using 4 threads
x265 [info]: Slices : 1
x265 [info]: frame threads / pool features : 2 / wpp(5 rows)
x265 [warning]: Source height < 720p; disabling lookahead-slices
x265 [info]: Coding QT: max CU size, min CU size : 64 / 8
x265 [info]: Residual QT: max TU size, max depth : 32 / 1 inter / 1 intra
x265 [info]: ME / range / subpel / merge : hex / 57 / 2 / 2
x265 [info]: Keyframe min / max / scenecut / bias: 24 / 250 / 40 / 5.00
x265 [info]: Lookahead / bframes / badapt : 20 / 4 / 2
x265 [info]: b-pyramid / weightp / weightb : 1 / 1 / 0
x265 [info]: References / ref-limit cu / depth : 3 / on / on
x265 [info]: AQ: mode / str / qg-size / cu-tree : 1 / 1.0 / 32 / 1
x265 [info]: Rate Control / qCompress : CRF-28.0 / 0.60
x265 [info]: tools: rd=3 psy-rd=2.00 rskip signhide tmvp strong-intra-smoothing
x265 [info]: tools: deblock sao
x265 [info]: frame I: 150, Avg QP:25.99 kb/s: 1778.45
x265 [info]: frame P: 3879, Avg QP:29.63 kb/s: 229.32
x265 [info]: frame B: 10286, Avg QP:35.41 kb/s: 40.07
x265 [info]: Weighted P-Frames: Y:5.5% UV:3.4%
x265 [info]: consecutive B-frames: 10.9% 7.3% 17.9% 43.3% 20.6%

encoded 14315 frames in 1662.62s (8.61 fps), 109.56 kb/s, Avg QP:33.75
pi@raspberrypi:~ $

```

**Hình 2.21: Kết quả của H265/HEVC với video Big Buck Bunny**

Qua cả 2 thử nghiệm với 2 loại nén ta có thể thấy rằng:

H265 nén mất khá nhiều thời gian hơn, tính toán lâu hơn. Tuy nhiên, bù lại, H265 có dung lượng nén gấp ~ 2.3 lần của H264.

### 2.3 Kết luận chương.

Trong phần này, các kỹ thuật giải mã video sẽ được nghiên cứu nhằm tìm ra bộ giải mã phù hợp nhất trong hệ thống video streaming. Cụ thể, các kỹ thuật giải mã video nói chung và các kỹ thuật tiên tiến được sử dụng trong chuẩn nén H.265 được giới thiệu. Ngoài ra, một số kỹ thuật giải mã trong truyền dẫn video qua mạng quang vô tuyến cũng đã được giới thiệu trong phần này. Các kỹ thuật này được đánh giá và so sánh với kỹ thuật trước đây như H.264 để thấy rõ các cải tiến của H.265. Cụ thể, các kỹ thuật áp dụng trong mỗi giai đoạn giải mã sẽ được so sánh và đánh giá ưu nhược điểm so với các kỹ thuật trong các chuẩn nén H.264 trước đây.

Các nghiên cứu trong phần này đã chỉ ra rằng, chất lượng video tại phía thu trong chuẩn nén H.265 có thể được cải thiện so với các chuẩn nén trước đây. Ngoài ra hiệu quả nén của H.265 cũng được tăng lên đáng kể. Tuy nhiên, cấu trúc của bộ giải mã H.265 phức tạp hơn so với H.264. Điều này cũng dẫn đến thời gian giải mã của H.265 cũng tăng lên so với H.264. Tuy nhiên, trong một số chế độ mã hóa của H.265, việc giải mã đạt được hiệu quả cao nhưng thời gian giải mã không tăng lên nhiều so với các chuẩn trước đây. Vì vậy, việc lựa chọn bộ mã hóa H.265 để triển khai trong hệ thống truyền dẫn video thời gian thực qua môi trường quang vô tuyến là phù hợp.

## CHƯƠNG 3. PHÂN TÍCH ĐÁNH GIÁ HIỆU NĂNG CỦA CHUẨN NÉN VIDEO HEVC TRÊN PHẦN MỀM HEVC

### 3.1. Giới thiệu phần mềm mô phỏng HM và các tham số mô phỏng.

Để đánh giá hiệu năng nén của chuẩn HEVC, luận văn sử dụng phần mềm mã nguồn mở HM (HEVC Test Model). Đây là phần mềm tham chiếu được triển khai tham chiếu tiêu chuẩn HEVC. Phần mềm được phát triển bởi Nhóm hợp tác chung về mã hóa video (JCT-VC) từ các chuyên gia ITU-T SG 16 và ISO / IEC SC29 WG11. Một trong những mục tiêu chính của phần mềm tham chiếu là cung cấp một cơ sở để tiến hành các mô phỏng để đánh giá hiệu năng của chuẩn nén HEVC. Phần mềm được tham khảo tại tài liệu [15]. Hình 3.1 là ví dụ về lệnh HEVCEncoder.exe được sử dụng để mã hóa chuỗi video.

```
D:\Nghienccu\Video coding Research\DVC_GOP\DVC testing>echo off
-----
DVC_HM simulation
-----
A subdirectory or file bin\ already exists.
A subdirectory or file logJM\ already exists.
A subdirectory or file logHM\ already exists.
A subdirectory or file logDVC\ already exists.
A subdirectory or file str\ already exists.
A subdirectory or file yuv\ already exists.
A subdirectory or file data\ already exists.
A subdirectory or file hmdata\ already exists.
A subdirectory or file jmdata\ already exists.
@call bin\HEVCEncoder.exe -c cfg\encoder_intra_main.cfg -i data\foreman_qcif.yuv -wdt 176 -hgt 144 --InputBitDepth=8
-fr 30 -f 300 --Level=2.1 -o hmdata\Key_foreman_qcif_40_hm.yuv -q 40
*****
** WARNING: --SEIDecodedPictureHash is now disabled by default. **
** Automatic verification of decoded pictures by a **
** decoder requires this option to be enabled. **
*****
```

**Hình 3.1: Thực hiện lệnh HEVC Encoder**

Để đánh giá hiệu năng của HEVC đối với các chuỗi video có đặc tính khác nhau, các tham số của môi trường mô phỏng được cho trong bảng 3.1 dưới đây:

**Bảng 3.1. Các tham số của môi trường mô phỏng**

Tham số	Giá trị
Số lượng chuỗi video	9 chuỗi
Độ phân giải	176 x 144
Tốc độ mã hóa	30 frame/s
Cấu hình máy tính	Windows, Core™ i7, 1.6GHz, 16GB RAM

### 3.2. Các chuỗi video dùng để đánh giá.

Trong luận văn này sử dụng 9 chuỗi video chuẩn QCIF (Quarter Common Intermediate File Format) với nội dung chuyển động khác nhau để so sánh hiệu năng

của các thuật toán trong các chế độ mã hóa khác nhau. Các chuỗi video chuẩn QCIF có độ phân giải thấp được ứng dụng chạy trên các thiết bị cấu hình thấp như mạng IoT.

Các chuỗi video này được phân loại thành 3 loại: Loại A, loại B, loại C với độ phức tạp chuyển động tăng dần. Các chuỗi video loại A có chuyển động thấp hay chậm. Các chuỗi loại B có chuyển động trung bình và chuỗi loại C có chuyển động phức tạp hay chuyển động nhanh. Loại A bao gồm các chuỗi: Akiyo, Hall và Mother-daughter. Các frame đầu tiên của chuỗi này được mô tả trong hình 3.2. Các frame đầu tiên của chuỗi loại B là Mobile, News, và Coastguard với chuyển động trung bình trình bày trong hình 3.3. Tương tự, hình 3.4 mô tả các frame đầu tiên của chuỗi thuộc loại C gồm Foreman, Carphone và Soccer với sự chuyển đổi đối tượng nhanh và tốc độ chuyển động cao.



Akiyo



Hall



Mother-daughter

**Hình 3.2: Các frame đầu tiên của chuỗi Akiyo, Claire, Mother-daughter**



Mobile



News



Coastguard

**Hình 3.3: Các frame đầu tiên của chuỗi News, Hall, Coastguard**



Foreman



Carphone



Soccer

**Hình 3.4: Các frame đầu tiên của chuỗi Foreman, Carphone, Soccer**

### 3.3. Các tham số đánh giá hiệu năng nén.

Để đánh giá hiệu năng của chuẩn nén HEVC đối với các chuỗi video có đặc tính khác nhau, hai tham số PSNR (Peak Signal-to-Noise Ratio) và số lượng bit dùng để mã hóa trong một giây (Bitrate) được sử dụng để so sánh hiệu năng nén. Tham số PSNR được sử dụng để đo mức độ giống nhau giữa hai chuỗi video trước khi nén và sau khi giải nén. Nói cách khác, PSNR phản ánh chất lượng của chuỗi video sau khi giải mã. Tham số này càng lớn đồng nghĩa với chất lượng chuỗi video sau khi giải nén càng giống với chất lượng chuỗi video gốc. Thông thường, chuỗi video có PSNR từ 30 dB trở lên sẽ được coi là có chất lượng tốt. PSNR được tính dựa trên công thức sau:

$$PSNR = 20 \cdot \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \quad (3.1)$$

Trong đó  $MAX_I$  là giá trị tối đa của điểm ảnh trên ảnh. Khi một điểm ảnh được biểu diễn bởi 8 bit thì giá trị của nó sẽ là 255. Tổng quát khi tín hiệu được biểu diễn bởi  $b$  bit trên một đơn vị lấy mẫu thì  $MAX_I$  bằng  $2^b - 1$

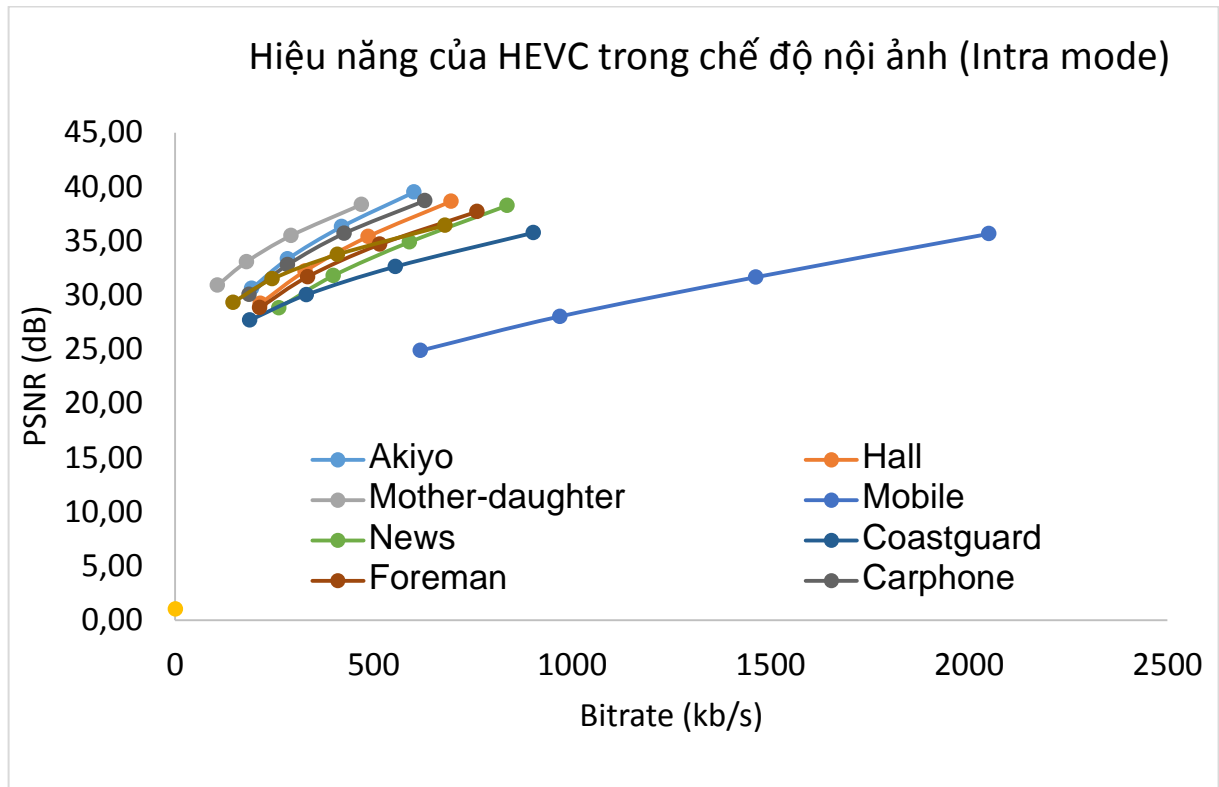
MSE (Mean Squared Error) là sai số toàn phương trung bình dùng cho ảnh hai chiều có kích thước là  $M \times N$ . Trong đó  $Org$  là ảnh gốc còn  $Rec$  là ảnh được khôi phục tương ứng. MSE được tính qua công thức:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (Org(i, j) - Rec(i, j))^2 \quad (3.2)$$

Bitrate là tham số phản ánh số lượng bit trung bình dùng để mã hóa các khung hình video trong một giây. Bitrate càng lớn nghĩa là số bit dùng để mã hóa video lớn. Nếu cùng một giá trị PSNR, chuỗi video nào có bitrate càng nhỏ nghĩa là hiệu quả nén chuỗi video đó càng cao.

### 3.4. Phân tích kết quả mô phỏng.

#### 3.4.1 Mã hóa video trong chế độ nội ảnh (Intra mode).



*Hình 3.5. So sánh hiệu năng nén của HEVC cho các chuỗi khác nhau*

**Bảng 3.2. So sánh chất lượng ảnh (PSNR-dB) sau giải nén của các chuỗi với các tham số lượng tử (QP) khác nhau**

	QP=28	QP=32	QP=36	QP=40
<b>Akiyo</b>	39,50	36,31	33,33	30,61
<b>Hall</b>	38,65	35,41	32,23	29,22
<b>Mother-daughter</b>	38,36	35,49	33,07	30,91
<b>Mobile</b>	35,66	31,66	28,03	24,88
<b>News</b>	38,26	34,90	31,79	28,82
<b>Coastguard</b>	35,75	32,63	30,04	27,69
<b>Foreman</b>	37,70	34,68	31,70	28,84
<b>Carphone</b>	38,72	35,70	32,79	30,07
<b>Soccer</b>	36,44	33,75	31,49	29,33

**Bảng 3.3. So sánh Bitrate (Kbps) của các chuỗi với các tham số lượng tử (QP) khác nhau**

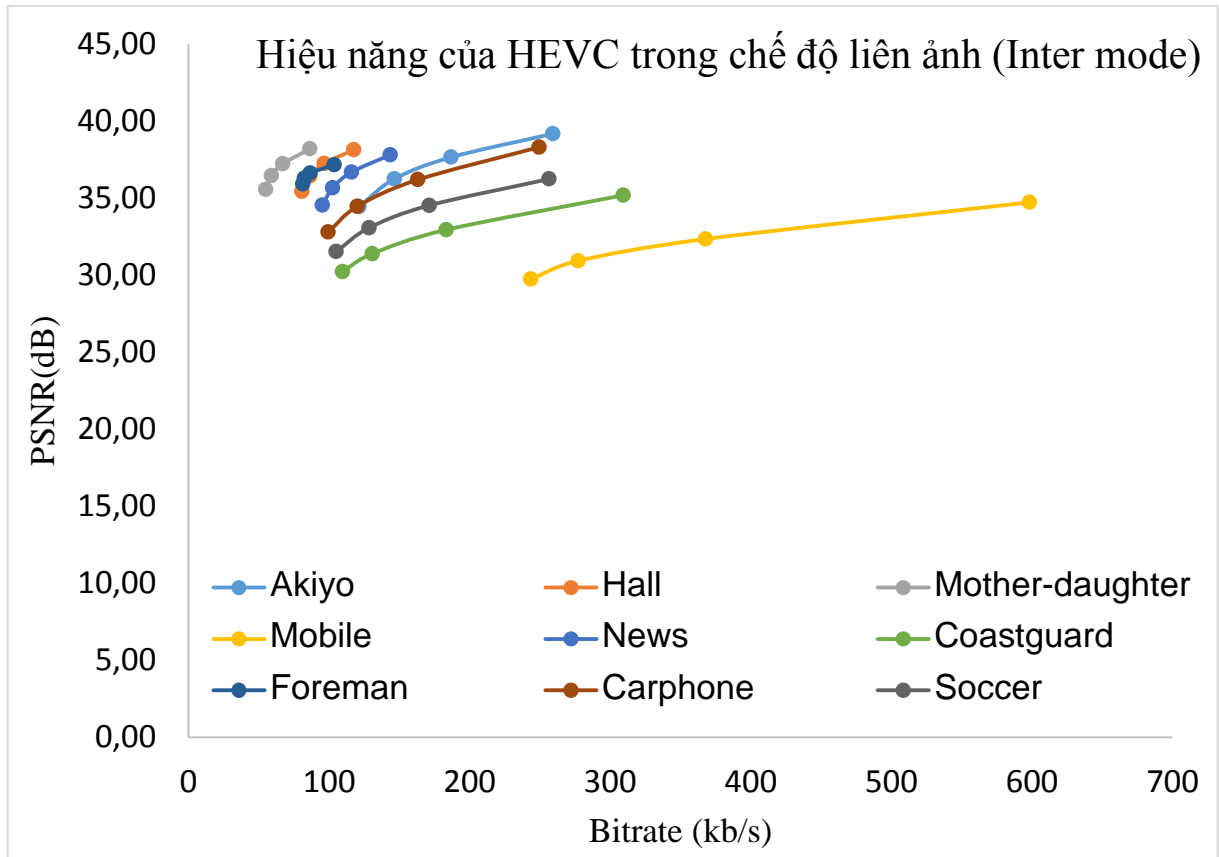
	QP=28	QP=32	QP=36	QP=40
<b>Akiyo</b>	601,77	419,36	283,3	192,72

<b>Hall</b>	695,09	486,5	327,44	214,55
<b>Mother-daughter</b>	469,45	291,81	179,67	106,68
<b>Mobile</b>	2050,54	1463,23	969,2	617,9
<b>News</b>	836,42	590,57	398,46	261,06
<b>Coastguard</b>	902,22	555,07	331,09	188,1
<b>Foreman</b>	760,5	515,8	333,8	213,2
<b>Carphone</b>	629,42	425,85	283,13	186,2
<b>Soccer</b>	679,98	409,08	244,58	145,88

Như kết quả đã chỉ ra trong hình 3.5 và các bảng 3.2, 3.3, ta thấy rằng trong chế độ nén nội ảnh các chuỗi video có chuyển động chậm và ít chuyển động (các chuỗi loại A) sẽ có chất lượng cao hơn trong khi cần ít bit hơn để mã hóa. Ngược lại, các chuỗi ở lớp B (có chuyển động phức tạp như Mobile) sẽ cần nhiều bit để mã hóa trong khi chất lượng hình ảnh đạt được không cao. Cụ thể, hình 3.2 cho thấy ứng với mỗi giá trị lượng tử QP, chuỗi Mobile có Bitrate lớn nhất trong khi PSNR đạt được là nhỏ nhất. Trong khi đó chuỗi Mother-daughter có Bitrate nhỏ nhất và PSNR đạt được chỉ sau chuỗi Akiyo. Như vậy, có thể thấy rằng, trong chế độ nội ảnh, các chuỗi có chuyển động chậm và ít chuyển động sẽ đạt được hiệu quả nén tốt nhất.

#### **3.4.2 Mã hóa video trong chế độ liên ảnh (Inter mode).**

Hình 3.6 và bảng 3.4, 3.5 chỉ ra hiệu năng nén của HEVC trong chế độ liên ảnh đối với các chuỗi video có đặc tính nén khác nhau. Bảng 3.5 cho thấy chuỗi Mobile vẫn là chuỗi cần nhiều bit để mã hóa nhất so với các chuỗi còn lại. Tuy nhiên, trong chế độ liên ảnh số lượng bit cần ít hơn nhiều so với chế độ nội ảnh. Bảng 3.4 cho thấy chất lượng trung bình của các video loại A vẫn cao hơn so với các video loại B & C. Với những video chuyển động nhanh như Mobile và Coastguard thì chất lượng thu được sẽ kém nhất. Lý do là vì đối với loại video chuyển động nhanh, ước lượng chuyển động và bù chuyển động trong quá trình mã hóa sẽ kém chính xác so với video chuyển động chậm. Do đó, chất lượng ảnh tại phía giải mã sẽ thấp.



Hình 3.6. So sánh hiệu năng nén của HEVC cho các chuỗi khác nhau

Bảng 3.4. So sánh chất lượng ảnh (PSNR-dB) sau giải nén của các chuỗi với các tham số lượng tử (QP) khác nhau

	QP=28	QP=32	QP=36	QP=40
Akiyo	<b>39,20</b>	<b>37,67</b>	36,26	34,46
Hall	38,15	37,27	36,45	35,45
Mother-daughter	38,21	37,24	<b>36,47</b>	35,57
Mobile	34,74	32,36	30,95	29,74
News	37,82	36,70	35,68	34,55
Coastguard	35,20	32,96	31,40	30,24
Foreman	37,18	36,64	36,29	<b>35,92</b>
Carphone	38,33	36,21	34,47	32,81
Soccer	36,25	34,54	33,08	31,54

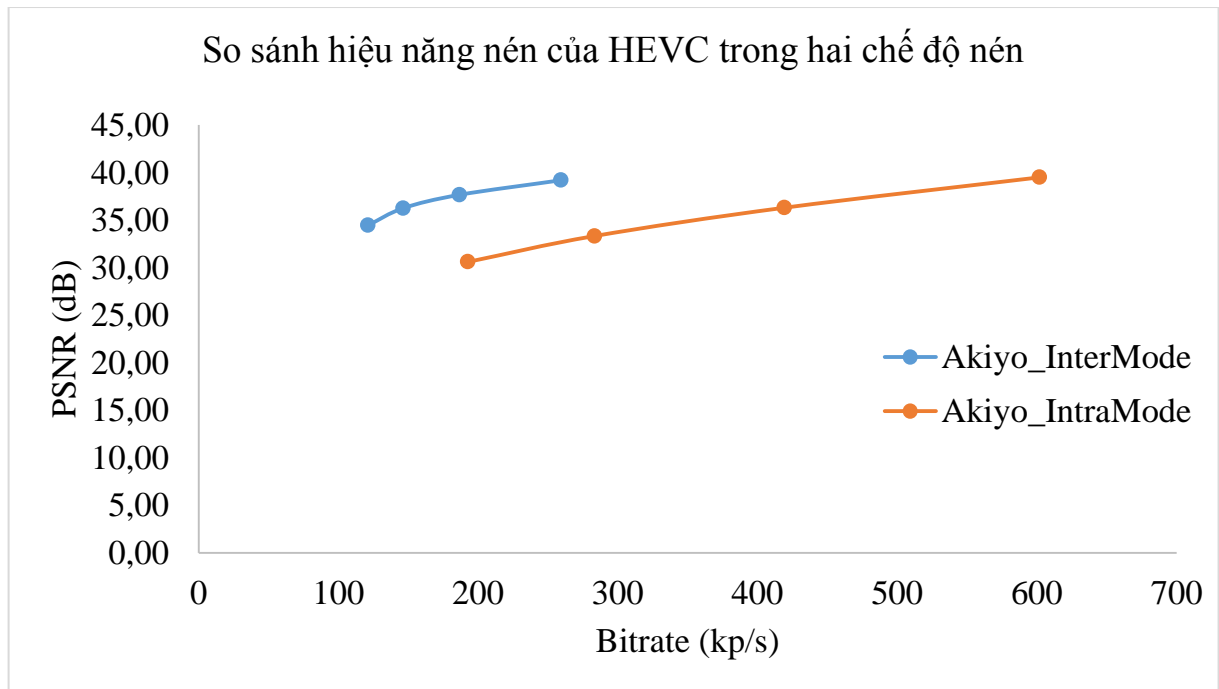
Bảng 3.5. So sánh Bitrate (Kbps) của các chuỗi với các tham số lượng tử (QP) khác nhau

	QP=28	QP=32	QP=36	QP=40
Akiyo	259,21	186,72	146,44	121,27
Hall	117,62	96,71	86,46	80,64



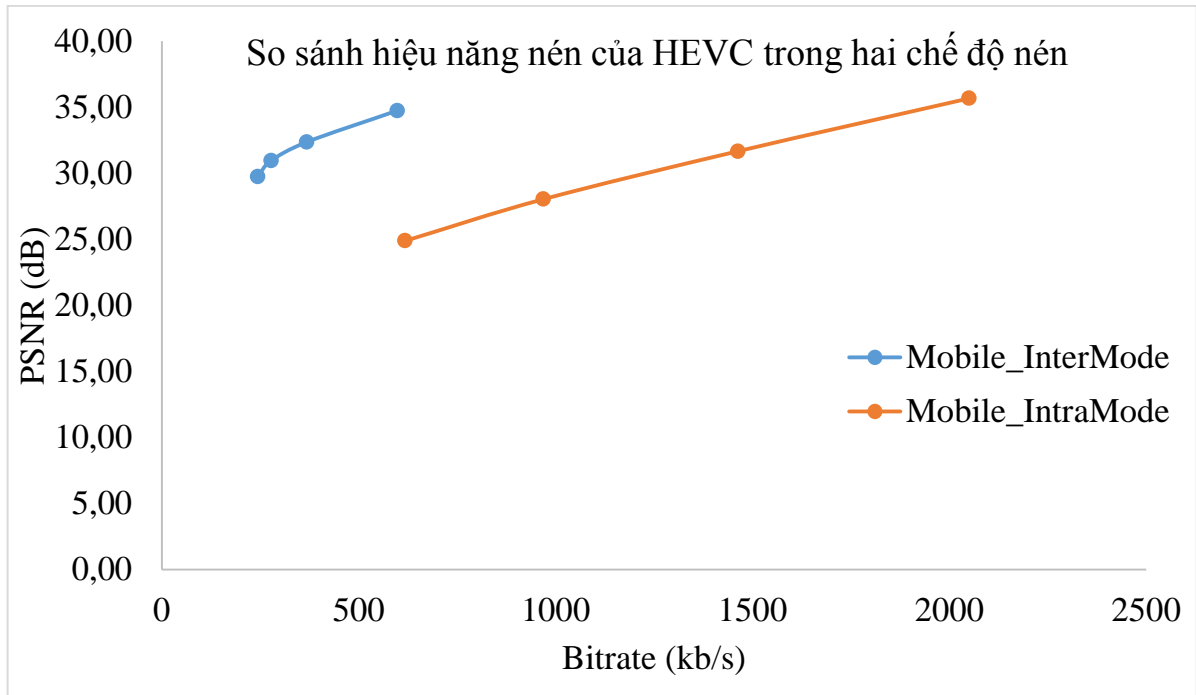
<b>Mother-daughter</b>	86,29	66,97	59,13	54,91
<b>Mobile</b>	598,23	367,88	277,28	243,45
<b>News</b>	143,45	115,9	102,53	95,15
<b>Coastguard</b>	309,35	183,32	130,7	109,55
<b>Foreman</b>	103,61	86,55	82,54	81,29
<b>Carphone</b>	249,49	163,1	120,04	99,39
<b>Soccer</b>	256,51	171,25	128,42	104,95

### 3.5. So sánh hai chế độ liên ảnh.

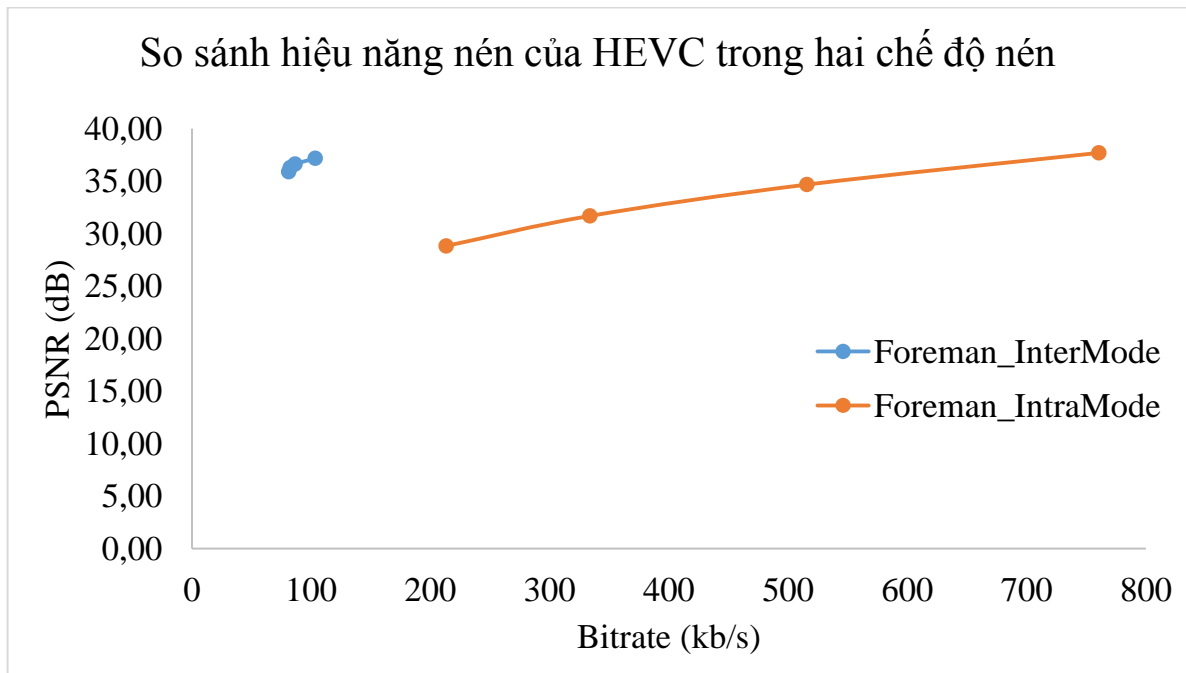


**Hình 3.7. So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Akiyo**





**Hình 3.8. So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Mobile**



**Hình 3.9. So sánh hiệu năng nén của HEVC trong hai chế độ nén của chuỗi Foreman**

Hình 3.7, 3.8 và 3.9 so sánh hiệu năng nén của HEVC trong hai chế độ liên ảnh và nội ảnh đối với 3 loại chuỗi: Ít chuyển động (Akiyo), chuyển động vừa (Mobile) và chuyển động nhanh (Foreman). Ta có thể thấy đối với chuỗi ít chuyển động, sự khác biệt giữa chế độ không lớn bằng hai loại chuỗi còn lại. Ngoài ra, trong chế độ nội ảnh

(Intra mode), sự khác biệt về giá trị lượng tử sẽ dẫn đến sự thay đổi rõ rệt về lượng bit cần để mã hóa và chất lượng ảnh tại phía giải mã.

### 3.6. Chế độ mã hóa Inter với kích thước chuỗi khác nhau

Để kiểm tra chế độ mã hóa với các kích thước chuỗi khác nhau, bốn chuỗi video được sử dụng để đánh giá bao gồm Coastguard, Suzie, Pamphlet và Harbor với các tham số được tóm tắt trong bảng 3.5. Khung hình đầu tiên của bốn chuỗi video này được hiển thị trong hình 3.5.

Test sequences	Spatial resolution	Number of frames	Quantization parameters
<b>Coastguard</b>	176x144	300	{26,30,34,38}
<b>Suzie</b>		150	{25,29,34,40}
<b>Pamphlet</b>		150	{25,29,34,40}
<b>Harbour</b>		150	{25,29,34,40}

Bảng 3. 6. Các tham số của chuỗi video thử nghiệm



Hình 3.10 Frame đầu tiên của các chuỗi video thử nghiệm

Để đánh giá thuật toán được đề xuất, chỉ số BD-PSNR (Bjntegaard-Delta PSNR) được sử dụng để so sánh, với BD-PSNR để cung cấp độ lợi tương đối giữa hai phương pháp, bằng cách đo sự khác biệt trung bình giữa hai đường cong đánh giá tỷ lệ sai lệch RD (RD – Rate-Distortion) với đường cong RD được chọn làm đường cong cơ sở. Nếu BD-PSNR là dương, nó có nghĩa là đường cong thứ hai tốt hơn đường cong cơ sở. Trong đánh giá này, các đường cong RD của GOP4 và phương pháp được đề xuất có tên là Adaptive GOP được so sánh với đường cong cơ sở GOP2.

Kết quả đánh giá hiệu năng về tỷ lệ sai lệch RD cho bốn chuỗi video thử nghiệm được trình bày trong bảng 3.7.

Theo kết quả biểu diễn trong bảng 3.7, các giá trị PSNR của phương pháp đề xuất tốt hơn các giá trị của GOP 4 và tương đương với các giá trị của GOP2. Giá trị bitrate của phương pháp được đề xuất cao hơn giá trị của GOP4 và thấp hơn giá trị của GOP2. Do đó, việc lựa chọn giữa GOP2 và GOP4 phụ thuộc vào sự cân bằng giữa PSNR và Bitrate. Kết quả cho thấy mức độ suy giảm chất lượng của video (về giá trị PSNR) trong phương pháp đề xuất là không đáng kể trong khi tiết kiệm được khá nhiều bitrate. Bảng 3.2 cho thấy khả năng tối ưu bitrate của phương pháp đề xuất là 3,37% và 9,62% so với GOP2 và GOP4, tương ứng.

Sequence	QP	GOP2		GOP4		Adaptive GOP	
		Bitrate	PSNR	Bitrate	PSNR	Bitrate	PSNR
	26	27760	38.18	28242	34.65	27735	38.14
Coastguard	30	17131	34.87	16140	32.48	17058	34.84
	34	9838	31.88	8228	30.36	9760	31.85
	38	5256	29.14	3781	28.23	5199	29.12
<b>Average</b>		<b>14996.25</b>	<b>33.52</b>	<b>14097.75</b>	<b>31.43</b>	<b>14938</b>	<b>33.49</b>
	26	18424	41.58	19719	41.26	18565	41.34
Suzie	30	10869	38.56	11172	38.23	10530	38.26
	34	5725	35.41	5588	35.15	5283	35.29
	38	2667	32.24	2353	32.04	2270	32.19
<b>Average</b>		<b>9421.25</b>	<b>36.95</b>	<b>9708</b>	<b>36.67</b>	<b>9162</b>	<b>36.77</b>
	26	23893.93	41.15	23128.28	41.35	22453.65	41.37
Pamphlet	30	15669.9	37.42	14900.7	37.51	14504.5	37.56
	34	9013.55	33.18	8567.73	33.24	8349.78	33.29
	38	3897.73	28.86	3667.88	28.91	3587.02	28.95
<b>Average</b>		<b>13118.78</b>	<b>35.15</b>	<b>12566.15</b>	<b>35.25</b>	<b>12223.74</b>	<b>35.29</b>
	26	45656.58	38.04	45680.28	37.62	45337.92	37.81
Harbour	30	29713.93	34.18	28617.86	33.73	28830.11	33.96
	34	16805.14	30.36	15471.99	30.03	15889.86	30.23
	38	7646.22	26.24	6768.94	26.09	7082.92	26.22
<b>Average</b>		<b>24955.47</b>	<b>32.2</b>	<b>24134.77</b>	<b>31.86</b>	<b>24285.2</b>	<b>32.06</b>

**Bảng 3.7 Hiệu năng của các chuỗi video thử nghiệm trong thuật toán đề xuất**

### 3.7. Kết luận chương

Chương 3 đã trình bày quá trình mô phỏng đo và đánh giá hiệu năng nén của chuẩn nén video HEVC đối với các chuỗi video có đặc tính khác nhau. Phần mềm mô phỏng được sử dụng là phần mềm HM của Nhóm hợp tác chung về mã hóa video (JCT-VC) từ các chuyên gia ITU-T SG 16 và ISO / IEC SC29 WG11. Các chuỗi video được thử nghiệm là các chuỗi video có các đặc tính thay đổi từ chuyển động chậm đến chuyển động nhanh, từ ít chuyển động đến nhiều chuyển động. Kết quả mô phỏng cho thấy chuẩn nén HEVC đạt được hiệu quả cao nhất đối với các chuỗi chuyển động chậm và ít chuyển động cả về chất lượng và hiệu năng nén.

## KẾT LUẬN

Luận văn đã tìm hiểu tổng quan về mã hóa và giải mã video các kỹ thuật giải mã video nói chung và các kỹ thuật tiên tiến được sử dụng trong chuẩn nén H.265. Các kỹ thuật này được đánh giá và so sánh với kỹ thuật trước đây như H.264 để thấy rõ các cải tiến của H.265. Cụ thể, các kỹ thuật áp dụng trong mỗi giai đoạn giải mã sẽ được so sánh và đánh giá ưu nhược điểm so với các kỹ thuật trong các chuẩn nén H.264 trước đây.

Để đánh giá hiệu năng nén của chuẩn HEVC, luận văn sử dụng phần mềm mã nguồn mở HM. Các nghiên cứu sử dụng phần mềm mô phỏng trong chương 3 đã chỉ ra rằng, chất lượng video tại phía thu trong chuẩn nén H.265 có thể được cải thiện so với các chuẩn nén trước đây. Ngoài ra hiệu quả nén của H.265 cũng được tăng lên đáng kể. Tuy nhiên, cấu trúc của bộ giải mã H.265 phức tạp hơn so với H.264. Điều này cũng dẫn đến thời gian giải mã của H.265 cũng tăng lên so với H.264. Tuy nhiên, trong một số chế độ mã hóa của H.265, việc giải mã đạt được hiệu quả cao nhưng thời gian giải mã không tăng lên nhiều so với các chuẩn trước đây.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

- [1] Vũ Hữu Tiến, Hà Đình Dũng, “Bài giảng: Xử lý và Truyền thông Đa phương tiện”, Học viện Công nghệ Bưu chính Viễn thông, 2016.
- [2] Đỗ Hoàng Tiến, Dương Thanh Phương, “Kỹ thuật truyền hình”, NXB Khoa học và Kỹ thuật 2004.
- [3] Luận văn Vũ Thị Quỳnh Anh, “Triển mô hình tap âm tương quan nhằm nâng cao hiệu năng mã hóa của kỹ thuật mã hóa Video phân tán DVC” Trang 13-23, 2019

### Tiếng Anh

- [4] J. R. Jain, A. K. Jam, "Displacement measurement and its application in interframe image coding", IEEE Trans. Commun., vol. COM-29, pp. 1799-1808, Dec. 1981.
- [5] Mohammed Ghanbari, “ Video coding: an introduction to standard codecs”, The Institution of Electrical Engineers, 1999.
- [6] R. Srinivasan, K.R. Rao, “Predoctove coding based on efficient motion estimation”, IEEE Trans. Commun., pp. 888-896, 2012.
- [7] R. Li, B. Zeng, M. L. Liou, "A new three-step search algorithm for block motionestimation", IEEE Trans. Circuits Syst. Video Technol., vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [8] JVT Editors (T. Wiegand, G. Sullivan, A. Luthra), Draft ITUT Recommendation and final draft international standard of joint video specification (ITU-T Rec.H.264 |ISO/IEC 14496- 10 AVC), JVT-G050r1, Geneva, May 2003.
- [9] L. Natrio, C. Brites, J. Ascenso, and F. Pereira, Side information extrapolation for low-delay pixel-domain distributed video coding, International Workshop on Very Low Bitrate Video, 2005.
- [10] A. Aaron and B. Girod, , Wyner-Ziv video coding with low-encoder complexity, Picture Coding Symposium, 2004.
- [11] [5] D. Slepian and J. Wolf, Noiseless Coding of Correlated Information Sources, IEEE Transactions on Information Theory, vol. 19, no. 4, pp.471-480, July 1973.
- [12] A. Wyner and J. Ziv, The Rate-Distortion Function for Source Coding with Side Information at the Decoder, IEEE Transactions on Information Theory, vol. 22, no. 1, pp.1-10, January 1976.
- [13] R. Puri and K. Ramchandran, PRISM: A new robust video coding architecture based on distributed compression principles, 40th Allerton Conf. Communication, Control and Computing,, Allerton, IL, USA, 2002.

- [14] A. Aaron, R. Thang, and B. Girod (2002) “Wyner-Ziv Coding of Motion Video”, in Proc. Asilomar Conference on Signals and Systems, Pacific Grove, CA, USA, November 2002.

**Trang Web**

- [15] <https://hevc.hhi.fraunhofer.de/>