ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ-VIỄN THÔNG
*****

BÁO CÁO ĐỒ ÁN
AUDIO - VIDEO
TỔNG QUAN VỀ CHUẨN HEVC/H.265

Giảng viên: ThS.Bùi An Đông

Nguyễn Hoài Trọng 19200536

Thành phố Hồ Chí Minh 2022

TỔNG QUAN VỀ CHUẨN NÉN HEVC

1. Giới thiệu chuẩn HEVC

Dữ liệu dùng cho các nền tảng lưu trữ các video, livestream ngày càng tăng lên do chất lượng, độ phân giải video càng tăng và chiếm phần lớn lưu lượng dữ liệu trên Internet. Do đó các chuẩn nén video lần lượt ra đời như H.264/AVC, H.265/HEVC, VP9, AV1,…

Mã hóa video hiệu quả cao (High Efficiency Video Coding - HEVC), còn được gọi là H.265 và MPEG-H Phần 2, là một tiêu chuẩn nén video, được thiết kế như một sự kế thừa cho AVC.

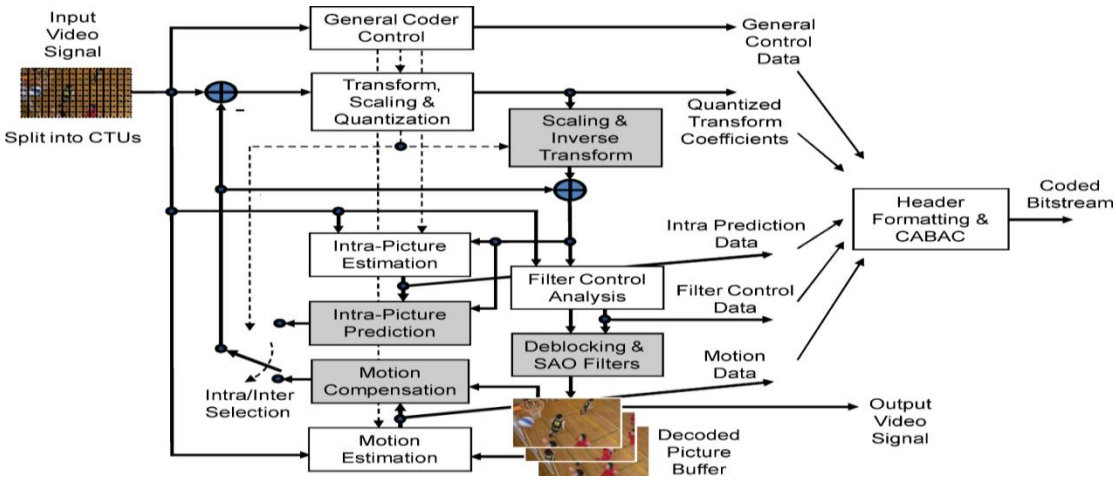
Chuẩn nén video HEVC đã cải thiện hiệu suất so với tiêu chuẩn trước đó. Nó có thể giảm tổng chi phí phân phối và lưu trữ video trong khi duy trì hoặc tăng chất lượng video.

So với AVC, HEVC cung cấp khả năng nén dữ liệu tốt hơn từ 25% đến 50% ở cùng mức chất lượng video hoặc chất lượng video được cải thiện đáng kể ở cùng tốc độ bit. Nó hỗ trợ độ phân giải lên tới 8192 × 4320, bao gồm 8K UHD.

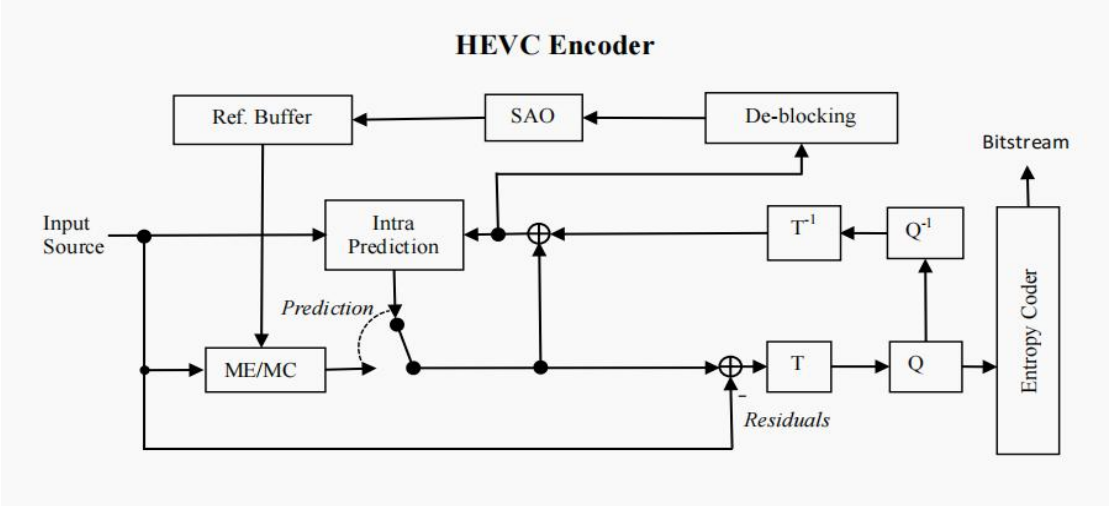
H.265 được phát triển bởi ITU-T (International Telecommunication Union) và ISO/IEC (International Organisation for Standardisation / International Electrotechnical Commission), được công bố lần đầu vào năm 2013. Nó là một tập hợp bao gồm các quy tắc và thuật toán dùng để mã hóa và giải mã các tín hiệu video.

2. Sơ đồ khối và chức năng

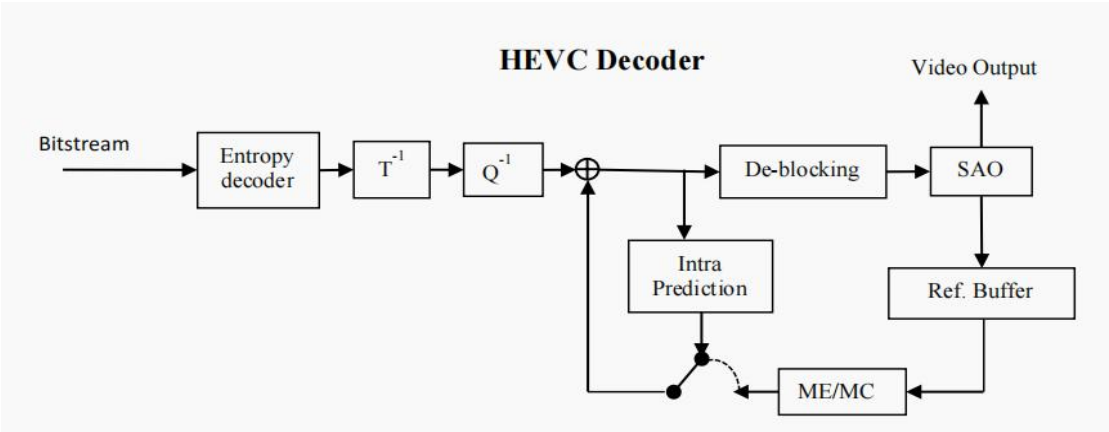
Quá trình nén ảnh được tóm tắt: Một bức ảnh ngõ vào gồm có 3 thành phần màu cơ bản R, G, B sẽ được biến đổi thành Y (độ sáng), Cb, Cr (độ màu). Sau đó, ảnh sẽ được tách thành các khối Macro Block (MB) có kích thuớc là NxN với n= 4, 8 hay 16 tùy thuộc vào độ phức tạp của bức ảnh. Ảnh đầu tiên hoặc điểm truy nhập ngẫu nhiên thì được dự đoán trong khối (Intra), các ảnh còn lại của dãy được dự đoán liên khối (Inter) và bù chuyển động từ các ảnh trước đó. Dữ liệu từ các MB cần được mã hóa sẽ đưa đến cả bộ trừ và bộ dự đoán chuyển động. Bộ dự đoán chuyển động sẽ so sánh các MB mới dựa vào với các MB tham khảo đã được đưa vào trước đó. Bộ dự đoán chuyển động sẽ tính toán vector chuyển động, vector này đặc trưng cho sự dịch chuyển theo cả 2 chiều ngang và thẳng đứng của MB mới cần được mã hóa so với frame tham khảo. Bộ dự đoán chuyển động cũng đồng thời gửi các MB tham khảo tới bộ trừ với MB mới cần được mã hóa để tạo ra các sai số tiên đoán đặc trưng cho sự sai khác giữa MB dự đoán và MB cần mã hóa. Sự sai khác này sẽ được biến đổi DCT nguyên để tạo ra tập hệ số biến đổi. Sau đó, dữ liệu được đưa qua bộ lượng tử hóa để làm giảm số lượng bít cần truyền. Đến đây, các hệ số lượng tử được chia làm 2 hướng, một hướng sắp xếp lại và đưa vào mã hóa Entropy, dữ liệu tiếp tục làm giảm đi một cách đáng kể, hướng còn lại đưa qua bộ giải lượng tử và biến đổi ngược để tạo ra khối sai số. Khối sai số được cộng với tín hiệu dự đoán, lọc tách khối tạo thành ảnh cấu trúc lại và được lưu trữ nhằm mục đích ước lượng và dự đoán chuyển động. Dữ liệu tại đầu ra bộ mã hóa Entropy sẽ kết hợp với vector chuyển động rồi truyền ra ngoài kênh truyền dưới dạng dòng bít nén và gửi tới bộ giải mã.



Hình 1: Sơ đồ khối Encoder và Decoder HEVC (với khối màu xám là Decoder)



Hình 2: Sơ đồ khối Encoder HEVC



Hình 3: Sơ đồ khối Decoder HEVC

(T - Transform; T^{-1} - Transform đảo; Q - Lượng tử hóa; Q^{-1} - Lượng tử hóa đảo ME - Ước lượng chuyển động; MC - Bù chuyển động; SAO - Sample adaptive offset)

2.1 Lấy mẫu ảnh

Để thể hiện tín hiệu video màu, HEVC thường sử dụng một không gian màu YCbCr với lấy mẫu 4:2:0. Thành phần Y còn được gọi là độ sáng và đại diện cho độ sáng. Hai thành phần màu Cb và Cr đại diện cho mức độ mà màu sắc lệch từ màu xám sang màu xanh và màu đỏ, tương ứng.

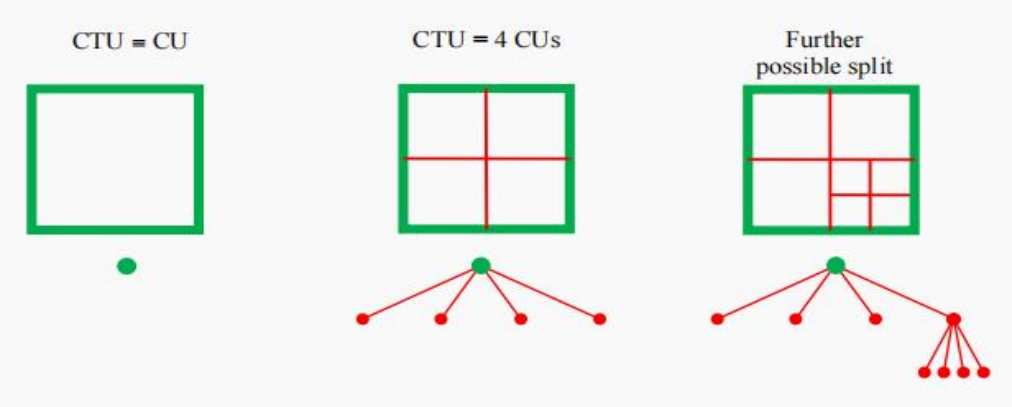
Bởi vì hệ thống thị giác của con người là nhiều hơn nhạy cảm với độ sáng hơn sắc độ, cấu trúc lấy mẫu 4:2:0 thường được sử dụng, trong đó mỗi thành phần sắc độ có một phần tư số lượng mẫu của thành phần độ sáng.

Các hình ảnh video thường được lấy mẫu dần dần với kích thước hình chữ nhật W×H, trong đó W là chiều rộng và H là chiều cao của hình ảnh tính theo mẫu độ sáng. Mỗi mảng thành phần màu, với tỷ lệ lấy mẫu 4:2:0 được W/2×H/2.

2.2 Chia ảnh thành Code Tree Units (CTUs)

Một hình ảnh được phân chia thành các Code Tree Unit (CTU), mà mỗi cái chứa CTB (Coding Tree Block) Luma CTB (CTB độ sáng) và Chroma CTB (CTB sắc độ).

Ở định dạng 4:2:0, mỗi thành phần sắc độ liên quan đến một CTU bao gồm N/2 × N/2 mẫu, N tối đa cho phép là 64. Ngược lại với các cấu trúc macroblock được sử dụng trong các tiêu chuẩn trước đó, chẳng hạn như H.264/AVC, CTU trong HEVC không chỉ lớn hơn mà còn cũng có các tùy chọn khác nhau để phân chia thành các khối nhỏ hơn.



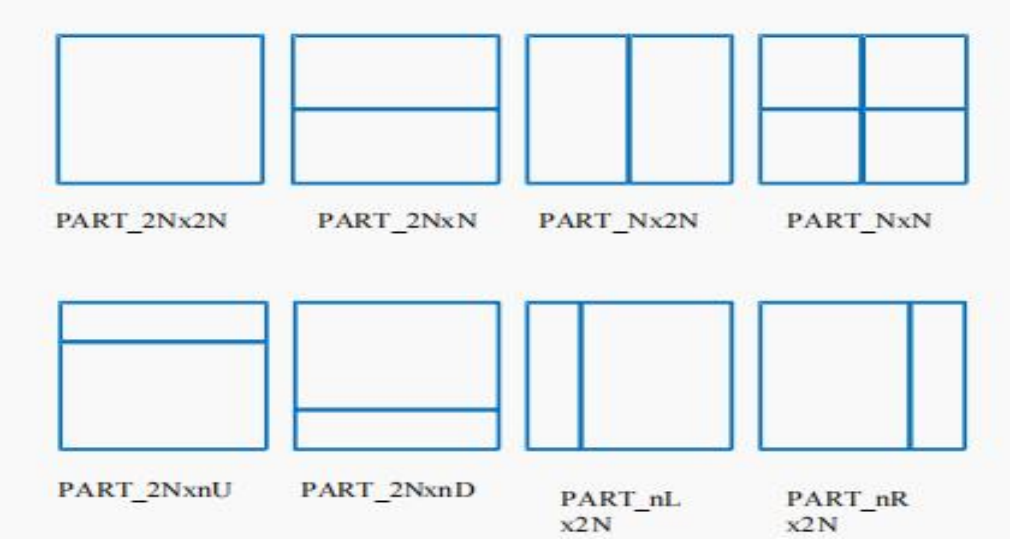
Hình 4: CTU được chia thành cấu trúc Quadtree

2.3 Prediction units and prediction blocks (PBs)

Tùy thuộc vào quyết định loại dự đoán cơ bản, các CB độ sáng và sắc độ có thể được chia nhỏ hơn nữa về kích thước và được dự đoán từ các khối dự đoán độ sáng và sắc độ (PB - Prediction Blocks). HEVC hỗ trợ các kích thước PB có thể thay đổi từ 64×64 xuống mẫu 4×4.

Đơn vị dự đoán (PU - Prediction units) là đơn vị cơ bản cho các quy trình dự đoán. Như minh họa trong (hình 5), một CU có thể được chia thành 1, 2 hoặc 4 PU. Phân vùng PU có thể không đối xứng hoặc đối xứng, nhằm mục đích khớp với các ranh giới hoặc cạnh khác nhau của các đối tượng trong ảnh.

Một CU được mã hóa liên kết có thể được chia thành tất cả tám chế độ phân vùng có thể. Ngược lại, một CU được mã hóa nội bộ chỉ có thể được phân chia đối xứng. CU với PART_2Nx2N chỉ có một PU trong khi CU với PART_NxN có 4 PU. Các chế độ dự đoán còn lại có 2 PU.

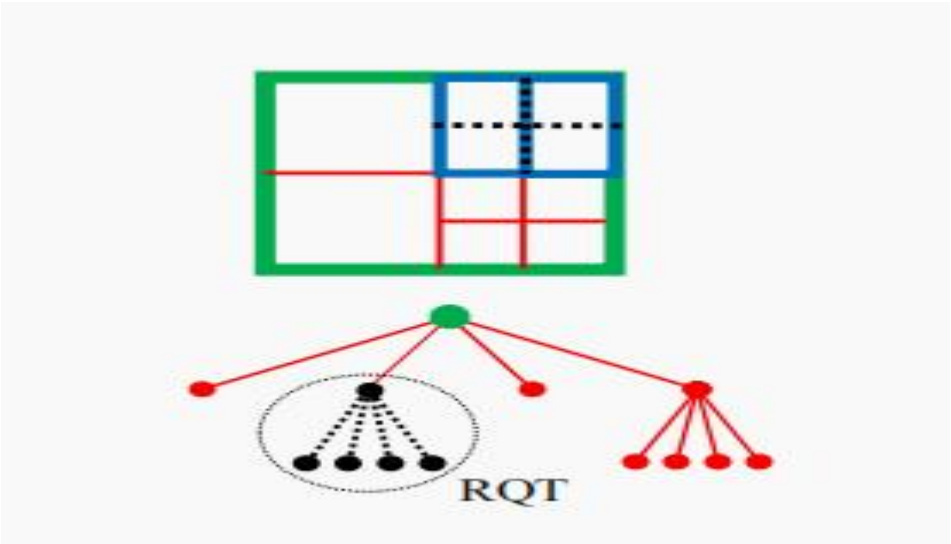


Hình 5: Partitioning cho khối PU

2.4 Transform unit structure

Transform unit (TU) là đơn vị cơ bản được sử dụng để biến đổi và lượng tử hóa quy trình. Mỗi CU (Coding Unit) chứa một hoặc nhiều TU. Trong khi phân tách PU xác định dự đoán trong các phần khác nhau của CU, phân tách TU xác định các phép biến đổi trong một CU. Mỗi TU bao gồm các khối vuông có kích thước từ 32 × 32 đến 4 × 4 pixel.

Việc chia CU thành các TU được mô tả bằng cấu trúc cây tứ giác. Trong trường hợp các cây tứ giác lồng vào nhau, cấu trúc phân chia là được gọi là Residual QuadTree (RQT). Một số hạn chế cụ thể được áp dụng tùy thuộc vào chế độ dự đoán, phân tách PU, thành phần nhất định (độ sáng hoặc sắc độ) và kích thước khối. Ví dụ thể hiện vị trí RQT liên quan đến sự phân chia CTU/PU được thể hiện trong (Hình 6). Các phép biến đổi và lượng tử hóa thực tế được thực hiện trên mỗi TU.



Hình 6: CTU chia thành PUs với cấu trúc cây

2.5 Dự đoán Intra

HEVC có 35 chế độ dự đoán intra cho thành phần độ sáng (Luma), trong đó bao gồm các chế độ dự đoán DC, Planar và 33 angular modes. Các góc dự đoán là bội số của 7,5 độ. Độ chính xác của dự đoán hướng là độ chính xác 1/32 góc độ. HEVC sử dụng gấp đôi số lượng mẫu tham chiếu bên trái so với AVC.

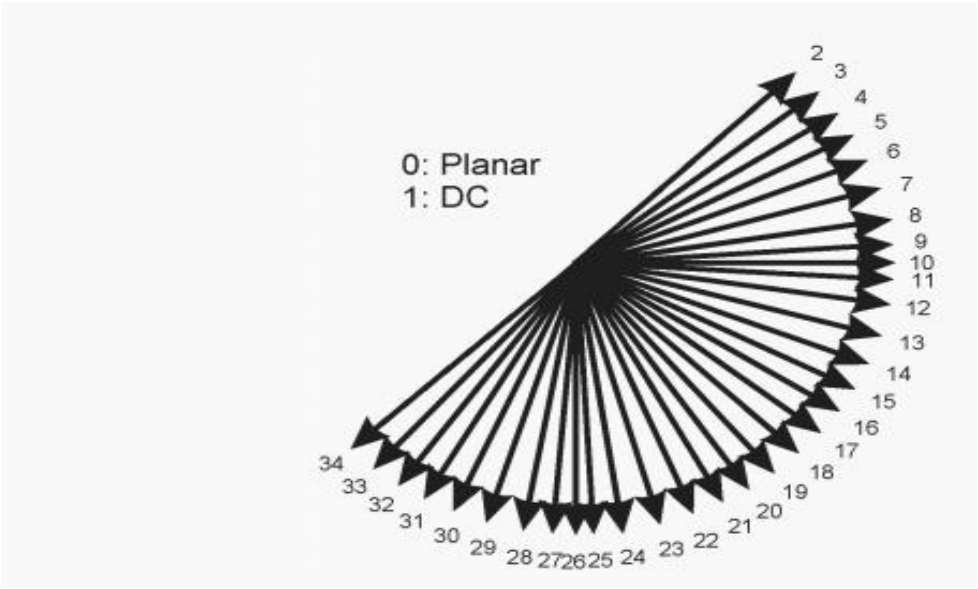
Intra−Angular Prediction: So với tám hướng dự đoán của H.264/MPEG-4 AVC, HEVC hỗ trợ tổng cộng 33 hướng dự đoán, ký hiệu là Intra−Angular[k], trong đó k là số chế độ từ 2 đến 34. Các góc được thiết kế có chủ ý để cung cấp vùng phủ sóng dày đặc hơn cho các góc gần ngang và gần thẳng đứng và phạm vi phủ sóng thô hơn cho các góc gần chéo để phản ánh quan sát mức độ phổ biến thống kê của các góc và hiệu quả của quá trình xử lý dự đoán tín hiệu.

Khi sử dụng chế độ Intra−Angular, mỗi TB được dự đoán theo hướng từ các mẫu lân cận không gian được xây dựng lại (nhưng chưa được lọc bởi các bộ lọc trong vòng lặp) trước đó được sử dụng cho dự đoán này.

Đối với một TB có kích thước N×N, tổng cộng của 4N+1 mẫu lân cận không gian có thể được sử dụng cho dự đoán, như trong (Hình 7). Khi có sẵn từ trước hoạt động giải mã, các mẫu từ các TB phía dưới bên trái có thể được sử dụng để dự đoán trong HEVC ngoài các mẫu từ TB tại bên trái, bên trên và bên phải của TB hiện tại. Quá trình dự đoán của các chế độ Intra−Angular có thể liên quan đến các mẫu ngoại suy từ tham chiếu dự kiến vị trí mẫu theo một hướng nhất định. Để loại bỏ nhu cầu chuyển đổi từng mẫu giữa tham chiếu bộ đệm hàng và cột, cho Intra−Angular[k] với k trong phạm vi từ 2–17, các mẫu nằm ở hàng trên là chiếu dưới dạng các mẫu bổ sung nằm ở cột bên trái; và với k trong khoảng 18–34, các mẫu nằm ở cột bên trái được chiếu như các mẫu nằm ở hàng trên.

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

Quá trình dự đoán của các chế độ Intra−Angular 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 để hỗ trợ kích thước khối 4×4, 8×8 và 16×16. Tính nhất quán của thiết kế này đặc biệt mong muốn vì HEVC hỗ trợ nhiều loại hơn kích thước TB và số lượng dự đoán tăng đáng kể hướng so với H.264/MPEG-4 AVC.



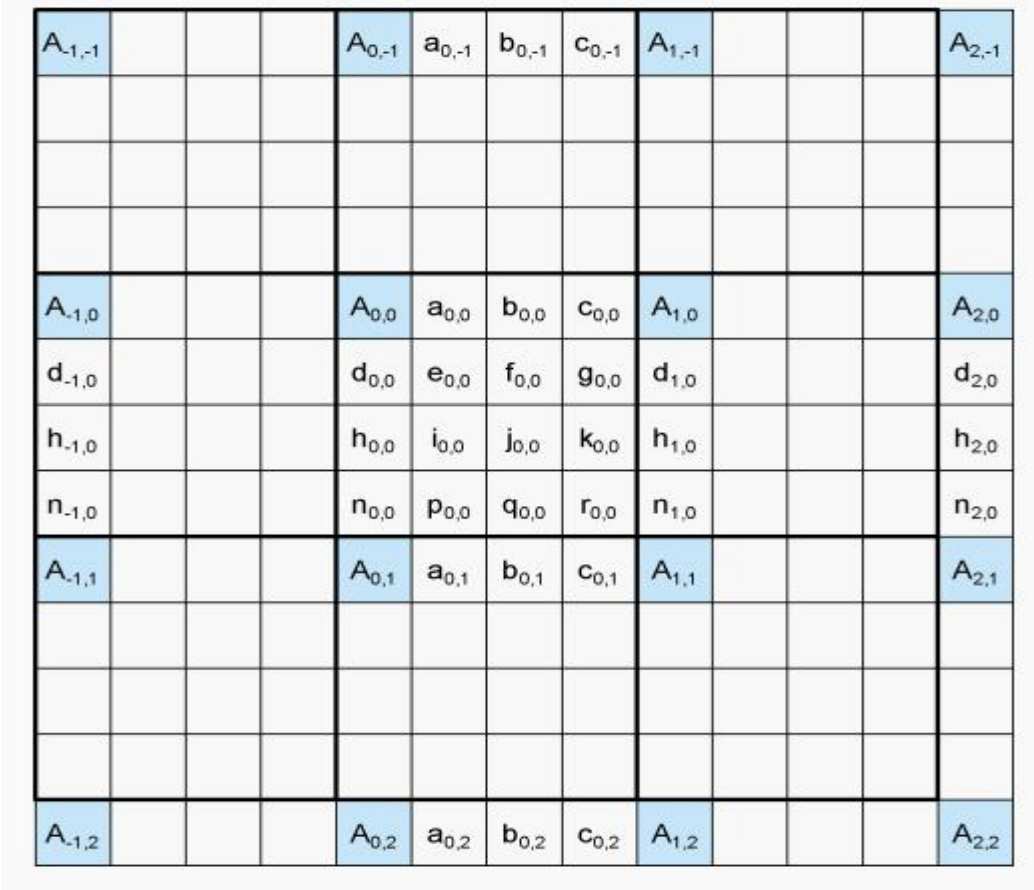
Hình 7: Modes và hướng cho dự đoán intrapicture

Intra−Planar and Intra−DC Prediction: Ngoài ra đến dự đoán Intra−Angular nhắm mục tiêu các khu vực có cường độ mạnh các cạnh định hướng, HEVC hỗ trợ hai dự đoán thay thế các phương thức, Intra−Planar và Intra−DC. Trong khi dự đoán Intra−DC sử dụng giá trị trung bình của các mẫu tham chiếu cho dự đoán, giá trị trung bình của hai dự đoán tuyến tính sử dụng bốn các mẫu tham chiếu góc được sử dụng trong dự đoán Intra−Planar để ngăn sự gián đoạn dọc theo ranh giới 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 kích thước độ sáng PB là 16×16 và dự đoán mặt phẳng của nó hoạt động hơi khác so với dự đoán phẳng trong HEVC Chế độ Intra−Planar cũng sử dụng bộ lọc làm mịn khi kích thước khối lớn hơn hoặc bằng 8 × 8 và làm mịn không được sử dụng (hoặc hữu ích) cho trường hợp Intra−DC.

2.6 Dự đoán Inter

Fractional Sample Interpolation: Các mẫu PB đối với CB được dự đoán trong ảnh được lấy từ CB của vùng khối tương ứng trong ảnh tham

chiếu được xác định bởi một chỉ số hình ảnh tham chiếu, mà là ở một vị trí thay thế bởi các thành phần ngang và dọc của vectơ chuyển động.



Hình 8: Integer and fractional sample positions for luma interpolation

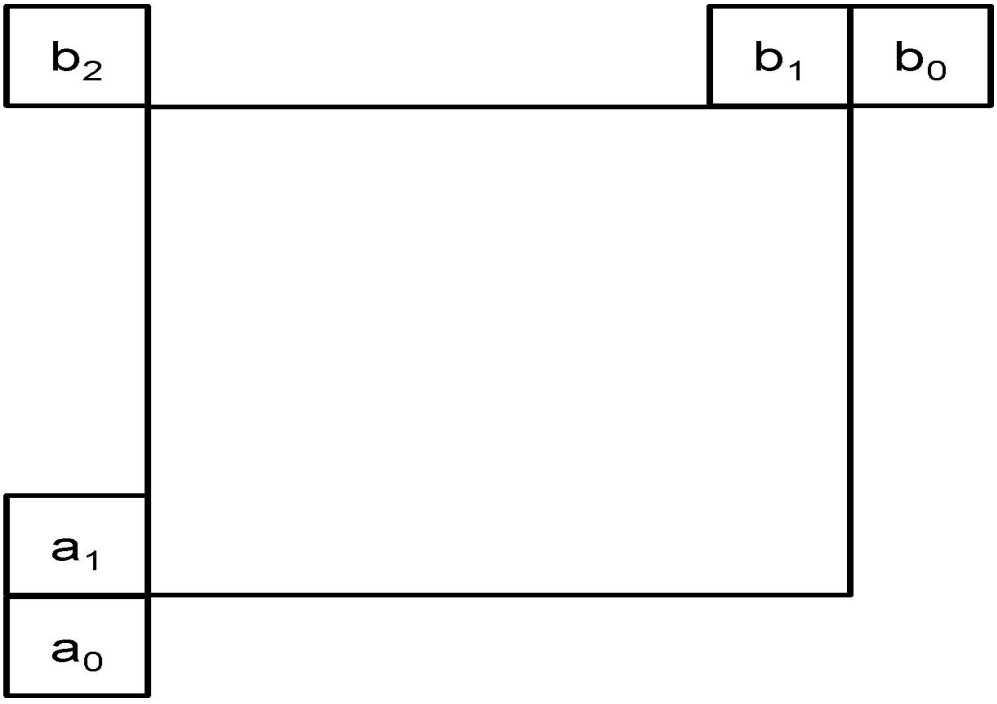
Merge Mode: Thông tin chuyển động thường bao gồm các giá trị dịch chuyển vectơ chuyển động ngang và dọc, một hoặc hai chỉ số hình ảnh tham chiếu, và, trong trường hợp các vùng dự đoán trong lát cắt B, nhận dạng tham chiếu nào danh sách ảnh được liên kết với mỗi chỉ mục. HEVC bao gồm một chế độ hợp nhất để lấy thông tin chuyển động từ không gian hoặc các khối lân cận tạm thời. Nó được ký hiệu là chế độ hợp nhất vì nó tạo thành một vùng hợp nhất chia sẻ tất cả thông tin chuyển động.

Có hai sự khác biệt quan trọng so với H.264/AVC. Đầu tiên, nó truyền thông tin chỉ mục đến chọn một trong số một số ứng cử viên có sẵn, theo

cách đôi khi được gọi là sơ đồ cạnh tranh vectơ chuyển động. Nó cũng xác định rõ ràng danh sách ảnh tham chiếu và chỉ mục ảnh tham chiếu, trong khi chế độ trực tiếp giả định rằng chúng có một số giá trị được xác định trước.

Motion Vector Prediction for Nonmerge Mode: Khi một CB dự đoán xen kẽ không được mã hóa theo cách bỏ qua hoặc các chế độ hợp nhất, vectơ chuyển động được mã hóa khác nhau bằng cách sử dụng một công cụ dự đoán véc tơ chuyển động. Tương tự như chế độ hợp nhất, HEVC cho phép bộ mã hóa chọn bộ dự đoán vectơ chuyển động trong số nhiều ứng cử viên dự đoán. sự khác biệt giữa bộ dự đoán và vectơ chuyển động thực tế và chỉ số của ứng cử viên được truyền đến bộ giải mã.

Chỉ có hai ứng cử viên chuyển động không gian được chọn theo đến sự sẵn có của năm ứng cử viên trong Hình 9. Đầu tiên ứng cử viên chuyển động không gian được chọn từ tập hợp các vị trí bên trái {a0, a1} và vị trí thứ hai từ tập hợp các vị trí trên {b0, b1, b2} theo tính khả dụng của chúng, trong khi vẫn giữ nguyên thứ tự tìm kiếm như được chỉ ra trong hai bộ.



Hình 9: Positions of spatial candidates of motion information

Khi chỉ số tham chiếu của PU lân cận không bằng với PU hiện tại, một phiên bản thu nhỏ của vectơ chuyển động được sử dụng. Vectơ chuyển động lân cận là được chia tỷ lệ theo khoảng cách thời gian giữa hiện tại hình ảnh và hình ảnh tham chiếu được biểu thị bằng chỉ số tham chiếu của PU lân cận và PU hiện tại, tương ứng. Khi hai ứng cử viên không gian có cùng các thành phần vectơ chuyển động, một ứng cử viên không gian dư thừa là loại trừ.

Khi số lượng bộ dự đoán vectơ chuyển động (Motion vector) không bằng hai và việc sử dụng dự đoán vectơ chuyển động tạm thời không rõ ràng bị vô hiệu hóa, ứng cử viên dự đoán vectơ chuyển động tạm thời được đưa vào. Điều này có nghĩa là ứng cử viên tạm thời hoàn toàn không được sử dụng khi hai ứng cử viên không gian có sẵn. Cuối cùng, một chuyển động bằng không vectơ được bao gồm lặp đi lặp lại cho đến khi số lượng vectơ chuyển động các ứng cử viên dự đoán bằng hai, điều này đảm bảo rằng số lượng bộ dự đoán vectơ chuyển động là hai. Vì vậy,

chỉ một mã hóa cờ là cần thiết để xác định dự đoán vectơ chuyển động nào là được sử dụng trong trường hợp chế độ không hợp nhất.

2.7 Transform, Scaling, and Quantization

HEVC sử dụng mã hóa biến đổi của lỗi dự đoán còn lại theo cách tương tự như trong các tiêu chuẩn trước đó. Khối residual được phân vùng thành nhiều TB vuông. Kích thước khối biến đổi được hỗ trợ là 4×4, 8×8, 16×16 và 32×32.

Core Transform: Các phép biến đổi hai chiều được tính toán bằng cách áp dụng các phép biến đổi 1-D theo chiều ngang và chiều dọc. Các yếu tố của ma trận biến đổi cốt lõi là xuất phát bằng cách xấp xỉ các hàm cơ sở DCT được chia tỷ lệ, dưới các cân nhắc như giới hạn phạm vi động cần thiết để tính toán biến đổi và tối đa hóa độ chính xác và gần với tính trực giao khi các mục ma trận là cụ thể như giá trị số nguyên.

$$H = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & 90 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89 \\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 & 25 & 70 & 90 & 80 & 43 & -9 & -57 & -87 \\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 & 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 & -43 & -90 & -57 & 25 & 87 & 70 & -9 & -80 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75 \\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 & 57 & 80 & -25 & -90 & -9 & 87 & 43 & -70 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 & -70 & -43 & 87 & 9 & -90 & 25 & 80 & -57 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50 \\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 & 80 & -9 & -70 & 87 & -25 & -57 & 90 & -43 \\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 & 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 & -87 & 57 & -9 & -43 & 80 & -90 & 70 & -25 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 75 & -50 & 18 \\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 & 90 & -87 & 80 & -70 & 57 & -43 & 25 & -9
\end{bmatrix}.$$

Hình 10: Example for matrix length-16 transform

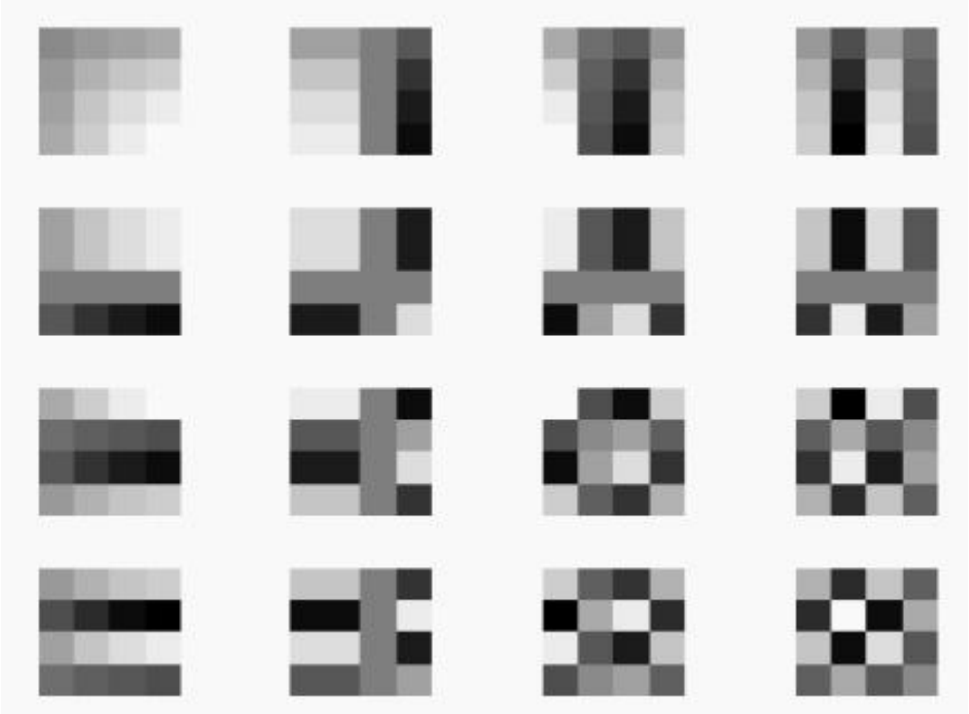
Do kích thước tăng lên của các biến đổi được hỗ trợ, giới hạn phạm vi động của các kết quả trung gian từ giai đoạn đầu tiên của quá trình chuyển đổi là khá quan trọng. HEVC chèn rõ ràng thao tác dịch chuyển 7-b sang phải và thao tác cắt 16-b sau giai đoạn biến đổi nghịch đảo 1-D

đầu tiên của biến đổi (giai đoạn biến đổi nghịch đảo dọc) để đảm bảo rằng tất cả các giá trị trung gian có thể được lưu trữ trong bộ nhớ 16-b (đối với video 8-b giải mã).

Alternative 4 × 4 Transform: Đối với khối biến đổi kích thước 4×4, một phép biến đổi số nguyên thay thế bắt nguồn từ một DST được áp dụng cho các khối dư độ sáng cho ảnh chụp nội suy chế độ dự đoán, với ma trận biến đổi các chức năng cơ bản của DST phù hợp hơn với thống kê thuộc tính mà biên độ còn lại có xu hướng tăng khi khoảng cách từ các mẫu ranh giới được sử dụng cho đoán trở nên lớn hơn. Về độ phức tạp, 4 × 4. Biến đổi kiểu DST không đòi hỏi nhiều tính toán hơn so với biến đổi kiểu 4×4 DCT và nó cung cấp giảm khoảng 1% tốc độ bit trong tính năng tiên đoán trong ảnh mã hóa. Việc sử dụng loại biến đổi DST bị hạn chế đối với chỉ các khối biến đổi độ sáng 4×4, vì đối với các trường hợp khác, cải thiện hiệu quả mã hóa bổ sung để bao gồm loại biến đổi bổ sung đã được tìm thấy là cận biên.

$$H = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}.$$

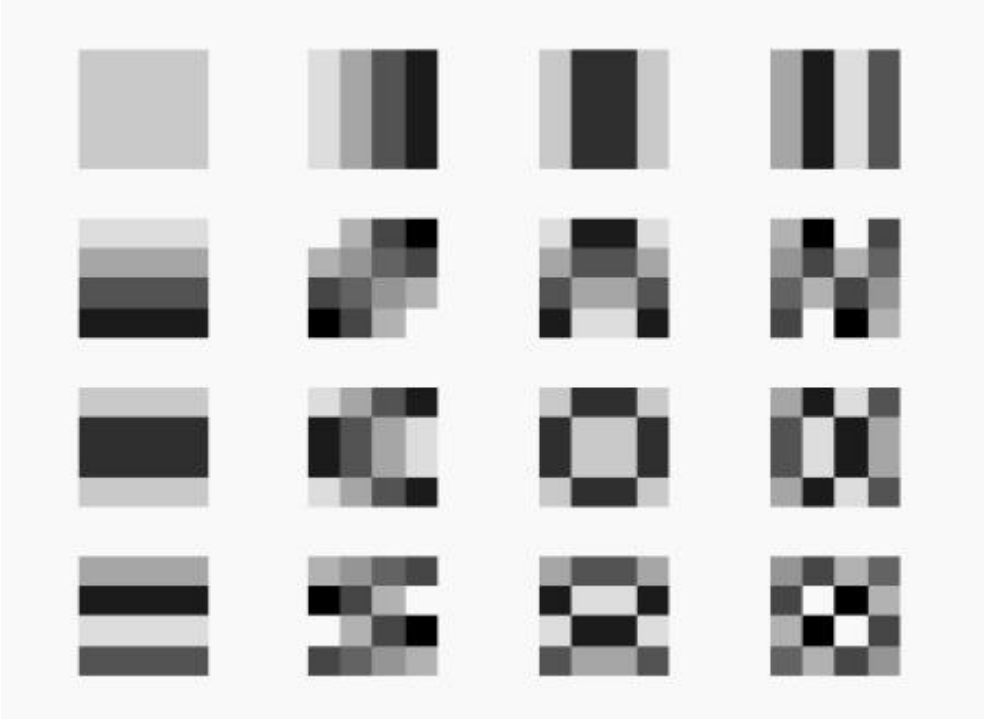
Hình 11: The transform matrix of 4-point DST



Hình 12: Basis patterns for 4 × 4 DST

$$\begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}$$

Hình 13: The transform matrix of 4-point DCT



Hình 14: Basis patterns for 4 × 4 DCT

Scaling and Quantization: Vì các hàng của ma trận dạng chuyển đổi là các giá trị gần đúng của các giá trị đồng nhất các hàm cơ bản được chia tỷ lệ của DCT trực giao, hoạt động tính tỷ lệ trước được kết hợp trong quá trình khử lượng tử của H.264/MPEG-4 AVC không cần thiết trong HEVC. Sự tránh né này tỷ lệ chức năng cơ sở cụ thể theo tần số rất hữu ích trong việc giảm kích thước bộ nhớ trung gian, đặc biệt khi xem xét rằng kích thước của biến đổi có thể lớn tới 32 × 32.

Đối với lượng tử hóa, HEVC về cơ bản sử dụng cùng một URQ lược đồ được điều khiển bởi một tham số lượng tử hóa (QP - Quantization Parameter) như trong H.264/MPEG-4 AVC. Phạm vi của các giá trị QP được xác định từ 0 đến 51 và tăng 6 lần lượng tử hóa kích thước bước sao cho ánh xạ các giá trị QP thành kích thước bước là xấp xỉ logarit. Ma trận tỷ lệ lượng tử hóa là cũng được hỗ trợ. Để giảm bộ nhớ cần thiết để lưu trữ tần số cụ thể giá trị tỷ lệ, chỉ ma trận lượng tử hóa có kích thước 4 × 4 và 8×8 được sử dụng. Đối với các phép biến đổi lớn

hơn 16 × 16 và 32×32, ma trận tỷ lệ 8×8 được gửi và được áp dụng bởi chia sẻ các giá trị trong các nhóm hệ số 2×2 và 4×4 trong các không gian con tần số, ngoại trừ các giá trị tại DC (tần số không) các vị trí, trong đó các giá trị riêng biệt được gửi và áp dụng.

2.8 Entropy Coding

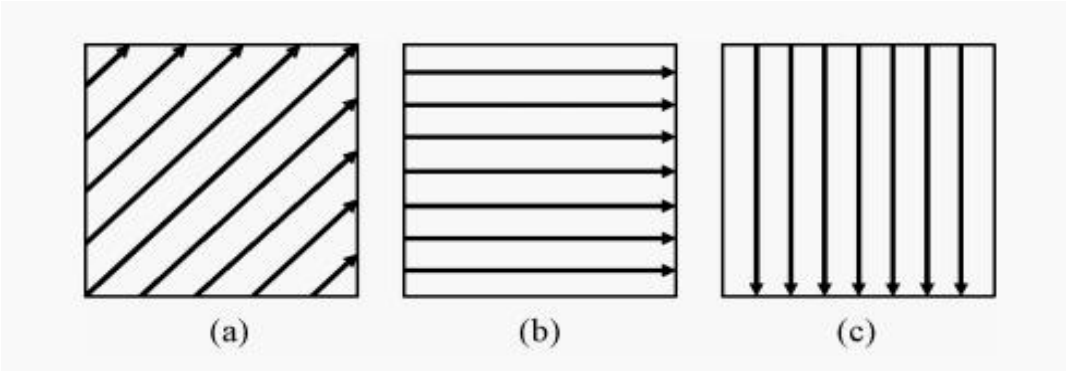
HEVC chỉ xác định một phương pháp mã hóa entropy, CABAC thay vì hai như trong H.264/MPEG-4 AVC. Thuật toán cốt lõi của CABAC không thay đổi và một số khía cạnh về cách nó được sử dụng trong thiết kế HEVC:

Context Modeling: Việc lựa chọn mô hình bối cảnh phù hợp được biết đến là yếu tố chính để cải thiện hiệu quả của mã hóa CABAC. Trong HEVC, độ sâu phân tách của cây mã hóa hoặc cây biến đổi được khai thác để lấy các chỉ số mô hình ngữ cảnh của các phần tử cú pháp khác nhau bên cạnh các phần tử lân cận không gian được sử dụng trong H.264/AVC.

Ví dụ: cờ bỏ qua phần tử cú pháp chỉ định liệu CB có được mã hóa là hình ảnh tương tác bị bỏ qua dự đoán hay không và cờ phần tử cú pháp split−coding−unit chỉ định liệu CB có được phân chia thêm hay không được mã hóa bằng cách sử dụng các mô hình ngữ cảnh dựa trên thông tin lân cận về mặt không gian. Phần tử cú pháp split−transform−flag xác định liệu TB có được phân chia thêm hay không và ba phần tử cú pháp chỉ định các hệ số biến đổi khác không cho từng thành phần màu, cbf−luma, cbf−cb và cbf−cr, được mã hóa dựa trên độ sâu phân chia của cây biến đổi.

Adaptive Coefficient Scanning: Quá trình quét hệ số được thực hiện trong các khối con 4×4 cho tất cả các kích thước TB (nghĩa là chỉ sử

dụng một vùng hệ số cho kích thước 4×4 TB và sử dụng nhiều vùng hệ số 4×4 trong các khối biến đổi lớn hơn). Ba phương pháp quét hệ số, quét chéo lên trên bên phải, quét ngang và quét dọc như trong Hình 15, được chọn hoàn toàn để mã hóa các hệ số biến đổi có kích thước 4×4 và 8×8 TB trong các vùng dự đoán trong ảnh. Việc lựa chọn thứ tự quét hệ số phụ thuộc vào hướng của dự đoán trong ảnh. Quét dọc được sử dụng khi hướng dự đoán gần với phương ngang và quét ngang được sử dụng khi hướng dự đoán gần với phương thẳng đứng. Đối với các hướng dự đoán khác, quét theo đường chéo lên bên phải được sử dụng.



Hình 15: Các kiểu scanning trong HEVC
(a - Diagonal up-right scan; b - Horizontal scan; c - Vertical scan)

Coefficient Coding: Tương tự như H.264/MPEG-4 AVC, HEVC truyền vị trí của hệ số biến đổi khác 0 cuối cùng, significance map, bit dấu và mức cho các hệ số biến đổi. Tuy nhiên, nhiều thay đổi khác nhau cho từng bộ phận đã được thực hiện, đặc biệt là để xử lý tốt hơn các tăng đáng kể kích thước TB.

Đầu tiên, các vị trí tọa độ tần số ngang và dọc của hệ số khác 0 cuối cùng được mã hóa cho TB trước khi gửi bản đồ ý nghĩa của các khối con 4x4 cho biết các hệ số biến đổi khác có giá trị khác 0, thay vì gửi một

loạt các hệ số cuối cùng. các cờ nhận dạng được xen kẽ với significance map như được thực hiện trong H.264/MPEG-4 AVC.

Một phương pháp được gọi là ẩn dữ liệu dấu hiệu được sử dụng để cải thiện khả năng nén hơn nữa. Các bit dấu được mã hóa theo điều kiện dựa trên số lượng và vị trí của các hệ số được mã hóa. Khi ẩn dữ liệu ký hiệu được sử dụng và có ít nhất hai hệ số khác không trong khối con 4×4 và sự khác biệt giữa vị trí quét của các hệ số khác không đầu tiên và cuối cùng là lớn hơn 3, bit dấu của hệ số khác 0 đầu tiên được lấy từ tính chẵn lẻ của tổng các biên độ hệ số.

Mặt khác, bit dấu được mã hóa bình thường. Về phía bộ mã hóa, điều này có thể được thực hiện bằng cách chọn một hệ số có biên độ gần với ranh giới của khoảng lượng tử hóa để buộc phải sử dụng khoảng lượng tử hóa liền kề trong trường hợp nếu không thì chẵn lẻ sẽ không chỉ ra dấu hiệu chính xác của hệ số đầu tiên. Điều này cho phép bit dấu được mã hóa với chi phí thấp hơn (theo thuật ngữ tỷ lệ biến dạng) so với khi nó được mã hóa riêng, bằng cách cho phép bộ mã hóa tự do lựa chọn biên độ hệ số biến đổi nào có thể được thay đổi với chi phí tỷ lệ biến dạng thấp nhất.

2.9 Lọc In-loop

Trong HEVC, hai bước xử lý, cụ thể là gỡ lỗi bộ lọc (DBF - DeBlocking Filter) theo sau là bộ lọc SAO (Sample Adapt Offset), được áp dụng cho các mẫu được tái tạo trước khi ghi chúng vào bộ giải mã bộ đệm hình ảnh trong vòng giải mã. DBF được dự định để giảm các tạo tác chặn do mã hóa dựa trên khối.

DBF tương tự như DBF của H.264/MPEG-4 AVC tiêu chuẩn, trong khi SAO mới được giới thiệu trong HEVC. Trong khi DBF chỉ được áp

dụng cho các mẫu nằm ở khối ranh giới, bộ lọc SAO được áp dụng thích ứng cho tất cả các mẫu đáp ứng các điều kiện nhất định, ví dụ, dựa trên độ dốc. Với sự phát triển của HEVC, nó cũng đã được coi là vận hành bước xử lý thứ ba được gọi là bộ lọc vòng lặp thích ứng (ALF - Adapt Loop Filter) sau bộ lọc SAO; tuy nhiên, tính năng ALF không đưa vào thiết kế cuối cùng.

2.10 Lọc De-blocking

Bộ lọc khử chặn trong HEVC tương tự như bộ lọc được sử dụng bởi H.264/AVC nhưng được đơn giản hóa và cải thiện để hỗ trợ tốt hơn (đặc biệt là xử lý song song kết cấu). HEVC giới hạn bộ lọc khử chặn ở các cạnh nằm trên các khối 8 × 8 trong khi H.264/AVC áp dụng lọc cho khối 4 × 4. Bộ lọc khử khối sử dụng các khối 8 × 8 vì nó không gây ra sự xuống cấp đáng kể. Một thay đổi khác là bộ lọc khử chặn trước tiên áp dụng lọc ngang cho các cạnh dọc. Mẫu được lọc sau đó được sử dụng để lọc các cạnh ngang. Điều này hỗ trợ nhiều triển khai song song.

2.11 Sample adaptive offset (SAO)

Độ lệch thích ứng mẫu (SAO) là một quá trình sửa đổi các mẫu bằng cách thêm các giá trị độ lệch vào các giá trị mẫu nhất định để giảm biến dạng. Nó được áp dụng cho các pixel được tạo lại sau bộ lọc khử chặn. SAO có thể được thực hiện dễ dàng vì có một bảng tra cứu cho phép xác định trước giá trị offset trước khi tiến hành bất kỳ công việc nào. Bảng tra cứu bao gồm sáu loại theo phân loại pixel được xây dựng lại. Sau đó, thêm Độ lệch dải (BO - Band Offset) hoặc Độ lệch cạnh (EO - Edge Offset) vào pixel tùy thuộc vào cường độ pixel hoặc thuộc tính cạnh.

Trong bù dải, giá trị bù được chọn theo giá trị mẫu. Toàn bộ phạm vi biên độ mẫu được chia thành 32 dải bằng nhau (mỗi dải có độ lệch riêng)

sau đó được chia thành hai nhóm, một dải trung tâm bao gồm 16 dải và một nhóm khác chứa các dải còn lại. Bộ mã hóa chọn một dải nhóm để áp dụng SAO.

EO sử dụng bốn mẫu pixel ba chiều một chiều để phân loại pixel theo thông tin hướng cạnh. Phân loại từng pixel bằng cách so sánh với hai pixel lân cận của nó. Sau đó, thêm phần bù của từng danh mục để bù giá trị pixel.

3. Thực hiện DCT ảnh với MATLAB

3.1 Công thức và code cho DCT 2D

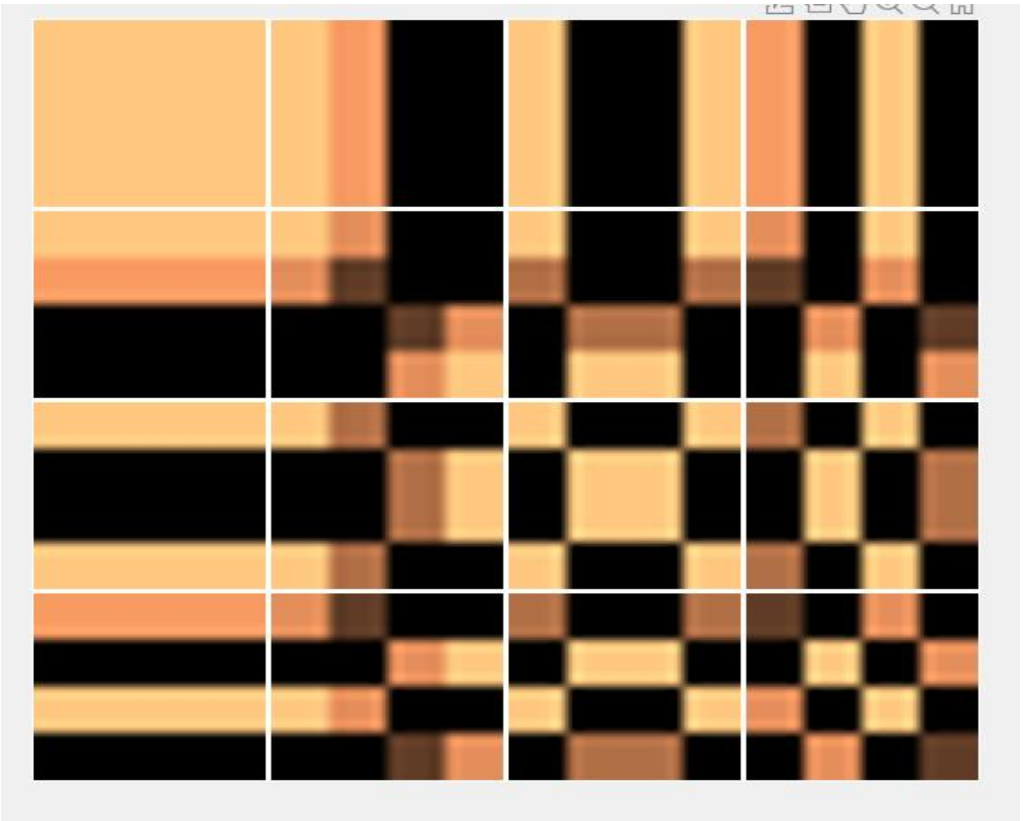
$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

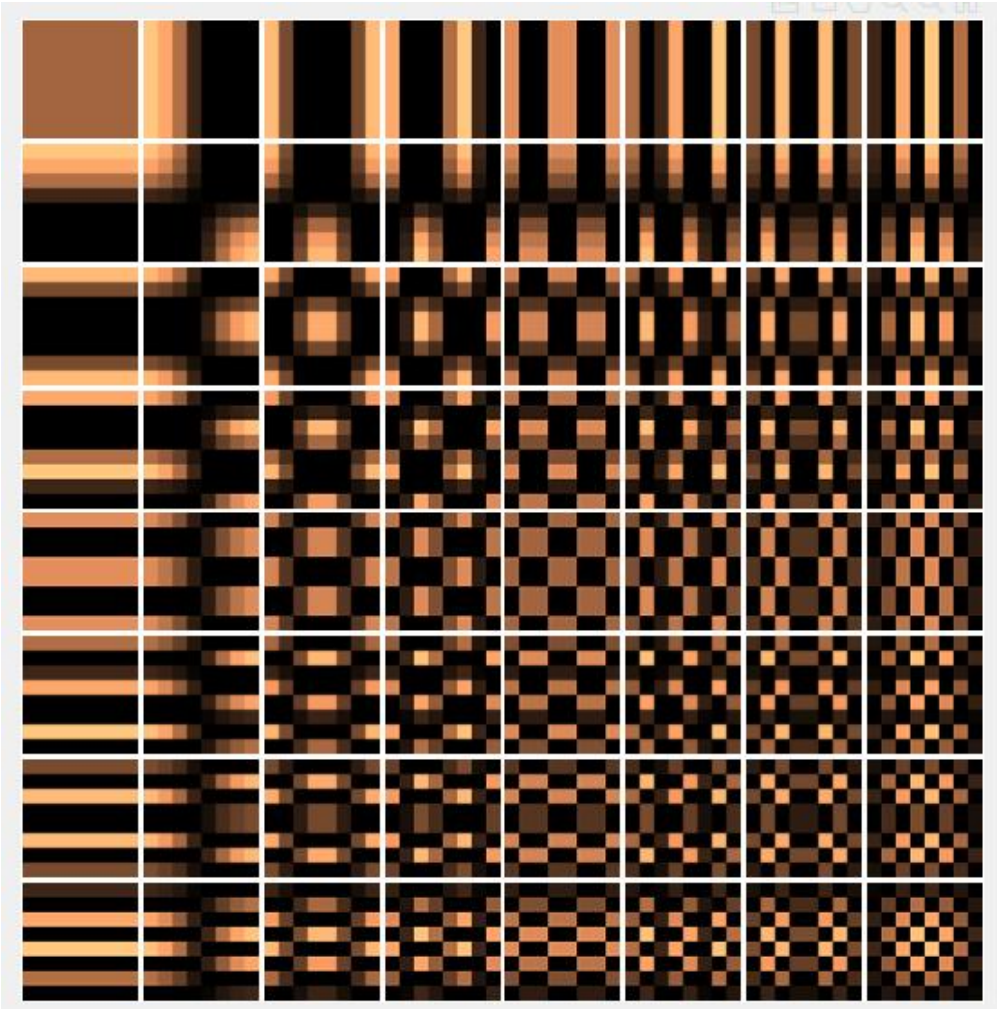
```
M=4; N=4;
B=zeros([4,4]);
for P=1:M
    p=P-1;
    for Q=1:N
        q=Q-1;
        if (p)==0 && (q)==0
            ap=1/sqrt(M);
            aq=1/sqrt(N);
        else
            ap=sqrt(2/M);
            aq=sqrt(2/N);
        end

        for x=1:M
            for y=1:N
                B(x,y)=ap*aq*cos((pi*(p)*(2*(x-
1)+1))/(2*M))*cos((pi*(q)*(2*(y-1)+1))/(2*N));
            end
        end
```

```
        basisimg = ind2rgb(im2uint8(B),copper);
        filename = strcat(num2str(p), num2str(q),
'.png');

imwrite(imresize(basisimg,[16,16],'nearest'),filename);
    end


end

path='C:\Users\USER\Documents\MATLAB';
imds = imageDatastore(fullfile(path,'*.png'));
montage(imds,'BackgroundColor','white','BorderSize',[2,2]
);
```

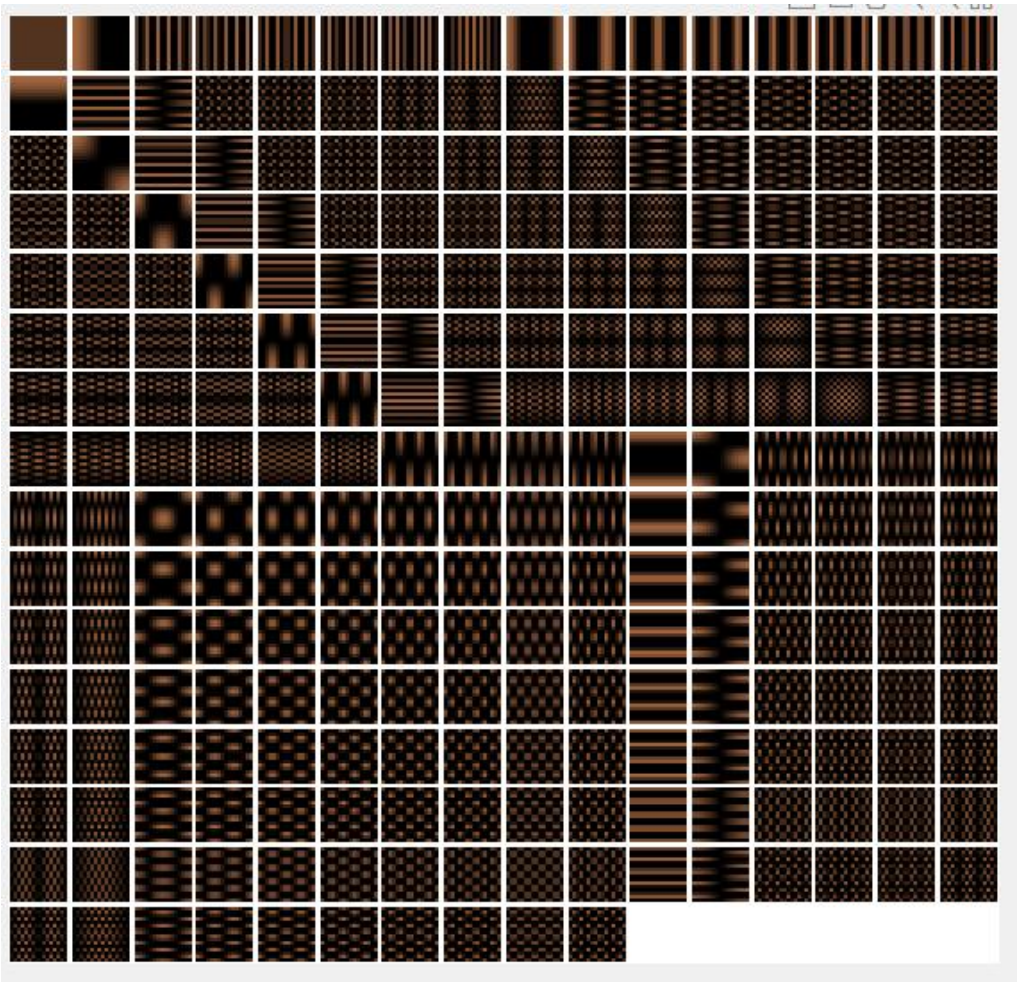
3.2 Kết quả DCT 2D



Hình 3.1: DCT chuyển đổi ảnh thành các block 4x4



Hình 3.2: DCT chuyển đổi ảnh thành các block 8x8



Hình 3.3: DCT chuyển đổi ảnh thành các block 16x16

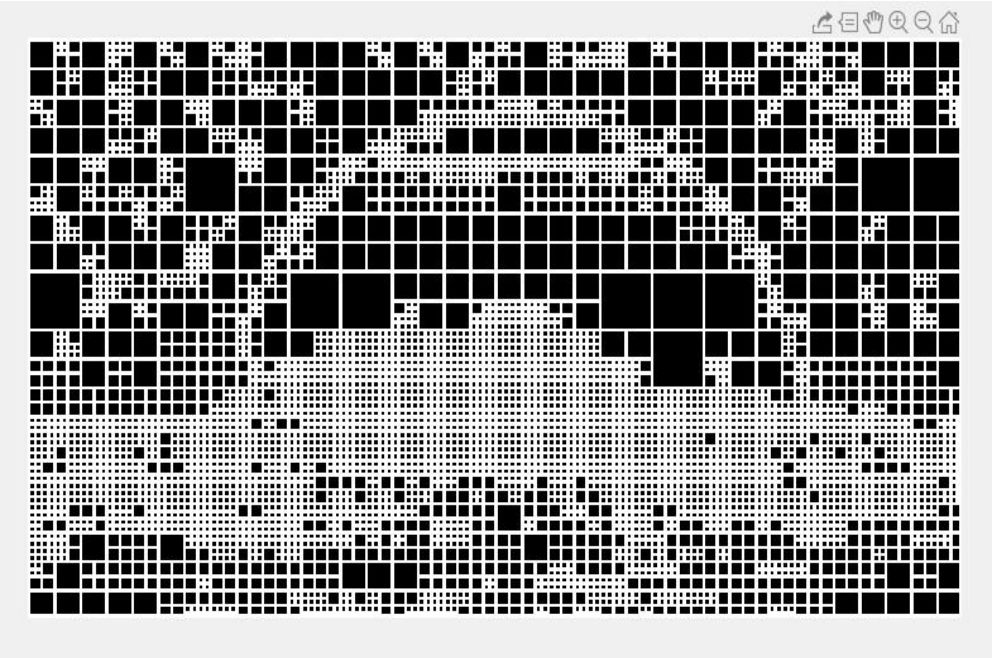
3.3 Mô phỏng cấu trúc quadtree khi DCT

```
clear
A = imread('C:\Users\USER\Pictures\Saved
Pictures\2.jpg');
[Height,Width,Depth] = size(A);
if mod(Height,16)~= 0
Height = floor(Height/16)*16;
end
if mod(Width,16)~= 0
Width = floor(Width/16)*16;
end
A1 = A(1:Height,1:Width,:);
clear A
A = A1;
if Depth == 3
A = rgb2ycbcr(A);
A1 = A(:,:,1);
end
```

```
clear A
A = A1;
y = varSizeDCTcoder(A,0.12,2);
function Aq = varSizeDCTcoder(A,Thresh,Qscale)

[Height,Width] = size(A);
S = qtdecomp(A,Thresh,[2,16]);
QuadBlks = repmat(uint8(0),size(S));
for dim = [2 4 8 16]
numBlks = length(find(S==dim));
if (numBlks > 0)
Val = repmat(uint8(1),[dim dim numBlks]);
Val(2:dim,2:dim,:) = 0;
QuadBlks = qtsetblk(QuadBlks,S,dim,Val);
end
end
QuadBlks(end,1:end) =1;
QuadBlks(1:end,end) = 1;
figure,imshow(QuadBlks,[])
Qsteps8 = [16 11 10 16 24 40 51 61;...
           12 12 14 19 26 58 60 55;...
           14 13 16 24 40 57 69 56;...
           14 17 22 29 51 87 80 62;...
           18 22 37 56 68 109 103 77;...
           24 35 55 64 81 104 113 92;...
           49 64 78 87 103 121 120 101;...
           72 92 95 98 112 100 103 99];

Qsteps2 = [8 34; 34 34];
Qsteps4 = [8 24 24 24; 24 24 24 24; 24 24 24 24; 24 24 24
24];
Qsteps16 = [4 16 16 16 16 16 16 16 16 16 16 16 16 16 16
16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16;...
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16];
Aq = uint8(zeros(Height,Width));
BlkPercent = zeros(4,1);
m = 1;
```

```
for dim = [2 4 8 16]
[x,y] = find(S == dim);
BlkPercent(m) = length(x)*dim*dim*100/(Height*Width);
for k = 1:length(x)
t = dct2(double(A(x(k):x(k)+dim-1,y(k):y(k)+dim-1)));
switch dim
    case 2
t = round(t ./ Qsteps2) .* Qsteps2;
case 4
t = round(t ./ Qsteps4) .* Qsteps4;
case 8
t = round(t ./ (Qscale*Qsteps8)) .* (Qscale*Qsteps8);
case 16
t = round(t ./ Qsteps16) .* Qsteps16;
end
Aq(x(k):x(k)+dim-1,y(k):y(k)+dim-1) = uint8(idct2(t));
end
m = m + 1;
end
figure,imshow(Aq)
mse = std2(double(A)-double(Aq));
sprintf('2x2 = %5.2f%%\t4x4 = %5.2f%%\t8x8
= %5.2f%%\t16x16 = %5.2f%%\n',BlkPercent)
sprintf('SNR = %4.2f dB\n',
20*log10(std2(double(A))/mse))
end
```

Hình 3.4: Ảnh gốc



Hình 3.5: Ảnh được chia thành block có kích thước từ 2x2 đến 16x16

4. Kết luận, đánh giá

HEVC/H.265 cũng tương tự như các chuẩn nén video thế hệ trước như AVC/H.264 nhưng có đôi chút khác biệt. Tìm hiểu về chuẩn HEVC/H.265 giúp hiểu thêm về quá trình và video được nén và truyền đi như thế nào thông qua quá trình DCT từng ảnh.

Đồ án chỉ tìm hiểu tổng quan về lý thuyết của chuẩn nén HEVC/H.265 và mô phỏng quá trình DCT ảnh bằng MATLAB nên còn khá hạn chế và thiếu sót.

TÀI LIỆU THAM KHẢO

[1] Z. R. Saleemi and G. Raja, "Requirement based transform coefficient coding architecture for DCT/DST for HEVC," 2017 International Symposium on Wireless Systems and Networks (ISWSN), Lahore, Pakistan, 2017, pp. 1-5, doi: 10.1109/ISWSN.2017.8250029.

[2] M. Porto et al, "Hardware Design of Fast HEVC 2-D IDCT Targeting Real-Time UHD 4K Applications," in 2015 IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS), Montevideo, 2015.

[3] M. A. Saleh, H. Hashim, N. M. Tahir and E. Hisham, "Review for High Efficiency Video Coding (HEVC)," 2014 IEEE Conference on Systems, Process and Control (ICSPC 2014), Kuala Lumpur, Malaysia, 2014, pp. 141-146, doi: 10.1109/SPC.2014.7086246.

[4] B. G. Haskell, F. W. Mounts, and J. C. Candy, “Interframe coding of videotelephone pictures,” Proc. IEEE, 60 (7), 792–800, 1972.

[5] A. N. Netravali and B. Prasada, “Adaptive quantization of picture signals using spatial masking,” Proc. IEEE, 65 (4), 536–548, 1977.

[6] W. A. Pearlman, “Adaptive cosine transform image coding with constant block distortion,” IEEE Trans. Comm., COM-38(5), 698–703, 1990.

[7] K. S. Thyagarajn and M. Merritt, “Contrast sensitive variance-based adaptive block size DCT image compression,” US Patent 6,529,634.

[8] A. Puri and A. Eleftheriadis, “MPEG-4: An object-based multimedia coding standard supporting mobile applications,” Mobile Netw. Appl., (3), 5–32, 1998.

[9] J. Lee and B. W. Dickinson, “Temporally adaptive motion interpolation exploiting temporal masking in visual perception,” IEEE Trans. Image Proc., 3 (5), 513–526, September 1994.

[10] A. Puri and R. Aravind, “Motion-compensated video coding with adaptive perceptual quantization,” IEEE Trans. Circuits Syst. Video Technol., 1 (4), 351–361, 1991.

[11] M. R. Pickering and J. F. Arnold, “A perceptually efficient VBR rate control algorithm,” IEEE Trans. Image Proc., 3 (5), 527–532, 1994.

[12] G. Pastuszak and A. Abramowski, "Algorithm and Architecture Design of the H.265/HEVC Intra Encoder," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 210-222, 2016.

[13] J. Ohm, W. Han, T. Wiegand and G. J. Sullivan, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012.

[14] J.-S. Park,W.-J. Nam, S.-M.Han, and S. Lee, “2-D large inverse transform (16 16, 32 32) for HEVC (high efficiency video coding),” J.Semicond. Technol. Sci., vol. 2, pp. 203–211, 2012.

[15] Yanxiang Wang, Charith Abhayaratne, Marta Mrak, Chapter 3 - High Efficiency Video Coding (HEVC) for Next Generation Video Applications, Editor(s): Sergios Theodoridis, Rama Chellappa, Academic Press Library in Signal Processing, Elsevier, Volume 5, 2014, Pages 93-117, ISSN 2351-9819, ISBN 9780124201491.

[16] VENU, M., RUKHSAR, M. J., & SINGH, B. N. (2015). Implementation High-Level Syntax Architecture for Efficient Integer DCT for HEVC.

[17] G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on

Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.