



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0049370

(51)^{2020.01} H04N 19/159; H04N 19/132; H04N
19/70; H04N 19/176; H04N 19/593;
H04N 19/11

(13) B

(21) 1-2022-00075

(22) 24/06/2020

(86) PCT/KR2020/008206 24/06/2020

(87) WO2020/262951 30/12/2020

(30) 10-2019-0075163 24/06/2019 KR; 10-2019-0080749 04/07/2019 KR; 10-2020-0077014 24/06/2020 KR

(45) 25/07/2025 448

(43) 25/05/2022 410A

(73) 1. HYUNDAI MOTOR COMPANY (KR)

12, Heolleung-ro Seocho-gu Seoul 06797 Republic of Korea

2. KIA CORPORATION (KR)

12, Heolleung-ro Seocho-gu Seoul 06797 Republic of Korea

3. EWha UNIVERSITY - INDUSTRY COLLABORATION FOUNDATION (KR)

52, Ewhayeodae-gil Seodaemun-gu Seoul 03760 Republic of Korea

(72) KANG, Je Won (KR); PARK, Sang Hyo (KR); PARK, Seung Wook (KR); LIM, Wha Pyeong (KR).

(74) Công ty TNHH Đại Tín và Liên Danh (DAITIN AND ASSOCIATES CO.,LTD)

(54) PHƯƠNG PHÁP GIẢI MÃ DỰ ĐOÁN NỘI BỘ CỦA DỮ LIỆU VIDEO

(21) 1-2022-00075

(57) Sáng chế đề xuất phương pháp để mã hóa dự đoán nội bộ dữ liệu video. Thiết bị bao gồm bộ giải mã được cấu hình để giải mã, từ dòng bit, phần tử cú pháp cho biết kiểu dự đoán nội bộ của khối hiện tại của dữ liệu video, và bộ dự đoán nội bộ được cấu hình để tạo khối dự đoán cho khối hiện tại bằng cách thực hiện có chọn lọc dự đoán nội bộ dựa trên ma trận hoặc dự đoán nội bộ thông thường dựa trên kiểu dự đoán nội bộ của khối hiện tại được chỉ ra bởi phần tử cú pháp.

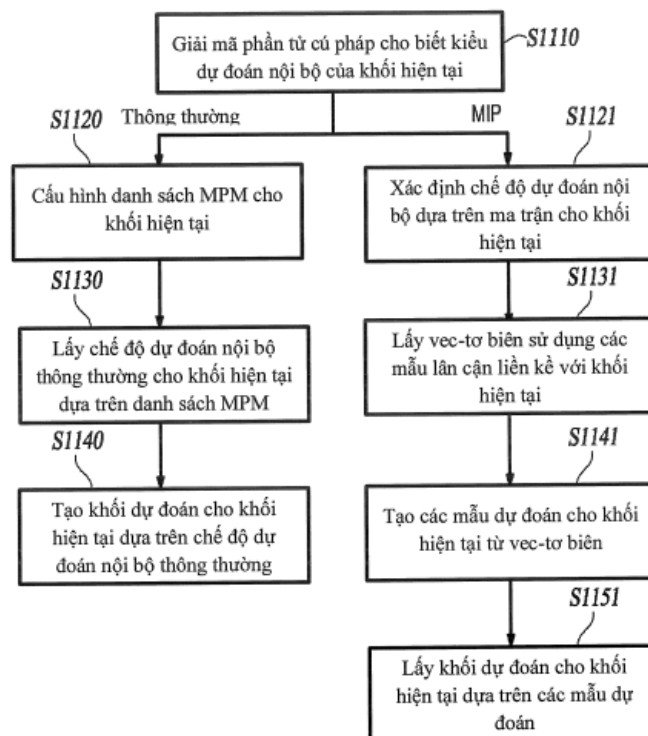


Fig.11

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến lĩnh vực mã hóa và giải mã dữ liệu video.

Tình trạng kỹ thuật của sáng chế

Vì khối lượng dữ liệu video lớn hơn dữ liệu thoại hoặc dữ liệu ảnh tĩnh, nên việc lưu trữ hoặc truyền dữ liệu video mà không cần xử lý để nén cần nhiều tài nguyên phần cứng bao gồm cả bộ nhớ. Theo đó, trong việc lưu trữ hoặc truyền dữ liệu video, dữ liệu video thường được nén sử dụng bộ mã hóa để lưu trữ hoặc truyền. Sau đó, bộ giải mã nhận dữ liệu video đã nén, giải nén và tái tạo dữ liệu video. Các kỹ thuật nén video theo kiểu này bao gồm H.264/AVC và mã hóa video hiệu năng cao (High Efficiency Video Coding, HEVC), giúp cải thiện khoảng 40% hiệu quả mã hóa so với H.264/AVC.

Tuy nhiên, kích thước hình ảnh, độ phân giải, và tốc độ khung hình đang dần được tăng lên, và theo đó lượng dữ liệu được mã hóa cũng tăng theo. Vì vậy, cần có một kỹ thuật nén mới có hiệu quả mã hóa và chất lượng hình ảnh cao hơn so với kỹ thuật nén hiện có.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất các kỹ thuật được cải tiến để mã hóa dự đoán nội bộ khối của dữ liệu video.

Theo khía cạnh thứ nhất của sáng chế, sáng chế đề xuất phương pháp giải mã dữ liệu video, bao gồm giải mã, từ dòng bit, phần tử cú pháp chỉ kiểu dự đoán nội bộ cho khối hiện tại của dữ liệu video, kiểu dự đoán nội bộ được chỉ ra từ các dự đoán nội bộ dựa trên ma trận (Matrix Based Intra-Prediction, MIP) và dự đoán nội bộ thông thường; và tạo khối dự đoán cho khối hiện tại bằng cách thực hiện có chọn lọc MIP hoặc dự đoán nội bộ thông thường dựa trên kiểu dự đoán nội bộ của khối hiện tại được chỉ ra bởi phần tử cú pháp.

Là một phần của quá trình tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ thông thường, phương pháp này bao gồm lấy ra các ứng viên của chế độ đúng nhất (Most Probable Mode, MPM) dựa trên chế độ dự đoán nội bộ thông thường của từng khối lân cận liền kề với khối hiện tại và cấu hình danh sách MPM cho khối hiện tại;

và lấy chế độ dự đoán nội bộ thông thường cho khối hiện tại dựa trên danh sách MPM. Khi kiểu dự đoán nội bộ của các khối lân cận là MIP, thì chế độ dự đoán nội bộ thông thường của khối lân cận được đặt ở chế độ PLANAR.

Là một phần của quá trình tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ thông thường, phương pháp này bao gồm giải mã, từ dòng bit, phần tử cú pháp chỉ ra chế độ MIP cho khối hiện tại; lấy vec-tơ biên sử dụng các mẫu lân cận liền kề với khối hiện tại dựa trên kích thước và hình dạng của khối hiện tại; tạo các mẫu dự đoán cho khối hiện tại dựa trên phép nhân vec-tơ ma trận giữa vec-tơ biên và ma trận được xác định trước cho chế độ MIP; và lấy được khối dự đoán cho khối hiện tại dựa trên các mẫu dự đoán.

Theo một khía cạnh khác của sáng chế, sáng chế đề xuất thiết bị giải mã dữ liệu video, bao gồm bộ giải mã được cấu hình để giải mã, từ dòng bit, phần tử cú pháp cho biết kiểu dự đoán nội bộ của khối hiện tại của dữ liệu video, kiểu dự đoán nội bộ được chỉ ra từ dự đoán nội bộ dựa trên ma trận (MIP) và dự đoán nội bộ thông thường; và bộ dự đoán nội bộ được cấu hình để tạo khối dự đoán cho khối hiện tại bằng cách thực hiện có chọn lọc MIP hoặc dự đoán nội bộ thông thường dựa trên kiểu dự đoán nội bộ của khối hiện tại được chỉ định bởi phần tử cú pháp.

Là một phần của quá trình tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ thông thường, công cụ dự đoán nội bộ được cấu hình để lấy ra các ứng viên của chế độ đúng nhất (MPM) dựa trên chế độ dự đoán nội bộ thông thường của mỗi khối lân cận liền kề với khối hiện tại và cấu hình danh sách MPM cho khối hiện tại; và lấy ra chế độ dự đoán nội bộ thông thường cho khối hiện tại dựa trên danh sách MPM. Trong việc lấy các ứng viên MPM, khi kiểu dự đoán nội bộ của các khối lân cận là MIP, thì công cụ dự đoán nội bộ đặt (coi) chế độ dự đoán nội bộ thông thường của khối lân cận là chế độ PLANAR.

Là một phần của quá trình tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ thông thường, bộ dự đoán nội bộ có thể giải mã, từ dòng bit, phần tử cú pháp chỉ ra chế độ MIP cho khối hiện tại; lấy được vec-tơ biên sử dụng các mẫu lân cận liền kề với khối hiện tại dựa trên kích thước và hình dạng của khối hiện tại; tạo các mẫu dự đoán cho khối hiện tại dựa trên phép nhân vec-tơ ma trận giữa vec-tơ biên và ma trận

được xác định trước cho chế độ MIP; và lấy khối dự đoán cho khối hiện tại dựa trên các mẫu dự đoán.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối mẫu minh họa thiết bị mã hóa video có thể triển khai các kỹ thuật của sáng chế.

Fig.2 là hình minh họa cấu trúc phân vùng khối sử dụng cấu trúc QTBT TT.

Fig.3a là hình minh họa các chế độ dự đoán nội bộ.

Fig.3b là hình minh họa các chế độ dự đoán nội bộ bao gồm các chế độ dự đoán nội bộ góc rộng.

Fig.4 là sơ đồ khối mẫu minh họa thiết bị giải mã video có thể triển khai các kỹ thuật của sáng chế.

Fig.5 là sơ đồ khái niệm minh họa quy trình chính của kỹ thuật MIP có thể được sử dụng trong các kỹ thuật của sáng chế.

Fig.6 là lưu đồ minh họa sơ đồ quy trình dự đoán khối dựa trên MIP theo phương án thực hiện của sáng chế.

Fig.7a-7c là các sơ đồ khái niệm minh họa phương pháp mẫu tạo vec-tơ biên mà để nhập vào phép nhân vec-tơ ma trận sử dụng các mẫu lân cận bên trái.

Fig.8a-8c là các sơ đồ khái niệm minh họa phương pháp mẫu để tạo vec-tơ biên mà để nhập vào phép nhân vec-tơ ma trận sử dụng các mẫu lân cận bên trên.

Fig.9 là hình minh họa các mẫu lân cận bên trái, vec-tơ biên, và các mẫu dự đoán được dự đoán của chúng.

Fig.10 là hình minh họa các mẫu lân cận bên trên, vec-tơ biên và các mẫu dự đoán được dự đoán của chúng.

Fig.11 là lưu đồ minh họa phương pháp giải mã dữ liệu video theo phương án thực hiện của sáng chế.

Mô tả chi tiết sáng chế

Sau đây là phần mô tả chi tiết sáng chế thông qua các phương án thực hiện cùng với việc tham chiếu đến các hình vẽ kèm theo. Cần lưu ý rằng, các số tham chiếu được thêm vào các phần tử thành phần trong các hình vẽ tương ứng, các số tham chiếu giống nhau chỉ định các phần tử giống nhau, mặc dù các phần tử được thể hiện trong các hình vẽ là khác nhau. Ngoài ra, trong phần mô tả dưới đây của sáng chế, để sáng chế trở nên dễ hiểu hơn, sẽ bỏ qua việc mô tả chi tiết về các chức năng và cấu hình được kết hợp đã biết.

Fig.1 là sơ đồ khối mẫu minh họa thiết bị mã hóa video có thể triển khai các kỹ thuật của sáng chế. Sau đây, thiết bị mã hóa video và các phần tử của thiết bị sẽ được mô tả cùng với việc tham chiếu đến Fig.1.

Thiết bị mã hóa video bao gồm bộ tách hình ảnh 110, bộ dự đoán 120, bộ trừ 130, bộ biến đổi 140, bộ lượng tử hóa 145, bộ điều chỉnh lại 150, bộ mã hóa entropy 155, bộ lượng tử hóa ngược 160, bộ biến đổi ngược 165, bộ cộng 170, bộ phân lọc 180 và bộ nhớ 190.

Mỗi phần tử của thiết bị mã hóa video có thể được triển khai trong phần cứng hoặc phần mềm hoặc kết hợp giữa phần cứng và phần mềm. Các chức năng của các phần tử tương ứng có thể được triển khai dưới dạng phần mềm, và bộ vi xử lý có thể được triển khai để thực thi các chức năng phần mềm tương ứng với các phần tử tương ứng.

Một video bao gồm nhiều hình ảnh. Mỗi hình ảnh được chia thành nhiều vùng và mã hóa được thực hiện trên mỗi vùng. Ví dụ, một hình ảnh được chia thành một hoặc nhiều ô hoặc/và lát cắt. Cụ thể, một hoặc nhiều ô có thể được xác định là một nhóm ô. Mỗi ô hoặc lát cắt được chia thành một hoặc nhiều đơn vị cây mã hóa (Coding Tree Unit, CTU). Mỗi CTU được chia thành một hoặc nhiều đơn vị mã hóa (Coding Unit, CU) theo cấu trúc cây. Thông tin áp dụng cho mỗi CU được mã hóa dưới dạng cú pháp của CU, và thông tin áp dụng cho các CU bao gồm trong một CTU chung được mã hóa dưới dạng cú pháp của CTU. Ngoài ra, thông tin áp dụng cho tất cả các khối trong một lát cắt chung được mã hóa dưới dạng cú pháp của tiêu đề lát cắt, và thông tin áp dụng cho tất cả các khối cấu thành ảnh được mã hóa trong bộ tham số ảnh (Picture Parameter Set, PPS) hoặc tiêu đề ảnh. Hơn nữa, thông tin chung mà được nhiều ảnh đề cập đến được mã hóa trong bộ tham số trình tự (Sequence Parameter Set, SPS). Ngoài ra, thông tin được đề cập đến bởi một hoặc nhiều

SPS chung được mã hóa trong bộ thông số video (Video Parameter Set, VPS). Thông tin áp dụng cho một ô hoặc nhóm ô chung có thể được mã hóa dưới dạng cú pháp của ô hoặc nhóm ô tiêu đề.

Bộ tách hình ảnh 110 được cấu hình để xác định kích thước của đơn vị cây mã hóa (CTU). Thông tin về kích thước của CTU (CTU Size) được mã hóa dưới dạng cú pháp của SPS hoặc PPS và được truyền đến thiết bị giải mã video. Bộ tách hình ảnh 110 được cấu hình để tách từng hình ảnh cấu thành video thành nhiều CTU có kích thước xác định trước, và sau đó chia đệ quy các CTU sử dụng cấu trúc cây. Trong cấu trúc cây, một nút lá đóng vai trò là đơn vị mã hóa (CU), là đơn vị cơ bản của mã hóa.

Cấu trúc cây có thể là cây tứ phân (QuadTree, QT), trong đó một nút (hoặc nút cha) được tách thành bốn nút con (hoặc nút con) có cùng kích thước, cây nhị phân (BinaryTree, BT), trong đó một nút được tách thành hai nút phụ, cây tam phân (TernaryTree, TT), trong đó một nút được chia thành ba nút phụ có tỷ lệ 1:2:1, hoặc cấu trúc được hình thành bởi sự kết hợp của hai hoặc nhiều cấu trúc QT, cấu trúc BT, và cấu trúc TT. Ví dụ, có thể sử dụng cấu trúc cây tứ phân cộng với cây nhị phân (QuadTree Plus BinaryTree, QTBT), hoặc có thể sử dụng cấu trúc cây tứ phân cộng với cây nhị phân cây tam phân (QuadTree Plus BinaryTree TernaryTree, QTBTBT). Ở đây, BTBT có thể được gọi chung là cây đa kiểu (Multiple-Type Tree, MTT).

Fig.2 là hình minh họa cấu trúc cây tách QTBTBT. Như được thể hiện trong Fig.2, CTU ban đầu có thể được tách trong cấu trúc QT. Việc tách QT có thể được lặp lại cho đến khi kích thước của khối tách đạt đến kích thước khối tối thiểu (MinQTSIZE) của nút lá được cấp phép trong QT. Cờ thứ nhất (QT_split_flag) cho biết liệu mỗi nút của cấu trúc QT có được chia thành bốn nút của lớp thấp hơn được mã hóa bởi bộ mã hóa entropy 155 và được báo hiệu tới thiết bị giải mã video hay không. Khi nút lá của QT nhỏ hơn kích thước khối tối đa (MaxBTSIZE) của nút gốc cho phép trong BT, nó có thể bị tách thành một hoặc nhiều cấu trúc BT hoặc cấu trúc TT. Cấu trúc BT và/hoặc cấu trúc TT có thể có nhiều hướng tách.

Ví dụ, có thể có hai hướng, cụ thể, hướng mà trong đó khối của nút được chia theo chiều ngang và hướng mà trong đó khối được chia theo chiều dọc. Như được thể hiện trong Fig.2, khi khởi động tách MTT, cờ thứ hai (mtt_split_flag) cho biết liệu các nút có được

tách hay không, cờ chỉ hướng phân tách (dọc hoặc ngang) trong trường hợp phân tách, và/hoặc cờ chỉ ra kiểu phân tách (nhị phân hoặc tam phân) được mã hóa bởi bộ mã hóa entropy 155 và được báo hiệu đến thiết bị giải mã video. Ngoài ra, trước khi mã hóa cờ thứ nhất (QT_split_flag) cho biết liệu mỗi nút có được chia thành 4 nút của lớp thấp hơn hay không, cờ phân tách CU (split_cu_flag) cho biết liệu nút được tách có thể được mã hóa hay không. Khi giá trị của cờ phân tách CU (split_cu_flag) chỉ rằng việc tách không được thực hiện, thì khối của nút sẽ trở thành nút lá trong cấu trúc cây tách và đảm nhiệm đơn vị mã hóa (CU), là đơn vị cơ bản của mã hóa. Khi giá trị của cờ phân tách CU (split_cu_flag) chỉ rằng quá trình phân tách được thực hiện, thì thiết bị mã hóa video bắt đầu mã hóa các cờ theo cách được mô tả ở trên, bắt đầu với cờ thứ nhất.

Khi QTBT được sử dụng làm ví dụ khác về cấu trúc cây, có thể có hai kiểu tách, đó là kiểu tách khối theo chiều ngang thành hai khối có cùng kích thước (tức là chia theo chiều ngang đối xứng) và kiểu tách theo chiều dọc tách một khối thành hai khối có cùng kích thước (tức là tách theo chiều dọc đối xứng). Cờ phân tách (split_flag) cho biết liệu mỗi nút của cấu trúc BT có được tách thành khối của lớp thấp hơn hay không và thông tin về kiểu phân tách cho biết kiểu phân tách được mã hóa bởi bộ mã hóa entropy 155 và được truyền đến bộ giải mã video. Còn có thể có thêm kiểu tách khối nút thành hai khối bất đối xứng. Kiểu tách không đối xứng có thể bao gồm kiểu chia khối thành hai khối hình chữ nhật với tỷ lệ kích thước 1:3, hoặc kiểu tách khối của nút theo đường chéo.

Các CU có thể có nhiều kích cỡ khác nhau tùy theo phân tách QTBT hoặc QTBT TT của CTU. Sau đây, một khối tương ứng với CU (tức là một nút lá của QTBT TT) được mã hóa hoặc giải mã được gọi là “khối hiện tại”. Khi tách QTBT TT được sử dụng, hình dạng của khối hiện tại có thể là hình vuông hoặc hình chữ nhật. Bộ dự đoán 120 được cấu hình để dự đoán khối hiện tại để tạo khối dự đoán. Bộ dự đoán 120 bao gồm bộ dự đoán nội bộ 122 và bộ dự đoán liên 124.

Nói chung, mỗi khối hiện tại trong ảnh có thể được mã hóa theo dự đoán. Ngoài ra, dự đoán khối hiện tại được thực hiện bằng kỹ thuật dự đoán nội bộ (sử dụng dữ liệu từ ảnh chứa khối hiện tại) hoặc kỹ thuật dự đoán liên (sử dụng dữ liệu từ ảnh được mã hóa trước khi ảnh chứa khối hiện tại). Dự đoán liên bao gồm cả dự đoán một hướng và dự đoán hai hướng.

Bộ phận dự đoán nội bộ 122 được cấu hình để dự đoán các pixel trong khối hiện tại sử dụng các pixel (pixel tham chiếu) được đặt xung quanh khối hiện tại trong hình ảnh hiện tại bao gồm khối hiện tại. Có nhiều chế độ dự đoán nội bộ theo các hướng dự đoán. Ví dụ, như được thể hiện trong Fig.3a, nhiều chế độ dự đoán nội bộ có thể bao gồm hai chế độ vô hướng, bao gồm chế độ phẳng và chế độ DC, và 65 chế độ có hướng. Các pixel lân cận và phương trình được sử dụng sẽ được xác định khác nhau cho từng chế độ dự đoán. Bảng dưới đây liệt kê các số và tên gọi của chế độ dự đoán nội bộ của chúng.

Bảng 1

Chế độ dự đoán nội bộ	Tên liên kết
0	INTRA_PLANAR
1	INTRA_DC
2 ... 66	INTRA_ANGULAR2 ... INTRA_ANGULAR66

Để dự đoán hướng hiệu quả cho khối hiện tại hình chữ nhật, có thể dùng thêm các chế độ có hướng (chế độ dự đoán nội bộ từ 67 đến 80 và -1 đến -14) được chỉ định bằng các mũi tên chấm trong Fig.3b. Các chế độ này có thể được gọi là “chế độ dự đoán nội bộ góc rộng”. Trong Fig.3b, các mũi tên chỉ ra các mẫu tham chiếu tương ứng được sử dụng để dự đoán, không dùng để chỉ hướng dự đoán. Hướng dự đoán thì ngược lại với hướng được chỉ bởi mũi tên. Chế độ dự đoán nội bộ góc rộng là chế độ mà trong đó dự đoán được thực hiện theo hướng ngược lại với chế độ định hướng cụ thể mà không cần truyền thêm bit khi khối hiện tại có hình chữ nhật.

Cụ thể, trong các chế độ dự đoán nội bộ góc rộng, một số chế độ dự đoán nội bộ góc rộng khả dụng cho khối hiện tại có thể được xác định dựa trên tỷ lệ chiều rộng và chiều cao của khối hiện tại hình chữ nhật. Ví dụ, các chế độ dự đoán nội bộ góc rộng với góc nhỏ hơn khoảng 45 độ (chế độ dự đoán nội bộ 67 đến 80) có thể được sử dụng khi khối hiện tại có hình dạng hình chữ nhật với chiều cao nhỏ hơn chiều rộng. Có thể sử dụng chế độ dự đoán nội bộ góc rộng với góc lớn hơn khoảng -135 độ (chế độ dự đoán nội bộ -1 đến -14) khi khối hiện tại có hình dạng hình chữ nhật với chiều cao lớn hơn chiều rộng.

Bộ dự đoán nội bộ 122 có thể được cấu hình để xác định chế độ dự đoán nội bộ sẽ được sử dụng để mã hóa khối hiện tại. Theo một số ví dụ, bộ dự đoán nội bộ 122 có thể được cấu hình để mã hóa khối hiện tại sử dụng một số chế độ dự đoán nội bộ và chọn chế độ dự đoán nội bộ thích hợp từ các chế độ đã thử nghiệm để sử dụng. Ví dụ, bộ dự đoán nội bộ 122 có thể được cấu hình để tính toán các giá trị độ biến dạng sử dụng phép phân tích độ biến dạng của một số chế độ dự đoán nội bộ đã thử nghiệm, và chọn chế độ dự đoán nội bộ có đặc điểm độ biến dạng tốt nhất trong các chế độ đã thử nghiệm.

Bộ dự đoán nội bộ 122 được cấu hình để chọn một chế độ dự đoán nội bộ trong các chế độ dự đoán nội bộ, và dự đoán khối hiện tại sử dụng các pixel lân cận (pixel tham chiếu) và một phương trình được xác định theo chế độ dự đoán nội bộ đã chọn. Thông tin về chế độ dự đoán nội bộ đã chọn được mã hóa bởi bộ mã hóa entropy 155 và được truyền đến thiết bị giải mã video. Ngoài ra, bộ dự đoán nội bộ 122 có thể được cấu hình để tạo khối dự đoán cho khối hiện tại, sử dụng dự đoán nội bộ dựa trên ma trận (MIP), sẽ được mô tả sau. Bộ dự đoán nội bộ 122 được cấu hình để tạo khối dự đoán cho khối hiện tại sử dụng vec-tơ biên lấy từ các mẫu được tái tạo ở bên trái của khối hiện tại và các mẫu được tái tạo bên trên khối hiện tại, ma trận xác định trước, và vec-tơ bù.

Ngoài ra, bộ dự đoán liên 124 được cấu hình để tạo khối dự đoán cho khối hiện tại thông qua bù chuyển động. Bộ dự đoán liên 124 được cấu hình để tìm kiếm khối tương đồng nhất với khối hiện tại trong ảnh tham chiếu đã được mã hóa và giải mã sớm hơn ảnh hiện tại, và tạo khối dự đoán cho khối hiện tại sử dụng khối đã tìm được. Sau đó, bộ dự đoán liên được cấu hình để tạo vec-tơ chuyển động tương ứng với sự thay thế giữa khối hiện tại trong hình ảnh hiện tại và khối dự đoán trong hình ảnh tham chiếu. Nói chung, ước tính chuyển động được thực hiện trên thành phần luma, và vec-tơ chuyển động được tính toán dựa trên thành phần luma được sử dụng cho cả thành phần luma và thành phần chroma. Thông tin chuyển động bao gồm thông tin về hình ảnh tham chiếu và thông tin về vec-tơ chuyển động được sử dụng để dự đoán khối hiện tại được mã hóa bởi bộ mã hóa entropy 155 và được truyền đến thiết bị giải mã video.

Bộ trừ 130 được cấu hình để trừ khối dự đoán được tạo bởi bộ dự đoán nội bộ 122 hoặc bộ dự đoán liên 124 khỏi khối hiện tại để tạo thành khối dư thừa. Bộ biến đổi 140 có

thể được cấu hình để tách khối dư thừa thành một hoặc nhiều khối biến đổi, và áp dụng phép biến đổi cho một hoặc nhiều khối biến đổi, từ đó biến đổi các giá trị dư thừa của khối biến đổi từ miền pixel sang miền tần số. Trong miền tần số, các khối đã biến đổi được gọi là khối hệ số chứa một hoặc nhiều giá trị hệ số biến đổi. Có thể sử dụng nhân biến đổi hai chiều để biến đổi, và nhân biến đổi một chiều có thể được sử dụng để biến đổi ngang và biến đổi dọc, tương ứng. Các nhân biến đổi có thể dựa trên biến đổi cosin rời rạc (Discrete Cosine Transform, DCT), biến đổi sin rời rạc (Discrete Sine Transform, DST) hoặc tương tự.

Bộ biến đổi 140 có thể được cấu hình để biến đổi tín hiệu dư thừa trong khối dư thừa sử dụng toàn bộ kích thước của khối dư thừa làm đơn vị biến đổi. Ngoài ra, bộ biến đổi 140 có thể được cấu hình để phân vùng khối dư thừa thành hai khối phụ theo hướng ngang hoặc hướng dọc, và chỉ biến đổi một trong hai khối phụ. Theo đó, kích thước của khối biến đổi có thể khác với kích thước của khối dư thừa (và do đó kích thước của khối dự đoán cũng vậy). Các giá trị mẫu dư thừa khác 0 có thể không tồn tại hoặc rất hiếm trong khối con chưa được biến đổi. Các mẫu dư thừa của khối con chưa được biến đổi không được báo hiệu, và có thể được thiết bị giải mã video coi là “0”. Có thể có nhiều kiểu phân vùng tùy theo hướng phân vùng và tỷ lệ phân vùng. Bộ biến đổi 140 có thể cung cấp thông tin về chế độ mã hóa (hoặc chế độ biến đổi) của khối dư thừa (ví dụ, thông tin cho biết liệu khối dư thừa được biến đổi hay khối con dư thừa được biến đổi, và thông tin cho biết loại phân vùng được chọn để phân vùng khối dư thừa thành các khối con, và thực hiện theo thông tin xác định khối con đã được chuyển đổi) tới bộ mã hóa entropy 155. Bộ mã hóa entropy 155 có thể được cấu hình để mã hóa thông tin về chế độ mã hóa (hoặc chế độ biến đổi) của khối dư thừa.

Bộ lượng tử hóa 145 có thể được cấu hình để lượng tử hóa hệ số biến đổi xuất từ bộ biến đổi 140, và xuất hệ số biến đổi lượng tử hóa tới bộ mã hóa entropy 155. Đối với một số khối hoặc khung, bộ lượng tử hóa 145 có thể được cấu hình để lượng tử hóa trực tiếp khối dư thừa liên quan mà không cần biến đổi. Bộ phận điều chỉnh lại 150 có thể được cấu hình để điều chỉnh lại các giá trị hệ số cho giá trị dư thừa đã lượng tử hóa. Bộ phận điều chỉnh lại 150 có thể được cấu hình để thay đổi mảng 2 chiều của hệ số thành chuỗi hệ số 1 chiều thông qua quét hệ số. Ví dụ, bộ phận điều chỉnh lại 150 có thể được cấu hình để quét các hệ số từ hệ số DC sang hệ số trong vùng tần số cao sử dụng quét zig-zag hoặc quét

chéo để xuất ra chuỗi hệ số 1 chiều. Tùy thuộc vào kích thước của đơn vị biến đổi và chế độ dự đoán nội bộ, quét dọc, trong đó mảng hai chiều của hệ số được quét theo hướng cột, hoặc hướng ngang, trong đó các hệ số hình dạng khối hai chiều có thể được quét theo hướng hàng, thay vì quét zig-zag. Nói cách khác, có thể xác định chế độ quét được sử dụng trong các loại sau bao gồm: quét zig-zag, quét chéo, quét dọc và quét ngang theo kích thước của đơn vị biến đổi và chế độ dự đoán nội bộ.

Bộ mã hóa entropy 155 được cấu hình để mã hóa hệ số biến đổi lượng tử hóa một chiều xuất từ bộ phận điều chỉnh lại 150 sử dụng các kỹ thuật mã hóa khác nhau như mã hóa nhị phân thích nghi (Context-based Adaptive Binary Arithmetic Code, CABAC) và mã hóa theo cấp số nhân Golomb, để tạo ra dòng bit. Bộ mã hóa entropy 155 được cấu hình để mã hóa thông tin như là kích thước CTU, cờ phân tách CU, cờ phân tách QT, kiểu phân tách MTT, và hướng phân tách MTT, được liên kết với phân tách khối, sao cho công cụ giải mã video có thể chia khối theo cách tương tự như trong thiết bị mã hóa video. Ngoài ra, bộ mã hóa entropy 155 được cấu hình để mã hóa thông tin về kiểu dự đoán cho biết liệu khối hiện tại được mã hóa bằng dự đoán nội bộ hay dự đoán liên, và mã hóa thông tin dự đoán nội bộ (tức là thông tin về chế độ dự đoán nội bộ) hoặc thông tin dự đoán liên (thông tin về chỉ số hình ảnh tham chiếu và vec-tơ chuyển động) theo kiểu dự đoán.

Bộ lượng tử hóa ngược 160 có thể được cấu hình để lượng tử hóa ngược các hệ số biến đổi lượng tử hóa xuất từ bộ lượng tử hóa 145 để tạo ra các hệ số biến đổi. Bộ biến đổi ngược 165 được cấu hình để biến đổi các hệ số biến đổi xuất từ bộ lượng tử hóa ngược 160 từ miền tần số sang miền không gian và tái tạo khối dư. Bộ cộng 170 được cấu hình để cộng khối dư thừa được tái tạo vào khối dự đoán được tạo bởi bộ dự đoán 120 để tái tạo khối hiện tại. Các pixel trong khối hiện tại đã được tái tạo dùng để làm pixel tham chiếu trong việc thực hiện dự đoán nội bộ của khối tiếp theo.

Bộ phận lọc 180 được cấu hình để lọc các pixel đã tái tạo nhằm giảm tạo tác nghẽn, tạo tác chuông, và tạo tác làm mờ tạo ra do dự đoán dựa trên khối và chuyển đổi/lượng tử hóa. Bộ phận lọc 180 có thể bao gồm bộ lọc gỡ lỗi 182 và bộ lọc phân bù tương thích pixel (Pixel Adaptive Offset, SAO) 184. Bộ lọc gỡ lỗi 180 được cấu hình để lọc biên giữa các khối được tái tạo để loại bỏ các tạo tác nghẽn do mã hóa/giải mã từng khối một gây ra, và bộ lọc SAO 184 được cấu hình để thực hiện lọc bổ sung trên video đã lọc gỡ lỗi. Bộ lọc SAO 184 là bộ lọc được sử dụng để bù cho sự chênh lệch giữa pixel được tái tạo và pixel

ban đầu bị mất do mã hóa. Các khối được tái tạo được lọc qua bộ lọc gỡ lỗi 182 và bộ lọc SAO 184 được lưu trữ trong bộ nhớ 190. Một khi tất cả các khối trong ảnh được tái tạo, ảnh đã được tái tạo có thể dùng để làm ảnh tham chiếu để dự đoán liên các khối trong hình ảnh sẽ được mã hóa tiếp theo.

Fig.4 là sơ đồ khối chức năng mẫu minh họa thiết bị giải mã video có thể triển khai các kỹ thuật của sáng chế. Sau đây, thiết bị giải mã video và các phần tử của thiết bị sẽ được mô tả cùng với việc tham chiếu đến Fig.4. Thiết bị giải mã video có thể bao gồm bộ giải mã entropy 410, bộ điều chỉnh lại 415, bộ lượng tử hóa ngược 420, bộ biến đổi ngược 430, bộ dự đoán 440, bộ cộng 450, bộ lọc 460, và bộ nhớ 470

Tương tự với thiết bị mã hóa video của Fig. 1, mỗi phần tử của thiết bị giải mã video có thể được thực hiện trong phần cứng, phần mềm, hoặc kết hợp giữa phần cứng và phần mềm. Hơn nữa, chức năng của từng phần tử có thể được triển khai trong phần mềm, và bộ vi xử lý có thể được triển khai để thực thi chức năng của phần mềm tương ứng với từng phần tử.

Bộ giải mã entropy 410 được cấu hình để xác định khối hiện tại sẽ được giải mã bằng cách giải mã dòng bit do thiết bị mã hóa video tạo ra và lấy thông tin liên quan đến phân tách khối, và lấy thông tin dự đoán và thông tin về tín hiệu dư thừa, và các loại tương tự theo yêu cầu để tái tạo khối hiện tại. Bộ giải mã entropy 410 được cấu hình để lấy thông tin về kích thước CTU từ bộ thông số trình tự (SPS) hoặc bộ thông số hình ảnh (PPS), xác định kích thước của CTU, và tách ảnh thành các CTU có kích thước xác định. Sau đó, bộ giải mã được cấu hình để xác định CTU là lớp trên cùng, nghĩa là nút gốc của cấu trúc cây, và lấy thông tin phân tách về CTU để tách CTU sử dụng cấu trúc cây.

Ví dụ, khi CTU được tách bằng cấu trúc QTBT, cờ thứ nhất (QT_split_flag) liên quan đến tách QT được lấy để chia mỗi nút thành bốn nút của lớp con. Đối với nút tương ứng với nút lá của QT, cờ thứ hai (MTT_split_flag) và thông tin về hướng tách (dọc/ngang) và/hoặc kiểu tách (nhị phân/tam phân) liên quan đến tách MTT được lấy để tách nút lá tương ứng trong cấu trúc MTT. Do đó, mỗi nút bên dưới nút lá của QT được tách đệ quy theo cấu trúc BT hoặc TT.

Theo ví dụ khác, khi CTU được tách bằng cấu trúc QTBT, cờ phân tách CU (split_cu_flag) cho biết liệu tách CU có thể được trích xuất hay không. Khi khối tương ứng

được tách, cờ thứ nhất (QT_split_flag) có thể được trích xuất. Trong thao tác phân tách, không hoặc nhiều tách MTT đệ quy có thể xảy ra cho mỗi nút sau khi không hoặc nhiều tách QT đệ quy. Ví dụ, CTU có thể trực tiếp trải qua quá trình tách MTT mà không cần tách QT, hoặc chỉ trải qua tách QT nhiều lần. Theo ví dụ khác, khi CTU được tách bằng cấu trúc QTBT, cờ thứ nhất (QT_split_flag) liên quan đến tách QT được lấy và mỗi nút được tách thành bốn nút của lớp thấp hơn. Sau đó, cờ phân tách (split_flag) cho biết liệu nút tương ứng với nút lá của QT có được tách thêm trong BT hay không và trích xuất được thông tin hướng tách.

Khi khối hiện tại chuẩn bị được giải mã được xác định thông qua phân tách trong cấu trúc cây, bộ giải mã entropy 410 được cấu hình để lấy thông tin về kiểu dự đoán cho biết khối hiện tại là dự đoán nội bộ hay dự đoán liên. Khi thông tin kiểu dự đoán chỉ là dự đoán nội bộ, thì bộ giải mã entropy 410 được cấu hình để trích xuất phần tử cú pháp cho thông tin dự đoán nội bộ (chế độ dự đoán nội bộ) cho khối hiện tại. Khi thông tin kiểu dự đoán chỉ là dự đoán liên, thì bộ giải mã entropy 410 được cấu hình để trích xuất phần tử cú pháp cho thông tin dự đoán liên, tức là, thông tin chỉ vec-tơ chuyển động và hình ảnh tham chiếu được tham chiếu đến bởi vec-tơ chuyển động.

Bộ giải mã entropy 410 được cấu hình để trích xuất thông tin về chế độ mã hóa của khối dư thừa (ví dụ, thông tin về việc liệu khối dư thừa được mã hóa chỉ một khối con của khối dư thừa có được mã hóa hay không, thông tin cho biết loại phân vùng được chọn để phân vùng khối dư thừa thành các khối con, thông tin xác định khối con dư thừa được mã hóa, các tham số lượng tử hóa, v.v.) từ dòng bit. Bộ giải mã entropy 410 cũng được cấu hình để trích xuất thông tin về các hệ số biến đổi lượng tử hóa của khối hiện tại dưới dạng thông tin về tín hiệu dư thừa.

Bộ phận điều chỉnh lại 415 có thể được cấu hình để thay đổi trình tự của các hệ số biến đổi lượng tử hóa một chiều được giải mã entropy bằng bộ giải mã entropy 410 thành mảng hệ số 2 chiều (tức là khối) theo thứ tự ngược lại của quét hệ số được thực hiện bởi thiết bị mã hóa video. Bộ lượng tử hóa ngược 420 được cấu hình để lượng tử hóa ngược các hệ số biến đổi lượng tử hóa. Bộ biến đổi ngược 430 được cấu hình để biến đổi ngược các hệ số biến đổi lượng tử hóa ngược từ miền tần số sang miền không gian dựa trên thông tin về chế độ mã hóa của khối dư thừa để tái tạo các tín hiệu dư thừa, từ đó tạo ra khối dư thừa được tái tạo cho khối hiện tại.

Khi thông tin về chế độ mã hóa của khối dư thừa chỉ ra rằng khối dư thừa của khối hiện tại đã được mã hóa bởi thiết bị mã hóa video, thì bộ biến đổi ngược 430 sử dụng kích thước của khối hiện tại (và do đó kích thước của khối dư thừa được tái tạo) làm đơn vị biến đổi cho các hệ số biến đổi lượng tử hóa ngược để thực hiện biến đổi ngược để tạo ra khối dư thừa được tái tạo cho khối hiện tại.

Khi thông tin về chế độ mã hóa của khối dư thừa chỉ ra rằng chỉ có một khối con của khối dư thừa đã được mã hóa bởi thiết bị mã hóa video, thì bộ biến đổi ngược 430 sử dụng kích thước của khối con đã biến đổi làm đơn vị biến đổi cho hệ số biến đổi lượng tử hóa để thực hiện biến đổi ngược nhằm tái tạo các tín hiệu dư thừa cho khối con đã biến đổi, và lấp đầy các tín hiệu dư thừa cho khối con chưa được biến đổi với giá trị “0” để tạo ra một khối dư thừa được tái tạo cho khối hiện tại.

Bộ dự đoán 440 có thể bao gồm bộ dự đoán nội bộ 442 và bộ dự đoán liên 444. Bộ dự đoán nội bộ 442 được kích hoạt khi kiểu dự đoán của khối hiện tại là dự đoán nội bộ, và bộ dự đoán liên 444 được kích hoạt khi kiểu dự đoán của khối hiện tại là dự đoán liên. Bộ dự đoán nội bộ 442 được cấu hình để xác định chế độ dự đoán nội bộ của khối hiện tại trong các chế độ dự đoán nội bộ dựa trên phần tử cú pháp cho chế độ dự đoán nội bộ được lấy từ bộ giải mã entropy 410, và dự đoán khối hiện tại sử dụng các pixel tham chiếu xung quanh khối hiện tại theo chế độ dự đoán nội bộ. Ngoài ra, bộ dự đoán nội bộ 442 có thể được cấu hình để tạo khối dự đoán cho khối hiện tại, sử dụng dự đoán nội bộ dựa trên ma trận (MIP), sẽ được mô tả sau. Bộ dự đoán nội bộ 442 có thể được cấu hình để tạo khối dự đoán cho khối hiện tại sử dụng vec-tơ biên lấy từ các mẫu được tái tạo ở bên trái khối hiện tại và các mẫu được tạo ở bên trên của khối hiện tại, và ma trận và vec-tơ bù được xác định trước.

Bộ dự đoán liên 444 được cấu hình để xác định vec-tơ chuyển động của khối hiện tại và hình ảnh tham chiếu được tham chiếu đến bởi vec-tơ chuyển động sử dụng phần tử cú pháp cho chế độ dự đoán nội bộ được lấy từ bộ giải mã entropy 410, và dự đoán khối hiện tại dựa trên vec-tơ chuyển động và hình ảnh tham chiếu. Bộ cộng 450 được cấu hình để tái tạo khối hiện tại bằng cách thêm khối dư thừa xuất từ bộ biến đổi ngược 430 và khối dự đoán xuất từ bộ dự đoán liên 444 hoặc bộ dự đoán nội bộ 442. Các pixel trong khối hiện tại được tái tạo để sử dụng làm pixel tham chiếu trong dự đoán nội bộ khối được giải mã tiếp theo.

Bộ phận lọc 460 có thể bao gồm bộ lọc gỡ lỗi 462 và bộ lọc SAO 464. Bộ lọc gỡ lỗi 462 thực hiện lọc-gỡ lỗi ranh giới giữa các khối được tái tạo để loại bỏ các tạo tác chặn do giải mã theo từng khối một. Bộ lọc SAO 464 được cấu hình để thực hiện lọc bổ sung trên khối được tái tạo sau khi lọc gỡ lỗi theo các phần bù tương ứng để bù cho sự chênh lệch giữa pixel được tái tạo và pixel ban đầu bị mất do mã hóa. Khối được tái tạo được lọc qua bộ lọc gỡ lỗi 462 và bộ lọc SAO 464 được lưu trữ trong bộ nhớ 470. Khi tất cả các khối trong hình ảnh được tái tạo, hình ảnh được tái tạo sẽ được sử dụng làm hình ảnh tham chiếu để dự đoán liên các khối trong hình ảnh được mã hóa tiếp theo.

Các kỹ thuật theo sáng chế thường liên quan đến mã hóa dự đoán nội bộ. Mô tả sau đây chủ yếu tập trung vào các kỹ thuật giải mã, tức là, các thao tác của bộ giải mã video. Các kỹ thuật mã hóa sẽ được mô tả ngắn gọn vì chúng đối lập với các kỹ thuật giải mã được mô tả toàn diện.

Trong cuộc hội thảo về tiêu chuẩn mã hóa video thế hệ tiếp theo (tức là mã hóa video đa năng (Versatile Video Coding, VVC)), một số công cụ mã hóa mới cho hiệu suất mã hóa tốt hơn so với mã hóa video hiệu năng cao (High Efficiency Video Coding, HEVC) đã được giới thiệu. Lựa chọn nhiều phép biến đổi (Multiple Transform Selection, MTS), là một kỹ thuật được giới thiệu trong VVC, dựa trên ba phép biến đổi lượng giác bao gồm DCT-2, DST-7 và DCT-8, và bộ mã hóa video có thể được cấu hình để chọn biến đổi ngang và biến đổi dọc, giúp giảm thiểu tổn thất độ biến dạng.

Bảng 2

Kiểu biến đổi	Chức năng cơ bản $T_i(j)$, $i, j = 0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ Trong đó $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

Đối với khối mà MTS được áp dụng, một hoặc nhiều phần tử cú pháp có thể được mã hóa để báo hiệu tương ứng với nhân biến đổi được sử dụng, theo hướng ngang và hướng dọc.

Ví dụ, ba cờ (MTS_CU_flag, MTS_Hor_flag, MTS_Ver_flag) có thể được sử dụng chung một bảng ánh xạ được xác định trước, chẳng hạn như Bảng 3.

Bảng 3

MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	Nội bộ/liên	
			Ngang	Dọc
0	-	-	DCT2	
1	0	0	DST7	DST7
	0	1	DCT8	DST7
	1	0	DST7	DCT8
	1	1	DCT8	DCT8

Thứ nhất, cờ (MTS_CU_flag) cho biết liệu DCT-2 có được sử dụng theo cả hai hướng ngang và dọc hay không. Khi DCT-2 không được áp dụng theo cả hai hướng, có thể báo hiệu cờ (MTS_Hor_flag) cho biết nhân biến đổi được áp dụng theo hướng ngang giữa DST-7 và DCT-8, và có thể báo hiệu cờ (MTS_Ver_flag) cho biết nhân biến đổi để được áp dụng theo hướng dọc giữa DST-7 và DCT-8. Có thể sử dụng DST-7 và DCT-8 cho kích thước khối lên đến 32×32 , và DCT-2 có thể được sử dụng cho kích thước khối lên đến 64×64 . Theo đó, MTS_CU_flag có thể được báo hiệu khi kích thước (chiều rộng và chiều cao) của khối nhỏ hơn hoặc bằng 32, và khối có ít nhất một tín hiệu dư thừa khác không. Khi MTS_CU_flag không xuất hiện, thì MTS_CU_flag có thể được suy ra là 0.

Ngoài ra, phần tử cú pháp (tu_mts_idx) biểu thị chỉ số nhân MTS có thể được sử dụng cùng với bảng ánh xạ xác định trước chẳng hạn như Bảng 4. Ngoài ra, tu_mts_idx cho biết chỉ số nhân MTS, và các nhân cho hướng ngang và dọc được chọn theo mỗi giá trị chỉ số. Trong bảng 4, trTypeHor và trTypeVer chỉ ra các loại nhân cho các hướng ngang và dọc. Cụ thể, 0 đại diện cho DCT-2, 1 đại diện cho DST-7, và 2 đại diện cho DCT-8. Ví dụ, khi tu_mts_idx = 2, DCT-8 được sử dụng theo hướng ngang và DST-7 được sử dụng theo hướng dọc.

Bảng 4

tu_mts_idx	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

Hơn nữa, tu_mts_idx chỉ được báo hiệu khi sps_explicit_mts_enabled_flag trong SPS là 1. Điều này được gọi là ExplicitMTS vì nó báo hiệu rõ ràng chỉ số MTS. Mặt khác, tu_mts_idx có thể không được chỉ ra rõ ràng, nhưng có thể được bộ mã hóa và bộ giải mã ngầm tính được, và được gọi là ImplicitMTS. ImplicitMTS có thể được thực hiện khi sps_explicit_mts_enabled_flag trong SPS bằng 0. Khi ImplicitMTS được sử dụng, thì có thể thực hiện lựa chọn nhân biến đổi dựa trên thông tin như chế độ dự đoán nội bộ, chế độ dự đoán liên được sử dụng cho khối, và kích thước của khối. Ví dụ, khi chiều rộng của khối lớn hơn hoặc bằng 4 và nhỏ hơn hoặc bằng 16, DST-7 được sử dụng làm nhân cho hướng ngang. Nếu không, DCT-2 được sử dụng. Ngoài ra, khi chiều cao của khối lớn hơn

hoặc bằng 4 và nhỏ hơn hoặc bằng 16, thì DST-7 được sử dụng làm nhân cho hướng dọc. Nếu không, DCT-2 được sử dụng. Nói cách khác, khi kích thước khối nhỏ, DST-7 được sử dụng vì tính có hướng của các tín hiệu dư thừa dự đoán có thể cao. Đối với một khối tương đối lớn, DCT-2 có thể được sử dụng làm nhân vì đặc tính của tín hiệu dư thừa là khá đồng đều là mạnh.

Bảng 5 thể hiện một phần của cú pháp đơn vị biến đổi được chỉ định trong bản thảo VVC 5. Trong cú pháp bên dưới, với mục đích dễ hiểu hơn, sẽ tô đậm các phần tử có màu xám.

Bảng 5

<code>transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {</code>	Descriptor
<code>(omitted)</code>	
<code>if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {</code>	
<code>if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)</code>	
<code>transform_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))</code>	
<code>tu_mts_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>if(tu_cbf_luma[x0][y0]) {</code>	
<code>if(!transform_skip_flag[x0][y0])</code>	
<code>residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)</code>	
<code>else</code>	
<code>residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)</code>	
<code>}</code>	
<code>(omitted)</code>	

Khối biến đổi có thể được mã hóa mà không cần thao tác biến đổi, được chỉ ra bởi phần tử cú pháp `transform_skip_flag` được báo hiệu cho khối biến đổi có ít nhất một hệ số khác 0. Kiểm soát cấp cao đối với chế độ mã hóa của khối biến đổi được thực hiện bởi

transform_skip_enabled_flag và log2_transform_skip_max_size_minus2, là các phần tử trong cú pháp cấp cao (High-Level Syntax, HLS) chẳng hạn như bộ tham số trình tự (SPS). Khi transform_skip_enabled_flag là 1, transform_skip_flag được mã hoá cho khối biến đổi có chiều cao và chiều rộng nhỏ hơn hoặc bằng “1 << (log2_transform_skip_max_size_minus2+2)”. Khi transform_skip_flag bằng 1, khối biến đổi liên quan được mã hoá ở chế độ bỏ qua biến đổi. Nếu không, có thể áp dụng phép biến đổi dựa trên MTS cho khối biến đổi liên quan.

Dự đoán nội bộ dựa trên ma trận (MIP) là kỹ thuật dự đoán nội bộ mới được giới thiệu trong VTM 5.0. Ý tưởng ban đầu là sử dụng kỹ thuật dự đoán nội bộ dựa trên mạng nơ-ron, nghĩa là, sử dụng mạng nơ-ron nhiều lớp để dự đoán các giá trị pixel hiện tại PU dựa trên các pixel được tái tạo liền kề. Tuy nhiên, do độ phức tạp cao của phương pháp dự đoán sử dụng mạng nơ-ron, nên kỹ thuật dự đoán nội bộ dựa trên biến đổi tuyến tính afin sử dụng các ma trận được đào tạo trước đã được giới thiệu.

Để dự đoán một khối hình chữ nhật PU có chiều rộng W và chiều cao H, MIP lấy ở đầu vào H các mẫu được tái tạo ở bên trái của khối và W mẫu được tái tạo ở phía trên của khối. Thu được các pixel dự đoán cuối cùng bằng phép trung bình, nhân vec-tơ ma trận, nội suy tuyến tính, và tương tự. Kích thước của khối mà MIP được áp dụng được phân thành ba loại như sau.

$$idx(W, H) = \begin{cases} 0 & \text{cho } W = H = 4 \\ 1 & \text{cho } \max(W, H) = 8 \\ 2 & \text{cho } \max(W, H) > 8 \end{cases}$$

Theo idx (W, H), số lượng chế độ MIP (numModes), kích thước biên (boundarySize), và kích thước khối dự đoán (predW, predH, predC) được xác định như sau. Trong bảng dưới đây, MipSizeId = idx (W, H).

Bảng 6

MipSizeId	numModes	boundarySize	predW	predH	predC
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	$\text{Min}(\text{nTbW}, 8)$	$\text{Min}(\text{nTbH}, 8)$	8

Fig.5 là sơ đồ khái niệm minh họa các quy trình chính của kỹ thuật MIP có thể được sử dụng trong các kỹ thuật của sáng chế.

(1) Phép trung bình

Mục đích chính của quá trình này là chuẩn hóa các mẫu tham chiếu. Tùy thuộc vào kích thước và hình dạng khối (tức là MipSizeId), thu được 4 hoặc 8 mẫu. Khi cả chiều rộng và chiều cao của khối hiện tại là 4 (tức là $W=H=4$), tổng cộng 4 mẫu, bao gồm 2 mẫu bên trái và 2 mẫu bên trên, sẽ thu được ($\text{boundarySize} = 2$). Trong trường hợp khác, tổng cộng 8 mẫu, bao gồm 4 mẫu bên trái và 4 mẫu bên trên, thu được ($\text{boundarySize} = 4$).

Như được thể hiện trong Fig.5, các mẫu lân cận bên trên được ký hiệu là $bdry^{top}$ và các mẫu lân cận bên trái được ký hiệu là $bdry^{left}$. Bằng cách thực hiện phép trung bình giữa $bdry^{top}$ và $bdry^{left}$, tương ứng, thu được các bộ mẫu được lấy xuống $bdry_{red}^{top}$ và $bdry_{red}^{left}$. Phép trung bình là quá trình lấy mẫu xuống như sau.

$$\text{redS}[x] = \left(\sum_{i=0}^{\text{bDwn}-1} \text{refS}[x * \text{bDwn} + i] + (1 \ll (\log_2(\text{bDwn}) - 1)) \right) \gg \text{Log2}(\text{bDwn})$$

Trong phương trình trên, bDwn biểu thị giá trị tỷ lệ lấy mẫu xuống ($\text{nTbs}/\text{boundarySize}$), và refS biểu thị mẫu tham chiếu ban đầu. RedS đã tính toán được lưu trữ là $bdry_{red}^{left}$ cho các lân cận bên trái và $bdry_{red}^{top}$ cho các lân cận bên trên.

Các mẫu tham chiếu được lấy mẫu xuống được ghép vào một vec-tơ có độ dài 4 hoặc 8. Vec-tơ đã giảm biên $bdry_{red}$ nhập với phép nhân ma trận vec-tơ được xác định theo phương trình bên dưới. Ví dụ, khi $W=H=4$ và chế độ MIP nhỏ hơn 18, vec-tơ biên được

tạo bằng cách ghép theo thứ tự của $bdry_{red}^{left}$ và $bdry_{red}^{top}$. Khi $W=H=4$ và chế độ MIP lớn hơn hoặc bằng 18, chúng được ghép theo thứ tự của $bdry_{red}^{left}$ và $bdry_{red}^{top}$. Trong phương trình sau, “mode” biểu thị chế độ MIP.

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}, bdry_{red}^{left}] & \text{cho } W = H = 4 \text{ và } mode < 18 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{cho } W = H = 4 \text{ và } mode \geq 18 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{cho } \max(W, H) = 8 \text{ và } mode < 10 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{cho } \max(W, H) = 8 \text{ và } mode \geq 10 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{cho } \max(W, H) > 8 \text{ và } mode < 6 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{cho } \max(W, H) > 8 \text{ và } mode \geq 6 \end{cases}$$

(2) Phép nhân ma trận-vec-tơ

Trong quá trình này, tín hiệu dự đoán được lấy mẫu xuống $pred_{red}$ của khối hiện tại được tạo ra từ vec-tơ đã giảm biên $bdry_{red}$. $pred_{red}$ là tổng của sản phẩm vec-tơ-ma trận và phần bù, được tính như sau.

$$pred_{red} = A \cdot bdry_{red} + b$$

Kích thước của $pred_{red}$ là $W_{red} \times H_{red}$. W_{red} và H_{red} được xác định theo kích thước và hình dạng của khối hiện tại như mô tả dưới đây. Ma trận A có số hàng bằng $W_{red} * H_{red}$, và có 4 cột khi $W=H=4$ hoặc 8 cột trong các trường hợp khác. Vec-tơ bù b là vec-tơ có kích thước $W_{red} * H_{red}$.

$$W_{red} = \begin{cases} 4 & \text{cho } \max(W, H) \leq 8 \\ \min(W, 8) & \text{cho } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{cho } \max(W, H) \leq 8 \\ \min(H, 8) & \text{cho } \max(W, H) > 8 \end{cases}$$

Tập hợp S_0 , S_1 , và S_2 của ma trận A và vec-tơ bù b có thể được sử dụng cho khối được xác định trước cho từng loại kích thước khối. Các chỉ số (0, 1, 2) của tập S được chọn theo MipSizeId đề cập ở trên (tức là $idx(W, H)$) và ma trận A và vec-tơ bù b được lấy từ một trong các tập S_0 , S_1 , và S_2 theo chế độ MIP được áp dụng cho khối hiện tại.

Tập hợp S_0 bao gồm 18 ma trận A_0 mỗi ma trận có 16 hàng và 4 cột, và 18 vec-tơ bù 16 chiều b_0 , và được sử dụng cho khối 4×4 . Tập hợp S_1 bao gồm 10 ma trận A_1 , mỗi ma

trận có 16 hàng và 8 cột và 10 vec-tơ bù 16 chiều b_1 , và được sử dụng cho các khối có kích thước 4×8 , 8×4 và 8×8 . Cuối cùng, tập hợp S_2 bao gồm 6 ma trận A_2 , mỗi ma trận có 64 hàng và 8 cột và 6 vec-tơ bù 64 chiều b_2 , và được sử dụng cho tất cả các hình dạng khối khác.

(3) Nội suy pixel

Nội suy là quá trình lấy mẫu lên. Như đã đề cập ở trên, $pred_{red}$ là tín hiệu dự đoán được lấy mẫu xuống của khối ban đầu. Cụ thể, khối dự đoán được lấy mẫu xuống có kích thước là $predW$ và $predH$ được định nghĩa như sau.

$$pred_{red}[x][y], \text{ với } x = 0..predW-1, y = 0..predH-1$$

Khối dự đoán có kích thước khối ban đầu ($nTbW$, $nTbH$) được tạo bằng cách nội suy tuyến tính tín hiệu dự đoán tại vị trí còn lại theo mỗi hướng được xác định như sau.

$$predSamples[x][y], \text{ với } x = 0..nTbW-1, y = 0..nTbH-1$$

Tùy thuộc vào hệ số tỷ lệ lấy mẫu lên theo chiều ngang và chiều dọc $upHor(=nTbW/predW)$ và $upVer(=nTbH/predH)$, một số hoặc tất cả các $predSamples$ được điền từ $pred_{red}$ như sau.

$$predSamples[(x+1) * upHor-1][(y+1) * upVer-1] = pred_{red}[x][y]$$

Khi $upHor = 1$, tất cả các vị trí ngang của các $predSamples$ từ $pred_{red}$ được lấp đầy. Khi $upVer=1$, tất cả các vị trí dọc của $predSamples$ từ $pred_{red}$ được lấp đầy.

Sau đó, các mẫu trống còn lại của $PredSamples$ được lấp đầy thông qua phép nội suy tuyến tính kép. Nội suy theo hướng ngang và nội suy theo hướng dọc là các quá trình lấy mẫu lên. Đối với phép nội suy của các mẫu bên trái và bên trên trong $predSamples$, các mẫu lấy mẫu xuống $bdry_{red}^{top}$ được gán cho các giá trị của $predSamples[x][-1]$, và các mẫu tham chiếu ban đầu ở bên trái được gán cho các giá trị của $predSamples[-1][y]$.

(4) Báo hiệu chế độ dự đoán nội bộ MIP

Đối với mỗi đơn vị mã hóa (CU) được mã hóa dự đoán nội bộ, truyền cờ cho biết liệu chế độ dự đoán nội bộ dựa trên ma trận (tức là chế độ MIP) có được áp dụng hay

không. Trong bản thảo 5 VVC, để báo hiệu chế độ MIP, danh sách MPM được sử dụng theo cách tương tự như chế độ dự đoán nội bộ truyền thống (sau đây, gọi là “chế độ dự đoán nội bộ thông thường”) khác với dự đoán nội bộ dựa trên ma trận. Ví dụ, `intra_mip_mpm_flag`, `intra_mip_mpm_idx`, và `intra_mip_mpm_remainder` được sử dụng cho báo hiệu chế độ MIP. `intra_mip_mpm_idx` được mã hóa bằng mã nhị phân đã cắt giảm, và `intra_mip_mpm_remainder` được mã hóa bằng mã độ dài cố định.

Tùy thuộc vào kích thước của khối mã hóa (CU), có thể hỗ trợ tối đa 35 chế độ MIP. Ví dụ, đối với CU có tối đa $(W, H) \leq 8$ và $W \cdot H < 32$, 35 chế độ khả dụng. Ngoài ra, 19 chế độ dự đoán và 11 chế độ dự đoán được sử dụng cho các CU có lần lượt là $\max(W, H) = 8$ và $\max(W, H) > 8$. Ngoài ra, một cặp chế độ (tức là hai chế độ) có thể chia sẻ chung một ma trận và vec-tơ bù để giảm yêu cầu bộ nhớ. Chế độ chia sẻ cụ thể được tính như sau. Ví dụ, đối với khối mã hóa 4×4 , chế độ 19 sử dụng ma trận chuyển vị của ma trận được gán cho chế độ 2.

$$m = \begin{cases} mode & \text{cho } W = H = 4 \text{ và } mode < 18 \\ mode - 17 & \text{cho } W = H = 4 \text{ và } mode \geq 18 \\ mode & \text{cho } \max(W, H) = 8 \text{ và } mode < 10 \\ mode - 9 & \text{cho } \max(W, H) = 8 \text{ và } mode \geq 10 \\ mode & \text{cho } \max(W, H) > 8 \text{ và } mode < 6 \\ mode - 5 & \text{cho } \max(W, H) > 8 \text{ và } mode \geq 6 \end{cases}$$

Khi có một khối mà MIP được áp dụng liên kế với khối mà dự đoán nội bộ thông thường khác ngoài MIP được áp dụng (sau đây được gọi là “khối thông thường”), có thể sử dụng bảng ánh xạ được xác định giữa các chế độ MIP và các chế độ thông thường để dẫn xuất MPM của khối thông thường. Bảng ánh xạ được sử dụng để dẫn xuất chế độ thông thường có các đặc điểm tương tự từ chế độ MIP của khối lân cận mà áp dụng MIP. Chế độ thông thường được dẫn xuất theo cách này được sử dụng để dẫn xuất MPM của khối thông thường. Tương tự, ngay cả khi MIP được áp dụng cho khối đã sắp xếp luma được sử dụng trong dẫn xuất DM chroma, chế độ thông thường của khối đã sắp xếp luma được dẫn xuất sử dụng bảng ánh xạ và chế độ dẫn xuất thông thường được sử dụng cho dẫn xuất DM chroma. Phương trình dưới đây thể hiện ánh xạ giữa các chế độ thông thường và các chế độ MIP sử dụng bảng ánh xạ.

$$predmode_{MIP} = map_regular_to_mip_{idx}[predmode_{regular}]$$

$$predmode_{regular} = map_mip_to_regular_{idx}[predmode_{MIP}]$$

Theo mô tả trên, trong bản thảo 5 VVC, khi một khối được dự đoán dựa trên MIP, các mẫu lân cận được tái tạo $bdry^{left}$ ở bên trái khối và các mẫu lân cận được tái tạo $bdry^{top}$ ở bên trên của khối luôn được sử dụng làm mẫu tham chiếu. Phương án này có thể làm giảm hiệu suất dự đoán nội bộ trong trường hợp kết cấu khối có các đặc điểm có hướng. Ngoài ra, trong bản thảo 5 VVC, để nội suy các mẫu trên cùng trong $predSamples$, tập mẫu được lấy mẫu xuống $bdry_{red}^{top}$ được phân bổ cho các giá trị của $predSamples[x][-1]$ như được thể hiện trong Fig.5, do đó làm cho quá trình nội suy phức tạp hơn cần thiết. Hơn nữa, trong bản thảo 5 VVC, vì danh sách MPM được sử dụng để báo hiệu cả chế độ MIP và chế độ thông thường, nên việc triển khai có thể rất phức tạp do yêu cầu nhiều kiểm tra và điều kiện như là yêu cầu ánh xạ giữa chế độ MIP và chế độ thông thường.

Theo quan điểm trên, sáng chế đề xuất một số kỹ thuật cải tiến cho mã hóa MPM. Theo một khía cạnh của sáng chế, sáng chế đề xuất phương án có thể làm giảm độ phức tạp của việc triển khai chế độ MIP và có tính đến đặc tính có hướng mà kết cấu của một khối có thể có.

Fig.6 là lưu đồ minh họa sơ đồ quy trình để dự đoán khối dựa trên MIP theo phương án thực hiện của sáng chế.

1. Giải mã chế độ MIP (S610)

Đối với CU được mã hóa trong chế độ dự đoán nội bộ, bộ giải mã video sẽ giải mã cờ cho biết kiểu dự đoán nội bộ có phải là MIP hay không. Khi MIP được áp dụng cho CU hiện tại, thì bộ giải mã video sẽ giải mã một phần tử cú pháp cho biết chế độ MIP được sử dụng trong CU hiện tại trong các chế độ MIP khả dụng.

Không giống như chế độ dự đoán nội bộ truyền thống (tức là chế độ dự đoán nội bộ thông thường), danh sách MPM có thể không được sử dụng để báo hiệu chế độ MIP. Thay vào đó, ví dụ, có thể sử dụng phần tử cú pháp (ví dụ, $intra_mip_mode$) có thể được mã hóa bằng mã nhị phân cắt ngắn và cho biết chế độ MIP được sử dụng trong CU hiện tại trong các chế độ MIP. Dưới đây là một phần của cú pháp đơn vị mã hóa mẫu được đề xuất dựa

trên bản thảo 5 VVC. Trong cú pháp bên dưới, với mục đích dễ hiểu hơn, sẽ tô đậm các phần tử có màu xám.

Bảng 7

<code>if(CuPredMode[x0][y0] == MODE_INTRA) {</code>
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>
<code>if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)</code>
<code>intra_bdpcm_flag[x0][y0]</code>
<code>if(intra_bdpcm_flag[x0][y0])</code>
<code>intra_bdpcm_dir_flag[x0][y0]</code>
<code>else {</code>
<code>if(sps_mip_enabled_flag && (Abs(Log2(cbWidth) - Log2(cbHeight)) <= 2) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)</code>
<code>intra_mip_flag[x0][y0]</code>
<code>if(intra_mip_flag[x0][y0])</code>
<code>intra_mip_mode[x0][y0]</code>

Khi `intra_mip_flag [x0][y0]` là 1, thì chỉ rằng kiểu dự đoán nội bộ của khối hiện tại là MIP. Khi `intra_mip_flag [x0][y0]` bằng 0, thì chỉ rằng kiểu dự đoán nội bộ của khối hiện tại là dự đoán nội bộ thông thường, không phải MIP. Khi không có `intra_mip_flag [x0][y0]`, thì có thể suy ra bằng 0. `intra_mip_mode [x0][y0]` chỉ định chế độ MIP được sử dụng cho khối hiện tại trong MIP.

Các mẫu lân cận của khối mã hóa hiện tại được sử dụng để tạo tín hiệu dự đoán có thể được xác định khác nhau tùy theo chế độ MIP. Theo một ví dụ, đối với khối mã hóa có kích thước 4×4, có sẵn 35 chế độ MIP, và các mẫu lân cận được sử dụng để tạo tín hiệu dự đoán có thể được xác định theo một dải các chế độ như sau.

<i>Các mẫu lân cận bên trái</i>	<i>cho mode < 12</i>
<i>Các mẫu lân cận bên trên</i>	<i>cho mode ≥ 24</i>
<i>Các mẫu lân cận bên trái và các mẫu lân cận bên trên</i>	<i>nếu không</i>

Khi các pixel của khối mã hóa hiện tại có đặc điểm ngang, thì có thể có lợi khi tạo tín hiệu dự đoán sử dụng các mẫu lân cận bên trái. Tương tự, khi các pixel của khối mã

hóa hiện tại có các đặc điểm dọc, thì có thể thuận lợi để tạo ra các tín hiệu dự đoán sử dụng các mẫu lân cận bên trên.

Để chọn chế độ MIP phù hợp với đặc tính có hướng của khối, bộ mã hóa video có thể được cấu hình để áp dụng các chế độ thông thường của dự đoán nội bộ cho khối và tính toán giá trị biến dạng của khối cho mỗi chế độ thông thường. Khi giá trị biến dạng của chế độ dự đoán ngang (Ví dụ, INTRA_ANGULAR_14 đến INTRA_ANGULAR_22) nhỏ hơn giá trị của các chế độ khác, hướng ngang có thể được xác định là hướng chủ đạo của khối mã hóa hiện tại. Ngược lại, khi giá trị biến dạng của chế độ dự đoán dọc (Ví dụ, INTRA_ANGULAR_46 đến INTRA_ANGULAR_54) nhỏ hơn so với các chế độ khác, hướng dọc có thể được xác định là hướng chủ đạo của khối mã hóa hiện tại.

Khi hướng ngang là hướng chủ đạo của khối mã hóa hiện tại, bộ mã hóa phân bổ một trong các chế độ MIP mà chỉ sử dụng các mẫu lân cận bên trái cho khối mã hóa hiện tại. Mặt khác, khi hướng dọc là hướng chính, bộ mã hóa có thể phân bổ một trong các chế độ MIP chỉ sử dụng bên trên các mẫu lân cận cho khối mã hóa hiện tại.

Sau đây, để sáng chế trở nên dễ hiểu, vec-tơ biên và phương pháp xác định các mẫu dự đoán từ vec-tơ biên sẽ được mô tả cho trường hợp sử dụng các mẫu lân cận bên trái và trường hợp sử dụng các mẫu lân cận bên trên. Khi sử dụng cả các mẫu lân cận bên trái và bên trên, quy trình về cơ bản có thể tương tự với quy trình được mô tả trong Fig.5.

2-1. Tạo vec-tơ biên - Sử dụng các mẫu lân cận bên trái (S620)

Fig.7a-7c là các sơ đồ khái niệm minh họa phương pháp mẫu để tạo một vec-tơ biên được nhập vào phép nhân vec-tơ ma trận sử dụng các mẫu lân cận bên trái. Ví dụ, như được minh họa trong Fig.7a, khi kích thước (chiều cao) của khối mã hóa hiện tại giống với kích thước của vec-tơ biên $bdry_{red}$, thì vec-tơ biên $bdry_{red}$ có cùng kích thước có thể được lấp đầy sử dụng tập mẫu lân cận bên trái $bdry_{left}$. Ví dụ, mỗi mẫu lân cận bên trái có thể được bao gồm trong mục nhập của vec-tơ biên.

Theo ví dụ khác, như được minh họa trong Fig.7b, vec-tơ biên có thể được lấp đầy bằng cách tính giá trị trung bình của hai pixel trong mỗi hàng sử dụng hai cột bên trái liền kề với khối mã hóa. Vẫn theo ví dụ khác, như được minh họa trong Fig.7c, vec-tơ biên $bdry_{red}$ có thể được lấp đầy sử dụng tập mẫu lấy mẫu xuống $bdry_{red}^{left}$ thu được từ tập mẫu

lân cận bên trái $bdry^{left}$. $bdry_{red}^{left}$ có thể thu được bằng cách tính giá trị trung bình $bdry^{left}$ cho mỗi hai mẫu.

Tùy thuộc vào kích thước của khối mã hóa, $bdry^{left}$ và $bdry_{red}^{left}$ có thể được sử dụng thích hợp. Ví dụ, như được thể hiện trong phương trình dưới đây, khi $H \leq 8$, thì $bdry^{left}$ được sử dụng. Nếu không, có thể sử dụng $bdry_{red}^{left}$.

$$bdry_{red} = \begin{cases} [bdry^{left}] & \text{cho } H \leq 8 \\ [bdry_{red}^{left}] & \text{nếu không} \end{cases}$$

2-2. Tạo vec-tơ biên-Sử dụng các mẫu lân cận bên trên (S620)

Fig.8a-8C là các sơ đồ khái niệm minh họa phương pháp mẫu để tạo vec-tơ biên được đưa vào phép nhân vec-tơ ma trận sử dụng các mẫu lân cận bên trên. Ví dụ, như được minh họa trong Fig.8a, khi kích thước (chiều rộng) của khối mã hóa hiện tại giống với kích thước của vec-tơ biên $bdry_{red}$, vec-tơ biên $bdry_{red}$ có cùng kích thước có thể được lấp đầy sử dụng tập mẫu lân cận bên trên. Ví dụ, mỗi mẫu lân cận bên trên có thể được bao gồm trong mục nhập của vec-tơ biên.

Theo ví dụ khác, như được minh họa trong Fig.8b, vec-tơ biên $bdry_{red}$ có thể được lấp đầy bằng cách tính trung bình hai pixel trong mỗi cột sử dụng hai hàng trên liền kề với khối mã hóa. Vẫn theo ví dụ khác, như được minh họa trong Fig.8c, vec-tơ biên $bdry_{red}$ có thể được lấp đầy sử dụng tập mẫu được lấy mẫu xuống $bdry_{red}^{top}$ thu được từ tập mẫu lân cận bên trên $bdry_{top}$. Ngoài ra, có thể thu được $bdry_{red}^{top}$ bằng cách lấy trung bình của hai mẫu $bdry_{top}$.

Tùy thuộc vào kích thước của khối mã hóa, $bdry_{top}$ và $bdry_{red}^{top}$ có thể được sử dụng thích hợp. Ví dụ, như được thể hiện trong phương trình dưới đây, khi $H \leq 8$, $bdry_{top}$ có thể được sử dụng. Nếu không, $bdry_{red}^{top}$ có thể được sử dụng.

$$bdry_{red} = \begin{cases} [bdry_{top}] & \text{cho } H \leq 8 \\ [bdry_{red}^{top}] & \text{nếu không} \end{cases}$$

3. Phép nhân ma trận-vec-tơ và phép cộng bù (S630)

Bộ giải mã video được cấu hình để thực hiện phép nhân vec-tơ ma trận trên vec-tơ biên $bdry_{red}$ và thêm vec-tơ bù vào đó. Ma trận A và vec-tơ bù b có thể được xác định theo chế độ đã giải mã MIP k . Kết quả là, tạo tín hiệu dự đoán của khối ban đầu hoặc tín hiệu dự đoán được lấy mẫu phụ $pred_{red}$. $pred_{red}$ được tính như sau.

$$pred_{red} = A \cdot bdry_{red} + b$$

Ở đây, kích thước của $pred_{red}$ là $W_{red} \times H_{red}$

Khi vec-tơ biên nhập vào phép nhân vec-tơ ma trận được tạo sử dụng các mẫu lân cận bên trái, W_{red} và H_{red} có thể được xác định theo kích thước của khối mã hóa như sau.

$$W_{red} = \begin{cases} 4 & \text{cho } \max(W, H) \leq 8 \\ \min(W, 8) & \text{cho } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{cho } H \leq 8 \\ 8 & \text{cho } H > 8 \end{cases}$$

Khi vec-tơ biên nhập vào phép nhân vec-tơ ma trận được tạo sử dụng các mẫu lân cận bên trên, W_{red} và H_{red} có thể được xác định theo kích thước của khối mã hóa như sau.

$$W_{red} = \begin{cases} 4 & \text{cho } W \leq 8 \\ 8 & \text{cho } W > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{cho } \max(W, H) \leq 8 \\ \min(H, 8) & \text{cho } \max(W, H) > 8 \end{cases}$$

Cần phải nội suy khi kích thước của $pred_{red}$ được tạo bị nhỏ hơn số lượng mẫu của khối mã hóa.

4. Nội suy tuyến tính (S640)

Fig.9 thể hiện các mẫu lân cận bên trái, vec-tơ biên và các mẫu dự đoán thu được từ đó. Theo Fig.9, các pixel được biểu thị bằng hình tam giác trong khối dự đoán $predSamples[x][y]$ cho khối mã hóa hiện tại có các giá trị được phân bổ từ $pred_{red}$ tạo được. Tham khảo Fig.9, các phương pháp nội suy khác nhau có thể được sử dụng để xác định các pixel trống trong khối dự đoán $predSamples[x][y]$.

Ví dụ, như trong trường hợp pixel 1, giá trị của pixel trống có thể được xác định sử dụng mẫu lân cận bên trái, mẫu lân cận bên trên, và giá trị của $pred_{red}$ phân bổ bên dưới

pixel trống. Theo ví dụ khác, như trong trường hợp pixel 2, giá trị pixel trống có thể được xác định sử dụng mẫu lân cận bên trên và giá trị của $pred_{red}$ phân bổ bên dưới pixel trống. Theo ví dụ khác, như trong trường hợp pixel 3, giá trị pixel trống có thể được xác định sử dụng hai giá trị của $pred_{red}$ được phân bổ ở bên trên và bên dưới pixel trống. Theo ví dụ khác, như trong trường hợp pixel 4, giá trị pixel trống có thể được xác định sử dụng mẫu lân cận bên trái, và hai giá trị của $pred_{red}$ phân bổ bên trên và bên dưới pixel trống.

Fig.10 thể hiện các mẫu lân cận bên trên, vec-tơ biên, và các mẫu dự đoán thu được từ đó. Theo Fig.10, các pixel được biểu thị bằng hình tam giác trong khối dự đoán $predSamples[x][y]$ cho khối mã hóa hiện tại có các giá trị được phân bổ từ $pred_{red}$ đã tạo.

Tham khảo Fig.10, các phương pháp nội suy khác nhau có thể được sử dụng để xác định các pixel trống trong khối dự đoán $predSamples[x][y]$. Ví dụ, như trong trường hợp pixel 1, giá trị pixel trống có thể được xác định sử dụng mẫu lân cận bên trái, mẫu lân cận bên trên, và giá trị của $pred_{red}$ phân bổ bên dưới pixel trống. Theo ví dụ khác, như trong trường hợp pixel 2, giá trị pixel trống có thể được xác định sử dụng mẫu lân cận bên trái, hai giá trị của $pred_{red}$ được phân bổ bên trên và bên dưới pixel trống. Theo ví dụ khác, như trong trường hợp pixel 3, giá trị pixel trống có thể được xác định sử dụng hai giá trị của $pred_{red}$ được phân bổ bên trên và bên dưới pixel trống. Theo ví dụ khác, như trong trường hợp pixel 4, giá trị pixel trống có thể được xác định sử dụng mẫu lân cận bên trên và giá trị của $pred_{red}$ phân bổ bên dưới pixel trống.

Không giống như các ví dụ được minh họa trong Fig.9 và 10, để xác định các pixel trống trong khối dự đoán $predSamples[x][y]$, các mẫu tham chiếu ban đầu $bdry_{top}$ có thể được phân bổ cho $predSamples[x][-1]$, và các mẫu tham chiếu ban đầu bên trái $bdry_{left}$ có thể được phân bổ cho $predSamples[-1][y]$. Trong phương án thực hiện nêu trên, đề xuất sơ đồ sử dụng có chọn lọc các mẫu lân cận của khối hiện tại theo chế độ MIP sao cho hướng của khối có thể được xem xét khi tạo khối dự đoán dựa trên MIP.

Theo một khía cạnh khác của sáng chế, thay vì thay đổi tập mẫu lân cận được sử dụng để tạo vec-tơ biên theo chế độ MIP, có thể sửa đổi các giá trị của ma trận A và phần bù b được áp dụng cho vec-tơ biên.

Ma trận A và phần bù b được áp dụng cho phép nhân vec-tơ ma trận và phép cộng bù cho vec-tơ biên có thể được chia thành hai phần như sau.

$$pred_{red} = A \begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + (b)$$

$$pred_{red} = (A1|A2) \begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \end{pmatrix}$$

Khi giá trị của $A2$ và $b2$ xấp xỉ 0, tín hiệu dự đoán $pred_{red}$ chủ yếu bị ảnh hưởng bởi pixel lân cận bên trái $bdry_{red}^{left}$, có thể hữu ích cho khối mà trong đó kết cấu của khối có hướng nằm ngang. Khi các giá trị của $A1$ và $b1$ xấp xỉ 0, thì tín hiệu dự đoán $pred_{red}$ chủ yếu bị ảnh hưởng bởi pixel lân cận bên trên $bdry_{red}^{top}$, có thể hữu ích cho khối mà trong đó kết cấu của khối có hướng dọc.

Bộ mã hóa và giải mã video có thể sử dụng các giá trị đã sửa đổi của ma trận A và phần bù b để áp dụng cho vec-tơ biên theo chế độ MIP của khối hiện tại. Ví dụ, đối với 4×4 CU, 35 chế độ MIP có thể khả dụng, và các giá trị của ma trận A và phần bù b được áp dụng cho vec-tơ biên có thể được sửa đổi theo phạm vi của chế độ.

$$\begin{cases} A2 \text{ và } b2 \cong 0 & \text{cho } mode < 12 \\ A1 \text{ và } b2 \cong 0 & \text{cho } mode \geq 24 \\ \text{Sử dụng } A \text{ và } b & \text{nếu không} \end{cases}$$

Theo mô tả ở trên, bản thảo 5 VVC hỗ trợ tới 35 chế độ MIP tùy theo kích thước và hình dạng của CU. Ví dụ, 35 chế độ khả dụng cho CU với $\max(W, H) \leq 8$ && $W \cdot H < 32$, 19 chế độ dự đoán khả dụng cho CU có $\max(W, H) = 8$, và 11 chế độ dự đoán khả dụng cho CU có $\max(W, H) > 8$. Ngoài ra, một cặp chế độ (hai chế độ) có thể chia sẻ chung một ma trận và vec-tơ bù để giảm yêu cầu bộ nhớ. Ví dụ, đối với khối mã hóa 4×4 , chế độ 19 sử dụng ma trận chuyển vị của ma trận được gán cho chế độ 2. Hơn nữa, bằng cách liên kết $bdry_{red}^{top}$ và $bdry_{red}^{left}$ theo thứ tự được xác định theo chế độ MIP và kích thước của khối, tạo được vec-tơ biên $bdry_{red}$ để nhập vào phép nhân ma trận vec-tơ.

Sáng chế đề xuất phương pháp nâng cao về cơ bản có thể đạt được cấp hiệu quả mã hóa tương đương trong khi giảm được độ phức tạp mã hóa. Theo khía cạnh khác của sáng chế, thay vì dùng sơ đồ hiện có mà trong đó chế độ sử dụng phép chuyển vị của ma trận được sử dụng bởi chế độ khác, có thể sử dụng sơ đồ mới được cấu hình để thay đổi thứ tự

ghép nối $bdry_{red}^{top}$ và $bdry_{red}^{left}$ tạo thành vec-tơ biên $bdry_{red}$ dùng cho phép nhân vec-tơ ma trận cho từng chế độ. Bộ mã hóa video có thể báo hiệu một phần tử cú pháp cho biết thứ tự ghép nối của $bdry_{red}^{top}$ và $bdry_{red}^{left}$ để tạo thành vec-tơ biên được mã hóa cho mỗi chế độ. Theo sơ đồ này, số lượng chế độ MIP khả dụng có thể giảm một nửa so với phương pháp thông thường, và có thể giảm độ phức tạp mã hóa về mặt tạo vec-tơ biên và nhân ma trận vec-tơ.

Như được mô tả liên quan đến cấu trúc cú pháp của Bảng 5, `transform_skip_flag`, là phần tử cú pháp liên quan đến biến đổi bỏ qua, được báo hiệu cho mỗi đơn vị biến đổi thỏa mãn các điều kiện nhất định. Khi `transform_skip_flag` không phải là TRUE, `tu_mts_idx`, là phần tử cú pháp liên quan đến lựa chọn nhiều biến đổi, phải được truyền theo cờ (`sps_explicit_mts_inter_enabled_flag`, `sps_explicit_mts_intra_enabled_flag`) trong SPS, và do đó gánh nặng được áp dụng liên quan đến số lượng bit được truyền trong dòng bit.

Tuy nhiên, lựa chọn nhiều biến đổi và bỏ qua biến đổi có liên quan chặt chẽ đến các đặc tính của tín hiệu dư thừa, và do đó có thể không hiệu quả khi luôn báo hiệu các phần tử cú pháp liên quan đến biến đổi bất kể chế độ dự đoán được sử dụng trong khối. Cần lưu ý rằng xem xét quá trình MIP bao gồm phép tính trung bình và phép nội suy, tín hiệu dư thừa của khối mà MIP được áp dụng có các thành phần tần số thấp chiếm ưu thế trong miền biến đổi, và có khả năng MIP sẽ không được áp dụng cho khối có mối tương quan thấp giữa các pixel, chẳng hạn như khối chứa chuyển động và kết cấu phức tạp. Do đó, có khả năng bỏ qua biến đổi không được áp dụng cho khối mà sử dụng MIP.

Theo quan điểm trên, phương án luôn báo hiệu `tu_mts_idx`, là phần tử cú pháp liên quan đến lựa chọn đa biến đổi, và `transform_skip_flag`, là phần tử cú pháp liên quan đến bỏ qua biến đổi bất kể MIP có được áp dụng cho khối hay không, phương án theo cách này không hiệu quả lắm. Theo một khía cạnh của sáng chế, báo hiệu các phần tử cú pháp liên quan đến biến đổi có thể được bỏ qua tùy thuộc vào MIP có được áp dụng hay không, và các phần tử cú pháp bị bỏ qua có thể được bộ giải mã video suy luận ngầm.

Theo phương án thực hiện, kỹ thuật MTS có thể được áp dụng ngầm theo cờ MIP. Như được hiển thị trong bảng bên dưới, `transform_skip_flag` và `tu_mts_idx` chỉ có thể được mã hoá khi `intra_mip_flag` không phải là TRUE. Khi `intra_mip_flag` là TRUE, thì

transform_skip_flag được suy ra là 0, tu_mts_idx cũng được suy ra là 0, và áp dụng MTS ngầm được suy ra. Theo đó, các nhân biến đổi theo hướng ngang và hướng dọc có thể được xác định tùy theo kích thước khối. Trong cú pháp bên dưới, màu xám của các phần tử được sử dụng để chỉ ra những thay đổi tiềm năng trong cú pháp hoặc cung cấp sự hiểu biết về nó.

Bảng 8

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	Descriptor
(omitted)	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]) && (!intra_mip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
(omitted)	

Theo một phương án thực hiện khác, kỹ thuật MTS có thể được áp dụng rõ ràng theo cờ MIP. Như được thể hiện trong bảng dưới đây, transform_skip_flag và tu_mts_idx chỉ có thể được mã hoá khi intra_mip_flag không phải là TRUE. Khi intra_mip_flag là TRUE, thì transform_skip_flag được suy ra là 0, tu_mts_idx được báo hiệu rõ ràng, và MTS rõ ràng được áp dụng. Trong cú pháp bên dưới, màu xám của các phần tử dùng để chỉ ra những thay đổi tiềm năng trong cú pháp hoặc cung cấp sự hiểu biết về nó.

Bảng 9

<code>transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {</code>	Descriptor
<code>(omitted)</code>	
<code>if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {</code>	
<code>if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))</code>	
<code>transform_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))</code>	
<code>tu_mts_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>if(tu_cbf_luma[x0][y0]) {</code>	
<code>if(!transform_skip_flag[x0][y0])</code>	
<code>residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)</code>	
<code>else</code>	
<code>residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)</code>	
<code>}</code>	
<code>(omitted)</code>	

Theo phương án thực hiện khả thi khác, MIP có thể không được áp dụng rõ ràng cho khối có mối tương quan thấp giữa các pixel, chẳng hạn như khối có kích thước nhỏ chứa chuyển động và kết cấu phức tạp. Bỏ qua biến đổi (Transform Skip, TS) có thể không được áp dụng cho CU 4x4 mà đã áp dụng MIP. Ngoài ra, MIP có thể không được áp dụng cho CU 4x4 mà đã áp dụng TS. Ngoài ra, MIP cùng với TS có thể không được áp dụng cho CU 4x4.

Theo phương án thông thường, mã hóa dự đoán nội bộ có thể sử dụng chế độ đúng nhất (MPM). Ví dụ, trong HEVC, danh sách của ba MPM được cấu hình từ các chế độ dự đoán nội bộ của các khối bên trái và bên trên. Hạn chế của phương pháp này là có nhiều chế độ (chế độ nội bộ khác với MPM) thuộc về các chế độ không phải MPM cần được mã hóa bằng nhiều bit hơn. Một số phương pháp đã được đề xuất để gia tăng số lượng MPM lên 3 mục nhập trở lên (ví dụ, chế độ 6 MPM). Tuy nhiên, cấu hình danh sách MPM theo

kiểu này với nhiều mục nhập hơn có thể yêu cầu nhiều kiểm tra và điều kiện hơn, điều này có thể làm cho việc triển khai trở nên phức tạp hơn.

Theo mô tả ở trên, trong bản thảo 5 VVC, vì mỗi danh sách MPM được sử dụng để báo hiệu chế độ MIP và chế độ thông thường, nên việc triển khai có thể phức tạp do yêu cầu nhiều kiểm tra và điều kiện, chẳng hạn như yêu cầu ánh xạ giữa chế độ MIP và chế độ thông thường để cấu hình danh sách MIP.

Để duy trì độ phức tạp của cấu hình danh sách MPM ở mức thấp, danh sách MPM bao gồm sáu ứng viên MPM có thể được cấu hình sử dụng các chế độ dự đoán nội bộ của khối lân cận bên trái và khối lân cận bên trên liên kết với khối hiện tại. Các ứng viên MPM có thể bao gồm chế độ dự đoán nội bộ mặc định (ví dụ, chế độ Planar), chế độ dự đoán nội bộ của các khối lân cận, và chế độ dự đoán nội bộ lấy từ chế độ dự đoán nội bộ của các khối lân cận. Khi chế độ dự đoán nội bộ của khối lân cận không được sử dụng (ví dụ, khi khối lân cận được dự đoán liên, hoặc khối lân cận nằm trong một lát cắt khác hoặc một ô khác), thì chế độ dự đoán nội bộ của khối lân cận có thể được đặt thành Planar.

Theo kiểu chế độ dự đoán nội bộ của chế độ (Left) của khối bên trái và chế độ (Above) của khối bên trên, được chia thành 4 trường hợp. Khi Left và Above khác nhau, và hai chế độ là chế độ có hướng, có thể thêm 4 trường hợp khác tùy theo sự chênh lệch của Left và Above để tạo danh sách MPM. Trong bảng dưới đây, Max biểu thị chế độ lớn hơn giữa Left và Above, và MIN biểu thị chế độ nhỏ hơn giữa Left và Above.

Bảng 10

Điều kiện	Điều kiện chi tiết	Các chế độ MPM
Chế độ Left và chế độ Above là chế độ có hướng và giống nhau		{Phẳng, Left, Left-1, Left+1, Left-2, Left+2}
Chế độ Left và chế độ Above khác nhau, và hai chế độ đều là chế độ có hướng	$\text{Max-Min} = 1$	{Phẳng, Left, Above, Min-1, Max+1, Min-2}
	$\text{Max-Min} \geq 62$	{Phẳng, Left, Above, Min+1, Max-1, Min+2}
	$\text{Max-Min} \geq 2$	{Phẳng, Left, Above, Min+1, Min-1, Max+1}
	$2 < \text{Max-Min} < 62$	{Phẳng, Left, Above, Min-1, Min+1, Max-1}
Chế độ Left và chế độ Above khác nhau, và chỉ một chế độ là có hướng		{Phẳng, Max, Max-1, Max+1, Max-2, Max+2}
Cả hai chế độ Left và chế độ Above là chế độ vô hướng (nghĩa là, Phẳng hoặc DC)		{Phẳng, DC, Góc50, Góc18, Góc46, Góc54}

Do các đặc điểm của kỹ thuật MIP bao gồm phép trung bình và phép nội suy, tín hiệu dư thừa của khối mà MIP được áp dụng có thể có thành phần tần số thấp chiếm ưu thế trong miền biến đổi. Lưu ý rằng đặc tính dư thừa này tương tự như tín hiệu dư thừa của khối mà áp dụng chế độ phẳng hoặc chế độ DC. Do đó, trong việc lấy danh sách MPM các khối được mã hóa ở chế độ dự đoán nội bộ thông thường, có thể hữu ích khi xem xét tính tương đồng của các tín hiệu dư thừa này.

Theo một khía cạnh của sáng chế, khi lấy danh sách MPM cho một khối (tức là khối thông thường) được mã hóa ở chế độ dự đoán nội bộ thông thường, khi khối lân cận được mã hóa ở chế độ MIP, chế độ dự đoán của khối lân cận có thể được coi là chế độ phẳng (hoặc chế độ DC). Ví dụ, khi chế độ MIP được áp dụng cho khối lân cận, chế độ phẳng (hoặc chế độ DC) có thể được thêm vào danh sách MPM thay cho chế độ MIP của khối lân cận.

Tương tự, ngay cả khi chế độ trực tiếp (Direct Modem, DM) chroma, nếu MIP được áp dụng cho một khối đã sắp xếp luma, thì chế độ dự đoán nội bộ của khối luma có thể được coi là chế độ phẳng (hoặc chế độ DC), thay vì sử dụng bảng ánh xạ giữa chế độ MIP và chế độ thông thường. Theo đó, bộ mã hóa phân tích phần tử cú pháp chỉ định chế độ dự đoán nội bộ cho một khối chroma, và phần tử cú pháp có thể chỉ ra rằng chế độ dự đoán nội bộ của khối chroma sử dụng chế độ dự đoán nội bộ của khối đã sắp xếp luma. Trong trường hợp đó, khi MIP được áp dụng cho khối đã sắp xếp luma, thì chế độ dự đoán nội bộ của khối luma có thể được coi là chế độ phẳng (hoặc chế độ DC). Nói cách khác, khi MIP được áp dụng cho khối đã sắp xếp luma ở chế độ trực tiếp (DM) chroma, thì có thể xác định rằng chế độ dự đoán nội bộ của khối chroma là chế độ phẳng (hoặc chế độ DC).

Bộ mã hóa video có thể báo hiệu cờ 1 bit (ví dụ, `mpm_flag`) cho biết liệu chế độ dự đoán nội bộ của khối hiện tại có tương ứng với MPM hay không. Điển hình là, khi chế độ dự đoán nội bộ của khối hiện tại tương ứng với MPM, chỉ số MPM cho biết một trong 6 MPM sẽ được báo hiệu bổ sung. Lưu ý rằng trong bảng trên, chế độ phẳng luôn được bao gồm trong danh sách MPM. Do đó, bộ mã hóa có thể hiệu quả khi báo hiệu rõ ràng liệu chế độ dự đoán nội bộ của khối hiện tại có phải là chế độ phẳng hay không (ví dụ, sử dụng cờ 1 bit) và để báo hiệu bổ sung chỉ số MPM cho biết một trong năm MPM khác khi chế độ dự đoán nội bộ của khối hiện tại tương tự với một trong năm MPM khác. Khi chế độ dự đoán nội bộ của khối hiện tại không tương ứng với bất kỳ MPM nào, thì phần tử cú pháp chỉ ra một trong 61 không phải MPM còn lại ngoại trừ 6 MPM có thể được mã hóa bằng mã nhị phân cắt ngắn.

Fig.11 là lưu đồ minh họa phương pháp giải mã dữ liệu video theo phương án thực hiện của sáng chế. Bộ giải mã video có thể được cấu hình để giải mã phần tử cú pháp cho biết kiểu dự đoán nội bộ của khối hiện tại của dữ liệu video từ dòng bit (S1110). Các kiểu dự đoán nội bộ bao gồm dự đoán nội bộ dựa trên ma trận (MIP) và dự đoán nội bộ thông thường. Phần tử cú pháp có thể là mã nhị phân cắt ngắn chỉ định một trong nhiều chế độ MIP được cấp phép cho kích thước và hình dạng của khối hiện tại.

Bộ giải mã video có thể được cấu hình để tạo khối dự đoán cho khối hiện tại bằng cách thực hiện có chọn lọc MIP hoặc dự đoán nội bộ thông thường dựa trên kiểu dự đoán nội bộ của khối hiện tại. Là một phần của việc tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ thông thường, bộ giải mã video có thể được cấu hình để

thực hiện các hoạt động sau (S1120 đến S1140). Bộ giải mã video có thể cấu hình danh sách MPM cho khối hiện tại bằng cách lấy ra các ứng viên chế độ đúng nhất (MPM) dựa trên các chế độ dự đoán nội bộ thông thường của các khối lân cận liền kề với khối hiện tại (S1120), và lấy chế độ dự đoán nội bộ thông thường cho khối hiện tại dựa trên danh sách MPM (S1130). Khi lấy các ứng viên MPM dựa trên các chế độ dự đoán nội bộ thông thường của các khối lân cận, bộ giải mã video có thể được cấu hình để đặt (coi) chế độ dự đoán nội bộ thông thường của khối lân cận là chế độ PLANAR khi kiểu dự đoán nội bộ của khối lân cận là dự đoán nội bộ dựa trên ma trận. Bộ giải mã có thể được cấu hình để tạo khối dự đoán cho khối hiện tại dựa trên chế độ dự đoán nội bộ thông thường của khối hiện tại (S1140).

Là một phần của việc tạo khối dự đoán cho khối hiện tại bằng cách thực hiện dự đoán nội bộ dựa trên ma trận, bộ giải mã video có thể được cấu hình để thực hiện các thao tác sau (S1121 đến S1151). Bộ giải mã video có thể được cấu hình để giải mã phần tử cú pháp cho biết chế độ dự đoán nội bộ dựa trên ma trận cho khối hiện tại từ dòng bit để xác định chế độ dự đoán nội bộ dựa trên ma trận cho khối hiện tại (S1121). Bộ giải mã video có thể được cấu hình để lấy vec-tơ biên sử dụng các mẫu lân cận liền kề với khối hiện tại dựa trên kích thước và hình dạng của khối hiện tại (S1131), và tạo các mẫu dự đoán cho khối hiện tại dựa trên phép nhân vec-tơ ma trận giữa một ma trận được xác định trước cho chế độ dự đoán nội bộ dựa trên ma trận cho khối hiện tại và vec-tơ biên (S1141). Bộ giải mã video có thể được cấu hình để lấy khối dự đoán cho khối hiện tại bằng cách thực hiện nội suy tuyến tính và cắt dựa trên các mẫu được dự đoán (S1151).

Bộ giải mã video có thể được cấu hình để lấy vec-tơ biên từ một hoặc hai tập hợp được chọn giữa tập hợp các mẫu lân cận bên trái và tập hợp các mẫu lân cận bên trên liền kề với khối hiện tại, tùy thuộc vào chế độ dự đoán nội bộ dựa trên ma trận cho khối hiện tại. Bộ giải mã video được cấu hình để giải mã, từ dòng bit, phần tử cú pháp chỉ thứ tự ghép nối của các mục nhập thứ nhất của vec-tơ biên lấy từ các mẫu lân cận bên trái liền kề với khối hiện tại và các mục nhập thứ hai của vec-tơ biên lấy từ các mẫu lân cận bên trên liền kề với khối hiện tại. Bộ giải mã video có thể được cấu hình để tạo vec-tơ biên bằng cách ghép nối các mục nhập thứ nhất và các mục nhập thứ hai theo thứ tự ghép nối được chỉ định.

Các mục nhập của vec-tơ biên có thể được lấp đầy bằng các giá trị được lấy mẫu xuống từ các mẫu lân cận bên trái liền kề với khối hiện tại hoặc với các mẫu lân cận bên trái liền kề với khối hiện tại theo kích thước và hình dạng của khối hiện tại. Bộ giải mã video có thể phân bổ các mẫu dự đoán cho các vị trí trong khối dự đoán để lấy khối dự đoán cho khối hiện tại dựa trên các mẫu được dự đoán. Để tạo ra các giá trị mẫu dự đoán cho các vị trí mà mẫu dự đoán không được phân bổ trong khối dự đoán, có thể thực hiện phép nội suy tuyến tính trên các mẫu dự đoán, các mẫu lân cận bên trái liền kề với khối hiện tại, và các mẫu lân cận bên trên liền kề với khối hiện tại.

Cần hiểu rằng các phương án ví dụ được mô tả ở trên có thể được thực hiện theo nhiều cách khác nhau. Các chức năng hoặc phương pháp được mô tả trong một hoặc nhiều ví dụ có thể được triển khai trong phần cứng, phần mềm, chương trình cơ sở hoặc bất kỳ sự kết hợp nào của chúng. Cần hiểu rằng các thành phần chức năng được mô tả ở đây đã được đặt là “bộ phận” để nhấn mạnh hơn nữa tính độc lập trong việc triển khai chúng.

Các chức năng hoặc phương pháp khác nhau được mô tả trong sáng chế có thể được thực hiện với các hướng dẫn được lưu trữ trong một phương tiện ghi tạm thời có thể được đọc và thực hiện bởi một hoặc nhiều bộ xử lý. Phương tiện ghi tạm thời bao gồm, ví dụ, tất cả các loại thiết bị ghi mà trong đó dữ liệu được lưu trữ ở dạng có thể đọc được bởi hệ thống máy tính. Ví dụ, phương tiện ghi tạm thời bao gồm phương tiện lưu trữ như bộ nhớ chỉ đọc có thể lập trình để xóa (Erasable Programmable Read Only Memory, EPROM), ổ đĩa flash, ổ đĩa quang, ổ cứng từ tính, và ổ đĩa thể rắn (Solid State Drives, SSD).

Dù sáng chế đã được mô tả ở trên thông qua các phương án thực hiện mẫu, nhưng các phương án thực hiện được mô tả chỉ mang tính minh họa mà không giới hạn phạm vi của sáng chế, người có trình độ trong cùng lĩnh vực kỹ thuật có thể thực hiện nhiều biến thể và sửa đổi khác mà vẫn thuộc phạm vi bảo hộ của sáng chế. Các phương án thực hiện mẫu của sáng chế được mô tả nhằm mục đích minh họa và làm sáng tỏ sáng chế. Theo đó, người có trình độ trung bình trong cùng lĩnh vực kỹ thuật cần hiểu rõ rằng phạm vi của sáng chế không giới hạn ở các phương án chính xác được mô tả ở trên mà phạm vi của sáng chế chỉ giới hạn bởi các yêu cầu bảo hộ và những yêu cầu tương đương.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dữ liệu video, bao gồm:

giải mã, từ dòng bit, phần tử cú pháp cho biết kiểu dự đoán nội bộ của khối luma hiện tại của dữ liệu video, kiểu dự đoán nội bộ được chỉ ra trong các dự đoán nội bộ dựa trên ma trận (Matrix Based Intra-Prediction, MIP) và dự đoán nội bộ thông thường; và

tạo khối dự đoán cho khối luma hiện tại bằng cách thực hiện có chọn lọc MIP hoặc dự đoán nội bộ thông thường dựa trên kiểu dự đoán nội bộ của khối luma hiện tại được chỉ ra bởi phần tử cú pháp,

trong đó việc tạo khối dự đoán cho khối luma hiện tại bằng cách thực hiện dự đoán nội bộ thông thường bao gồm:

lấy các ứng viên chế độ đúng nhất (Most Probable Mode, MPM) dựa trên chế độ dự đoán nội bộ thông thường của từng khối lân cận liền kề với khối luma hiện tại và cấu hình danh sách MPM cho khối luma hiện tại; và

lấy một chế độ dự đoán nội bộ thông thường cho khối luma hiện tại dựa trên danh sách MPM,

trong đó, để xác định rằng kiểu dự đoán nội bộ của khối lân cận là MIP, chế độ dự đoán nội bộ thông thường của khối lân cận được đặt là chế độ PLANAR, và

trong đó, khi MIP được thực hiện trên khối luma hiện tại, và chế độ dự đoán nội bộ của khối luma hiện tại được sử dụng làm chế độ dự đoán nội bộ của khối chroma tương ứng với khối luma hiện tại, chế độ dự đoán nội bộ thông thường cho khối luma hiện tại được đặt là chế độ PLANAR,

trong đó quá trình tạo khối dự đoán cho khối luma hiện tại bằng cách thực hiện MIP bao gồm:

giải mã, từ dòng bit, phần tử cú pháp chỉ ra chế độ MIP cho khối luma hiện tại;

lấy vec-tơ biên sử dụng các mẫu lân cận liền kề với khối luma hiện tại dựa trên chiều rộng và chiều cao của khối luma hiện tại;

tạo các mẫu dự đoán cho khối luma hiện tại dựa trên phép nhân vec-tơ ma trận giữa vec-tơ biên và ma trận được xác định trước cho chế độ MIP; và

lấy khối dự đoán cho khối luma hiện tại dựa trên các mẫu dự đoán, và

trong đó lấy vec-tơ biên sử dụng các mẫu lân cận liền kề với khối luma hiện tại bao gồm:

giải mã, từ dòng bit, phần tử cú pháp chỉ ra thứ tự ghép nối của các mục nhập thứ nhất của vec-tơ biên lấy từ các mẫu lân cận bên trái liền kề với khối luma hiện tại và các mục nhập thứ hai của vec-tơ biên lấy từ các mẫu lân cận bên trên liền kề với khối luma hiện tại; và

ghép nối các mục nhập thứ nhất và các mục nhập thứ hai theo thứ tự ghép nối và từ đó tạo ra vec-tơ biên.

2. Phương pháp theo điểm 1, trong đó phần tử cú pháp chỉ chế độ MIP cho khối luma hiện tại là mã nhị phân cắt ngắn chỉ định một trong nhiều chế độ MIP được cấp phép cho chiều rộng và chiều cao của khối luma hiện tại.

3. Phương pháp theo điểm 2, trong đó các mục nhập của vec-tơ biên được lấp đầy bằng các giá trị được lấy mẫu xuống từ các mẫu lân cận bên trái liền kề với khối luma hiện tại hoặc với các mẫu lân cận bên trái liền kề với khối luma hiện tại theo chiều rộng và chiều cao của khối luma hiện tại.

4. Phương pháp theo điểm 2, trong đó lấy khối dự đoán cho khối luma hiện tại dựa trên các mẫu được dự đoán bao gồm:

phân bổ các mẫu dự đoán vào các vị trí trong khối dự đoán; và

tạo các giá trị mẫu dự đoán cho các vị trí mà các mẫu dự đoán không được phân bổ trong khối dự đoán, sử dụng nội suy tuyến tính cho các mẫu dự đoán, các mẫu lân cận bên trái liền kề với khối luma hiện tại, và các mẫu lân cận bên trên liền kề với khối luma hiện tại.

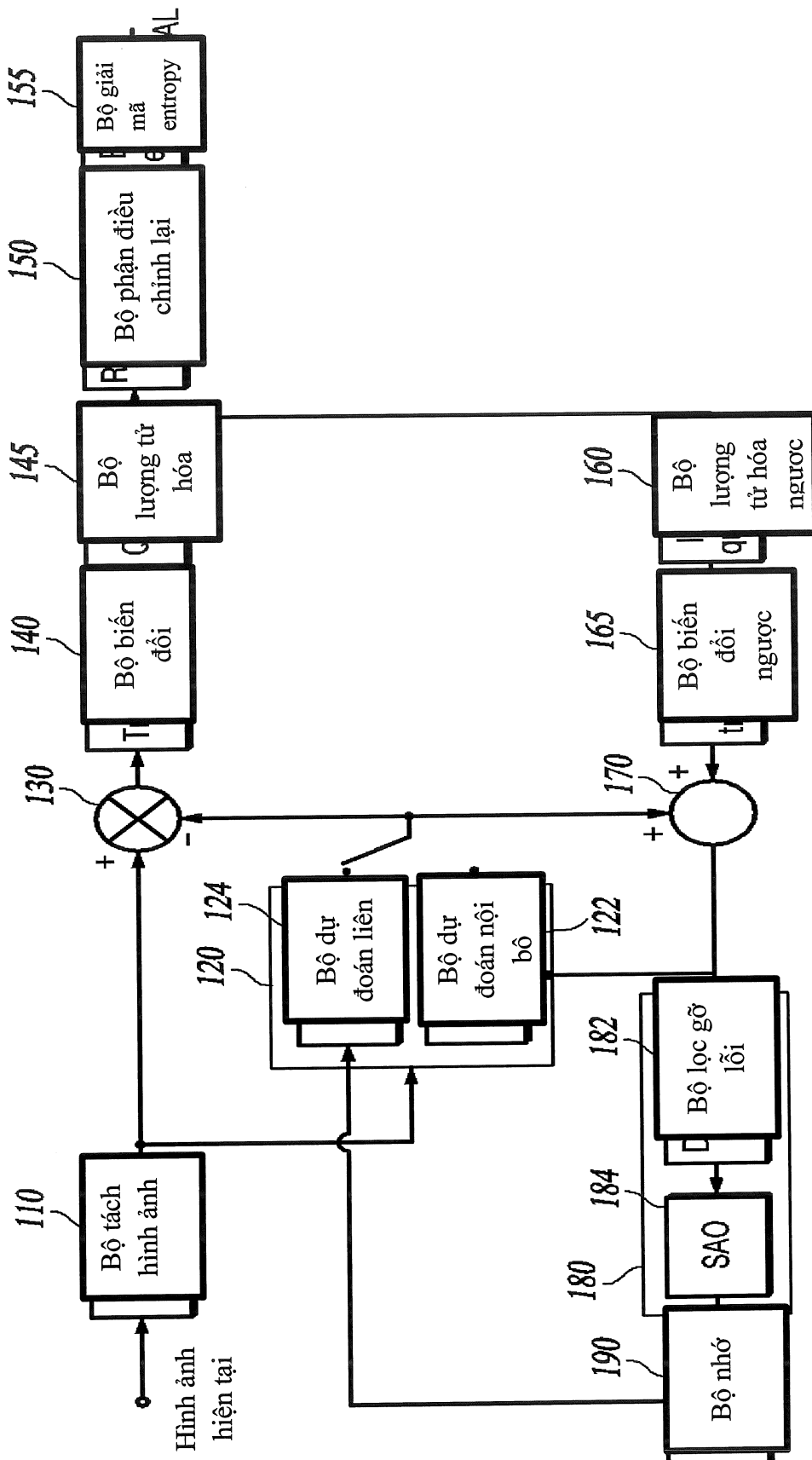


Fig.1

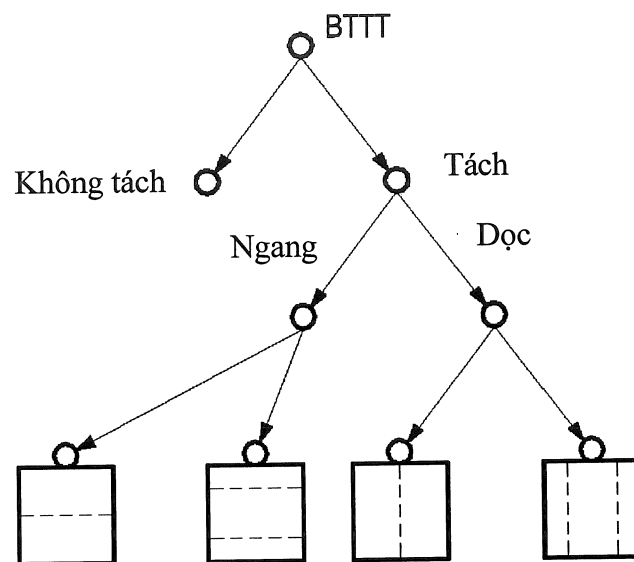
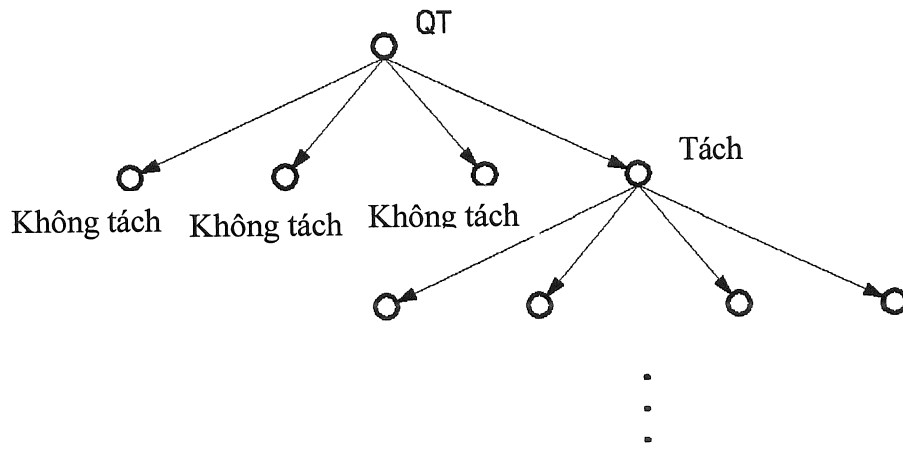


Fig.2

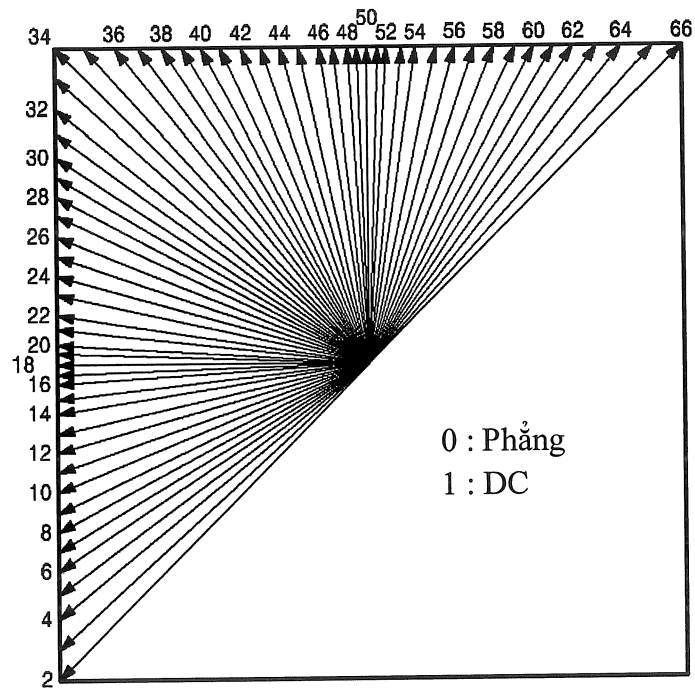


Fig.3a

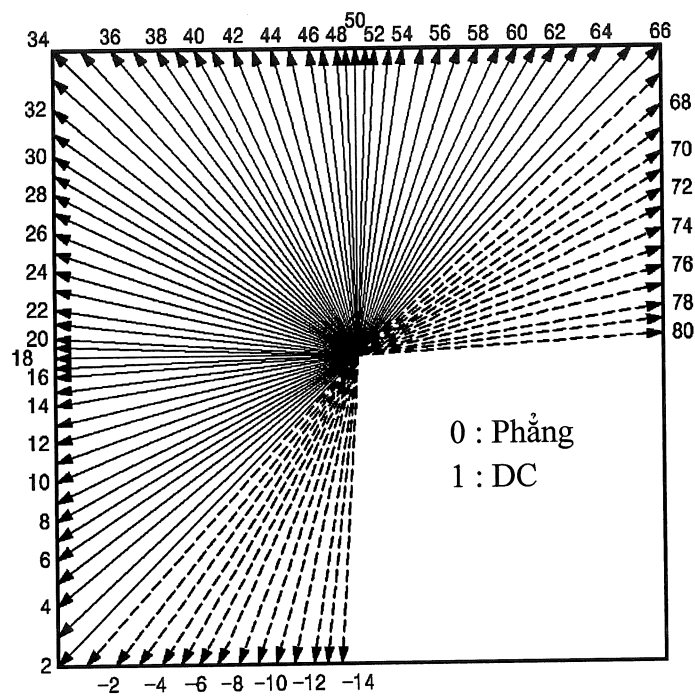


Fig.3b

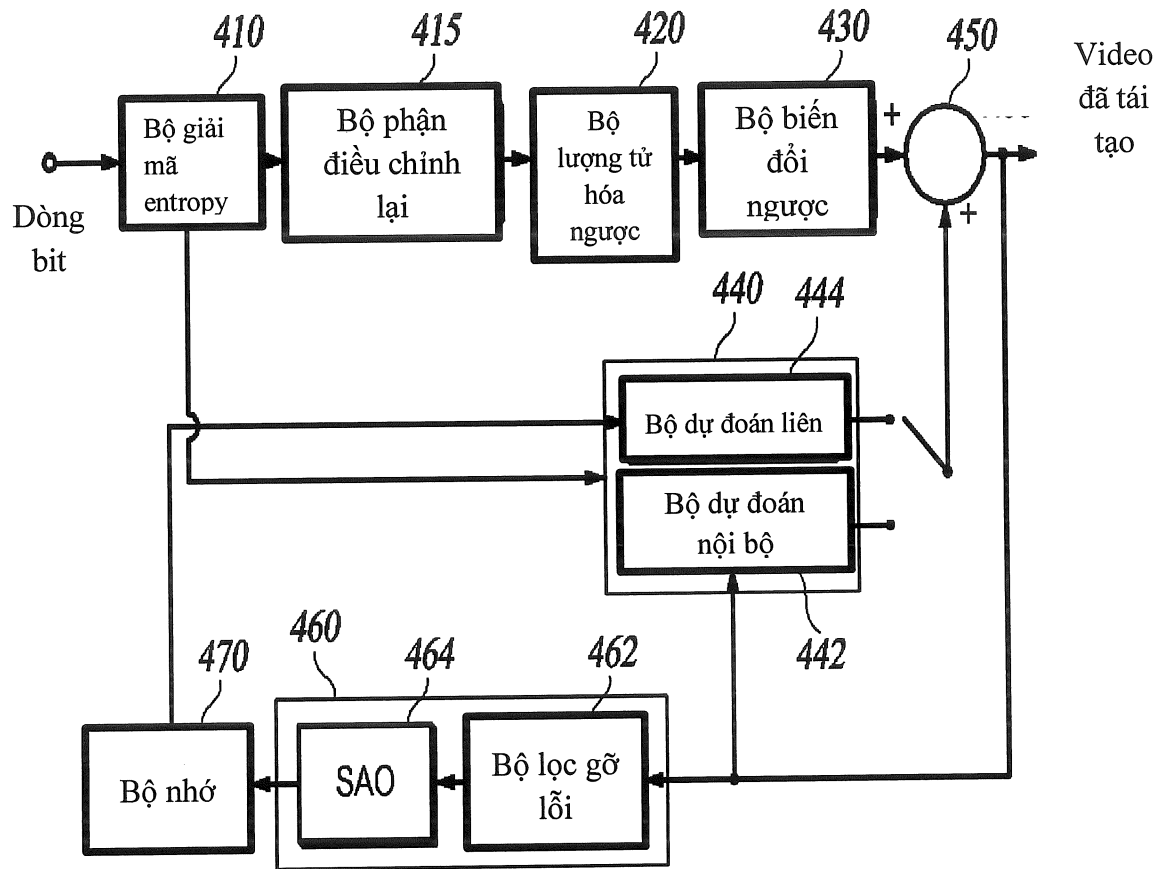


Fig.4

1. Phép trung bình
2. Phép nhân vec-tơ ma trận
3. Phép nội suy tuyến tính

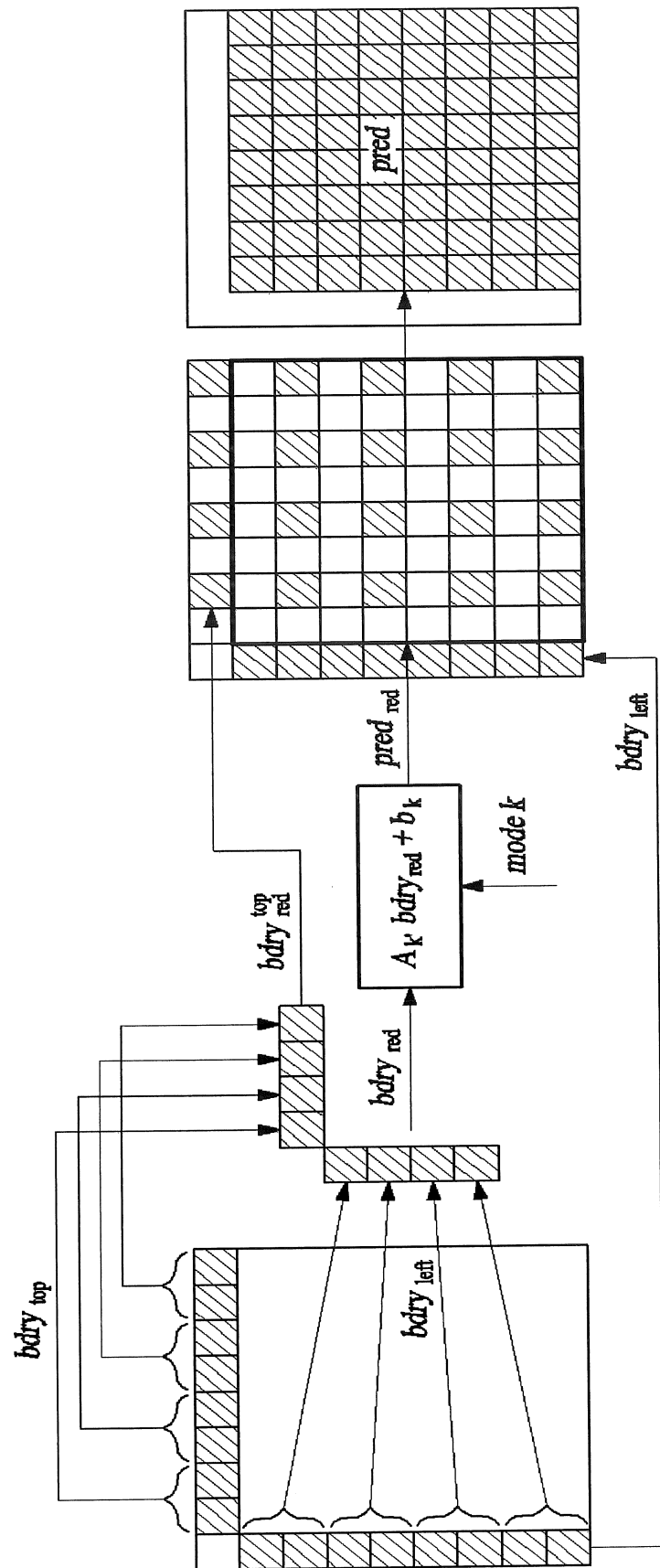


Fig.5

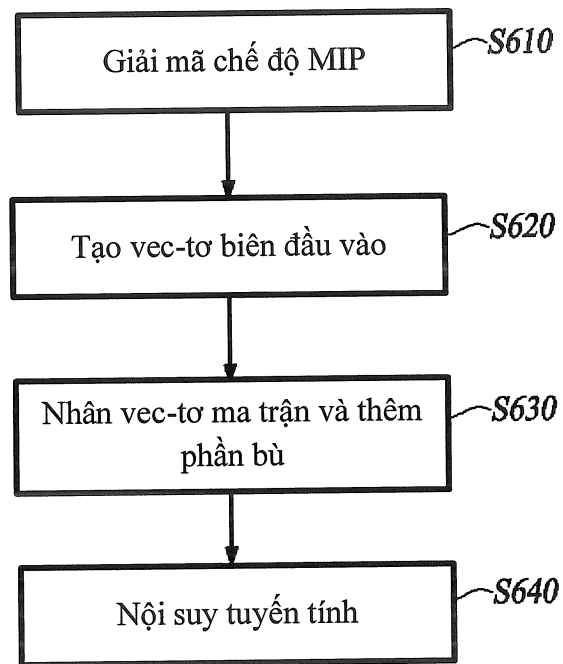


Fig.6

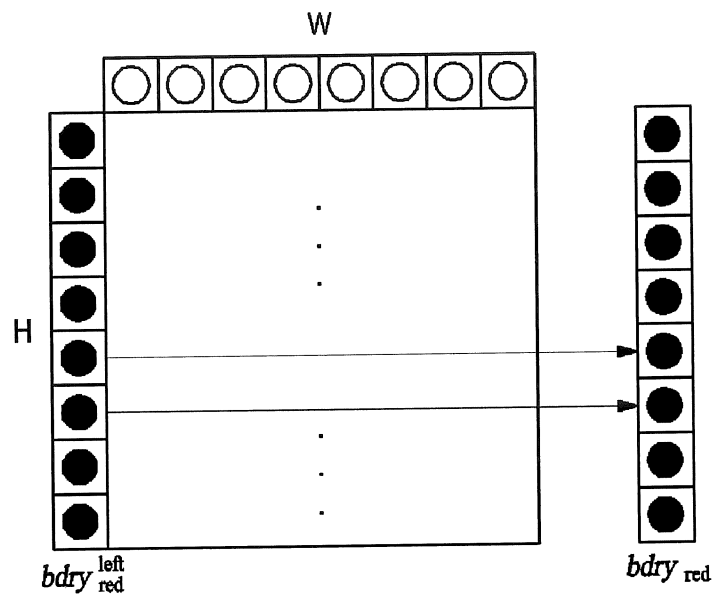


Fig.7a

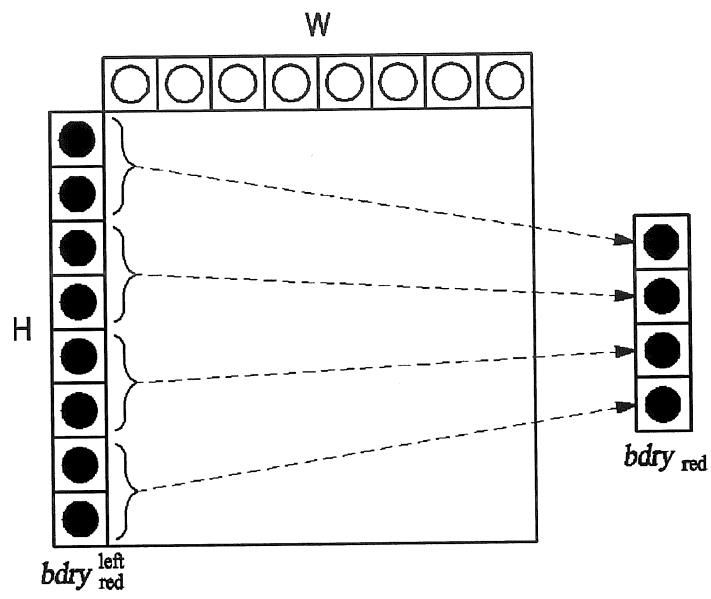


Fig. 7b

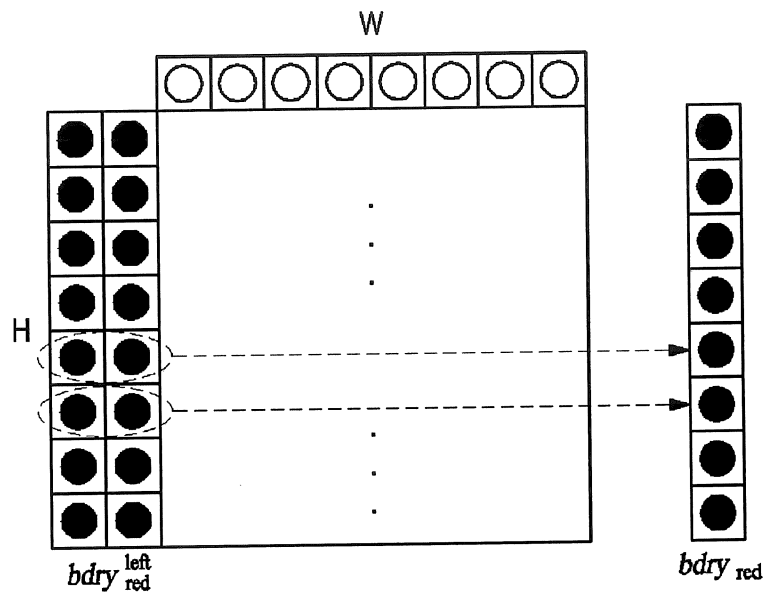


Fig. 7c

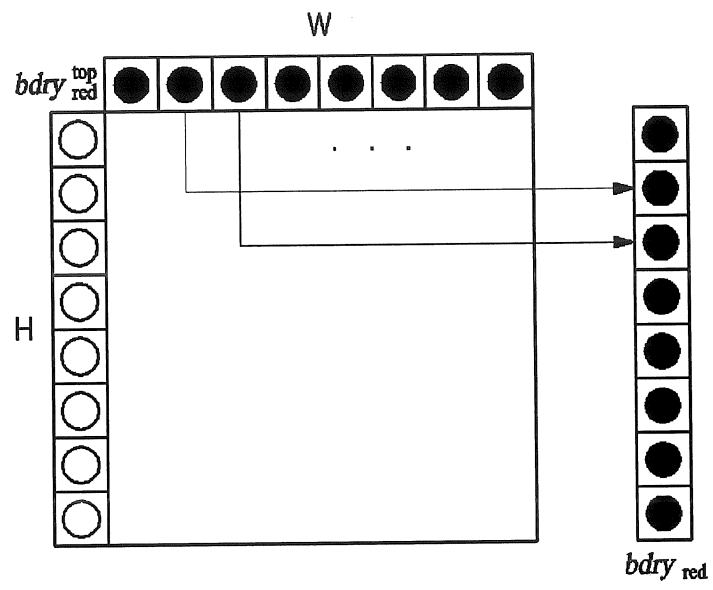


Fig.8a

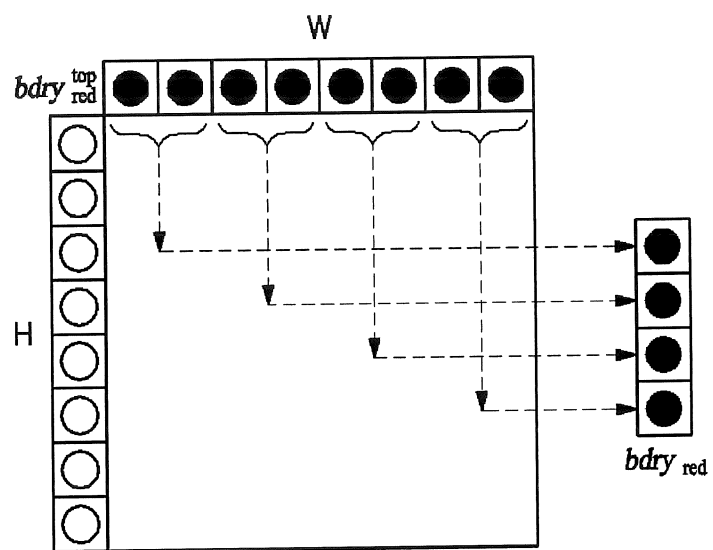


Fig.8b

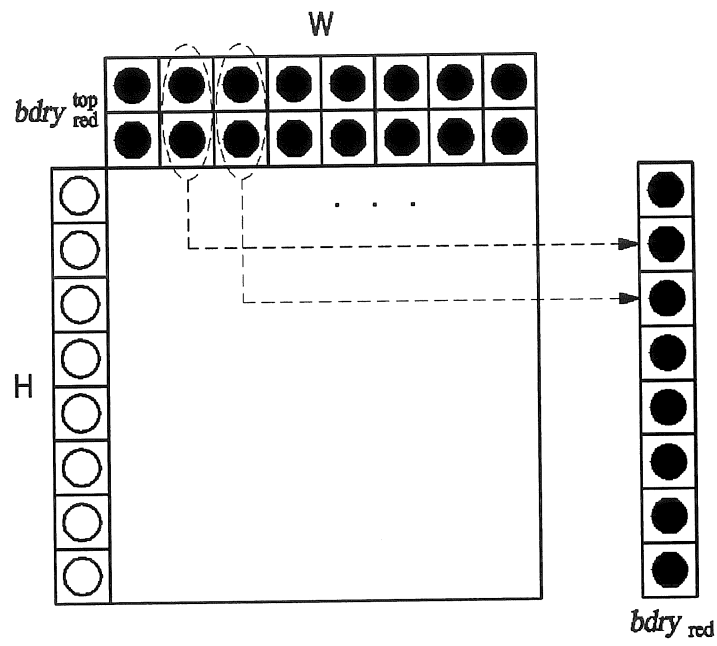


Fig.8c

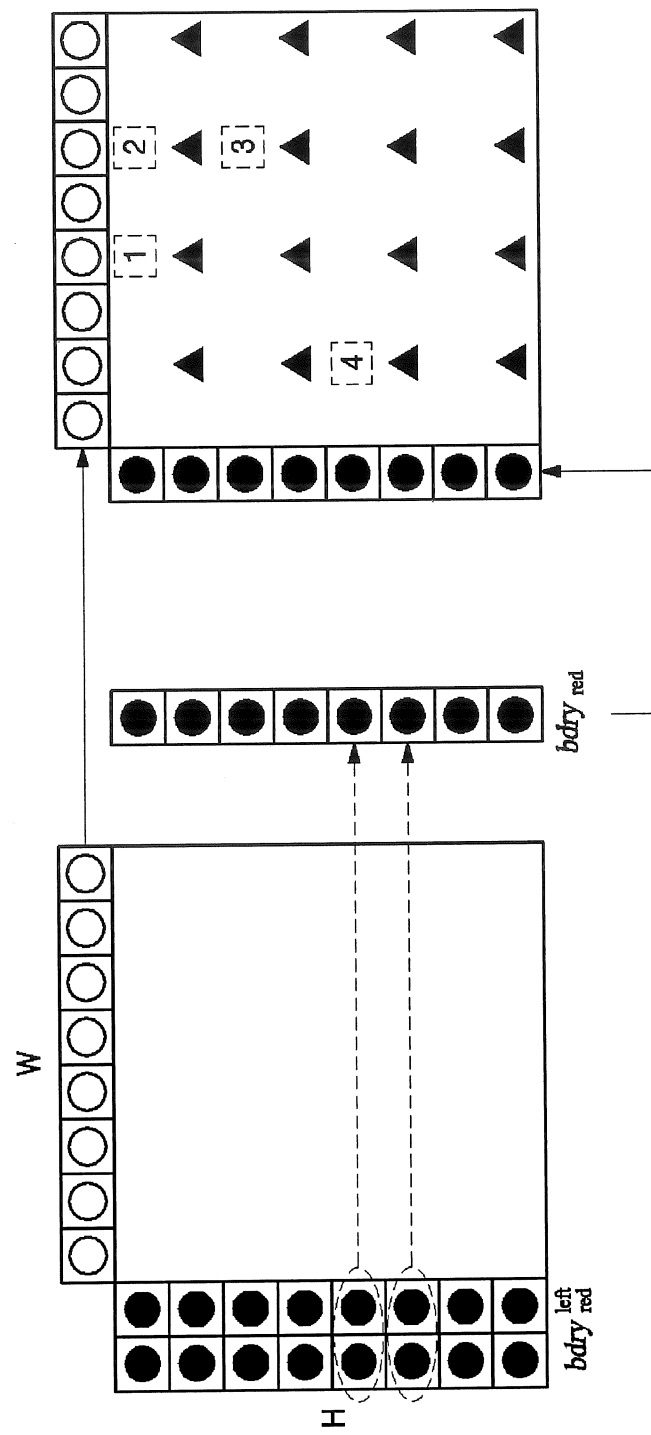


Fig.9

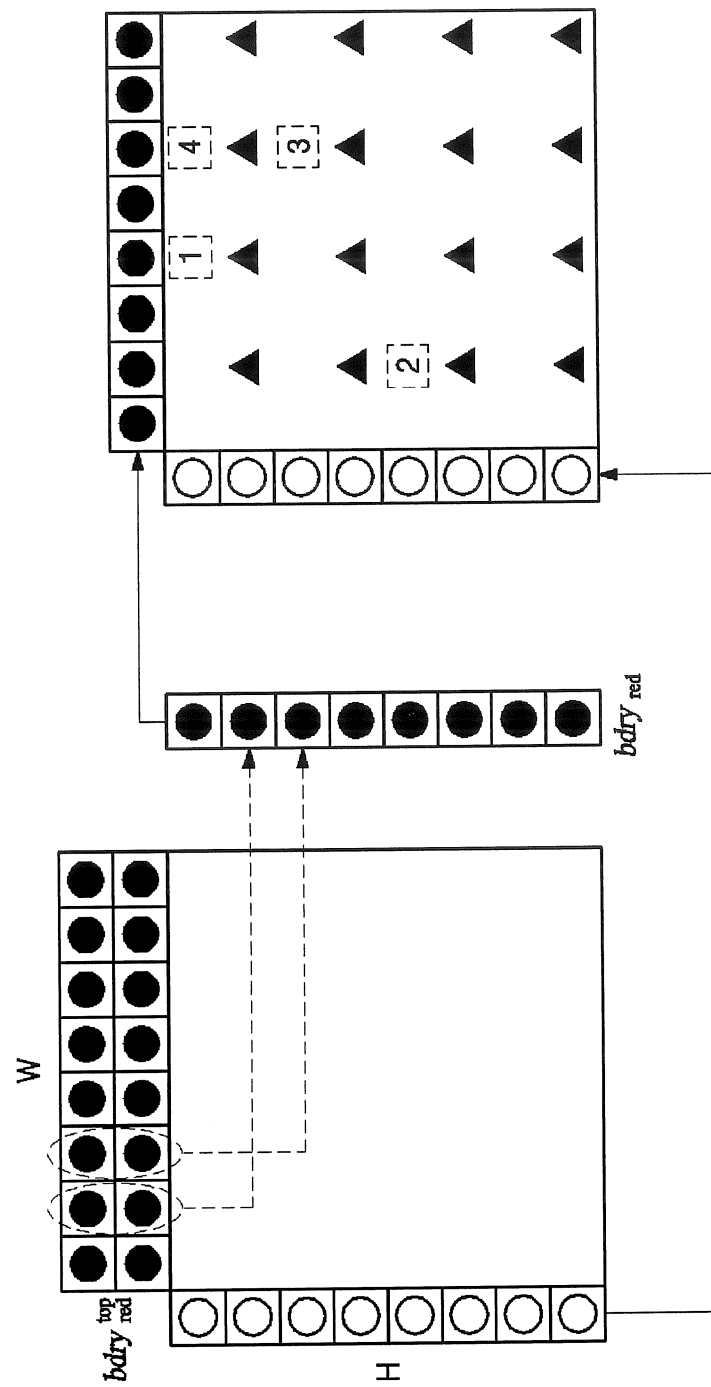


Fig.10

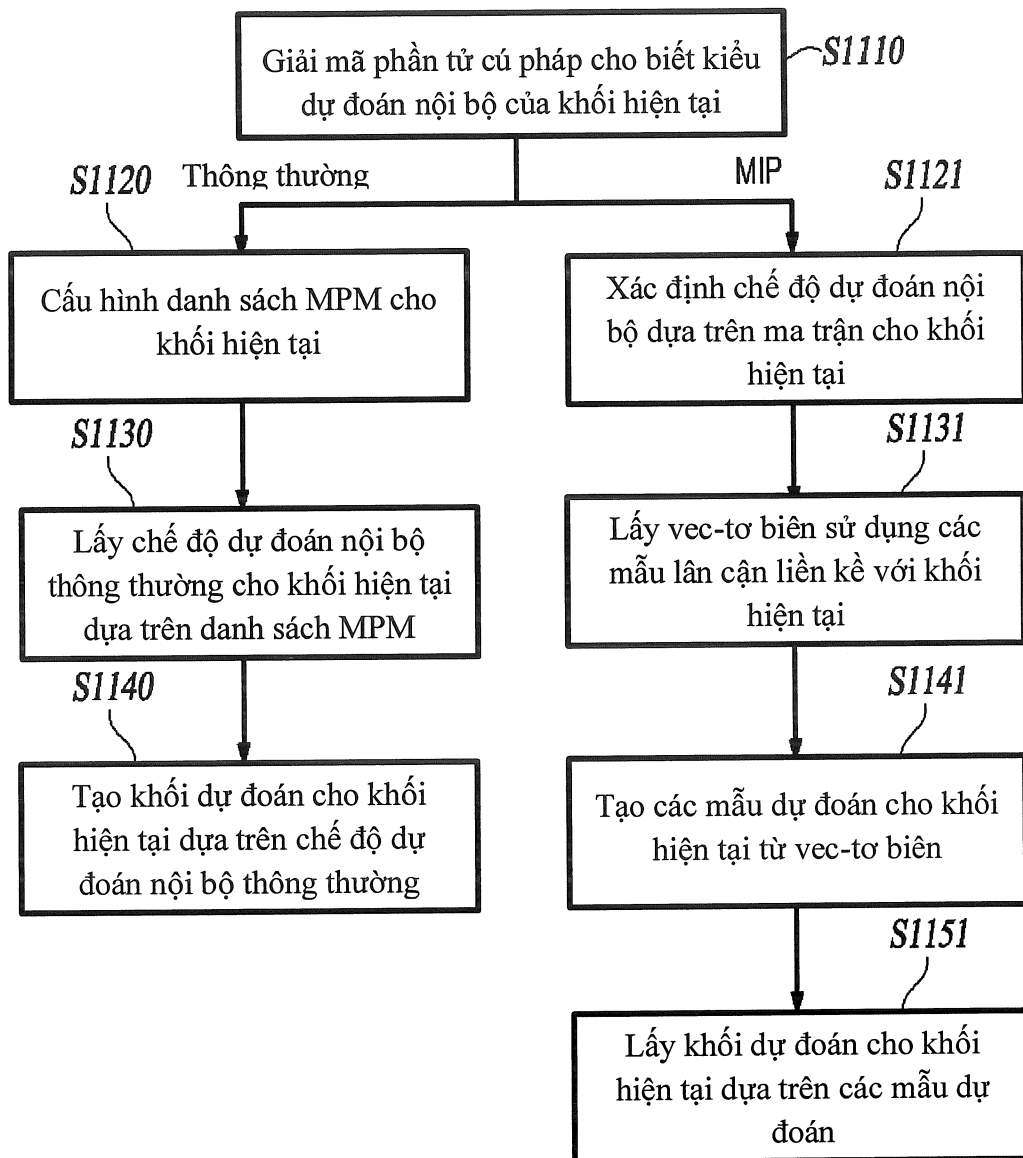


Fig.11