



(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)



1-0042703

(51)^{2020.01} H04N 19/109; H04N 19/117; H04N 19/119; H04N 19/70; H04N 19/176; H04N 19/503; H04N 19/51; H04N 19/105; H04N 19/122 (13) B

(21) 1-2021-03045

(22) 08/11/2019

(86) PCT/KR2019/015198 08/11/2019

(87) WO 2020/096426 14/05/2020

(30) 10-2018-0136262 08/11/2018 KR; 10-2018-0167979 21/12/2018 KR

(45) 27/01/2025 442

(43) 25/08/2021 401

(73) GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD. (CN)
No.18, Haibin Road, Wusha, Chang'an Dongguan, Guangdong 523860, China

(72) LEE, Bae Keun (KR).

(74) Công ty TNHH Sở hữu trí tuệ Vàng (GINTASSET CO., LTD.)

(54) PHƯƠNG PHÁP GIẢI MÃ VIDEO, PHƯƠNG PHÁP LẬP MÃ VIDEO, THIẾT BỊ GIẢI MÃ VIDEO, THIẾT BỊ LẬP MÃ VIDEO, BỘ GIẢI MÃ VIDEO, BỘ LẬP MÃ VIDEO VÀ VẬT GHI CÓ THỂ ĐỌC ĐƯỢC BẰNG MÁY TÍNH

(21) 1-2021-03045

(57) Sáng chế đề cập đến phương pháp giải mã video, phương pháp lập mã video, thiết bị giải mã video, thiết bị lập mã video, bộ giải mã video, bộ lập mã video và vật ghi có thể được được bằng máy tính. Phương pháp giải mã video bao gồm các bước: xác định xem phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không; tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời; định rõ ứng cử hợp nhất dành cho khối hiện thời dựa trên danh sách ứng cử hợp nhất; và suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất.

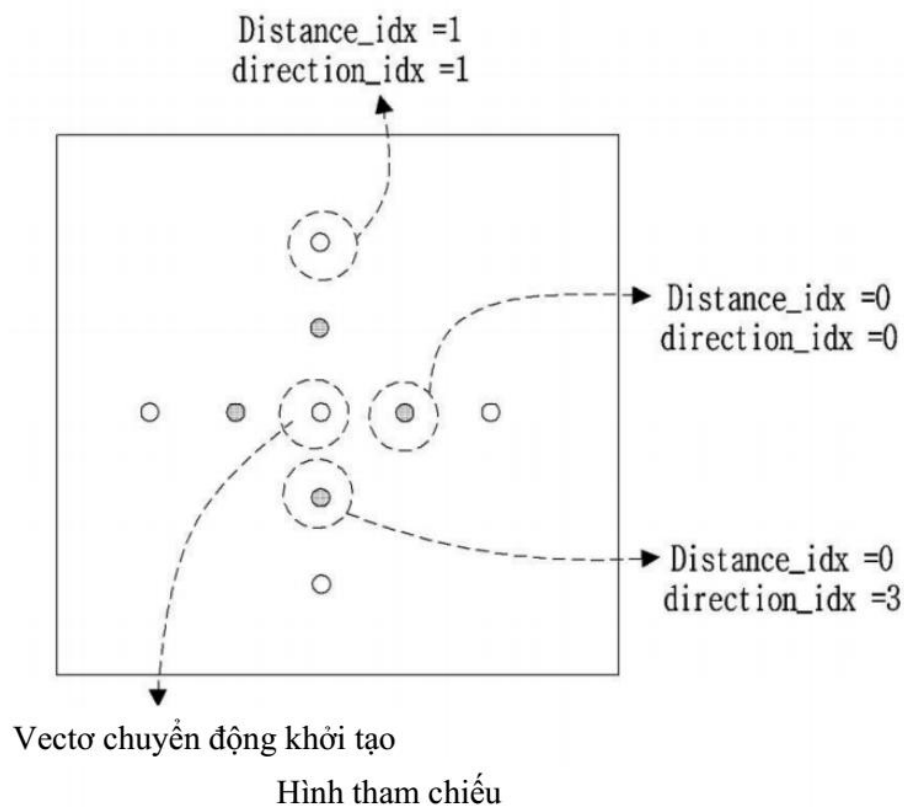


Fig.28

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp giải mã video, phương pháp lập mã video, thiết bị giải mã video, thiết bị lập mã video, bộ giải mã video, bộ lập mã video và vật ghi có thể đọc được bằng máy tính.

Tình trạng kỹ thuật của sáng chế

Khi các tấm hiển thị ngày càng lớn hơn, thì các dịch vụ video có chất lượng cao hơn nữa ngày càng được yêu cầu ngày càng nhiều hơn. Vấn đề lớn nhất của các dịch vụ video độ nét cao tăng lên đáng kể về dung lượng dữ liệu, và để giải quyết vấn đề này, các nghiên cứu để cải thiện tỷ lệ nén video được tiến hành một cách chủ động. Một ví dụ đại diện, nhóm chuyên gia hình động (Motion Picture Experts Group, MPEG) và nhóm chuyên gia mã hóa video (Video Coding Experts Group, VCEG) thuộc liên hiệp viễn thông quốc tế-thông tin vô tuyến (International Telecommunication Union-Telecommunication, ITU-T) đã thành lập nhóm nghiên cứu chung về mã hóa video (Joint Collaborative Team on Video Coding, JCT-VC) năm 2009. JCT-VC đã đề xuất kỹ thuật mã hóa video hiệu quả cao (High Efficiency Video Coding, HEVC), mà là tiêu chuẩn nén video có hiệu năng nén cao gấp đôi hiệu năng nén của H.264/AVC, và được chấp thuận là tiêu chuẩn vào ngày 25 tháng 1 năm 2013. Với sự tiến bộ nhanh chóng trong các dịch vụ video độ nét cao, hiệu năng của HEVC dần bộc lộ những hạn chế của nó.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất phương pháp để sàng lọc vector chuyển động được suy ra từ ứng cử hợp nhất dựa trên vector độ lệch khi lập mã/giải mã tín hiệu video, và thiết bị để thực hiện phương pháp này.

Mục đích khác của sáng chế là đề xuất phương pháp để báo hiệu vector độ lệch khi lập mã/giải mã tín hiệu video, và thiết bị để thực hiện phương pháp này.

Các vấn đề kỹ thuật sẽ đạt được theo sáng chế không bị hạn chế bởi các vấn đề kỹ thuật được đề cập ở trên, và các vấn đề khác không được đề cập có thể được hiểu rõ bởi người có hiểu biết trung bình về lĩnh vực kỹ thuật này từ phần mô tả sau đây.

Phương pháp để giải mã/lập mã tín hiệu video theo sáng chế bao gồm các bước: xác định xem phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không; tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời; định rõ ứng cử hợp nhất dành cho khối hiện thời dựa trên danh sách ứng cử hợp nhất; và suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất. Vào lúc này, khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm là số nhiều, thì ứng cử hợp nhất của khối hiện thời được chọn lựa dựa trên thông tin chỉ số được giải mã từ luồng bit chỉ báo một ứng cử bất kỳ trong số các ứng cử hợp nhất, và khi số lượng tối đa là 1, thì ứng cử hợp nhất được xác định mà không giải mã thông tin chỉ số.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, độ lớn của vector độ lệch có thể được xác định dựa trên thông tin chỉ số thứ nhất định rõ một ứng cử bất kỳ trong số các ứng cử độ lớn chuyển động.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động có thể được thiết đặt khác nhau theo giá trị của cờ chỉ báo các giá trị bằng số của các ứng cử độ lớn chuyển động.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, cờ có thể được báo hiệu ở mức hình.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động có thể được thiết đặt khác nhau theo độ chính xác vector chuyển động dành cho khối hiện thời.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, độ lớn của vector độ lệch có thể được nhận bằng cách áp dụng toán tử dịch cho giá trị được chỉ báo bởi ứng cử độ lớn chuyển động được định rõ bởi thông tin chỉ số thứ nhất.

Theo phương pháp lập mã và giải mã tín hiệu video theo sáng chế, chiều của vector độ lệch có thể được xác định dựa trên thông tin chỉ số thứ hai định rõ một ứng cử bất kỳ trong số các ứng cử chiều vector.

Các dấu hiệu được tóm tắt ngắn gọn ở trên dựa vào sáng chế chỉ đơn thuần là các khía cạnh làm ví dụ của phần mô tả chi tiết của sáng chế mà sẽ được mô tả bên dưới, và không nhằm hạn chế phạm vi của sáng chế.

Theo sáng chế, hiệu quả dự báo liên cấu trúc có thể được cải thiện bằng cách sàng lọc vector chuyển động của ứng cử hợp nhất dựa trên vector độ lệch.

Theo sáng chế, hiệu quả dự báo liên cấu trúc có thể được cải thiện bằng cách xác định một cách thích nghi độ lớn và chiều của vector độ lệch.

Các hiệu quả mà có thể được nhận từ sáng chế không bị hạn chế bởi các hiệu quả được đề cập ở trên, và các hiệu quả khác không được đề cập có thể được hiểu rõ bởi người có hiểu biết trung bình về lĩnh vực kỹ thuật này từ phần mô tả sau đây.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ sơ đồ khối thể hiện bộ lập mã video theo một phương án của sáng chế.

Fig.2 là hình vẽ sơ đồ khối thể hiện bộ giải mã video theo một phương án của sáng chế.

Fig.3 là hình vẽ thể hiện đơn vị cây mã hóa cơ bản theo một phương án của sáng chế.

Fig.4 là hình vẽ thể hiện các kiểu phân chia khác nhau của khối mã hóa.

Fig.5 là hình vẽ thể hiện mẫu phân chia của đơn vị cây mã hóa.

Fig.6 là hình vẽ thể hiện các hình dạng của các đơn vị cơ bản dữ liệu.

Fig.7 và Fig.8 là các hình vẽ thể hiện các ví dụ về việc phân chia khối mã hóa thành nhiều khối con.

Fig.9 là hình vẽ lưu đồ thể hiện phương pháp dự báo liên cấu trúc theo một phương án của sáng chế.

Fig.10 là hình vẽ thể hiện các chuyển động không tuyến tính của đối tượng.

Fig.11 là hình vẽ lưu đồ thể hiện phương pháp dự báo liên cấu trúc dựa trên chuyển động afin theo một phương án của sáng chế.

Fig.12 là hình vẽ thể hiện một ví dụ về các vector hạt giống afin của mỗi mô hình chuyển động afin.

Fig.13 là hình vẽ thể hiện một ví dụ về các vector afin của các khối con trong mô hình chuyển động 4 thông số.

Fig.14 là hình vẽ lưu đồ thể hiện quy trình để suy ra thông tin chuyển động của khối hiện thời sử dụng chế độ hợp nhất.

Fig.15 là hình vẽ thể hiện một ví dụ về các khối ứng cử được sử dụng để suy ra ứng cử hợp nhất.

Fig.16 là hình vẽ thể hiện các vị trí của các mẫu tham chiếu.

Fig.17 là hình vẽ thể hiện một ví dụ về các khối ứng cử được sử dụng để suy ra ứng cử hợp nhất.

Fig.18 là hình vẽ thể hiện một ví dụ mà trong đó vị trí của mẫu tham chiếu được thay đổi.

Fig.19 là hình vẽ thể hiện một ví dụ mà trong đó vị trí của mẫu tham chiếu được thay đổi.

Fig.20 là hình vẽ lưu đồ thể hiện quy trình để cập nhật danh sách thông tin chuyển động liên vùng.

Fig.21 là hình vẽ thể hiện một phương án để cập nhật danh sách ứng cử hợp nhất liên vùng.

Fig.22 là hình vẽ thể hiện một ví dụ mà trong đó chỉ số của ứng cử hợp nhất liên vùng được lưu trước đó được cập nhật.

Fig.23 là hình vẽ thể hiện vị trí của khối con đại diện.

Fig.24 là hình vẽ thể hiện một ví dụ mà trong đó danh sách thông tin chuyển động liên vùng được tạo ra cho mỗi chế độ dự báo liên cấu trúc.

Fig.25 là hình vẽ thể hiện một ví dụ mà trong đó ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động dài hạn được cộng vào danh sách ứng cử hợp nhất.

Fig.26 là hình vẽ thể hiện một ví dụ mà trong đó kiểm dư thừa được thực hiện chỉ trên một số ứng cử hợp nhất.

Fig.27 là hình vẽ thể hiện một ví dụ mà trong đó kiểm dư thừa được bỏ qua đối với ứng cử hợp nhất cụ thể.

Fig.28 là hình vẽ thể hiện vector độ lệch theo các giá trị `distance_idx` chỉ báo độ lớn của vector độ lệch và `direction_idx` chỉ báo chiều của vector độ lệch.

Fig.29 là hình vẽ thể hiện vector độ lệch theo các giá trị `distance_idx` chỉ báo độ lớn của vector độ lệch và `direction_idx` chỉ báo chiều của vector độ lệch.

Fig.30 là hình vẽ thể hiện các mẫu phân chia của khối mã hóa khi kỹ thuật phân chia dạng tam giác được áp dụng.

Fig.31 là hình vẽ thể hiện một ví dụ mà trong đó các vector độ lệch của mỗi đơn vị con được thiết đặt khác nhau.

Fig.32 là hình vẽ thể hiện các ứng cử vector chuyển động mà ứng cử hợp nhất sàng lọc có thể nhận lấy.

Fig.33 là hình vẽ thể hiện cấu hình của danh sách độ lệch sàng lọc hợp nhất.

Fig.34 và Fig.35 là các hình vẽ thể hiện các vector độ lệch được định rõ bởi các ứng cử độ lệch hợp nhất.

Fig.36 là hình vẽ thể hiện các khối ứng cử được sử dụng để suy ra các ứng cử dự báo vector chuyển động.

Fig.37 là hình vẽ thể hiện các ứng cử vector chuyển động mà có thể được thiết đặt là ứng cử dự báo vector chuyển động sàng lọc.

Fig.38 là hình vẽ thể hiện cấu hình của danh sách độ lệch sàng lọc vector dự báo.

Mô tả chi tiết sáng chế

Sau đây, phương án được ưu tiên của sáng chế sẽ được mô tả chi tiết có dựa vào các hình vẽ kèm theo.

Hoạt động lập mã và giải mã video được thực hiện theo đơn vị khối. Ví dụ, quy trình lập mã/giải mã chẳng hạn như lượng tử biến đổi, dự báo, lọc vòng trong, cấu thành lại hoặc hoạt động tương tự có thể được thực hiện trên khối mã hóa, khối biến đổi, hoặc khối dự báo.

Sau đây, khối sẽ được lập mã/giải mã sẽ được gọi là ‘khối hiện thời’. Ví dụ, khối hiện thời có thể thể hiện khối mã hóa, khối biến đổi hoặc khối dự báo theo bước xử lý lập mã/giải mã hiện thời.

Ngoài ra, có thể được hiểu rằng thuật ngữ ‘đơn vị’ được sử dụng trong phần mô tả kỹ thuật chỉ báo đơn vị cơ bản để thực hiện quy trình lập mã/giải mã cụ thể, và thuật ngữ ‘khối’ chỉ báo mảng mẫu có kích thước được xác định trước. Trừ khi có quy định khác, ‘khối’ và ‘đơn vị’ có thể được sử dụng có ý nghĩa giống nhau. Ví dụ, theo một phương án được mô tả bên dưới, có thể được hiểu rằng khối mã hóa và đơn vị mã hóa có ý nghĩa giống nhau.

Fig.1 là hình vẽ sơ đồ khối thể hiện bộ lập mã video theo một phương án của sáng chế.

Tham khảo Fig.1, thiết bị lập mã video 100 có thể bao gồm phần phân chia hình 110, phần dự báo 120 và 125, phần biến đổi 130, phần lượng tử 135, phần sắp xếp lại 160, phần mã hóa entropy 165, phần lượng tử ngược 140, phần biến đổi ngược 145, phần bộ lọc 150, và bộ nhớ 155.

Mỗi thành phần được thể hiện trên Fig.1 được thể hiện độc lập để biểu diễn các chức năng đặc trưng khác nhau trong thiết bị lập mã video, và không có nghĩa là mỗi thành phần được tạo thành bởi đơn vị cấu hình phần cứng riêng biệt hoặc một phần mềm. Tức là, mỗi thành phần được chứa sẽ được liệt kê như một thành phần để thuận tiện cho việc giải thích, và ít nhất hai trong số các thành phần này có thể được kết hợp để tạo thành một thành phần, hoặc một thành phần có thể được chia thành nhiều thành phần để thực hiện một chức năng. Các phương án tích hợp và các phương án riêng biệt của các thành phần này cũng nằm trong phạm vi của sáng chế nếu chúng không nằm ngoài bản chất của sáng chế.

Ngoài ra, một số thành phần trong số các thành phần này không phải là các thành phần cơ bản để thực hiện các chức năng cơ bản theo sáng chế, nhưng có thể là các thành phần tùy chọn chỉ để cải thiện hiệu năng. Sáng chế có thể được thực hiện bằng cách chỉ bao gồm các thành phần cơ bản để thực hiện bản chất của sáng chế không bao gồm các thành phần được sử dụng để cải thiện hiệu năng, và kết cấu chỉ bao gồm các thành phần cơ bản không bao gồm các thành phần tùy chọn được sử dụng để cải thiện hiệu năng cũng nằm trong phạm vi của sáng chế.

Phần phân chia hình 110 có thể phân chia hình được nhập vào thành ít nhất một đơn vị xử lý. Vào lúc này, đơn vị xử lý có thể là đơn vị dự báo (prediction unit, PU), đơn vị biến đổi (transform unit, TU), hoặc đơn vị mã hóa (coding unit, CU). Phần phân chia hình 110 có thể phân chia hình thành dạng kết hợp của nhiều đơn vị mã hóa, đơn vị dự báo, và đơn vị biến đổi, và lập mã hình bằng cách chọn lựa dạng kết hợp của đơn vị mã hóa, đơn vị dự báo, và đơn vị biến đổi dựa trên chuẩn mức định trước (ví dụ, hàm giá trị).

Ví dụ, một hình có thể được phân chia thành nhiều đơn vị mã hóa. Để phân chia các đơn vị mã hóa trong hình, cấu trúc cây đệ quy chẳng hạn như cấu trúc cây tứ phân có thể được sử dụng. Video hoặc đơn vị mã hóa được phân chia thành các đơn vị mã hóa khác nhau sử dụng đơn vị mã hóa lớn nhất là nút gốc có thể được phân chia để có nhiều nút con cái nhất như số lượng các đơn vị mã hóa được phân chia. Đơn vị mã hóa mà không được phân chia thêm nữa theo giới hạn định trước trở thành nút lá. Tức là, khi được giả định rằng chỉ việc phân chia hình vuông là có thể đối với một đơn vị mã hóa, thì một đơn vị mã hóa này có thể được phân chia thành thành bốn đơn vị mã hóa khác nhau.

Sau đây, theo một phương án của sáng chế, đơn vị mã hóa có thể được sử dụng như ý nghĩa của đơn vị thực hiện lập mã hoặc ý nghĩa của đơn vị thực hiện giải mã.

Đơn vị dự báo có thể là đơn vị mà được phân chia theo hình dạng trong đó ít nhất một hình dạng vuông, hình chữ nhật hoặc hình dạng tương tự có cùng kích thước trong một đơn vị mã hóa, hoặc có thể là một đơn vị dự báo bất kỳ, trong số các đơn vị dự báo được phân chia trong một đơn vị mã hóa, tức là được phân chia để có hình dạng và/hoặc kích thước khác nhau với kích thước của đơn vị dự báo khác.

Nếu đơn vị mã hóa không phải là đơn vị mã hóa nhỏ nhất khi đơn vị dự báo mà thực hiện dự báo nội cấu trúc dựa trên đơn vị mã hóa được tạo ra, dự báo nội cấu trúc có thể được thực hiện mà không phân chia hình thành nhiều đơn vị dự báo $N \times N$.

Phần dự báo 120 và 125 có thể bao gồm phần dự báo liên cấu trúc 120 mà thực hiện dự báo liên cấu trúc và phần dự báo nội cấu trúc 125 mà thực hiện dự báo nội cấu trúc. Có thể được xác định xem sử dụng dự báo liên cấu trúc hay để thực hiện dự báo nội cấu trúc đối với đơn vị dự báo, và xác định thông tin cụ thể (ví dụ, chế độ dự báo nội cấu trúc, vector chuyển động, hình tham chiếu, v.v.) theo mỗi phương pháp dự báo. Vào lúc này, đơn vị xử

lý để thực hiện dự báo có thể khác với đơn vị xử lý để xác định phương pháp dự báo và nội dung cụ thể. Ví dụ, phương pháp dự báo và chế độ dự báo có thể được xác định theo đơn vị dự báo, và hoạt động dự báo có thể được thực hiện theo đơn vị biến đổi. Hệ số còn dư (khối còn dư) giữa khối dự báo được cấu thành lại và khối ban đầu có thể được nhập vào trong phần biến đổi 130. Ngoài ra, thông tin chế độ dự báo, thông tin vector chuyển động và thông tin tương tự được sử dụng để dự báo có thể được lập mã bởi phần mã hóa entropy 165 cùng với hệ số còn dư và được truyền tới bộ giải mã. Khi chế độ lập mã cụ thể được sử dụng, thì khối ban đầu có thể được lập mã và được truyền tới bộ giải mã mà không tạo ra khối dự báo thông qua phần dự báo 120 và 125.

Phần dự báo liên cấu trúc 120 có thể dự báo đơn vị dự báo dựa trên thông tin trên ít nhất một hình trong số các hình trước hoặc sau hình hiện thời, và trong một số trường hợp, có thể dự báo đơn vị dự báo dựa trên thông tin trên vùng cục bộ mà đã được lập mã trong hình hiện thời. Phần dự báo liên cấu trúc 120 có thể bao gồm phần nội suy hình tham chiếu, phần dự báo chuyển động, và phần bù chuyển động.

Phần nội suy hình tham chiếu có thể thu thông tin hình tham chiếu từ bộ nhớ 155 và tạo ra thông tin điểm ảnh của số nguyên các điểm ảnh hoặc nhỏ hơn từ hình tham chiếu. Trong trường hợp điểm ảnh độ sáng, bộ lọc nội suy 8 nhánh dựa trên DCT với hệ số bộ lọc thay đổi có thể được sử dụng để tạo ra thông tin điểm ảnh của số nguyên các điểm ảnh hoặc nhỏ hơn theo đơn vị 1/4 điểm ảnh. Trong trường hợp tín hiệu sắc độ, bộ lọc nội suy 4 nhánh dựa trên DCT với hệ số bộ lọc thay đổi có thể được sử dụng để tạo ra thông tin điểm ảnh của số nguyên các điểm ảnh hoặc nhỏ hơn theo đơn vị 1/8 điểm ảnh.

Phần dự báo chuyển động có thể thực hiện dự báo chuyển động dựa trên hình tham chiếu được nội suy bởi phần nội suy hình tham chiếu. Các phương pháp khác nhau chẳng hạn như thuật toán so khớp khối dựa trên tìm kiếm đầy đủ (full search-based block matching algorithm, FBMA), tìm kiếm ba bước (three-step search, TSS), và thuật toán tìm kiếm ba bước mới (new three-step search, NTS) có thể được sử dụng như phương pháp để tính vector chuyển động. Vector chuyển động có thể có giá trị vector chuyển động có đơn vị 1/2 hoặc 1/4 điểm ảnh dựa trên các điểm ảnh được nội suy. Phần dự báo chuyển động có thể dự báo đơn vị dự báo hiện thời bằng cách thay đổi chế độ dự báo chuyển động. Các phương pháp

khác nhau chẳng hạn như chế độ bỏ qua, chế độ hợp nhất, chế độ dự báo vector chuyển động cải tiến (advanced motion vector prediction, AMVP), chế độ sao chép khối nội cấu trúc và phương pháp tương tự có thể được sử dụng như chế độ dự báo chuyển động.

Phân dự báo nội cấu trúc 125 có thể tạo ra đơn vị dự báo dựa trên thông tin trên các điểm ảnh tham chiếu trong vùng lân cận của khối hiện thời, mà là thông tin điểm ảnh trong hình hiện thời. Khi khối trong vùng lân cận của đơn vị dự báo hiện thời là khối mà trên đó dự báo liên cấu trúc đã được thực hiện và do đó điểm ảnh tham chiếu là điểm ảnh mà trên đó dự báo liên cấu trúc đã được thực hiện, điểm ảnh tham chiếu được chứa trong khối mà trên đó dự báo liên cấu trúc đã được thực hiện có thể được sử dụng thay cho thông tin điểm ảnh tham chiếu của khối trong vùng lân cận mà trên đó dự báo nội cấu trúc đã được thực hiện. Tức là, khi điểm ảnh tham chiếu không khả dụng, thì ít nhất một điểm ảnh tham chiếu trong số các điểm ảnh tham chiếu có thể được sử dụng thay cho thông tin điểm ảnh tham chiếu không khả dụng.

Trong dự báo nội cấu trúc, chế độ dự báo có thể có chế độ dự báo góc mà sử dụng thông tin điểm ảnh tham chiếu theo chiều dự báo, và chế độ dự báo không góc mà không sử dụng thông tin chiều khi thực hiện dự báo. Chế độ để dự báo thông tin độ sáng có thể khác với chế độ để dự báo thông tin sắc độ, và thông tin chế độ dự báo nội cấu trúc được sử dụng để dự báo thông tin độ sáng hoặc thông tin tín hiệu độ sáng được dự báo có thể được sử dụng để dự báo thông tin sắc độ.

Nếu kích thước của đơn vị dự báo giống như kích thước của đơn vị biến đổi khi dự báo nội cấu trúc được thực hiện, thì dự báo nội cấu trúc có thể được thực hiện đối với đơn vị dự báo dựa trên điểm ảnh trên cạnh bên trái, và điểm ảnh trên cạnh bên trái-trên cùng, và điểm ảnh trên cạnh trên cùng của đơn vị dự báo. Tuy nhiên, nếu kích thước của đơn vị dự báo khác với kích thước của đơn vị biến đổi khi dự báo nội cấu trúc được thực hiện, thì dự báo nội cấu trúc có thể được thực hiện sử dụng điểm ảnh tham chiếu dựa trên đơn vị biến đổi. Ngoài ra, dự báo nội cấu trúc sử dụng phân chia $N \times N$ có thể được sử dụng chỉ đối với đơn vị mã hóa nhỏ nhất.

Phương pháp dự báo nội cấu trúc có thể tạo ra khối dự báo sau khi áp dụng bộ lọc san bằng nội cấu trúc thích nghi (Adaptive Intra Smoothing, AIS) cho điểm ảnh tham chiếu

theo chế độ dự báo. Kiểu bộ lọc AIS được áp dụng cho điểm ảnh tham chiếu có thể thay đổi. Để thực hiện phương pháp dự báo nội cấu trúc, thì chế độ dự báo nội cấu trúc của đơn vị dự báo hiện thời có thể được dự báo từ chế độ dự báo nội cấu trúc của đơn vị dự báo hiện có trong vùng lân cận của đơn vị dự báo hiện thời. Khi chế độ dự báo của đơn vị dự báo hiện thời được dự báo sử dụng thông tin chế độ được dự báo từ đơn vị dự báo lân cận, nếu các chế độ dự báo nội cấu trúc của đơn vị dự báo hiện thời giống như đơn vị dự báo trong vùng lân cận, thì thông tin chỉ báo rằng các chế độ dự báo của đơn vị dự báo hiện thời giống như đơn vị dự báo trong vùng lân cận này có thể được truyền sử dụng thông tin cờ định trước, và nếu các chế độ dự báo của đơn vị dự báo hiện thời và đơn vị dự báo trong vùng lân cận khác nhau, thì thông tin chế độ dự báo của khối hiện thời có thể được lập mã bằng cách thực hiện mã hóa entropy.

Ngoài ra, khối còn dư bao gồm đơn vị dự báo mà đã được thực hiện dự báo dựa trên đơn vị dự báo được tạo ra bởi phần dự báo 120 và 125 và thông tin hệ số còn dư, mà là giá trị chênh lệch của đơn vị dự báo với khối ban đầu, có thể được tạo ra. Khối còn dư được tạo ra có thể được nhập vào trong phần biến đổi 130.

Phần biến đổi 130 có thể biến đổi khối còn dư bao gồm khối ban đầu và thông tin hệ số còn dư của đơn vị dự báo được tạo ra thông qua phần dự báo 120 và 125 sử dụng phương pháp biến đổi chẳng hạn như biến đổi cosin rời rạc (Discrete Cosine Transform, DCT) hoặc biến đổi sin rời rạc (Discrete Sine Transform, DST). Ở đây, lõi biến đổi DCT bao gồm ít nhất một trong số DCT2 và DCT8, và lõi biến đổi DST bao gồm DST7. Việc có áp dụng hay không DCT hoặc DST để biến đổi khối còn dư có thể được xác định dựa trên thông tin chế độ dự báo nội cấu trúc của đơn vị dự báo được sử dụng để tạo ra khối còn dư. Biến đổi trên khối còn dư có thể được bỏ qua. Cờ chỉ báo việc có bỏ qua hay không biến đổi trên khối còn dư có thể được lập mã. Việc bỏ qua biến đổi có thể được cho phép đối với khối còn dư có kích thước nhỏ hơn hoặc bằng ngưỡng, thành phần độ sáng, hoặc thành phần sắc độ dưới định dạng 4 : 4 : 4.

Phần lượng tử 135 có thể lượng tử các giá trị được biến đổi vào trong miền tần số bởi phần biến đổi 130. Các hệ số lượng tử có thể thay đổi theo khối hoặc độ quan trọng của

video. Giá trị được tính bởi phần lượng tử 135 có thể được cung cấp cho phần lượng tử ngược 140 và phần sắp xếp lại 160.

Phần sắp xếp lại 160 có thể sắp xếp lại các giá trị hệ số đối với các hệ số còn dư được lượng tử.

Phần sắp xếp lại 160 có thể thay đổi các hệ số của hình dạng khối hai chiều thành hình dạng vector một chiều thông qua phương pháp quét hệ số. Ví dụ, phần sắp xếp lại 160 có thể quét các hệ số DC lên tới các hệ số miền tần số cao sử dụng phương pháp quét zig-zag, và thay đổi các hệ số thành hình dạng vector một chiều. Theo kích thước của đơn vị biến đổi và chế độ dự báo nội cấu trúc, phương pháp quét theo chiều dọc để quét các hệ số của hình dạng khối hai chiều theo chiều cột và phương pháp quét theo chiều ngang để quét các hệ số của hình dạng khối hai chiều theo chiều hàng có thể được sử dụng thay vì phương pháp quét zig-zag. Tức là, theo kích thước của đơn vị biến đổi và chế độ dự báo nội cấu trúc, phương pháp quét mà sẽ được sử dụng có thể được xác định trong số phương pháp quét zig-zag, quét theo chiều dọc, và quét theo chiều ngang.

Phần mã hóa entropy 165 có thể thực hiện mã hóa entropy dựa trên các giá trị được tính bởi phần sắp xếp lại 160. Mã hóa entropy có thể sử dụng các phương pháp lập mã khác nhau chẳng hạn như phương pháp Exponential Golomb, mã hóa chiều dài biến thiên thích nghi ngữ cảnh (Context-Adaptive Variable Length Coding, CAVLC), mã hóa số học nhị phân thích nghi theo ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC), và phương pháp tương tự.

Phần mã hóa entropy 165 có thể lập mã thông tin khác nhau chẳng hạn như thông tin hệ số còn dư và thông tin kiểu khối của đơn vị mã hóa, thông tin chế độ dự báo, thông tin đơn vị phân chia, thông tin đơn vị dự báo và thông tin đơn vị truyền dẫn, thông tin vector chuyển động, thông tin khung tham chiếu, thông tin nội suy khối, và thông tin lọc được nhập từ phần sắp xếp lại 160 và các phần dự báo 120 và 125.

Phần mã hóa entropy 165 có thể lập mã entropy giá trị hệ số của đơn vị mã hóa được nhập từ phần sắp xếp lại 160.

Phần lượng tử ngược 140 và phần biến đổi ngược 145 lượng tử ngược các giá trị được lượng tử bởi phần lượng tử 135 và biến đổi ngược các giá trị được biến đổi bởi phần biến

đôi 130. Hệ số còn dư được tạo ra bởi phần lượng tử ngược 140 và phần biến đổi ngược 145 có thể được kết hợp với đơn vị dự báo được dự báo thông qua phần ước lượng chuyển động, phần bù chuyển động, và phần dự báo nội cấu trúc được chứa trong phần dự báo 120 và 125 để tạo ra khối được cấu thành lại.

Phần bộ lọc 150 có thể bao gồm ít nhất một trong số bộ lọc giải khối, đơn vị hiệu chỉnh độ lệch, và bộ lọc vòng thích nghi (adaptive loop filter, ALF).

Bộ lọc giải khối có thể loại bỏ độ méo khối được tạo ra bởi biên giữa các khối trong hình được cấu thành lại. Để xác định việc có thực hiện hay không việc giải khối, việc có áp dụng hay không bộ lọc giải khối cho khối hiện thời có thể được xác định dựa trên các điểm ảnh được chứa trong một số cột hoặc hàng được chứa trong khối này. Bộ lọc mạnh hoặc bộ lọc yếu có thể được áp dụng theo cường độ lọc giải khối cần thiết khi bộ lọc giải khối được áp dụng cho một khối. Ngoài ra, khi hoạt động lọc theo chiều dọc và lọc theo chiều ngang được thực hiện khi áp dụng bộ lọc giải khối, thì hoạt động lọc theo chiều dọc và lọc theo chiều ngang có thể được xử lý song song.

Đơn vị hiệu chỉnh độ lệch có thể hiệu chỉnh độ lệch cho video gốc theo đơn vị điểm ảnh đối với hình mà trên đó hoạt động giải khối đã được thực hiện. Để thực hiện hiệu chỉnh độ lệch đối với hình cụ thể, thì có thể sử dụng phương pháp chia các điểm ảnh được chứa trong video thành số lượng nhất định các vùng, xác định vùng để thực hiện độ lệch, và áp dụng độ lệch cho vùng này, hoặc phương pháp để áp dụng độ lệch có tính đến thông tin cạnh biên của mỗi điểm ảnh.

Lọc vòng thích nghi (ALF) có thể được thực hiện dựa trên giá trị được nhận bằng cách so sánh video được cấu thành lại và được lọc với video gốc. Sau khi chia các điểm ảnh được chứa trong hình thành các nhóm định trước, thì một bộ lọc sẽ được áp dụng cho nhóm tương ứng có thể được xác định, và hoạt động lọc có thể được thực hiện khác nhau đối với mỗi nhóm. Tín hiệu độ sáng, mà là thông tin liên quan đến việc có áp dụng hay không ALF, có thể được truyền cho mỗi đơn vị mã hóa (CU), và hình dạng và hệ số bộ lọc của bộ lọc ALF sẽ được áp dụng có thể thay đổi theo mỗi khối. Ngoài ra, bộ lọc ALF có cùng một kiểu (kiểu cố định) có thể được áp dụng bất kể đặc trưng của khối sẽ được áp dụng.

Bộ nhớ 155 có thể lưu khối hoặc hình được cấu thành lại được tính thông qua phần bộ lọc 150, và khối hoặc hình được cấu thành lại và được lưu có thể được cung cấp cho phần dự báo 120 và 125 khi dự báo liên cấu trúc được thực hiện.

Fig.2 là hình vẽ sơ đồ khối thể hiện bộ giải mã video theo một phương án của sáng chế.

Tham khảo Fig.2, bộ giải mã video 200 có thể bao gồm phần giải mã entropy 210, phần sắp xếp lại 215, phần lượng tử ngược 220, phần biến đổi ngược 225, phần dự báo 230 và 235, phần bộ lọc 240, và bộ nhớ 245.

Khi luồng bit video được nhập từ bộ lập mã video, luồng bit được nhập vào có thể được giải mã theo thủ tục ngược lại với thủ tục của bộ lập mã video.

Phần giải mã entropy 210 có thể thực hiện giải mã entropy theo thủ tục ngược lại với thủ tục để thực hiện mã hóa entropy trong phần giải mã entropy của bộ lập mã video. Ví dụ, các phương pháp khác nhau tương ứng với phương pháp được thực hiện bởi bộ lập mã video, chẳng hạn như Exponential Golomb, mã hóa chiều dài biến thiên thích nghi ngữ cảnh (CAVLC), và mã hóa số học nhị phân thích nghi theo ngữ cảnh (CABAC), có thể được áp dụng.

Phần giải mã entropy 210 có thể giải mã thông tin liên quan đến dự báo nội cấu trúc và dự báo liên cấu trúc được thực hiện bởi bộ lập mã.

Phần sắp xếp lại 215 có thể thực hiện sắp xếp lại trên luồng bit được giải mã entropy bởi phần giải mã entropy 210 dựa trên phương pháp sắp xếp lại được thực hiện bởi bộ lập mã. Các hệ số được biểu diễn theo hình dạng vector một chiều có thể được cấu thành lại và được sắp xếp lại như các hệ số của hình dạng khối hai chiều. Phần sắp xếp lại 215 có thể thu thông tin liên quan đến hoạt động quét hệ số được thực hiện bởi phần lập mã và thực hiện hoạt động cấu thành lại thông qua phương pháp quét ngược dựa trên thứ bậc quét được thực hiện bởi phần lập mã tương ứng.

Phần lượng tử ngược 220 có thể thực hiện lượng tử ngược dựa trên thông số lượng tử được cung cấp bởi bộ lập mã và giá trị hệ số của khối được sắp xếp lại.

Phần biến đổi ngược 225 có thể thực hiện biến đổi ngược trên hoạt động biến đổi, tức là, DCT hoặc DST, được thực hiện bởi phần biến đổi trên kết quả của hoạt động lượng tử

được thực hiện bởi bộ lập mã video, tức là, DCT ngược hoặc DST ngược. Ở đây, lỗi biến đổi DCT có thể bao gồm ít nhất một trong số DCT2 và DCT8, và lỗi biến đổi DST có thể bao gồm DST7. Ngoài ra, khi hoạt động biến đổi được bỏ qua trong bộ lập mã video, thì phần biến đổi ngược 225 có thể không thực hiện biến đổi ngược. Biến đổi ngược có thể được thực hiện dựa trên đơn vị truyền dẫn được xác định bởi bộ lập mã video. Phần biến đổi ngược 225 của bộ giải mã video có thể thực hiện có chọn lựa kỹ thuật biến đổi (ví dụ, DCT hoặc DST) theo nhiều mảnh thông tin chẳng hạn như phương pháp dự báo, kích thước của khối hiện thời, chiều dự báo và thông tin tương tự.

Phần dự báo 230 và 235 có thể tạo ra khối dự báo dựa trên thông tin liên quan đến việc tạo ra khối dự báo được cung cấp bởi bộ giải mã entropy 210 và thông tin trên khối được giải mã trước đó hoặc hình được cung cấp bởi bộ nhớ 245.

Như được mô tả ở trên, nếu kích thước của đơn vị dự báo và kích thước của đơn vị biến đổi giống nhau khi dự báo nội cấu trúc được thực hiện theo cùng cách thức như hoạt động của bộ lập mã video, thì dự báo nội cấu trúc được thực hiện trên đơn vị dự báo dựa trên điểm ảnh hiện có trên cạnh bên trái, điểm ảnh trên cạnh bên trái-trên cùng, và điểm ảnh trên cạnh trên cùng của đơn vị dự báo. Tuy nhiên, nếu kích thước của đơn vị dự báo và kích thước của đơn vị biến đổi khác nhau khi dự báo nội cấu trúc được thực hiện, thì dự báo nội cấu trúc có thể được thực hiện sử dụng điểm ảnh tham chiếu dựa trên đơn vị biến đổi. Ngoài ra, dự báo nội cấu trúc sử dụng phân chia $N \times N$ có thể được sử dụng chỉ đối với đơn vị mã hóa nhỏ nhất.

Phần dự báo 230 và 235 có thể bao gồm phần xác định đơn vị dự báo, phần dự báo liên cấu trúc, và phần dự báo nội cấu trúc. Phần xác định đơn vị dự báo có thể thu thông tin khác nhau chẳng hạn như thông tin đơn vị dự báo được nhập từ phần giải mã entropy 210, thông tin chế độ dự báo của phương pháp dự báo nội cấu trúc, thông tin liên quan đến dự báo chuyển động của phương pháp dự báo liên cấu trúc, và thông tin tương tự. Sau đó, phần xác định đơn vị dự báo có thể nhận dạng đơn vị dự báo từ đơn vị mã hóa hiện thời, và xác định việc đơn vị dự báo có thực hiện dự báo liên cấu trúc hoặc dự báo nội cấu trúc hay không. Phần dự báo liên cấu trúc 230 có thể thực hiện dự báo liên cấu trúc trên đơn vị dự báo hiện thời dựa trên thông tin được chứa trong ít nhất một hình trong số các hình trước

hoặc sau hình hiện thời bao gồm đơn vị dự báo hiện thời bằng cách sử dụng thông tin cần thiết cho dự báo liên cấu trúc của đơn vị dự báo hiện thời được cung cấp bởi bộ lập mã video. Ngoài ra, phần dự báo liên cấu trúc 230 có thể thực hiện dự báo liên cấu trúc dựa trên thông tin trên vùng cục bộ được cấu thành lại trước đó trong hình hiện thời bao gồm đơn vị dự báo hiện thời.

Để thực hiện dự báo liên cấu trúc, thì có thể được xác định, dựa trên đơn vị mã hóa, việc phương pháp dự báo chuyển động của đơn vị dự báo được chứa trong đơn vị mã hóa tương ứng có phải là chế độ bỏ qua, chế độ hợp nhất, chế độ dự báo vectơ chuyển động cải tiến (chế độ AMVP), hoặc chế độ sao chép khối nội cấu trúc hay không.

Phần dự báo nội cấu trúc 235 có thể tạo ra khối dự báo dựa trên thông tin trên điểm ảnh trong hình hiện thời. Khi đơn vị dự báo là đơn vị dự báo mà đã thực hiện dự báo nội cấu trúc, thì dự báo nội cấu trúc có thể được thực hiện dựa trên thông tin chế độ dự báo nội cấu trúc của đơn vị dự báo được cung cấp bởi bộ lập mã video. Phần dự báo nội cấu trúc 235 có thể bao gồm bộ lọc san bằng nội cấu trúc thích nghi (AIS), phần nội suy điểm ảnh tham chiếu, và bộ lọc DC. Bộ lọc AIS là một phần mà thực hiện hoạt động lọc trên điểm ảnh tham chiếu của khối hiện thời, và có thể xác định việc có áp dụng hay không bộ lọc theo chế độ dự báo của đơn vị dự báo hiện thời và áp dụng bộ lọc này. Hoạt động lọc AIS có thể được thực hiện trên điểm ảnh tham chiếu của khối hiện thời bằng cách sử dụng chế độ dự báo và thông tin bộ lọc AIS của đơn vị dự báo được cung cấp bởi bộ lập mã video. Khi chế độ dự báo của khối hiện thời là chế độ mà không thực hiện hoạt động lọc AIS, thì bộ lọc AIS có thể không được áp dụng.

Khi chế độ dự báo của đơn vị dự báo là đơn vị dự báo mà thực hiện dự báo nội cấu trúc dựa trên giá trị điểm ảnh được nhận bằng cách nội suy điểm ảnh tham chiếu, phần nội suy điểm ảnh tham chiếu có thể tạo ra điểm ảnh tham chiếu của đơn vị điểm ảnh có giá trị nguyên hoặc giá trị nhỏ hơn so với giá trị nguyên bằng cách nội suy điểm ảnh tham chiếu. Khi chế độ dự báo của đơn vị dự báo hiện thời là chế độ dự báo mà tạo ra khối dự báo mà không nội suy điểm ảnh tham chiếu, thì điểm ảnh tham chiếu có thể không được nội suy. Bộ lọc DC có thể tạo ra khối dự báo thông qua hoạt động lọc khi chế độ dự báo của khối hiện thời là chế độ DC.

Khối hoặc hình được cấu thành lại có thể được cung cấp cho phần bộ lọc 240. Phần bộ lọc 240 có thể bao gồm bộ lọc giải khối, đơn vị hiệu chỉnh độ lệch, và ALF.

Thông tin về việc bộ lọc giải khối có được áp dụng cho khối hoặc hình tương ứng hay không và thông tin về việc bộ lọc mạnh hoặc bộ lọc yếu có được áp dụng hay không khi bộ lọc giải khối được áp dụng có thể được cung cấp bởi bộ lập mã video. Bộ lọc giải khối của bộ giải mã video có thể được cung cấp với thông tin liên quan đến bộ lọc giải khối được cung cấp bởi bộ lập mã video, và bộ giải mã video có thể thực hiện hoạt động lọc giải khối trên khối tương ứng.

Đơn vị hiệu chỉnh độ lệch có thể thực hiện hiệu chỉnh độ lệch trên hình được cấu thành lại dựa trên kiểu hiệu chỉnh độ lệch và thông tin giá trị độ lệch được áp dụng cho video khi hoạt động lập mã được thực hiện.

ALF có thể được áp dụng cho đơn vị mã hóa dựa trên thông tin về việc có áp dụng hay không ALF và thông tin về các hệ số ALF được cung cấp bởi bộ lập mã. Thông tin ALF có thể được cung cấp sẽ được chứa trong tập thông số cụ thể.

Bộ nhớ 245 có thể lưu hình hoặc khối được cấu thành lại và sử dụng như hình tham chiếu hoặc khối tham chiếu và có thể cung cấp hình được cấu thành lại này cho đơn vị xuất.

Fig.3 là hình vẽ thể hiện đơn vị cây mã hóa cơ bản theo một phương án của sáng chế.

Khối mã hóa có kích thước tối đa có thể được định nghĩa là khối cây mã hóa. Hình được phân chia thành nhiều đơn vị cây mã hóa (CTU). Đơn vị cây mã hóa là đơn vị mã hóa có kích thước tối đa và có thể được gọi là đơn vị mã hóa lớn (Large Coding Unit, LCU). Fig.3 thể hiện một ví dụ mà trong đó hình được phân chia thành nhiều đơn vị cây mã hóa.

Kích thước của đơn vị cây mã hóa có thể được định nghĩa ở mức hình hoặc mức trình tự. Để làm được điều này, thông tin chỉ báo kích thước của đơn vị cây mã hóa có thể được báo hiệu thông qua tập thông số hình hoặc tập thông số trình tự.

Ví dụ, kích thước của đơn vị cây mã hóa đối với toàn bộ hình theo trình tự có thể được thiết đặt là 128×128 . Ngoài ra, ở mức hình, một kích thước bất kỳ trong số 128×128 và 256×256 có thể được xác định là kích thước của đơn vị cây mã hóa. Ví dụ, kích thước của đơn vị cây mã hóa có thể được thiết đặt là 128×128 trong hình thứ nhất, và kích thước của đơn vị cây mã hóa có thể được thiết đặt là 256×256 trong hình thứ hai.

Các khối mã hóa có thể được tạo ra bằng cách phân chia đơn vị cây mã hóa. Khối mã hóa chỉ báo đơn vị cơ bản để thực hiện lập mã/giải mã. Ví dụ, hoạt động dự báo hoặc biến đổi có thể được thực hiện cho mỗi khối mã hóa, hoặc chế độ lập mã dự báo có thể được xác định cho mỗi khối mã hóa. Ở đây, chế độ lập mã dự báo chỉ báo phương pháp để tạo ra hình dự báo. Ví dụ, chế độ lập mã dự báo có thể bao gồm dự báo trong hình (dự báo nội cấu trúc), dự báo giữa các hình (dự báo liên cấu trúc), tham chiếu hình hiện thời (current picture referencing, CPR) hoặc sao chép khối nội cấu trúc (intra-block copy, IBC), hoặc dự báo được kết hợp. Đối với khối mã hóa, khối dự báo có thể được tạo ra bằng cách sử dụng ít nhất một chế độ lập mã dự báo trong số dự báo nội cấu trúc, dự báo liên cấu trúc, tham chiếu hình hiện thời, và dự báo được kết hợp.

Thông tin chỉ báo chế độ lập mã dự báo của khối hiện thời có thể được báo hiệu thông qua luồng bit. Ví dụ, thông tin có thể là cờ 1-bit chỉ báo việc chế độ lập mã dự báo có phải là chế độ nội cấu trúc hoặc chế độ liên cấu trúc hay không. Chỉ khi chế độ lập mã dự báo của khối hiện thời được xác định là chế độ liên cấu trúc, thì chế độ tham chiếu hình hiện thời hoặc chế độ dự báo được kết hợp có thể được sử dụng.

Chế độ tham chiếu hình hiện thời là dành cho việc thiết đặt hình hiện thời là hình tham chiếu và nhận khối dự báo của khối hiện thời từ vùng mà đã được lập mã/giải mã trong hình hiện thời. Ở đây, hình hiện thời có nghĩa là hình bao gồm khối hiện thời. Thông tin chỉ báo chế độ tham chiếu hình hiện thời có được áp dụng cho khối hiện thời hay không có thể được báo hiệu thông qua luồng bit. Ví dụ, thông tin có thể là cờ 1-bit. Khi cờ này có giá trị là true, thì chế độ lập mã dự báo của khối hiện thời có thể được xác định là chế độ tham chiếu hình hiện thời, và khi cờ này có giá trị là false, thì chế độ dự báo của khối hiện thời có thể được xác định là dự báo liên cấu trúc.

Ngoài ra, chế độ lập mã dự báo của khối hiện thời có thể được xác định dựa trên chỉ số hình tham chiếu. Ví dụ, khi chỉ số hình tham chiếu chỉ báo hình hiện thời, thì chế độ lập mã dự báo của khối hiện thời có thể được xác định là chế độ tham chiếu hình hiện thời. Khi chỉ số hình tham chiếu chỉ báo hình khác với hình hiện thời, thì chế độ lập mã dự báo của khối hiện thời có thể được xác định là chế độ dự báo liên cấu trúc. Tức là, chế độ tham chiếu hình hiện thời là phương pháp dự báo sử dụng thông tin trên một vùng mà trong đó

hoạt động lập mã/giải mã đã được hoàn thành trong hình hiện thời, và chế độ dự báo liên cấu trúc là phương pháp dự báo sử dụng thông tin trên hình khác mà trong đó hoạt động lập mã/giải mã đã được hoàn thành.

Chế độ dự báo được kết hợp thể biểu diễn độ lập mã mà trong đó hai hoặc nhiều chế độ trong số dự báo nội cấu trúc, dự báo liên cấu trúc, và chế độ tham chiếu hình hiện thời được kết hợp. Ví dụ, khi chế độ dự báo được kết hợp được áp dụng, thì khối dự báo thứ nhất có thể được tạo ra dựa trên một chế độ trong số dự báo nội cấu trúc, dự báo liên cấu trúc, và tham chiếu hình hiện thời, và khối dự báo thứ hai có thể được tạo ra dựa trên chế độ khác. Khi khối dự báo thứ nhất và khối dự báo thứ hai được tạo ra, thì khối dự báo cuối cùng có thể được tạo ra thông qua phép tính trung bình hoặc phép tính tổng trọng số của khối dự báo thứ nhất và khối dự báo thứ hai. Thông tin chỉ báo việc chế độ dự báo được kết hợp có được áp dụng hay không có thể được báo hiệu thông qua luồng bit. Thông tin có thể là cờ 1-bit.

Fig.4 là hình vẽ thể hiện các kiểu phân chia khác nhau của khối mã hóa.

Khối mã hóa có thể được phân chia thành nhiều khối mã hóa dựa trên phân chia cây tứ phân, phân chia cây nhị phân, hoặc phân chia cây tam phân. Khối mã hóa được phân chia có thể được phân chia lại thành nhiều khối mã hóa dựa trên phân chia cây tứ phân, phân chia cây nhị phân, hoặc phân chia cây tam phân.

Phân chia cây tứ phân dùng để chỉ kỹ thuật phân chia mà phân chia khối hiện thời thành bốn khối. Kết quả của việc phân chia cây tứ phân, khối hiện thời có thể được phân chia thành bốn phần phân chia có hình dạng vuông (xem ‘SPLIT_QT’ trên Fig.4(a)).

Phân chia cây nhị phân dùng để chỉ kỹ thuật phân chia mà phân chia khối hiện thời thành hai khối. Việc phân chia khối hiện thời thành hai khối dọc theo chiều dọc (tức là, sử dụng đường nằm dọc cắt qua khối hiện thời) có thể được gọi là phân chia cây nhị phân theo chiều dọc, và phân chia khối hiện thời thành hai khối dọc theo chiều ngang (tức là, sử dụng đường nằm ngang cắt qua khối hiện thời) có thể được gọi là phân chia cây nhị phân theo chiều ngang. Kết quả của việc phân chia cây nhị phân, khối hiện thời có thể được phân chia thành hai phần phân chia có hình dạng không vuông. ‘SPLIT_BT_VER’ trên Fig.4(b) thể

hiện kết quả của việc phân chia cây nhị phân theo chiều dọc, và ‘SPLIT_BT_HOR’ trên Fig.4(c) thể hiện kết quả của việc phân chia cây nhị phân theo chiều ngang.

Phân chia cây tam phân dùng để chỉ kỹ thuật phân chia mà phân chia khối hiện thời thành ba khối. Việc phân chia khối hiện thời thành ba khối dọc theo chiều dọc (tức là, sử dụng hai đường nằm dọc cắt qua khối hiện thời) có thể được gọi là phân chia cây tam phân theo chiều dọc, và việc phân chia khối hiện thời thành ba khối dọc theo chiều ngang (tức là, sử dụng hai đường nằm ngang cắt qua khối hiện thời) có thể được gọi là phân chia cây tam phân theo chiều ngang. Kết quả của phân chia cây tam phân, khối hiện thời có thể được phân chia thành ba phần phân chia có hình dạng không vuông. Vào lúc này, chiều rộng/chiều cao của phần phân chia được định vị ở trung tâm của khối hiện thời có thể lớn gấp đôi chiều rộng/chiều cao của các phần phân chia khác. ‘SPLIT_TT_VER’ trên Fig.4(d) thể hiện kết quả của việc phân chia cây tam phân theo chiều dọc, và ‘SPLIT_TT_HOR’ trên Fig.4(e) thể hiện kết quả của việc phân chia cây tam phân theo chiều ngang.

Số lần phân chia đơn vị cây mã hóa có thể được định nghĩa là độ sâu phân chia. Độ sâu phân chia tối đa của đơn vị cây mã hóa có thể được xác định ở mức trình tự hoặc mức hình. Do đó, độ sâu phân chia tối đa của đơn vị cây mã hóa có thể khác nhau đối với mỗi trình tự hoặc hình.

Ngoài ra, độ sâu phân chia tối đa đối với mỗi kỹ thuật phân chia có thể được xác định độc lập. Ví dụ, độ sâu phân chia tối đa được cho phép đối với phân chia cây tứ phân có thể khác với độ sâu phân chia tối đa được cho phép đối với phân chia cây nhị phân và/hoặc phân chia cây tam phân.

Bộ lập mã có thể báo hiệu thông tin chỉ báo ít nhất một trong số kiểu phân chia và độ sâu phân chia của khối hiện thời thông qua luồng bit. Bộ giải mã có thể xác định kiểu phân chia và độ sâu phân chia của đơn vị cây mã hóa dựa trên thông tin được phân tích cú pháp từ luồng bit.

Fig.5 là hình vẽ thể hiện mẫu phân chia của đơn vị cây mã hóa.

Việc phân chia khối mã hóa sử dụng kỹ thuật phân chia chẳng hạn như phân chia cây tứ phân, phân chia cây nhị phân, và/hoặc phân chia cây tam phân có thể được gọi là phân chia đa cây.

Các khối mã hóa được tạo ra bằng cách áp dụng phân chia đa cây cho khối mã hóa có thể được gọi là các khối mã hóa phía dưới. Khi độ sâu phân chia của khối mã hóa là k , thì độ sâu phân chia của các khối mã hóa phía dưới được thiết đặt là $k + 1$.

Ngược lại, đối với các khối mã hóa có độ sâu phân chia $k + 1$, thì khối mã hóa có độ sâu phân chia k có thể được gọi là khối mã hóa phía trên.

Kiểu phân chia của khối mã hóa hiện thời có thể được xác định dựa trên ít nhất một trong số kiểu phân chia của khối mã hóa phía trên và kiểu phân chia của khối mã hóa lân cận. Ở đây, khối mã hóa lân cận là khối mã hóa gần kề với khối mã hóa hiện thời và có thể bao gồm ít nhất một trong số khối lân cận trên cùng và khối lân cận bên trái của khối mã hóa hiện thời, và khối lân cận gần kề với góc bên trái-trên cùng. Ở đây, kiểu phân chia có thể bao gồm ít nhất một trong số việc có hay không phân chia cây tứ phân, việc có hay không phân chia cây nhị phân, chiều phân chia cây nhị phân, việc có hay không phân chia cây tam phân, và chiều phân chia cây tam phân.

Để xác định kiểu phân chia của khối mã hóa, thông tin chỉ báo việc khối mã hóa có thể được phân chia hay không có thể được báo hiệu thông qua luồng bit. Thông tin này là cờ 1-bit 'split_cu_flag', và khi cờ này có giá trị là true, thì nó chỉ báo rằng khối mã hóa được phân chia bởi kỹ thuật phân chia đa cây.

Khi split_cu_flag có giá trị là true, thì thông tin chỉ báo việc khối mã hóa có được phân chia cây tứ phân hay không có thể được báo hiệu thông qua luồng bit. Thông tin này là cờ 1-bit split_qt_flag, và khi cờ này có giá trị là true, thì khối mã hóa có thể được phân chia thành bốn khối.

Ví dụ, trong ví dụ được thể hiện trên Fig.5, khi đơn vị cây mã hóa được phân chia cây tứ phân, thì bốn khối mã hóa có độ sâu phân chia là 1 được tạo ra. Ngoài ra, được thể hiện rằng phân chia cây tứ phân được áp dụng lại cho các khối mã hóa thứ nhất và thứ tư trong số bốn khối mã hóa được tạo ra là kết quả của phân chia cây tứ phân. Kết quả là, bốn khối mã hóa có độ sâu phân chia là 2 có thể được tạo ra.

Ngoài ra, các khối mã hóa có độ sâu phân chia là 3 có thể được tạo ra bằng cách áp dụng phân chia cây tứ phân lại cho khối mã hóa có độ sâu phân chia là 2.

Khi phân chia cây tứ phân không được áp dụng cho khối mã hóa, việc phân chia cây nhị phân hoặc phân chia cây tam phân có được thực hiện trên khối mã hóa hay không có thể được xác định có tính đến ít nhất một trong số kích thước của khối mã hóa, việc khối mã hóa có được định vị ở biên hình hay không, độ sâu phân chia tối đa, và kiểu phân chia của khối lân cận. Khi được xác định để thực hiện phân chia cây nhị phân hoặc phân chia cây tam phân trên khối mã hóa, thì thông tin chỉ báo chiều phân chia có thể được báo hiệu thông qua luồng bit. Thông tin này có thể là cờ 1-bit `mtt_split_cu_vertical_flag`. Dựa trên cờ này, việc chiều phân chia có phải là chiều dọc hoặc chiều ngang hay không có thể được xác định. Ngoài ra, thông tin chỉ báo việc phân chia cây nhị phân hoặc phân chia cây tam phân có được áp dụng cho khối mã hóa hay không có thể được báo hiệu thông qua luồng bit. Thông tin này có thể là cờ 1-bit `mtt_split_cu_binary_flag`. Dựa trên cờ này, việc phân chia cây nhị phân hoặc phân chia cây tam phân có được áp dụng cho khối mã hóa hay không có thể được xác định.

Ví dụ, trong ví dụ được thể hiện trên Fig.5, được thể hiện rằng phân chia cây nhị phân theo chiều dọc được áp dụng cho khối mã hóa có độ sâu phân chia là 1, phân chia cây tam phân theo chiều dọc được áp dụng cho khối mã hóa cạnh bên trái trong số các khối mã hóa được tạo ra như kết quả của việc phân chia, và phân chia cây nhị phân theo chiều dọc được áp dụng cho khối mã hóa cạnh bên phải.

Khi thiết bị để lập mã hoặc giải mã video được thực hiện, thì có vấn đề ở chỗ là vùng lớn hơn so với giá trị ngưỡng khó để xử lý do hiệu năng phần cứng. Ví dụ, có vấn đề ở chỗ là khi có thể xử lý đồng thời lên tới 4096 mẫu dựa trên hiệu năng phần cứng, thì các đơn vị dữ liệu có kích thước 64×64 phải được truy nhập và xử lý dư thừa, và dữ liệu không thể được xử lý đồng thời đối với các vùng có nhiều mẫu hơn so với 4096 mẫu. Tương tự điều này, đơn vị cơ bản để xử lý dữ liệu có thể được định nghĩa là đơn vị cơ bản dữ liệu dựa trên đường ống dữ liệu (Virtual Processing Data Unit, VPDU, sau đây, được gọi là đơn vị cơ bản dữ liệu).

Đơn vị cơ bản dữ liệu có thể được phân loại thành kiểu hình vuông, kiểu không phải hình vuông hoặc kiểu không phải hình chữ nhật.

Fig.6 là hình vẽ thể hiện các hình dạng của các đơn vị cơ bản dữ liệu.

Các đơn vị cơ bản dữ liệu có thể bao gồm các mẫu nhiều như hoặc nhỏ hơn so với số lượng tối đa của các mẫu mà có thể được xử lý đồng thời. Ví dụ, như được thể hiện trong ví dụ trên Fig.6(a), các khối hình vuông có kích thước 64×64 có thể được thiết đặt là các đơn vị cơ bản dữ liệu. Ngoài ra, các khối không phải hình vuông có thể được thiết đặt là các đơn vị cơ bản dữ liệu. Ví dụ, như được thể hiện trong ví dụ trên Fig.6(b) hoặc Fig.6(c), khối có kích thước 32×128 hoặc khối có kích thước 64×32 có thể được thiết đặt là đơn vị cơ bản dữ liệu.

Mặc dù không được thể hiện trên hình vẽ, các đơn vị cơ bản dữ liệu hình tam giác, hình dạng chữ L, và các đơn vị cơ bản dữ liệu hình đa giác có thể được định nghĩa.

Thông tin để xác định đơn vị cơ bản dữ liệu có thể được báo hiệu thông qua luồng bit. Thông tin này có thể dành cho việc xác định ít nhất một trong số kích thước và hình dạng của đơn vị cơ bản dữ liệu. Dựa trên thông tin này, việc có cho phép hay không đơn vị cơ bản dữ liệu không phải hình vuông hoặc có cho phép hay không đơn vị cơ bản dữ liệu không phải hình chữ nhật có thể được xác định.

Ngoài ra, ít nhất một trong số kích thước và hình dạng của đơn vị cơ bản dữ liệu có thể được định trước trong bộ lập mã và bộ giải mã.

Việc có cho phép hay không kiểu phân chia của khối mã hóa có thể được xác định có tính đến kích thước của đơn vị cơ bản dữ liệu. Ví dụ, khi khối mã hóa được tạo ra là kết quả của việc phân chia khối mã hóa lớn hơn so với đơn vị cơ bản dữ liệu, thì việc phân chia này có thể không được cho phép. Ngoài ra, khi khối mã hóa không phải hình vuông được tạo ra là kết quả của việc phân chia khối mã hóa lớn hơn so với đơn vị cơ bản dữ liệu, thì việc phân chia này có thể không được cho phép. Ví dụ, khi chiều rộng hoặc chiều cao của khối mã hóa lớn hơn so với giá trị ngưỡng hoặc khi số lượng các mẫu được chứa trong khối mã hóa lớn hơn so với giá trị ngưỡng, thì phân chia cây nhị phân hoặc phân chia cây tam phân có thể không được cho phép. Do đó, việc lập mã thông tin có liên quan đến phân chia cây nhị phân hoặc phân chia cây tam phân có thể được bỏ qua.

Ngoài ra, có thể được thiết đặt là nhất thiết phải phân chia khối mã hóa lớn hơn so với đơn vị cơ bản dữ liệu. Ngoài ra, có thể được thiết đặt là nhất thiết phải thực hiện phân chia cây nhị phân hoặc phân chia cây tam phân trên khối mã hóa lớn hơn so với đơn vị cơ bản

dữ liệu. Do đó, đối với khối mã hóa lớn hơn so với đơn vị cơ bản dữ liệu, thì mặc dù flag `split_flag` chỉ báo việc có phân chia hay không khối mã hóa không được lập mã, giá trị của cờ có thể được suy ra là 1.

Ví dụ khác, khối mã hóa lớn hơn so với đơn vị cơ bản dữ liệu có thể được phân chia thành nhiều khối con. Ở đây, khối con có thể được thiết đặt là đơn vị dự báo, mà là đơn vị cơ bản để dự báo, hoặc đơn vị biến đổi, mà là đơn vị cơ bản để biến đổi và/hoặc lượng tử. Vào lúc này, việc phân chia khối mã hóa thành nhiều đơn vị dự báo có thể được định nghĩa là phân chia đơn vị dự báo VPDU, và việc phân chia khối mã hóa thành nhiều đơn vị biến đổi có thể được định nghĩa là phân chia đơn vị biến đổi VPDU.

Ít nhất một trong số hoạt động phân chia đơn vị dự báo VPDU và hoạt động phân chia đơn vị biến đổi VPDU có thể được áp dụng cho khối mã hóa. Kiểu phân chia của khối mã hóa theo ứng dụng của hoạt động phân chia đơn vị dự báo VPDU có thể được thiết đặt là giống như kiểu phân chia của khối mã hóa theo ứng dụng của hoạt động phân chia đơn vị biến đổi VPDU.

Khi chỉ hoạt động phân chia đơn vị dự báo VPDU được áp dụng cho khối mã hóa, thì hoạt động dự báo được thực hiện cho mỗi khối con, nhưng hoạt động biến đổi và/hoặc lượng tử có thể được thực hiện cho khối mã hóa. Vào lúc này, chế độ dự báo chẳng hạn như chế độ lập mã dự báo, chế độ dự báo nội cấu trúc, hoặc chế độ dự báo liên cấu trúc có thể được xác định đối với khối mã hóa.

Khi chỉ hoạt động phân chia đơn vị biến đổi VPDU được áp dụng cho khối mã hóa, thì hoạt động dự báo được thực hiện cho khối con, nhưng hoạt động biến đổi và/hoặc lượng tử có thể được thực hiện cho mỗi khối con.

Fig.7 và Fig.8 là các hình vẽ thể hiện các ví dụ về việc phân chia khối mã hóa thành nhiều khối con.

Fig.7 là hình vẽ thể hiện mẫu phân chia khi chỉ đơn vị cơ bản dữ liệu hình vuông được cho phép, và Fig.8 là hình vẽ thể hiện mẫu phân chia khi đơn vị cơ bản dữ liệu hình vuông và đơn vị cơ bản dữ liệu không phải hình vuông được cho phép.

Khi được giả định rằng chỉ các đơn vị cơ bản dữ liệu hình vuông được cho phép, trên Fig.7(a) và Fig.7(b), CU0 và CU2 được định nghĩa là hai VPDU khác nhau, và CU1 được

định nghĩa là bốn VPDU khác nhau. Do đó, CU0 và CU2 có thể được phân chia thành hai khối con, và CU1 có thể được phân chia thành bốn khối con.

Khi được giả định rằng các đơn vị cơ bản dữ liệu hình vuông và các đơn vị cơ bản dữ liệu không phải hình vuông được cho phép, trên Fig.8(a) và Fig.8(b), CU0 và CU2 có thể được định nghĩa là một VPDU, trong khi CU1 có thể được định nghĩa sử dụng hai VPDU khác nhau. Do đó, CU0 và CU2 không được phân chia thành các khối con, trong khi CU1 có thể được phân chia thành hai khối con.

Vào lúc này, CU1 có thể được phân chia thành các khối con hình vuông hoặc các khối con không phải hình vuông. Ví dụ, CU1 có thể được phân chia thành hai khối con hình vuông dựa trên đường nằm ngang mà phân chia CU1 trên và dưới. Ngoài ra, CU1 có thể được phân chia thành hai khối con không phải hình vuông dựa trên đường nằm dọc mà phân chia CU1 trái và phải.

Khi có nhiều ứng cử kiểu phân chia có thể áp dụng được cho khối mã hóa, thì thông tin chỉ báo một ứng cử bất kỳ trong số nhiều ứng cử kiểu phân chia có thể được báo hiệu thông qua luồng bit. Ví dụ, thông tin này có thể chỉ báo việc khối mã hóa có được phân chia thành các khối hình vuông hay không hoặc việc khối mã hóa có được phân chia thành các khối không phải hình vuông hay không.

Ngoài ra, việc phân chia khối mã hóa thành các khối hình vuông có thể được thiết đặt là có độ ưu tiên cao hơn so với độ ưu tiên phân chia khối mã hóa thành các khối không phải hình vuông. Ví dụ, việc phân chia khối mã hóa thành các khối không phải hình vuông có thể được cho phép khi không được cho phép phân chia khối mã hóa thành các khối hình vuông.

Ngoài ra, kiểu phân chia của khối mã hóa có thể được xác định dựa trên kiểu phân chia của khối mã hóa nút cha mẹ. Ví dụ, có thể được thiết đặt là phân chia khối mã hóa thành các khối hình vuông khi khối mã hóa nút cha mẹ được phân chia dựa trên cây tam phân. Mặt khác, có thể được thiết đặt là phân chia khối mã hóa thành các khối không phải hình vuông khi khối mã hóa nút cha mẹ được phân chia dựa trên cây nhị phân hoặc cây tam phân.

Dự báo liên cấu trúc là chế độ lập mã dự báo mà dự báo khối hiện thời bằng cách sử dụng thông tin của hình trước đó. Ví dụ, khối ở cùng vị trí như khối hiện thời trong hình trước đó (sau đây, khối được sắp xếp cùng một chỗ) có thể được thiết đặt là khối dự báo của khối hiện thời. Sau đây, khối dự báo được tạo ra dựa trên khối ở cùng vị trí như khối hiện thời sẽ được gọi là khối dự báo được sắp xếp cùng một chỗ.

Mặt khác, khi đối tượng hiện có trong hình trước đó đã được di chuyển tới vị trí khác trong hình hiện thời, thì khối hiện thời có thể được dự báo một cách hiệu quả bằng cách sử dụng chuyển động của đối tượng. Ví dụ, khi chiều di chuyển và kích thước của đối tượng có thể được biết bằng cách so sánh hình trước đó và hình hiện thời, thì khối dự báo (hoặc hình dự báo) của khối hiện thời có thể được tạo ra có tính đến thông tin chuyển động của đối tượng. Sau đây, khối dự báo được tạo ra sử dụng thông tin chuyển động có thể được gọi là khối dự báo chuyển động.

Khối còn dư có thể được tạo ra bằng cách trừ đi khối dự báo khỏi khối hiện thời. Vào lúc này, khi có chuyển động của đối tượng, thì năng lượng của khối còn dư có thể được làm giảm xuống bằng cách sử dụng khối dự báo chuyển động thay vì khối dự báo được sắp xếp cùng một chỗ, và do đó, hiệu năng nén của khối còn dư có thể được cải thiện.

Như được mô tả ở trên, việc tạo ra khối dự báo bằng cách sử dụng thông tin chuyển động có thể được gọi là dự báo bù chuyển động. Trong hầu hết hoạt động dự báo liên cấu trúc, khối dự báo có thể được tạo ra dựa trên dự báo bù chuyển động.

Thông tin chuyển động có thể bao gồm ít nhất một trong số vector chuyển động, chỉ số hình tham chiếu, chiều dự báo, và chỉ số trọng số hai chiều. Vector chuyển động biểu diễn chiều di chuyển và kích thước của đối tượng. Chỉ số hình tham chiếu định rõ hình tham chiếu của khối hiện thời trong số các hình tham chiếu được chứa trong danh sách hình tham chiếu. Chiều dự báo chỉ báo một chiều dự báo bất kỳ trong số dự báo L0 một chiều, dự báo L1 một chiều, và dự báo hai chiều (L0 dự báo L0 và dự báo L1). Theo chiều dự báo của khối hiện thời, ít nhất một trong số thông tin chuyển động theo chiều L0 và thông tin chuyển động theo chiều L1 có thể được sử dụng. Chỉ số trọng số hai chiều định rõ giá trị trọng số được áp dụng cho L0 khối dự báo và giá trị trọng số được áp dụng cho khối dự báo L1.

Fig.9 là hình vẽ lưu đồ thể hiện phương pháp dự báo liên cấu trúc theo một phương án của sáng chế.

Tham khảo Fig.9, phương pháp dự báo liên cấu trúc bao gồm các bước để xác định chế độ dự báo liên cấu trúc của khối hiện thời (S901), đạt được thông tin chuyển động của khối hiện thời theo chế độ dự báo liên cấu trúc được xác định (S902), và thực hiện dự báo bù chuyển động đối với khối hiện thời dựa trên thông tin chuyển động đạt được (S903).

Ở đây, chế độ dự báo liên cấu trúc biểu diễn các kỹ thuật khác nhau để xác định thông tin chuyển động của khối hiện thời, và có thể bao gồm chế độ dự báo liên cấu trúc mà sử dụng thông tin chuyển động tịnh tiến và chế độ dự báo liên cấu trúc mà sử dụng thông tin chuyển động afin. Ví dụ, chế độ dự báo liên cấu trúc sử dụng thông tin chuyển động tịnh tiến có thể bao gồm chế độ hợp nhất và chế độ dự báo vector chuyển động cải tiến, và chế độ dự báo liên cấu trúc sử dụng thông tin chuyển động afin có thể bao gồm chế độ hợp nhất afin và chế độ dự báo vector chuyển động cải tiến afin. Thông tin chuyển động của khối hiện thời có thể được xác định dựa trên khối lân cận gần kề với khối hiện thời hoặc thông tin được phân tích cú pháp từ luồng bit theo chế độ dự báo liên cấu trúc.

Sau đây, phương pháp dự báo liên cấu trúc sử dụng thông tin chuyển động afin sẽ được mô tả chi tiết.

Fig.10 là hình vẽ thể hiện các chuyển động không tuyến tính của đối tượng.

Chuyển động không tuyến tính của đối tượng có thể được tạo ra trong video. Ví dụ, như được thể hiện trong ví dụ trên Fig.10, chuyển động không tuyến tính của đối tượng, chẳng hạn như phóng to, thu nhỏ, xoay, biến đổi afin hoặc chuyển động tương tự của camera, có thể xảy ra. Khi chuyển động không tuyến tính của đối tượng xảy ra, chuyển động của đối tượng không thể được biểu diễn một cách hiệu quả với vector chuyển động tịnh tiến. Do đó, hiệu quả lập mã có thể được cải thiện bằng cách sử dụng chuyển động afin thay vì chuyển động tịnh tiến trong vùng mà chuyển động không tuyến tính của đối tượng xảy ra.

Fig.11 là hình vẽ lưu đồ thể hiện phương pháp dự báo liên cấu trúc dựa trên chuyển động afin theo một phương án của sáng chế.

Việc kỹ thuật dự báo liên cấu trúc dựa trên chuyển động afin có được áp dụng cho khối hiện thời hay không có thể được xác định dựa trên thông tin được phân tích cú pháp

từ luồng bit. Cụ thể là, việc kỹ thuật dự báo liên cấu trúc dựa trên chuyển động affin có được áp dụng cho khối hiện thời hay không có thể được xác định dựa trên ít nhất một trong số cờ chỉ báo việc chế độ hợp nhất affin có được áp dụng cho khối hiện thời hay không và cờ chỉ báo việc chế độ dự báo vector chuyển động cải tiến affin có được áp dụng cho khối hiện thời hay không.

Khi kỹ thuật dự báo liên cấu trúc dựa trên chuyển động affin được áp dụng cho khối hiện thời, thì mô hình chuyển động affin của khối hiện thời có thể được xác định (S1101). Mô hình chuyển động affin có thể được xác định là ít nhất một trong số mô hình chuyển động affin sáu thông số và mô hình chuyển động affin bốn thông số. Mô hình chuyển động affin sáu thông số biểu diễn chuyển động affin sử dụng sáu thông số, và mô hình chuyển động affin bốn thông số biểu diễn chuyển động affin sử dụng bốn thông số.

Phương trình 1 biểu diễn chuyển động affin sử dụng sáu thông số. Chuyển động affin biểu diễn chuyển động không tịnh tiến đối với vùng định trước được xác định bởi các vector hạt giống affin.

【Phương trình 1】

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

Khi chuyển động affin được biểu diễn sử dụng sáu thông số, thì chuyển động phức tạp có thể được biểu diễn. Tuy nhiên, khi số lượng các bit được yêu cầu để lập mã mỗi thông số tăng lên, thì hiệu quả lập mã có thể bị hạ thấp. Do đó, chuyển động affin có thể được biểu diễn sử dụng bốn thông số. Phương trình 2 biểu diễn chuyển động affin sử dụng bốn thông số.

【Phương trình 2】

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

Thông tin để xác định mô hình chuyển động affin của khối hiện thời có thể được lập mã và được báo hiệu thông qua luồng bit. Ví dụ, thông tin có thể là cờ 1-bit ‘affine_type_flag’. Khi giá trị của cờ là 0, thì có thể chỉ báo rằng mô hình chuyển động affin bốn thông số được áp dụng, và khi giá trị của cờ là 1, thì có thể chỉ báo rằng mô hình chuyển

động afin sáu thông số được áp dụng. Cờ này có thể được lập mã theo đơn vị phân chia, khung lát, hoặc khối (ví dụ, theo đơn vị khối mã hóa hoặc cây mã hóa). Khi cờ này được báo hiệu ở mức phân chia, thì mô hình chuyển động afin được xác định ở mức phân chia có thể được áp dụng cho tất cả các khối thuộc về phân chia này.

Ngoài ra, mô hình chuyển động afin của khối hiện thời có thể được xác định dựa trên chế độ dự báo liên cấu trúc afin của khối hiện thời. Ví dụ, khi chế độ hợp nhất afin được áp dụng, thì mô hình chuyển động afin của khối hiện thời có thể được xác định là mô hình chuyển động 4 thông số. Mặt khác, khi chế độ dự báo vector chuyển động cải tiến afin được áp dụng, thì thông tin để xác định mô hình chuyển động afin của khối hiện thời có thể được lập mã và được báo hiệu thông qua luồng bit. Ví dụ, khi chế độ dự báo vector chuyển động cải tiến afin được áp dụng cho khối hiện thời, thì mô hình chuyển động afin của khối hiện thời có thể được xác định dựa trên cờ 1-bit 'affine_type_flag'.

Kế tiếp, vector hạt giống afin của khối hiện thời có thể được suy ra (S1102). Khi mô hình chuyển động afin bốn thông số được chọn lựa, thì các vector chuyển động ở hai điểm điều khiển của khối hiện thời có thể được suy ra. Mặt khác, khi mô hình chuyển động afin sáu thông số được chọn lựa, thì các vector chuyển động ở ba điểm điều khiển của khối hiện thời có thể được suy ra. Vector chuyển động ở điểm điều khiển có thể được gọi là vector hạt giống afin. Điểm điều khiển có thể bao gồm ít nhất một trong số góc bên trái-trên cùng, góc bên phải-trên cùng, và góc bên trái-dưới cùng của khối hiện thời.

Fig.12 là hình vẽ thể hiện một ví dụ về các vector hạt giống afin của mỗi mô hình chuyển động afin.

Trong mô hình chuyển động afin bốn thông số, các vector hạt giống afin có thể được suy ra dành cho hai vị trí trong số góc bên trái-trên cùng, góc bên phải-trên cùng, và góc bên trái-dưới cùng. Ví dụ, như được thể hiện trong ví dụ trên Fig.12(a), khi mô hình chuyển động afin bốn thông số được chọn lựa, thì vector afin có thể được suy ra sử dụng vector hạt giống afin sv0 dành cho góc bên trái-trên cùng của khối hiện thời (ví dụ, mẫu bên trái-trên cùng (x_0, y_0)) và vector hạt giống afin sv1 dành cho góc bên phải-trên cùng của khối hiện thời (ví dụ, mẫu bên phải-trên cùng (x_1, y_1)). Cũng có thể sử dụng vector hạt giống afin dành cho góc bên trái-dưới cùng thay vì vector hạt giống afin dành cho góc bên trái-trên

cùng, hoặc sử dụng vector hạt giống afin dành cho góc bên trái-dưới cùng thay vì vector hạt giống afin dành cho góc bên phải-trên cùng.

Trong mô hình chuyển động afin sáu thông số, các vector hạt giống afin có thể được suy ra dành cho góc bên trái-trên cùng, góc bên phải-trên cùng, và góc bên trái-dưới cùng. Ví dụ, như được thể hiện trong ví dụ trên Fig.12(b), khi mô hình chuyển động afin sáu thông số được chọn lựa, thì vector afin có thể được suy ra sử dụng vector hạt giống afin sv_0 dành cho góc bên trái-trên cùng của khối hiện thời (ví dụ, mẫu bên trái-trên cùng (x_0, y_0)), vector hạt giống afin sv_1 dành cho góc bên phải-trên cùng của khối hiện thời (ví dụ, mẫu bên phải-trên cùng (x_1, y_1)), và vector hạt giống afin sv_2 dành cho góc bên trái-dưới cùng của khối hiện thời (ví dụ, mẫu bên trái-dưới cùng (x_2, y_2)).

Trong các phương án được mô tả bên dưới, trong mô hình chuyển động afin bốn thông số, các vector hạt giống afin của điểm điều khiển bên trái-trên cùng và điểm điều khiển bên phải-trên cùng sẽ lần lượt được gọi là vector hạt giống afin thứ nhất và vector hạt giống afin thứ hai. Theo các phương án sử dụng vector hạt giống afin thứ nhất và vector hạt giống afin thứ hai được mô tả bên dưới, ít nhất một trong số vector hạt giống afin thứ nhất và vector hạt giống afin thứ hai có thể được thay thế với vector hạt giống afin của điểm điều khiển bên trái-dưới cùng (vector hạt giống afin thứ ba) hoặc vector hạt giống afin của điểm điều khiển bên phải-dưới cùng (vector hạt giống afin thứ tư).

Ngoài ra, trong mô hình chuyển động afin sáu thông số, các vector hạt giống afin của điểm điều khiển bên trái-trên cùng, điểm điều khiển bên phải-trên cùng, và điểm điều khiển bên trái-dưới cùng sẽ lần lượt được gọi là vector hạt giống afin thứ nhất, vector hạt giống afin thứ hai, và vector hạt giống afin thứ ba. Theo các phương án sử dụng vector hạt giống afin thứ nhất, vector hạt giống afin thứ hai, và vector hạt giống afin thứ ba được mô tả bên dưới, ít nhất một trong số vector hạt giống afin thứ nhất, thì vector hạt giống afin thứ hai, và vector hạt giống afin thứ ba có thể được thay thế với vector hạt giống afin của điểm điều khiển bên phải-dưới cùng (vector hạt giống afin thứ tư).

Vector afin có thể được suy ra cho mỗi khối con bằng cách sử dụng các vector hạt giống afin (S1103). Ở đây, vector afin biểu diễn vector chuyển động tịnh tiến được suy ra dựa trên

các vector hạt giống afin. Vector afin của khối con có thể được gọi là vector chuyển động khối con afin hoặc vector chuyển động khối con.

Fig.13 là hình vẽ thể hiện một ví dụ về các vector afin của các khối con trong mô hình chuyển động 4 thông số.

Vector afin của khối con có thể được suy ra dựa trên vị trí của điểm điều khiển, vị trí của khối con, và vector hạt giống afin. Ví dụ, phương trình 3 thể hiện một ví dụ để suy ra vector khối con afin.

【Phương trình 3】

$$\begin{aligned} v_x &= \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(y_1 - y_0)}(y - y_0) + sv_{0x} \\ v_y &= \frac{(sv_{1y} - sv_{0y})}{(y_1 - y_0)}(y - y_0) + \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(x - x_0) + sv_{0y} \end{aligned}$$

Trong phương trình 3, (x, y) thể hiện vị trí của khối con. Ở đây, vị trí của khối con chỉ báo vị trí của mẫu tham chiếu được chứa trong khối con. Mẫu tham chiếu có thể là mẫu được đặt ở vị trí góc bên trái-trên cùng của khối con, hoặc mẫu mà trong đó ít nhất một trong số các tọa độ trục x và trục y là điểm trung tâm. (x₀, y₀) thể hiện vị trí của điểm điều khiển thứ nhất, và (sv_{0x}, sv_{0y}) thể hiện vector hạt giống afin thứ nhất. Ngoài ra, (x₁, y₁) thể hiện vị trí của điểm điều khiển thứ hai, và (sv_{1x}, sv_{1y}) thể hiện vector hạt giống afin thứ hai.

Khi điểm điều khiển thứ nhất và điểm điều khiển thứ hai lần lượt tương ứng với góc bên trái-trên cùng và góc bên phải-trên cùng của khối hiện thời, x₁-x₀ có thể được thiết đặt là giá trị bằng chiều rộng của khối hiện thời.

Say đây, dự báo bù chuyển động dành cho mỗi khối con có thể được thực hiện sử dụng vector afin của mỗi khối con (S1104). Kết quả của việc thực hiện dự báo bù chuyển động, khối dự báo dành cho mỗi khối con có thể được tạo ra. Các khối dự báo của các khối con có thể được thiết đặt là các khối dự báo của khối hiện thời.

Kế tiếp, phương pháp dự báo liên cấu trúc sử dụng thông tin chuyển động tịnh tiến sẽ được mô tả chi tiết.

Thông tin chuyển động của khối hiện thời có thể được suy ra từ thông tin chuyển động của khối khác. Ở đây, khối khác có thể là khối được lập mã/giải mã bằng cách dự báo liên cấu trúc trước khối hiện thời. Việc thiết đặt thông tin chuyển động của khối hiện thời bằng

thông tin chuyển động của khối khác có thể được định nghĩa là chế độ hợp nhất. Ngoài ra, việc thiết đặt vector chuyển động của khối khác là giá trị dự báo của vector chuyển động của khối hiện thời có thể được định nghĩa là chế độ dự báo vector chuyển động cải tiến.

Fig.14 là hình vẽ lưu đồ thể hiện quy trình để suy ra thông tin chuyển động của khối hiện thời sử dụng chế độ hợp nhất.

Ứng cử hợp nhất của khối hiện thời có thể được suy ra (S1401). Ứng cử hợp nhất của khối hiện thời có thể được suy ra từ khối được lập mã/giải mã bằng cách dự báo liên cấu trúc trước khối hiện thời.

Fig.15 là hình vẽ thể hiện một ví dụ về các khối ứng cử được sử dụng để suy ra ứng cử hợp nhất.

Các khối ứng cử có thể bao gồm ít nhất một trong số các khối lân cận bao gồm mẫu gần kề với khối hiện thời hoặc các khối không lân cận bao gồm mẫu không gần kề với khối hiện thời. Sau đây, các mẫu để xác định các khối ứng cử được định nghĩa là các mẫu tham chiếu. Ngoài ra, mẫu tham chiếu gần kề với khối hiện thời được gọi là mẫu tham chiếu lân cận, và mẫu tham chiếu không gần kề với khối hiện thời được gọi là mẫu tham chiếu không lân cận.

Mẫu tham chiếu lân cận có thể được chứa trong cột lân cận của cột ngoài cùng bên trái của khối hiện thời hoặc hàng lân cận của hàng trên cùng của khối hiện thời. Ví dụ, khi các tọa độ của mẫu bên trái-trên cùng của khối hiện thời là $(0, 0)$, ít nhất một trong số khối bao gồm mẫu tham chiếu ở vị trí của $(-1, H-1)$, khối bao gồm mẫu tham chiếu ở vị trí của $(W-1, -1)$, khối bao gồm mẫu tham chiếu ở vị trí của $(W, -1)$, khối bao gồm mẫu tham chiếu ở vị trí của $(-1, H)$, và khối bao gồm mẫu tham chiếu ở vị trí của $(-1, -1)$ có thể được sử dụng làm khối ứng cử. Tham khảo hình vẽ, các khối lân cận có chỉ số 0 đến 4 có thể được sử dụng làm các khối ứng cử.

Mẫu tham chiếu không lân cận biểu diễn mẫu mà có ít nhất một trong số khoảng cách trục x và khoảng cách trục y từ mẫu tham chiếu gần kề với khối hiện thời có giá trị định trước. Ví dụ, ít nhất một trong số khối bao gồm mẫu tham chiếu mà có khoảng cách trục x từ mẫu tham chiếu bên trái là giá trị định trước, khối bao gồm mẫu không lân cận mà có khoảng cách trục y từ mẫu tham chiếu trên cùng là giá trị định trước, và khối bao gồm mẫu

không lân cận mà có khoảng cách trục x và khoảng cách trục y từ mẫu tham chiếu bên trái-trên cùng là các giá trị định trước có thể được sử dụng làm khối ứng cử. Các giá trị định trước có thể là số tự nhiên chẳng hạn như 4, 8, 12, 16 hoặc số tương tự. Tham khảo hình vẽ, ít nhất một trong số các khối có chỉ số 5 đến 26 có thể được sử dụng làm khối lân cận.

Mẫu không được đặt ở vị trí trên cùng một đường nằm dọc, đường nằm ngang, hoặc đường chéo là mẫu tham chiếu lân cận có thể được thiết đặt là mẫu tham chiếu không lân cận.

Fig.16 là hình vẽ thể hiện các vị trí của các mẫu tham chiếu.

Như được thể hiện trong ví dụ trên Fig.16, các tọa độ x của các mẫu tham chiếu không lân cận trên cùng có thể được thiết đặt khác với các tọa độ x của các mẫu tham chiếu lân cận trên cùng. Ví dụ, khi vị trí của mẫu tham chiếu lân cận trên cùng là $(W-1, -1)$, thì vị trí của mẫu tham chiếu không lân cận trên cùng được tách riêng là N từ mẫu tham chiếu lân cận trên cùng trên trục y có thể được thiết đặt là $((W/2)-1, -1-N)$, và vị trí của mẫu tham chiếu không lân cận trên cùng được tách riêng là $2N$ từ mẫu tham chiếu lân cận trên cùng trên trục y có thể được thiết đặt là $(0, -1-2N)$. Tức là, vị trí của mẫu tham chiếu không gần kề có thể được xác định dựa trên vị trí của mẫu tham chiếu gần kề và khoảng cách từ mẫu tham chiếu gần kề.

Sau đây, khối ứng cử bao gồm mẫu tham chiếu lân cận trong số các khối ứng cử được gọi là khối lân cận, và khối bao gồm mẫu tham chiếu không lân cận được gọi là khối không lân cận.

Khi khoảng cách giữa khối hiện thời và khối ứng cử lớn hơn hoặc bằng giá trị ngưỡng, thì khối ứng cử có thể được thiết đặt là không khả dụng như ứng cử hợp nhất. Giá trị ngưỡng có thể được xác định dựa trên kích thước của đơn vị cây mã hóa. Ví dụ, giá trị ngưỡng có thể được thiết đặt là chiều cao (`ctu_height`) của đơn vị cây mã hóa hoặc giá trị được nhận bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi chiều cao (ví dụ, $\text{ctu_height} \pm N$) của đơn vị cây mã hóa. Độ lệch N là giá trị được định trước trong bộ lập mã và bộ giải mã, và có thể được thiết đặt là 4, 8, 16, 32 hoặc `ctu_height`.

Khi độ chênh lệch giữa tọa độ trục y của khối hiện thời và tọa độ trục y của mẫu được chứa trong khối ứng cử lớn hơn so với giá trị ngưỡng, thì khối ứng cử có thể được xác định là không khả dụng như ứng cử hợp nhất.

Ngoài ra, khối ứng cử mà không thuộc về cùng một đơn vị cây mã hóa như khối hiện thời có thể được thiết đặt là không khả dụng như ứng cử hợp nhất. Ví dụ, khi mẫu tham chiếu lệch khối biên trên cùng của đơn vị cây mã hóa mà khối hiện thời thuộc về, thì khối ứng cử bao gồm mẫu tham chiếu có thể được thiết đặt là không khả dụng như ứng cử hợp nhất.

Khi biên trên cùng của khối hiện thời gần kề với biên trên cùng của đơn vị cây mã hóa, thì nhiều khối ứng cử được xác định là không khả dụng như ứng cử hợp nhất, và do đó hiệu quả lập mã/giải mã của khối hiện thời có thể giảm xuống. Để giải quyết vấn đề này, các khối ứng cử có thể được thiết đặt sao cho số lượng các khối ứng cử được đặt ở vị trí trên cạnh bên trái của khối hiện thời lớn hơn so với số lượng các khối ứng cử được đặt ở vị trí trên phần trên cùng của khối hiện thời.

Fig.17 là hình vẽ thể hiện một ví dụ về các khối ứng cử được sử dụng để suy ra ứng cử hợp nhất.

Như được thể hiện trong ví dụ trên Fig.17, các khối trên cùng thuộc về N cột khối trên cùng của khối hiện thời và các khối cạnh bên trái thuộc về M cột khối cạnh bên trái của khối hiện thời có thể được thiết đặt là các khối ứng cử. Vào lúc này, số lượng các khối ứng cử cạnh bên trái có thể được thiết đặt là lớn hơn so với số lượng các khối ứng cử trên cùng bằng cách thiết đặt M lớn hơn N.

Ví dụ, độ chênh lệch giữa tọa độ trục y của mẫu tham chiếu trong khối hiện thời và tọa độ trục y của khối trên cùng mà có thể được sử dụng làm khối ứng cử có thể được thiết đặt không vượt quá N lần chiều cao của khối hiện thời. Ngoài ra, độ chênh lệch giữa tọa độ trục x của mẫu tham chiếu trong khối hiện thời và tọa độ trục x của khối cạnh bên trái mà có thể được sử dụng làm khối ứng cử có thể được thiết đặt không vượt quá M lần chiều rộng của khối hiện thời.

Ví dụ, trong ví dụ được thể hiện trên Fig.17, được thể hiện rằng cách khối thuộc về hai cột khối trên cùng của khối hiện thời và các khối thuộc về năm cột khối bên trái của khối hiện thời được thiết đặt là các khối ứng cử.

Ví dụ khác, khi khối ứng cử không thuộc về đơn vị cây mã hóa giống như của khối hiện thời, thì ứng cử hợp nhất có thể được suy ra sử dụng khối thuộc về cùng một đơn vị cây mã hóa như khối hiện thời hoặc khối bao gồm mẫu tham chiếu gần kề với biên của đơn vị cây mã hóa, thay vì khối ứng cử.

Fig.18 là hình vẽ thể hiện một ví dụ mà trong đó vị trí của mẫu tham chiếu được thay đổi.

Khi mẫu tham chiếu được chứa trong đơn vị cây mã hóa khác với khối hiện thời, và mẫu tham chiếu không gần kề với biên của đơn vị cây mã hóa, thì khối ứng cử có thể được xác định sử dụng mẫu tham chiếu gần kề với biên của đơn vị cây mã hóa, thay vì mẫu tham chiếu.

Ví dụ, trong các ví dụ được thể hiện trên Fig.18(a) và Fig.18(b), khi biên trên cùng của khối hiện thời và biên trên cùng của đơn vị cây mã hóa tiếp xúc với nhau, thì các mẫu tham chiếu trên phần trên cùng của khối hiện thời thuộc về đơn vị cây mã hóa khác với khối hiện thời. Trong số các mẫu tham chiếu thuộc về đơn vị cây mã hóa khác với khối hiện thời, thì mẫu tham chiếu không gần kề với biên trên cùng của đơn vị cây mã hóa có thể được thay thế với mẫu gần kề với biên trên cùng của đơn vị cây mã hóa.

Ví dụ, như được thể hiện trong ví dụ trên Fig.18(a), mẫu tham chiếu ở vị trí 6 được thay thế với mẫu ở vị trí 6' được đặt ở vị trí biên trên cùng của đơn vị cây mã hóa, và như được thể hiện trong ví dụ trên Fig.18(b), mẫu tham chiếu ở vị trí 15 được thay thế với mẫu ở vị trí 15' được đặt ở vị trí biên trên cùng của đơn vị cây mã hóa. Vào lúc này, tọa độ y của mẫu thay thế được thay đổi tới vị trí gần kề với đơn vị cây mã hóa, và tọa độ x của mẫu thay thế có thể được thiết đặt bằng với mẫu tham chiếu. Ví dụ, mẫu ở vị trí 6' có thể có cùng tọa độ x như mẫu ở vị trí 6, và mẫu ở vị trí 15' có thể có cùng tọa độ x như mẫu ở vị trí 15.

Ngoài ra, giá trị được nhận bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi tọa độ x của mẫu tham chiếu có thể được thiết đặt là tọa độ x của mẫu thay thế. Ví dụ, khi các tọa

độ x của mẫu tham chiếu lân cận được đặt ở vị trí trên cùng của khối hiện thời và mẫu tham chiếu không lân cận giống nhau, thì giá trị được nhận bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi tọa độ x của mẫu tham chiếu có thể được thiết đặt là tọa độ x của mẫu thay thế. Việc này để ngăn mẫu thay thế thay thế mẫu tham chiếu không lân cận không được đặt ở cùng một vị trí như mẫu tham chiếu không lân cận hoặc mẫu tham chiếu lân cận khác.

Fig.19 là hình vẽ thể hiện một ví dụ mà trong đó vị trí của mẫu tham chiếu được thay đổi.

Thay thế mẫu tham chiếu mà được chứa trong đơn vị cây mã hóa khác với khối hiện thời và không gần kề với biên của đơn vị cây mã hóa với mẫu được đặt ở vị trí biên của đơn vị cây mã hóa, thì giá trị được nhận bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi tọa độ x của mẫu tham chiếu có thể được thiết đặt là tọa độ x của mẫu thay thế.

Ví dụ, trong ví dụ được thể hiện trên Fig.19, mẫu tham chiếu ở vị trí 6 và mẫu tham chiếu ở vị trí 15 lần lượt có thể được thay thế với mẫu ở vị trí 6' và mẫu ở vị trí 15', mà các tọa độ y giống như của hàng gần kề với biên trên cùng của đơn vị cây mã hóa. Vào lúc này, tọa độ x của mẫu ở vị trí 6' có thể được thiết đặt là giá trị nhận được bằng cách trừ đi $W/2$ khỏi tọa độ x của mẫu tham chiếu ở vị trí 6, và tọa độ x của mẫu ở vị trí 15' có thể được thiết đặt là giá trị nhận được bằng cách trừ đi $W-1$ khỏi tọa độ x của mẫu tham chiếu ở vị trí 15.

Không giống như các ví dụ được thể hiện trên Fig.18 và Fig.19, tọa độ y của hàng được đặt ở vị trí trên phần trên cùng của hàng trên cùng của khối hiện thời hoặc tọa độ y của biên trên cùng của đơn vị cây mã hóa có thể được thiết đặt là tọa độ y của mẫu thay thế.

Mặc dù không được thể hiện trên hình vẽ, mẫu thay thế mẫu tham chiếu có thể được xác định dựa trên biên cạnh bên trái của đơn vị cây mã hóa. Ví dụ, khi mẫu tham chiếu không được chứa trong cùng một đơn vị cây mã hóa như khối hiện thời và không gần kề với biên cạnh bên trái của đơn vị cây mã hóa, thì mẫu tham chiếu có thể được thay thế với mẫu gần kề với biên cạnh bên trái của đơn vị cây mã hóa. Vào lúc này, mẫu thay thế có thể có tọa độ y giống như của mẫu tham chiếu, hoặc có thể có tọa độ y được nhận bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi tọa độ y của mẫu tham chiếu.

Sau đây, khối bao gồm mẫu thay thế có thể được thiết đặt là khối ứng cử, và ứng cử hợp nhất của khối hiện thời có thể được suy ra dựa trên khối ứng cử.

Ứng cử hợp nhất cũng có thể được suy ra từ khối lân cận về mặt thời gian được chứa trong hình khác với khối hiện thời. Ví dụ, ứng cử hợp nhất có thể được suy ra từ khối được sắp xếp cùng một chỗ được chứa trong hình được sắp xếp cùng một chỗ.

Thông tin chuyển động của ứng cử hợp nhất có thể được thiết đặt bằng với thông tin chuyển động của khối ứng cử. Ví dụ, ít nhất một trong số vectơ chuyển động, chỉ số hình tham chiếu, chiều dự báo, và chỉ số trọng số hai hướng của khối ứng cử có thể được thiết đặt là thông tin chuyển động của ứng cử hợp nhất.

Danh sách ứng cử hợp nhất bao gồm các ứng cử hợp nhất có thể được tạo ra (S1402). Các ứng cử hợp nhất có thể được chia thành ứng cử hợp nhất gần kề được suy ra từ khối lân cận gần kề với khối hiện thời và ứng cử hợp nhất không gần kề được suy ra từ khối không lân cận.

Các chỉ số của các ứng cử hợp nhất trong danh sách ứng cử hợp nhất có thể được gán theo thứ bậc định trước. Ví dụ, chỉ số được gán cho ứng cử hợp nhất gần kề có thể có giá trị nhỏ hơn so với chỉ số được gán cho ứng cử hợp nhất không gần kề. Ngoài ra, chỉ số có thể được gán cho mỗi ứng cử hợp nhất dựa trên chỉ số của mỗi khối được thể hiện trên Fig.15 hoặc Fig.17.

Khi nhiều ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất, thì ít nhất một trong số nhiều ứng cử hợp nhất có thể được chọn lựa (S1403). Vào lúc này, thông tin chỉ báo việc thông tin chuyển động của khối hiện thời có được suy ra từ ứng cử hợp nhất gần kề hay không có thể được báo hiệu thông qua luồng bit. Thông tin có thể là cờ 1-bit. Ví dụ, phần tử cú pháp `isAdjacentMergeFlag` chỉ báo việc thông tin chuyển động của khối hiện thời có được suy ra từ ứng cử hợp nhất gần kề hay không có thể được báo hiệu thông qua luồng bit. Khi giá trị của phần tử cú pháp `isAdjacentMergeFlag` là 1, thì thông tin chuyển động của khối hiện thời có thể được suy ra dựa trên ứng cử hợp nhất gần kề. Mặt khác, khi giá trị của phần tử cú pháp `isAdjacentMergeFlag` là 0, thì thông tin chuyển động của khối hiện thời có thể được suy ra dựa trên ứng cử hợp nhất không gần kề.

Bảng 1 thể hiện bảng cú pháp bao gồm phần tử cú pháp `isAdjacentMergeFlag`.

【Bảng 1】

coding_unit (x0, y0, cbWidth, cbHeight, treeType) {	Bộ mô tả
if (slice_type != I) {	
pred_mode_flag	ae(v)
}	
if (CuPredMode[x0][y0] == MODE_INTRA) {	
if (treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag[x0][y0]	
if (intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
nếu không thì	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if (treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else {	
if (cu_skip_flag[x0][y0]) {	
if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjacentMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)
}	
}	
} else { /* MODE_INTER*/	
merge_flag[x0][y0]	ae(v)
if (merge_flag[x0][y0]){	
if (MaxNumMergeCand > 1){	

isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)
}	
}	
}	
if (CuPredMode[x0][y0] != MODE_INTRA)	
cu_cbf	ae(v)
if (cu_cbf) {	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

Thông tin để định rõ một ứng cử bất kỳ trong số nhiều ứng cử hợp nhất có thể được báo hiệu thông qua luồng bit. Ví dụ, thông tin chỉ báo chỉ số của một ứng cử bất kỳ trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất có thể được báo hiệu thông qua luồng bit.

Khi isAdjacentMergeflag là 1, thì phần tử cú pháp merge_idx định rõ một ứng cử bất kỳ trong số các ứng cử hợp nhất gần kề có thể được báo hiệu. Giá trị tối đa của phần tử cú pháp merge_idx có thể được thiết đặt là giá trị được nhận bằng cách trừ đi 1 khỏi số lượng các ứng cử hợp nhất gần kề.

Khi isAdjacentMergeflag là 0, thì phần tử cú pháp NA_merge_idx định rõ một ứng cử bất kỳ trong số các ứng cử hợp nhất không gần kề có thể được báo hiệu. Phần tử cú pháp NA_merge_idx biểu diễn giá trị nhận được bằng cách trừ đi số lượng các ứng cử hợp nhất gần kề khỏi chỉ số của ứng cử hợp nhất không gần kề. Bộ giải mã có thể chọn lựa ứng cử hợp nhất không gần kề bằng cách cộng số lượng các ứng cử hợp nhất gần kề vào chỉ số được định rõ bởi NA_merge_idx.

Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với giá trị ngưỡng, thì ứng cử hợp nhất được chứa trong danh sách thông tin chuyển

động liên vùng có thể được cộng vào danh sách ứng cử hợp nhất. Ở đây, giá trị ngưỡng có thể là số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm hoặc giá trị nhận được bằng cách trừ đi độ lệch khỏi số lượng tối đa của các ứng cử hợp nhất. Độ lệch này có thể là số tự nhiên chẳng hạn như 1, 2 hoặc số tương tự. Danh sách thông tin chuyển động liên vùng có thể bao gồm ứng cử hợp nhất được suy ra dựa trên khối được lập mã/giải mã trước khối hiện thời.

Danh sách thông tin chuyển động liên vùng bao gồm ứng cử hợp nhất được suy ra từ khối được lập mã/giải mã dựa trên dự báo liên cấu trúc trong hình hiện thời. Ví dụ, thông tin chuyển động của ứng cử hợp nhất được chứa trong danh sách thông tin chuyển động liên vùng có thể được thiết đặt bằng với thông tin chuyển động của khối được lập mã/giải mã dựa trên dự báo liên cấu trúc. Ở đây, thông tin chuyển động có thể bao gồm ít nhất một trong số vectơ chuyển động, chỉ số hình tham chiếu, chiều dự báo, và chỉ số trọng số hai hướng.

Để thuận tiện cho việc giải thích, ứng cử hợp nhất được chứa trong danh sách thông tin chuyển động liên vùng sẽ được gọi là ứng cử hợp nhất liên vùng.

Số lượng tối đa của các ứng cử hợp nhất mà danh sách thông tin chuyển động liên vùng có thể bao gồm có thể được định trước bởi bộ lập mã và bộ giải mã. Ví dụ, số lượng tối đa của các ứng cử hợp nhất mà có thể được chứa trong danh sách thông tin chuyển động liên vùng có thể là 1, 2, 3, 4, 5, 6, 7, 8 hoặc lớn hơn (ví dụ, 16).

Ngoài ra, thông tin chỉ báo số lượng tối đa của các ứng cử hợp nhất trong danh sách thông tin chuyển động liên vùng có thể được báo hiệu thông qua luồng bit. Thông tin này có thể được báo hiệu ở mức trình tự, hình, hoặc phân chia.

Ngoài ra, số lượng tối đa của các ứng cử hợp nhất của danh sách thông tin chuyển động liên vùng có thể được xác định theo kích thước của hình, kích thước của phân chia, hoặc kích thước của đơn vị cây mã hóa.

Danh sách thông tin chuyển động liên vùng có thể được khởi tạo theo đơn vị hình, phân chia, khung lát, gạch lát, đơn vị cây mã hóa, hoặc đường đơn vị cây mã hóa (hàng hoặc cột). Ví dụ, khi phân chia được khởi tạo, thì danh sách thông tin chuyển động liên

vùng cũng được khởi tạo, và danh sách thông tin chuyển động liên vùng không thể bao gồm ứng cử hợp bất kỳ.

Ngoài ra, thông tin chỉ báo việc có khởi tạo hay không danh sách thông tin chuyển động liên vùng có thể được báo hiệu thông qua luồng bit. Thông tin này có thể được báo hiệu ở mức phần chia, khung lát, gạch lát, hoặc khối. Cho đến khi thông tin chỉ báo để khởi tạo danh sách thông tin chuyển động liên vùng, thì danh sách thông tin chuyển động liên vùng được tạo cấu hình trước đó có thể được sử dụng.

Ngoài ra, thông tin về ứng cử hợp nhất liên vùng khởi tạo có thể được báo hiệu thông qua tập thông số hình hoặc phần đầu phần chia. Mặc dù phần chia được khởi tạo, nhưng danh sách thông tin chuyển động liên vùng có thể bao gồm ứng cử hợp nhất liên vùng khởi tạo. Do đó, ứng cử hợp nhất liên vùng có thể được sử dụng cho khối mà là đích lập mã/giải mã thứ nhất trong phần chia.

Các khối được lập mã/giải mã theo thứ bậc lập mã/giải mã, và các khối được lập mã/giải mã dựa trên dự báo liên cấu trúc có thể được thiết đặt theo trình tự là ứng cử hợp nhất liên vùng theo thứ bậc lập mã/giải mã.

Fig.20 là hình vẽ lưu đồ thể hiện quy trình để cập nhật danh sách thông tin chuyển động liên vùng.

Khi dự báo liên cấu trúc được thực hiện trên khối hiện thời (S2001), thì ứng cử hợp nhất liên vùng có thể được suy ra dựa trên khối hiện thời (S2002). Thông tin chuyển động của ứng cử hợp nhất liên vùng có thể được thiết đặt bằng với thông tin chuyển động của khối hiện thời.

Khi danh sách thông tin chuyển động liên vùng trống (S2003), thì ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời có thể được cộng vào danh sách thông tin chuyển động liên vùng (S2004).

Khi danh sách thông tin chuyển động liên vùng đã bao gồm ứng cử hợp nhất liên vùng (S2003), thì kiểm dư thừa có thể được thực hiện trên thông tin chuyển động của khối hiện thời (hoặc ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời) (S2005). Kiểm dư thừa là để xác định việc thông tin chuyển động của ứng cử hợp nhất liên vùng được lưu trước đó trong danh sách thông tin chuyển động liên vùng và thông tin chuyển động của

khối hiện thời có giống nhau hay không. Kiểm dư thừa có thể được thực hiện trên tất cả các ứng cử hợp nhất liên vùng được lưu trước đó trong danh sách thông tin chuyển động liên vùng. Ngoài ra, kiểm dư thừa có thể được thực hiện trên các ứng cử hợp nhất liên vùng có chỉ số lớn hơn so với giá trị ngưỡng hoặc nhỏ hơn so với giá trị ngưỡng trong số các ứng cử hợp nhất liên vùng được lưu trước đó trong danh sách thông tin chuyển động liên vùng.

Khi ứng cử hợp nhất liên vùng có cùng thông tin chuyển động như thông tin chuyển động của khối hiện thời không được chứa, thì ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời có thể được cộng vào danh sách thông tin chuyển động liên vùng (S2008). Việc các ứng cử hợp nhất liên vùng có giống nhau hay không có thể được xác định dựa trên việc thông tin chuyển động (ví dụ, vectơ chuyển động và/hoặc chỉ số hình tham chiếu) của các ứng cử hợp nhất liên vùng có giống nhau hay không.

Vào lúc này, khi số lượng tối đa của các ứng cử hợp nhất liên vùng đã được lưu trong danh sách thông tin chuyển động liên vùng (S2006), ứng cử hợp nhất liên vùng cũ nhất được xóa (S2007), và ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời có thể được cộng vào danh sách thông tin chuyển động liên vùng (S2008).

Mỗi ứng cử hợp nhất liên vùng có thể được nhận dạng bằng chỉ số. Khi ứng cử hợp nhất liên vùng được suy ra từ khối hiện thời được cộng vào danh sách thông tin chuyển động liên vùng, thì chỉ số thấp nhất (ví dụ, 0) được gán cho ứng cử hợp nhất liên vùng, và các chỉ số của các ứng cử hợp nhất liên vùng được lưu trước đó có thể được tăng lên 1. Vào lúc này, khi số lượng tối đa của các ứng cử hợp nhất liên vùng đã được lưu trong danh sách thông tin chuyển động liên vùng, thì ứng cử hợp nhất liên vùng có chỉ số lớn nhất được loại bỏ.

Ngoài ra, khi ứng cử hợp nhất liên vùng được suy ra từ khối hiện thời được cộng vào danh sách thông tin chuyển động liên vùng, thì chỉ số lớn nhất có thể được gán cho ứng cử hợp nhất liên vùng. Ví dụ, khi số lượng các ứng cử hợp nhất liên vùng được lưu trước đó trong danh sách thông tin chuyển động liên vùng nhỏ hơn so với giá trị tối đa, thì chỉ số có cùng giá trị như số lượng các ứng cử hợp nhất liên vùng được lưu trước đó có thể được gán cho ứng cử hợp nhất liên vùng. Ngoài ra, khi số lượng các ứng cử hợp nhất liên vùng được lưu trước đó trong danh sách thông tin chuyển động liên vùng giống như giá trị tối đa, thì

chỉ số trừ đi 1 khỏi giá trị tối đa có thể được gán cho ứng cử hợp nhất liên vùng. Ngoài ra, ứng cử hợp nhất liên vùng có chỉ số nhỏ nhất được loại bỏ, và các chỉ số của các ứng cử hợp nhất liên vùng còn lại được lưu trước đó có thể bị giảm đi 1.

Fig.21 là hình vẽ thể hiện một phương án để cập nhật danh sách ứng cử hợp nhất liên vùng.

Được giả định rằng khi ứng cử hợp nhất liên vùng được suy ra từ khối hiện thời được cộng vào danh sách ứng cử hợp nhất liên vùng, thì chỉ số lớn nhất được gán cho ứng cử hợp nhất liên vùng. Ngoài ra, được giả định rằng số lượng tối đa của các ứng cử hợp nhất liên vùng đã được lưu trong danh sách ứng cử hợp nhất liên vùng.

Khi ứng cử hợp nhất liên vùng $HmvpCand[n + 1]$ được suy ra từ khối hiện thời được cộng vào danh sách ứng cử hợp nhất liên vùng $HmvpCandList$, thì ứng cử hợp nhất liên vùng $HmvpCand[0]$ có chỉ số nhỏ nhất trong số các ứng cử hợp nhất liên vùng được lưu trước đó được xóa, và các chỉ số của các ứng cử hợp nhất liên vùng còn lại có thể bị giảm đi 1. Ngoài ra, chỉ số của ứng cử hợp nhất liên vùng $HmvpCand[n + 1]$ được suy ra từ khối hiện thời có thể được thiết đặt là giá trị tối đa (n trong ví dụ được thể hiện trên Fig.21).

Khi ứng cử hợp nhất liên vùng giống như ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời được lưu trước đó (S2005), thì ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời có thể không được cộng vào danh sách thông tin chuyển động liên vùng (S2009).

Ngoài ra, khi ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời được cộng vào danh sách thông tin chuyển động liên vùng, thì ứng cử hợp nhất liên vùng được lưu trước đó mà giống như ứng cử hợp nhất liên vùng có thể được loại bỏ. Trong trường hợp này, nhận được hiệu quả giống như việc mới cập nhật chỉ số của ứng cử hợp nhất liên vùng được lưu trước đó.

Fig.22 là hình vẽ thể hiện một ví dụ mà trong đó chỉ số của ứng cử hợp nhất liên vùng được lưu trước đó được cập nhật.

Khi chỉ số của ứng cử hợp nhất liên vùng được lưu trước đó $mvCand$ mà giống như ứng cử hợp nhất liên vùng $mvCand$ được suy ra dựa trên khối hiện thời là $hIdx$, thì ứng cử

hợp nhất liên vùng được lưu trước đó được xóa, và các chỉ số của các ứng cử hợp nhất liên vùng có chỉ số lớn hơn so với $hIdx$ có thể bị giảm đi 1. Ví dụ, trong ví dụ được thể hiện trên Fig.22, được thể hiện rằng $HmvpCand[2]$ giống như $mvCand$ được xóa khỏi danh sách thông tin chuyển động liên vùng $HmvpCandList$, và các chỉ số của $HmvpCand[3]$ đến $HmvpCand[n]$ bị giảm đi 1.

Ngoài ra, ứng cử hợp nhất liên vùng $mvCand$ được suy ra dựa trên khối hiện thời có thể được cộng vào phần cuối của danh sách thông tin chuyển động liên vùng.

Ngoài ra, chỉ số được gán cho ứng cử hợp nhất liên vùng được lưu trước đó mà giống như ứng cử hợp nhất liên vùng được suy ra dựa trên khối hiện thời có thể được cập nhật. Ví dụ, chỉ số của ứng cử hợp nhất liên vùng được lưu trước đó có thể được thay đổi thành giá trị tối thiểu hoặc giá trị tối đa.

Có thể được thiết đặt để không cộng thông tin chuyển động của các khối được chứa trong vùng định trước vào danh sách thông tin chuyển động liên vùng. Ví dụ, ứng cử hợp nhất liên vùng được suy ra dựa trên thông tin chuyển động của khối được chứa trong vùng xử lý hợp nhất có thể không được cộng vào danh sách thông tin chuyển động liên vùng. Vì thứ bậc lập mã/giải mã không được định nghĩa cho các khối được chứa trong vùng xử lý hợp nhất, nên không thích hợp để sử dụng thông tin chuyển động của một khối bất kỳ trong số các khối để dự báo liên cấu trúc của khối khác. Do đó, các ứng cử hợp nhất liên vùng được suy ra dựa trên các khối được chứa trong vùng xử lý hợp nhất có thể không được cộng vào danh sách thông tin chuyển động liên vùng.

Khi dự báo bù chuyển động được thực hiện theo đơn vị khối con, thì ứng cử hợp nhất liên vùng có thể được suy ra dựa trên thông tin chuyển động của khối con đại diện trong số nhiều khối con được chứa trong khối hiện thời. Ví dụ, khi ứng cử hợp nhất khối con được sử dụng cho khối hiện thời, thì ứng cử hợp nhất liên vùng có thể được suy ra dựa trên thông tin chuyển động của khối con đại diện trong số các khối con.

Các vector chuyển động của các khối con có thể được suy ra theo thứ bậc sau đây. Trước hết, một ứng cử bất kỳ trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất của khối hiện thời được chọn lựa, và vector dịch khởi tạo ($shVector$) có thể được suy ra dựa trên vector chuyển động của ứng cử hợp nhất được chọn lựa. Sau đó, khối

con được dịch, mà trong đó vị trí của mẫu tham chiếu là $(xColSb, yColSb)$, có thể được suy ra là vector dịch khởi tạo được cộng vào ở vị trí (xSb, ySb) của mẫu tham chiếu (ví dụ, mẫu bên trái-trên cùng hoặc mẫu ở trung tâm) của mỗi khối con trong khối mã hóa. Phương trình 4 thể hiện phương trình để suy ra khối con được dịch.

【Phương trình 4】

$$(xColSb, yColSb) = (xSb + shVector[0] \gg 4, ySb + shVector[1] \gg 4)$$

Sau đó, vector chuyển động của khối được sắp xếp cùng một chỗ tương ứng với vị trí trung tâm của khối con bao gồm $(xColSb, yColSb)$ có thể được thiết đặt là vector chuyển động của khối con bao gồm (xSb, ySb) .

Khối con đại diện có thể có nghĩa là khối con bao gồm mẫu bên trái-trên cùng hoặc mẫu ở trung tâm của khối hiện thời.

Fig.23 là hình vẽ thể hiện vị trí của khối con đại diện.

Fig.23(a) thể hiện một ví dụ mà trong đó khối con được đặt ở vị trí bên trái-trên cùng của khối hiện thời được thiết đặt như khối con đại diện, và Fig.23(b) thể hiện một ví dụ mà trong đó khối con được đặt ở vị trí trung tâm của khối hiện thời được thiết đặt như khối con đại diện. Khi dự báo bù chuyển động được thực hiện theo đơn vị khối con, thì ứng cử hợp nhất liên vùng của khối hiện thời có thể được suy ra dựa trên vector chuyển động của khối con bao gồm mẫu bên trái-trên cùng của khối hiện thời hoặc khối con bao gồm mẫu ở trung tâm của khối hiện thời.

Có thể được xác định việc có sử dụng hay không khối hiện thời là ứng cử hợp nhất liên vùng, dựa trên chế độ dự báo liên cấu trúc của khối hiện thời. Ví dụ, khối được lập mã/giải mã dựa trên mô hình chuyển động afin có thể được thiết đặt là không khả dụng như ứng cử hợp nhất liên vùng. Do đó, mặc dù khối hiện thời được lập mã/giải mã bởi dự báo liên cấu trúc, khi chế độ dự báo liên cấu trúc của khối hiện thời là chế độ dự báo afin, thì danh sách thông tin chuyển động liên vùng có thể không được cập nhật dựa trên khối hiện thời.

Ngoài ra, ứng cử hợp nhất liên vùng có thể được suy ra dựa trên ít nhất một vector khối con trong số các khối con được chứa trong khối được lập mã/giải mã dựa trên mô hình chuyển động afin. Ví dụ, ứng cử hợp nhất liên vùng có thể được suy ra sử dụng khối con

được đặt ở vị trí bên trái-trên cùng, khối con được đặt ở vị trí trung tâm, hoặc khối con được đặt ở vị trí cạnh bên phải-trên cùng của khối hiện thời. Ngoài ra, giá trị trung bình của các vector khối con của nhiều khối con có thể được thiết đặt là vector chuyển động của ứng cử hợp nhất liên vùng.

Ngoài ra, ứng cử hợp nhất liên vùng có thể được suy ra dựa trên giá trị trung bình của các vector hạt giống afin của khối được lập mã/giải mã dựa trên mô hình chuyển động afin. Ví dụ, trung bình của ít nhất một trong số vector hạt giống afin thứ nhất, vector hạt giống afin thứ hai, và vector hạt giống afin thứ ba của khối hiện thời có thể được thiết đặt là vector chuyển động của ứng cử hợp nhất liên vùng.

Ngoài ra, danh sách thông tin chuyển động liên vùng có thể được tạo cấu hình cho mỗi chế độ dự báo liên cấu trúc. Ví dụ, ít nhất một trong số danh sách thông tin chuyển động liên vùng dành cho khối được lập mã/giải mã bằng sao chép khối nội cấu trúc, danh sách thông tin chuyển động liên vùng dành cho khối được lập mã/giải mã dựa trên mô hình chuyển động tịnh tiến, và danh sách thông tin chuyển động liên vùng dành cho khối được lập mã/giải mã dựa trên mô hình chuyển động afin có thể được định nghĩa. Theo chế độ dự báo liên cấu trúc của khối hiện thời, danh sách bất kỳ trong số nhiều danh sách thông tin chuyển động ứng cử liên vùng có thể được chọn lựa.

Fig.24 là hình vẽ thể hiện một ví dụ mà trong đó danh sách thông tin chuyển động liên vùng được tạo ra cho mỗi chế độ dự báo liên cấu trúc.

Khi khối được lập mã/giải mã dựa trên mô hình chuyển động không phải afin, thì ứng cử hợp nhất liên vùng mvCand được suy ra dựa trên khối có thể được cộng vào danh sách thông tin chuyển động không phải afin liên vùng HmvpCandList. Mặt khác, khi khối được lập mã/giải mã dựa trên mô hình chuyển động afin, thì ứng cử hợp nhất liên vùng mvAfCand được suy ra dựa trên khối có thể được cộng vào danh sách thông tin chuyển động afin liên vùng HmvpAfCandList.

Các vector hạt giống afin của khối được lập mã/giải mã dựa trên mô hình chuyển động afin có thể được lưu trong ứng cử hợp nhất liên vùng được suy ra từ khối này. Do đó, ứng cử hợp nhất liên vùng có thể được sử dụng làm ứng cử hợp nhất để suy ra vector hạt giống afin của khối hiện thời.

Ngoài danh sách thông tin chuyển động liên vùng được mô tả ở trên, danh sách thông tin liên vùng bổ sung có thể được định nghĩa. Ngoài danh sách thông tin chuyển động liên vùng được mô tả ở trên (sau đây, được gọi là danh sách thông tin liên vùng thứ nhất), danh sách thông tin chuyển động dài hạn (sau đây, được gọi là danh sách thông tin liên vùng thứ hai) có thể được định nghĩa. Ở đây, danh sách thông tin chuyển động dài hạn bao gồm các ứng cử hợp nhất dài hạn.

Khi cả hai danh sách thông tin liên vùng thứ nhất và danh sách thông tin liên vùng thứ hai trống, thì trước hết, các ứng cử hợp nhất liên vùng có thể được cộng vào danh sách thông tin liên vùng thứ hai. Chỉ sau khi số lượng các ứng cử hợp nhất liên vùng khả dụng đạt đến số lượng tối đa trong danh sách thông tin liên vùng thứ hai, thì các ứng cử hợp nhất liên vùng có thể được cộng vào danh sách thông tin liên vùng thứ nhất.

Ngoài ra, một ứng cử hợp nhất liên vùng có thể được cộng vào cả hai danh sách thông tin liên vùng thứ hai và danh sách thông tin liên vùng thứ nhất.

Vào lúc này, danh sách thông tin liên vùng thứ hai, cấu hình mà đã được hoàn thành, có thể không được cập nhật thêm nữa. Ngoài ra, khi vùng được giải mã lớn hơn hoặc bằng tỷ số định trước của phần chia, thì danh sách thông tin liên vùng thứ hai có thể được cập nhật. Ngoài ra, danh sách thông tin liên vùng thứ hai có thể được cập nhật đối với mỗi N đường đơn vị cây mã hóa.

Mặt khác, danh sách thông tin liên vùng thứ nhất có thể được cập nhật bất cứ khi nào khối được lập mã/giải mã bởi dự báo liên cấu trúc được tạo ra. Tuy nhiên, có thể được thiết đặt là không sử dụng ứng cử hợp nhất liên vùng được cộng vào danh sách thông tin liên vùng thứ hai, để cập nhật danh sách thông tin liên vùng thứ nhất.

Thông tin để chọn lựa một danh sách bất kỳ trong số danh sách thông tin liên vùng thứ nhất và danh sách thông tin liên vùng thứ hai có thể được báo hiệu thông qua luồng bit. Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với giá trị ngưỡng, thì các ứng cử hợp nhất được chứa trong danh sách thông tin chuyển động liên vùng được chỉ báo bởi thông tin có thể được cộng vào danh sách ứng cử hợp nhất.

Ngoài ra, danh sách thông tin chuyển động liên vùng có thể được chọn lựa dựa trên kích thước và hình dạng của khối hiện thời, chế độ dự báo liên cấu trúc, việc dự báo hai

chiều có được cho phép hay không, việc sàng lọc vector chuyển động có được cho phép hay không, hoặc việc phân chia hình tam giác có được cho phép hay không.

Ngoài ra, mặc dù ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin liên vùng thứ nhất được cộng vào, khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với số lượng tối đa của các hợp nhất, thì các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin liên vùng thứ hai có thể được cộng vào danh sách ứng cử hợp nhất.

Fig.25 là hình vẽ thể hiện một ví dụ mà trong đó ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động dài hạn được cộng vào danh sách ứng cử hợp nhất.

Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với số lượng tối đa, thì các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin liên vùng thứ nhất HmvpCandList có thể được cộng vào danh sách ứng cử hợp nhất. Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với số lượng tối đa mặc dù các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin liên vùng thứ nhất được cộng vào danh sách ứng cử hợp nhất, thì các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động dài hạn HmvpLTCandList có thể được cộng vào danh sách ứng cử hợp nhất.

Bảng 2 thể hiện quy trình để cộng các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động dài hạn vào danh sách ứng cử hợp nhất.

【Bảng 2】

Đối với mỗi ứng cử trong HMVPCandList với chỉ số HMVPLTIdx = 1..

numHMVPLTCand, thì các bước có thứ bậc sau đây được lặp lại cho đến khi combStop có giá trị true

sameMotion được thiết đặt có giá trị là FALSE

Nếu hmvStop có giá trị là FALSE và numCurrMergecand nhỏ hơn so với

(MaxNumMergeCand-1), thì hmvLT được thiết đặt có giá trị là TRUE

Nếu HMVPLTCandList[NumLTHmvp-HMVPLTIdx] có cùng các vector chuyển động và cùng các chỉ số tham chiếu với mergeCandList[i] bất kỳ với I là 0..

numOrigMergeCand-1 và HasBeenPruned[i] có giá trị bằng FALSE, thì sameMotion được thiết đặt có giá trị là true

Nếu sameMotion có giá trị bằng false, thì mergeCandList[numCurrMergeCand++] được thiết đặt là HMVPLTCandList[NumLTHmvp-HMVPLTIdx]

- Nếu numCurrMergeCand bằng (MaxNumMergeCand-1), thì hmvLTStop được thiết đặt có giá trị bằng TRUE

Ứng cử hợp nhất liên vùng có thể được thiết đặt để bao gồm thông tin bổ sung, ngoài thông tin chuyển động. Ví dụ, đối với ứng cử hợp nhất liên vùng, kích thước, hình dạng, hoặc thông tin phân chia của khối có thể được lưu bổ sung. Khi danh sách ứng cử hợp nhất của khối hiện thời được cấu thành, thì chỉ các ứng cử hợp nhất liên vùng có kích thước, hình dạng, hoặc thông tin phân chia giống hoặc tương tự với khối hiện thời được sử dụng trong số các ứng cử hợp nhất liên vùng, hoặc các ứng cử hợp nhất liên vùng có kích thước, hình dạng, hoặc thông tin phân chia giống hoặc tương tự với khối hiện thời có thể được cộng vào danh sách ứng cử hợp nhất ở vị trí thứ nhất.

Ngoài ra, danh sách thông tin chuyển động liên vùng có thể được tạo ra cho mỗi thông tin trong số kích thước, hình dạng, hoặc thông tin phân chia của khối. Trong số nhiều danh sách thông tin chuyển động liên vùng, thì danh sách ứng cử hợp nhất của khối hiện thời có thể được tạo ra bằng cách sử dụng danh sách thông tin chuyển động liên vùng tương ứng với hình dạng, kích thước, hoặc thông tin phân chia của khối hiện thời.

Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất của khối hiện thời nhỏ hơn so với giá trị ngưỡng, thì các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động liên vùng có thể được cộng vào danh sách ứng cử hợp nhất. Quy trình bổ sung được thực hiện theo thứ bậc tăng dần hoặc giảm dần dựa trên chỉ số. Ví dụ, ứng cử hợp nhất liên vùng có chỉ số lớn nhất có thể được cộng đầu tiên vào danh sách ứng cử hợp nhất.

Khi muốn cộng ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động liên vùng vào danh sách ứng cử hợp nhất, thì kiểm dư thừa có thể được thực hiện giữa ứng cử hợp nhất liên vùng và các ứng cử hợp nhất được lưu trước đó trong danh sách ứng cử hợp nhất.

Ví dụ, bảng 3 thể hiện quy trình trong đó ứng cử hợp nhất liên vùng được cộng vào danh sách ứng cử hợp nhất.

【Bảng 3】

Đối với mỗi ứng cử trong HMVPCandList với chỉ số HMVPIdx = 1..
numCheckedHMVPCand, thì các bước có thứ bậc sau đây được lặp lại cho đến khi
combStop có giá trị true
sameMotion được thiết đặt có giá trị là false
Nếu HMVPCandList[NumHmvp-HMVPIdx] có cùng các vector chuyển động và cùng
các chỉ số tham chiếu với mergeCandList[i] bất kỳ với I là 0.. numOrigMergeCand-1
và HasBeenPruned[i] có giá trị bằng false, thì sameMotion được thiết đặt có giá trị là
true
Nếu sameMotion có giá trị bằng false, thì mergeCandList[numCurrMergeCand++]
được thiết đặt là HMVPCandList[NumHmvp-HMVPIdx]
- Nếu numCurrMergeCand bằng (MaxNumMergeCand-1), thì hmvpStop được thiết đặt
có giá trị bằng TRUE

Kiểm dư thừa có thể được thực hiện chỉ trên một số ứng cử trong số các ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động liên vùng. Ví dụ, kiểm dư thừa có thể được thực hiện chỉ trên các ứng cử hợp nhất liên vùng có chỉ số lớn hơn so với giá trị ngưỡng hoặc nhỏ hơn so với giá trị ngưỡng. Ngoài ra, kiểm dư thừa có thể được thực

hiện chỉ trên N ứng cử hợp nhất có chỉ số lớn nhất hoặc N ứng cử hợp nhất có chỉ số nhỏ nhất.

Ngoài ra, kiểm dư thừa có thể được thực hiện chỉ trên một số ứng cử trong số các ứng cử hợp nhất được lưu trước đó trong danh sách ứng cử hợp nhất. Ví dụ, kiểm dư thừa có thể được thực hiện chỉ trên ứng cử hợp nhất có chỉ số lớn hơn so với giá trị ngưỡng hoặc nhỏ hơn so với giá trị ngưỡng, hoặc trên ứng cử hợp nhất được suy ra từ khối ở vị trí cụ thể. Ở đây, vị trí cụ thể có thể bao gồm ít nhất một trong số khối lân cận bên trái, khối lân cận trên cùng, khối lân cận bên phải-trên cùng, và khối lân cận bên trái-dưới cùng của khối hiện thời.

Fig.26 là hình vẽ thể hiện một ví dụ mà trong đó kiểm dư thừa được thực hiện chỉ trên một số ứng cử hợp nhất.

Khi muốn cộng ứng cử hợp nhất liên vùng HmvpCand[j] vào danh sách ứng cử hợp nhất, thì kiểm dư thừa có thể được thực hiện trên ứng cử hợp nhất liên vùng với hai ứng cử hợp nhất mergeCandList[NumMerge-2] và mergeCandList[NumMerge-1] có các chỉ số lớn nhất. Ở đây, NumMerge có thể biểu diễn số lượng các ứng cử hợp nhất không gian và các ứng cử hợp nhất thời gian khả dụng.

Không giống như ví dụ được thể hiện trên hình vẽ này, khi muốn cộng ứng cử hợp nhất liên vùng HmvpCand[j] vào danh sách ứng cử hợp nhất, thì kiểm dư thừa có thể được thực hiện trên ứng cử hợp nhất liên vùng lên tới hai ứng cử hợp nhất có chỉ số nhỏ nhất. Ví dụ, có thể kiểm tra xem mergeCandList[0] và mergeCandList[1] có giống như HmvpCand[j] hay không. Ngoài ra, kiểm dư thừa có thể được thực hiện chỉ trên các ứng cử được suy ra ở vị trí cụ thể. Ví dụ, kiểm dư thừa có thể được thực hiện trên ít nhất một trong số ứng cử hợp nhất được suy ra từ khối lân cận được đặt ở vị trí trên cạnh bên trái của khối hiện thời và ứng cử hợp nhất được suy ra từ khối lân cận được đặt ở vị trí trên phần trên cùng khối hiện thời. Khi ứng cử hợp nhất được suy ra ở vị trí cụ thể không tồn tại trong danh sách ứng cử hợp nhất, thì ứng cử hợp nhất liên vùng có thể được cộng vào danh sách ứng cử hợp nhất mà không kiểm dư thừa.

Khi ứng cử hợp nhất giống như ứng cử hợp nhất liên vùng thứ nhất được tìm thấy và kiểm dư thừa được thực hiện trên ứng cử hợp nhất liên vùng thứ hai, thì kiểm dư thừa với ứng cử hợp nhất giống như ứng cử hợp nhất liên vùng thứ nhất có thể được bỏ qua.

Fig.27 là hình vẽ thể hiện một ví dụ mà trong đó kiểm dư thừa được bỏ qua đối với ứng cử hợp nhất cụ thể.

Khi muốn cộng ứng cử hợp nhất liên vùng $HmvpCand[i]$ có chỉ số i vào danh sách ứng cử hợp nhất, thì kiểm dư thừa được thực hiện giữa ứng cử hợp nhất liên vùng và các ứng cử hợp nhất được lưu trước đó trong danh sách ứng cử hợp nhất. Vào lúc này, khi ứng cử hợp nhất $mergeCandList[j]$ giống như ứng cử hợp nhất liên vùng $HmvpCand[i]$ được tìm thấy, thì kiểm dư thừa có thể được thực hiện giữa ứng cử hợp nhất liên vùng $HmvpCand[i-1]$ có chỉ số $i-1$ và các ứng cử hợp nhất mà không cộng ứng cử hợp nhất liên vùng $HmvpCand[i]$ vào danh sách ứng cử hợp nhất. Vào lúc này, kiểm dư thừa giữa ứng cử hợp nhất liên vùng $HmvpCand[i-1]$ và ứng cử hợp nhất $mergeCandList[j]$ có thể được bỏ qua.

Ví dụ, trong ví dụ được thể hiện trên Fig.27, được xác định rằng $HmvpCand[i]$ và $mergeCandList[2]$ giống nhau. Do đó, $HmvpCand[i]$ không được cộng vào danh sách ứng cử hợp nhất, và kiểm dư thừa có thể được thực hiện trên $HmvpCand[i-1]$. Vào lúc này, kiểm dư thừa giữa $HmvpCand[i-1]$ và $mergeCandList[2]$ có thể được bỏ qua.

Khi số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất của khối hiện thời nhỏ hơn so với giá trị ngưỡng, thì có thể chứa thêm ít nhất một trong số ứng cử hợp nhất theo cặp và ứng cử hợp nhất zero, ngoài ứng cử hợp nhất liên vùng. Ứng cử hợp nhất theo cặp có nghĩa là ứng cử hợp nhất có giá trị trung bình của các vector chuyển động của hai hoặc nhiều ứng cử hợp nhất là vector chuyển động, và ứng cử hợp nhất zero có nghĩa là ứng cử hợp nhất có vector chuyển động 0.

Ứng cử hợp nhất có thể được cộng vào danh sách ứng cử hợp nhất của khối hiện thời theo thứ bậc sau đây.

Ứng cử hợp nhất không gian - Ứng cử hợp nhất thời gian - ứng cử hợp nhất liên vùng
- (Ứng cử hợp nhất afin liên vùng) - Ứng cử hợp nhất theo cặp - Ứng cử hợp nhất zero

Ứng cử hợp nhất không gian có nghĩa là ứng cử hợp nhất được suy ra từ ít nhất một trong số khối lân cận và khối không lân cận, và ứng cử hợp nhất thời gian có nghĩa là ứng cử hợp nhất được suy ra từ hình tham chiếu trước đó. Ứng cử hợp nhất afin liên vùng biểu diễn ứng cử hợp nhất liên vùng được suy ra từ khối được lập mã/giải mã với mô hình chuyển động afin.

Danh sách thông tin chuyển động liên vùng cũng có thể được sử dụng trong chế độ dự báo vector chuyển động cải tiến. Ví dụ, khi số lượng các ứng cử dự báo vector chuyển động được chứa trong danh sách ứng cử dự báo vector chuyển động của khối hiện thời nhỏ hơn so với giá trị ngưỡng, thì ứng cử hợp nhất liên vùng được chứa trong danh sách thông tin chuyển động liên vùng có thể được thiết đặt là ứng cử dự báo vector chuyển động dành cho khối hiện thời. Cụ thể là, vector chuyển động của ứng cử hợp nhất liên vùng có thể được thiết đặt là ứng cử dự báo vector chuyển động.

Khi một ứng cử bất kỳ trong số các ứng cử dự báo vector chuyển động được chứa trong danh sách ứng cử dự báo vector chuyển động của khối hiện thời được chọn lựa, thì ứng cử được chọn lựa có thể được thiết đặt là bộ dự báo vector chuyển động của khối hiện thời. Sau đây, sau khi hệ số còn dư vector chuyển động của khối hiện thời được giải mã, thì vector chuyển động của khối hiện thời có thể nhận được bằng cách cộng bộ dự báo vector chuyển động và hệ số còn dư vector chuyển động.

Danh sách ứng cử dự báo vector chuyển động của khối hiện thời có thể được tạo cấu hình theo thứ bậc sau đây.

Ứng cử dự báo vector chuyển động không gian - Ứng cử dự báo vector chuyển động thời gian - Ứng cử hợp nhất liên vùng - (Ứng cử hợp nhất afin liên vùng) - -Ứng cử dự báo vector chuyển động zero

Ứng cử dự báo vector chuyển động không gian có nghĩa là ứng cử dự báo vector chuyển động được suy ra từ ít nhất một trong số khối lân cận và khối không lân cận, và ứng cử dự báo vector chuyển động thời gian có nghĩa là ứng cử dự báo vector chuyển động được suy ra từ hình tham chiếu trước đó. Ứng cử hợp nhất afin liên vùng biểu diễn ứng cử dự báo vector chuyển động liên vùng được suy ra từ khối được lập mã/giải mã với mô hình chuyển động

afin. Ứng cử dự báo vector chuyển động zero biểu diễn ứng cử có giá trị vector chuyển động 0.

Khi ứng cử hợp nhất của khối hiện thời được chọn lựa, thì vector chuyển động của ứng cử hợp nhất được chọn lựa được thiết đặt như vector chuyển động khởi tạo, và dự báo bù chuyển động có thể được thực hiện dành cho khối hiện thời sử dụng vector chuyển động được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động khởi tạo. Việc suy ra vector chuyển động mới bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất có thể được định nghĩa như phương pháp mã hóa độ chênh lệch chuyển động hợp nhất.

Thông tin chỉ báo việc có sử dụng hay không phương pháp lập mã độ lệch hợp nhất có thể được báo hiệu thông qua luồng bit. Thông tin này có thể là cờ `merge_offset_vector_flag` có một bit. Ví dụ, khi giá trị của `merge_offset_vector_flag` là 1, thì nó chỉ báo rằng phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời. Khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất. Khi giá trị của `merge_offset_vector_flag` là 0, thì nó chỉ báo rằng phương pháp mã hóa độ chênh lệch chuyển động hợp nhất không được áp dụng cho khối hiện thời. Khi phương pháp lập mã độ lệch hợp nhất không được áp dụng, thì vector chuyển động của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động của khối hiện thời.

Cờ có thể được báo hiệu chỉ khi giá trị của `skip_flag` chỉ báo việc chế độ bỏ qua được áp dụng là true hay không hoặc khi giá trị của `merge_flag` chỉ báo việc chế độ hợp nhất được áp dụng là true hay không. Ví dụ, khi giá trị của `skip_flag` chỉ báo việc chế độ bỏ qua được áp dụng cho khối hiện thời là 1 hay không hoặc khi giá trị của `merge_flag` chỉ báo việc chế độ hợp nhất được áp dụng cho khối hiện thời là 1 hay không, thì `merge_offset_vector_flag` có thể được lập mã và được báo hiệu.

Khi được xác định rằng phương pháp lập mã độ lệch hợp nhất được áp dụng cho khối hiện thời, thì ít nhất một trong số thông tin định rõ một thông tin bất kỳ trong số các ứng cử

hợp nhất được chứa trong danh sách ứng cử hợp nhất, thông tin chỉ báo độ lớn của vector độ lệch, và thông tin chỉ báo chiều của vector độ lệch có thể được báo hiệu bổ sung.

Thông tin để xác định số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm có thể được báo hiệu thông qua luồng bit. Ví dụ, số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm có thể được thiết đặt là số nguyên 6 hoặc nhỏ hơn.

Khi được xác định rằng phương pháp lập mã độ lệch hợp nhất được áp dụng cho khối hiện thời, thì chỉ số lượng tối đa của các ứng cử hợp nhất được thiết đặt trước có thể được thiết đặt là vector chuyển động khởi tạo của khối hiện thời. Tức là, số lượng các ứng cử hợp nhất mà có thể được sử dụng bởi khối hiện thời có thể được xác định thích nghi theo việc phương pháp lập mã độ lệch hợp nhất có được áp dụng hay không. Ví dụ, khi giá trị của `merge_offset_vector_flag` được thiết đặt là 0, thì số lượng tối đa của các ứng cử hợp nhất mà có thể được sử dụng bởi khối hiện thời có thể được thiết đặt là M, trong khi đó khi giá trị của `merge_offset_vector_flag` được thiết đặt là 1, thì số lượng tối đa của các ứng cử hợp nhất mà có thể được sử dụng bởi khối hiện thời có thể được thiết đặt là N. Ở đây, M thể hiện số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm, và N thể hiện số nguyên bằng hoặc nhỏ hơn M.

Ví dụ, khi M là 6 và N là 2, thì hai ứng cử hợp nhất có chỉ số nhỏ nhất trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất có thể được thiết đặt là khả dụng đối với khối hiện thời. Do đó, vector chuyển động của ứng cử hợp nhất có giá trị chỉ số 0 hoặc vector chuyển động của ứng cử hợp nhất có giá trị chỉ số 1 có thể được thiết đặt là vector chuyển động khởi tạo của khối hiện thời. Khi M và N giống nhau (ví dụ, khi M và N là 2), thì tất cả các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất có thể được thiết đặt là khả dụng đối với khối hiện thời.

Ngoài ra, việc khối lân cận có thể được sử dụng như ứng cử hợp nhất có thể được xác định dựa trên việc phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không. Ví dụ, khi giá trị của `merge_offset_vector_flag` là 1, thì ít nhất một trong số khối lân cận gần kề với góc bên phải-trên cùng của khối hiện thời, khối lân cận gần kề với góc bên trái-trên cùng, và khối lân cận gần kề với góc bên trái-dưới cùng

có thể được thiết đặt là không khả dụng như ứng cử hợp nhất. Do đó, khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của ít nhất một trong số khối lân cận gần kề với góc bên phải-trên cùng của khối hiện thời, khối lân cận gần kề với góc bên trái-trên cùng, và khối lân cận gần kề với góc bên trái-dưới cùng có thể không được thiết đặt là vector chuyển động khởi tạo. Ngoài ra, khi giá trị của `merge_offset_vector_flag` là 1, thì khối lân cận về mặt thời gian của khối hiện thời có thể được thiết đặt là không khả dụng như ứng cử hợp nhất.

Khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì có thể được thiết đặt không sử dụng ít nhất một trong số ứng cử hợp nhất theo cặp và ứng cử hợp nhất zero. Do đó, khi giá trị của `merge_offset_vector_flag` là 1, thì ít nhất một trong số ứng cử hợp nhất theo cặp và ứng cử hợp nhất zero có thể không được cộng vào danh sách ứng cử hợp nhất mặc dù số lượng các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất nhỏ hơn so với số lượng tối đa.

Vector chuyển động của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động khởi tạo của khối hiện thời. Vào lúc này, khi số lượng các ứng cử hợp nhất mà có thể được sử dụng bởi khối hiện thời là số nhiều, thì thông tin định rõ một ứng cử trong số nhiều ứng cử hợp nhất có thể được báo hiệu thông qua luồng bit. Ví dụ, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm lớn hơn so với 1, thì thông tin `merge_idx` chỉ báo một ứng cử bất kỳ trong số nhiều ứng cử hợp nhất có thể được báo hiệu thông qua luồng bit. Tức là, theo phương pháp lập mã độ lệch hợp nhất, thì ứng cử hợp nhất có thể được định rõ bởi thông tin `merge_idx` để định rõ một ứng cử bất kỳ trong số nhiều ứng cử hợp nhất. Vector chuyển động khởi tạo của khối hiện thời có thể được thiết đặt là vector chuyển động của ứng cử hợp nhất được chỉ báo bởi `merge_idx`.

Mặt khác, khi số lượng các ứng cử hợp nhất mà có thể được sử dụng bởi khối hiện thời là 1, thì việc báo hiệu về thông tin để định rõ ứng cử hợp nhất có thể được bỏ qua. Ví dụ, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm không lớn hơn 1, thì việc báo hiệu về thông tin `merge_idx` để định rõ ứng cử hợp nhất có thể được bỏ qua. Tức là, theo phương pháp lập mã độ lệch hợp nhất, khi một ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất, thì việc lập mã của thông tin `merge_idx`

để định rõ ứng cử hợp nhất có thể được bỏ qua, và vector chuyển động khởi tạo có thể được xác định dựa trên ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất. Vector chuyển động của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động khởi tạo của khối hiện thời.

Ví dụ khác, sau khi ứng cử hợp nhất của khối hiện thời được xác định, thì việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được xác định. Ví dụ, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm lớn hơn so với 1, thì thông tin `merge_idx` để định rõ một ứng cử bất kỳ trong số trong số các ứng cử hợp nhất có thể được báo hiệu. Sau khi ứng cử hợp nhất được chọn lựa dựa trên `merge_idx`, `merge_offset_vector_flag` chỉ báo việc phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không có thể được giải mã. Bảng 4 thể hiện bảng cú pháp theo phương án được mô tả ở trên.

【Bảng 4】

<code>coding_unit (x0, y0, cbWidth, cbHeight, treeType) {</code>	Bộ mô tả
<code>if (slice_type != I) {</code>	
<code>cu_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if (cu_skip_flag[x0][y0] == 0)</code>	
<code>pred_mode_flag</code>	<code>ae(v)</code>
<code>}</code>	
<code>if (CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code>if (treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>intra_luma_mpm_flag[x0][y0]</code>	
<code>if (intra_luma_mpm_flag[x0][y0])</code>	
<code>intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
nếu không thì	
<code>intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	

if (treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
} else { /* MODE_INTER */	
if (cu_skip_flag[x0][y0]) {	
if (merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_idx [x0][y0]	ae(v)
direction_idx [x0][y0]	ae(v)
}	
}	
} else {	
merge_flag [x0][y0]	ae(v)
if (merge_flag[x0][y0]) {	
if (merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx [x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_idx [x0][y0]	ae(v)
direction_idx [x0][y0]	ae(v)
}	
}	
} else {	
if (slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if (sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)

if (sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
}	

Ví dụ khác, sau khi ứng cử hợp nhất của khối hiện thời được xác định, việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được xác định chỉ khi chỉ số của ứng cử hợp nhất được xác định nhỏ hơn so với số lượng tối đa của các ứng cử hợp nhất mà có thể được sử dụng khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng. Ví dụ, chỉ khi giá trị của thông tin chỉ số merge_idx nhỏ hơn so với N, merge_offset_vector_flag chỉ báo việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được lập mã và được lập mã. Khi giá trị của thông tin chỉ số merge_idx bằng hoặc lớn hơn N, thì hoạt động lập mã merge_offset_vector_flag có thể được bỏ qua. Khi hoạt động lập mã merge_offset_vector_flag được bỏ qua, thì có thể được xác định rằng phương pháp mã hóa độ chênh lệch chuyển động hợp nhất không được áp dụng cho khối hiện thời.

Ngoài ra, sau khi ứng cử hợp nhất của khối hiện thời được xác định, việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được xác định có tính đến việc ứng cử hợp nhất được xác định có thông tin chuyển động hai chiều hoặc thông tin chuyển động một chiều hay không. Ví dụ, merge_offset_vector_flag chỉ báo việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được lập mã và chỉ được báo hiệu khi khi giá trị của thông tin chỉ số merge_idx nhỏ hơn so với N và ứng cử hợp nhất được chọn lựa bởi thông tin chỉ số có thông tin chuyển động hai chiều. Ngoài ra, merge_offset_vector_flag chỉ báo việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được lập mã và chỉ được báo hiệu khi giá trị của thông tin chỉ số merge_idx nhỏ hơn so với N và ứng cử hợp nhất được chọn lựa bởi thông tin chỉ số có thông tin chuyển động một chiều.

Ngoài ra, việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có thể được xác định dựa trên ít nhất một trong số kích thước của khối hiện thời,

hình dạng của khối hiện thời, và việc khối hiện thời có tiếp xúc với biên của đơn vị cây mã hóa hay không. Khi ít nhất một trong số kích thước của khối hiện thời, kích thước của khối hiện thời, và việc khối hiện thời có tiếp xúc với biên của đơn vị cây mã hóa hay không không thỏa mãn điều kiện đặt trước, thì hoạt động lập mã merge_offset_vector_flag chỉ báo việc có áp dụng hay không phương pháp mã hóa độ chênh lệch chuyển động hợp nhất cho khối hiện thời có thể được bỏ qua.

Khi ứng cử hợp nhất được chọn lựa, thì vector chuyển động của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động khởi tạo của khối hiện thời. Lúc đó, vector độ lệch có thể được xác định bằng cách giải mã thông tin chỉ báo độ lớn của vector độ lệch và thông tin chỉ báo chiều của vector độ lệch. Vector độ lệch có thể có thành phần chiều ngang hoặc thành phần chiều dọc.

Thông tin chỉ báo độ lớn của vector độ lệch có thể là thông tin chỉ số chỉ báo một ứng cử bất kỳ trong số các ứng cử độ lớn chuyển động. Ví dụ, thông tin chỉ số distance_idx chỉ báo một ứng cử bất kỳ trong số các ứng cử độ lớn chuyển động có thể được báo hiệu thông qua luồng bit. Bảng 5 thể hiện dạng nhị phân của thông tin chỉ số distance_idx và các giá trị của biến variable DistFromMergeMV để xác định độ lớn của vector độ lệch theo distance_idx.

【Bảng 5】

distance_idx[x][y]	Dạng nhị phân	DistFromMergeMV[x0][y0]
0	0	1
1	10	2
2	110	4
3	1110	8
4	11110	16
5	111110	32
6	1111110	64
7	1111111	128

Độ lớn của vector độ lệch có thể được suy ra bằng cách chia biến *DistFromMergeMV* cho giá trị đặt trước. Phương trình 5 thể hiện một ví dụ để xác định độ lớn của vector độ lệch.

【Phương trình 5】

$$abs(offsetMV) = DistFromMergeMV \ll 2$$

Theo phương trình 5, giá trị nhận được bằng cách chia biến *DistFromMergeMV* cho 4 hoặc giá trị nhận được bằng cách dịch biến *DistFromMergeMV* sang bên trái bởi 2 có thể được thiết đặt là độ lớn của vector độ lệch.

Số lượng lớn hơn của các ứng cử độ lớn chuyển động hoặc số lượng nhỏ hơn của các ứng cử độ lớn chuyển động so với ví dụ được thể hiện trong bảng 5 có thể được sử dụng, hoặc phạm vi của các ứng cử kích thước độ lệch vector chuyển động có thể được thiết đặt khác với ví dụ được thể hiện trong bảng 5. Ví dụ, độ lớn của thành phần chiều ngang hoặc thành phần chiều dọc của vector độ lệch có thể được thiết đặt không lớn hơn 2 khoảng cách mẫu. Bảng 6 thể hiện dạng nhị phân của thông tin chỉ số *distance_idx* và các giá trị của biến *DistFromMergeMV* để xác định độ lớn của vector độ lệch theo *distance_idx*.

【Bảng 6】

<i>distance_idx</i> [x][y]	Dạng nhị phân	<i>DistFromMergeMV</i> [x0][y0]
0	0	1
1	10	2
2	110	4
3	111	8

Ngoài ra, phạm vi của các ứng cử kích thước độ lệch vector chuyển động có thể được thiết đặt khác nhau dựa trên độ chính xác vector chuyển động. Ví dụ, khi độ chính xác vector chuyển động dành cho khối hiện thời là đơn vị điểm ảnh dạng phân số, thì các giá trị của biến *DistFromMergeMV* tương ứng với các giá trị của thông tin chỉ số *distance_idx* có thể được thiết đặt là 1, 2, 4, 8, 16 hoặc giá trị tương tự. Ở đây, đơn vị điểm ảnh dạng phân số bao gồm ít nhất một đơn vị trong số 1/16 đơn vị điểm ảnh, một phần tám đơn vị điểm ảnh, một phần tư đơn vị điểm ảnh, và một phần hai đơn vị điểm ảnh. Mặt khác, khi độ chính xác

vector chuyển động dành cho khối hiện thời là đơn vị điểm ảnh nguyên, thì các giá trị của biến `DistFromMergeMV` tương ứng với các giá trị của thông tin chỉ số `distance_idx` có thể được thiết đặt là 4, 8, 16, 32, 64, và giá trị tương tự. Tức là, bảng được tham chiếu để xác định `DistFromMergeMV` có thể được thiết đặt khác nhau theo độ chính xác vector chuyển động dành cho khối hiện thời.

Ví dụ, khi độ chính xác vector chuyển động của khối hiện thời hoặc ứng cử hợp nhất là một phân tử đơn vị điểm ảnh, thì biến `DistFromMergeMV` được chỉ báo bởi `distance_idx` có thể được suy ra sử dụng bảng 5. Mặt khác, khi độ chính xác vector chuyển động của khối hiện thời hoặc ứng cử hợp nhất là đơn vị điểm ảnh nguyên, thì giá trị nhận được bằng cách lấy N lần (ví dụ, 4 lần) của giá trị của biến `DistFromMergeMV` được chỉ báo bởi `distance_idx` trong bảng 5 có thể được suy ra là giá trị của biến `DistFromMergeMV`.

Thông tin để xác định độ chính xác vector chuyển động có thể được báo hiệu thông qua luồng bit. Ví dụ, thông tin này có thể được báo hiệu ở mức trình tự, hình, phần chia, hoặc khối. Do đó, phạm vi của các ứng cử độ lớn chuyển động có thể được thiết đặt khác nhau theo thông tin có liên quan đến độ chính xác vector chuyển động được báo hiệu thông qua luồng bit. Ngoài ra, độ chính xác vector chuyển động có thể được xác định dựa trên ứng cử hợp nhất của khối hiện thời. Ví dụ, độ chính xác vector chuyển động của khối hiện thời có thể được thiết đặt giống như độ chính xác vector chuyển động của ứng cử hợp nhất.

Ngoài ra, thông tin để xác định phạm vi tìm kiếm của vector độ lệch có thể được báo hiệu thông qua luồng bit. Ít nhất một thông tin trong số số lượng các ứng cử độ lớn chuyển động, giá trị tối thiểu trong số các ứng cử độ lớn chuyển động, và giá trị tối đa trong số các ứng cử độ lớn chuyển động có thể được xác định dựa trên phạm vi tìm kiếm. Ví dụ, `flag_merge_offset_vector` để xác định phạm vi tìm kiếm của vector độ lệch có thể được báo hiệu thông qua luồng bit. Thông tin này có thể được báo hiệu thông qua phần đầu trình tự, phần đầu hình, hoặc phần đầu phần chia.

Ví dụ, khi giá trị của `merge_offset_extend_range_flag` là 0, thì độ lớn của vector độ lệch có thể được thiết đặt không vượt quá 2. Do đó, giá trị tối đa của `DistFromMergeMV` có thể được thiết đặt là 8. Mặt khác, khi giá trị của `merge_offset_extend_range_flag` là 1,

thì độ lớn của vector độ lệch có thể được thiết đặt không vượt quá 32 khoảng cách mẫu. Do đó, giá trị tối đa của `DistFromMergeMV` có thể được thiết đặt là 128.

Độ lớn của vector độ lệch có thể được xác định sử dụng cờ chỉ báo việc độ lớn của vector độ lệch có lớn hơn so với giá trị ngưỡng hay không. Ví dụ, cờ `distance_flag` chỉ báo việc độ lớn của vector độ lệch có lớn hơn so với giá trị ngưỡng hay không có thể được báo hiệu thông qua luồng bit. Giá trị ngưỡng có thể là 1, 2, 4, 8 hoặc 16. Ví dụ, khi `distance_flag` là 1, thì nó chỉ báo rằng độ lớn của vector độ lệch lớn hơn 4. Mặt khác, khi `distance_flag` là 0, thì nó chỉ báo rằng độ lớn của vector độ lệch là 4 hoặc thấp hơn.

Khi độ lớn của vector độ lệch lớn hơn so với giá trị ngưỡng, giá trị chênh lệch giữa độ lớn của vector độ lệch và giá trị ngưỡng có thể được suy ra sử dụng thông tin chỉ số `distance_idx`. Ngoài ra, khi độ lớn của vector độ lệch thấp hơn hoặc bằng giá trị ngưỡng, thì độ lớn của vector độ lệch có thể được xác định sử dụng thông tin chỉ số `distance_idx`. Bảng 7 là bảng cú pháp thể hiện quy trình để lập mã `distance_flag` và `distance_idx`.

【Bảng 7】

<code>coding_unit (x0, y0, cbWidth, cbHeight, treeType) {</code>	Bộ mô tả
<code>if (slice_type != I) {</code>	
<code>cu_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if (cu_skip_flag[x0][y0] == 0)</code>	
<code>pred_mode_flag</code>	<code>ae(v)</code>
<code>}</code>	
<code>if (CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code>if (treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>intra_luma_mpm_flag[x0][y0]</code>	
<code>if (intra_luma_mpm_flag[x0][y0])</code>	
<code>intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
nếu không thì	
<code>intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	

if (treeType == SINGLE_TREE treeType ==	
DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else { /* MODE_INTER */	
if (cu_skip_flag[x0][y0]) {	
if (merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx[x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_flag[x0][y0]	ae(v)
distance_idx[x0][y0]	
direction_idx[x0][y0]	ae(v)
}	
}	
} else {	
merge_flag[x0][y0]	ae(v)
if (merge_flag[x0][y0]) {	
if (merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx[x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if (merge_idx < 2 && merge_offset_vector_flag) {	
distance_flag[x0][y0]	ae(v)
distance_idx[x0][y0]	ae(v)
direction_idx[x0][y0]	ae(v)
}	
}	
} else {	
if (slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)

if (sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if (sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
}	

Phương trình 6 thể hiện một ví dụ để suy ra biến *DistFromMergeMV* để xác định độ lớn của vector độ lệch sử dụng *distance_flag* và *distance_idx*.

【Phương trình 6】

$$DistFromMergeMV = N * distance_flag + (1 \ll distance_idx)$$

Trong phương trình 6, giá trị của *distance_flag* có thể được thiết đặt là 1 hoặc 0. Giá trị của *distance_idx* có thể được thiết đặt là 1, 2, 4, 8, 16, 32, 64, 128 hoặc giá trị tương tự. *N* thể hiện hệ số được xác định bởi giá trị ngưỡng. Ví dụ, khi giá trị ngưỡng là 4, thì *N* có thể được thiết đặt là 16.

Thông tin chỉ báo chiều của vector độ lệch có thể là thông tin chỉ số chỉ báo một ứng cử bất kỳ trong số các ứng cử chiều vector. Ví dụ, thông tin chỉ số *direction_idx* chỉ báo một ứng cử bất kỳ trong số các ứng cử chiều vector có thể được báo hiệu thông qua luồng bit. Bảng 8 thể hiện dạng nhị phân của thông tin chỉ số *direction_idx* và các chiều của vector độ lệch theo *direction_idx*.

【Bảng 8】

<i>direction_idx[x][y]</i>	Dạng nhị phân	<i>sign[x][y][0]</i>	<i>sign[x][y][1]</i>
0	00	+1	0
1	01	-1	0
2	10	0	+1
3	11	0	-1

Trong bảng 8, *sign[0]* chỉ báo chiều ngang, và *sign[1]* chỉ báo chiều dọc. +1 chỉ báo rằng giá trị của thành phần x hoặc thành phần y của vector độ lệch là dương (+), và -1 chỉ báo rằng giá trị của thành phần x hoặc thành phần y của vector độ lệch là âm (-). Phương

trình 7 thể hiện một ví dụ để xác định vector độ lệch dựa trên độ lớn và chiều của vector độ lệch.

【Phương trình 7】

$$\text{offsetMV}[0] = \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMV}[1] = \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

Trong phương trình 7, offsetMV[0] thể hiện thành phần chiều dọc của vector độ lệch, và offsetMV[1] thể hiện thành phần chiều ngang của vector độ lệch.

Fig.28 là hình vẽ thể hiện vector độ lệch theo các giá trị distance_idx chỉ báo độ lớn của vector độ lệch và direction_idx chỉ báo chiều của vector độ lệch.

Như được thể hiện trong ví dụ trên Fig.28, độ lớn và chiều của vector độ lệch có thể được xác định theo các giá trị distance_idx và direction_idx. Độ lớn tối đa của vector độ lệch có thể được thiết đặt không vượt quá giá trị ngưỡng. Ở đây, giá trị ngưỡng có thể có giá trị định trước trong bộ lập mã và bộ giải mã. Ví dụ, giá trị ngưỡng có thể là 32 khoảng cách mẫu. Ngoài ra, giá trị ngưỡng có thể được xác định theo độ lớn của vector chuyển động khởi tạo. Ví dụ, giá trị ngưỡng đối với chiều ngang có thể được thiết đặt dựa trên độ lớn của thành phần chiều ngang của vector chuyển động khởi tạo, và giá trị ngưỡng đối với chiều dọc có thể được thiết đặt dựa trên độ lớn của thành phần chiều dọc của vector chuyển động khởi tạo.

Khi ứng cử hợp nhất có thông tin chuyển động hai chiều, thì vector chuyển động L0 của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động khởi tạo L0 của khối hiện thời, và vector chuyển động L1 của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động khởi tạo L1 của khối hiện thời. Vào lúc này, vector độ lệch L0 và vector độ lệch L1 có thể được xác định có tính đến giá trị chênh lệch thứ bậc được xuất ra giữa hình tham chiếu L0 của ứng cử hợp nhất và hình hiện thời (sau đây, được gọi là giá trị chênh lệch L0) và giá trị chênh lệch thứ bậc được xuất ra giữa L1 hình tham chiếu của ứng cử hợp nhất và hình hiện thời (sau đây, được gọi là giá trị chênh lệch L1).

Trước hết, khi dấu của giá trị chênh lệch L0 và giá trị chênh lệch L1 giống nhau, thì vector độ lệch L0 và vector độ lệch L1 có thể được thiết đặt là giống nhau. Mặt khác, khi dấu của giá trị chênh lệch L0 và giá trị chênh lệch L1 khác nhau, thì vector độ lệch L1 có thể được thiết đặt theo chiều ngược lại với vector độ lệch L0.

Độ lớn của vector độ lệch L0 và độ lớn của vector độ lệch L1 có thể được thiết đặt là giống nhau. Ngoài ra, độ lớn của vector độ lệch L1 có thể được xác định bằng cách chia tỷ lệ vector độ lệch L0 dựa trên giá trị chênh lệch L0 và giá trị chênh lệch L1.

Ví dụ, phương trình 8 thể hiện vector độ lệch L0 và vector độ lệch L1 khi dấu của giá trị chênh lệch L0 và giá trị chênh lệch L1 giống nhau.

【Phương trình 8】

$$\text{offsetMVL0}[0] = \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMVL0}[1] \equiv \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

$$\text{offsetMVL1}[0] \equiv \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMVL1}[1] = \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

Trong phương trình 8, offsetMVL0[0] chỉ báo thành phần chiều ngang của vector độ lệch L0, và offsetMVL0[1] chỉ báo thành phần chiều dọc của vector độ lệch L0. offsetMVL1[0] chỉ báo thành phần chiều ngang của vector độ lệch L1, và offsetMVL1[1] chỉ báo thành phần chiều dọc của vector độ lệch L1.

Phương trình 9 thể hiện vector độ lệch L0 và vector độ lệch L1 khi dấu của giá trị chênh lệch L0 và giá trị chênh lệch L1 khác nhau.

【Phương trình 9】

$$\text{offsetMVL0}[0] = \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMVL0}[1] \equiv \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

$$\text{offsetMVL1}[0] \equiv -1 * \text{abs}(\text{offsetMV}) * \text{sign}[0]$$

$$\text{offsetMVL1}[1] = -1 * \text{abs}(\text{offsetMV}) * \text{sign}[1]$$

Nhiều hơn bốn ứng cử chiều vector có thể được định nghĩa. Các bảng 9 và 10 thể hiện các ví dụ mà trong đó tám ứng cử chiều vector được định nghĩa.

【Bảng 9】

direction_idx[x][y]	Dạng nhị phân	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1	+1
5	101	+1	-1
6	110	-1	+1
7	111	-1	-1

【Bảng 10】

direction_idx[x][y]	Dạng nhị phân	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1/2	+1/2
5	101	+1/2	-1/2
6	110	-1/2	+1/2
7	111	-1/2	-1/2

Trong các bảng 9 và 10, khi các giá trị tuyệt đối của sign[0] và sign[1] lớn hơn 0, thì nó chỉ báo rằng vector độ lệch theo đường chéo. Khi bảng 9 được sử dụng, thì các độ lớn của các thành phần trục x và trục y của vector độ lệch đường chéo được thiết đặt là abs(offsetMV), trong khi bảng 10 được sử dụng, thì các độ lớn của các thành phần trục x và trục y của vector độ lệch đường chéo được thiết đặt là abs(offsetMV/2).

Fig.29 là hình vẽ thể hiện vector độ lệch theo các giá trị distance_idx chỉ báo độ lớn của vector độ lệch và direction_idx chỉ báo chiều của vector độ lệch.

Fig.29(a) là hình vẽ thể hiện một ví dụ khi bảng 9 được áp dụng, và Fig.29(b) là hình vẽ thể hiện một ví dụ khi bảng 10 được áp dụng.

Thông tin để xác định ít nhất một trong số số lượng và kích thước của các ứng cử chiều vector có thể được báo hiệu thông qua luồng bit. Ví dụ, flag merge_offset_direction_range_flag để xác định các ứng cử chiều vector có thể được báo hiệu thông qua luồng bit. Còn có thể được báo hiệu ở mức trình tự, hình, hoặc phần chia. Ví dụ, khi giá trị của cờ là 0, thì bốn ứng cử chiều vector được lấy ví dụ trong bảng 8 có thể được sử dụng. Mặt khác, khi giá trị của cờ là 1, thì tám ứng cử chiều vector được lấy ví dụ trong bảng 9 hoặc bảng 10 có thể được sử dụng.

Ngoài ra, ít nhất một trong số số lượng và kích thước của các ứng cử chiều vector có thể được xác định dựa trên độ lớn của vector độ lệch. Ví dụ, khi giá trị của biến DistFromMergeMV để xác định độ lớn của vector độ lệch bằng hoặc nhỏ hơn so với giá trị

ngưỡng, thì tám ứng cử chiều vector được lấy ví dụ trong bảng 9 hoặc 10 có thể được sử dụng. Mặt khác, khi giá trị của biến *DistFromMergeMV* lớn hơn so với giá trị ngưỡng, thì bốn ứng cử chiều vector được lấy ví dụ trong bảng 8 có thể được sử dụng.

Ngoài ra, ít nhất một trong số số lượng và kích thước của các ứng cử chiều vector có thể được xác định dựa trên giá trị *MVx* của thành phần x và giá trị *MVy* của thành phần y của vector chuyển động khởi tạo. Ví dụ, khi độ chênh lệch giữa *MVx* và *MVy* hoặc giá trị tuyệt đối của độ chênh lệch này nhỏ hơn hoặc bằng giá trị ngưỡng, thì tám ứng cử chiều vector được lấy ví dụ trong bảng 9 hoặc 10 có thể được sử dụng. Mặt khác, khi độ chênh lệch giữa *MVx* và *MVy* hoặc giá trị tuyệt đối của độ chênh lệch này lớn hơn so với giá trị ngưỡng, thì bốn ứng cử chiều vector được lấy ví dụ trong bảng 8 có thể được sử dụng.

Vector chuyển động của khối hiện thời có thể được suy ra bằng cách cộng vector độ lệch vào vector chuyển động khởi tạo. Phương trình 10 thể hiện một ví dụ để xác định vector chuyển động của khối hiện thời.

【Phương trình 10】

$$mvL0[0] = mergeMVL0[0] + offsetMVL0[0]$$

$$mvL1[0] = mergeMVL1[0] + offsetMVL1[0]$$

$$mvL1[1] = mergeMVL1[1] + offsetMVL1[1]$$

Trong phương trình 10, *mvL0* thể hiện vector chuyển động L0 của khối hiện thời, và *mvL1* thể hiện vector chuyển động L1 của khối hiện thời. *mergeMVL0* thể hiện vector chuyển động khởi tạo L0 của khối hiện thời (tức là, vector chuyển động L0 của ứng cử hợp nhất), và *mergeMVL1* thể hiện vector chuyển động khởi tạo L1 của khối hiện thời. [0] chỉ báo thành phần chiều ngang của vector chuyển động, và [1] chỉ báo thành phần chiều dọc của vector chuyển động.

Ngay cả khi dự báo liên cấu trúc được thực hiện trên mỗi đơn vị con sau khi khối mã hóa được phân chia thành nhiều đơn vị con, thì phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có thể được áp dụng. Ở đây, kỹ thuật thực hiện dự báo liên cấu trúc theo đơn vị con có thể bao gồm ít nhất một kỹ thuật trong số dự báo vector chuyển động theo thời gian cải tiến (ATMVP), dự báo vector chuyển động thời gian không gian (Spatial Temporal Motion Vector Prediction, STMVP), và kỹ thuật phân chia dạng tam giác.

Ví dụ, vector chuyển động khởi tạo có thể được suy ra như sau theo phương pháp ATMVP.

Trước hết, vector dịch khởi tạo có thể được suy ra sử dụng vector chuyển động của ứng cử hợp nhất được suy ra từ khối lân cận gần kề với khối mã hóa. Ngoài ra, khối dịch của khối con được chứa trong khối mã hóa có thể được suy ra sử dụng vector dịch khởi tạo. Phương trình 11 thể hiện vị trí của khối dịch.

【Phương trình 11】

$$(xColSb, yColSb) = (xSb + shVector[0] \gg 4, ySb + shVector[1] \gg 4)$$

Trong phương trình 11, (xColSb, yColSb) thể hiện vị trí của mẫu bên trái-trên cùng của khối dịch, và (xSb, ySb) thể hiện vị trí của mẫu bên trái-trên cùng của khối con. shVector thể hiện vector dịch.

Khi khối dịch được xác định, thì vector chuyển động của khối được sắp xếp cùng một chỗ ở vị trí giống như của khối dịch trong hình được sắp xếp cùng một chỗ có thể được thiết đặt là vector chuyển động của khối con. Tức là, vector chuyển động của khối được sắp xếp cùng một chỗ bao gồm mẫu ở vị trí của (xColSb, yColSb) trong khối được sắp xếp cùng một chỗ có thể được thiết đặt là vector chuyển động của khối con bao gồm mẫu ở vị trí của (xSb, ySb).

Khi kỹ thuật phân chia dạng tam giác được áp dụng, thì khối mã hóa có thể được phân chia thành các đơn vị con hình tam giác. Ví dụ, khối mã hóa có thể được phân chia thành hai đơn vị con bởi đường chéo nối các góc bên trái-trên cùng và bên phải-dưới cùng của khối mã hóa hoặc đường chéo nối các góc bên phải-trên cùng và bên trái-dưới cùng của khối mã hóa.

Fig.30 là hình vẽ thể hiện các mẫu phân chia của khối mã hóa khi kỹ thuật phân chia dạng tam giác được áp dụng.

Thông tin chuyển động của mỗi đơn vị con hình tam giác có thể được định rõ bởi ứng cử hợp nhất. Để làm được điều này, thông tin chỉ số chỉ báo một ứng cử bất kỳ trong số các ứng cử hợp nhất có thể được báo hiệu cho mỗi đơn vị con. Ví dụ, thông tin chỉ số merge_1st_idx của đơn vị con thứ nhất có thể định rõ ứng cử hợp nhất của đơn vị con thứ

nhất, và thông tin chỉ số `merge_2nd_idx` của đơn vị con thứ hai có thể định rõ ứng cử hợp nhất của đơn vị con thứ hai.

Vector chuyển động khởi tạo của mỗi đơn vị con có thể được xác định một cách độc lập. Ví dụ, khi mô hình chuyển động afin được áp dụng cho khối mã hóa, thì các vector afin của các khối con được suy ra từ các vector hạt giống afin của khối mã hóa có thể được thiết đặt là các vector chuyển động khởi tạo của các khối con. Vector chuyển động của mỗi khối con có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động khởi tạo.

Khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối mã hóa được phân chia thành nhiều đơn vị con, thì nhiều đơn vị con có thể được thiết đặt để sử dụng cùng một vector độ lệch. Tức là, vector chuyển động khởi tạo của mỗi đơn vị con trong số nhiều đơn vị con có thể được thay đổi sử dụng cùng một vector độ lệch.

Ngoài ra, khối mã hóa được phân chia thành nhiều đơn vị con, và vector độ lệch của mỗi đơn vị con có thể được xác định một cách độc lập. Do đó, vector độ lệch của ít nhất một trong số các đơn vị con có thể được thiết đặt khác với các vector độ lệch của các đơn vị con khác.

Fig.31 là hình vẽ thể hiện một ví dụ mà trong đó các vector độ lệch của mỗi đơn vị con được thiết đặt khác nhau.

Như được thể hiện trong ví dụ trên Fig.31, thông tin `distance_idx` chỉ báo độ lớn của vector độ lệch và `direction_idx` chỉ báo chiều của vector độ lệch có thể được lập mã và được báo hiệu cho mỗi đơn vị con.

Ngoài ra, độ lớn của vector độ lệch có thể được thiết đặt giống nhau cho tất cả các đơn vị con, và chiều của vector độ lệch có thể được thiết đặt một cách độc lập cho các đơn vị con. Ví dụ, có thể được thiết đặt để chia sẻ giá trị của `distance_idx` được báo hiệu ở mức mã hóa trong số các đơn vị con, và `direction_idx` có thể được lập mã và được báo hiệu cho mỗi đơn vị con.

Ngoài ra, chiều của vector độ lệch có thể được thiết đặt giống nhau cho tất cả các đơn vị con, và độ lớn của vector độ lệch có thể được thiết đặt một cách độc lập cho các đơn vị con. Ví dụ, có thể được thiết đặt để chia sẻ giá trị của `direction_idx` được báo hiệu ở mức

mã hóa trong số các đơn vị con, và `distance_idx` có thể được lập mã và được báo hiệu cho mỗi đơn vị con.

Phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có thể được áp dụng chỉ với một số đơn vị con trong số nhiều đơn vị con được tạo ra bằng cách phân chia khối mã hóa. Ví dụ, khi khối hiện thời được phân chia thành đơn vị con thứ nhất và đơn vị con thứ hai, thì vector chuyển động của đơn vị con thứ nhất có thể được thiết đặt giống như vector chuyển động của ứng cử hợp nhất, và vector chuyển động của đơn vị con thứ hai có thể được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất.

Thay vì báo hiệu thông tin để xác định vector độ lệch, thì bộ giải mã có thể suy ra vector độ lệch. Cụ thể là, vector độ lệch có thể được suy ra sử dụng giá trị trung bình đối với các gradient chiều ngang và giá trị trung bình đối với các gradient chiều dọc của các mẫu dự báo được chứa trong khối con.

Ở đây, gradient có thể được suy ra dựa trên độ chênh lệch giữa mẫu được cấu thành lại tương ứng với mẫu dự báo trong hình tham chiếu và mẫu lân cận gần kề với mẫu được cấu thành lại. Ví dụ, gradient chiều ngang có thể chỉ báo độ chênh lệch giữa mẫu được cấu thành lại và mẫu được cấu thành lại lân cận trên cạnh bên trái và/hoặc cạnh bên phải, và gradient chiều dọc có thể chỉ báo độ chênh lệch giữa mẫu được cấu thành lại và mẫu được cấu thành lại lân cận trên cạnh trên cùng và/hoặc cạnh dưới cùng.

Trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất, thì ứng cử hợp nhất có vector chuyển động được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất tham chiếu trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất có thể được cộng vào danh sách ứng cử hợp nhất. Ứng cử hợp nhất có vector chuyển động được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất tham chiếu có thể được gọi là ứng cử hợp nhất sàng lọc.

Thông tin chuyển động còn lại không bao gồm vector chuyển động của ứng cử hợp nhất sàng lọc có thể được thiết đặt giống như của ứng cử hợp nhất tham chiếu.

Fig.32 là hình vẽ thể hiện các ứng cử vector chuyển động mà ứng cử hợp nhất sàng lọc có thể nhận lấy.

Khi vector chuyển động của ứng cử hợp nhất tham chiếu là $(MvLX[0], MvLX[1])$, thì vector chuyển động của ứng cử hợp nhất sàng lọc có thể được suy ra bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi ít nhất một trong số thành phần x và thành phần y của vector chuyển động của ứng cử hợp nhất tham chiếu. Ví dụ, vector chuyển động của ứng cử hợp nhất sàng lọc có thể được thiết đặt là $(MvLX[0]+M, MvLX[1])$, $(MvLX[0]-M, MvLX[1])$, $(MvLX[0], MvLX[1]+M)$ hoặc $(MvLX[0], MvLX[1]-M)$. M thể hiện độ lớn của vector độ lệch.

Ứng cử hợp nhất tham chiếu có thể là ứng cử hợp nhất có giá trị chỉ số định trước trong danh sách ứng cử hợp nhất. Ví dụ, ứng cử hợp nhất có giá trị chỉ số nhỏ nhất (tức là, ứng cử hợp nhất có giá trị chỉ số 0) hoặc ứng cử hợp nhất có giá trị chỉ số lớn nhất trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất có thể được thiết đặt là ứng cử hợp nhất tham chiếu. Ngoài ra, ứng cử hợp nhất liên vùng có giá trị chỉ số nhỏ nhất hoặc ứng cử hợp nhất liên vùng có giá trị chỉ số lớn nhất trong danh sách thông tin chuyển động liên vùng có thể được thiết đặt là ứng cử hợp nhất tham chiếu.

Ngoài ra, ứng cử hợp nhất có giá trị chỉ số nhỏ nhất trong số các ứng cử hợp nhất có thông tin chuyển động hai chiều có thể được thiết đặt là ứng cử hợp nhất tham chiếu. Tức là, khi các khối ứng cử được tìm kiếm theo trình tự, thì ứng cử hợp nhất hai chiều được tìm thấy thứ nhất có thể được thiết đặt là ứng cử hợp nhất tham chiếu.

Ứng cử hợp nhất cơ bản có thể được chọn lựa dựa trên kích thước của khối hiện thời, hình dạng của khối hiện thời, hoặc việc khối hiện thời có tiếp xúc với biên của đơn vị cây mã hóa hay không. Ví dụ, khi khối hiện thời có hình dạng vuông hoặc khối hiện thời có hình dạng không vuông mà chiều cao lớn hơn so với chiều rộng, thì ứng cử hợp nhất có chỉ số 0 hoặc ứng cử hợp nhất được suy ra từ khối lân cận được đặt ở vị trí trên phần trên cùng của khối hiện thời có thể được thiết đặt là ứng cử hợp nhất tham chiếu. Khi khối hiện thời có hình dạng không vuông mà chiều rộng lớn hơn so với chiều cao, thì ứng cử hợp nhất có chỉ số 1 hoặc ứng cử hợp nhất được suy ra từ khối lân cận được đặt ở vị trí trên cạnh bên trái của khối hiện thời có thể được thiết đặt là ứng cử hợp nhất tham chiếu.

Ngoài ra, thông tin định rõ ứng cử hợp nhất tham chiếu có thể được báo hiệu thông qua luồng bit. Thông tin này có thể là thông tin chỉ số định rõ một ứng cử bất kỳ trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất.

Thông tin chỉ báo việc có sử dụng hay không ứng cử hợp nhất sàng lọc có thể được báo hiệu thông qua luồng bit. Thông tin có thể là cờ 1-bit. Khi giá trị của cờ là 1, thì ứng cử hợp nhất sàng lọc được tạo ra dựa trên ứng cử hợp nhất tham chiếu có thể được cộng vào danh sách ứng cử hợp nhất. Mặt khác, khi giá trị của cờ là 0, thì danh sách ứng cử hợp nhất có thể không bao gồm ứng cử hợp nhất sàng lọc.

Ngoài ra, khi số lượng các ứng cử hợp nhất được cộng trước đó vào danh sách ứng cử hợp nhất nhỏ hơn so với số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm, thì ứng cử hợp nhất sàng lọc có thể được cộng vào danh sách ứng cử hợp nhất. Ở đây, các ứng cử hợp nhất được cộng vào trước đó có thể bao gồm ít nhất một trong số ứng cử hợp nhất không gian, ứng cử hợp nhất thời gian, ứng cử hợp nhất liên vùng, và ứng cử hợp nhất theo cặp. Ví dụ, khi số lượng của ít nhất một trong số các ứng cử hợp nhất không gian, các ứng cử hợp nhất thời gian, và các ứng cử hợp nhất liên vùng được chứa trong danh sách ứng cử hợp nhất nhỏ hơn hoặc bằng giá trị ngưỡng, thì ứng cử hợp nhất sàng lọc có thể được cộng vào danh sách ứng cử hợp nhất.

Ngoài ra, khi số lượng các ứng cử hợp nhất được cộng trước đó vào danh sách ứng cử hợp nhất lớn hơn hoặc bằng giá trị ngưỡng, thì ứng cử hợp nhất sàng lọc có thể được sử dụng.

Số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm có thể được thiết đặt khác nhau theo việc ứng cử hợp nhất sàng lọc có được sử dụng hay không. Ví dụ, khi được thiết đặt để không sử dụng ứng cử hợp nhất sàng lọc, thì số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm có thể được thiết đặt là N , trong khi được thiết đặt để sử dụng ứng cử hợp nhất sàng lọc, thì số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm có thể được thiết đặt là $N + n$.

Ứng cử hợp nhất sàng lọc có thể có chỉ số lớn hơn chỉ số của các ứng cử hợp nhất được cộng trước đó vào danh sách ứng cử hợp nhất. Ví dụ, bảng 11 thể hiện một ví dụ để tạo cấu hình danh sách ứng cử hợp nhất.

【Bảng 11】

mergeCand[0]
mergeCand[1]
mergeCand[2]
mergeCand[3]
mergeCand[4]
mergeCand[5]
mergeCand[6]: Ứng cử hợp nhất sàng lọc mà có vector chuyển động là $(mvLx[0]+M, mvLx[1])$
mergeCand[7]: Ứng cử hợp nhất sàng lọc mà có vector chuyển động là $(mvLx[0]-M, mvLx[1])$
mergeCand[8]: Ứng cử hợp nhất sàng lọc mà có vector chuyển động là $(mvLx[0], mvLx[1]+M)$
mergeCand[9]: Ứng cử hợp nhất sàng lọc mà có vector chuyển động là $(mvLx[0], mvLx[1]-M)$

Trong bảng 11, mergeCand[X] thể hiện ứng cử hợp nhất có chỉ số X. mvLx[0] thể hiện vector chuyển động thành phần x của ứng cử hợp nhất tham chiếu, và mvLx[1] thể hiện vector chuyển động thành phần y của ứng cử hợp nhất tham chiếu. Ví dụ, khi ứng cử hợp nhất tham chiếu là mergeCand[0], mvLx[0] và mvLx[1] có thể chỉ báo vector chuyển động của mergeCand[0].

Độ lớn M của vector độ lệch có thể được định trước trong bộ lập mã và bộ giải mã. Ví dụ, độ lớn M của vector độ lệch có thể được thiết đặt là số nguyên nhỏ hơn hoặc bằng 4, chẳng hạn như 1 hoặc 4.

Ngoài ra, thông tin để xác định vector độ lệch có thể được báo hiệu thông qua luồng bit. Thông tin này có thể được báo hiệu ở mức trình tự, hình, phần chia, hoặc khối. Ví dụ, vector độ lệch có thể được xác định sử dụng ít nhất một trong số thông tin distance_idx để

xác định độ lớn của vector độ lệch và thông tin `direction_idx` để xác định chiều của vector độ lệch được mô tả ở trên.

Như được thể hiện trong ví dụ trong bảng 11, ít nhất một ứng cử hợp nhất sàng lọc được suy ra dựa trên ứng cử hợp nhất tham chiếu có thể được cộng vào danh sách ứng cử hợp nhất. Khi có ứng cử hợp nhất có thông tin chuyển động giống như của ứng cử hợp nhất sàng lọc trong số các ứng cử hợp nhất được cộng vào trước đó, thì ứng cử hợp nhất sàng lọc có thể không được cộng vào danh sách ứng cử hợp nhất. Ví dụ, khi ứng cử hợp nhất sàng lọc được suy ra dựa trên ứng cử hợp nhất tham chiếu `mergeCand[0]` giống như một ứng cử bất kỳ trong số các ứng cử hợp nhất của `mergeCand[1]` đến `mergeCand[5]`, thì ứng cử hợp nhất sàng lọc có thể không được cộng vào danh sách ứng cử hợp nhất.

Ngoài ra, khi có ứng cử hợp nhất có thông tin chuyển động giống như của ứng cử hợp nhất sàng lọc, thì ứng cử hợp nhất sàng lọc có thể được suy ra lại bằng cách thay đổi vector độ lệch hoặc bằng cách thiết đặt lại ứng cử hợp nhất có thông tin chuyển động giống như của ứng cử hợp nhất sàng lọc như ứng cử hợp nhất sàng lọc. Ví dụ, khi thông tin chuyển động của ứng cử hợp nhất sàng lọc `mergeCand[6]` được suy ra dựa trên ứng cử hợp nhất tham chiếu `mergeCand[0]` giống như của ứng cử hợp nhất `mergeCand[2]`, thì vector chuyển động của ứng cử hợp nhất sàng lọc `mergeCand[6]` có thể được thay đổi thành giá trị nhận được bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất [2]. Ví dụ, vector chuyển động của `mergeCand[6]` có thể được thay đổi từ $(\text{mergeCand}[0]_{\text{mxLx}}[0] + M, \text{mergeCand}[0]_{\text{mvLx}}[1])$ thành $(\text{mergeCand}[2]_{\text{mxLx}}[0] + M, \text{mergeCand}[2]_{\text{mvLx}}[1])$. Ở đây, `mergeCand[X]_mvLx` thể hiện vector chuyển động của ứng cử hợp nhất có chỉ số X.

Ví dụ khác, vector độ lệch có thể được xác định sử dụng danh sách độ lệch sàng lọc hợp nhất bao gồm ít nhất một ứng cử độ lệch hợp nhất. Khi ứng cử hợp nhất được định rõ bởi thông tin chỉ số của khối hiện thời là ứng cử hợp nhất tham chiếu, thì vector độ lệch có thể được xác định sử dụng danh sách độ lệch sàng lọc hợp nhất. Ngoài ra, vector chuyển động của khối hiện thời có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất. Ứng cử hợp nhất tham chiếu có thể là ứng cử hợp nhất có giá trị chỉ số định trước trong danh sách ứng cử hợp nhất. Ví dụ, trong

số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất, thì ứng cử hợp nhất có giá trị chỉ số nhỏ nhất (tức là, ứng cử hợp nhất có giá trị chỉ số 0) hoặc ứng cử hợp nhất có giá trị chỉ số lớn nhất có thể được thiết đặt là ứng cử hợp nhất tham chiếu. Ngoài ra, ứng cử hợp nhất liên vùng có giá trị chỉ số nhỏ nhất hoặc ứng cử hợp nhất liên vùng có giá trị chỉ số lớn nhất trong danh sách thông tin chuyển động liên vùng có thể được thiết đặt là ứng cử hợp nhất tham chiếu.

Fig.33 là hình vẽ thể hiện cấu hình của danh sách độ lệch sàng lọc hợp nhất.

Trên Fig.33, được giả định rằng ứng cử hợp nhất tham chiếu là ứng cử hợp nhất có chỉ số 6.

Khi chỉ số của ứng cử hợp nhất được định rõ bởi thông tin chỉ số `merge_idx` chỉ báo một ứng cử bất kỳ trong số các ứng cử hợp nhất không phải là 6, thì vector chuyển động của ứng cử hợp nhất có thể được thiết đặt là vector chuyển động của khối hiện thời.

Mặt khác, khi chỉ số của ứng cử hợp nhất được định rõ bởi thông tin chỉ số `merge_idx` là 6, thì vector độ lệch có thể được suy ra sử dụng danh sách độ lệch sàng lọc hợp nhất. Thông tin chỉ số `MrgOffset_idx` định rõ một ứng cử bất kỳ trong số các ứng cử độ lệch hợp nhất được chứa trong danh sách độ lệch sàng lọc hợp nhất có thể được báo hiệu thông qua luồng bit.

Khi vector độ lệch được định rõ, thì vector chuyển động của khối hiện thời có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử hợp nhất tham chiếu.

Danh sách độ lệch sàng lọc hợp nhất có thể bao gồm ít nhất một ứng cử độ lệch hợp nhất. Ví dụ, số lượng của các ứng cử độ lệch hợp nhất mà danh sách độ lệch sàng lọc hợp nhất bao gồm có thể là 4, 8 hoặc 16.

Fig.34 và Fig.35 là các hình vẽ thể hiện các vector độ lệch được định rõ bởi các ứng cử độ lệch hợp nhất.

Fig.34 thể hiện một ví dụ mà trong đó số lượng của các ứng cử độ lệch hợp nhất là 8, và Fig.35 thể hiện một ví dụ mà trong đó số lượng của các ứng cử độ lệch hợp nhất là 16.

Như được thể hiện trong ví dụ trên Fig.34(a), các vector độ lệch được chỉ báo bởi các ứng cử độ lệch hợp nhất có thể được thiết đặt sao cho giá trị tuyệt đối của vector chuyển

động của chiều ngang và/hoặc giá trị tuyệt đối của vector chuyển động của chiều dọc có thể có giá trị cố định. Ngoài ra, như được thể hiện trong ví dụ trên Fig.35, các vector độ lệch được chỉ báo bởi các ứng cử độ lệch hợp nhất có chỉ số nhỏ hơn so với giá trị ngưỡng trong số các ứng cử độ lệch hợp nhất có thể được thiết đặt sao cho giá trị tuyệt đối của vector chuyển động của chiều ngang và/hoặc giá trị tuyệt đối của vector chuyển động của chiều dọc có thể có giá trị thứ nhất, và các vector độ lệch được chỉ báo bởi các ứng cử độ lệch hợp nhất khác có thể được thiết đặt sao cho giá trị tuyệt đối của vector chuyển động của chiều ngang và/hoặc giá trị tuyệt đối của vector chuyển động của chiều dọc có thể có giá trị thứ hai.

Ngoài ra, như được thể hiện trên Fig.34(b), các vector độ lệch được chỉ báo bởi các ứng cử độ lệch hợp nhất có thể được thiết đặt sao cho tổng của giá trị tuyệt đối của vector chuyển động của chiều ngang và giá trị tuyệt đối của vector chuyển động của chiều dọc có thể có giá trị cố định.

Nhiều ứng cử hợp nhất tham chiếu có thể được thiết đặt. Ví dụ, trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất, hai ứng cử hợp nhất có chỉ số nhỏ nhất có thể được thiết đặt là các ứng cử hợp nhất tham chiếu. Do đó, khi chỉ số của ứng cử hợp nhất được định rõ bởi thông tin chỉ số `merge_idx` là 0 hoặc 1, thì vector độ lệch có thể được suy ra sử dụng danh sách độ lệch sàng lọc hợp nhất. Ngoài ra, ứng cử hợp nhất có chỉ số nhỏ nhất trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất và ứng cử hợp nhất có chỉ số lớn nhất trong số các ứng cử hợp nhất được chứa trong danh sách ứng cử hợp nhất liên vùng có thể được thiết đặt là các ứng cử hợp nhất tham chiếu.

Trong chế độ dự báo vector chuyển động cải tiến, vector chuyển động của khối hiện thời có thể được suy ra bằng cách cộng vector độ chênh lệch chuyển động vào vector dự báo chuyển động. Vector dự báo chuyển động của khối hiện thời có thể được xác định dựa trên danh sách ứng cử dự báo vector chuyển động bao gồm ít nhất một ứng cử vector dự báo chuyển động. Ví dụ, một ứng cử bất kỳ trong số các ứng cử vector dự báo chuyển động có thể được thiết đặt là vector dự báo chuyển động của khối hiện thời.

Ứng cử dự báo vector chuyển động có thể được suy ra dựa trên ít nhất một trong số khối lân cận về mặt không gian của khối hiện thời và khối lân cận về mặt thời gian của khối hiện thời.

Fig.36 là hình vẽ thể hiện các khối ứng cử được sử dụng để suy ra các ứng cử dự báo vector chuyển động.

Khối lân cận về mặt không gian có thể bao gồm các khối lân cận trên cùng được đặt ở vị trí trên phần trên cùng của khối hiện thời và các khối lân cận bên trái được đặt ở vị trí trên cạnh bên trái của khối hiện thời. Các khối lân cận trên cùng có thể bao gồm ít nhất một trong số khối B0 bao gồm mẫu ở vị trí $(x_{Cb}+CbW, y_{Cb}-1)$, khối B1 bao gồm mẫu ở vị trí $(x_{Cb}+CbW-1, y_{Cb}-1)$, khối B2 bao gồm mẫu ở vị trí $(x_{Cb}-1, y_{Cb}-1)$, và khối B3 bao gồm mẫu ở vị trí $(x_{Cb}, y_{Cb}-1)$. Ở đây, (x_{Cb}, y_{Cb}) thể hiện vị trí của mẫu bên trái-trên cùng của khối hiện thời, và CbW thể hiện chiều rộng của khối hiện thời. Các khối lân cận bên trái có thể bao gồm ít nhất một trong số khối A0 bao gồm mẫu ở vị trí $(x_{Cb}-1, y_{Cb}+CbH)$, khối A1 bao gồm mẫu ở vị trí $(x_{Cb}-1, y_{Cb}+CbH-1)$, và khối A2 bao gồm mẫu ở vị trí $(x_{Cb}-1, y_{Cb})$. Ở đây, CbH thể hiện chiều cao của khối hiện thời.

Khối lân cận về mặt thời gian có thể bao gồm ít nhất một trong số khối C0 bao gồm mẫu ở vị trí trung tâm của khối có vị trí và kích thước giống như của khối hiện thời trong khối được sắp xếp cùng một chỗ, và khối C1 bao gồm mẫu gần kề với góc bên phải-dưới cùng của khối này.

Số lượng tối đa của các ứng cử dự báo vector chuyển động mà danh sách ứng cử dự báo vector chuyển động có thể bao gồm có thể là hai. Trình tự để suy ra các ứng cử dự báo vector chuyển động như được mô tả bên dưới.

1. Khi ít nhất một trong số khối lân cận bên trái A0 và khối lân cận bên trái A1 khả dụng, thì vector chuyển động của khối khả dụng được thiết đặt là ứng cử dự báo vector chuyển động.

2. Khi ít nhất một trong số khối lân cận trên cùng B0, khối lân cận trên cùng B1, và khối lân cận trên cùng B2 khả dụng, thì vector chuyển động của khối khả dụng được thiết đặt là ứng cử dự báo vector chuyển động.

3. Khi khối lân cận về mặt thời gian khả dụng, thì vector chuyển động thời gian được thiết đặt là ứng cử dự báo vector chuyển động.

4. Vector chuyển động zero được thiết đặt là ứng cử dự báo vector chuyển động.

Ngoài ra, khi số lượng các ứng cử dự báo vector chuyển động được suy ra theo trình tự 1 đến 3 nhỏ hơn hai, thì vector chuyển động được chứa trong danh sách thông tin chuyển động liên vùng có thể được thiết đặt là ứng cử dự báo vector chuyển động. Khi danh sách thông tin chuyển động liên vùng khả dụng, thì các ứng cử dự báo vector chuyển động có thể được suy ra theo trình tự được mô tả bên dưới.

1. Khi ít nhất một trong số khối lân cận bên trái A0 và khối lân cận bên trái A1 khả dụng, thì vector chuyển động của khối khả dụng được thiết đặt là ứng cử dự báo vector chuyển động.

2. Khi ít nhất một trong số khối lân cận trên cùng B0, khối lân cận trên cùng B1, và khối lân cận trên cùng B2 khả dụng, thì vector chuyển động của khối khả dụng được thiết đặt là ứng cử dự báo vector chuyển động.

3. Khi khối lân cận về mặt thời gian khả dụng, thì vector chuyển động thời gian được thiết đặt là ứng cử dự báo vector chuyển động.

4. Vector chuyển động được chứa trong danh sách thông tin chuyển động liên vùng được thiết đặt là ứng cử dự báo vector chuyển động.

5. Vector chuyển động zero được thiết đặt là ứng cử dự báo vector chuyển động.

Ứng cử dự báo vector chuyển động có vector chuyển động được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử dự báo vector chuyển động tham chiếu có thể được cộng vào danh sách ứng cử dự báo vector chuyển động. Ứng cử dự báo vector chuyển động có vector chuyển động được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử dự báo vector chuyển động tham chiếu có thể được gọi là ứng cử dự báo vector chuyển động sàng lọc.

Fig.37 là hình vẽ thể hiện các ứng cử vector chuyển động mà có thể được thiết đặt là ứng cử dự báo vector chuyển động sàng lọc.

Khi vector chuyển động của ứng cử dự báo vector chuyển động tham chiếu là $(MvpLX[0], MvpLX[1])$, thì vector chuyển động của ứng cử dự báo vector chuyển động sàng

lọc có thể được suy ra bằng cách cộng hoặc trừ đi độ lệch vào hoặc khỏi ít nhất một trong số thành phần x và thành phần y của vector chuyển động của ứng cử dự báo vector chuyển động tham chiếu. Ví dụ, vector chuyển động của ứng cử dự báo vector chuyển động sàng lọc có thể được thiết đặt là $(MvpLX[0]+M, MvpLX[1])$, $(MvpLX[0]-M, MvpLX[1])$, $(MvpLX[0], MvpLX[1]+M)$ hoặc $(MvpLX[0], MvpLX[1]-M)$. M thể hiện độ lớn của vector độ lệch.

Độ lớn M của vector độ lệch có thể được định trước trong bộ lập mã và bộ giải mã. Ví dụ, độ lớn M của vector độ lệch có thể được thiết đặt là số nguyên nhỏ hơn hoặc bằng 4, chẳng hạn như 1 hoặc 4.

Ngoài ra, thông tin để xác định vector độ lệch có thể được báo hiệu thông qua luồng bit. Thông tin này có thể được báo hiệu ở mức trình tự, hình, phần chia, hoặc khối. Ví dụ, vector độ lệch có thể được xác định sử dụng ít nhất một thông tin trong số thông tin distance_idx để xác định độ lớn của vector độ lệch và thông tin direction_idx để xác định chiều của vector độ lệch được mô tả ở trên.

Ứng cử dự báo vector chuyển động tham chiếu có thể là ứng cử dự báo vector chuyển động có giá trị chỉ số định trước trong danh sách ứng cử dự báo vector chuyển động. Ví dụ, trong số các ứng cử dự báo vector chuyển động được chứa trong danh sách ứng cử dự báo vector chuyển động, thì ứng cử dự báo vector chuyển động có giá trị chỉ số 0 hoặc ứng cử dự báo vector chuyển động có giá trị chỉ số 1 có thể được thiết đặt là ứng cử dự báo vector chuyển động tham chiếu.

Ví dụ khác, vector độ lệch có thể được xác định sử dụng danh sách độ lệch sàng lọc hợp nhất bao gồm ít nhất một ứng cử độ lệch vector dự báo. Khi ứng cử dự báo vector chuyển động được định rõ bởi thông tin chỉ số của khối hiện thời là ứng cử dự báo vector chuyển động tham chiếu, thì vector độ lệch có thể được xác định sử dụng danh sách độ lệch sàng lọc vector dự báo. Ngoài ra, vector dự báo chuyển động của khối hiện thời có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử dự báo vector chuyển động. Ứng cử dự báo vector chuyển động tham chiếu có thể là ứng cử dự báo vector chuyển động có giá trị chỉ số định trước trong danh sách ứng cử dự báo vector chuyển động. Ví dụ, trong số các ứng cử dự báo vector chuyển động được chứa trong danh

sách ứng cử dự báo vector chuyển động, thì ứng cử dự báo vector chuyển động có giá trị chỉ số nhỏ nhất hoặc ứng cử dự báo vector chuyển động có giá trị chỉ số lớn nhất có thể được thiết đặt là ứng cử dự báo vector chuyển động tham chiếu.

Khi vector độ lệch được tính sử dụng danh sách sàng lọc độ lệch vector dự báo, thì số lượng tối đa của các ứng cử vector dự báo mà danh sách ứng cử vector dự báo có thể bao gồm có thể được thiết đặt là giá trị lớn hơn 2.

Fig.38 là hình vẽ thể hiện cấu hình của danh sách độ lệch sàng lọc vector dự báo.

Trên Fig.38, được giả định rằng ứng cử vector dự báo tham chiếu là ứng cử vector dự báo có chỉ số 2.

Khi chỉ số của ứng cử vector dự báo được định rõ bởi thông tin chỉ số AMVPcand_idx chỉ báo một ứng cử bất kỳ trong số các ứng cử vector dự báo không phải là 2, thì vector chuyển động của ứng cử vector dự báo có thể được thiết đặt là vector dự báo chuyển động của khối hiện thời.

Mặt khác, khi chỉ số của ứng cử vector dự báo được định rõ bởi thông tin chỉ số AMVPcand_idx là 2, thì vector độ lệch có thể được suy ra sử dụng danh sách độ lệch sàng lọc vector dự báo. Thông tin chỉ số AMVPOffset_idx định rõ một ứng cử bất kỳ trong số các ứng cử độ lệch vector dự báo được chứa trong danh sách độ lệch sàng lọc vector dự báo có thể được báo hiệu thông qua luồng bit.

Khi vector độ lệch được định rõ, thì vector dự báo chuyển động của khối hiện thời có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động của ứng cử vector dự báo tham chiếu.

Ngay cả khi khối mã hóa được lập mã dựa trên mô hình chuyển động afin, thì kỹ thuật sàng lọc của vector chuyển động có thể được sử dụng. Ví dụ, khi chế độ dự báo vector chuyển động cải tiến afin được áp dụng, thì các vector hạt giống afin của khối mã hóa có thể được suy ra bằng cách cộng vector độ chênh lệch hạt giống afin vào vector dự báo hạt giống afin. Ở đây, vector dự báo hạt giống afin có thể được suy ra dựa trên vector hạt giống afin của khối lân cận về mặt không gian hoặc khối lân cận về mặt thời gian của khối mã hóa. Vector độ chênh lệch hạt giống afin có thể được xác định dựa trên thông tin được báo hiệu từ luồng bit. Vào lúc này, cùng một vector độ chênh lệch hạt giống afin có thể được áp dụng cho tất

cả các điểm điều khiển. Ngoài ra, thông tin để xác định vector hạt giống afin có thể được báo hiệu cho mỗi điểm điều khiển.

Khi vector afin dành cho khối con được suy ra dựa trên các vector hạt giống afin của khối mã hóa, thì vector afin có thể được thiết đặt là vector chuyển động khởi tạo, và sau đó vector độ lệch có thể được suy ra. Vector chuyển động của mỗi khối con có thể được suy ra bằng cách cộng hoặc trừ đi vector độ lệch vào hoặc khỏi vector chuyển động khởi tạo.

Thay vì báo hiệu thông tin để xác định vector độ lệch, thì bộ giải mã có thể suy ra vector độ lệch. Cụ thể là, vector độ lệch có thể được suy ra sử dụng giá trị trung bình đối với các gradien chiều ngang và giá trị trung bình đối với các gradien chiều dọc của các mẫu dự báo được chứa trong khối con.

Dự báo nội cấu trúc để dự báo khối hiện thời sử dụng các mẫu được cấu thành lại mà đã được lập mã/giải mã trong vùng lân cận của khối hiện thời. Vào lúc này, các mẫu được cấu thành lại trước bộ lọc vòng trong được áp dụng có thể được sử dụng dành cho việc dự báo nội cấu trúc của khối hiện thời.

Kỹ thuật dự báo nội cấu trúc bao gồm dự báo nội cấu trúc dựa trên ma trận, và dự báo nội cấu trúc thông thường có tính đến chiều so với các mẫu được cấu thành lại lân cận. Thông tin chỉ báo kỹ thuật dự báo nội cấu trúc của khối hiện thời có thể được báo hiệu thông qua luồng bit. Thông tin có thể là cờ 1-bit. Ngoài ra, kỹ thuật dự báo nội cấu trúc của khối hiện thời có thể được xác định dựa trên ít nhất một trong số vị trí, kích thước, và hình dạng của khối hiện thời, hoặc dựa trên kỹ thuật dự báo nội cấu trúc của khối lân cận. Ví dụ, khi khối hiện thời tồn tại trên biên hình, thì có thể được thiết đặt để không áp dụng dự báo nội cấu trúc dựa trên ma trận cho khối hiện thời.

Dự báo nội cấu trúc dựa trên ma trận là phương pháp để đạt được khối dự báo của khối hiện thời bởi bộ lập mã và bộ giải mã dựa trên tích ma trận giữa ma trận được lưu trước đó và các mẫu được cấu thành lại trong vùng lân cận của khối hiện thời. Thông tin để định rõ một ma trận bất kỳ trong số nhiều ma trận được lưu trước đó có thể được báo hiệu thông qua luồng bit. Bộ giải mã có thể xác định ma trận để dự báo nội cấu trúc của khối hiện thời dựa trên thông tin và kích thước của khối hiện thời.

Dự báo nội cấu trúc thông thường là phương pháp để đạt được khối dự báo dành cho khối hiện thời dựa trên chế độ dự báo nội cấu trúc không góc hoặc chế độ dự báo nội cấu trúc có góc.

Hình còn dư được suy ra có thể được suy ra bằng cách trừ đi video dự báo khỏi video gốc. Vào lúc này, khi video còn dư được thay đổi tới miền tần số, thì chất lượng video chủ thể của video không bị hạ thấp đáng kể mặc dù các thành phần tần số cao trong số các thành phần tần số bị loại bỏ. Do đó, khi các giá trị của các thành phần tần số cao được biến đổi thành giá trị nhỏ hoặc các giá trị của các thành phần tần số cao được thiết đặt là 0, thì có tác dụng làm tăng hiệu quả nén mà không tạo ra méo trực quan đáng kể. Bằng cách phản ánh đặc trưng này, khối hiện thời có thể được biến đổi để phân giải video còn dư thành các thành phần tần số hai chiều. Hoạt động biến đổi có thể được thực hiện sử dụng kỹ thuật biến đổi chẳng hạn như biến đổi cosin rời rạc (DCT) hoặc biến đổi sin rời rạc (DST).

Sau khi khối hiện thời được biến đổi sử dụng DCT hoặc DST, thì khối hiện thời được biến đổi có thể được biến đổi lại. Vào lúc này, hoạt động biến đổi dựa trên DCT hoặc DST có thể được định nghĩa là biến đổi thứ nhất, và việc biến đổi lại khối mà biến đổi thứ nhất được áp dụng có thể được định nghĩa là biến đổi thứ hai.

Biến đổi thứ nhất có thể được thực hiện sử dụng một ứng cử bất kỳ trong số nhiều ứng cử lỗi biến đổi. Ví dụ, biến đổi thứ nhất có thể được thực hiện sử dụng một lỗi biến đổi bất kỳ trong số DCT2, DCT8, hoặc DCT7.

Các lỗi biến đổi khác nhau có thể được sử dụng cho chiều ngang và chiều dọc. Thông tin chỉ báo dạng kết hợp của lỗi biến đổi của chiều ngang và lỗi biến đổi của chiều dọc có thể được báo hiệu thông qua luồng bit.

Các đơn vị để thực hiện biến đổi thứ nhất và biến đổi thứ hai có thể khác nhau. Ví dụ, biến đổi thứ nhất có thể được thực hiện trên khối 8×8 , và biến đổi thứ hai có thể được thực hiện trên khối con có kích thước 4×4 trong số khối 8×8 được biến đổi. Vào lúc này, các hệ số biến đổi của các vùng còn dư mà không được thực hiện biến đổi thứ hai có thể được thiết đặt là 0.

Ngoài ra, biến đổi thứ nhất có thể được thực hiện trên khối 4×4 , và biến đổi thứ hai có thể được thực hiện trên vùng có kích thước 8×8 bao gồm khối 4×4 được biến đổi.

Thông tin chỉ báo việc biến đổi thứ hai có được thực hiện hay không có thể được báo hiệu thông qua luồng bit.

Bộ giải mã có thể thực hiện biến đổi ngược của biến đổi thứ hai (biến đổi ngược thứ hai), và có thể thực hiện biến đổi ngược của biến đổi thứ nhất (biến đổi ngược thứ nhất) trên kết quả của biến đổi ngược. Kết quả của việc thực hiện biến đổi ngược thứ hai và biến đổi ngược thứ nhất, có thể đạt được các tín hiệu còn dư dành cho khôi hiện thời.

Hoạt động lượng tử là để làm giảm năng lượng của khối, và quy trình lượng tử bao gồm quy trình để chia hệ số biến đổi cho giá trị không đổi cụ thể. Giá trị không đổi có thể được suy ra bằng thông số lượng tử, và thông số lượng tử có thể được định nghĩa là giá trị giữa 1 và 63.

Khi bộ lập mã thực hiện biến đổi và lượng tử, thì bộ giải mã có thể đạt được khối còn dư thông qua lượng tử ngược và biến đổi ngược. Bộ giải mã có thể đạt được khối được cấu thành lại dành cho khôi hiện thời bằng cách cộng khối dự báo và khối còn dư.

Khi đạt được khối được cấu thành lại của khối hiện thời, thì tổn hao của thông tin xảy ra trong quy trình lượng tử và lập mã có thể được làm giảm xuống thông qua hoạt động lọc vòng trong. Bộ lọc vòng trong có thể bao gồm ít nhất một trong số bộ lọc giải khối, bộ lọc độ lệch thích nghi mẫu (sample adaptive offset filter, SAO), và bộ lọc vòng thích nghi (ALF).

Việc áp dụng các phương án được mô tả ở trên tập trung vào quy trình giải mã hoặc quy trình lập mã cho quy trình lập mã hoặc quy trình giải mã đều nằm trong phạm vi của sáng chế. Việc thay đổi các phương án được mô tả theo thứ bậc định trước theo thứ bậc khác với thứ bậc được mô tả cũng nằm trong phạm vi của sáng chế.

Mặc dù các phương án ở trên đã được mô tả dựa trên một loạt các bước và lưu đồ, nhưng sáng chế không bị hạn chế bởi thứ tự thời gian của các bước và lưu đồ của sáng chế, và có thể được thực hiện đồng thời hoặc theo thứ tự khác nhau nếu cần. Ngoài ra, mỗi thành phần (ví dụ, các đơn vị, môđun, v.v.) cấu thành sơ đồ khối theo các phương án được mô tả ở trên có thể được thực hiện như thiết bị phần cứng hoặc phần mềm, hoặc nhiều thành phần có thể được kết hợp để được thực hiện như một thiết bị phần cứng hoặc phần mềm. Các phương án được mô tả ở trên có thể được thực hiện dưới dạng các lệnh chương trình mà có

thể được thực thi thông qua các thành phần máy tính và được ghi trong vật ghi có thể đọc được bằng máy tính. Vật ghi có thể đọc được bằng máy tính có thể bao gồm các lệnh chương trình, tập tin dữ liệu, các cấu trúc dữ liệu hoặc dạng tương tự dưới dạng độc lập hoặc kết hợp. Vật ghi có thể đọc được bằng máy tính bao gồm, ví dụ, vật từ tính chẳng hạn như đĩa cứng, đĩa mềm và băng từ, vật ghi quang tính chẳng hạn như bộ nhớ chỉ đọc dạng đĩa compac (compact disc read-only memory, CD-ROM) và đĩa đa năng kỹ thuật số (digital versatile disc, DVD), vật ghi quang từ chẳng hạn như đĩa mềm, và các thiết bị phần cứng được tạo cấu hình đặc biệt để lưu và thực thi các lệnh chương trình, chẳng hạn như bộ nhớ chỉ đọc (read only memory, ROM), bộ nhớ truy nhập ngẫu nhiên (random access memory, RAM), bộ nhớ dạng flash hoặc vật ghi tương tự. Các thiết bị phần cứng được mô tả ở trên có thể được tạo cấu hình để vận hành sử dụng một hoặc nhiều môđun phần mềm để thực hiện quy trình của sáng chế, và ngược lại.

Sáng chế có thể được áp dụng cho thiết bị điện tử mà lập mã và giải mã video.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã video bao gồm các bước:

xác định xem phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không;

tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời;

định rõ ứng cử hợp nhất dành cho khối hiện thời dựa trên danh sách ứng cử hợp nhất;

và

suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất, trong đó khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất,

khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm là số nhiều, thì ứng cử hợp nhất của khối hiện thời được chọn lựa dựa trên thông tin chỉ số được giải mã từ luồng bit chỉ báo một ứng cử bất kỳ trong số các ứng cử hợp nhất, và

khi số lượng tối đa là 1, thì ứng cử hợp nhất được xác định mà không giải mã thông tin chỉ số;

độ lớn của vector độ lệch được xác định dựa trên thông tin chỉ số thứ nhất định rõ một ứng cử trong số các ứng cử độ lớn chuyển động, trong đó ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo giá trị của cờ chỉ báo các giá trị bằng số của các ứng cử độ lớn chuyển động, hoặc ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo độ chính xác vector chuyển động dành cho khối hiện thời.

2. Phương pháp theo điểm 1, trong đó cờ được báo hiệu ở mức hình.

3. Phương pháp theo điểm 1, trong đó độ lớn của vector độ lệch được nhận bằng cách áp dụng toán tử dịch cho giá trị được chỉ báo bởi ứng cử độ lớn chuyển động được định rõ bởi thông tin chỉ số thứ nhất.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ điểm 1 đến 3, trong đó chiều của vector độ lệch được xác định dựa trên thông tin chỉ số thứ hai định rõ một ứng cử trong số

các ứng cử chiều vector.

5. Phương pháp lập mã video bao gồm các bước:

xác định xem phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không;

tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời;

định rõ ứng cử hợp nhất dành cho khối hiện thời dựa trên danh sách ứng cử hợp nhất;

và

suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất, trong đó khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất,

khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm là số nhiều, thì thông tin chỉ số chỉ báo ứng cử hợp nhất của khối hiện thời trong số các ứng cử hợp nhất được lập mã, và

khi số lượng tối đa là 1, thì bước lập mã của thông tin chỉ số được bỏ qua;

trong đó phương pháp này còn bao gồm các bước: bước lập mã thông tin chỉ số thứ nhất định rõ ứng cử độ lớn chuyển động chỉ báo độ lớn của vector độ lệch trong số nhiều ứng cử độ lớn chuyển động, và lập mã chỉ báo các giá trị bằng số của các ứng cử độ lớn chuyển động, trong đó ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo giá trị của cờ, hoặc ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo độ chính xác vector chuyển động dành cho khối hiện thời.

6. Phương pháp theo điểm 5, trong đó cờ được lập mã ở mức hình.

7. Phương pháp theo điểm 5, trong đó ứng cử độ lớn chuyển động có giá trị được suy ra bằng cách áp dụng toán tử dịch cho độ lớn của vector độ lệch.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ điểm 5 đến 7, trong đó phương pháp này còn bao gồm bước lập mã thông tin chỉ số thứ hai để định rõ ứng cử chiều vector chỉ báo chiều của vector độ lệch trong số nhiều ứng cử chiều vector.

9. Thiết bị giải mã video bao gồm phần dự báo liên cấu trúc để xác định xem phương pháp

mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không, tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời, định rõ ứng cử hợp nhất dành cho khối hiện thời dựa trên danh sách ứng cử hợp nhất, và suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất, trong đó

khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm là số nhiều, thì ứng cử hợp nhất của khối hiện thời được chọn lựa dựa trên thông tin chỉ số được giải mã từ luồng bit chỉ báo một ứng cử bất kỳ trong số các ứng cử hợp nhất, và khi số lượng tối đa là 1, thì ứng cử hợp nhất được xác định mà không giải mã thông tin chỉ số;

trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để xác định độ lớn của vector độ lệch dựa trên thông tin chỉ số thứ nhất định rõ một ứng cử trong số các ứng cử độ lớn chuyển động, và trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để thiết đặt khác nhau ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động theo giá trị của cờ chỉ báo các giá trị bằng số của các ứng cử độ lớn chuyển động, hoặc thiết đặt khác nhau ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động theo độ chính xác vector chuyển động dành cho khối hiện thời.

10. Thiết bị giải mã video theo điểm 9, trong đó cờ được báo hiệu ở mức hình.

11. Thiết bị giải mã video theo điểm 9, trong đó độ lớn của vector độ lệch được nhận bằng cách áp dụng toán tử dịch cho giá trị được chỉ báo bởi ứng cử độ lớn chuyển động được định rõ bởi thông tin chỉ số thứ nhất.

12. Thiết bị giải mã video theo điểm bất kỳ trong số các điểm từ điểm 9 đến 11, trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để xác định chiều của vector độ lệch dựa trên thông tin chỉ số thứ hai định rõ một ứng cử trong số các ứng cử chiều vector.

13. Thiết bị lập mã video bao gồm phần dự báo liên cấu trúc để xác định xem phương pháp mã hóa độ chênh lệch chuyển động hợp nhất có được áp dụng cho khối hiện thời hay không, tạo ra danh sách ứng cử hợp nhất dành cho khối hiện thời, định rõ ứng cử hợp nhất dành

cho khối hiện thời dựa trên danh sách ứng cử hợp nhất, và suy ra vector chuyển động dành cho khối hiện thời dựa trên ứng cử hợp nhất, trong đó

khi phương pháp mã hóa độ chênh lệch chuyển động hợp nhất được áp dụng cho khối hiện thời, thì vector chuyển động của khối hiện thời được suy ra bằng cách cộng vector độ lệch vào vector chuyển động của ứng cử hợp nhất, khi số lượng tối đa của các ứng cử hợp nhất mà danh sách ứng cử hợp nhất có thể bao gồm là số nhiều, thì thông tin chỉ số chỉ báo ứng cử hợp nhất của khối hiện thời trong số các ứng cử hợp nhất được lập mã, và khi số lượng tối đa là 1, thì bước lập mã của thông tin chỉ số được bỏ qua;

trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để lập mã thông tin chỉ số thứ nhất định rõ ứng cử độ lớn chuyển động chỉ báo độ lớn của vector độ lệch trong số nhiều ứng cử độ lớn chuyển động, trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để lập mã chỉ báo các giá trị bằng số của các ứng cử độ lớn chuyển động, trong đó ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo giá trị của cờ, hoặc ít nhất một giá trị trong số giá trị tối đa và giá trị tối thiểu của các ứng cử độ lớn chuyển động được thiết đặt khác nhau theo độ chính xác vector chuyển động dành cho khối hiện thời.

14. Thiết bị lập mã video theo điểm 13, trong đó cờ được lập mã ở mức hình.

15. Thiết bị lập mã video theo điểm 13, trong đó ứng cử độ lớn chuyển động có giá trị được suy ra bằng cách áp dụng toán tử dịch cho độ lớn của vector độ lệch.

16. Thiết bị lập mã video theo điểm bất kỳ trong số các điểm từ điểm 13 đến 15, trong đó phần dự báo liên cấu trúc còn được tạo cấu hình để lập mã thông tin chỉ số thứ hai định rõ ứng cử chiều vector chỉ báo chiều của vector độ lệch trong số nhiều ứng cử chiều vector.

17. Bộ giải mã video bao gồm:

ít nhất một bộ xử lý; và

bộ nhớ lưu các chương trình máy tính mà, khi được thực thi bởi ít nhất một bộ xử lý, thì làm cho ít nhất một bộ xử lý này thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 1 đến 4.

18. Bộ lập mã video bao gồm:

ít nhất một bộ xử lý; và

bộ nhớ lưu các chương trình máy tính mà, khi được thực thi bởi ít nhất một bộ xử lý, thì làm cho ít nhất một bộ xử lý này thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 5 đến 8.

19. Vật ghi có thể đọc được bằng máy tính bao gồm các lệnh mà, khi được thực thi bởi ít nhất một bộ xử lý, thì làm cho ít nhất một bộ xử lý thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 1 đến 4.

20. Vật ghi có thể đọc được bằng máy tính bao gồm các lệnh mà, khi được thực thi bởi ít nhất một bộ xử lý, thì làm cho ít nhất một bộ xử lý thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 5 đến 8.

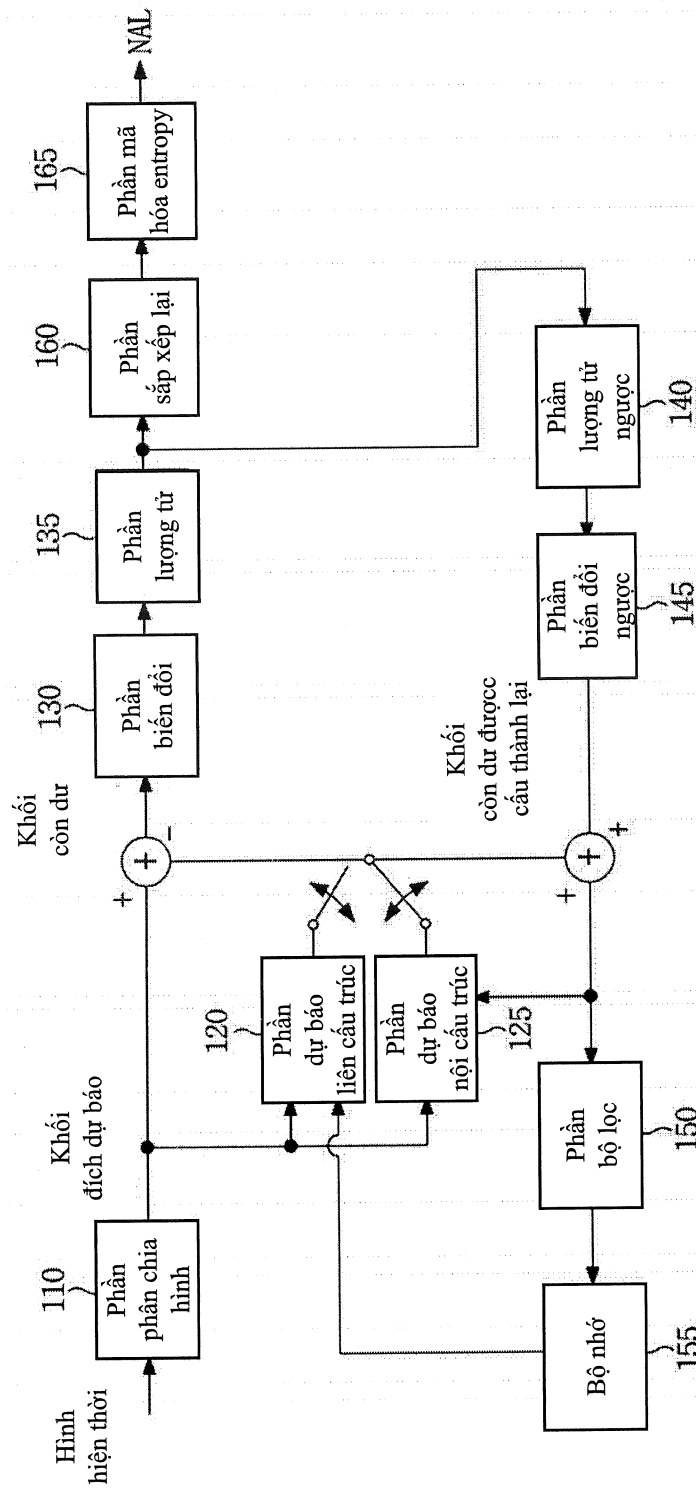


Fig.1

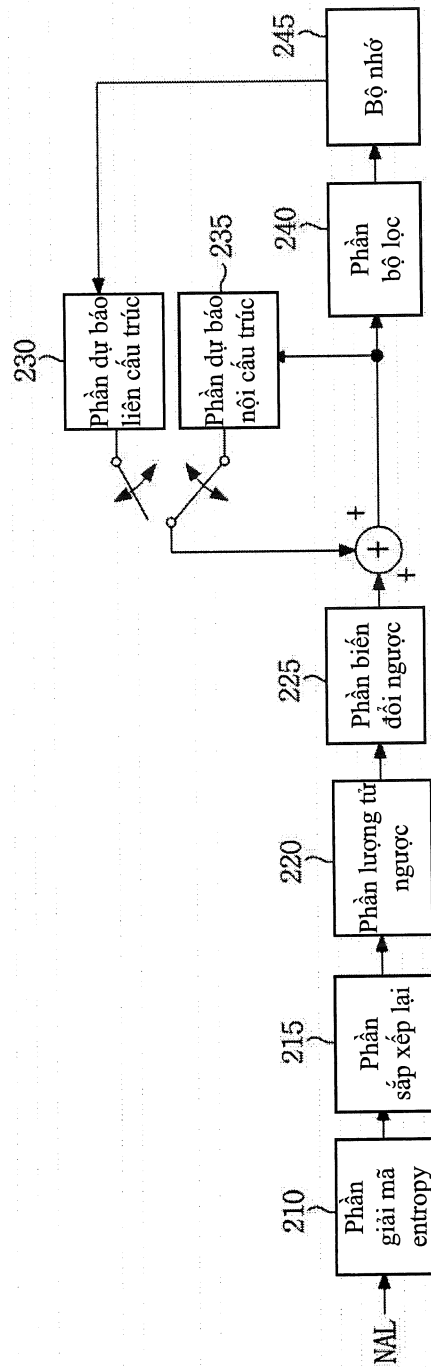


Fig.2

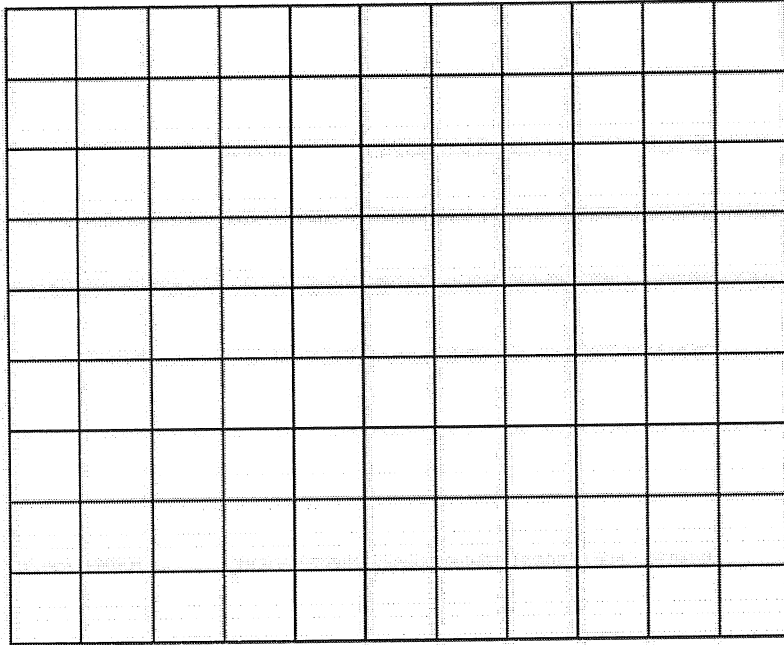


Fig.3

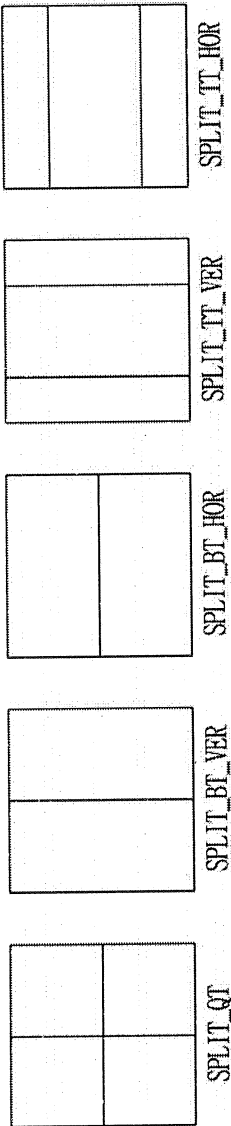


Fig.4

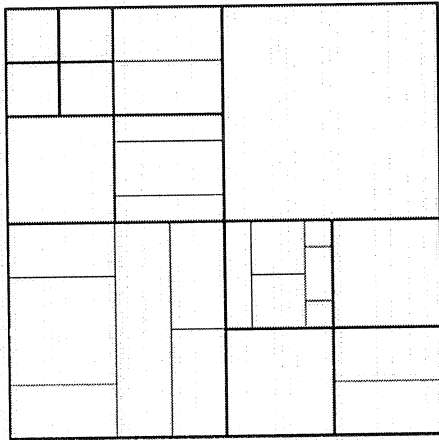


Fig.5

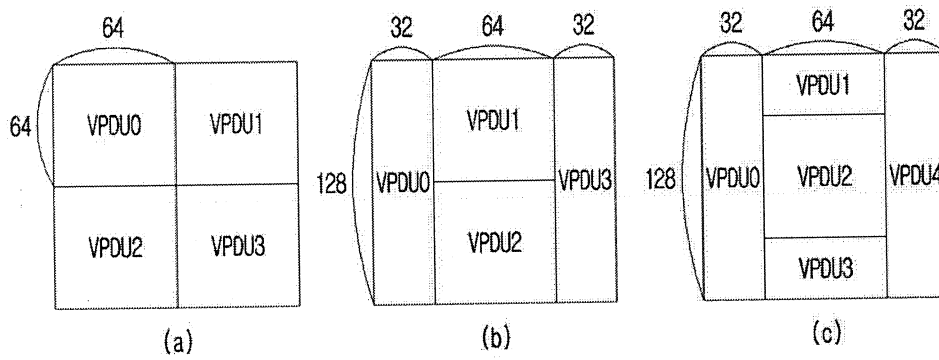


Fig.6

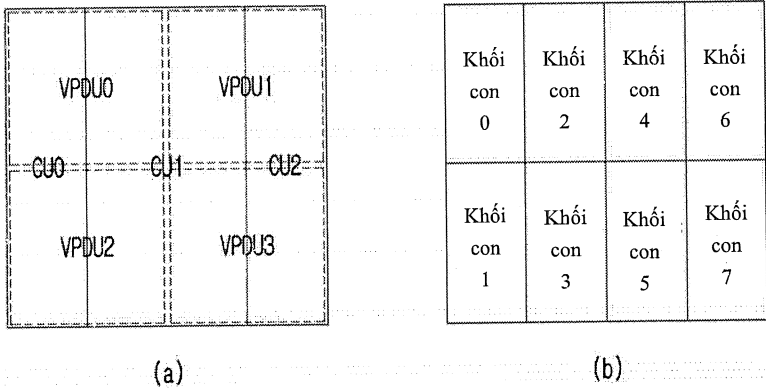


Fig.7

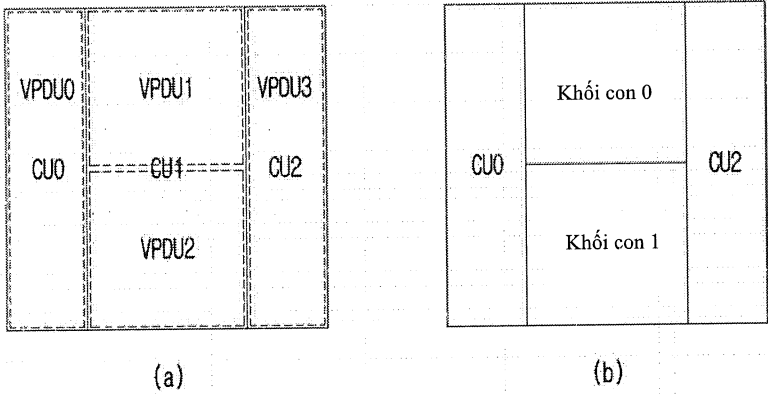


Fig.8

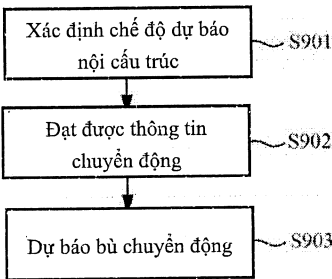


Fig.9

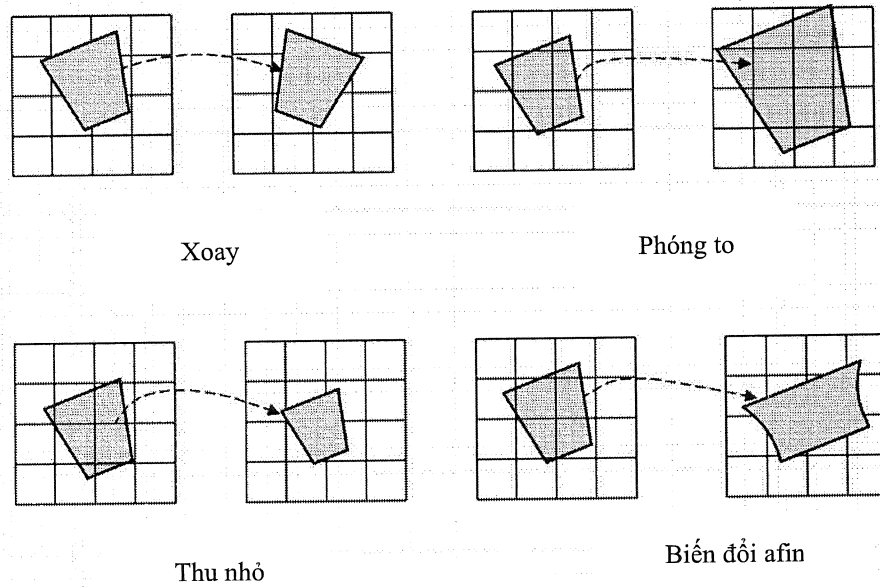


Fig.10

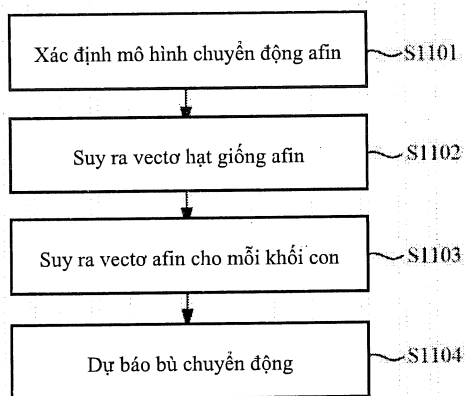


Fig.11

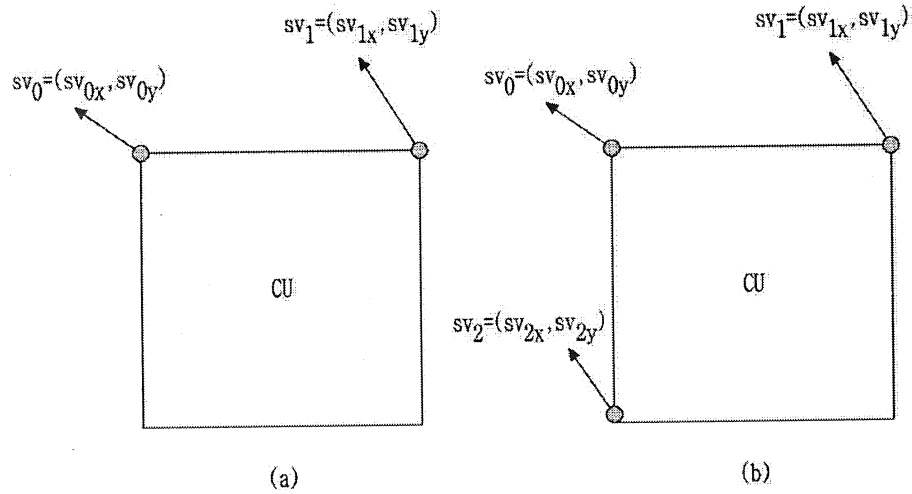


Fig.12

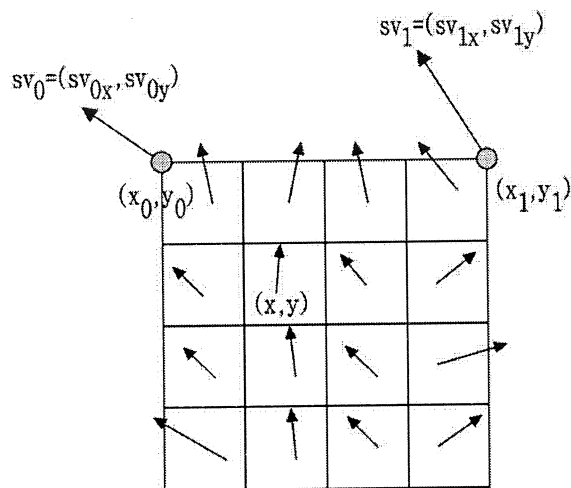


Fig.13

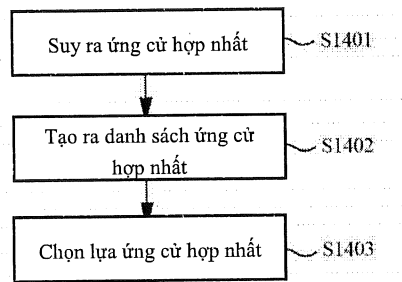


Fig.14

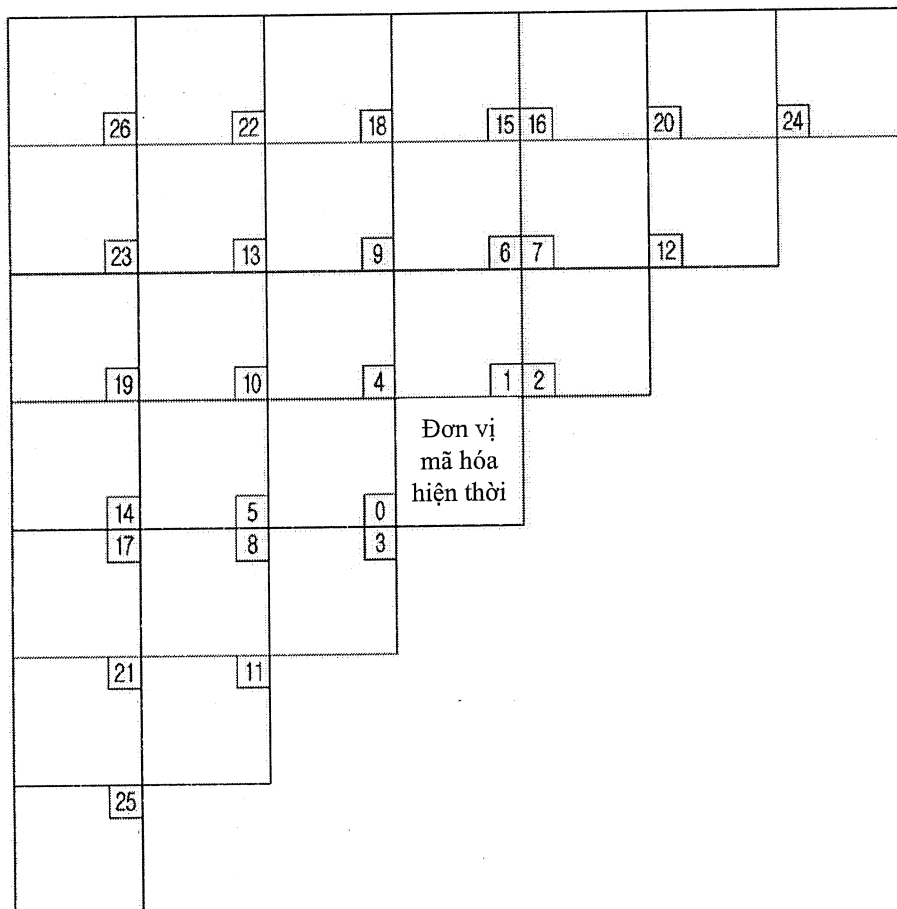


Fig.15

10/29

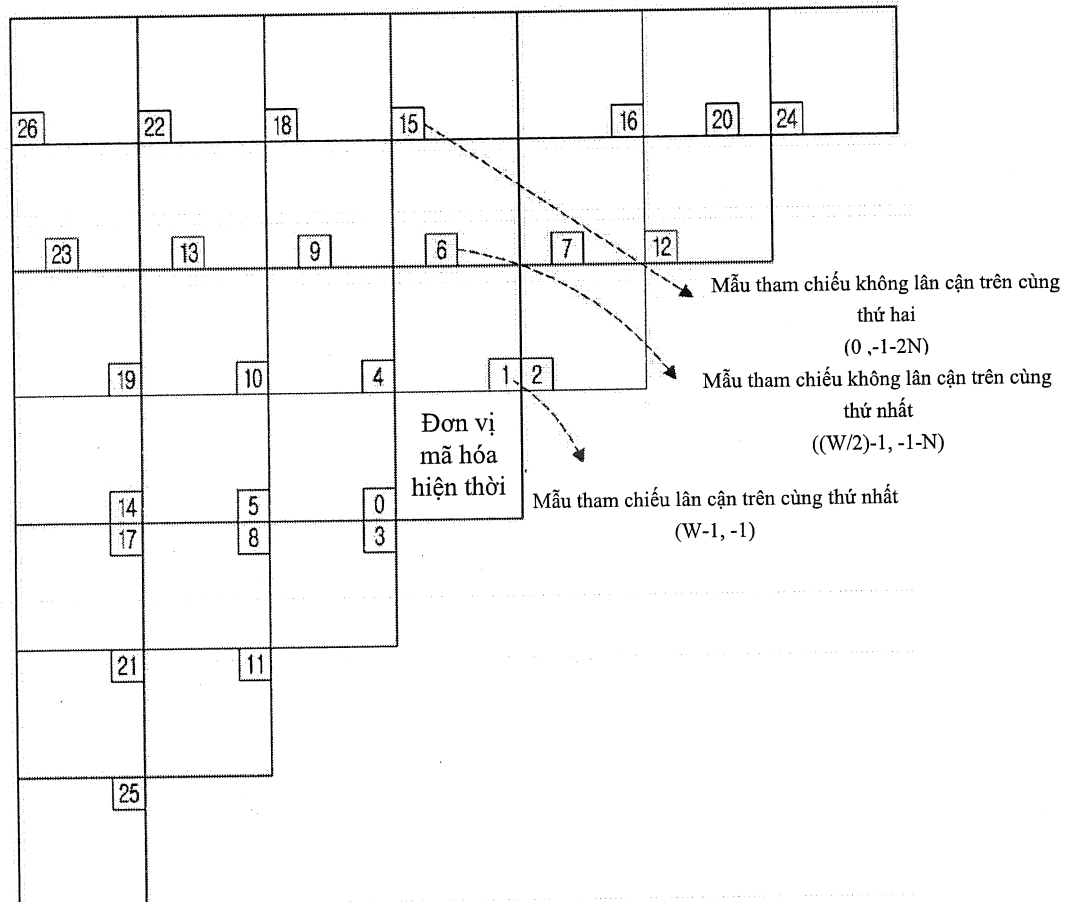


Fig.16

11/29

28	22	16	13	9	6	7	12
29	23	18	10	4	1	2	
21	15	14	5	0	Đơn vị mã hóa hiện thời		
30	24	17	8	3			
31	25	19	11				
32	26	20					
33	27						
34							

Fig.17

12/29

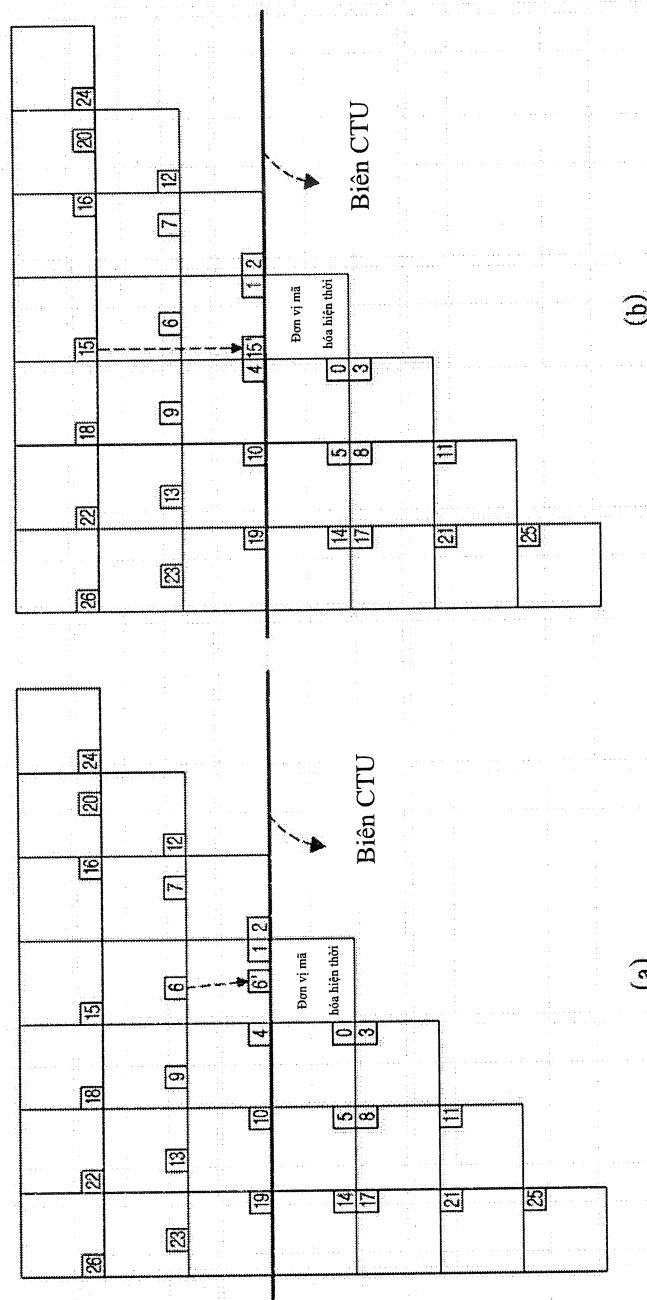


Fig.18

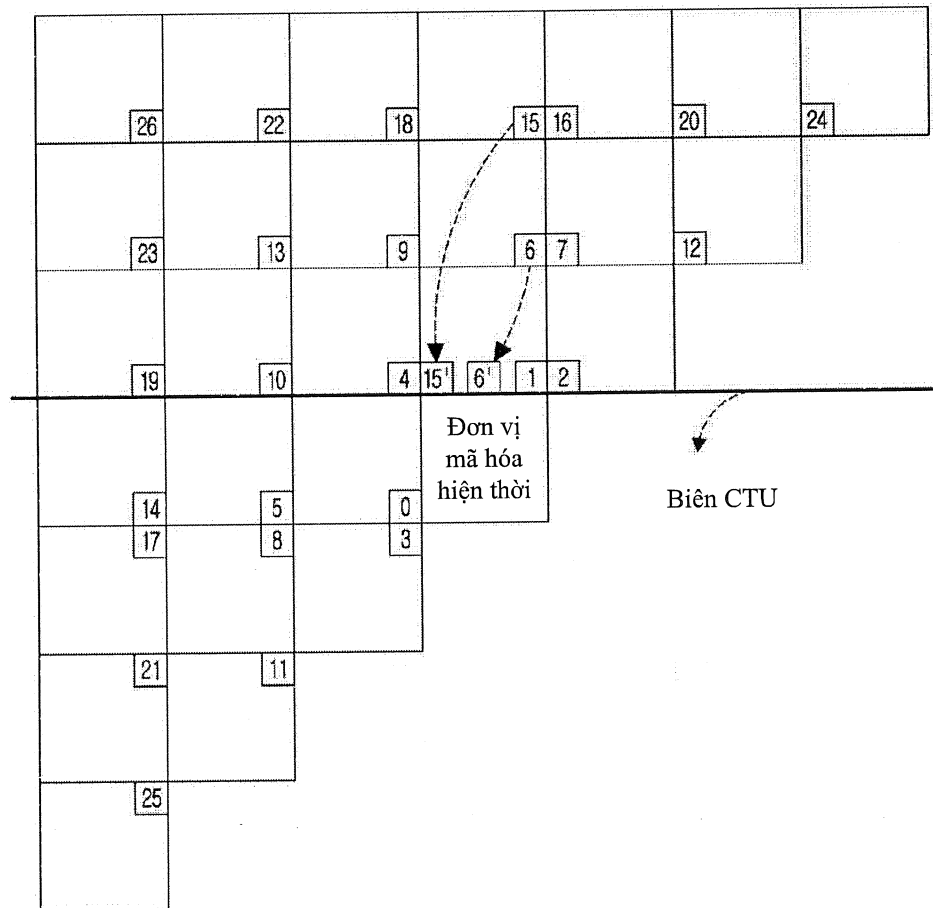


Fig.19

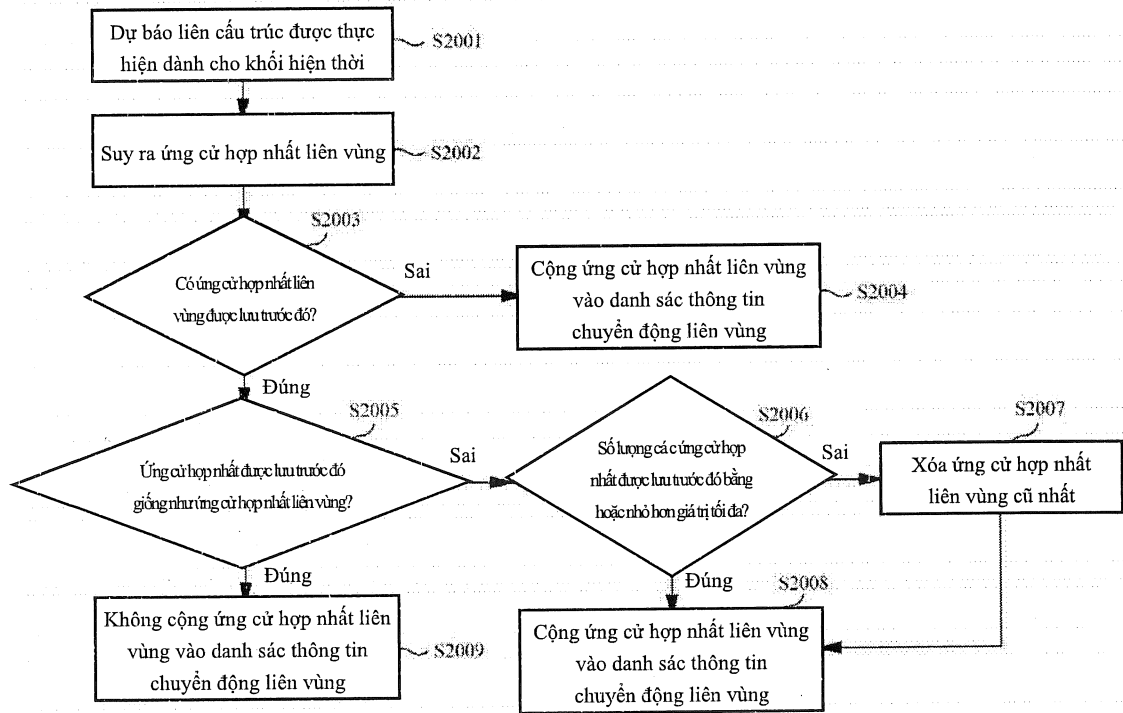


Fig.20

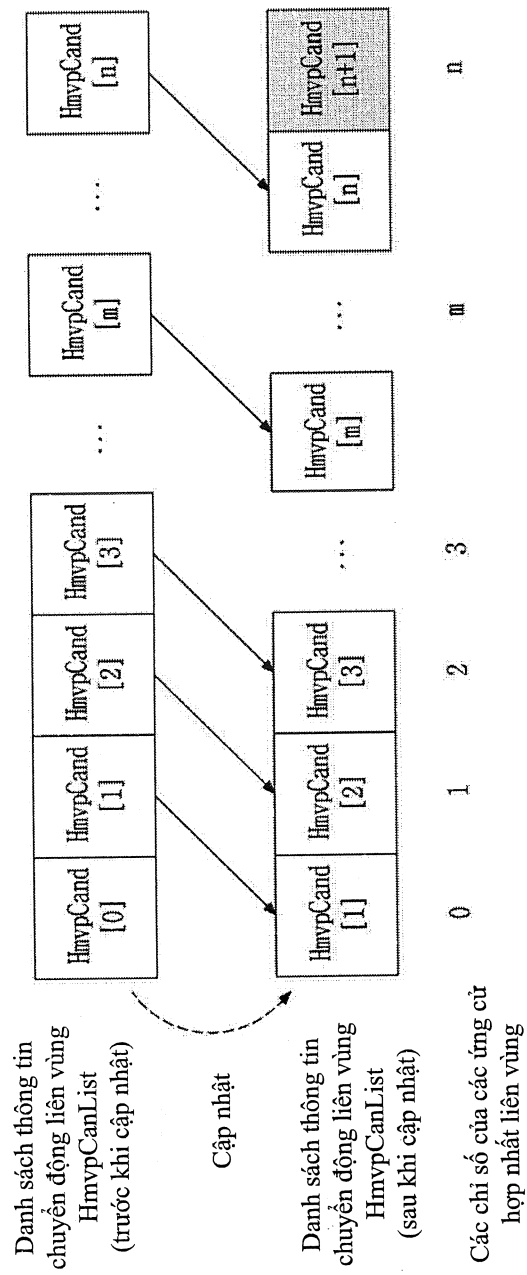


Fig.21

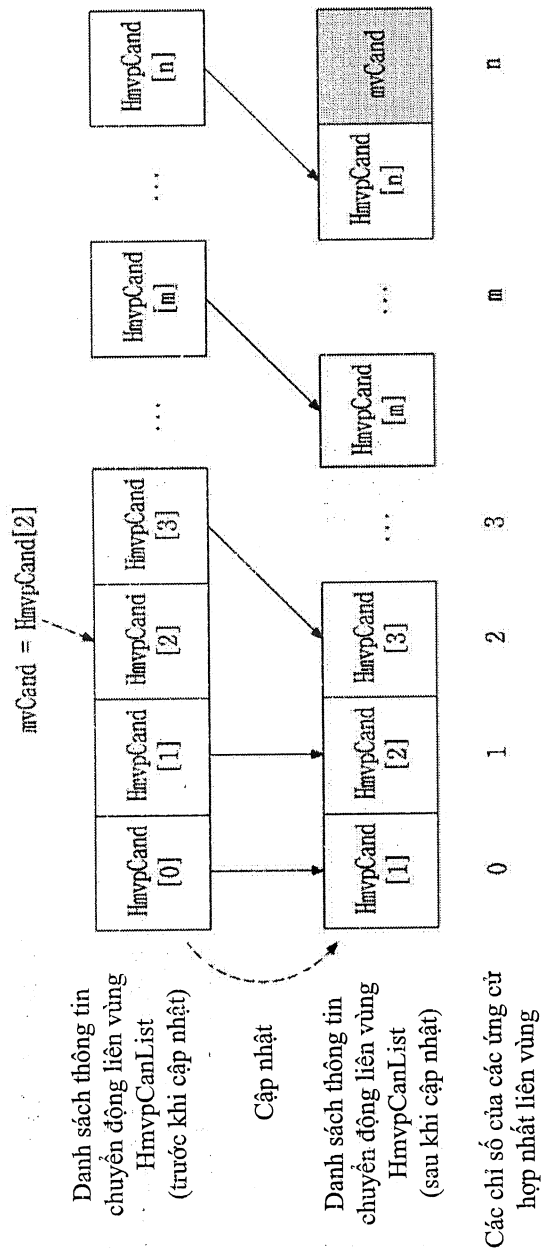
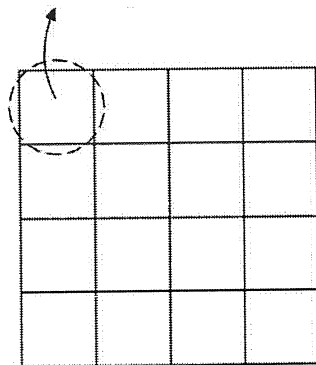


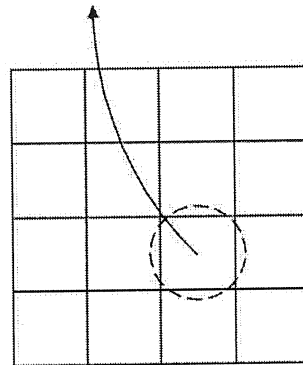
Fig.22

Khối đại diện



(a)

Khối đại diện



(b)

Fig.23

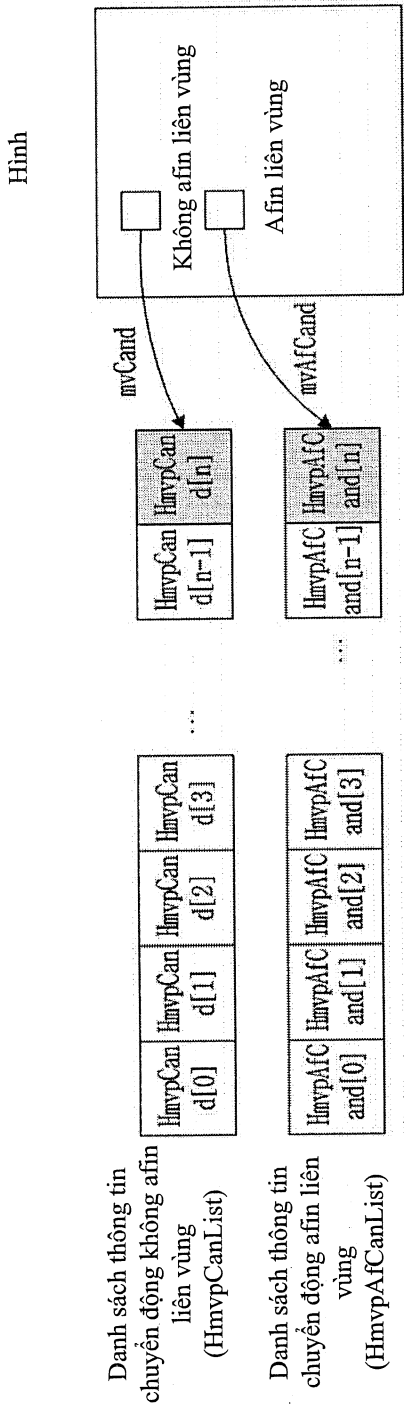


Fig.24

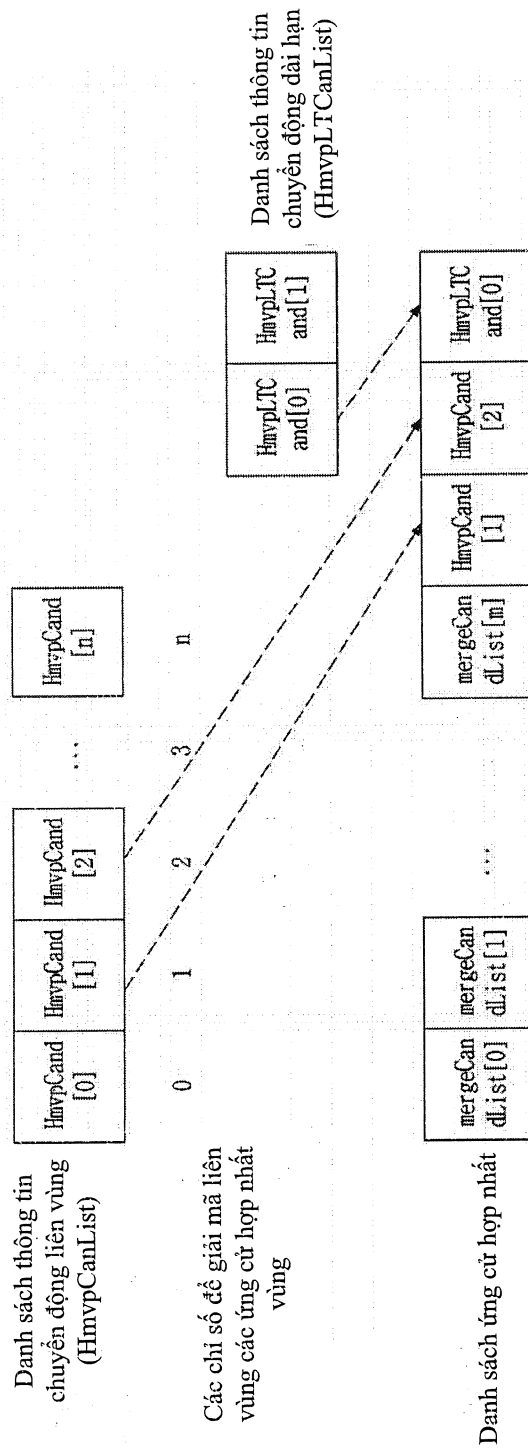


Fig.25

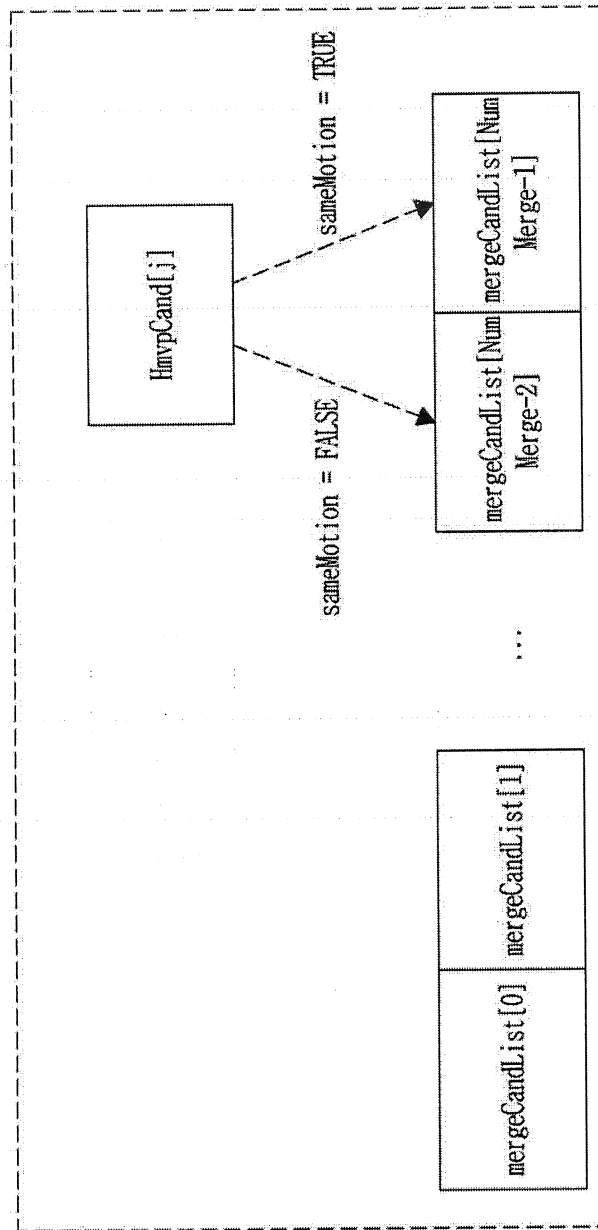


Fig.26

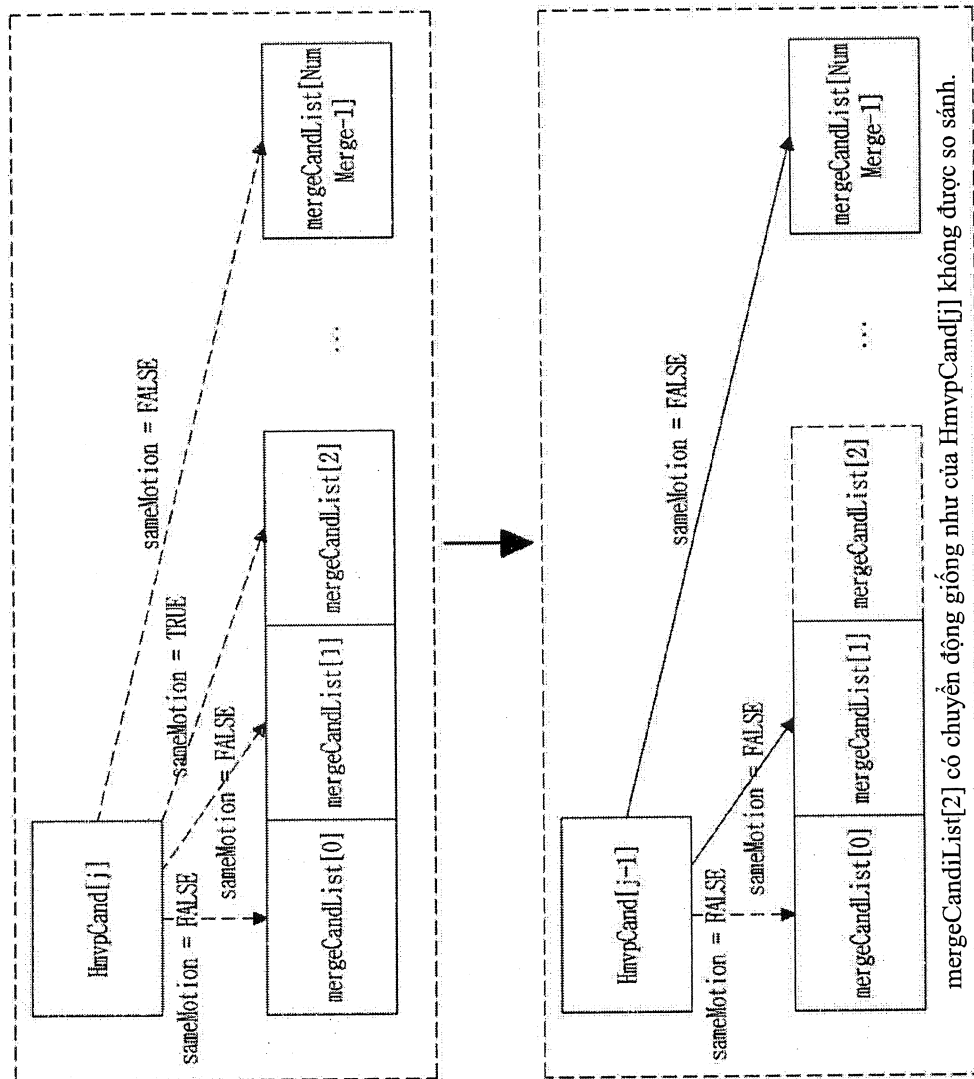


Fig.27

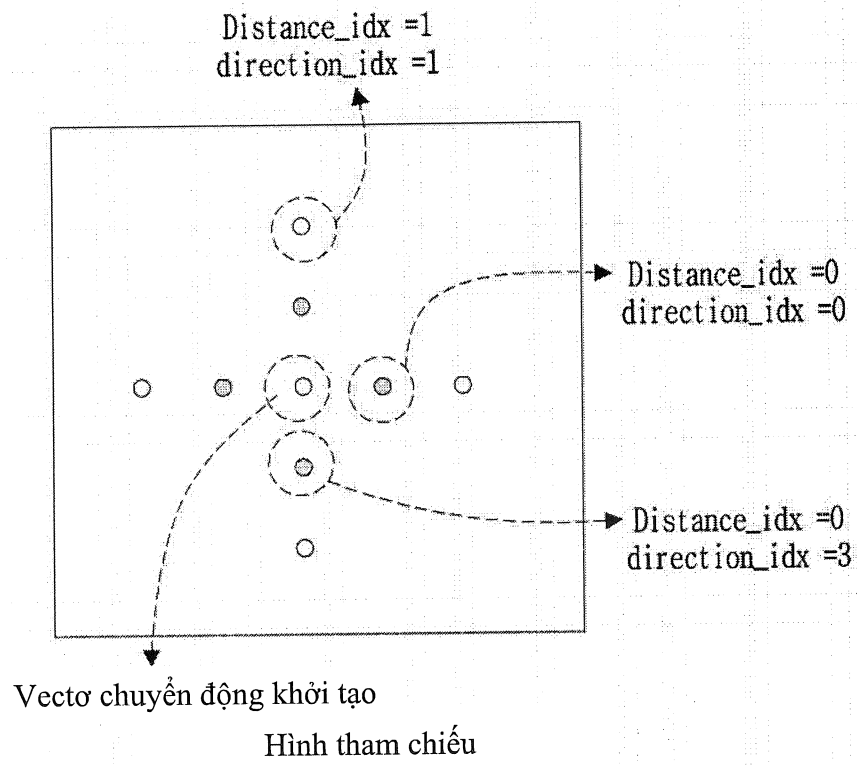


Fig.28

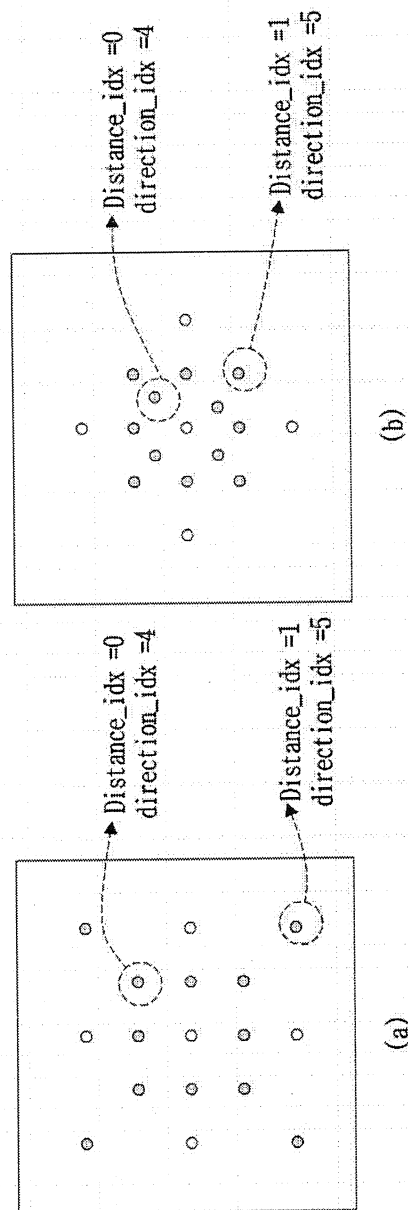


Fig.29

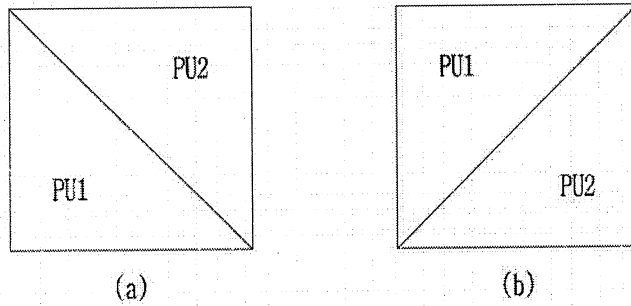


Fig.30

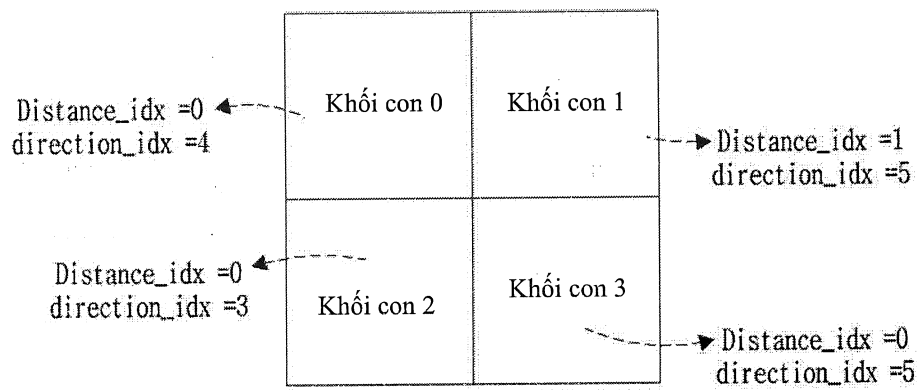


Fig.31

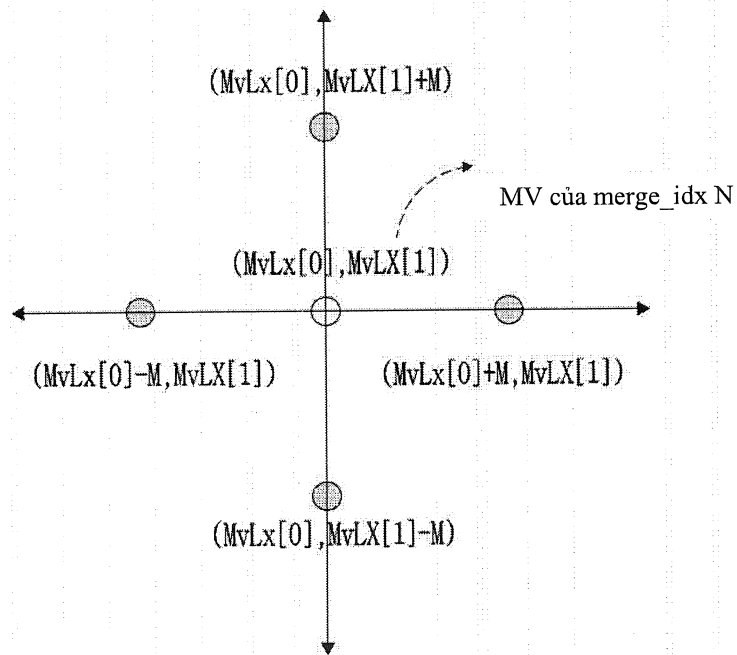


Fig.32

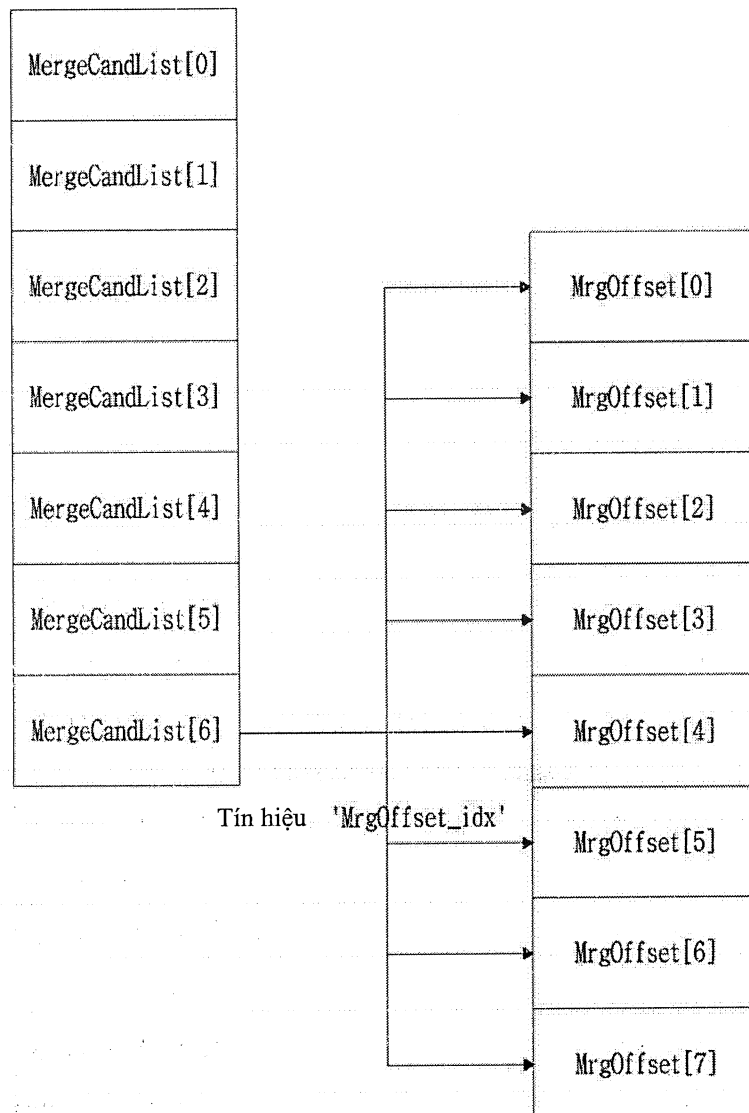


Fig.33

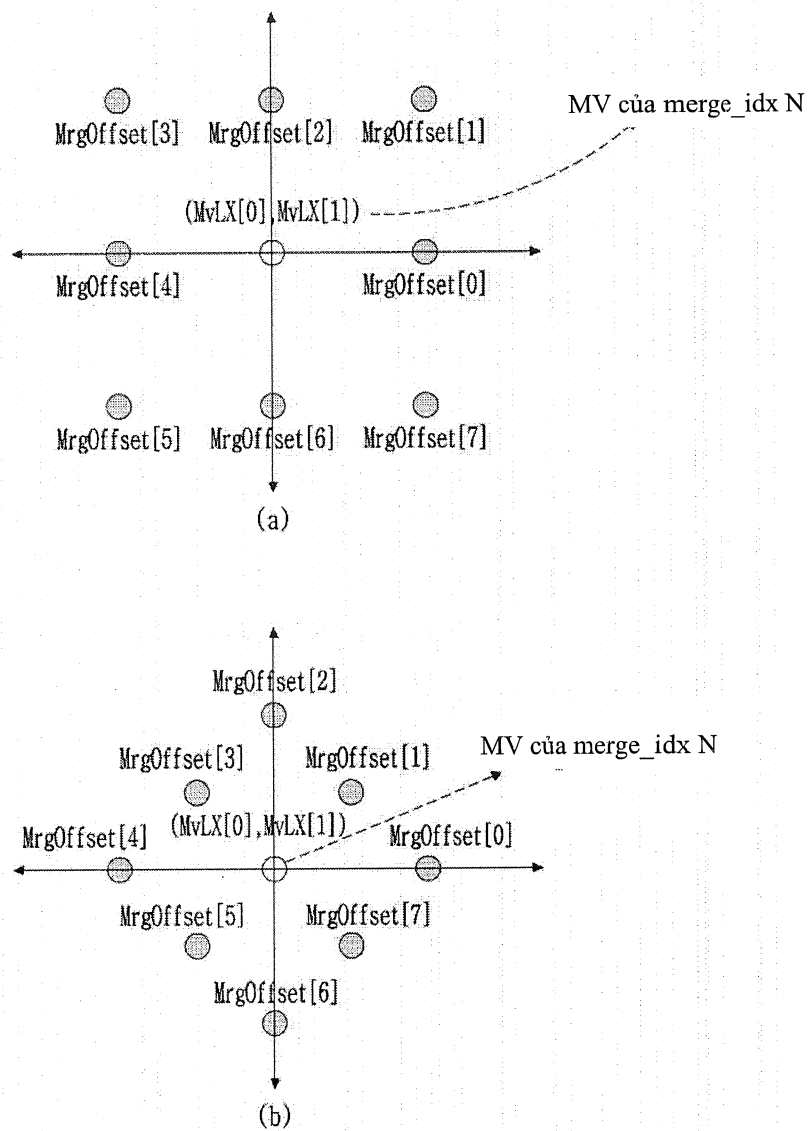


Fig.34

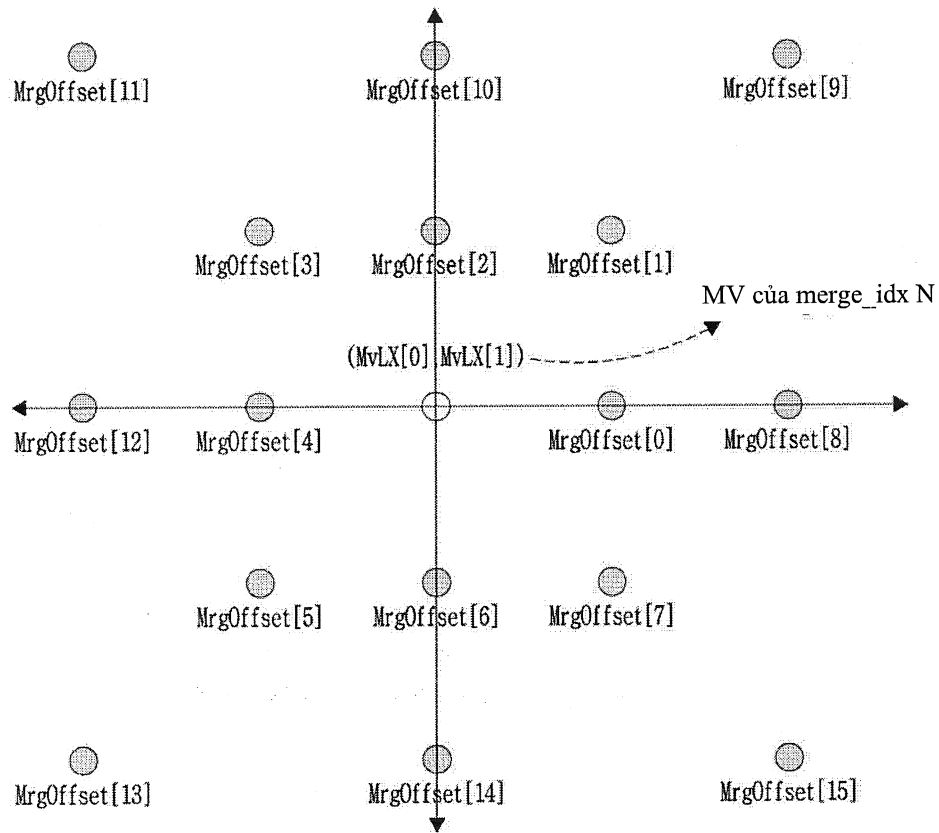


Fig.35

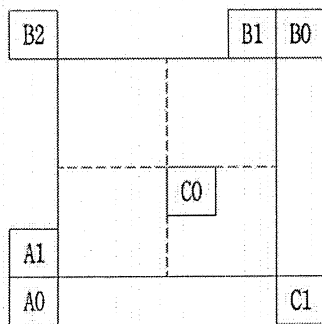


Fig.36

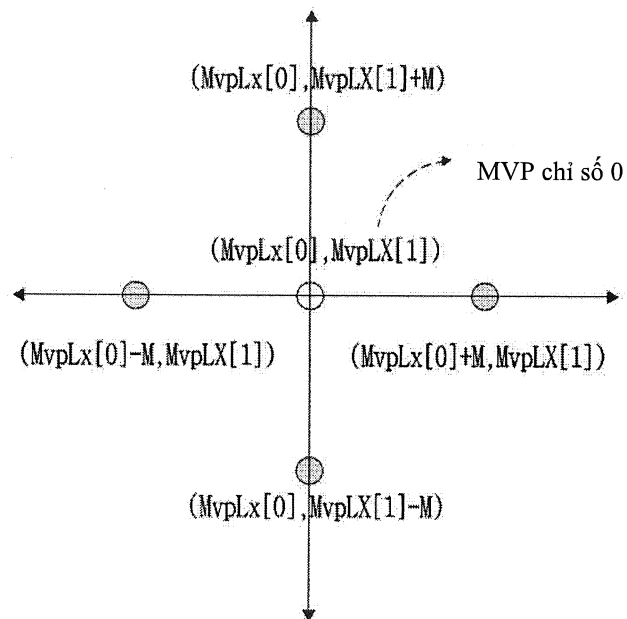


Fig.37

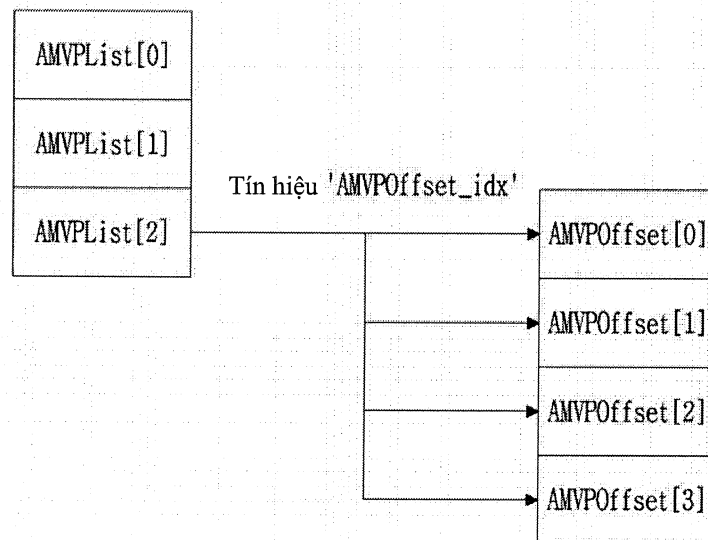


Fig.38