



(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)



1-0045122

(51)^{2020.01} H04N 19/11; H04N 19/132; H04N
19/176; H04N 19/86; H04N 19/513;
H04N 19/593; H04N 19/70; H04N
19/119; H04N 19/186

(13) B

(21) 1-2021-01904

(22) 20/09/2019

(86) PCT/KR2019/012290 20/09/2019

(87) WO2020/060327 A1 26/03/2020

(30) 10-2018-0114342 21/09/2018 KR; 10-2018-0114344 21/09/2018 KR; 10-2018-0114343 21/09/2018 KR

(45) 25/04/2025 445

(43) 25/06/2021 399A

(71) 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 Dịch vụ Sở hữu trí tuệ KENFOX (KENFOX IP SERVICE CO.,LTD.)

(54) PHƯƠNG PHÁP GIẢI MÃ VIDEO, PHƯƠNG PHÁP MÃ HÓA VIDEO, BỘ GIẢI MÃ VIDEO, BỘ MÃ HÓA VIDEO VÀ PHƯƠNG TIỆN ĐỌC ĐƯỢC BẰNG MÁY TÍNH KHÔNG TẠM THỜI

(21) 1-2021-01904

(57) Sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video, bộ giải mã video, bộ mã hóa video và phương tiện đọc được bằng máy tính không tạm thời, trong đó phương pháp giải mã video theo sáng chế bao gồm các bước là: tạo ra danh mục ứng viên hợp nhất trong khối hiện tại; định rõ một trong số nhiều ứng viên hợp nhất được đưa vào danh mục ứng viên hợp nhất; suy ra vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được định rõ; suy ra vector affin đối với khối con trong khối hiện tại, sử dụng vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại; và thực hiện dự báo bù chuyển động cho khối con dựa trên vector affin.

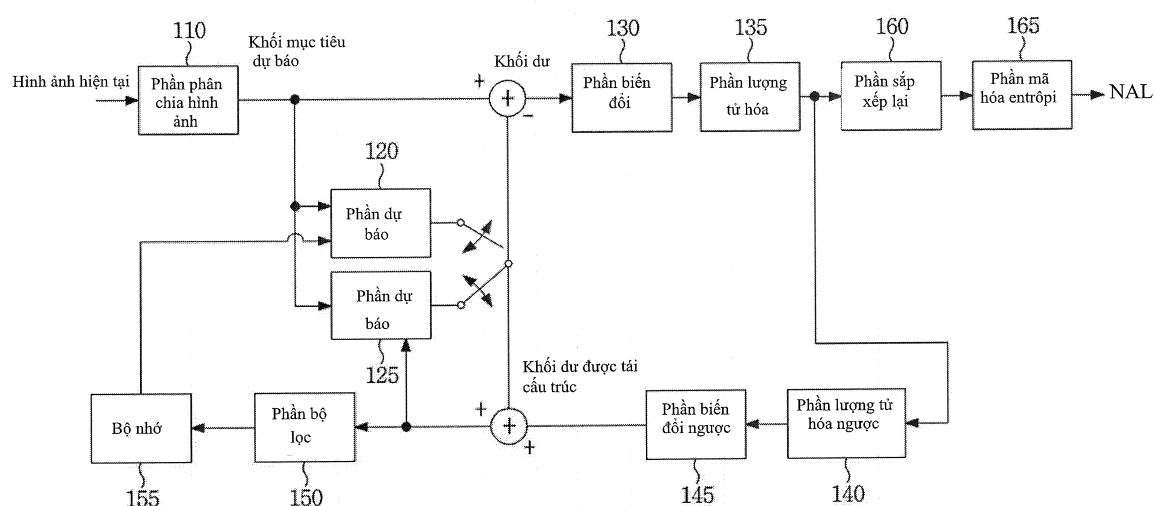


Fig.1

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video, bộ giải mã video, bộ mã hóa video và phương tiện đọc được bằng máy tính không tạm thời.

Tình trạng kỹ thuật của sáng chế

Do các bảng hiển thị ngày càng trở nên lớn hơn, các dịch vụ video có chất lượng cao hơn được yêu cầu ngày càng nhiều hơn. Vấn đề lớn nhất của các dịch vụ video độ nét cao là việc tăng đáng kể khối lượng dữ liệu, và để giải quyết vấn đề này, các cuộc nghiên cứu để cải thiện tốc độ nén video được tích cực thực hiện. Theo một ví dụ đại diện, nhóm các chuyên gia hình ảnh chuyển động (Motion Picture Experts Group - MPEG) và nhóm các chuyên gia mã hóa video (Video Coding Experts Group - VCEG) thuộc ban tiêu chuẩn hoá viễn thông của liên minh viễn thông quốc tế (International Telecommunication Union-Telecommunication - ITU-T) đã thành lập nhóm hợp tác chung về mã hóa video (Joint Collaborative Team on Video Coding - JCT-VC) vào năm 2009. JCT-VC đã đề xuất mã hóa video hiệu suất cao (High Efficiency Video Coding - HEVC), đây là tiêu chuẩn nén video có hiệu năng nén cao gấp khoảng hai lần hiệu năng nén của H.264/AVC, và tiêu chuẩn này được phê chuẩn vào ngày 25/01/2013. Với sự tiến bộ nhanh chóng trong các dịch vụ video độ nét cao, hiệu năng của HEVC dần bộc lộ các nhược điểm của tiêu chuẩn này.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất phương pháp dự báo liên khung sử dụng mô hình affin, và thiết bị dùng để thực hiện phương pháp này, trong việc mã hóa/giải mã tín hiệu video.

Một mục đích khác của sáng chế là đề xuất phương pháp suy ra vector hạt affin sử dụng vector chuyển động tịnh tiến của khối con, và thiết bị dùng để thực hiện phương pháp, trong việc mã hóa/giải mã tín hiệu video.

Một mục đích khác của sáng chế là đề xuất phương pháp suy ra vector hạt affin bằng

cách biến đổi khoảng cách giữa khối lân cận và khối hiện tại thành chuỗi lũy thừa của 2, và thiết bị dùng để thực hiện phương pháp, trong việc mã hóa/giải mã tín hiệu video.

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

Phương pháp giải mã và mã hóa tín hiệu video theo sáng chế gồm các bước là: tạo ra danh mục ứng viên hợp nhất đối với khối hiện tại; định rõ một trong số nhiều ứng viên hợp nhất được đưa vào danh mục ứng viên hợp nhất; suy ra vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được định rõ; suy ra vector affin đối với khối con trong khối hiện tại bằng cách sử dụng vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại; và thực hiện dự báo bù chuyển động cho khối con dựa trên vector affin. Tại điểm này, khối con là vùng có kích cỡ nhỏ hơn kích cỡ của khối hiện tại. Ngoài ra, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất có thể được suy ra dựa trên thông tin chuyển động của khối lân cận gần kề khối hiện tại.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, khi khối lân cận được đưa vào đơn vị cây mã hóa khác với đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất có thể được suy ra dựa trên các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải của khối lân cận.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, khối con dưới cùng-bên trái có thể gồm mẫu tham chiếu dưới cùng-bên trái được bố trí tại góc dưới cùng-bên trái của khối lân cận, và khối con dưới cùng-bên phải có thể gồm mẫu tham chiếu dưới cùng-bên phải được bố trí tại góc dưới cùng-bên trái của khối lân cận.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất có thể được suy ra dựa trên trị số thu được dựa trên phép toán dịch chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất có thể được suy ra dựa trên trị số thu được bằng cách thêm khoảng cách ngang giữa mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải và độ lệch.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, vector hạt affin

thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất có thể được suy ra dựa trên trị số thu được dựa trên phép toán dịch chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất có thể được suy ra dựa trên khoảng cách giữa mẫu lân cận gần kề mặt phải của mẫu tham chiếu dưới cùng-bên phải và mẫu tham chiếu dưới cùng-bên trái.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, danh mục ứng viên hợp nhất gồm ứng viên hợp nhất thứ nhất được suy ra dựa trên khối lân cận đỉnh được xác định là khối sẵn có thứ nhất trong số các khối lân cận đỉnh được bố trí trên đỉnh của khối hiện tại, và ứng viên hợp nhất thứ hai được suy ra dựa trên khối lân cận trái được xác định là khối sẵn có thứ nhất trong số các khối lân cận trái được bố trí trên mặt trái của khối hiện tại.

Theo phương pháp giải mã và mã hóa tín hiệu video theo sáng chế, khi khối lân cận được đưa vào đơn vị cây mã hóa giống như đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất có thể được suy ra dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của khối lân cận.

Các đặc tính được tóm lược trên đây liên quan đến sáng chế chỉ là các khía cạnh dùng làm ví dụ của phần mô tả chi tiết của sáng chế sẽ được mô tả dưới đây, và không giới hạn phạm vi của sáng chế.

Theo sáng chế, có tác dụng cải thiện hiệu suất dự báo thông qua phương pháp dự báo liên khung sử dụng mô hình affin.

Theo sáng chế, có tác dụng cải thiện hiệu suất mã hóa bằng cách suy ra vector hạt affin sử dụng vector chuyển động tịnh tiến của khối con.

Theo sáng chế, có tác dụng cải thiện hiệu suất mã hóa bằng cách suy ra vector hạt affin bằng cách biến đổi khoảng cách giữa khối lân cận và khối hiện tại thành chuỗi lũy thừa của 2.

Các tác dụng có thể thu được từ sáng chế không bị giới hạn ở các tác dụng được đề cập trên đây, và các tác dụng khác không được đề cập có thể được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng từ phần mô tả sau đây.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối thể hiện bộ mã hóa video theo phương án của sáng chế.

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

Fig.3 là hình vẽ thể hiện đơn vị cây mã hóa cơ bản theo 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 hình phân chia của đơn vị cây mã hóa.

Fig.6 là lưu đồ minh họa phương pháp dự báo liên khung theo phương án của sáng chế.

Fig.7 là hình vẽ thể hiện chuyển động phi tuyến tính của đối tượng.

Fig.8 là lưu đồ minh họa phương pháp dự báo liên khung dựa trên chuyển động affin theo phương án của sáng chế.

Fig.9 là hình vẽ thể hiện ví dụ về các vectơ hạt affin của mỗi mô hình chuyển động affin.

Fig.10 là hình vẽ thể hiện ví dụ về các vectơ affin của các khối con trong mô hình chuyển động 4 thông số.

Fig.11 là hình vẽ thể hiện khối lân cận có thể được sử dụng để suy ra ứng viên hợp nhất.

Fig.12 là hình vẽ thể hiện ví dụ về suy ra vectơ hạt affin của khối hiện tại dựa trên vectơ hạt affin của khối lân cận affin.

Fig.13 là hình vẽ thể hiện ví dụ về việc đặt vectơ chuyển động của khối con là vectơ hạt affin của khối lân cận affin.

Fig.14 đến Fig.16 là các hình vẽ thể hiện các vị trí của các mẫu tham chiếu.

Fig.17 là hình vẽ thể hiện ví dụ về việc ứng dụng phương pháp suy ra vectơ hợp nhất affin biến đổi.

Fig.18 là hình vẽ thể hiện ví dụ về suy ra vectơ hạt affin của khối hiện tại dựa trên khối lân cận không phải affin.

Fig.19 là hình vẽ thể hiện ví dụ về thay thế khối lân cận bằng một khối lân cận khác.

Fig.20 và Fig.21 là các hình vẽ thể hiện luồng của phương pháp dự báo bù chuyển động sử dụng nhiều ứng viên hợp nhất.

Fig.22 là lưu đồ minh họa phương pháp dự báo nội khung theo phương án của sáng chế.

Fig.23 là hình vẽ thể hiện các mẫu tham chiếu được đưa vào mỗi đường mẫu tham chiếu.

Fig.24 là hình vẽ thể hiện các phương thức dự báo nội khung.

Fig.25 và Fig.26 là các hình vẽ thể hiện ví dụ về mảng một chiều sắp xếp các mẫu tham chiếu trong một đường.

Fig.27 là hình vẽ thể hiện ví dụ về các góc được tạo ra giữa các phương thức dự báo nội khung góc và đường thẳng song song với trục x.

Fig.28 là hình vẽ thể hiện phương án thu được các mẫu dự báo khi khối hiện tại là hình dạng không phải hình vuông.

Fig.29 là hình vẽ thể hiện các phương thức dự báo nội khung góc rộng.

Fig.30 là lưu đồ minh họa quy trình xác định độ bền khối.

Fig.31 là hình vẽ thể hiện các ứng viên bộ lọc được xác định trước.

Fig.32 là hình vẽ thể hiện ví dụ về phân chia hình ảnh thành nhiều ngói.

Fig.33 là hình vẽ thể hiện mẫu hình phân chia của hình ảnh theo kỹ thuật xếp ngói linh hoạt.

Fig.34 là hình vẽ thể hiện ví dụ về gán ID ngói cho mỗi đơn vị cây mã hóa.

Fig.35 là hình vẽ thể hiện ví dụ về xác định có chọn lọc xem liệu có ứng dụng bộ lọc trong vòng lặp cho mỗi ngói hay không.

Mô tả chi tiết sáng chế

Trong phần dưới đây, phương án của sáng chế sẽ được mô tả chi tiết kết hợp với các hình vẽ kèm theo.

Việc mã hóa và giải mã video được thực hiện bởi đơn vị của khối. Ví dụ như, quy trình mã hóa/giải mã chẳng hạn như biến đổi, lượng tử hóa, dự báo, lọc trong vòng lặp, tái cấu trúc hoặc hoạt động tương tự có thể được thực hiện trên khối mã hóa, khối biến đổi, hoặc khối dự báo.

Trong phần dưới đây, khối cần được mã hóa/giải mã sẽ được gọi là ‘khối hiện tại’. Ví dụ như, khối hiện tại có thể thể hiện khối mã hóa, khối biến đổi hoặc khối dự báo theo bước quy trình mã hóa/giải mã hiện tại.

Ngoài ra, có thể hiểu rằng thuật ngữ ‘đơn vị’ được sử dụng trong bản mô tả này biểu thị đơn vị cơ bản để thực hiện quy trình mã hóa/giải mã cụ thể, và thuật ngữ ‘khối’ biểu thị mảng mẫu có kích cỡ được xác định trước. Trừ khi được trình bày khác đi, ‘khối’ và ‘đơn vị’ có thể được sử dụng để có cùng ý nghĩa. Ví dụ như, theo phương án được mô tả dưới đây, có thể hiểu rằng khối mã hóa và bộ phận mã hóa có cùng ý nghĩa.

Fig.1 là sơ đồ khối thể hiện bộ mã hóa video theo phương án của sáng chế.

Như được thể hiện trên Fig.1, thiết bị mã hóa video 100 có thể gồm phần phân chia hình ảnh 110, phần dự báo 120 và 125, phần biến đổi 130, phần lượng tử hóa 135, phần sắp xếp lại 160, phần mã hóa entropi 165, phần lượng tử hóa ngược 140, phần biến đổi

ngược 145, phần bộ lọc 150, và bộ nhớ 155.

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

Ngoài ra, một vài trong số các thành phần đều không phải là các thành phần thiết yếu thực hiện các chức năng thiết yếu trong 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 gồm các thành phần cần thiết để thực hiện bản chất của sáng chế loại trừ các thành phần được sử dụng để cải thiện hiệu năng, và cấu trúc chỉ gồm các thành phần thiết yếu loại trừ các thành phần tùy chọn được sử dụng để cải thiện hiệu năng cũng được đưa vào phạm vi của sáng chế.

Phân phân chia hình ảnh 110 có thể phân chia hình ảnh đầu vào thành ít nhất một bộ phận xử lý. Tại điểm này, bộ phận xử lý có thể là bộ phận dự báo (Prediction Unit - PU), bộ phận biến đổi (Transform Unit - TU), hoặc bộ phận mã hóa (Coding Unit - CU). Phân phân chia hình ảnh 110 có thể phân chia hình ảnh thành tổ hợp của nhiều bộ phận mã hóa, bộ phận dự báo, và bộ phận biến đổi, và mã hóa hình ảnh bằng cách chọn tổ hợp của bộ phận mã hóa, bộ phận dự báo, và bộ phận biến đổi dựa trên tiêu chuẩn được xác định trước (ví dụ như, hàm trị số).

Ví dụ như, một hình ảnh có thể được phân chia thành nhiều bộ phận mã hóa. Để phân chia các bộ phận mã hóa trong hình ảnh, cấu trúc cây đệ quy chẳng hạn như cấu trúc cây tứ phân có thể được sử dụng. Bộ phận mã hóa được phân chia thành các bộ phận mã hóa khác nhau sử dụng video hoặc bộ phận mã hóa lớn nhất làm nghiệm có thể được phân chi để có nhiều nút con như số lượng được phân chia các bộ phận mã hóa. Bộ phận mã hóa không được phân chia nữa theo giới hạn được xác định trước trở thành nút lá. Tức là, khi giả sử rằng chỉ có thể phân chia vuông cho một bộ phận mã hóa, một bộ phận mã hóa này có thể được phân chia thành tận bốn bộ phận mã hóa khác nhau.

Trong nội dung mô tả dưới đây, theo phương án của sáng chế, bộ phận mã hóa có

thể được sử dụng làm ý nghĩa của đơn vị thực hiện việc mã hóa hoặc ý nghĩa của đơn vị thực hiện việc giải mã.

Bộ phận dự báo có thể là bộ phận được phân chia trong hình dạng của ít nhất một hình vuông, hình chữ nhật hoặc hình tương tự của cùng một kích cỡ trong một bộ phận mã hóa, hoặc đó có thể là bất kỳ một bộ phận dự báo nào, trong số các bộ phận dự báo được phân chia trong một bộ phận mã hóa, mà được phân chia để có hình dạng và/hoặc kích cỡ khác với hình dạng và/hoặc kích cỡ của một bộ phận dự báo khác.

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

Phần dự báo 120 và 125 có thể gồm phần dự báo liên khung 120 mà thực hiện dự báo liên khung và phần dự báo nội khung 125 mà thực hiện dự báo nội khung. Có thể xác định việc liệu sử dụng dự báo liên khung hay thực hiện dự báo nội khung cho bộ phận dự báo, và xác định thông tin cụ thể (ví dụ như, phương thức dự báo nội khung, vector chuyển động, hình ảnh tham chiếu, v.v.) theo mỗi phương pháp dự báo. Tại điểm này, bộ phận xử lý để thực hiện việc dự báo có thể là khác với bộ phận xử lý để xác định phương pháp dự báo và nội dung cụ thể. Ví dụ như, phương pháp dự báo và phương thức dự báo có thể được xác định trong bộ phận dự báo, và việc dự báo có thể được thực hiện trong bộ phận biến đổi. Hệ số dư (khối dư) giữa khối dự báo được tạo ra và khối gốc có thể được nhập vào phần biến đổi 130. Ngoài ra, thông tin phương thức dự báo, thông tin vector chuyển động và thông tin tương tự được sử dụng để dự báo có thể được mã hóa bởi phần mã hóa entropi 165 cùng với hệ số dư và được truyền đến bộ giải mã. Khi phương thức mã hóa cụ thể được sử dụng, khối gốc có thể được mã hóa theo hoàn cảnh thực tế và được truyền đến bộ giải mã mà không tạo ra khối dự báo thông qua phần dự báo 120 và 125.

Phần dự báo liên khung 120 có thể dự báo bộ phận dự báo dựa trên thông tin về ít nhất một hình ảnh trong số các hình ảnh trước khi hoặc sau khi hình ảnh hiện tại, và trong một số trường hợp, nó có thể dự báo bộ phận dự báo dựa trên thông tin về khu vực cục bộ đã được mã hóa trong hình ảnh hiện tại. Phần dự báo liên khung 120 có thể gồm phần nội suy hình ảnh tham chiếu, bộ phận dự báo chuyển động, và bộ phận bù chuyển động.

Phần nội suy hình ảnh tham chiếu có thể nhận thông tin hình ảnh tham chiếu từ bộ nhớ 155 và tạo ra thông tin điểm ảnh của số nguyên của các điểm ảnh hoặc nhỏ hơn từ

hình ảnh tham chiếu. Trong trường hợp điểm ảnh độ sáng, bộ lọc nội suy 8 nhánh dựa trên DCT với hệ số lọc biến đổi có thể được sử dụng để tạo ra thông tin điểm ảnh của số nguyên của các điểm ảnh hoặc nhỏ hơn bằng đơn vị của $1/4$ điểm ảnh. Trong trường hợp tín hiệu chênh lệch màu, bộ lọc nội suy 4 nhánh dựa trên DCT với hệ số lọc biến đổi có thể được sử dụng để tạo ra thông tin điểm ảnh của số nguyên của các điểm ảnh hoặc nhỏ hơn bằng đơn vị của $1/8$ điểm ảnh.

Bộ phận dự báo chuyển động có thể thực hiện dự báo chuyển động dựa trên hình ảnh tham chiếu được nội suy bởi phần nội suy hình ảnh tham chiếu. Các phương pháp khác nhau chẳng hạn như thuật toán ghép 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 algorithm - NTS) có thể được sử dụng làm phương pháp tính toán vector chuyển động. Vector chuyển động có thể có trị số vector chuyển động của đơn vị là $1/2$ hoặc $1/4$ điểm ảnh dựa trên các điểm ảnh được nội suy. Bộ phận dự báo chuyển động có thể dự báo bộ phận dự báo hiện tại bằng phương pháp dự báo chuyển động biến đổi. Các phương pháp khác nhau chẳng hạn như phương pháp nhảy, phương pháp hợp nhất, phương pháp dự báo vector chuyển động tiên tiến (Advanced Motion Vector Prediction - AMVP), phương pháp sao chép nội khối và phương pháp tương tự có thể được sử dụng làm phương pháp dự báo chuyển động.

Phần dự báo nội khung 125 có thể tạo ra bộ phận dự báo dựa trên thông tin về các điểm ảnh tham chiếu quanh khối hiện tại, đây là thông tin điểm ảnh trong hình ảnh hiện tại. Khi khối trong miền lân cận của bộ phận dự báo hiện tại là khối mà trên đó dự báo liên khung đã được thực hiện và do đó điểm ảnh tham chiếu là điểm ảnh mà trên đó dự báo liên khung đã được thực hiện, điểm ảnh tham chiếu được đưa vào khối mà trên đó dự báo liên khung đã được thực hiện có thể được sử dụng thay cho thông tin điểm ảnh tham chiếu của khối trong miền lân cận mà trên đó dự báo nội khung đã được thực hiện. Tức là, khi điểm ảnh tham chiếu là không sẵn có, ít nhất một điểm ảnh tham chiếu trong số các điểm ảnh tham chiếu sẵn có có thể được sử dụng thay cho thông tin điểm ảnh tham chiếu không sẵn có.

Trong dự báo nội khung, phương thức dự báo có thể có phương thức dự báo góc mà sử dụng thông tin điểm ảnh tham chiếu theo hướng dự báo, và phương thức dự báo không góc mà không sử dụng thông tin hướng khi thực hiện việc dự báo. Phương thức dự báo thông tin độ sáng có thể là khác với phương thức dự báo thông tin chênh lệch màu, và thông tin phương thức dự báo nội khung 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 chênh lệch màu.

Nếu kích cỡ của bộ phận dự báo giống như kích cỡ của bộ phận biến đổi khi dự báo nội khung được thực hiện, dự báo nội khung có thể được thực hiện cho bộ phận dự báo dựa trên điểm ảnh ở mặt trái, điểm ảnh ở mặt trên cùng-bên trái, và điểm ảnh trên đỉnh của bộ phận dự báo. Tuy nhiên, nếu kích cỡ của bộ phận dự báo khác với kích cỡ của bộ phận biến đổi khi dự báo nội khung được thực hiện, dự báo nội khung có thể được thực hiện sử dụng điểm ảnh tham chiếu dựa trên bộ phận biến đổi. Ngoài ra, dự báo nội khung sử dụng phương thức phân chia $N \times N$ có thể chỉ được sử dụng cho bộ phận mã hóa nhỏ nhất.

Phương pháp dự báo nội khung có thể tạo ra khối dự báo sau khi ứng dụng bộ lọc tron nội khung thích ứng (Adaptive Intra Smoothing - AIS) cho điểm ảnh tham chiếu theo phương thức dự báo. Kiểu bộ lọc AIS được ứng dụng cho điểm ảnh tham chiếu có thể thay đổi. Để thực hiện phương pháp dự báo nội khung, phương thức dự báo nội khung của bộ phận dự báo hiện tại có thể được dự báo từ phương thức dự báo nội khung của bộ phận dự báo tồn tại ở miền lân cận của bộ phận dự báo hiện tại. Khi phương thức dự báo của bộ phận dự báo hiện tại được dự báo sử dụng thông tin phương thức được dự báo từ bộ phận dự báo lân cận, nếu các phương thức dự báo nội khung của bộ phận dự báo hiện tại giống như bộ phận dự báo trong miền lân cận, thông tin biểu thị rằng các phương thức dự báo của bộ phận dự báo hiện tại giống như bộ phận dự báo trong miền lân cận có thể được truyền bằng cách sử dụng thông tin cờ được xác định trước, và nếu các phương thức dự báo của bộ phận dự báo hiện tại và bộ phận dự báo trong miền lân cận là khác nhau, thông tin phương thức dự báo của khối hiện tại có thể được mã hóa bằng cách thực hiện mã hóa entrôpi.

Ngoài ra, khối dư gồm bộ phận dự báo đã thực hiện việc dự báo dựa trên bộ phận dự báo được tạo ra bởi phần dự báo 120 và 125 và thông tin hệ số dư, đây là trị số chênh lệch của bộ phận dự báo với khối gốc, có thể được tạo ra. Khối dư được tạo ra có thể được nhập vào phần biến đổi 130.

Phần biến đổi 130 có thể biến đổi khối dư gồm khối gốc và thông tin hệ số dư của bộ phận 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), biến đổi sin rời rạc (Discrete Sine Transform - DST), hoặc nhảy biến đổi. Việc ứng dụng DCT, DST hay KLT để biến đổi khối dư có thể được xác định dựa trên thông tin phương

thức dự báo nội khung của bộ phận dự báo được sử dụng để tạo ra khối dư.

Phần lượng tử hóa 135 có thể lượng tử hóa các trị số được biến đổi thành miền tần số bởi phần biến đổi 130. Các hệ số lượng tử hóa có thể thay đổi theo khối hoặc trị số của video. Trị số được tính toán bởi phần lượng tử hóa 135 có thể được cung cấp cho phần lượng tử hóa ngược 140 và phần sắp xếp lại 160.

Phần sắp xếp lại 160 có thể sắp xếp lại các trị số hệ số cho các hệ số dư được lượng tử hóa.

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 như, phần sắp xếp lại 160 có thể quét các hệ số DC lên đến các hệ số miền tần số cao sử dụng phương pháp quét hình chữ z, và thay đổi các hệ số thành hình dạng vector một chiều. Theo kích cỡ của bộ phận biến đổi và phương thức dự báo nội khung, quét dọc là quét các hệ số của hình dạng khối hai chiều theo hướng cột và quét ngang là quét các hệ số của hình dạng khối hai chiều theo hướng hàng có thể được sử dụng thay vì quét hình chữ z. Tức là, theo kích cỡ của bộ phận biến đổi và phương thức dự báo nội khung, phương pháp quét sẽ được sử dụng có thể được xác định trong số quét hình chữ z, quét hướng dọc, và quét hướng ngang.

Phần mã hóa entropi 165 có thể thực hiện mã hóa entropi dựa trên các trị số được tính toán bởi phần sắp xếp lại 160. Mã hóa entropi có thể sử dụng các phương pháp mã hóa khác nhau chẳng hạn như Golomb hàm mũ, mã hóa độ dài có thể thay đổi thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding - CAVLC), mã hóa số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding - CABAC), và phương pháp tương tự.

Phần mã hóa entropi 165 có thể mã hóa các thông tin khác nhau chẳng hạn như thông tin hệ số dư và thông tin kiểu khối của bộ phận mã hóa, thông tin phương thức dự báo, thông tin bộ phận phân chia, thông tin bộ phận dự báo và thông tin bộ phận truyền, thông tin vector chuyển động, thông tin khung tham chiếu, thông tin nội suy khối, và thông tin lọc được nhập vào từ phần sắp xếp lại 160 và các phần dự báo 120 và 125.

Phần mã hóa entropi 165 có thể mã hóa entropi trị số hệ số của bộ phận mã hóa được nhập vào từ phần sắp xếp lại 160.

Phần lượng tử hóa ngược 140 và phần biến đổi ngược 145 lượng tử hóa ngược các trị số được lượng tử hóa bởi phần lượng tử hóa 135 và biến đổi ngược các trị số được biến đổi bởi phần biến đổi 130. Hệ số dư được tạo ra bởi phần lượng tử hóa ngược 140

và phần biến đổi ngược 145 có thể được kết hợp với bộ phận dự báo được dự báo thông qua phần ước lượng chuyển động, bộ phận bù chuyển động, và phần dự báo nội khung được đưa vào phần dự báo 120 và 125 để tạo ra khối được tái cấu trúc.

Phần bộ lọc 150 có thể gồm ít nhất một trong số bộ lọc giải khối, bộ phận điều chỉnh độ lệch, và bộ lọc vòng lặp thích ứng (Adaptive Loop Filter - ALF).

Bộ lọc giải khối có thể loại bỏ độ méo khối được tạo ra bởi ranh giới giữa các khối trong hình ảnh được tái cấu trúc. Để xác định việc liệu có hay không thực hiện việc giải khối, việc liệu hoặc không ứng dụng bộ lọc giải khối cho khối hiện tại có thể được xác định dựa trên các điểm ảnh được đưa vào một số cột hoặc hàng được đưa vào khối. Bộ lọc khỏe hoặc bộ lọc yếu có thể được ứng dụng theo độ bền lọc giải khối cần thiết khi bộ lọc giải khối được ứng dụng cho khối. Ngoài ra, khi lọc hướng dọc và lọc hướng ngang được thực hiện trong việc ứng dụng bộ lọc giải khối, lọc hướng ngang và lọc hướng dọc có thể được xử lý song song.

Bộ phận điều chỉnh độ lệch có thể điều chỉnh độ lệch cho video gốc bằng đơn vị điểm ảnh cho video mà trên đó việc giải khối đã được thực hiện. Để thực hiện điều chỉnh độ lệch cho hình ảnh cụ thể, có thể sử dụng phương pháp phân chia các điểm ảnh được đưa vào video thành một số khu vực nhất định, xác định khu vực để thực hiện độ lệch, và ứng dụng độ lệch cho khu vực, hoặc phương pháp áp dụng độ lệch xét đến thông tin cạnh biên của mỗi điểm ảnh.

Lọc vòng lặp thích ứng (Adaptive Loop Filtering - ALF) có thể được thực hiện dựa trên trị số thu được bằng cách so sánh video được lọc và tái cấu trúc và video gốc. Sau khi chia các điểm ảnh được đưa vào video thành các nhóm được xác định trước, một bộ lọc cần được ứng dụng cho nhóm tương ứng có thể được xác định, và việc lọc có thể được thực hiện theo cách khác nhau cho mỗi nhóm. Tín hiệu độ sáng, là thông tin liên quan đến việc liệu có hay không ứng dụng ALF, có thể được truyền cho mỗi bộ phận mã hóa (Coding Unit - CU), và hình dạng và hệ số lọc của bộ lọc ALF cần được ứng dụng có thể thay đổi theo mỗi khối. Ngoài ra, bộ lọc ALF của cùng một loại (loại cố định) có thể được ứng dụng bất kể đặc điểm của khối cần được ứng dụng.

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

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

Như được thể hiện trên Fig.2, bộ giải mã video 200 có thể gồm phần giải mã entropi

210, phần sắp xếp lại 215, phần lượng tử hóa ngược 220, phần biến đổi ngược 225, phần dự báo 230 và 235, phần bộ lọc 240, và bộ nhớ 245.

Khi video dòng bit được nhập vào từ bộ mã hóa video, dòng bit đầu vào có thể được giải mã theo quy trình ngược với quy trình của bộ mã hóa video.

Phần giải mã entropi 210 có thể thực hiện giải mã entropi theo quy trình ngược với quy trình thực hiện mã hóa entropi trong phần giải mã entropi của bộ mã hóa video. Ví dụ như, các phương pháp khác nhau tương ứng với phương pháp được thực hiện bởi bộ mã hóa video, chẳng hạn như Golomb hàm mũ, mã hóa độ dài có thể thay đổi thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding - CAVLC), và mã hóa số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding - CABAC), có thể được ứng dụng.

Phần giải mã entropi 210 có thể giải mã thông tin liên quan đến dự báo nội khung và dự báo liên khung được thực hiện bởi bộ mã hóa.

Phần sắp xếp lại 215 có thể thực hiện việc sắp xếp lại trên dòng bit được giải mã entropi bởi phần giải mã entropi 210 dựa trên phương pháp sắp xếp lại được thực hiện bởi bộ mã hóa. Các hệ số được thể hiện ở hình dạng vector một chiều có thể được tái cấu trúc và được bố trí lại như các hệ số của hình dạng khối hai chiều. Phần sắp xếp lại 215 có thể nhận thông tin liên quan đến việc quét hệ số được thực hiện bởi phần giải mã và thực hiện việc tái cấu trúc thông qua phương pháp quét ngược dựa trên thứ tự quét được thực hiện bởi phần giải mã tương ứng.

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

Phần biến đổi ngược 225 có thể thực hiện việc biến đổi ngược, tức là, DCT ngược, DST ngược, hoặc KLT ngược, đối với việc biến đổi, tức là, DCT, DST, hoặc KLT, được thực hiện bởi phần biến đổi theo kết quả lượng tử hóa được thực hiện bởi bộ mã hóa video. Việc biến đổi ngược có thể được thực hiện dựa trên bộ phận truyền được xác định bởi bộ mã hóa video. Phần biến đổi ngược 225 của bộ giải mã video có thể thực hiện chọn lọc kỹ thuật biến đổi (ví dụ như, DCT, DST, KLT) theo nhiều thông tin chẳng hạn như phương pháp dự báo, kích cỡ của khối hiện tại, hướng 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 entropi 210 và thông tin trên hình ảnh hoặc khối được giải mã trước được cung cấp bởi bộ nhớ 245.

Như được mô tả trên đây, nếu kích cỡ của bộ phận dự báo và kích cỡ của bộ phận

biến đổi là giống nhau khi dự báo nội khung được thực hiện trong cùng cách thức như hoạt động của bộ mã hóa video, dự báo nội khung được thực hiện trên bộ phận dự báo dựa trên điểm ảnh tồn tại ở mặt trái, điểm ảnh ở mặt trên cùng-bên trái, và điểm ảnh trên đỉnh của bộ phận dự báo. Tuy nhiên, nếu kích cỡ của bộ phận dự báo và kích cỡ của bộ phận biến đổi là khác nhau khi dự báo nội khung được thực hiện, dự báo nội khung có thể được thực hiện sử dụng điểm ảnh tham chiếu dựa trên bộ phận biến đổi. Ngoài ra, dự báo nội khung sử dụng phương thức phân chia $N \times N$ có thể chỉ được sử dụng cho bộ phận mã hóa nhỏ nhất.

Phần dự báo 230 và 235 có thể gồm phần xác định bộ phận dự báo, phần dự báo liên khung, và phần dự báo nội khung. Phần xác định bộ phận dự báo có thể nhận các thông tin khác nhau chẳng hạn như bộ phận dự báo thông tin được nhập vào từ phần giải mã entropi 210, thông tin phương thức dự báo của phương pháp dự báo nội khung, thông tin liên quan đến dự báo chuyển động của phương pháp dự báo liên khung, và thông tin tương tự, nhận dạng bộ phận dự báo từ bộ phận mã hóa hiện tại, và xác định việc liệu bộ phận dự báo thực hiện dự báo liên khung hoặc dự báo nội khung. Phần dự báo liên khung 230 có thể thực hiện dự báo liên khung trên bộ phận dự báo hiện tại dựa trên thông tin được đưa vào ít nhất một hình ảnh trong số các hình ảnh trước khi hoặc sau khi hình ảnh hiện tại gồm bộ phận dự báo hiện tại bằng cách sử dụng thông tin cần thiết cho dự báo liên khung của bộ phận dự báo hiện tại được cung cấp bởi bộ mã hóa video. Theo lựa chọn, phần dự báo liên khung 230 có thể thực hiện dự báo liên khung dựa trên thông tin về khu vực cục bộ được tái cấu trúc trước trong hình ảnh hiện tại gồm bộ phận dự báo hiện tại.

Để thực hiện dự báo liên khung, có thể xác định, dựa trên bộ phận mã hóa, việc liệu phương pháp dự báo chuyển động của bộ phận dự báo được đưa vào bộ phận mã hóa tương ứng là phương thức nhảy, phương thức hợp nhất, phương thức dự báo vector chuyển động tiên tiến (Advanced Motion Vector Prediction mode - AMVP mode), hay phương thức sao chép nội khối.

Phần dự báo nội khung 235 có thể tạo ra khối dự báo dựa trên thông tin về điểm ảnh trong hình ảnh hiện tại. Khi bộ phận dự báo là bộ phận dự báo mà đã thực hiện dự báo nội khung, dự báo nội khung có thể được thực hiện dựa trên thông tin phương thức dự báo nội khung của bộ phận dự báo được cung cấp bởi bộ mã hóa video. Phần dự báo nội khung 235 có thể gồm bộ lọc tron nội khung thích ứng (Adaptive Intra Smoothing - AIS), phần điểm ảnh tham chiếu nội suy, và bộ lọc DC. Bộ lọc AIS là phần thực hiện

việc lọc trên điểm ảnh tham chiếu của khối hiện tại, và có thể xác định việc liệu hoặc không ứng dụng bộ lọc theo phương thức dự báo của bộ phận dự báo hiện tại và ứng dụng bộ lọc. Việc lọc AIS có thể được thực hiện trên điểm ảnh tham chiếu của khối hiện tại bằng cách sử dụng phương thức dự báo và thông tin bộ lọc AIS của bộ phận dự báo được cung cấp bởi bộ mã hóa video. Khi phương thức dự báo của khối hiện tại là phương thức không thực hiện việc lọc AIS, bộ lọc AIS có thể không được ứng dụng.

Khi phương thức dự báo của bộ phận dự báo là bộ phận dự báo mà thực hiện dự báo nội khung dựa trên trị số điểm ảnh thu được bằng cách nội suy điểm ảnh tham chiếu, phần điểm ảnh tham chiếu nội suy có thể tạo ra điểm ảnh tham chiếu của đơn vị điểm ảnh có trị số kiểu số nguyên hoặc nhỏ hơn bằng cách nội suy điểm ảnh tham chiếu. Khi phương thức dự báo của bộ phận dự báo hiện tại là phương thức dự báo mà tạo ra khối dự báo mà không nội suy điểm ảnh tham chiếu, điểm ảnh tham chiếu có thể không được nội suy. Bộ lọc DC có thể tạo ra khối dự báo thông qua lọc khi phương thức dự báo của khối hiện tại là phương thức DC.

Hình ảnh hoặc khối được tái cấu trúc có thể được cung cấp cho phần bộ lọc 240. Phần bộ lọc 240 có thể gồm bộ lọc giải khối, bộ phận điều chỉnh độ lệch, và ALF.

Thông tin về việc liệu bộ lọc giải khối được ứng dụng cho hình ảnh hoặc khối tương ứng và thông tin về việc liệu bộ lọc khỏe hoặc bộ lọc yếu có được ứng dụng khi bộ lọc giải khối được ứng dụng có thể được cung cấp bởi bộ mã hóa video. Bộ lọc giải khối của bộ giải mã video có thể được cung cấp thông tin liên quan đến bộ lọc giải khối được cung cấp bởi bộ mã hóa video, và bộ giải mã video có thể thực hiện việc lọc giải khối trên khối tương ứng.

Bộ phận điều chỉnh độ lệch có thể thực hiện điều chỉnh độ lệch trên video được tái cấu trúc dựa trên điều chỉnh kiểu độ lệch và trị số độ lệch thông tin được ứng dụng cho video khi việc mã hóa được thực hiện.

ALF có thể được ứng dụng cho bộ phận mã hóa dựa trên thông tin về việc liệu có hay không ứng dụng ALF và thông tin trên các hệ số ALF được cung cấp bởi bộ mã hóa. Thông tin ALF có thể được cung cấp cần được đưa vào tập hợp thông số cụ thể.

Bộ nhớ 245 có thể lưu trữ khối hoặc hình ảnh được tái cấu trúc và sử dụng nó làm hình ảnh tham chiếu hoặc khối tham chiếu và có thể cung cấp hình ảnh được tái cấu trúc cho bộ phận đầu ra.

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

Khối mã hóa có kích cỡ tối đa có thể được xác định là khối cây mã hóa. Hình ảnh

được phân chia thành nhiều đơn vị cây mã hóa (Coding Tree Unit - CTU). Đơn vị cây mã hóa là bộ phận mã hóa có kích cỡ tối đa và có thể được gọi là bộ phận mã hóa lớn (Large Coding Unit - LCU). Fig.3 thể hiện ví dụ trong đó hình ảnh được phân chia thành nhiều đơn vị cây mã hóa.

Kích cỡ của đơn vị cây mã hóa có thể được xác định ở cấp độ hình ảnh hoặc cấp độ trình tự. Để thực hiện được điều này, thông tin biểu thị kích cỡ của đơn vị cây mã hóa có thể được báo hiệu thông qua tập hợp thông số hình ảnh hoặc tập hợp thông số trình tự.

Ví dụ như, kích cỡ của đơn vị cây mã hóa cho toàn bộ hình ảnh theo trình tự có thể được đặt thành 128 x 128. Theo lựa chọn, ở cấp độ hình ảnh, bất kỳ một trong số 128 x 128 và 256 x 256 có thể được xác định là kích cỡ của đơn vị cây mã hóa. Ví dụ như, kích cỡ của đơn vị cây mã hóa có thể được đặt thành 128 x 128 ở hình ảnh thứ nhất, và kích cỡ của đơn vị cây mã hóa có thể được đặt thành 256 x 256 ở hình ảnh thứ hai.

Các khối mã hóa có thể được tạo ra bằng phân chia đơn vị cây mã hóa. Khối mã hóa biểu thị đơn vị cơ bản để thực hiện việc mã hóa/giải mã. Ví dụ như, việc dự báo hoặc biến đổi có thể được thực hiện cho mỗi khối mã hóa, hoặc phương thức mã hóa dự báo có thể được xác định cho mỗi khối mã hóa. Ở đây, phương thức mã hóa dự báo biểu thị phương pháp tạo ra video dự báo. Ví dụ như, phương thức mã hóa dự báo có thể gồm dự báo trong màn hình (dự báo nội màn hình), dự báo giữa các màn hình (dự báo liên màn hình), tham chiếu hình ảnh hiện tại (Current Picture Referencing - CPR) hoặc sao chép nội khối (Intra-Block Copy - IBC), hoặc dự báo kết hợp. Đối với khối mã hóa, khối dự báo có thể được tạo ra bằng cách sử dụng ít nhất một phương thức mã hóa dự báo trong số dự báo nội khung, dự báo liên khung, tham chiếu hình ảnh hiện tại, và dự báo kết hợp.

Thông tin biểu thị phương thức mã hóa dự báo của khối hiện tại có thể được báo hiệu thông qua dòng bit. Ví dụ như, thông tin có thể là cờ 1 bit biểu thị việc liệu phương thức mã hóa dự báo là phương thức nội khung hay phương thức liên khung. Chỉ khi phương thức mã hóa dự báo của khối hiện tại được xác định là phương thức liên khung, tham chiếu hình ảnh hiện tại hoặc dự báo kết hợp có thể được sử dụng.

Tham chiếu hình ảnh hiện tại là để đặt hình ảnh hiện tại là hình ảnh tham chiếu và thu được khối dự báo của khối hiện tại từ khu vực đã được mã hóa/giải mã rồi trong hình ảnh hiện tại. Ở đây, hình ảnh hiện tại nghĩa là hình ảnh gồm khối hiện tại. Thông tin biểu thị việc liệu tham chiếu hình ảnh hiện tại có được ứng dụng cho khối hiện tại có thể được báo hiệu thông qua dòng bit. Ví dụ như, thông tin có thể là cờ 1 bit. Khi cờ là đúng, phương thức mã hóa dự báo của khối hiện tại có thể được xác định là tham chiếu hình

ảnh hiện tại, và khi còn là sai, phương thức dự báo của khối hiện tại có thể được xác định là dự báo liên khung.

Theo lựa chọn, phương thức mã hóa dự báo của khối hiện tại có thể được xác định dựa trên chỉ số hình ảnh tham chiếu. Ví dụ như, khi chỉ số hình ảnh tham chiếu biểu thị hình ảnh hiện tại, phương thức mã hóa dự báo của khối hiện tại có thể được xác định là tham chiếu hình ảnh hiện tại. Khi chỉ số hình ảnh tham chiếu biểu thị hình ảnh không phải là hình ảnh hiện tại, phương thức mã hóa dự báo của khối hiện tại có thể được xác định là dự báo liên khung. Tức là, tham chiếu hình ảnh hiện tại là phương pháp dự báo sử dụng thông tin về khu vực trong đó việc mã hóa/giải mã đã được hoàn thành trong hình ảnh hiện tại, và dự báo liên khung là phương pháp dự báo sử dụng thông tin về một hình ảnh khác mà trong đó việc mã hóa/giải mã đã được hoàn thành.

Việc dự báo kết hợp thể hiện phương thức mã hóa trong đó hai hoặc nhiều hơn hai trong số dự báo nội khung, dự báo liên khung, và tham chiếu hình ảnh hiện tại được kết hợp. Ví dụ như, khi việc dự báo kết hợp được ứng dụng, khối dự báo thứ nhất có thể được tạo ra dựa trên một trong số dự báo nội khung, dự báo liên khung, và tham chiếu hình ảnh hiện tại, và khối dự báo thứ hai có thể được tạo ra dựa trên một khối dự báo khác. Khi khối dự báo thứ nhất và khối dự báo thứ hai được tạo ra, khối dự báo cuối cùng có thể được tạo ra thông qua phép tính trị số trung bình hoặc phép tính tổng có trọng số của khối dự báo thứ nhất và khối dự báo thứ hai. Thông tin biểu thị việc liệu có hay không việc dự báo kết hợp được ứng dụng có thể được báo hiệu thông qua dòng bit. Thông tin có thể là còn 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 đề cập đến kỹ thuật phân chia dùng để phân chia khối hiện tại thành bốn khối. Do phân chia cây tứ phân, khối hiện tại có thể được phân chia thành bốn phần có dạng hình vuông (xem 'SPLIT_QT' trên Fig.4 (a)).

Phân chia cây nhị phân đề cập đến kỹ thuật phân chia dùng để phân chia khối hiện tại thành hai khối. Phân chia khối hiện tại thành hai khối dọc theo hướng dọc (tức là, sử dụng đường dọc đi qua khối hiện tại) có thể được gọi là phân chia cây nhị phân hướng dọc, và phân chia khối hiện tại thành hai khối dọc theo hướng ngang (tức là, sử dụng

đường ngang đi qua khối hiện tại) có thể được gọi là phân chia cây nhị phân hướng ngang. Do phân chia cây nhị phân, khối hiện tại có thể được phân chia thành hai phần có hình dạng không phải hình vuông. ‘SPLIT_BT_VER’ trên Fig.4 (b) thể hiện kết quả của phân chia cây nhị phân hướng dọc, và ‘SPLIT_BT_HOR’ trên Fig.4 (c) thể hiện kết quả của phân chia cây nhị phân hướng ngang.

Phân chia cây tam phân đề cập đến kỹ thuật phân chia dùng để phân chia khối hiện tại thành ba khối. Phân chia khối hiện tại thành ba khối dọc theo hướng dọc (tức là, sử dụng hai đường dọc đi qua khối hiện tại) có thể được gọi là phân chia cây tam phân hướng dọc, và phân chia khối hiện tại thành ba khối dọc theo hướng ngang (tức là, sử dụng hai đường ngang đi qua khối hiện tại) có thể được gọi là phân chia cây tam phân hướng ngang. Do phân chia cây tam phân, khối hiện tại có thể được phân chia thành ba phần có hình dạng không phải hình vuông. Tại điểm này, độ rộng/độ cao của phân chia được bố trí ở trung tâm của khối hiện tại có thể là lớp gấp hai lần độ rộng/độ cao của các phần kia. ‘SPLIT_TT_VER’ trên Fig.4 (d) thể hiện kết quả của phân chia cây tam phân hướng dọc, và ‘SPLIT_TT_HOR’ trên Fig.4 (e) thể hiện kết quả của phân chia cây tam phân hướng ngang.

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

Theo lựa chọn, độ sâu phân chia tối đa cho mỗi kỹ thuật phân chia có thể được xác định riêng. Ví dụ như, độ sâu phân chia tối đa được phép cho phân chia cây tứ phân có thể là khác với độ sâu phân chia tối đa được phép cho phân chia cây nhị phân và/hoặc phân chia cây tam phân.

Bộ mã hóa có thể báo hiệu thông tin biểu thị ít nhất một trong số kiểu phân chia và độ sâu phân chia của khối hiện tại thông qua dòng 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 tách từ dòng bit.

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

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 nhiều cây.

Khối mã hóa được tạo ra bằng cách ứng dụng phân chia nhiều cây cho khối mã hóa

có thể được gọi là các khối mã hóa dưới. Khi độ sâu phân chia của khối mã hóa là k , độ sâu phân chia của khối mã hóa dưới được đặt thành $k + 1$.

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

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

Để xác định kiểu phân chia của khối mã hóa, thông tin biểu thị việc liệu có hay không khối mã hóa có thể được phân chia có thể được báo hiệu thông qua dòng bit. Thông tin là cờ 1 bit của 'split_cu_flag', và khi cờ là đúng, điều đó biểu thị rằng khối mã hóa được phân chia bằng kỹ thuật phân chia cây đề mục ($\rightarrow$ tứ phân).

Khi split_cu_flag là đúng, thông tin biểu thị việc liệu khối mã hóa là cây tứ phân được phân chia có thể được báo hiệu thông qua dòng bit. Thông tin là cờ 1 bit của split_qt_flag, và khi cờ là đúng, khối mã hóa có thể được phân chia thành bốn khối.

Ví dụ như, trong ví dụ được thể hiện trên Fig.5, do đơn vị cây mã hóa là cây tứ phân được phân chia, bốn khối mã hóa có độ sâu phân chia của 1 được tạo ra. Ngoài ra, điều được thể hiện là phân chia cây tứ phân được ứng dụng lần nữa cho các khối mã hóa thứ nhất và thứ tư trong số bốn khối mã hóa được tạo ra do 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 ứng dụng phân chia cây tứ phân lần nữa cho khối mã hóa có độ sâu phân chia là 2.

Khi phân chia cây tứ phân không được ứng dụng cho khối mã hóa, việc liệu phân chia cây nhị phân hoặc phân chia cây tam phân được thực hiện trên khối mã hóa có thể được xác định xét đến ít nhất một trong số kích cỡ của khối mã hóa, việc liệu khối mã hóa được bố trí tại ranh giới hình ảnh, độ sâu phân chia tối đa, và kiểu phân chia của khối lân cận. Khi xác định thực hiện phân chia cây nhị phân hoặc phân chia cây tam phân trên khối mã hóa, thông tin biểu thị hướng phân chia có thể được báo hiệu thông qua dòng bit. Thông tin có thể là cờ 1 bit của mtt_split_cu_vertical_flag. Dựa trên cờ,

việc liệu hướng phân chia là hướng dọc hoặc hướng ngang có thể được xác định. Ngoài ra, thông tin biểu thị việc liệu phân chia cây nhị phân hoặc phân chia cây tam phân được ứng dụng cho khối mã hóa có thể được báo hiệu thông qua dòng bit. Thông tin có thể là cờ 1 bit của `mtt_split_cu_binary_flag`. Dựa trên cờ, việc liệu phân chia cây nhị phân hoặc phân chia cây tam phân được ứng dụng cho khối mã hóa có thể được xác định.

Ví dụ như, trong ví dụ được thể hiện trên Fig.5, điều được thể hiện là phân chia cây nhị phân hướng dọc được ứng dụng cho khối mã hóa có độ sâu phân chia là 1, phân chia cây tam phân hướng dọc được ứng dụng cho khối mã hóa bên trái trong số khối mã hóa được tạo ra do phân chia, và phân chia cây nhị phân hướng dọc được ứng dụng cho khối mã hóa bên phải.

Dự báo liên khung là phương thức mã hóa dự báo dự báo khối hiện tại bằng cách sử dụng thông tin của hình ảnh trước. Ví dụ như, khối ở cùng vị trí với khối hiện tại trong hình ảnh trước (trong phần dưới đây, khối được sắp xếp) có thể được đặt là khối dự báo của khối hiện tại. Trong phần dưới đây, khối dự báo được tạo ra dựa trên khối ở cùng vị trí với khối hiện tại sẽ được gọi là khối dự báo được sắp xếp.

Mặt khác, khi vật hiện có trong hình ảnh trước đã di chuyển đến một vị trí khác trong hình ảnh hiện tại, khối hiện tại có thể được dự báo hiệu quả bằng cách sử dụng chuyển động của vật. Ví dụ như, khi hướng chuyển động và kích cỡ của vật có thể được biết bằng cách so sánh hình ảnh trước và hình ảnh hiện tại, khối dự báo (hoặc video dự báo) của khối hiện tại có thể được tạo ra xét đến thông tin chuyển động của vật. Trong nội dung dưới đây, khối dự báo được tạo ra bằng cách sử dụng thông tin chuyển động có thể được gọi là khối dự báo chuyển động.

Khối dư có thể được tạo ra bằng cách trừ khối dự báo từ khối hiện tại. Tại điểm này, khi có chuyển động của vật, năng lượng của khối dư có thể được giảm bớt bằng cách sử dụng khối dự báo chuyển động thay vì khối dự báo được sắp xếp, và do đó, hiệu năng nén của khối dư có thể được cải thiện.

Như được mô tả trên đây, tạo ra khối dự báo bằng cách sử dụng thông tin chuyển động có thể được gọi là dự báo bù chuyển động. Trong phần lớn dự báo liên khung, khối dự báo có thể được tạo ra dựa trên dự báo bù chuyển động.

Thông tin chuyển động có thể gồm ít nhất một trong số vector chuyển động, chỉ số hình ảnh tham chiếu, hướng dự báo, và chỉ số trọng số hai hướng. Vector chuyển động thể hiện hướng chuyển động và kích cỡ của vật. Chỉ số hình ảnh tham chiếu định rõ hình ảnh tham chiếu của khối hiện tại trong số các hình ảnh tham chiếu được đưa vào danh

mục hình ảnh tham chiếu. Hướng dự báo biểu thị bất kỳ một trong số dự báo L0 đơn hướng, dự báo L1 đơn hướng, và dự báo hai hướng (dự báo L0 và dự báo L1). Theo hướng dự báo của khối hiện tại, ít nhất một trong số thông tin chuyển động theo hướng L0 và thông tin chuyển động theo hướng L1 có thể được sử dụng. Chỉ số trọng số hai hướng định rõ trị số định trọng số được ứng dụng cho khối dự báo L0 và trị số định trọng số được ứng dụng cho khối dự báo L1.

Fig.6 là lưu đồ minh họa phương pháp dự báo liên khung theo phương án của sáng chế.

Như được thể hiện trên Fig.6, phương pháp dự báo liên khung gồm các bước là xác định phương thức dự báo liên khung của khối hiện tại (S601), thu được thông tin chuyển động của khối hiện tại theo phương thức dự báo liên khung được xác định (S602), và thực hiện dự báo bù chuyển động cho khối hiện tại dựa trên thông tin chuyển động thu được (S603).

Ở đây, phương thức dự báo liên khung thể hiện các kỹ thuật khác nhau để xác định thông tin chuyển động của khối hiện tại, và có thể gồm phương thức dự báo liên khung mà sử dụng thông tin chuyển động tịnh tiến và phương thức dự báo liên khung mà sử dụng thông tin chuyển động affin. Ví dụ như, phương thức dự báo liên khung sử dụng thông tin chuyển động tịnh tiến có thể gồm phương thức hợp nhất và vector chuyển động phương thức dự báo, và phương thức dự báo liên khung sử dụng thông tin chuyển động affin có thể gồm phương thức hợp nhất affin và vector chuyển động affin phương thức dự báo. Thông tin chuyển động của khối hiện tại có thể được xác định dựa trên khối lân cận gần kề khối hiện tại hoặc thông tin được phân tách từ dòng bit theo phương thức dự báo liên khung.

Trong nội dung dưới đây, phương pháp dự báo liên khung sử dụng thông tin chuyển động affin sẽ được mô tả chi tiết.

Fig.7 là hình vẽ thể hiện chuyển động phi tuyến tính của đối tượng.

Chuyển động phi tuyến tính của vật có thể được tạo ra trong video. Ví dụ như, như được thể hiện trong ví dụ trên Fig.7, chuyển động phi tuyến tính của đối tượng, chẳng hạn như thu nhỏ, phóng to, xoay, biến đổi affin hoặc chuyển động tương tự của máy ảnh, có thể xảy ra. Khi chuyển động phi tuyến tính của đối tượng xảy ra, chuyển động của đối tượng không thể được biểu hiện hiệu quả với vector chuyển động tịnh tiến. Do đó, hiệu suất mã hóa có thể được cải thiện bằng cách sử dụng chuyển động affin thay vì chuyển động tuyến tính ở khu vực mà ở đó chuyển động phi tuyến tính của đối tượng xảy ra.

Fig.8 là lưu đồ minh họa phương pháp dự báo liên khung dựa trên chuyển động affin theo phương án của sáng chế.

Việc liệu kỹ thuật dự báo liên khung dựa trên chuyển động affin được ứng dụng cho khối hiện tại có thể được xác định dựa trên thông tin được phân tách từ dòng bit. Cụ thể là, việc liệu kỹ thuật dự báo liên khung dựa trên chuyển động affin được ứng dụng cho khối hiện tại có thể được xác định dựa trên ít nhất một trong số cờ biểu thị việc liệu phương thức hợp nhất affin được ứng dụng cho khối hiện tại và cờ biểu thị việc liệu vector chuyển động affin phương thức dự báo được ứng dụng cho khối hiện tại.

Khi kỹ thuật dự báo liên khung dựa trên chuyển động affin được ứng dụng cho khối hiện tại, mô hình chuyển động affin của khối hiện tại có thể được xác định (S801). 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 hiện chuyển động affin sử dụng sáu thông số, và mô hình chuyển động affin bốn thông số biểu hiện chuyển động affin sử dụng bốn thông số.

Phương trình 1 biểu hiện chuyển động affin sử dụng sáu thông số. Chuyển động affin thể hiện chuyển động tịnh tiến cho khu vực được xác định trước được xác định bởi các vector hạt affin.

【Phương trình 1】

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

Khi chuyển động affin được biểu hiện sử dụng sáu thông số, chuyển động phức tạp có thể được biểu hiện. Tuy nhiên, do số lượng bit cần thiết cho việc mã hóa mỗi trong số các thông số tăng lên, hiệu suất mã hóa có thể bị giảm. Do đó, chuyển động affin có thể được biểu hiện sử dụng bốn thông số. Phương trình 2 biểu hiện 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 tại có thể được mã hóa và được báo hiệu thông qua dòng bit. Ví dụ như, thông tin có thể là cờ 1 bit của 'affine_type_flag'. Khi trị số của cờ là 0, điều đó có thể biểu thị rằng mô hình chuyển động affin 4 thông số được ứng dụng, và khi trị số của cờ là 1, điều đó có thể biểu thị

rằng mô hình chuyển động affin 6 thông số được ứng dụng. Cờ có thể được mã hóa bằng đơn vị của phiên, ngôi, hoặc khối (ví dụ như, bằng đơn vị của khối mã hóa hoặc cây mã hóa). Khi cờ được báo hiệu ở cấp độ phiên, mô hình chuyển động affin được xác định ở cấp độ phiên có thể được ứng dụng cho tất cả các khối thuộc về phiên.

Theo lựa chọn, mô hình chuyển động affin của khối hiện tại có thể được xác định dựa trên phương thức dự báo liên khung affin của khối hiện tại. Ví dụ như, khi phương thức hợp nhất affin được ứng dụng, mô hình chuyển động affin của khối hiện tại có thể được xác định là mô hình chuyển động 4 thông số. Mặt khác, khi vector chuyển động affin phương thức dự báo được ứng dụng, thông tin để xác định mô hình chuyển động affin của khối hiện tại có thể được mã hóa và được báo hiệu thông qua dòng bit. Ví dụ như, khi vector chuyển động affin phương thức dự báo được ứng dụng cho khối hiện tại, mô hình chuyển động affin của khối hiện tại có thể được xác định dựa trên cờ 1 bit của 'affine_type_flag'.

Tiếp theo, vector hạt affin của khối hiện tại có thể được suy ra (S802). Khi mô hình chuyển động affin 4 thông số được chọn, các vector chuyển động tại hai điểm điều khiển của khối hiện tại có thể được suy ra. Mặt khác, khi mô hình chuyển động affin 6 thông số được chọn, các vector chuyển động tại ba điểm điều khiển của khối hiện tại có thể được suy ra. Vector chuyển động tại điểm điều khiển có thể được gọi là vector hạt affin. Điểm điều khiển có thể gồm ít nhất một trong số góc trên cùng-bên trái, góc trên cùng-bên phải, và góc dưới cùng-bên trái của khối hiện tại.

Fig.9 là hình vẽ thể hiện ví dụ về các vector hạt affin của mỗi mô hình chuyển động affin.

Trong mô hình chuyển động affin 4 thông số, các vector hạt affin có thể được suy ra cho hai trong số góc trên cùng-bên trái, góc trên cùng-bên phải, và góc dưới cùng-bên trái. Ví dụ như, như được thể hiện trong ví dụ trên Fig.9(a), khi mô hình chuyển động affin 4 thông số được chọn, vector affin có thể được suy ra sử dụng vector hạt affin sv_0 cho góc trên cùng-bên trái của khối hiện tại (ví dụ như, mẫu trên cùng-bên trái (x_1, y_1)) và vector hạt affin sv_1 cho góc trên cùng-bên phải của khối hiện tại (ví dụ như, mẫu trên cùng-bên phải (x_1, y_1)). Cũng có thể sử dụng vector hạt affin cho góc dưới cùng-bên trái thay vì vector hạt affin cho góc trên cùng-bên trái, hoặc sử dụng vector hạt affin cho góc dưới cùng-bên trái thay vì vector hạt affin cho góc trên cùng-bên phải.

Trong mô hình chuyển động affin 6 thông số, các vector hạt affin có thể được suy ra cho góc trên cùng-bên trái, góc trên cùng-bên phải, và góc dưới cùng-bên trái. Ví dụ như,

như được thể hiện trong ví dụ trên Fig.9(b), khi mô hình chuyển động affin 6 thông số được chọn, vector affin có thể được suy ra sử dụng vector hạt affin sv_0 cho góc trên cùng-bên trái của khối hiện tại (ví dụ như, mẫu trên cùng-bên trái (x_1, y_1)), vector hạt affin sv_1 cho góc trên cùng-bên phải của khối hiện tại (ví dụ như, mẫu trên cùng-bên phải (x_1, y_1)), và vector hạt affin sv_2 cho góc trên cùng-bên trái của khối hiện tại (ví dụ như, mẫu trên cùng-bên trái (x_2, y_2)).

Theo phương án được mô tả dưới đây, trong mô hình chuyển động affin 4 thông số, các vector hạt affin của điểm điều khiển trên cùng-bên trái và điểm điều khiển trên cùng-bên phải sẽ được gọi là vector hạt affin thứ nhất và vector hạt affin thứ hai, tương ứng. Trong các phương án sử dụng vector hạt affin thứ nhất và vector hạt affin thứ hai được mô tả dưới đây, ít nhất một trong số vector hạt affin thứ nhất và vector hạt affin thứ hai có thể bị thay thế bằng vector hạt affin của điểm điều khiển dưới cùng-bên trái (vector hạt affin thứ ba) hoặc vector hạt affin của điểm điều khiển dưới cùng-bên phải (vector hạt affin thứ tư).

Ngoài ra, trong mô hình chuyển động affin 6 thông số, các vector hạt affin của điểm điều khiển trên cùng-bên trái, điểm điều khiển trên cùng-bên phải, và điểm điều khiển dưới cùng-bên trái sẽ được gọi là vector hạt affin thứ nhất, vector hạt affin thứ hai, và vector hạt affin thứ ba, tương ứng. Trong các phương án sử dụng vector hạt affin thứ nhất, vector hạt affin thứ hai, và vector hạt affin thứ ba được mô tả dưới đây, ít nhất một trong số vector hạt affin thứ nhất, vector hạt affin thứ hai, và vector hạt affin thứ ba có thể bị thay thế bằng vector hạt affin của điểm điều khiển dưới cùng-bên phải (vector hạt affin thứ tư).

Vector affin có thể được suy ra cho mỗi khối con bằng cách sử dụng các vector hạt affin (S803). Ở đây, vector affin thể hiện vector chuyển động tịnh tiến được suy ra dựa trên các vector hạt affin. Vector affin của khối con có thể được gọi là khối con affin vector chuyển động hoặc vector chuyển động khối con.

Fig.10 là hình vẽ thể hiện ví dụ về các vector affin của các khối con trong mô hình chuyển động 4 thông số.

Vector affin của khối con có thể được suy ra dựa trên vị trí của điểm điều khiển, vị trí của khối con, và vector hạt affin. Ví dụ như, Phương trình 3 thể hiện ví dụ về suy ra khối con affin vector.

【Phương trình 3】

$$v_x = \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)}(y - y_0) + sv_{0x}$$

$$v_y = \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)}(x - x_0) - \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(y - y_0) + sv_{0y}$$

Trong phương trình 3, (x, y) biểu thị vị trí của khối con. Ở đây, vị trí của khối con biểu thị vị trí của mẫu tham chiếu được đưa vào khối con. Mẫu tham chiếu có thể là mẫu được bố trí ở góc trên cùng-bên trái của khối con, hoặc mẫu trong đó có ít nhất một trong số hệ tọa độ trục x và trục y là điểm tâm. (x_0, y_0) biểu thị vị trí của điểm điều khiển thứ nhất, và (sv_{0x}, sv_{0y}) biểu thị vector hạt affin thứ nhất. Ngoài ra, (x_1, y_1) biểu thị vị trí của điểm điều khiển thứ hai, và (sv_{1x}, sv_{1y}) biểu thị vector hạt affin thứ hai.

Khi điểm điều khiển thứ nhất và điểm điều khiển thứ hai tương ứng với góc trên cùng-bên trái và góc trên cùng-bên phải của khối hiện tại tương ứng, $x_1 - x_0$ có thể được đặt thành trị số bằng độ rộng của khối hiện tại.

Sau đó, dự báo bù chuyển động cho mỗi khối con có thể được thực hiện sử dụng vector affin của mỗi khối con (S804). Do thực hiện dự báo bù chuyển động, khối dự báo cho mỗi khối con có thể được tạo ra. Các khối dự báo của các khối con có thể được đặt là các khối dự báo của khối hiện tại.

Vector hạt affin của khối hiện tại có thể được suy ra dựa trên vector hạt affin của khối lân cận gần kề khối hiện tại. Khi phương thức dự báo liên khung của khối hiện tại là phương thức hợp nhất affin, vector hạt affin của ứng viên hợp nhất được đưa vào danh mục ứng viên hợp nhất có thể được xác định là vector hạt affin của khối hiện tại. Ngoài ra, khi phương thức dự báo liên khung của khối hiện tại là phương thức hợp nhất affin, thông tin chuyển động gồm ít nhất một trong số chỉ số hình ảnh tham chiếu của khối hiện tại, cờ dự báo của hướng cụ thể, và trị số định trọng số hai hướng có thể được đặt giống như hướng của ứng viên hợp nhất.

Ứng viên hợp nhất có thể được suy ra dựa trên khối lân cận của khối hiện tại. Khối lân cận có thể gồm ít nhất một trong số khối lân cận không gian gần kề khối hiện tại về không gian, và khối lân cận thời gian được đưa vào hình ảnh khác với hình ảnh hiện tại.

Fig.11 là hình vẽ thể hiện khối lân cận có thể được sử dụng để suy ra ứng viên hợp nhất.

Khối lân cận của khối hiện tại có thể gồm ít nhất một trong số khối lân cận (A) gần kề mặt trái của khối hiện tại, khối lân cận (B) gần kề đỉnh của khối hiện tại, khối lân cận (C) gần kề góc trên cùng-bên phải của khối hiện tại, khối lân cận (D) gần kề góc dưới cùng-bên trái của khối hiện tại, và khối lân cận (E) gần kề góc trên cùng-bên trái của khối hiện tại. Khi hệ tọa độ của mẫu trên cùng-bên trái của khối hiện tại là (x_0, y_0) , khối

lân cận trái A gồm mẫu tại vị trí của (x_0-1, y_0+H-1) , và khối lân cận đỉnh B gồm mẫu tại vị trí của (x_0+W-1, y_0-1) . Ở đây, W và H tương ứng biểu thị độ rộng và độ cao của khối hiện tại. Khối lân cận trên cùng-bên phải C gồm mẫu tại vị trí của (x_0+W, y_0-1) , và khối lân cận trái dưới cùng-bên trái D gồm mẫu tại vị trí của (x_0-1, y_0+H) . Khối lân cận trên cùng-bên trái E gồm mẫu tại vị trí của (x_0-1, y_0-1) .

Khi khối lân cận được mã hóa trong phương thức dự báo liên khung affine, vector hạt affine của ứng viên hợp nhất có thể được suy ra dựa trên vector hạt affine của khối lân cận. Trong nội dung dưới đây, khối lân cận được mã hóa trong phương thức dự báo liên khung affine sẽ được gọi là khối lân cận affine.

Ứng viên hợp nhất cho khối hiện tại có thể được tạo ra bằng cách tìm kiếm các khối lân cận theo thứ tự quét được xác định trước. Thứ tự quét có thể được xác định trước trong bộ mã hóa và bộ giải mã. Ví dụ như, các khối lân cận có thể được tìm kiếm theo thứ tự A, B, C, D và E. Ngoài ra, các ứng viên hợp nhất có thể lần lượt được suy ra từ các khối lân cận affine được tìm kiếm. Theo lựa chọn, thứ tự quét có thể được xác định theo kiểu thích ứng dựa trên ít nhất một trong số kích cỡ, hình dạng, và mô hình chuyển động affine của khối hiện tại. Tức là, các thứ tự quét của các khối có ít nhất một trong số kích cỡ, hình dạng, và mô hình chuyển động affine là khác có thể là khác với nhau.

Theo lựa chọn, các khối được bố trí trên đỉnh của khối hiện tại lần lượt được tìm kiếm, và ứng viên hợp nhất được suy ra từ khối lân cận affine được tìm thấy đầu tiên, và các khối được bố trí trên mặt trái của khối hiện tại lần lượt được tìm kiếm, và ứng viên hợp nhất có thể được suy ra từ khối lân cận affine được tìm thấy đầu tiên. Ở đây, các khối lân cận được bố trí trên đỉnh của khối hiện tại gồm ít nhất một trong số khối lân cận E, khối lân cận B, và khối lân cận C, và các khối được bố trí trên mặt trái của khối hiện tại có thể gồm ít nhất một trong số khối A và khối D. Tại điểm này, khối lân cận E cũng có thể được phân loại là khối được bố trí trên mặt trái của khối hiện tại.

Mặc dù không được thể hiện, ứng viên hợp nhất có thể được suy ra từ khối lân cận thời gian của khối hiện tại. Ở đây, khối lân cận thời gian có thể gồm khối tại vị trí giống như vị trí của khối hiện tại trong hình ảnh được sắp xếp hoặc khối gần kề đó. Cụ thể là, khi khối lân cận thời gian của khối hiện tại được mã hóa trong phương thức dự báo liên khung affine, ứng viên hợp nhất có thể được suy ra dựa trên vector hạt affine của ứng viên hợp nhất thời gian.

Danh mục ứng viên hợp nhất gồm các ứng viên hợp nhất có thể được tạo ra, và vector hạt affine của một trong số các ứng viên hợp nhất được đưa vào danh mục ứng viên

hợp nhất có thể được xác định là vector hạt affin của khối hiện tại. Để thực hiện được điều này, thông tin chỉ số để nhận dạng bất kỳ một trong số các ứng viên hợp nhất có thể được mã hóa và được truyền thông qua dòng bit.

Theo một ví dụ khác, các khối lân cận được tìm kiếm theo thứ tự quét, và vector hạt affin của khối hiện tại có thể được suy ra từ vector hạt affin của khối lân cận affin được tìm thấy đầu tiên.

Như được mô tả trên đây, trong phương thức hợp nhất affin, vector hạt affin của khối hiện tại có thể được suy ra sử dụng vector hạt affin của khối lân cận.

Khi phương thức dự báo liên khung của khối hiện tại là vector chuyển động affin phương thức dự báo, vector hạt affin của ứng viên dự báo vector chuyển động được đưa vào danh mục ứng viên dự báo vector chuyển động có thể được xác định là vector hạt affin trị số dự báo của khối hiện tại. Vector hạt affin của khối hiện tại có thể được suy ra bằng cách thêm vector hạt affin trị số chênh lệch vào vector hạt affin trị số dự báo.

Ứng viên dự báo vector hạt affin có thể được suy ra dựa trên khối lân cận của khối hiện tại. Cụ thể là, các khối lân cận được bố trí trên đỉnh của khối hiện tại được tìm kiếm theo thứ tự quét được xác định trước, và ứng viên dự báo vector hạt affin thứ nhất có thể được suy ra từ khối lân cận affin được tìm thấy đầu tiên. Ngoài ra, các khối lân cận được bố trí trên mặt trái của khối hiện tại được tìm kiếm theo thứ tự quét được xác định trước, và ứng viên dự báo vector hạt affin thứ hai có thể được suy ra từ khối lân cận affin được tìm thấy đầu tiên.

Thông tin để xác định vector hạt affin trị số chênh lệch có thể được mã hóa và được truyền thông qua dòng bit. Thông tin có thể gồm thông tin kích cỡ biểu thị độ lớn của vector hạt affin trị số chênh lệch, và thông tin mã biểu thị dấu hiệu của vector hạt affin trị số chênh lệch. Vector hạt affin trị số chênh lệch cho mỗi điểm điều khiển có thể được đặt là giống nhau. Theo lựa chọn, trị số chênh lệch vector tâm affin có thể được đặt khác nhau cho mỗi điểm điều khiển.

Như được mô tả trên đây, vector hạt affin của ứng viên hợp nhất hoặc ứng viên dự báo vector hạt affin có thể được suy ra từ vector hạt affin của khối lân cận affin, và vector hạt affin của khối hiện tại có thể được suy ra sử dụng vector hạt affin được suy ra của ứng viên hợp nhất hoặc ứng viên dự báo vector hạt affin. Theo lựa chọn, sau khi tìm kiếm các khối lân cận affin theo thứ tự quét được xác định trước, vector hạt affin của khối hiện tại có thể được suy ra từ vector hạt affin của khối lân cận affin được tìm thấy đầu tiên.

Trong nội dung dưới đây, phương pháp suy ra vector hạt affin của khối hiện tại, ứng

viên hợp nhất, hoặc ứng viên dự báo vector hạt affin từ vector hạt affin của khối lân cận affin sẽ được mô tả chi tiết. Trong các phương án được mô tả dưới đây, có thể hiểu rằng suy ra vector hạt affin của khối hiện tại là suy ra vector hạt affin của ứng viên hợp nhất hoặc suy ra vector hạt affin của ứng viên dự báo vector hạt affin.

Fig.12 là hình vẽ thể hiện ví dụ về suy ra vector hạt affin của khối hiện tại dựa trên vector hạt affin của khối lân cận affin.

Khi vector hạt affin thứ nhất nv_0 cho điểm điều khiển trên cùng-bên trái và vector hạt affin thứ hai nv_1 cho điểm điều khiển trên cùng-bên phải được lưu trữ cho khối lân cận affin, vector hạt affin thứ ba nv_2 cho điểm điều khiển dưới cùng-bên trái của khối lân cận affin có thể được suy ra dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai. Phương trình 4 thể hiện ví dụ về suy ra vector hạt affin thứ ba.

【Phương trình 4】

$$nv_{2x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + nv_{0x}$$

$$nv_{2y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + nv_{0y}$$

Trong phương trình 4, (nv_{0x}, nv_{0y}) biểu thị vector hạt affin thứ nhất nv_0 , (nv_{1x}, nv_{1y}) biểu thị vector hạt affin thứ hai nv_1 , và (nv_{2x}, nv_{2y}) biểu thị vector hạt affin thứ ba nv_2 . Ngoài ra, (x_{n0}, y_{n0}) biểu thị vị trí của điểm điều khiển thứ nhất, (x_{n1}, y_{n1}) biểu thị vị trí của điểm điều khiển thứ hai, và (x_{n2}, y_{n2}) biểu thị vị trí của điểm điều khiển thứ ba.

Sau đó, vector hạt affin của khối hiện tại có thể được suy ra sử dụng vector hạt affin thứ nhất, vector hạt affin thứ hai, và vector hạt affin thứ ba. Phương trình 5 thể hiện ví dụ về suy ra vector hạt affin thứ nhất v_0 của khối hiện tại, và phương trình 6 thể hiện ví dụ về suy ra vector hạt affin thứ hai v_1 của khối hiện tại.

【Phương trình 5】

$$v_{0x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_0 - x_{n0}) - \frac{(nv_{2x} - nv_{0x})}{(y_{n2} - x_{n0})} (y_0 - y_{n0}) + nv_{0x}$$

$$v_{0y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_0 - x_{n0}) - \frac{(nv_{2y} - nv_{0y})}{(y_{n2} - x_{n0})} (y_0 - y_{n0}) + nv_{0y}$$

【Phương trình 6】

$$v_{1x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_1 - x_{n0}) - \frac{(nv_{2x} - nv_{0x})}{(y_{n2} - x_{n0})} (y_1 - y_{n0}) + nv_{0x}$$

$$v_{1y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_1 - x_{n0}) - \frac{(nv_{2y} - nv_{0y})}{(y_{n2} - x_{n0})} (y_1 - y_{n0}) + nv_{0y}$$

Trong các phương trình 5 và 6, (v_{0x}, v_{0y}) biểu thị vector hạt affin thứ nhất sv_0 của khối hiện tại, và (v_{1x}, v_{1y}) biểu thị vector hạt affin thứ hai sv_1 của khối hiện tại. Ngoài ra, (x_0, y_0) biểu thị vị trí của điểm điều khiển thứ nhất, và (x_1, y_1) biểu thị vị trí của điểm điều khiển thứ hai. Ví dụ như, điểm điều khiển thứ nhất biểu thị góc trên cùng-bên trái của khối hiện tại, và điểm điều khiển thứ hai biểu thị góc trên cùng-bên phải của khối hiện tại.

Trong ví dụ được mô tả trên đây, nội dung đã được mô tả là các vector hạt affin của khối hiện tại được suy ra sử dụng ba vector hạt affin cho khối lân cận affin. Theo một ví dụ khác, các vector hạt affin của khối hiện tại có thể được suy ra chỉ sử dụng hai trong số các vector hạt affin của khối lân cận affin.

Theo lựa chọn, các vector hạt affin của khối hiện tại có thể được suy ra sử dụng vector hạt affin thứ tư cho góc dưới cùng-bên phải, thay vì vector hạt affin thứ nhất của góc trên cùng-bên trái, vector hạt affin thứ hai của góc trên cùng-bên phải, hoặc vector hạt affin thứ ba của góc dưới cùng-bên trái của khối lân cận affin.

Cụ thể là, khi ranh giới đỉnh của khối hiện tại tiếp xúc với ranh giới đỉnh của đơn vị cây mã hóa, và điều được mong muốn là sử dụng vector hạt affin cho điểm điều khiển đỉnh (ví dụ như, góc trên cùng-bên trái hoặc góc trên cùng-bên phải) của khối lân cận affin gần kề đỉnh của khối hiện tại (trong phần dưới đây, được gọi là khối lân cận affin đỉnh), các khối này nên được lưu trữ trước trong bộ nhớ, và vấn đề tăng về số lượng bộ đệm đường dây có thể xảy ra. Do đó, khi ranh giới đỉnh của khối hiện tại tiếp xúc với ranh giới đỉnh của đơn vị cây mã hóa, nó có thể được đặt để sử dụng vector hạt affin của điểm điều khiển đáy (ví dụ như, góc dưới cùng-bên trái hoặc góc dưới cùng-bên phải), thay vì vector hạt affin của điểm điều khiển đỉnh, cho khối lân cận affin đỉnh. Ví dụ như, các vector hạt affin của khối hiện tại có thể được suy ra sử dụng vector hạt affin thứ ba cho góc dưới cùng-bên trái và vector hạt affin thứ tư cho góc dưới cùng-bên phải của khối lân cận affin đỉnh. Tại điểm này, các vector hạt affin cho các góc đáy có thể được sao chép từ các vector hạt affin cho các góc đỉnh hoặc được suy ra từ các vector hạt affin cho các góc đỉnh. Ví dụ như, vector hạt affin thứ nhất, vector hạt affin thứ hai, hoặc vector hạt affin thứ ba có thể được chuyển hóa thành hoặc thay thế cho vector hạt affin thứ tư cho góc dưới cùng-bên phải.

Các phương trình 7 và 8 thể hiện ví dụ về suy ra vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại sử dụng vector hạt affin thứ ba cho điểm điều khiển dưới cùng-bên trái và vector hạt affin thứ tư cho điểm điều khiển dưới cùng-bên phải của

vector affin lân cận.

【Phương trình 7】

$$v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y}$$

【Phương trình 8】

$$v_{1x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y}$$

Trong các phương trình 7 và 8, (x_{n2}, y_{n2}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên trái của khối lân cận affin, và (x_{n3}, y_{n3}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên phải của khối lân cận affin. (x_0, y_0) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên trái của khối hiện tại, và (x_1, y_1) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên phải của khối hiện tại. (nv_{2x}, nv_{2y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên trái của khối lân cận affin (tức là, vector hạt affin thứ ba), và (nv_{3x}, nv_{3y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên phải của khối lân cận affin (tức là, vector hạt affin thứ tư). (v_{0x}, v_{0y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên trái của khối hiện tại (tức là, vector hạt affin thứ nhất), và (v_{1x}, v_{1y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên phải của khối hiện tại (tức là, vector hạt affin thứ hai).

Phép tính chia được đưa vào các phương trình 7 và 8 có thể bị thay đổi thành phép toán dịch chuyển. Phép toán dịch chuyển có thể được thực hiện dựa trên trị số được suy ra bởi độ rộng giữa điểm điều khiển dưới cùng-bên trái và điểm điều khiển dưới cùng-bên phải (tức là, $(x_{n3} - x_{n2})$).

Như được thể hiện trong ví dụ được mô tả trên đây, các vector hạt affin của khối hiện tại có thể được suy ra từ các vector hạt affin của khối lân cận affin đã được mã hóa/giải mã. Để thực hiện được điều này, các vector hạt affin của khối lân cận affin đã được mã hóa/giải mã nên được lưu trữ trong bộ nhớ. Tuy nhiên, do các vector hạt affin của khối lân cận affin được lưu trữ trong bộ nhớ ngoài các vector chuyển động tịnh tiến (tức là, các vector affin) của các khối con được đưa vào khối lân cận affin ra, có vấn đề tăng việc sử dụng bộ nhớ. Để giải quyết vấn đề này, vector hạt affin của khối hiện tại có

thể được suy ra sử dụng vector chuyển động của khối con gần kề điểm điều khiển của khối lân cận affin, thay vì vector hạt affin của khối lân cận affin. Tức là, vector chuyển động của khối con gần kề điểm điều khiển của khối lân cận affin có thể được đặt là vector hạt affin của khối lân cận affin. Ở đây, khối con là khối có kích cỡ/hình dạng được xác định trước trong bộ mã hóa và bộ giải mã, và có thể là khối có kích cỡ/hình dạng cơ bản để lưu trữ vector chuyển động. Ví dụ như, khối con có thể là khối hình vuông có kích cỡ 4×4 . Theo lựa chọn, vector chuyển động cho vị trí mẫu cụ thể có thể được đặt là vector hạt affin của khối lân cận affin.

Fig.13 là hình vẽ thể hiện ví dụ về việc đặt vector chuyển động của khối con là vector hạt affin của khối lân cận affin.

Vector chuyển động của khối con gần kề điểm điều khiển có thể được đặt là vector hạt affin của điểm điều khiển. Ví dụ như, trong ví dụ được thể hiện trên Fig.13, vector chuyển động (nv_{4x} , nv_{4y}) của khối con (khối con dưới cùng-bên trái) gần kề góc dưới cùng-bên trái của khối lân cận affin có thể được đặt là vector hạt affin (nv_{2x} , nv_{2y}) của điểm điều khiển dưới cùng-bên trái, và vector chuyển động (nv_{5x} , nv_{5y}) của khối con (khối con dưới cùng-bên phải) gần kề góc dưới cùng-bên phải có thể được đặt là vector hạt affin (nv_{3x} , nv_{3y}) của điểm điều khiển ở góc đáy- phải. Ở đây, khối con dưới cùng-bên trái có thể có nghĩa là khối con gồm mẫu (ví dụ như, mẫu tại vị trí của $(x_{n2}$, y_{n2-1})) gần kề điểm điều khiển dưới cùng-bên trái $(x_{n2}$, y_{n2}) trong khối affin lân cận, và khối con dưới cùng-bên phải có thể có nghĩa là khối gồm mẫu (ví dụ như, mẫu tại vị trí của $(x_{n3-1}$, y_{n3-1})) gần kề điểm điều khiển dưới cùng-bên phải $(x_{n3}$, y_{n3}) trong khối affin lân cận. Khi các vector hạt affin của khối hiện tại được suy ra dựa trên các phương trình 7 và 8, vector hạt affin thứ ba của khối lân cận affin có thể được thay thế bằng vector chuyển động của khối con dưới cùng-bên trái, và vector hạt affin thứ tư có thể được thay thế bằng vector chuyển động của khối con dưới cùng-bên phải.

Trong nội dung dưới đây, theo phương án được mô tả dưới đây, khối con được sử dụng làm vector hạt affin của khối lân cận affin sẽ được gọi là khối con affin.

Theo phương án của sáng chế, khối con affin có thể được xác định dựa trên mẫu tại vị trí cụ thể. Ví dụ như, khối con gồm mẫu tại vị trí cụ thể có thể được đặt là khối con affin. Trong phần dưới đây, mẫu tại vị trí cụ thể sẽ được gọi là mẫu tham chiếu affin. Ngoài ra, mẫu tham chiếu được sử dụng để xác định khối con affin của điểm điều khiển dưới cùng-bên trái được gọi là mẫu tham chiếu dưới cùng-bên trái, và mẫu tham chiếu được sử dụng để xác định khối con affin của điểm điều khiển dưới cùng-bên phải được

gọi là mẫu tham chiếu dưới cùng-bên phải.

Mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải có thể được chọn từ các mẫu được đưa vào khối lân cận affin. Ví dụ như, ít nhất một trong số mẫu trên cùng-bên trái, mẫu dưới cùng-bên trái, mẫu trên cùng-bên phải, và mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu dưới cùng-bên trái, và ít nhất một trong số mẫu trên cùng-bên trái, mẫu dưới cùng-bên trái, mẫu trên cùng-bên phải, và mẫu dưới cùng-bên trái của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu dưới cùng-bên phải. Do đó, các vectơ chuyển động của khối con dưới cùng-bên trái gồm mẫu tham chiếu dưới cùng-bên trái và khối con dưới cùng-bên phải gồm mẫu tham chiếu dưới cùng-bên phải có thể được đặt tương ứng là vectơ hạt affin cho điểm điều khiển dưới cùng-bên trái và vectơ hạt affin cho điểm điều khiển dưới cùng-bên phải.

Theo một ví dụ khác, ít nhất một trong số mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải có thể được đặt là mẫu được bố trí bên ngoài khối lân cận affin. Nội dung này sẽ được mô tả chi tiết kết hợp với các hình vẽ từ Fig.14 đến Fig.16.

Fig.14 đến Fig.16 là các hình vẽ thể hiện các vị trí của các mẫu tham chiếu.

Như được thể hiện trong ví dụ trên Fig.14(a), mẫu trên cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều khiển dưới cùng-bên trái. Do đó, khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu được bố trí trên mặt phải của mẫu trên cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con gần kề mặt phải của khối con dưới cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Theo lựa chọn, như được thể hiện trong ví dụ trên Fig.14(b), mẫu được bố trí trên mặt trái của mẫu trên cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều khiển dưới cùng-bên trái. Do đó, khối con gần kề mặt trái của khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu trên cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con dưới

cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Theo lựa chọn, như được thể hiện trong ví dụ trên Fig.15(a), mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều khiển dưới cùng-bên trái. Do đó, khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu được bố trí trên mặt phải của mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con gần kề mặt phải của khối con dưới cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Theo lựa chọn, như được thể hiện trong ví dụ trên Fig.15(b), mẫu được bố trí trên mặt trái của mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều khiển dưới cùng-bên trái. Do đó, khối con gần kề mặt trái của khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con dưới cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Theo lựa chọn, như được thể hiện trong ví dụ trên Fig.16(a), mẫu được bố trí giữa mẫu trên cùng-bên trái và mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái (ví dụ như, mẫu giữa-trái) có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều khiển dưới cùng-bên trái. Do đó, khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu được bố trí trên mặt phải của mẫu được bố trí giữa mẫu trên cùng-bên phải và mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải (ví dụ như, mẫu giữa-phải) có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con gần kề mặt phải của khối con dưới cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Theo lựa chọn, như được thể hiện trong ví dụ trên Fig.16 (b), mẫu được bố trí trên mặt trái của mẫu được bố trí giữa mẫu trên cùng-bên trái và mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}) cho điểm điều

khối dưới cùng-bên trái. Do đó, khối con gần kề mặt trái của khối con dưới cùng-bên trái gồm mẫu tham chiếu (x_{n4} và y_{n4}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên trái.

Mẫu được bố trí giữa mẫu trên cùng-bên phải và mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}) cho điểm điều khiển dưới cùng-bên phải. Do đó, khối con dưới cùng-bên phải gồm mẫu tham chiếu (x_{n5} và y_{n5}) có thể được đặt là khối con affin cho điểm điều khiển dưới cùng-bên phải.

Khi các vector hạt affin của khối hiện tại được suy ra dựa trên các phương trình 7 và 8, vector hạt affin thứ ba của khối lân cận affin có thể được thay thế bằng vector chuyển động của khối con affin cho điểm điều khiển dưới cùng-bên trái, và vector hạt affin thứ tư có thể được thay thế bằng vector chuyển động của khối con affin cho điểm điều khiển dưới cùng-bên phải. Ngoài ra, vị trí của điểm điều khiển dưới cùng-bên trái có thể được thay thế bằng vị trí của mẫu tham chiếu dưới cùng-bên trái, và vị trí của điểm điều khiển dưới cùng-bên phải có thể được thay thế bằng vị trí của mẫu tham chiếu dưới cùng-bên phải.

Không giống như các phương án được mô tả được thể hiện trên các Fig.14 đến Fig.16, khối con gồm mẫu gần kề mẫu tham chiếu có thể được đặt là khối con affin. Cụ thể là, mẫu được bố trí bên ngoài khối con lân cận affin có thể được đặt là mẫu tham chiếu, và khối con được đưa vào khối lân cận affin có thể được đặt là khối con affin. Ví dụ như, trong ví dụ được thể hiện trên Fig.14 (a), mẫu được bố trí trên mặt phải của mẫu trên cùng-bên phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}), và khối con dưới cùng-bên phải có thể được đặt là khối con affin cho góc dưới cùng-bên phải. Theo lựa chọn, trong ví dụ được thể hiện trên Fig.14 (b), mẫu được bố trí ở bên trái của mẫu trên cùng-bên trái của khối con dưới cùng-bên trái có thể được đặt là mẫu tham chiếu (x_{n4} , y_{n4}), và khối con dưới cùng-bên trái có thể được đặt là khối con affin cho góc dưới cùng-bên trái.

Các phương án được mô tả trên đây có thể được ứng dụng như nhau cho các Fig.15 và Fig.16. Tức là, trong ví dụ được thể hiện trên Fig.15 (a) hoặc Fig.16 (a), mẫu được bố trí trên mặt phải của mẫu dưới cùng-bên phải hoặc mẫu giữa-phải của khối con dưới cùng-bên phải có thể được đặt là mẫu tham chiếu (x_{n5} , y_{n5}), và khối con dưới cùng-bên phải có thể được đặt là khối con affin cho góc dưới cùng-bên phải. Theo lựa chọn, trong ví dụ được thể hiện trên Fig.15 (b) hoặc Fig.16 (b), mẫu được bố trí trên mặt trái của mẫu dưới cùng-bên trái hoặc mẫu giữa-trái của khối con dưới cùng-bên trái có thể được đặt là

mẫu tham chiếu (x_{n4}, y_{n4}) , và khối con dưới cùng-bên trái có thể được đặt là khối con affin cho góc dưới cùng-bên trái.

Như được thể hiện trong các ví dụ được mô tả trên đây, vector hạt affin của khối lân cận affin có thể được suy ra sử dụng vector chuyển động của khối con affin. Để thực hiện được điều này, trong trường hợp khối đã được mã hóa/giải mã, vector chuyển động có thể được lưu trữ bằng đơn vị của khối con.

Theo một ví dụ khác, sau khi lưu trữ số lượng tối thiểu các vector hạt affin cho khối lân cận affin, vector chuyển động của khối con affin có thể được suy ra sử dụng các vector hạt affin được lưu trữ.

Các phương trình 9 và 10 thể hiện ví dụ về suy ra vector chuyển động của khối con affin sử dụng các vector hạt affin của khối lân cận affin.

【Phương trình 9】

$$nv_{4x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0x}$$

$$nv_{4y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0y}$$

【Phương trình 10】

$$nv_{5x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0x}$$

$$nv_{5y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0y}$$

Trong các phương trình 9 và 10, (nv_{4x}, nv_{4y}) biểu thị vector chuyển động của khối con affin cho điểm điều khiển dưới cùng-bên trái, và (nv_{5x}, nv_{5y}) biểu thị vector chuyển động của khối con affin cho điểm điều khiển dưới cùng-bên phải. Do vector chuyển động của khối con affin và vector hạt affin của điểm điều khiển được đặt ngang bằng nhau, (nv_{4x}, nv_{4y}) có thể được thay thế bằng vector hạt affin (nv_{2x}, nv_{2y}) cho điểm điều khiển dưới cùng-bên trái, hoặc (nv_{5x}, nv_{5y}) có thể được thay thế bằng vector hạt affin (nv_{3x}, nv_{3y}) cho điểm điều khiển dưới cùng-bên phải.

(x_{n4}, y_{n4}) biểu thị vị trí của mẫu tham chiếu cho khối con dưới cùng-bên trái. Theo lựa chọn, vị trí trung tâm của khối con dưới cùng-bên trái hoặc vị trí của điểm điều khiển dưới cùng-bên trái có thể được sử dụng thay vì vị trí này. (x_{n5}, y_{n5}) biểu thị vị trí của mẫu tham chiếu cho khối con dưới cùng-bên phải. Theo lựa chọn, vị trí trung tâm của khối con dưới cùng-bên phải hoặc vị trí của điểm điều khiển dưới cùng-bên phải có thể được

sử dụng thay vì vị trí này.

Các phương trình 9 và 10 có thể được ứng dụng khi khối hiện tại không tiếp xúc với ranh giới của đơn vị cây mã hóa. Khi khối hiện tại tiếp xúc với ranh giới đỉnh của đơn vị cây mã hóa, thay vì sử dụng các phương trình 9 và 10, vector chuyển động tịnh tiến của khối con affin được xác định dựa trên mẫu tham chiếu dưới cùng-bên trái có thể được đặt là vector hạt affin thứ ba, và vector chuyển động tịnh tiến của khối con affin được xác định dựa trên mẫu tham chiếu dưới cùng-bên phải có thể được đặt là vector hạt affin thứ tư.

Trong các phương trình 7 và 8, $(x_{n3}-x_{n2})$ biểu thị độ rộng giữa điểm điều khiển dưới cùng-bên trái và điểm điều khiển dưới cùng-bên phải. Như được mô tả trên đây, x_{n3} có thể được thay thế bằng vị trí x_{n5} của mẫu tham chiếu dưới cùng-bên phải, và x_{n2} có thể được thay thế bằng vị trí x_{n4} của mẫu tham chiếu dưới cùng-bên trái. Trong phần dưới đây, $(x_{n3}-x_{n2})$ hoặc trị số thu được bằng cách thay thế các vị trí của phương trình nêu trên với các vị trí của các mẫu tham chiếu (ví dụ như, $(x_{n5}-x_{n4})$) được xác định là biến số W_{seed} , và biến số được gọi là độ rộng vector hạt con.

Theo các vị trí của các mẫu tham chiếu, trường hợp mà độ rộng vector hạt con không phải là chuỗi lũy thừa của 2 (ví dụ như, 2^n) có thể xảy ra. Ví dụ như, khi mẫu dưới cùng-bên trái của khối con dưới cùng-bên trái được đặt là mẫu tham chiếu dưới cùng-bên trái, và mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải được đặt là mẫu tham chiếu dưới cùng-bên phải, độ rộng vector hạt con không phải là bội số của 2. Như được mô tả trên đây, khi độ rộng vector hạt con không phải là chuỗi lũy thừa của 2, độ rộng vector hạt con có thể được biến đổi là chuỗi lũy thừa của 2. Việc biến đổi có thể gồm cộng hoặc trừ độ lệch vào hoặc từ độ rộng vector hạt con, hoặc sử dụng vị trí của mẫu gần kề mẫu tham chiếu thay vì vị trí của mẫu tham chiếu. Ví dụ như, độ rộng vector hạt con được biến đổi có thể được suy ra bằng cách thêm 1 vào độ rộng giữa mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải. Theo lựa chọn, độ rộng giữa mẫu tham chiếu lân cận gần kề mặt phải của mẫu tham chiếu dưới cùng-bên phải và mẫu tham chiếu dưới cùng-bên trái có thể được đặt là độ rộng vector hạt con được biến đổi. Sau đó, vector hạt affin của khối hiện tại có thể được suy ra bằng cách ứng dụng độ rộng vector hạt con được biến đổi thành các phương trình 7 và 8.

Phép tính chia được đưa vào các phương trình 7 và 8 có thể bị thay đổi thành phép toán dịch chuyển. Phép toán dịch chuyển có thể được thực hiện dựa trên trị số được suy ra bởi độ rộng vector hạt con được biến đổi (tức là, trị số được biểu hiện là chuỗi lũy thừa của 2).

Khi mẫu tham chiếu được sử dụng để xác định khối con affin không thuộc về khối lân cận affin, vector hạt affin của khối lân cận affin có thể được suy ra dựa trên ví dụ gần kề mẫu tham chiếu trong số các mẫu được đưa vào khối lân cận affin. Cụ thể là, vector chuyển động tịnh tiến của khối con gồm mẫu gần kề mẫu tham chiếu trong khối lân cận affin (trong phần dưới đây, được gọi là mẫu tham chiếu lân cận) có thể được đặt là vector hạt affin của khối lân cận affin. Bằng cách này, phương pháp suy ra vector hạt affin sử dụng mẫu tham chiếu lân cận có thể được xác định là phương pháp suy ra vector hợp nhất affin biến đổi.

Fig.17 là hình vẽ thể hiện ví dụ về việc ứng dụng phương pháp suy ra vector hợp nhất affin biến đổi.

Khi mẫu tham chiếu dưới cùng-bên phải (x_{n5} , y_{n5}) của khối lân cận affin E không thuộc về khối lân cận affin, vector hạt affin có thể được suy ra dựa trên ví dụ (x_{n5-1} , y_{n5}) gần kề mặt trái của mẫu tham chiếu dưới cùng-bên phải trong số các mẫu được đưa vào khối lân cận affin. Cụ thể là, vector chuyển động tịnh tiến của khối con gồm mẫu tham chiếu lân cận (x_{n5-1} , y_{n5}) có thể được đặt là vector hạt affin của điểm điều khiển dưới cùng-bên phải.

Trong ví dụ được thể hiện trên Fig.17, điều được thể hiện là mẫu gần kề mặt phải của mẫu trên cùng-bên phải của khối con dưới cùng-bên phải là mẫu tham chiếu dưới cùng-bên phải. Ngay cả khi mẫu gần kề mặt phải của mẫu dưới cùng-bên phải của khối con dưới cùng-bên phải hoặc mẫu gần kề mặt phải của mẫu giữa-phải của khối con dưới cùng-bên phải được đặt là mẫu tham chiếu dưới cùng-bên phải, vector hạt affin có thể được suy ra dựa trên ví dụ gần kề mặt trái của mẫu tham chiếu lân cận.

Ngoài ra, ngay cả khi mẫu tham chiếu dưới cùng-bên trái không thuộc về khối lân cận affin, theo phương án được mô tả trên đây, vector hạt affin có thể được suy ra dựa trên ví dụ gần kề mặt phải của mẫu tham chiếu dưới cùng-bên trái.

Do vị trí của mẫu tham chiếu và khối con để suy ra vector hạt affin được đặt khác nhau, độ rộng vector hạt con có thể được đặt thành chuỗi lũy thừa của 2.

Ứng viên hợp nhất, ứng viên dự báo vector hạt affin, hoặc vector hạt affin của khối hiện tại có thể được suy ra sử dụng khối lân cận đã không được mã hóa trong phương thức liên khung affin trong miền lân cận của khối hiện tại. Cụ thể là, các khối không được mã hóa trong phương thức liên khung affin có thể được kết hợp, và sự kết hợp của chúng có thể được đặt là ứng viên hợp nhất hoặc ứng viên dự báo vector hạt affin. Ví dụ như, sự kết hợp của ít nhất một trong số vector chuyển động của bất kỳ một trong số các

khối lân cận gần kề góc trên cùng-bên trái của khối hiện tại, vector chuyển động của bất kỳ một trong số các khối lân cận gần kề góc trên cùng-bên phải của khối hiện tại, và vector chuyển động của bất kỳ một trong số các khối lân cận gần kề góc dưới cùng-bên trái của khối hiện tại có thể được đặt là ứng viên hợp nhất hoặc ứng viên dự báo vector hạt affin. Tại điểm này, vector chuyển động của khối lân cận gần kề góc trên cùng-bên trái, vector chuyển động của khối lân cận gần kề góc trên cùng-bên phải, và vector chuyển động của khối lân cận gần kề góc dưới cùng-bên trái có thể được đặt tương ứng là vector hạt affin thứ nhất cho điểm điều khiển trên cùng-bên trái, vector hạt affin thứ hai cho điểm điều khiển trên cùng-bên phải, và vector hạt affin thứ ba cho điểm điều khiển dưới cùng-bên trái.

Theo lựa chọn, ứng viên hợp nhất, ứng viên dự báo vector hạt affin, hoặc vector hạt affin của khối hiện tại có thể được suy ra sử dụng khối lân cận không được mã hóa trong phương thức liên khung affin trong phương pháp suy ra vector hợp nhất affin biến đổi được mô tả trên đây. Trong phần dưới đây, khối lân cận không được mã hóa trong phương thức liên khung affin sẽ được gọi là khối lân cận không phải affin.

Fig.18 là hình vẽ thể hiện ví dụ về suy ra vector hạt affin của khối hiện tại dựa trên khối lân cận không phải affin.

Trong ví dụ được thể hiện trên Fig.18, giả sử rằng tất cả các khối lân cận gần kề khối hiện tại là các khối lân cận không phải affin.

Khi mong muốn suy ra vector hạt affin của khối hiện tại từ khối lân cận không phải affin A trong số các khối lân cận gần kề khối hiện tại, mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải có thể được đặt cho khối A. Ví dụ như, mẫu gần kề mặt trái của mẫu dưới cùng-bên trái của khối A có thể được đặt là mẫu tham chiếu dưới cùng-bên trái, và mẫu dưới cùng-bên phải của khối A có thể được đặt là mẫu tham chiếu dưới cùng-bên phải. Do mẫu tham chiếu dưới cùng-bên trái không ở trong khối A, vector chuyển động của khối con gồm mẫu gần kề mặt phải của mẫu tham chiếu dưới cùng-bên trái có thể được đặt là vector hạt affin thứ ba của khối A. Ngoài ra, vector chuyển động của khối con gồm mẫu tham chiếu dưới cùng-bên phải có thể được đặt là vector hạt affin thứ tư của khối A. Sau đó, vector hạt affin thứ nhất và vector hạt affin thứ hai cho khối hiện tại có thể được suy ra từ khối A dựa trên các phương trình 9 và 10.

Phương pháp suy ra vector hạt affin từ khối lân cận không phải affin có thể được sử dụng chỉ khi dự báo bù chuyển động cho khối lân cận không phải affin được thực hiện bằng đơn vị của khối con. Ở đây, kỹ thuật dự báo để thực hiện dự báo bù chuyển động

bằng đơn vị của khối con có thể gồm ít nhất một trong số STMVP, ATMVP, luồng quang học hai hướng (Bi-Directional Optical Flow - BIO), bù chuyển động của khối xếp chồng (Overlapped Block Motion Compensation - OBMC), và lọc vector chuyển động phía bộ giải mã (Decoder side Motion Vector Refinement - DMVR).

Theo phương án được mô tả trên đây, nội dung đã được mô tả là khi ranh giới đỉnh của khối hiện tại tiếp xúc với ranh giới của đơn vị cây mã hóa, ứng viên hợp nhất, ứng viên dự báo vector hạt affin, hoặc vector hạt affin của khối hiện tại được suy ra sử dụng vector hạt affin thứ ba cho điểm điều khiển dưới cùng-bên trái và vector hạt affin thứ tư cho điểm điều khiển dưới cùng-bên phải của khối lân cận affin được bố trí trên đỉnh của khối hiện tại.

Theo một ví dụ khác, khi ranh giới đỉnh của khối hiện tại tiếp xúc với ranh giới của đơn vị cây mã hóa, và khối lân cận được bố trí trên đỉnh của khối hiện tại thuộc về đơn vị cây mã hóa khác với đơn vị cây mã hóa của khối hiện tại, ứng viên hợp nhất, ứng viên dự báo vector hạt affin, hoặc vector hạt affin của khối hiện tại có thể được suy ra sử dụng khối lân cận gần nhất với khối lân cận trong số các khối được đưa vào đơn vị cây mã hóa mà khối hiện tại thuộc về, thay vì khối lân cận.

Trong ví dụ được thể hiện trên Fig.19, điều được thể hiện là khối hiện tại tiếp xúc với ranh giới đỉnh của đơn vị cây mã hóa, và các khối B, C và E được bố trí trên đỉnh của khối hiện tại thuộc về đơn vị cây mã hóa khác với đơn vị cây mã hóa của khối hiện tại. Do đó, thay vì sử dụng khối E, vector hạt affin của khối hiện tại có thể được suy ra sử dụng khối F gần kề khối E trong số các khối được đưa vào đơn vị cây mã hóa mà khối hiện tại thuộc về.

Vì lợi ích dự báo bù chuyển động của khối hiện tại, các vector hạt affin của nhiều khối có thể được sử dụng. Ví dụ như, có thể chọn nhiều ứng viên hợp nhất từ danh mục ứng viên hợp nhất, và suy ra vector hạt affin hoặc vector khối con cho khối hiện tại dựa trên các vector hạt affin của các ứng viên hợp nhất được chọn. Việc mã hóa/giải mã khối hiện tại sử dụng các vector hạt affin của nhiều khối có thể được gọi là phương pháp mã hóa hợp nhất nhiều affin.

Thông tin biểu thị việc liệu có hay không phương pháp mã hóa hợp nhất nhiều affin được ứng dụng cho khối hiện tại có thể được mã hóa và được truyền thông qua dòng bit. Theo lựa chọn, việc liệu hoặc không ứng dụng phương pháp mã hóa hợp nhất nhiều affin cho khối hiện tại có thể được xác định dựa trên ít nhất một trong số số lượng các khối lân cận affin trong số các khối lân cận gần kề khối hiện tại, số lượng các ứng viên hợp nhất

được đưa vào danh mục ứng viên hợp nhất, và mô hình chuyển động affin của khối hiện tại.

Fig.20 và Fig.21 là các hình vẽ thể hiện luồng của phương pháp dự báo bù chuyển động sử dụng nhiều ứng viên hợp nhất.

Fig.20 là hình vẽ thể hiện ví dụ về suy ra các vector hạt affin của khối hiện tại sử dụng các vector hạt affin của nhiều ứng viên hợp nhất. Fig.21 là hình vẽ thể hiện ví dụ về suy ra vector chuyển động của mỗi khối con sử dụng các vector hạt affin của nhiều ứng viên hợp nhất.

Các vector hạt affin của khối hiện tại có thể được tạo ra dựa trên tổng, hiệu số, trị số trung bình, hoặc phép tính tổng có trọng số của các vector hạt affin của hai ứng viên hợp nhất.

Các phương trình 11 và 12 thể hiện ví dụ về suy ra vector hạt affin của khối hiện tại thông qua việc tính tổng các vector hạt affin của các ứng viên hợp nhất.

【Phương trình 11】

$$(sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + (sv_{2x}, sv_{2y})$$

【Phương trình 12】

$$(sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + (sv_{3x}, sv_{3y})$$

Trong các phương trình 11 và 12, sv_4 biểu thị vector hạt affin thứ nhất của khối hiện tại, sv_0 biểu thị vector hạt affin thứ nhất của ứng viên hợp nhất thứ nhất, và sv_2 biểu thị vector hạt affin thứ nhất của ứng viên hợp nhất thứ hai. Ngoài ra, sv_5 biểu thị vector hạt affin thứ hai của khối hiện tại, sv_1 biểu thị vector hạt affin thứ hai của ứng viên hợp nhất thứ nhất, và sv_3 biểu thị vector hạt affin thứ hai của ứng viên hợp nhất thứ hai.

Ngoài ra, các phương trình 13 và 14 thể hiện ví dụ về suy ra vector hạt affin của khối hiện tại thông qua phép tính tổng có trọng số của các vector hạt affin của các ứng viên hợp nhất.

【Phương trình 13】

$$(sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + ((sv_{2x}, sv_{2y}) \gg 1)$$

【Phương trình 14】

$$(sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + ((sv_{3x}, sv_{3y}) \gg 1)$$

Theo một ví dụ khác, khối con thứ nhất vector chuyển động và khối con thứ hai vector chuyển động có thể được tạo ra cho mỗi khối con trong khối hiện tại, dựa trên các vector hạt affin của ứng viên hợp nhất thứ nhất và các vector hạt affin của ứng viên hợp

nhất thứ hai. Sau đó, vector chuyển động khối con cuối cùng có thể được tạo ra dựa trên tổng, hiệu số, trị số trung bình, hoặc phép tính tổng có trọng số của khối con thứ nhất vector chuyển động và khối con thứ hai vector chuyển động.

Phương trình 15 thể hiện ví dụ về suy ra vector chuyển động khối con cuối cùng thông qua phép tính tổng khối con thứ nhất vector chuyển động và khối con thứ hai vector chuyển động.

【Phương trình 15】

$$(V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + (V_{1x}, V_{1y})$$

Trong phương trình 15, V_0 biểu thị khối con thứ nhất vector chuyển động, V_1 biểu thị khối con thứ hai vector chuyển động, và V_2 biểu thị vector chuyển động khối con cuối cùng.

Ngoài ra, phương trình 16 thể hiện ví dụ về suy ra vector chuyển động khối con cuối cùng thông qua phép tính tổng có trọng số của khối con thứ nhất vector chuyển động và khối con thứ hai vector chuyển động.

【Phương trình 16】

$$(V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + ((V_{1x}, V_{1y}) \gg 1)$$

Dự báo nội khung cho dự báo khối hiện tại sử dụng các mẫu được tái cấu trúc đã được mã hóa/giải mã quanh khối hiện tại. Tại điểm này, các mẫu được tái cấu trúc trước khi bộ lọc trong vòng lặp được ứng dụng có thể được sử dụng để dự báo nội khung của khối hiện tại.

Kỹ thuật dự báo nội khung gồm dự báo nội khung dựa trên ma trận, và dự báo nội khung chung xét đến hướng so với các mẫu được tái cấu trúc lân cận. Thông tin biểu thị kỹ thuật dự báo nội khung của khối hiện tại có thể được báo hiệu thông qua dòng bit. Thông tin có thể là cờ 1 bit. Theo lựa chọn, kỹ thuật dự báo nội khung của khối hiện tại có thể được xác định dựa trên ít nhất một trong số vị trí, kích cỡ, và hình dạng của khối hiện tại, hoặc dựa trên kỹ thuật dự báo nội khung của khối lân cận. Ví dụ như, khi khối hiện tại tồn tại qua ranh giới hình ảnh, có thể đặt không ứng dụng dự báo nội khung dựa trên ma trận dự báo nội khung đến khối hiện tại.

Dự báo nội khung dựa trên ma trận dự báo nội khung là phương pháp thu được khối dự báo của khối hiện tại bởi bộ mã hóa và bộ giải mã dựa trên sản phẩm ma trận giữa ma trận được lưu trữ trước và các mẫu được tái cấu trúc quanh khối hiện tại. Thông tin để định rõ bất kỳ một trong số nhiều ma trận được lưu trữ trước có thể được báo hiệu thông

qua dòng bit. Bộ giải mã có thể xác định ma trận để dự báo nội khung khối hiện tại dựa trên thông tin và kích cỡ của khối hiện tại.

Dự báo nội khung chung là phương pháp thu được khối dự báo cho khối hiện tại dựa trên phương thức dự báo không góc nội khung hoặc phương thức dự báo góc nội khung. Trong phần dưới đây, quy trình thực hiện dự báo nội khung dựa trên dự báo nội khung chung sẽ được mô tả chi tiết hơn kết hợp với các hình vẽ.

Fig.22 là lưu đồ minh họa phương pháp dự báo nội khung theo phương án của sáng chế.

Đường mẫu tham chiếu của khối hiện tại có thể được xác định (S2201). Đường mẫu tham chiếu nghĩa là tập hợp các mẫu tham chiếu được đưa vào đường thứ k ngoài mặt đỉnh và/hoặc mặt trái của khối hiện tại ra. Các mẫu tham chiếu có thể được suy ra từ các mẫu được tái cấu trúc đã được mã hóa/giải mã quanh khối hiện tại.

Thông tin chỉ số để nhận dạng đường mẫu tham chiếu của khối hiện tại trong số nhiều đường mẫu tham chiếu có thể được báo hiệu thông qua dòng bit. Nhiều đường mẫu tham chiếu có thể gồm ít nhất một trong số đường thứ nhất, đường thứ hai, đường thứ ba, và đường thứ tư trên mặt đỉnh và/hoặc mặt trái của khối hiện tại. Bảng 1 thể hiện các chỉ số được gán tương ứng cho các đường mẫu tham chiếu. Trong bảng 1, giả sử rằng đường thứ nhất, đường thứ hai, và đường thứ tư được sử dụng là các ứng viên đường mẫu tham chiếu.

【Bảng 1】

Chỉ số	Các đường mẫu tham chiếu
0	Đường mẫu tham chiếu thứ nhất
1	Đường mẫu tham chiếu thứ hai
2	Đường mẫu tham chiếu thứ tư

Đường mẫu tham chiếu của khối hiện tại có thể được xác định dựa trên ít nhất một trong số vị trí, kích cỡ, và hình dạng của khối hiện tại, hoặc dựa trên phương thức mã hóa dự báo của khối lân cận. Ví dụ như, khi khối hiện tại tiếp xúc với ranh giới của hình ảnh, ngói, phiến, hoặc đơn vị cây mã hóa, đường mẫu tham chiếu thứ nhất có thể được xác định là đường mẫu tham chiếu của khối hiện tại.

Đường mẫu tham chiếu có thể gồm các mẫu tham chiếu đỉnh được bố trí trên đỉnh của khối hiện tại và các mẫu tham chiếu trái được bố trí trên mặt trái của khối hiện tại. Các mẫu tham chiếu đỉnh và các mẫu tham chiếu bên trái có thể được suy ra từ các mẫu được tái cấu trúc quanh khối hiện tại. Các mẫu được tái cấu trúc có thể ở trạng thái trước

khi bộ lọc trong vòng lặp được ứng dụng.

Fig.23 là hình vẽ thể hiện các mẫu tham chiếu được đưa vào mỗi đường mẫu tham chiếu.

Theo phương thức dự báo nội khung của khối hiện tại, mẫu dự báo có thể là thu được bằng cách sử dụng ít nhất một trong số các mẫu tham chiếu thuộc về đường mẫu tham chiếu.

Tiếp theo, phương thức dự báo nội khung của khối hiện tại có thể được xác định (S2202). Đối với phương thức dự báo nội khung của khối hiện tại, ít nhất một trong số phương thức dự báo không góc nội khung và phương thức dự báo góc nội khung có thể được xác định là phương thức dự báo nội khung của khối hiện tại. Phương thức dự báo không góc nội khung gồm phương thức phẳng và phương thức DC, và phương thức dự báo góc nội khung gồm 33 hoặc 65 phương thức giữa hướng xiên đáy-bên trái và hướng xiên đỉnh-bên phải.

Fig.24 là hình vẽ thể hiện các phương thức dự báo nội khung.

Fig.24 (a) thể hiện 35 phương thức dự báo nội khung, và Fig.24 (b) thể hiện 67 phương thức dự báo nội khung.

Số lượng các phương thức dự báo nội khung lớn hơn hoặc nhỏ hơn số lượng được thể hiện trên Fig.24 có thể được xác định.

Phương thức đúng nhất (Most Probable Mode - MPM) có thể được đặt dựa trên phương thức dự báo nội khung của khối lân cận gần kề khối hiện tại. Ở đây, khối lân cận có thể gồm khối lân cận trái gần kề mặt trái của khối hiện tại và khối lân cận đỉnh gần kề đỉnh của khối hiện tại. Khi hệ tọa độ của mẫu trên cùng-bên trái của khối hiện tại là $(0, 0)$, khối lân cận trái có thể gồm mẫu tại vị trí của $(-1, 0)$, $(-1, H-1)$ hoặc $(-1, (H-1)/2)$. Ở đây, H biểu thị độ cao của khối hiện tại. Khối lân cận đỉnh có thể gồm mẫu tại vị trí của $(0, -1)$, $(W-1, -1)$ hoặc $((W-1)/2, -1)$. Ở đây, W biểu thị độ rộng của khối hiện tại.

Khi khối lân cận được mã hóa bằng dự báo nội khung bình thường, MPM có thể được suy ra dựa trên phương thức dự báo nội khung của khối lân cận. Cụ thể là, phương thức dự báo nội khung của khối lân cận trái có thể được đặt thành `candIntraPredModeA` biến thiên, và phương thức dự báo nội khung của khối lân cận đỉnh có thể được đặt thành `candIntraPredModeB` biến thiên.

Tại điểm này, khi khối lân cận là không sẵn có (ví dụ như, khi khối lân cận chưa được mã hóa/được giải mã hoặc khi vị trí của khối lân cận nằm ngoài ranh giới hình ảnh), khi khối lân cận được mã hóa bằng dự báo nội khung dựa trên ma trận, khi khối lân

cận được mã hóa bằng dự báo liên khung, hoặc khi khối lân cận được đưa vào đơn vị cây mã hóa khác với khối hiện tại, `candIntraPredModeX` biến thiên (ở đây, X là A hoặc B) được suy ra dựa trên phương thức dự báo nội khung của khối lân cận có thể được đặt là phương thức mặc định. Ở đây, phương thức mặc định có thể gồm ít nhất một trong số phương thức phẳng, phương thức DC, phương thức hướng dọc, và phương thức hướng ngang.

Theo lựa chọn, khi khối lân cận được mã hóa bằng dự báo nội khung dựa trên ma trận, phương thức dự báo nội khung tương ứng với trị số chỉ số để định rõ bất kỳ một trong số các ma trận có thể được đặt thành `candIntraPredModeX`. Để thực hiện được điều này, bảng tra cứu thể hiện mối tương quan ánh xạ giữa các trị số chỉ số để định rõ ma trận và các phương thức dự báo nội khung có thể được lưu trữ trước trong bộ mã hóa và bộ giải mã.

Các MPM có thể được suy ra dựa trên `candIntraPredModeA` biến thiên và `candIntraPredModeB` biến thiên. Số lượng MPM được đưa vào danh mục MPM có thể được đặt trước trong bộ mã hóa và bộ giải mã. Số lượng MPM được đưa vào danh mục MPM có thể được đặt sẵn trong bộ mã hóa và bộ giải mã. Ví dụ như, số lượng MPM có thể là 3, 4, 5 hoặc 6. Theo lựa chọn, thông tin biểu thị số lượng MPM có thể được báo hiệu thông qua dòng bit. Theo lựa chọn, số lượng MPM có thể được xác định dựa trên ít nhất một trong số phương thức mã hóa dự báo của khối lân cận và kích cỡ và hình dạng của khối hiện tại.

Theo các phương án được mô tả dưới đây, giả sử rằng số lượng MPM là ba, và ba MPM này sẽ được gọi là `MPM[0]`, `MPM[1]` và `MPM[2]`. Khi số lượng MPM là nhiều hơn ba, các MPM có thể được tạo cấu hình để gồm ba MPM được mô tả trong các phương án được mô tả dưới đây.

Khi `candIntraPredA` và `candIntraPredB` là giống nhau, và `candIntraPredA` là phương thức phẳng hoặc phương thức DC, `MPM[0]` và `MPM[1]` có thể được đặt tương ứng thành phương thức phẳng và phương thức DC. `MPM[2]` có thể được đặt thành phương thức dự báo nội khung hướng dọc, phương thức dự báo nội khung hướng ngang, hoặc phương thức dự báo nội khung hướng xiên. Phương thức dự báo nội khung hướng xiên có thể là hướng xiên đáy-bên trái phương thức dự báo nội khung, phương thức dự báo nội khung hướng trên cùng-bên trái, hoặc phương thức dự báo nội khung hướng trên cùng-bên phải.

Khi `candIntraPredA` và `candIntraPredB` là giống nhau, và `candIntraPredA` là

phương thức dự báo góc nội khung, MPM[0] có thể được đặt giống như candIntraPredA. MPM[1] và MPM[2] có thể được đặt tương ứng thành phương thức dự báo nội khung tương tự với candIntraPredA. Phương thức dự báo nội khung tương tự với candIntraPredA có thể là phương thức dự báo nội khung trong đó hiệu số giữa candIntraPredA và chỉ số là ± 1 hoặc ± 2 . Phép toán môđun (%) và độ lệch có thể được sử dụng để suy ra phương thức dự báo nội khung tương tự với candIntraPredA.

Khi candIntraPredA và candIntraPredB là khác nhau, MPM[0] có thể được đặt giống như candIntraPredA, và MPM[1] có thể được đặt giống như candIntraPredB. Tại điểm này, khi cả candIntraPredA và candIntraPredB là phương thức dự báo không góc nội khung, MPM[2] có thể được đặt thành phương thức dự báo nội khung hướng dọc, phương thức dự báo nội khung hướng ngang, hoặc phương thức dự báo nội khung hướng xiên. Theo lựa chọn, khi ít nhất một trong số candIntraPredA và candIntraPredB là phương thức dự báo góc nội khung, MPM[2] có thể được đặt thành phương thức dự báo nội khung được suy ra bằng cách cộng hoặc trừ độ lệch vào hoặc từ phương thức phẳng, phương thức DC, hoặc một trong số candIntraPredA và candIntraPredB có trị số lớn hơn. Ở đây, độ lệch có thể là 1 hoặc 2.

Danh mục MPM gồm nhiều MPM được tạo ra, và thông tin biểu thị việc liệu MPM mà giống như phương thức dự báo nội khung của khối hiện tại được đưa vào danh mục MPM có thể được báo hiệu thông qua dòng bit. Thông tin là cờ 1 bit và có thể được gọi là cờ MPM. Khi cờ MPM biểu thị rằng MPM giống như của khối hiện tại được đưa vào danh mục MPM, thông tin chỉ số nhận dạng một trong số các MPM có thể được báo hiệu thông qua dòng bit. MPM được định rõ bởi thông tin chỉ số có thể được đặt là phương thức dự báo nội khung của khối hiện tại. Khi cờ MPM biểu thị rằng MPM giống như của khối hiện tại không được đưa vào danh mục MPM, thông tin phương thức dự báo nội khung biểu thị bất kỳ một trong số các phương thức dự báo nội khung dự không phải là các MPM có thể được báo hiệu thông qua dòng bit. Thông tin phương thức dự báo nội khung biểu thị trị số chỉ số tương ứng với phương thức dự báo nội khung của khối hiện tại khi các chỉ số được gán lại cho các phương thức dự báo nội khung dự loại trừ các MPM. Bộ giải mã có thể sắp xếp các MPM theo thứ tự tăng lên và xác định phương thức dự báo nội khung của khối hiện tại bằng cách so sánh thông tin phương thức dự báo với các MPM. Ví dụ như, khi thông tin phương thức dự báo là bằng hoặc nhỏ hơn MPM, phương thức dự báo nội khung của khối hiện tại có thể được suy ra bằng cách thêm 1 vào thông tin phương thức dự báo.

Thay vì đặt phương thức mặc định thành MPM, thông tin biểu thị việc liệu có hay

không phương thức dự báo nội khung của khối hiện tại là phương thức mặc định có thể được báo hiệu thông qua dòng bit. Thông tin là cờ 1 bit, và cờ có thể được gọi là cờ phương thức mặc định. Cờ phương thức mặc định có thể được báo hiệu chỉ khi cờ MPM biểu thị rằng MPM giống như của khối hiện tại được đưa vào danh mục MPM. Như được mô tả trên đây, phương thức mặc định có thể gồm ít nhất một trong số phương thức phẳng, phương thức DC, phương thức hướng dọc, và phương thức hướng ngang. Ví dụ như, khi phương thức phẳng được đặt là phương thức mặc định, cờ phương thức mặc định có thể biểu thị việc liệu có hay không phương thức dự báo nội khung của khối hiện tại là phương thức phẳng. Khi cờ phương thức mặc định biểu thị rằng phương thức dự báo nội khung của khối hiện tại không phải là phương thức mặc định, một trong số các MPM được biểu thị bởi thông tin chỉ số có thể được đặt là phương thức dự báo nội khung của khối hiện tại.

Khi nhiều phương thức dự báo nội khung được đặt là các phương thức mặc định, thông tin chỉ số biểu thị bất kỳ một trong số các phương thức mặc định có thể còn được báo hiệu. Phương thức dự báo nội khung của khối hiện tại có thể được đặt là phương thức mặc định được biểu thị bởi thông tin chỉ số.

Khi chỉ số của đường mẫu tham chiếu của khối hiện tại không phải là 0, có thể đặt không sử dụng phương thức mặc định. Do đó, khi chỉ số của đường mẫu tham chiếu không phải là 0, cờ phương thức mặc định không được báo hiệu, và trị số của cờ phương thức mặc định có thể được đặt thành trị số được xác định trước (tức là, sai).

Khi phương thức dự báo nội khung của khối hiện tại được xác định, các mẫu dự báo có thể là thu được cho khối hiện tại dựa trên phương thức dự báo nội khung được xác định (S2203).

Khi phương thức DC được chọn, các mẫu dự báo được tạo ra cho khối hiện tại dựa trên trị số trung bình của các mẫu tham chiếu. Cụ thể là, các trị số của tất cả các mẫu trong khối dự báo có thể được tạo ra dựa trên trị số trung bình của các mẫu tham chiếu. Trị số trung bình có thể được suy ra sử dụng ít nhất một trong số các mẫu tham chiếu đỉnh được bố trí trên đỉnh của khối hiện tại và các mẫu tham chiếu trái được bố trí trên mặt trái của khối hiện tại.

Phụ thuộc vào hình dạng của khối hiện tại, số lượng hoặc phạm vi của các mẫu tham chiếu được sử dụng để suy ra trị số trung bình có thể thay đổi. Ví dụ như, khi khối hiện tại là khối có hình dạng không phải hình vuông có độ rộng lớn hơn độ cao, trị số trung bình có thể được tính toán chỉ sử dụng các mẫu tham chiếu đỉnh. Mặt khác, khi

khối hiện tại là khối có hình dạng không phải hình vuông có độ rộng nhỏ hơn độ cao, trị số trung bình có thể được tính toán chỉ sử dụng các mẫu tham chiếu bên trái. Tức là, khi độ rộng và độ cao của khối hiện tại là khác nhau, trị số trung bình có thể được tính toán chỉ sử dụng các mẫu tham chiếu gần kề khối dài hơn. Theo lựa chọn, dựa trên tỷ lệ độ rộng so với độ cao của khối hiện tại, việc liệu có hay không tính toán trị số trung bình chỉ sử dụng các mẫu tham chiếu đỉnh hoặc việc liệu có hay không tính toán trị số trung bình chỉ sử dụng các mẫu tham chiếu bên trái có thể được xác định.

Khi phương thức phẳng được chọn, mẫu dự báo có thể là thu được sử dụng hướng ngang mẫu dự báo và hướng dọc mẫu dự báo. Ở đây, hướng ngang mẫu dự báo thu được dựa trên mẫu tham chiếu trái và mẫu tham chiếu phải được bố trí trên đường ngang giống như hướng của mẫu dự báo, và hướng dọc mẫu dự báo thu được dựa trên mẫu tham chiếu đỉnh và vị trí mẫu tham chiếu đáy được bố trí trên đường dọc giống như đường dọc của mẫu dự báo. Ở đây, mẫu tham chiếu phải có thể được tạo ra bằng cách sao chép mẫu tham chiếu gần kề góc trên cùng-bên phải của khối hiện tại, và mẫu tham chiếu đáy có thể được tạo ra bằng cách sao chép mẫu tham chiếu gần kề góc dưới cùng-bên trái của khối hiện tại. Hướng ngang mẫu dự báo có thể là thu được dựa trên phép tính tổng có trọng số của mẫu tham chiếu trái và mẫu tham chiếu phải, và hướng dọc mẫu dự báo có thể là thu được dựa trên phép tính tổng có trọng số của mẫu tham chiếu đỉnh và mẫu tham chiếu đáy. Tại điểm này, trị số định trọng số được gán cho mỗi mẫu tham chiếu có thể được xác định theo vị trí của mẫu dự báo. Mẫu dự báo có thể là thu được dựa trên phép tính trị số trung bình hoặc phép tính tổng có trọng số của hướng ngang mẫu dự báo và hướng dọc mẫu dự báo. Khi phép tính tổng có trọng số được thực hiện, các trị số định trọng số được ứng dụng cho hướng ngang mẫu dự báo và hướng dọc mẫu dự báo có thể được xác định dựa trên vị trí của mẫu dự báo.

Khi phương thức dự báo góc được chọn, thông số biểu thị hướng dự báo (hoặc góc dự báo) của phương thức dự báo góc được chọn có thể được xác định. Bảng 2 thể hiện thông số hướng nội khung intraPredAng của mỗi phương thức dự báo nội khung.

【Bảng 2】

PredModeIntr	1	2	3	4	5	6	7
a	-	32	26	21	17	13	9
IntraPredAng							
PredModeIntr	8	9	10	11	12	13	14

aIntraPredAng	5	2	0	-2	-5	-9	-13
PredModeIntr	15	16	17	18	19	20	21
aIntraPredAng	-17	-21	-26	-32	-26	-21	-17
PredModeIntr	22	23	24	25	26	27	28
aIntraPredAng	-13	-9	-5	-2	0	2	5
PredModeIntr	29	30	31	32	33	34	
aIntraPredAng	9	13	17	21	26	32	

Khi 35 phương thức dự báo nội khung được xác định, bảng 2 thể hiện thông số hướng nội khung của mỗi phương thức dự báo nội khung có chỉ số của bất kỳ một trong số 2 đến 34. Khi nhiều hơn 33 phương thức dự báo nội khung góc được xác định, bảng 2 còn được chia nhỏ thêm để đặt thông số hướng nội khung của mỗi phương thức dự báo góc nội khung.

Sau khi sắp xếp các mẫu tham chiếu đỉnh và các mẫu tham chiếu bên trái của khối hiện tại trong một đường, mẫu dự báo có thể là thu được dựa trên trị số của thông số hướng nội khung. Tại điểm này, khi trị số của thông số hướng nội khung là trị số âm, các mẫu tham chiếu bên trái và các mẫu tham chiếu đỉnh có thể được sắp xếp trong một đường.

Các Fig.25 và Fig. 26 là các hình vẽ thể hiện ví dụ về mảng một chiều sắp xếp các mẫu tham chiếu trong một đường.

Fig.25 là hình vẽ thể hiện ví dụ về hướng dọc mảng một chiều sắp xếp các mẫu tham chiếu theo hướng dọc, và Fig.26 là hình vẽ thể hiện ví dụ về hướng ngang mảng một chiều sắp xếp các mẫu tham chiếu theo hướng ngang. Các phương án trên Fig.25 và Fig.26 sẽ được mô tả dựa trên giả định rằng 35 phương thức dự báo nội khung được xác định.

Khi chỉ số phương thức dự báo nội khung là bất kỳ một trong số 11 đến 18, hướng ngang mảng một chiều trong đó các mẫu tham chiếu đỉnh được xoay ngược chiều kim đồng hồ có thể được ứng dụng, và khi chỉ số phương thức dự báo nội khung là bất kỳ một trong số 19 đến 25, hướng dọc mảng một chiều trong đó các mẫu tham chiếu bên trái được xoay theo chiều kim đồng hồ có thể được ứng dụng. Khi sắp xếp các mẫu tham chiếu trong một đường, góc phương thức dự báo nội khung có thể được xem xét.

Thông số xác định mẫu tham chiếu có thể được xác định dựa trên thông số hướng nội khung. Thông số xác định mẫu tham chiếu có thể gồm chỉ số mẫu tham chiếu để định rõ mẫu tham chiếu và thông số trị số định trọng số để xác định trị số định trọng số được

ứng dụng cho mẫu tham chiếu.

Chỉ số mẫu tham chiếu $iIdx$ và thông số trị số định trọng số i_{fact} có thể là thu được tương ứng thông qua các phương trình 17 và 18 được thể hiện dưới đây.

【Phương trình 17】

$$iIdx = (y + 1) * P_{ang}/32$$

【Phương trình 18】

$$i_{fact} = [(y + 1) * P_{ang}] \& 31$$

Trong các phương trình 17 và 18, P_{ang} biểu thị thông số hướng nội khung. Mẫu tham chiếu được định rõ bởi chỉ số mẫu tham chiếu $iIdx$ tương ứng với điểm ảnh số nguyên.

Ít nhất một hoặc nhiều mẫu tham chiếu có thể được định rõ để suy ra mẫu dự báo. Cụ thể là, vị trí của mẫu tham chiếu được sử dụng để suy ra mẫu dự báo có thể được định rõ xét đến độ nghiêng của phương thức dự báo. Ví dụ như, mẫu tham chiếu được sử dụng để suy ra mẫu dự báo có thể được định rõ sử dụng chỉ số mẫu tham chiếu $iIdx$.

Tại điểm này, khi độ nghiêng của phương thức dự báo nội khung không được biểu hiện bởi một mẫu tham chiếu, mẫu dự báo có thể được tạo ra bằng cách nội suy nhiều mẫu tham chiếu. Ví dụ như, khi độ nghiêng của phương thức dự báo nội khung là trị số giữa độ nghiêng giữa mẫu dự báo và mẫu tham chiếu thứ nhất và độ nghiêng giữa mẫu dự báo và mẫu tham chiếu thứ hai, mẫu dự báo có thể là thu được bằng cách nội suy mẫu tham chiếu thứ nhất và mẫu tham chiếu thứ hai. Tức là, khi đường góc theo góc dự báo nội khung không đi qua mẫu tham chiếu được bố trí tại điểm ảnh số nguyên, mẫu dự báo có thể là thu được bằng cách nội suy các mẫu tham chiếu gần kề các mặt trái và phải hoặc các mặt trên và dưới của vị trí mà ở đó đường góc đi qua.

Phương trình 19 thể hiện ví dụ về thu được mẫu dự báo dựa trên các mẫu tham chiếu.

【Phương trình 19】

$$P(x, y) = ((32 - i_{fact})/32) * Ref_1D(x + iIdx + 1) + (i_{fact}/32) * Ref_1D(x + iIdx + 2)$$

Trong phương trình 19, P biểu thị mẫu dự báo, và Ref_1D biểu thị bất kỳ một trong số được bố trí theo một chiều các mẫu tham chiếu. Tại điểm này, vị trí của mẫu tham chiếu có thể được xác định bởi vị trí (x, y) của mẫu dự báo và chỉ số $iIdx$ của mẫu tham chiếu.

Khi độ nghiêng của phương thức dự báo nội khung có thể được biểu hiện là một

mẫu tham chiếu, thông số trị số định trọng số i_{fact} được đặt thành 0. Do đó, phương trình 19 có thể được đơn giản hóa như được thể hiện theo phương trình 20.

【Phương trình 20】

$$P(x, y) = \text{Ref_1D}(x + i\text{Idx} + 1)$$

Dự báo nội khung cho khối hiện tại có thể được thực hiện dựa trên nhiều phương thức dự báo nội khung. Ví dụ như, phương thức dự báo nội khung có thể được suy ra cho mỗi mẫu dự báo, và mẫu dự báo có thể được suy ra dựa trên phương thức dự báo nội khung được gán cho mỗi mẫu dự báo.

Theo lựa chọn, phương thức dự báo nội khung có thể được suy ra cho mỗi vùng, và dự báo nội khung cho mỗi vùng có thể được thực hiện dựa trên phương thức dự báo nội khung được gán cho mỗi vùng. Ở đây, vùng có thể gồm ít nhất một mẫu. Ít nhất một trong số kích cỡ và hình dạng của vùng có thể được xác định theo kiểu thích ứng dựa trên ít nhất một trong số kích cỡ, hình dạng, và phương thức dự báo nội khung của khối hiện tại. Theo lựa chọn, ít nhất một trong số kích cỡ và hình dạng của vùng có thể được xác định trước trong bộ mã hóa và bộ giải mã là độc lập từ kích cỡ hoặc hình dạng của khối hiện tại.

Theo lựa chọn, dự báo nội khung có thể được thực hiện dựa trên mỗi trong số nhiều dự báo nội khung, và mẫu dự báo cuối cùng có thể được suy ra dựa trên phép tính trị số trung bình hoặc phép tính tổng có trọng số của nhiều mẫu dự báo thu được thông qua nhiều dự báo nội khung. Ví dụ như, mẫu dự báo thứ nhất có thể là thu được bằng cách thực hiện dự báo nội khung dựa trên phương thức dự báo nội khung thứ nhất, và mẫu dự báo thứ hai có thể là thu được bằng cách thực hiện dự báo nội khung dựa trên phương thức dự báo nội khung thứ hai. Sau đó, mẫu dự báo cuối cùng có thể là thu được dựa trên phép tính trị số trung bình hoặc phép tính tổng có trọng số của mẫu dự báo thứ nhất và mẫu dự báo thứ hai. Tại điểm này, trị số định trọng số được gán cho mỗi trong số mẫu dự báo thứ nhất và mẫu dự báo thứ hai có thể được xác định xét đến ít nhất một trong số việc liệu có hay không phương thức dự báo nội khung thứ nhất là phương thức dự báo góc/không góc, việc liệu có hay không phương thức dự báo nội khung thứ hai là phương thức dự báo góc/không góc, và phương thức dự báo nội khung của khối lân cận.

Nhiều phương thức dự báo nội khung có thể là sự kết hợp của phương thức dự báo không góc nội khung và phương thức dự báo góc, sự kết hợp của các phương thức dự báo góc, hoặc sự kết hợp của các phương thức dự báo không góc.

Fig.27 là hình vẽ thể hiện ví dụ về các góc được tạo ra giữa các phương thức dự báo

nội khung góc và đường thẳng song song với trục x.

Như được thể hiện trong ví dụ trên Fig.27, các phương thức dự báo góc có thể tồn tại giữa hướng xiên đáy-bên trái và hướng xiên đỉnh-bên phải. Mô tả về góc được tạo ra bởi trục x và phương thức dự báo góc, các phương thức dự báo góc có thể tồn tại giữa 45 độ (hướng xiên đáy-bên trái) và -135 độ (hướng xiên đỉnh-bên phải).

Khi khối hiện tại là hình dạng không phải hình vuông, theo phương thức dự báo nội khung của khối hiện tại, có thể có trường hợp suy ra mẫu dự báo bằng cách sử dụng mẫu tham chiếu xa hơn so với mẫu dự báo, thay vì mẫu tham chiếu còn gần hơn với mẫu dự báo, trong số các mẫu tham chiếu được bố trí trên đường góc theo góc dự báo nội khung.

Fig.28 là hình vẽ thể hiện phương án thu được các mẫu dự báo khi khối hiện tại là hình dạng không phải hình vuông.

Ví dụ như, như được thể hiện trong ví dụ trên Fig.28(a), giả sử rằng khối hiện tại là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, và phương thức dự báo nội khung của khối hiện tại là phương thức dự báo góc nội khung có góc giữa 0 và 45 độ. Trong trường hợp nêu trên, khi mẫu dự báo A gần cột phải của khối hiện tại được suy ra, có thể có trường hợp sử dụng mẫu tham chiếu trái L xa so với mẫu dự báo, thay vì mẫu tham chiếu đỉnh T gần với mẫu dự báo, trong số các mẫu tham chiếu được bố trí trên phương thức góc theo góc.

Theo một ví dụ khác, như được thể hiện trong ví dụ trên Fig.28 (b), giả sử rằng khối hiện tại là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, và phương thức dự báo nội khung của khối hiện tại là phương thức dự báo góc nội khung có góc giữa -90 và -135 độ. Trong trường hợp nêu trên, khi mẫu dự báo A gần hàng đáy của khối hiện tại được suy ra, có thể có trường hợp sử dụng mẫu tham chiếu đỉnh T xa so với mẫu dự báo, thay vì mẫu tham chiếu trái L gần với mẫu dự báo, trong số các mẫu tham chiếu được bố trí trên phương thức góc theo góc.

Để giải quyết vấn đề được mô tả trên đây, khi khối hiện tại là hình dạng không phải hình vuông, phương thức dự báo nội khung của khối hiện tại có thể được thay thế bằng phương thức dự báo nội khung của hướng đối diện. Do đó, các phương thức dự báo góc có góc lớn hơn hoặc nhỏ hơn góc của các phương thức dự báo góc được thể hiện trên Fig.24 có thể được sử dụng cho khối có hình dạng không phải hình vuông. Phương thức dự báo góc nội khung như thế này có thể được xác định là phương thức dự báo nội khung góc rộng. Phương thức dự báo nội khung góc rộng thể hiện phương thức dự báo góc nội khung không thuộc về phạm vi là 45 đến -135 độ.

Fig.29 là hình vẽ thể hiện các phương thức dự báo nội khung góc rộng.

Trong ví dụ được thể hiện trên Fig.29, các phương thức dự báo nội khung với chỉ số của -1 đến -14 và các phương thức dự báo nội khung với chỉ số của 67 đến 80 thể hiện các phương thức dự báo nội khung góc rộng.

Mặc dù 14 phương thức dự báo nội khung góc rộng (-1 đến -14) có góc lớn hơn 45 độ và 14 phương thức dự báo nội khung góc rộng (67 đến 80) có góc nhỏ hơn -135 độ được thể hiện trên Fig.29, số lượng lớn hơn hoặc nhỏ hơn của các phương thức dự báo nội khung góc rộng có thể được xác định.

Khi phương thức dự báo nội khung góc rộng được sử dụng, độ dài của các mẫu tham chiếu đỉnh có thể được đặt thành $2W+1$, và độ dài của các mẫu tham chiếu bên trái có thể được đặt thành $2H+1$.

Theo phương thức dự báo nội khung góc rộng được sử dụng, mẫu A được thể hiện trên Fig.28(a) có thể được dự báo sử dụng mẫu tham chiếu T, và mẫu A được thể hiện trên Fig.28(b) có thể được dự báo sử dụng mẫu tham chiếu L.

Bằng cách thêm các phương thức dự báo nội khung hiện có và N phương thức dự báo nội khung góc rộng, tổng số $67 + N$ các phương thức dự báo nội khung có thể được sử dụng. Ví dụ như, bảng 3 thể hiện các thông số hướng nội khung của các phương thức dự báo nội khung khi 20 các phương thức dự báo nội khung góc rộng được xác định.

【Bảng 3】

PredModeIntra	-10	-9	-8	-7	-6	-5	-4	-3	-2
intraPredAngle	114	93	79	68	60	54	49	45	39
PredModeIntra	-1	2	3	4	5	6	7	8	9
intraPredAngle	35	32	29	26	23	21	19	17	15
PredModeIntra	10	11	12	13	14	15	16	17	18
intraPredAngle	13	11	9	7	5	3	2	1	0
PredModeIntra	19	20	21	22	23	24	25	26	27
intraPredAngle	-1	-2	-3	-5	-7	-9	-11	-13	-15
PredModeIntra	28	29	30	31	32	33	34	35	36
intraPredAngle	-17	-19	-21	-23	-26	-29	-32	-29	-26
PredModeIntra	37	38	39	40	41	42	43	44	45
intraPredAngle	-23	-21	-19	-17	-15	-13	-11	-9	-7
PredModeIntra	46	47	48	49	50	51	52	53	54

intraPredAngle	-5	-3	-2	-1	0	1	2	3	5
PredModeIntra	55	56	57	58	59	60	61	62	63
intraPredAngle	7	9	11	13	15	17	19	21	23
PredModeIntra	64	65	66	67	68	69	70	71	72
intraPredAngle	26	29	32	35	39	45	49	54	60
PredModeIntra	73	74	75	76					
intraPredAngle	68	79	93	114					

Khi khối hiện tại là hình dạng không phải hình vuông và phương thức dự báo nội khung của khối hiện tại thu được ở bước S2202 thuộc về dải biến đổi, phương thức dự báo nội khung của khối hiện tại có thể được biến đổi thành phương thức dự báo nội khung góc rộng. Dải biến đổi có thể được xác định dựa trên ít nhất một trong số kích cỡ, hình dạng, và tỷ lệ của khối hiện tại. Ở đây, tỷ lệ có thể thể hiện tỷ lệ độ rộng so với độ cao của khối hiện tại.

Khi khối hiện tại là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, dải biến đổi có thể được đặt từ chỉ số phương thức dự báo nội khung (ví dụ như, 66) của hướng xiên đỉnh-bên phải (chỉ số của phương thức dự báo nội khung theo hướng xiên đỉnh-bên phải - N). Ở đây, N có thể được xác định dựa trên tỷ lệ của khối hiện tại. Khi phương thức dự báo nội khung của khối hiện tại thuộc về dải biến đổi, phương thức dự báo nội khung có thể được biến đổi thành phương thức dự báo nội khung góc rộng. Phép biến đổi có thể là trừ trị số được xác định trước từ phương thức dự báo nội khung, và trị số được xác định trước có thể là tổng số (ví dụ như, 67) của các phương thức dự báo nội khung loại trừ các phương thức dự báo nội khung góc rộng.

Theo phương án được mô tả trên đây, các phương thức dự báo nội khung thứ 66 đến 53 có thể được biến đổi tương ứng thành các phương thức dự báo nội khung góc rộng thứ -1 đến -14.

Khi khối hiện tại là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, dải biến đổi có thể được đặt từ chỉ số phương thức dự báo nội khung (ví dụ như, 2) của hướng xiên đáy-bên trái (chỉ số của phương thức dự báo nội khung theo hướng xiên đáy-bên trái + M). Ở đây, M có thể được xác định dựa trên tỷ lệ của khối hiện tại. Khi phương thức dự báo nội khung của khối hiện tại thuộc về dải biến đổi, phương thức dự báo nội khung có thể được biến đổi thành phương thức dự báo nội khung góc rộng. Phép biến đổi có thể là cộng trị số được xác định trước vào phương thức dự báo nội khung, và trị số được xác định trước có thể là tổng số (ví dụ như, 65) của các phương thức dự báo

nội khung góc loại trừ các phương thức dự báo nội khung góc rộng.

Theo phương án được mô tả trên đây, các phương thức dự báo nội khung thứ 2 đến 15 có thể được biến đổi tương ứng thành các phương thức dự báo nội khung góc rộng thứ 67 đến 80.

Trong phần dưới đây, các phương thức dự báo nội khung thuộc về dải biến đổi sẽ được gọi là các phương thức dự báo thay thế nội khung góc rộng.

Dải biến đổi có thể được xác định dựa trên tỷ lệ của khối hiện tại. Ví dụ như, các bảng 4 và 5 thể hiện dải biến đổi khi 35 phương thức dự báo nội khung và 67 phương thức dự báo nội khung tương ứng được xác định loại trừ các phương thức dự báo nội khung góc rộng.

【Bảng 4】

Điều kiện	Các phương thức dự báo nội khung bị thay thế
$W/H = 2$	Các phương thức 2, 3, 4
$W/H > 2$	Các phương thức 2, 3, 4, 5, 6
$W/H = 1$	Không có
$H/W = 1/2$	Các phương thức 32, 33, 34
$H/W < 1/2$	Các phương thức 30, 31, 32, 33, 34

【Bảng 5】

Điều kiện	Các phương thức dự báo nội khung bị thay thế
$W/H = 2$	Các phương thức 2, 3, 4, 5, 6, 7
$W/H > 2$	Các phương thức 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
$W/H = 1$	Không có
$H/W = 1/2$	Các phương thức 61, 62, 63, 64, 65, 66
$H/W < 1/2$	Các phương thức 57, 58, 59, 60, 61, 62, 63, 64, 65, 66

Như được thể hiện trong các ví dụ của các bảng 4 và 5, số lượng các phương thức dự báo thay thế nội khung góc rộng thuộc về dải biến đổi có thể thay đổi theo tỷ lệ của khối hiện tại.

Khi các phương thức dự báo nội khung góc rộng được sử dụng ngoài các phương thức dự báo nội khung hiện có ra, các tài nguyên cần thiết cho mã hóa các phương thức dự báo nội khung góc rộng tăng lên, và do đó hiệu suất mã hóa có thể bị giảm. Do đó, hiệu suất mã hóa có thể được cải thiện bằng cách mã hóa các phương thức dự báo nội

khung thay thế cho các phương thức dự báo nội khung góc rộng, thay vì mã hóa các phương thức dự báo nội khung góc rộng theo hoàn cảnh thực tế.

Ví dụ như, khi khối hiện tại được mã hóa bằng cách sử dụng phương thức dự báo nội khung góc rộng thứ 67, phương thức dự báo nội khung thứ 2, đây là phương thức dự báo nội khung thay thế của phương thức dự báo nội khung góc rộng thứ 67, có thể được mã hóa bằng cách sử dụng phương thức dự báo nội khung của khối hiện tại. Ngoài ra, khi khối hiện tại được mã hóa bằng cách sử dụng phương thức dự báo nội khung góc rộng thứ -1, phương thức dự báo nội khung thứ 66, đây là phương thức dự báo nội khung thay thế của phương thức dự báo nội khung góc rộng thứ -1, có thể được mã hóa bằng cách sử dụng phương thức dự báo nội khung của khối hiện tại.

Bộ giải mã có thể giải mã phương thức dự báo nội khung của khối hiện tại và xác định việc liệu phương thức dự báo nội khung được giải mã thuộc về dải biến đổi. Khi phương thức dự báo nội khung được giải mã là phương thức dự báo nội khung thay thế góc rộng, phương thức dự báo nội khung có thể được biến đổi thành phương thức dự báo nội khung góc rộng.

Theo lựa chọn, khi khối hiện tại được mã hóa bằng cách sử dụng phương thức dự báo nội khung góc rộng, phương thức dự báo nội khung góc rộng có thể được mã hóa theo hoàn cảnh thực tế.

Việc mã hóa phương thức dự báo nội khung có thể được thực hiện dựa trên danh mục MPM được mô tả trên đây. Trong phần dưới đây, phương pháp tạo cấu hình danh mục MPM sẽ được mô tả chi tiết. Trong các phương án được mô tả dưới đây, giả sử rằng 10 các phương thức dự báo nội khung góc rộng (-1 đến -10) với góc lớn hơn 45 độ và 10 phương thức dự báo nội khung góc rộng (67 đến 76) với góc nhỏ hơn -135 độ được xác định.

Khi khối lân cận được mã hóa bằng cách sử dụng phương thức dự báo nội khung góc rộng, MPM có thể được đặt dựa trên phương thức dự báo nội khung thay thế góc rộng tương ứng với phương thức dự báo nội khung góc rộng. Ví dụ như, khi khối lân cận được mã hóa bằng cách sử dụng phương thức dự báo nội khung góc rộng, `candIntraPredX` biến thiên (X là A hoặc B) có thể được đặt thành phương thức dự báo nội khung thay thế góc rộng.

Theo lựa chọn, phương pháp suy ra MPM có thể được xác định theo hình dạng của khối hiện tại. Ví dụ như, khi khối hiện tại là dạng hình vuông có độ rộng giống như độ cao, `candIntraPredX` có thể được đặt thành phương thức dự báo nội khung thay thế góc

rộng. Mặt khác, khi khối hiện tại là hình dạng không phải hình vuông, candIntraPredX có thể được đặt thành phương thức dự báo nội khung góc rộng.

Theo lựa chọn, việc liệu có hay không để đặt candIntraPredX thành phương thức dự báo nội khung góc rộng có thể được xác định dựa trên việc liệu phương thức dự báo nội khung góc rộng của khối lân cận có thể được ứng dụng cho khối hiện tại. Ví dụ như, khi khối hiện tại là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, phương thức dự báo nội khung góc rộng có chỉ số lớn hơn chỉ số của phương thức dự báo nội khung của hướng xiên đỉnh-bên phải được đặt là candIntraPredX theo hoàn cảnh thực tế. Tuy nhiên, phương thức dự báo nội khung thay thế góc rộng tương ứng với nội dung đó được đặt là candIntraPredX cho phương thức dự báo nội khung góc rộng có chỉ số nhỏ hơn chỉ số của phương thức dự báo nội khung của hướng xiên đáy-bên trái. Mặt khác, khi khối hiện tại là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, phương thức dự báo nội khung góc rộng có chỉ số nhỏ hơn chỉ số của phương thức dự báo nội khung của hướng xiên đáy-bên trái được đặt là candIntraPredX theo hoàn cảnh thực tế. Tuy nhiên, phương thức dự báo nội khung thay thế góc rộng tương ứng với nội dung đó được đặt là candIntraPredX cho phương thức dự báo nội khung góc rộng có chỉ số lớn hơn chỉ số của phương thức dự báo nội khung của hướng xiên đỉnh-bên phải.

Tức là, việc liệu có hay không để suy ra MPM sử dụng phương thức dự báo nội khung góc rộng theo thực tế, hoặc việc liệu có hay không để suy ra MPM sử dụng phương thức dự báo nội khung thay thế góc rộng có thể được xác định theo việc liệu khối lân cận được mã hóa trong phương thức dự báo nội khung góc rộng có hình dạng giống như hoặc tương tự với hình dạng của khối hiện tại.

Theo lựa chọn, phương thức dự báo nội khung góc rộng của khối lân cận có thể được đặt là candIntraPredX bất kể hình dạng của khối hiện tại.

Tóm lại, candIntraPredX có thể được đặt thành phương thức dự báo nội khung góc rộng hoặc phương thức dự báo nội khung thay thế góc rộng của khối lân cận.

MPM có thể được suy ra dựa trên candIntraPredA và candIntraPredB . Trong trường hợp này, MPM có thể được suy ra trong phương thức dự báo nội khung tương tự với candIntraPredA hoặc candIntraPredB . Phương thức dự báo nội khung tương tự với candIntraPredA hoặc candIntraPredB có thể được suy ra dựa trên phép toán môđun và độ lệch. Tại điểm này, hằng số và độ lệch được sử dụng cho phép toán môđun có thể được xác định khác nhau theo hình dạng của khối hiện tại.

Bảng 6 thể hiện ví dụ về suy ra MPM theo hình dạng của khối hiện tại.

【Bảng 6】

i) Nếu cuWidth là bằng cuHeight

$\text{candModeList}[0] = \text{candIntraPredModeA}$

$\text{candModeList}[1] = 2 + ((\text{candIntraPredModeA} + 61) \% 64)$

$\text{candModeList}[2] = 2 + ((\text{candIntraPredModeA} - 1) \% 64)$

ii) nếu không (tức là, cuWidth không bằng cuHeight)

$\text{candModeList}[0] = \text{candIntraPredModeA}$

$\text{candModeList}[1] = 2 + ((\text{candIntraPredModeA} + 81) \% 84)$

$\text{candModeList}[2] = 2 + ((\text{candIntraPredModeA} - 1) \% 84)$

Giả sử rằng candIntraPredA và candIntraPredB là giống nhau, và candIntraPredA là phương thức dự báo góc nội khung. Khi khối hiện tại là dạng hình vuông, phương thức dự báo nội khung tương tự với candIntraPredA có thể là thu được dựa trên phép toán môđun dựa trên trị số thu được bằng cách trừ 1 từ tổng số các phương thức dự báo nội khung góc loại trừ các phương thức dự báo nội khung góc rộng. Ví dụ như, khi số lượng các phương thức dự báo nội khung góc không phải là các phương thức dự báo nội khung góc rộng là 65, MPM có thể được suy ra dựa trên trị số được suy ra dựa trên candIntraPredA và phép toán môđun là 64. Mặt khác, khi khối hiện tại là hình dạng không phải hình vuông, phương thức dự báo nội khung tương tự với candIntraPredA có thể là thu được dựa trên phép toán môđun dựa trên trị số thu được bằng cách trừ 1 từ tổng số các phương thức dự báo nội khung góc gồm các phương thức dự báo nội khung góc rộng. Ví dụ như, khi số lượng các phương thức dự báo nội khung góc rộng là 20, MPM có thể được suy ra dựa trên trị số được suy ra dựa trên candIntraPredA và phép toán môđun là 84.

Khi hàng số được sử dụng cho phép toán môđun được đặt khác nhau theo hình dạng của khối hiện tại, việc liệu phương thức dự báo nội khung góc rộng có thể được đặt thành phương thức dự báo góc nội khung tương tự với candIntraPredA có thể được xác định. Ví dụ như, phương thức dự báo nội khung góc rộng có thể không được đặt là phương thức dự báo góc nội khung tương tự với candIntraPredA trong phép toán môđun sử dụng 64, trong khi phương thức dự báo nội khung góc rộng có thể được đặt là phương thức dự báo góc nội khung tương tự với candIntraPredA trong phép toán môđun sử dụng 84.

Theo lựa chọn, khi candIntraPredA và candIntraPredB là giống nhau, MPM có thể được suy ra xét đến hình dạng của khối hiện tại và việc liệu candIntraPredA là phương thức dự báo nội khung góc rộng.

Bảng 7 thể hiện ví dụ về suy ra MPM theo hình dạng của khối hiện tại.

【Bảng 7】

```
- Nếu candIntraPredModeB là bằng candIntraPredModeA, áp dụng nội dung sau
đây:
-- Nếu candIntraPredModeA nhỏ hơn 2 (tức là, bằng INTRA_PLANAR hoặc
INTRA_DC), candModeList[x] với x = 0...2 được suy ra như sau:
candModeList[0] = INTRA_PLANAR
candModeList[1] = INTRA_DC
candModeList[2] = INTRA_ANGULAR50
-- nếu không candModeList[x] với x = 0.. 2 được suy ra như sau:
i) Nếu cuWidth là bằng cuHeight
1. nếu candIntrapredA nhỏ hơn 0 hoặc candIntraPredModeA lớn hơn 66
candModeList[0] = INTRA_PLANAR
candModeList[1] = INTRA_DC
candModeList[2] = INTRA_ANGULAR50
2. nếu không nếu candIntraPredA >= 0 && candIntraPredModeA < 67)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
ii) nếu không nếu cuWidth lớn hơn cuHeight
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)
iii) nếu không (tức là, cuWidth lớn hơn cuHeight)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = -8 + ((candIntraPredModeA - 1) % 84)
- Nếu không (candIntraPredModeB không bằng candIntraPredModeA), áp dụng nội
dung sau đây:
-- candModeList[0] và candModeList[1] được suy ra như sau:
candModeList[0] = candIntraPredModeA
candModeList[1] = candIntraPredModeB
```

```
-- Nếu không candModeList[0] và candModeList[1] là bằng INTRA_PLANAR,
candModeList[2] được đặt thành INTRA_PLANAR
-- Nếu không, nếu không gì trong số candModeList[0] và candModeList[1] là bằng
INTRA_DC, candModeList[2] được đặt bằng INTRA_DC,
-- Nếu không, candModeList[2] được đặt thành INTRA_ANGULAR 50.
```

Giả sử rằng candIntraPredA và candIntraPredB là giống nhau.

Khi khối hiện tại là dạng hình vuông và candIntraPredA là phương thức dự báo nội khung góc rộng, các MPM có thể được đặt thành các phương thức mặc định. Ví dụ như, MPM[0], MPM[1], và MPM[2] có thể được đặt tương ứng thành phương thức phẳng, phương thức DC, và phương thức dự báo nội khung dọc.

Khi khối hiện tại là dạng hình vuông và candIntraPredA không phải là phương thức dự báo nội khung góc rộng, nhưng phương thức dự báo góc nội khung, các MPM có thể được đặt thành candIntraPredA và phương thức dự báo góc nội khung tương tự với nội dung đó. Ví dụ như, MPM[0] có thể được đặt thành candIntraPredA, và MPM[1] và MPM[2] có thể được đặt thành candIntraPredA và phương thức dự báo góc nội khung tương tự với nội dung đó.

Khi khối hiện tại là hình dạng không phải hình vuông và candIntraPredA là phương thức dự báo góc nội khung, các MPM có thể được đặt thành candIntraPredA và phương thức dự báo góc nội khung tương tự với nội dung đó. Ví dụ như, MPM[0] có thể được đặt thành candIntraPredA, và MPM[1] và MPM[2] có thể được đặt thành candIntraPredA và phương thức dự báo góc nội khung tương tự với nội dung đó.

Phương thức dự báo góc nội khung tương tự với candIntraPredA có thể được suy ra sử dụng phép toán môđun và độ lệch. Tại điểm này, hằng số được sử dụng cho phép toán môđun có thể thay đổi theo hình dạng của khối hiện tại. Ngoài ra, độ lệch được sử dụng để suy ra phương thức dự báo góc nội khung tương tự với candIntraPredA có thể được đặt khác nhau theo hình dạng của khối hiện tại. Ví dụ như, khi khối hiện tại là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, phương thức dự báo góc nội khung tương tự với candIntraPredA có thể được suy ra sử dụng độ lệch 2. Mặt khác, khi khối hiện tại là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, phương thức dự báo góc nội khung tương tự với candIntraPredA có thể được suy ra sử dụng các độ lệch 2 và -8.

Theo lựa chọn, các MPM có thể được suy ra xét đến việc liệu candIntraPredX là phương thức dự báo nội khung góc rộng có chỉ số lớn nhất hoặc nhỏ nhất.

Bảng 8 thể hiện ví dụ về suy ra MPM xét đến chỉ số của phương thức dự báo nội khung góc rộng.

【Bảng 8】

- Nếu `candIntraPredModeB` là bằng `candIntraPredModeA`, áp dụng nội dung sau đây:

-- Nếu `candIntraPredModeA` nhỏ hơn 2 (tức là, bằng `INTRA_PLANAR` hoặc `INTRA_DC`), `candModeList[x]` với $x = 0 \dots 2$ được suy ra như sau:

`candModeList[0] = INTRA_PLANAR`
`candModeList[1] = INTRA_DC`
`candModeList[2] = INTRA_ANGULAR50`

-- nếu không `candModeList[x]` với $x = 0 \dots 2$ được suy ra như sau:

i) Nếu (`candIntraPredModeA < 0 && candIntraPredModeA != -10`)

`candModeList[0] = candIntraPredModeA`
`candModeList[1] = 2 + ((candIntraPredModeA - 3) % 84)`
`candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)`

ii) nếu không, nếu (`candIntraPredModeA = -10`)

`candModeList[0] = candIntraPredModeA`
`candModeList[1] = -1`
`candModeList[2] = -9`

iii) nếu không, nếu (`candIntraPredModeA > 67 && candIntraPredModeA < 76`)

`candModeList[0] = candIntraPredModeA`
`candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)`
`candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)`

iv) nếu không, `candIntraPredModeA` là bằng 67

`candModeList[0] = candIntraPredModeA`
`candModeList[1] = 66`
`candModeList[2] = 68`

v) nếu không, `candIntraPredModeA` là bằng 77

`candModeList[0] = candIntraPredModeA`
`candModeList[1] = 76`
`candModeList[2] = 67`

vi) nếu không, nếu (`candIntraPredModeA >= 0 && candIntraPredModeA < 67`)

```

candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
- Nếu không (candIntraPredModeB không bằng candIntraPredModeA), áp dụng nội
dung sau đây:
-- candModeList[0] và candModeList[1] được suy ra như sau:
candModeList[0] = candIntraPredModeA
candModeList[1] = candIntraPredModeB
-- Nếu không candModeList[0] và candModeList[1] là bằng INTRA_PLANAR,
candModeList[2] được đặt thành INTRA_PLANAR
-- Nếu không, nếu không gì trong số candModeList[0] và candModeList[1] là bằng
INTRA_DC, candModeList[2] được đặt bằng INTRA_DC,
-- Nếu không, candModeList[2] được đặt thành INTRA_ANGULAR 50.

```

Giả sử rằng `candIntraPredA` và `candIntraPredB` là giống nhau. Để thuận tiện cho việc giải thích, các phương thức dự báo nội khung góc rộng có trị số chỉ số nhỏ hơn trị số chỉ số của phương thức dự báo nội khung của hướng xiên đáy-bên trái được gọi là các phương thức dự báo nội khung góc rộng hướng đáy, và các phương thức dự báo nội khung góc rộng có trị số chỉ số lớn hơn trị số chỉ số của phương thức dự báo nội khung của hướng xiên đỉnh-bên phải được gọi là các phương thức dự báo nội khung góc rộng hướng phải.

Khi `candIntraPredA` là phương thức dự báo nội khung góc rộng hướng đáy, các MPM có thể được đặt thành `candIntraPredA` và phương thức dự báo góc nội khung tương tự với nội dung đó. Tại điểm này, khi `candIntraPredA` là phương thức dự báo nội khung góc rộng hướng đáy có trị số nhỏ nhất, MPM có thể được đặt thành phương thức dự báo nội khung góc rộng hướng đáy có trị số chỉ số được xác định trước. Ở đây, chỉ số được xác định trước có thể là chỉ số có trị số lớn nhất trong số các chỉ số của các phương thức dự báo nội khung góc rộng hướng đáy. Ví dụ như, khi `candIntraPredA` là -10, `MPM[0]`, `MPM[1]`, và `MPM[2]` có thể được đặt thành -10, -1, và -9, tương ứng.

Khi `candIntraPredA` là phương thức dự báo nội khung góc rộng hướng phải, các MPM có thể được đặt thành `candIntraPredA` và phương thức dự báo góc nội khung tương tự với nội dung đó. Tại điểm này, khi `candIntraPredA` là phương thức dự báo nội khung góc rộng hướng phải có trị số lớn nhất, MPM có thể được đặt thành phương thức dự báo nội khung góc rộng hướng phải có trị số chỉ số được xác định trước. Ở đây, chỉ số

được xác định trước có thể là chỉ số có trị số nhỏ nhất trong số các chỉ số của các phương thức dự báo nội khung góc rộng hướng phải. Ví dụ như, khi `candIntraPredA` là 77, `MPM[0]`, `MPM[1]`, và `MPM[2]` có thể được đặt tương ứng thành 77, 76, và 67.

Theo lựa chọn, khi chỉ số được suy ra bằng cách trừ hoặc cộng 1 từ hoặc vào chỉ số của `candIntraPredA` nhỏ hơn trị số nhỏ nhất hoặc lớn hơn trị số lớn nhất trong số các chỉ số của các phương thức dự báo nội khung, MPM có thể được đặt thành phương thức mặc định. Ở đây, phương thức mặc định có thể gồm ít nhất một trong số phương thức phẳng, phương thức DC, phương thức dự báo nội khung dọc, phương thức dự báo nội khung ngang, và phương thức dự báo nội khung chéo.

Theo lựa chọn, khi chỉ số được suy ra bằng cách trừ hoặc cộng 1 từ hoặc vào chỉ số của `candIntraPredA` nhỏ hơn trị số nhỏ nhất hoặc lớn hơn trị số lớn nhất trong số các chỉ số của các phương thức dự báo nội khung, MPM có thể được đặt thành phương thức dự báo nội khung ngược với `candIntraPredA` hoặc phương thức dự báo nội khung tương tự với phương thức dự báo nội khung ngược với `candIntraPredA`.

Theo lựa chọn, các ứng viên MPM có thể được suy ra xét đến hình dạng của khối hiện tại và hình dạng của khối lân cận. Ví dụ như, phương pháp suy ra các MPM khi cả khối hiện tại và khối lân cận là các dạng không phải hình vuông có thể là khác với phương pháp suy ra các MPM khi khối hiện tại là dạng hình vuông và khối lân cận là hình dạng không phải hình vuông.

Các MPM trong danh mục MPM có thể được bố trí lại (được sắp xếp lại thứ tự) xét đến ít nhất một trong số kích cỡ của khối hiện tại, hình dạng của khối hiện tại, kích cỡ của khối lân cận, và hình dạng của khối lân cận. Ở đây, việc sắp xếp lại biểu thị việc gán lại các chỉ số được gán cho mỗi trong số các MPM. Ví dụ như, chỉ số nhỏ hơn có thể được gán cho MPM giống như phương thức dự báo nội khung của khối lân cận có kích cỡ hoặc hình dạng giống như kích cỡ hoặc hình dạng của khối hiện tại.

Giả sử rằng `MPM[0]` và `MPM[1]` được đặt tương ứng thành phương thức dự báo nội khung `candIntraPredA` của khối lân cận trái và phương thức dự báo nội khung `candIntraPredB` của khối lân cận đỉnh.

Khi khối hiện tại và khối lân cận đỉnh là các dạng không phải hình vuông có độ rộng lớn hơn độ cao, các MPM có thể được bố trí lại sao cho phương thức dự báo nội khung `candIntraPredB` của khối lân cận đỉnh có thể có chỉ số nhỏ hơn. Tức là, `candIntraPredB` có thể được bố trí lại cho `MPM[0]` và `candIntraPredA` có thể được bố trí lại cho `MPM[1]`.

Theo lựa chọn, khi khối hiện tại và khối lân cận đỉnh là các dạng không phải hình vuông có độ cao lớn hơn độ rộng, các MPM có thể được bố trí lại sao cho phương thức dự báo nội khung candIntraPredB của khối lân cận đỉnh có chỉ số nhỏ hơn. Tức là, candIntraPredB có thể được bố trí lại cho $\text{MPM}[0]$ và candIntraPredA có thể được bố trí lại cho $\text{MPM}[1]$.

Theo lựa chọn, khi khối hiện tại và khối lân cận đỉnh là các dạng hình vuông, các MPM có thể được bố trí lại sao cho phương thức dự báo nội khung candIntraPredB của khối lân cận đỉnh có chỉ số nhỏ hơn. Tức là, candIntraPredB có thể được bố trí lại cho $\text{MPM}[0]$ và candIntraPredA có thể được bố trí lại cho $\text{MPM}[1]$.

Thay vì sắp xếp lại các MPM, khi candIntraPredX ban đầu được gán cho MPM, ít nhất một trong số kích cỡ của khối hiện tại, hình dạng của khối hiện tại, kích cỡ của khối lân cận, và hình dạng của khối lân cận có thể được xem xét.

Các MPM có thể được bố trí lại dựa trên kích cỡ hoặc hình dạng của khối hiện tại. Ví dụ như, khi khối hiện tại là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, các MPM có thể được bố trí lại theo thứ tự giảm dần. Mặt khác, khi khối hiện tại là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, các MPM có thể được bố trí lại theo thứ tự tăng lên.

Video dư thừa được suy ra có thể được suy ra bằng cách trừ video dự báo từ video gốc. Tại điểm này, khi video dư thừa bị thay đổi thành miền tần số, chất lượng video chủ quan của video không bị giảm đáng kể mặc dù các thành phần tần số cao trong số các thành phần tần số bị loại bỏ. Do đó, khi các trị số của các thành phần tần số cao được chuyển đổi để nhỏ hoặc các trị số của các thành phần tần số cao được đặt thành 0, có tác dụng tăng hiệu suất nén mà không tạo ra hiện tượng méo trực quan đáng kể. Bằng cách phản ánh đặc điểm này, khối hiện tại có thể được biến đổi để phân giải video dư thừa thành các thành phần tần số hai chiều. Phép biến đổi có thể được thực hiện sử dụng kỹ thuật biến đổi chẳng hạn như Biến đổi Cosin Rời rạc (Discrete Cosine Transform - DST) hoặc Biến đổi Sin Rời rạc (Discrete Sine Transform - DST).

DCT phân giải (hoặc biến đổi) video dư thừa thành các thành phần tần số hai chiều sử dụng phép biến đổi cosin, và DST phân giải (hoặc biến đổi) video dư thừa thành các thành phần tần số hai chiều sử dụng phương thức biến đổi sin. Do biến đổi video dư thừa, các thành phần tần số có thể được biểu hiện là video cơ sở. Ví dụ như, khi biến đổi DCT được thực hiện trên khối có kích cỡ $N \times N$, nhiều thành phần mẫu hình cơ bản như N^2 có thể thu được. Kích cỡ của mỗi trong số các thành phần mẫu hình cơ bản được đưa

vào khối có kích cỡ $N \times N$ có thể thu được thông qua phép biến đổi. Theo kỹ thuật biến đổi được sử dụng cho phép biến đổi đó, kích cỡ của thành phần mẫu hình cơ bản có thể được gọi là hệ số DCT hoặc hệ số DST.

Kỹ thuật biến đổi DCT chủ yếu được sử dụng để biến đổi video trong đó nhiều thành phần tần số thấp khác không được phân bố. Kỹ thuật biến đổi DST chủ yếu được sử dụng cho các video trong đó nhiều thành phần tần số cao được phân bố.

Video dư thừa có thể được biến đổi sử dụng kỹ thuật biến đổi không phải là DCT hoặc DST.

Trong phần dưới đây, sự biến đổi video dư thừa thành các thành phần tần số hai chiều sẽ được gọi là việc biến đổi video hai chiều. Ngoài ra, kích cỡ của các thành phần mẫu hình cơ bản thu được do sự biến đổi được gọi là hệ số biến đổi. Ví dụ như, hệ số biến đổi có thể có nghĩa là hệ số DCT hoặc hệ số DST. Khi cả phép biến đổi thứ nhất và phép biến đổi thứ hai được mô tả dưới đây được ứng dụng, hệ số biến đổi có thể có nghĩa là kích cỡ của thành phần mẫu hình cơ bản được tạo ra do việc biến đổi thứ hai.

Kỹ thuật biến đổi có thể được xác định cho mỗi khối. Kỹ thuật biến đổi có thể được xác định dựa trên ít nhất một trong số phương thức mã hóa dự báo của khối hiện tại, hình dạng của khối hiện tại, và kích cỡ của khối hiện tại. Ví dụ như, khi khối hiện tại được mã hóa theo phương thức dự báo nội khung và kích cỡ của khối hiện tại nhỏ hơn $N \times N$, việc biến đổi có thể được thực hiện sử dụng kỹ thuật biến đổi DST. Mặt khác, khi các điều kiện nêu trên không được thỏa mãn, việc biến đổi có thể được thực hiện sử dụng kỹ thuật biến đổi DCT.

Việc biến đổi video hai chiều có thể không được thực hiện cho một số khối của video dư thừa. Không thực hiện việc biến đổi video hai chiều có thể được gọi là nhảy biến đổi. Khi nhảy biến đổi được ứng dụng, việc lượng tử hóa có thể được ứng dụng cho các hệ số dư đã không bị biến đổi.

Sau khi khối hiện tại được biến đổi bằng cách sử dụng DCT hoặc DST, khối hiện tại được biến đổi có thể được biến đổi lại. Tại điểm này, việc biến đổi dựa trên DCT hoặc DST có thể được xác định là việc biến đổi thứ nhất, và biến đổi lại khối mà việc biến đổi thứ nhất được ứng dụng cho có thể được xác định là việc biến đổi thứ hai.

Việc biến đổi thứ nhất có thể được thực hiện sử dụng bất kỳ một trong số nhiều ứng viên lõi biến đổi. Ví dụ như, việc biến đổi thứ nhất có thể được thực hiện sử dụng bất kỳ một trong số DCT2, DCT8, hoặc DCT7.

Các lõi biến đổi khác nhau có thể được sử dụng cho hướng ngang và hướng dọc.

Thông tin biểu thị sự kết hợp của lỗi biến đổi của hướng ngang và lỗi biến đổi của hướng dọc có thể được báo hiệu thông qua dòng bit.

Các bộ phận để thực hiện việc biến đổi thứ nhất và việc biến đổi thứ hai có thể khác nhau. Ví dụ như, việc biến đổi thứ nhất có thể được thực hiện trên khối 8×8 , và việc biến đổi thứ hai có thể được thực hiện trên khối con có kích cỡ 4×4 trong số khối 8×8 được biến đổi. Tại điểm này, các hệ số biến đổi của các vùng dư đã không được thực hiện việc biến đổi thứ hai có thể được đặt thành 0.

Theo lựa chọn, việc biến đổi thứ nhất có thể được thực hiện trên khối 4×4 , và việc biến đổi thứ hai có thể được thực hiện trên vùng có kích cỡ 8×8 gồm khối 4×4 được biến đổi.

Thông tin biểu thị liệu việc biến đổi thứ hai đã được thực hiện có thể được báo hiệu thông qua dòng bit hay không.

Bộ giải mã có thể thực hiện việc biến đổi ngược của việc biến đổi thứ hai (việc biến đổi ngược thứ hai), và có thể thực hiện việc biến đổi ngược của việc biến đổi thứ nhất (việc biến đổi ngược thứ nhất) theo kết quả của việc biến đổi ngược. Do thực hiện việc biến đổi ngược thứ hai và việc biến đổi ngược thứ nhất, các tín hiệu dư cho khối hiện tại có thể là thu được.

Việc lượng tử hóa là để giảm năng lượng của khối, và các quy trình lượng tử hóa gồm quy trình chia hệ số biến đổi bởi trị số hằng số cụ thể. Trị số hằng số có thể được suy ra bởi thông số lượng tử hóa, và thông số lượng tử hóa có thể được xác định là trị số giữa 1 và 63.

Khi bộ mã hóa thực hiện việc biến đổi và lượng tử hóa, bộ giải mã có thể thu được khối dư thông qua lượng tử hóa ngược và việc biến đổi ngược. Bộ giải mã có thể thu được khối được tái cấu trúc cho khối hiện tại bằng cách thêm khối dự báo và khối dư.

Khi thu được khối được tái cấu trúc của khối hiện tại, việc mất thông tin xảy ra trong quy trình mã hóa và lượng tử hóa có thể được giảm bớt thông qua việc lọc trong vòng lặp. Bộ lọc trong vòng lặp có thể gồm ít nhất một trong số bộ lọc giải khối, bộ lọc độ lệch thích ứng mẫu (Sample Adaptive Offset filter - SAO), và bộ lọc vòng lặp thích ứng (Adaptive Loop Filter - ALF). Trong phần dưới đây, khối được tái cấu trúc trước khi bộ lọc trong vòng lặp được ứng dụng được gọi là khối được tái cấu trúc thứ nhất, và khối được tái cấu trúc sau khi bộ lọc trong vòng lặp được ứng dụng được gọi là khối được tái cấu trúc thứ hai.

Khối được tái cấu trúc thứ hai có thể là thu được bằng cách ứng dụng ít nhất một

trong số bộ lọc giải khối, SAO, và ALF cho khối được tái cấu trúc thứ nhất. Tại điểm này, SAO hoặc ALF có thể được ứng dụng sau khi bộ lọc giải khối được ứng dụng.

Bộ lọc giải khối là để giảm nhẹ việc làm giảm chất lượng video (thành phần lạ khối lớn) tại ranh giới của khối, điều này xảy ra khi việc lượng tử hóa được thực hiện trên các khối riêng. Để ứng dụng bộ lọc giải khối, độ bền khối (Block Strength - BS) giữa khối được tái cấu trúc thứ nhất và khối được tái cấu trúc lân cận có thể được xác định.

Fig.30 là lưu đồ minh họa quy trình xác định độ bền khối.

Trong ví dụ được thể hiện trên Fig.30, P biểu thị khối được tái cấu trúc thứ nhất, và Q biểu thị khối được tái cấu trúc lân cận. Ở đây, khối được tái cấu trúc lân cận có thể là khối gần kề mặt trái hoặc mặt đỉnh của khối hiện tại.

Trong ví dụ được thể hiện trên Fig.30, điều được thể hiện là độ bền khối được xác định xét đến các phương thức mã hóa dự báo của P và Q, việc liệu hệ số biến đổi khác không được đưa vào, việc liệu dự báo liên khung được thực hiện sử dụng cùng một hình ảnh tham chiếu, và việc liệu trị số chênh lệch của các vector chuyển động lớn hơn hoặc bằng trị số ngưỡng.

Việc liệu hoặc không ứng dụng bộ lọc giải khối có thể được xác định dựa trên độ bền khối. Ví dụ như, khi độ bền khối là 0, việc lọc có thể không được thực hiện.

SAO là để giảm nhẹ hiện tượng rung lắc (thành phần lạ rung lắc) xảy ra khi việc lượng tử hóa được thực hiện trong miền tần số. SAO có thể được thực hiện bằng cách cộng hoặc trừ độ lệch được xác định xét đến mẫu hình của video thứ nhất được tái cấu trúc. Phương pháp xác định độ lệch gồm độ lệch cạnh biên (Edge Offset - EO) hoặc độ lệch băng thông. EO biểu thị phương pháp xác định độ lệch của mẫu hiện tại theo mẫu hình của các điểm ảnh xung quanh. BO biểu thị phương pháp áp dụng độ lệch chung cho tập hợp các điểm ảnh có các trị số độ sáng tương tự trong một vùng. Cụ thể là, độ sáng điểm ảnh có thể được phân chia thành 32 phần bằng nhau, và các điểm ảnh có các trị số độ sáng tương tự có thể được đặt là một tập hợp. Ví dụ như, bốn băng thông gần kề trong số 32 băng thông có thể được đặt là một nhóm, và cùng một trị số độ lệch có thể được ứng dụng cho các mẫu thuộc về bốn băng thông.

ALF là phương pháp tạo ra video thứ hai được tái cấu trúc bằng cách ứng dụng bộ lọc có kích cỡ/hình dạng được xác định trước cho video thứ nhất được tái cấu trúc hoặc video được tái cấu trúc mà bộ lọc giải khối đã được ứng dụng cho. Phương trình 21 thể hiện ví dụ về việc ứng dụng ALF.

【Phương trình 21】

$$R'(i, j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} f(k, l) \cdot R(i + k, j + l)$$

Bất kỳ một trong số các ứng viên bộ lọc được xác định trước có thể được chọn bằng đơn vị hình ảnh, đơn vị cây mã hóa, khối mã hóa, khối dự báo, hoặc khối biến đổi. Mỗi trong số các ứng viên bộ lọc có thể khác nhau về kích cỡ hoặc hình dạng.

Fig.31 là hình vẽ thể hiện các ứng viên bộ lọc được xác định trước.

Như được thể hiện trong ví dụ trên Fig.31, ít nhất một trong số các dạng hình thoi 5 x 5, 7 x 7 và 9 x 9 có thể được chọn.

Chỉ dạng hình thoi của kích cỡ 5 x 5 có thể được sử dụng đối với thành phần màu.

Vì lợi ích việc mã hóa thời gian thực hoặc độ trễ thấp các video có độ phân giải cao chẳng hạn như các video toàn cảnh, các video 360 độ, hoặc các video 4K/8K UHD (Ultra High Definition – Độ phân giải Siêu Cao), phương pháp phân chia hình ảnh thành nhiều vùng và mã hóa/giải mã nhiều vùng song song có thể được xem xét. Để thực hiện được điều này, hình ảnh có thể được phân chia thành các ngói, tức là, đơn vị cơ bản của việc mã hóa/giải mã song song, và các ngói có thể được xử lý song song.

Các ngói có thể bị ràng buộc để có dạng hình chữ nhật. Khi mã hóa/giải mã ngói, dữ liệu của các ngói khác không được sử dụng. Bảng xác suất của ngữ cảnh mã hóa số học nhị phân thích ứng ngữ cảnh (Context Adaptive Binary Arithmetic Coding - CABAC) có thể được khởi tạo bằng đơn vị của ngói, và có thể đặt không ứng dụng bộ lọc trong vòng lặp tại các ranh giới của các ngói.

Fig.32 là hình vẽ thể hiện ví dụ về phân chia hình ảnh thành nhiều ngói.

Ngói gồm ít nhất một đơn vị cây mã hóa, và ranh giới của ngói xếp chồng với ranh giới của đơn vị cây mã hóa.

Như được thể hiện trong ví dụ trên Fig.32, hình ảnh có thể được phân chia thành nhiều tập hợp ngói. Thông tin để phân chia hình ảnh thành nhiều tập hợp ngói có thể được báo hiệu thông qua dòng bit.

Theo kiểu phân chia của hình ảnh, các ngói có thể có cùng kích cỡ trong tất cả các vùng ngoại trừ các ranh giới hình ảnh.

Theo lựa chọn, hình ảnh có thể được phân chia sao cho các ngói gần kề theo hướng ngang có thể có cùng độ cao, hoặc hình ảnh có thể được phân chia sao cho các ngói gần kề theo hướng dọc có thể có cùng độ rộng.

Do hình ảnh được phân chia sử dụng ít nhất một trong số đường dọc và đường

ngang đi qua hình ảnh, mỗi trong số các ngói thuộc về cột và/hoặc hàng khác. Theo phương án được mô tả dưới đây, cột mà ngói thuộc về được gọi là cột ngói, và hàng mà ngói thuộc về được gọi là hàng ngói.

Thông tin để xác định hình dạng của phân chia hình ảnh thành các ngói có thể được báo hiệu thông qua dòng bit. Thông tin có thể được mã hóa và được báo hiệu thông qua tập hợp thông số hình ảnh hoặc tập hợp thông số trình tự. Thông tin là để xác định số lượng ngói trong hình ảnh, và có thể gồm thông tin biểu thị số lượng các hàng ngói và thông tin biểu thị số lượng cột ngói. Ví dụ như, phần tử cú pháp `num_tile_columns_minus1` biểu thị trị số thu được bằng cách trừ 1 từ số lượng cột ngói, và phần tử cú pháp `num_tile_rows_minus1` biểu thị trị số thu được bằng cách trừ 1 từ số lượng các hàng ngói.

Trong ví dụ được thể hiện trên Fig.32, do số lượng cột ngói là 4 và số lượng các hàng ngói là 3, `num_tile_columns_minus1` có thể là 3, và `num_tile_rows_minus1` có thể là 2.

Khi hình ảnh được phân chia thành nhiều ngói, thông tin biểu thị kích cỡ của ngói có thể được báo hiệu thông qua dòng bit. Ví dụ như, khi hình ảnh được phân chia thành nhiều cột ngói, thông tin biểu thị độ rộng của mỗi cột ngói được báo hiệu thông qua dòng bit, và khi hình ảnh được phân chia thành nhiều hàng ngói, thông tin biểu thị độ cao của mỗi hàng ngói được báo hiệu thông qua dòng bit. Ví dụ như, phần tử cú pháp `column_width_minus1` biểu thị độ rộng của cột ngói có thể được mã hóa và được báo hiệu cho mỗi cột ngói, và phần tử cú pháp `row_height_minus1` biểu thị độ cao của hàng ngói có thể được mã hóa và được báo hiệu cho mỗi hàng ngói.

`Column_width_minus1` có thể biểu thị trị số thu được bằng cách trừ 1 từ độ rộng của cột ngói, và `row_height_minus1` có thể biểu thị trị số thu được bằng cách trừ 1 từ độ cao của hàng ngói.

Việc mã hóa `column_width_minus1` có thể bị bỏ qua cho cột ngói cuối cùng, và việc mã hóa `row_height_minus1` có thể bị bỏ qua cho hàng ngói cuối cùng. Độ rộng của cột ngói cuối cùng và độ cao của hàng cuối cùng có thể được suy ra xét đến kích cỡ hình ảnh.

Bộ giải mã có thể xác định kích cỡ ngói dựa trên `column_width_minus1` và `row_height_minus1`.

Bảng 9 thể hiện bảng cú pháp để phân chia hình ảnh thành các ngói.

【Bảng 9】

...	
if (tile_enabled_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if (!uniform_spacing_flag) {	
for (i = 0; i < num_tile_columns_minus1: i++)	
column_width_minus1[i]	ue(v)
for (i = 0; i < num_tile_rows_minus1: i++)	
row_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

Như được thể hiện trên bảng 9, phần tử cú pháp **num_tile_columns_minus1** biểu thị số lượng cột ngói và phần tử cú pháp **num_tile_rows_minus1** biểu thị số lượng các hàng ngói có thể được báo hiệu.

Tiếp theo, phần tử cú pháp **uniform_spacing_flag** biểu thị việc liệu hình ảnh được phân chia thành các ngói có kích cỡ bằng nhau có thể được báo hiệu. Khi **uniform_spacing_flag** là đúng, các ngói ở khu vực còn lại loại trừ các ranh giới hình ảnh có thể được phân chia với kích cỡ bằng nhau.

Khi **uniform_spacing_flag** là sai, phần tử cú pháp **column_width_minus1** biểu thị độ rộng của mỗi cột ngói và phần tử cú pháp **row_height_minus1** biểu thị độ cao của mỗi hàng ngói có thể được báo hiệu.

Phần tử cú pháp **loop_filter_across_tiles_enabled_flag** biểu thị việc liệu có được phép to use a bộ lọc vòng lặp tại các ranh giới ngói.

Cột ngói có độ rộng nhỏ nhất trong số cột ngói có thể được gọi là ngói độ rộng tối thiểu, và hàng ngói có độ cao nhỏ nhất trong số các hàng ngói có thể được gọi là ngói độ cao tối thiểu. Thông tin biểu thị độ rộng của ngói độ rộng tối thiểu và thông tin biểu thị độ cao của ngói độ cao tối thiểu có thể được báo hiệu thông qua dòng bit. Ví dụ như, phần tử cú pháp **min_column_width_minus1** biểu thị trị số thu được bằng cách trừ 1 từ độ rộng của ngói độ rộng tối thiểu, và phần tử cú pháp **min_row_height_minus1** biểu thị trị số thu được bằng cách trừ 1 từ độ cao của ngói độ cao tối thiểu.

Thông tin biểu thị trị số chênh lệch với độ rộng ngói nhỏ nhất có thể được báo hiệu cho mỗi cột ngói. Ví dụ như, phần tử cú pháp `diff_column_width` biểu thị trị số chênh lệch về độ rộng giữa cột ngói hiện tại và cột ngói tối thiểu. Trị số của sự khác biệt về độ rộng có thể được biểu hiện là trị số chênh lệch về số lượng cột đơn vị cây mã hóa. Bộ giải mã có thể suy ra độ rộng của ngói hiện tại bằng cách thêm độ rộng của ngói độ rộng tối thiểu được suy ra dựa trên `min_column_width_minus1` và trị số của sự khác biệt về độ rộng được suy ra dựa trên `diff_column_width`.

Ngoài ra, thông tin biểu thị trị số chênh lệch với độ cao ngói tối thiểu có thể được báo hiệu cho mỗi hàng ngói. Ví dụ như, phần tử cú pháp `diff_row_height` biểu thị trị số chênh lệch về độ cao giữa ngói hiện tại hàng và hàng ngói tối thiểu. Trị số của sự khác biệt về độ cao có thể được biểu hiện là trị số chênh lệch về số lượng hàng đơn vị cây mã hóa. Bộ giải mã có thể suy ra độ cao của ngói hiện tại bằng cách thêm độ cao của ngói độ cao tối thiểu được suy ra dựa trên `min_row_height_minus1` và trị số của sự khác biệt về độ cao được suy ra dựa trên `diff_row_height`.

Bảng 10 thể hiện bảng cú pháp gồm thông tin về sự khác biệt kích cỡ.

【Bảng 10】

...	
<code>if (tile_enabled_flag) {</code>	
<code>num_tile_columns_minus1</code>	<code>ue(v)</code>
<code>num_tile_rows_minus1</code>	<code>ue(v)</code>
<code>uniform_spacing_flag</code>	<code>u(1)</code>
<code>if (!uniform_spacing_flag) {</code>	
<code>min_column_width_minus1</code>	
<code>min_row_height_minus1</code>	
<code>for (i = 0; i < num_tile_columns_minus1: i++)</code>	
<code>diff_column_width[i]</code>	<code>ue(v)</code>
<code>for (i = 0; i < num_tile_rows_minus1: i++)</code>	
<code>diff_row_height[i]</code>	<code>ue(v)</code>
<code>}</code>	
<code>loop_filter_across_tiles_enabled_flag</code>	<code>u(1)</code>
<code>}</code>	
...	

Hình ảnh có thể được phân chia sao cho các ngói gần kề ngang có thể có các độ cao khác nhau, hoặc hình ảnh có thể được phân chia sao cho các ngói gần kề dọc có thể có các độ rộng khác nhau. Phương pháp phân chia hình ảnh như thế này có thể được gọi là phương pháp phân chia ngói linh hoạt, và các ngói được phân chia thông qua phương pháp phân chia ngói linh hoạt có thể được gọi là các ngói linh hoạt.

Fig.33 là hình vẽ thể hiện mẫu hình phân chia của hình ảnh theo kỹ thuật xếp ngói linh hoạt.

Thứ tự tìm kiếm các ngói được tạo ra bằng phân chia hình ảnh có thể theo thứ tự quét được xác định trước. Ngoài ra, chỉ số có thể được gán cho mỗi trong số các ngói theo thứ tự quét được xác định trước.

Thứ tự quét các ngói có thể là bất kỳ một trong số quét mảnh, quét chéo, quét hướng dọc, và quét hướng ngang. Fig.33(a) đến Fig.33(d) tương ứng thể hiện các ví dụ về việc gán chỉ số cho mỗi trong số các ngói theo quét mảnh, quét chéo, quét hướng dọc, và quét hướng ngang.

Thứ tự quét tiếp theo có thể được xác định theo kích cỡ hoặc vị trí của ngói hiện tại. Ví dụ như, khi độ cao của ngói hiện tại và độ cao của ngói gần kề mặt phải của ngói hiện tại là khác nhau (ví dụ như, khi độ cao của ngói lân cận phải lớn hơn độ cao của ngói hiện tại), ngói được bố trí ở phía cực trái trong số các ngói được đặt trên đường dọc giống như đường dọc của ngói gần kề đáy của ngói hiện tại có thể được xác định là các mục tiêu quét sau khi ngói hiện tại.

Thứ tự quét các ngói có thể được xác định bằng đơn vị của hình ảnh hoặc trình tự.

Theo lựa chọn, thứ tự quét các ngói có thể được xác định xét đến kích cỡ của ngói thứ nhất trong hình ảnh. Ví dụ như, khi độ rộng của ngói thứ nhất lớn hơn độ cao, thứ tự quét các ngói có thể được đặt thành quét ngang. Khi độ cao của ngói thứ nhất lớn hơn độ rộng, thứ tự quét các ngói có thể được đặt thành quét dọc. Khi độ rộng của ngói thứ nhất giống như độ cao, thứ tự quét các ngói có thể được đặt thành quét mảnh hoặc quét chéo.

Thông tin biểu thị tổng số lượng ngói có thể được báo hiệu thông qua dòng bit. Ví dụ như, khi kỹ thuật xếp ngói linh hoạt được ứng dụng, phần tử cú pháp `number_of_tiles_in_picture_minus2` được suy ra bằng cách trừ 2 từ tổng số lượng ngói trong hình ảnh có thể được báo hiệu. Bộ giải mã có thể nhận ra số lượng số lượng ngói được đưa vào hình ảnh hiện tại dựa trên `number_of_tiles_in_picture_minus2`.

Bảng 11 thể hiện bảng cú pháp gồm thông tin về số lượng ngói.

【Bảng 11】

pic_parameter_set_rbsp() {	Bộ mô tả
...	
tiles_enabled_flag	u(1)
...	
if (tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i = 0; i < number_of_tiles_in_picture_minus2 + 2; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

Để giảm số lượng bit cần thiết để mã hóa kích cỡ của ngói, thông tin biểu thị kích cỡ của ngói con có thể được mã hóa và được báo hiệu. Ngói con là đơn vị cơ bản cấu thành ngói, và mỗi ngói có thể được tạo cấu hình để gồm ít nhất một ngói con. Ngói con có thể gồm một hoặc nhiều đơn vị cây mã hóa.

Ví dụ như, phần tử cú pháp **subtile_width_minus1** biểu thị trị số thu được bằng cách trừ 1 từ độ rộng của ngói con. Phần tử cú pháp **subtile_height_minus1** biểu thị trị số thu được bằng cách trừ 1 từ độ cao của ngói con.

Thông tin biểu thị việc liệu các ngói còn lại không phải là ngói thứ nhất có kích cỡ giống như kích cỡ của ngói trước có thể được mã hóa và được báo hiệu. Ví dụ như, phần tử cú pháp **use_previous_tile_size_flag** biểu thị việc liệu kích cỡ của ngói hiện tại giống như kích cỡ của ngói trước. Khi **use_previous_tile_size_flag** là đúng, điều đó biểu thị rằng kích cỡ của ngói hiện tại giống như kích cỡ của ngói trước. Khi **use_previous_tile_size_flag** là sai, thông tin biểu thị kích cỡ của ngói hiện tại có thể

được mã hóa và được báo hiệu. Đối với ngôi thứ nhất, việc mã hóa `use_previous_tile_size_flag` có thể là bị bỏ qua, và trị số của cờ có thể được đặt thành sai.

Thông tin biểu thị kích cỡ ngôi có thể gồm phần tử cú pháp `tile_width_minus1[i]` biểu thị độ rộng của ngôi thứ *i* và phần tử cú pháp `tile_height_minus1[i]` biểu thị độ cao của ngôi thứ *i*.

Thông tin biểu thị kích cỡ ngôi có thể biểu thị trị số chênh lệch với kích cỡ của ngôi con. Do kích cỡ thông tin của ngôi con được sử dụng, hiệu suất mã hóa/giải mã có thể được nâng cao bằng cách giảm số lượng bit cần thiết cho việc mã hóa kích cỡ của mỗi ngôi. Ví dụ như, độ rộng của ngôi thứ *i* `tileWidth` có thể được suy ra dựa trên phương trình 22 được thể hiện dưới đây, và độ cao của ngôi thứ *i* `tileHeight` có thể được suy ra dựa trên phương trình 23 được thể hiện dưới đây.

【Phương trình 22】

$$tileWidth = (subtile_width_minus1 + 1) * (tile_width_minus1[i] + 1)$$

【Phương trình 23】

$$tileHeight = (subtile_height_minus1 + 1) * (tile_height_minus[i] + 1)$$

Theo lựa chọn, việc mã hóa thông tin kích cỡ ngôi con có thể là bị bỏ qua, và kích cỡ của ngôi thứ *i* có thể được mã hóa theo hoàn cảnh thực tế như thông tin kích cỡ ngôi. Thông tin kích cỡ ngôi con có thể theo tùy chọn được mã hóa. Thông tin biểu thị việc liệu có hay không thông tin kích cỡ ngôi con được mã hóa có thể được báo hiệu thông qua tập hợp thông số video, tập hợp thông số trình tự, hoặc tập hợp thông số hình ảnh.

Thông tin liên quan đến kích cỡ ngôi được mô tả trên đây có thể được mã hóa và được báo hiệu là biểu thị số lượng các đơn vị cây mã hóa. Ví dụ như, `column_width_minus1`, `min_column_width_minus1`, `subtile_width_minus1`, hoặc `tile_width_minus1` có thể biểu thị số lượng cột đơn vị cây mã hóa được đưa vào ngôi. Ngoài ra, `diff_column_width` có thể biểu thị trị số chênh lệch giữa số lượng cột đơn vị cây mã hóa được đưa vào ngôi độ rộng tối thiểu và số lượng cột đơn vị cây mã hóa được đưa vào ngôi hiện tại.

Ngoài ra, `row_height_minus1`, `min_row_height_minus1`, `subtile_height_minus1`, hoặc `tile_height_minus1` có thể biểu thị số lượng hàng đơn vị cây mã hóa được đưa vào ngôi. Ngoài ra, `diff_row_height` có thể biểu thị trị số chênh lệch giữa số lượng hàng đơn vị cây mã hóa được đưa vào ngôi độ cao tối thiểu và số lượng hàng đơn vị cây mã hóa được đưa vào ngôi hiện tại.

Bộ giải mã có thể xác định kích cỡ của ngói dựa trên số lượng cột đơn vị cây mã hóa và/hoặc số lượng hàng đơn vị cây mã hóa được suy ra dựa trên các phần tử cú pháp và kích cỡ của đơn vị cây mã hóa. Ví dụ như, độ rộng của ngói thứ i có thể được đặt thành $(\text{tile_width_minus1}[i] + 1) \times (\text{độ rộng của đơn vị cây mã hóa})$, và độ cao của ngói thứ i có thể được đặt thành $(\text{tile_height_minus1}[i] + 1) \times (\text{độ cao của đơn vị cây mã hóa})$.

Trong khi đó, thông tin biểu thị kích cỡ của đơn vị cây mã hóa có thể được báo hiệu thông qua tập hợp thông số trình tự hoặc tập hợp thông số hình ảnh.

Trong bảng 11, nội dung được mô tả là phần tử cú pháp `use_previous_tile_size_flag` biểu thị việc liệu kích cỡ của ngói hiện tại giống như kích cỡ của ngói trước được sử dụng. Theo một ví dụ khác, thông tin biểu thị việc liệu độ rộng của ngói hiện tại giống như độ rộng của ngói trước hoặc thông tin biểu thị việc liệu độ cao của ngói hiện tại giống như độ cao của ngói trước có thể được mã hóa và được báo hiệu.

Bảng 12 thể hiện bảng cú pháp gồm thông tin biểu thị việc liệu độ rộng của ngói hiện tại giống như độ rộng của ngói trước.

【Bảng 12】

<code>pic_parameter_set_rbsp() {</code>	Bộ mô tả
<code>...</code>	
<code>tiles_enabled_flag</code>	<code>u(1)</code>
<code>...</code>	
<code>CoveredWidthByTile = 0;</code>	
<code>CoveredHeightByTile = 0;</code>	
<code>PrevWidthByTile = 0;</code>	
<code>PrevHeightByTile = 0;</code>	
<code>if (tiles_enabled_flag) {</code>	
<code> number_of_tiles_in_picture_minus2</code>	<code>ue(v)</code>
<code> for (i = 0; i < number_of_tiles_in_picture_minus2 + 2; i++) {</code>	
<code> if (i > 0)</code>	
<code> use_previous_tile_size_flag</code>	<code>u(1)</code>
<code> if (use_previous_tile_size_flag == 0) {</code>	
<code> use_previous_tile_width_flag</code>	<code>ue(v)</code>
<code> if (use_previous_tile_width_flag == 0) {</code>	<code>ue(v)</code>

tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
} else {	
tile_height_minus1[i]	ue(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

Phần tử cú pháp `use_previous_tile_width_flag` biểu thị việc liệu độ rộng của ngói hiện tại giống như độ rộng của ngói trước. Khi `use_previous_tile_width_flag` là đúng, độ rộng của ngói hiện tại có thể được đặt giống như độ rộng của ngói trước. Trong trường hợp này, việc mã hóa thông tin biểu thị độ rộng của ngói hiện tại có thể là bị bỏ qua, và độ rộng của ngói hiện tại có thể được suy ra từ độ rộng của ngói trước.

Khi `use_previous_tile_width_flag` là sai, thông tin biểu thị độ rộng của ngói hiện tại có thể được báo hiệu. Ví dụ như, `tile_width_minus1[i]` có thể biểu thị trị số thu được bằng cách trừ 1 từ độ rộng của ngói thứ *i*.

Phần tử cú pháp `use_previous_tile_width_flag` có thể được mã hóa và được báo hiệu chỉ khi nội dung được xác định là kích cỡ của ngói hiện tại khác với kích cỡ của ngói trước (ví dụ như, khi trị số của `use_previous_tile_size_flag` là 0).

`Tile_width_minus1[i]` có thể có trị số thu được bằng cách trừ 1 từ số lượng cột đơn vị cây mã hóa được đưa vào ngói thứ *i*. Bộ giải mã có thể suy ra số lượng cột đơn vị cây mã hóa thuộc về ngói thứ *i* bằng cách thêm 1 vào `tile_width_minus1[i]`, và tính toán độ rộng ngói bằng cách nhân trị số được suy ra bởi độ rộng của đơn vị cây mã hóa.

Bảng 13 thể hiện bảng cú pháp còn gồm thông tin biểu thị việc liệu độ cao của ngói hiện tại giống như độ cao của ngói trước.

【Bảng 13】

pic_parameter_set_rbsp() {	Bộ mô tả
...	
tiles_enabled_flag	u(1)
...	
CoveredWidthByTile = 0;	

CoveredHeightByTile = 0;	
PrevWidthByTile = 0;	
PrevHeightByTile = 0;	
if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
for (i = 0; i < number_of_tiles_in_picture_minus2 + 2; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
use_previous_tile_width_flag	ue(v)
if (use_previous_tile_width_flag == 0) {	ue(v)
tile_width_minus1[i]	ue(v)
tile_width_minus1[i] += 1;	
use_previous_tile_height_flag	
if (use_previous_tile_height_flag == 0){	
tile_height_minus1[i]	ue(v)
tile_height_minus1[i] += 1;	
}	
} else {	
use_previous_tile_height_flag	
if (use_previous_tile_height_flag == 0){	
tile_height_minus1[i]	ue(v)
tile_height += 1;	
}	
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

Phần tử cú pháp use_previous_tile_height_flag biểu thị việc liệu độ cao của ngói hiện tại giống như độ cao của ngói trước. Khi use_previous_tile_height_flag là đúng, độ

cao của ngói hiện tại có thể được đặt giống như độ cao của ngói trước. Trong trường hợp này, việc mã hóa thông tin biểu thị độ cao của ngói hiện tại có thể là bị bỏ qua, và độ cao của ngói hiện tại có thể được suy ra từ độ cao của ngói trước.

Khi `use_previous_tile_height_flag` là sai, thông tin biểu thị độ cao của ngói hiện tại có thể được báo hiệu. Ví dụ như, `tile_height_minus1[i]` có thể biểu thị trị số thu được bằng cách trừ 1 từ độ cao của ngói thứ *i*.

Phần tử cú pháp `use_previous_tile_height_flag` có thể được mã hóa và được báo hiệu chỉ khi nội dung được xác định là kích cỡ của ngói hiện tại khác với kích cỡ của ngói trước (ví dụ như, khi trị số của `use_previous_tile_size_flag` là 0). Ngoài ra, phần tử cú pháp `use_previous_tile_height_flag` có thể được báo hiệu chỉ khi `use_previous_tile_width_flag` là sai.

Bảng 12 thể hiện ví dụ về trường hợp sử dụng `use_previous_tile_width_flag`, và bảng 13 thể hiện ví dụ về trường hợp sử dụng `use_previous_tile_width_flag` và `use_previous_tile_height_flag`. Mặc dù không được thể hiện trong các bảng nêu trên, việc mã hóa `use_previous_tile_width_flag` có thể là bị bỏ qua, và chỉ `use_previous_tile_height_flag` có thể được sử dụng.

Phần tử cú pháp nào trong số `use_previous_tile_height_flag` và `use_previous_tile_size_flag` sẽ được sử dụng có thể được xác định dựa trên ít nhất một trong số thứ tự quét ngói, độ rộng và độ cao của ngói thứ nhất, và độ rộng và độ cao của ngói trước. Ví dụ như, khi thứ tự quét ngói là hướng dọc, `use_previous_tile_height_flag` có thể được sử dụng, trong khi khi thứ tự quét ngói là hướng ngang, `use_previous_tile_width_flag` có thể được sử dụng. Theo lựa chọn, khi ngói thứ nhất hoặc ngói trước là hình dạng không phải là hình vuông có độ rộng lớn hơn độ cao, `use_previous_tile_width_flag` có thể được sử dụng, trong khi khi ngói thứ nhất hoặc ngói trước là hình dạng không phải là hình vuông có độ cao lớn hơn độ rộng, `use_previous_tile_height_flag` có thể được sử dụng.

Trong khi báo hiệu số lượng ngói được đưa vào hình ảnh, việc mã hóa thông tin liên quan đến kích cỡ ngói có thể bị bỏ qua cho ngói cuối cùng.

Bảng 14 thể hiện ví dụ trong đó việc mã hóa thông tin kích cỡ ngói bị bỏ qua cho ngói cuối cùng.

【Bảng 14】

<code>pic_parameter_set_rbsp() {</code>	Bộ mô tả
--	----------

...	
tiles_enabled_flag	u(1)
...	
if (tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i = 0; i < number_of_tiles_in_picture_minus2 + 1; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

Khi kích cỡ của các ngói không phải là ngói cuối cùng được định rõ, vùng còn lại trong ảnh có thể được đặt là ngói cuối cùng.

Cho mỗi đơn vị cây mã hóa, ký hiệu nhận dạng để nhận dạng ngói mà đơn vị cây mã hóa thuộc về có thể được gán (trong phần dưới đây, được gọi là ID ngói-TileID).

Fig.34 là hình vẽ thể hiện ví dụ về gán ID ngói cho mỗi đơn vị cây mã hóa.

Cùng một ID ngói có thể được gán cho các đơn vị cây mã hóa thuộc về cùng một ngói. Cụ thể là, N-th TileID có thể được gán cho các đơn vị cây mã hóa thuộc về Tile N.

Các biến x và y biểu thị vị trí của đơn vị cây mã hóa trong hình ảnh có thể được xác định để xác định ID ngói được gán cho mỗi đơn vị cây mã hóa. Ở đây, x biểu thị trị số thu được bằng cách chia tọa độ trục x trong số vị trí (x0, y0) của mẫu trên cùng-bên trái của đơn vị cây mã hóa bởi độ rộng của đơn vị cây mã hóa, và y biểu thị trị số thu được bằng cách chia tọa độ trục y trong số vị trí (x0, y0) của mẫu trên cùng-bên trái của đơn vị cây mã hóa bởi độ cao của đơn vị cây mã hóa. Cụ thể là, x và y có thể được suy ra bởi các phương trình 24 và 25 được thể hiện dưới đây.

【Phương trình 24】

$$x = (x0 / (CTU\ width))$$

【Phương trình 25】

$$y = (y0 / (CTU\ height))$$

Gán ID ngói cho mỗi đơn vị cây mã hóa có thể được thực hiện thông qua quy trình được mô tả dưới đây.

i) Khởi tạo ID ngói

ID ngói của mỗi đơn vị cây mã hóa có thể được khởi tạo thành trị số thu được bằng cách trừ 1 từ số lượng ngói trong hình ảnh.

【Bảng 15】

Nếu `tiles_enabled_flag` là bằng 1, trị số của biến số `tile_id` và trị số của mảng hai chiều `TileId` được định rõ như sau:

```
tile_id = 0
```

```
đối với (y = 0; y < PicHeightInCtbsY; y++)
```

```
- đối với (x = 0; x < PicWidthInCtbsY; x++)
```

```
-- TileId[x][y] = number_of_tiles_in_picture_minus2 + 1
```

```
tile_height_minus1[number_of_tiles_in_picture_minus2 + 1] = 0
```

```
tile_width_minus1[number_of_tiles_in_picture_minus2 + 1] = 0
```

ii) Suy ra ID ngói

【Bảng 16】

Trị số của biến số `tile_id` và trị số của phép gán `CTU_tile` mảng hai chiều được suy ra như sau:

```
đối với (ctu_y = 0; ctu_y < (tile_height_minus1[i] + 1)*(subtile_height_minus1 + 1) &&
(tile_coordinate_y + ctu_y) < PicHeightInCtbsY; ctu_y++)
```

```
- đối với (ctu_x = 0; ctu_x < (tile_width_minus1[i] + 1)*(subtile_width_minus1 + 1) &&
(tile_coordinate_x + ctu_x) < PicWidthInCtbsY; ctu_x++)
```

```
-- TileId[tile_coordinate_x + ctu_x][tile_coordinate_y + ctu_y] = tile_id
```

```
tile_id++
```

Theo phương án được mô tả trên đây, nội dung đã được mô tả là cờ biểu thị việc liệu có được phép để ứng dụng bộ lọc trong vòng lặp tại các ranh giới của các ngói được

báo hiệu thông qua tập hợp thông số hình ảnh. Tuy nhiên, khi đặt không sử dụng bộ lọc trong vòng lặp tại tất cả các ranh giới ngói, có thể có vấn đề là làm giảm chất lượng hình ảnh chủ quan và làm giảm hiệu suất giải mã.

Do đó, thông tin biểu thị việc liệu có hay không để cho phép bộ lọc trong vòng lặp cho mỗi ngói có thể được mã hóa và được báo hiệu.

Fig.35 là hình vẽ thể hiện ví dụ về xác định có chọn lọc xem liệu có ứng dụng bộ lọc trong vòng lặp cho mỗi ngói hay không.

Như được thể hiện trong ví dụ trên Fig.35, việc liệu có hay không để cho phép bộ lọc trong vòng lặp (ví dụ như, bộ lọc giải khối, SAO và/hoặc ALF) tại ranh giới dọc hoặc ngang có thể được xác định cho mỗi ngói.

Bảng 17 thể hiện ví dụ trong đó thông tin biểu thị việc liệu có hay không để cho phép bộ lọc trong vòng lặp được mã hóa cho mỗi ngói.

【Bảng 17】

pic_parameter_set_rbsp() {	Bộ mô tả
...	
tiles_enabled_flag	u(1)
...	
if (tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i = 0; i < number_of_tiles_in_picture_minus2 + 1; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_flag[i]	u(1)
}	
}	

...	
-----	--

Trong ví dụ của bảng 17, phần tử cú pháp `loop_filter_across_tiles_flag[i]` biểu thị việc liệu có được phép để ứng dụng bộ lọc trong vòng lặp cho ngói thứ *i*. Khi trị số của `loop_filter_across_tiles_flag[i]` là 1, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể được sử dụng tại các ranh giới dọc và ngang của ngói có ID ngói là *i*. Khi trị số của `loop_filter_across_tiles_flag[i]` là 0, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể không được sử dụng tại các ranh giới dọc và ngang của ngói có ID ngói là *i*.

Thông tin biểu thị việc liệu có hay không để cho phép bộ lọc trong vòng lặp cho mỗi trong số các hướng dọc và ngang có thể được mã hóa.

Bảng 18 thể hiện ví dụ trong đó thông tin biểu thị việc liệu có hay không để cho phép bộ lọc trong vòng lặp được mã hóa riêng cho hướng ngang và hướng dọc.

【Bảng 18】

<code>pic_parameter_set_rbsp() {</code>	Bộ mô tả
...	
<code>tiles_enabled_flag</code>	<code>u(1)</code>
..	
<code>if (tiles_enabled_flag) {</code>	
<code>number_of_tiles_in_picture_minus2</code>	<code>ue(v)</code>
<code>subtile_width_minus1</code>	<code>ue(v)</code>
<code>subtile_height_minus1</code>	<code>ue(v)</code>
<code>for (i = 0; i < number_of_tiles_in_picture_minus2 + 1; i++) {</code>	
<code>if (i > 0)</code>	
<code>use_previous_tile_size_flag</code>	<code>u(1)</code>
<code>if (use_previous_tile_size_flag == 0) {</code>	
<code>tile_width_minus1[i]</code>	<code>ue(v)</code>
<code>tile_height_minus1[i]</code>	<code>ue(v)</code>
<code>}</code>	
<code>loop_filter_hor_across_tiles_flag[i]</code>	<code>u(1)</code>
<code>loop_filter_ver_across_tiles_flag[i]</code>	<code>u(1)</code>
<code>}</code>	
<code>}</code>	

...	
-----	--

Trong ví dụ của bảng 18, phần tử cú pháp `loop_filter_hor_across_tiles_flag[i]` biểu thị việc liệu có được phép để ứng dụng bộ lọc trong vòng lặp đi qua ngói thứ i theo hướng ngang. Phần tử cú pháp `loop_filter_ver_across_tiles_flag[i]` biểu thị việc liệu có được phép để ứng dụng bộ lọc trong vòng lặp đi qua ngói thứ i theo hướng dọc.

Khi trị số của `loop_filter_hor_across_tiles_flag[i]` là 1, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể được sử dụng tại ranh giới ngang của ngói có ID ngói là i . Khi trị số của `loop_filter_hor_across_tiles_flag[i]` là 0, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể không được sử dụng tại ranh giới ngang của ngói có ID ngói là i .

Khi trị số của `loop_filter_ver_across_tiles_flag[i]` là 1, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể được sử dụng tại ranh giới dọc của ngói có ID ngói là i , và khi trị số của `loop_filter_ver_across_tiles_flag[i]` là 0, điều đó biểu thị rằng bộ lọc trong vòng lặp có thể không được sử dụng tại ranh giới dọc của ngói có ID ngói là i .

Theo lựa chọn, thông tin biểu thị việc liệu bộ lọc trong vòng lặp được phép cho nhóm ngói gồm nhiều ngói có thể được mã hóa và được báo hiệu. Việc liệu có hay không để cho phép bộ lọc trong vòng lặp cho nhiều ngói được đưa vào nhóm ngói có thể được xác định by thông tin.

Để xác định nhóm ngói, ít nhất một trong số số lượng ngói thuộc về nhóm ngói, kích cỡ của nhóm ngói, và thông tin phân chia của hình ảnh có thể được báo hiệu thông qua dòng bit. Theo lựa chọn, vùng có kích cỡ được xác định trước trong bộ mã hóa và bộ giải mã có thể được đặt là nhóm ngói.

Việc mã hóa thông tin biểu thị việc liệu bộ lọc trong vòng lặp được phép có thể là bị bỏ qua, và việc liệu có hay không để cho phép bộ lọc trong vòng lặp có thể được xác định dựa trên ít nhất một trong số số lượng đơn vị cây mã hóa được đưa vào ngói, độ rộng của ngói, và độ cao của ngói. Ví dụ như, khi độ rộng ngói nhỏ hơn trị số tham chiếu, bộ lọc trong vòng lặp cho hướng ngang có thể được phép, và khi độ cao ngói nhỏ hơn trị số tham chiếu, bộ lọc trong vòng lặp cho hướng dọc có thể được phép.

Khi bộ lọc trong vòng lặp được sử dụng tại ranh giới ngói, dữ liệu được tái cấu trúc bên ngoài ngói có thể được tạo ra dựa trên dữ liệu được đưa vào ngói. Tại điểm này, video được tái cấu trúc bên ngoài ngói có thể là thu được đệm hoặc nội suy dữ liệu được đưa vào ngói. Sau đó, bộ lọc trong vòng lặp có thể được ứng dụng sử dụng dữ liệu được tái cấu trúc bên ngoài ngói.

Ứng dụng các phương án được mô tả trên đây tập trung vào quy trình giải mã hoặc

quy trình mã hóa cho quy trình mã hóa hoặc quy trình giải mã được đưa vào phạm vi của sáng chế. Việc thay đổi các phương án được mô tả theo thứ tự được xác định trước theo thứ tự khác với thứ tự được mô tả cũng được đưa vào phạm vi của sáng chế.

Mặc dù các phương án nêu trên đã được mô tả dựa trên một loạt các bước hoặc các lưu đồ, điều này không giới hạn thứ tự chuỗi thời gian của sáng chế, và có thể được thực hiện đồng thời hoặc theo thứ tự khác nhau theo yêu cầu. Ngoài ra, mỗi trong số các thành phần (ví dụ như, các đơn vị, các môđun, v.v.) cấu thành sơ đồ khối trong các phương án được mô tả trên đây có thể được thực hiện là thiết bị phần cứng hoặc phần mềm, hoặc nhiều thành phần có thể được kết hợp cần được thực hiện là thiết bị phần cứng đơn lẻ hoặc phần mềm. Các phương án được mô tả trên đây có thể được thực hiện dưới dạng các lệnh chương trình có thể được thực hiện thông qua các thành phần máy tính khác nhau và được ghi theo phương tiện ghi có thể đọc được bằng máy tính. Phương tiện ghi có thể đọc được bằng máy tính có thể gồm các lệnh chương trình, các tệp dữ liệu, các cấu trúc dữ liệu phương tiện tương tự độc lập hoặc kết hợp với nhau. Phương tiện ghi có thể đọc được bằng máy tính gồm, ví dụ như, phương tiện từ tính chẳng hạn như đĩa cứng, đĩa mềm và băng từ tính, phương tiện ghi quang chẳng hạn như CD-ROM và DVD, phương tiện từ quang chẳng hạn như đĩa mềm quang học, và các thiết bị phần cứng được tạo cấu hình cụ thể để lưu trữ và thực hiện các lệnh chương trình, chẳng hạn như ROM, RAM, bộ nhớ cực nhanh và thiết bị tương tự. Các thiết bị phần cứng được mô tả trên đây có thể được tạo cấu hình để vận hành bằng cách sử dụng một hoặc nhiều môđun phần mềm để thực hiện quy trình của sáng chế, và ngược lại.

Sáng chế có thể được ứng dụng cho thiết bị điện tử dùng để mã hóa và giải mã video.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã video bao gồm:

bước tạo ra danh mục ứng viên hợp nhất đối với khối hiện tại;

bước định rõ một trong số nhiều ứng viên hợp nhất được đưa vào danh mục ứng viên hợp nhất;

bước suy ra vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được định rõ;

bước suy ra vector affin đối với khối con trong khối hiện tại bằng cách sử dụng vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại, trong đó khối con là vùng có kích cỡ nhỏ hơn kích cỡ của khối hiện tại; và

bước thực hiện dự báo bù chuyển động cho khối con dựa trên vector affin, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên thông tin chuyển động của khối lân cận gần kề khối hiện tại;

trong đó khi khối lân cận được đưa vào đơn vị cây mã hóa khác với đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải của khối lân cận.

2. Phương pháp theo điểm 1, trong đó khối con dưới cùng-bên trái gồm mẫu tham chiếu dưới cùng-bên trái được bố trí tại góc dưới cùng-bên trái của khối lân cận, và khối con dưới cùng-bên phải gồm mẫu tham chiếu dưới cùng-bên phải được bố trí tại góc dưới cùng-bên trái của khối lân cận.

3. Phương pháp theo điểm 2, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên trị số thu được dựa trên phép toán dịch chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất được suy ra dựa trên trị số thu được bằng cách thêm khoảng cách ngang giữa mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải và độ lệch.

4. Phương pháp theo điểm 2, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên trị số thu được dựa trên phép toán dịch

chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất được suy ra dựa trên khoảng cách giữa mẫu lân cận gần kề mặt phải của mẫu tham chiếu dưới cùng-bên phải và mẫu tham chiếu dưới cùng-bên trái.

5. Phương pháp theo điểm 1, trong đó danh mục ứng viên hợp nhất gồm ứng viên hợp nhất thứ nhất được suy ra dựa trên khối lân cận đỉnh được xác định là khối sẵn có thứ nhất trong số các khối lân cận đỉnh được bố trí trên đỉnh của khối hiện tại, và ứng viên hợp nhất thứ hai được suy ra dựa trên khối lân cận trái được xác định là khối sẵn có thứ nhất trong số các khối lân cận trái được bố trí trên mặt trái của khối hiện tại.

6. Phương pháp theo điểm 1, trong đó khi khối lân cận được đưa vào đơn vị cây mã hóa giống như đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của khối lân cận.

7. Phương pháp theo điểm 1, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại được suy ra sử dụng vector hạt affin thứ ba cho điểm điều khiển dưới cùng-bên trái và vector hạt affin thứ tư cho điểm điều khiển dưới cùng-bên phải của khối lân cận affin

$$\begin{aligned} v_{0x} &= \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x} \\ v_{0y} &= \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y} \\ v_{1x} &= \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x} \\ v_{1y} &= \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y} \end{aligned}$$

(x_{n2}, y_{n2}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên trái của khối lân cận affin, và (x_{n3}, y_{n3}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên phải của khối lân cận affin;

(x_0, y_0) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên trái của khối hiện tại, và (x_1, y_1) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên phải của khối hiện tại;

(nv_{2x}, nv_{2y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên trái của

vector hạt affin thứ ba, và (nv_{3x}, nv_{3y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên phải của vector hạt affin thứ tư;

(v_{0x}, v_{0y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên trái của vector hạt affin thứ nhất, và (v_{1x}, v_{1y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên phải của vector hạt affin thứ hai.

8. Phương pháp mã hóa video bao gồm:

bước tạo ra danh mục ứng viên hợp nhất đối với khối hiện tại;

bước định rõ một trong số nhiều ứng viên hợp nhất được đưa vào danh mục ứng viên hợp nhất;

bước suy ra vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được định rõ;

bước suy ra vector affin đối với khối con trong khối hiện tại bằng cách sử dụng vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại, trong đó khối con là vùng có kích cỡ nhỏ hơn kích cỡ của khối hiện tại; và

bước thực hiện dự báo bù chuyển động cho khối con dựa trên vector affin, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên thông tin chuyển động của khối lân cận gần kề khối hiện tại.

trong đó khi khối lân cận được đưa vào đơn vị cây mã hóa khác với đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải của khối lân cận.

9. Phương pháp theo điểm 8, trong đó khối con dưới cùng-bên trái gồm mẫu tham chiếu dưới cùng-bên trái được bố trí tại góc dưới cùng-bên trái của khối lân cận, và khối con dưới cùng-bên phải gồm mẫu tham chiếu dưới cùng-bên phải được bố trí tại góc dưới cùng-bên trái của khối lân cận.

10. Phương pháp theo điểm 9, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên trị số thu được dựa trên phép toán dịch chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất được

suy ra dựa trên trị số thu được bằng cách thêm khoảng cách ngang giữa mẫu tham chiếu dưới cùng-bên trái và mẫu tham chiếu dưới cùng-bên phải và độ lệch.

11. Phương pháp theo điểm 9, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên trị số thu được dựa trên phép toán dịch chuyển được thực hiện trên trị số chênh lệch giữa các vector chuyển động của khối con dưới cùng-bên trái và khối con dưới cùng-bên phải sử dụng bội suất, và bội suất được suy ra dựa trên khoảng cách giữa mẫu lân cận gần kề mặt phải của mẫu tham chiếu dưới cùng-bên phải và mẫu tham chiếu dưới cùng-bên trái.

12. Phương pháp theo điểm 8, trong đó danh mục ứng viên hợp nhất gồm ứng viên hợp nhất thứ nhất được suy ra dựa trên khối lân cận đỉnh được xác định là khối sẵn có thứ nhất trong số các khối lân cận đỉnh được bố trí trên đỉnh của khối hiện tại, và ứng viên hợp nhất thứ hai được suy ra dựa trên khối lân cận trái được xác định là khối sẵn có thứ nhất trong số các khối lân cận trái được bố trí trên mặt trái của khối hiện tại.

13. Phương pháp theo điểm 8, trong đó khi khối lân cận được đưa vào đơn vị cây mã hóa giống như đơn vị cây mã hóa của khối hiện tại, vector hạt affin thứ nhất và vector hạt affin thứ hai của ứng viên hợp nhất được suy ra dựa trên vector hạt affin thứ nhất và vector hạt affin thứ hai của khối lân cận.

14. Phương pháp theo điểm 8, trong đó vector hạt affin thứ nhất và vector hạt affin thứ hai của khối hiện tại được suy ra sử dụng vector hạt affin thứ ba cho điểm điều khiển dưới cùng-bên trái và vector hạt affin thứ tư cho điểm điều khiển dưới cùng-bên phải của khối lân cận affin

$$v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y}$$

$$v_{1x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y}$$

(x_{n2}, y_{n2}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên trái của khối lân cận

affin, và (x_{n3}, y_{n3}) biểu thị hệ tọa độ của điểm điều khiển dưới cùng-bên phải của khối lân cận affin;

(x_0, y_0) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên trái của khối hiện tại, và (x_1, y_1) biểu thị hệ tọa độ của điểm điều khiển trên cùng-bên phải của khối hiện tại;

(nv_{2x}, nv_{2y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên trái của vector hạt affin thứ ba, và (nv_{3x}, nv_{3y}) biểu thị vector hạt affin của điểm điều khiển dưới cùng-bên phải của vector hạt affin thứ tư;

(v_{0x}, v_{0y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên trái của vector hạt affin thứ nhất, và (v_{1x}, v_{1y}) biểu thị vector hạt affin của điểm điều khiển trên cùng-bên phải của vector hạt affin thứ hai.

15. Bộ giải mã video được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 1 đến điểm 7.

16. Bộ mã hóa video được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ điểm 8 đến điểm 14.

17. Phương tiện đọc được bằng máy tính không tạm thời có lưu trữ trên đó các lệnh mà khi được thực hiện bởi bộ xử lý, làm cho bộ xử lý thực hiện phương pháp giải mã video theo điểm bất kỳ trong số các điểm từ điểm 1 đến điểm 7.

18. Phương tiện đọc được bằng máy tính không tạm thời có lưu trữ trên đó các lệnh mà khi được thực hiện bởi bộ xử lý, làm cho bộ xử lý thực hiện phương pháp mã hóa video theo điểm bất kỳ trong số các điểm từ điểm 8 đến điểm 14.

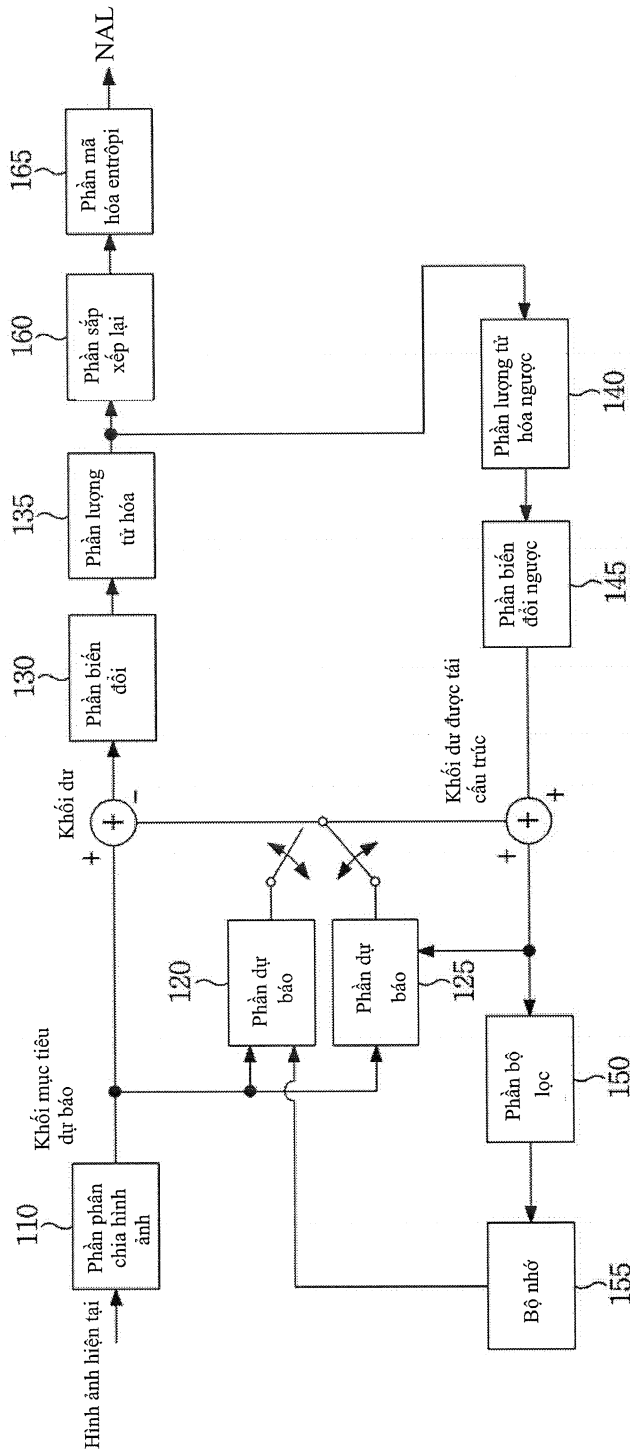


Fig.1

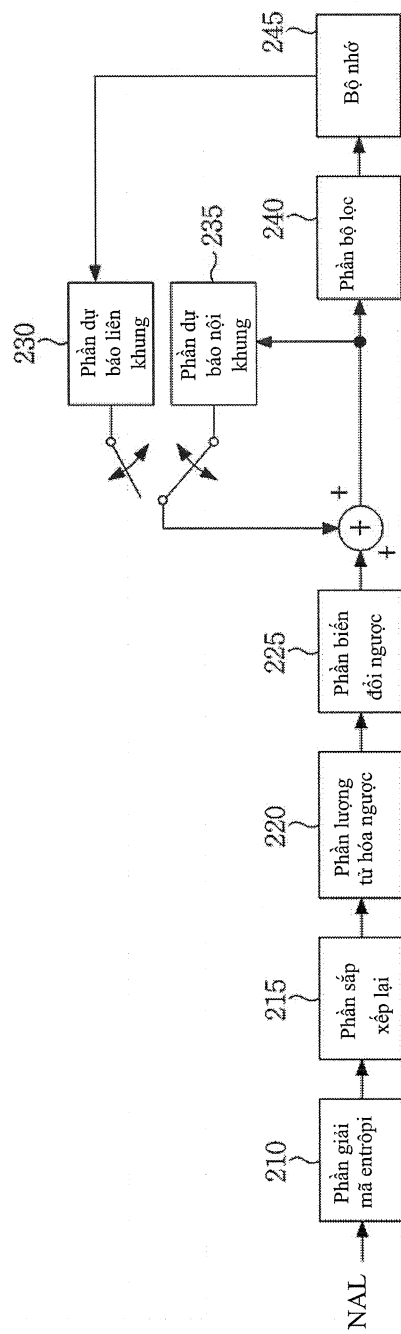


Fig.2

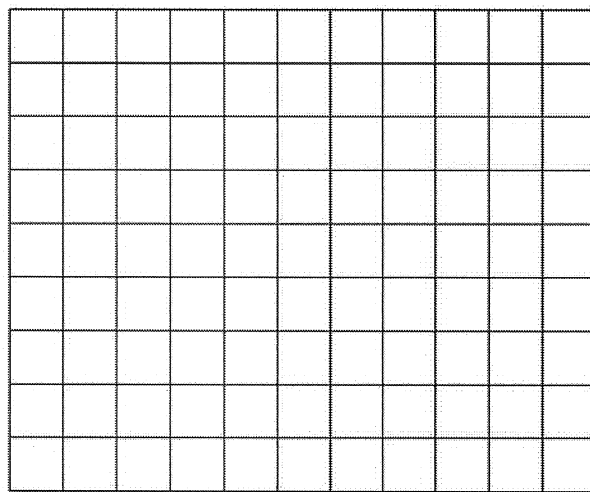


Fig.3

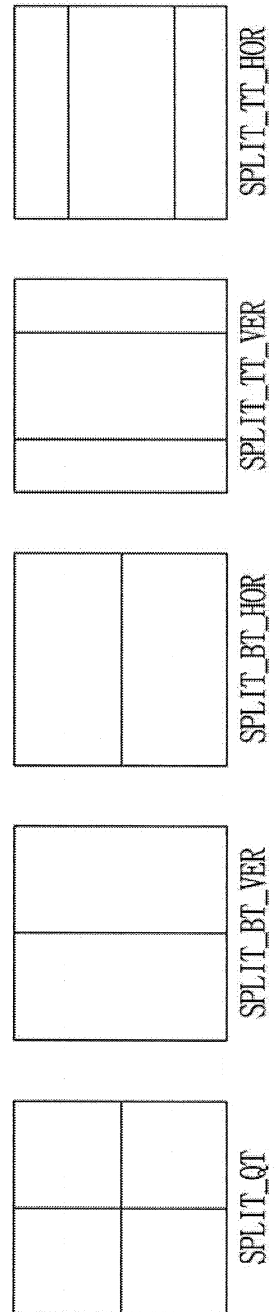


Fig.4

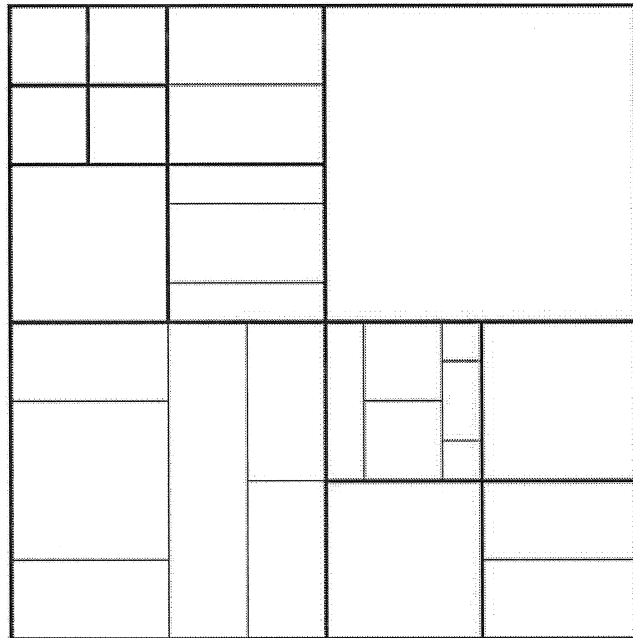


Fig.5

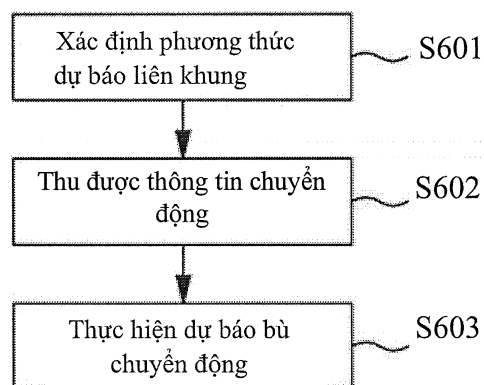


Fig.6

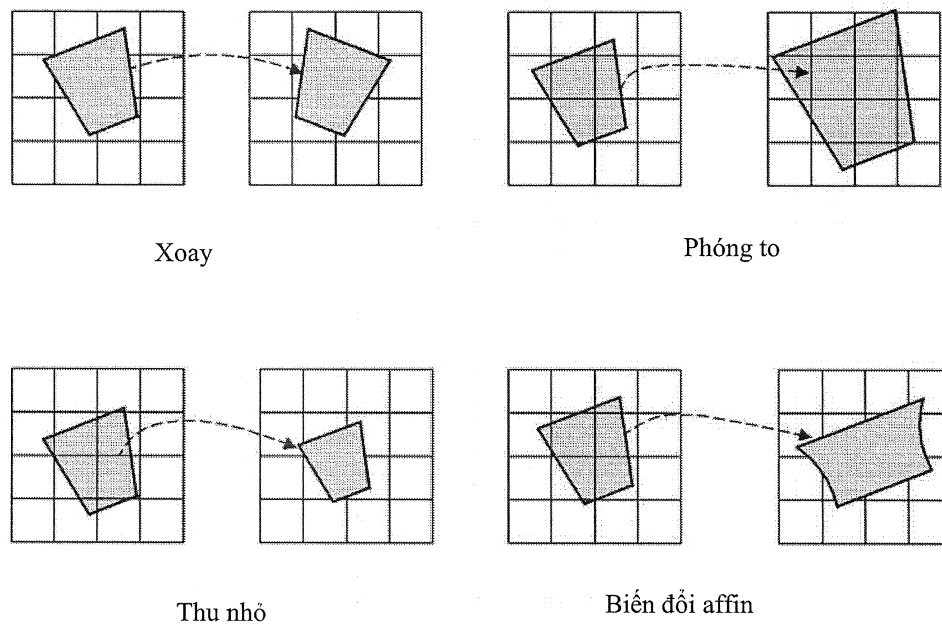


Fig.7

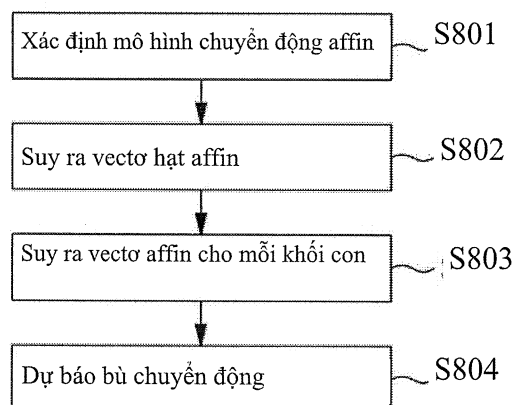


Fig.8

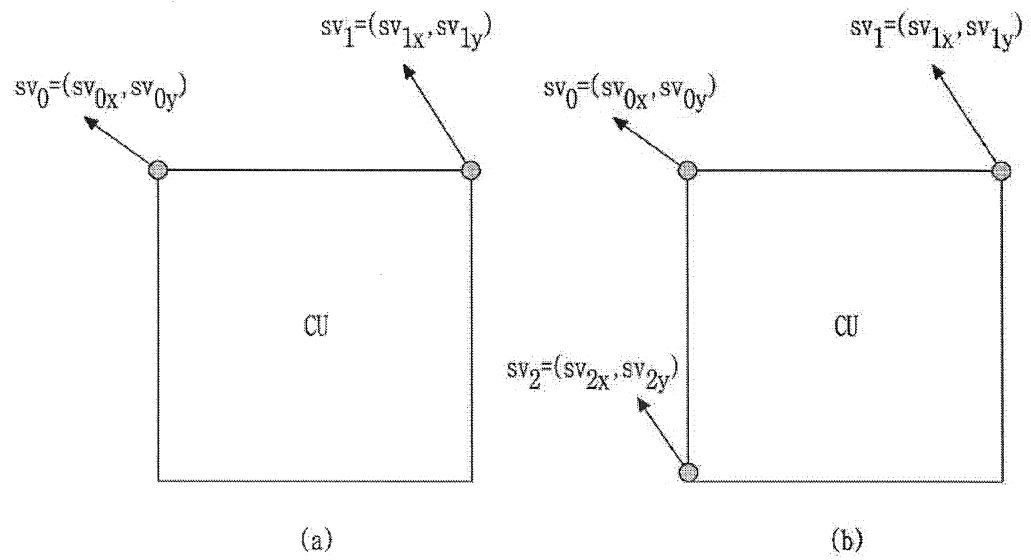


Fig.9

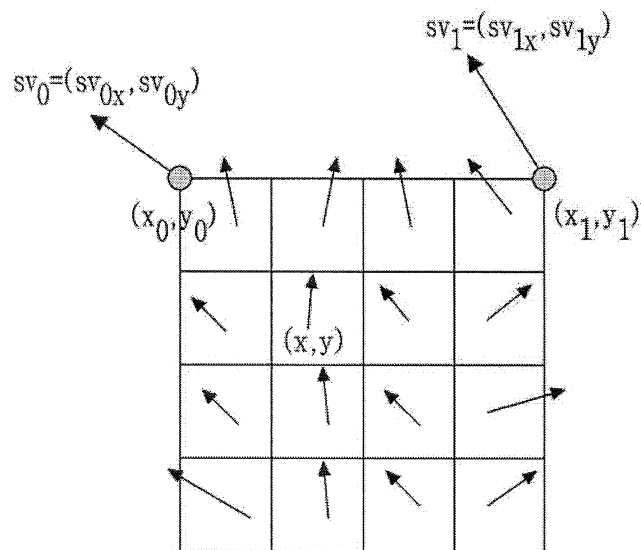


Fig.10

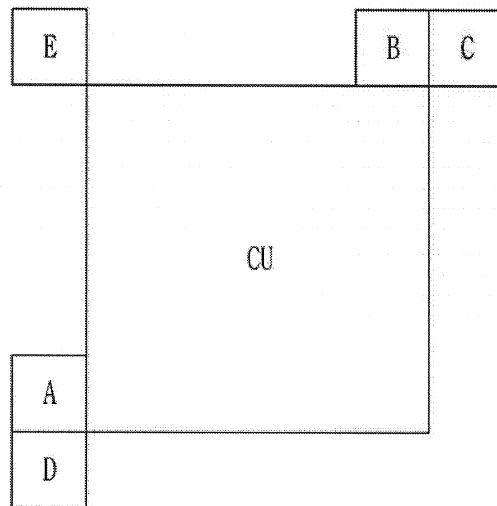


Fig.11

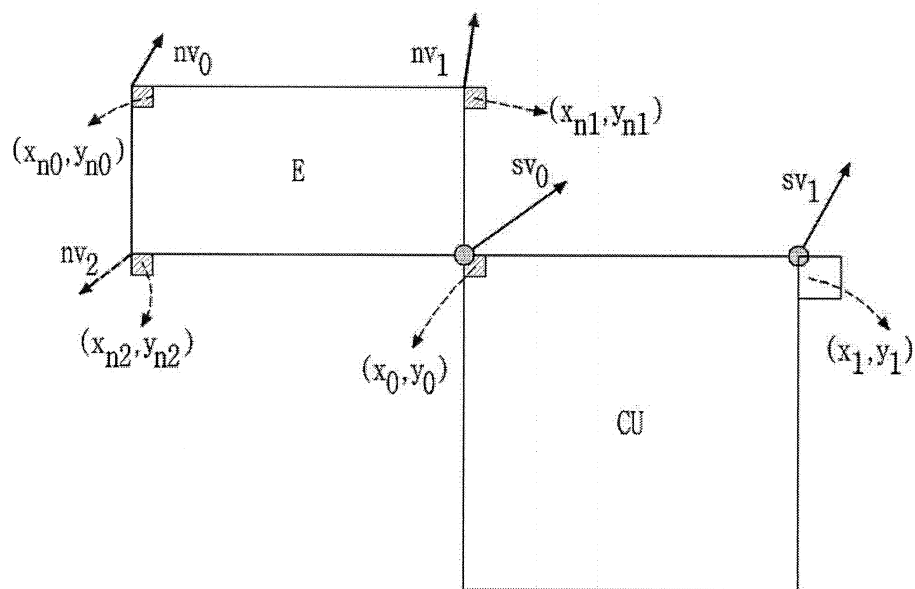


Fig.12

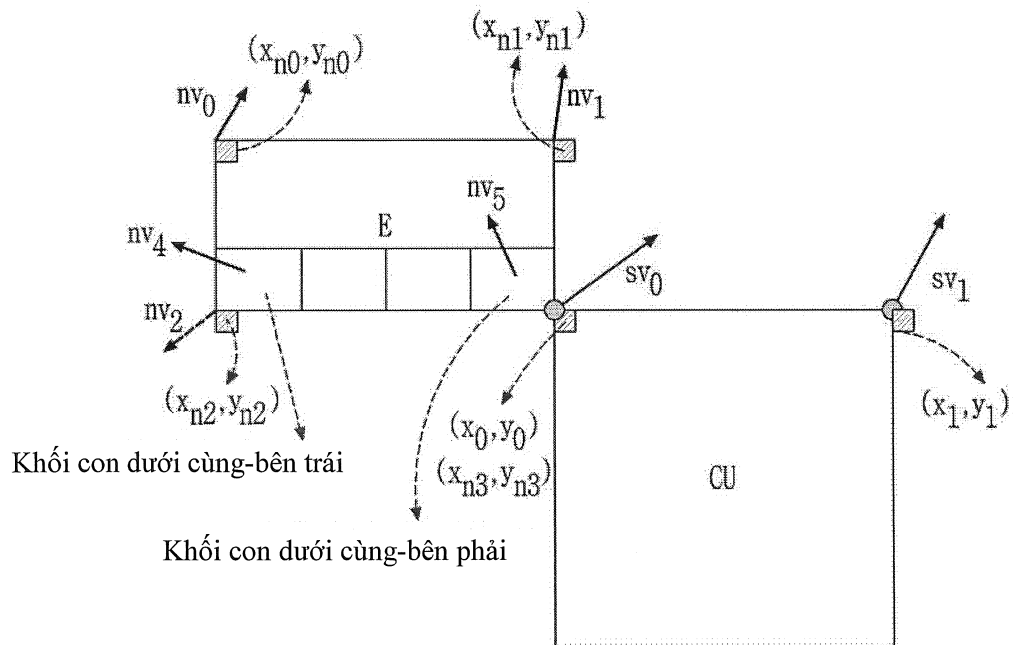


Fig.13

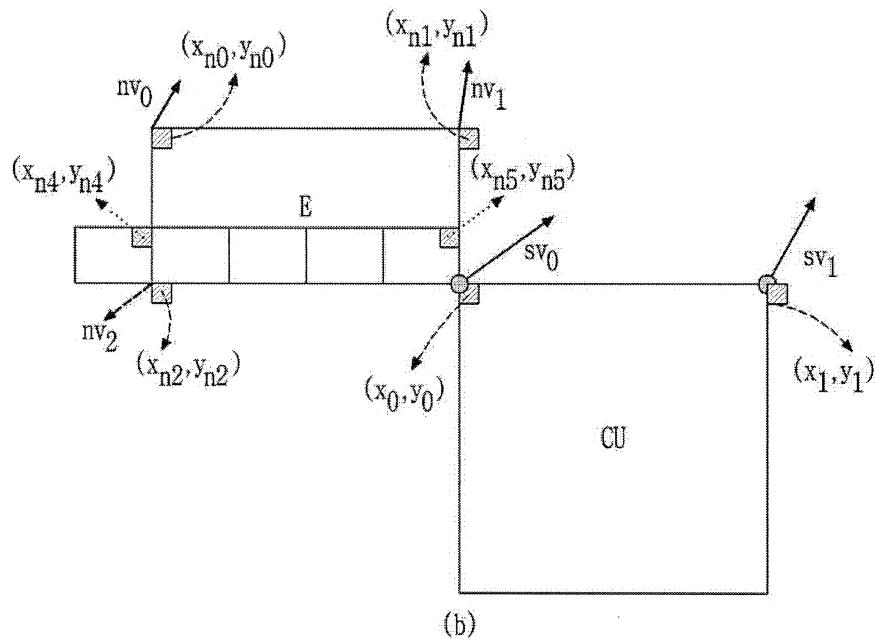
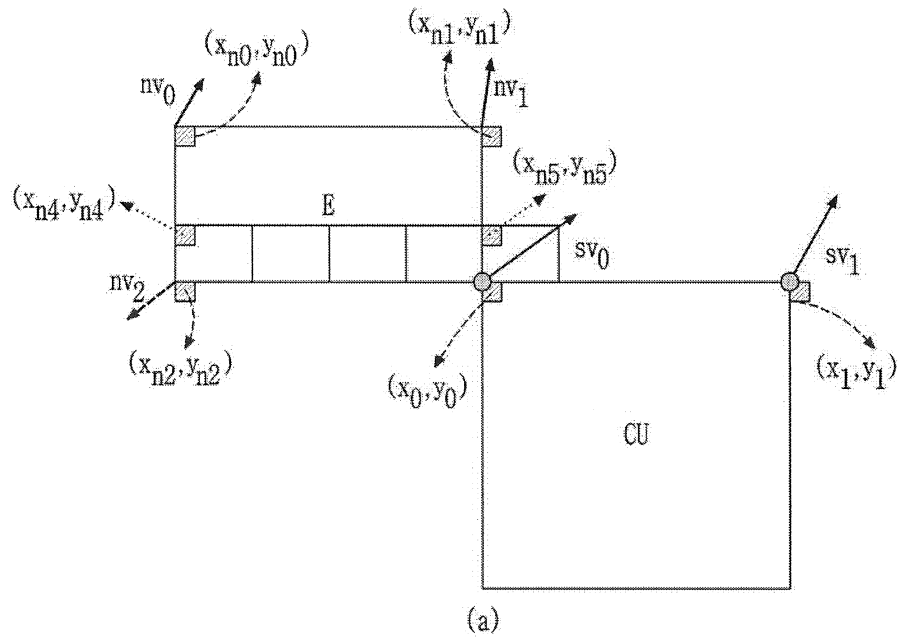


Fig.14

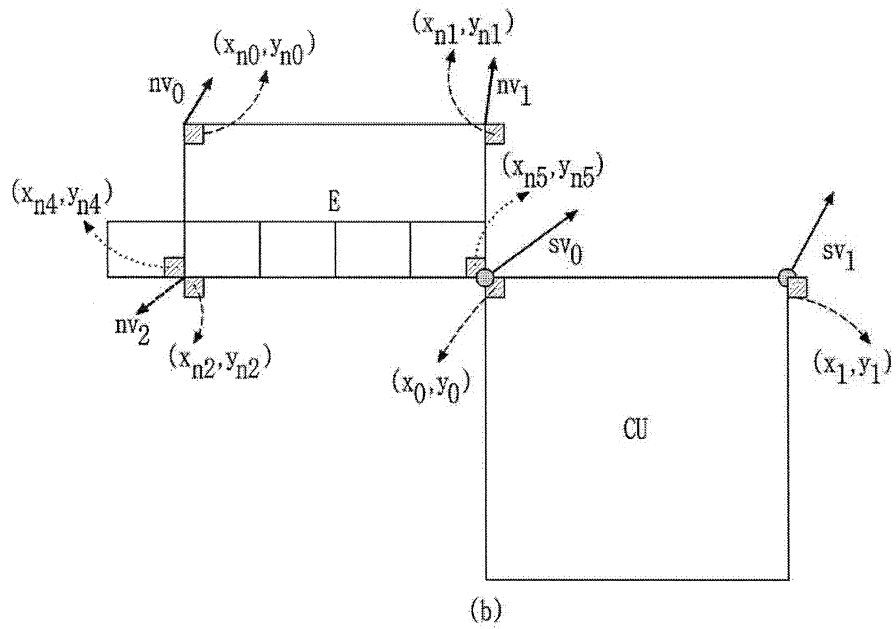
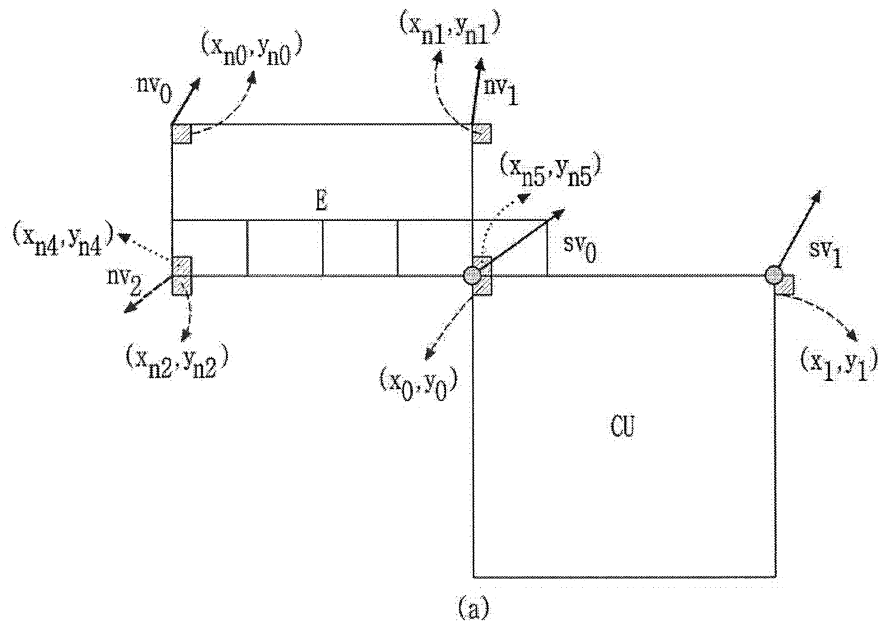


Fig.15

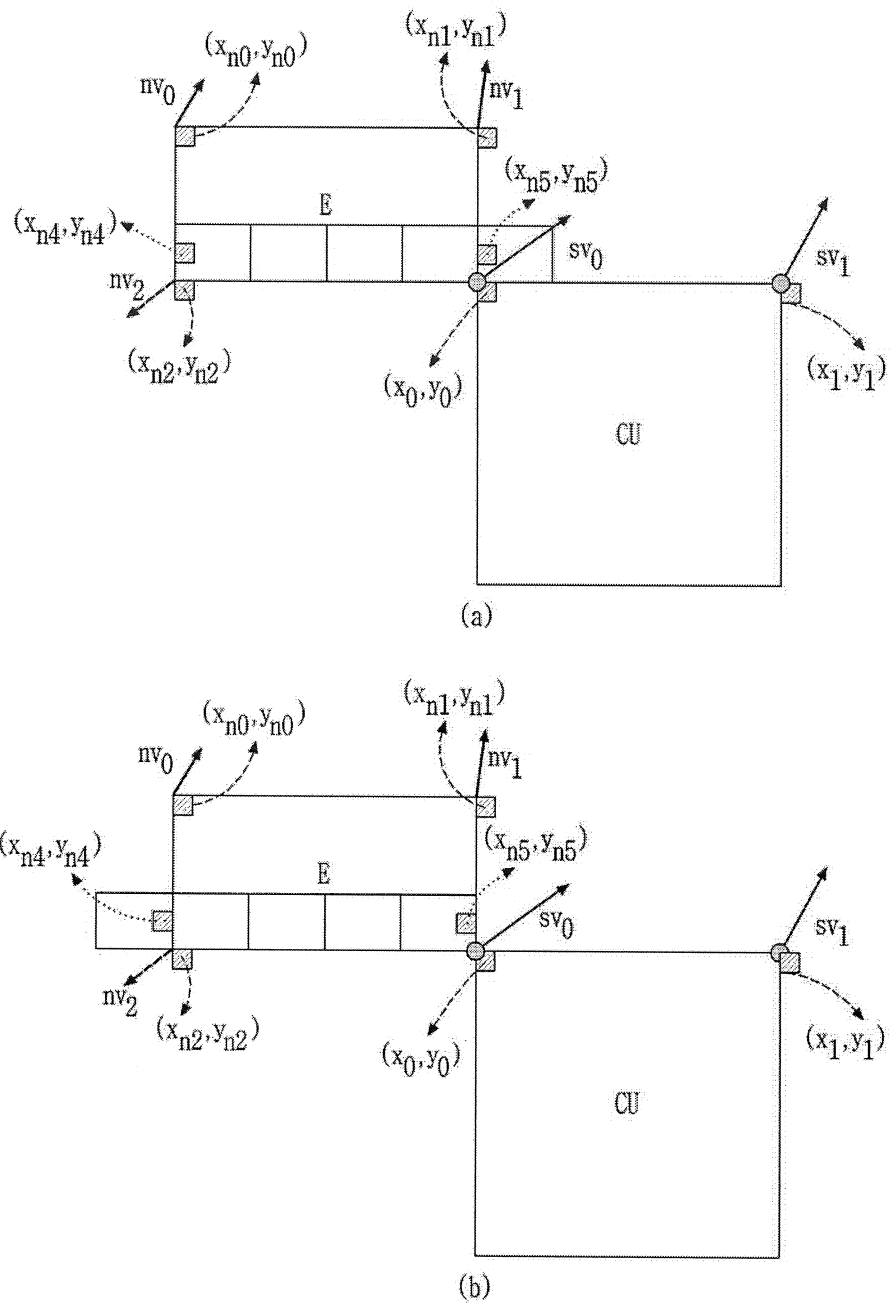


Fig.16

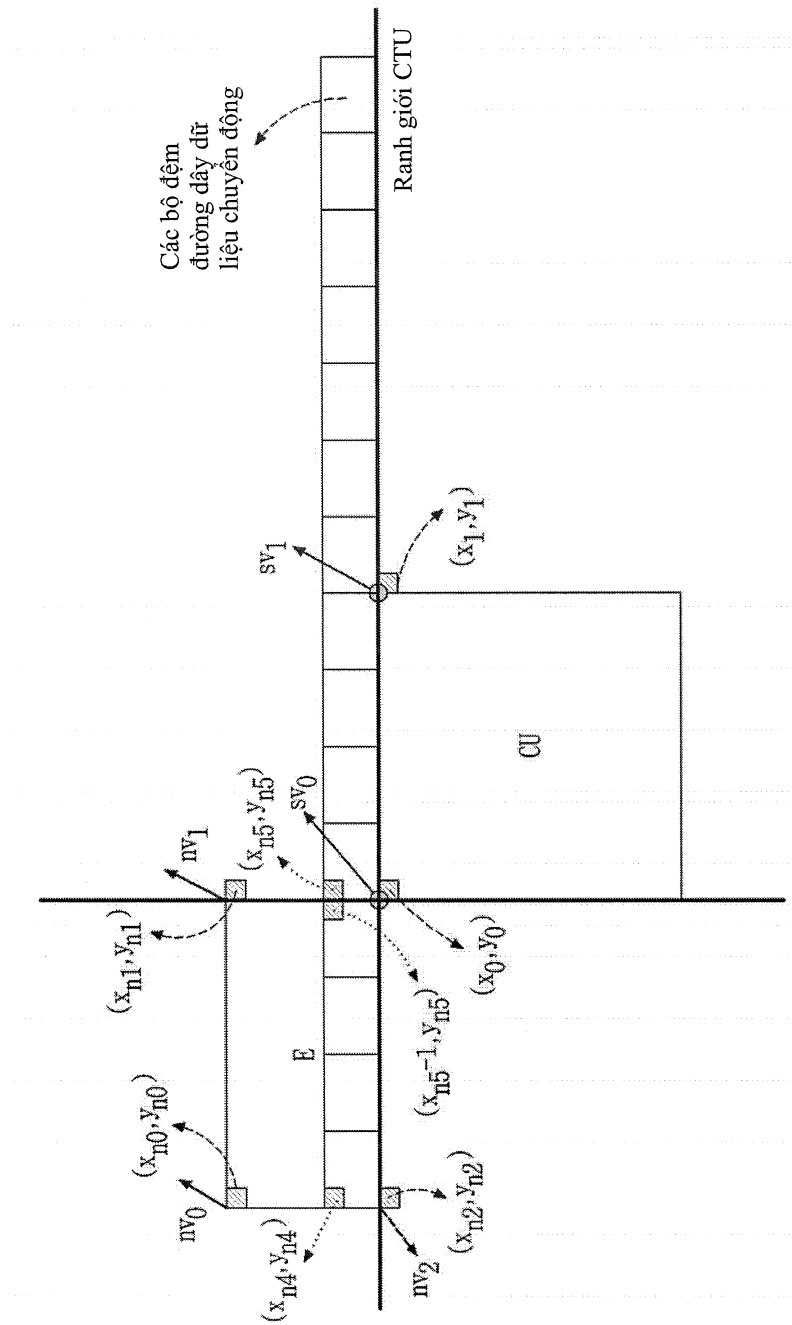


Fig.17

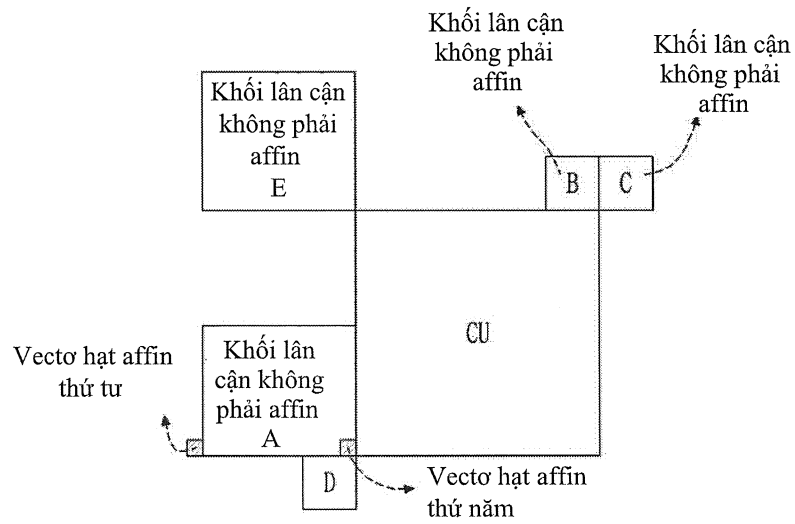


Fig.18

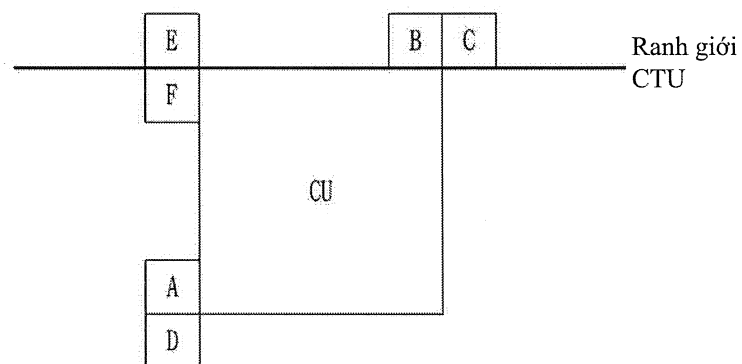


Fig.19

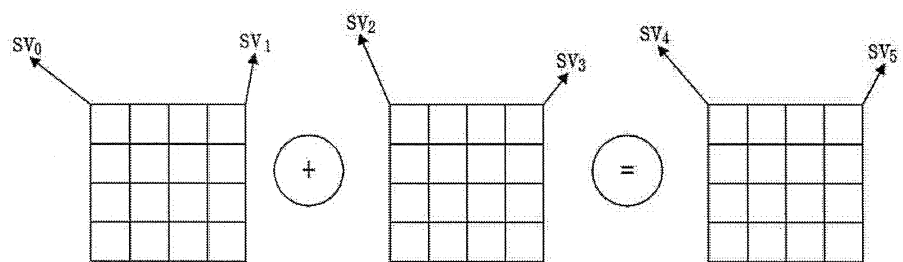


Fig.20

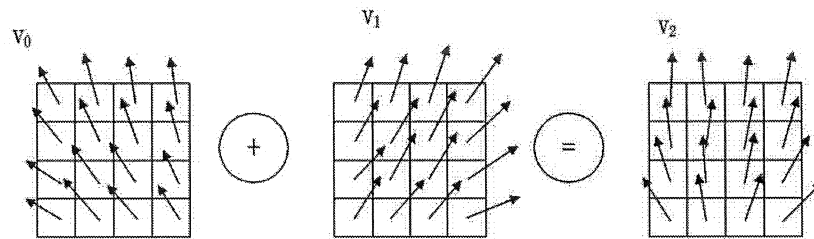


Fig.21

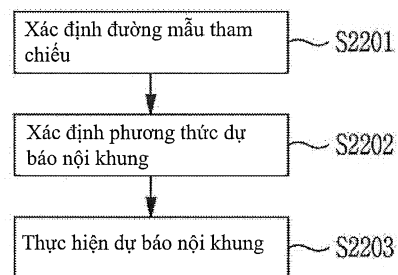


Fig.22

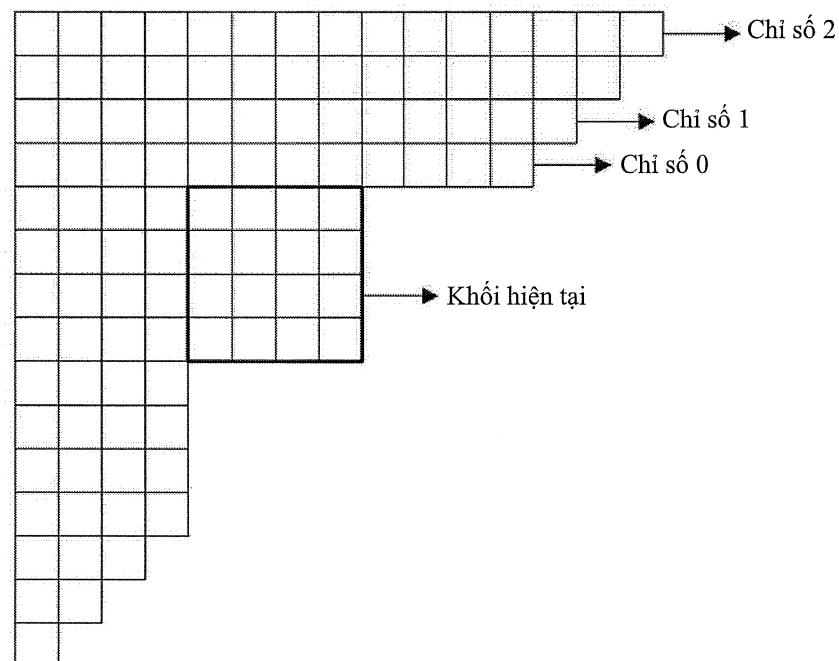


Fig.23

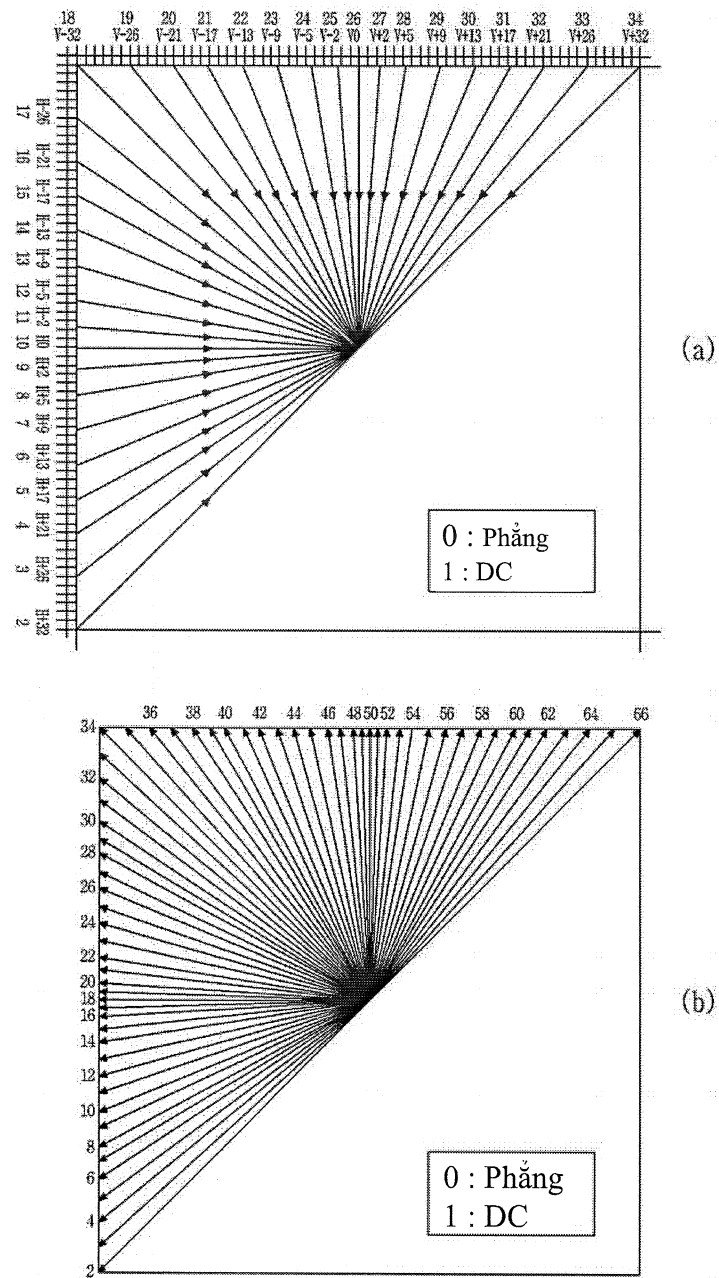


Fig.24

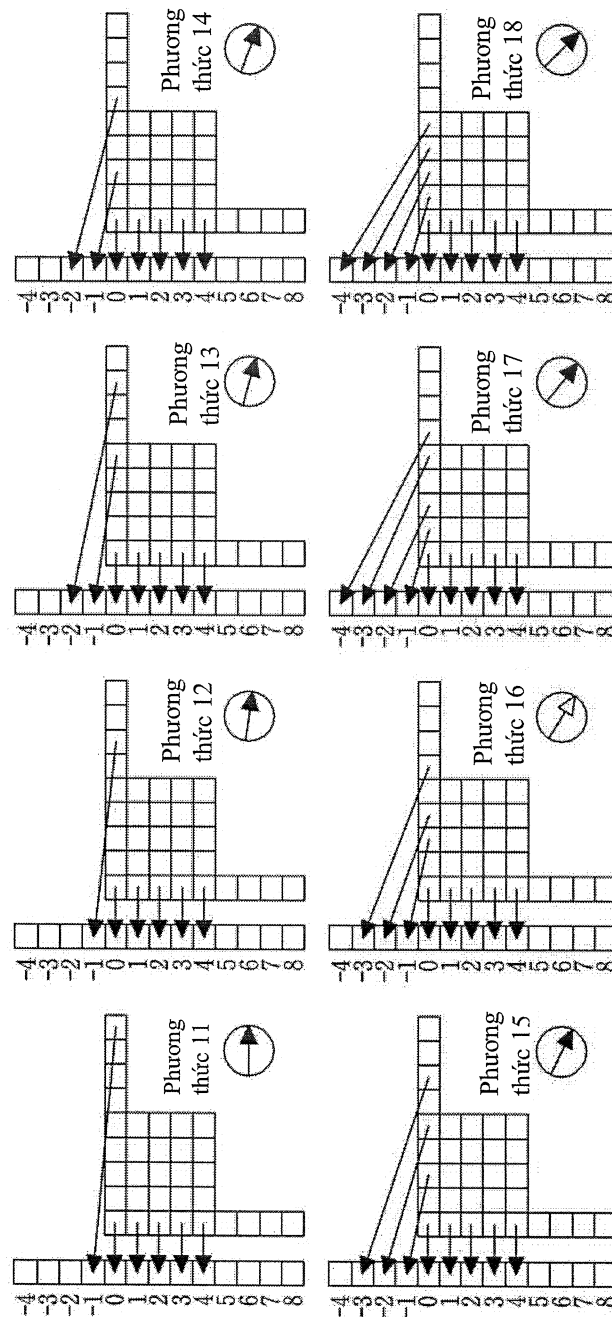


Fig.25

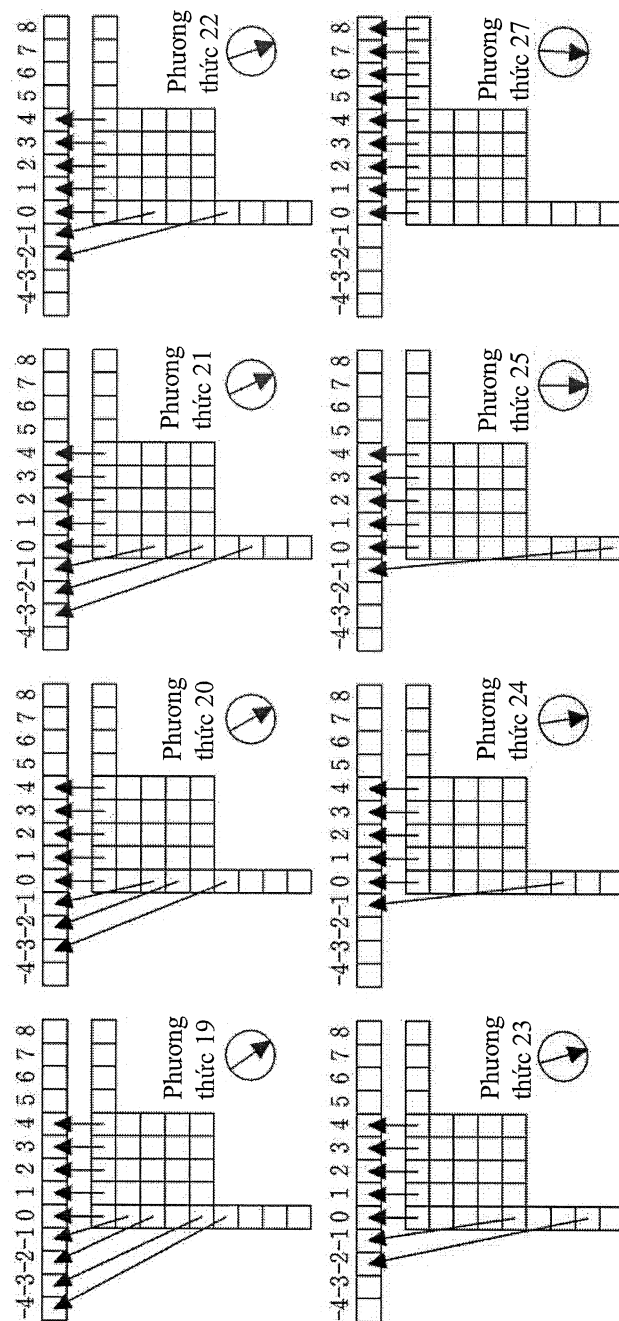


Fig.26

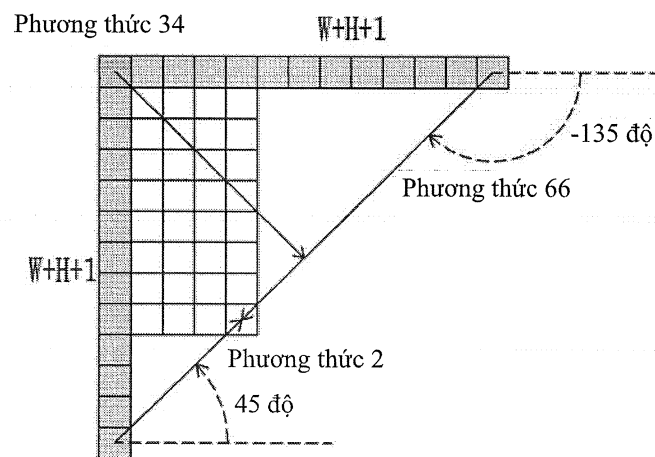


Fig.27

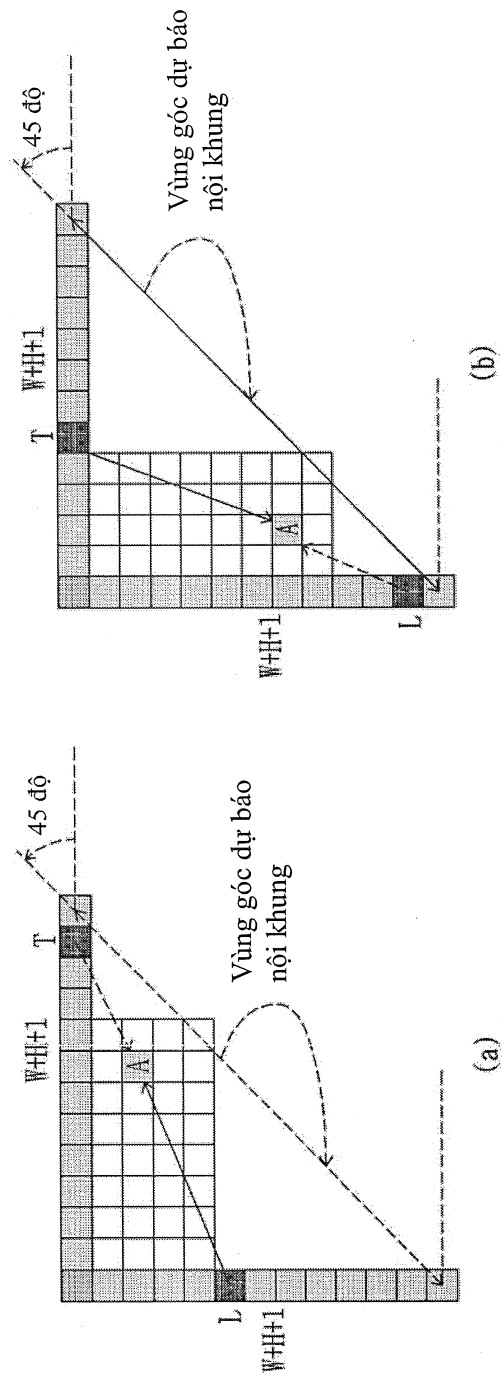


Fig.28

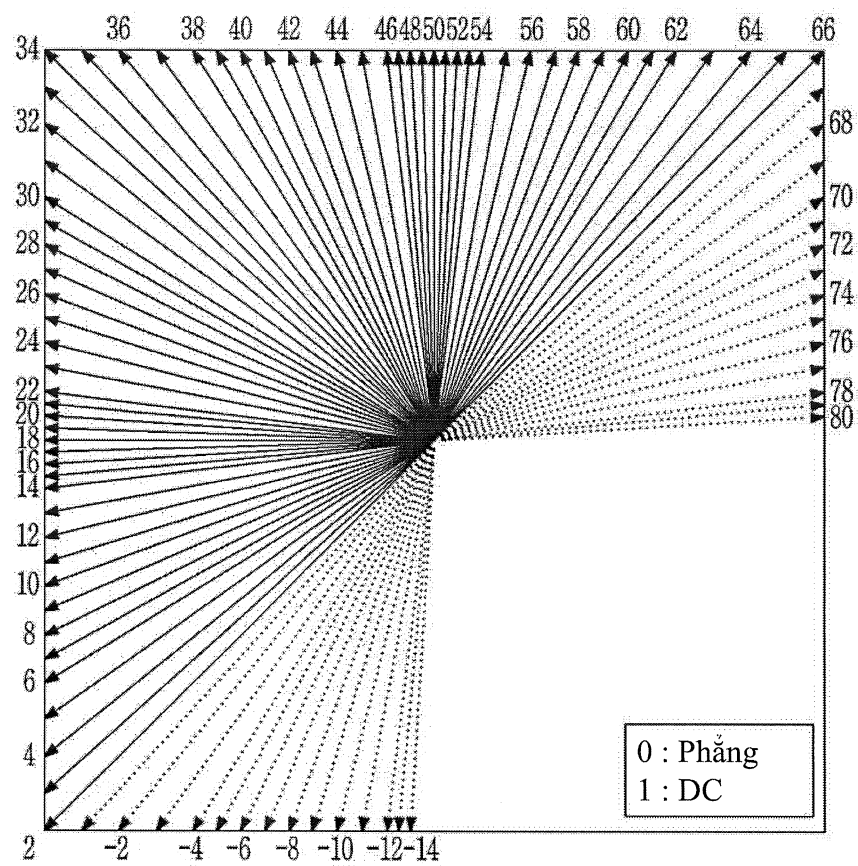


Fig.29

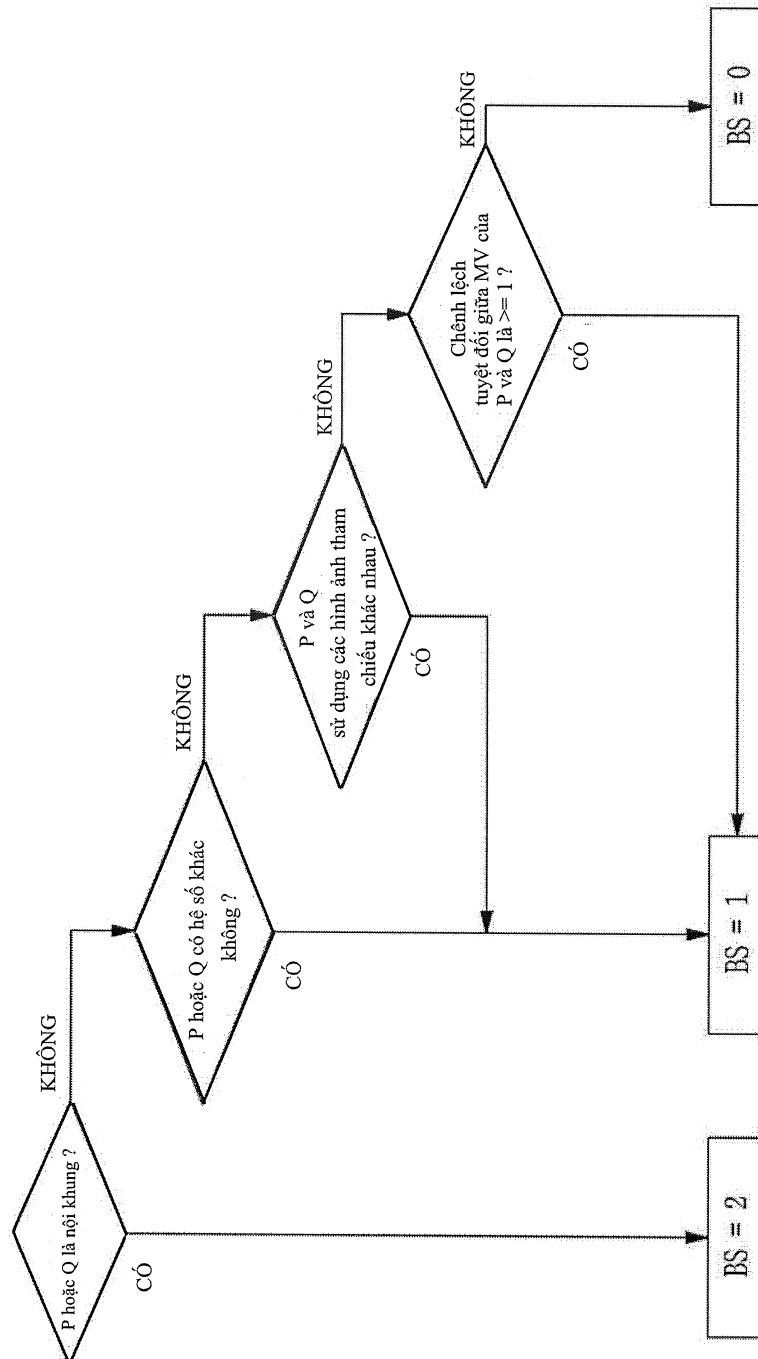


Fig.30

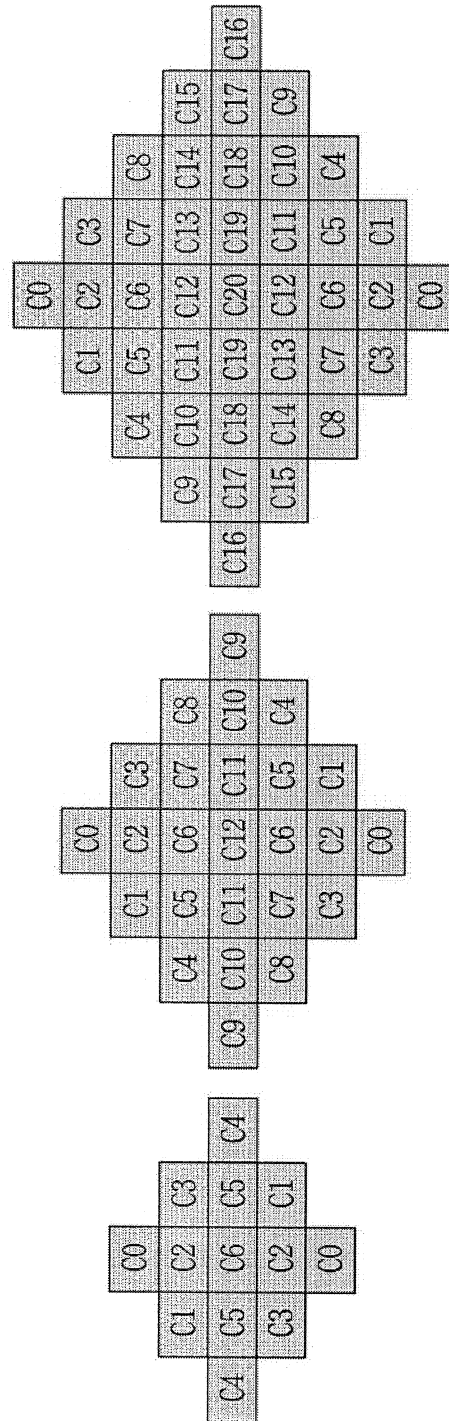


Fig.31

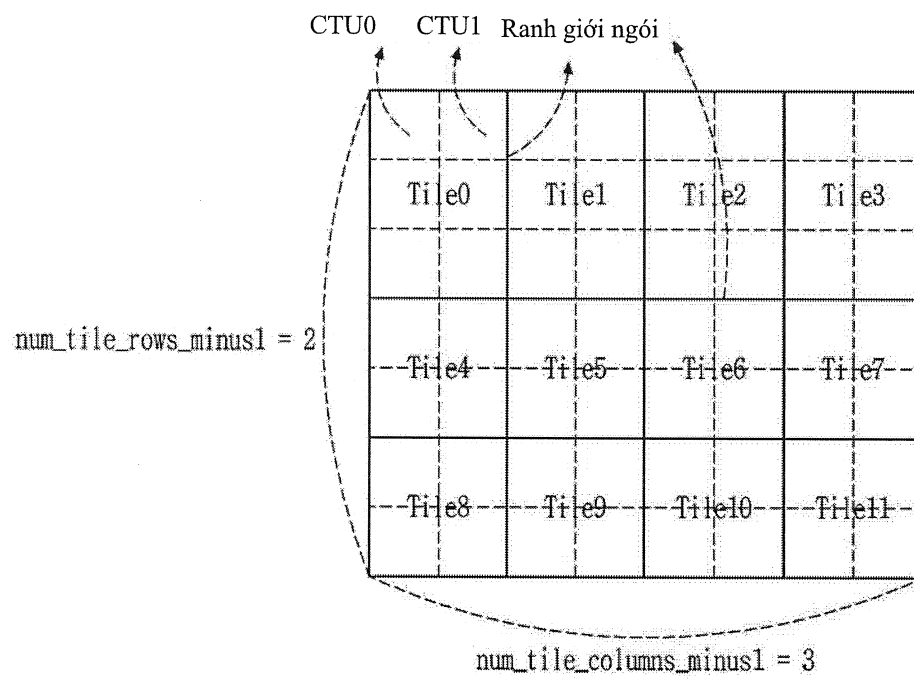


Fig.32

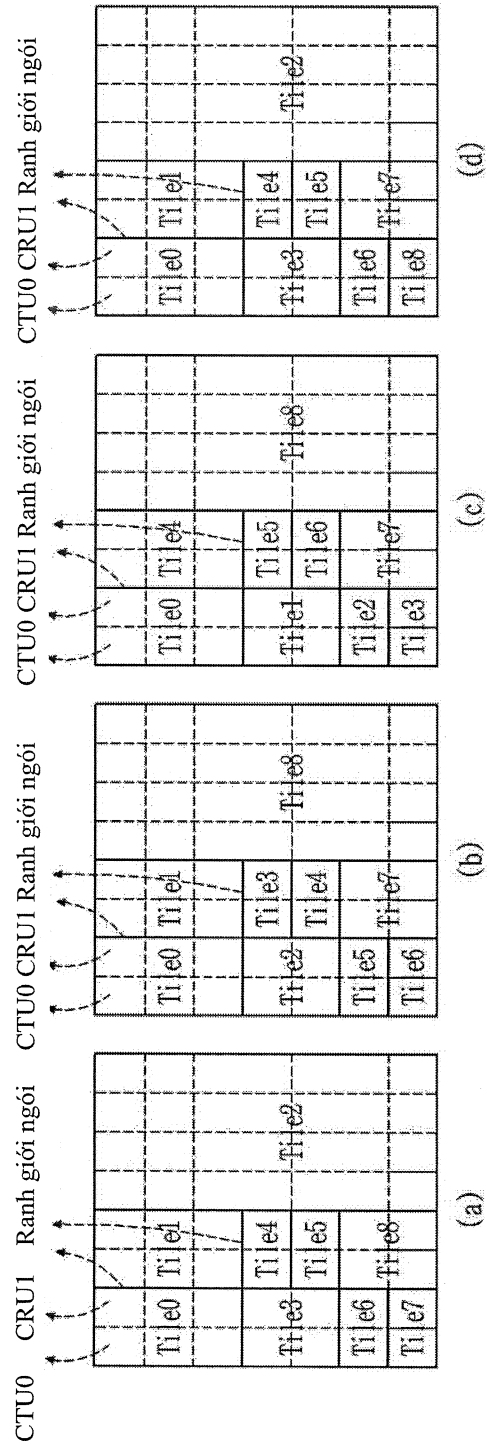


Fig.33

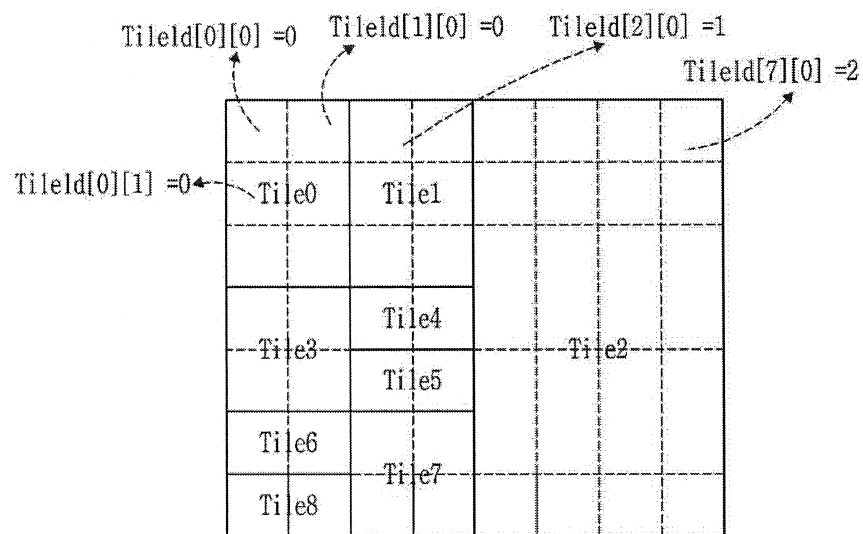


Fig.34

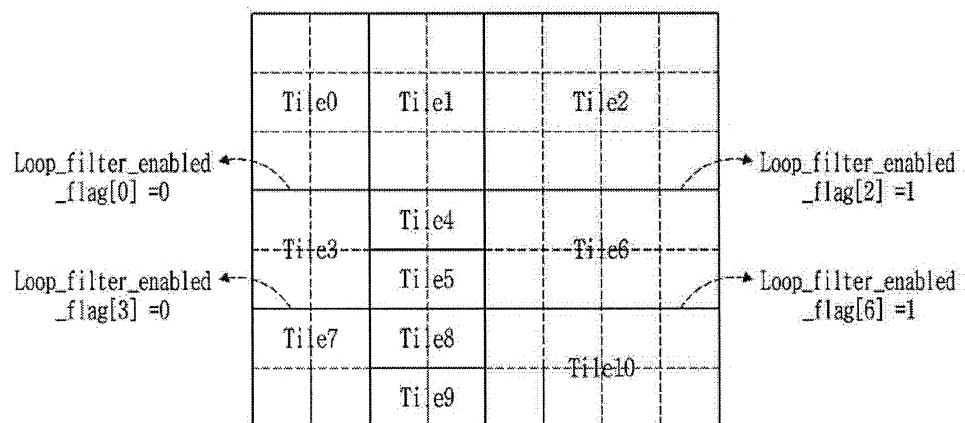


Fig.35