



(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)



1-0048231

(51)^{2020.01} H04N 19/119; H04N 19/174

(13) B

(21) 1-2021-08485

(22) 08/10/2020

(86) PCT/KR2020/013796 08/10/2020

(87) WO 2021/071313 15/04/2021

(30) 10-2019-0124956 09/10/2019 KR; 10-2019-0125745 11/10/2019 KR

(45) 25/07/2025 448

(43) 25/10/2022 415A

(73) Apple Inc. (US)

One Apple Park Way, Cupertino, California 95014, United States of America

(72) LEE, Bae Keun (KR).

(74) Công ty TNHH Lê & Lê (LE & LE)

(54) PHƯƠNG PHÁP GIẢI MÃ VIDEO VÀ PHƯƠNG PHÁP MÃ HÓA VIDEO

(21) 1-2021-08485

(57) Sáng chế đề cập đến phương pháp giải mã video bao gồm phân chia ảnh hiện tại thành các ô, giải mã thông tin phân chia mà biểu diễn loại lát, trong đó thông tin phân chia biểu diễn rằng lát chữ nhật có được áp dụng hay không và giải mã thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi thông tin phân chia biểu diễn rằng lát chữ nhật được áp dụng.

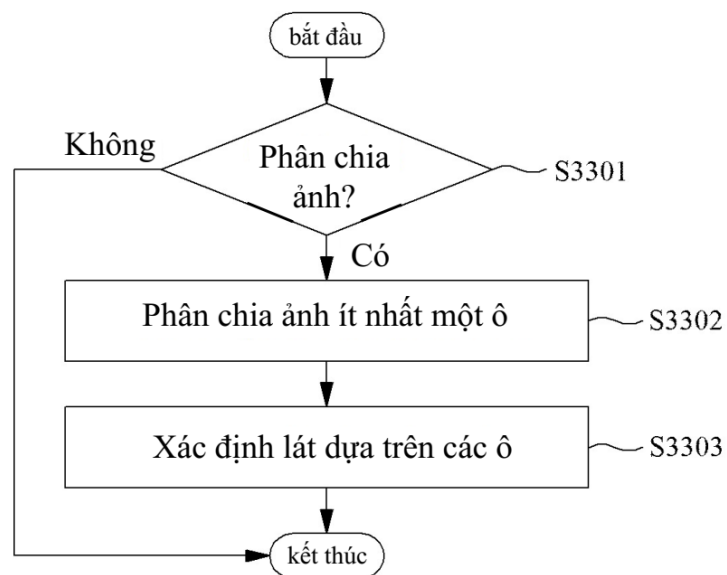


Fig. 33

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp mã hóa/giải mã tín hiệu video và thiết bị mã hóa/giải mã tín hiệu video.

Tình trạng kỹ thuật của sáng chế

Do các panen hiển thị trở nên lớn hơn, dịch vụ video có chất lượng cao hơn được yêu cầu. Vấn đề lớn nhất với dịch vụ video độ phân giải cao là lượng dữ liệu bị tăng lên đáng kể. Để giải quyết vấn đề nêu trên, nghiên cứu để cải thiện tốc độ nén video đang được thực hiện tích cực. Đối với ví dụ đại diện, Đội hợp tác chung về mã hóa Video (JCT-VC) đã được thành lập năm 2009 bởi Nhóm chuyên gia ảnh động (MPEG) và nhóm chuyên gia mã hóa Video (VCEG) theo Ủy ban viễn thông quốc tế-Viễn thông (ITU-T). JCT-VC đề xuất mã hóa video hiệu quả cao (HEVC-High Efficiency Video Coding), tiêu chuẩn nén video mà có hiệu năng nén gấp đôi của H.264/AVC, và đã được chấp nhận như là tiêu chuẩn vào 25/1/2013. Tuy nhiên, với sự phát triển nhanh của các dịch vụ video phân giải cao, hiệu năng của HEVC đang dần thể hiện những hạn chế.

Bản chất kỹ thuật của sáng chế

Mục đích kỹ thuật:

Mục đích của sáng chế là đề xuất phương pháp phân chia ảnh thành nhiều ô hoặc nhiều lát trong việc mã hóa/giải mã tín hiệu video, và thiết bị thực hiện phương pháp này.

Mục đích của sáng chế là đề xuất phương pháp phân chia lát dựa trên chỉ số ô trong việc phân chia ảnh thành các ô, và thiết bị thực hiện phương pháp này.

Mục đích của sáng chế là đề xuất phương pháp phân chia lát dựa trên thông tin chênh lệch với lát trước đó trong việc phân chia ảnh thành các lát, và thiết bị thực hiện phương pháp này.

Các mục đích kỹ thuật có thể thu được từ sáng chế không bị giới hạn ở các

mục đích kỹ thuật nêu trên, và các mục đích kỹ thuật không được đề cập khác có thể được hiểu rõ ràng từ phần mô tả sau đây bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật mà sáng chế liên quan đến.

Giải pháp kỹ thuật:

Phương pháp giải mã tín hiệu video theo sáng chế bao gồm phân chia ảnh hiện tại thành các ô, giải mã thông tin phân chia mà biểu diễn loại lát, trong đó thông tin phân chia biểu diễn rằng lát chữ nhật có được áp dụng hay không và giải mã thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi thông tin phân chia biểu diễn rằng lát chữ nhật được áp dụng. Trong trường hợp này, khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được phân tích hay không có thể được xác định dựa trên độ cao của ô mà lát được bao gồm.

Phương pháp mã hóa tín hiệu video theo sáng chế có thể bao gồm phân chia ảnh hiện tại thành các ô, xác định rằng lát chữ nhật có được áp dụng hay không và mã hóa thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi lát chữ nhật được áp dụng. Trong trường hợp này, khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được mã hóa hay không có thể được xác định dựa trên độ cao của ô mà lát được bao gồm.

Trong phương pháp giải mã tín hiệu video theo sáng chế, việc phân tích thông tin số lượng có thể được bỏ qua khi ô được cấu hình với một hàng đơn vị cây mã hóa.

Trong phương pháp giải mã tín hiệu video theo sáng chế, khi ô được cấu hình với nhiều hàng đơn vị cây mã hóa, thông tin số lượng có thể được phân tích và thông tin số lượng có thể biểu diễn số lượng lát mà thông tin độ cao của nó được báo hiệu rõ ràng.

Trong phương pháp giải mã tín hiệu video theo sáng chế, thông tin độ cao thứ hai có thể được phân tích đối với lát thứ nhất mà chỉ số của nó nhỏ hơn số

lượng được chỉ báo bởi thông tin số lượng trong số ít nhất một lát được chứa trong ô.

Trong phương pháp giải mã tín hiệu video theo sáng chế, thông tin độ cao thứ hai có thể không được phân tích đối với lát thứ hai mà chỉ số của nó là tương tự hoặc lớn hơn số lượng được chỉ báo bởi thông tin số lượng.

Trong phương pháp giải mã tín hiệu video theo sáng chế, độ cao của lát thứ hai có thể được thiết lập là giá trị nhỏ hơn của độ cao của lát thứ ba mà thông tin độ cao thứ hai được phân tích cuối cùng hoặc độ cao còn lại trong ô.

Cần được hiểu rằng các dấu hiệu được tóm tắt nêu trên là các khía cạnh ví dụ của phần mô tả chi tiết sau đây của sáng chế mà không làm giới hạn phạm vi của sáng chế.

Hiệu quả của sáng chế:

Theo sáng chế, hiệu quả mã hóa/giải mã có thể được cải thiện bằng cách phân chia ảnh thành các ô hoặc các lát.

Theo sáng chế, hiệu quả mã hóa/giải mã có thể được cải thiện bằng cách phân chia lát dựa trên chỉ số ô.

Theo sáng chế, hiệu quả mã hóa/giải mã có thể được cải thiện bằng cách phân chia lát dựa trên thông tin chênh lệch với lát trước đó.

Các hiệu quả có thể thu được từ sáng chế có thể không bị giới hạn ở hiệu quả nêu trên, và các hiệu quả không được đề cập khác có thể được hiểu rõ ràng từ phần mô tả sau đây bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật mà sáng chế liên quan đến.

Mô tả vắn tắt các hình vẽ

FIG.1 là hình vẽ thể hiện sơ đồ khối của thiết bị mã hóa video (bộ mã hóa) theo phương án của sáng chế;

FIG.2 là hình vẽ thể hiện sơ đồ khối của thiết bị giải mã video (bộ giải mã)

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 loại phân chia khác nhau của khối mã hóa.

FIG.5 là hình vẽ của ví dụ thể hiện khía cạnh phân chia CTU.

FIG.6 là lưu đồ của phương pháp dự đoán liên đới theo phương án của sáng chế.

FIG.7 là sơ đồ minh họa chuyển động phi tuyến của đối tượng.

FIG.8 là lưu đồ của phương pháp dự đoán liên đới dựa trên chuyển động afin theo phương án của sáng chế.

FIG.9 là sơ đồ minh họa các vectơ hạt afin đối với mỗi mô hình chuyển động afin.

FIG.10 là sơ đồ minh họa các vectơ afin của các khối con trong mô hình chuyển động 4 tham số.

FIG.11 là sơ đồ minh họa các khối ứng viên được sử dụng để thu nhận ứng viên hợp nhất.

FIG.12 là sơ đồ để giải thích khía cạnh cập nhật của Bảng thông tin chuyển động.

FIG.13 là sơ đồ để giải thích khía cạnh cập nhật của Bảng thông tin chuyển động.

FIG.14 là sơ đồ thể hiện khía cạnh cập nhật của Bảng thông tin chuyển động.

FIG.15 là sơ đồ thể hiện ví dụ trong đó chỉ số của ứng viên thông tin chuyển động được lưu giữ được làm mới.

FIG.16 là sơ đồ thể hiện ví dụ trong đó việc kiểm tra dư thừa được thực

hiện chỉ đối với một phần của các ứng viên hợp nhất.

FIG.17 là sơ đồ thể hiện ví dụ trong đó việc kiểm tra dư thừa với ứng viên hợp nhất cụ thể được bỏ qua.

FIG.18 là sơ đồ thể hiện ví dụ trong đó khối ứng viên được chứa trong cùng vùng xử lý hợp nhất như khối hiện tại được thiết lập là không khả dụng như là ứng viên hợp nhất.

FIG.19 là sơ đồ thể hiện ví dụ thu nhận ứng viên hợp nhất đối với khối hiện tại khi khối hiện tại được chứa trong vùng xử lý hợp nhất.

FIG.20 là sơ đồ thể hiện Bảng thông tin chuyển động tạm thời.

FIG.21 là sơ đồ thể hiện ví dụ trong đó Bảng thông tin chuyển động và Bảng thông tin chuyển động tạm thời được hợp nhất.

FIG.22 là lưu đồ của phương pháp nội dự đoán theo phương án của sáng chế.

FIG.23 là sơ đồ thể hiện các chế độ nội dự đoán.

Các Fig.24 và Fig.25 là sơ đồ thể hiện ví dụ của mảng một chiều trong đó các mẫu tham chiếu được sắp xếp trong hàng.

FIG.26 là sơ đồ minh họa góc được tạo thành bởi các chế độ nội dự đoán có hướng với đường thẳng song song với trục x.

FIG.27 là sơ đồ thể hiện khía cạnh trong đó mẫu dự đoán được thu nhận trong trường hợp mà khối hiện tại có dạng không phải hình vuông.

FIG.28 là sơ đồ thể hiện các chế độ nội dự đoán góc rộng.

FIG.29 là sơ đồ thể hiện ví dụ trong đó việc bỏ qua biến đổi có được thực hiện hay không được xác định theo khối con.

FIG.30 là sơ đồ thể hiện ví dụ trong đó các khối con sử dụng cùng loại biến đổi.

FIG.31 là lưu đồ thể hiện xử lý xác định cường độ khối.

FIG.32 thể hiện các ứng viên lọc định trước.

FIG.33 là sơ đồ thể hiện phương pháp phân chia ảnh theo phương án của sáng chế.

FIG.34 thể hiện ví dụ trong đó ảnh được phân chia thành các ô.

FIG.35 là sơ đồ để giải thích khía cạnh tạo ra đơn vị viên.

Các Fig.36 và Fig.37 là các sơ đồ thể hiện ví dụ trong đó lát được xác định dựa trên thứ tự quét mảnh.

FIG.38 là sơ đồ thể hiện ví dụ trong đó chỉ lát trong dạng hình chữ nhật được cho phép.

Mô tả chi tiết sáng chế

Sau đây, các phương án của sáng chế sẽ được mô tả chi tiết có viện dẫn đến các hình vẽ kèm theo.

Việc mã hóa và giải mã ảnh được thực hiện trên cơ sở của khối. Trong ví dụ của sáng chế, đối với khối mã hóa, khối biến đổi, hoặc khối dự đoán, các xử lý mã hóa/giải mã như biến đổi, lượng tử hóa, dự đoán, lọc vòng trong, khôi phục, v.v có thể được thực hiện.

Sau đây, khối đích mã hóa/giải mã được gọi là "khối hiện tại". Trong ví dụ của sáng chế, khối hiện tại có thể biểu diễn khối mã hóa, khối biến đổi, hoặc khối dự đoán theo xử lý mã hóa/giải mã hiện tại.

Ngoài ra, thuật ngữ "đơn vị" được sử dụng trong bản mô tả này biểu diễn đơn vị cơ sở để thực hiện xử lý mã hóa/giải mã cụ thể, và "khối" có thể được hiểu là để biểu diễn mảng mẫu có kích cỡ định trước. Trừ khi được thể hiện khác, "khối" và "đơn vị" có thể được sử dụng dụng hoán đổi. Trong ví dụ của sáng chế, trong các ví dụ được mô tả sau đây, khối mã hóa và đơn vị mã hóa có thể được hiểu là có cùng ý nghĩa như nhau.

FIG.1 là hình vẽ thể hiện sơ đồ khối của thiết bị mã hóa ảnh (bộ mã hóa) theo phương án của sáng chế.

Viện dẫn tới FIG.1, thiết bị mã hóa ảnh 100 có thể bao gồm bộ phân chia ảnh 110, các bộ dự đoán 120 và 125, bộ biến đổi 130, bộ lượng tử hóa 135, bộ sắp xếp lại 160, bộ mã hóa entropy 165, bộ giải lượng tử 140, bộ biến đổi ngược 145, bộ lọc 150, và bộ nhớ 155.

Các bộ phận được mô tả trong FIG.1 được minh họa độc lập để thể hiện các chức năng đặc trưng khác nhau trong thiết bị mã hóa ảnh, và FIG.1 không có nghĩa rằng mỗi bộ phận được cấu thành bởi phần cứng riêng biệt hoặc một bộ phận phần mềm. Tức là, mỗi bộ phận chỉ được đánh số nhằm thuận tiện cho việc giải thích, ít nhất hai thành phần trong số các thành phần tương ứng có thể cấu thành một thành phần hoặc một thành phần có thể được phân chia thành nhiều thành phần mà có thể thực hiện các chức năng của chúng. Ngay cả phương án tích hợp các thành phần tương ứng và phương án phân chia thành phần cũng được nằm trong phạm vi của sáng chế trừ khi chúng nằm ngoài bản chất của sáng chế.

Ngoài ra, một vài thành phần không phải các thành phần cần thiết mà thực hiện các chức năng cơ bản của sáng chế nhưng là các thành phần tùy chọn chỉ để cải thiện hiệu năng. Sáng chế có thể được thực hiện với thành phần cần thiết để thực hiện bản chất của sáng chế ngoài thành phần được sử dụng chỉ để cải thiện hiệu năng và cấu trúc bao gồm chỉ thành phần cần thiết ngoài thành phần tùy chọn được sử dụng chỉ để cải thiện hiệu năng cũng được nằm trong phạm vi sáng chế.

Bộ phân chia ảnh 110 có thể phân chia ảnh đầu vào thành ít nhất một đơn vị xử lý. Theo đó, đơn vị xử lý có thể là đơn vị dự đoán(PU-prediction unit), đơn vị biến đổi(TU-transform unit) hoặc đơn vị mã hóa(CU-đơn vị mã hóa). Trong bộ phân chia ảnh 110, một ảnh có thể được phân chia thành các kết hợp của các đơn vị mã hóa, đơn vị dự đoán, và đơn vị biến đổi, và ảnh có thể được mã hóa bằng cách lựa chọn kết hợp của các đơn vị mã hóa, các đơn vị dự đoán, và các đơn vị biến đổi theo điều kiện định trước (ví dụ, hàm giá trị).

Ví dụ, một ảnh có thể được phân chia thành nhiều đơn vị mã hóa. Để phân chia ảnh thành các đơn vị mã hóa, cấu trúc cây đệ quy như cấu trúc dạng cây tứ phân có thể được sử dụng, và đơn vị mã hóa mà bắt nguồn từ gốc như ảnh đơn hoặc đơn vị mã hóa lớn nhất có thể được phân chia thành các đơn vị mã hóa khác và có thể có các nút con nhiều bằng các đơn vị mã hóa được phân chia. Đơn vị mã hóa mà không còn được phân chia theo giới hạn định trước trở thành nút nhánh. Tức là, khi giả thiết rằng chỉ phân chia hình vuông là khả dụng đối với một đơn

vị mã hóa, một đơn vị mã hóa có thể được phân chia thành tối đa bốn đơn vị mã hóa khác.

Sau đây, trong phương án của sáng chế, đơn vị mã hóa có thể được sử dụng như đơn vị để mã hóa và có thể được sử dụng như là đơn vị để giải mã.

Đơn vị dự đoán có thể được thu nhận bằng cách phân chia một đơn vị mã hóa thành ít nhất một hình vuông hoặc hình chữ nhật có cùng kích cỡ, hoặc một đơn vị mã hóa có thể được phân chia thành các đơn vị dự đoán theo cách mà một đơn vị dự đoán có thể khác với đơn vị dự đoán khác về dạng và/hoặc kích cỡ.

Trong khi tạo ra đơn vị dự đoán dựa trên khối mã hóa mà việc nội dự đoán cần được thực hiện, khi đơn vị mã hóa không phải đơn vị mã hóa nhỏ nhất, việc nội dự đoán có thể được thực hiện mà không cần thực hiện việc phân chia thành các đơn vị dự đoán $N \times N$.

Các bộ dự đoán 120 và 125 có thể bao gồm bộ dự đoán liên đới 120 thực hiện việc dự đoán liên đới và bộ nội dự đoán 125 thực hiện việc nội dự đoán. Việc có thực hiện hay không dự đoán liên đới hoặc nội dự đoán trên đơn vị dự đoán có thể được xác định, và thông tin chi tiết (ví dụ, chế độ nội dự đoán, vector chuyển động, ảnh tham chiếu, v.v.) theo mỗi phương pháp dự đoán có thể được xác định. Theo đó, đơn vị xử lý mà trên đó việc dự đoán được thực hiện có thể khác với đơn vị xử lý cho phương pháp dự đoán, và chi tiết của nó được xác định. Ví dụ, phương pháp dự đoán, chế độ dự đoán, v.v có thể được xác định trên cơ sở của đơn vị dự đoán, và việc dự đoán có thể được thực hiện trên cơ sở của đơn vị biến đổi. Giá trị dư (khối dư) giữa khối dự đoán được tạo ra và khối gốc có thể được đưa vào bộ biến đổi 130. Ngoài ra, thông tin chế độ dự đoán được sử dụng cho việc dự đoán, thông tin vector chuyển động, v.v có thể được mã hóa sử dụng giá trị dư bởi bộ mã hóa entropy 165 và có thể được truyền tới bộ giải mã. Khi chế độ mã hóa cụ thể được sử dụng, khối gốc được mã hóa như thực tại và được truyền tới bộ giải mã mà không tạo ra khối dự đoán thông qua bộ dự đoán 120 hoặc 125.

Bộ dự đoán liên đới 120 có thể dự đoán đơn vị dự đoán trên cơ sở của thông tin về ít nhất một trong số ảnh trước đó và ảnh tiếp theo của ảnh hiện tại, hoặc trong một vài trường hợp, có thể dự đoán đơn vị dự đoán trên cơ sở của thông tin về một vài vùng được mã hóa trong ảnh hiện tại. Bộ dự đoán liên đới 120 có thể bao gồm bộ nội suy ảnh tham chiếu, bộ dự đoán chuyển động, và bộ bù chuyển động.

Bộ nội suy ảnh tham chiếu có thể thu thông tin ảnh tham chiếu từ bộ nhớ 155, và tạo ra thông tin điểm ảnh của điểm ảnh tại số nguyên hoặc nhỏ hơn từ ảnh tham chiếu. Trong trường hợp của điểm ảnh độ chói, bộ lọc nội suy dựa trên DCT 8 nhánh mà có các hệ số khác nhau có thể được sử dụng để tạo ra thông tin điểm ảnh về điểm ảnh tại số nguyên hoặc nhỏ hơn đối với $1/4$ đơn vị điểm ảnh. Trong trường hợp của tín hiệu sắc độ, bộ lọc nội suy dựa trên DCT 4 nhánh mà có các hệ số lọc khác nhau có thể được sử dụng để tạo ra thông tin điểm ảnh về điểm ảnh tại số nguyên hoặc nhỏ hơn đối với $1/8$ đơn vị điểm ảnh.

Bộ dự đoán chuyển động có thể thực hiện việc dự đoán chuyển động dựa trên ảnh tham chiếu được nội suy bởi bộ nội suy ảnh tham chiếu. Đối với các phương pháp tính toán vector chuyển động, các phương pháp khác nhau, như thuật toán so khớp khối dựa trên tìm kiếm toàn phần (FBMA-full search-based block matching algorithm), thuật toán tìm kiếm ba bước (TSS-three step search), thuật toán tìm kiếm ba-bước mới (NTS-new three-step), v.v, có thể được sử dụng. Vector chuyển động có thể có giá trị vector chuyển động trong đơn vị của $1/2$ hoặc $1/4$ điểm ảnh trên cơ sở của điểm ảnh được nội suy. Bộ dự đoán chuyển động có thể dự đoán đơn vị dự đoán hiện tại bằng cách thay đổi phương pháp dự đoán chuyển động. Đối với các phương pháp dự đoán chuyển động, các phương pháp khác nhau, như phương pháp bỏ qua, phương pháp hợp nhất, phương pháp AMVP (Advanced Motion Vector Prediction - Dự đoán vector chuyển động cải tiến), phương pháp sao chép nội khối, v.v, có thể được sử dụng.

Bộ nội dự đoán 125 có thể tạo ra đơn vị dự đoán trên cơ sở của thông tin về điểm ảnh tham chiếu xung quanh khối hiện tại, mà là thông tin điểm ảnh trong ảnh hiện tại. Khi khối lân cận của đơn vị dự đoán hiện tại là khối mà để việc dự đoán liên đới được thực hiện, và do đó điểm ảnh tham chiếu là điểm ảnh mà để việc dự đoán liên đới được thực hiện, điểm ảnh tham chiếu được chứa trong khối mà để việc dự đoán liên đới được thực hiện có thể được thay thế bởi thông tin về điểm ảnh tham chiếu của khối lân cận mà để việc nội dự đoán được thực hiện. Nói cách khác, khi điểm ảnh tham chiếu không khả dụng, ít nhất một điểm ảnh tham chiếu của các điểm ảnh tham chiếu khả dụng có thể được sử dụng thay thế cho thông tin điểm ảnh tham chiếu không khả dụng.

Chế độ dự đoán trong việc nội dự đoán có thể bao gồm chế độ dự đoán có hướng sử dụng thông tin điểm ảnh tham chiếu theo chiều dự đoán và chế độ vô

hướng không sử dụng thông tin có hướng khi thực hiện việc dự đoán. Chế độ để dự đoán thông tin độ chói có thể khác với chế độ dự đoán để thông tin sắc độ. Để dự đoán thông tin sắc độ, thông tin về chế độ nội dự đoán được sử dụng để dự đoán thông tin độ chói hoặc thông tin về tín hiệu độ chói được dự đoán có thể được sử dụng.

Trong khi thực hiện việc nội dự đoán, khi đơn vị dự đoán là đồng nhất về kích cỡ với đơn vị biến đổi, việc nội dự đoán có thể được thực hiện trên đơn vị dự đoán trên cơ sở của các điểm ảnh có vị trí tại bên trái, trên cùng-bên trái, và trên cùng của đơn vị dự đoán. Tuy nhiên, trong khi thực hiện việc nội dự đoán, khi đơn vị dự đoán khác về kích cỡ với đơn vị biến đổi, việc nội dự đoán có thể được thực hiện bằng cách sử dụng điểm ảnh tham chiếu dựa trên đơn vị biến đổi. Ngoài ra, việc nội dự đoán sử dụng phân chia $N \times N$ có thể chỉ được sử dụng đối với đơn vị mã hóa nhỏ nhất.

Trong phương pháp nội dự đoán, khối dự đoán có thể được tạo ra sau khi áp dụng lọc nội san bằng thích nghi (AIS-adaptive intra smoothing) tới điểm ảnh tham chiếu theo chế độ dự đoán. Loại của bộ lọc AIS được áp dụng tới điểm ảnh tham chiếu có thể thay đổi. Để thực hiện phương pháp nội dự đoán, chế độ nội dự đoán đối với đơn vị dự đoán hiện tại có thể được dự đoán từ chế độ nội dự đoán của đơn vị dự đoán có mặt xung quanh đơn vị dự đoán hiện tại. Trong khi dự đoán chế độ dự đoán đối với đơn vị dự đoán hiện tại bằng cách sử dụng thông tin chế độ được dự đoán từ đơn vị dự đoán lân cận, khi chế độ nội dự đoán đối với đơn vị dự đoán hiện tại đồng nhất với chế độ nội dự đoán của đơn vị dự đoán lân cận, thông tin mà chỉ báo rằng đơn vị dự đoán hiện tại và đơn vị dự đoán lân cận có cùng chế độ dự đoán có thể được truyền nhờ sử dụng thông tin cờ định trước. Khi chế độ dự đoán đối với đơn vị dự đoán hiện tại khác với các chế độ dự đoán của các đơn vị dự đoán lân cận, việc mã hóa entropy có thể được thực hiện để mã hóa thông tin về chế độ dự đoán đối với khối hiện tại.

Ngoài ra, khối dư có thể được tạo ra mà bao gồm thông tin về giá trị dư mà là giá trị chênh lệch giữa đơn vị dự đoán mà để việc dự đoán được thực hiện bởi bộ dự đoán 120 hoặc 125, và khối gốc của đơn vị dự đoán. Khối dư được tạo ra có thể được đưa vào bộ biến đổi 130.

Bộ biến đổi 130 có thể thực hiện việc biến đổi trên khối dư, mà bao gồm thông tin về giá trị dư giữa khối gốc và đơn vị dự đoán được tạo ra bởi bộ dự đoán

120 hoặc 125, bằng cách sử dụng phương pháp biến đổi như biến đổi cosin rời rạc (DCT-discrete cosine transform) hoặc biến đổi sin rời rạc (DST-discrete sine transform). Theo đó, lõi biến đổi DCT bao gồm ít nhất một trong số DCT2 hoặc DCT8 và lõi biến đổi DST bao gồm DST7. Việc có áp dụng hay không DCT, hoặc DST để thực hiện việc biến đổi trên khối dư có thể được xác định trên cơ sở của thông tin về chế độ nội dự đoán của đơn vị dự đoán mà được sử dụng để tạo ra khối dư. Có thể bỏ qua việc biến đổi đối với khối dư. Còn mà chỉ báo rằng có bỏ qua biến đổi hay không đối với khối dư có thể được mã hóa. Việc bỏ qua biến đổi có thể được cho phép đối với khối dư mà kích cỡ của nó nhỏ hơn hoặc bằng giá trị ngưỡng, khối dư của thành phần độ chói, hoặc khối dư của thành phần sắc độ dưới khuôn dạng 4:4:4.

Bộ lượng tử hóa 135 có thể thực hiện việc lượng tử hóa trên các giá trị được biến đổi thành miền tần số bởi bộ biến đổi 130. Hệ số lượng tử hóa có thể thay đổi theo khối hoặc độ quan trọng của ảnh. Các giá trị được tính toán trong bộ lượng tử hóa 135 có thể được cấp tới bộ giải lượng tử 140 và bộ sắp xếp lại 160.

Bộ sắp xếp lại 160 có thể thực hiện việc sắp xếp lại trên các giá trị hệ số đối với các giá trị dư được lượng tử hóa.

Bộ sắp xếp lại 160 có thể thay đổi các hệ số dưới dạng của khối hai chiều thành các hệ số dưới dạng vector một chiều thông qua phương pháp quét hệ số. Ví dụ, bộ sắp xếp lại 160 có thể quét từ hệ số DC tới hệ số trong miền tần số cao nhờ sử dụng phương pháp quét zic-zắc để thay đổi các hệ số thành dạng của vector một chiều. Theo kích cỡ và chế độ nội dự đoán của đơn vị dự đoán, ngoài việc quét zic-zắc, việc quét theo chiều dọc trong đó các hệ số trong dạng của khối hai chiều được quét trong chiều cột, hoặc việc quét theo chiều ngang trong đó các hệ số trong dạng của khối hai chiều được quét trong chiều hàng có thể được sử dụng. Nói cách khác, phương pháp quét nào trong số việc quét zic-zắc, việc quét theo chiều dọc, và việc quét theo chiều ngang được sử dụng có thể được xác định theo kích cỡ và chế độ nội dự đoán của đơn vị biến đổi.

Bộ mã hóa entropy 165 có thể thực hiện việc mã hóa entropy trên cơ sở của các giá trị được tính toán bởi bộ sắp xếp lại 160. Việc mã hóa entropy có thể sử dụng các phương pháp mã hóa khác nhau, ví dụ, mã hóa Golomb số mũ, mã hóa độ dài biến thiên thích nghi ngữ cảnh (CAVLC-context-adaptive variable

length coding), hoặc mã hóa số học nhị phân thích nghi ngữ cảnh (CABAC-context-adaptive binary arithmetic coding).

Bộ mã hóa entropy 165 có thể mã hóa các loại thông tin khác nhau, như thông tin về hệ số giá trị dư và thông tin về loại khối của đơn vị mã hóa, thông tin về chế độ dự đoán, thông tin về đơn vị phân chia, thông tin về đơn vị dự đoán, thông tin về đơn vị phân chia, thông tin về đơn vị dự đoán và thông tin về bộ truyền, thông tin về vectơ chuyển động, thông tin về khung tham chiếu, thông tin về nội suy khối, thông tin lọc, v.v thu được từ bộ sắp xếp lại 160 và các bộ dự đoán 120 và 125.

Bộ mã hóa entropy 165 có thể mã hóa entropy các hệ số của đơn vị mã hóa được đưa vào từ bộ sắp xếp lại 160.

Bộ giải lượng tử 140 có thể thực hiện việc giải lượng tử trên các giá trị được lượng tử hóa trong bộ lượng tử hóa 135, và bộ biến đổi ngược 145 có thể thực hiện việc biến đổi ngược trên các giá trị được biến đổi trong bộ biến đổi 130. Giá trị dư được tạo ra bởi bộ giải lượng tử 140 và bộ biến đổi ngược 145 có thể được thêm với đơn vị dự đoán được dự đoán bởi bộ đánh giá chuyển động, bộ bù chuyển động, hoặc bộ nội dự đoán mà được chứa trong các bộ dự đoán 120 và 125 để tạo ra khối được khôi phục.

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

Bộ lọc giải khối có thể loại bỏ méo khối mà xảy ra do các biên giữa các khối trong ảnh được khôi phục. Để xác định việc có thực hiện giải khối hay không, việc có áp dụng hay không bộ lọc giải khối tới khối hiện tại có thể được xác định trên cơ sở của các điểm ảnh được chứa trong một vài hàng và cột được chứa trong khối. Khi bộ lọc giải khối được áp dụng tới khối, bộ lọc mạnh hoặc bộ lọc yếu được áp dụng theo cường độ lọc giải khối được yêu cầu. Ngoài ra, trong khi áp dụng bộ lọc giải khối, khi thực hiện việc lọc theo chiều ngang và việc lọc theo chiều dọc, việc lọc theo chiều ngang và việc lọc theo chiều dọc có thể được cấu hình để được xử lý song song.

Bộ hiệu chỉnh độ dịch có thể hiệu chỉnh ảnh gốc bởi độ dịch trong đơn vị của điểm ảnh so với ảnh mà để việc giải khối được thực hiện. Để thực hiện hiệu chỉnh độ dịch trên ảnh cụ thể, phương pháp áp dụng độ dịch tới vùng mà được

xác định sau khi phân chia các điểm ảnh của ảnh thành số lượng vùng định trước, hoặc phương pháp áp dụng độ dịch theo thông tin biên của mỗi điểm ảnh có thể được sử dụng.

Việc lọc vòng thích nghi (ALF-Adaptive loop filtering) có thể được thực hiện trên cơ sở của giá trị thu được bằng cách so sánh ảnh khôi phục được lọc với ảnh gốc. Các điểm ảnh được chứa trong ảnh có thể được phân chia thành các nhóm định trước, bộ lọc được áp dụng tới mỗi nhóm có thể được xác định, và việc lọc có thể được thực hiện riêng biệt trên mỗi nhóm. Thông tin về việc có áp dụng ALF hay không và có thể được truyền đối với mỗi đơn vị mã hóa (CU) đối với tín hiệu độ chói, và dạng và hệ số lọc của bộ lọc ALF cần được áp dụng có thể thay đổi trên cơ sở của mỗi khối. Ngoài ra, bộ lọc ALF mà có cùng dạng (dạng cố định) có thể được áp dụng bất kể đặc tính của khối mà bộ lọc sẽ được áp dụng tới.

Trong bộ nhớ 155, ảnh hoặc khối được khôi phục được tính toán thông qua bộ lọc 150 có thể được lưu trữ. Ảnh hoặc khối được khôi phục được lưu trữ có thể được cấp tới bộ dự đoán 120 hoặc 125 khi thực hiện việc dự đoán liên đới.

FIG.2 là hình vẽ thể hiện sơ đồ khối của thiết bị giải mã ảnh (bộ giải mã) theo phương án của sáng chế.

Viện dẫn tới FIG.2, thiết bị giải mã ảnh 200 có thể bao gồm bộ giải mã entropy 210, bộ sắp xếp lại 215, bộ giải lượng tử 220, bộ biến đổi ngược 225, các bộ dự đoán 230 và 235, bộ lọc 240 và bộ nhớ 245.

Khi dòng bit ảnh được đưa vào từ bộ mã hóa, dòng bit đầu vào có thể được giải mã theo xử lý ngược của thiết bị mã hóa ảnh.

Bộ giải mã entropy 210 có thể thực hiện việc giải mã entropy theo xử lý ngược của việc mã hóa entropy bởi bộ mã hóa entropy của bộ mã hóa ảnh. Ví dụ, kết hợp với các phương pháp được thực hiện bởi thiết bị mã hóa ảnh, các phương pháp khác nhau, như mã hóa Golomb số mũ, mã hóa độ dài biến thiên thích nghi ngữ cảnh (CAVLC-context-adaptive variable length coding), hoặc mã hóa số học nhị phân thích nghi ngữ cảnh (CABAC-context-adaptive binary arithmetic coding) có thể được áp dụng.

Bộ giải mã entropy 210 có thể giải mã thông tin về việc nội dự đoán và dự đoán liên đới được thực hiện bởi bộ mã hóa.

Bộ sắp xếp lại 215 có thể thực hiện việc sắp xếp lại trên dòng bit được giải mã entropy bởi bộ giải mã entropy 210 trên cơ sở của phương pháp sắp xếp lại được sử dụng trong bộ mã hóa. Các hệ số được biểu diễn dưới dạng của vector một chiều có thể được khôi phục và được sắp xếp lại thành các hệ số trong dạng của khối hai chiều. Bộ sắp xếp lại 215 có thể thực hiện việc sắp xếp lại thông qua phương pháp thu thông tin liên quan đến việc quét hệ số được thực hiện trong bộ mã hóa và quét ngược trên cơ sở của thứ tự quét được thực hiện trong bộ mã hóa.

Bộ giải lượng tử 220 có thể thực hiện việc giải lượng tử trên cơ sở của tham số lượng tử hóa thu được từ bộ mã hóa và các giá trị hệ số của khối được sắp xếp lại.

Bộ biến đổi ngược 225 có thể thực hiện, việc biến đổi ngược, mà là DCT ngược hoặc DST ngược, so với việc biến đổi, mà là DCT hoặc DST, được thực hiện trên kết quả lượng tử hóa bởi bộ biến đổi trong bộ mã hóa ảnh. Theo đó, lõi biến đổi DCT có thể bao gồm ít nhất một trong số DCT2 hoặc DCT8, và lõi biến đổi DST có thể bao gồm DST7. Ngoài ra, khi việc biến đổi được bỏ qua trong bộ mã hóa ảnh, việc biến đổi ngược cũng không được thực hiện trong bộ biến đổi ngược 225. Việc biến đổi ngược có thể được thực hiện trên cơ sở của bộ truyền được xác định bởi bộ mã hóa ảnh. Bộ biến đổi ngược 225 của bộ giải mã ảnh có thể thực hiện chọn lọc phương pháp biến đổi (ví dụ, DCT, hoặc DST) theo các đoạn thông tin, như phương pháp dự đoán, kích cỡ của khối hiện tại, chiều dự đoán, v.v.

Bộ dự đoán 230 hoặc 235 có thể tạo ra khối dự đoán trên cơ sở của thông tin liên quan đến khối dự đoán thu được từ bộ giải mã entropy 210 và thông tin khối hoặc ảnh được giải mã trước đó thu được từ bộ nhớ 245.

Như được mô tả nêu trên, đối với hoạt động của bộ mã hóa ảnh, trong khi thực hiện việc nội dự đoán, khi đơn vị dự đoán đồng nhất về kích cỡ với đơn vị biến đổi, việc nội dự đoán có thể được thực hiện trên đơn vị dự đoán trên cơ sở của các điểm ảnh có vị trí tại bên trái, trên cùng-bên trái, và trên cùng của đơn vị dự đoán. Tuy nhiên, trong khi thực hiện việc nội dự đoán, khi đơn vị dự đoán khác về kích cỡ với đơn vị biến đổi, việc nội dự đoán có thể được thực hiện bằng cách sử dụng điểm ảnh tham chiếu dựa trên đơn vị biến đổi. Ngoài ra, việc nội dự đoán sử dụng phân chia $N \times N$ có thể chỉ được sử dụng đối với đơn vị mã hóa nhỏ nhất.

Các bộ dự đoán 230 và 235 có thể bao gồm môđun xác định PU, bộ dự

đoán liên đới, và bộ nội dự đoán. Bộ xác định PU có thể thu các loại thông tin khác nhau, như thông tin về đơn vị dự đoán, thông tin về chế độ dự đoán của phương pháp nội dự đoán, thông tin về dự đoán chuyển động của phương pháp dự đoán liên đới, v.v mà được đưa vào từ bộ giải mã entropy 210, chia đơn vị dự đoán trong đơn vị mã hóa hiện tại, và xác định rằng việc dự đoán liên đới hay việc nội dự đoán được thực hiện trên đơn vị dự đoán. Nhờ sử dụng thông tin được yêu cầu trong việc dự đoán liên đới của đơn vị dự đoán hiện tại thu được từ bộ mã hóa ảnh, bộ dự đoán liên đới 230 có thể thực hiện việc dự đoán liên đới trên đơn vị dự đoán hiện tại trên cơ sở của thông tin về ít nhất một trong số ảnh trước đó và ảnh tiếp theo của ảnh hiện tại mà bao gồm đơn vị dự đoán hiện tại. Ngoài ra, việc dự đoán liên đới có thể được thực hiện trên cơ sở của thông tin về một vài vùng được khôi phục trước trong ảnh hiện tại mà bao gồm đơn vị dự đoán hiện tại.

Để thực hiện việc dự đoán liên đới, phương pháp nào trong số chế độ bỏ qua, chế độ hợp nhất, chế độ AMVP, hoặc chế độ sao chép nội khối được sử dụng như là phương pháp dự đoán chuyển động đối với đơn vị dự đoán được chứa trong đơn vị mã hóa có thể được xác định trên cơ sở của đơn vị mã hóa.

Bộ nội dự đoán 235 có thể tạo ra khối dự đoán trên cơ sở của thông tin về điểm ảnh trong ảnh hiện tại. Khi đơn vị dự đoán là đơn vị dự đoán mà việc nội dự đoán đã được thực hiện, việc nội dự đoán có thể được thực hiện trên cơ sở của thông tin về chế độ nội dự đoán của đơn vị dự đoán thu được từ bộ mã hóa ảnh. Bộ nội dự đoán 235 có thể bao gồm bộ lọc san bằng trong thích nghi (AIS-adaptive intra smoothing), điểm môđun nội suy ảnh tham chiếu, hoặc bộ lọc DC. Bộ lọc AIS có thể thực hiện việc lọc trên điểm ảnh tham chiếu của khối hiện tại, và việc có áp dụng lọc hay không có thể được xác định theo chế độ dự đoán đối với đơn vị dự đoán hiện tại. Chế độ dự đoán của đơn vị dự đoán và thông tin về bộ lọc AIS mà thu được từ bộ mã hóa ảnh có thể được sử dụng khi thực hiện việc lọc AIS trên điểm ảnh tham chiếu của khối hiện tại. Khi chế độ dự đoán đối với khối hiện tại là chế độ mà việc lọc AIS không được áp dụng tới, bộ lọc AIS có thể không được áp dụng.

Khi chế độ dự đoán của đơn vị dự đoán là chế độ dự đoán mà việc nội dự đoán được thực hiện trên cơ sở của giá trị điểm ảnh thu được bằng cách nội suy các điểm ảnh tham chiếu, bộ nội suy điểm ảnh tham chiếu có thể nội suy các điểm ảnh tham chiếu để tạo ra điểm ảnh tham chiếu mà có đơn vị nguyên hoặc nhỏ hơn.

Khi chế độ dự đoán đối với đơn vị dự đoán hiện tại là chế độ dự đoán trong đó khối dự đoán được tạo ra mà không nội suy các điểm ảnh tham chiếu, các điểm ảnh tham chiếu có thể không được nội suy. Bộ lọc DC có thể tạo ra khối dự đoán thông qua việc lọc khi chế độ dự đoán đối với khối hiện tại là chế độ DC.

Khối hoặc ảnh được khôi phục có thể được cấp tới bộ lọc 240. Bộ lọc 240 có thể bao gồm bộ lọc giải khối, môđun hiệu chỉnh độ dịch, và ALF.

Thông tin về việc bộ lọc giải khối có được áp dụng tới ảnh hoặc khối tương ứng hay không và thông tin về việc bộ lọc mạnh hoặc bộ lọc yếu được áp dụng khi bộ lọc giải khối được áp dụng có thể thu được từ bộ mã hóa ảnh. Bộ lọc giải khối của bộ giải mã ảnh có thể thu thông tin về bộ lọc giải khối từ bộ mã hóa ảnh, và bộ giải mã ảnh có thể thực hiện việc lọc giải khối trên khối tương ứng.

Bộ hiệu chỉnh độ dịch có thể thực hiện việc hiệu chỉnh độ dịch trên ảnh được khôi phục trên cơ sở của loại hiệu chỉnh độ dịch, thông tin về giá trị độ dịch, v.v được áp dụng tới ảnh khi thực hiện việc mã hóa.

ALF có thể được áp dụng tới đơn vị mã hóa trên cơ sở của thông tin về có áp dụng ALF hay không, thông tin về hệ số ALF, v.v thu được từ bộ mã hóa. Thông tin ALF có thể được cung cấp bằng cách được chứa trong tập hợp tham số cụ thể.

Trong bộ nhớ 245, ảnh hoặc khối được khôi phục có thể được lưu trữ để được sử dụng như là ảnh tham chiếu hoặc khối tham chiếu, và ảnh được khôi phục có thể được cấp tới bộ 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 lớn nhất có thể được xác định là khối cây mã hóa. Ảnh đơn có thể được phân chia thành các đơn vị cây mã hóa (CTU-đơn vị cây mã hóa). CTU có thể là đơn vị mã hóa có kích cỡ lớn nhất, và có thể được gọi là đơn vị mã hóa lớn nhất (LCU-largest coding unit). FIG.3 là hình vẽ thể hiện ví dụ trong đó ảnh đơn được phân chia thành các CTU.

Kích cỡ của CTU có thể được xác định trong mức ảnh hoặc mức chuỗi. Tương tự, thông tin mà biểu diễn kích cỡ của CTU có thể được báo hiệu thông qua tập hợp tham số ảnh hoặc tập hợp tham số chuỗi.

Trong ví dụ của sáng chế, kích cỡ của CTU đối với toàn bộ ảnh trong chuỗi có thể được thiết lập là 128x128. Ngoài ra, bất kỳ một trong số kích cỡ 128x128 hoặc 256x256 có thể được xác định là kích cỡ của CTU trong mức ảnh. Trong ví dụ của sáng chế, CTU có thể được thiết lập để có kích cỡ bằng 128x128 trong ảnh thứ nhất, và kích cỡ bằng 256x256 trong ảnh thứ hai.

Các khối mã hóa có thể được tạo ra bằng cách phân chia CTU. Khối mã hóa biểu diễn đơn vị cơ bản để thực hiện việc mã hóa/giải mã. Trong ví dụ của sáng chế, việc dự đoán hoặc biến đổi có thể được thực hiện đối với mỗi khối mã hóa, hoặc chế độ mã hóa dự đoán có thể được xác định đối với mỗi khối mã hóa. Theo đó, chế độ mã hóa dự đoán biểu diễn phương pháp tạo ra ảnh dự đoán. Trong ví dụ của sáng chế, chế độ mã hóa dự đoán có thể bao gồm việc nội dự đoán, việc dự đoán liên đới, tham chiếu ảnh hiện tại (CPR-current picture referencing), sao chép nội khối (IBC-intra block copy) hoặc dự đoán kết hợp. Đối với khối mã hóa, khối dự đoán của khối mã hóa có thể được tạo ra nhờ sử dụng chế độ mã hóa dự đoán của ít nhất một trong số việc nội dự đoán, việc dự đoán liên đới, việc tham chiếu ảnh hiện tại, hoặc dự đoán được kết hợp.

Thông tin mà biểu diễn chế độ mã hóa dự đoán đối với khối hiện tại có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, thông tin này có thể là cờ 1-bit mà biểu diễn rằng chế độ mã hóa dự đoán là chế độ nội dự đoán hay dự đoán liên đới. Khi chế độ mã hóa dự đoán đối với khối hiện tại được xác định là chế độ liên đới, việc tham chiếu ảnh hiện tại hoặc dự đoán kết hợp có thể khả dụng.

Việc tham chiếu ảnh hiện tại là thiết lập ảnh hiện tại như là ảnh tham chiếu và thu nhận khối dự đoán của khối hiện tại từ vùng mà đã được mã hóa/giải mã trong ảnh hiện tại. Theo đó, ảnh hiện tại có nghĩa là ảnh mà bao gồm khối hiện tại. Thông tin mà biểu diễn rằng việc tham chiếu ảnh hiện tại có được áp dụng tới khối hiện tại hay không có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, thông tin này có thể là cờ 1-bit. Khi cờ là đúng (TRUE), chế độ mã hóa dự đoán đối với khối hiện tại có thể được xác định như là việc tham chiếu ảnh hiện tại, và khi cờ là sai (FALSE), chế độ mã hóa dự đoán đối với khối hiện tại có thể được xác định như là việc dự đoán liên đới.

Ngoài ra, chế độ mã hóa dự đoán đối với khối hiện tại có thể được xác định trên cơ sở của chỉ số ảnh tham chiếu. Trong ví dụ của sáng chế, khi chỉ số ảnh tham chiếu chỉ báo ảnh hiện tại, chế độ mã hóa dự đoán đối với khối hiện tại có

thể được xác định như là việc tham chiếu ảnh hiện tại. Khi chỉ số ảnh tham chiếu chỉ báo ảnh khác ngoài ảnh hiện tại, chế độ mã hóa dự đoán đối với khối hiện tại có thể được xác định như là việc dự đoán liên đới. Nói cách khác, việc tham chiếu ảnh hiện tại là phương pháp dự đoán sử dụng thông tin về vùng mà đã được mã hóa/giải mã trong ảnh hiện tại, và việc dự đoán liên đới là phương pháp dự đoán sử dụng thông tin về ảnh khác mà đã được mã hóa/giải mã.

Việc dự đoán kết hợp biểu diễn chế độ mã hóa kết hợp mà kết hợp ít nhất hai trong số việc nội dự đoán, việc dự đoán liên đới, và việc tham chiếu ảnh hiện tại. Trong ví dụ của sáng chế, khi việc dự đoán kết hợp được áp dụng, khối dự đoán thứ nhất có thể được tạo ra trên cơ sở của bất kỳ một trong số việc nội dự đoán, việc dự đoán liên đới hoặc việc tham chiếu ảnh hiện tại, và khối dự đoán thứ hai có thể được tạo ra trên cơ sở của việc nội dự đoán, việc dự đoán liên đới hoặc việc tham chiếu ảnh hiện tại khác. Khi khối dự đoán thứ nhất và khối dự đoán thứ hai được tạo ra, khối dự đoán cuối cùng có thể được tạo ra bằng cách tính toán giá trị trung bình hoặc tổng có trọng số của khối dự đoán thứ nhất và khối dự đoán thứ hai. Thông tin mà biểu diễn rằng có áp dụng hay không việc dự đoán kết hợp tới khối hiện tại có thể được báo hiệu trong dòng bit. Thông tin này có thể là cờ 1-bit.

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

Khối mã hóa có thể được phân chia thành các khối mã hóa trên cơ sở của việc phân chia dạng cây tứ phân, việc phân chia dạng cây nhị phân hoặc việc phân chia dạng cây tam phân. Khối mã hóa được phân chia có thể được phân chia lại thành các khối mã hóa trên cơ sở của việc phân chia dạng cây tứ phân, việc phân chia dạng cây nhị phân hoặc việc phân chia dạng cây tam phân.

việc phân chia dạng cây tứ phân biểu diễn phương pháp phân chia khối hiện tại thành bốn khối. Theo kết quả của việc phân chia dạng cây tứ phân, khối hiện tại có thể được phân chia thành bốn phân đoạn hình vuông (viện dẫn tới "SPLIT_QT" trên FIG.4 (a)).

Việc phân chia dạng cây nhị phân biểu diễn phương pháp phân chia khối hiện tại thành bốn khối. Việc phân chia khối hiện tại thành hai khối theo chiều dọc (tức là, sử dụng đường dọc cắt qua khối hiện tại) có thể được gọi là việc phân chia dạng cây nhị phân theo chiều dọc, và việc phân chia khối hiện tại thành hai khối theo chiều ngang (tức là, sử dụng đường ngang cắt qua khối hiện tại) có thể được

gọi là việc phân chia dạng cây nhị phân theo chiều ngang. Theo kết quả của việc phân chia dạng cây nhị phân, khối hiện tại có thể được phân chia thành hai phân đoạn không phải hình vuông. "SPLIT_BT_VER" của FIG.4 (b) là hình vẽ thể hiện kết quả của việc phân chia dạng cây nhị phân có chiều dọc, và "SPLIT_BT_HOR" của FIG.4 (c) là hình vẽ thể hiện kết quả của việc phân chia dạng cây nhị phân theo chiều ngang.

Việc phân chia dạng cây tam phân biểu diễn phương pháp phân chia khối hiện tại thành ba khối. Việc phân chia khối hiện tại thành ba khối dọc theo chiều dọc (tức là, sử dụng hai đường dọc cắt qua khối hiện tại) có thể được gọi là việc phân chia dạng cây tam phân theo chiều dọc, và việc phân chia khối hiện tại thành ba khối theo chiều ngang (tức là, sử dụng hai đường ngang cắt qua khối hiện tại) có thể được gọi là việc phân chia dạng cây tam phân theo chiều ngang. Theo kết quả của việc phân chia dạng cây tam phân, khối hiện tại có thể được phân chia thành ba phân đoạn không phải hình vuông. Theo đó, độ rộng/độ cao của phân đoạn có vị trí tại trung tâm của khối hiện tại có thể cao gấp đôi độ rộng/độ cao của các phân đoạn khác. "SPLIT_TT_VER" của FIG.4 (d) là hình vẽ thể hiện kết quả của việc phân chia dạng cây tam phân theo chiều dọc, và "SPLIT_TT_HOR" của FIG.4 (e) là hình vẽ thể hiện kết quả của việc phân chia dạng cây tam phân theo chiều ngang.

Số lần phân chia của CTU có thể được xác định là độ sâu phân chia. Độ sâu phân chia lớn nhất của CTU có thể được xác định trong mức chuỗi hoặc ảnh. Do đó, độ sâu phân chia lớn nhất của CTU có thể thay đổi trên cơ sở của chuỗi hoặc ảnh.

Ngoài ra, độ sâu phân chia lớn nhất có thể được xác định độc lập đối với mỗi phương pháp phân chia. Trong ví dụ của sáng chế, độ sâu phân chia lớn nhất trong đó việc phân chia dạng cây tứ phân được cho phép có thể khác với độ sâu phân chia lớn nhất trong đó việc phân chia dạng cây nhị phân và/hoặc việc phân chia dạng cây tam phân được cho phép.

Bộ mã hóa có thể báo hiệu thông tin mà biểu diễn ít nhất một trong số loại phân chia và độ sâu phân chia của khối hiện tại trong dòng bit. Bộ giải mã có thể xác định loại phân chia và độ sâu phân chia của CTU trên cơ sở của thông tin thu được bằng cách phân tích dòng bit.

FIG.5 là hình vẽ của ví dụ thể hiện khía cạnh phân chia CTU.

Việc phân chia khối mã hóa nhờ sử dụng việc phân chia dạng cây tứ phân, việc phân chia dạng cây nhị phân và/hoặc việc phân chia dạng cây tam phân có thể được gọi là việc phân chia dạng cây đa mức.

Các khối mã hóa được tạo ra bằng cách phân chia khối mã hóa bằng cách áp dụng việc phân chia dạng cây đa mức có thể được gọi là các khối mã hóa con. Khi độ sâu phân chia của khối mã hóa là k , độ sâu phân chia của các khối mã hóa con được thiết lập là $k + 1$.

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

Loại phân chia của khối mã hóa hiện tại có thể được xác định trên cơ sở của ít nhất một trong số loại phân chia của khối mã hóa gốc và loại phân chia của khối mã hóa lân cận. Theo đó, khối mã hóa lân cận có thể là khối liền kề với khối mã hóa hiện tại, và bao gồm ít nhất một trong số khối lân cận trên cùng, khối lân cận bên trái, hoặc khối lân cận liền kề với góc trên cùng-bên trái của khối mã hóa hiện tại. Theo đó, loại phân chia có thể bao gồm việc có áp dụng hay không việc phân chia dạng cây tứ phân, có áp dụng hay không việc phân chia dạng cây nhị phân, chiều của việc phân chia dạng cây nhị phân, có áp dụng hay không việc phân chia dạng cây tam phân, hoặc chiều của việc phân chia dạng cây tam phân.

Để xác định loại phân chia của khối mã hóa, thông tin mà biểu diễn rằng khối mã hóa có được phân chia hay không có thể được báo hiệu trong dòng bit. Thông tin này là cờ 1-bit của "split_cu_flag", và khi cờ là đúng (TRUE), có thể biểu diễn rằng khối mã hóa được phân chia bởi phương pháp phân chia dạng cây đa mức.

Khi split_cu_flag là đúng (TRUE), thông tin mà biểu diễn rằng khối mã hóa có được phân chia hay không bởi việc phân chia dạng cây tứ phân có thể được báo hiệu trong dòng bit. Thông tin là cờ 1-bit của split_qt_flag, và khi cờ là đúng (TRUE), khối mã hóa có thể được phân chia thành bốn khối.

Trong ví dụ của sáng chế, trong ví dụ được thể hiện trên FIG.5, CTU được phân chia bởi việc phân chia dạng cây tứ phân, và do đó bốn khối mã hóa mà có độ sâu phân chia bằng 1 được tạo ra. Ngoài ra, được thể hiện rằng việc phân chia dạng cây tứ phân được áp dụng lần nữa tới khối mã hóa thứ nhất và khối mã hóa thứ tư trong số bốn khối mã hóa được tạo ra bởi việc phân chia dạng cây tứ phân.

Kết quả là, bốn khối mã hóa mà có độ sâu phân chia bằng 2 có thể được tạo ra.

Ngoài ra, bằng cách áp dụng lần nữa việc phân chia dạng cây tứ phân tới khối mã hóa mà có độ sâu phân chia bằng 2, khối mã hóa mà có độ sâu phân chia bằng 3 có thể được tạo ra.

Khi việc phân chia dạng cây tứ phân không được áp dụng tới khối mã hóa, việc có thực hiện hay không việc phân chia dạng cây nhị phân hoặc việc phân chia dạng cây tam phân đối với khối mã hóa có thể được xác định theo ít nhất một trong số kích cỡ của khối mã hóa, việc khối mã hóa có vị trí tại biên ảnh hay không, độ sâu phân chia lớn nhất, hoặc loại phân chia của khối lân cận. Khi được xác định để thực hiện việc phân chia dạng cây nhị phân hoặc việc phân chia dạng cây tam phân đối với khối mã hóa, thông tin mà biểu diễn chiều phân chia có thể được báo hiệu trong dòng bit. Thông tin này có thể là cờ 1-bit của `mtt_split_cu_vertical_flag`. Việc chiều phân chia là chiều dọc hay chiều ngang có thể được xác định trên cơ sở của cờ. Ngoài ra, thông tin mà biểu diễn việc phân chia nào trong số việc phân chia dạng cây nhị phân hoặc việc phân chia dạng cây tam phân được áp dụng tới khối mã hóa có thể được báo hiệu trong dòng bit. Thông tin này có thể là cờ 1-bit của `mtt_split_cu_binary_flag`. Việc phân chia dạng cây nhị phân được áp dụng tới khối mã hóa hay việc phân chia dạng cây tam phân được áp dụng tới khối mã hóa có thể được xác định trên cơ sở của cờ.

Trong ví dụ của sáng chế, trong ví dụ được thể hiện trên FIG.5, việc phân chia dạng cây nhị phân có chiều dọc được áp dụng tới khối mã hóa mà có độ sâu phân chia bằng 1, việc phân chia dạng cây tam phân theo chiều dọc được áp dụng tới khối mã hóa bên trái trong số các khối mã hóa được tạo ra bằng cách phân chia, và việc phân chia dạng cây nhị phân có chiều dọc được áp dụng tới khối mã hóa bên phải.

Việc dự đoán liên đới là chế độ mã hóa dự đoán mà dự đoán khối hiện tại nhờ sử dụng thông tin về ảnh trước đó. Trong ví dụ của sáng chế, khối (sau đây gọi là khối được sắp xếp theo thứ tự) tại cùng vị trí với khối hiện tại trong ảnh trước đó có thể được thiết lập là khối dự đoán của khối hiện tại. Sau đây, khối dự đoán được tạo ra trên cơ sở của khối được sắp xếp theo thứ tự của khối hiện tại có thể được gọi là khối dự đoán được sắp xếp theo thứ tự.

Ngược lại, khi đối tượng hiện diện trong ảnh trước đó đã di chuyển tới vị trí khác trong ảnh hiện tại, khối hiện tại có thể được dự đoán hiệu quả nhờ sử dụng

các chuyển động của đối tượng. Ví dụ, khi chiều chuyển động và kích cỡ của đối tượng được xác định bằng cách so sánh ảnh trước đó với ảnh hiện tại, khối dự đoán (hoặc ảnh dự đoán) của khối hiện tại có thể được tạo ra theo thông tin chuyển động của các đối tượng. Sau đây, khối dự đoán được tạo ra nhờ sử dụng thông tin chuyển động có thể được gọi là khối dự đoán chuyển động.

Khối dự có thể được tạo ra bằng cách trừ khối dự đoán từ khối hiện tại. Theo đó, trong trường hợp trong đó đối tượng di chuyển, năng lượng của khối dự có thể được làm giảm nhờ sử dụng khối dự đoán chuyển động ngoài sử dụng khối dự đoán được sắp xếp theo thứ tự, và do đó hiệu năng nén của khối dự có thể được cải thiện.

Như nêu trên, việc tạo ra khối dự đoán bằng cách sử dụng thông tin chuyển động có thể được gọi là dự đoán đánh giá chuyển động. Trong hầu hết việc dự đoán liên đới, khối dự đoán có thể được tạo ra trên cơ sở của dự đoán bù chuyển động.

Thông tin chuyển động có thể bao gồm ít nhất một trong số vectơ chuyển động, chỉ số ảnh tham chiếu, chiều dự đoán, và chỉ số hệ số trọng số hai chiều. Vectơ chuyển động biểu diễn chiều chuyển động của đối tượng và độ lớn. Chỉ số ảnh tham chiếu chỉ rõ ảnh tham chiếu của khối hiện tại trong số các ảnh tham chiếu được chứa trong danh sách ảnh tham chiếu. Chiều dự đoán chỉ báo bất kỳ một trong số dự đoán đơn hướng L0, dự đoán đơn hướng L1, hoặc dự đoán song hướng (dự đoán L0 và dự đoán L1). Ít nhất một trong số thông tin chuyển động chiều L0 và thông tin chuyển động chiều L1 có thể được sử dụng theo chiều dự đoán của khối hiện tại. Chỉ số hệ số trọng số hai chiều chỉ rõ hệ số trọng số được áp dụng tới khối dự đoán L0 và hệ số trọng số được áp dụng tới khối dự đoán L1.

FIG.6 là lưu đồ của phương pháp dự đoán liên đới theo phương án của sáng chế.

Viện dẫn tới FIG.6, phương pháp dự đoán liên đới bao gồm xác định chế độ dự đoán liên đới đối với khối hiện tại (S601), thu nhận thông tin chuyển động của khối hiện tại theo chế độ dự đoán liên đới được xác định (S602), và thực hiện dự đoán bù chuyển động đối với khối hiện tại trên cơ sở của thông tin chuyển động thu được (S603).

Theo đó, chế độ dự đoán liên đới có thể biểu diễn các phương pháp khác

nhau để xác định thông tin chuyển động của khối hiện tại, và bao gồm chế độ dự đoán liên đới nhờ sử dụng thông tin chuyển động dịch, chế độ dự đoán liên đới sử dụng thông tin chuyển động afin. Trong ví dụ của sáng chế, chế độ dự đoán liên đới sử dụng thông tin chuyển động dịch có thể bao gồm chế độ hợp nhất và chế độ dự đoán vector chuyển động, và chế độ dự đoán liên đới sử dụng thông tin chuyển động afin có thể bao gồm chế độ hợp nhất afin và chế độ dự đoán vector chuyển động afin. Thông tin chuyển động trên khối hiện tại có thể được xác định trên cơ sở của khối lân cận mà lân cận với khối hiện tại hoặc thông tin thu được bằng cách phân tích dòng bit. Thông tin chuyển động của khối hiện tại có thể được thu nhận từ thông tin chuyển động của khối khác. Theo đó, khối khác có thể là khối được mã hóa/giải mã bởi việc dự đoán liên đới trước khối hiện tại. Thiết lập thông tin chuyển động của khối hiện tại là tương tự như thông tin chuyển động của khối khác có thể được xác định là chế độ hợp nhất. Ngoài ra, thiết lập vector chuyển động của khối khác như là giá trị dự đoán của vector chuyển động của khối hiện tại có thể được xác định là chế độ dự đoán vector chuyển động.

Sau đây, phương pháp dự đoán liên đới nhờ sử dụng thông tin chuyển động afin được mô tả chi tiết.

FIG.7 là sơ đồ minh họa chuyển động phi tuyến của đối tượng.

Trong video, chuyển động phi tuyến của đối tượng có thể diễn ra. Theo ví dụ của sáng chế, như trong ví dụ được thể hiện trên FIG.7, chuyển động phi tuyến của đối tượng có thể diễn ra như phóng to, thu nhỏ camera, quay hoặc biến đổi afin, v.v. Đối với chuyển động phi tuyến của đối tượng, vector chuyển động dịch có thể không biểu diễn một cách hiệu quả chuyển động của đối tượng. Do đó, đối với vùng trong đó chuyển động phi tuyến của đối tượng diễn ra, hiệu quả mã hóa có thể được cải thiện bằng cách sử dụng chuyển động afin, thay vì chuyển động dịch.

FIG.8 là lưu đồ của phương pháp dự đoán liên đới dựa trên chuyển động afin theo phương án của sáng chế.

Việc phương pháp dự đoán liên đới dựa trên chuyển động afin có được áp dụng tới khối hiện tại hay không có thể được xác định dựa trên thông tin được phân tích từ dòng bit. Cụ thể, dựa trên ít nhất một trong số cờ mà chỉ báo rằng chế độ hợp nhất afin có được áp dụng tới khối hiện tại hay không hoặc cờ mà chỉ báo rằng chế độ dự đoán vector chuyển động afin có được áp dụng tới khối hiện tại hay

không, phương pháp dự đoán liên đới dựa trên chuyển động afin có được áp dụng tới khối hiện tại hay không sẽ được xác định.

Khi phương pháp dự đoán liên đới dựa trên chuyển động afin được áp dụng tới khối hiện tại, mô hình chuyển động afin đối với khối hiện tại có thể được xác định S801. Mô hình chuyển động afin có thể được xác định như là ít nhất một trong số mô hình chuyển động afin 6 tham số hoặc mô hình chuyển động afin 4 tham số. Mô hình chuyển động afin 6 tham số biểu diễn chuyển động afin bằng cách sử dụng 6 tham số và mô hình chuyển động afin 4 tham số biểu diễn chuyển động afin bằng cách sử dụng 4 tham số.

Công thức 1 biểu diễn chuyển động afin bằng cách sử dụng 6 tham số. Chuyển động afin biểu diễn chuyển động dịch đối với vùng định trước được xác định bởi các vectơ hạt afin.

【Công thức 1】

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

Trong trường hợp biểu diễn chuyển động afin bằng cách sử dụng 6 tham số, chuyển động phức tạp có thể được biểu diễn, nhưng hiệu quả mã hóa có thể bị giảm đi do nhiều bit hơn cần để mã hóa mỗi tham số. Do đó, chuyển động afin có thể được biểu diễn bằng cách sử dụng 4 tham số. Công thức 2 biểu diễn chuyển động afin bằng cách sử dụng 4 tham số.

【Công thức 2】

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

Thông tin để xác định mô hình chuyển động afin của khối hiện tại có thể được mã hóa và báo hiệu trong dòng bit. Theo ví dụ của sáng chế, thông tin này có thể là cờ 1-bit, 'affine_type_flag'. Nếu giá trị của cờ là 0, có thể biểu diễn rằng mô hình chuyển động afin 4 tham số được áp dụng và nếu giá trị của cờ là 1, có thể biểu diễn rằng mô hình chuyển động afin 6 tham số được áp dụng. Cờ có thể được mã hóa trong đơn vị của lát, ô, hoặc khối (ví dụ, khối mã hóa hoặc đơn vị cây mã hóa). Khi cờ được báo hiệu tại mức lát, mô hình chuyển động afin được xác định tại mức lát có thể được áp dụng tới tất cả các khối thuộc về lát.

Ngoài ra, dựa trên chế độ dự đoán liên đới afin của khối hiện tại, mô hình chuyển động afin của khối hiện tại có thể được xác định. Theo ví dụ của sáng chế, khi chế độ hợp nhất afin được áp dụng, mô hình chuyển động afin của khối hiện tại có thể được xác định như là mô hình chuyển động 4 tham số. Mặt khác, khi chế độ dự đoán vector chuyển động afin được áp dụng, thông tin để xác định mô hình chuyển động afin của khối hiện tại có thể được mã hóa và báo hiệu trong dòng bit. Theo ví dụ của sáng chế, khi chế độ dự đoán vector chuyển động afin được áp dụng tới khối hiện tại, mô hình chuyển động afin của khối hiện tại có thể được xác định dựa trên cờ 1-bit, 'affine_type_flag'.

Tiếp theo, các vector hạt afin của khối hiện tại có thể được thu nhận S802. Khi mô hình chuyển động afin 4 tham số được lựa chọn, các vector chuyển động tại hai điểm điều khiển đối với khối hiện tại có thể được thu nhận. Mặt khác, khi mô hình chuyển động afin 6 tham số được lựa chọn, các vector chuyển động tại ba điểm điều khiển đối với khối hiện tại có thể được thu nhận. Vector chuyển động tại điểm điều khiển có thể được gọi là vector hạt afin. Điểm điều khiển có thể bao gồm ít nhất một trong số góc bên trái-trên cùng, góc bên phải-trên cùng hoặc góc bên trái-dưới cùng của khối hiện tại.

FIG.9 là sơ đồ minh họa các vector hạt afin đối với mỗi mô hình chuyển động afin.

Trong mô hình chuyển động afin 4 tham số, các vector hạt afin đối với hai trong số góc bên trái-trên cùng, góc bên phải-trên cùng hoặc góc bên trái-dưới cùng có thể được thu nhận. Theo ví dụ của sáng chế, như trong ví dụ được thể hiện trên FIG.9 (a), khi mô hình chuyển động afin 4 tham số được lựa chọn, vector afin có thể được thu nhận bằng cách sử dụng vector hạt afin sv0 đối với góc bên trái-trên cùng của khối hiện tại (ví dụ mẫu bên trái-trên cùng (x1, y1)) và vector hạt afin sv1 đối với góc bên phải-trên cùng của khối hiện tại (ví dụ mẫu bên phải-trên cùng (x1, y1)). Có thể sử dụng vector hạt afin đối với góc bên trái-dưới cùng thay vì vector hạt afin đối với góc bên trái-trên cùng hoặc sử dụng vector hạt afin đối với góc bên trái-dưới cùng thay vì vector hạt afin đối với góc bên phải-trên cùng.

Trong mô hình chuyển động afin 6 tham số, các vector hạt afin đối với góc bên trái-trên cùng, góc bên phải-trên cùng và góc bên trái-dưới cùng có thể được thu nhận. Theo ví dụ của sáng chế, như trong ví dụ được thể hiện trong FIG.9 (b),

khi mô hình chuyển động afin 6 tham số được lựa chọn, vector afin có thể được thu nhận bằng cách sử dụng vector hạt afin sv0 đối với góc bên trái-trên cùng của khối hiện tại (ví dụ mẫu bên trái-trên cùng (x_1, y_1)), vector hạt afin sv1 đối với góc bên phải-trên cùng của khối hiện tại (ví dụ mẫu bên phải-trên cùng (x_1, y_1)) và vector hạt afin sv2 đối với góc bên trái-trên cùng của khối hiện tại (ví dụ mẫu bên trái-trên cùng (x_2, y_2)).

Trong phương án được mô tả sau đây, theo mô hình chuyển động afin 4 tham số, các vector hạt afin tại điểm điều khiển bên trái-trên cùng và điểm điều khiển bên phải-trên cùng được gọi là vector hạt afin thứ nhất và vector hạt afin thứ hai, một cách lần lượt. Trong các phương án được mô tả sau đây mà sử dụng vector hạt afin thứ nhất và vector hạt afin thứ hai, ít nhất một trong số vector hạt afin thứ nhất và vector hạt afin thứ hai có thể được thay thế bởi vector hạt afin tại điểm điều khiển bên trái-dưới cùng (vector hạt afin thứ ba) hoặc vector hạt afin tại điểm điều khiển bên phải-dưới cùng (vector hạt afin thứ tư).

Ngoài ra, theo mô hình chuyển động afin 6 tham số, các vector hạt afin tại điểm điều khiển bên trái-trên cùng, điểm điều khiển bên phải-trên cùng và điểm điều khiển bên trái-dưới cùng được gọi là vector hạt afin thứ nhất, vector hạt afin thứ hai và vector hạt afin thứ ba, một cách lần lượt. Trong các phương án được mô tả sau đây mà sử dụng vector hạt afin thứ nhất, vector hạt afin thứ hai và vector hạt afin thứ ba, ít nhất một trong số vector hạt afin thứ nhất, vector hạt afin thứ hai và vector hạt afin thứ ba có thể được thay thế bởi vector hạt afin tại điểm điều khiển bên phải-dưới cùng (vector hạt afin thứ tư).

Vector afin có thể được thu nhận theo khối con bằng cách sử dụng các vector hạt afin S803. Liên quan đến việc này, vector afin biểu diễn vector chuyển động dịch được thu nhận dựa trên các vector hạt afin. Vector afin của khối con có thể được gọi là vector chuyển động khối con afin hoặc vector chuyển động khối con.

FIG.10 là sơ đồ minh họa các vector afin của các khối con theo mô hình chuyển động 4 tham số.

Vector afin của khối con có thể được thu nhận dựa trên vị trí của điểm điều khiển, vị trí của khối con và vector hạt afin. Theo ví dụ của sáng chế, công thức 3 biểu diễn ví dụ về việc thu nhận vector chuyển động khối con afin.

【Công thức 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 công thức 3, (x, y) biểu diễn vị trí của khối con. Liên quan đến việc này, vị trí của khối con biểu diễn vị trí của mẫu cơ sở được chứa trong khối con. Mẫu cơ sở có thể là mẫu có vị trí tại góc bên trái-trên cùng của khối con hoặc mẫu mà ít nhất một trong số tọa độ trục x hoặc tọa độ trục y nằm tại vị trí trung tâm. (x₀, y₀) biểu diễn vị trí của điểm điều khiển thứ nhất và (sv_{0x}, sv_{0y}) biểu diễn vector hạt afin thứ nhất. Ngoài ra, (x₁, y₁) biểu diễn vị trí của điểm điều khiển thứ hai và (sv_{1x}, sv_{1y}) biểu diễn vector hạt afin thứ hai.

Khi điểm điều khiển thứ nhất và điểm điều khiển thứ hai tương ứng với góc bên trái-trên cùng và góc bên phải-trên cùng của khối hiện tại, một cách lần lượt, x₁-x₀ có thể được thiết lập là giá trị đồng nhất với độ rộng của khối hiện tại.

Sau đó, dự đoán bù chuyển động đối với mỗi khối con có thể được thực hiện bằng cách sử dụng vector afin của mỗi khối con S804. Theo kết quả của việc thực hiện dự đoán bù chuyển động, khối dự đoán đối với mỗi khối con có thể được tạo ra. Các khối dự đoán của các khối con có thể được thiết lập là khối dự đoán của khối hiện tại.

Vector hạt afin của khối hiện tại có thể được thu nhận dựa trên vector hạt afin của khối lân cận mà lân cận khối hiện tại. Khi chế độ dự đoán liên đới của khối hiện tại là chế độ hợp nhất afin, vector hạt afin của ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất có thể được xác định như là vector hạt afin của khối hiện tại. Ngoài ra, khi chế độ dự đoán liên đới của khối hiện tại là chế độ hợp nhất afin, thông tin chuyển động bao gồm ít nhất một trong số chỉ số ảnh tham chiếu, cờ dự đoán có hướng cụ thể hoặc trọng số hai chiều của khối hiện tại có thể cũng được thiết lập tương tự như ứng viên hợp nhất.

Tiếp theo, phương pháp dự đoán liên đới sử dụng thông tin chuyển động dịch sẽ được mô tả chi tiết.

Thông tin chuyển động của khối hiện tại có thể được thu nhận từ thông tin

chuyển động của khối khác. Theo đó, khối khác có thể là khối được mã hóa/giải mã bởi việc dự đoán liên đới trước khối hiện tại. Thiết lập thông tin chuyển động của khối hiện tại là tương tự như thông tin chuyển động của khối khác có thể được xác định là chế độ hợp nhất. Ngoài ra, thiết lập vector chuyển động của khối khác như là giá trị dự đoán của vector chuyển động của khối hiện tại có thể được xác định là chế độ dự đoán vector chuyển động.

FIG.11 là lưu đồ của xử lý thu nhận thông tin chuyển động của khối hiện tại dưới chế độ hợp nhất.

Ứng viên hợp nhất của khối hiện tại có thể được thu nhận S1101. Ứng viên hợp nhất của khối hiện tại có thể được thu nhận từ khối được mã hóa/giải mã bởi việc dự đoán liên đới trước khối hiện tại.

FIG.12 là sơ đồ minh họa các khối ứng viên được sử dụng để thu nhận ứng viên hợp nhất.

Các khối ứng viên có thể bao gồm ít nhất một trong số các khối lân cận bao gồm mẫu liền kề với khối hiện tại hoặc các khối không lân cận bao gồm mẫu không liền kề với khối hiện tại. Sau đây, các mẫu mà xác định các khối ứng viên được xác định là các mẫu cơ sở. Ngoài ra, mẫu cơ sở liền kề với khối hiện tại được gọi là mẫu cơ sở lân cận và mẫu cơ sở không liền kề với khối hiện tại được gọi là mẫu cơ sở không lân cận.

Mẫu cơ sở lân cận có thể được chứa trong cột lân cận của cột ngoài cùng bên trái của khối hiện tại hoặc hàng lân cận của hàng trên cùng của khối hiện tại. Trong ví dụ của sáng chế, khi tọa độ của mẫu bên trái-trên cùng của khối hiện tại là $(0,0)$, ít nhất một trong số khối bao gồm mẫu cơ sở tại vị trí $(-1, H-1)$, $(W-1, -1)$, $(W, -1)$, $(-1, H)$ hoặc $(-1, 1)$ có thể được sử dụng là khối ứng viên. Việc dẫn tới sơ đồ, các khối lân cận có chỉ số 0 đến 4 có thể được sử dụng như là các khối ứng viên.

Mẫu cơ sở không lân cận biểu diễn mẫu mà ít nhất một trong số khoảng cách trục x hoặc khoảng cách trục y với mẫu cơ sở liền kề với khối hiện tại có giá trị định trước. Trong ví dụ của sáng chế, ít nhất một trong số khối bao gồm mẫu cơ sở mà khoảng cách trục x với mẫu cơ sở bên trái là giá trị định trước, khối bao gồm mẫu không lân cận mà khoảng cách trục y với mẫu cơ sở trên cùng là giá trị định trước hoặc khối bao gồm mẫu không lân cận mà khoảng cách trục x và

khoảng cách trực y với mẫu cơ sở bên trái-trên cùng là giá trị định trước có thể được sử dụng như là khối ứng viên. Giá trị định trước có thể là số tự nhiên như 4, 8, 12, 16, v.v. Việc dẫn tới sơ đồ, ít nhất một trong số các khối trong chỉ số 5 đến 26 có thể được sử dụng như là khối ứng viên.

Mẫu mà không có vị trí trên cùng đường dọc, đường ngang, hoặc đường chéo như mẫu cơ sở lân cận có thể được thiết lập là mẫu cơ sở không lân cận.

Thông tin chuyển động của ứng viên hợp nhất có thể được thiết lập tương tự như thông tin chuyển động của khối ứng viên. Trong ví dụ của sáng chế, ít nhất một trong số vector chuyển động, chỉ số ảnh tham chiếu, chiều dự đoán hoặc chỉ số trọng số hai chiều của khối ứng viên có thể được thiết lập là thông tin chuyển động của ứng viên hợp nhất.

Danh sách ứng viên hợp nhất bao gồm ứng viên hợp nhất có thể được tạo ra S1102.

Chỉ số của các ứng viên hợp nhất trong danh sách ứng viên hợp nhất có thể được gán theo thứ tự định trước. Trong ví dụ của sáng chế, chỉ số có thể được gán trong thứ tự của ứng viên hợp nhất thu được từ khối lân cận bên trái, ứng viên hợp nhất thu được từ khối lân cận trên cùng, ứng viên hợp nhất thu được từ khối lân cận bên phải-trên cùng, ứng viên hợp nhất thu được từ khối lân cận bên trái-dưới cùng, ứng viên hợp nhất thu được từ khối lân cận bên trái-trên cùng và ứng viên hợp nhất thu được từ khối lân cận theo thời gian.

Khi các ứng viên hợp nhất được chứa trong ứng viên hợp nhất, ít nhất một trong số các ứng viên hợp nhất có thể được lựa chọn S1103. Cụ thể, thông tin để chỉ rõ bất kỳ một trong số các ứng viên hợp nhất có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, thông tin, `merge_idx`, mà biểu diễn chỉ số của bất kỳ một trong số các ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất có thể được báo hiệu trong dòng bit.

Khi số lượng ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất nhỏ hơn ngưỡng, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất như là ứng viên hợp nhất. Theo đó, ngưỡng có thể là số lượng lớn nhất của ứng viên hợp nhất mà có thể được chứa trong danh sách ứng viên hợp nhất hoặc giá trị trong đó độ dịch được trừ từ số lượng lớn nhất của ứng viên hợp nhất. Độ dịch có thể là số

tự nhiên như 1 hoặc 2, v.v

Bảng thông tin chuyển động bao gồm ứng viên thông tin chuyển động thu được từ khối được mã hóa/giải mã dựa trên việc dự đoán liên đới trong ảnh hiện tại. Trong ví dụ của sáng chế, thông tin chuyển động của ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể được thiết lập tương tự như thông tin chuyển động của khối được mã hóa/giải mã dựa trên việc dự đoán liên đới. Theo đó, thông tin chuyển động có thể bao gồm ít nhất một trong số vector chuyển động, chỉ số ảnh tham chiếu, chiều dự đoán hoặc chỉ số trọng số hai chiều.

Ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động cũng có thể được gọi là ứng viên hợp nhất vùng liên đới hoặc ứng viên hợp nhất vùng dự đoán.

Số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động có thể được xác định trước trong bộ mã hóa và bộ giải mã. Trong ví dụ của sáng chế, số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động có thể là 1, 2, 3, 4, 5, 6, 7, 8 hoặc lớn hơn (ví dụ 16).

Ngoài ra, thông tin mà biểu diễn số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động có thể được báo hiệu trong dòng bit. Thông tin này có thể được báo hiệu trong mức chuỗi, ảnh, hoặc lát. Thông tin này có thể biểu diễn số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động. Ngoài ra, thông tin này có thể biểu diễn độ chênh lệch giữa số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động và số lượng ứng viên hợp nhất lớn nhất mà có thể được chứa trong danh sách ứng viên hợp nhất.

Ngoài ra, số lượng ứng viên thông tin chuyển động lớn nhất mà có thể được chứa trong Bảng thông tin chuyển động có thể được xác định theo kích cỡ ảnh, kích cỡ lát hoặc kích cỡ đơn vị cây mã hóa.

Bảng thông tin chuyển động có thể được khởi tạo trong đơn vị của ảnh, lát, ô, thanh, đơn vị cây mã hóa hoặc dòng đơn vị cây mã hóa (hàng hoặc cột). Trong ví dụ của sáng chế, khi lát được khởi tạo, Bảng thông tin chuyển động cũng được khởi tạo, do đó Bảng thông tin chuyển động có thể không bao gồm bất kỳ ứng

viên thông tin chuyển động.

Ngoài ra, thông tin mà biểu diễn rằng Bảng thông tin chuyển động sẽ được khởi tạo hay không có thể được báo hiệu trong dòng bit. Thông tin này có thể được báo hiệu trong mức lát, ô, thành hoặc khối. Cho đến khi thông tin này chỉ báo việc khởi tạo của Bảng thông tin chuyển động, Bảng thông tin chuyển động được cấu hình trước có thể được sử dụng.

Ngoài ra, thông tin về ứng viên thông tin chuyển động khởi tạo có thể được báo hiệu trong tập hợp tham số ảnh hoặc thông tin tiêu đề lát. Mặc dù lát được khởi tạo, Bảng thông tin chuyển động có thể bao gồm ứng viên thông tin chuyển động khởi tạo. Do đó, ứng viên thông tin chuyển động khởi tạo có thể được sử dụng đối với khối mà là đích mã hóa/giải mã đầu tiên trong lát.

Ngoài ra, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động của đơn vị cây mã hóa trước đó có thể được thiết lập là ứng viên thông tin chuyển động khởi tạo. Trong ví dụ của sáng chế, ứng viên thông tin chuyển động với chỉ số nhỏ nhất hoặc với chỉ số lớn nhất trong số các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động của đơn vị cây mã hóa trước đó có thể được thiết lập là ứng viên thông tin chuyển động khởi tạo.

Các khối được mã hóa/giải mã trong thứ tự mã hóa/giải mã, và các khối được mã hóa/giải mã dựa trên việc dự đoán liên đới có thể được thiết lập tuần tự là ứng viên thông tin chuyển động trong thứ tự mã hóa/giải mã.

FIG.13 là sơ đồ để giải thích khía cạnh cập nhật của Bảng thông tin chuyển động.

Đối với khối hiện tại, khi việc dự đoán liên đới được thực hiện S1301, ứng viên thông tin chuyển động có thể được thu nhận dựa trên khối hiện tại S1302. Thông tin chuyển động của ứng viên thông tin chuyển động có thể được thiết lập tương tự như của khối hiện tại.

Khi Bảng thông tin chuyển động là trống S1303, ứng viên thông tin chuyển động thu được dựa trên khối hiện tại có thể được thêm vào Bảng thông tin chuyển động S1304.

Khi Bảng thông tin chuyển động đã bao gồm ứng viên thông tin chuyển

động S1303, việc kiểm tra dư thừa đối với thông tin chuyển động của khối hiện tại (hoặc, ứng viên thông tin chuyển động thu được dựa trên điều này) có thể được thực hiện S1305. Việc kiểm tra dư thừa là để xác định rằng thông tin chuyển động của ứng viên thông tin chuyển động được lưu trữ trước trong Bảng thông tin chuyển động là tương tự như thông tin chuyển động của khối hiện tại. Việc kiểm tra dư thừa có thể được thực hiện đối với tất cả ứng viên thông tin chuyển động được lưu trữ trước trong Bảng thông tin chuyển động. Ngoài ra, việc kiểm tra dư thừa có thể được thực hiện đối với các ứng viên thông tin chuyển động với chỉ số vượt quá hoặc dưới ngưỡng trong số các ứng viên thông tin chuyển động được lưu trữ trước trong Bảng thông tin chuyển động. Ngoài ra, việc kiểm tra dư thừa có thể được thực hiện đối với các ứng viên thông tin chuyển động được xác định trước. Trong ví dụ của sáng chế, 2 các ứng viên thông tin chuyển động với các chỉ số nhỏ nhất hoặc với các chỉ số lớn nhất có thể được xác định như là các đích cho việc kiểm tra dư thừa.

Khi ứng viên thông tin chuyển động với cùng thông tin chuyển động như khối hiện tại không được bao gồm, ứng viên thông tin chuyển động thu được dựa trên khối hiện tại có thể được thêm vào Bảng thông tin chuyển động S1308. Việc các ứng viên thông tin chuyển động có đồng nhất hay không có thể được xác định dựa trên việc thông tin chuyển động (ví dụ vectơ chuyển động/chỉ số ảnh tham chiếu, v.v) của các ứng viên thông tin chuyển động có đồng nhất hay không.

Theo đó, khi số lượng ứng viên thông tin chuyển động lớn nhất đã được lưu trữ trong Bảng thông tin chuyển động S1306, ứng viên thông tin chuyển động cũ nhất có thể được xóa S1307 và ứng viên thông tin chuyển động thu được dựa trên khối hiện tại có thể được thêm vào Bảng thông tin chuyển động S1308. Theo đó, ứng viên thông tin chuyển động cũ nhất có thể là ứng viên thông tin chuyển động với chỉ số lớn nhất hoặc chỉ số nhỏ nhất.

Các ứng viên thông tin chuyển động có thể được nhận dạng bởi chỉ số tương ứng. Khi ứng viên thông tin chuyển động thu được từ khối hiện tại được thêm vào Bảng thông tin chuyển động, chỉ số nhỏ nhất (ví dụ 0) có thể được gán tới ứng viên thông tin chuyển động và các chỉ số của các ứng viên thông tin chuyển động được lưu trữ trước có thể được tăng thêm 1. Theo đó, khi số lượng ứng viên thông tin chuyển động lớn nhất đã được lưu trữ trong Bảng thông tin chuyển động, ứng viên thông tin chuyển động với chỉ số lớn nhất được loại bỏ.

Ngoài ra, khi ứng viên thông tin chuyển động thu được từ khối hiện tại được thêm vào Bảng thông tin chuyển động, chỉ số lớn nhất có thể được gán tới ứng viên thông tin chuyển động. Trong ví dụ của sáng chế, khi số lượng ứng viên thông tin chuyển động được lưu trữ trước trong Bảng thông tin chuyển động nhỏ hơn giá trị lớn nhất, chỉ số với cùng giá trị như số lượng ứng viên thông tin chuyển động được lưu trữ trước có thể được gán tới ứng viên thông tin chuyển động. Ngoài ra, khi số lượng ứng viên thông tin chuyển động được lưu trữ trước trong Bảng thông tin chuyển động bằng giá trị lớn nhất, chỉ số mà trừ 1 từ giá trị lớn nhất có thể được gán tới ứng viên thông tin chuyển động. Ngoài ra, ứng viên thông tin chuyển động với chỉ số nhỏ nhất được loại bỏ và các chỉ số của các ứng viên thông tin chuyển động dư được lưu trữ trước được giảm đi 1.

FIG.14 là sơ đồ thể hiện khía cạnh cập nhật của Bảng thông tin chuyển động.

Giả thiết rằng khi ứng viên thông tin chuyển động thu được từ khối hiện tại được thêm vào Bảng thông tin chuyển động, chỉ số lớn nhất được gán tới ứng viên thông tin chuyển động. Ngoài ra, giả thiết rằng số lượng ứng viên thông tin chuyển động lớn nhất đã được lưu trữ trong Bảng thông tin chuyển động.

Khi ứng viên thông tin chuyển động $HmvpCand[n+1]$ thu được từ khối hiện tại được thêm vào Bảng thông tin chuyển động $HmvpCandList$, ứng viên thông tin chuyển động $HmvpCand[0]$ với chỉ số nhỏ nhất trong số các ứng viên thông tin chuyển động được lưu trữ trước có thể được xóa và các chỉ số của các ứng viên thông tin chuyển động dư có thể được giảm đi 1. Ngoài ra, chỉ số của ứng viên thông tin chuyển động $HmvpCand[n+1]$ thu được từ khối hiện tại có thể được thiết lập là giá trị lớn nhất (ví dụ được thể hiện trên FIG.14 là n).

Khi ứng viên thông tin chuyển động đồng nhất với ứng viên thông tin chuyển động thu được dựa trên khối hiện tại được lưu trữ trước S1305, ứng viên thông tin chuyển động thu được dựa trên khối hiện tại có thể không được thêm vào Bảng thông tin chuyển động S1309.

Ngoài ra, trong khi ứng viên thông tin chuyển động thu được dựa trên khối hiện tại được thêm vào Bảng thông tin chuyển động, ứng viên thông tin chuyển động được lưu trữ trước đồng nhất với ứng viên thông tin chuyển động có thể được loại bỏ. Trong trường hợp này, có thể gây ra cùng hiệu ứng như khi chỉ số của ứng viên thông tin chuyển động được lưu trữ trước được cập nhật mới.

FIG.15 là sơ đồ thể hiện ví dụ trong đó chỉ số của ứng viên thông tin chuyển động được lưu trữ trước được cập nhật.

Khi chỉ số của ứng viên thông tin chuyển động được lưu trữ trước đồng nhất với ứng viên thông tin chuyển động mvCand thu được từ khối hiện tại là hIdx, ứng viên thông tin chuyển động được lưu trữ trước có thể được loại bỏ và chỉ số của các ứng viên thông tin chuyển động với chỉ số lớn hơn hIdx có thể được giảm đi 1. Trong ví dụ của sáng chế, ví dụ được thể hiện trên FIG.15 thể hiện rằng HmvpCand[2] đồng nhất với mvCand được xóa trong Bảng thông tin chuyển động HmvpCandList và chỉ số từ HmvpCand[3] đến HmvpCand[n] được giảm đi 1.

Và ứng viên thông tin chuyển động mvCand thu được dựa trên khối hiện tại có thể được thêm vào phần cuối của Bảng thông tin chuyển động.

Ngoài ra, chỉ số được gán tới ứng viên thông tin chuyển động được lưu trữ trước đồng nhất với ứng viên thông tin chuyển động thu được dựa trên khối hiện tại có thể được cập nhật. Ví dụ, chỉ số của ứng viên thông tin chuyển động được lưu trữ trước có thể được thay đổi thành giá trị nhỏ nhất hoặc giá trị lớn nhất.

Thông tin chuyển động của các khối được chứa trong vùng định trước có thể được thiết lập để không được thêm vào Bảng thông tin chuyển động. Trong ví dụ của sáng chế, ứng viên thông tin chuyển động thu được dựa trên thông tin chuyển động của khối được chứa trong vùng xử lý hợp nhất có thể không được thêm vào Bảng thông tin chuyển động. Do thứ tự mã hóa/giải mã đối với các khối được chứa trong vùng xử lý hợp nhất không được xác định, việc sử dụng thông tin chuyển động của bất kỳ một trong số khối cho việc dự đoán liên đới của khối khác là không hợp lý. Do đó, các ứng viên thông tin chuyển động thu được dựa trên các khối được chứa trong vùng xử lý hợp nhất có thể không được thêm vào Bảng thông tin chuyển động.

Ngoài ra, thông tin chuyển động của khối nhỏ hơn kích cỡ được thiết lập trước có thể được thiết lập để không được thêm vào Bảng thông tin chuyển động. Trong ví dụ của sáng chế, ứng viên thông tin chuyển động thu được dựa trên thông tin chuyển động của khối mã hóa mà độ rộng hoặc độ cao của nó nhỏ hơn 4 hoặc 8 hoặc thông tin chuyển động của khối mã hóa có kích cỡ 4x4 có thể không được thêm vào Bảng thông tin chuyển động.

Dựa trên chế độ dự đoán liên đới của khối hiện tại, có thể được xác định

rằng khối hiện tại sẽ được sử dụng như là ứng viên thông tin chuyển động hay không. Trong ví dụ của sáng chế, khối được mã hóa/giải mã dựa trên mô hình chuyển động afin có thể được thiết lập là không khả dụng như ứng viên thông tin chuyển động. Do đó, mặc dù khối hiện tại được mã hóa/giải mã bởi việc dự đoán liên đới, Bảng thông tin chuyển động có thể không được cập nhật dựa trên khối hiện tại khi chế độ dự đoán liên đới của khối hiện tại là chế độ dự đoán afin.

Ứng viên thông tin chuyển động có thể được thiết lập để bao gồm thông tin bổ sung ngoại trừ thông tin chuyển động. Trong ví dụ của sáng chế, ít nhất một trong số kích cỡ, dạng hoặc thông tin phân chia của khối có thể được lưu trữ thêm trong ứng viên thông tin chuyển động. Khi danh sách ứng viên hợp nhất của khối hiện tại được cấu hình, chỉ ứng viên thông tin chuyển động mà kích cỡ, dạng hoặc thông tin phân chia của nó là đồng nhất hoặc tương tự với khối hiện tại trong số các ứng viên thông tin chuyển động có thể được sử dụng hoặc ứng viên thông tin chuyển động mà kích cỡ, dạng hoặc thông tin phân chia của nó là đồng nhất hoặc tương tự với khối hiện tại có thể được thêm vào danh sách ứng viên hợp nhất trước.

Khi số lượng ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất của khối hiện tại nhỏ hơn ngưỡng, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất như là ứng viên hợp nhất. Xử lý bổ sung được thực hiện trong thứ tự mà thể hiện thứ tự được sắp xếp của các chỉ số của các ứng viên thông tin chuyển động theo thứ tự tăng dần hoặc giảm dần. Trong ví dụ của sáng chế, ứng viên thông tin chuyển động với chỉ số lớn nhất có thể được thêm đầu tiên vào danh sách ứng viên hợp nhất của khối hiện tại.

Khi ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động được thêm vào danh sách ứng viên hợp nhất, việc kiểm tra dư thừa giữa ứng viên thông tin chuyển động và các ứng viên hợp nhất được lưu trữ trước trong danh sách ứng viên hợp nhất có thể được thực hiện. Theo kết quả của việc kiểm tra dư thừa, ứng viên thông tin chuyển động với cùng thông tin chuyển động như ứng viên hợp nhất được lưu trữ trước có thể không được thêm vào danh sách ứng viên hợp nhất.

Việc kiểm tra dư thừa có thể được thực hiện chỉ đối với một phần của các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động.

Trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với ứng viên thông tin chuyển động với chỉ số cao hơn hoặc dưới ngưỡng. Ngoài ra, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với N ứng viên thông tin chuyển động với chỉ số lớn nhất hoặc chỉ số nhỏ nhất.

Ngoài ra, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với một phần của các ứng viên hợp nhất được lưu trữ trước trong danh sách ứng viên hợp nhất. Trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với ứng viên hợp nhất mà chỉ số của nó cao hơn hoặc dưới ngưỡng hoặc ứng viên hợp nhất thu được từ khối tại vị trí cụ thể. Theo đó, vị trí cụ thể có thể bao gồm ít nhất một trong số khối lân cận bên trái, khối lân cận trên cùng, khối lân cận bên phải-trên cùng hoặc khối lân cận bên trái-dưới cùng của khối hiện tại.

FIG.16 là sơ đồ thể hiện ví dụ trong đó việc kiểm tra dư thừa được thực hiện chỉ đối với một phần của các ứng viên hợp nhất.

Khi ứng viên thông tin chuyển động `HmvpCand[j]` được thêm vào danh sách ứng viên hợp nhất, việc kiểm tra dư thừa với 2 ứng viên hợp nhất có chỉ số lớn nhất, `mergeCandList[NumMerge-2]` và `mergeCandList[NumMerge-1]`, có thể được thực hiện đối với ứng viên thông tin chuyển động. Theo đó, `NumMerge` có thể thể hiện số lượng ứng viên hợp nhất theo không gian khả dụng và ứng viên hợp nhất theo thời gian.

Không giống ví dụ được thể hiện, khi ứng viên thông tin chuyển động `HmvpCand[j]` được thêm vào danh sách ứng viên hợp nhất, việc kiểm tra dư thừa với 2 ứng viên hợp nhất có chỉ số nhỏ nhất có thể được thực hiện đối với ứng viên thông tin chuyển động. Ví dụ, có thể được kiểm tra rằng `mergeCandList[0]` và `mergeCandList[1]` có đồng nhất với `HmvpCand[j]` hay không.

Ngoài ra, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với ứng viên hợp nhất thu được từ vị trí cụ thể. trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện đối với ít nhất một trong số ứng viên hợp nhất thu được từ khối lân cận có vị trí tại bên trái của khối hiện tại hoặc tại trên cùng của khối hiện tại. Khi không có ứng viên hợp nhất thu được từ vị trí cụ thể trong danh sách ứng viên hợp nhất, ứng viên thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất mà không cần kiểm tra dư thừa.

Khi ứng viên thông tin chuyển động `HmvpCand[j]` được thêm vào danh

sách ứng viên hợp nhất, việc kiểm tra dư thừa với 2 ứng viên hợp nhất có chỉ số lớn nhất, `mergeCandList[NumMerge-2]` và `mergeCandList[NumMerge-1]`, có thể được thực hiện đối với ứng viên thông tin chuyển động. Theo đó, `NumMerge` có thể thể hiện số lượng ứng viên hợp nhất theo không gian khả dụng và ứng viên hợp nhất theo thời gian.

Việc kiểm tra dư thừa với ứng viên hợp nhất có thể được thực hiện chỉ đối với một phần của các ứng viên thông tin chuyển động. Trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với N ứng viên thông tin chuyển động với chỉ số lớn hoặc nhỏ trong số các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động. Trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với các ứng viên thông tin chuyển động với chỉ số mà số lượng và độ chênh lệch các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động nằm dưới ngưỡng. Khi ngưỡng bằng 2, việc kiểm tra dư thừa có thể được thực hiện chỉ đối với 3 ứng viên thông tin chuyển động với giá trị chỉ số lớn nhất trong số các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động. Việc kiểm tra dư thừa có thể được bỏ qua đối với các ứng viên thông tin chuyển động ngoại trừ 3 ứng viên thông tin chuyển động nêu trên. Khi việc kiểm tra dư thừa được bỏ qua, ứng viên thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất bất kể thông tin chuyển động tương tự như ứng viên hợp nhất có tồn tại hay không.

Ngược lại, việc kiểm tra dư thừa được thiết lập là chỉ được thực hiện đối với các ứng viên thông tin chuyển động với chỉ số mà số lượng và độ chênh lệch của các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động nằm trên ngưỡng.

Số lượng của ứng viên thông tin chuyển động mà việc kiểm tra dư thừa được thực hiện có thể được xác định lại trong bộ mã hóa và bộ giải mã. Trong ví dụ của sáng chế, ngưỡng có thể là số nguyên như 0, 1 hoặc 2.

Ngoài ra, ngưỡng có thể được xác định dựa trên ít nhất một trong số lượng của ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất hoặc số lượng của các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động.

Khi ứng viên hợp nhất đồng nhất với ứng viên thông tin chuyển động thứ nhất được tìm thấy, việc kiểm tra dư thừa với ứng viên hợp nhất đồng nhất với

ứng viên thông tin chuyển động thứ nhất có thể được bỏ qua trong việc kiểm tra dư thừa đối với ứng viên thông tin chuyển động thứ hai.

FIG.17 là sơ đồ thể hiện ví dụ trong đó việc kiểm tra dư thừa với ứng viên hợp nhất cụ thể được bỏ qua.

Khi ứng viên thông tin chuyển động $HmvpCand[i]$ mà chỉ số của nó là i được thêm vào danh sách ứng viên hợp nhất, việc kiểm tra dư thừa giữa ứng viên thông tin chuyển động và các ứng viên hợp nhất được lưu trữ trước trong danh sách ứng viên hợp nhất được thực hiện. Theo đó, khi ứng viên hợp nhất $mergeCandlist[j]$ đồng nhất với ứng viên thông tin chuyển động $HmvpCand[i]$ được tìm thấy, việc kiểm tra dư thừa giữa ứng viên thông tin chuyển động $HmvpCand[i-1]$ mà chỉ số của nó là $i-1$ và các ứng viên hợp nhất có thể được thực hiện mà không cần thêm ứng viên thông tin chuyển động $HmvpCand[i]$ vào danh sách ứng viên hợp nhất. Theo đó, việc kiểm tra dư thừa giữa ứng viên thông tin chuyển động $HmvpCand[i-1]$ và ứng viên hợp nhất $mergeCandList[j]$ có thể được bỏ qua.

Trong ví dụ của sáng chế, trong ví dụ được thể hiện trên FIG.17, được xác định rằng $HmvpCand[i]$ và $mergeCandList[2]$ là đồng nhất. Do đó, việc kiểm tra dư thừa đối với $HmvpCand[i-1]$ có thể được thực hiện mà không cần thêm $HmvpCand[i]$ vào danh sách ứng viên hợp nhất. Theo đó, việc kiểm tra dư thừa giữa $HmvpCand[i-1]$ và $mergeCandList[2]$ có thể được bỏ qua.

Khi số lượng của ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất của khối hiện tại nhỏ hơn ngưỡng, ít nhất một trong số ứng viên hợp nhất cặp đôi hoặc ứng viên hợp nhất 0 có thể được bao gồm thêm ngoại trừ ứng viên thông tin chuyển động. Ứng viên hợp nhất cặp đôi có nghĩa là ứng viên hợp nhất mà có giá trị thu được từ việc lấy trung bình các vector chuyển động của nhiều hơn 2 ứng viên hợp nhất như là vector chuyển động và ứng viên hợp nhất 0 có nghĩa là ứng viên hợp nhất mà vector chuyển động của nó là 0.

Đối với danh sách ứng viên hợp nhất của khối hiện tại, ứng viên hợp nhất có thể được thêm vào theo thứ tự sau đây.

Ứng viên hợp nhất theo không gian – ứng viên hợp nhất theo thời gian gian – ứng viên thông tin chuyển động – (ứng viên thông tin chuyển động afin) – ứng viên hợp nhất cặp đôi – ứng viên hợp nhất 0

Ứng viên hợp nhất theo không gian có nghĩa là ứng viên hợp nhất thu được từ ít nhất một của khối lân cận hoặc khối không lân cận và ứng viên hợp nhất thời gian có nghĩa là ứng viên hợp nhất thu được từ ảnh tham chiếu trước đó. Ứng viên thông tin chuyển động affin biểu diễn ứng viên thông tin chuyển động thu được từ khối được mã hóa/giải mã bởi mô hình chuyển động affin.

Bảng thông tin chuyển động có thể cũng được sử dụng trong chế độ dự đoán vector chuyển động. Trong ví dụ của sáng chế, khi số lượng ứng viên dự đoán vector chuyển động được chứa trong danh sách ứng viên dự đoán vector chuyển động của khối hiện tại nhỏ hơn ngưỡng, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể được thiết lập là ứng viên dự đoán vector chuyển động đối với khối hiện tại. Cụ thể, vector chuyển động của ứng viên thông tin chuyển động có thể được thiết lập là ứng viên dự đoán vector chuyển động.

Nếu bất kỳ một trong số các ứng viên dự đoán vector chuyển động được chứa trong danh sách ứng viên dự đoán vector chuyển động của khối hiện tại được lựa chọn, ứng viên được lựa chọn có thể được thiết lập là tham số dự đoán vector chuyển động của khối hiện tại. Sau đó, sau khi giá trị dư vector chuyển động của khối hiện tại được giải mã, vector chuyển động của khối hiện tại có thể được thu nhận bằng cách cộng tham số dự đoán vector chuyển động và giá trị dư vector chuyển động.

Danh sách ứng viên dự đoán vector chuyển động của khối hiện tại có thể được cấu hình trong thứ tự sau đây.

Ứng viên dự đoán vector chuyển động không gian – Ứng viên dự đoán vector chuyển động thời gian – Ứng viên thông tin chuyển động – (Ứng viên thông tin chuyển động affin) – ứng viên dự đoán vector chuyển động 0

Ứng viên dự đoán vector chuyển động không gian có nghĩa là ứng viên dự đoán vector chuyển động thu được từ ít nhất một trong số khối lân cận hoặc khối không lân cận và ứng viên dự đoán vector chuyển động theo thời gian có nghĩa là ứng viên dự đoán vector chuyển động thu được từ ảnh tham chiếu trước đó. Ứng viên thông tin chuyển động affin biểu diễn ứng viên thông tin chuyển động thu được từ khối được mã hóa/giải mã bởi mô hình chuyển động affin. Ứng viên dự đoán vector chuyển động 0 biểu diễn ứng viên mà giá trị của vector chuyển động là 0.

Vùng xử lý hợp nhất lớn hơn khối mã hóa có thể được xác định. Các khối mã hóa được chứa trong vùng xử lý hợp nhất có thể được xử lý song song mà không được mã hóa/giải mã tuần tự. Theo đó, không được mã hóa/giải mã tuần tự có nghĩa là thứ tự của việc mã hóa/giải mã không được xác định. Do đó, xử lý mã hóa/giải mã của các khối được chứa trong vùng xử lý hợp nhất có thể được xử lý độc lập. Ngoài ra, các khối được chứa trong vùng xử lý hợp nhất có thể chia sẻ các ứng viên hợp nhất. Theo đó, các ứng viên hợp nhất có thể được thu nhận dựa trên vùng xử lý hợp nhất.

Theo đặc điểm nêu trên, vùng xử lý hợp nhất có thể được gọi là vùng xử lý song song, vùng hợp nhất được chia sẻ (SMR-shared merge region) hoặc vùng ước lượng hợp nhất (MER-merge estimation region).

Ứng viên hợp nhất của khối hiện tại có thể được thu nhận dựa trên khối mã hóa. Tuy nhiên, khi khối hiện tại được chứa trong vùng xử lý hợp nhất lớn hơn khối hiện tại, khối ứng viên được chứa trong cùng vùng xử lý hợp nhất như khối hiện tại có thể được thiết lập là không khả dụng như ứng viên hợp nhất.

FIG.18 là sơ đồ thể hiện ví dụ trong đó khối ứng viên được chứa trong cùng vùng xử lý hợp nhất như khối hiện tại được thiết lập là không khả dụng như là ứng viên hợp nhất.

Trong ví dụ được thể hiện trên FIG.18 (a), trong việc mã hóa/giải mã CU5, các khối bao gồm các mẫu cơ sở liền kề với CU5 có thể được thiết lập là các khối ứng viên. Theo đó, các khối ứng viên X3 và X4 được chứa trong cùng vùng xử lý hợp nhất như CU5 có thể được thiết lập là không khả dụng như là ứng viên hợp nhất của CU5. Tuy nhiên, các khối ứng viên X0, X1 và X2 không được chứa trong cùng vùng xử lý hợp nhất như CU5 có thể được thiết lập là khả dụng như là ứng viên hợp nhất.

Trong ví dụ được thể hiện trên FIG.18 (b), trong việc mã hóa/giải mã của CU8, các khối bao gồm các mẫu cơ sở liền kề với CU8 có thể được thiết lập là các khối ứng viên. Theo đó, các khối ứng viên X6, X7 và X8 được chứa trong cùng vùng xử lý hợp nhất như CU8 có thể được thiết lập là không khả dụng như là ứng viên hợp nhất. Tuy nhiên, các khối ứng viên X5 và X9 không được chứa trong cùng vùng xử lý hợp nhất như CU8 có thể được thiết lập là khả dụng như là ứng viên hợp nhất.

Ngoài ra, khi khối hiện tại được chứa trong vùng xử lý hợp nhất, khối lân cận liền kề với khối hiện tại và vùng xử lý hợp nhất có thể được thiết lập là khối ứng viên.

FIG.19 là sơ đồ thể hiện ví dụ mà thu nhận ứng viên hợp nhất đối với khối hiện tại khi khối hiện tại được chứa trong vùng xử lý hợp nhất.

Như trong ví dụ được thể hiện trên FIG.19 (a), các khối lân cận liền kề với khối hiện tại có thể được thiết lập là các khối ứng viên để thu nhận ứng viên hợp nhất của khối hiện tại. Theo đó, khối ứng viên được chứa trong cùng vùng xử lý hợp nhất như khối hiện tại có thể được thiết lập là không khả dụng như là ứng viên hợp nhất. Trong ví dụ của sáng chế, trong khi thu nhận ứng viên hợp nhất đối với khối mã hóa CU3, khối lân cận trên cùng y3 và khối lân cận bên phải-trên cùng y4 được chứa trong cùng vùng xử lý hợp nhất như khối mã hóa CU3 có thể được thiết lập là không khả dụng như là ứng viên hợp nhất của khối mã hóa CU3.

Bằng cách quét các khối lân cận liền kề với khối hiện tại trong thứ tự định trước, ứng viên hợp nhất có thể được thu nhận. Trong ví dụ của sáng chế, thứ tự định trước có thể là thứ tự y1, y3, y4, y0 và y2.

Khi số lượng các ứng viên hợp nhất mà có thể được thu nhận từ các khối lân cận liền kề với khối hiện tại nhỏ hơn giá trị mà độ dịch được trừ từ số lượng các ứng viên hợp nhất lớn nhất hoặc số lượng lớn nhất, ứng viên hợp nhất đối với khối hiện tại có thể được thu nhận bằng cách sử dụng các khối lân cận liền kề với vùng xử lý hợp nhất tương tự ví dụ được thể hiện trên FIG.19 (b). Trong ví dụ của sáng chế, các khối lân cận liền kề với vùng xử lý hợp nhất bao gồm khối mã hóa CU3 có thể được thiết lập là các khối ứng viên đối với khối mã hóa CU3. Theo đó, các khối lân cận liền kề với vùng xử lý hợp nhất có thể bao gồm ít nhất một trong số khối lân cận bên trái x1, khối lân cận trên cùng x3, khối lân cận bên trái-dưới cùng x0, khối lân cận bên phải-trên cùng x4 hoặc khối lân cận bên trái-trên cùng x2.

Bằng cách quét các khối lân cận liền kề với vùng xử lý hợp nhất trong thứ tự định trước, ứng viên hợp nhất có thể được thu nhận. Trong ví dụ của sáng chế, thứ tự định trước có thể là thứ tự x1, x3, x4, x0 và x2.

Tóm lại, ứng viên hợp nhất trên khối mã hóa CU3 mà bao gồm trong vùng xử lý hợp nhất có thể được thu nhận bằng cách quét các khối ứng viên trong thứ

tự quét sau đây.

$(y1, y3, y4, y0, y2, x1, x3, x4, x0, x2)$

Tuy nhiên, thứ tự quét của các khối ứng viên được minh họa nêu trên chỉ thể hiện ví dụ của sáng chế và các khối ứng viên có thể được quét trong thứ tự khác ví dụ nêu trên. Ngoài ra, thứ tự quét có thể được xác định thích nghi dựa trên ít nhất một trong số kích cỡ hoặc dạng của khối hiện tại hoặc vùng xử lý hợp nhất.

Vùng xử lý hợp nhất có thể là hình vuông hoặc không phải hình vuông. Thông tin để xác định vùng xử lý hợp nhất có thể được báo hiệu trong dòng bit. Thông tin này có thể bao gồm ít nhất một trong số thông tin mà biểu diễn dạng của vùng xử lý hợp nhất hoặc thông tin mà biểu diễn kích cỡ của vùng xử lý hợp nhất. Khi vùng xử lý hợp nhất không phải hình vuông, ít nhất một trong số thông tin mà biểu diễn kích cỡ của vùng xử lý hợp nhất, thông tin mà biểu diễn độ rộng hoặc độ cao của vùng xử lý hợp nhất hoặc thông tin mà biểu diễn tỷ lệ giữa độ rộng và độ cao của vùng xử lý hợp nhất có thể được báo hiệu trong dòng bit.

Kích cỡ của vùng xử lý hợp nhất có thể được xác định dựa trên ít nhất một trong số thông tin được báo hiệu trong dòng bit, độ phân giải ảnh, kích cỡ của lát hoặc kích cỡ của ô.

Nếu dự đoán bù chuyển động được thực hiện đối với khối được chứa trong vùng xử lý hợp nhất, ứng viên thông tin chuyển động thu được dựa trên thông tin chuyển động của khối trong đó dự đoán bù chuyển động được thực hiện có thể được thêm vào Bảng thông tin chuyển động.

Tuy nhiên, nếu ứng viên thông tin chuyển động thu được từ khối được chứa trong vùng xử lý hợp nhất được thêm vào Bảng thông tin chuyển động, trường hợp có thể diễn ra trong đó ứng viên thông tin chuyển động thu được từ khối được sử dụng trong việc mã hóa/giải mã của khối khác trong vùng xử lý hợp nhất mà việc mã hóa/giải mã thực sự thấp hơn khối. Nói cách khác, mặc dù sự phụ thuộc giữa các khối sẽ được loại trừ trong khi mã hóa/giải mã các khối được chứa trong vùng xử lý hợp nhất, trường hợp có thể xảy ra trong đó việc bù dự đoán chuyển động được thực hiện bằng cách sử dụng thông tin chuyển động của khối khác được chứa trong vùng xử lý hợp nhất. Để giải quyết vấn đề này, mặc dù việc mã hóa/giải mã của khối được chứa trong vùng xử lý hợp nhất được hoàn thành, thông tin chuyển động của khối mà việc mã hóa/giải mã của nó được hoàn

thành có thể không được thêm vào Bảng thông tin chuyển động.

Ngoài ra, Bảng thông tin chuyển động có thể được cập nhật nhờ sử dụng chỉ khối tại vị trí định trước trong vùng xử lý hợp nhất. Các ví dụ của vị trí định trước có thể bao gồm ít nhất một trong số khối có vị trí tại trên cùng bên trái của vùng xử lý hợp nhất, khối có vị trí tại trên cùng bên phải của vùng xử lý hợp nhất, khối có vị trí tại dưới cùng bên trái của vùng xử lý hợp nhất, khối có vị trí tại dưới cùng bên phải của vùng xử lý hợp nhất, khối có vị trí trong trung tâm của vùng xử lý hợp nhất, khối liền kề với biên bên phải của vùng xử lý hợp nhất, và khối liền kề với biên dưới cùng của vùng xử lý hợp nhất. Theo ví dụ của sáng chế, Bảng thông tin chuyển động có thể được cập nhật chỉ với thông tin chuyển động của khối liền kề với góc dưới cùng bên phải của vùng xử lý hợp nhất và Bảng thông tin chuyển động có thể không được cập nhật với thông tin chuyển động của các khối khác.

Ngoài ra, sau khi việc giải mã tất cả khối được chứa trong vùng xử lý hợp nhất được hoàn thành, ứng viên thông tin chuyển động thu được từ các khối có thể được thêm vào Bảng thông tin chuyển động. Tức là, trong khi các khối được chứa trong vùng xử lý hợp nhất được mã hóa/giải mã, Bảng thông tin chuyển động có thể không được cập nhật.

Trong ví dụ của sáng chế, nếu việc dự đoán bù chuyển động được thực hiện đối với các khối được chứa trong vùng xử lý hợp nhất, ứng viên thông tin chuyển động thu được từ các khối có thể được thêm vào Bảng thông tin chuyển động theo thứ tự định trước. Theo đó, thứ tự định trước có thể được xác định trong thứ tự quét của các khối mã hóa trong vùng xử lý hợp nhất hoặc đơn vị cây mã hóa. Thứ tự quét có thể là ít nhất một trong số quét mảnh, quét ngang, quét dọc hoặc quét zigzắc. Ngoài ra, thứ tự định trước có thể được xác định dựa trên thông tin chuyển động của mỗi khối hoặc số lượng khối với cùng thông tin chuyển động.

Ngoài ra, ứng viên thông tin chuyển động bao gồm thông tin chuyển động vô hướng có thể được thêm vào Bảng thông tin chuyển động trước ứng viên thông tin chuyển động bao gồm thông tin chuyển động hai chiều. Ngược lại, ứng viên thông tin chuyển động bao gồm thông tin chuyển động hai chiều có thể được thêm vào Bảng thông tin chuyển động trước ứng viên thông tin chuyển động bao gồm thông tin chuyển động vô hướng.

Ngoài ra, ứng viên thông tin chuyển động có thể được thêm vào Bảng

thông tin chuyển động trong thứ tự của tần số cao để sử dụng hoặc tần số thấp để sử dụng trong vùng xử lý hợp nhất hoặc đơn vị cây mã hóa.

Khi khối hiện tại được chứa trong vùng xử lý hợp nhất và số lượng các ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất của khối hiện tại nhỏ hơn số lượng lớn nhất, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất. Theo đó, ứng viên thông tin chuyển động thu được từ khối được chứa trong cùng vùng xử lý hợp nhất như khối hiện tại có thể được thiết lập để không được thêm vào danh sách ứng viên hợp nhất của khối hiện tại.

Ngoài ra, khi khối hiện tại được chứa trong vùng xử lý hợp nhất, có thể được thiết lập để không sử dụng ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động. Nói cách khác, mặc dù số lượng các ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất của khối hiện tại nhỏ hơn số lượng lớn nhất, ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động có thể không được thêm vào danh sách ứng viên hợp nhất.

Trong ví dụ khác, Bảng thông tin chuyển động trên vùng xử lý hợp nhất hoặc đơn vị cây mã hóa có thể được cấu hình. Bảng thông tin chuyển động này đóng vai trò để lưu trữ tạm thời thông tin chuyển động của các khối được chứa trong vùng xử lý hợp nhất. Để phân biệt giữa Bảng thông tin chuyển động chung và Bảng thông tin chuyển động đối với vùng xử lý hợp nhất hoặc đơn vị cây mã hóa, Bảng thông tin chuyển động đối với vùng xử lý hợp nhất hoặc đơn vị cây mã hóa được gọi là Bảng thông tin chuyển động tạm thời. Và ứng viên thông tin chuyển động được lưu trữ trong Bảng thông tin chuyển động tạm thời được gọi là ứng viên thông tin chuyển động tạm thời.

FIG.20 là sơ đồ thể hiện Bảng thông tin chuyển động tạm thời.

Bảng thông tin chuyển động tạm thời đối với đơn vị cây mã hóa hoặc vùng xử lý hợp nhất có thể được cấu hình. Khi dự đoán bù chuyển động được thực hiện đối với khối hiện tại được chứa trong đơn vị cây mã hóa hoặc vùng xử lý hợp nhất, thông tin chuyển động của khối có thể không được thêm vào Bảng thông tin chuyển động HmvpCandList. Thay vì đó, ứng viên thông tin chuyển động tạm thời thu được từ khối có thể được thêm vào Bảng thông tin chuyển động tạm thời HmvpMERCandList. Nói cách khác, ứng viên thông tin chuyển động tạm thời được thêm vào Bảng thông tin chuyển động tạm thời có thể không được thêm vào

Bảng thông tin chuyển động. Do đó, Bảng thông tin chuyển động có thể không bao gồm ứng viên thông tin chuyển động thu được dựa trên thông tin chuyển động của các khối được chứa trong đơn vị cây mã hóa hoặc vùng xử lý hợp nhất bao gồm khối hiện tại.

Ngoài ra, chỉ thông tin chuyển động của một vài khối được chứa trong vùng xử lý hợp nhất có thể được thêm vào Bảng thông tin chuyển động tạm thời. Theo ví dụ của sáng chế, chỉ các khối tại các vị trí định trước trong vùng xử lý hợp nhất có thể được sử dụng để cập nhật Bảng thông tin chuyển động. Các vị trí định trước có thể bao gồm ít nhất một trong số khối có vị trí tại trên cùng bên trái của vùng xử lý hợp nhất, khối có vị trí tại trên cùng bên phải của vùng xử lý hợp nhất, khối có vị trí tại dưới cùng bên trái của vùng xử lý hợp nhất, khối có vị trí tại dưới cùng bên phải của vùng xử lý hợp nhất, khối có vị trí trong trung tâm của vùng xử lý hợp nhất, khối liền kề với biên bên phải của vùng xử lý hợp nhất, và khối liền kề với biên dưới cùng của vùng xử lý hợp nhất. Theo ví dụ của sáng chế, chỉ thông tin chuyển động của khối liền kề với góc dưới cùng bên phải của vùng xử lý hợp nhất có thể được thêm vào Bảng thông tin chuyển động tạm thời và thông tin chuyển động của các khối khác có thể không được thêm vào Bảng thông tin chuyển động tạm thời.

Số lượng lớn nhất của các ứng viên thông tin chuyển động tạm thời mà Bảng thông tin chuyển động tạm thời có thể bao gồm có thể được thiết lập bằng số lượng ứng viên thông tin chuyển động lớn nhất mà Bảng thông tin chuyển động có thể bao gồm. Ngoài ra, số lượng các ứng viên thông tin chuyển động tạm thời lớn nhất mà Bảng thông tin chuyển động tạm thời có thể bao gồm có thể được xác định theo kích cỡ của đơn vị cây mã hóa hoặc vùng xử lý hợp nhất. Ngoài ra, số lượng lớn nhất của các ứng viên thông tin chuyển động tạm thời mà Bảng thông tin chuyển động tạm thời có thể bao gồm có thể được thiết lập nhỏ hơn số lượng ứng viên thông tin chuyển động lớn nhất mà Bảng thông tin chuyển động có thể bao gồm.

Khối hiện tại được chứa trong đơn vị cây mã hóa hoặc vùng xử lý hợp nhất có thể được thiết lập để không sử dụng Bảng thông tin chuyển động tạm thời trên đơn vị cây mã hóa hoặc vùng xử lý hợp nhất tương ứng. Nói cách khác, khi số lượng các ứng viên hợp nhất được chứa trong danh sách ứng viên hợp nhất của khối hiện tại nhỏ hơn ngưỡng, ứng viên thông tin chuyển động được chứa trong

Bảng thông tin chuyển động có thể được thêm vào danh sách ứng viên hợp nhất và ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời có thể không được thêm vào danh sách ứng viên hợp nhất. Do đó, thông tin chuyển động của khối khác bao gồm trong cùng đơn vị cây mã hóa hoặc cùng vùng xử lý hợp nhất như khối hiện tại có thể không được sử dụng đối với dự đoán bù chuyển động của khối hiện tại.

Nếu việc mã hóa/giải mã của tất cả khối được chứa trong đơn vị cây mã hóa hoặc vùng xử lý hợp nhất được hoàn thành, Bảng thông tin chuyển động và Bảng thông tin chuyển động tạm thời có thể được hợp nhất.

FIG.21 là sơ đồ thể hiện ví dụ trong đó Bảng thông tin chuyển động và Bảng thông tin chuyển động tạm thời được hợp nhất.

Nếu việc mã hóa/giải mã của tất cả khối được chứa trong đơn vị cây mã hóa hoặc vùng xử lý hợp nhất được hoàn thành, ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời có thể được cập nhật trong Bảng thông tin chuyển động như trong ví dụ được thể hiện trên FIG.21.

Theo đó, các ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời có thể được thêm vào Bảng thông tin chuyển động trong thứ tự được chèn trong Bảng thông tin chuyển động tạm thời. (Nói cách khác, trong thứ tự tăng dần hoặc thứ tự giảm dần của giá trị chỉ số)

Trong ví dụ khác, các ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời có thể được thêm vào Bảng thông tin chuyển động trong thứ tự định trước. Theo đó, thứ tự định trước có thể được xác định trong thứ tự quét của các khối mã hóa trong vùng xử lý hợp nhất hoặc đơn vị cây mã hóa. Thứ tự quét có thể là ít nhất một trong số quét mảnh, quét ngang, quét dọc hoặc quét ziczăc. Ngoài ra, thứ tự định trước có thể được xác định dựa trên thông tin chuyển động của mỗi khối hoặc số lượng khối với cùng thông tin chuyển động.

Ngoài ra, ứng viên thông tin chuyển động tạm thời bao gồm thông tin chuyển động vô hướng có thể được thêm vào Bảng thông tin chuyển động trước ứng viên thông tin chuyển động tạm thời bao gồm thông tin chuyển động hai chiều. Ngược lại, ứng viên thông tin chuyển động tạm thời bao gồm thông tin chuyển động hai chiều có thể được thêm vào Bảng thông tin chuyển động trước ứng viên

thông tin chuyển động tạm thời bao gồm thông tin chuyển động vô hướng.

Ngoài ra, ứng viên thông tin chuyển động tạm thời có thể được thêm vào Bảng thông tin chuyển động trong thứ tự của tần số cao để sử dụng hoặc tần số thấp để sử dụng trong vùng xử lý hợp nhất hoặc đơn vị cây mã hóa.

Trong trường hợp mà ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời được thêm vào Bảng thông tin chuyển động, việc kiểm tra dư thừa đối với ứng viên thông tin chuyển động tạm thời có thể được thực hiện. Trong ví dụ của sáng chế, khi cùng ứng viên thông tin chuyển động như ứng viên thông tin chuyển động tạm thời được chứa trong Bảng thông tin chuyển động tạm thời được lưu trữ trước trong Bảng thông tin chuyển động, ứng viên thông tin chuyển động tạm thời có thể không được thêm vào Bảng thông tin chuyển động. Theo đó, việc kiểm tra dư thừa có thể được thực hiện đối với một phần của các ứng viên thông tin chuyển động được chứa trong Bảng thông tin chuyển động. Trong ví dụ của sáng chế, việc kiểm tra dư thừa có thể được thực hiện đối với các ứng viên thông tin chuyển động với chỉ số cao hơn hoặc dưới ngưỡng. Trong ví dụ của sáng chế, khi ứng viên thông tin chuyển động tạm thời bằng ứng viên thông tin chuyển động với chỉ số cao hơn giá trị định trước, ứng viên thông tin chuyển động tạm thời có thể không được thêm vào Bảng thông tin chuyển động.

Có thể giới hạn việc sử dụng của ứng viên thông tin chuyển động thu được từ khối được chứa trong cùng đơn vị cây mã hóa hoặc cùng vùng xử lý hợp nhất như khối hiện tại như ứng viên hợp nhất của khối hiện tại. Đối với điều này, thông tin địa chỉ của khối có thể được lưu trữ bổ sung đối với ứng viên thông tin chuyển động. Thông tin địa chỉ của khối có thể bao gồm ít nhất một trong số vị trí của khối, địa chỉ của khối, chỉ số của khối, vị trí của vùng xử lý hợp nhất trong đó khối được bao gồm, địa chỉ của vùng xử lý hợp nhất trong đó khối được bao gồm, chỉ số của vùng xử lý hợp nhất trong đó khối được bao gồm, vị trí của vùng cây mã hóa trong đó khối được bao gồm, địa chỉ của vùng cây mã hóa trong đó khối được bao gồm hoặc chỉ số của vùng cây mã hóa trong đó khối được bao gồm.

Việc nội dự đoán là để dự đoán khối hiện tại bằng cách sử dụng mẫu được khôi phục mà đã được mã hóa/giải mã và nằm xung quanh khối hiện tại. Theo đó, mẫu được khôi phục trước khi áp dụng lọc vòng trong có thể được sử dụng đối với việc nội dự đoán của khối hiện tại.

Phương pháp nội dự đoán bao gồm việc nội dự đoán dựa trên ma trận và việc nội dự đoán theo chiều với mẫu khôi phục lân cận. Thông tin mà chỉ báo phương pháp nội dự đoán của khối hiện tại có thể được báo hiệu trong dòng bit. Thông tin này có thể là cờ 1-bit. Ngoài ra, việc nội dự đoán của khối hiện tại có thể được xác định trên cơ sở của ít nhất một trong số vị trí của khối hiện tại, kích cỡ của khối hiện tại, dạng của khối hiện tại, hoặc phương pháp nội dự đoán của khối lân cận. Trong ví dụ của sáng chế, khi khối hiện tại được hiện diện cắt qua biên ảnh, có thể được thiết lập sao cho phương pháp nội dự đoán dựa trên ma trận không được áp dụng tới khối hiện tại.

Phương pháp nội dự đoán dựa trên ma trận là phương pháp thu nhận khối dự đoán của khối hiện tại trên cơ sở của tích ma trận của ma trận được lưu trữ trong bộ mã hóa và bộ giải mã, và các mẫu khôi phục xung quanh khối hiện tại. Thông tin để chỉ rõ bất kỳ một trong số các ma trận được lưu trữ trước có thể được báo hiệu trong dòng bit. Bộ giải mã có thể xác định ma trận để thực hiện việc nội dự đoán trên khối hiện tại trên cơ sở của thông tin nêu trên và kích cỡ của khối hiện tại.

Việc nội dự đoán chung là phương pháp thu nhận khối dự đoán của khối hiện tại trên cơ sở của chế độ nội dự đoán vô hướng hoặc chế độ nội dự đoán có hướng. Sau đây, viện dẫn tới các hình vẽ, xử lý của việc nội dự đoán dựa trên việc nội dự đoán chung sẽ được mô tả chi tiết.

FIG.22 là lưu đồ của phương pháp nội dự đoán theo phương án của sáng chế.

Dòng mẫu tham chiếu của khối hiện tại có thể được xác định S2201. Dòng mẫu tham chiếu có nghĩa là nhóm của các mẫu tham chiếu được chứa trong dòng thứ k nằm cách phần trên cùng và/hoặc bên trái của khối hiện tại. Mẫu tham chiếu có thể được thu nhận từ mẫu được khôi phục được mã hóa/giải mã xung quanh khối hiện tại.

Thông tin chỉ số mà nhận dạng dòng mẫu tham chiếu đối với khối hiện tại trong số các dòng mẫu tham chiếu có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, thông tin chỉ số, `intra_luma_ref_idx`, để chỉ rõ dòng mẫu tham chiếu của khối hiện tại có thể được báo hiệu trong dòng bit. Thông tin chỉ số có thể được báo hiệu theo khối mã hóa.

Các dòng mẫu tham chiếu có thể bao gồm ít nhất một trong số dòng thứ nhất, dòng thứ hai hoặc dòng thứ ba tại trên cùng và/hoặc bên trái của khối hiện tại. Dòng mẫu tham chiếu bao gồm hàng liền kề với trên cùng của khối hiện tại và cột liền kề với bên trái của khối hiện tại trong số các dòng mẫu tham chiếu có thể được gọi là dòng mẫu tham chiếu lân cận, và các dòng mẫu tham chiếu còn lại có thể được gọi là dòng mẫu tham chiếu không lân cận.

Bảng 1 thể hiện chỉ số được gán tới mỗi dòng mẫu tham chiếu ứng viên.

【Bảng 1】

Index (intra_luma_ref_idx)	Dòng mẫu tham chiếu
0	Dòng mẫu tham chiếu lân cận
1	Dòng mẫu tham chiếu không lân cận thứ nhất
2	Dòng mẫu tham chiếu không lân cận thứ hai

Dựa trên ít nhất một trong số vị trí, kích cỡ, dạng của khối hiện tại hoặc chế độ mã hóa dự đoán của khối lân cận, dòng mẫu tham chiếu của khối hiện tại có thể được xác định. Trong một ví dụ, khi khối hiện tại liền kề với biên của ảnh, ô, lát hoặc đơn vị cây mã hóa, dòng mẫu tham chiếu liền kề có thể được xác định như là dòng mẫu tham chiếu của khối hiện tại.

Dòng mẫu tham chiếu có thể bao gồm các mẫu tham chiếu trên cùng có vị trí tại trên cùng của khối hiện tại và các mẫu tham chiếu bên trái có vị trí tại bên trái của khối hiện tại. Các mẫu tham chiếu trên cùng và các mẫu tham chiếu bên trái có thể được thu nhận từ các mẫu được khôi phục nằm xung quanh khối hiện tại. Các mẫu được khôi phục có thể nằm trong trạng thái trước khi bộ lọc vòng trong được áp dụng.

Tiếp theo, chế độ nội dự đoán của khối hiện tại có thể được xác định S2202. Đối với chế độ nội dự đoán của khối hiện tại, ít nhất một trong số chế độ nội dự đoán vô hướng hoặc chế độ nội dự đoán có hướng có thể được xác định như là chế độ nội dự đoán của khối hiện tại. Các chế độ nội dự đoán vô hướng bao gồm chế độ phẳng và DC và các chế độ nội dự đoán có hướng bao gồm 33 hoặc 65 chế độ từ chiều đường chéo bên trái-dưới cùng tới chiều đường chéo bên phải-trên cùng.

FIG.23 là sơ đồ thể hiện các chế độ nội dự đoán.

FIG.23 (a) thể hiện 35 chế độ nội dự đoán và FIG.23 (b) thể hiện 67 chế độ nội dự đoán.

Số lượng chế độ nội dự đoán lớn hơn hoặc nhỏ hơn so với được thể hiện trên FIG.23 có thể được xác định.

Dựa trên chế độ nội dự đoán của khối lân cận liền kề với khối hiện tại, MPM (Most Probable Mode-Chế độ có khả năng xảy ra nhất) có thể được thiết lập. Theo đó, khối lân cận có thể bao gồm khối lân cận bên trái liền kề với bên trái của khối hiện tại và khối lân cận trên cùng liền kề với trên cùng của khối hiện tại.

Số lượng MPM được chứa trong danh sách MPM có thể được thiết lập trước trong bộ mã hóa và bộ giải mã. Trong ví dụ của sáng chế, số lượng MPM có thể là 3, 4, 5 hoặc 6. Ngoài ra, thông tin mà biểu diễn số lượng MPM có thể được báo hiệu trong dòng bit. Ngoài ra, dựa trên ít nhất một trong số chế độ mã hóa dự đoán của khối lân cận, kích cỡ, dạng hoặc chỉ số dòng mẫu tham chiếu của khối hiện tại, số lượng MPM có thể được xác định. Trong ví dụ của sáng chế, trong khi N MPM có thể được sử dụng khi dòng mẫu tham chiếu lân cận được xác định như là dòng mẫu tham chiếu của khối hiện tại, M MPM có thể được sử dụng khi dòng mẫu tham chiếu không lân cận được xác định như là dòng mẫu tham chiếu của khối hiện tại. Khi M là số tự nhiên nhỏ hơn N, trong ví dụ của sáng chế, N có thể là 6 và M có thể là 5, 4 hoặc 3. Do đó, trong khi chế độ nội dự đoán của khối hiện tại có thể được xác định như là bất kỳ một trong số 6 chế độ nội dự đoán ứng viên khi chỉ số của dòng mẫu tham chiếu của khối hiện tại là 0 và cờ MPM là đúng, chế độ nội dự đoán của khối hiện tại có thể được xác định như là bất kỳ một trong số 5 chế độ nội dự đoán ứng viên khi chỉ số của dòng mẫu tham chiếu của khối hiện tại lớn hơn 0 và cờ là đúng.

Ngoài ra, số lượng cố định (ví dụ 6 hoặc 5) của các ứng viên MPM có thể được sử dụng bất kể chỉ số của dòng mẫu tham chiếu của khối hiện tại.

Danh sách MPM bao gồm các MPM có thể được tạo ra và thông tin mà chỉ báo rằng MPM tương tự như chế độ nội dự đoán của khối hiện tại có được chứa trong danh sách MPM hay không có thể được báo hiệu trong dòng bit. Khi thông tin này là cờ 1-bit, thông tin này có thể được gọi là cờ MPM. Khi cờ MPM

biểu diễn cùng MPM như khối hiện tại được chứa trong danh sách MPM, thông tin chỉ số mà nhận dạng một trong số các MPM có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, thông tin chỉ số, `mpm_idx`, mà chỉ rõ bất kỳ một trong số các MPM có thể được báo hiệu trong dòng bit. MPM được chỉ rõ bởi thông tin chỉ số có thể được thiết lập là chế độ nội dự đoán của khối hiện tại. Khi cờ MPM biểu diễn cùng MPM như khối hiện tại không được chứa trong danh sách MPM, thông tin chế độ còn lại mà chỉ báo bất kỳ một trong số các chế độ nội dự đoán còn lại ngoại trừ các MPM có thể được báo hiệu trong dòng bit. Thông tin chế độ còn lại biểu diễn giá trị chỉ số tương ứng với chế độ nội dự đoán của khối hiện tại khi chỉ số được gán lại tới các chế độ nội dự đoán còn lại ngoại trừ các MPM. Bộ giải mã có thể có thể xác định chế độ nội dự đoán của khối hiện tại bằng cách sắp xếp các MPM trong thứ tự giảm dần và so sánh thông tin chế độ còn lại với các MPM. Trong ví dụ của sáng chế, khi thông tin chế độ còn lại bằng hoặc nhỏ hơn MPM, chế độ nội dự đoán của khối hiện tại có thể được thu nhận bằng cách cộng 1 vào thông tin chế độ còn lại.

Khi thu nhận chế độ nội dự đoán của khối hiện tại, so sánh một phần của các MPM với thông tin chế độ còn lại có thể được bỏ qua. Trong ví dụ của sáng chế, các MPM trong chế độ nội dự đoán vô hướng trong số các MPM có thể được loại trừ khỏi đích sánh. Khi các chế độ nội dự đoán vô hướng được thiết lập như là các MPM, thông tin chế độ còn lại chỉ báo rõ ràng chế độ nội dự đoán có hướng, vì vậy chế độ nội dự đoán của khối hiện tại có thể được thu nhận bằng cách so sánh các MPM còn lại ngoại trừ các chế độ nội dự đoán vô hướng với thông tin chế độ còn lại. Thay vì loại trừ các chế độ nội dự đoán vô hướng từ đích so sánh, giá trị kết quả có thể được so sánh với các MPM còn lại sau khi cộng số lượng chế độ nội dự đoán vô hướng vào thông tin chế độ còn lại.

Thay vì thiết lập chế độ mặc định như là MPM, thông tin mà chỉ báo rằng chế độ nội dự đoán của khối hiện tại có phải là chế độ mặc định hay không có thể được báo hiệu trong dòng bit. Thông tin này là cờ 1-bit và cờ có thể được gọi là cờ chế độ mặc định. Cờ chế độ mặc định có thể được báo hiệu chỉ khi cờ MPM biểu diễn rằng cùng MPM như khối hiện tại được chứa trong danh sách MPM. Như được mô tả nêu trên, chế độ mặc định có thể bao gồm ít nhất một trong số chế độ phẳng, DC, chế độ chiều dọc hoặc chế độ chiều ngang. Trong ví dụ của sáng chế, khi chế độ phẳng được thiết lập như là chế độ mặc định, cờ chế độ mặc định có thể chỉ báo rằng chế độ nội dự đoán của khối hiện tại có phải là chế độ

phẳng hay không. Khi cờ chế độ mặc định chỉ báo rằng chế độ nội dự đoán của khối hiện tại không phải chế độ mặc định, một trong số các MPM được chỉ báo bởi thông tin chỉ số có thể được thiết lập là chế độ nội dự đoán của khối hiện tại.

Khi cờ chế độ mặc định được sử dụng, có thể được thiết lập rằng chế độ nội dự đoán tương tự với chế độ mặc định không được thiết lập là MPM. Trong ví dụ của sáng chế, khi cờ chế độ mặc định chỉ báo rằng chế độ nội dự đoán của khối hiện tại có phải là chế độ phẳng hay không, chế độ nội dự đoán của khối hiện tại có thể được thu nhận bằng cách sử dụng 5 MPM ngoại trừ MPM tương ứng với chế độ phẳng.

Khi các chế độ nội dự đoán được thiết lập là các chế độ mặc định, thông tin chỉ số mà chỉ báo bất kỳ một trong số các chế độ mặc định có thể còn được báo hiệu. Chế độ nội dự đoán của khối hiện tại có thể được thiết lập là chế độ mặc định được chỉ báo bởi thông tin chỉ số.

Khi chỉ số của dòng mẫu tham chiếu của khối hiện tại không phải 0, có thể được thiết lập để không sử dụng chế độ mặc định. Trong ví dụ của sáng chế, khi dòng mẫu tham chiếu không lân cận được xác định là dòng mẫu tham chiếu của khối hiện tại, có thể được thiết lập để không sử dụng chế độ nội dự đoán vô hướng như chế độ DC hoặc chế độ phẳng. Do đó, khi chỉ số của dòng mẫu tham chiếu không phải 0, cờ chế độ mặc định có thể không được báo hiệu và giá trị của cờ chế độ mặc định có thể được hiểu là giá trị định trước (tức là sai).

Khi chế độ nội dự đoán của khối hiện tại được xác định, các mẫu dự đoán đối với khối hiện tại có thể được thu nhận dựa trên chế độ nội dự đoán định trước S2203.

Khi chế độ DC được lựa chọn, các mẫu dự đoán đối với khối hiện tại có thể được tạo ra dựa trên giá trị trung bình của các mẫu tham chiếu. Một cách chi tiết, các giá trị của tất cả mẫu nằm trong khối dự đoán có thể được tạo ra dựa trên giá trị trung bình của các mẫu tham chiếu. Giá trị trung bình có thể được thu nhận nhờ sử dụng ít nhất một trong số các mẫu tham chiếu trên cùng liền kề với trên cùng của khối hiện tại, và các mẫu tham chiếu bên trái liền kề với bên trái của khối hiện tại.

Số lượng hoặc phạm vi của các mẫu tham chiếu được sử dụng khi thu nhận giá trị trung bình có thể thay đổi dựa trên dạng của khối hiện tại. Trong ví dụ của

sáng chế, khi khối hiện tại là khối không phải hình vuông trong đó độ rộng lớn hơn độ cao, giá trị trung bình có thể được tính toán bằng cách sử dụng các mẫu tham chiếu trên cùng. Ngược lại, khi khối hiện tại là khối không phải hình vuông trong đó độ rộng nhỏ hơn độ cao, giá trị trung bình có thể được tính toán bằng cách sử dụng các mẫu tham chiếu bên trái. Nói cách khác, khi độ rộng và độ cao của khối hiện tại là khác nhau, các mẫu tham chiếu liền kề với độ dài lớn hơn có thể được áp dụng để tính toán giá trị trung bình. Ngoài ra, việc tính toán giá trị trung bình bằng cách sử dụng các mẫu tham chiếu trên cùng hay bằng cách sử dụng các mẫu tham chiếu bên trái có thể được xác định trên cơ sở của tỷ lệ giữa độ rộng và độ cao của khối hiện tại.

Khi chế độ phẳng được lựa chọn, mẫu dự đoán có thể được thu nhận bằng cách sử dụng mẫu dự đoán theo chiều ngang và mẫu dự đoán theo chiều dọc. Theo đó, mẫu dự đoán theo chiều ngang có thể được thu nhận trên cơ sở của mẫu tham chiếu bên trái và mẫu tham chiếu bên phải mà có vị trí tại cùng dòng theo chiều ngang với mẫu dự đoán, và mẫu dự đoán theo chiều dọc có thể được thu nhận trên cơ sở của mẫu tham chiếu trên cùng và mẫu tham chiếu dưới cùng mà có vị trí tại cùng dòng theo chiều dọc với mẫu dự đoán. Theo đó, mẫu tham chiếu bên phải có thể được tạo ra bằng cách sao chép mẫu tham chiếu liền kề với góc trên cùng-bên phải của khối hiện tại, và mẫu tham chiếu dưới cùng có thể được tạo ra bằng cách sao chép mẫu tham chiếu liền kề với góc dưới bên trái của khối hiện tại. Mẫu dự đoán theo chiều ngang có thể được thu nhận trên cơ sở của tổng có trọng số của mẫu tham chiếu bên trái và mẫu tham chiếu bên phải, và mẫu dự đoán theo chiều dọc có thể được thu nhận trên cơ sở của tổng có trọng số của mẫu tham chiếu trên cùng và mẫu tham chiếu dưới cùng. Theo đó, hệ số trọng số được gán tới mỗi mẫu tham chiếu có thể được xác định theo vị trí của mẫu dự đoán. Mẫu dự đoán có thể được thu nhận trên cơ sở của giá trị trung bình hoặc tổng có trọng số của mẫu dự đoán theo chiều ngang và mẫu dự đoán theo chiều dọc. Khi tổng có trọng số được sử dụng, hệ số trọng số được gán tới mẫu dự đoán theo chiều ngang và mẫu dự đoán theo chiều dọc có thể được xác định trên cơ sở của vị trí của mẫu dự đoán.

Khi chế độ dự đoán có hướng được lựa chọn, tham số mà biểu diễn chiều dự đoán (hoặc góc dự đoán) của chế độ dự đoán có hướng được lựa chọn có thể được xác định. Bảng 2 dưới đây biểu diễn tham số nội dự đoán có hướng của intraPredAng đối với mỗi chế độ nội dự đoán.

【Bảng 2】

PredModeIntra	1	2	3	4	5	6	7
IntraPredAng	-	32	26	21	17	13	9
PredModeIntra	8	9	10	11	12	13	14
IntraPredAng	5	2	0	-2	-5	-9	-13
PredModeIntra	15	16	17	18	19	20	21
IntraPredAng	-17	-21	-26	-32	-26	-21	-17
PredModeIntra	22	23	24	25	26	27	28
IntraPredAng	-13	-9	-5	-2	0	2	5
PredModeIntra	29	30	31	32	33	34	
IntraPredAng	9	13	17	21	26	32	

Bảng 2 biểu diễn tham số nội dự đoán có hướng của mỗi chế độ nội dự đoán trong đó chỉ số của nó là một trong số 2 đến 34 khi 35 chế độ nội dự đoán được xác định. Khi các chế độ nội dự đoán có hướng được xác định lớn hơn 33, tham số nội dự đoán có hướng của mỗi chế độ nội dự đoán có thể được thiết lập bằng cách phân chia Bảng 2.

Các mẫu tham chiếu trên cùng và các mẫu tham chiếu bên trái đối với khối hiện tại được sắp xếp theo dòng, và sau đó mẫu dự đoán có thể được thu nhận trên cơ sở của giá trị của tham số nội dự đoán có hướng. Theo đó, khi giá trị của tham số nội dự đoán có hướng là giá trị âm, các mẫu tham chiếu bên trái và các mẫu tham chiếu trên cùng có thể được sắp xếp theo dòng.

Các FIG.24 và FIG.25 lần lượt là các hình vẽ thể hiện các ví dụ về sắp xếp một chiều trong đó các mẫu tham chiếu được sắp xếp theo dòng.

FIG.24 là hình vẽ thể hiện sắp xếp một-chiều theo chiều dọc trong đó các mẫu tham chiếu được sắp xếp trong chiều dọc, và FIG.25 là hình vẽ thể hiện sắp xếp một-chiều theo chiều ngang trong đó các mẫu tham chiếu được sắp xếp trong chiều ngang. Các ví dụ của các FIG.24 và FIG.25 sẽ được mô tả trong giả thiết rằng 35 chế độ nội dự đoán được xác định.

Khi chỉ số chế độ nội dự đoán là bất kỳ một trong số 11 đến 18, sắp xếp một chiều theo chiều ngang có thể được áp dụng trong đó các mẫu tham chiếu

trên cùng được quay ngược chiều kim đồng hồ, và khi chỉ số chế độ nội dự đoán là bất kỳ một trong số 19 đến 25, sắp xếp một chiều theo chiều dọc có thể được áp dụng trong đó các mẫu tham chiếu bên trái được quay theo chiều kim đồng hồ. Khi sắp xếp các mẫu tham chiếu theo dòng, góc chế độ nội dự đoán có thể được xem xét.

Mẫu tham chiếu mà xác định tham số có thể được xác định trên cơ sở của tham số nội dự đoán có hướng. Mẫu tham chiếu mà xác định tham số có thể bao gồm chỉ số mẫu tham chiếu để chỉ rõ mẫu, và tham số hệ số trọng số để xác định hệ số trọng số được áp dụng tới mẫu tham chiếu.

Chỉ số mẫu tham chiếu, i_{Idx} , và tham số hệ số trọng số, i_{fact} , có thể được lần lượt thu được thông qua các phương trình 4 và 5 dưới đây.

【Công thức 4】

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

【Công thức 5】

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

Trong các phương trình 4 và 5, P_{ang} biểu diễn tham số nội dự đoán có hướng. Mẫu tham chiếu được chỉ rõ bởi chỉ số mẫu tham chiếu i_{Idx} tương ứng với điểm ảnh nguyên.

Để thu nhận mẫu dự đoán, ít nhất một mẫu tham chiếu có thể được chỉ rõ. Một cách chi tiết, theo độ nghiêng của chế độ dự đoán, vị trí của mẫu tham chiếu được sử dụng để thu nhận mẫu dự đoán có thể được chỉ rõ. Trong ví dụ của sáng chế, mẫu tham chiếu được sử dụng để thu nhận mẫu dự đoán có thể được chỉ rõ bằng cách sử dụng chỉ số mẫu tham chiếu của i_{Idx} .

Theo đó, khi độ nghiêng của chế độ nội dự đoán không được biểu diễn bởi một mẫu tham chiếu, mẫu dự đoán có thể được tạo ra bằng cách thực hiện việc nội suy trên nhiều mẫu tham chiếu. Trong ví dụ của sáng chế, khi độ nghiêng của chế độ nội dự đoán là giá trị giữa độ nghiêng giữa mẫu dự đoán và mẫu tham chiếu thứ nhất, và độ nghiêng giữa mẫu dự đoán và mẫu tham chiếu thứ hai, mẫu dự đoán có thể được thu nhận bằng cách thực hiện việc nội suy trên mẫu tham chiếu thứ nhất và mẫu tham chiếu thứ hai. Nói cách khác, khi đường tạo góc theo

góc nội dự đoán không đi qua mẫu tham chiếu có vị trí tại điểm ảnh nguyên, mẫu dự đoán có thể được thu nhận bằng cách thực hiện việc nội suy trên các mẫu tham chiếu có vị trí liền kề với bên trái và bên phải, hoặc trên cùng và dưới cùng của vị trí trong đó đường tạo góc đi qua.

Phương trình 6 dưới đây biểu diễn ví dụ về việc thu nhận mẫu dự đoán trên cơ sở của các mẫu tham chiếu.

【Công thức 6】

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

Trong phương trình 6, P biểu diễn mẫu dự đoán, và Ref_1D biểu diễn bất kỳ một trong số các mẫu tham chiếu mà được sắp xếp theo dòng. Theo đó, vị trí của mẫu tham chiếu có thể được xác định bởi vị trí (x, y) của mẫu dự đoán và chỉ số mẫu tham chiếu của iIdx.

Khi độ nghiêng của chế độ nội dự đoán có thể được biểu diễn bởi một mẫu tham chiếu, tham số hệ số trọng số của i_{fact} được thiết lập là 0. Do đó, phương trình 6 có thể được đơn giản hóa như phương trình 7 dưới đây.

【Công thức 7】

$$P(x,y)=Ref_1D(x+iIdx+1)$$

Việc nội dự đoán đối với khối hiện tại có thể được thực hiện trên cơ sở của các chế độ nội dự đoán. Trong ví dụ của sáng chế, chế độ nội dự đoán có thể được thu nhận đối với mỗi mẫu dự đoán, và mẫu dự đoán có thể được thu nhận trên cơ sở của chế độ nội dự đoán được gán tới mỗi mẫu dự đoán.

Ngoài ra, chế độ nội dự đoán có thể được thu nhận đối với mỗi vùng, việc nội dự đoán đối với mỗi vùng có thể được thực hiện trên cơ sở của chế độ nội dự đoán được gán tới mỗi vùng. Theo đó, vùng có thể bao gồm ít nhất một mẫu. Ít nhất một trong số kích cỡ và dạng của vùng có thể được xác định thích nghi trên cơ sở của ít nhất một trong số kích cỡ của khối hiện tại, dạng của khối hiện tại, và chế độ nội dự đoán đối với khối hiện tại. Ngoài ra, ít nhất một trong số kích cỡ và dạng của vùng có thể được xác định trước trong bộ mã hóa và bộ giải mã độc lập với kích cỡ hoặc dạng của khối hiện tại.

FIG.26 là hình vẽ thể hiện độ góc được tạo thành giữa các chế độ nội dự đoán có hướng và đường thẳng song song với trục x.

Đối với ví dụ được thể hiện trên FIG.26, các chế độ dự đoán có hướng có thể được biểu diễn giữa chiều đường chéo phía dưới-bên trái và chiều đường chéo trên cùng-bên phải. Việc mô tả độ góc giữa trục x và chế độ dự đoán có hướng, các chế độ dự đoán có hướng có thể được biểu diễn từ 45 độ (chiều đường chéo dưới cùng-bên trái) đến -135 độ (chiều đường chéo trên cùng-bên phải).

Khi khối hiện tại không phải hình vuông, trường hợp có thể được biểu diễn trong đó mẫu dự đoán được thu nhận bằng cách sử dụng, trong số các mẫu tham chiếu có vị trí tại đường tạo góc theo góc nội dự đoán, mẫu tham chiếu mà có vị trí xa hơn so với mẫu tham chiếu gần với mẫu dự đoán theo chế độ nội dự đoán đối với khối hiện tại.

FIG.27 là hình vẽ thể hiện khía cạnh của việc thu nhận mẫu dự đoán khi khối hiện tại không phải hình vuông.

Trong ví dụ của sáng chế, như ví dụ được thể hiện trên FIG.27 (a), giả thiết rằng khối hiện tại không phải hình vuông trong đó độ rộng lớn hơn độ cao, và chế độ nội dự đoán đối với khối hiện tại là chế độ nội dự đoán có hướng mà có góc từ 0 độ đến 45 độ. Trong trường hợp nêu trên, khi thu nhận mẫu dự đoán A xung quanh cột bên phải của khối hiện tại, trong số các mẫu tham chiếu có vị trí tại chế độ góc theo độ góc nêu trên, thay vì sử dụng mẫu tham chiếu trên cùng T gần với mẫu dự đoán, trường hợp có thể được biểu diễn trong đó mẫu tham chiếu bên trái L nằm cách xa mẫu dự đoán được sử dụng.

Trong ví dụ khác, như ví dụ được thể hiện trên FIG.27 (b), giả thiết rằng khối hiện tại không phải hình vuông trong đó độ cao lớn hơn độ rộng, và chế độ nội dự đoán đối với khối hiện tại là chế độ nội dự đoán có hướng từ -90 độ tới -135 độ. Trong các trường hợp nêu trên, khi thu nhận mẫu dự đoán A xung quanh hàng dưới cùng của khối hiện tại, trong số các mẫu tham chiếu có vị trí tại chế độ góc theo độ góc nêu trên, thay vì sử dụng mẫu tham chiếu bên trái L gần với mẫu dự đoán, trường hợp có thể được biểu diễn trong đó mẫu tham chiếu trên cùng T nằm cách xa mẫu dự đoán được sử dụng.

Để giải quyết vấn đề nêu trên, khi khối hiện tại không phải hình vuông, chế độ nội dự đoán đối với khối hiện tại có thể được thay thế bởi chế độ nội dự

đoán trong chiều ngược lại. Do đó, đối với khối không phải hình vuông, các chế độ dự đoán có hướng mà có các góc lớn hơn hoặc nhỏ hơn so với của các chế độ dự đoán có hướng được thể hiện trên FIG.23 có thể được sử dụng. Chế độ nội dự đoán có hướng nêu trên có thể được xác định là chế độ nội dự đoán góc rộng. Chế độ nội dự đoán góc rộng biểu diễn chế độ nội dự đoán có hướng mà không thuộc về phạm vi của 45 độ đến -135 độ.

FIG.28 là hình vẽ thể hiện các chế độ nội dự đoán góc rộng.

Trong ví dụ thể hiện trong FIG.28, các chế độ nội dự đoán mà có các chỉ số từ -1 đến -14 và các chế độ nội dự đoán mà có các chỉ số từ 67 đến 80 biểu diễn các chế độ nội dự đoán góc rộng.

Trong FIG.28, 14 các chế độ nội dự đoán góc rộng (từ -1 đến -14) mà góc lớn hơn 45 độ và 4 chế độ nội dự đoán góc rộng (từ 67 đến 80) có góc nhỏ hơn -135 độ được thể hiện. Tuy nhiên, số lượng chế độ nội dự đoán góc rộng nhiều hơn hoặc ít hơn có thể được xác định.

Khi chế độ nội dự đoán góc rộng được sử dụng, độ dài của các mẫu tham chiếu trên cùng có thể được thiết lập là $2W + 1$, và độ dài của các mẫu tham chiếu bên trái có thể được thiết lập là $2H + 1$.

Bằng cách sử dụng chế độ nội dự đoán góc rộng, mẫu A được thể hiện trên FIG.28 (a) có thể được dự đoán bằng cách sử dụng mẫu tham chiếu T, và mẫu A được thể hiện trên FIG.28 (b) có thể được dự đoán bởi mẫu tham chiếu L.

Ngoài các chế độ nội dự đoán truyền thống và N chế độ nội dự đoán góc rộng, tổng số $67 + N$ chế độ nội dự đoán có thể được sử dụng. Trong ví dụ của sáng chế, Bảng 3 biểu diễn tham số nội dự đoán có hướng đối với các chế độ nội dự đoán khi 20 chế độ nội dự đoán 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 không phải hình vuông, và chế độ nội dự đoán đối với khối hiện tại mà thu được trong S2502 thuộc về phạm vi biến đổi, chế độ nội dự đoán đối với khối hiện tại có thể được biến đổi thành chế độ nội dự đoán góc rộng. Phạm vi biến đổi có thể được xác định trên cơ sở của ít nhất một trong số kích cỡ, dạng, hoặc tỷ lệ của khối hiện tại. Theo đó, tỷ lệ này có thể biểu diễn tỷ lệ giữa độ rộng và độ cao của khối hiện tại.

Khi khối hiện tại không phải hình vuông trong đó độ rộng lớn hơn độ cao, phạm vi biến đổi có thể được thiết lập từ chỉ số chế độ nội dự đoán (ví dụ, 66) của chiều đường chéo trên cùng-bên phải thành (chỉ số chế độ nội dự đoán của chiều đường chéo trên cùng-bên phải – N). Theo đó, N có thể được xác định trên cơ sở của tỷ lệ của khối hiện tại. Khi chế độ nội dự đoán đối với khối hiện tại thuộc về phạm vi biến đổi, chế độ nội dự đoán có thể được biến đổi thành chế độ nội dự đoán góc rộng. Việc biến đổi có thể là trừ giá trị định trước từ chế độ nội dự đoán, và giá trị định trước có thể là tổng số (ví dụ, 67) chế độ nội dự đoán ngoại trừ chế độ nội dự đoán góc rộng.

Trong ví dụ nêu trên, chế độ nội dự đoán từ số 66 đến số 53 có thể được biến đổi lần lượt thành các chế độ nội dự đoán góc rộng từ số -1 đến số -14.

Khi khối hiện tại không phải hình vuông trong đó độ cao lớn hơn độ rộng,

phạm vi biến đổi có thể được thiết lập từ chỉ số chế độ nội dự đoán (ví dụ, 2) của chiều đường chéo dưới cùng-bên trái tới (chỉ số chế độ nội dự đoán của chiều đường chéo phía dưới-bên trái + M). Theo đó, M có thể được xác định trên cơ sở của tỷ lệ của khối hiện tại. Khi chế độ nội dự đoán đối với khối hiện tại thuộc về phạm vi biến đổi, chế độ nội dự đoán có thể được biến đổi thành chế độ nội dự đoán góc rộng. Việc biến đổi có thể là cộng giá trị định trước vào chế độ nội dự đoán, và giá trị định trước có thể là tổng số (ví dụ, 65) chế độ nội dự đoán có hướng ngoại trừ chế độ nội dự đoán góc rộng.

Trong ví dụ nêu trên, chế độ nội dự đoán từ số 2 đến số 15 có thể được biến đổi lần lượt thành các chế độ nội dự đoán góc rộng từ số 67 đến số 80.

Sau đây, các chế độ nội dự đoán thuộc về phạm vi biến đổi được gọi là các chế độ thay thế dự đoán trong góc rộng.

Phạm vi biến đổi có thể được xác định trên cơ sở của tỷ lệ của khối hiện tại. Trong ví dụ của sáng chế, các bảng 4 và 5 lần lượt thể hiện phạm vi biến đổi của trường hợp trong đó 35 chế độ nội dự đoán ngoại trừ chế độ nội dự đoán góc rộng được xác định, và trường hợp trong đó 67 chế độ nội dự đoán ngoại trừ chế độ nội dự đoán góc rộng được xác định.

【Bảng 4】

Điều kiện	Chế độ nội dự đoán được thay thế
$W/H = 2$	Các chế độ 2, 3, 4
$W/H > 2$	Các chế độ 2, 3, 4, 5, 6
$W/H = 1$	Không
$H/W = 1/2$	Các chế độ 32, 33, 34
$H/W < 1/2$	Các chế độ 30, 31, 32, 33, 34

【Bảng 5】

Điều kiện	Chế độ nội dự đoán được thay thế
$W/H = 2$	các chế độ 2, 3, 4, 5, 6, 7
$W/H > 2$	Các chế độ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
$W/H = 1$	Không
$H/W = 1/2$	các chế độ 61, 62, 63, 64, 65, 66
$H/W < 1/2$	Các chế độ 57, 58, 59, 60, 61, 62, 63, 64, 65, 66

Đối với các ví dụ được thể hiện trong các bảng 4 và 5, số lượng chế độ thay thế dự đoán trong góc rộng được chứa trong phạm vi biến đổi có thể thay đổi theo tỷ lệ của khối hiện tại.

Tỷ lệ của khối hiện tại có thể còn được phân chia thêm để thiết lập phạm vi biến đổi như được thể hiện trong Bảng 6 dưới đây.

【Bảng 6】

Điều kiện	Chế độ nội dự đoán được thay thế
$W/H = 16$	Các chế độ 12, 13, 14, 15
$W/H = 8$	Các chế độ 12, 13
$W/H = 4$	Các chế độ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
$H/W = 2$	các chế độ 2, 3, 4, 5, 6, 7
$H/W = 1$	Không
$W/H = 1/2$	các chế độ 61, 62, 63, 64, 65, 66
$W/H = 1/4$	Các chế độ 57, 58, 59, 60, 61, 62, 63, 64, 65, 66
$W/H = 1/8$	Các chế độ 55, 56
$H/W = 1/16$	Các chế độ 53, 54, 55, 56

Khi dòng mẫu tham chiếu không lân cận được xác định là dòng mẫu tham chiếu đối với khối hiện tại hoặc khi phương pháp mã hóa nội dự đoán đa dòng để lựa chọn một trong số các dòng mẫu tham chiếu được sử dụng, phương pháp dự đoán có thể được cấu hình để không sử dụng chế độ nội dự đoán góc rộng. Tức là, mặc dù khối hiện tại có dạng không phải hình vuông, và chế độ nội dự đoán đối với khối hiện tại thuộc về phạm vi biến đổi, chế độ nội dự đoán đối với khối hiện tại có thể không được biến đổi thành chế độ nội dự đoán góc rộng.

Ngoài ra, khi chế độ nội dự đoán đối với khối hiện tại được xác định là chế độ nội dự đoán góc rộng, phương pháp dự đoán có thể được cấu hình sao cho các dòng mẫu tham chiếu không lân cận là không khả dụng như là dòng mẫu tham chiếu đối với khối hiện tại hoặc có thể được cấu hình để không sử dụng phương pháp mã hóa nội dự đoán đa dòng để lựa chọn một trong số các dòng mẫu tham chiếu. Khi phương pháp mã hóa nội dự đoán đa dòng không được sử dụng, dòng mẫu tham chiếu lân cận có thể được xác định như là dòng mẫu tham chiếu đối với khối hiện tại.

Khi chế độ nội dự đoán góc rộng không được sử dụng, mỗi refW và refH có thể được thiết lập là tổng của $n\text{TbW}$ và $n\text{TbH}$. Do đó, dòng mẫu tham chiếu không lân cận nằm cách khối hiện tại bởi giá trị i có thể bao gồm $(n\text{TbW} + n\text{TbH} + \text{offsetX}[i])$ các mẫu tham chiếu trên cùng và $(n\text{TbW} + n\text{TbH} + \text{offsetY}[i])$ các mẫu tham chiếu bên trái ngoại trừ mẫu tham chiếu bên trái-trên cùng. Tức là, dòng mẫu tham chiếu không lân cận nằm cách khối hiện tại bởi giá trị i có thể bao gồm $(2n\text{TbW} + 2n\text{TbH} + \text{offsetX}[i] + \text{offsetY}[i] + 1)$ mẫu tham chiếu. Ví dụ, khi giá trị của whRatio lớn hơn 1, giá trị của offsetX có thể được thiết lập để lớn hơn giá trị của offsetY . Trong một ví dụ, khi giá trị của offsetX có thể được thiết lập là 1, và giá trị của offsetY có thể được thiết lập là 0. Ngược lại, khi giá trị của whRatio nhỏ hơn 1, giá trị của offsetY có thể được thiết lập lớn hơn giá trị của offsetX . Trong một ví dụ, giá trị của offsetX có thể được thiết lập là 0, và giá trị của offsetY có thể được thiết lập là 1.

Do các chế độ nội dự đoán góc rộng được sử dụng ngoài các chế độ nội dự đoán truyền thống, tài nguyên để mã hóa các chế độ nội dự đoán góc rộng có thể được tăng lên, và do đó hiệu quả mã hóa có thể bị làm giảm. Do đó, thay vì mã hóa chế độ nội dự đoán góc rộng như thực tại, chế độ nội dự đoán được thay thế đối với các chế độ nội dự đoán góc rộng được mã hóa để cải thiện hiệu quả mã hóa.

Trong ví dụ của sáng chế, khi khối hiện tại được mã hóa bằng cách sử dụng chế độ nội dự đoán góc rộng của số 67, số 2 mà là chế độ nội dự đoán thay thế góc rộng của số 67 có thể được mã hóa như là chế độ nội dự đoán đối với khối hiện tại. Ngoài ra, khi khối hiện tại được mã hóa bằng cách sử dụng chế độ nội dự đoán góc rộng của số -1, số 66 mà là chế độ nội dự đoán thay thế góc rộng của số -1 có thể được mã hóa như là chế độ nội dự đoán đối với khối hiện tại.

Bộ giải mã có thể giải mã chế độ nội dự đoán đối với khối hiện tại, và xác định rằng chế độ nội dự đoán được giải mã có thuộc về phạm vi biến đổi hay không. Khi chế độ nội dự đoán được giải mã là chế độ nội dự đoán thay thế góc rộng, chế độ nội dự đoán có thể được biến đổi thành chế độ nội dự đoán góc rộng.

Ngoài ra, khi khối hiện tại được mã hóa thông qua chế độ nội dự đoán góc rộng, chế độ nội dự đoán góc rộng có thể được mã hóa như thực tại.

Việc mã hóa chế độ nội dự đoán có thể được thực hiện dựa trên danh sách MPM được mô tả nêu trên. Cụ thể, khi khối lân cận được mã hóa trong chế độ nội

dự đoán góc rộng, MPM có thể được thiết lập dựa trên chế độ nội dự đoán thay thế góc rộng tương ứng với chế độ nội dự đoán góc rộng.

Ảnh dư có thể được thu nhận bằng cách trừ ảnh dư đoán từ ảnh gốc. Theo đó, khi ảnh dư được chuyển đổi thành miền tần số, ngay cả mặc dù các thành phần tần số cao được loại bỏ khỏi các thành phần tần số, chất lượng ảnh mục tiêu của ảnh không giảm đi đáng kể. Do đó, khi các giá trị của các thành phần tần số cao được biến đổi thành các giá trị nhỏ, hoặc khi các giá trị của các thành phần tần số cao được thiết lập là 0, hiệu quả nén có thể được tăng lên mà không gây ra độ méo trực quan lớn. Phản ánh đặc tính nêu trên, việc biến đổi có thể được thực hiện trên khối hiện tại để phân giải ảnh dư thành các thành phần tần số hai chiều. Việc biến đổi có thể được thực hiện bằng cách sử dụng các phương pháp biến đổi như DCT (discrete cosine transform-Biến đổi côsin rời rạc), DST (discrete sine transform-Biến đổi sin rời rạc), v.v.

DCT là để phân giải (hoặc biến đổi) ảnh dư thành các thành phần tần số hai chiều nhờ sử dụng biến đổi côsin, và DST là để phân giải (hoặc biến đổi) ảnh dư thành các thành phần tần số hai chiều nhờ sử dụng biến đổi sin. Theo kết quả biến đổi ảnh dư, các thành phần tần số có thể được biểu diễn như là ảnh gốc. Theo ví dụ của sáng chế, khi biến đổi DCT được thực hiện đối với khối có kích cỡ $N \times N$, $N/2$ thành phần mẫu cơ sở có thể được thu nhận. Kích cỡ của mỗi thành phần mẫu cơ sở được chứa trong khối có kích cỡ $N \times N$ có thể được thu nhận thông qua việc biến đổi. Theo phương pháp biến đổi được sử dụng, kích cỡ của thành phần mẫu cơ sở có thể được gọi là hệ số DCT hoặc hệ số DST.

Phương pháp biến đổi DCT được sử dụng chủ yếu để biến đổi ảnh mà nhiều thành phần tần số thấp khác 0 được phân phối. Phương pháp biến đổi DST được sử dụng chủ yếu cho ảnh mà nhiều thành phần tần số cao được phân phối.

Cũng có thể biến đổi ảnh dư bằng cách sử dụng phương pháp biến đổi khác ngoài DCT hoặc DST.

Sau đây, việc biến đổi ảnh dư thành các thành phần tần số hai chiều được gọi là biến đổi ảnh hai chiều. Ngoài ra, kích cỡ của các thành phần mẫu cơ sở thu được bởi việc biến đổi được gọi là hệ số biến đổi. Trong ví dụ của sáng chế, hệ số biến đổi có thể có nghĩa là hệ số DCT hoặc hệ số DST. Khi cả biến đổi thứ nhất và biến đổi thứ hai được mô tả sau đây được áp dụng, hệ số biến đổi có thể có nghĩa là thành phần mẫu cơ bản được tạo ra bởi kết quả của biến đổi thứ hai.

Ngoài ra, mẫu dư mà việc bỏ qua biến đổi được áp dụng cũng được gọi là hệ số biến đổi.

Phương pháp biến đổi có thể được xác định trong đơn vị của khối. Phương pháp biến đổi có thể được xác định dựa trên ít nhất một trong số chế độ mã hóa dự đoán của khối hiện tại, kích cỡ của khối hiện tại hoặc dạng của khối hiện tại. Trong ví dụ của sáng chế, khi khối hiện tại được mã hóa bởi chế độ nội dự đoán và kích cỡ của khối hiện tại nhỏ hơn $N \times N$, việc biến đổi có thể được thực hiện bằng cách sử dụng phương pháp biến đổi DST. Mặt khác, khi điều kiện nêu trên không được thỏa mãn, việc biến đổi có thể được thực hiện bằng cách sử dụng phương pháp biến đổi DCT.

Việc biến đổi ảnh hai chiều có thể không được thực hiện đối với một vài khối của ảnh dư. Việc không thực hiện biến đổi ảnh hai chiều có thể được gọi là bỏ qua biến đổi. Việc bỏ qua biến đổi biểu diễn rằng biến đổi thứ nhất và biến đổi thứ hai không được áp dụng tới khối hiện tại. Khi việc bỏ qua biến đổi được áp dụng, việc lượng tử hóa có thể được áp dụng tới các giá trị dư mà việc biến đổi không được thực hiện.

Việc bỏ qua biến đổi có được cho phép hay không đối với khối hiện tại có thể được xác định dựa trên ít nhất một trong số kích cỡ hoặc dạng của khối hiện tại. Trong ví dụ của sáng chế, chỉ khi kích cỡ của khối hiện tại nhỏ hơn giá trị ngưỡng, việc bỏ qua biến đổi có thể được áp dụng. Giá trị ngưỡng liên quan đến ít nhất một trong số độ rộng, độ cao hoặc số lượng mẫu của khối hiện tại, và có thể được xác định là 32×32 , v.v. Ngoài ra, việc bỏ qua biến đổi có thể được cho phép chỉ đối với khối vuông. Trong ví dụ của sáng chế, việc bỏ qua biến đổi có thể được cho phép đối với khối vuông có kích cỡ 32×32 , 16×16 , 8×8 hoặc 4×4 . Ngoài ra, chỉ khi phương pháp nội mã hóa phân vùng con không được sử dụng, việc bỏ qua biến đổi có thể được cho phép.

Ngoài ra, khi phương pháp nội mã hóa phân vùng con được áp dụng tới khối hiện tại, có thể được xác định đối với mỗi khối con rằng có áp dụng việc bỏ qua biến đổi hay không.

FIG.29 là sơ đồ thể hiện ví dụ trong đó việc bỏ qua biến đổi có được thực hiện hay không được xác định theo khối con.

Việc bỏ qua biến đổi có thể được áp dụng chỉ đối với một phần của các

khối con. Trong ví dụ của sáng chế, như trong ví dụ được thể hiện trên FIG.29, có thể được thiết lập để áp dụng việc bỏ qua biến đổi tới khối con tại vị trí trên cùng của khối hiện tại và không áp dụng việc bỏ qua biến đổi tới khối con tại vị trí dưới cùng.

Loại biến đổi của khối con mà việc bỏ qua biến đổi không được cho phép có thể được xác định dựa trên thông tin được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, loại biến đổi có thể được xác định dựa trên `tu_mts_idx` mà sẽ được mô tả sau đây.

Ngoài ra, loại biến đổi của khối con có thể được xác định dựa trên kích cỡ của khối con. Trong ví dụ của sáng chế, loại biến đổi chiều ngang có thể được xác định dựa trên việc độ rộng của khối con có lớn hơn hoặc bằng và/hoặc nhỏ hơn hoặc bằng giá trị ngưỡng hay không, và loại biến đổi chiều dọc có thể được xác định dựa trên việc độ cao của khối con có lớn hơn hoặc bằng và/hoặc nhỏ hơn hoặc bằng giá trị ngưỡng hay không.

Sau khi thực hiện việc biến đổi trên khối hiện tại bằng cách sử dụng DCT hoặc DST, việc biến đổi có thể được thực hiện lần nữa trên khối hiện tại được biến đổi. Theo đó, việc biến đổi dựa trên DCT hoặc DST có thể được xác định là biến đổi thứ nhất, và thực hiện việc biến đổi lần nữa trên khối mà biến đổi thứ nhất được áp dụng tới có thể được xác định là biến đổi thứ hai.

Biến đổi thứ nhất có thể được thực hiện bằng cách sử dụng bất kỳ một trong số các ứng viên lõi biến đổi. Trong ví dụ của sáng chế, biến đổi thứ nhất có thể được thực hiện bằng cách sử dụng bất kỳ một trong số DCT2, DCT8, hoặc DST7.

Các lõi biến đổi khác nhau có thể được sử dụng đối với chiều ngang và chiều dọc. Thông tin mà biểu diễn kết hợp của lõi biến đổi của chiều ngang và lõi biến đổi của chiều dọc có thể được báo hiệu trong dòng bit.

Đơn vị xử lý của biến đổi thứ nhất có thể khác với biến đổi thứ hai. Trong ví dụ của sáng chế, biến đổi thứ nhất có thể được thực hiện trên khối 8x8, và biến đổi thứ hai có thể được thực hiện trên khối con có kích cỡ 4x4 trong khối 8x8 được biến đổi. Ngoài ra, biến đổi thứ hai có thể được thực hiện đối với các hệ số biến đổi mà thuộc về 3 khối con có kích cỡ 4x4. 3 khối con có thể bao gồm khối con có vị trí tại trên cùng-bên trái của khối hiện tại, khối con lân cận bên phải của

khối con và khối con lân cận dưới cùng của khối con. Ngoài ra, biến đổi thứ hai có thể được thực hiện đối với khối có kích cỡ 8×8 .

Cũng có thể rằng các hệ số biến đổi trong vùng còn lại mà trên đó biến đổi thứ hai không được thực hiện có thể được thiết lập là 0.

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

Thông tin mà biểu diễn rằng có thực hiện biến đổi thứ hai hay không có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, cờ mà biểu diễn rằng biến đổi thứ hai có được thực hiện hay không, hoặc thông tin chỉ số mà chỉ rõ rằng biến đổi thứ hai có được thực hiện hay không và hạt nhân biến đổi được sử dụng cho biến đổi thứ hai có thể được báo hiệu. Trong ví dụ của sáng chế, khi thông tin chỉ số là 0, điều này biểu diễn rằng biến đổi thứ hai không được thực hiện đối với khối hiện tại. Mặt khác, khi thông tin chỉ số lớn hơn 0, hạt nhân biến đổi đối với biến đổi thứ hai có thể được xác định bởi thông tin chỉ số.

Ngoài ra, việc có thực hiện biến đổi thứ hai hay không có thể được xác định dựa trên lỗi biến đổi chiều ngang và lỗi biến đổi chiều dọc có đồng nhất với nhau hay không. Trong một ví dụ, biến đổi thứ hai có thể được thực hiện chỉ khi lỗi biến đổi chiều ngang và lỗi biến đổi chiều dọc là đồng nhất với nhau. Ngoài ra, biến đổi thứ hai có thể được thực hiện chỉ khi lỗi biến đổi chiều ngang và lỗi biến đổi chiều dọc là khác nhau.

Ngoài ra, biến đổi thứ hai có thể được cho phép chỉ khi lỗi biến đổi định trước được sử dụng đối với biến đổi theo chiều ngang và biến đổi theo chiều dọc. Trong một ví dụ, khi lỗi biến đổi DCT2 được sử dụng cho biến đổi trong chiều ngang và biến đổi trong chiều dọc, biến đổi thứ hai có thể được cho phép. Ngoài ra, khi phương pháp nội mã hóa phân vùng con được áp dụng tới khối hiện tại, biến đổi thứ hai có thể được cho phép chỉ khi lỗi biến đổi DCT2 được sử dụng cho việc biến đổi trong chiều ngang và biến đổi trong chiều dọc.

Ngoài ra, có thể được xác định rằng việc có thực hiện hay không biến đổi thứ hai dựa trên số lượng các hệ số biến đổi không phải 0 của khối hiện tại. Trong một ví dụ, khi số lượng của các hệ số biến đổi không phải 0 của khối hiện tại nhỏ hơn hoặc bằng ngưỡng, phương pháp dự đoán có thể được cấu hình để không sử

dụng biến đổi thứ hai. Khi số lượng của các hệ số biến đổi không phải 0 của khối hiện tại lớn hơn ngưỡng, phương pháp dự đoán có thể có được cấu hình để sử dụng biến đổi thứ hai. Miễn là khối hiện tại được mã hóa nhờ sử dụng việc nội dự đoán, phương pháp dự đoán có thể được cấu hình để sử dụng biến đổi thứ hai.

Ngoài ra, việc biến đổi thứ hai được thực hiện hay không có thể được xác định dựa trên vị trí của hệ số biến đổi cuối cùng không phải 0 của khối hiện tại. Trong ví dụ của sáng chế, khi ít nhất một trong số tọa độ trục x hoặc tọa độ trục y của hệ số biến đổi không phải 0 cuối cùng của khối hiện tại lớn hơn giá trị ngưỡng, hoặc khi ít nhất một trong số tọa độ trục x hoặc tọa độ trục y của khối con mà hệ số biến đổi không phải 0 cuối cùng của khối hiện tại thuộc về đó lớn hơn giá trị ngưỡng, biến đổi thứ hai có thể không được thực hiện. Trong trường hợp này, giá trị ngưỡng có thể được xác định trước trong thiết bị mã hóa và thiết bị giải mã. Ngoài ra, giá trị ngưỡng có thể được xác định dựa trên kích cỡ hoặc dạng của khối hiện tại.

Ngoài ra, khi chỉ hệ số biến đổi của thành phần DC tồn tại trong khối hiện tại, có thể được thiết lập không thực hiện biến đổi thứ hai. Trong trường hợp này, thành phần DC biểu diễn hệ số biến đổi tại vị trí trên cùng-bên trái trong khối hiện tại.

Ngoài ra, khi việc nội dự đoán dựa trên ma trận được áp dụng tới khối hiện tại, có thể được thiết lập không thực hiện biến đổi thứ hai.

Thông tin mà biểu diễn loại biến đổi của khối hiện tại có thể được báo hiệu trong dòng bit. Thông tin này có thể là thông tin chỉ số, `tu_mts_idx`, mà biểu diễn một trong số các kết hợp của loại biến đổi đối với chiều ngang và loại biến đổi đối với chiều dọc.

Dựa trên các ứng viên loại biến đổi được chỉ rõ bởi thông tin chỉ số, `tu_mts_idx`, lỗi biến đổi đối với chiều dọc và lỗi biến đổi đối với chiều ngang có thể được xác định. Bảng 7 biểu diễn các kết hợp loại biến đổi theo `tu_mts_idx`.

【Bảng 7】

tu_mts_idx	Loại biến đổi	
	Ngang	Dọc
0	DCT-II	DCT-II
1	DST-VII	DST-VII

tu_mts_idx	Loại biến đổi	
	Ngang	Dọc
2	DCT-VIII	DST-VII
3	DST-VII	DCT-VIII
4	DCT-VIII	DCT-VIII

Loại biến đổi có thể được xác định như là một trong số DCT2, DST7 hoặc DCT8. Ngoài ra, việc bỏ qua biến đổi có thể được chèn trong ứng viên loại biến đổi.

Khi bảng 7 được sử dụng, DCT2 có thể được áp dụng trong chiều ngang và trong chiều dọc khi tu_mts_idx là 0. Khi tu_mts_idx là 2, DCT8 có thể được áp dụng trong chiều ngang và DCT7 có thể được áp dụng trong chiều dọc.

Khi phương pháp nội mã hóa phân vùng con được áp dụng, lỗi biến đổi của khối con có thể được xác định một cách độc lập. Trong ví dụ của sáng chế, thông tin để chỉ rõ ứng viên kết hợp loại biến đổi có thể được mã hóa và được báo hiệu theo khối con. Do đó, lỗi biến đổi giữa các khối con có thể khác nhau.

Ngoài ra, các khối con có thể sử dụng cùng loại biến đổi. Trong trường hợp này, tu_mts_idx mà chỉ rõ ứng viên kết hợp loại biến đổi có thể được báo hiệu chỉ đối với khối con thứ nhất. Ngoài ra, tu_mts_idx có thể được báo hiệu trong mức khối mã hóa và loại biến đổi của các khối con có thể được xác định bằng cách viện dẫn tới tu_mts_idx được báo hiệu trong mức khối mã hóa. Ngoài ra, loại biến đổi có thể được xác định dựa trên ít nhất một trong số kích cỡ, dạng hoặc chế độ nội dự đoán của một trong số các khối con và loại biến đổi được xác định có thể được thiết lập để được sử dụng cho tất cả khối con.

FIG.30 là sơ đồ thể hiện ví dụ trong đó các khối con sử dụng cùng loại biến đổi.

Khi khối mã hóa được phân chia trong chiều ngang, loại biến đổi của khối con tại vị trí trên cùng của khối mã hóa (Sub-CU0) có thể được thiết lập tương tự như của khối con tại vị trí dưới cùng (Sub-CU1). Trong ví dụ của sáng chế, như trong ví dụ được thể hiện trong FIG.30 (a), khi loại biến đổi chiều ngang và loại biến đổi chiều dọc được xác định dựa trên tu_mts_idx được báo hiệu đối với khối con trên cùng, loại biến đổi được xác định có thể cũng được áp dụng tới khối con dưới cùng.

Khi khối mã hóa được phân chia trong chiều dọc, loại biến đổi của khối con tại vị trí bên trái của khối mã hóa (Sub-CU0) có thể được thiết lập tương tự như của khối con tại vị trí bên phải (Sub-CU1). Trong ví dụ của sáng chế, như trong ví dụ được thể hiện trong FIG.30 (b), khi loại biến đổi chiều ngang và loại biến đổi chiều dọc xác định dựa trên `tu_mts_idx` được báo hiệu cho khối con bên trái, loại biến đổi được xác định cũng có thể được áp dụng tới khối con bên phải.

Việc thông tin chỉ số có được mã hóa hay không có thể được xác định dựa trên ít nhất một trong số kích cỡ hoặc dạng của khối hiện tại, số lượng hệ số không phải 0, việc biến đổi thứ hai có được thực hiện hay không hoặc việc phương pháp nội mã hóa phân vùng con có được áp dụng hay không. Trong ví dụ của sáng chế, khi phương pháp nội mã hóa phân vùng con được áp dụng tới khối hiện tại, hoặc khi số lượng hệ số không phải 0 bằng hoặc nhỏ hơn giá trị ngưỡng, báo hiệu của thông tin chỉ số có thể được bỏ qua. Khi báo hiệu của thông tin chỉ số được bỏ qua, loại biến đổi mặc định có thể được áp dụng tới khối hiện tại.

Loại biến đổi mặc định có thể bao gồm ít nhất một trong số DCT2 hoặc DST7. Khi có các loại biến đổi mặc định, một trong số các loại biến đổi mặc định có thể được lựa chọn bằng cách xem xét ít nhất một trong số kích cỡ, dạng hoặc chế độ nội dự đoán của khối hiện tại, biến đổi thứ hai có được thực hiện hay không hoặc phương pháp nội mã hóa phân vùng con có được áp dụng hay không. Trong ví dụ của sáng chế, một trong số các loại biến đổi có thể được xác định như là loại biến đổi chiều ngang dựa trên việc độ rộng của khối hiện tại có nằm trong phạm vi được thiết lập trước hay không, và một trong số các loại biến đổi có thể được xác định như là loại biến đổi chiều dọc dựa trên việc độ cao của khối hiện tại có nằm trong phạm vi được thiết lập trước hay không. Ngoài ra, chế độ mặc định có thể được xác định khác nhau theo kích cỡ, dạng hoặc chế độ nội dự đoán của khối hiện tại hoặc việc biến đổi thứ hai có được thực hiện hay không.

Ngoài ra, khi chỉ hệ số biến đổi của thành phần DC tồn tại trong khối hiện tại, loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được thiết lập là loại biến đổi mặc định. Trong ví dụ của sáng chế, khi chỉ hệ số biến đổi của thành phần DC tồn tại trong khối hiện tại, loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được thiết lập là DCT2.

Giá trị ngưỡng có thể được xác định dựa trên kích cỡ hoặc dạng của khối hiện tại. Trong ví dụ của sáng chế, khi kích cỡ của khối hiện tại bằng hoặc nhỏ

hơn 32×32 , giá trị ngưỡng có thể được thiết lập là 2, và khi khối hiện tại lớn hơn 32×32 (ví dụ, khi khối hiện tại là khối mã hóa có kích cỡ 32×64 hoặc 64×32), giá trị ngưỡng có thể được thiết lập là 4.

Các bảng tra cứu có thể được lưu trữ trước trong thiết bị mã hóa/thiết bị giải mã. Ít nhất một trong số giá trị chỉ số được gán tới các ứng viên kết hợp loại biến đổi, loại của các ứng viên kết hợp loại biến đổi hoặc số lượng các ứng viên kết hợp loại biến đổi có thể khác nhau đối với mỗi bảng tra cứu.

Dựa trên ít nhất một trong số kích cỡ, dạng hoặc chế độ nội dự đoán của khối hiện tại, việc biến đổi thứ hai có được áp dụng hay không, hoặc việc bỏ qua biến đổi có được áp dụng tới khối lân cận hay không, bảng tra cứu đối với khối hiện tại có thể được lựa chọn.

Trong ví dụ của sáng chế, khi kích cỡ của khối hiện tại nhỏ hơn hoặc bằng 4×4 , hoặc khi khối hiện tại được mã hóa bằng cách dự đoán liên đới, bảng tra cứu thứ nhất có thể được sử dụng và khi kích cỡ của khối hiện tại lớn hơn 4×4 , hoặc khi khối hiện tại được mã hóa bằng cách nội dự đoán, bảng tra cứu thứ hai có thể được sử dụng.

Ngoài ra, thông tin mà chỉ báo một trong số các bảng tra cứu có thể được báo hiệu trong dòng bit. Thiết bị giải mã có thể lựa chọn bảng tra cứu đối với khối hiện tại dựa trên thông tin này.

Trong ví dụ khác, chỉ số được gán tới ứng viên kết hợp loại biến đổi có thể được xác định thích nghi, dựa trên ít nhất một trong số kích cỡ, dạng, chế độ mã hóa dự đoán hoặc chế độ nội dự đoán của khối hiện tại, biến đổi thứ hai có được áp dụng không, hoặc việc bỏ qua biến đổi có được áp dụng tới khối lân cận hay không. Trong ví dụ của sáng chế, chỉ số được gán tới việc bỏ qua biến đổi khi kích cỡ của khối hiện tại là 4×4 có thể nhỏ hơn chỉ số được gán tới việc bỏ qua biến đổi khi kích cỡ của khối hiện tại lớn hơn 4×4 . Cụ thể, khi kích cỡ của khối hiện tại là 4×4 , chỉ số 0 có thể được gán tới việc bỏ qua biến đổi và khi khối hiện tại lớn hơn 4×4 và nhỏ hơn hoặc bằng 16×16 , chỉ số lớn hơn 0 (ví dụ, chỉ số 1) có thể được gán tới việc bỏ qua biến đổi. Khi khối hiện tại lớn hơn 16×16 , giá trị cực đại (ví dụ, 5) có thể được gán tới chỉ số của việc bỏ qua biến đổi.

Ngoài ra, khi khối hiện tại được mã hóa bằng cách dự đoán liên đới, chỉ số 0 có thể được gán tới việc bỏ qua biến đổi. Khi khối hiện tại được mã hóa bằng

cách nội dự đoán, chỉ số lớn hơn 0 (ví dụ, chỉ số 1) có thể được gán tới việc bỏ qua biến đổi.

Ngoài ra, khi khối hiện tại là khối có kích cỡ 4x4 được mã hóa bằng cách dự đoán liên đới, chỉ số 0 có thể được gán tới việc bỏ qua biến đổi. Mặt khác, khi khối hiện tại không được mã hóa bằng cách dự đoán liên đới, hoặc khi khối hiện tại lớn hơn 4x4, chỉ số lớn hơn 0 (ví dụ, chỉ số 1) có thể được gán tới việc bỏ qua biến đổi.

Cũng có thể sử dụng các ứng viên kết hợp loại biến đổi khác với các ứng viên kết hợp loại biến đổi được đánh số trong bảng 7. Trong ví dụ của sáng chế, ứng viên kết hợp loại biến đổi mà bao gồm việc bỏ qua biến đổi được áp dụng tới một trong số biến đổi chiều ngang hoặc biến đổi chiều dọc và lỗi biến đổi như DCT2, DCT8 hoặc DST7, v.v được áp dụng tới biến đổi khác có thể được sử dụng. Trong trường hợp này, việc bỏ qua biến đổi được sử dụng như là ứng viên loại biến đổi cho chiều ngang hay chiều dọc có thể được xác định dựa trên ít nhất một trong số kích cỡ (ví dụ, độ rộng và/hoặc độ cao), dạng, chế độ mã hóa dự đoán hoặc chế độ nội dự đoán của khối hiện tại.

Thông tin mà chỉ báo rằng thông tin chỉ số để xác định loại biến đổi của khối hiện tại có được báo hiệu rõ ràng hay không có thể được báo hiệu trong dòng bit. Trong ví dụ của sáng chế, `sps_explicit_intra_mts_flag`, thông tin mà biểu diễn rằng việc xác định loại biến đổi rõ ràng có được cho phép đối với khối được mã hóa bằng cách nội dự đoán hay không, và/hoặc `sps_explicit_inter_mts_flag`, thông tin mà biểu diễn rằng việc xác định loại biến đổi rõ ràng có được cho phép đối với khối được mã hóa bằng cách dự đoán liên đới hay không, có thể được báo hiệu tại mức chuỗi.

Khi việc xác định loại biến đổi rõ ràng được cho phép, loại biến đổi của khối hiện tại có thể được xác định dựa trên thông tin chỉ số, `tu_mts_idx`, được báo hiệu trong dòng bit. Mặt khác, khi việc xác định loại biến đổi rõ ràng không được cho phép, loại biến đổi có thể được xác định dựa trên ít nhất một trong số kích cỡ hoặc dạng của khối hiện tại, việc có cho phép thực hiện biến đổi hay không trong đơn vị của khối con, vị trí của khối con bao gồm hệ số biến đổi không phải 0, việc biến đổi thứ hai có được thực hiện hay không, hoặc việc phương pháp nội mã hóa phân vùng con có được áp dụng hay không. Trong ví dụ của sáng chế, loại biến đổi chiều ngang của khối hiện tại có thể được xác định dựa trên độ rộng của khối

hiện tại và loại biến đổi chiều dọc của khối hiện tại có thể được xác định dựa trên độ cao của khối hiện tại. Ví dụ, khi độ rộng của khối hiện tại nhỏ hơn 4 hoặc lớn hơn 16, loại biến đổi chiều ngang có thể được xác định là DCT2. Nếu không phải, loại biến đổi chiều ngang có thể được xác định là DST7. Khi độ cao của khối hiện tại nhỏ hơn 4 hoặc lớn hơn 16, loại biến đổi chiều dọc có thể được xác định là DCT2. Nếu không phải, loại biến đổi chiều dọc có thể được xác định là DST7. Trong trường hợp này, giá trị ngưỡng mà cần được so sánh với độ rộng và độ cao có thể được xác định dựa trên ít nhất một trong số kích cỡ, dạng hoặc chế độ nội dự đoán của khối hiện tại để xác định loại biến đổi chiều ngang và loại biến đổi chiều dọc.

Ngoài ra, khi khối hiện tại có dạng hình vuông mà độ cao và độ rộng là giống nhau, loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được thiết lập giống nhau, tuy nhiên khi khối hiện tại có dạng không phải hình vuông mà độ cao và độ rộng khác nhau, loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được thiết lập khác nhau. Trong ví dụ của sáng chế, khi độ rộng của khối hiện tại lớn hơn độ cao, loại biến đổi chiều ngang có thể được xác định là DST7 và loại biến đổi chiều dọc có thể được xác định là DCT2. Khi độ cao của khối hiện tại lớn hơn độ rộng, loại biến đổi chiều dọc có thể được xác định là DST7 và loại biến đổi chiều ngang có thể được xác định là DCT2.

Số lượng và/hoặc loại của các ứng viên loại biến đổi hoặc số lượng và/hoặc loại của các ứng viên kết hợp loại biến đổi có thể khác nhau tùy theo vào việc xác định loại biến đổi rõ ràng có được cho phép hay không. Trong một ví dụ, khi việc xác định loại biến đổi rõ ràng được cho phép, DCT2, DST7 và DCT8 có thể được sử dụng là các ứng viên loại biến đổi. Do đó, mỗi loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được thiết lập là DCT2, DST8 hoặc DCT8. Khi việc xác định loại biến đổi rõ ràng không được cho phép, chỉ DCT2 và DST7 có thể được sử dụng như là ứng viên loại biến đổi. Do đó, mỗi loại biến đổi chiều ngang và loại biến đổi chiều dọc có thể được xác định là DCT2 hoặc DST7.

Bộ giải mã có thể thực hiện việc biến đổi ngược (biến đổi ngược thứ hai) tới biến đổi thứ hai và có thể thực hiện biến đổi ngược (biến đổi ngược thứ nhất) tới biến đổi thứ nhất thu được biến đổi ngược thứ hai. Theo kết quả thực hiện việc biến đổi ngược thứ hai và việc biến đổi ngược thứ nhất, các tín hiệu dư đối với khối hiện tại có thể được thu nhận.

Khi việc biến đổi và lượng tử hóa được thực hiện bởi bộ mã hóa, bộ giải mã có thể thu nhận khối dư thông qua việc lượng tử hóa ngược và biến đổi ngược. Bộ giải mã có thể cộng khối dư đoán và khối dư với nhau để thu nhận khối được khôi phục đối với khối hiện tại.

Khi khối được khôi phục của khối hiện tại được thu nhận, tổn hao thông tin như xuất hiện trong xử lý lượng tử hóa và mã hóa có thể được làm giảm thông qua lọc vòng trong. Bộ lọc vòng trong có thể bao gồm ít nhất một trong số bộ lọc giải khối, bộ lọc dịch thích nghi mẫu (SAO-sample adaptive offset), hoặc bộ lọc vòng thích nghi (ALF-adaptive loop filter). Sau đây, khối được khôi phục trước khi bộ lọc vòng trong được áp dụng được gọi là khối được khôi phục thứ nhất và khối được khôi phục sau khi bộ lọc vòng trong được áp dụng được gọi là khối được khôi phục thứ hai.

Khối được khôi phục thứ hai có thể được thu nhận bằng cách áp dụng ít nhất một trong số bộ lọc giải khối, SAO hoặc ALF tới khối được khôi phục thứ nhất. Theo đó, SAO hoặc ALF có thể được áp dụng sau khi bộ lọc giải khối được áp dụng.

Bộ lọc giải khối là để làm giảm sự suy giảm chất lượng (ví dụ, méo khối) trên biên của khối mà xảy ra khi việc lượng tử hóa được thực hiện theo khối. Để áp dụng bộ lọc giải khối, cường độ khối (BS) giữa khối được khôi phục thứ nhất và khối được khôi phục lân cận có thể được xác định.

FIG.31 là lưu đồ thể hiện xử lý xác định cường độ khối.

Trong ví dụ được thể hiện trên FIG.31, P biểu diễn khối được khôi phục thứ nhất và Q biểu diễn khối được khôi phục lân cận. Liên quan đến việc này, khối được khôi phục lân cận có thể lân cận với bên trái hoặc trên cùng của khối hiện tại.

Ví dụ được thể hiện trên FIG.31 thể hiện rằng cường độ khối được xác định bằng cách xem xét chế độ mã hóa dự đoán của P và Q, rằng hệ số biến đổi không phải 0 có được bao gồm hay không, việc dự đoán liên đới có được thực hiện bằng cách sử dụng cùng ảnh tham chiếu hay không hoặc việc giá trị chênh lệch của các vectơ chuyển động có bằng hoặc lớn hơn giá trị ngưỡng hay không.

Dựa trên cường độ khối, việc bộ lọc giải khối có được áp dụng hay không có thể được xác định. Trong ví dụ của sáng chế, khi cường độ khối bằng 0, việc

lọc có thể không được thực hiện.

SAO là để làm giảm méo rung mà xảy ra khi việc lượng tử hóa được thực hiện trong vùng tần số. SAO có thể được thực hiện bằng cách cộng hoặc trừ độ dịch được xác định bằng cách xem xét mẫu của ảnh được khôi phục thứ nhất. Phương pháp xác định độ dịch bao gồm độ dịch biên (EO-Edge Offset) hoặc độ dịch băng (BO-Band Offset). EO biểu diễn phương pháp xác định độ dịch của mẫu hiện tại theo mẫu của các điểm ảnh lân cận. BO biểu diễn phương pháp mà áp dụng độ dịch chung tới nhóm của các điểm ảnh với giá trị độ sáng giống nhau trong vùng. Cụ thể, độ sáng điểm ảnh có thể được chia thành 32 phân đồng nhất và các điểm ảnh với giá trị độ sáng tương tự có thể được thiết lập là một nhóm. Trong ví dụ của sáng chế, 4 băng lân cận trong số 32 băng có thể được thiết lập là một nhóm và giá trị độ dịch tương tự có thể được áp dụng tới các mẫu thuộc về 4 băng.

ALF là phương pháp mà tạo ra ảnh được khôi phục thứ hai bằng cách áp dụng bộ lọc với kích cỡ/dạng định trước tới ảnh được khôi phục thứ nhất hoặc ảnh được khôi phục mà bộ lọc giải khối được áp dụng. Công thức 8 sau đây biểu diễn ví dụ trong đó ALF được áp dụng.

【Công thức 8】

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

Bất kỳ một trong số các ứng viên lọc định trước có thể được lựa chọn trong cơ sở của ảnh, đơn vị cây mã hóa, khối mã hóa, khối dự đoán hoặc khối biến đổi. Ít nhất một trong số kích cỡ hoặc dạng của mỗi ứng viên lọc có thể khác nhau.

FIG.32 biểu diễn các ứng viên lọc định trước.

Như trong ví dụ được thể hiện trên FIG.32, ít nhất bất kỳ một trong số dạng kim cương có kích cỡ 5x5, 7x7 hoặc 9x9 có thể được lựa chọn.

Chỉ dạng kim cương có kích cỡ 5x5 có thể được sử dụng cho thành phần sắc độ.

Phương pháp phân chia ảnh thành các vùng và việc mã hóa/giải mã các

vùng một cách song song có thể được xem xét đối với mã hóa thời gian thực hoặc độ trễ thấp của ảnh phân giải cao như video toàn cảnh, video 360 độ hoặc 4K/8K UHD (Ultra High Definition - Độ phân giải siêu cao). Cụ thể, theo mục đích xử lý, ảnh có thể được phân chia thành các ô hoặc các lát (hoặc các nhóm ô).

Ô biểu diễn đơn vị cơ sở để mã hóa/giải mã song song. Mỗi ô có thể được xử lý song song. Ô có thể có dạng chữ nhật. Ngoài ra, ô không phải chữ nhật có thể được cho phép.

Thông tin mà biểu diễn rằng ô không phải chữ nhật có được cho phép hay không hoặc có ô không phải chữ nhật hay không có thể được báo hiệu trong dòng bit.

Trong việc mã hóa/giải mã ô, có thể được thiết lập để không sử dụng dữ liệu của ô khác. Xử lý song song của các ô có thể được hỗ trợ bằng cách loại bỏ sự phụ thuộc mã hóa/giải mã giữa các ô. Cụ thể, Bảng xác suất của ngữ cảnh CABAC (Context Adaptive Binary Arithmetic Coding-Mã hóa số học nhị phân thích nghi ngữ cảnh) có thể được khởi tạo theo ô và việc lọc nội vòng có thể được thiết lập để không được áp dụng trên biên của các ô. Ngoài ra, dữ liệu trong ô khác có thể không được sử dụng như là ứng viên để thu nhận vector chuyển động. Ví dụ, dữ liệu trong ô khác có thể được thiết lập để không cần được sử dụng như là ứng viên hợp nhất, ứng viên dự đoán vector chuyển động (ứng viên AMVP) hoặc ứng viên thông tin chuyển động. Ngoài ra, dữ liệu trong ô khác có thể được thiết lập để không cần được sử dụng cho việc tính toán ngữ cảnh của ký tự.

Thông tin về mã hóa/giải mã video có thể được báo hiệu bởi thông tin tiêu đề lát. Thông tin được báo hiệu bởi thông tin tiêu đề lát có thể được áp dụng chung tới các đơn vị cây mã hóa hoặc các ô được chứa trong lát. Lát cũng có thể được gọi là nhóm ô.

FIG.33 là sơ đồ thể hiện phương pháp phân chia ảnh theo phương án của sáng chế.

Đầu tiên, việc ảnh hiện tại có được phân chia thành nhiều đơn vị xử lý hay không có thể được xác định S3310. Liên quan đến việc này, đơn vị xử lý có thể bao gồm ít nhất một trong số ô hoặc lát. Theo ví dụ của sáng chế, cú pháp, `no_pic_partition_flag`, mà chỉ báo rằng ảnh hiện tại có được phân chia thành các ô hoặc các lát hay không, có thể được báo hiệu trong dòng bit. Nếu giá trị của cú

pháp, `no_pic_partition_flag`, bằng 0, điều này biểu diễn rằng ảnh hiện tại được phân chia thành ít nhất một ô hoặc ít nhất một lát. Mặt khác, nếu giá trị của cú pháp, `no_pic_partition_flag`, bằng 1, điều này biểu diễn rằng ảnh hiện tại không được phân chia thành các ô hoặc các lát.

Khi được xác định rằng ảnh hiện tại không cần được phân chia thành các đơn vị xử lý, xử lý phân chia của ảnh hiện tại có thể được kết thúc. Liên quan đến việc này, có thể được hiểu rằng ảnh hiện tại bao gồm một ô và một lát (hoặc một nhóm ô).

Ngoài ra, thông tin mà biểu diễn rằng các ô có tồn tại trong ảnh hay không có thể được báo hiệu trong dòng bit. Thông tin này có thể bao gồm ít nhất một trong số cờ 1-bit mà biểu diễn rằng các ô tồn tại trong ảnh hoặc thông tin mà chỉ rõ số lượng ô trong ảnh.

Khi được xác định rằng ảnh hiện tại cần được phân chia thành các đơn vị xử lý, thông tin phân chia ô có thể được báo hiệu trong dòng bit. Ảnh có thể được phân chia thành ít nhất một ô dựa trên thông tin phân chia ô được báo hiệu S3320.

Khi ảnh hiện tại được phân chia thành các ô, lát có thể được xác định bằng cách kết hợp các ô hoặc phân chia ô S3330.

Sau đây, theo sáng chế, phương pháp phân chia ô và phương pháp xác định lát sẽ được mô tả chi tiết.

FIG.34 biểu diễn ví dụ trong đó ảnh được phân chia thành các ô.

Ô có thể bao gồm ít nhất một đơn vị cây mã hóa. Biên của ô có thể được thiết lập để khớp với biên của đơn vị cây mã hóa. Nói cách khác, loại phân chia mà một đơn vị cây mã hóa được phân chia thành nhiều có thể không được cho phép.

Khi ảnh được phân chia thành nhiều ô, độ cao của các ô liên kế hoặc độ rộng của các ô liên kế có thể được thiết lập để có cùng giá trị.

Theo ví dụ của sáng chế, như trong ví dụ được thể hiện trên FIG.34, độ cao của các ô thuộc về cùng hàng ô và/hoặc độ rộng của các ô thuộc về cùng cột ô có thể được thiết lập giống nhau. Các ô thuộc về cùng hàng ô có thể được gọi là tập hợp ô theo chiều ngang và các ô thuộc về cùng cột ô có thể được gọi là tập hợp ô theo chiều dọc.

Ngoài ra, thông tin mà biểu diễn rằng độ rộng và/hoặc độ cao của ô cần được mã hóa/giải mã có được thiết lập tương tự như độ rộng và/hoặc độ cao của ô trước đó hay không có thể được báo hiệu.

Thông tin mà biểu diễn dạng phân chia của ảnh có thể được báo hiệu trong dòng bit. Thông tin có thể được mã hóa và báo hiệu bởi tập hợp tham số ảnh, tập hợp tham số chuỗi hoặc thông tin tiêu đề lát.

Thông tin để xác định kích cỡ ô có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, phần tử cú pháp, `tile_width_minus1[i]`, mà biểu diễn độ rộng của cột ô thứ *i*, và phần tử cú pháp, `tile_height_minus1[i]`, mà biểu diễn độ cao của hàng ô thứ *i* có thể được mã hóa trong dòng bit.

Khi các kích cỡ của các cột ô liên tiếp là giống nhau, thông tin độ rộng có thể được báo hiệu chỉ đối với một trong số các cột ô liên tiếp và việc mã hóa/giải mã thông tin độ rộng có thể được bỏ qua đối với các cột ô còn lại.

Theo ví dụ của sáng chế, khi mỗi độ rộng từ cột ô thứ *i* đến cột ô cuối cùng trong ảnh là giống nhau, cú pháp, `tile_cols_width_minus1[i]`, có thể được báo hiệu chỉ đối với cột ô thứ *i* và độ rộng của cột ô còn lại có thể được thiết lập tương tự như độ rộng của cột ô thứ *i*.

Thông tin mà biểu diễn rằng các độ rộng của các cột ô liên tiếp có giống nhau hay không có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, cú pháp, `rem_tile_col_equal_flag[i]`, biểu diễn rằng các độ rộng từ cột ô mà chỉ số của nó là *i* đến cột ô mà chỉ số của nó là (N-1) có giống nhau hay không. Trong trường hợp này, N có thể biểu diễn số lượng cột ô. Nói cách khác, cột ô mà chỉ số của nó là (N-1) có thể biểu diễn cột ô cuối cùng trong ảnh.

Khi cú pháp, `rem_tile_col_equal_flag[i]`, là 1, cú pháp, `tile_cols_width_minus1[i]`, mà biểu diễn độ rộng có thể được báo hiệu đối với cột ô mà chỉ số của nó là *i*, trong khi việc mã hóa/giải mã của cú pháp mà biểu diễn độ rộng của cột ô có thể được bỏ qua đối với các cột ô mà chỉ số của nó lớn hơn *i*.

Khi việc mã hóa/giải mã của cú pháp mà biểu diễn độ rộng của cột ô được bỏ qua, giá trị nhỏ nhất của độ rộng của cột ô thứ *i* và độ rộng của vùng còn lại có thể được thiết lập là độ rộng của cột ô tiếp theo. Theo ví dụ của sáng chế, khi độ rộng của cột ô thứ *i* được thiết lập là biến, `PrevTileColWidth`, và độ rộng của vùng

còn lại trong ảnh được thiết lập là biến, `RemainingColWidth`, kích cỡ của cột ô mà chỉ số của nó lớn hơn `i` có thể được thiết lập là giá trị nhỏ hơn của biến, `PrevTileColWidth`, và biến, `RemainingColWidth`.

Theo ví dụ khác của sáng chế, cú pháp, `rem_tile_col_equal_flag[i]`, có thể được thiết lập để biểu diễn rằng các độ rộng từ cột ô mà chỉ số của nó là 0 đến cột ô mà chỉ số của nó là `i` có giống nhau hay không.

Khi các kích cỡ của các hàng ô liên tiếp là giống nhau, thông tin độ cao có thể được báo hiệu chỉ đối với một trong số các hàng ô liên tiếp và việc mã hóa/giải mã thông tin độ cao có thể được bỏ qua đối với các hàng ô còn lại.

Theo ví dụ của sáng chế, khi các độ cao từ hàng ô thứ `i` đến hàng ô cuối cùng trong ảnh là giống nhau, cú pháp, `tile_rows_height_minus1[i]`, có thể được báo hiệu chỉ đối với hàng ô thứ `i` và độ cao của hàng ô còn lại có thể được thiết lập tương tự như độ cao của hàng ô thứ `i`.

Thông tin mà biểu diễn rằng các độ cao của các hàng ô liên tiếp có giống nhau hay không có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, cú pháp, `rem_tile_row_equal_flag[i]`, biểu diễn rằng các độ cao từ hàng ô mà chỉ số của nó là `i` đến hàng ô mà chỉ số của nó là $(N-1)$ có giống nhau hay không. Trong trường hợp này, N có thể biểu diễn số lượng hàng ô. Nói cách khác, hàng ô mà chỉ số của nó là $(N-1)$ có thể biểu diễn hàng ô cuối cùng trong ảnh.

Khi cú pháp, `rem_tile_row_equal_flag[i]`, là 1, cú pháp, `tile_rows_height_minus1[i]`, mà biểu diễn độ cao có thể được báo hiệu đối với hàng ô mà chỉ số của nó là `i`, trong khi việc mã hóa/giải mã của cú pháp mà biểu diễn độ cao của hàng ô có thể được bỏ qua đối với các hàng ô mà chỉ số của nó lớn hơn `i`.

Khi việc mã hóa/giải mã của cú pháp mà biểu diễn độ cao của hàng ô được bỏ qua, giá trị nhỏ nhất của độ cao của hàng ô thứ `i` và độ cao của vùng còn lại có thể được thiết lập là độ cao của hàng ô tiếp theo. Theo ví dụ của sáng chế, khi độ cao của hàng ô thứ `i` được thiết lập là biến, `PrevTileRowHeight`, và độ cao của vùng còn lại trong ảnh được thiết lập là biến, `RemainingRowHeight`, kích cỡ của hàng ô mà chỉ số của nó lớn hơn `i` có thể được thiết lập là giá trị nhỏ hơn của biến, `PrevTileRowHeight`, và biến, `RemainingRowHeight`.

Trong ví dụ khác, cú pháp, `rem_tile_row_equal_flag[i]`, có thể được thiết

lập để biểu diễn rằng các độ cao từ hàng ô mà chỉ số của nó là 0 đến hàng ô mà chỉ số của nó là i có giống nhau hay không.

Bảng 8 minh họa cấu trúc cú pháp bao gồm cú pháp, `rem_tile_col_equal_flag[i]`, và cú pháp, `rem_tile_row_equal_flag[i]`.

【Bảng 8】

<code>pic_parameter_set_rbsp() {</code>	Mô tả
<code>...</code>	
<code>if(!single_tile_in_pic_flag) {</code>	
<code>...</code>	
<code>num_tile_columns_minus1</code>	ue(v)
<code>num_tile_rows_minus1</code>	ue(v)
<code>for(i = 0, remWidthInCtbsY = PicWidthInCtbsY; i < num_tile_columns_minus1 && (i == 0 !rem_tile_col_equal_flag[i - 1]); remWidthInCtbsY -= tile_column_width_minus1[i] + 1, i++) {</code>	
<code>if(remWidthInCtbsY % (num_tile_columns_minus1 + 1 - i) == 0)</code>	
<code>rem_tile_col_equal_flag[i]</code>	u(1)
<code>if(!rem_tile_col_equal_flag[i])</code>	
<code>tile_column_width_minus1[i]</code>	ue(v)
<code>}</code>	
<code>for(i = 0, remHeightInCtbsY = PicHeightInCtbsY; i < num_tile_rows_minus1 && (i == 0 !rem_tile_row_equal_flag[i - 1]); remHeightInCtbsY -= tile_row_height_minus1[i] + 1, i++) {</code>	
<code>if(remHeightInCtbsY % (num_tile_rows_minus1 + 1 - i) == 0)</code>	
<code>rem_tile_row_equal_flag[i]</code>	u(1)
<code>if(!rem_tile_row_equal_flag[i])</code>	
<code>tile_row_height_minus1[i]</code>	ue(v)
<code>}</code>	

Thông tin để chỉ rõ số lượng cột ô mà độ rộng của nó được báo hiệu rõ ràng trong ảnh hiện tại có thể được báo hiệu trong dòng bit. Theo ví dụ của sáng chế, cú pháp, `num_exp_tile_columns_minus1`, để xác định số lượng cột ô mà độ rộng của nó được báo hiệu, có thể được báo hiệu trong dòng bit. Cú pháp, `num_exp_tile_columns_minus1`, có thể là giá trị trừ 1 từ số lượng cột ô mà độ rộng của nó được báo hiệu.

Bằng với số lượng được xác định dựa trên cú pháp, `num_exp_tile_columns_minus1`, cú pháp để chỉ rõ độ rộng của cột ô có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, cú pháp, `tile_width_minus1[i]`, mà biểu diễn độ rộng của cột ô thứ *i*, có thể được báo hiệu trong dòng bit. Cú pháp, `tile_width_minus1[i]`, có thể biểu diễn giá trị mà trừ 1 từ số lượng cột đơn vị cây mã hóa được chứa trong hàng ô.

Khi chỉ số *i* của cột ô nhỏ hơn số lượng cột ô mà độ rộng của nó được báo hiệu rõ ràng, độ rộng của cột ô tương ứng có thể được xác định dựa trên cú pháp, `tile_width_minus1[i]`, được báo hiệu trong dòng bit.

Mặt khác, khi chỉ số *j* của cột ô bằng hoặc lớn hơn số lượng cột ô mà độ rộng của nó được báo hiệu rõ ràng, độ rộng của cột ô tương ứng có thể được xác định dựa trên cú pháp, `tile_width_minus1[i]`, mà được báo hiệu cuối cùng. Trong trường hợp này, *i* có thể biểu diễn chỉ số của cột ô mà độ rộng của nó được báo hiệu cuối cùng và có thể là số nguyên nhỏ hơn *j*. Theo ví dụ của sáng chế, khi giá trị mà trừ độ rộng của các cột ô trước đó từ độ rộng của ảnh hiện tại bằng hoặc lớn hơn giá trị mà cộng 1 vào cú pháp, `tile_width_minus1[i]`, độ rộng của cột ô *j* có thể được thiết lập là giá trị mà cộng 1 vào cú pháp, `tile_width_minus1[i]`. Mặt khác, khi giá trị mà trừ các độ rộng của các cột ô trước đó từ độ rộng của ảnh hiện tại nhỏ hơn giá trị mà cộng 1 vào cú pháp, `tile_width_minus1[i]`, giá trị chênh lệch mà trừ các độ rộng của các cột ô trước đó từ độ rộng của ảnh hiện tại có thể được thiết lập là độ rộng của cột ô *j*.

Bảng 9 thể hiện xử lý trong đó độ rộng của cột ô được xác định.

【Bảng 9】

```
remainingWidthInCtbsY = PicWidthInCtbsY
for( i = 0; i <= num_exp_tile_columns_minus1; i++ ) {
    colWidth[ i ] = tile_column_width_minus1[ i ] + 1
    remainingWidthInCtbsY -= colWidth[ i ]
}
uniformTileColWidth =
tile_column_width_minus1[ num_exp_tile_columns_minus1 ] + 1
while( remainingWidthInCtbsY >= uniformTileColWidth ) {
```

```

colWidth[ i++ ] = uniformTileColWidth
remainingWidthInCtbsY -= uniformTileColWidth
}
if( remainingWidthInCtbsY > 0 )
    colWidth[ i++ ] = remainingWidthInCtbsY
NumTileColumns = i

```

Trong Bảng 9, biến, *PicWidthInCtbsY*, biểu diễn số lượng cột đơn vị cây mã hóa được bao gồm bởi ảnh hiện tại. Theo ví dụ của sáng chế, biến, *PicWidthInCtbsY*, có thể được thu nhận như trong công thức 9 sau đây.

【Công thức 9】

$$PicWidthInCtbsY = Ceil (pic_width_in_luma_samples / CtbSizeY)$$

Biến, *reminingWidthInCtbsY*, biểu diễn giá trị mà trừ các độ rộng được tích lũy của các cột ô từ biến, *PicWidthInCtbsY*. Theo ví dụ của sáng chế, đối với cột ô mà chỉ số của nó là *i*, *reminingWidthInCtbsY* có thể được thu nhận bằng cách trừ giá trị mà là tổng của các độ rộng từ các cột ô thứ 0 đến thứ (*i*-1) từ *PicWidthInCtbsY*.

Như trong ví dụ được thể hiện trong Bảng 9, biến, *uniformTileColWidth*, có thể được thu nhận bằng cách cộng 1 vào *tile_column_width_minus1[num_exp_tile_columns_minus1]*, độ rộng của cột ô mà được báo hiệu rõ ràng cuối cùng. Trong trường hợp này, khi biến, *remainingWidthInCtbY*, mà biểu diễn số lượng cột đơn vị cây mã hóa còn lại trong ảnh hiện tại bằng hoặc lớn hơn biến, *uniformTileColWidth*, vùng còn lại có thể được phân chia trong kích cỡ của biến, *uniformTileColWidth*.

Mặt khác, khi biến, *remainingWidthInCtbY*, mà biểu diễn số lượng cột đơn vị cây mã hóa còn lại trong ảnh hiện tại nhỏ hơn biến, *uniformTileColWidth*, vùng còn lại có thể được thiết lập là cột ô cuối cùng như thực tại.

Mặt khác, khi chỉ số *j* của cột ô bằng hoặc lớn hơn số lượng cột ô mà độ rộng của nó được báo hiệu rõ ràng, độ rộng của cột ô tương ứng có thể được thiết lập là giá trị nhỏ hơn của biến *uniformTileColWidth* và *remainingWidthInCtbY*.

Nói cách khác, độ rộng của các cột ô còn lại ngoại trừ các cột ô mà độ rộng của nó được báo hiệu rõ ràng có thể có giá trị nhỏ hơn hoặc bằng độ rộng của cột ô cuối cùng trong số các cột ô mà độ rộng của nó được báo hiệu rõ ràng.

Ngoài ra, thông tin để chỉ rõ số lượng hàng ô mà độ cao của nó trong ảnh hiện tại được báo hiệu có thể được báo hiệu trong dòng bit. Theo ví dụ của sáng chế, cú pháp, `num_exp_tile_rows_minus1` để xác định số lượng hàng ô mà độ cao của nó được báo hiệu, có thể được báo hiệu trong dòng bit. Cú pháp, `num_exp_tile_rows_minus1`, có thể là giá trị mà trừ 1 từ số lượng hàng ô mà độ cao của nó được báo hiệu.

Bằng với số lượng được xác định dựa trên cú pháp, `num_exp_tile_rows_minus1`, cú pháp để chỉ rõ độ cao của hàng ô có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, cú pháp, `tile_height_minus1[i]`, mà biểu diễn độ cao của hàng ô thứ *i* có thể được báo hiệu trong dòng bit. Cú pháp, `tile_height_minus1[i]`, có thể biểu diễn giá trị mà trừ 1 từ số lượng hàng đơn vị cây mã hóa được chứa trong hàng ô.

Khi chỉ số *i* của hàng ô nhỏ hơn số lượng hàng ô mà độ cao của nó được báo hiệu rõ ràng, độ cao của hàng ô tương ứng có thể được xác định dựa trên cú pháp, `tile_height_minus1[i]`, được báo hiệu trong dòng bit.

Mặt khác, khi chỉ số *j* của hàng ô bằng hoặc lớn hơn số lượng hàng ô mà độ cao của nó được báo hiệu rõ ràng, độ cao của hàng ô tương ứng có thể được xác định dựa trên cú pháp, `tile_height_minus1[l]`, mà được báo hiệu cuối cùng. Trong trường hợp này, *l* có thể biểu diễn chỉ số của hàng ô mà độ cao của nó được báo hiệu cuối cùng và có thể là số nguyên nhỏ hơn *j*.

Theo ví dụ của sáng chế, khi giá trị mà trừ các độ cao của các hàng ô trước đó từ độ cao của ảnh hiện tại bằng hoặc lớn hơn giá trị mà cộng 1 vào cú pháp, `tile_height_minus1[l]`, độ cao của hàng ô *j* có thể được thiết lập là giá trị mà cộng 1 vào cú pháp, `tile_height_minus1[l]`. Mặt khác, khi giá trị mà trừ các độ cao của các hàng ô trước đó từ độ cao của ảnh hiện tại nhỏ hơn giá trị mà cộng 1 vào cú pháp, `tile_height_minus1[l]`, giá trị chênh lệch mà trừ các độ cao của các hàng ô trước đó từ độ cao của ảnh hiện tại có thể được thiết lập là độ cao của hàng ô *j*.

Bảng 10 thể hiện xử lý trong đó độ cao của hàng ô được xác định.

【Bảng 10】

```

remainingHeightInCtbsY = PicHeightInCtbsY
for( j = 0; j < num_exp_tile_rows_minus1; j++ ) {
    RowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    remainingHeightInCtbsY -= RowHeight[ j ]
}
uniformTileRowHeight =
tile_row_height_minus1[ num_exp_tile_rows_minus1 ] + 1
while( remainingHeightInCtbsY >= uniformTileRowHeight ) {
    RowHeight[ j++ ] = uniformTileRowHeight
    remainingHeightInCtbsY -= uniformTileRowHeight
}
if( remainingHeightInCtbsY > 0 )
    RowHeight[ j++ ] = remainingHeightInCtbsY
NumTileRows = j

```

Trong Bảng 10, biến, *PicHeightInCtbsY*, biểu diễn số lượng hàng đơn vị cây mã hóa được chứa bởi ảnh hiện tại. Theo ví dụ của sáng chế, biến, *PicWidthInCtbsY*, có thể được thu nhận như trong công thức 10 sau đây.

【Công thức 10】

$$PicHeightInCtbsY = Ceil(pic_height_in_luma_samples / CtbSizeY)$$

Biến, *reminingHeightInCtbsY*, biểu diễn giá trị mà trừ các độ cao được tích lũy của các hàng ô từ biến, *PicHeightInCtbsY*. Theo ví dụ của sáng chế, đối với hàng ô mà chỉ số của nó là *i*, *remainingHeightInCtbsY* có thể được thu nhận bằng cách trừ giá trị mà là tổng của các độ cao từ các hàng ô thứ 0 đến (*i*-1) từ *PicHeightInCtbsY*.

Như trong ví dụ được thể hiện trong Bảng 10, biến, *uniformTileRowHeight*, có thể được thu nhận bằng cách cộng 1 vào

`tile_row_height_minus1[num_exp_tile_rows_minus1]`, độ cao của hàng ô mà được báo hiệu rõ ràng cuối cùng. Trong trường hợp này, khi biến, `remainingHeightInCtbY`, mà biểu diễn số lượng hàng đơn vị cây mã hóa còn lại trong ảnh hiện tại bằng hoặc lớn hơn biến, `uniformTileRowHeight`, vùng còn lại có thể được phân chia theo kích cỡ của biến, `uniformTileRowHeight`.

Mặt khác, khi biến, `remainingHeightInCtbY`, mà biểu diễn số lượng hàng đơn vị cây mã hóa còn lại trong ảnh hiện tại nhỏ hơn biến, `uniformTileRowHeight`, vùng còn lại có thể được thiết lập là hàng ô cuối cùng như thực tại.

Mặt khác, khi chỉ số `j` của hàng ô bằng hoặc lớn hơn số lượng hàng ô mà độ cao của nó được báo hiệu rõ ràng, độ cao của hàng ô tương ứng có thể được thiết lập là giá trị nhỏ hơn của biến `uniformTileRowHeight` và `remainingHeightInCtbY`.

Nói cách khác, độ cao của các hàng ô còn lại ngoại trừ các hàng ô mà độ cao của nó được báo hiệu rõ ràng có thể có giá trị nhỏ hơn hoặc bằng độ rộng của hàng ô cuối cùng trong số các hàng ô mà độ cao của nó được báo hiệu rõ ràng.

Bảng 11 minh họa Bảng cú pháp mà bao gồm cú pháp biểu diễn số lượng cột ô mà độ rộng của nó được báo hiệu rõ ràng và cú pháp mà biểu diễn số lượng hàng ô mà độ cao của nó được báo hiệu rõ ràng.

【Bảng 11】

<code>pic_parameter_set_rbsp()</code> {	Mô tả
...	
<code>single_tile_in_pic_flag</code>	<code>u(1)</code>
<code>if(!single_tile_in_pic_flag) {</code>	
<code>log2_pps_ctu_size_minus5</code>	<code>u(2)</code>
<code>num_exp_tile_columns_minus1</code>	<code>ue(v)</code>
<code>num_exp_tile_rows_minus1</code>	<code>ue(v)</code>
<code>for(i = 0; i <= num_exp_tile_columns_minus1; i++)</code>	
<code>tile_column_width_minus1[i]</code>	<code>ue(v)</code>
<code>for(i = 0; i <= num_exp_tile_rows_minus1; i++)</code>	
<code>tile_row_height_minus1[i]</code>	<code>ue(v)</code>
<code>loop_filter_across_tiles_enabled_flag</code>	<code>u(1)</code>
<code>loop_filter_across_slices_enabled_flag</code>	<code>u(1)</code>

Trong khi đó, cú pháp, `loop_filter_across_tiles_enabled_flag`, biểu diễn

rằng có được cho phép áp dụng lọc nội vòng hay không trên biên của các ô trong ảnh mà liên quan đến tập hợp tham số ảnh. Trong trường hợp này, bộ lọc nội vòng có thể bao gồm ít nhất một trong số bộ lọc giải khối, ALF hoặc SAO. Khi giá trị của cờ, `loop_filter_across_tiles_enabled_flag`, là 1, điều này biểu diễn rằng lọc nội vòng mà đi qua biên của các ô trong ảnh mà viện dẫn đến tập hợp tham số ảnh có thể được áp dụng. Mặt khác, khi giá trị của cờ, `loop_filter_across_tiles_enabled_flag`, là 0, điều này biểu diễn rằng không được phép áp dụng lọc nội vòng trên biên của các ô trong ảnh mà viện dẫn đến tập hợp tham số ảnh.

Cú pháp, `loop_filter_across_slices_enabled_flag`, biểu diễn rằng có được cho phép áp dụng lọc nội vòng hay không trên biên của các lát trong ảnh mà viện dẫn đến tập hợp tham số ảnh. Trong trường hợp này, bộ lọc nội vòng có thể bao gồm ít nhất một trong số bộ lọc giải khối, ALF hoặc SAO. khi giá trị của cờ, `loop_filter_across_slices_enabled_flag`, là 1, điều này biểu diễn rằng lọc nội vòng mà đi qua biên của các lát trong ảnh mà viện dẫn đến tập hợp tham số ảnh có thể được áp dụng. Mặt khác, khi giá trị của cờ, `loop_filter_across_slices_enabled_flag`, là 0, điều này biểu diễn rằng không được phép áp dụng lọc nội vòng trên biên của các lát trong ảnh mà viện dẫn đến tập hợp tham số ảnh.

Ô có thể được phân chia đệ quy. Theo ví dụ của sáng chế, một ô có thể được phân chia thành nhiều ô.

Mỗi ô được tạo ra bằng cách phân chia ô có thể được gọi là ô con hoặc đơn vị viên. Đơn vị viên có thể là đơn vị xử lý song song. Theo ví dụ của sáng chế, đơn vị viên có thể được mã hóa/giải mã độc lập nhau. Trong việc mã hóa/giải mã khối được chứa trong viên, có thể được thiết lập để không sử dụng dữ liệu của khối khác. Theo ví dụ của sáng chế, mẫu được chứa trong viên khác có thể được thiết lập là không khả dụng như mẫu tham chiếu cho việc nội dự đoán. Ngoài ra, dữ liệu trong viên khác có thể được thiết lập để không cần được sử dụng như là ứng viên hợp nhất, ứng viên dự đoán vector chuyển động (ứng viên AMVP) hoặc ứng viên thông tin chuyển động. Ngoài ra, dữ liệu trong viên khác có thể không được sử dụng cho việc tính toán ngưỡng cảnh của ký tự.

FIG.35 là sơ đồ để giải thích khía cạnh tạo ra đơn vị viên.

Đơn vị viên có thể được tạo ra bằng cách phân chia ô trong chiều ngang.

Theo ví dụ của sáng chế, ví dụ được thể hiện trên FIG.35 thể hiện rằng các ô thuộc về cột ô cuối cùng trong ảnh hiện tại được phân chia thành 2 đơn vị viên.

Khi được giả định rằng việc quét mảnh được áp dụng tới các ô, việc quét mảnh có thể được áp dụng giữa các đơn vị viên. Theo ví dụ của sáng chế, sau khi quét tất cả các đơn vị viên được chứa trong ô cụ thể, ô tiếp theo có thể được quét. Nói cách khác, đơn vị viên có thể có cùng trạng thái như ô.

Biên của đơn vị viên có thể khớp với của đơn vị cây mã hóa. Nói cách khác, ít nhất một hàng đơn vị cây mã hóa trong ô có thể được xác định là đơn vị viên.

Ô và đơn vị viên có thể được nhận dạng bởi chỉ số ô. Chỉ số ô có thể được gán tới mỗi ô và đơn vị viên trong thứ tự quét mảnh. Khi một ô được phân chia thành các đơn vị viên, chỉ số ô có thể được gán tới mỗi đơn vị viên.

Trong phương án sau đây, thuật ngữ, 'ô', có thể bao gồm các ô được tạo ra bằng cách phân chia ô (tức là, ô con hoặc đơn vị viên) cũng như ô.

Ít nhất một hoặc nhiều ô có thể được xác định là một đơn vị xử lý. Theo ví dụ của sáng chế, các ô có thể được xác định là một lát. Lát có thể được gọi là nhóm ô.

Ngoài ra, một ô có thể được phân chia thành các đơn vị xử lý. Theo ví dụ của sáng chế, ô có thể được phân chia thành các lát. Liên quan đến việc này, một lát có thể bao gồm ít nhất một cột đơn vị cây mã hóa. Khi ô được phân chia thành các lát, thông tin mà biểu diễn độ cao của mỗi lát có thể được báo hiệu trong dòng bit.

Thông tin mã hóa/giải mã ảnh có thể được báo hiệu thông qua thông tin tiêu đề lát. Thông tin được báo hiệu thông qua thông tin tiêu đề lát có thể được áp dụng chung tới các ô và/hoặc các khối thuộc về lát.

Thông tin mà biểu diễn loại lát có thể được báo hiệu thông qua dòng bit. Thông tin này biểu diễn phương pháp xác định lát trong ảnh hiện tại. Theo ví dụ của sáng chế, cú pháp, `rect_slice_flag`, mà biểu diễn loại lát có thể được báo hiệu trong dòng bit.

Cú pháp, `rect_slice_flag`, biểu diễn rằng lát có được xác định dựa trên thứ tự quét mảnh của các ô hay không hoặc lát có được xác định trong dạng chữ nhật

hay không. Theo ví dụ của sáng chế, khi `rect_slice_flag` là 0, điều này biểu diễn rằng lát được xác định dựa trên thứ tự quét mảnh của các ô. Mặt khác, khi `rect_slice_flag` là 1, điều này biểu diễn rằng lát được xác định trong dạng chữ nhật.

Sau đây, hai phương pháp để xác định lát sẽ được mô tả chi tiết.

Phương pháp xác định dựa trên quét mảnh là để chỉ rõ ít nhất một hoặc nhiều ô theo thứ tự quét mảnh và xác định ít nhất một hoặc nhiều ô được chỉ rõ như là lát. Khi phương pháp xác định dựa trên quét mảnh được tuân theo, một hoặc nhiều ô liên tiếp có thể được xác định là lát. Trong trường hợp này, các ô liên tiếp có thể được xác định theo thứ tự quét mảnh. Khi lát quét mảnh được áp dụng, lát không phải chữ nhật có thể được tạo ra.

Các Fig.36 và Fig.37 là các sơ đồ thể hiện ví dụ trong đó lát được xác định dựa trên thứ tự quét mảnh.

Theo ví dụ của sáng chế, trong ví dụ được thể hiện trên FIG.36, khi giả thiết rằng lát thứ nhất, lát 0, bao gồm 3 ô, lát thứ nhất, lát 0, có thể được xác định là bao gồm ô 0 đến ô 2 theo thứ tự quét mảnh. Khi Giả thiết rằng lát thứ hai, lát 1, bao gồm 6 ô, lát thứ hai, lát 1, có thể được xác định là bao gồm ô 3 đến ô 8 theo thứ tự quét mảnh. Lát cuối cùng, lát 2, có thể bao gồm các ô dư, ô 9 đến ô 11, theo thứ tự quét mảnh.

Khi lát được xác định dựa trên thứ tự quét mảnh, thông tin mà biểu diễn số lượng ô được chứa bởi mỗi lát có thể được báo hiệu. Báo hiệu của thông tin mà biểu diễn số lượng ô được chứa bởi lát có thể được bỏ qua đối với lát cuối cùng.

Khi lát bao gồm nhiều ô, độ rộng hoặc độ cao của các ô được chứa trong lát có thể khác nhau. Theo ví dụ của sáng chế, được thể hiện rằng độ cao của ô 3 trong số các ô được chứa bởi lát thứ hai, lát 1, là khác với các ô khác.

Phương pháp xác định lát trong dạng chữ nhật là phương pháp phân chia mà cho phép chỉ lát trong dạng chữ nhật. Khi phương pháp xác định lát trong dạng chữ nhật được áp dụng, các ô có vị trí tại bốn góc của lát thuộc về cùng hàng hoặc cùng cột.

FIG.38 là sơ đồ thể hiện ví dụ trong đó chỉ lát trong dạng hình chữ nhật được cho phép.

Như trong ví dụ được thể hiện trên FIG.38, lát thứ tư, lát 3, bao gồm ô 5,

ô 6, ô 9 và ô 10. Như trong ví dụ được thể hiện, khi lát bao gồm nhiều ô, hình chữ nhật mà có ô trên cùng-bên trái và ô dưới cùng-bên phải như là hai góc của nó có thể được xác định là một lát.

Biên của lát có thể khớp với biên của ảnh và/hoặc biên của ô. Theo ví dụ của sáng chế, biên bên trái hoặc biên trên cùng của lát có thể được thiết lập là biên của ảnh, hoặc biên bên trái hoặc biên trên cùng của lát có thể được thiết lập là biên của ô.

Ngoài ra, khi phương pháp xác định lát chữ nhật được áp dụng, một ô có thể được phân chia thành nhiều lát chữ nhật.

Khi phương pháp xác định lát chữ nhật được áp dụng (ví dụ, khi `rect_slice_flag` is 1), thông tin mà biểu diễn rằng ảnh có được cấu hình với một lát hay không có thể được báo hiệu. Theo ví dụ của sáng chế, cú pháp, `one_slice_in_pic_flag`, mà biểu diễn rằng số lượng lát trong ảnh có bằng 1 hay không có thể được báo hiệu trong dòng bit. Khi cờ, `one_slice_in_pic_flag`, là 1, điều này biểu diễn rằng ảnh được cấu hình với một lát. Mặt khác, khi cờ, `one_slice_in_pic_flag`, là 0, điều này biểu diễn rằng ảnh được cấu hình với ít nhất 2 lát hoặc nhiều hơn. Khi giá trị của cờ, `one_slice_in_pic_flag`, là 0, thông tin cấu hình của lát có thể được báo hiệu bổ sung. Theo ví dụ của sáng chế, Bảng 12 minh họa Bảng cú pháp bao gồm cú pháp `one_slice_in_pic_flag`.

【Bảng 12】

<code>pic_parameter_set_rbsp() {</code>	Mô tả
<code>pps_pic_parameter_set_id</code>	ue(v)
<code>pps_seq_parameter_set_id</code>	ue(v)
<code>single_tile_in_pic_flag</code>	u(1)
...	
<code>single_brick_per_slice_flag</code>	u(1)
<code>if(!single_brick_per_slice_flag)</code>	
<code>rect_slice_flag</code>	u(1)
<code>if(rect_slice_flag && !single_brick_per_slice_flag) {</code>	
<code>one_slice_in_pic_flag</code>	u(1)

if (!one_slice_in_pic_flag){	
num_slices_in_pic_minus2	ue(v)
for(i = 0; i <= num_slices_in_pic_minus2+1; i++) {	
if(i > 0)	
top_left_brick_idx [i]	u(v)
bottom_right_brick_idx_delta[i]	u(v)
}	
}	
}	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	

Như được minh họa, khi `one_slice_in_pic_flag` là 1, việc mã hóa thông tin liên quan đến cấu hình của lát mà sẽ được mô tả sau đây (ví dụ, cú pháp, `top_left_brick_idx[i]`, mà biểu diễn chỉ số của ô được chứa bởi lát, và/hoặc cú pháp, `bottom_right_brick_idx_delta[i]`, mà biểu diễn độ chênh lệch chỉ số giữa ô được chứa bởi lát thứ nhất và ô được chứa bởi lát thứ hai, v.v) có thể được bỏ qua. Nói cách khác, `one_slice_in_pic_flag` có thể được sử dụng để xác định rằng thông tin liên quan đến cấu hình của lát có được mã hóa hay không.

Ảnh có thể biểu diễn Ảnh con. Ảnh có thể được phân chia thành ít nhất một ảnh con. Thông tin liên quan đến việc phân chia ảnh con có thể được báo hiệu tại mức chuỗi. Theo ví dụ của sáng chế, thông tin liên quan đến phân chia ảnh con mà được báo hiệu thông qua tập tham số chuỗi có thể được áp dụng chung tới tất cả các ảnh mà viện dẫn tới tập hợp tham số chuỗi.

Thông tin liên quan đến phân chia ảnh con có thể bao gồm thông tin mà biểu diễn rằng ảnh có được phân chia thành các ảnh con hay không, thông tin mà biểu diễn số lượng ảnh con, thông tin mà biểu diễn kích cỡ của ảnh con hoặc thông tin mà biểu diễn rằng ảnh con có được xem là ảnh độc lập hay không.

Cú pháp, `one_slice_in_pic_flag`, mà biểu diễn rằng có được cấu hình với

một lát hay không có thể được báo hiệu theo ảnh con.

Ngoài ra, cú pháp, `one_slice_in_pic_flag`, mà biểu diễn rằng mỗi ảnh con có được cấu hình với một lát hay không có thể được báo hiệu tại mức chuỗi hoặc mức ảnh. Theo ví dụ của sáng chế, khi cờ, `one_slice_in_pic_flag`, là 1, điều này biểu diễn rằng tất cả các ảnh con trong ảnh hiện tại được cấu hình với một lát. Trong trường hợp này, việc mã hóa của thông tin cấu hình lát có thể được bỏ qua và các lát có thể được xác định dựa trên thông tin phân chia của ảnh.

Mặt khác, khi cờ, `one_slice_in_pic_flag`, là 0, điều này biểu diễn rằng ít nhất một ảnh con trong ảnh hiện tại được cấu hình với nhiều lát. Trong trường hợp này, thông tin cấu hình lát có thể được mã hóa riêng biệt với thông tin phân chia ảnh con.

Khi được xác định rằng ảnh được cấu hình với các lát, cú pháp mà biểu diễn số lượng lát trong ảnh có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, cú pháp, `num_slices_in_pic_minus2`, mà biểu diễn số lượng lát trong ảnh có thể được báo hiệu trong dòng bit. Cú pháp, `num_slices_in_pic_minus2`, có thể biểu diễn giá trị mà trừ 2 từ số lượng lát trong ảnh. Do đó, bộ giải mã có thể xác định tổng số lượng lát bằng cách cộng 2 vào giá trị được chỉ báo bởi cú pháp, `num_slices_in_pic_minus2`.

Thay vì cú pháp, `num_slices_in_pic_minus2`, cú pháp, `num_slices_in_pic_minus1`, mà biểu diễn giá trị mà trừ 1 từ số lượng lát trong ảnh có thể được mã hóa.

Khi phương pháp xác định lát trong dạng chữ nhật được áp dụng, thông tin để nhận dạng các ô được chứa bởi mỗi laats có thể được báo hiệu để xác định các ô được chứa bởi mỗi lát. Thông tin này có thể được sử dụng để chỉ rõ ít nhất một trong số ô thứ nhất hoặc ô cuối cùng của lát. Thứ tự giữa các ô có thể được xác định theo thứ tự quét định trước. Theo ví dụ của sáng chế, khi thứ tự quét mảnh được áp dụng, ô thứ nhất chỉ báo ô tại vị trí trên cùng-bên trái của lát và ô cuối cùng chỉ báo ô tại vị trí dưới cùng-bên phải của lát.

Thông tin để nhận dạng ít nhất một trong số chỉ số của ô tại vị trí trên cùng-bên trái của lát hoặc chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được báo hiệu trong dòng bit. Theo ví dụ của sáng chế, ít nhất một trong số cú pháp, `top_left_tile_idx`, để nhận dạng chỉ số của ô tại vị trí trên cùng-bên trái của lát,

hoặc cú pháp, `bottom_right_tile_idx`, để nhận dạng chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được báo hiệu trong dòng bit. Việc mã hóa của ít nhất một trong số cú pháp để nhận dạng chỉ số của ô tại vị trí trên cùng-bên trái của lát hoặc cú pháp để nhận dạng chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được bỏ qua đối với lát cuối cùng. Vùng còn lại ngoại trừ vùng được chiếm giữ bởi các lát trước đó trong ảnh hiện tại có thể được thiết lập là lát cuối cùng.

Theo ví dụ của sáng chế, trong ví dụ được thể hiện trên FIG.38, `top_left_tile_idx` mà biểu diễn chỉ số của ô tại vị trí trên cùng-bên trái của lát, và `bottom_right_tile_idx` mà biểu diễn chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được báo hiệu đối với mỗi lát 0, lát 1 và lát 2. Mặt khác, việc mã hóa của `top_left_tile_idx` và `bottom_right_tile_idx` có thể được bỏ qua đối với lát cuối cùng, lát 3, trong ảnh. Chỉ số của ô trên cùng-bên trái của lát cuối cùng, lát 3, trong ảnh, có thể được thiết lập là chỉ số của ô tại vị trí trên cùng-bên trái trong vùng còn lại ngoại trừ lát 0, lát 1 và lát 2 trong ảnh và chỉ số của ô dưới cùng-bên phải của lát 3 có thể được thiết lập là chỉ số của ô tại vị trí dưới cùng-bên phải trong vùng còn lại (hoặc, ô tại vị trí dưới cùng-bên phải của ảnh).

Ngoài ra, thông tin chênh lệch để chỉ rõ chỉ số của ô được chứa trong lát có thể được mã hóa và báo hiệu. Theo ví dụ của sáng chế, thông tin mà biểu diễn giá trị chênh lệch của chỉ số của ô tại vị trí trên cùng-bên trái của lát và chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được báo hiệu trong dòng bit. Theo ví dụ của sáng chế, cú pháp, `top_left_tile_idx[i]`, để nhận dạng chỉ số của ô tại vị trí trên cùng-bên trái của lát, và cú pháp, `bottom_right_tile_idx_delta[i]`, mà biểu diễn độ chênh lệch giữa chỉ số của ô tại vị trí trên cùng-bên trái của lát và chỉ số của ô tại vị trí dưới cùng-bên phải của lát, có thể được báo hiệu đối với lát thứ i . Chỉ số của ô dưới cùng-bên phải của lát thứ i có thể được thu nhận bằng cách tính tổng `top_left_tile_idx[i]` và `bottom_right_tile_idx_delta[i]`. Việc mã hóa của ít nhất một trong số cú pháp mà biểu diễn chỉ số của ô tại vị trí trên cùng-bên trái của lát hoặc cú pháp mà biểu diễn độ chênh lệch của chỉ số của ô tại vị trí trên cùng-bên trái của lát và chỉ số của ô tại vị trí dưới cùng-bên phải của lát có thể được bỏ qua đối với lát cuối cùng. Vùng còn lại ngoại trừ vùng được chiếm giữ bởi các lát trước đó trong ảnh hiện tại có thể được thiết lập là lát cuối cùng.

Ngoài ra, ít nhất một trong số thông tin chênh lệch chỉ số ô theo chiều ngang hoặc thông tin chênh lệch chỉ số ô theo chiều dọc có thể được mã hóa và

báo hiệu. Thông tin chênh lệch chỉ số ô theo chiều ngang có thể biểu diễn độ chênh lệch giữa chỉ số của ô thứ nhất và chỉ số của ô ngoài cùng bên phải được chứa trong cùng hàng ô như ô thứ nhất. Thông tin chênh lệch chỉ số ô theo chiều dọc có thể biểu diễn độ chênh lệch giữa chỉ số của ô thứ nhất và chỉ số của ô thấp nhất được chứa trong cùng cột ô như ô thứ nhất.

Theo vị trí của lát, loại thông tin chênh lệch cần được mã hóa có thể được xác định. Theo ví dụ của sáng chế, đối với lát liền kề với biên bên phải hoặc biên dưới cùng của ảnh, thông tin chênh lệch chỉ số theo chiều ngang hoặc thông tin chênh lệch chỉ số ô theo chiều dọc có thể được mã hóa và báo hiệu. Số lượng bit có thể được làm giảm bằng cách mã hóa thông tin chênh lệch chỉ số ô theo chiều ngang hoặc thông tin chênh lệch chỉ số theo chiều dọc thay vì thông tin mà biểu diễn độ chênh lệch chỉ số giữa ô trên cùng-bên trái và ô trên cùng-bên phải.

Ngoài ra, khi ảnh được phân chia thành ít nhất một hoặc nhiều hàng lát, thông tin mà biểu diễn độ chênh lệch chỉ số giữa ô trên cùng-bên trái và ô dưới cùng-bên phải có thể được mã hóa đối với lát liền kề với biên bên trái của ảnh. Các lát thuộc về cùng hàng có thể được thiết lập để có cùng độ cao.

Thông tin chênh lệch giữa chỉ số của ô được chứa trong lát thứ nhất và chỉ số của ô được chứa trong lát thứ hai có thể được mã hóa và báo hiệu. Trong trường hợp này, lát thứ nhất và lát thứ hai có thể được xác định dựa trên thứ tự quét và mỗi lát có thể được chiếm giữ bởi ô khác nhau.

Theo ví dụ của sáng chế, khi ô mà cấu thành lát thứ nhất và ô mà cấu thành lát thứ hai là khác nhau, chỉ số của lát thứ hai có thể được thu nhận bằng cách cộng 1 vào chỉ số i của lát thứ nhất.

Ngoài ra, khi ô thứ nhất được phân chia để bao gồm các lát, thông tin chênh lệch giữa lát thứ nhất được chứa trong ô thứ nhất và lát thứ hai bao gồm ô thứ hai mà là ô tiếp theo của ô thứ nhất trong thứ tự quét hoặc giữa lát thứ nhất được chứa trong ô thứ nhất và lát thứ hai được chứa trong ô thứ hai có thể được mã hóa. Trong trường hợp này, thông tin chênh lệch có thể được mã hóa chỉ đối với lát thứ nhất hoặc lát cuối cùng trong số các lát được chứa trong ô thứ nhất.

Ô được sử dụng để thu nhận thông tin chênh lệch giữa lát thứ nhất và lát thứ hai có thể có vị trí tại trên cùng-bên trái, trên cùng-bên phải, dưới cùng-bên phải, dưới cùng-bên trái hoặc trung tâm của lát.

Bảng 13 minh họa Bảng cú pháp bao gồm thông tin chênh lệch.

【Bảng 13】

pic_parameter_set_rbsp() {	Mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
single_tile_in_pic_flag	u(1)
...	
rect_slice_flag	u(1)
if(rect_slice_flag && !single_brick_per_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i <= num_slices_in_pic_minus1; i++) {	
if(i > 0)	
top_left_brick_idx_delta[i]	u(v)
bottom_right_brick_idx_delta[i]	u(v)
}	
}	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	

Theo ví dụ của sáng chế, cú pháp, **top_left_brick_idx_delta[i]**, mà biểu diễn độ chênh lệch chỉ số giữa ô trên cùng-bên trái của lát thứ *i* và ô trên cùng-bên trái của lát trước đó (tức là, lát thứ (*i*-1)), có thể được báo hiệu trong dòng bit.

Dựa trên thông tin chênh lệch, vị trí của ô trên cùng-bên trái trong lát thứ *i* có thể được xác định. Cụ thể, chỉ số của ô trên cùng-bên trái của lát thứ *i* có thể được thu nhận bằng cách cộng chỉ số của ô trên cùng-bên trái của lát thứ (*i*-1) và cú pháp, **top_left_brick_idx_delta[i]**. Công thức 11 biểu diễn ví dụ về việc thu nhận chỉ số của ô trên cùng-bên trái của lát thứ *i*.

【Công thức 11】

$$TopLeftBrickIdx[i] = TopLeftBrickIdx[i-1] + top_left_brick_idx_delta[i]$$

Trong công thức 11, TopLeftBrickIdx[i-1] biểu diễn chỉ số của ô trên cùng-bên trái của lát thứ (i-1).

Việc mã hóa của cú pháp, top_left_brick_idx_delta[i], mà biểu diễn thông tin chênh lệch với lát trước đó có thể được bỏ qua đối với lát thứ nhất (tức là, lát mà chỉ số i là 0). Cú pháp, bottom_right_brick_idx_delta[i], mà biểu diễn độ chênh lệch chỉ số giữa ô trên cùng-bên trái và ô trên cùng-bên phải của lát có thể được mã hóa và báo hiệu đối với lát thứ nhất.

Bảng 14 là ví dụ mà thể hiện xử lý trong đó lát mà mỗi ô thuộc về đó được chỉ rõ bằng cách sử dụng thông tin chênh lệch.

【Bảng 14】

```

NumBricksInSlice[ i ] = 0
TopLeftBrickIdx[0] = 0
TopLeftBrickIdx[i] = top_left_brick_idx_delta[i] + TopLeftBrickIdx[i-1]
botRightBkIdx = TopLeftBrickIdx[i] + bottom_right_brick_idx_delta[ i ]
for( j = 0; j < NumBricksInPic; j++) {
    if( BrickColBd[ j ] >= BrickColBd[ TopLeftBrickIdx[i]] ) &&
        BrickColBd[ j ]                                     <=
BrickColBd[ botRightBkIdx ] &&
        BrickRowBd[ j ]                                     >=
BrickRowBd[ TopLeftBrickIdx [ i ] ] &&
        BrickRowBd[ j ]                                     <=
BrickColBd[ botRightBkIdx ] ) {
    NumBricksInSlice[ i ]++
    BricksToSliceMap[ j ] = i
}
}

```

NumBricksInSlice[i] biểu diễn số lượng ô được chứa bởi lát i.

`TopLeftBrickIdx[i]` biểu diễn chỉ số của ô trên cùng-bên trái của lát i . `botRightBkIdx` biểu diễn chỉ số của ô trên cùng-bên phải. `BrickColBd[j]` biểu diễn chỉ số của cột ô mà ô j thuộc về. `BrickRowBd[j]` biểu diễn chỉ số của hàng ô mà ô j thuộc về. `BricksToSliceMap[j] = i` biểu diễn rằng ô j được thêm vào lát i .

Thay vì cú pháp, `top_left_brick_idx_delta[i]`, cú pháp, `bottom_right_brick_idx_delta[i]`, mà biểu diễn độ chênh lệch chỉ số giữa ô dưới cùng-bên phải của lát thứ i - và ô dưới cùng-bên phải của lát trước đó (tức là, lát thứ $(i-1)$) có thể được báo hiệu trong dòng bit. Trong trường hợp này, chỉ số của ô dưới cùng-bên phải của lát thứ i có thể được thu nhận bằng cách cộng cú pháp, `bottom_right_brick_idx_delta[i]`, vào chỉ số của ô dưới cùng-bên phải của lát trước đó.

Đối với lát thứ nhất, giá trị của `bottom_right_brick_idx_delta[0]` có thể được thiết lập là tương tự như chỉ số của ô dưới cùng-bên phải của lát thứ nhất. Nói cách khác, chỉ số của ô dưới cùng-bên phải của lát thứ nhất có thể được thiết lập là tương tự như cú pháp, `bottom_right_brick_idx_delta[0]`.

Thông tin mà biểu diễn rằng chỉ số của ô được chứa trong lát thứ i trong ảnh có giá trị lớn hơn chỉ số của ô được chứa trong lát thứ $(i-1)$ hay không có thể được mã hóa và báo hiệu trong dòng bit. Theo ví dụ của sáng chế, cú pháp, `bottom_right_increasing_order_flag`, có thể được báo hiệu trong dòng bit. Khi cờ, `bottom_right_increasing_order_flag`, là 1, điều này biểu diễn rằng chỉ số của ô được chứa trong lát thứ i trong ảnh luôn có giá trị lớn hơn chỉ số của ô được chứa trong lát thứ $i-1$. Mặt khác, khi `bottom_right_increasing_order_flag` là 0, điều này biểu diễn rằng chỉ số của ô được chứa trong ít nhất một lát trong ảnh có thể có giá trị nhỏ hơn chỉ số của ô được chứa trong lát trước đó.

Lát có thể được xác định bằng cách sử dụng chỉ số của ô trên cùng-bên trái trong lát và thông tin mà chỉ rõ ít nhất một trong số độ rộng hoặc độ cao của lát. Theo ví dụ của sáng chế, ít nhất một trong số cú pháp, `slice_width_in_tiles_minus1[i]`, mà biểu diễn độ rộng của lát thứ i , hoặc cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn độ cao của lát thứ i có thể được báo hiệu trong dòng bit.

Cú pháp, `slice_width_in_tiles_minus1[i]`, biểu diễn giá trị mà trừ 1 từ số lượng cột ô được chứa trong lát thứ i . Cú pháp, `slice_height_in_tiles_minus1[i]`, biểu diễn giá trị mà trừ 1 từ số lượng hàng ô được chứa trong lát thứ i .

Lát thứ i có thể được cấu thành với số lượng cột ô bằng số lượng được xác định dựa trên cú pháp, `slice_width_in_tiles_minus1[i]`, và số lượng hàng ô bằng số lượng được xác định dựa trên cú pháp, `slice_height_in_tiles_minus1[i]`. Trong trường hợp này, ô trên cùng-bên trái của cột ô thứ i có thể có giá trị chỉ số được xác định dựa trên `top_left_brick_idx_delta[i]`.

Ngoài ra, cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn độ cao của lát có thể được báo hiệu chỉ đối với lát liền kề với biên bên trái của ảnh, và việc mã hóa của cú pháp, `slice_height_in_tiles_minus1[i]`, có thể được bỏ qua đối với lát khác. Độ cao của lát mà việc mã hóa của cú pháp, `slice_height_in_tiles_minus1[i]`, được bỏ qua có thể được thiết lập tương tự như của lát mà liền kề biên bên trái của ảnh hiện tại trong số các lát được chứa trong cùng hàng.

`TopLeftBrickIdx[0]`, chỉ số của ô trên cùng-bên trái của lát thứ nhất, có thể được thiết lập là 0. Do đó, đối với lát thứ hai (tức là, lát mà chỉ số i là 1), cú pháp, `top_left_brick_idx_delta[i]`, về bản chất có cùng giá trị như chỉ số của ô trên cùng-bên trái. Do đó, cú pháp, `top_left_brick_idx[i]`, thay vì cú pháp, `top_left_brick_idx_delta[i]`, có thể được báo hiệu đối với lát thứ hai. Trong trường hợp này, `top_left_brick_idx[i]` biểu diễn chỉ số của ô trên cùng-bên trái của lát thứ i .

Trong ví dụ khác, cú pháp, `top_left_brick_idx_delta[i]`, có thể được thiết lập là độ chênh lệch giữa ô thứ nhất của lát thứ i và ô thứ nhất của lát thứ $(i+1)$. Nói cách khác, chỉ số của ô trên cùng-bên trái của lát thứ $(i+1)$ có thể được thu nhận bằng cách tính tổng chỉ số của ô trên cùng-bên trái của lát thứ i và cú pháp, `top_left_brick_idx_delta[i]`, được báo hiệu đối với lát thứ i . Khi cú pháp, `top_left_brick_idx_delta[i]`, biểu diễn thông tin chênh lệch giữa lát thứ i và lát thứ $(i+1)$, việc mã hóa của cú pháp, `top_left_brick_idx_delta[i]`, có thể được bỏ qua đối với lát cuối cùng.

Lát có thể được xác định bằng cách sử dụng Bảng cú pháp như trong Bảng 15 sau đây.

【Bảng 15】

pic_parameter_set_rbsp() {	Mô tả
...	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
log2_pps_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
rect_slice_flag	u(1)
if(rect_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i++]	ue(v)
}	
if(tile_idx_delta_present_flag && i <	

num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
...	
}	
}	
...	
}	

Khi cờ, rect_slice_flag, mà biểu loại lát được mã hóa và giá trị của cờ, rect_slice_flag, là 1, cú pháp, num_slices_in_pic_minus1, mà biểu diễn số lượng lát trong ảnh có thể được mã hóa.

Khi được xác định rằng ảnh bao gồm nhiều lát, cú pháp, slice_width_in_tiles_minus1[i], mà biểu diễn độ rộng của mỗi lát, và cú pháp, slice_height_in_tiles_minus1[i], mà biểu diễn độ cao của mỗi lát có thể được mã hóa và báo hiệu.

Ngoài ra, cú pháp, tile_idx_delta[i], mà biểu diễn độ chênh lệch giữa chỉ số của ô được chứa trong lát thứ i và chỉ số của ô được chứa trong lát thứ (i+1) có thể được báo hiệu. Đối với ô cuối cùng, việc mã hóa của cú pháp, tile_idx_delta[i], có thể được bỏ qua.

Cú pháp được thu nhận bằng cách trừ độ dịch từ độ chênh lệch chỉ số ô giữa hai lát có thể được mã hóa. Độ dịch có thể là số tự nhiên như 1 hoặc 2. Theo ví dụ của sáng chế, cú pháp, tile_idx_delta_minus1[i], có thể được báo hiệu trong dòng bit. Độ chênh lệch giữa chỉ số của ô được chứa trong lát thứ i và chỉ số của ô được chứa trong lát thứ (i+1) có thể được thu nhận bằng cách cộng 1 vào cú pháp, tile_idx_delta_minus1[i].

Thông tin chênh lệch của chỉ số ô có thể được mã hóa/giải mã khi ảnh hiện tại bao gồm nhiều lát. Nói cách khác, khi ảnh hiện tại được cấu hình với một lát, báo hiệu của cú pháp, tile_idx_delta[i], mà biểu diễn độ chênh lệch của chỉ số ô, có thể được bỏ qua. Ví dụ, khi cờ, one_slice_in_pic_flag, mà biểu diễn rằng ảnh

hiện tại có được cấu hình với một lát hay không bằng 1, báo hiệu của cú pháp, `tile_idx_delta[i]`, có thể được bỏ qua.

Ngoài ra, bộ mã hóa có thể xác định rằng có mã hóa hay không cú pháp, `tile_idx_delta[i]`, mà biểu diễn độ chênh lệch của chỉ số ô và có thể mã hóa cờ, `tile_idx_delta_present_flag`, mà biểu diễn rằng cú pháp, `tile_idx_delta[i]`, có được mã hóa hay không theo việc xác định này. Khi giá trị của cờ, `tile_idx_delta_present_flag`, bằng 0, điều này biểu diễn rằng cú pháp, `tile_idx_delta[i]`, không được mã hóa/giải mã. Khi giá trị của cờ, `tile_idx_delta_present_flag`, bằng 1, cú pháp, `tile_idx_delta[i]`, có thể được mã hóa/giải mã đối với ít nhất một lát.

Một ô có thể được phân chia thành nhiều lát. Theo ví dụ của sáng chế, các lát có thể được tạo ra bằng cách phân chia ô trong chiều ngang.

Khi được xác định rằng lát không bao gồm nhiều ô, việc ô có được phân chia thành nhiều lát có thể được xác định. Theo ví dụ của sáng chế, khi cả cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, là 0, điều này biểu diễn rằng lát được cấu thành với chỉ một ô, hoặc ô được phân chia thành nhiều lát.

Khi cả cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, bằng 0, thông tin mà biểu diễn rằng ô có được phân chia thành nhiều lát hay không có thể được báo hiệu.

Theo ví dụ của sáng chế, cú pháp, `num_slices_in_tile_minus1[i]`, mà biểu diễn số lượng lát được chứa bởi ô có thể được báo hiệu. Cú pháp, `num_slices_in_tile_minus1[i]`, biểu diễn giá trị mà trừ 1 từ số lượng lát được chứa bởi lát thứ *i*.

Khi ô bao gồm lát thứ *i* được cấu hình với nhiều lát, cú pháp mà biểu diễn độ cao của mỗi lát trong ô có thể được mã hóa. Theo ví dụ của sáng chế, cú pháp, `slice_height_in_ctu_minus1[i][j]`, mà biểu diễn độ cao của lát thứ *j* trong ô bao gồm lát thứ *i* (tức là, lát thứ $(i+j)$), có thể được báo hiệu. Cú pháp, `slice_height_in_ctu_minus1[i][j]`, có thể biểu diễn số lượng hàng đơn vị cây mã hóa được chứa bởi lát thứ *j* trong ô bao gồm lát thứ *i*. Bảng 16 minh họa cấu trúc cú pháp mà bao gồm cú pháp, `slice_height_in_ctu_minus1[i][j]`, mà biểu diễn độ cao của lát.

【Bảng 16】

pic_parameter_set_rbsp() {	Mô tả
...	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
log2_pps_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
rect_slice_flag	u(1)
if(rect_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
if(RowHeight[TileIdx[i] / NumTileColumns] > 1)	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++){	
if(RowHeight[TileIdx[i] / NumTileColumns] > 2)	

slice_height_in_ctu_minus1[i][j]	ue(v)
}	
}	
if (slice_width_in_tiles_minus1[i] != 0 slice_height_in_tiles_minus1[i] != 0) {	
tile_idx_delta[i]	se(v)
}	
}	
signalled_slice_id_flag	u(1)
if(signalled_slice_id_flag) {	
signalled_slice_id_length_minus1	ue(v)
for(i = 0; i <= num_slices_in_pic_minus1; i++)	
slice_id[i]	u(v)
}	
}	
}	
...	
}	

Trong trường hợp này, dựa trên độ cao của ô, việc cú pháp mà biểu diễn số lượng lát trong ô có được mã hóa/giải mã hay không có thể được xác định. Theo ví dụ của sáng chế, khi ô bao gồm lát thứ i được cấu hình với một hàng đơn vị cây mã hóa, việc mã hóa/giải mã của cú pháp, num_slices_in_tile_minus1, mà biểu diễn số lượng lát trong ô có thể được bỏ qua. Khi việc mã hóa/giải mã của cú pháp, num_slices_in_tile_minus1, được bỏ qua, giá trị của nó có thể được giả định là 0. Nói cách khác, lát thứ i và ô bao gồm lát thứ i có thể được thiết lập để có cùng kích cỡ và dạng. Trong Bảng 16, biến, TileIdx[i], chỉ báo chỉ số của ô bao gồm lát thứ i và biến, NumTileColumn, biểu diễn số lượng cột ô. TileIdx[i]/NumTileColumn trả về hàng mà ô bao gồm lát thứ i được bao gồm. Biến, RowHeight[TileIdx[i]/NumTileColumn], biểu diễn độ cao của hàng ô hoặc

ô bao gồm lát thứ i .

Trong trường hợp này, do các ô thuộc về cùng hàng ô có cùng độ cao, có thể được hiểu rằng việc cú pháp mà biểu diễn số lượng lát trong ô có được mã hóa/giải mã hay không được xác định dựa trên độ cao của hàng ô.

Dựa trên độ cao của ô hoặc hàng ô, việc cú pháp mà biểu diễn độ cao của lát có được mã hóa/giải mã hay không có thể được xác định. Theo ví dụ của sáng chế, khi ô bao gồm lát thứ i được cấu hình với một hàng đơn vị cây mã hóa, hoặc khi ô bao gồm lát thứ i được cấu hình với hai hàng đơn vị cây mã hóa, việc mã hóa/giải mã của cú pháp, `slice_height_in_ctu_minus1[i][j]`, mà biểu diễn độ cao của lát có thể được bỏ qua. Giá trị của cú pháp, `slice_height_in_ctu_minus1[i][j]`, có thể được giả định thích nghi theo số lượng lát trong ô. Theo ví dụ của sáng chế, khi số lượng lát trong ô là 1, độ cao của lát có thể được thiết lập tương tự như độ cao của ô. Ngoài ra, khi số lượng lát trong ô là 2 và độ cao của ô là 2, độ cao của mỗi lát có thể được thiết lập là 1. Trong Bảng 16, biến, `TileIdx[i]`, chỉ báo chỉ số của ô bao gồm lát thứ i và biến, `NumTileColumn`, biểu diễn số lượng cột ô. `TileIdx[i]/NumTileColumn` trả về hàng mà ô bao gồm lát thứ i được bao gồm. Biến, `RowHeight[TileIdx[i]/NumTileColumn]`, biểu diễn độ cao của ô bao gồm lát thứ i .

Thay vì cú pháp, `num_slices_in_tile_minus1`, mà biểu diễn số lượng lát nêu trên, cú pháp, `num_exp_slices_in_tile[i]`, mà biểu diễn thông tin biểu diễn số lượng độ cao lát mà được mã hóa rõ ràng, có thể được mã hóa. Cú pháp, `num_exp_slices_in_tile[i]`, có thể có giá trị mà tương tự hoặc nhỏ hơn số lượng lát được chứa bởi ô.

Khi cú pháp, `num_exp_slices_in_tile[i]`, là 0, điều này biểu diễn rằng ô không được phân chia thành các lát. Khi cú pháp, `num_exp_slices_in_tile[i]`, lớn hơn 0, điều này biểu diễn rằng ô có thể được phân chia thành các lát.

Thông tin mà biểu diễn độ cao của lát có thể được báo hiệu với số lượng bằng giá trị được chỉ báo bởi cú pháp, `num_exp_slices_in_tile[i]`. Theo ví dụ của sáng chế, khi cú pháp, `num_exp_slices_in_tile[i]`, lớn hơn 1, cú pháp, `exp_slice_height_in_ctu_minus1[j]`, mà biểu diễn độ cao của lát thứ j trong ô, có thể được báo hiệu.

Như trong phương án nêu trên, việc có mã hóa hay không thông tin số

lượng, `exp_slice_height_in_ctu_minus1[j]`, có thể được xác định dựa trên độ cao của ô hoặc hàng ô. Theo ví dụ của sáng chế, khi ô bao gồm lát thứ i được cấu hình với một hàng đơn vị cây mã hóa, việc mã hóa/giải mã của cú pháp, `exp_slice_height_in_ctu_minus1[j]`, có thể được bỏ qua. Khi việc mã hóa/giải mã của cú pháp, `exp_slice_height_in_ctu_minus1[j]`, được bỏ qua, giá trị của nó có thể được giả định bằng 0.

Khi chỉ số j của lát nhỏ hơn số lượng lát mà độ cao của nó được báo hiệu rõ ràng, độ rộng của lát j có thể được xác định dựa trên cú pháp, `exp_slice_height_in_ctu_minus1[j]`, được báo hiệu trong dòng bit. Mặt khác, khi chỉ số k của lát bằng hoặc lớn hơn số lượng lát mà độ cao của nó được báo hiệu rõ ràng, độ cao của lát k có thể được xác định dựa trên cú pháp, `exp_slice_height_in_ctu_minus1[l]`, mà được báo hiệu cuối cùng. Trong trường hợp này, l có thể biểu diễn chỉ số của lát mà độ cao của nó được báo hiệu cuối cùng và có thể là số nguyên nhỏ hơn k .

Theo ví dụ của sáng chế, khi giá trị mà trừ độ cao của các lát trước đó từ độ cao của ô bằng hoặc lớn hơn giá trị mà cộng 1 vào cú pháp, `exp_slice_height_in_ctu_minus1[l]`, độ cao của lát k có thể được thiết lập là giá trị mà cộng 1 vào cú pháp, `exp_slice_height_in_ctu_minus1[l]`. Mặt khác, khi giá trị mà trừ độ cao của các lát trước đó từ độ cao của ô nhỏ hơn giá trị mà cộng 1 vào cú pháp, `exp_slice_height_in_ctu_minus1[l]`, giá trị chênh lệch mà trừ độ cao của các lát trước đó từ độ cao của ô có thể được thiết lập là độ cao của lát k .

Nói cách khác, độ cao của các lát còn lại ngoại trừ các lát mà độ cao của nó được báo hiệu rõ ràng có thể có giá trị nhỏ hơn hoặc tương tự như độ cao của lát cuối cùng của các lát mà độ cao của nó được báo hiệu rõ ràng.

Trong ví dụ nêu trên, được minh họa rằng cú pháp, `one_slice_in_pic_flag`, mà biểu diễn ảnh có được cấu hình với một lát hay không có được mã hóa hay không được xác định chỉ khi phương pháp xác định lát chữ nhật được áp dụng. Ngược với ví dụ, việc cú pháp, `rect_slice_flag`, có được mã hóa hay không có thể được xác định theo giá trị của cờ, `one_slice_in_pic_flag`. Theo ví dụ của sáng chế, khi giá trị của `one_slice_in_pic_flag` bằng 1, việc mã hóa của `rect_slice_flag` có thể được bỏ qua.

Khi số lượng cột ô trong ảnh hiện tại là 1, việc mã hóa của cú pháp, `slice_width_in_tiles_minus1`, có thể được bỏ qua. Ngoài ra, khi số lượng hàng ô

trong ảnh hiện tại là 1, việc mã hóa của cú pháp, `slice_height_in_tiles_minus1`, có thể được bỏ qua.

Ngoài ra, việc mã hóa của ít nhất một trong số thông tin mà biểu diễn độ rộng của lát hoặc thông tin mà biểu diễn độ cao của lát có thể được bỏ qua.

Theo ví dụ của sáng chế, việc mã hóa của cú pháp, `slice_width_in_tiles_minus1`, mà biểu diễn độ rộng của lát có thể được bỏ qua và khoảng cách với ô tại vị trí định trước trong lát liền kề có thể được thiết lập là độ rộng của lát. Cụ thể, ô trên cùng-bên trái của lát thứ $(i+1)$ liền kề với bên phải của lát thứ i có thể được chỉ rõ bởi cú pháp, `top_left_brick_idx_delta[i]`. Độ rộng của lát thứ i có thể được thu nhận bởi độ chênh lệch giữa tọa độ x của ô trên cùng-bên trái trong lát thứ i (ví dụ, tọa độ x của mẫu trên cùng-bên trái) và tọa độ x của ô trên cùng-bên trái trong lát thứ $(i+1)$ (ví dụ, tọa độ x của mẫu trên cùng-bên trái).

Ngoài ra, việc mã hóa của cú pháp, `slice_height_in_tiles_minus1`, mà biểu diễn độ cao của lát có thể được bỏ qua và khoảng cách với ô tại vị trí trong lát liền kề có thể được thiết lập là độ rộng của lát. Cụ thể, ô trên cùng-bên trái của lát thứ j có vị trí tại dưới cùng của lát thứ i có thể được chỉ rõ bởi cú pháp, `top_left_brick_idx_delta[j-1]`. Độ cao của lát thứ i có thể được thu nhận bởi độ chênh lệch giữa tọa độ y của ô trên cùng-bên trái trong lát thứ i (ví dụ, tọa độ y của mẫu trên cùng-bên trái) và tọa độ y của ô trên cùng-bên trái trong lát thứ j (ví dụ, tọa độ y của mẫu trên cùng-bên trái).

Ngoài ra, trong việc xác định lát trong dạng chữ nhật, thông tin mà biểu diễn rằng thông tin giá trị chênh lệch có được sử dụng hay không có thể được báo hiệu trong dòng bit. Theo ví dụ của sáng chế, `tile_idx_delta_present_flag` mà biểu diễn rằng thông tin giá trị chênh lệch có được sử dụng hay không có thể được báo hiệu trong dòng bit. Khi giá trị của cú pháp, `tile_idx_delta_present_flag`, là 1, điều này biểu diễn rằng cú pháp mà biểu diễn giá trị chênh lệch của chỉ số ô được mã hóa và được báo hiệu. Theo ví dụ của sáng chế, khi giá trị của cú pháp, `tile_idx_delta_present_flag`, là 1, lát thứ i có thể được xác định bởi cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn kích cỡ của lát và thông tin giá trị chênh lệch để xác định vị trí của ô trên cùng-bên trái hoặc vị trí của ô trên cùng-bên phải trong lát (ví dụ, `top_left_brick_idx_delta[i-1]` hoặc `bottom_right_brick_idx_delta[i]`).

Khi giá trị của cú pháp, `tile_idx_delta_present_flag`, là 0, lát thứ i có thể

được xác định bởi cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn kích cỡ của lát. Khi giá trị của cú pháp, `tile_idx_delta_present_flag`, là 0, các lát tại vị trí bên phải của lát liền kề với biên bên trái của ảnh hiện tại có thể có cùng độ cao. Do đó, thông tin mà biểu diễn độ cao của lát có thể được báo hiệu chỉ đối với lát mà liền kề với biên bên trái của ảnh hiện tại và báo hiệu của thông tin mà biểu diễn độ cao của lát có thể được bỏ qua đối với các lát mà không liền kề với biên bên trái của ảnh hiện tại.

Cú pháp, `tile_idx_delta_present_flag`, mà biểu diễn rằng thông tin mà biểu diễn độ chênh lệch của chỉ số ô có được mã hóa hay không có thể được mã hóa/giải mã khi số lượng lát được chứa trong ảnh hiện tại bằng hoặc lớn hơn giá trị ngưỡng. Trong trường hợp này, giá trị ngưỡng có thể được thiết lập là số nguyên như 0, 1, hoặc 2, v.v. Theo ví dụ của sáng chế, chỉ khi số lượng lát được chứa trong ảnh hiện tại bằng hoặc lớn hơn 1 hoặc 2, cú pháp, `tile_idx_delta_present_flag`, có thể được mã hóa/giải mã. Khi số lượng lát được chứa trong ảnh hiện tại nhỏ hơn giá trị ngưỡng, việc mã hóa/giải mã của `tile_idx_delta_present_flag` có thể được bỏ qua và giá trị của nó có thể được giả định bằng 0. Có thể xác định lát dựa trên chỉ số của đơn vị cây mã hóa thay vì chỉ số của ô.

Trong ví dụ nêu trên, giả thiết rằng lát được xác định bằng cách đưa thứ tự tới các ô theo thứ tự quét màn hình.

Khi ô được cấu hình với nhiều lát, thông tin mà biểu diễn kích cỡ của lát và/hoặc thông tin chênh lệch chỉ số ô có thể được báo hiệu chỉ đối với lát thứ nhất. Trong trường hợp này, thông tin mà biểu diễn kích cỡ của lát có thể bao gồm ít nhất một trong số cú pháp, `slice_width_in_tiles_minus1[i]`, mà biểu diễn độ rộng của lát, và cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn độ cao của lát.

Bảng 17 biểu diễn ví dụ cho việc này.

【Bảng 17】

<code>pic_parameter_set_rbsp() {</code>	Mô tả
<code>...</code>	
<code>single_tile_in_pic_flag</code>	u(1)
<code>if(!single_tile_in_pic_flag) {</code>	
<code>log2_pps_ctu_size_minus5</code>	u(2)

num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
rect_slice_flag	u(1)
if(rect_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
isFirstSliceInTile[i] = 1 ;	
isFirstSliceInTile[num_slices_in_pic_minus1] = 0 ;	
isFirstSliceInTile[num_slice_in_pic_minus1] = 0 ;	
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if (isFirstSliceInTile[i]){	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
}	
if((slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) && isFirstSliceInTile[i]) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++){	
slice_height_in_ctu_minus1[i+j]	ue(v)
if (j>0)	
isFirstSliceInTile[i++] = 0 ;	
}	
}	
if (slice_width_in_tiles_minus1[i] != 0 slice_height_in_tiles_minus1[i] != 0 isFirstSliceInTile[i]) {	
tile_idx_delta[i]	se(v)

}	
}	
...	
}	
}	
...	
}	

Biến, `isFirstSliceInTile[i]`, mà biểu diễn rằng lát có phải là lát thứ nhất hay không trong ô có thể được thiết lập. Khi giá trị của biến, `isFirstSliceInTile[i]`, là 1, điều này biểu diễn rằng lát là lát thứ nhất trong ô. Mặt khác, khi giá trị của biến, `isFirstSliceInTile[i]`, là 0, điều này biểu diễn rằng lát không phải lát thứ nhất trong ô.

Biến, `isFirstSliceInTile[i]`, có thể được khởi tạo bằng 1. Do đó, khi lát thứ i được cấu hình với ít nhất một hoặc nhiều ô, không phải một phần ô, giá trị của biến, `isFirstSliceInTile[i]`, đối với ô thứ i , có thể được thiết lập là 1.

Khi giá trị của biến, `isFirstSliceInTile[i]`, bằng 1, cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn kích cỡ của lát có thể được mã hóa/giải mã.

Trong trường hợp này, khi cả cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, là 0, thông tin số lượng mà biểu diễn số lượng lát trong ô có thể được báo hiệu. Khi các lát được chứa trong ô, thông tin độ cao đối với ít nhất một trong số các lát có thể được mã hóa/giải mã. Trong trường hợp này, sau khi giải mã thông tin độ cao của lát thứ hai, giá trị của biến, `isFirstTileInTile[i++]`, có thể được biến đổi thành 0.

Do đó, từ lát thứ hai, việc mã hóa của cú pháp, `slice_width_in_tiles_minus1[i]`, mà biểu diễn độ rộng của lát, và cú pháp, `slice_height_in_tiles_minus1[i]`, mà biểu diễn độ cao của lát có thể được bỏ qua.

Cú pháp, `tile_idx_delta[i]`, mà biểu diễn độ chênh lệch chỉ số ô có thể được mã hóa/giải mã chỉ đối với ô thứ nhất. Theo ví dụ của sáng chế, cú pháp, `tile_idx_delta[i]`, có thể được báo hiệu đối với ít nhất một trong số trường hợp khi cú pháp, `slice_width_in_tiles_minus1[i]`, không phải 0, trường hợp khi cú pháp,

`slice_height_in_tiles_minus1[i]`, không phải 0 hoặc trường hợp khi biến, `isFirstSliceInTile[i]`, bằng 1.

Khi một ô được phân chia thành nhiều lát (tức là, khi cả cú pháp, `slice_width_in_tiles_minus1[i]`, và cú pháp, `slice_height_in_tiles_minus1[i]`, là 0), giá trị của biến, `isFirstSliceInTile`, là 1 đối với chỉ lát thứ nhất và giá trị của biến, `isFirstSliceInTile`, là 0 từ lát thứ hai. Do đó, cú pháp, `tile_idx_delta[i]`, có thể được mã hóa chỉ đối với lát thứ nhất. Khi ô cuối cùng trong ảnh được cấu hình với nhiều lát, `tile_idx_delta[i]` có thể không được báo hiệu.

Việc áp dụng các phương án như được mô tả về xử lý giải mã hoặc xử lý mã hóa tới xử lý mã hóa hoặc xử lý giải mã một cách tương ứng có thể được chứa trong phạm vi của sáng chế. Trong phạm vi của sáng chế, các phương án trong đó các hoạt động diễn ra trong thứ tự định trước có thể được cải biến thành phương án trong đó các hoạt động diễn ra trong thứ tự khác với thứ tự định trước.

Mặc dù phương án nêu trên được mô tả dựa trên chuỗi hoạt động hoặc lưu đồ, phương án này không giới hạn thứ tự theo trình tự thời gian của các hoạt động của phương pháp này. Trong ví dụ khác, các hoạt động có thể được thực hiện một cách đồng thời hoặc trong thứ tự khác nếu cần thiết. Ngoài ra, trong phương án nêu trên, mỗi bộ phận (ví dụ, đơn vị, môđun, v.v) mà cấu thành sơ đồ khối có thể được thực hiện dưới dạng của thiết bị phần cứng hoặc phần mềm. Các bộ phận có thể được kết hợp với nhau vào một bộ phận mà có thể được thực hiện nhờ sử dụng một thiết bị phần cứng hoặc phần mềm. Phương án được mô tả nêu trên có thể được thực hiện nhờ sử dụng các lệnh chương trình mà có thể được thực thi thông qua các bộ phận máy tính khác nhau. Các lệnh có thể được ghi trong phương tiện lưu trữ đọc được bởi máy tính. Phương tiện lưu trữ đọc được bởi máy tính có thể chứa trong đó các lệnh chương trình, các tệp dữ liệu, các cấu trúc dữ liệu, hoặc loại tương tự một cách độc lập hoặc kết hợp với nhau. Các ví dụ của các phương tiện lưu trữ đọc được bởi máy tính bao gồm phương tiện từ tính như các đĩa cứng, đĩa mềm (floppy), và các băng từ, phương tiện lưu trữ quang như các CD-ROM, DVD, và phương tiện quang từ như các đĩa quang mềm, và các thiết bị phần cứng như ROM, RAM, bộ nhớ chớp, và loại tương tự có cấu trúc cụ thể để lưu trữ trong đó và thực thi các lệnh chương trình. Thiết bị phần cứng có thể có cấu trúc để hoạt động như là một hoặc nhiều môđun phần mềm để thực hiện xử lý theo sáng chế, và ngược lại.

Khả năng áp dụng công nghiệp

Sáng chế có thể được áp dụng tới thiết bị điện tử mà mã hóa/giải mã video.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã video, phương pháp này bao gồm:

phân chia ảnh hiện tại thành nhiều ô;

giải mã thông tin phân chia mà biểu diễn loại lát, trong đó thông tin phân chia biểu diễn rằng lát chữ nhật có được áp dụng hay không; và

giải mã thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi thông tin phân chia biểu diễn rằng lát chữ nhật được áp dụng,

trong đó khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được phân tích hay không được xác định dựa trên độ cao của ô trong đó lát được bao gồm,

trong đó việc phân tích của thông tin số lượng được bỏ qua khi ô được cấu hình với một hàng đơn vị cây mã hóa,

trong đó khi ô được cấu hình với nhiều hàng đơn vị cây mã hóa, thông tin số lượng được phân tích, và

trong đó thông tin số lượng biểu diễn số lượng lát mà thông tin độ cao của lát được báo hiệu rõ ràng.

2. Phương pháp giải mã video, phương pháp này bao gồm:

phân chia ảnh hiện tại thành nhiều ô;

giải mã thông tin phân chia mà biểu diễn loại lát, trong đó thông tin phân chia biểu diễn rằng lát chữ nhật có được áp dụng hay không; và

giải mã thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi thông tin phân chia biểu diễn rằng lát chữ nhật được áp dụng,

trong đó khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được phân tích hay không được xác định dựa trên độ cao của ô trong đó lát được bao gồm,

trong đó việc phân tích của thông tin số lượng được bỏ qua khi ô được cấu hình với một hàng đơn vị cây mã hóa,

trong đó khi ô được cấu hình với nhiều hàng đơn vị cây mã hóa, thông tin số lượng được phân tích, và

trong đó thông tin số lượng biểu diễn số lượng lát mà thông tin độ cao của lát được báo hiệu rõ ràng,

trong đó thông tin độ cao thứ hai được phân tích đối với lát thứ nhất mà chỉ số của nó nhỏ hơn số lượng được chỉ báo bởi thông tin số lượng trong số ít nhất một lát được chứa trong ô.

3. Phương pháp theo điểm 2, trong đó thông tin độ cao thứ hai không được phân tích đối với lát thứ hai mà chỉ số của lát này là tương tự hoặc lớn hơn số lượng được chỉ báo bởi thông tin số lượng.

4. Phương pháp theo điểm 3, trong đó độ cao của lát thứ hai được thiết lập là giá trị nhỏ hơn của độ cao của lát thứ ba mà thông tin độ cao thứ hai được phân tích cuối cùng hoặc độ cao còn lại trong ô.

5. Phương pháp mã hóa video, phương pháp này bao gồm:

phân chia ảnh hiện tại thành nhiều ô;

xác định rằng lát chữ nhật có được áp dụng hay không; và

mã hóa thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi lát chữ nhật được áp dụng,

trong đó khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được mã hóa hay không được xác định dựa trên độ cao của ô trong đó lát được bao gồm,

trong đó khi ô được cấu hình với một hàng đơn vị cây mã hóa, việc mã hóa của thông tin số lượng được bỏ qua,

trong đó khi ô được cấu hình với nhiều hàng đơn vị cây mã hóa, thông tin số lượng được mã hóa, và

thông tin số lượng biểu diễn số lượng lát mà thông tin độ cao của lát được báo hiệu rõ ràng.

6. Phương pháp mã hóa video, phương pháp này bao gồm:

phân chia ảnh hiện tại thành nhiều ô;

xác định rằng lát chữ nhật có được áp dụng hay không; và

giải mã thông tin độ rộng thứ nhất và thông tin độ cao thứ nhất của lát khi lát chữ nhật được áp dụng,

trong đó khi thông tin độ rộng thứ nhất chỉ báo một cột ô và thông tin độ cao thứ nhất chỉ báo một hàng ô, việc thông tin số lượng có được mã hóa hay không được xác định dựa trên độ cao của ô trong đó lát được bao gồm,

trong đó khi ô được cấu hình với một hàng đơn vị cây mã hóa, mã hóa thông tin số lượng được bỏ qua,

trong đó khi ô được cấu hình với nhiều hàng đơn vị cây mã hóa, thông tin số lượng được mã hóa, và

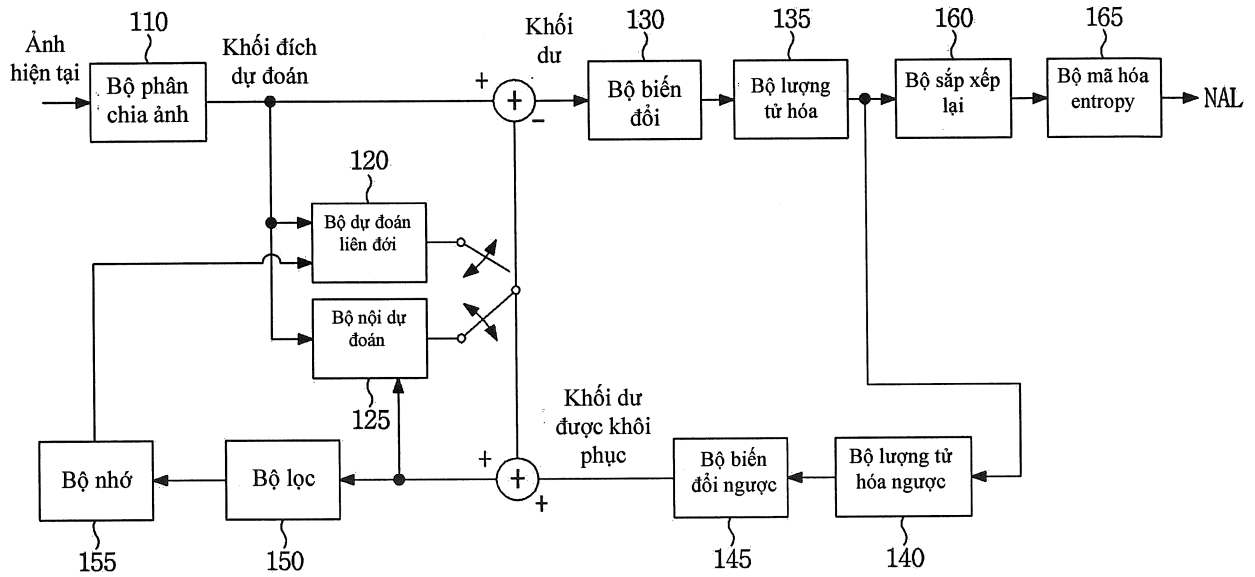
trong đó thông tin số lượng biểu diễn số lượng lát mà thông tin độ cao của lát được báo hiệu rõ ràng, và

trong đó thông tin độ cao thứ hai được mã hóa đối với lát thứ nhất mà chỉ số của nó nhỏ hơn số lượng được chỉ báo bởi thông tin số lượng trong số ít nhất một lát được chứa trong ô.

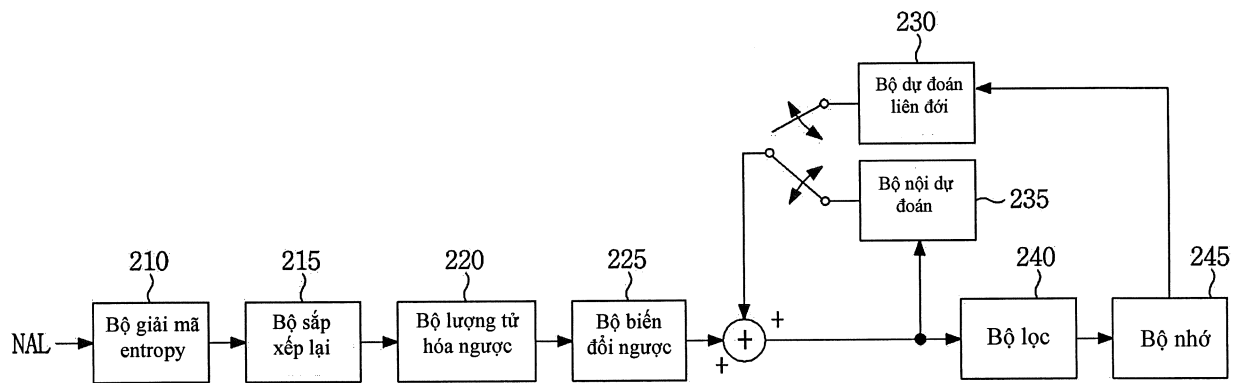
7. Phương pháp theo điểm 6, trong đó thông tin độ cao thứ hai không được mã hóa đối với lát thứ hai mà chỉ số của lát này là tương tự hoặc lớn hơn số lượng được chỉ báo bởi thông tin số lượng.

8. Phương pháp theo điểm 7, trong đó độ cao của lát thứ hai có giá trị nhỏ hơn độ cao của lát thứ ba mà thông tin độ cao thứ hai được phân tích cuối cùng hoặc độ cao còn lại trong ô.

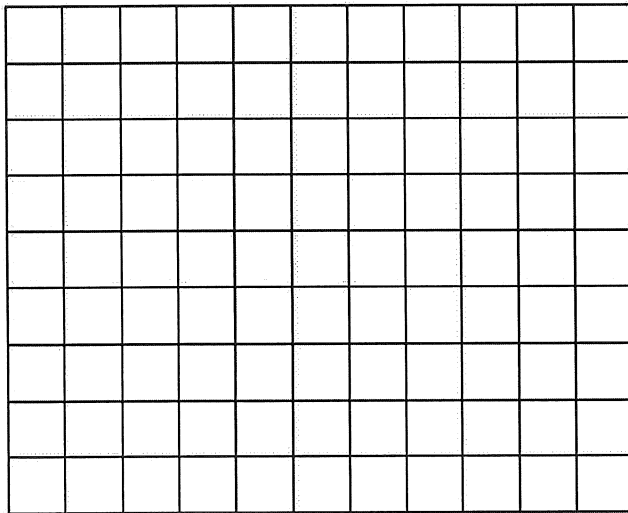
【FIG. 1】



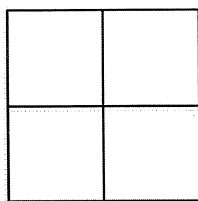
【FIG. 2】



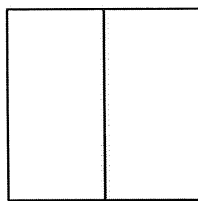
【FIG. 3】



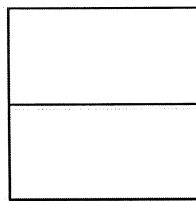
【FIG. 4】



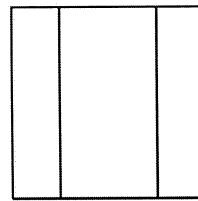
SPLIT_QT



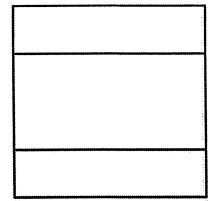
SPLIT_BT_VER



SPLIT_BT_HOR

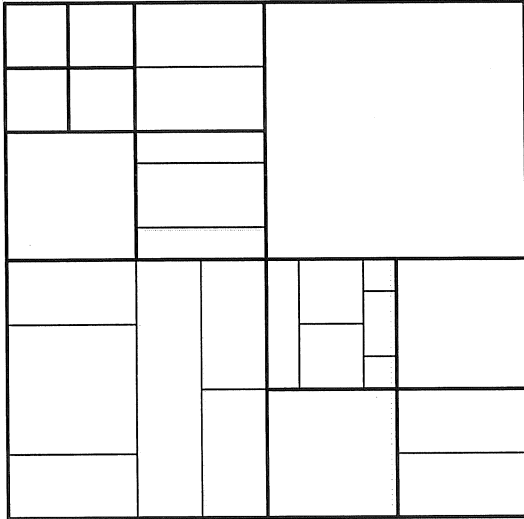


SPLIT_TT_VER

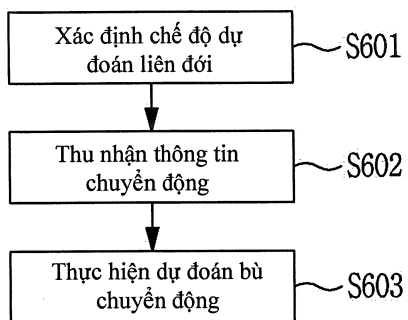


SPLIT_TT_HOR

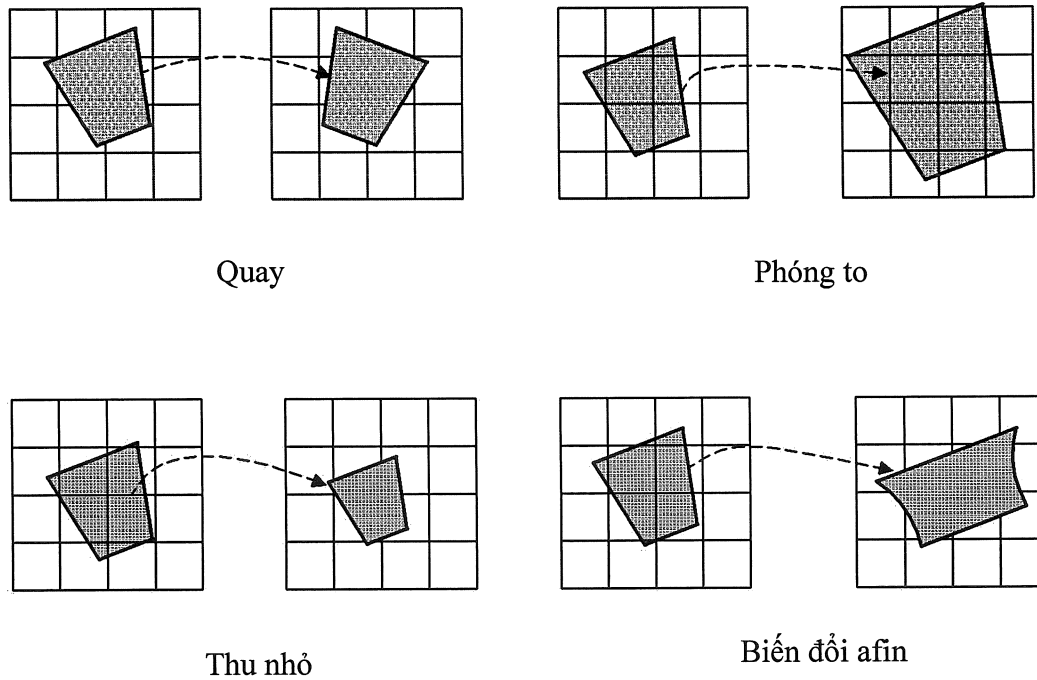
【FIG. 5】



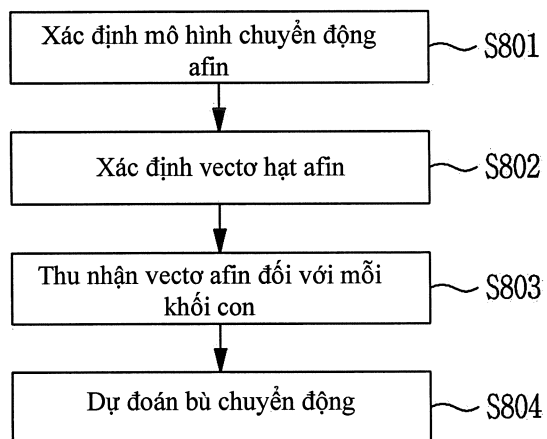
【FIG. 6】



【FIG. 7】

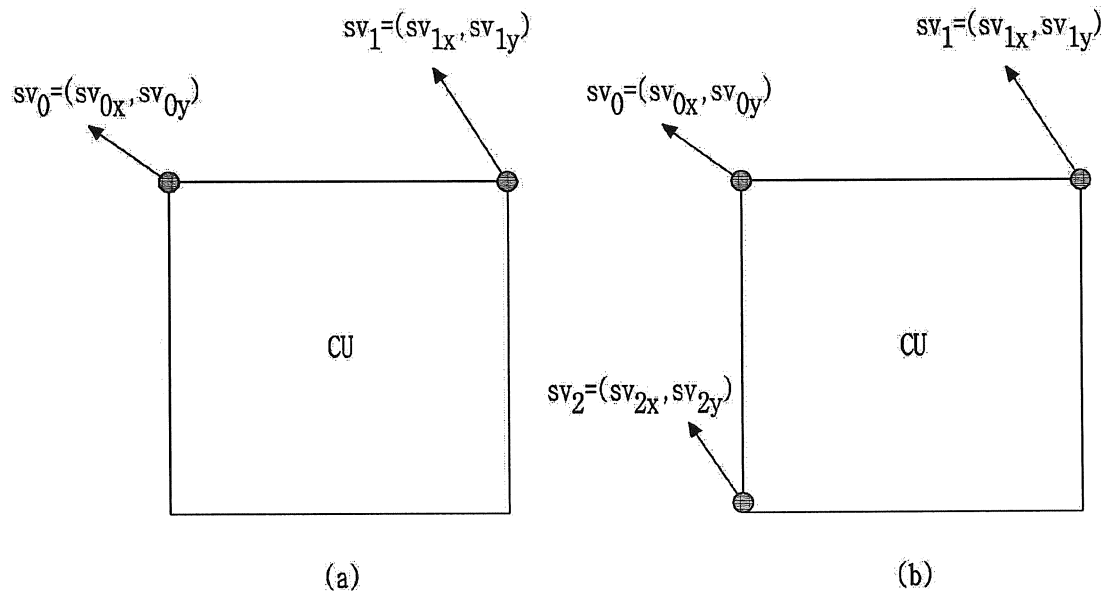


【FIG. 8】

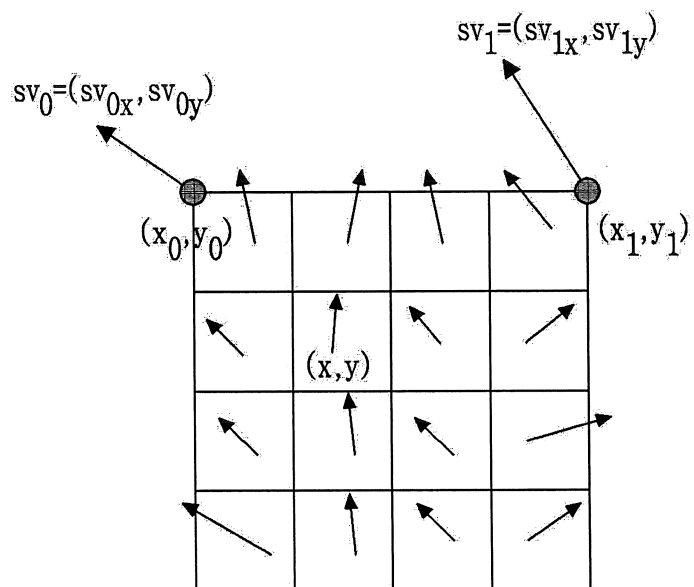


5/23

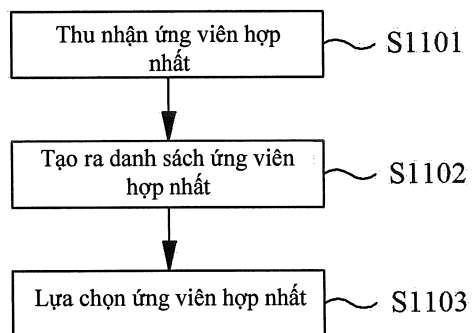
【FIG. 9】



【FIG. 10】



【FIG. 11】

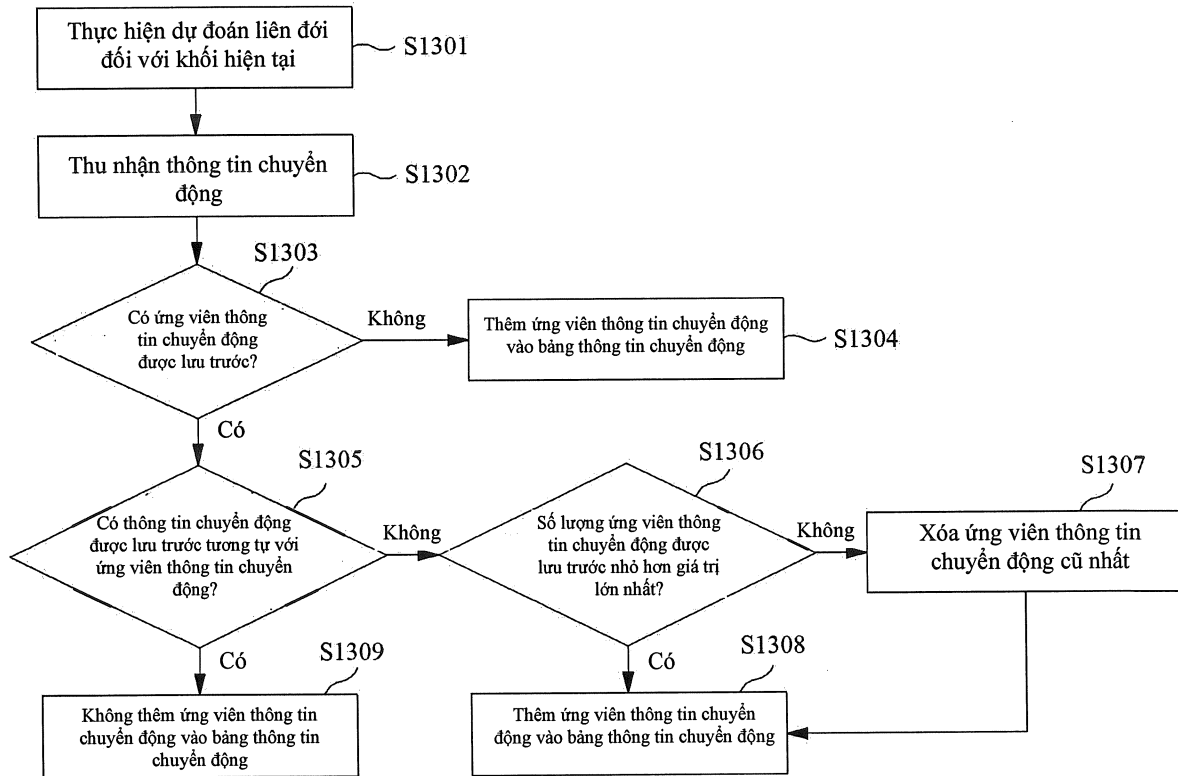


7/23

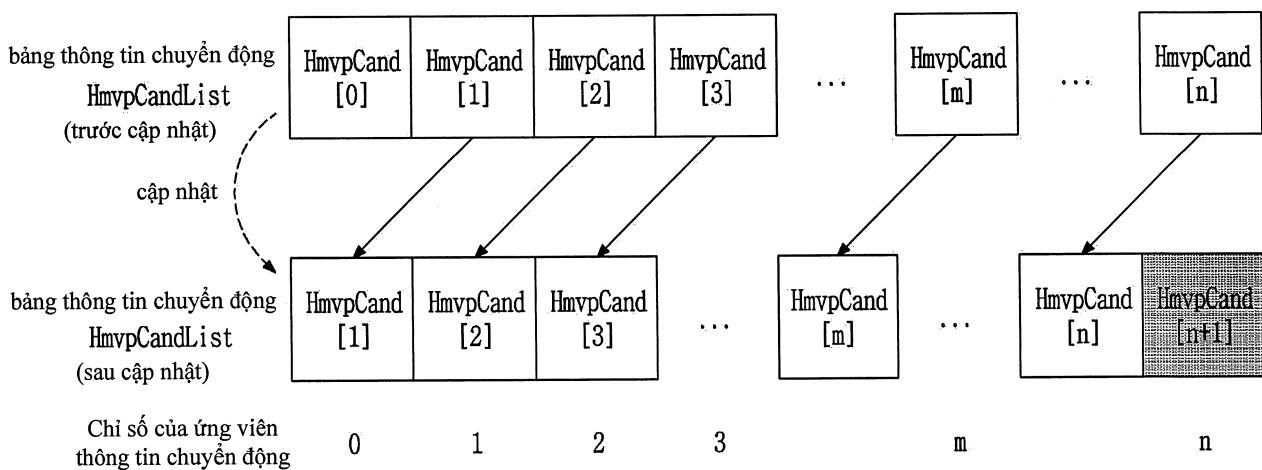
【FIG. 12】

26	22	18	15	16	20	24
23	13	9	6	7	12	
19	10	4	1	2		
14	5	0	Đơn vị mã hóa hiện tại			
17	8	3				
21	11					
25						

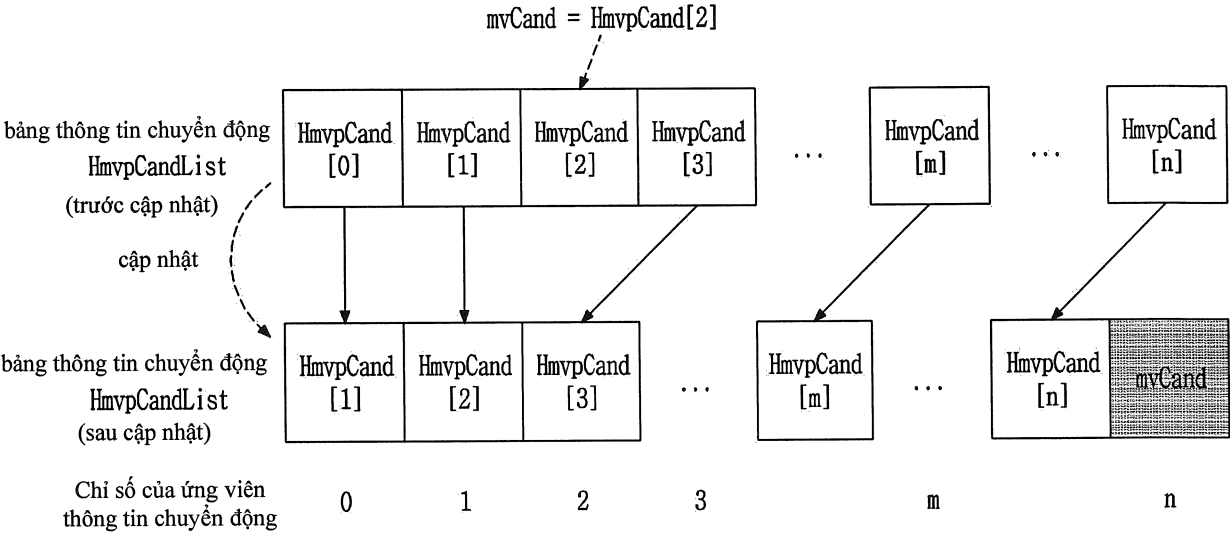
【FIG. 13】



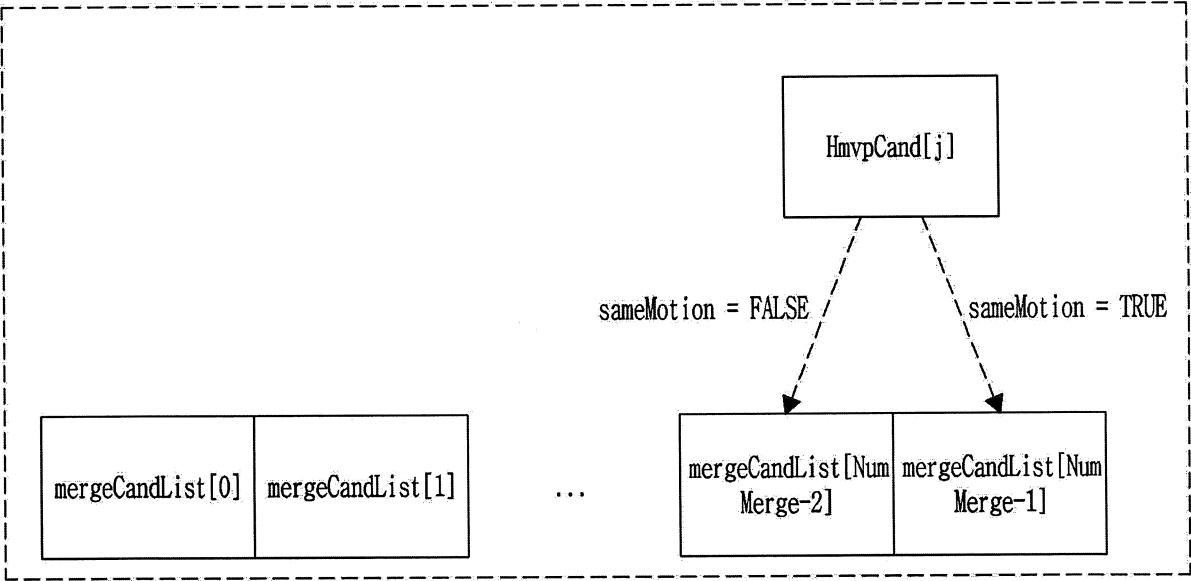
【FIG. 14】



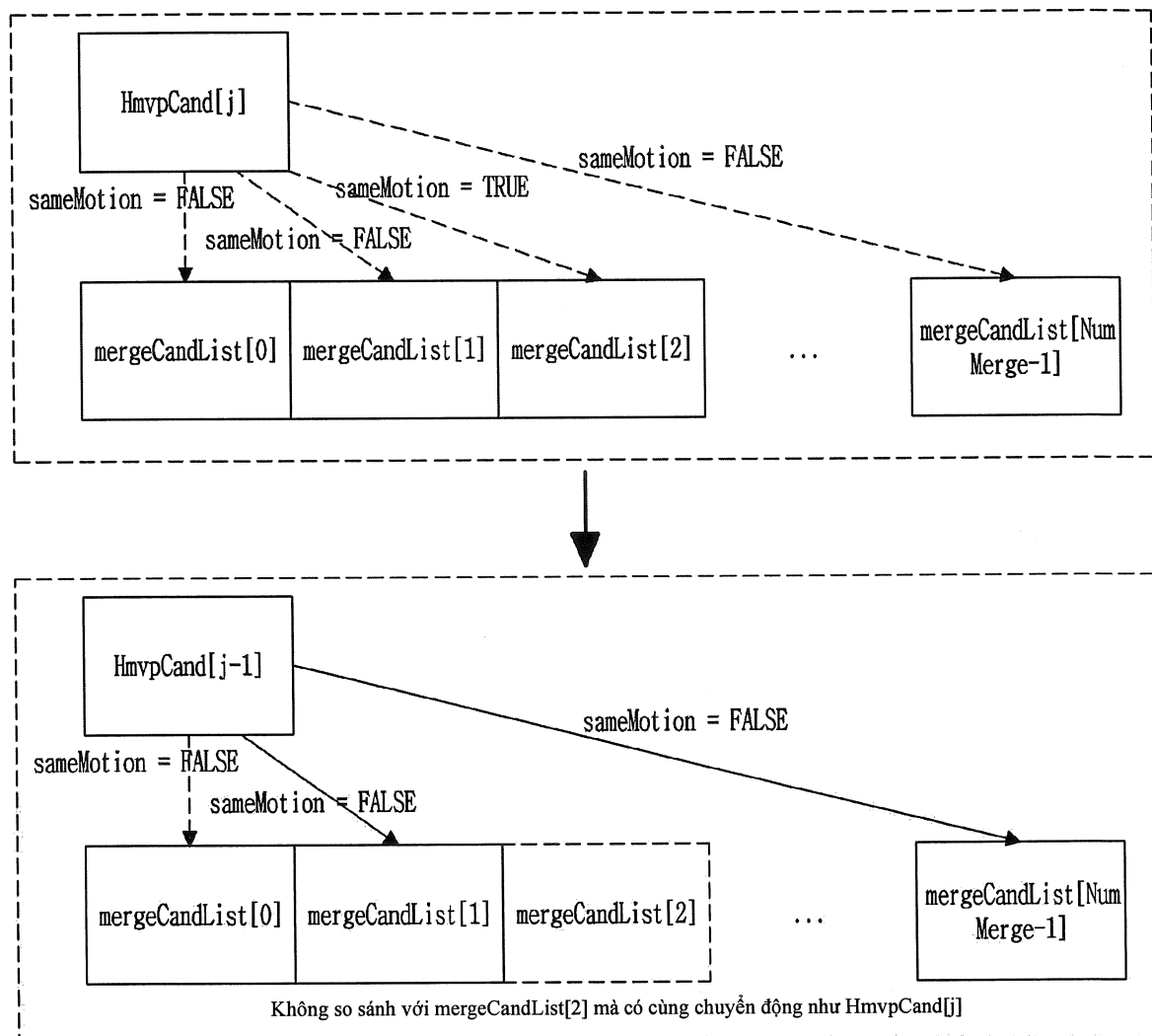
【FIG. 15】



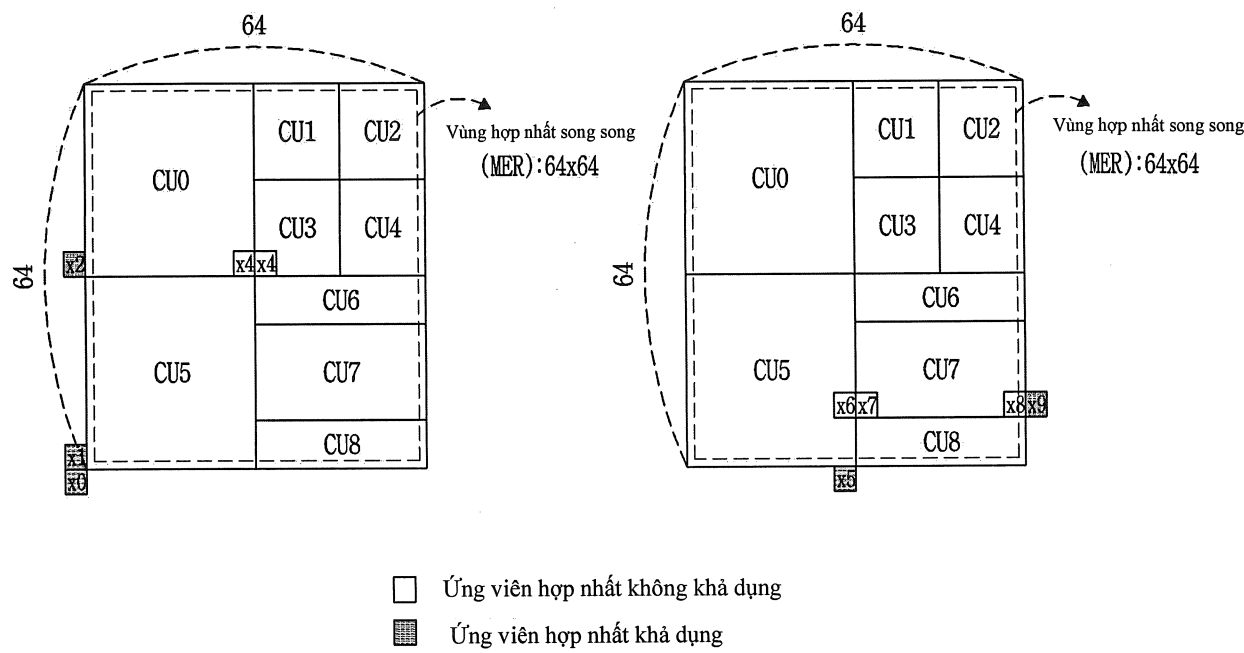
【FIG. 16】



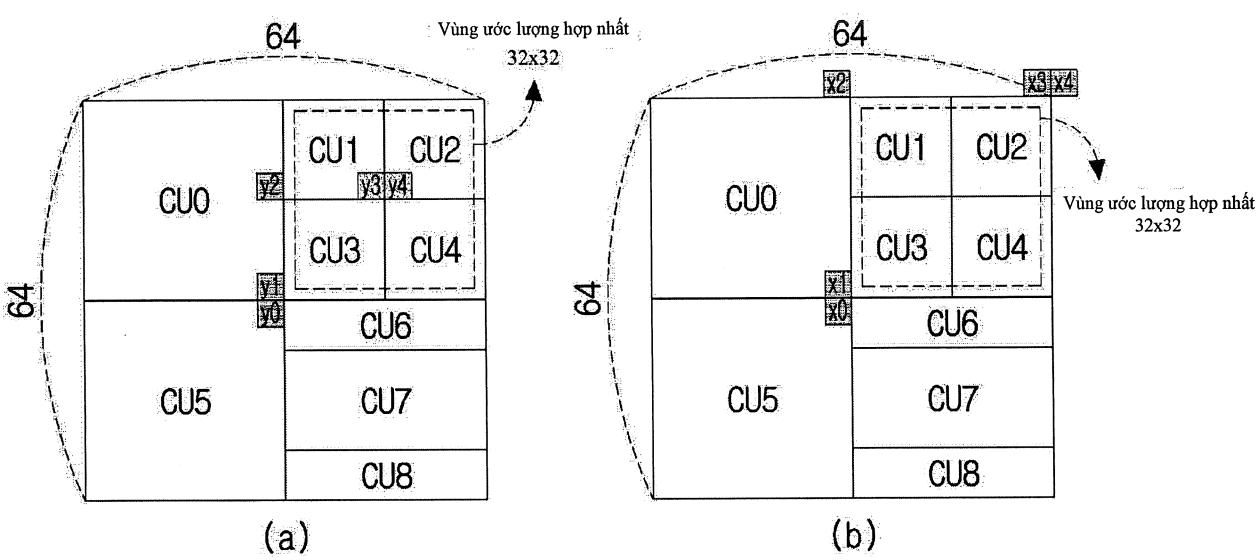
【FIG. 17】



【FIG. 18】

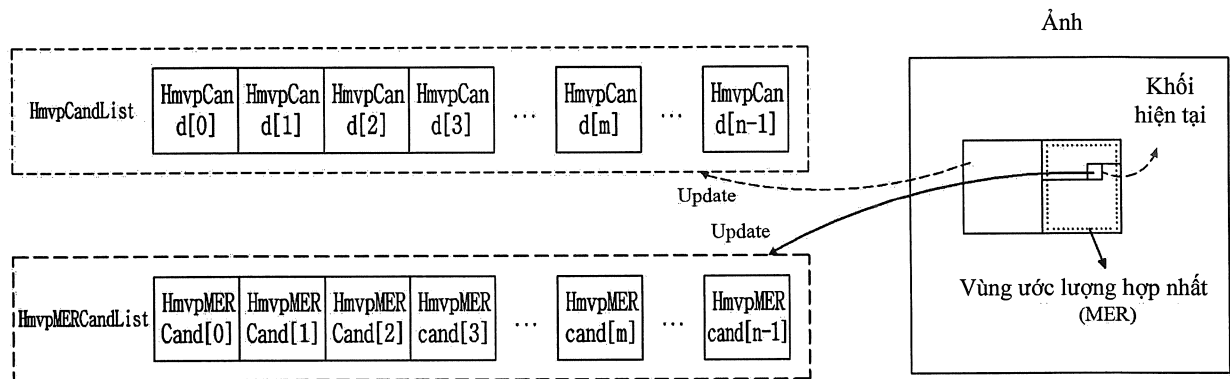


【FIG. 19】

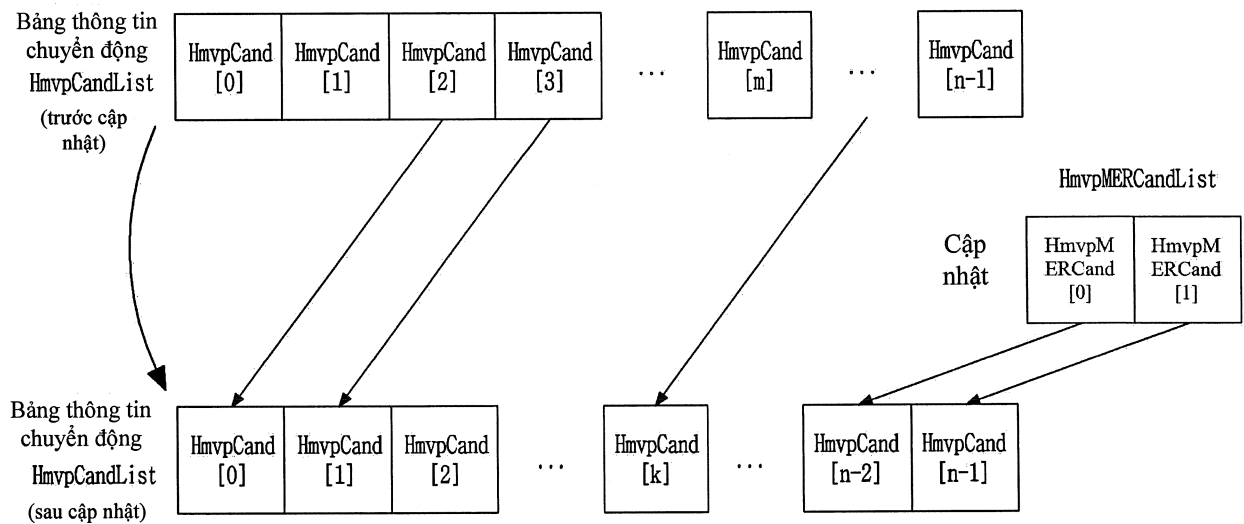


12/23

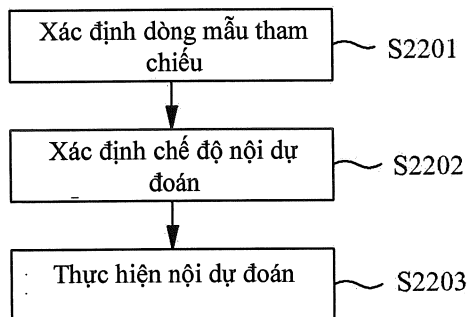
【FIG. 20】



【FIG. 21】

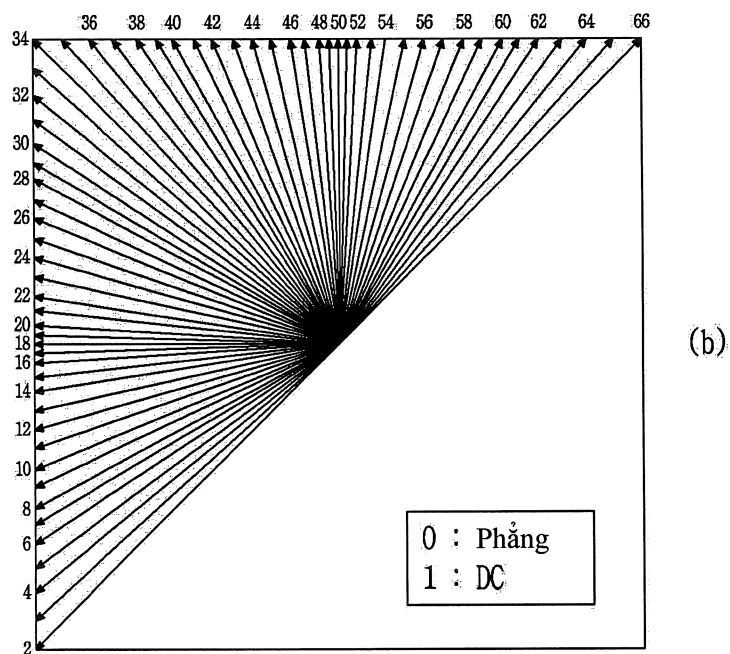
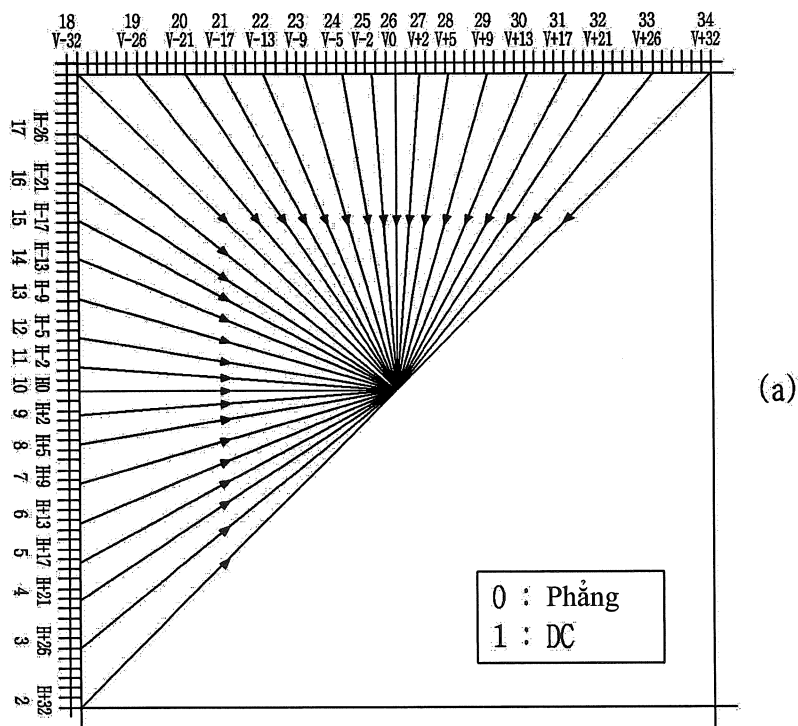


【FIG. 22】



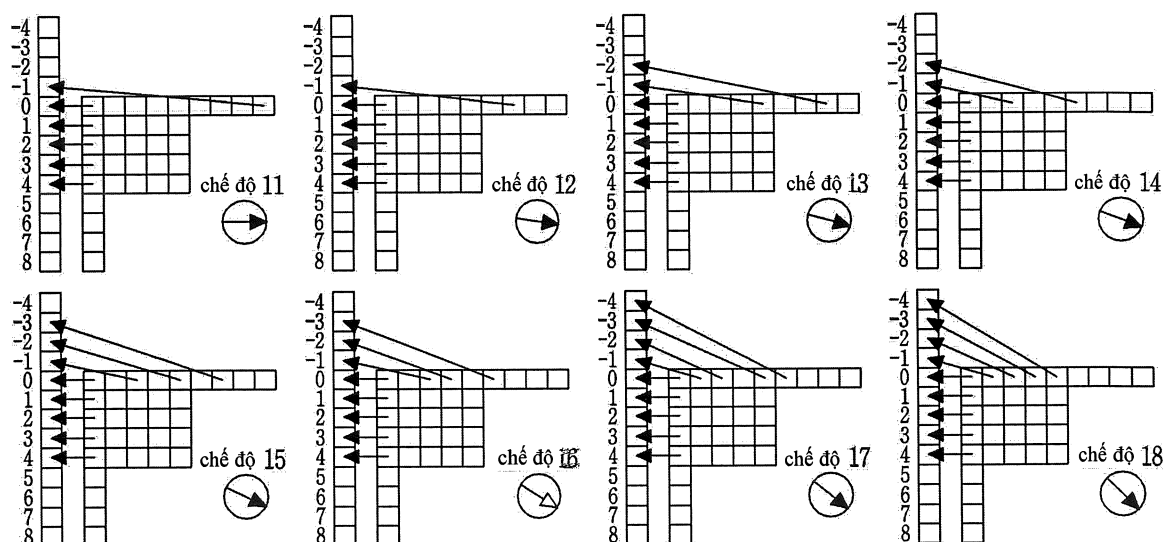
14/23

【FIG. 23】

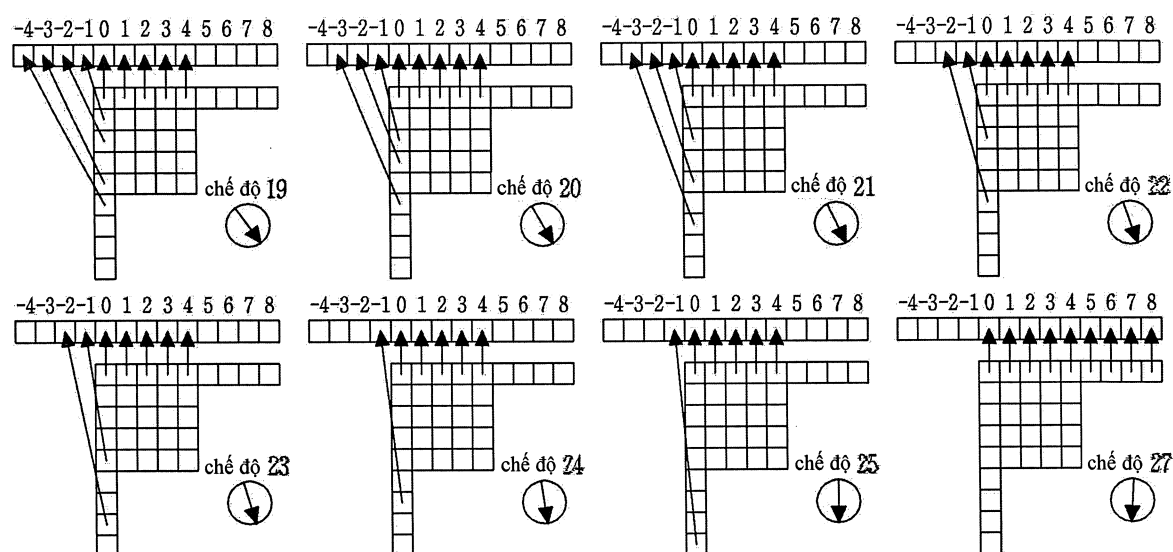


15/23

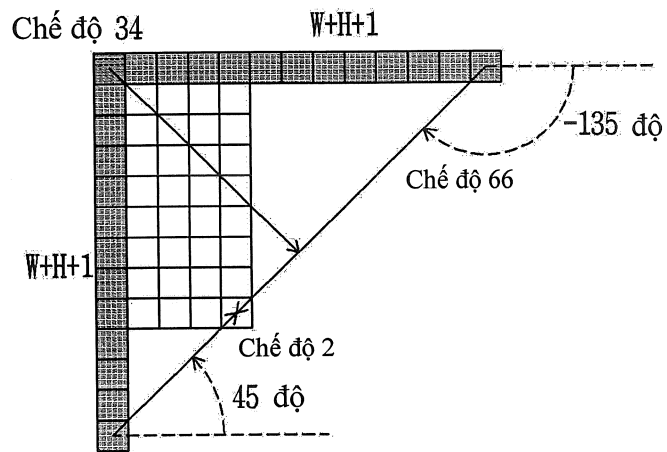
【FIG. 24】



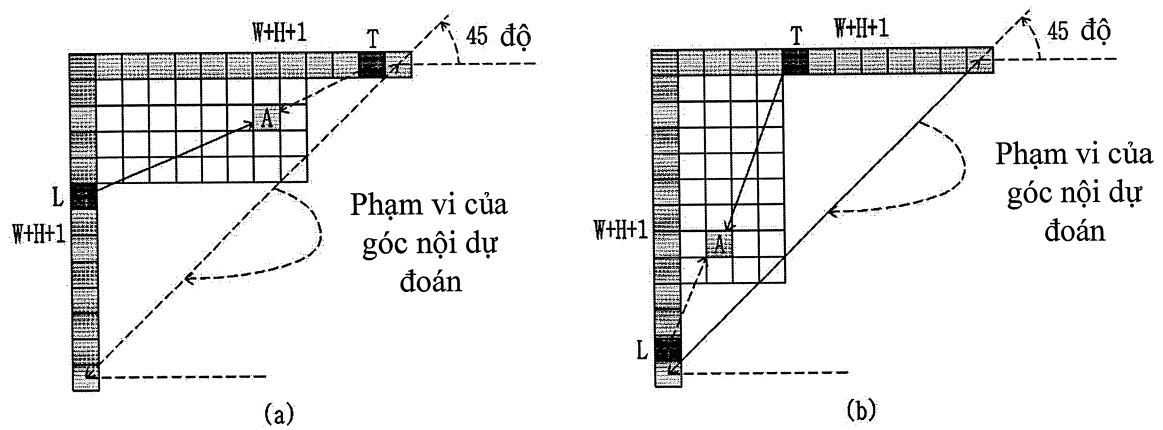
【FIG. 25】



【FIG. 26】

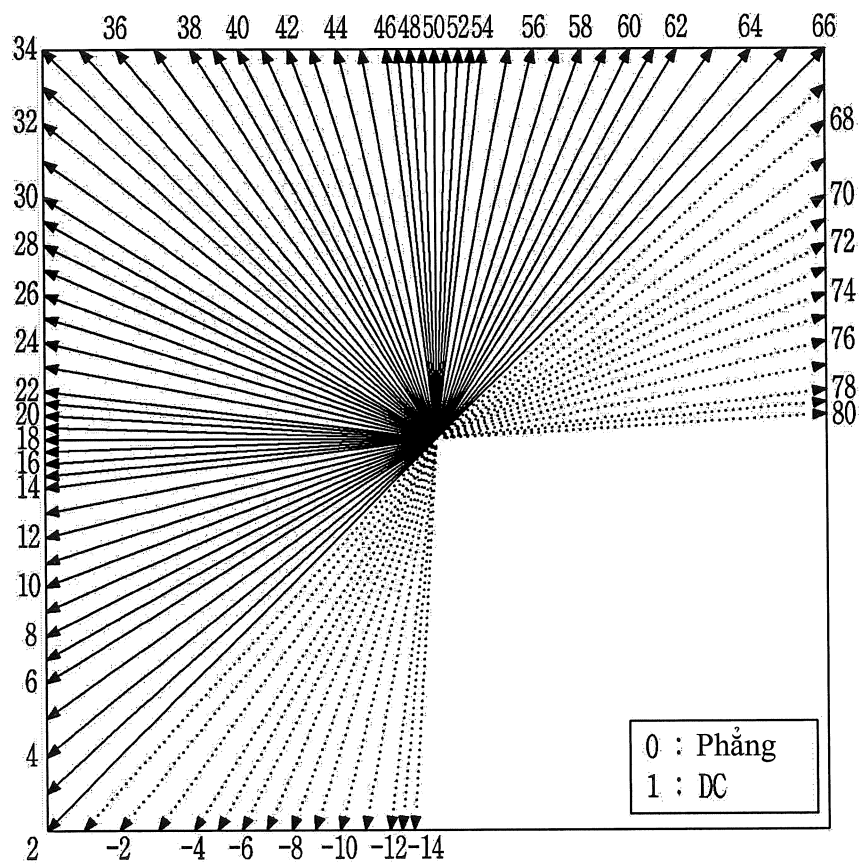


【FIG. 27】

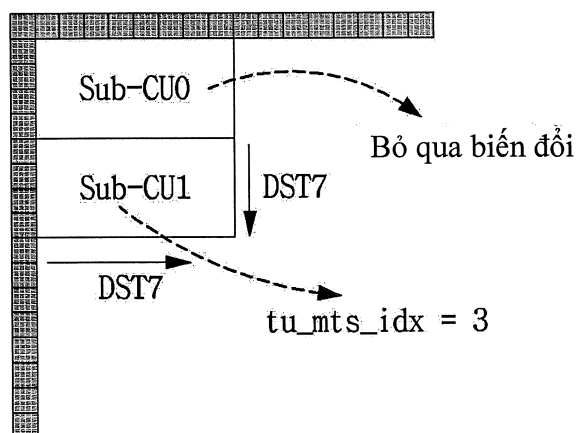


17/23

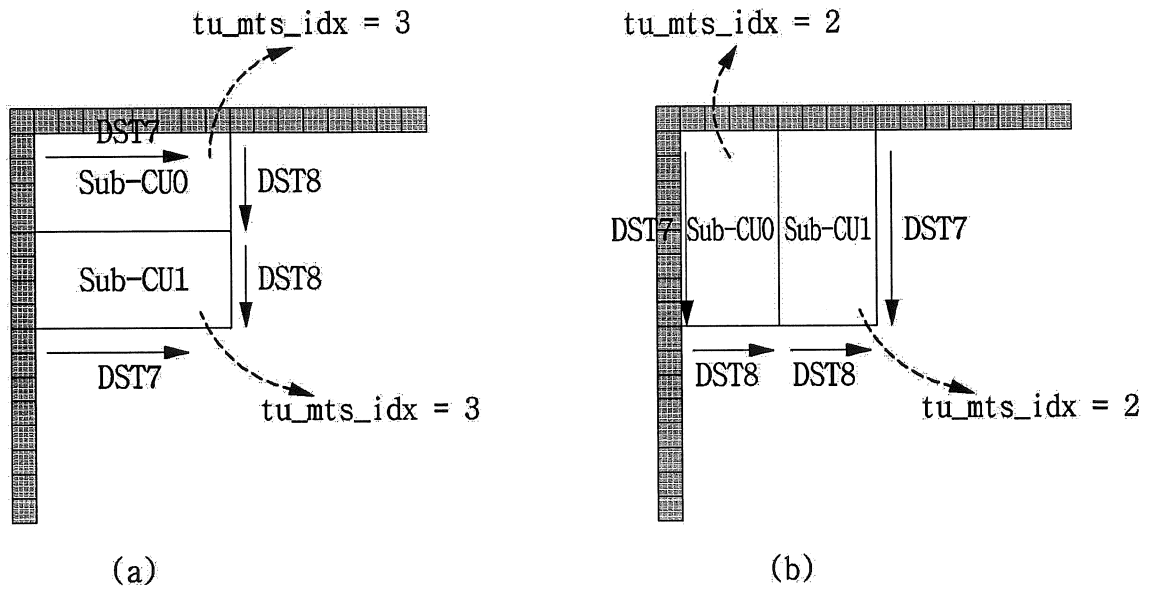
【FIG. 28】



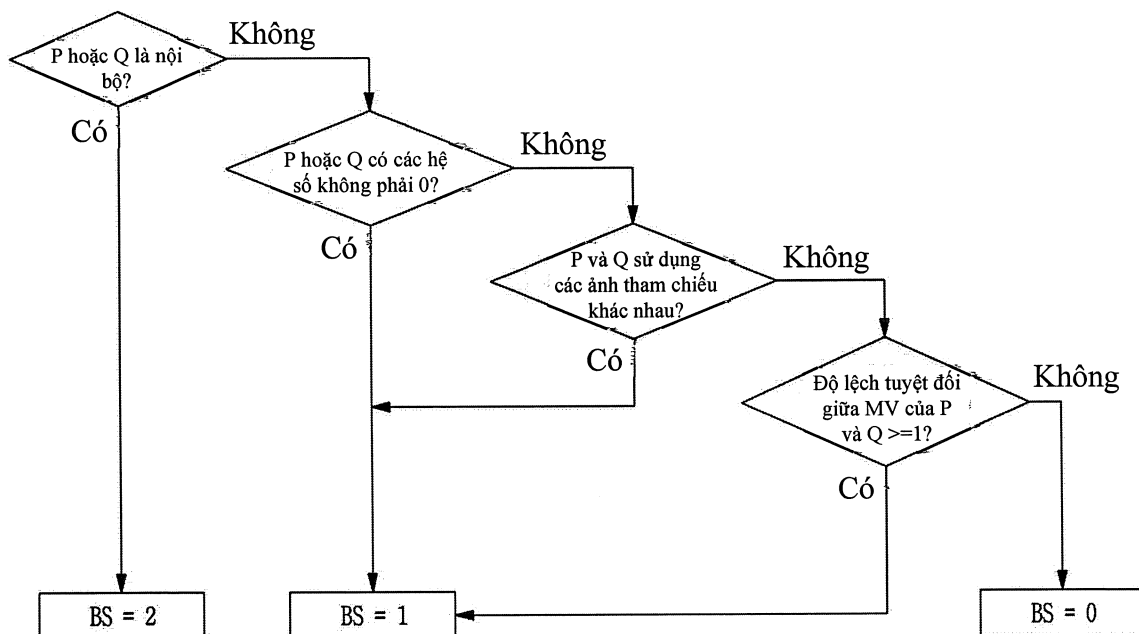
【FIG. 29】



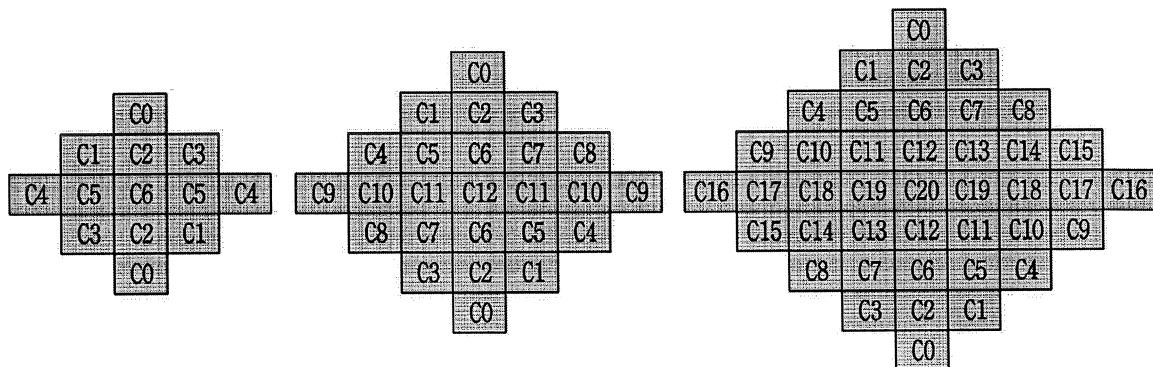
【FIG. 30】



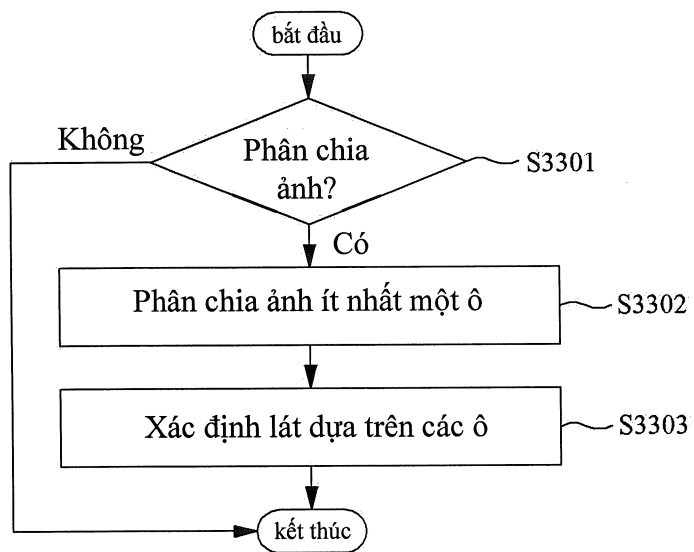
【FIG. 31】



【FIG. 32】

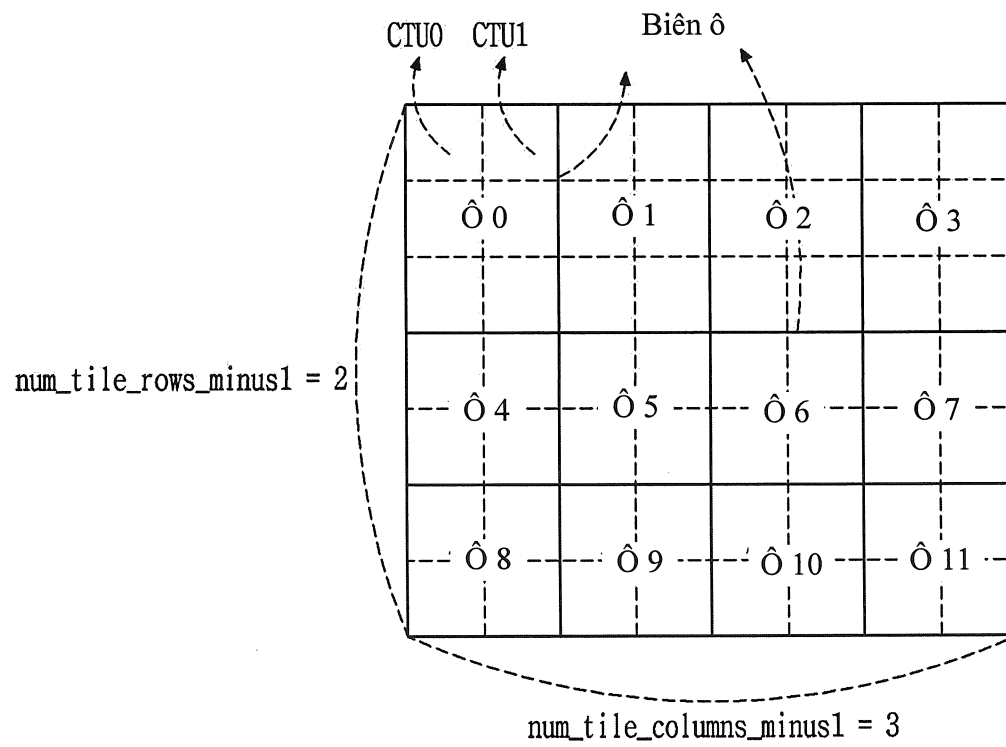


【FIG. 33】

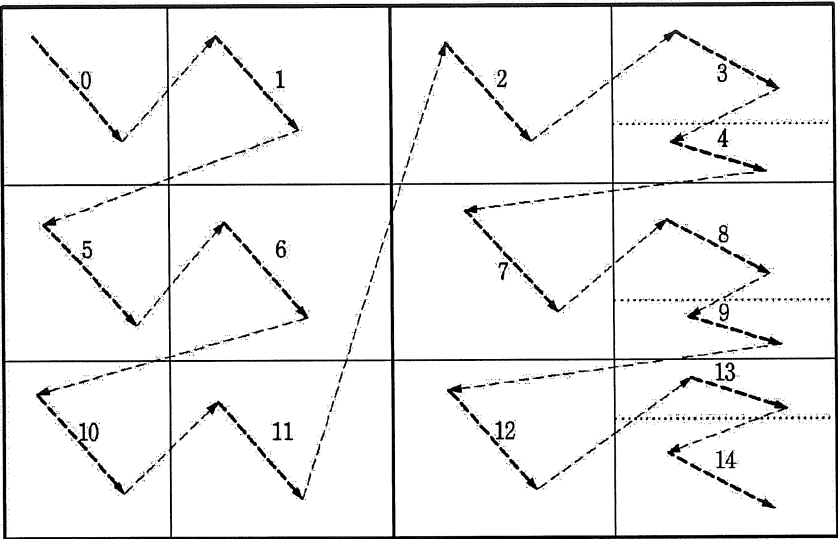
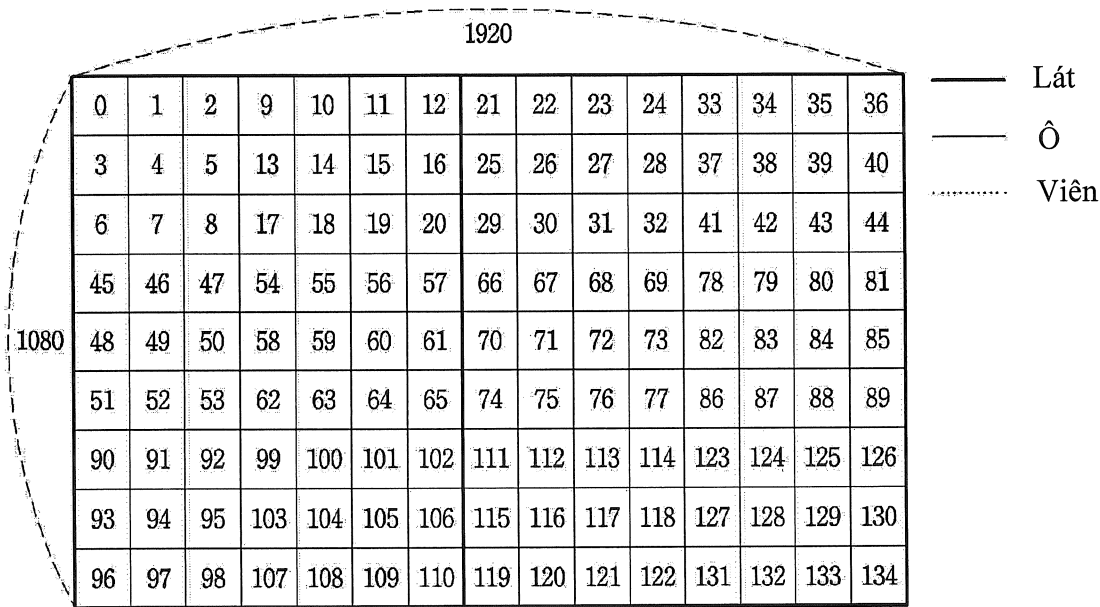


20/23

【FIG. 34】

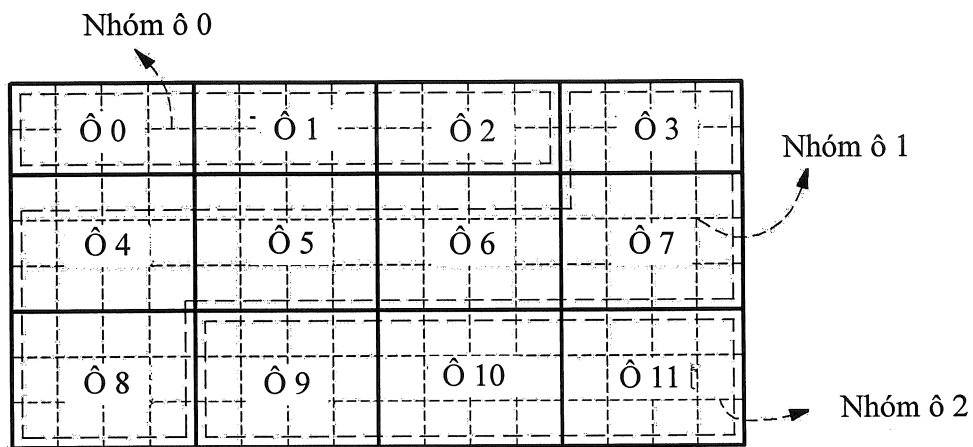


【FIG. 35】

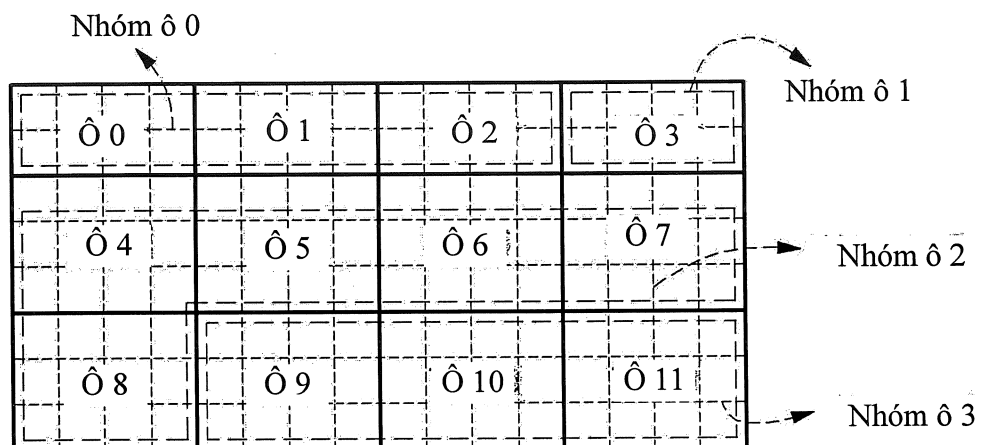


22/23

【FIG. 36】



【FIG. 37】



23/23

【FIG. 38】

