



(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)



1-0035931

(51)^{2020.01} H04N 19/105; H04N 19/176; H04N 19/593; H04N 19/139

(13) B

(21) 1-2021-07963

(22) 19/06/2020

(86) PCT/JP2020/024137 19/06/2020

(87) WO2020/256102 24/12/2020

(30) 2019-114395 20/06/2019 JP

(45) 26/06/2023 423

(43) 25/03/2022 408

(73) JVCKenwood Corporation (JP)

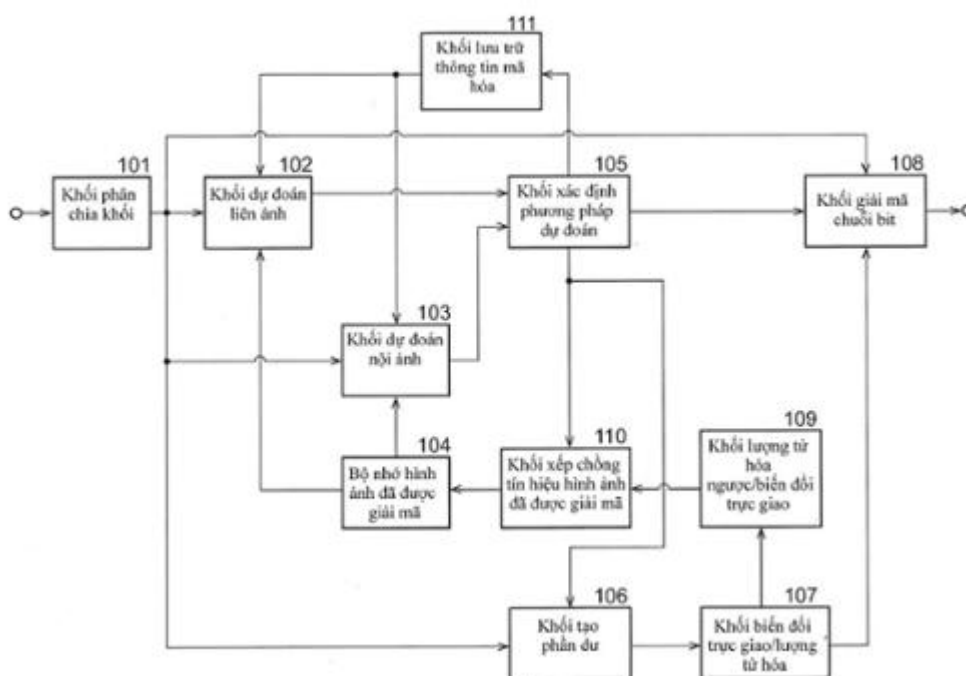
3-12, Moriyacho, Kanagawa-ku, Yokohama-shi, Kanagawa 2210022, Japan

(72) Hideki TAKEHARA (JP); Hiroya NAKAMURA (JP); Satoru SAKAZUME (JP); Shigeru FUKUSHIMA (JP); Toru KUMAKURA (JP); Hiroyuki KURASHIGE (JP).

(74) Văn phòng Luật sư Ân Nam (ANNAM IP & LAW)

(54) THIẾT BỊ MÃ HÓA HÌNH ẢNH, THIẾT BỊ GIẢI MÃ HÌNH ẢNH, PHƯƠNG PHÁP MÃ HÓA HÌNH ẢNH VÀ PHƯƠNG PHÁP GIẢI MÃ HÌNH ẢNH

(57) Sáng chế đề xuất thiết bị mã hóa hình ảnh, thiết bị này bao gồm: khối rút ra ứng viên vector khối được tạo cấu hình để rút ra các ứng viên vector khối của khối đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa; khối lựa chọn được tạo cấu hình để lựa chọn vector khối đã lựa chọn được từ các ứng viên vector khối; khối hiệu chỉnh vị trí tham chiếu được tạo cấu hình để thực hiện hiệu chỉnh liên quan đến khối tham chiếu sẽ được đề cập tới bởi vector khối đã được lựa chọn do vậy vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu; trong đó mẫu đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã dưới dạng giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các kỹ thuật mã hóa và giải mã hình ảnh trong đó hình ảnh được phân chia thành các khối và việc dự đoán được thực hiện.

Tình trạng kỹ thuật của sáng chế

Trong quy trình mã hóa và giải mã hình ảnh, hình ảnh đích được phân chia thành các khối, mỗi khối là một nhóm với số lượng mẫu định trước, và quy trình xử lý được thực hiện theo đơn vị khối. Việc phân chia hình ảnh thành các khối thích hợp cùng với các thiết lập về chế độ dự đoán nội ảnh và chế độ dự đoán liên ảnh thích hợp cho phép cải thiện hiệu quả mã hóa.

Tài liệu sáng chế 1 bộc lộ kỹ thuật dự đoán nội ảnh từ việc thu được hình ảnh được dự đoán bằng cách sử dụng các mẫu đã giải mã được ở lân cận khối đích mã hóa/giải mã.

[Tài liệu sáng chế 1] JP 2009-246975 A.

Bản chất kỹ thuật của sáng chế

Tuy nhiên, kỹ thuật trong Tài liệu sáng chế 1 chỉ sử dụng các mẫu giải mã được ở lân cận khối đích mã hóa/giải mã, để dự đoán, và điều này khiến cho hiệu quả dự đoán không cao.

Theo một khía cạnh của sáng chế để giải quyết vấn đề nêu trên, sáng chế đề xuất thiết bị bao gồm: khối rút ra ứng viên vector khối để rút ra các ứng viên vector khối của khối đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa; bộ lựa chọn để lựa chọn vector khối được lựa chọn từ các ứng viên vector khối; và khối hiệu chỉnh vị trí tham chiếu thực hiện việc hiệu chỉnh liên quan

đến khối tham chiếu sẽ được xét tới bởi vector khối đã lựa chọn được do đó vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu, trong đó mẫu được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã như là giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu.

Theo sáng chế, có thể đạt được quy trình mã hóa/giải mã hình ảnh hiệu quả cao và tải thấp.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối của thiết bị mã hóa hình ảnh theo phương án của sáng chế.

Fig.2 là sơ đồ khối của thiết bị giải mã hình ảnh theo phương án của sáng chế.

Fig.3 là lưu đồ minh họa hoạt động tách khối cây.

Fig.4 là biểu đồ minh họa trạng thái của quy trình tách hình ảnh đầu vào thành các khối cây.

Fig.5 là biểu đồ minh họa z-scan.

Fig.6A là biểu đồ minh họa hình dạng tách của khối.

Fig.6B là biểu đồ minh họa hình dạng tách của khối.

Fig.6C là biểu đồ minh họa hình dạng tách của khối.

Fig.6D là biểu đồ minh họa hình dạng tách của khối.

Fig.6E là biểu đồ minh họa hình dạng tách của khối.

Fig.7 là lưu đồ minh họa hoạt động phân chia khối thành bốn phần.

Fig.8 là lưu đồ minh họa hoạt động phân chia khối thành hai hoặc ba phần.

Fig.9 là cú pháp để thể hiện hình dạng của sự phân chia các khối.

Fig.10A là biểu đồ minh họa dự đoán nội ảnh.

Fig.10B là biểu đồ minh họa dự đoán nội ảnh.

Fig.11 là biểu đồ minh họa các khối tham chiếu cho dự đoán liên ảnh.

Fig.12A là cú pháp để thể hiện chế độ dự đoán khối mã hóa.

Fig.12B là cú pháp để thể hiện chế độ dự đoán khối mã hóa.

Fig.13 là biểu đồ minh họa tương quan giữa các phần tử cú pháp và các chế độ liên quan đến dự đoán liên ảnh.

Fig.14 là biểu đồ minh họa Bộ chuyển động liên kết tại hai điểm điều khiển.

Fig.15 là biểu đồ minh họa Bộ chuyển động liên kết tại ba điểm điều khiển.

Fig.16 là sơ đồ khối của cấu hình chi tiết của khối dự đoán liên ảnh 102 trên Fig.1.

Fig.17 là sơ đồ khối của cấu hình chi tiết của khối rút ra chế độ dự đoán vector chuyển động thông thường 301 trên Fig.16.

Fig.18 là sơ đồ khối của cấu hình chi tiết của khối rút ra chế độ hợp nhất thông thường 302 trên Fig.16.

Fig.19 là lưu đồ minh họa quy trình rút ra chế độ dự đoán vector chuyển động thông thường của khối rút ra chế độ dự đoán vector chuyển động thông thường 301 trên Fig.16.

Fig.20 là lưu đồ minh họa thủ tục xử lý của quy trình rút ra chế độ dự đoán vector chuyển động thông thường.

Fig.21 là lưu đồ minh họa thủ tục xử lý của quy trình rút ra chế độ hợp nhất thông thường.

Fig.22 là sơ đồ khối của cấu hình chi tiết của khối dự đoán liên ảnh 203 trên Fig.2.

Fig.23 là sơ đồ khối của cấu hình chi tiết của khối rút ra chế độ dự đoán vector

chuyển động thông thường 401 trên Fig.22.

Fig.24 là sơ đồ khối của cấu hình chi tiết của khối rút ra chế độ hợp nhất thông thường 402 trên Fig.22.

Fig.25 là lưu đồ minh họa quy trình rút ra chế độ dự đoán vector chuyển động thông thường của khối rút ra chế độ dự đoán vector chuyển động thông thường 401 trên Fig.22.

Fig.26 là biểu đồ minh họa thủ tục của quy trình khởi tạo/cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.27 là lưu đồ thủ tục xử lý việc xác nhận các phần tử giống nhau thủ tục của quy trình khởi tạo/cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.28 là lưu đồ thủ tục xử lý dịch phần tử của quy trình khởi tạo/cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.29 là lưu đồ minh họa thủ tục xử lý rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.30 là lưu đồ minh họa thủ tục xử lý rút ra ứng viên hợp nhất dựa trên lịch sử.

Fig.31A là biểu đồ minh họa ví dụ về quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.31B là biểu đồ minh họa ví dụ về quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.31C là biểu đồ minh họa ví dụ về quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Fig.32 là biểu đồ minh họa dự đoán bù chuyển động trong trường hợp dự đoán

L0 được thực hiện và hình ảnh tham chiếu (RefL0Pic) của L0 ở tại thời điểm trước hình ảnh đích (CurPic).

Fig.33 là biểu đồ minh họa dự đoán bù chuyển động trong trường hợp dự đoán L0 được thực hiện và hình ảnh tham chiếu của dự đoán L0 ở tại thời điểm sau hình ảnh đích.

Fig.34 là biểu đồ minh họa hướng dự đoán của dự đoán bù chuyển động trong dự đoán kép trong đó hình ảnh tham chiếu dự đoán L0 tại thời điểm trước hình ảnh đích và hình ảnh tham chiếu dự đoán L1 ở tại thời điểm sau hình ảnh đích.

Fig.35 là biểu đồ minh họa hướng dự đoán của dự đoán bù chuyển động trong dự đoán kép trong đó hình ảnh tham chiếu dự đoán L0 và hình ảnh tham chiếu dự đoán L1 tại thời điểm trước hình ảnh đích.

Fig.36 là biểu đồ minh họa hướng dự đoán của dự đoán bù chuyển động trong dự đoán kép trong đó hình ảnh tham chiếu dự đoán L0 và hình ảnh tham chiếu dự đoán L1 tại thời điểm sau hình ảnh đích.

Fig.37 là biểu đồ minh họa ví dụ về cấu hình phần cứng của thiết bị mã hóa-giải mã theo phương án của sáng chế.

Fig.38 là lưu đồ minh họa thủ tục xử lý rút ra ứng viên hợp nhất trung bình.

Fig.39A là sơ đồ minh họa vùng tham chiếu hợp lệ của bản sao khối nội ảnh.

Fig.39B là sơ đồ minh họa vùng tham chiếu hợp lệ của bản sao khối nội ảnh.

Fig.40 là sơ đồ khối của cấu hình chi tiết của khối dự đoán nội ảnh 103 trên Fig.1.

Fig.41 là sơ đồ khối của cấu hình chi tiết của khối dự đoán nội ảnh 204 trên Fig.2.

Fig.42 là sơ đồ khối của khối dự đoán bản sao khối nội ảnh 352.

Fig.43 là sơ đồ khối của khối dự đoán bản sao khối nội ảnh 362.

Fig.44 là lưu đồ minh họa quy trình sao chép khối dự đoán nội ảnh của khối dự đoán bản sao khối nội ảnh 352.

Fig.45 là lưu đồ minh họa quy trình sao chép khối dự đoán nội ảnh của khối dự đoán bản sao khối nội ảnh 362.

Fig.46 là lưu đồ minh họa quy trình sao chép khối nội ảnh hợp nhất.

Fig.47 là lưu đồ minh họa thủ tục xử lý của quy trình rút ra chế độ vector khối của bản sao khối dự đoán nội ảnh.

Fig.48 là sơ đồ minh họa quy trình được thực hiện bởi khối hiệu chỉnh vị trí tham chiếu 380 và khối hiệu chỉnh vị trí tham chiếu 480.

Fig.49 là sơ đồ minh họa trạng thái của việc hiệu chỉnh vị trí tham chiếu.

Fig.50A là sơ đồ minh họa các vị trí phía trên bên trái và phía dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật.

Fig.50B là sơ đồ minh họa các vị trí phía trên bên trái và phía dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật.

Fig.50C là sơ đồ minh họa các vị trí phía trên bên trái và phía dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật.

Fig.50D là sơ đồ minh họa các vị trí phía trên bên trái và phía dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật.

Fig.51 là sơ đồ minh họa quy trình hiệu chỉnh vị trí tham chiếu của vị trí mà vùng có thể tham chiếu không phải là hình chữ nhật.

Fig.52A là sơ đồ minh họa trạng thái của việc hiệu chỉnh vị trí tham chiếu.

Fig.52B là sơ đồ minh họa trạng thái của việc hiệu chỉnh vị trí tham chiếu.

Fig.53 là sơ đồ minh họa quy trình được thực hiện bởi khối hiệu chỉnh vị trí

tham chiều 380 và khối hiệu chỉnh vị trí tham chiều 480.

Fig.54A là sơ đồ minh họa trạng thái của việc phân chia vùng có thể tham chiều thành hai.

Fig.54B là sơ đồ minh họa trạng thái của việc phân chia vùng có thể tham chiều thành hai.

Fig.54C là sơ đồ minh họa trạng thái của việc phân chia vùng có thể tham chiều thành hai.

Fig.54D là sơ đồ minh họa trạng thái của việc phân chia vùng có thể tham chiều thành hai.

Fig.55 là sơ đồ minh họa quy trình phân chia vùng có thể tham chiều thành hai và hiệu chỉnh mỗi vị trí tham chiều.

Mô tả chi tiết sáng chế

Sau đây là phần định nghĩa các công nghệ và các thuật ngữ kỹ thuật được sử dụng trong phương án của sáng chế.

Khối cây

Trong phương án này, hình ảnh đích mã hóa/giải mã được chia đều thành kích thước định trước. Đơn vị này được gọi là khối cây. Ở Fig.4 kích thước của khối cây được thiết lập là 128×128 mẫu, tuy nhiên, kích thước của khối cây không giới hạn ở đó và có thể được thiết lập với kích thước bất kỳ. Khối cây đích (tương ứng với đích mã hóa trong quy trình mã hóa và đích giải mã trong quy trình giải mã) được chuyển theo kiểu quét mảnh, tức là, theo thứ tự từ trái sang phải và từ trên xuống dưới. Phần bên trong của mỗi khối cây có thể được tách một cách đệ quy hơn nữa. Khối mã hóa/giải mã là kết quả của việc phân chia đệ quy khối cây được gọi là khối mã hóa. Khối cây và khối mã hóa được gọi chung là khối. Việc thực hiện phân chia thích hợp

các khối cho phép mã hóa hiệu quả. Kích thước của khối cây có thể là giá trị cố định được định trước bởi thiết bị mã hóa và thiết bị giải mã, hoặc có thể tuân theo cấu hình trong đó kích thước của khối cây được định trước bởi thiết bị mã hóa được truyền đến thiết bị giải mã. Ở đây, kích thước tối đa của khối cây là 128×128 mẫu, và kích thước tối thiểu của khối cây là 16×16 mẫu. Kích thước tối đa của khối mã hóa là 64×64 mẫu, và kích thước tối thiểu của khối mã hóa là 4×4 mẫu.

Chế độ dự đoán

Việc chuyển đổi được thực hiện giữa dự đoán nội ảnh (MODE_INTRA) thực hiện dự đoán từ tín hiệu hình ảnh được xử lý của hình ảnh đích và dự đoán liên ảnh (MODE_INTER) thực hiện dự đoán từ tín hiệu hình ảnh của hình ảnh được xử lý trong các đơn vị của các khối mã hóa đích.

Hình ảnh đã được xử lý được sử dụng, trong quy trình mã hóa, cho hình ảnh thu được bằng cách giải mã tín hiệu đã được mã hóa, tín hiệu hình ảnh, khối cây, khối, khối mã hóa, hoặc tương tự. Hình ảnh đã được xử lý được sử dụng, trong quy trình giải mã, cho hình ảnh đã được giải mã, tín hiệu hình ảnh, khối cây, khối, khối mã hóa, hoặc tương tự.

Chế độ xác định dự đoán nội ảnh (MODE_INTRA) và dự đoán liên ảnh (MODE_INTER) được gọi là chế độ dự đoán (PredMode). Chế độ dự đoán (PredMode) có giá trị là dự đoán nội ảnh (MODE_INTRA) hoặc dự đoán liên ảnh (MODE_INTER).

Dự đoán bản sao khối nội ảnh

[0012] Dự đoán bản sao khối nội ảnh là quy trình mã hóa/giải mã khối đích cùng với việc tham chiếu tới mẫu giải mã được trong hình ảnh đích như là giá trị dự đoán. Khoảng cách từ khối đích tới mẫu tham chiếu được đại diện bởi vector khối.

Vecto khối đề cập tới hình ảnh đích và hình ảnh tham chiếu được xác định duy nhất. Bởi vậy, không cần phải có chỉ số tham chiếu. Sự khác biệt giữa vectơ khối và vectơ chuyển động là để xem hình ảnh tham chiếu là hình ảnh đích hay hình ảnh đã được xử lý. Ngoài ra, trong trường hợp vectơ khối, có thể chọn độ chính xác mẫu 1 hoặc độ chính xác mẫu 4 có thể được lựa chọn bằng cách sử dụng độ phân giải vectơ chuyển động thích ứng (adaptive motion vector resolution - AMVR).

Trong bản sao khối nội ảnh, có thể lựa chọn từ hai chế độ: chế độ bản sao khối dự đoán nội ảnh hoặc chế độ bản sao khối hợp nhất nội ảnh.

Chế độ bản sao khối dự đoán nội ảnh là chế độ xác định vectơ khối của khối đích trên cơ sở dự đoán vectơ khối rút ra được từ thông tin đã được xử lý và trên cơ sở sự độ chênh lệch vectơ khối. Dự đoán vectơ khối được rút ra từ khối đã được xử lý ở lân cận khối đích và từ chỉ số để chỉ ra dự đoán vectơ khối. Chỉ số để chỉ ra dự đoán vectơ khối và sự độ chênh lệch vectơ khối được xác định trong luồng bit.

Chế độ bản sao khối hợp nhất nội ảnh là chế độ rút ra thông tin dự đoán bản sao khối nội ảnh của khối đích từ thông tin dự đoán bản sao khối nội ảnh của khối đã được xử lý ở lân cận khối đích mà không truyền độ chênh lệch vectơ chuyển động.

Dự đoán liên ảnh

Trong dự đoán liên ảnh, việc dự đoán được thực hiện từ tín hiệu hình ảnh của hình ảnh được xử lý, có thể sử dụng các hình ảnh đã được xử lý làm các hình ảnh tham chiếu. Hai kiểu danh sách tham chiếu L0 (danh sách tham chiếu 0) và L1 (danh sách tham chiếu 1) được xác định để quản lý các hình ảnh tham chiếu. Hình ảnh tham chiếu được chỉ ra bằng cách sử dụng chỉ số tham chiếu trong mỗi danh sách. Trong Lát P, dự đoán L0 (Pred_L0) có thể được sử dụng. Trong B lát, dự đoán L0 (Pred_L0), dự đoán L1 (Pred_L1), và dự đoán kép (Pred_BI) có thể được sử dụng. Dự đoán L0 (Pred_L0)

là dự đoán liên ảnh đề cập đến hình ảnh tham chiếu được quản lý bởi L0, còn dự đoán L1 (Pred_L1) là dự đoán liên ảnh đề cập đến hình ảnh tham chiếu được quản lý bởi L1. Dự đoán kép (Pred_BI) là dự đoán liên ảnh trong đó cả dự đoán L0 và dự đoán L1 được thực hiện và đề cập đến một hình ảnh tham chiếu được quản lý trong mỗi L0 và L1. Thông tin chỉ ra dự đoán L0, dự đoán L1, và dự đoán kép được gọi là chế độ dự đoán liên ảnh. Trong quy trình xử lý sau đây, giả sử rằng việc xử lý sẽ được thực hiện đối với mỗi L0 và L1 cho các hằng số và các biến bao gồm hậu tố LX trong đầu ra.

Chế độ dự đoán vector chuyển động

Chế độ dự đoán vector chuyển động là chế độ truyền chỉ số để chỉ ra dự đoán vector chuyển động, sai phân vector chuyển động, chế độ dự đoán liên ảnh, và chỉ số tham chiếu, và xác định thông tin dự đoán liên ảnh của khối đích. Dự đoán vector chuyển động được lấy từ ứng viên dự đoán vector chuyển động có được từ khối đã được xử lý ở lân cận khối đích hoặc khối thuộc về hình ảnh đã được xử lý và được đặt tại cùng vị trí với hoặc trong vùng lân cận (liền kề) khối đích, và từ chỉ số để chỉ ra dự đoán vector chuyển động.

Chế độ hợp nhất

Chế độ hợp nhất là chế độ lấy thông tin dự đoán liên ảnh khối đích từ thông tin dự đoán liên ảnh của khối đã được xử lý ở lân cận khối đích, hoặc khối thuộc về hình ảnh đã được xử lý và được đặt tại cùng vị trí với khối đích hoặc trong vùng lân cận (liền kề) khối đích, mà không truyền sai phân vector chuyển động hoặc chỉ số tham chiếu.

Khối được xử lý ở lân cận khối đích và thông tin dự đoán liên ảnh của khối đã được xử lý được xác định là các ứng viên hợp nhất theo không gian. Các khối thuộc về hình ảnh đã được xử lý và được đặt tại cùng vị trí với khối đích hoặc trong vùng lân

cận (liền kề) khối đích, và thông tin dự đoán liên ảnh có được từ thông tin dự đoán liên ảnh của khối được xác định là các ứng viên hợp nhất theo thời gian. Mỗi ứng viên hợp nhất được thêm vào danh sách ứng viên hợp nhất. Ứng viên hợp nhất được sử dụng cho việc dự đoán của khối đích được chỉ ra bởi chỉ số hợp nhất.

Khối lân cận

Fig.11 là biểu đồ minh họa các khối tham chiếu được sử dụng để lấy thông tin dự đoán liên ảnh trong chế độ dự đoán vectơ chuyển động và chế độ hợp nhất. A0, A1, A2, B0, B1, B2, và B3 là các khối đã được xử lý ở lân cận khối đích. T0 là khối thuộc về hình ảnh đã được xử lý và được đặt tại cùng vị trí với khối đích hoặc trong vùng lân cận (liền kề) khối đích, trong hình ảnh đích.

A1 và A2 là các khối nằm phía bên trái của khối mã hóa đích và ở lân cận khối mã hóa đích. B1 và B3 là các khối nằm trên khối mã hóa đích và ở lân cận khối mã hóa đích. A0, B0, và B2 là các khối được đặt tương ứng ở phía dưới bên trái, phía trên bên phải, và phía trên bên trái của khối mã hóa đích.

Sau đây là phần mô tả chi tiết về cách xử lý các khối lân cận trong chế độ dự đoán vectơ chuyển động và chế độ hợp nhất.

Bù chuyển động liên kết

Trước tiên bù chuyển động liên kết chia khối mã hóa thành các khối con của đơn vị định trước và rồi sau đó xác định riêng vectơ chuyển động cho mỗi của khối đã con đã được phân chia để thực hiện bù chuyển động. vectơ chuyển động của mỗi khối con được rút ra trên cơ sở một hoặc nhiều điểm điều khiển có được từ thông tin dự đoán liên ảnh của khối đã được xử lý ở lân cận khối đích, hoặc khối thuộc về hình ảnh đã được xử lý và được đặt tại cùng vị trí với hoặc trong vùng lân cận (liền kề) khối đích. Tuy phương án thiết lập kích thước của khối con là 4×4 mẫu, nhưng kích thước

của khối con không giới hạn ở đó, và vector chuyển động có thể được tính trong các đơn vị của các mẫu.

Fig.14 minh họa ví dụ về bù chuyển động liên kết trong trường hợp có hai điểm điều khiển. Trong trường hợp này, mỗi điểm trong số hai điểm điều khiển có hai tham số, tức là, thành phần theo phương ngang và thành phần theo phương đứng. Theo đó, phép biến đổi liên kết có hai điểm điều khiển được gọi là phép biến đổi liên kết bốn tham số. CP1 và CP2 trên Fig.14 là các điểm điều khiển.

Fig.15 minh họa ví dụ về bù chuyển động liên kết trong trường hợp có ba điểm điều khiển. Trong trường hợp này, mỗi điểm trong số ba điểm điều khiển có hai tham số, tức là, thành phần theo phương ngang và thành phần theo phương đứng. Theo đó, phép biến đổi liên kết có ba điểm điều khiển được gọi là phép biến đổi liên kết sáu tham số. CP1, CP2, và CP3 trên Fig.15 là các điểm điều khiển.

Bù chuyển động liên kết có thể được sử dụng cho chế độ bất kỳ trong chế độ dự đoán vector chuyển động và chế độ hợp nhất. Chế độ áp dụng bù chuyển động liên kết trong chế độ dự đoán vector chuyển động được gọi là chế độ dự đoán vector chuyển động khối con. Chế độ áp dụng bù chuyển động liên kết trong chế độ hợp nhất được gọi là chế độ hợp nhất khối con.

Cú pháp mã hóa khối

Cú pháp để thể hiện chế độ dự đoán của khối mã hóa sẽ được mô tả tham chiếu đến các hình vẽ Fig.12A, Fig.12B, và Fig.13. `pred_mode_flag` trên Fig.12A là cờ thể hiện có phải là chế độ dự đoán liên ảnh hay không. Đặt `pred_mode_flag` 0 thể hiện dự đoán liên ảnh còn đặt `pred_mode_flag` 1 thể hiện dự đoán nội ảnh. Trong trường hợp dự đoán nội ảnh, `pred_mode_ibc_flag` là cờ thể hiện có phải là chế độ dự đoán bản sao khối được truyền hay không. Trong trường hợp dự đoán bản sao khối nội ảnh

(pred_mode_ibc_flag=1), merge_flag được truyền đi. merge_flag là cờ thể hiện chế độ được thiết lập là chế độ sao chép khối nội ảnh hợp nhất hay chế độ sao chép khối nội ảnh dự đoán. Trong trường hợp chế độ sao chép khối nội ảnh hợp nhất (merge_flag=1), chỉ số merge_idx được truyền đi. Khi dự đoán không phải là dự đoán bản sao khối nội ảnh (pred_mode_ibc_flag = 0), dự đoán được thiết lập là dự đoán nội ảnh thông thường, và thông tin intra_pred_mode của dự đoán nội ảnh thông thường được truyền đi.

Trong trường hợp dự đoán liên ảnh, merge_flag là cờ được truyền đi. merge_flag là cờ thể hiện chế độ sử dụng là chế độ hợp nhất hay chế độ dự đoán vector chuyển động. Trong trường hợp chế độ dự đoán vector chuyển động (merge_flag = 0), cờ inter_affine_flag thể hiện có áp dụng chế độ dự đoán vector chuyển động khối con hay không được truyền đi. Trong trường hợp áp dụng chế độ dự đoán vector chuyển động khối con (inter_affine_flag = 1), cu_affine_type_flag được truyền đi. cu_affine_type_flag là cờ để xác định số lượng điểm điều khiển trong chế độ dự đoán vector chuyển động khối con.

Ngược lại, trong trường hợp chế độ hợp nhất (merge_flag = 1), merge_khối con_flag trên Fig.12B được truyền đi. merge_khối con_flag là cờ thể hiện có áp dụng chế độ hợp nhất khối con hay không. Trong trường hợp chế độ hợp nhất khối con (merge_khối con_flag = 1), chỉ số hợp nhất merge_khối con_idx được truyền đi. Ngược lại, trong trường hợp chế độ không phải là chế độ hợp nhất khối con (merge_khối con_flag = 0), cờ merge_triangle_flag thể hiện có áp dụng chế độ hợp nhất tam giác hay không được truyền đi. Trong trường hợp áp dụng chế độ hợp nhất tam giác (merge_triangle_flag = 1), các chỉ số tam giác hợp nhất merge_triangle_idx0 và merge_triangle_idx1 được truyền đi cho hướng tách khối merge_triangle_split_dir,

và cho mỗi phần trong số hai phần phân tách. Trong trường hợp không áp dụng chế độ hợp nhất tam giác, ($\text{merge_triangle_flag} = 0$), chỉ số hợp nhất merge_idx được truyền đi.

Fig.13 minh họa giá trị của mỗi phần tử cú pháp trong dự đoán liên ảnh và chế độ dự đoán tương ứng. $\text{merge_flag} = 0$ và $\text{inter_affine_flag} = 0$ tương ứng với chế độ dự đoán vector chuyển động thông thường (Inter Pred Mode). $\text{merge_flag} = 0$ và $\text{inter_affine_flag} = 1$ tương ứng với chế độ dự đoán vector chuyển động khối con (Inter Affine Mode). $\text{merge_flag} = 1$, $\text{merge_khối con_flag} = 0$, và $\text{merge_triangle_flag} = 0$ tương ứng với chế độ hợp nhất thông thường (Chế độ hợp nhất). $\text{merge_flag} = 1$, $\text{merge_khối con_flag} = 0$, và $\text{merge_triangle_flag} = 1$ tương ứng với chế độ hợp nhất tam giác (Chế độ hợp nhất tam giác). $\text{merge_flag} = 1$, $\text{merge_khối con_flag} = 1$ tương ứng với chế độ hợp nhất khối con (Chế độ hợp nhất liên kết).

POC

Số thứ tự hình ảnh (POC) là biến được kết hợp với hình ảnh sẽ được mã hóa, và được thiết lập là giá trị tăng dần theo thứ tự đầu ra hình ảnh. Giá trị POC giúp có thể phân biệt xem các hình ảnh có giống nhau hay không, phân biệt mối quan hệ tuần tự giữa các ảnh theo thứ tự đầu ra hoặc rút ra khoảng cách giữa các ảnh. Ví dụ, có thể xác định rằng hai hình ảnh có cùng giá trị POC là các hình ảnh giống hệt nhau. Trong trường hợp các POC của hai hình ảnh có giá trị khác nhau, hình ảnh có giá trị POC nhỏ hơn có thể được xác định là hình ảnh được xuất sớm hơn. Sự khác biệt giữa các POC của hai hình ảnh có thể thể hiện khoảng cách giữa các hình ảnh trên trục thời gian.

Phương án thứ nhất

Thiết bị mã hóa hình ảnh 100 và thiết bị giải mã hình ảnh 200 theo phương án

thứ nhất của sáng chế sẽ được mô tả.

Fig.1 là sơ đồ khối của thiết bị mã hóa hình ảnh 100 theo phương án thứ nhất. Thiết bị mã hóa hình ảnh 100 theo phương án bao gồm khối phân chia khối 101, khối dự đoán liên ảnh 102, khối dự đoán nội ảnh 103, bộ nhớ hình ảnh đã được giải mã 104, khối xác định phương pháp dự đoán 105, khối tạo phần dư 106, khối lượng tử hóa/biến đổi trực giao 107, khối mã hóa chuỗi bit 108, khối lượng tử hóa ngược/biến đổi trực giao ngược 109, khối xếp chồng tín hiệu hình ảnh đã được giải mã 110, và bộ nhớ lưu trữ thông tin mã hóa 111.

Khối phân chia khối 101 chia đệ quy hình ảnh đầu vào để tạo khối mã hóa. Khối phân chia khối 101 bao gồm: bộ chia bốn thực hiện chia khối đích chia theo cả phương ngang và phương đứng; và bộ chia nhị phân bậc ba thực hiện chia khối đích chia một trong phương ngang hoặc phương đứng. Khối phân chia khối 101 thiết lập khối mã hóa đã được tạo ra là khối mã hóa đích, và đưa tín hiệu hình ảnh của khối mã hóa đích đến khối dự đoán liên ảnh 102, khối dự đoán nội ảnh 103, và khối tạo phần dư 106. Ngoài ra, khối phân chia khối 101 đưa thông tin thể hiện cấu trúc phân chia đệ quy được xác định tới khối mã hóa chuỗi bit 108. Hoạt động chi tiết của khối phân chia khối 101 sẽ được mô tả sau đây.

Khối dự đoán liên ảnh 102 thực hiện dự đoán liên ảnh khối mã hóa đích. Khối dự đoán liên ảnh 102 rút ra các ứng viên dự đoán liên ảnh từ thông tin dự đoán liên ảnh được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 và tín hiệu hình ảnh đã được giải mã được lưu trong bộ nhớ hình ảnh đã được giải mã 104, lựa chọn chế độ dự đoán liên ảnh thích hợp từ các ứng viên được rút ra, và đưa chế độ dự đoán liên ảnh lựa chọn được và tín hiệu hình ảnh được dự đoán tương ứng với chế độ dự đoán liên ảnh lựa chọn được tới khối xác định phương pháp dự đoán 105. Cấu hình và hoạt động chi

tiết của khối dự đoán liên ảnh 102 sẽ được mô tả sau đây.

Khối dự đoán nội ảnh 103 thực hiện dự đoán nội ảnh trên khối mã hóa đích. Khối dự đoán nội ảnh 103 xem tín hiệu hình ảnh đã được giải mã được lưu trong bộ nhớ hình ảnh đã được giải mã 104 như là mẫu tham chiếu, và thực hiện dự đoán nội ảnh dựa trên thông tin mã hóa chẳng hạn như chế độ dự đoán nội ảnh được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 và nhờ đó tạo ra tín hiệu hình ảnh được dự đoán. Trong dự đoán nội ảnh, khối dự đoán nội ảnh 103 lựa chọn chế độ dự đoán nội ảnh thích hợp từ các chế độ dự đoán nội ảnh, và đưa chế độ dự đoán nội ảnh lựa chọn được và tín hiệu hình ảnh được dự đoán đã được lựa chọn tương ứng với chế độ dự đoán nội ảnh lựa chọn được tới khối xác định phương pháp dự đoán 105. Cấu hình và hoạt động chi tiết của khối dự đoán nội ảnh 103 sẽ được mô tả chi tiết sau đây.

Bộ nhớ hình ảnh đã được giải mã 104 lưu hình ảnh đã được giải mã được tạo bởi khối xếp chồng tín hiệu hình ảnh đã được giải mã 110. Bộ nhớ hình ảnh đã được giải mã 104 đưa hình ảnh giải mã đã lưu tới khối dự đoán liên ảnh 102 và khối dự đoán nội ảnh 103.

Khối xác định phương pháp dự đoán 105 đánh giá mỗi dự đoán nội ảnh và dự đoán liên ảnh bằng cách sử dụng thông tin mã hóa, lượng mã hóa của phần dư, lượng méo giữa tín hiệu hình ảnh được dự đoán và tín hiệu hình ảnh đích, hoặc tương tự, và nhờ đó xác định chế độ dự đoán tối ưu. Trong trường hợp dự đoán nội ảnh, khối xác định phương pháp dự đoán 105 đưa thông tin dự đoán nội ảnh chẳng hạn như chế độ dự đoán nội ảnh tới khối mã hóa chuỗi bit 108 như là thông tin mã hóa. Trong trường hợp chế độ hợp nhất của dự đoán liên ảnh, khối xác định phương pháp dự đoán 105 đưa thông tin dự đoán liên ảnh chẳng hạn như chỉ số hợp nhất và thông tin (cờ hợp nhất khối con) thể hiện chế độ có phải là chế độ hợp nhất khối con hay không tới khối

mã hóa chuỗi bit 108 như là thông tin mã hóa. Trong trường hợp chế độ dự đoán vector chuyển động của dự đoán liên ảnh, khối xác định phương pháp dự đoán 105 đưa thông tin dự đoán liên ảnh chẳng hạn như chế độ dự đoán liên ảnh, chỉ số dự đoán vector chuyển động, chỉ số tham chiếu của L0 hoặc L1, sai phân vector chuyển động, hoặc thông tin thể hiện chế độ có phải là chế độ dự đoán vector chuyển động khối con hay không (cờ dự đoán vector chuyển động khối con) tới khối mã hóa chuỗi bit 108 như là thông tin mã hóa. Khối xác định phương pháp dự đoán 105 còn đưa thông tin mã hóa xác định được tới bộ nhớ lưu trữ thông tin mã hóa 111. Khối xác định phương pháp dự đoán 105 đưa tín hiệu hình ảnh được dự đoán tới khối tạo phần dư 106 và khối xếp chồng tín hiệu hình ảnh đã được giải mã 110.

Khối tạo phần dư 106 tạo ra phần dư lấy tín hiệu hình ảnh đích trừ đi tín hiệu hình ảnh được dự đoán, và đưa phần dư tạo ra tới khối lượng tử hóa/biến đổi trực giao 107.

Khối lượng tử hóa/biến đổi trực giao 107 thực hiện biến đổi trực giao và lượng tử hóa trên phần dư theo tham số lượng tử và nhờ đó tạo ra phần dư đã được biến đổi trực giao và lượng tử hóa, và sau đó đưa phần dư tạo ra tới khối mã hóa chuỗi bit 108 và khối lượng tử hóa ngược/biến đổi trực giao ngược 109.

Khối mã hóa chuỗi bit 108 mã hóa, ngoài các chuỗi, hình ảnh, lát, và thông tin trong các đơn vị của các khối mã hóa, khối mã hóa chuỗi bit 108 mã hóa thông tin mã hóa tương ứng với phương pháp dự đoán được định trước bởi khối xác định phương pháp dự đoán 105 cho mỗi của các khối mã hóa. Đặc biệt, khối mã hóa chuỗi bit 108 mã hóa chế độ dự đoán PredMode cho mỗi khối mã hóa. Trong trường hợp chế độ dự đoán là dự đoán liên ảnh (MODE_INTER), khối mã hóa chuỗi bit 108 mã hóa thông tin mã hóa (thông tin dự đoán liên ảnh) chẳng hạn như cờ để xác định xem chế độ có

phải là chế độ hợp nhất hay không, cờ hợp nhất khối con, chỉ số hợp nhất trong chế độ hợp nhất, chế độ dự đoán liên ảnh trong các chế độ không phải chế độ hợp nhất, chỉ số dự đoán vector chuyển động, thông tin liên quan đến các sai phân vector chuyển động, và cờ dự đoán vector chuyển động khối con, trên cơ sở cú pháp quy định trước (quy tắc cú pháp của chuỗi bit) và nhờ đó tạo ra chuỗi bit thứ nhất. Trong trường hợp chế độ dự đoán là dự đoán nội ảnh (MODE_INTRA), khối mã hóa chuỗi bit 108 mã hóa cờ để xác định xem chế độ có phải là sao chép khối nội ảnh dựa trên cú pháp đã quy định hay không. Trong trường hợp sao chép khối nội ảnh, thông tin mã hóa (thông tin dự đoán nội ảnh) chẳng hạn như chỉ số hợp nhất cho trường hợp chế độ hợp nhất, chỉ số dự đoán vector khối, và độ chênh lệch vector khối cho chế độ không phải chế độ hợp nhất được mã hóa dựa trên cú pháp quy định trước, thông tin mã hóa (thông tin dự đoán nội ảnh) chẳng hạn như chế độ dự đoán nội ảnh được mã hóa dựa trên cú pháp quy định trước. Chuỗi bit thứ nhất được tái dựng bởi việc mã hóa nêu trên. Ngoài ra, khối mã hóa chuỗi bit 108 thực hiện mã hóa entropy trên phần dư đã được biến đổi trực giao và lượng tử hóa dựa trên cú pháp quy định trước và nhờ đó tạo ra chuỗi bit thứ hai. Khối mã hóa chuỗi bit 108 ghép chuỗi bit thứ nhất và chuỗi bit thứ hai dựa trên cú pháp quy định trước, và xuất ra luồng bit.

Khối lượng tử hóa ngược/biến đổi trực giao ngược 109 thực hiện lượng tử hóa ngược và biến đổi trực giao ngược trên phần dư đã được biến đổi trực giao/lượng tử hóa ngược được đưa tới từ khối lượng tử hóa/biến đổi trực giao 107 và nhờ đó tính toán phần dư, và sau đó đưa phần dư tính toán tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 110.

Khối xếp chồng tín hiệu hình ảnh đã được giải mã 110 xếp chồng tín hiệu hình ảnh được dự đoán theo xác định của khối xác định phương pháp dự đoán 105 cùng với

phần dư đã trải qua lượng tử hóa ngược/biến đổi trực giao ngược bởi khối lượng tử hóa ngược/biến đổi trực giao ngược 109, nhờ đó tạo ra hình ảnh đã được giải mã, và lưu hình ảnh giải mã được tạo ra trong bộ nhớ hình ảnh đã được giải mã 104. Khối xếp chồng tín hiệu hình ảnh đã được giải mã 110 có thể thực hiện xử lý lọc để giảm méo chẳng hạn như méo khối do mã hóa trên hình ảnh đã được giải mã, và sau đó có thể lưu hình ảnh đã được giải mã trong bộ nhớ hình ảnh đã được giải mã 104.

Bộ nhớ lưu trữ thông tin mã hóa 111 lưu thông tin mã hóa chẳng hạn như chế độ dự đoán (dự đoán liên ảnh hoặc dự đoán nội ảnh) được định trước bởi khối xác định phương pháp dự đoán 105. Trong trường hợp dự đoán liên ảnh, thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 bao gồm thông tin dự đoán liên ảnh chẳng hạn như vector chuyển động xác định được, các chỉ số tham chiếu của danh sách tham chiếu L0 và L1, và danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử. Trong trường hợp dự đoán liên ảnh chế độ hợp nhất, thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 bao gồm, ngoài thông tin nêu trên, chỉ số hợp nhất và thông tin dự đoán liên ảnh bao gồm thông tin thể hiện chế độ có phải là chế độ hợp nhất khối con (cờ hợp nhất khối con). Trong trường hợp chế độ dự đoán vector chuyển động của dự đoán liên ảnh, thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 bao gồm, ngoài thông tin nêu trên, thông tin dự đoán liên ảnh chẳng hạn như chế độ dự đoán liên ảnh, chỉ số dự đoán vector chuyển động, sai phân vector chuyển động, và thông tin thể hiện chế độ có phải là chế độ dự đoán vector chuyển động khối con hay không (cờ dự đoán vector chuyển động khối con). Trong trường hợp dự đoán nội ảnh, thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa 111 bao gồm thông tin dự đoán nội ảnh chẳng hạn như chế độ dự đoán nội ảnh xác định được.

Fig.2 là sơ đồ khối minh họa cấu hình của thiết bị giải mã hình ảnh theo phương án của sáng chế tương ứng với thiết bị mã hóa hình ảnh trên Fig.1. Thiết bị giải mã hình ảnh theo phương án bao gồm khối giải mã chuỗi bit 201, khối phân chia khối 202, khối dự đoán liên ảnh 203, khối dự đoán nội ảnh 204, bộ nhớ lưu trữ thông tin mã hóa 205, khối lượng tử hóa ngược/biến đổi trực giao ngược 206, và khối xếp chồng tín hiệu hình ảnh đã được giải mã 207, và bộ nhớ hình ảnh đã được giải mã 208.

Do quy trình giải mã của thiết bị giải mã hình ảnh trên Fig.2 tương ứng với quy trình giải mã ở phía thiết bị mã hóa hình ảnh trên Fig.1. Theo đó, mỗi cấu hình của bộ nhớ lưu trữ thông tin mã hóa 205, khối lượng tử hóa ngược/biến đổi trực giao ngược 206, khối xếp chồng tín hiệu hình ảnh đã được giải mã 207, và bộ nhớ hình ảnh đã được giải mã 208 trên Fig.2 lần lượt có chức năng tương ứng với mỗi cấu hình của bộ nhớ lưu trữ thông tin mã hóa 111, khối lượng tử hóa ngược/biến đổi trực giao ngược 109, khối xếp chồng tín hiệu hình ảnh đã được giải mã 110, và bộ nhớ hình ảnh đã được giải mã 104 của thiết bị mã hóa hình ảnh trên Fig.1.

Luồng bit được đưa tới khối giải mã chuỗi bit 201 được tách dựa trên quy tắc cú pháp quy định trước. Khối giải mã chuỗi bit 201 giải mã chuỗi bit thứ nhất đã được tách, và nhờ đó thu được hình ảnh, chuỗi, lát, thông tin trong các đơn vị của các khối mã hóa, và thông tin mã hóa trong các đơn vị của các khối mã hóa. Đặc biệt, khối giải mã chuỗi bit 201 giải mã chế độ dự đoán PredMode mà phân biệt chế độ dự đoán có phải là dự đoán liên ảnh (MODE_INTER) hay dự đoán nội ảnh (MODE_INTRA) trong các đơn vị của khối mã hóa. Trong trường hợp chế độ dự đoán là dự đoán liên ảnh (MODE_INTER), khối giải mã chuỗi bit 201 giải mã thông tin mã hóa (thông tin dự đoán liên ảnh) liên quan đến cờ để phân biệt chế độ có phải là chế độ hợp nhất hay không, chỉ số hợp nhất trong trường hợp chế độ hợp nhất, cờ hợp nhất khối con, và dự

đoán liên ảnh trong trường hợp chế độ dự đoán vectơ chuyển động, chỉ số dự đoán vectơ chuyển động, sai phân vectơ chuyển động, cờ dự đoán vectơ chuyển động khối con hoặc tương tự theo cú pháp quy định trước, và sau đó, đưa thông tin mã hóa (thông tin dự đoán liên ảnh) tới bộ nhớ lưu trữ thông tin mã hóa 205 thông qua khối dự đoán liên ảnh 203 và khối phân chia khối 202. Trong trường hợp chế độ dự đoán là dự đoán nội ảnh (MODE_INTRA), khối giải mã chuỗi bit 201 giải mã cờ để xác định xem chế độ có phải là sao chép khối nội ảnh hay không. Trong trường hợp sao chép khối nội ảnh, thông tin mã hóa (thông tin dự đoán nội ảnh) chẳng hạn như chỉ số hợp nhất cho chế độ hợp nhất, chỉ số dự đoán vectơ khối và độ chênh lệch vectơ khối cho chế độ không phải chế độ hợp nhất được giải mã dựa trên cú pháp được quy định trước. Trong trường hợp sao chép khối nội ảnh, thông tin mã hóa (thông tin dự đoán nội ảnh) chẳng hạn như chế độ dự đoán nội ảnh được giải mã dựa trên cú pháp được quy định trước. Nhờ việc giải mã nêu trên, thông tin mã hóa (thông tin dự đoán nội ảnh) được cung cấp cho bộ nhớ lưu trữ thông tin mã hóa 205 thông qua khối dự đoán liên ảnh 203 hoặc khối dự đoán nội ảnh 204, và thông qua khối phân chia khối 202. Khối giải mã chuỗi bit 201 giải mã chuỗi bit thứ hai đã được tách và tính toán phần dư đã được biến đổi trực giao/lượng tử hóa ngược, và sau đó, đưa phần dư đã được biến đổi trực giao/lượng tử hóa ngược tới khối lượng tử hóa ngược/biến đổi trực giao ngược 206.

Khi chế độ dự đoán PredMode của khối mã hóa đích là dự đoán liên ảnh (MODE_INTER) và chế độ dự đoán vectơ chuyển động, khối dự đoán liên ảnh 203 sử dụng thông tin mã hóa của tín hiệu hình ảnh đã được giải mã đã được lưu trong bộ nhớ lưu trữ thông tin mã hóa 205 để rút ra các ứng viên dự đoán vectơ chuyển động. Khối dự đoán liên ảnh 203 sau đó thêm các ứng viên dự đoán vectơ chuyển động được rút ra vào danh sách ứng viên dự đoán vectơ chuyển động như sau. Khối dự đoán liên

ảnh 203 lựa chọn dự đoán vector chuyển động tương ứng với chỉ số dự đoán vector chuyển động sẽ được giải mã và được đưa tới bởi khối giải mã chuỗi bit 201 từ các ứng viên dự đoán vector chuyển động được ghi trong danh sách ứng viên dự đoán vector chuyển động. Sau đó, khối dự đoán liên ảnh 203 tính toán vector chuyển động dựa trên sai phân vector chuyển động được giải mã bởi khối giải mã chuỗi bit 201 và dự đoán vector chuyển động lựa chọn được, và lưu vector chuyển động tính toán được trong bộ nhớ lưu trữ thông tin mã hóa 205 cùng với thông tin mã hóa khác. Ở đây, thông tin mã hóa của khối mã hóa sẽ được đưa tới và được lưu bao gồm chế độ dự đoán PredMode, các cờ predFlagL0[xP][yP] và predFlagL1[xP][yP] thể hiện sử dụng dự đoán L0 hay dự đoán L1, các chỉ số tham chiếu refIdxL0[xP][yP] và refIdxL1[xP][yP] của L0 và L1; và các vector chuyển động mvL0[xP][yP] và mvL1[xP][yP] của L0 và L1. Ở đây, xP và yP là các chỉ số thể hiện vị trí của mẫu phía trên bên trái của khối mã hóa trong hình ảnh. Trong trường hợp chế độ dự đoán PredMode là dự đoán liên ảnh (MODE_INTER) và chế độ dự đoán liên ảnh là dự đoán L0 (Pred_L0), cờ predFlagL0 thể hiện sử dụng dự đoán L0 được thiết lập là 1 và cờ predFlagL1 thể hiện sử dụng dự đoán L1 được thiết lập là 0. Trong trường hợp chế độ dự đoán liên ảnh là dự đoán L1 (Pred_L1), cờ predFlagL0 thể hiện sử dụng dự đoán L0 được thiết lập là 0 và cờ predFlagL1 thể hiện sử dụng dự đoán L1 được thiết lập là 1. Trong trường hợp chế độ dự đoán liên ảnh là dự đoán kép (Pred_BI), cả cờ predFlagL0 thể hiện sử dụng dự đoán L0 và cờ predFlagL1 thể hiện sử dụng dự đoán L1 được thiết lập là 1. Ngoài ra, khi chế độ dự đoán PredMode của khối mã hóa đích ở trong dự đoán liên ảnh (MODE_INTER) và chế độ hợp nhất, rút ra được ứng viên hợp nhất. Bằng cách sử dụng thông tin mã hóa của khối mã hóa được giải mã được lưu sẵn trong bộ nhớ lưu trữ thông tin mã hóa 205, các ứng viên hợp nhất được rút ra và được

ghi trong danh sách ứng viên hợp nhất được mô tả sau đây. Sau đó, ứng viên hợp nhất tương ứng với chỉ số hợp nhất được giải mã bởi khối giải mã chuỗi bit 201 và được đưa tới được lựa chọn từ các ứng viên hợp nhất được ghi trong danh sách ứng viên hợp nhất, và sau đó, thông tin dự đoán liên ảnh chẳng hạn như các cờ $\text{predFlagL0}[xP][yP]$ và $\text{predFlagL1}[xP][yP]$ thể hiện sử dụng dự đoán L0 và dự đoán L1 của ứng viên hợp nhất chọn được, các chỉ số tham chiếu $\text{refIdxL0}[xP][yP]$ và $\text{refIdxL1}[xP][yP]$ của L0 và L1, và các vector chuyển động $\text{mvL0}[xP][yP]$ và $\text{mvL1}[xP][yP]$ của L0 và L1 là để lưu trong bộ nhớ lưu trữ thông tin mã hóa 205. Ở đây, xP và yP là các chỉ số thể hiện vị trí của mẫu phía trên bên trái của khối mã hóa trong hình ảnh. Cấu hình và hoạt động chi tiết của khối dự đoán liên ảnh 203 sẽ được mô tả sau đây.

Khối dự đoán nội ảnh 204 thực hiện dự đoán nội ảnh khi chế độ dự đoán PredMode của khối mã hóa đích là dự đoán nội ảnh (MODE_INTRA). thông tin mã hóa được giải mã bởi khối giải mã chuỗi bit 201 bao gồm chế độ dự đoán nội ảnh. Khối dự đoán nội ảnh 204 tạo ra tín hiệu hình ảnh được dự đoán bởi dự đoán nội ảnh từ tín hiệu hình ảnh đã được giải mã được lưu trong bộ nhớ hình ảnh đã được giải mã 208 theo chế độ dự đoán nội ảnh có trong thông tin mã hóa được giải mã bởi khối giải mã chuỗi bit 201. Khối dự đoán nội ảnh 204 sau đó đưa tín hiệu hình ảnh được tạo ra được dự đoán tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207. Khối dự đoán nội ảnh 204 tương ứng với khối dự đoán nội ảnh 103 của thiết bị mã hóa hình ảnh 100, và vì vậy thực hiện quy trình tương tự với quy trình của khối dự đoán nội ảnh 103.

Khối lượng tử hóa ngược/biến đổi trực giao ngược 206 thực hiện biến đổi trực giao ngược/lượng tử hóa ngược trên phần dư đã được biến đổi trực giao/lượng tử hóa được giải mã trong khối giải mã chuỗi bit 201, và nhờ đó thu được phần dư được biến đổi trực giao ngược/lượng tử hóa ngược.

Khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 xếp chồng tín hiệu hình ảnh được dự đoán được dự đoán nội ảnh bởi khối dự đoán liên ảnh 203 hoặc tín hiệu hình ảnh dự đoán được dự đoán nội ảnh bởi khối dự đoán nội ảnh 204 bằng phần dư đã được biến đổi trực giao ngược/lượng tử hóa ngược bởi khối lượng tử hóa ngược/biến đổi trực giao ngược 206, nhờ đó giải mã tín hiệu hình ảnh đã được giải mã. Khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 sau đó lưu tín hiệu hình ảnh đã được giải mã đã được giải mã, trong bộ nhớ hình ảnh đã được giải mã 208. Khi lưu hình ảnh đã được giải mã trong bộ nhớ hình ảnh đã được giải mã 208, khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 có thể thực hiện xử lý lọc trên hình ảnh đã được giải mã để giảm méo khối hoặc tương tự do mã hóa, và sau đó có thể lưu hình ảnh đã được giải mã trong bộ nhớ hình ảnh đã được giải mã 208.

Tiếp theo, hoạt động của khối phân chia khối 101 trong thiết bị mã hóa hình ảnh 100 sẽ được mô tả. Fig.3 là lưu đồ minh họa hoạt động tách hình ảnh thành các khối cây và còn tách các khối cây. Đầu tiên, hình ảnh đầu vào được phân chia thành các khối cây có kích thước định trước (bước S1001). Mỗi khối cây được quét theo thứ tự định trước, tức là, theo kiểu quét mảnh (bước S1002), và khối cây đích được phân chia nội (bước S1003).

Fig.7 là lưu đồ minh họa hoạt động chi tiết của quy trình phân chia trong bước S1003. Đầu tiên, xác định xem có nên phân chia khối đích thành bốn hay không (bước S1101).

Trong trường hợp xác định rằng khối đích được phân chia thành bốn, khối đích sẽ được chia thành bốn (bước S1102). Mỗi khối thu được bằng cách tách khối đích được quét theo kiểu quét Z, tức là, theo thứ tự phía trên bên trái, phía trên bên phải, phía dưới bên trái, và phía dưới bên phải (bước S1103). Fig.5 minh họa ví dụ về

kiểu quét Z, và 601 trên Fig.6A minh họa ví dụ trong đó khối đích được chia thành bốn. Các chữ số từ 0 tới 3 của 601 trên Fig.6A thể hiện thứ tự xử lý. Sau đó, quy trình phân chia trên Fig.7 được thực hiện đệ quy cho mỗi khối được phân chia trong bước S1101 (bước S1104).

Trong trường hợp xác định rằng khối không được chia thành bốn, khối đích sẽ được chia thành hai hoặc ba, tức là, phân chia nhị phân-tam phân (bước S1105).

Fig.8 là lưu đồ minh họa hoạt động chi tiết quy trình phân chia nhị phân-tam phân trong bước S1105. Đầu tiên, xác định xem phân chia nhị phân-tam phân có được thực hiện trên khối đích hay không, tức là, phân chia nhị phân hoặc phân chia tam phân sẽ được thực hiện (bước S1201).

Trong trường hợp không xác định phân chia nhị phân-tam phân sẽ được thực hiện trên khối đích, tức là, trong trường hợp xác định không phân chia khối đích, việc phân chia kết thúc (bước S1211). Tức là, ngoài ra quy trình phân chia đệ quy sẽ không được thực hiện trên khối đã được phân chia bởi quy trình phân chia đệ quy.

Trong trường hợp xác định rằng phân chia nhị phân-tam phân sẽ được thực hiện trên khối đích, ngoài ra, xác định được rằng có phân chia khối đích thành hai hay không (bước S1202).

Trong trường hợp xác định rằng khối đích được phân chia thành hai, ngoài ra, xác định được rằng có phân chia khối đích theo phương (đứng) từ trên xuống dưới hay không (bước S1203), và sau đó dựa trên kết quả nhận được, khối đích sẽ được phân chia nhị phân theo phương (đứng) từ trên xuống dưới (bước S1204), hoặc khối đích sẽ được phân chia nhị phân theo phương (ngang) từ trái qua phải (bước S1205). Là kết quả của bước S1204, khối đích được phân chia nhị phân theo phương từ trên xuống dưới (phương đứng) như được minh họa trên 602 trên Fig.6B. Là kết quả của bước

S1205, khối đích được phân chia nhị phân theo chiều từ phải qua trái (phương ngang) như được minh họa trên 604 trên Fig.6D.

Trong bước S1202, trong trường hợp không xác định khối đích được phân chia thành hai, tức là, trong trường hợp xác định rằng khối đích được phân chia thành ba, ngoài ra, xác định được rằng có phân chia khối đích thành ba phần là phần trên, phần giữa, phần dưới (phương đứng) (bước S1206). Dựa trên kết quả thu được, khối đích được chia thành ba phần là phần trên, phần giữa và phần dưới (phương đứng) (bước S1207), hoặc phần bên trái, phần giữa và phần bên phải (phương ngang) (bước S1208). Là kết quả của bước S1207, khối đích được chia thành ba phần là phần trên, phần giữa và phần dưới (phương đứng) như được minh họa trên 603 trên Fig.6C. Là kết quả của bước S1208, khối đích được chia thành ba phần là phần bên trái, phần giữa và phần bên phải (phương ngang) như được minh họa trên 605 trên Fig.6E.

Sau khi thực hiện một trong số các bước S1204, S1205, S1207, hoặc S1208, mỗi khối thu được bằng cách tách khối đích được quét theo thứ tự từ trái sang phải và từ trên xuống dưới. (bước S1209). Các số hiệu từ 0 tới 2 của 602 tới 605 trên các hình vẽ từ Fig.6B tới Fig.6E thể hiện thứ tự xử lý. Đối với mỗi khối đã được phân chia, quy trình phân chia nhị phân-tam phân trên Fig.8 được thực hiện đệ quy (bước S1210).

Trong quy trình phân chia đệ quy các khối được mô tả ở đây, mức độ thích hợp của quy trình phân tách có thể bị giới hạn trên cơ sở số lượng phân chia, kích thước khối đích, hoặc tương tự. thông tin giới hạn mức độ thích hợp của quy trình phân tách có thể thấy ở cấu hình trong đó thông tin không được truyền đi bằng cách thực hiện thỏa thuận sơ bộ giữa thiết bị mã hóa và thiết bị giải mã, hoặc ở cấu hình trong đó thiết bị mã hóa xác định thông tin để giới hạn mức độ thích hợp của quy trình phân tách và ghi thông tin thành các chuỗi bit, nhờ đó truyền thông tin tới thiết bị giải mã.

Khi một khối nhất định được phân chia, khối trước khi phân chia được gọi là khối mẹ, và khối sau phân chia được gọi là khối con.

Tiếp theo, hoạt động của khối phân chia khối 202 trong thiết bị giải mã hình ảnh 200 sẽ được mô tả. Khối phân chia khối 202 chia khối cây bằng cách sử dụng thủ tục xử lý tương tự với thủ tục xử lý của khối phân chia khối 101 của thiết bị mã hóa hình ảnh 100. Cần lưu ý rằng có sự khác biệt rằng mặc dù khối phân chia khối 101 của thiết bị mã hóa hình ảnh 100 xác định sự phân chia tối ưu hình dạng của các khối bằng cách áp dụng phương pháp tối ưu chẳng hạn như ước lượng hình dạng tối ưu bằng nhận dạng hình ảnh hoặc tối ưu tỷ lệ méo, nhưng khối phân chia khối 202 của thiết bị giải mã hình ảnh 200 xác định sự phân chia hình dạng của các khối bằng cách giải mã sự phân chia các khối thông tin được ghi trong chuỗi bit.

Fig.9 minh họa cú pháp (các quy tắc cú pháp của chuỗi bit) liên quan đến sự phân chia các khối theo phương án thứ nhất. `Coding_quadtree()` biểu thị cú pháp quy trình phân chia khối làm bốn. `Multi_type_tree()` biểu thị cú pháp quy trình phân chia khối thành hai hoặc ba. `Qt_split` là cờ thể hiện có phân chia khối thành bốn phần hay không. Trong trường hợp phân chia khối thành bốn, sẽ đặt `qt_split = 1`. Trong trường hợp không phân chia khối thành bốn, sẽ đặt `qt_split = 0`. Trong trường hợp phân chia khối thành bốn (`qt_split = 1`), quy trình phân chia làm bốn sẽ được thực hiện một cách đệ quy trên mỗi khối đã được phân chia (`coding_quadtree (0)`, `coding_quadtree (1)`, `coding_quadtree (2)`, `coding_quadtree (3)`), trong đó các đối số từ 0 tới 3 tương ứng với các số hiệu của 601 trên Fig.6A). Trong trường hợp sự phân chia làm bốn sẽ không được thực hiện (`qt_split = 0`), sự phân chia sau đó được xác định theo `multi_type_tree()`. `mtt_split` là cờ thể hiện xem có thực hiện phân chia thêm hay không. Trong trường hợp cần thực hiện tách tiếp (`mtt_split = 1`), việc truyền `mtt_split_vertical`

là cờ thể hiện thực hiện phân chia theo phương đứng hay phương ngang và `mtt_split_binary` là cờ xác định sẽ thực hiện phân chia khối thành hai hay ba. `mtt_split_vertical = 1` thể hiện phân chia theo phương đứng, và `mtt_split_vertical = 0` thể hiện phân chia theo phương ngang. `mtt_split_binary = 1` thể hiện rằng khối được phân chia nhị phân, và `mtt_split_binary = 0` thể hiện rằng khối được phân chia tam phân. Trong trường hợp khối sẽ được phân chia nhị phân (`mtt_split_binary = 1`), quy trình phân chia được thực hiện một cách đệ quy trên mỗi khối trong số hai khối được phân chia (`multi_type_tree (0)` và `multi_type_tree (1)` trong đó các đối số từ 0 tới 1 tương ứng với các số hiệu trên 602 hoặc 604 trên các hình vẽ từ Fig.6B tới Fig.6D). Trong trường hợp khối sẽ được phân chia tam phân (`mtt_split_binary = 0`), quy trình phân chia được thực hiện một cách đệ quy trên mỗi khối trong số ba khối (`multi_type_tree (0)`, `multi_type_tree (1)`, và `multi_type_tree (2)`, trong đó các số hiệu từ 0 tới 2 tương ứng với các số hiệu trên 603 trên Fig.6B hoặc 605 trên Fig.6E). Thực hiện gọi một cách đệ quy `multi_type_tree` cho đến khi `mtt_split = 0` sẽ đạt được sự phân chia các khối theo kiểu phân cấp.

Dự đoán nội ảnh

Phương pháp dự đoán nội ảnh theo phương án của sáng chế được thực hiện trong khối dự đoán nội ảnh 103 của thiết bị mã hóa hình ảnh 100 trên Fig.1 và khối dự đoán nội ảnh 204 của thiết bị giải mã hình ảnh 200 trên Fig.2.

Phương pháp dự đoán nội ảnh theo sáng chế sẽ được mô tả cùng với các hình vẽ kèm theo. Phương pháp dự đoán nội ảnh được thực hiện trong bất kỳ quy trình mã hóa và giải mã trong các đơn vị của các khối mã hóa.

Khối dự đoán nội ảnh 103 ở phía mã hóa

Fig.40 là sơ đồ minh họa cấu hình chi tiết của khối dự đoán nội ảnh 103 của

thiết bị mã hóa hình ảnh 100 trên Fig.1. Khối dự đoán nội ảnh thông thường 351 tạo ra tín hiệu hình ảnh đã dự đoán bởi dự đoán nội ảnh thông thường từ các mẫu đã được giải mã lân cận khối mã hóa đích, lựa chọn chế độ dự đoán nội ảnh thích hợp từ nhiều chế độ dự đoán nội ảnh, và cung cấp chế độ dự đoán nội ảnh được lựa chọn và tín hiệu hình ảnh đã được dự đoán tương ứng với chế độ dự đoán nội ảnh được lựa chọn tới khối xác định phương pháp dự đoán 105. Các hình vẽ Fig.10A và Fig.10B minh họa các ví dụ về dự đoán nội ảnh. Fig.10A minh họa mối tương quan giữa hướng dự đoán của dự đoán nội ảnh thông thường và số lượng chế độ dự đoán nội ảnh. Ví dụ, chế độ dự đoán nội ảnh 50 sao chép các mẫu ở phương đứng và bởi vậy tái dựng hình ảnh dự đoán nội ảnh. Chế độ dự đoán nội ảnh 1 là Chế độ DC trong đó tất cả các giá trị mẫu của khối đích được thiết lập thành giá trị trung bình của các mẫu tham chiếu. Chế độ dự đoán nội ảnh 0 là chế độ phẳng trong đó hình ảnh dự đoán nội ảnh hai chiều được tạo ra từ các mẫu tham chiếu trong phương đứng và phương ngang. Fig.10B là ví dụ của việc tái dựng hình ảnh dự đoán nội ảnh trong trường hợp chế độ dự đoán nội ảnh 40. Đối với mỗi ví dụ của khối đích, giá trị của mẫu tham chiếu trong phương được thể hiện bởi chế độ dự đoán nội ảnh được sao chép lại. Trong trường hợp mẫu tham chiếu trong chế độ dự đoán nội ảnh không nằm ở vị trí nguyên, giá trị mẫu tham chiếu được xác định bởi phép nội suy từ các giá trị mẫu tham chiếu tại các vị trí nguyên ở lân cận.

Khối dự đoán sao chép khối nội ảnh 352 thu vùng đã được giải mã của tín hiệu hình ảnh giống với khối mã hóa đích thu được từ bộ nhớ hình ảnh đã được giải mã 104, tạo ra tín hiệu hình ảnh đã dự đoán bởi quy trình sao chép khối nội ảnh, và cung cấp tín hiệu hình ảnh đã được dự đoán tới khối xác định phương pháp dự đoán 105. Cấu hình và quy trình xử lý chi tiết của khối dự đoán sao chép khối nội ảnh 352 sẽ được mô tả chi tiết sau đây.

Khối dự đoán nội ảnh 204 ở phía giải mã

Fig.41 là sơ đồ minh họa cấu hình chi tiết của khối dự đoán nội ảnh 204 của thiết bị giải mã hình ảnh 200 trên Fig.2.

Khối dự đoán nội ảnh thông thường 361 tạo ra tín hiệu hình ảnh đã dự đoán bởi dự đoán nội ảnh thông thường từ các mẫu đã được giải mã lân cận khối mã hóa đích, lựa chọn chế độ dự đoán nội ảnh thích hợp từ nhiều chế độ dự đoán nội ảnh, và thu chế độ dự đoán nội ảnh được lựa chọn và tín hiệu hình ảnh đã được dự đoán tương ứng với chế độ dự đoán nội ảnh được lựa chọn. Tín hiệu hình ảnh đã được dự đoán này được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 thông qua chuyển mạch 364. Do quy trình xử lý của khối dự đoán nội ảnh thông thường 361 trên Fig.41 tương đương với quy trình của khối dự đoán nội ảnh thông thường 351 trên Fig.40, do đó phần mô tả chi tiết được bỏ qua.

Khối dự đoán sao chép khối nội ảnh 362 thu vùng đã được giải mã của tín hiệu hình ảnh giống với khối mã hóa đích thu được từ bộ nhớ hình ảnh đã được giải mã 208, và tạo ra tín hiệu hình ảnh đã được dự đoán bởi quy trình sao chép khối nội ảnh. Tín hiệu hình ảnh đã được dự đoán này được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 thông qua chuyển mạch 364. Cấu hình chi tiết và quy trình xử lý của khối dự đoán sao chép khối nội ảnh 362 sẽ được mô tả chi tiết sau đây.

Dự đoán liên ảnh

Phương pháp dự đoán liên ảnh theo phương án được thực hiện trong khối dự đoán liên ảnh 102 của thiết bị mã hóa hình ảnh trên Fig.1 và khối dự đoán liên ảnh 203 của thiết bị giải mã hình ảnh trên Fig.2.

Phương pháp dự đoán liên ảnh theo phương án sẽ được mô tả tham chiếu đến các hình vẽ. Phương pháp dự đoán liên ảnh được thực hiện bất kỳ quy trình mã hóa và

giải mã nào trong các đơn vị của các khối mã hóa.

Khối dự đoán liên ảnh 102 ở phía mã hóa

Fig.16 là biểu đồ minh họa cấu hình chi tiết của khối dự đoán liên ảnh 102 của thiết bị mã hóa hình ảnh trên Fig.1. Khối rút ra chế độ dự đoán vector chuyển động thông thường 301 rút ra các ứng viên dự đoán vector chuyển động thông thường, lựa chọn dự đoán vector chuyển động, và tính toán sai phân vector chuyển động giữa dự đoán vector chuyển động lựa chọn được và vector chuyển động phát hiện được. Chế độ dự đoán liên ảnh phát hiện được, chỉ số tham chiếu, vector chuyển động, và sai phân vector chuyển động tính toán được sẽ là thông tin dự đoán liên ảnh của chế độ dự đoán vector chuyển động thông thường. Thông tin dự đoán liên ảnh này được đưa tới khối xác định chế độ dự đoán liên ảnh 305. Quy trình xử lý và cấu hình chi tiết của khối rút ra chế độ dự đoán vector chuyển động thông thường 301 sẽ được mô tả sau đây.

Khối rút ra chế độ hợp nhất thông thường 302 rút ra các ứng viên hợp nhất thông thường, lựa chọn ứng viên hợp nhất thông thường, và thu được thông tin dự đoán liên ảnh của chế độ hợp nhất thông thường. Thông tin dự đoán liên ảnh này được đưa tới khối xác định chế độ dự đoán liên ảnh 305. Quy trình xử lý và cấu hình chi tiết của khối rút ra chế độ hợp nhất thông thường 302 sẽ được mô tả sau đây.

Khối rút ra chế độ dự đoán vector chuyển động khối con 303 rút ra các ứng viên dự đoán vector chuyển động khối con, lựa chọn dự đoán vector chuyển động khối con, và tính toán sai phân vector chuyển động giữa dự đoán vector chuyển động khối con lựa chọn được và vector chuyển động phát hiện được. Chế độ dự đoán liên ảnh phát hiện được, chỉ số tham chiếu, vector chuyển động, và sai phân vector chuyển động tính toán được sẽ là thông tin dự đoán liên ảnh của chế độ dự đoán vector chuyển động khối con. Thông tin dự đoán liên ảnh này được đưa tới khối xác định chế độ dự đoán

liên ảnh 305.

Khối rút ra chế độ hợp nhất khối con 304 rút ra các ứng viên hợp nhất khối con, lựa chọn ứng viên hợp nhất khối con, và thu được thông tin dự đoán liên ảnh của chế độ hợp nhất khối con. Thông tin dự đoán liên ảnh này được đưa tới khối xác định chế độ dự đoán liên ảnh 305.

Trong khối xác định chế độ dự đoán liên ảnh 305 xác định thông tin dự đoán liên ảnh dựa trên thông tin dự đoán liên ảnh được đưa tới từ khối rút ra chế độ dự đoán vector chuyển động thông thường 301, khối rút ra chế độ hợp nhất thông thường 302, khối rút ra chế độ dự đoán vector chuyển động khối con 303, và khối rút ra chế độ hợp nhất khối con 304. Thông tin dự đoán liên ảnh theo kết quả xác định được đưa tới từ khối xác định chế độ dự đoán liên ảnh 305 tới khối dự đoán bù chuyển động 306.

Khối dự đoán bù chuyển động 306 thực hiện dự đoán liên ảnh trên tín hiệu hình ảnh tham chiếu được lưu trong bộ nhớ hình ảnh đã được giải mã 104 dựa trên thông tin dự đoán liên ảnh xác định được. Quy trình xử lý và cấu hình chi tiết của khối dự đoán bù chuyển động 306 sẽ được mô tả sau đây.

Khối dự đoán liên ảnh 203 ở phía giải mã

Fig.22 là biểu đồ minh họa cấu hình chi tiết của khối dự đoán liên ảnh 203 của thiết bị giải mã hình ảnh trên Fig.2.

Khối rút ra chế độ dự đoán vector chuyển động thông thường 401 rút ra các ứng viên dự đoán vector chuyển động thông thường, lựa chọn dự đoán vector chuyển động, tính toán giá trị thêm vào thu được bằng cách thêm dự đoán vector chuyển động lựa chọn được và sai phân vector chuyển động đã được giải mã, và thiết lập giá trị thêm vào này như là vector chuyển động. chế độ dự đoán liên ảnh được giải mã, chỉ số tham chiếu, vector chuyển động sẽ là thông tin dự đoán liên ảnh của chế độ dự đoán vector

chuyển động thông thường. Thông tin dự đoán liên ảnh này được đưa tới khối dự đoán bù chuyển động 406 thông qua khối chuyển mạch 408. Quy trình xử lý và cấu hình chi tiết của khối rút ra chế độ dự đoán vector chuyển động thông thường 401 sẽ được mô tả sau đây.

Khối rút ra chế độ hợp nhất thông thường 402 rút ra các ứng viên hợp nhất thông thường, lựa chọn ứng viên hợp nhất thông thường, và thu được thông tin dự đoán liên ảnh của chế độ hợp nhất thông thường. Thông tin dự đoán liên ảnh này được đưa tới khối dự đoán bù chuyển động 406 thông qua khối chuyển mạch 408. Quy trình xử lý và cấu hình chi tiết của khối rút ra chế độ hợp nhất thông thường 402 sẽ được mô tả sau đây.

Khối rút ra chế độ dự đoán vector chuyển động khối con 403 rút ra các ứng viên dự đoán vector chuyển động khối con, lựa chọn dự đoán vector chuyển động khối con, và tính toán giá trị thêm vào thu được bằng cách thêm dự đoán vector chuyển động khối con lựa chọn được và sai phân vector chuyển động đã được giải mã, và thiết lập giá trị thêm vào này như là vector chuyển động. chế độ dự đoán liên ảnh được giải mã, chỉ số tham chiếu, và vector chuyển động sẽ là thông tin dự đoán liên ảnh của chế độ dự đoán vector chuyển động khối con. Thông tin dự đoán liên ảnh này được đưa tới khối dự đoán bù chuyển động 406 thông qua khối chuyển mạch 408.

Khối rút ra chế độ hợp nhất khối con 404 rút ra các ứng viên hợp nhất khối con, lựa chọn ứng viên hợp nhất khối con, và thu được thông tin dự đoán liên ảnh của chế độ hợp nhất khối con. Thông tin dự đoán liên ảnh này được đưa tới khối dự đoán bù chuyển động 406 thông qua khối chuyển mạch 408.

Khối dự đoán bù chuyển động 406 thực hiện dự đoán liên ảnh trên tín hiệu hình ảnh tham chiếu được lưu trong bộ nhớ hình ảnh đã được giải mã 208 dựa trên

thông tin dự đoán liên ảnh xác định được. Quy trình xử lý và cấu hình chi tiết của khối dự đoán bù chuyển động 406 tương tự với khối dự đoán bù chuyển động 306 ở phía mã hóa.

Khối rút ra chế độ dự đoán vector chuyển động thông thường (AMVP thông thường)

Khối rút ra chế độ dự đoán vector chuyển động thông thường 301 trên Fig.17 bao gồm khối rút ra ứng viên dự đoán vector chuyển động không gian 321, khối rút ra ứng viên dự đoán vector chuyển động thời gian 322, khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 323, khối chọn ứng viên dự đoán vector chuyển động 325, khối phát hiện vector chuyển động thông thường 326, khối lựa chọn ứng viên dự đoán vector chuyển động 327, và khối trừ vector chuyển động 328.

Khối rút ra chế độ dự đoán vector chuyển động thông thường 401 trên Fig.23 bao gồm khối rút ra ứng viên dự đoán vector chuyển động không gian 421, khối rút ra ứng viên dự đoán vector chuyển động thời gian 422, khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 423, khối chọn ứng viên dự đoán vector chuyển động 425, khối lựa chọn ứng viên dự đoán vector chuyển động 426, và khối cộng vector chuyển động 427.

Các thủ tục xử lý của khối rút ra chế độ dự đoán vector chuyển động thông thường 301 ở phía mã hóa và khối rút ra chế độ dự đoán vector chuyển động thông thường 401 ở phía giải mã sẽ được mô tả tham chiếu tương ứng đến các lưu đồ trên Fig.19 và Fig.25. Fig.19 là lưu đồ minh họa thủ tục xử lý rút ra chế độ dự đoán vector chuyển động thông thường được thực hiện bởi khối rút ra chế độ vector chuyển động thông thường 301 ở phía mã hóa. Fig.25 là lưu đồ minh họa thủ tục xử lý rút ra chế độ dự đoán vector chuyển động thông thường được thực hiện bởi khối rút ra chế độ vector

chuyển động thông thường 401 ở phía giải mã.

Khối rút ra chế độ dự đoán vector chuyển động thông thường (AMVP thông thường): phía mã hóa

Thủ tục xử lý rút ra chế độ dự đoán vector chuyển động thông thường ở phía mã hóa sẽ được mô tả tham chiếu đến các hình vẽ Fig.19. Trong phần mô tả thủ tục xử lý trên Fig.19, thuật ngữ “thông thường” được minh họa trên Fig.19 sẽ được lược bỏ trong một số trường hợp.

Đầu tiên, khối phát hiện vector chuyển động thông thường 326 phát hiện vector chuyển động thông thường cho mỗi chế độ dự đoán liên ảnh và các chỉ số tham chiếu (bước S100 trên Fig.19).

Sau đó, sai phân vector chuyển động của vector chuyển động được sử dụng trong dự đoán liên ảnh trong chế độ dự đoán vector chuyển động thông thường được tính toán cho mỗi L0 và L1 (từ bước S101 tới S106 trên Fig.19) trong khối rút ra ứng viên dự đoán vector chuyển động không gian 321, khối rút ra ứng viên dự đoán vector chuyển động thời gian 322, khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 323, khối độn ứng viên dự đoán vector chuyển động 325, khối lựa chọn ứng viên dự đoán vector chuyển động 327, và khối trừ vector chuyển động 328. Đặc biệt, trong trường hợp chế độ dự đoán PredMode khối đích là dự đoán liên ảnh (MODE_INTER) và chế độ dự đoán liên ảnh là dự đoán L0 (Pred_L0), sẽ tính toán được danh sách ứng viên dự đoán vector chuyển động mvpListL0 của L0. Sau đó, dự đoán vector chuyển động mvpL0 được lựa chọn, và sau đó, sai phân vector chuyển động mvdL0 của vector chuyển động mvL0 của L0 được tính toán. Trong trường hợp chế độ dự đoán liên ảnh khối đích là dự đoán L1 (Pred_L1), danh sách ứng viên dự đoán vector chuyển động mvpListL1 của L1 được tính toán. Sau đó, dự đoán vector chuyển

động $mvpL1$ được lựa chọn, và sau đó sai phân vector chuyển động $mvdL1$ của vector chuyển động $mvL1$ của $L1$ được tính toán. Trong trường hợp chế độ dự đoán liên ảnh khối đích là dự đoán kép (Pred_BI), cả dự đoán $L0$ và dự đoán $L1$ được thực hiện. danh sách ứng viên dự đoán vector chuyển động $mvpListL0$ của $L0$ được tính toán và dự đoán vector chuyển động $mvpL0$ của $L0$ được lựa chọn, và sau đó sai phân vector chuyển động $mvdL0$ của vector chuyển động $mvL0$ của $L0$ được tính toán. Cùng với việc tính toán này, danh sách ứng viên dự đoán vector chuyển động $mvpListL1$ của $L1$ được tính toán và dự đoán vector chuyển động $mvpL1$ của $L1$ được tính toán, và sau đó, sai phân vector chuyển động $mvdL1$ của vector chuyển động $mvL1$ của $L1$ được tính toán.

Quy trình tính toán sai phân vector chuyển động được thực hiện cho mỗi $L0$ và $L1$, trong đó quy trình tính toán là quy trình chung trong cả $L0$ và $L1$. Theo đó, $L0$ và $L1$ sẽ được biểu thị là LX như là thủ tục chung. Trong quy trình tính toán sai phân vector chuyển động của $L0$, X của LX được thiết lập là 0, còn trong quy trình tính toán sai phân vector chuyển động của $L1$, X của LX được thiết lập là 1. Ngoài ra, trong trường hợp thông tin trên danh sách khác được xem là thay cho một LX trong quy trình tính toán sai phân vector chuyển động của một LX , danh sách khác sẽ được biểu thị là LY .

Trong trường hợp vector chuyển động $mvLX$ của LX được sử dụng (bước S102 trên Fig.19: ĐÚNG), ứng viên dự đoán vector chuyển động của LX được tính toán, nhờ đó tạo ra danh sách ứng viên dự đoán vector chuyển động $mvpListLX$ của LX (bước S103 trên Fig.19). Trong khối rút ra chế độ dự đoán vector chuyển động thông thường 301, khối rút ra ứng viên dự đoán vector chuyển động không gian 321, khối rút ra ứng viên dự đoán vector chuyển động thời gian 322, khối rút ra ứng viên dự đoán vector

chuyển động dựa trên lịch sử 323, và khối độn ứng viên dự đoán vector chuyển động 325 rút ra các ứng viên dự đoán vector chuyển động và nhờ đó tạo ra danh sách ứng viên dự đoán vector chuyển động `mvpListLX`. thủ tục xử lý chi tiết của bước S103 trên Fig.19 sẽ được mô tả sau đây bằng cách sử dụng lưu đồ trên Fig.20.

Sau đó, khối lựa chọn ứng viên dự đoán vector chuyển động 327 lựa chọn dự đoán vector chuyển động `mvpLX` của LX từ danh sách ứng viên dự đoán vector chuyển động `mvpListLX` của LX (bước S104 trên Fig.19). Ở đây, một phần tử (phần tử thứ i tính từ 0) trong danh sách ứng viên dự đoán vector chuyển động `mvpListLX` được thể hiện là `mvpListLX[i]`. Sai phân vector chuyển động, là chênh lệch giữa vector chuyển động `mvLX` và mỗi ứng viên dự đoán vector chuyển động `mvpListLX[i]` được lưu trong danh sách ứng viên dự đoán vector chuyển động `mvpListLX`, được tính toán. Lượng mã hóa tại thời điểm mã hóa các sai phân vector chuyển động này được tính toán cho mỗi phần tử (ứng viên dự đoán vector chuyển động) của danh sách ứng viên dự đoán vector chuyển động `mvpListLX`. Sau đó, ứng viên dự đoán vector chuyển động `mvpListLX[i]` mà tối thiểu hóa lượng mã hóa cho mỗi ứng viên dự đoán vector chuyển động trong số các phần tử đơn lẻ được ghi trong danh sách ứng viên dự đoán vector chuyển động `mvpListLX` được lựa chọn làm dự đoán vector chuyển động `mvpLX`, và thu được chỉ số i của nó. Trong trường hợp có các ứng viên dự đoán vector chuyển động có lượng mã hóa được tạo ra tối thiểu trong danh sách ứng viên dự đoán vector chuyển động `mvpListLX`, ứng viên dự đoán vector chuyển động `mvpListLX[i]` có chỉ số i trong danh sách ứng viên dự đoán vector chuyển động `mvpListLX` được thể hiện bởi số nhỏ được lựa chọn làm dự đoán vector chuyển động tối ưu `mvpLX`, và thu được chỉ số i của nó.

Sau đó, khối trừ vector chuyển động 328 trừ dự đoán vector chuyển động lựa

chọn được $mvplX$ của LX từ vector chuyển động $mvLX$ của LX và tính toán sai phân vector chuyển động $mvdLX$ của LX theo biểu thức: $mvdLX = mvLX - mvplX$ (bước S105 trên Fig.19).

Khởi rút ra chế độ dự đoán vector chuyển động thông thường (AMVP thông thường): phía giải mã

Tiếp theo, thủ tục xử lý chế độ dự đoán vector chuyển động thông thường ở phía giải mã sẽ được mô tả tham chiếu đến các hình vẽ Fig.25. Ở phía giải mã, khởi rút ra ứng viên dự đoán vector chuyển động không gian 421, khởi rút ra ứng viên dự đoán vector chuyển động thời gian 422, khởi rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 423, và khởi động ứng viên dự đoán vector chuyển động 425 tính toán riêng các vector chuyển động được sử dụng trong dự đoán liên ảnh của chế độ dự đoán vector chuyển động thông thường cho mỗi $L0$ và $L1$ (từ bước S201 tới S206 trên Fig.25). Đặc biệt, trong trường hợp chế độ dự đoán $PredMode$ khởi đích là dự đoán liên ảnh ($MODE_INTER$) và chế độ dự đoán liên ảnh khởi đích là dự đoán $L0$ ($Pred_L0$), danh sách ứng viên dự đoán vector chuyển động $mvplListL0$ của $L0$ được tính toán. Sau đó, dự đoán vector chuyển động $mvplL0$ được lựa chọn, và sau đó, vector chuyển động $mvL0$ của $L0$ được tính toán. Trong trường hợp chế độ dự đoán liên ảnh khởi đích là dự đoán $L1$ ($Pred_L1$), Danh sách ứng viên dự đoán vector chuyển động $L1$ $mvplListL1$ được tính toán. Sau đó, dự đoán vector chuyển động $mvplL1$ được lựa chọn, và vector chuyển động $L1$ $mvL1$ được tính toán. Trong trường hợp chế độ dự đoán liên ảnh khởi đích là dự đoán kép ($Pred_BI$), cả dự đoán $L0$ và dự đoán $L1$ được thực hiện. Danh sách ứng viên dự đoán vector chuyển động $mvplListL0$ của $L0$ được tính toán và dự đoán vector chuyển động $mvplL0$ của $L0$ được lựa chọn, và sau đó vector chuyển động $mvL0$ của $L0$ được tính toán. Cùng với sự tính toán này, danh sách ứng viên dự đoán

vector chuyển động $mvpListL1$ của $L1$ được tính toán và dự đoán vector chuyển động $mvpL1$ của $L1$ được tính toán, và sau đó, vector chuyển động $mvL1$ của $L1$ được tính toán.

Tương tự phía mã hóa, phía giải mã thực hiện quy trình tính toán vector chuyển động cho mỗi $L0$ và $L1$, trong đó quy trình là quy trình chung trong cả $L0$ và $L1$. Theo đó, $L0$ và $L1$ sẽ được biểu thị là LX như là thủ tục chung. LX biểu thị chế độ dự đoán liên ảnh được sử dụng cho dự đoán liên ảnh của khối mã hóa đích. X là 0 trong quy trình tính toán vector chuyển động của $L0$, và X là 1 trong quy trình tính toán vector chuyển động của $L1$. Ngoài ra, trong trường hợp thông tin trên danh sách tham chiếu khác được xem là để thay cho danh sách tham chiếu giống LX sẽ được tính trong quy trình tính toán vector chuyển động của LX , danh sách tham chiếu khác sẽ được thể hiện là LY .

Trong trường hợp vector chuyển động $mvLX$ của LX được sử dụng (bước S202 trên Fig.25: ĐÚNG), ứng viên dự đoán vector chuyển động của LX được tính toán để tạo ra danh sách ứng viên dự đoán vector chuyển động $mvpListLX$ của LX (bước S203 trên Fig.25). trong khối rút ra chế độ dự đoán vector chuyển động thông thường 401, khối rút ra ứng viên dự đoán vector chuyển động không gian 421, khối rút ra ứng viên dự đoán vector chuyển động thời gian 422, khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 423, và khối độn ứng viên dự đoán vector chuyển động 425 tính toán các ứng viên dự đoán vector chuyển động và nhờ đó tạo ra danh sách ứng viên dự đoán vector chuyển động $mvpListLX$. Thủ tục xử lý chi tiết của bước S203 trên Fig.25 sẽ được mô tả sau đây bằng cách sử dụng lưu đồ trên Fig.20.

Sau đó, khối lựa chọn ứng viên dự đoán vector chuyển động 426 tách ứng viên dự đoán vector chuyển động $mvpListLX[mvpIdxLX]$ tương ứng với chỉ số dự đoán

vector chuyển động mvpldxLX đã được giải mã và được đưa tới bởi khối giải mã chuỗi bit 201 từ danh sách ứng viên dự đoán vector chuyển động mvplListLX, làm dự đoán vector chuyển động lựa chọn được mvplX (bước S204 trên Fig.25).

Sau đó, khối cộng vector chuyển động 427 cộng sai phân vector chuyển động mvdLX của LX và dự đoán vector chuyển động mvplX của LX đã được giải mã và được đưa tới bởi khối giải mã chuỗi bit 201 và tính toán vector chuyển động mvLX theo biểu thức: $mvLX = mvplX + mvdLX$ (bước S205 trên Fig.25).

Khối rút ra chế độ dự đoán vector chuyển động thông thường (AMVP thông thường): phương pháp dự đoán vector chuyển động

Fig.20 là lưu đồ minh họa thủ tục xử lý của quy trình rút ra chế độ dự đoán vector chuyển động thông thường có chức năng giống như khối rút ra chế độ dự đoán vector chuyển động thông thường 301 của thiết bị mã hóa hình ảnh và khối rút ra chế độ dự đoán vector chuyển động thông thường 401 của thiết bị giải mã hình ảnh theo phương án của sáng chế.

Mỗi khối rút ra chế độ dự đoán vector chuyển động thông thường 301 và khối rút ra chế độ dự đoán vector chuyển động thông thường 401 bao gồm danh sách ứng viên dự đoán vector chuyển động mvplListLX. Danh sách ứng viên dự đoán vector chuyển động mvplListLX có cấu trúc dạng danh sách, và bao gồm vùng lưu trữ để lưu, ở dạng các phần tử, chỉ số dự đoán vector chuyển động thể hiện vị trí trong danh sách ứng viên dự đoán vector chuyển động và ứng viên dự đoán vector chuyển động tương ứng với chỉ số. Số hiệu của chỉ số dự đoán vector chuyển động bắt đầu từ 0, và ứng viên dự đoán vector chuyển động sẽ được lưu trong vùng lưu trữ của danh sách ứng viên dự đoán vector chuyển động mvplListLX. Trong phương án này, giả sử rằng danh sách ứng viên dự đoán vector chuyển động mvplListLX có thể ghi ít nhất hai ứng viên

dự đoán vector chuyển động (dưới dạng thông tin dự đoán liên ảnh). Ngoài ra, biến numCurrMvpCand thể hiện số ứng viên dự đoán vector chuyển động được ghi trong danh sách ứng viên dự đoán vector chuyển động mvplistLX được thiết lập là 0.

Mỗi khối rút ra ứng viên dự đoán vector chuyển động không gian 321 và 421 rút ra ứng viên dự đoán vector chuyển động từ các khối ở lân cận phía bên trái. Quy trình này rút ra dự đoán vector chuyển động mvLXA với tham chiếu đến thông tin dự đoán liên ảnh của khối ở lân cận phía bên trái (A0 hoặc A1 trên Fig.11), cụ thể, cờ thể hiện ứng viên dự đoán vector chuyển động có thể được sử dụng hay không, vector chuyển động, chỉ số tham chiếu, hoặc tương tự, và cộng mvLXA được rút ra và danh sách ứng viên dự đoán vector chuyển động mvplistLX (bước S301 trên Fig.20). Lưu ý rằng X là 0 trong dự đoán L0 và X là 1 trong dự đoán L1 (áp dụng tương tự cho phần sau đây). Sau đó, khối rút ra ứng viên dự đoán vector chuyển động không gian 321 và 421 rút ra ứng viên dự đoán vector chuyển động từ khối lân cận phía trên. Quy trình này rút ra dự đoán vector chuyển động mvLXB với tham chiếu đến thông tin dự đoán liên ảnh của khối lân cận phía trên (B0, B1 hoặc B2 trên Fig.11), cụ thể, cờ thể hiện ứng viên dự đoán vector chuyển động có thể được sử dụng hay không, vector chuyển động, chỉ số tham chiếu, hoặc tương tự, Khi mvLXA được rút ra và mvLXB được rút ra không bằng nhau, mvLXB được thêm vào danh sách ứng viên dự đoán vector chuyển động mvplistLX (bước S302 trên Fig.20). Quy trình trong bước S301 và S302 trên Fig.20 được đưa ra là quy trình chung ngoại trừ vị trí và số lượng các khối tham chiếu lân cận khác nhau, và cờ availableFlagLXN thể hiện ứng viên dự đoán vector chuyển động của khối mã hóa có thể được sử dụng hay không, và vector chuyển động mvLXN, chỉ số tham chiếu refIdxN (N thể hiện hoặc B, áp dụng tương tự cho phần sau đây) sẽ được rút ra trong các quy trình này.

Sau đó, mỗi khối rút ra ứng viên dự đoán vector chuyển động thời gian 322 và 422 rút ra ứng viên dự đoán vector chuyển động từ khối trong hình ảnh có khác biệt về thời gian so với hình ảnh đích hiện tại. Quy trình này rút ra cờ availableFlagLXCol thể hiện ứng viên dự đoán vector chuyển động của khối mã hóa của hình ảnh có khác biệt về thời gian có thể được sử dụng, và vector chuyển động mvLXCol, chỉ số tham chiếu refIdxCol, và danh sách tham chiếu listCol, và cộng mvLXCol vào danh sách ứng viên dự đoán vector chuyển động mvplistLX (bước S303 trên Fig.20).

Lưu ý rằng giả sử rằng quy trình của khối rút ra ứng viên dự đoán vector chuyển động thời gian 322 và 422 có thể được bỏ qua trong các đơn vị của chuỗi (SPS), hình ảnh (PPS), hoặc lát.

Sau đó, khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 323 và 423 thêm các ứng viên dự đoán vector chuyển động dựa trên lịch sử được ghi trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList vào danh sách ứng viên dự đoán vector chuyển động mvplistLX. (Bước S304 trên Fig.20). Chi tiết về thủ tục xử lý ghi nhận trong bước S304 sẽ được mô tả sau đây tham chiếu đến lưu đồ trên Fig.29.

Sau đó, khối chọn ứng viên dự đoán vector chuyển động 325 và 425 thêm ứng viên dự đoán vector chuyển động có giá trị định trước chẳng hạn như (0, 0) cho đến khi danh sách ứng viên dự đoán vector chuyển động mvplistLX được thỏa mãn (S305 trên Fig.20).

Khối rút ra chế độ hợp nhất thông thường (hợp nhất thông thường)

Khối rút ra chế độ hợp nhất thông thường 302 trên Fig.18 bao gồm khối rút ra ứng viên hợp nhất không gian 341, khối rút ra ứng viên hợp nhất theo thời gian 342, khối rút ra ứng viên hợp nhất trung bình 344, khối rút ra ứng viên hợp nhất dựa trên

lịch sử 345, khối độn ứng viên hợp nhất 346, và khối lựa chọn ứng viên hợp nhất 347.

Khối rút ra chế độ hợp nhất thông thường 402 trên Fig.24 bao gồm khối rút ra ứng viên hợp nhất không gian 441, khối rút ra ứng viên hợp nhất theo thời gian 442, khối rút ra ứng viên hợp nhất trung bình 444, khối rút ra ứng viên hợp nhất dựa trên lịch sử 445, khối độn ứng viên hợp nhất 446, và khối lựa chọn ứng viên hợp nhất 447.

Fig.21 là lưu đồ minh họa thủ tục của quy trình rút ra chế độ hợp nhất thông thường có chức năng giống như khối rút ra chế độ hợp nhất thông thường 302 của thiết bị mã hóa hình ảnh và khối rút ra chế độ hợp nhất thông thường 402 của thiết bị giải mã hình ảnh theo phương án của sáng chế.

Sau đây, các quy trình khác nhau sẽ được mô tả theo từng bước. Phần mô tả sau đây là trường hợp kiểu lát lát_type là Lát B trừ khi quy định khác đi. Tuy nhiên, sáng chế cũng có thể được áp dụng cho trường hợp lát P. Lưu ý rằng, chỉ có dự đoán L0 (Pred_L0) là chế độ dự đoán liên ảnh, mà không có dự đoán L1 (Pred_L1) hoặc dự đoán kép (Pred_BI) trong trường hợp kiểu lát lát_type là lát P. Theo đó, có thể bỏ qua thủ tục liên quan đến L1 trong trường hợp này.

Khối rút ra chế độ hợp nhất thông thường 302 và khối rút ra chế độ hợp nhất thông thường 402 bao gồm danh sách ứng viên hợp nhất mergeCandList. Danh sách ứng viên hợp nhất mergeCandList có cấu trúc dạng danh sách, và bao gồm vùng lưu trữ để lưu, ở dạng các phần tử, chỉ số hợp nhất thể hiện vị trí trong danh sách ứng viên hợp nhất và ứng viên hợp nhất tương ứng với chỉ số. Số hiệu chỉ số hợp nhất bắt đầu từ 0, và ứng viên hợp nhất được lưu trong vùng lưu trữ của danh sách ứng viên hợp nhất mergeCandList. Trong quy trình xử lý tiếp theo, ứng viên hợp nhất của chỉ số hợp nhất i được ghi trong danh sách ứng viên hợp nhất mergeCandList sẽ là được thể hiện bởi mergeCandList[i]. Trong phương án này, giả sử rằng danh sách ứng viên hợp nhất

mergeCandList có thể ghi tối thiểu sáu ứng viên hợp nhất (ở dạng thông tin dự đoán liên ảnh). Ngoài ra, biến numCurrMergeCand thể hiện số hiệu ứng viên hợp nhất được ghi trong danh sách ứng viên hợp nhất mergeCandList được thiết lập là 0.

Khối rút ra ứng viên hợp nhất không gian 341 và khối rút ra ứng viên hợp nhất không gian 441 rút ra ứng viên hợp nhất không gian của mỗi khối (B1, A1, B0, A0, B2 trên Fig.11) ở lân cận khối đích theo thứ tự B1, A1, B0, A0, và B2, từ thông tin mã hóa được lưu hoặc trong bộ nhớ lưu trữ thông tin mã hóa 111 của thiết bị mã hóa hình ảnh hoặc trong bộ nhớ lưu trữ thông tin mã hóa 205 của thiết bị giải mã hình ảnh, và sau đó, ghi các ứng viên hợp nhất theo không gian được rút ra vào danh sách ứng viên hợp nhất mergeCandList (bước S401 trên Fig.21). Ở đây, N thể hiện một trong số B1, A1, B0, A0, B2 hoặc ứng viên hợp nhất theo thời gian Col sẽ được xác định. Các biến sẽ được rút ra bao gồm cờ availableFlagN thể hiện thông tin dự đoán liên ảnh của khối N có thể được sử dụng là ứng viên hợp nhất không gian hay không, chỉ số tham chiếu refIdxL0N của L0 và chỉ số tham chiếu refIdxL1N của L1 của ứng viên hợp nhất không gian N, cờ dự đoán L0 predFlagL0N thể hiện dự đoán L0 có được thực hiện hay không, cờ dự đoán L1 predFlagL1N thể hiện dự đoán L1 có được thực hiện hay không, vectơ chuyển động mvL0N của L0, và vectơ chuyển động mvL1N của L1. Tuy nhiên, do ứng viên hợp nhất trong phương án này được rút ra mà không tham chiếu tới thông tin dự đoán liên ảnh của khối có trong khối mã hóa đích, sẽ không rút ra được ứng viên hợp nhất không gian bằng cách sử dụng thông tin dự đoán liên ảnh của khối có trong khối mã hóa đích.

Sau đó, khối rút ra ứng viên hợp nhất theo thời gian 342 và khối rút ra ứng viên hợp nhất theo thời gian 442 rút ra các ứng viên hợp nhất theo thời gian từ hình ảnh có khác biệt về thời gian, và ghi các ứng viên hợp nhất theo thời gian rút ra được

vào danh sách ứng viên hợp nhất mergeCandList (bước S402 trên Fig.21). Các biến được rút ra bao gồm cờ availableFlagCol thể hiện ứng viên hợp nhất theo thời gian có thể được sử dụng hay không, cờ dự đoán L0 predFlagL0Col thể hiện dự đoán L0 của ứng viên hợp nhất theo thời gian có được thực hiện hay không, cờ dự đoán L1 predFlagL1Col thể hiện dự đoán L1 có được thực hiện hay không, và vector chuyển động mvL0Col của L0, và vector chuyển động mvL1Col của L1.

Lưu ý rằng giả định rằng quy trình của khối rút ra ứng viên hợp nhất theo thời gian 342 và 442 có thể được bỏ qua trong các đơn vị của trình tự (SPS), hình ảnh (PPS), hoặc lát.

Sau đó, khối rút ra ứng viên hợp nhất dựa trên lịch sử 345 và khối rút ra ứng viên hợp nhất dựa trên lịch sử 445 ghi các ứng viên dự đoán vector chuyển động dựa trên lịch sử được ghi trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList, vào danh sách ứng viên hợp nhất mergeCandList (bước S403 trên Fig.21).

Trong trường hợp số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList nhỏ hơn số ứng viên hợp nhất tối đa MaxNumMergeCand, ứng viên hợp nhất dựa trên lịch sử được rút ra cùng với số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList bị giới hạn ở số lượng ứng viên hợp nhất tối đa MaxNumMergeCand, và sau đó được ghi vào danh sách ứng viên hợp nhất mergeCandList.

Sau đó, khối rút ra ứng viên hợp nhất trung bình 344 và khối rút ra ứng viên hợp nhất trung bình 444 rút ra ứng viên hợp nhất trung bình từ danh sách ứng viên hợp nhất mergeCandList, và thêm ứng viên hợp nhất trung bình rút ra được vào danh sách ứng viên hợp nhất mergeCandList (bước S404 trên Fig.21).

Trong trường hợp số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList nhỏ hơn số lượng ứng viên hợp nhất tối đa MaxNumMergeCand , ứng viên hợp nhất trung bình được rút ra cùng với số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList bị giới hạn ở số lượng ứng viên hợp nhất tối đa MaxNumMergeCand , và sau đó được ghi vào danh sách ứng viên hợp nhất mergeCandList .

Ở đây, ứng viên hợp nhất trung bình là ứng viên hợp nhất mới bao gồm vector chuyển động thu được bằng cách lấy trung bình các vector chuyển động của ứng viên hợp nhất thứ nhất và ứng viên hợp nhất thứ hai được ghi trong danh sách ứng viên hợp nhất mergeCandList cho mỗi dự đoán $L0$ và dự đoán $L1$.

Sau đó, trong khối độn ứng viên hợp nhất 346 và khối độn ứng viên hợp nhất 446, trong trường hợp số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList nhỏ hơn số lượng ứng viên hợp nhất tối đa MaxNumMergeCand , ứng viên hợp nhất bổ sung được rút ra cùng với số ứng viên hợp nhất numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất mergeCandList bị giới hạn ở số lượng ứng viên hợp nhất tối đa MaxNumMergeCand , và sau đó được ghi vào danh sách ứng viên hợp nhất mergeCandList (bước S405 trên Fig.21). Trong lát P, ứng viên hợp nhất có vector chuyển động có giá trị $(0, 0)$ và chế độ dự đoán của dự đoán $L0$ (Pred_L0) được thêm vào cùng với số lượng ứng viên hợp nhất tối đa MaxNumMergeCand như là giới hạn trên. Trong B lát, ứng viên hợp nhất có chế độ dự đoán của dự đoán kép (Pred_BI) và vector chuyển động có giá trị $(0, 0)$ được thêm vào. Chỉ số tham chiếu tại thời điểm bổ sung ứng viên hợp nhất khác với chỉ số tham chiếu đã được thêm.

Sau đó, khối lựa chọn ứng viên hợp nhất 347 và khối lựa chọn ứng viên hợp

nhất 447 lựa chọn ứng viên hợp nhất trong số các ứng viên hợp nhất được ghi trong danh sách ứng viên hợp nhất mergeCandList. Khối lựa chọn ứng viên hợp nhất 347 ở phía mã hóa tính toán lượng mã hóa và lượng méo, và nhờ đó lựa chọn ứng viên hợp nhất, và sau đó, đưa chỉ số hợp nhất thể hiện ứng viên hợp nhất chọn được và thông tin dự đoán liên ảnh của ứng viên hợp nhất tới khối dự đoán bù chuyển động 306 thông qua khối xác định chế độ dự đoán liên ảnh 305. Ngược lại, khối lựa chọn ứng viên hợp nhất 447 ở phía giải mã lựa chọn ứng viên hợp nhất dựa trên chỉ số hợp nhất đã được giải mã, và đưa ứng viên hợp nhất chọn được tới khối dự đoán bù chuyển động 406.

Cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử

Tiếp theo, là phần mô tả chi tiết quy trình khởi tạo và cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList có trong bộ nhớ lưu trữ thông tin mã hóa 111 ở phía mã hóa và bộ nhớ lưu trữ thông tin mã hóa 205 ở phía giải mã. Fig.26 là lưu đồ minh họa thủ tục khởi tạo/cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Trong phương án này, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được cập nhật trong bộ nhớ lưu trữ thông tin mã hóa 111 và bộ nhớ lưu trữ thông tin mã hóa 205. Ngoài ra, khối cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử có thể là có trong khối dự đoán liên ảnh 102 và khối dự đoán liên ảnh 203 để cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList.

Các thiết lập ban đầu của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được thực hiện ở phần đầu của lát. Ở phía mã hóa, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được cập nhật trong trường hợp chế độ dự đoán vector chuyển động thông thường hoặc chế độ

hợp nhất thông thường được lựa chọn bởi khối xác định phương pháp dự đoán 105. Ở phía giải mã, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được cập nhật trong trường hợp thông tin dự đoán được giải mã bởi khối giải mã chuỗi bit 201 là chế độ dự đoán vector chuyển động thông thường hoặc chế độ hợp nhất thông thường.

Thông tin dự đoán liên ảnh được sử dụng tại thời điểm thực hiện dự đoán liên ảnh trong chế độ dự đoán vector chuyển động thông thường hoặc chế độ hợp nhất thông thường sẽ được ghi trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList, ở dạng ứng viên thông tin dự đoán liên ảnh hMvpCand. Ứng viên thông tin dự đoán liên ảnh hMvpCand bao gồm chỉ số tham chiếu refIdxL0 của L0 và chỉ số tham chiếu refIdxL1 của L1, cờ dự đoán predFlagL0 thể hiện dự đoán L0 có được thực hiện hay không, cờ dự đoán predFlagL1 thể hiện dự đoán L1 có được thực hiện hay không, vector chuyển động mvL0 của L0 và vector chuyển động mvL1 của L1.

Trong trường hợp có thông tin dự đoán liên ảnh có giá trị giống với ứng viên thông tin dự đoán liên ảnh hMvpCand trong số các phần tử (tức là, thông tin dự đoán liên ảnh) được ghi trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList có trong bộ nhớ lưu trữ thông tin mã hóa 111 ở phía mã hóa và bộ nhớ lưu trữ thông tin mã hóa 205 ở phía giải mã, phần tử sẽ được xóa khỏi danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList. Ngược lại, trong trường hợp không có thông tin dự đoán liên ảnh có giá trị giống với ứng viên thông tin dự đoán liên ảnh hMvpCand, phần tử đầu của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList sẽ bị xóa đi, và ứng viên thông tin dự đoán liên ảnh hMvpCand sẽ được thêm vào cuối danh sách ứng viên dự đoán vector

chuyển động dựa trên lịch sử HmvpCandList.

Số phần tử của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList có trong bộ nhớ lưu trữ thông tin mã hóa 111 ở phía mã hóa và bộ nhớ lưu trữ thông tin mã hóa 205 ở phía giải mã của sáng chế được thiết lập là sáu.

Đầu tiên, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được khởi tạo trong các khối của lát (bước S2101 trên Fig.26). Tất cả phần tử của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList không có ở phần đầu của lát, và số NumHmvpCand (số ứng viên hiện tại) của các ứng viên dự đoán vector chuyển động dựa trên lịch sử được ghi trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được thiết lập là 0.

Mặc dù việc khởi tạo danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList sẽ được thực hiện trong các khối của lát (khối mã hóa thứ nhất của lát), việc khởi tạo có thể được thực hiện trong các khối của các hình ảnh, gạch, hoặc các hàng khối cây.

Sau đó, quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList sau đây được thực hiện lặp lại cho mỗi khối mã hóa trong lát (từ bước S2102 tới S2107 trên Fig.26).

Đầu tiên, các thiết lập ban đầu được thực hiện trong các đơn vị của các khối mã hóa. Cờ 'identicalCandExist' thể hiện tồn tại ứng viên giống nhau có được thiết lập là giá trị FALSE (sai) hay không, chỉ số đích xóa 'removeIdx' thể hiện ứng viên đích xóa được thiết lập là 0 (bước S2103 trên Fig.26).

Xác định được xem có ứng viên thông tin dự đoán liên ảnh hMvpCand được ghi hay không (bước S2104 trên Fig.26). Trong trường hợp khối xác định phương

pháp dự đoán 105 ở phía mã hóa xác định chế độ dự đoán vector chuyển động thông thường hoặc chế độ hợp nhất thông thường, hoặc khối giải mã chuỗi bit 201 ở phía giải mã thực hiện giải mã ở dạng chế độ dự đoán vector chuyển động thông thường hoặc chế độ hợp nhất thông thường, thông tin dự đoán liên ảnh tương ứng được thiết lập là ứng viên thông tin dự đoán liên ảnh hMvpCand sẽ được ghi. Trong trường hợp khối xác định phương pháp dự đoán 105 ở phía mã hóa xác định chế độ dự đoán nội ảnh, chế độ dự đoán vector chuyển động khối con hoặc chế độ hợp nhất khối con, hoặc trong trường hợp khối giải mã chuỗi bit 201 ở phía giải mã thực hiện giải mã như chế độ dự đoán nội ảnh, chế độ dự đoán vector chuyển động khối con, hoặc chế độ hợp nhất khối con, quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList sẽ không được thực hiện, và không có ứng viên thông tin dự đoán liên ảnh hMvpCand nào được ghi. Trong trường hợp không có ứng viên thông tin dự đoán liên ảnh hMvpCand nào được ghi, sẽ bỏ qua các bước từ bước S2105 tới S2106 (bước S2104 trên Fig.26: SAI). Trong trường hợp có ứng viên thông tin dự đoán liên ảnh hMvpCand sẽ được ghi, quy trình từ bước S2105 trở đi sẽ được thực hiện (bước S2104 trên Fig.26: ĐÚNG).

Sau đó, xác định xem các phần tử đơn lẻ của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList bao gồm phần tử (thông tin dự đoán liên ảnh) có giá trị giống với ứng viên thông tin dự đoán liên ảnh hMvpCand sẽ được ghi, tức là, có tồn tại phần tử giống nhau không (bước S2105 trên Fig.26). Fig.27 là lưu đồ thủ tục xử lý để xác nhận các phần tử giống nhau. Trong trường hợp giá trị của số các ứng viên dự đoán vector chuyển động dựa trên lịch sử NumHmvpCand là 0 (bước S2121: SAI trên Fig.27), danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList trống, và không tồn tại ứng viên giống nhau. Theo đó, sẽ

bỏ qua các bước từ S2122 tới S2125 trên Fig.27, hoàn tất thủ tục xử lý để xác nhận phần tử giống nhau. Trong trường hợp giá trị của số NumHmvpCand của các ứng viên dự đoán vector chuyển động dựa trên lịch sử lớn hơn 0 (ĐÚNG trong bước S2121 trên Fig.27), quy trình bước S2123 sẽ được lặp lại từ chỉ số dự đoán vector chuyển động dựa trên lịch sử hMvpIdx từ 0 tới NumHmvpCand-1 (từ bước S2122 tới S2125 trên Fig.27). Đầu tiên, việc so sánh được thực hiện xem phần tử thứ hMvpIdx HmvpCandList[hMvpIdx] tính từ 0 trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử giống với ứng viên thông tin dự đoán liên ảnh hMvpCand (bước S2123 trên Fig.27). Trong trường hợp chúng giống nhau (bước S2123 trên Fig.27: ĐÚNG), cờ identicalCandExist thể hiện tồn tại ứng viên giống nhau có được thiết lập là giá trị ĐÚNG (TRUE), và chỉ số đích xóa removeIdx thể hiện vị trí của phần tử sẽ được xóa được thiết lập là giá trị hiện tại của chỉ số dự đoán vector chuyển động dựa trên lịch sử hMvpIdx, và quy trình xác nhận phần tử giống nhau sẽ kết thúc. Trong trường hợp chúng không giống nhau (bước S2123 trên Fig.27: SAI), hMvpIdx tăng lên một. Trong trường hợp chỉ số dự đoán vector chuyển động dựa trên lịch sử hMvpIdx nhỏ hơn hoặc bằng NumHmvpCand-1, thực hiện xử lý các bước từ S2123 trở về sau.

Quay trở lại lưu đồ trên Fig.26, quy trình dịch và thêm phần tử của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được thực hiện (bước S2106 trên Fig.26). Fig.28 là lưu đồ của thủ tục xử lý dịch/thêm phần tử của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList trong bước S2106 trên Fig.26. Đầu tiên, xác định xem có thêm phần tử mới sau khi loại bỏ phần tử được lưu trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList, hay thêm phần tử mới mà không loại bỏ phần tử. Đặc biệt, việc so sánh được thực hiện xem cờ identicalCandExist thể hiện tồn tại ứng viên giống nhau

có là ĐÚNG hay không, và NumHmvpCand có là 6 hay không (bước S2141 trên Fig.28). Trong trường hợp một trong các điều kiện là cờ identicalCandExist thể hiện tồn tại ứng viên giống nhau là ĐÚNG hoặc số ứng viên hiện tại NumHmvpCand là 6 được thỏa mãn (bước S2141 trên Fig.28: ĐÚNG), phần tử được lưu trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được loại bỏ và sau đó phần tử mới sẽ được thêm vào. giá trị ban đầu của chỉ số i được thiết lập là giá trị removeIdx + 1. Quy trình dịch phần tử bước S2143 được lặp lại từ giá trị khởi tạo này tới NumHmvpCand. (Từ bước S2142 tới S2144 trên Fig.28). Bằng việc sao chép phần HmvpCandList[i] vào HmvpCandList[i-1], các phần tử được dịch lên (bước S2143 trên Fig.28) và i tăng lên một (từ bước S2142 tới S2144 trên Fig.28). Sau đó, ứng viên thông tin dự đoán liên ảnh hMvpCand thứ (NumHmvpCand-1) được thêm vào HmvpCandList [NumHmvpCand-1] từ 0 tương ứng cuối danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử (bước S2145 trên Fig.28), và quy trình dịch/thêm phần tử vào danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList sẽ được hoàn tất. Ngược lại, trong trường hợp không có các điều kiện là cờ identicalCandExist thể hiện xem tồn tại ứng viên giống nhau có ĐÚNG hay không và NumHmvpCand là 6 được đáp ứng (bước S2141 trên Fig.28: SAI), ứng viên thông tin dự đoán liên ảnh hMvpCand sẽ được thêm vào cuối danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử mà không loại bỏ phần tử được lưu trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList (bước S2146 trên Fig.28). Ở đây, cuối danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử là HmvpCandList [NumHmvpCand] thứ NumHmvpCand tính từ 0. Ngoài ra, NumHmvpCand tăng lên một, và quy trình dịch và thêm phần tử vào danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được kết thúc.

Fig.31 là hình minh họa ví dụ về quy trình cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử. Trong trường hợp phần tử mới được thêm vào danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList trong đó sáu phần tử (thông tin dự đoán liên ảnh) đã được ghi, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được so sánh với thông tin dự đoán liên ảnh mới theo thứ tự từ phần tử đầu (Fig.31A). Khi phần tử mới có giá trị giống với phần tử thứ ba HMVP2 từ đầu danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList, phần tử HMVP2 bị xóa khỏi danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList và sau đó các phần tử từ HMVP3 tới HMVP5 được dịch (được sao chép) lên một, và phần tử mới được thêm vào cuối danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList (Fig.31B) để hoàn tất việc cập nhật danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList (Fig.31C).

Quy trình rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử

Tiếp theo, là phần mô tả chi tiết phương pháp rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử từ danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList. Thủ tục này tương ứng với thủ tục xử lý của bước S304 trên Fig.20 liên quan đến quy trình chung được thực hiện bởi khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 323 của khối rút ra chế độ dự đoán vector chuyển động thông thường 301 ở phía mã hóa và khối rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử 423 của khối rút ra chế độ dự đoán vector chuyển động thông thường 401 ở phía giải mã. Fig.29 là lưu đồ minh họa thủ tục xử lý rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử.

Trong trường hợp số ứng viên dự đoán vector chuyển động hiện tại

numCurrMvpCand lớn hơn hoặc bằng số phần tử tối đa của danh sách ứng viên dự đoán vector chuyển động.mvpListLX (ở đây, 2), hoặc số các ứng viên dự đoán vector chuyển động dựa trên lịch sử NumHmvpCand là 0 (bước S2201 trên Fig.29: SAI), các bước từ S2202 tới S2209 trên Fig.29 sẽ được bỏ qua, và thủ tục xử lý rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử sẽ được hoàn tất. Trong trường hợp số lượng hiện tại numCurrMvpCand của các ứng viên dự đoán vector chuyển động nhỏ hơn 2, là số phần tử tối đa của danh sách ứng viên dự đoán vector chuyển động.mvpListLX, và trong trường hợp giá trị của số NumHmvpCand các ứng viên dự đoán vector chuyển động dựa trên lịch sử lớn hơn 0 (bước S2201 trên Fig.29: ĐÚNG), các bước từ S2202 tới S2209 trên Fig.29 sẽ được thực hiện.

Sau đó, các bước từ S2203 tới S2208 trên Fig.29 được lặp lại cho đến khi chỉ số i là từ 1 tới nhỏ hơn 4 hoặc số các ứng viên dự đoán vector chuyển động dựa trên lịch sử numCheckedHMOVPCand (các bước từ S2202 tới S2209 trên Fig.29). Trong trường hợp số ứng viên dự đoán vector chuyển động hiện tại numCurrMvpCand lớn hơn hoặc bằng 2, là số phần tử tối đa của danh sách ứng viên dự đoán vector chuyển động.mvpListLX (bước S2203 trên Fig.29: SAI), quy trình từ các bước từ S2204 tới S2209 trên Fig.29 sẽ được bỏ qua và thủ tục xử lý rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử sẽ được hoàn tất. Trong trường hợp số ứng viên dự đoán vector chuyển động hiện tại numCurrMvpCand nhỏ hơn 2 là số các phần tử trong danh sách ứng viên dự đoán vector chuyển động tối đa.mvpListLX (bước S2203 trên Fig.29: ĐÚNG), quy trình trong các bước từ S2204 trở đi trên Fig.29 sẽ được thực hiện.

Sau đó, quy trình trong các bước từ S2205 tới S2207 được thực hiện đối với trường hợp Y là 0 và Y là 1 (L0 và L1) (các bước từ S2204 tới S2208 trên Fig.29). Trong trường hợp số ứng viên dự đoán vector chuyển động hiện tại numCurrMvpCand

lớn hơn hoặc bằng 2, là số phần tử tối đa của danh sách ứng viên dự đoán vector chuyển động.mvpListLX (bước S2205 trên Fig.29: SAI), quy trình từ các bước từ S2206 tới S2209 trên Fig.29 sẽ được bỏ qua và thủ tục xử lý rút ra ứng viên dự đoán vector chuyển động dựa trên lịch sử sẽ được hoàn tất. Trong trường hợp số ứng viên dự đoán vector chuyển động hiện tại numCurrMvpCand nhỏ hơn 2 là số các phần tử trong danh sách ứng viên dự đoán vector chuyển động tối đa.mvpListLX (bước S2205 trên Fig.29: ĐÚNG), quy trình trong các bước từ S2206 trở đi trên Fig.29 sẽ được thực hiện.

Tiếp theo, trong trường hợp danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList bao gồm phần tử có cùng chỉ số tham chiếu với chỉ số tham chiếu.refIdxLX của vector chuyển động đích mã hóa/giải mã và khác với bất kỳ phần tử nào trong danh sách dự đoán vector chuyển động.mvpListLX (bước S2206: ĐÚNG trên Fig.29), vector chuyển động của LY của ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList [NumHmvpCand-i] được thêm vào phần tử thứ numCurrMvpCand.mvpListLX[numCurrMvpCand] tính từ 0 trong danh sách ứng viên dự đoán vector chuyển động (bước S2207 trên Fig.29), và số lượng hiện tại các ứng viên dự đoán vector chuyển động numCurrMvpCand tăng lên một. Trong trường hợp không có phần tử trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList mà có cùng chỉ số tham chiếu với chỉ số tham chiếu.refIdxLX của vector chuyển động đích mã hóa/giải mã và khác với phần tử bất kỳ của danh sách dự đoán vector chuyển động.mvpListLX (bước S2206 trên Fig.29: SAI), quy trình thêm trong bước S2207 sẽ được bỏ qua.

Quy trình các bước từ S2205 tới S2207 trên Fig.29 được thực hiện cho cả L0 và L1 (các bước từ S2204 tới S2208 trên Fig.29). Chỉ số i tăng lên một, và khi chỉ số i

nhỏ hơn hoặc bằng 4 hoặc số các ứng viên dự đoán vector chuyển động dựa trên lịch sử NumHmvpCand, quy trình bước S2203 trở đi sẽ được thực hiện lại (các bước từ S2202 tới S2209 trên Fig.29).

Thủ tục rút ra ứng viên hợp nhất dựa trên lịch sử

Sau đây là phần mô tả chi tiết phương pháp rút ra ứng viên hợp nhất dựa trên lịch sử từ danh sách ứng viên hợp nhất dựa trên lịch sử HmvpCandList, thủ tục xử lý của bước S404 trên Fig.21, là quy trình chung, khối rút ra ứng viên hợp nhất dựa trên lịch sử 345 của khối rút ra chế độ hợp nhất thông thường 302 ở phía mã hóa và khối rút ra ứng viên hợp nhất dựa trên lịch sử 445 của khối rút ra chế độ hợp nhất thông thường 402 ở phía giải mã. Fig.30 là lưu đồ minh họa thủ tục xử lý để rút ra ứng viên hợp nhất dựa trên lịch sử.

Đầu tiên, quy trình khởi tạo được thực hiện (bước S2301 trên Fig.30). Mỗi phần tử từ 0 tới (numCurrMergeCand - 1) của isPruned[i] được thiết lập là giá trị SAI, và biến numOrigMergeCand được thiết lập là số numCurrMergeCand của số các phần tử được ghi trong danh sách ứng viên hợp nhất hiện tại.

Sau đó, giá trị ban đầu của chỉ số hMvpIdx được thiết lập là 1, và quy trình thêm vào từ bước S2303 tới bước S2310 trên Fig.30 được lặp lại từ giá trị khởi tạo này tới NumHmvpCand (các bước từ S2302 tới S2311 trên Fig.30). Khi số numCurrMergeCand của các phần tử được ghi trong danh sách ứng viên hợp nhất hiện tại không nhỏ hơn hoặc bằng (số lượng ứng viên hợp nhất tối đa MaxNumMergeCand - 1), các ứng viên hợp nhất đã được thêm vào tất cả các phần tử trong danh sách ứng viên hợp nhất. Theo đó, thủ tục rút ra ứng viên hợp nhất dựa trên lịch sử sẽ được hoàn tất (SAI trong bước S2303 trên Fig.30). Trong trường hợp số numCurrMergeCand các phần tử được ghi trong danh sách ứng viên hợp nhất hiện tại nhỏ hơn hoặc bằng (số

lượng ứng viên hợp nhất tối đa $\text{MaxNumMergeCand}-1$), quy trình bước S2304 trở đi sẽ được thực hiện. sameMotion được thiết lập là giá trị của SAI (bước S2304 trên Fig.30). Sau đó, giá trị ban đầu của chỉ số i được thiết lập là 0, và quy trình các bước từ S2306 và S2307 trên Fig.30 được thực hiện từ giá trị khởi tạo này tới $\text{numOrigMergeCand}-1$ (các bước từ S2305 tới S2308 trên Fig.30). Việc so sánh được thực hiện để xem phần tử thứ $(\text{NumHmvpCand}-\text{hMvpIdx})$ $\text{HmvpCandList}[\text{NumHmvpCand}-\text{hMvpIdx}]$ tính từ 0 trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử là giá trị giống với phần tử thứ i $\text{mergeCandList}[i]$ tính từ 0 trong danh sách ứng viên hợp nhất (bước S2306 trên Fig.30).

Các ứng viên hợp nhất được xác định có cùng trong trường hợp tất cả các phần tử thành phần (chế độ dự đoán liên ảnh, chỉ số tham chiếu, vector chuyển động) của ứng viên hợp nhất có cùng giá trị. Trong trường hợp các ứng viên hợp nhất có cùng giá trị và $\text{isPruned}[i]$ được thiết lập là SAI (ĐÚNG trong bước S2306 trên Fig.30), cả sameMotion và $\text{isPruned}[i]$ sẽ được đặt là ĐÚNG (bước S2307 trên Fig.30). Trong trường hợp các giá trị không giống nhau (SAI trong bước S2306 trên Fig.30), quy trình trong bước S2307 sẽ được bỏ qua. Sau khi hoàn tất việc lặp lại các bước từ S2305 tới bước S2308 trên Fig.30, việc so sánh được thực hiện xem sameMotion có là SAI hay không (bước S2309 trên Fig.30). Trong trường hợp sameMotion là SAI (bước S2309: ĐÚNG trên Fig.30), tức là, phần tử thứ $(\text{NumHmvpCand}-\text{hMvpId})$ $\text{HmvpCandList}[\text{NumHmvpCand}-\text{hMvpIdx}]$ tính từ 0 trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử không có trong mergeCandList , và vì vậy, phần tử $\text{HmvpCandList}[\text{NumHmvpCand}-\text{hMvpIdx}]$ tức là tính từ 0 của danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử được thêm vào $\text{mergeCandList}[\text{numCurrMergeCand}]$ tức là phần tử thứ numCurrMergeCand trong

danh sách ứng viên hợp nhất, và numCurrMergeCand tăng lên một (bước S2310 trên Fig.30). Chỉ số hMvpIdx tăng lên một (bước S2302 trên Fig.30), và quy trình các bước từ S2302 tới S2311 trên Fig.30 được lặp lại.

Sau khi hoàn tất việc xác nhận tất cả các phần tử trong danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử hoặc hoàn tất việc thêm ứng viên hợp nhất vào tất cả các phần tử trong danh sách ứng viên hợp nhất, thủ tục rút ra ứng viên hợp nhất dựa trên lịch sử được hoàn tất.

Quy trình rút ra ứng viên hợp nhất trung bình

Sau đây là phân mô tả chi tiết phương pháp rút ra ứng viên hợp nhất trung bình, thủ tục xử lý của bước S403 trên Fig.21, là quy trình chung khối rút ra ứng viên hợp nhất trung bình 344 của khối rút ra chế độ hợp nhất thông thường 302 ở phía mã hóa và khối rút ra ứng viên hợp nhất trung bình 444 của khối rút ra chế độ hợp nhất thông thường 402 ở phía giải mã. Fig.38 là lưu đồ minh họa thủ tục của quy trình xử lý rút ra ứng viên hợp nhất trung bình.

Đầu tiên, quy trình khởi tạo được thực hiện (bước S1301 trên Fig.38). biến numOrigMergeCand được thiết lập là số các phần tử numCurrMergeCand được ghi trong danh sách ứng viên hợp nhất hiện tại.

Sau đó, việc quét được thực hiện tuần tự từ đầu danh sách ứng viên hợp nhất để xác định hai mảnh thông tin chuyển động. Chỉ số i thể hiện thông tin chuyển động thứ nhất được thiết lập sao cho chỉ số $i = 0$, và chỉ số j thể hiện thông tin chuyển động thứ hai được thiết lập sao cho chỉ số $j = 1$. (Các bước từ S1302 tới S1303 trên Fig.38). Khi số numCurrMergeCand các phần tử được ghi trong danh sách ứng viên hợp nhất hiện tại không nhỏ hơn hoặc bằng (số lượng ứng viên hợp nhất tối đa MaxNumMergeCand-1), các ứng viên hợp nhất đã được thêm vào tất cả các phần tử

trong danh sách ứng viên hợp nhất. Theo đó, thủ tục rút ra ứng viên hợp nhất dựa trên lịch sử sẽ được hoàn tất (bước S1304 trên Fig.38). Trong trường hợp số numCurrMergeCand của các phần tử được ghi trong danh sách ứng viên hợp nhất hiện tại nhỏ hơn hoặc bằng (số lượng ứng viên hợp nhất tối đa MaxNumMergeCand-1), quy trình bước S1305 trở đi sẽ được thực hiện.

Việc xác định được thực hiện để xác định xem có phải cả thông tin chuyển động thứ i mergeCandList[i] của danh sách ứng viên hợp nhất và thông tin chuyển động thứ j mergeCandList[j] của danh sách ứng viên hợp nhất đều không hợp lệ hay không (bước S1305 trên Fig.38). Trong trường hợp cả hai không hợp lệ, quy trình xử lý phần tử tiếp theo mà không rút ra ứng viên hợp nhất trung bình của mergeCandList[i] hoặc mergeCandList[j]. Trong trường hợp cả mergeCandList[i] và mergeCandList[j] đều không hợp lệ không được đáp ứng, quy trình sau đây được lặp lại với X được đặt là 0 và 1 (các bước từ S1306 tới S1314 trên Fig.38).

Việc xác định được thực hiện để xem dự đoán LX của mergeCandList[i] hợp lệ hay không (bước S1307 trên Fig.38). Trong trường hợp dự đoán LX của mergeCandList[i] hợp lệ, việc xác định được thực hiện để xem dự đoán LX của mergeCandList[j] hợp lệ (bước S1308 trên Fig.38). Trong trường hợp dự đoán LX của mergeCandList[j] hợp lệ, tức là, trong trường hợp cả dự đoán LX của mergeCandList[i] và dự đoán LX của mergeCandList[j] đều hợp lệ, vector chuyển động của dự đoán LX thu được bằng cách lấy trung bình vector chuyển động của dự đoán LX của mergeCandList[i] và vector chuyển động của dự đoán LX của mergeCandList[j] sẽ được rút ra, và ứng viên hợp nhất trung bình của dự đoán LX có chỉ số tham chiếu của dự đoán LX của mergeCandList[i] sẽ được rút ra, bởi vậy thiết lập được là dự đoán LX của averageCand, và dự đoán LX của averageCand hợp lệ

(bước S1309 trên Fig.38). Trong bước S1308 trên Fig.38, trong trường hợp dự đoán LX của mergeCandList[j] không hợp lệ, tức là, trong trường hợp dự đoán LX của mergeCandList[i] hợp lệ và dự đoán LX của mergeCandList[j] không hợp lệ, vector chuyển động của dự đoán LX của mergeCandList[i] và ứng viên hợp nhất trung bình của dự đoán LX có chỉ số tham chiếu sẽ được rút ra, do vậy thiết lập được là dự đoán LX của averageCand, và dự đoán LX của averageCand hợp lệ (bước S1310 trên Fig.38). Trong trường hợp dự đoán LX của mergeCandList[i] không hợp lệ trong bước S1307 trên Fig.38, việc xác định được thực hiện để xem dự đoán LX của mergeCandList[j] hợp lệ hay không (bước S1311 trên Fig.38). Trong trường hợp dự đoán LX của mergeCandList[j] hợp lệ, tức là, trong trường hợp dự đoán LX của mergeCandList[i] không hợp lệ và dự đoán LX của mergeCandList[j] hợp lệ, vector chuyển động của dự đoán LX của mergeCandList[j] và ứng viên hợp nhất trung bình của dự đoán LX có chỉ số tham chiếu sẽ được rút ra, do vậy thiết lập được là dự đoán LX của averageCand, và dự đoán LX của averageCand hợp lệ (bước S1312 trên Fig.38). Trong bước S1311 trên Fig.38, trong trường hợp dự đoán LX của mergeCandList [j] không hợp lệ, tức là, trong trường hợp dự đoán LX của mergeCandList[i] và dự đoán LX của mergeCandList[j] đều không hợp lệ, dự đoán LX của averageCand sẽ không hợp lệ (bước S1312 trên Fig.38).

Ứng viên hợp nhất trung bình averageCand của dự đoán L0, dự đoán L1 hoặc dự đoán kép được tạo ra như trên được thêm vào numCurrMergeCand-th mergeCandList[numCurrMergeCand] của danh sách ứng viên hợp nhất, và numCurrMergeCand tăng lên một (bước S1315 trên Fig.42). This completes quy trình rút ra ứng viên hợp nhất trung bình.

Ứng viên hợp nhất trung bình thu được bằng cách lấy trung bình trong

mỗi thành phần theo phương ngang của vector chuyển động và thành phần theo phương đứng của vector chuyển động.

Quy trình dự đoán bù chuyển động

Khối dự đoán bù chuyển động 306 thu được vị trí và kích thước của khối mà hiện đang trải qua quy trình dự đoán trong mã hóa. Ngoài ra, khối dự đoán bù chuyển động 306 thu được thông tin dự đoán liên ảnh từ khối xác định chế độ dự đoán liên ảnh 305. Chỉ số tham chiếu và vector chuyển động có được từ thông tin dự đoán liên ảnh thu được, và hình ảnh tham chiếu được chỉ ra bởi chỉ số tham chiếu trong bộ nhớ hình ảnh đã được giải mã 104 được dịch từ cùng vị trí với tín hiệu hình ảnh của khối đã được dự đoán đã trải qua quy trình dự đoán đi một lượng của vector chuyển động. Tín hiệu hình ảnh của vị trí sau khi dịch được thu lại và sau đó tạo ra được tín hiệu dự đoán.

Trong trường hợp dự đoán được thực hiện từ hình ảnh tham chiếu tín hiệu, chẳng hạn như khi chế độ dự đoán liên ảnh trong dự đoán liên ảnh là dự đoán L0 hoặc dự đoán L1, tín hiệu dự đoán thu được từ một hình ảnh tham chiếu được thiết lập là tín hiệu dự đoán bù chuyển động. Trong trường hợp dự đoán được thực hiện từ hai hình ảnh tham chiếu, chẳng hạn như khi chế độ dự đoán liên ảnh là dự đoán kép, trung bình trọng số của các tín hiệu dự đoán thu được từ hai hình ảnh tham chiếu được thiết lập là tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động thu được được đưa tới khối xác định phương pháp dự đoán 105. Ở đây, tỷ lệ trung bình trọng số trong dự đoán kép được thiết lập là 1: 1. Ngoài ra, trung bình trọng số có thể là một tỷ lệ khác. Ví dụ, tỷ lệ trọng số có thể được thiết lập sao cho khoảng cách hình ảnh giữa hình ảnh đích dự đoán và hình ảnh tham chiếu càng thấp, thì tỷ lệ trọng số càng cao. Việc tính toán tỷ lệ trọng số cũng có thể được thực hiện bằng cách sử dụng bảng tương

đương giữa các khoảng cách hình ảnh và các tỷ lệ trọng số.

Khối dự đoán bù chuyển động 406 có chức năng tương tự với khối dự đoán bù chuyển động 306 ở phía mã hóa. Khối dự đoán bù chuyển động 406 thu được thông tin dự đoán liên ảnh từ khối rút ra chế độ dự đoán vector chuyển động thông thường 401, khối rút ra chế độ hợp nhất thông thường 402, khối rút ra chế độ dự đoán vector chuyển động khối con 403, và khối rút ra chế độ hợp nhất khối con 404, thông qua khối chuyển mạch 408. Khối dự đoán bù chuyển động 406 đưa tín hiệu dự đoán bù chuyển động thu được tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207.

Chế độ dự đoán liên ảnh

Quy trình thực hiện dự đoán từ hình ảnh tham chiếu đơn được gọi là dự đoán đơn. Dự đoán đơn thực hiện dự đoán của dự đoán L0 hoặc dự đoán L1 bằng cách sử dụng một trong hai hình ảnh tham chiếu được ghi trong danh sách tham chiếu L0 hoặc L1.

Fig.32 minh họa trường hợp dự đoán đơn trong đó hình ảnh tham chiếu (RefL0Pic) của L0 tại thời điểm trước hình ảnh đích (CurPic). Fig.33 minh họa trường hợp dự đoán đơn trong đó hình ảnh tham chiếu của dự đoán L0 ở tại thời điểm sau hình ảnh đích. Tương tự, dự đoán đơn có thể được thực hiện bằng cách thay thế hình ảnh tham chiếu dự đoán L0 trên Fig.32 và Fig.33 bằng hình ảnh tham chiếu dự đoán L1 (RefL1Pic).

Quy trình thực hiện dự đoán từ hai hình ảnh tham chiếu được gọi là dự đoán kép. Dự đoán kép thực hiện dự đoán, được thể hiện ở dạng kép, bằng cách sử dụng cả dự đoán L0 và dự đoán L1. Fig.34 minh họa trường hợp dự đoán kép trong đó hình ảnh tham chiếu dự đoán L0 tại thời điểm trước hình ảnh đích và hình ảnh tham chiếu dự đoán L1 ở tại thời điểm sau hình ảnh đích. Fig.35 minh họa trường hợp dự đoán

kép trong đó hình ảnh tham chiếu cho dự đoán L0 và hình ảnh tham chiếu cho dự đoán L1 tại thời điểm trước hình ảnh đích. Fig.36 minh họa trường hợp dự đoán kép trong đó hình ảnh tham chiếu cho dự đoán L0 và hình ảnh tham chiếu cho dự đoán L1 tại thời điểm sau hình ảnh đích.

Theo cách này, có thể sử dụng dự đoán mà không cần hạn chế mối quan hệ giữa kiểu dự đoán của L0/L1 và thời gian L từ 0 trở về trước và L1 trở về sau. Ngoài ra, dự đoán kép có thể thực hiện mỗi dự đoán L0 và dự đoán L1 bằng cách sử dụng cùng một hình ảnh tham chiếu. Việc xác định xem thực hiện dự đoán bù chuyển động trong dự đoán đơn hay dự đoán kép được thực hiện dựa trên thông tin (ví dụ, cờ) thể hiện sử dụng dự đoán L0 và sử dụng dự đoán L1, ví dụ.

Chỉ số tham chiếu

Theo phương án của sáng chế, có thể lựa chọn hình ảnh tham chiếu tối ưu từ các hình ảnh tham chiếu trong dự đoán bù chuyển động để cải thiện được độ chính xác của dự đoán bù chuyển động. Bởi vậy, hình ảnh tham chiếu được sử dụng trong dự đoán bù chuyển động sẽ được sử dụng làm chỉ số tham chiếu, và chỉ số tham chiếu được mã hóa trong luồng bit cùng với sai phân vector chuyển động.

Quy trình bù chuyển động dựa trên chế độ dự đoán vector chuyển động thông thường

Như được minh họa trên khối dự đoán liên ảnh 102 ở phía mã hóa trên Fig.16, trong trường hợp thông tin dự đoán liên ảnh bởi khối rút ra chế độ dự đoán vector chuyển động thông thường 301 đã được lựa chọn trên khối xác định chế độ dự đoán liên ảnh 305, khối dự đoán bù chuyển động 306 thu được thông tin dự đoán liên ảnh này từ khối xác định chế độ dự đoán liên ảnh 305, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích và nhờ đó tạo ra tín hiệu dự

đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xác định phương pháp dự đoán 105.

Tương tự, như được minh họa trên khối dự đoán liên ảnh 203 ở phía giải mã trên Fig.22, trong trường hợp khối chuyển mạch 408 được kết nối với khối rút ra chế độ dự đoán vector chuyển động thông thường 401 trong suốt quy trình giải mã, khối dự đoán bù chuyển động 406 thu được thông tin dự đoán liên ảnh bởi khối rút ra chế độ dự đoán vector chuyển động thông thường 401, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích và nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207.

Quy trình xử lý bù chuyển động dựa trên chế độ hợp nhất thông thường

Như được minh họa trên khối dự đoán liên ảnh 102 ở phía mã hóa trên Fig.16, trong trường hợp thông tin dự đoán liên ảnh bởi khối rút ra chế độ hợp nhất thông thường 302 đã được lựa chọn trên khối xác định chế độ dự đoán liên ảnh 305, khối dự đoán bù chuyển động 306 thu được Thông tin dự đoán liên ảnh này từ khối xác định chế độ dự đoán liên ảnh 305, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xác định phương pháp dự đoán 105.

Tương tự, như được minh họa trên khối dự đoán liên ảnh 203 ở phía giải mã trên Fig.22, trong trường hợp khối chuyển mạch 408 được kết nối với khối rút ra chế độ hợp nhất thông thường 402 trong suốt quy trình giải mã, khối dự đoán bù chuyển động 406 thu được thông tin dự đoán liên ảnh bởi khối rút ra chế độ hợp nhất thông thường 402, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển

động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207.

Quy trình xử lý bù chuyển động dựa trên chế độ dự đoán vector chuyển động khối con

Như được minh họa trên khối dự đoán liên ảnh 102 ở phía mã hóa trên Fig.16, trong trường hợp thông tin dự đoán liên ảnh bởi khối rút ra chế độ dự đoán vector chuyển động khối con 303 đã được lựa chọn trên khối xác định chế độ dự đoán liên ảnh 305, khối dự đoán bù chuyển động 306 thu được thông tin dự đoán liên ảnh này từ khối xác định chế độ dự đoán liên ảnh 305, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xác định phương pháp dự đoán 105.

Tương tự, như được minh họa trên khối dự đoán liên ảnh 203 ở phía giải mã trên Fig.22, trong trường hợp khối chuyển mạch 408 được kết nối với khối rút ra chế độ dự đoán vector chuyển động khối con 403 trong suốt quy trình giải mã, khối dự đoán bù chuyển động 406 thu được thông tin dự đoán liên ảnh bởi khối rút ra chế độ dự đoán vector chuyển động khối con 403, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207.

Quy trình xử lý bù chuyển động dựa trên chế độ hợp nhất khối con

Như được minh họa trên khối dự đoán liên ảnh 102 ở phía mã hóa trên Fig.16, trong trường hợp thông tin dự đoán liên ảnh bởi khối rút ra chế độ hợp nhất khối con

304 đã được lựa chọn trên khối xác định chế độ dự đoán liên ảnh 305, khối dự đoán bù chuyển động 306 thu được thông tin dự đoán liên ảnh này từ khối xác định chế độ dự đoán liên ảnh 305, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xác định phương pháp dự đoán 105.

Tương tự, như được minh họa trên khối dự đoán liên ảnh 203 ở phía giải mã trên Fig.22, trong trường hợp khối chuyển mạch 408 được kết nối với khối rút ra chế độ hợp nhất khối con 404 trong suốt quy trình giải mã, khối dự đoán bù chuyển động 406 thu được thông tin dự đoán liên ảnh bởi khối rút ra chế độ hợp nhất khối con 404, và rút ra chế độ dự đoán liên ảnh, chỉ số tham chiếu, và vector chuyển động của khối đích, nhờ đó tạo ra tín hiệu dự đoán bù chuyển động. Tín hiệu dự đoán bù chuyển động tạo ra được đưa tới khối xếp chồng tín hiệu hình ảnh đã được giải mã 207.

Quy trình xử lý bù chuyển động dựa trên dự đoán biến đổi tinh

Trong chế độ dự đoán vector chuyển động thông thường và chế độ hợp nhất thông thường, bù chuyển động bằng cách sử dụng mô hình afin có thể được sử dụng dựa trên các cờ sau. Các cờ sau đây phản ánh các điều kiện dự đoán liên ảnh được định trước bởi khối xác định chế độ dự đoán liên ảnh 305 trong quy trình mã hóa, và được mã hóa trong luồng bit. Trong quy trình giải mã, việc có thực hiện bù chuyển động bằng cách sử dụng mô hình afin dựa trên các cờ sau trong luồng bit hay không được chỉ ra.

`sps_affine_enabled_flag` thể hiện bù chuyển động bằng cách sử dụng mô hình afin có thể được sử dụng trong dự đoán liên ảnh hay không. Khi `sps_affine_enabled_flag` là 0, quy trình bị ngăn thực hiện bù chuyển động bởi mô hình afin trong các đơn vị của chuỗi. Ngoài ra, `inter_affine_flag` và `cu_affine_type_flag`

không được truyền trong khối mã hóa (CU) cú pháp của việc mã hóa chuỗi video. Khi `sps_affine_enabled_flag` là 1, bù chuyển động bởi mô hình afin có thể được sử dụng trong mã hóa chuỗi video.

`sps_affine_type_flag` thể hiện bù chuyển động bằng cách sử dụng 6-tham số mô hình afin có thể được sử dụng trong dự đoán liên ảnh hay không. Khi `sps_affine_type_flag` là 0, quy trình bị ngăn thực hiện bù chuyển động bằng cách sử dụng 6 tham số mô hình afin. Ngoài ra, `cu_affine_type_flag` không được truyền đi trong CU cú pháp của mã hóa chuỗi video. Khi `sps_affine_type_flag` là 1, bù chuyển động dựa trên 6 tham số mô hình afin có thể được sử dụng trong mã hóa chuỗi video. Trong trường hợp `sps_affine_type_flag` không tồn tại, nó sẽ là 0.

Trong trường hợp giải mã P hoặc B lát, khi `inter_affine_flag` là 1 CU hiện tại, bù chuyển động bằng cách sử dụng mô hình afin được sử dụng để tạo ra tín hiệu dự đoán bù chuyển động của CU hiện tại. Khi `inter_affine_flag` là 0, mô hình afin không được sử dụng cho CU hiện tại. Trong trường hợp `inter_affine_flag` không tồn tại, nó sẽ là 0.

Trong trường hợp giải mã lát P hoặc B, khi `cu_affine_type_flag` là 1 trong CU hiện tại, bù chuyển động bằng cách sử dụng 6 tham số mô hình afin được sử dụng để tạo ra tín hiệu dự đoán bù chuyển động của CU hiện tại. Khi `cu_affine_type_flag` là 0, bù chuyển động bằng cách sử dụng bốn tham số mô hình afin được sử dụng để tạo ra tín hiệu dự đoán bù chuyển động của CU đang được xử lý.

Chỉ số tham chiếu và vectơ chuyển động được suy ra trong các đơn vị của các khối con trong Bù chuyển động dựa trên mô hình afin. Theo đó, tín hiệu dự đoán bù chuyển động được tạo ra bằng cách sử dụng chỉ số tham chiếu và vectơ chuyển động để được xử lý trong các đơn vị của khối con.

Bốn tham số mô hình afin là chế độ trong đó vector chuyển động của khối con được lấy từ bốn tham số của thành phần theo phương ngang và thành phần theo phương đứng của mỗi vector chuyển động của hai điểm điều khiển, và bù chuyển động được thực hiện trong các đơn vị của các khối con.

Bản sao khối nội ảnh (IBC)

Vùng tham chiếu hợp lệ của bản sao khối nội ảnh sẽ được mô tả tham chiếu tới Fig.39. Fig.39A là một trường hợp mẫu trong đó một vùng tham chiếu hợp lệ được xác định bằng cách giả định đơn vị khối cây mã hóa làm khối chuẩn sao chép khối nội ảnh. Các khối 500, 501, 502, 503 và 504 trên Fig.39A là các khối cây mã hóa, và khối 504 trong số này là khối cây mã hóa đích. Khối 505 là một khối mã hóa đích. Các khối cây mã hóa được xử lý theo thứ tự 500, 501, 502, 503 và 504. Trong trường hợp này, ba khối cây mã hóa 501, 502 và 503 được xử lý ngay trước khi khối cây mã hóa 504 bao gồm khối mã hóa đích 505 được đặt làm vùng tham chiếu hợp lệ của khối mã hóa đích 505. Khối cây mã hóa được xử lý trước khối cây mã hóa 501 và các vùng bao gồm trong khối cây mã hóa 504 bao gồm khối mã hóa đích 505 đều được đặt thành các vùng tham chiếu không hợp lệ bất kể việc xử lý có được hoàn thành trước khối mã hóa đích 505 hay không.

Fig.39B minh họa một trường hợp mẫu trong đó một vùng tham chiếu hợp lệ được xác định bằng cách giả định một đơn vị thu được bằng cách tách đơn vị khối cây mã hóa làm khối chuẩn sao chép khối nội ảnh. Trên Fig.39B, các khối 515 và 516 là các khối cây mã hóa và khối 516 là khối cây mã hóa đích. Khối cây mã hóa 515 được chia thành 506, 507, 508 và 509, và khối cây mã hóa 516 được chia thành các khối 510, 511, 512 và 513. Số tham chiếu 514 là khối mã hóa đích. Các khối chuẩn sao chép khối nội ảnh được xử lý theo thứ tự 506, 507, 508, 509, 510, 511, 512 và 513. Trong

trường hợp này, ba khối chuẩn sao chép khối nội ảnh 508, 509, 510 được xử lý ngay trước khối chuẩn sao chép khối nội ảnh 511 bao gồm khối mã hóa đích 514 được đặt làm vùng tham chiếu hợp lệ của khối mã hóa đích 514. Khối cây mã hóa được xử lý trước khối chuẩn sao chép khối nội ảnh 508 và tất cả các vùng bao gồm trong khối chuẩn sao chép khối nội ảnh 511 bao gồm khối mã hóa đích 514 đều được đặt thành các vùng tham chiếu không hợp lệ bất kể việc xử lý có được hoàn thành trước khối mã hóa đích 514 hay không.

Dự đoán bản sao khối nội ảnh: phía mã hóa

Quy trình xử lý bản sao nội khối dự đoán ở phía mã hóa sẽ được mô tả với tham chiếu đến Fig.44.

Đầu tiên, khối phát hiện vector khối 375 phát hiện vector khối mvL (bước S4500 trên Fig.44). Sau đó, khối rút ra ứng viên vector khối không gian IBC 371, khối dự đoán ứng viên vector khối dựa trên lịch sử IBC 372, khối bổ sung ứng viên dự đoán vector khối IBC 373, khối chọn ứng viên dự đoán vector khối IBC 376, khối trừ vector khối 378 tính toán chênh lệch vector khối của vector khối được sử dụng trong chế độ dự đoán vector khối (các bước S4501 đến S4503 trên Fig.44).

Các ứng viên dự đoán vector khối được tính toán để xây dựng danh sách ứng viên vector khối.mvpList (bước S4501 trên Fig.44). Trong khối dự đoán bản sao khối nội ảnh 352, khối rút ra ứng viên vector khối không gian IBC 371, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 372 và ứng viên dự đoán vector khối IBC bổ sung 373 lấy đa số các ứng viên dự đoán vector khối và xây dựng một danh sách ứng viên dự đoán khối.mvpList. Quy trình xử lý chi tiết của bước S4501 trên Fig.44 sẽ được mô tả dưới đây bằng cách sử dụng lưu đồ trên Fig.47.

Sau đó, khối chọn ứng viên dự đoán vector khối IBC 376 chọn dự đoán vector

khối mvpL từ danh sách ứng viên dự đoán vector khối mvpListL (bước S4502 trên Fig.44). Sự khác biệt vector khối, đó là sự khác biệt giữa vector khối mvL và mỗi ứng viên dự đoán vector khối mvpListL [i] được lưu trữ trong danh sách ứng viên dự đoán vector khối mvpListL, được tính toán. Một số lượng mã tại thời điểm mã hóa các chênh lệch vector khối này được tính cho từng yếu tố của danh sách ứng viên dự đoán vector khối mvpListL. Sau đó, ứng viên dự đoán vector khối mvpListL [i] giảm thiểu số lượng mã cho mỗi ứng viên dự đoán vector khối trong số các yếu tố riêng lẻ được ghi trong danh sách ứng viên dự đoán vector khối MVpListL được chọn làm mvpL dự đoán vector khối và chỉ số i của nó thu được. Trong trường hợp có sự đa dạng của các ứng viên dự đoán vector khối có số lượng mã được tạo tối thiểu trong danh sách ứng viên dự đoán vector khối mvpListL, ứng viên dự đoán vector khối mvpListL [i] có chỉ số i trong danh sách ứng viên dự đoán vector khối MVpListL được đại diện bởi một số nhỏ được chọn làm mvpL dự đoán vector khối tối ưu, và thu được chỉ số i của nó.

Sau đó, khối trừ vector khối 378 trừ đi mvpL vector khối đã chọn khối vector khối mvL, và do đó tính toán chênh lệch vector khối mvdL như sau:

$$\text{mvdL} = \text{mvL} - \text{mvpL} \text{ (bước S4503 trên Fig.44).}$$

Dự đoán bản sao khối nội ảnh: phía giải mã

Tiếp theo, quy trình xử lý chế độ dự đoán vector khối bình thường ở phía giải mã sẽ được mô tả liên quan đến Fig.45. Ở phía giải mã, khối dự đoán ứng viên vector khối không gian IBC 471, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 472 và khối bổ sung dự đoán vector khối IBC 473 tính toán vector khối được sử dụng trong chế độ dự đoán vector khối (bước S4600 đến S4602 trên Fig.45). Cụ thể, danh sách ứng viên dự đoán vector khối mvpListL được tính toán, dự đoán vector khối mvpL được chọn, và sau đó vector khối mvL được tính toán.

Các ứng viên dự đoán vector khối được tính toán để xây dựng danh sách ứng viên dự đoán vector khối MVPLISTL (bước S4601 trên Fig.45). Trong khối dự đoán bản sao khối nội ảnh 362, khối rút ra ứng viên vector khối không gian IBC 471, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 472 và khối bổ sung vector khối IBC 473 tính toán đa số các ứng viên dự đoán vector khối và xây dựng danh sách ứng viên dự đoán vector khối mvplListL. Phần mô tả quy trình xử lý chi tiết của bước S4601 trên Fig.45 được bỏ qua. Sau đó, khối chọn ứng viên dự đoán vector khối IBC 476 trích xuất ứng viên dự đoán vector khối mvplListL [mvplIdxL] tương ứng với chỉ số dự đoán vector khối mvplIdxL được giải mã và cung cấp bởi bộ giải mã chuỗi bit 201, từ danh sách ứng viên dự đoán vector khối mvplListL, dưới dạng dự đoán vector khối được chọn mvpl (bước S4601 trên Fig.45). Sau đó, khối cộng vector khối 478 thêm sự khác biệt vector khối mvdL và bộ dự đoán vector khối mvpl được giải mã và cung cấp bởi bộ giải mã chuỗi bit 201 và tính toán (bước S4602 trên Fig.45) vector khối mvL như sau:

$$mvL = mvpl + mvdL$$

Chế độ dự đoán vector khối: phương pháp dự đoán vector khối

Fig.47 là lưu đồ minh họa quy trình xử lý rút ra chế độ sao chép nội khối dự đoán có một hàm chung với khối dự đoán bản sao khối nội ảnh 352 của thiết bị mã hóa hình ảnh chuyển động và khối dự đoán bản sao khối nội ảnh 362 của thiết bị giải mã hình ảnh chuyển động theo phương án của sáng chế.

Mỗi khối dự đoán bản sao khối nội ảnh 352 và khối dự đoán bản sao khối nội ảnh 362 bao gồm một danh sách ứng viên dự đoán vector khối mvplListL. Danh sách ứng viên dự đoán vector khối mvplListL có cấu trúc danh sách và bao gồm một vùng lưu trữ lưu trữ, dưới dạng các yếu tố, chỉ số dự đoán vector khối cho biết vị trí trong danh sách ứng viên dự đoán vector khối và ứng viên dự đoán vector khối tương ứng với

chỉ số. Số chỉ số dự đoán vector khối bắt đầu từ 0 và các ứng viên dự đoán vector khối sẽ được lưu trữ trong vùng lưu trữ của danh sách ứng viên dự đoán vector khối mvplistL. Trong phương án này, người ta cho rằng danh sách ứng viên dự đoán vector khối mvplistL có thể ghi nhận ba ứng viên dự đoán vector khối. Hơn nữa, biến numCurrMvpIbcCand cho biết số lượng ứng viên dự đoán vector khối đã ghi nhận trong danh sách ứng viên dự đoán vector khối mvplistL được đặt là 0.

Các khối rút ra ứng viên khối không gian IBC 371 và 471 lấy được một ứng viên dự đoán vector khối từ khối lân cận ở phía bên trái (bước S4801 trên Fig.47). Trong quy trình này, cờ có availableFlagLA cho biết liệu ứng viên dự đoán vector khối của khối lân cận (A0 hoặc A1) ở phía bên trái có thể sử dụng được và vector khối mvLA, được rút ra và mvLA được rút ra sẽ được thêm vào danh sách ứng viên dự đoán vector khối mvplistL. Sau đó, các khối rút ra ứng viên khối không gian IBC 371 và 471 lấy được một ứng viên dự đoán vector khối từ khối lân cận (B0, B1 hoặc B2) ở phía trên (bước S4802 trên Fig.47). Quy trình này lấy cờ availableFlagLB cho biết liệu ứng viên dự đoán vector chuyển động của khối lân cận ở phía trên có thể sử dụng được hay không và vector khối mvLB. Khi mvLA và mvLB không bằng nhau, mvLB vector khối sẽ được thêm vào danh sách ứng viên dự đoán vector khối MVplistL. Các quy trình trong các bước S4801 và S4802 trên Fig.47 được cung cấp như một quy trình chung ngoại trừ các vị trí và số lượng các khối lân cận tham chiếu là khác nhau. Các quy trình này lấy cờ availableFlagLN cho biết liệu một ứng viên dự đoán vector khối của một khối mã hóa có thể sử dụng được hay không và vector chuyển động mvLN, (N chỉ ra A hoặc B, tương tự áp dụng sau đây).

Sau đó, các khối rút ra ứng viên khối dựa trên lịch sử IBC 372 và 472 thêm các ứng viên vector khối dựa trên lịch sử đã ghi nhận trong danh sách ứng viên vector

khối dựa trên lịch sử HmvpIbcCandList vào danh sách ứng viên dự đoán vector khối.mvpListL. (Bước S4803 trên Fig.47). Đối với các chi tiết của quy trình xử lý ghi nhận của bước S4803, nó là đủ để thực hiện các hoạt động tương tự như thao tác minh họa trong lưu đồ trên Fig.29, nghĩa là, hoạt động trong trường hợp vector chuyển động được thay thế bằng vector khối, danh sách chỉ số tham chiếu được thay thế bằng L0, danh sách ứng viên dự đoán vector chuyển động dựa trên lịch sử HmvpCandList được thay thế bằng danh sách ứng viên vector khối dựa trên lịch sử HmvpIbcCandList, và do đó, phần mô tả về các quy trình này sẽ được bỏ qua.

Sau đó, khối bổ sung dự đoán vector khối IBC 373 và 473 thêm vector khối có giá trị được xác định trước như (0, 0) cho đến khi danh sách ứng viên dự đoán vector khối.mvpListL thỏa mãn (S4804 trên Fig.47).

Khối rút ra chế độ sao chép khối nội ảnh hợp nhất

Khối dự đoán bản sao khối nội ảnh 352 trên Fig.42 bao gồm khối rút ra ứng viên khối không gian IBC 371, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 372, khối bổ sung vector khối IBC 373, khối hiệu chỉnh vị trí tham chiếu 380, khối hiệu chỉnh ranh giới vùng tham chiếu 381, khối chọn ứng viên hợp nhất IBC 374 và khối xác định chế độ dự đoán IBC 377.

Khối dự đoán bản sao khối nội ảnh 362 trên Fig.43 bao gồm khối rút ra ứng viên khối không gian IBC 471, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 472, khối bổ sung vector khối IBC 473, khối chọn ứng viên hợp nhất IBC 474, khối hiệu chỉnh vị trí tham chiếu 480, khối hiệu chỉnh ranh giới vùng tham chiếu 481 và khối sao chép khối 477.

Fig.46 là lưu đồ minh họa quy trình rút ra chế độ sao chép khối nội ảnh hợp nhất có một hàm chung với khối dự đoán bản sao khối nội ảnh 352 của thiết bị mã hóa

hình ảnh chuyển động và khối dự đoán bản sao khối nội ảnh 362 của thiết bị giải mã hình ảnh chuyển động theo phương án của sáng chế.

Khối dự đoán bản sao khối nội ảnh 352 và khối dự đoán bản sao khối nội ảnh 362 bao gồm danh sách ứng viên sao chép khối nội ảnh hợp nhất hợp mergeIbcCandList. Danh sách ứng viên sao chép nội ảnh hợp nhất mergeIbcCandList có cấu trúc danh sách và bao gồm: vùng lưu trữ lưu trữ, dưới dạng các yếu tố, chỉ số hợp nhất cho biết vị trí bên trong ứng viên bản sao khối nội ảnh hợp nhất; và ứng viên bản sao khối nội ảnh hợp nhất tương ứng với chỉ số. Số lượng chỉ số hợp nhất bắt đầu từ 0 và ứng viên bản sao khối nội ảnh hợp nhất được lưu trữ trong vùng lưu trữ của danh sách ứng viên bản sao khối nội ảnh hợp nhất mergeIbcCandList. Trong các quy trình tiếp theo, ứng viên hợp nhất của chỉ số hợp nhất i đã ghi nhận trong danh sách ứng viên bản sao khối nội ảnh mergeIbcCandList sẽ được đại diện bởi mergeIbcCandList [i]. Trong phương án này, người ta cho rằng danh sách ứng viên hợp nhất hợp nhấtCandList có thể ghi nhận ít nhất ba ứng viên bản sao khối trong hợp nhất. Hơn nữa, biến numCurrMergeIbcCand cho biết số lượng các ứng viên bản sao khối nội ảnh hợp nhất đã ghi nhận trong danh sách ứng viên sao chép khối nội ảnh mergeIbcCandList được đặt thành 0.

Khối rút ra ứng viên khối không gian IBC 371 và khối rút ra ứng viên khối không gian IBC 471 lấy các ứng viên hợp nhất không gian A và B của các khối lân cận ở bên trái và các mặt trên của khối đích từ thông tin mã hóa được lưu trữ trong bộ nhớ lưu trữ thông tin mã hóa 111 của thiết bị mã hóa hình ảnh chuyển động hoặc bộ nhớ lưu trữ thông tin mã hóa 205 của thiết bị giải mã hình ảnh đang di chuyển, và sau đó thêm các ứng viên hợp nhất không gian được rút ra vào danh sách ứng viên sao chép nội ảnh hợp nhất Danh sách ứng viên bản sao nội ảnh mergeIbcCandList (bước S4701

trên Fig.46). Ở đây, N chỉ ra một trong những ứng viên hợp nhất không gian A và B sẽ được xác định. Cờ availableFlagN và vector khối mvL cho biết liệu thông tin dự đoán bản sao khối nội ảnh của khối N có thể được sử dụng làm cho vector khối không gian hợp nhất ứng viên N được rút ra hay không. Tuy nhiên, trong phương án này, ứng viên hợp nhất vector khối được rút ra mà không tham chiếu đến các khối mã hóa khác có trong khối bao gồm khối mã hóa đích. Theo đó, ứng viên hợp nhất vector khối không gian bao gồm trong khối bao gồm khối mã hóa đích sẽ không được rút ra.

Sau đó, khối rút ra ứng viên vector khối dựa trên lịch sử IBC 372 và khối rút ra ứng viên vector khối dựa trên lịch sử IBC 472 thêm các ứng viên dự đoán vector khối dựa trên lịch sử đã ghi nhận trong danh sách ứng viên dự đoán vector khối dựa trên lịch sử HmvpIbcCandList, vào danh sách ứng viên sao chép khối nội ảnh hợp nhất mergeIbcCandList (bước S4702 trên Fig.46). Trong ví dụ này, trong trường hợp vector khối đã được thêm vào mergeIbcCandList và vector khối của ứng viên dự đoán vector khối dựa trên lịch sử có cùng giá trị, việc bổ sung vào mergeIbcCandList sẽ không được thực hiện.

Sau đó, trong trường hợp số ứng viên hợp nhất numCurrMergeIbcCand ghi nhận trong danh sách ứng viên sao chép khối nội ảnh mergeIbcCandList nhỏ hơn số ứng viên hợp nhất khối nội ảnh MaxNumMergeIbcCand, khối bổ sung dự đoán vector khối IBC 373 và khối bổ sung dự đoán vector khối IBC 473 lấy thêm ứng viên hợp nhất khối nội ảnh sao cho số lượng ứng viên hợp nhất numrMergeIbcCand ghi nhận trong danh sách ứng viên sao chép khối nội ảnh hợp nhất mergeIbcCandList nằm trong giới hạn trên của số lượng ứng viên hợp nhất MaxNumMergeIbcCand và thêm ứng viên được rút ra vào danh sách ứng viên sao chép khối nội ảnh mergeIbcCandList (bước S4703 trên Fig.46). Vector khối có giá trị (0,0) sẽ được thêm vào danh sách ứng

viên sao chép nội ảnh hợp nhất MergeIbcCandList trong giới hạn trên của số lượng ứng viên hợp nhất MaxNumMergeIbcCand.

Sau đó, khối chọn ứng viên hợp nhất IBC 374 và khối chọn ứng viên hợp nhất IBC 474 chọn một trong số các ứng viên hợp nhất khối nội ảnh đã ghi nhận trong danh sách ứng viên sao chép khối nội ảnh mergeIbcCandList (bước S4704 trên Fig.46). Khối chọn ứng viên hợp nhất IBC 374 có được một hình ảnh được giải mã ở vị trí tham chiếu từ bộ nhớ ảnh được giải mã 104, tính toán số lượng mã và số lượng biến dạng, do đó chọn một ứng viên hợp nhất, và sau đó cung cấp một chỉ số hợp nhất cho biết ứng viên hợp nhất khối nội ảnh đã chọn vào khối xác định chế độ dự đoán IBC 377. Khối xác định chế độ dự đoán IBC 377 tính toán số lượng mã và số lượng biến dạng, do đó chọn chế độ hợp nhất, sau đó cung cấp kết quả cho khối xác định phương pháp dự đoán 105. Ngược lại, khối chọn ứng viên hợp nhất IBC 474 ở phía giải mã chọn một ứng viên hợp nhất khối nội ảnh trên cơ sở chỉ số hợp nhất được giải mã, sau đó cung cấp ứng viên hợp nhất khối nội ảnh đã chọn cho khối hiệu chỉnh vị trí tham chiếu 480.

Sau đó, khối hiệu chỉnh vị trí tham chiếu 380 và khối hiệu chỉnh vị trí tham chiếu 480 thực hiện quy trình hiệu chỉnh vị trí tham chiếu trên ứng viên hợp nhất khối nội ảnh (bước S4705 trên Fig.46). Chi tiết về các quy trình của khối hiệu chỉnh vị trí tham chiếu 380 và khối hiệu chỉnh vị trí tham chiếu 480 sẽ được mô tả dưới đây.

Sau đó, khối hiệu chỉnh ranh giới vùng tham chiếu 381 và khối hiệu chỉnh ranh giới vùng tham chiếu 481 thực hiện quy trình hiệu chỉnh ranh giới vùng tham chiếu đối với ứng viên hợp nhất khối nội ảnh (bước S4706 trên Fig.46). Chi tiết về các quy trình được thực hiện bởi khối hiệu chỉnh vị trí tham chiếu 381 và khối hiệu chỉnh vị trí tham chiếu 481 sẽ được mô tả dưới đây.

Khối sao chép khối 477 có được một hình ảnh được giải mã ở vị trí tham chiếu từ bộ nhớ hình ảnh được giải mã 208 và cung cấp hình ảnh thu được cho khối thêm vào tín hiệu hình ảnh được giải mã 207. Ở đây, khối sao chép khối 477 sao chép một thành phần độ sáng và một thành phần màu.

Vector khối mvL trên đại diện cho vector khối độ sáng. Khi định dạng màu là 4:2:0, vector khối màu mvC thu được như sau:

$$mvC = ((mvL \gg (3 + 2)) * 32$$

Dựa trên phương trình trên, mỗi thành phần x và y của mvC sẽ được xử lý.

Khối hiệu chỉnh vị trí tham chiếu

Fig.48 là lưu đồ minh họa một quy trình được thực hiện bởi khối hiệu chỉnh vị trí tham chiếu 380 và khối hiệu chỉnh vị trí tham chiếu th 480. Ở đây giả định rằng đơn vị của khối chuẩn sao chép khối nội ảnh là khối cây mã hóa (CTU) và kích thước của nó không phải là 128×128 mẫu.

Đầu tiên, các vị trí trên bên trái và dưới bên phải của một khối tham chiếu được tính toán (S6001). Khối tham chiếu đề cập đến một khối mà khối mã hóa đích đề cập đến việc sử dụng vector khối. Khi xác định được trên khối tham chiếu rằng vị trí trên bên trái là (xRefTL, yRefTL) và vị trí dưới bên phải là (xRefBR, yRefBR), những điều sau đây sẽ thu được:

$$(xRefTL, yRefTL) = (xCb + (mvL[0] \gg 4), yCb + (mvL[1] \gg 4))$$

$$(xRefBR, yRefBR) = (xRefTL + cbWidth - 1, yRefTL + cbHeight - 1)$$

Ở đây, vị trí của khối mã hóa đích là (xCb, yCb), vector khối là (mvL[0], mvL[1]), chiều rộng của khối mã hóa đích là cbWidth và chiều cao là cbHeight.

Tiếp theo, người ta xác định xem kích thước CTU là 128×128 mẫu (S6002). Ở đây, vì kích thước không phải là 128×128 mẫu (S6002: SAI), các vị trí trên bên trái

và dưới bên phải của vùng có thể tham chiếu được tính toán (S6003). Khi vị trí trên bên trái là (xAvlTL, yAvlTL) và vị trí dưới bên phải là (xAvlBR, yAvlBR) trên vùng có thể tham chiếu, các vị trí riêng lẻ sẽ thu được như sau:

$$\begin{aligned}
 NL &= \text{Min}(1, 7 - \text{CtbLog2SizeY}) - (1 \ll (((7 - \text{CtbLog2SizeY}) \ll 1)) \\
 (xAvlTL, yAvlTL) &= (((xCb \gg \text{CtbLog2SizeY}) + NL) \ll \text{CtbLog2SizeY}, \\
 (yCb \gg \text{CtbLog2SizeY}) &\ll \text{CtbLog2SizeY}) \\
 (xAvlBR, yAvlBR) &= (((xCb \gg \text{CtbLog2SizeY}) \ll \text{CtbLog2SizeY}) - 1, \\
 (((((\gg \text{CtbLog2SizeY}) + 1) &\ll \text{CtbLog2SizeY}) - 1)
 \end{aligned}$$

Ở đây, kích thước của CTU là CtbLog2SizeY.

Tiếp theo, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng x có nhỏ hơn phía trên bên trái của vùng tham chiếu (S6004) hay không. Khi xác định là sai (S6004: SAI), quy trình này tiến hành quy trình tiếp theo (S6006). Ngược lại, khi quyết định là đúng (S6004: ĐÚNG), vị trí tham chiếu trong hướng x sẽ được hiệu chỉnh phù hợp với phía trên bên trái của vùng có thể tham chiếu (S6005).

Fig.49 là sơ đồ minh họa trạng thái hiệu chỉnh vị trí tham chiếu. Số tham chiếu 6001 biểu thị khối cây mã hóa đích, 6002 biểu thị khối mã hóa đích và 6003 biểu thị một vùng có thể tham chiếu. Bây giờ, giả sử rằng một khối tham chiếu r2 nằm ở 6011, vị trí tham chiếu trong hướng x nhỏ hơn phía trên bên trái của vùng có thể tham chiếu (S6004: ĐÚNG). Do đó, vị trí tham chiếu được hiệu chỉnh vào vị trí của 6012 giả định $xRefTL = xAvlTL$ (S6005). Ở đây, vì $xRefBR = xRefTL + cbWidth - 1$ giữ như trong S6001, $xRefBR$ cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của $xRefTL$. Trong lần hiệu chỉnh vị trí tham chiếu này, vectơ khối mvL [0] có thể được hiệu chỉnh. Tức là, hiệu chỉnh được thực hiện như:

$$mvL[0] = (xAvlTL - xCb) \ll 4.$$

Điều này thỏa mãn $x_{\text{RefTL}} = x_{\text{AvlTL}}$, cho phép hiệu chỉnh vị trí tham chiếu.

Theo cách này, trong trường hợp khối tham chiếu nằm ngoài vùng tham chiếu, việc hiệu chỉnh vị trí tham chiếu sẽ cho phép tham chiếu khối tham chiếu.

Bây giờ, người ta cho rằng một số vector khối trong danh sách ứng viên vector khối được xây dựng bởi khối dự đoán bản sao khối nội ảnh 352 nằm ngoài vùng có thể tham chiếu. Trong trường hợp vị trí tham chiếu không được hiệu chỉnh, sẽ không thể tham khảo các vector khối này, vô hiệu hóa việc thêm các vector khối này cho các ứng viên cho chế độ hợp nhất IBC. Ngược lại, trong trường hợp vị trí tham chiếu sẽ được hiệu chỉnh trong sáng chế, tất cả các vector khối trong danh sách ứng viên vector khối được xây dựng sẽ nằm trong vùng có thể tham chiếu. Điều này sẽ đặt tất cả các vector khối có thể tham chiếu, cho phép bổ sung tất cả các vector khối cho các ứng viên cho chế độ hợp nhất IBC. Do đó, bộ chọn chế độ hợp nhất IBC 374 có thể chọn chế độ dự đoán tối ưu từ mỗi ứng viên chế độ hợp nhất IBC tương ứng với tất cả các vector khối, từ đó cải thiện hiệu quả mã hóa.

Bây giờ, người ta cho rằng một số vector khối trong danh sách ứng viên vector khối được xây dựng bởi khối dự đoán bản sao khối nội ảnh 362 nằm ngoài vùng có thể tham chiếu. Trong trường hợp vị trí tham chiếu không được hiệu chỉnh, sẽ không thể tham khảo các vector khối này, vô hiệu hóa giải mã chế độ hợp nhất IBC bằng cách sử dụng các vector khối này. Trong thiết bị mã hóa không theo sáng chế, chỉ số hợp nhất cho biết chế độ hợp nhất IBC giả định các vector khối đó hoạt động như không được mã hóa. Tuy nhiên, một chỉ số hợp nhất như vậy có thể được mã hóa để tạo ra một luồng bit do lỗi hoạt động hoặc tương tự. Trong các trường hợp khác, một phần của luồng bit có thể bị mất do mất gói hoặc tương tự và kết quả giải mã có thể là chỉ số hợp nhất như vậy. Khi cố gắng giải mã luồng bit không đầy đủ như vậy, có khả năng

truy cập bộ nhớ ảnh được giải mã ở vị trí không chính xác trong nỗ lực đề cập đến bên ngoài vùng có thể tham chiếu. Kết quả là, kết quả giải mã khác nhau tùy thuộc vào thiết bị giải mã hoặc quy trình giải mã bị dừng lại. Ngược lại, trong trường hợp vị trí tham chiếu sẽ được hiệu chỉnh trong sáng chế, tất cả các vector khối trong danh sách ứng viên vector khối được xây dựng sẽ nằm trong vùng có thể tham chiếu. Do đó, ngay cả khi một luồng bit không đầy đủ như vậy được giải mã, vị trí tham chiếu sẽ được hiệu chỉnh bên trong vùng có thể tham chiếu, cho phép tham chiếu. Theo cách này, việc hiệu chỉnh vị trí tham chiếu sẽ đảm bảo phạm vi truy cập bộ nhớ. Điều này dẫn đến việc mua lại kết quả giải mã tương tự bởi thiết bị giải mã, cho phép tiếp tục quy trình giải mã, dẫn đến cải thiện năng suất của thiết bị giải mã.

Hơn nữa, trong trường hợp vector khối sẽ được hiệu chỉnh trong việc hiệu chỉnh vị trí tham chiếu, mục tiêu là vector khối độ sáng. Ở đây, một vector khối màu được tính từ vector khối độ sáng. Tức là, hiệu chỉnh vector khối độ sáng cũng sẽ hiệu chỉnh vector khối màu. Do đó, không cần phải hiệu chỉnh lại vị trí tham chiếu trong màu sắc. Lượng xử lý có thể được giảm so với trường hợp không có hiệu chỉnh vector khối được thực hiện và cần phải xác định xem tham chiếu có thể dựa trên cả độ sáng và màu sắc hay không.

Ngoài ra, trong trường hợp vector khối phải được hiệu chỉnh trong việc hiệu chỉnh vị trí tham chiếu, vector khối đã hiệu chỉnh sẽ được lưu trữ trong bộ nhớ lưu trữ thông tin mã hóa 111 hoặc bộ nhớ lưu trữ thông tin mã hóa 205 dưới dạng vector khối của khối mã hóa đích. Tức là, vị trí tham chiếu đã hiệu chỉnh và vị trí được chỉ định bởi vector khối là như nhau. Ở đây, việc lưu trữ giải mã dẫn đến bộ nhớ ảnh được giải mã đôi khi bao gồm quy trình lọc bỏ chặn. Trong quy trình lọc này, năng suất của bộ lọc được kiểm soát bởi sự khác biệt giữa các vector khối của hai khối đối diện với ranh

giới khối. So với trường hợp vector khối không được hiệu chỉnh và vị trí tham chiếu được hiệu chỉnh khác với vị trí được chỉ ra bởi vector khối, có thể có được cường độ bộ lọc thích hợp hơn nữa, từ đó cải thiện hiệu quả mã hóa.

Sau đó, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng y có nhỏ hơn phía trên bên trái của vùng tham chiếu (S6006) hay không. Khi xác định là sai (S6006: SAI), quy trình này tiến hành quy trình tiếp theo (S6008). Ngược lại, khi quyết định là đúng (S6006: ĐÚNG), vị trí tham chiếu theo hướng y sẽ được hiệu chỉnh phù hợp với phía trên bên trái của vùng có thể tham chiếu (S6007).

Bây giờ, giả sử rằng một khối tham chiếu r4 được đặt tại 6021, vị trí tham chiếu theo hướng y nhỏ hơn phía trên bên trái của vùng tham chiếu (S6006: ĐÚNG). Do đó, vị trí tham chiếu được hiệu chỉnh về vị trí 6022 giả định $y_{RefTL} = y_{AvlTL}$ (S6007). Ở đây, vì $y_{RefBR} = y_{RefTL} + cbHeight - 1$ giữ như trong S6001, y_{RefBR} cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của y_{RefTL} . Trong lần hiệu chỉnh vị trí tham chiếu này, vector khối $mvL[1]$ có thể được hiệu chỉnh. Tức là, hiệu chỉnh được thực hiện như:

$$mvL[1] = (y_{AvlTL} - y_{Cb}) \ll 4$$

Điều này thỏa mãn $y_{RefTL} = y_{AvlTL}$, cho phép hiệu chỉnh vị trí tham chiếu.

Sau đó, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng x có lớn hơn bên phải dưới của vùng tham chiếu (S6008) hay không. Khi xác định là sai (S6008: SAI), quy trình này tiến hành quy trình tiếp theo (S6010). Ngược lại, khi quyết định là đúng (S6008: ĐÚNG), vị trí tham chiếu trong hướng x sẽ được hiệu chỉnh phù hợp với bên phải dưới của vùng tham chiếu (S6009).

Bây giờ, giả sử rằng một khối tham chiếu r7 nằm ở 6031, vị trí tham chiếu theo hướng x lớn hơn bên dưới bên phải của vùng tham chiếu (S6008: ĐÚNG). Do đó,

vị trí tham chiếu được hiệu chỉnh về vị trí 6032 giả định $xRefBR = xAvlBR$ (S6009). Ở đây, vì $xRefBR = xRefTL + cbWidth-1$ giữ như trong S6001, nghĩa là $xRefTL = xRefBR - (cbWidth-1)$ giữ, $xRefTL$ cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của $xRefBR$. Trong lần hiệu chỉnh vị trí tham chiếu này, vector khối $mvL[0]$ có thể được hiệu chỉnh. Tức là, việc hiệu chỉnh được thực hiện như sau:

$$mvL[0] = (xAvlBR - (xCb + cbWidth-1)) \ll 4$$

Điều này thỏa mãn $xRefBR = xAvlBR$, cho phép hiệu chỉnh vị trí tham chiếu.

Sau đó, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng y có lớn hơn bên phải dưới của vùng tham chiếu (S6010) hay không. Khi xác định là sai (S6010: SAI), quy trình kết thúc. Ngược lại, khi quyết định là đúng (S6010: ĐÚNG), vị trí tham chiếu theo hướng y sẽ được hiệu chỉnh phù hợp với bên phải dưới của vùng tham chiếu (S6011).

Bây giờ, giả sử rằng một khối tham chiếu $r5$ nằm ở 6041, vị trí tham chiếu theo hướng y lớn hơn bên dưới bên phải của vùng tham chiếu (S6010: ĐÚNG). Do đó, vị trí tham chiếu được hiệu chỉnh về vị trí 6042 giả sử $yRefBR = yAvlBR$ (S6011). Ở đây, vì $yRefBR = yRefTL + cbHeight-1$ giữ như trong S6001, nghĩa là $yRefTL = yRefBR - (cbHeight-1)$ giữ, $yRefTL$ cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của $yRefBR$. Trong lần hiệu chỉnh vị trí tham chiếu này, vector khối $mvL[1]$ có thể được hiệu chỉnh. Tức là, hiệu chỉnh được thực hiện như:

$$mvL[1] = (yAvlBR - (yCb + cbHeight-1)) \ll 4$$

Điều này thỏa mãn $yRefBR = yAvlBR$, cho phép hiệu chỉnh vị trí tham chiếu.

Ở đây, một trường hợp mà một khối tham chiếu $r1$ được đặt tại 6051 sẽ được mô tả. Trong trường hợp này, vị trí tham chiếu trong hướng x được hiệu chỉnh tương tự như trường hợp khối tham chiếu là $r2$. Hơn nữa, tương tự như trường hợp khối tham

chiều là r4, vị trí tham chiếu theo hướng y được hiệu chỉnh. Do đó, khối tham chiếu r1 sẽ được đặt tại 6052 bên trong vùng tham chiếu.

Trong trường hợp một khối tham chiếu r3 nằm ở 6061, trong đó khối tham chiếu r6 nằm ở 6062, hoặc khối tham chiếu r8 nằm ở 6063, vị trí tham chiếu trong mỗi hướng x và y được hiệu chỉnh tương tự như trường hợp trên. Kết quả là, mỗi khối tham chiếu được đặt bên trong vùng có thể tham chiếu.

Quy trình trên hoàn thành quy trình đối với trường hợp kích thước CTU không phải là 128×128 mẫu. Ngược lại, trong trường hợp kích thước của CTU là 128×128 mẫu (S6002: ĐÚNG), vị trí trên bên trái và dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật được tính toán (S6012).

Fig.50 là sơ đồ minh họa các vị trí trên bên trái và dưới bên phải khi vùng có thể tham chiếu là hình chữ nhật. Trong trường hợp trên Fig.50A, khối cây mã hóa đích 6101 được chia thành bốn và một khối mã hóa đích 6102 nằm ở phía trên bên trái của các vị trí chia. Tại thời điểm này, vùng tham chiếu có hình chữ L đảo ngược giống như phần nở vào năm 6103. Trong trường hợp vùng có thể tham chiếu là hình chữ nhật, phạm vi sẽ là phạm vi của vùng hình chữ nhật 6103. Trong trường hợp vùng có thể tham chiếu là hình chữ nhật và khi phần trên bên trái của khối tham chiếu được định nghĩa là (xRefTL, yRefTL) và phần dưới bên phải được định nghĩa là (xRefBR, yRefBR), những điều sau đây sẽ thu được:

$$\text{offset}[4] = \{0, 64, 128, 128\}$$

$$\text{NL} = -\text{offset}[3-\text{blk_idx}], \text{NR} = \text{offset}[\text{blk_idx}]$$

$$(\text{xAvlTL}, \text{yAvlTL}) = ((\text{xCb} \gg \text{CtbLog2SizeY}) \ll \text{CtbLog2SizeY} + \text{NL},$$

$$(\text{yCb} \gg \text{CtbLog2SizeY}) \ll \text{CtbLog2SizeY}))$$

$$(\text{xAvlBR}, \text{yAvlBR}) = (((\text{xCb} \gg \text{CtbLog2SizeY}) \ll \text{CtbLog2SizeY}) - 1 + \text{NR},$$

$$(((yCb \gg CtbLog2SizeY) + 1) \ll CtbLog2SizeY) - 1).$$

Ở đây, blk_idx là một chỉ số cho biết vị trí của khối mã hóa đích. Trong trường hợp khối mã hóa đích nằm ở vị trí trên bên trái sau khi chia khối cây mã hóa đích, sẽ thiết lập được; blk_idx = 0. Tương tự, trong trường hợp các khối mã hóa đích nằm ở vị trí trên bên phải, dưới bên trái và dưới bên phải, blk_idx sẽ được đặt thành 1, 2 và 3. Fig.50A là sơ đồ minh họa một trường hợp blk_idx = 0. Tương tự như vậy, các hình vẽ từ Fig.50B đến Fig.50D là sơ đồ minh họa các trường hợp blk_idx = 1 đến blk_idx = 3, tương ứng.

Tiếp theo, vị trí tham chiếu của vùng tham chiếu không hình chữ nhật được hiệu chỉnh (S6013). Fig.51 là lưu đồ minh họa một quy trình hiệu chỉnh vị trí tham chiếu của một phần của vùng có thể tham chiếu không phải là hình chữ nhật. Đầu tiên, vị trí trên bên trái của vùng tham chiếu được tính (S6021). Vì vùng có thể tham chiếu là phần gạch chéo trên Fig.50, có hai vị trí trên bên trái 6111 và 6112, ngoại trừ trường hợp blk_idx = 3. Khi hai điểm này lần lượt là (X1, Y1), (X2, Y2), các vị trí được tính là:

$$\text{offset}[4] = \{64, 128, 64, 0\}, \text{NL} = \text{offset}[\text{blk_idx}]$$

$$(X1, Y1) = (xAvlTL, yAvlTL + 64)$$

$$(X2, Y2) = (xAvlTL + \text{NL}, yAvlTL)$$

Tiếp theo, người ta xác định xem vị trí tham chiếu có được hiệu chỉnh theo phía trên bên trái của vùng tham chiếu (S6022) hay không. Trong quyết định này, xác định được là đúng trong trường hợp blk_idx = 3 không thỏa mãn và khối tham chiếu nằm trong một vùng nhỏ hơn X2 và Y1 (S6022: ĐÚNG). Trong trường hợp sai (S6022: SAI), quy trình tiến hành quy trình tiếp theo (S6026).

Tiếp theo, người ta xác định xem sự khác biệt giữa khối tham chiếu và vùng có

thể tham chiếu trong hướng x có nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng y (S6023). Trong trường hợp xác định là đúng (S6023: ĐÚNG), vị trí tham chiếu trong hướng x sẽ được hiệu chỉnh (S6024). Ngược lại, trong trường hợp xác định là sai (S6023: SAI), vị trí tham chiếu trong hướng y sẽ được hiệu chỉnh (S6025).

Fig.52A là sơ đồ minh họa trạng thái hiệu chỉnh vị trí tham chiếu trong S6024 và S6025. Bây giờ, blk_idx = 0. Giả sử rằng khối tham chiếu r1 nằm ở 6201, blk_idx = 3 không thỏa mãn và phần trên bên trái của khối tham chiếu nằm trong một vùng nhỏ hơn X2 (phương x 6112) và Y1 (phương Y 6111) (S6022: ĐÚNG). Hơn nữa, sự khác biệt giữa khối tham chiếu và vùng có thể tham chiếu trong hướng x nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu trong hướng y (S6023: ĐÚNG). Do đó, vị trí tham chiếu theo hướng x sẽ được hiệu chỉnh về vị trí 6202 giả định xRefTL = xAvlTL + NL (S6024). Ở đây, vì xRefBR = xRefTL + cbWidth-1 giữ như trong S6001, xRefBR cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của xRefTL. Trong lần hiệu chỉnh vị trí tham chiếu này, vector khối mvL [0] có thể được hiệu chỉnh. Tức là, hiệu chỉnh được thực hiện như:

$$mvL [0] = (xAvlTL + NL - xCb) \ll 4$$

Điều này thỏa mãn xRefTL = xAvlTL + NL, cho phép hiệu chỉnh vị trí tham chiếu.

Ngược lại, khi khối tham chiếu r2 nằm ở 6203, blk_idx = 3 không thỏa mãn và phần trên bên trái của khối tham chiếu nằm ở một vùng nhỏ hơn X2 (6112 phương x) và Y1 (6111 phương y) (S6022: ĐÚNG). Hơn nữa, chênh lệch giữa khối tham chiếu và vùng tham chiếu trong hướng x không nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng y (S6023: SAI). Do đó, vị trí tham chiếu theo hướng y

được hiệu chỉnh về vị trí 6204 giả sử $y_{RefTL} = y_{AvlTL} + 64$ (S6025). Ở đây, vì $y_{RefBR} = y_{RefTL} + cbHeight-1$ giữ như trong S6001, y_{RefBR} cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của y_{RefTL} . Trong lần hiệu chỉnh vị trí tham chiếu này, vectơ khối $mvL[0]$ có thể được hiệu chỉnh. Tức là, việc hiệu chỉnh được thực hiện như sau:

$$mvL[1] = (y_{AvlTL} + 64 - y_{Cb}) \ll 4$$

Điều này thỏa mãn $y_{RefTL} = y_{AvlTL} + 64$, cho phép hiệu chỉnh vị trí tham chiếu.

Ở đây, giả định rằng khối tham chiếu $r3$ nằm ở 6205. Trong trường hợp này, sự khác biệt giữa khối tham chiếu và vùng có thể tham chiếu theo hướng x nhỏ hơn sự khác biệt giữa khối tham chiếu và vùng có thể tham chiếu theo hướng y (S6023: ĐÚNG). Theo đó, vị trí tham chiếu theo hướng x được hiệu chỉnh tương tự như khối tham chiếu $r1$ để nằm ở 6206 (S6024). Tại thời điểm này, khối tham chiếu nằm ngoài vùng có thể tham chiếu. Tuy nhiên, vị trí tham chiếu theo hướng y được hiệu chỉnh bởi các quy trình của S6006 và S6007 được mô tả bên dưới. Do đó, khối tham chiếu được đặt bên trong vùng có thể tham chiếu.

Sau đó, vị trí phía dưới bên phải của vùng có thể tham chiếu được tính toán (S6026). Vì vùng có thể tham chiếu là một phần được tạo ra trên Fig.50, có hai vị trí phía dưới bên phải, 6113 và 6114, ngoại trừ trường hợp $blk_idx = 0$. Khi hai điểm này lần lượt là $(X3, Y3)$ và $(X4, Y4)$, vị trí được tính là:

$$offset[4] = \{0, 64, 128, 64\}, NR = offset[blk_idx]$$

$$(X3, Y3) = (x_{AvlBR}, y_{AvlBR}-64)$$

$$(X4, Y4) = (x_{AvlBR}-NR, y_{AvlBR})$$

Tiếp theo, sẽ xác định xem vị trí tham chiếu sẽ được hiệu chỉnh theo phía dưới

bên phải của vùng có thể tham chiếu (S6027) hay không. Trong xác định này, xác định được là đúng khi blk_idx không phải là 0 và khối tham chiếu nằm trong vùng lớn hơn X4 và Y3 (S6027: ĐÚNG). Trong trường hợp xác định là sai (S6027: SAI), quy trình kết thúc.

Tiếp theo, xác định được xem liệu sự khác biệt giữa khối tham chiếu và vùng có thể tham chiếu theo hướng x có nhỏ hơn sự khác biệt giữa khối tham chiếu và vùng có thể tham chiếu theo hướng y hay không (S6028). Trong trường hợp xác định là đúng (S6028: ĐÚNG), vị trí tham chiếu theo hướng x sẽ được hiệu chỉnh (S6029). Ngược lại, trong trường hợp xác định là sai (S6028: SAI), vị trí tham chiếu theo hướng y sẽ được hiệu chỉnh (S6030).

Fig.52B là sơ đồ minh họa trạng thái hiệu chỉnh vị trí tham chiếu trong S6029 và S6030. Bây giờ, blk_idx = 3. Giả sử khối tham chiếu r1 nằm ở 6211, blk_idx = 0 không thỏa mãn và phần dưới bên phải của khối tham chiếu nằm trong một vùng lớn hơn X4 (phương x 6114) và Y3 (phương y 6113) (S6027: ĐÚNG). Hơn nữa, chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng x nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng y (S6028: ĐÚNG). Do đó, vị trí tham chiếu trong hướng x được hiệu chỉnh về vị trí 6212 giả định xRefBR = xAvlBR (S6029). Ở đây, vì xRefBR = xRefTL + cbWidth-1 giữ như trong S6001, nghĩa là xRefTL = xRefBR - (cbWidth-1) giữ, xRefTL cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của xRefBR. Trong việc hiệu chỉnh vị trí tham chiếu này, vector khối mvL [0] có thể được hiệu chỉnh. Tức là, việc hiệu chỉnh được thực hiện như sau:

$$mvL[0] = (xAvlBR - NR - (xCb + cbWidth - 1)) \ll 4$$

Điều này thỏa mãn xRefBR = xAvlBR, cho phép hiệu chỉnh vị trí tham chiếu.

Ngược lại, giả sử rằng khối tham chiếu r2 nằm ở 6213, blk_idx = 0 không thỏa

mã và bên dưới bên phải của khối tham chiếu nằm trong một vùng lớn hơn X4 (phương x 6114) và Y3 (phương y 6113) (S6027: ĐÚNG). Hơn nữa, chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng x không nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng y (S6028: SAI). Do đó, vị trí tham chiếu hướng y được hiệu chỉnh về vị trí 6214 giả sử $y_{RefBR} = y_{AvlBR}$ (S6030). Ở đây, vì $y_{RefBR} = y_{RefTL} + cbHeight - 1$ giữ như trong S6001, nghĩa là $y_{RefTL} = y_{RefBR} - (cbHeight - 1)$ giữ, y_{RefTL} cũng sẽ được hiệu chỉnh cùng với sự hiệu chỉnh của y_{RefBR} . Trong lần hiệu chỉnh vị trí tham chiếu này, vector khối $mvL[1]$ có thể được hiệu chỉnh. Tức là, việc hiệu chỉnh được thực hiện như sau:

$$mvL[1] = (y_{AvlBR} - 64 - (yCb + cbHeight - 1)) \ll 4$$

Điều này thỏa mãn $y_{RefBR} = y_{AvlBR}$, cho phép hiệu chỉnh vị trí tham chiếu.

Ở đây, người ta cho rằng khối tham chiếu $r3$ nằm ở 6215. Trong trường hợp này, chênh lệch giữa khối tham chiếu và vùng có thể tham chiếu theo hướng x không nhỏ hơn chênh lệch giữa khối tham chiếu và vùng tham chiếu theo hướng y (S6028: SAI). Theo đó, vị trí tham chiếu theo hướng y được hiệu chỉnh tương tự như khối tham chiếu $r2$ để được đặt tại 6216 (S6030). Tại thời điểm này, khối tham chiếu nằm ngoài vùng có thể tham chiếu. Tuy nhiên, vị trí tham chiếu trong hướng x được hiệu chỉnh bởi các quy trình của S6008 và S6009 được mô tả dưới đây. Do đó, khối tham chiếu được đặt bên trong vùng có thể tham chiếu.

Trên Fig.52, quy trình việc hiệu chỉnh vị trí tham chiếu đã được mô tả bằng cách sử dụng các trường hợp mẫu khi $blk_idx = 0$ và $blk_idx = 3$. Trong trường hợp $blk_idx = 1$ hoặc $blk_idx = 2$, quy trình việc hiệu chỉnh vị trí tham chiếu được thực hiện tương tự như các trường hợp $blk_idx = 0$ và $blk_idx = 3$.

Sau quy trình hiệu chỉnh vị trí tham chiếu của vùng tham chiếu không hình

chữ nhật (S6013), các quy trình của S6004 đến S6011 được thực hiện. Thủ tục trên hoàn thành quy trình đối với trường hợp kích thước CTU là 128×128 mẫu.

Bây giờ, người ta cho rằng trong quy trình việc hiệu chỉnh vị trí tham chiếu của phần không hình chữ nhật của vùng tham chiếu (S6013), quy trình hiệu chỉnh vị trí tham chiếu theo hướng x được thực hiện (S6024) phù hợp với phía trên bên trái của vùng có thể tham chiếu. Trong trường hợp này, vị trí tham chiếu trong hướng x của khối tham chiếu sẽ không trở nên nhỏ hơn phía trên bên trái của vùng tham chiếu, và do đó, xác định trong S6004 luôn là sai (S6004: SAI). Do đó, trong trường hợp quy trình S6024 đã được thực hiện, các quy trình của S6004 và S6005 có thể bị ngăn chặn. Tương tự như vậy, trong trường hợp xử lý S6025 đã được thực hiện, các quy trình của S6006 và S6007 có thể bị ngăn chặn. Trong trường hợp quy trình S6029 đã được thực hiện, các quy trình của S6008 và S6009 có thể bị ngăn chặn. Trong trường hợp quy trình S6030 đã được thực hiện, các quy trình của S6010 và S6011 có thể bị ngăn chặn. Hơn nữa, có thể sử dụng cấu hình, trong lưu đồ trên Fig.51, trong đó quy trình so sánh bước S6023 được bỏ qua và bước S6024 luôn được thực hiện, hoặc bước S6025 luôn được thực hiện. Tương tự như vậy, có thể sử dụng cấu hình trong đó quy trình so sánh của bước S6028 được bỏ qua và bước S6029 luôn được thực hiện, hoặc bước S6030 luôn được thực hiện. Trong cấu hình như vậy, vị trí tham chiếu có thể được hiệu chỉnh bằng một quy trình đơn giản.

Trên Fig.48, vị trí tham chiếu được hiệu chỉnh bằng cách sử dụng các quy trình của S6012, S6013 và S6004 đến S6011 trong trường hợp kích thước của CTU là 128×128 mẫu. Thay vào đó, như minh họa trên Fig.53, điều này cũng có thể được thực hiện bằng quy trình chia vùng có thể tham chiếu thành hai và sau đó hiệu chỉnh từng vị trí tham chiếu (S6101).

Fig.54 là sơ đồ minh họa trạng thái chia một vùng có thể tham chiếu thành hai. Không giống như vùng có thể tham chiếu hình chữ nhật trên Fig.50, vùng có thể tham chiếu được chia thành hai trên Fig.54. Trong trường hợp khối mã hóa đích (6102) nằm ở vị trí trên bên trái sau khi chia khối cây mã hóa đích (6101), cài đặt sẽ là: blk_idx = 0. Tương tự, trong trường hợp các khối mã hóa đích nằm ở vị trí trên bên phải, dưới bên trái và dưới bên phải, blk_idx sẽ được đặt thành 1, 2 và 3. Fig.54A là sơ đồ minh họa một trường hợp blk_idx = 0. Tương tự như vậy, Fig.54B đến 54D là sơ đồ minh họa các trường hợp blk_idx = 1, blk_idx = 2 và blk_idx = 3, tương ứng. Hơn nữa, một vùng có thể tham chiếu (6301) được định nghĩa là vùng A có thể tham chiếu và vùng có thể tham chiếu khác (6302) được định nghĩa là vùng B có thể tham chiếu.

Fig.55 là lưu đồ minh họa một quy trình (S6101) chia vùng tham chiếu thành hai và hiệu chỉnh từng vị trí tham chiếu của các vùng có thể tham chiếu. Trên Fig.55, quy trình tương tự như trên Fig.48 được chỉ định bởi cùng một số bước, và mô tả của nó sẽ được bỏ qua. Đầu tiên, các vị trí trên bên trái và dưới bên phải của vùng A có thể tham chiếu được tính toán (S6111). Khi phía trên bên trái của vùng A có thể tham chiếu được định nghĩa là (xAvtlTL, yAvtlTL) và phía dưới bên phải là (xAvtlBR, yAvtlBR), những điều sau đây sẽ thu được:

$$xOffsetTL[4] = \{-128, -128, -64, 0\}, yOffsetTL[4] = \{64, 64, 64, 0\}$$

$$xOffsetBR[4] = \{0, 0, 0, 128\}, yOffsetBR[4] = \{128, 128, 128, 64\}$$

$$(xAvtlTL, yAvtlTL) = ((xCb \gg CtbLog2SizeY) \ll CtbLog2SizeY$$

$$+ xOffsetTL[blk_idx],$$

$$(yCb \gg CtbLog2SizeY) \ll CtbLog2SizeY + yOffsetTL[blk_idx])$$

$$(xAvtlBR, yAvtlBR) = (((xCb \gg CtbLog2SizeY) \ll CtbLog2SizeY) - 1$$

$$+ xOffsetBR[blk_idx],$$

$$(((yCb \gg CtbLog2SizeY) + 1) \ll CtbLog2SizeY) - 1 + yOffsetBR [blk_idx])$$

Tiếp theo, tính toán về việc khối tham chiếu có nằm ngoài vùng tham chiếu A có được thực hiện như sau (S6112):

$$out_xRefTL = xRefTL < xAvlTL$$

$$out_yRefTL = yRefTL < yAvlTL$$

$$out_xRefBR = xRefBR > xAvlBR$$

$$out_yRefBR = yRefBR > yAvlBR$$

Tiếp theo, các vị trí trên bên trái và dưới bên phải của vùng B có thể tham chiếu được tính toán (S6113). Khi phía trên bên trái của vùng B có thể tham chiếu được định nghĩa là (xAvlTL, yAvlTL) và phía dưới bên phải là (xAvlBR, yAvlBR), những điều sau đây thu được:

$$xOffsetTL [4] = \{-64, 0, 0, 0\}, yOffsetTL [4] = \{0, 0, 0, 0\}$$

$$xOffsetBR [4] = \{0, 64, 128, 64\}, yOffsetBR [4] = \{128, 64, 64, 128\}$$

$$(xAvlTL, yAvlTL) = ((xCb \gg CtbLog2SizeY) \ll CtbLog2SizeY + xOffsetTL [blk_idx],$$

$$(yCb \gg CtbLog2SizeY) \ll CtbLog2SizeY + yOffsetTL [blk_idx])$$

$$(xAvlBR, yAvlBR) = (((xCb \gg CtbLog2SizeY) \ll CtbLog2SizeY) - 1 + xOffsetBR [blk_idx],$$

$$(((yCb \gg CtbLog2SizeY) + 1) \ll CtbLog2SizeY) - 1 + yOffsetBR [blk_idx])$$

Tiếp theo, người ta xác định vị trí tham chiếu của khối tham chiếu theo hướng x có nhỏ hơn phía trên bên trái của vùng tham chiếu A hay không và vị trí tham chiếu của khối tham chiếu theo hướng x có nhỏ hơn phía trên bên trái của vùng tham chiếu B (S6114) hay không. Khi xác định là sai (S6114: SAI), quy trình này tiến hành quy trình tiếp theo (S6116). Ngược lại, khi quyết định là đúng (S6114: ĐÚNG), vị trí tham chiếu

trong hướng x sẽ được hiệu chỉnh phù hợp với phía trên bên trái của vùng có thể tham chiếu B (S6005). Vì quy trình của S6005 đã được mô tả, phần mô tả được bỏ qua.

Sau đó, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng y có nhỏ hơn phía trên bên trái của vùng tham chiếu A hay không và vị trí tham chiếu của khối tham chiếu theo hướng y có nhỏ hơn phía trên bên trái của vùng tham chiếu B (S6116) hay không. Khi xác định là sai (S6116: SAI), quy trình này tiến hành quy trình tiếp theo (S6118). Ngược lại, khi quyết định là đúng (S6116: ĐÚNG), vị trí tham chiếu theo hướng y sẽ được hiệu chỉnh phù hợp với phía trên bên trái của vùng có thể tham chiếu B (S6007). Do quy trình xử lý S6007 đã được mô tả, phần mô tả được bỏ qua.

Tiếp theo, người ta xác định vị trí tham chiếu trong hướng x của khối tham chiếu có lớn hơn bên phải dưới của vùng tham chiếu A hay không và vị trí tham chiếu của khối tham chiếu theo hướng x có lớn hơn bên dưới của vùng tham chiếu B (S6118) hay không. Khi xác định là sai (S6118: SAI), quy trình này tiến hành quy trình tiếp theo (S6120). Ngược lại, khi quyết định là đúng (S6118: ĐÚNG), vị trí tham chiếu trong hướng x sẽ được hiệu chỉnh phù hợp với bên phải dưới của vùng tham chiếu B (S6009). Do quy trình xử lý S6009 đã được mô tả, phần mô tả được bỏ qua.

Tiếp theo, người ta xác định xem vị trí tham chiếu của khối tham chiếu theo hướng y có lớn hơn bên phải dưới của vùng tham chiếu A hay không và vị trí tham chiếu của khối tham chiếu theo hướng y có lớn hơn bên phải dưới của vùng tham chiếu B (S6120) hay không. Khi xác định là sai (S6120: SAI), quy trình kết thúc. Ngược lại, khi quyết định là đúng (S6120: ĐÚNG), vị trí tham chiếu theo hướng y sẽ được hiệu chỉnh phù hợp với bên phải dưới của vùng tham chiếu B (S6011). Do quy trình xử lý S6011 đã được mô tả, phần mô tả được bỏ qua.

Như đã mô tả ở trên, ngay cả trong trường hợp khối tham chiếu nằm ngoài vùng tham chiếu khi kích thước của CTU là 128×128 mẫu, vị trí tham chiếu có thể được hiệu chỉnh và trở nên có thể tham chiếu. Hơn nữa, bằng cách chia vùng có thể tham chiếu thành hai và hiệu chỉnh từng vị trí tham chiếu, có thể đơn giản hóa quy trình và giảm số lượng tính toán. Ở đây, một vùng có thể tham chiếu (6301) được định nghĩa là vùng A có thể tham chiếu, trong khi vùng có thể tham chiếu khác (6302) được định nghĩa là vùng B có thể tham chiếu. Ngoài ra, vùng A có thể tham chiếu và vùng B có thể tham chiếu có thể được trao đổi, và một vùng có thể tham chiếu (6301) có thể được xử lý như vùng B có thể tham chiếu trong khi vùng có thể tham chiếu khác (6302) có thể được xử lý như vùng có thể tham chiếu A.

Trong ví dụ hiện tại, người ta xác định xem kích thước của CTU là 128×128 mẫu (S6002) và quy trình được chuyển đổi cho phù hợp. Quyết định này có thể được thay thế bằng quyết định xem khối chuẩn bản sao khối nội ảnh có được cung cấp trong đơn vị thu được bằng cách tách khối cây mã hóa hay xác định xem kích thước của CTU có lớn hơn kích thước tối đa của khối mã hóa hay không.

Trong tất cả các phương án được mô tả ở trên, nhiều kỹ thuật có thể được kết hợp với nhau.

Trong tất cả các phương án được mô tả ở trên, luồng bit được xuất ra từ thiết bị mã hóa hình ảnh có định dạng dữ liệu cụ thể như vậy sẽ được giải mã theo phương pháp giả mã sử dụng trong phương án này. Luồng bit có thể được cung cấp bằng cách được ghi và được cung cấp trong môi trường lưu trữ có thể đọc được bằng máy tính hoặc tương tự, chẳng hạn như HDD, SSD, bộ nhớ flash hoặc ổ đĩa quang, hoặc có thể được cung cấp từ máy chủ thông qua mạng hữu tuyến hoặc vô tuyến. Theo đó, thiết bị giải mã hình ảnh tương ứng với thiết bị mã hóa hình ảnh có khả năng giải mã luồng bit

có định dạng dữ liệu cụ thể bất kể kiểu thiết bị nào.

Trong trường hợp mạng không dây hoặc mạng có dây được sử dụng để trao đổi luồng bit giữa thiết bị mã hóa hình ảnh và thiết bị giải mã hình ảnh, luồng bit có thể được biến đổi thành dạng dữ liệu thích hợp với định dạng truyền của kênh truyền trong quy trình truyền dẫn. Trong trường hợp này, có thiết bị truyền biến đổi luồng bit được xuất ra từ thiết bị mã hóa hình ảnh thành dữ liệu được mã hóa ở định dạng dữ liệu phù hợp với định dạng truyền của kênh truyền và truyền dữ liệu được mã hóa vào mạng, và thiết bị nhận nhận dữ liệu được mã hóa từ mạng sẽ được khôi phục lại thành luồng bit và đưa luồng bit tới thiết bị giải mã hình ảnh. Thiết bị truyền bao gồm bộ nhớ đệm luồng bit được xuất ra từ thiết bị mã hóa hình ảnh, khối xử lý gói đóng gói luồng bit, và bộ truyền truyền dữ liệu mã hóa đã được đóng gói thông qua mạng. Thiết bị nhận bao gồm khối nhận để nhận dữ liệu mã hóa đã được đóng gói qua mạng, bộ nhớ đệm dữ liệu mã hóa nhận được, và khối xử lý gói đóng gói dữ liệu được mã hóa để tạo ra luồng bit và đưa luồng bit được tạo ra tới thiết bị giải mã hình ảnh.

Trong trường hợp mạng hữu tuyến hoặc vô tuyến được sử dụng để trao đổi luồng bit giữa thiết bị mã hóa hình ảnh và thiết bị giải mã hình ảnh, điều này có thể cung cấp, ngoài thiết bị phát và thiết bị thu, thiết bị chuyển tiếp nhận dữ liệu được mã hóa được truyền bởi thiết bị truyền và cung cấp dữ liệu đã được mã hóa cho thiết bị thu. Ngoài ra, thiết bị chuyển tiếp bao gồm bộ thu nhận dữ liệu được mã hóa được đóng gói được truyền bởi thiết bị truyền, bộ nhớ đệm dữ liệu được mã hóa đã nhận và bộ phát truyền dữ liệu được mã hóa được đóng gói tới mạng. Ngoài ra, thiết bị chuyển tiếp có thể bao gồm khối xử lý gói nhận thực hiện xử lý đóng gói dữ liệu được mã hóa được đóng gói để tích lũy các luồng bit, phương tiện ghi lưu trữ luồng bit và khối xử lý gói truyền tải đóng gói các luồng bit.

Ngoài ra, khối hiển thị hiển thị hình ảnh đã được giải mã bởi thiết bị giải mã hình ảnh có thể được bổ sung cho cấu hình, như là thiết bị hiển thị. Trong trường hợp đó, khối hiển thị đọc ra tín hiệu hình ảnh đã được giải mã được tạo bởi khối xếp chồng tín hiệu hình ảnh đã được giải mã 207 và được lưu trong bộ nhớ hình ảnh đã được giải mã 208, và hiển thị tín hiệu trên màn hình.

Ngoài ra, khối hình ảnh có thể được thêm vào cấu hình để hoạt động như thiết bị hình ảnh bằng cách đưa hình ảnh thu nhận được tới thiết bị mã hóa hình ảnh. Trong trường hợp đó, khối hình ảnh đưa tín hiệu hình ảnh thu nhận được tới khối phân chia khối 101.

Fig.37 minh họa ví dụ về cấu hình phân cứng của thiết bị mã hóa-giải mã theo phương án của sáng chế. Thiết bị mã hóa-giải mã bao gồm các cấu hình của thiết bị mã hóa hình ảnh và thiết bị giải mã hình ảnh theo các phương án của sáng chế. thiết bị mã hóa-giải mã 9000 bao gồm CPU 9001, bộ mã hóa IC 9002, I/O giao diện 9003, bộ nhớ 9004, ổ đĩa quang 9005, giao diện mạng 9006, và giao diện video 9009, trong đó các khối riêng lẻ được kết hợp với nhau bởi bus 9010.

Bộ mã hóa hình ảnh 9007 và giải mã hình ảnh 9008 thường được triển khai là bộ mã hóa IC 9002. Quy trình mã hóa hình ảnh của thiết bị mã hóa hình ảnh theo các phương án của sáng chế được thực hiện bởi bộ mã hóa hình ảnh 9007. Quy trình giải mã hình ảnh trong thiết bị giải mã hình ảnh theo phương án của sáng chế được thực hiện bởi bộ giải mã hình ảnh 9008. Giao diện I/O 9003 được triển khai là giao diện USB, ví dụ, và kết nối với bàn phím bên ngoài 9104, chuột 9105, hoặc thiết bị tương tự. CPU 9001 điều khiển thiết bị mã hóa-giải mã 9000 dựa trên đầu vào là Thao tác của người dùng thông qua giao diện I/O 9003 để thực hiện thao tác mà người dùng mong muốn. Thao tác của người dùng trên bàn phím 9104, chuột 9105, hoặc thiết bị

tương tự bao gồm việc lựa chọn sẽ thực hiện chức năng mã hóa hay giải mã, thiết lập chất lượng mã hóa, đích đầu vào/đầu ra của luồng bit, đích đầu vào/đầu ra của hình ảnh, hoặc tương tự.

Trong trường hợp người dùng mong muốn thao tác tái tạo hình ảnh được ghi trên phương tiện ghi đĩa 9100, ổ đĩa quang 9005 đọc ra luồng bit từ phương tiện ghi đĩa 9100 được lắp vào, và truyền luồng bit được đọc ra tới bộ giải mã hình ảnh 9008 của bộ mã hóa IC 9002 thông qua bus 9010. Bộ giải mã hình ảnh 9008 thực hiện quy trình giải mã hình ảnh trong thiết bị giải mã hình ảnh trên luồng bit được đưa vào theo các phương án của sáng chế, và truyền hình ảnh đã được giải mã tới màn hình ngoại vi 9103 thông qua giao diện video 9009. Thiết bị mã hóa-giải mã 9000 có giao diện mạng 9006, và có thể kết nối tới máy chủ phân bố ngoại vi 9106 và thiết bị đầu cuối di động 9107 qua mạng 9101. Trong trường hợp người dùng mong muốn sao chép lại hình ảnh được ghi trên máy chủ phân phối 9106 hoặc thiết bị đầu cuối di động 9107 thay cho hình ảnh được ghi trên phương tiện ghi đĩa 9100, giao diện mạng 9006 thu được luồng bit từ mạng 9101 thay cho đọc ra luồng bit từ phương tiện ghi đĩa đầu vào 9100. Trong trường hợp người dùng mong muốn sao chép lại hình ảnh được ghi trong bộ nhớ 9004, quy trình giải mã hình ảnh được thực hiện bởi thiết bị giải mã hình ảnh theo phương án của sáng chế trên luồng bit được ghi trong bộ nhớ 9004.

Trong trường hợp người dùng mong muốn thực hiện hoạt động mã hóa hình ảnh thu được bởi camera ngoại vi 9102 và ghi hình ảnh trong bộ nhớ 9004, giao diện video 9009 lấy hình ảnh từ camera 9102, và truyền hình ảnh tới khối mã hóa hình ảnh 9007 của bộ mã hóa IC 9002 thông qua bus 9010. Khối mã hóa hình ảnh 9007 thực hiện quy trình mã hóa hình ảnh bởi thiết bị mã hóa hình ảnh theo phương án của sáng chế trên hình ảnh được đưa vào thông qua giao diện video 9009 và nhờ đó tạo ra luồng

bit. Sau đó, luồng bit được truyền đến bộ nhớ 9004 thông qua bus 9010. Trong trường hợp người dùng mong muốn ghi luồng bit trên phương tiện ghi đĩa 9100 thay cho bộ nhớ 9004, ổ đĩa quang 9005 ghi luồng bit trên phương tiện ghi đĩa 9100 được lắp vào.

Cũng có thể triển khai cấu hình phần cứng có thiết bị mã hóa hình ảnh và không có thiết bị giải mã hình ảnh, hoặc cấu hình phần cứng có thiết bị giải mã hình ảnh và không có thiết bị mã hóa hình ảnh. Cấu hình phần cứng này được thực hiện bằng cách thay thế bộ mã hóa IC 9002 bằng khối mã hóa hình ảnh 9007 hoặc bộ giải mã hình ảnh 9008.

Quy trình nêu trên liên quan đến mã hóa và giải mã có thể được triển khai một cách tự nhiên như thiết bị truyền, lưu và nhận sử dụng phần cứng, và ngoài ra, quy trình có thể được thực hiện bởi phần mềm được lưu trong bộ nhớ chỉ đọc (ROM), bộ nhớ flash, hoặc tương tự, hoặc bởi phần mềm được thiết kế cho máy tính hoặc tương tự. Chương trình phần sụn và chương trình phần mềm có thể được cung cấp bằng cách ghi trên môi trường lưu trữ có thể đọc được bởi máy tính hoặc tương tự, có thể được cung cấp từ máy chủ thông qua mạng có dây hoặc không dây, hoặc có thể được cung cấp thông qua quảng bá dữ liệu bởi hệ thống truyền hình số mặt đất hoặc vệ tinh.

Sáng chế đã được mô tả tham chiếu đến các phương án này. Các phương án được mô tả nêu trên được đưa ra chỉ nhằm mục đích minh họa. Thay vào đó, người có trình độ trung bình trong lĩnh vực có thể hình dung được các ví dụ hiệu chỉnh đối khác nhau có thể được thực hiện bằng cách thực hiện các tổ hợp phần tử hoặc quy trình được mô tả ở trên, cũng nằm trong phạm vi kỹ thuật của sáng chế.

Khả năng ứng dụng trong công nghiệp

Sáng chế có thể được sử dụng cho các kỹ thuật mã hóa và giải mã hình ảnh để phân chia hình ảnh thành các khối để thực hiện dự đoán.

Danh sách số tham chiếu

- 100 thiết bị mã hóa hình ảnh
- 101 khối phân chia khối
- 102 khối dự đoán liên ảnh
- 103 khối dự đoán nội ảnh
- 104 bộ nhớ hình ảnh đã được giải mã
- 105 khối xác định phương pháp dự đoán
- 106 khối tạo phần dư
- 107 khối lượng tử hóa/biến đổi trực giao
- 108 khối mã hóa chuỗi bit
- 109 khối lượng tử hóa ngược/biến đổi trực giao ngược
- 110 khối xếp chồng tín hiệu hình ảnh đã được giải mã
- 111 bộ nhớ lưu trữ thông tin mã hóa
- 200 thiết bị giải mã hình ảnh
- 201 khối giải mã chuỗi bit
- 202 khối phân chia khối
- 203 khối dự đoán liên ảnh
- 204 khối dự đoán nội ảnh
- 205 bộ nhớ lưu trữ thông tin mã hóa
- 206 khối lượng tử hóa ngược/biến đổi trực giao ngược
- 207 khối xếp chồng tín hiệu hình ảnh đã được giải mã
- 208 bộ nhớ hình ảnh đã được giải mã

YÊU CẦU BẢO HỘ

1. Thiết bị mã hóa hình ảnh thực hiện mã hóa trong các đơn vị của các khối chuẩn sao chép khối nội ảnh, thiết bị này bao gồm:

khối rút ra ứng viên vector khối được tạo cấu hình để rút ra các ứng viên vector khối của khối đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa;

khối lựa chọn được tạo cấu hình để lựa chọn vector khối đã lựa chọn được từ các ứng viên vector khối;

khối hiệu chỉnh vị trí tham chiếu được tạo cấu hình để thực hiện hiệu chỉnh liên quan đến khối tham chiếu sẽ được đề cập tới bởi vector khối đã được lựa chọn do vậy vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu; và

khối dự đoán được tạo cấu hình để thu mẫu đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã dưới dạng giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu,

trong đó khối hiệu chỉnh vị trí tham chiếu thiết lập số lượng khối chuẩn sao chép khối nội ảnh cố định trải qua quy trình mã hóa ngay trước khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng có thể tham chiếu, và thiết lập khối chuẩn sao chép khối nội ảnh trước vùng có thể tham chiếu và khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng tham chiếu không hợp lệ bất kể quy trình mã hóa có hoàn tất hay không.

2. Phương pháp mã hóa hình ảnh thực hiện mã hóa trong các đơn vị của các khối chuẩn sao chép khối nội ảnh, phương pháp này bao gồm:

bước rút ra ứng viên vector khối để rút ra các ứng viên vector khối của khối

đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa;

bước lựa chọn để lựa chọn vector khối đã lựa chọn được từ các ứng viên vector khối;

bước hiệu chỉnh vị trí tham chiếu để thực hiện hiệu chỉnh liên quan đến khối tham chiếu sẽ được đề cập tới bởi vector khối đã được lựa chọn do vậy vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu; và

bước dự đoán để thu mẫu đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã dưới dạng giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu,

trong đó bước hiệu chỉnh vị trí tham chiếu thiết lập số lượng khối chuẩn sao chép khối nội ảnh cố định trải qua quy trình mã hóa ngay trước khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng có thể tham chiếu, và thiết lập khối chuẩn sao chép khối nội ảnh trước vùng có thể tham chiếu và khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng tham chiếu không hợp lệ bất kể quy trình mã hóa có hoàn tất hay không.

3. Thiết bị giải mã hình ảnh thực hiện giải mã trong các đơn vị của các khối chuẩn sao chép khối nội ảnh, thiết bị này bao gồm:

khối rút ra ứng viên vector khối được tạo cấu hình để rút ra các ứng viên vector khối của khối đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa;

khối lựa chọn được tạo cấu hình để lựa chọn vector khối đã lựa chọn được từ các ứng viên vector khối;

khối hiệu chỉnh vị trí tham chiếu được tạo cấu hình để thực hiện hiệu chỉnh liên quan đến khối tham chiếu sẽ được đề cập tới bởi vector khối đã được lựa chọn do vậy vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu; và

khối dự đoán được tạo cấu hình để thu mẫu đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã dưới dạng giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu,

trong đó khối hiệu chỉnh vị trí tham chiếu thiết lập số lượng khối chuẩn sao chép khối nội ảnh cố định trải qua quy trình mã hóa ngay trước khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng có thể tham chiếu, và thiết lập khối chuẩn sao chép khối nội ảnh trước vùng có thể tham chiếu và khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng tham chiếu không hợp lệ bất kể quy trình mã hóa có hoàn tất hay không.

4. Phương pháp giải mã hình ảnh thực hiện giải mã trong các đơn vị của các khối chuẩn sao chép khối nội ảnh, phương pháp này bao gồm:

bước rút ra ứng viên vector khối để rút ra các ứng viên vector khối của khối đích trong hình ảnh đích từ thông tin mã hóa được lưu trong bộ nhớ lưu trữ thông tin mã hóa;

bước lựa chọn để lựa chọn vector khối đã lựa chọn được từ các ứng viên vector khối;

bước hiệu chỉnh vị trí tham chiếu để thực hiện hiệu chỉnh liên quan đến khối tham chiếu sẽ được đề cập tới bởi vector khối đã được lựa chọn do vậy vị trí tham chiếu của khối tham chiếu sẽ được hiệu chỉnh để đề cập tới phía bên trong của vùng có thể tham chiếu; và

bước dự đoán để thu mẫu đã được giải mã trong hình ảnh đích thu được từ bộ nhớ hình ảnh đã được giải mã dưới dạng giá trị dự đoán của khối đích dựa trên vị trí tham chiếu của khối tham chiếu,

trong đó bước hiệu chỉnh vị trí tham chiếu thiết lập số lượng khối chuẩn sao chép khối nội ảnh cố định trải qua quy trình mã hóa ngay trước khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng có thể tham chiếu, và thiết lập khối chuẩn sao chép khối nội ảnh trước vùng có thể tham chiếu và khối chuẩn sao chép khối nội ảnh bao gồm khối đích như là vùng tham chiếu không hợp lệ bất kể quy trình mã hóa có hoàn tất hay không.

FIG.1

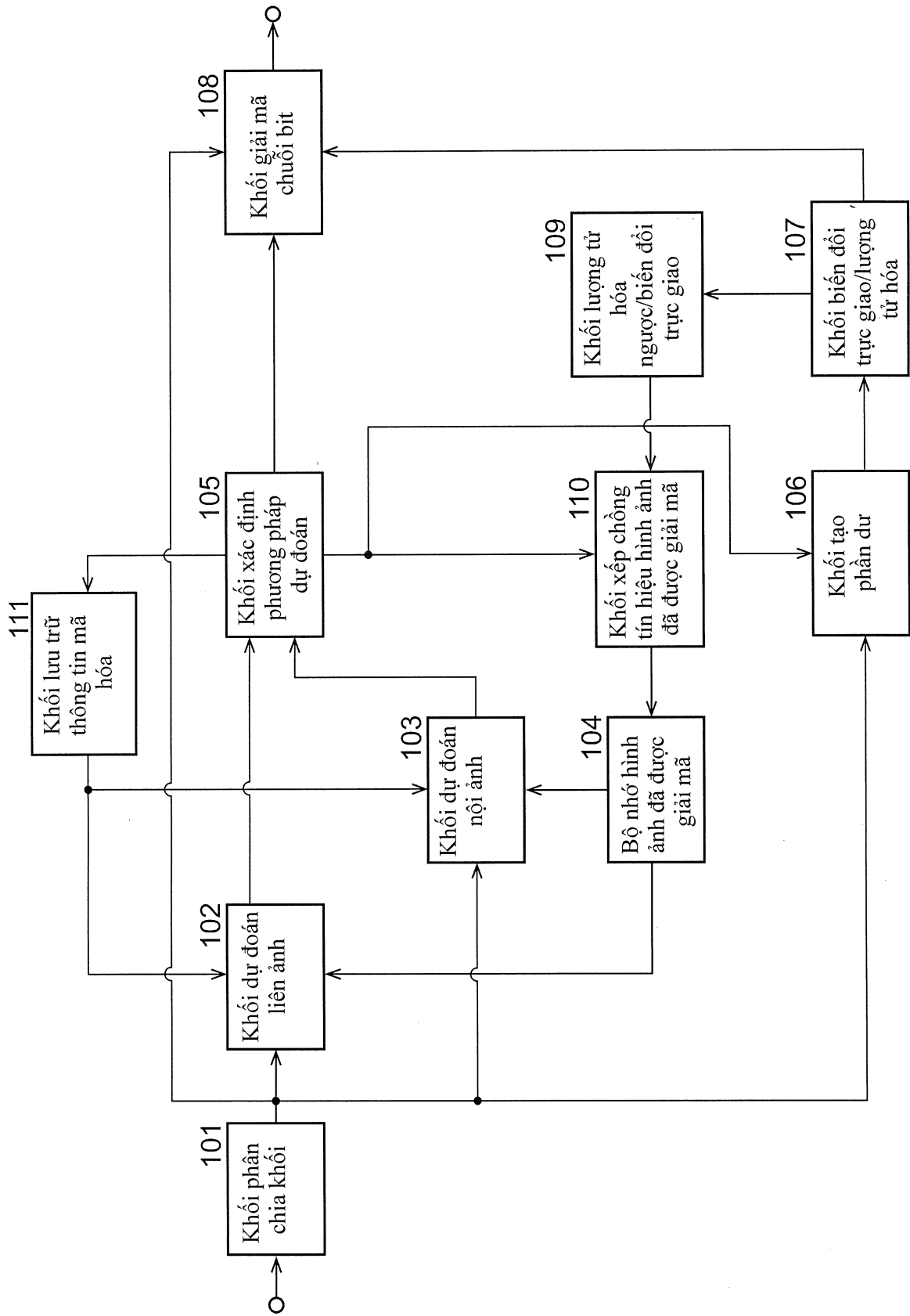


FIG.2

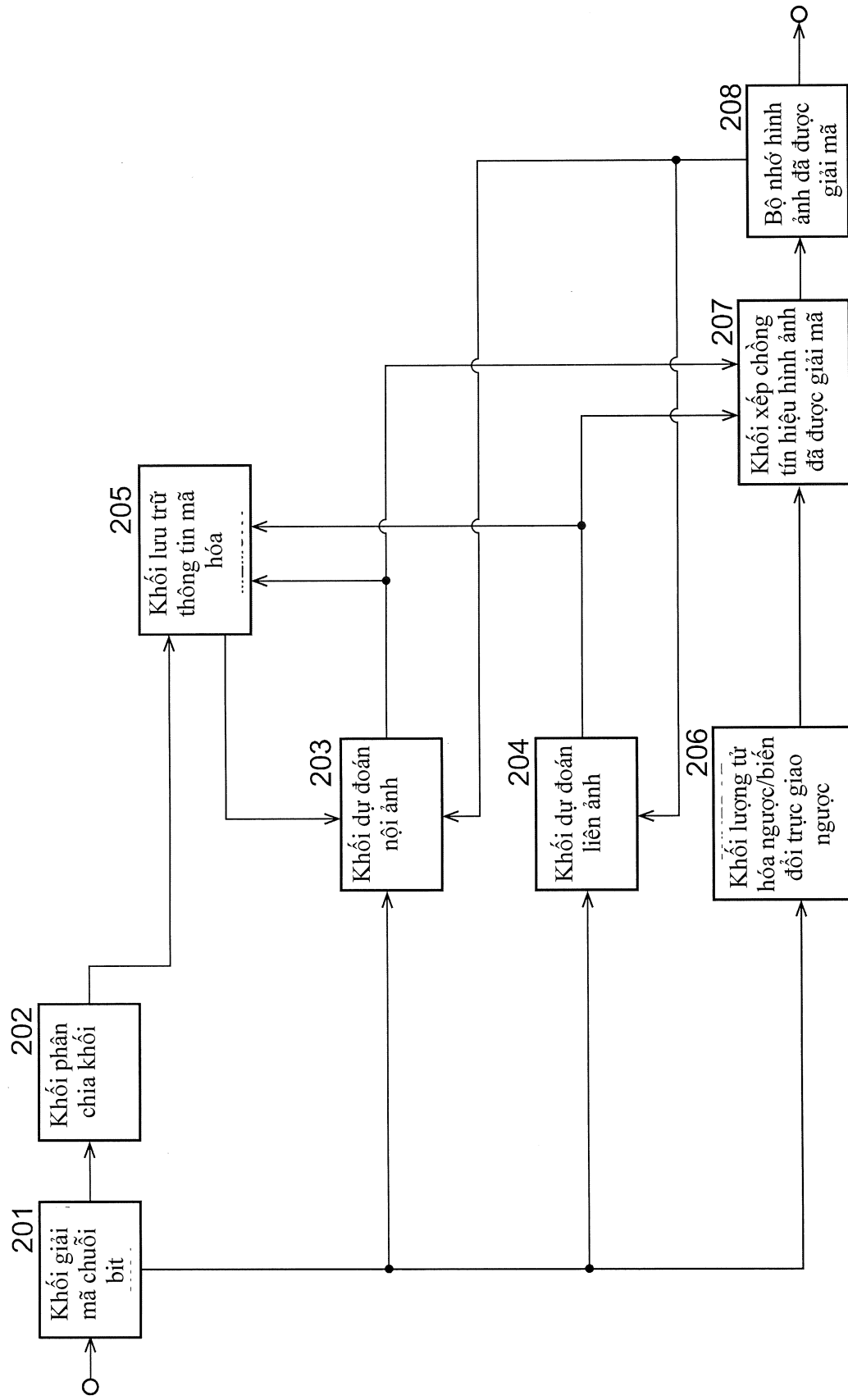


FIG.3

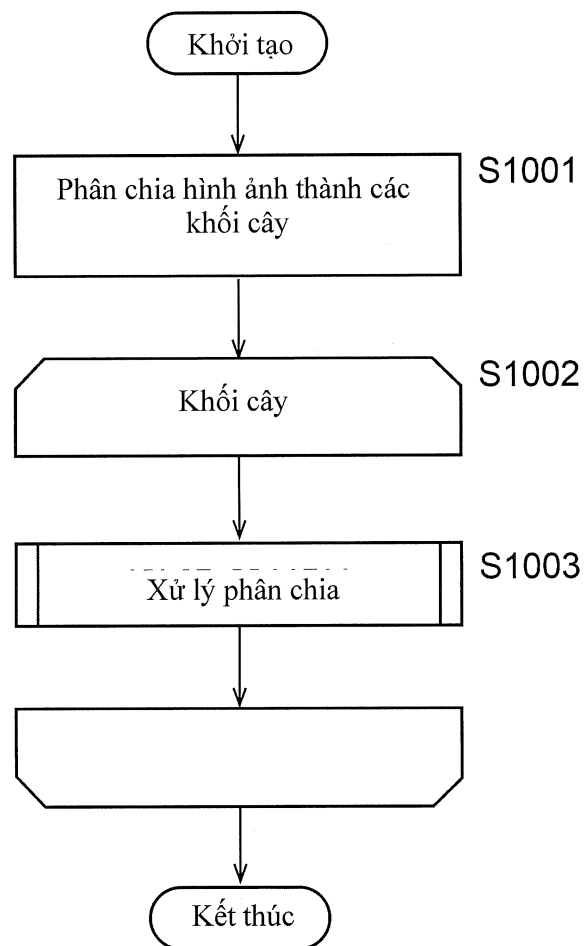


FIG.4

Khối cây
(128x128)

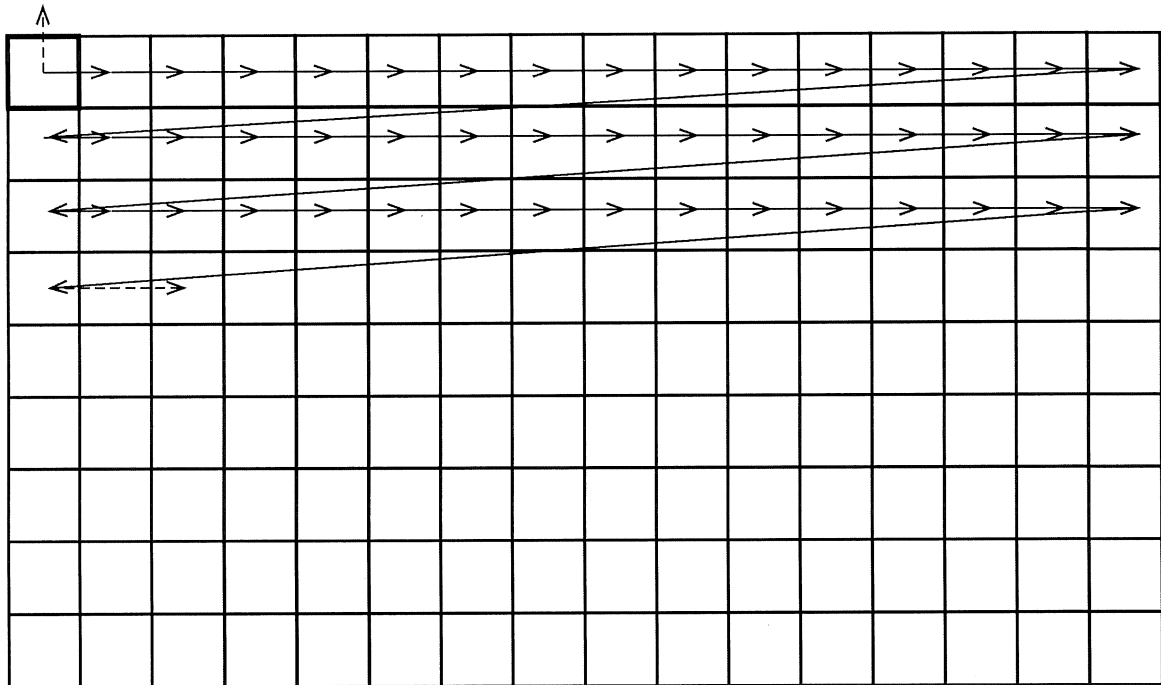


FIG.5

Khối cây
(128x128)

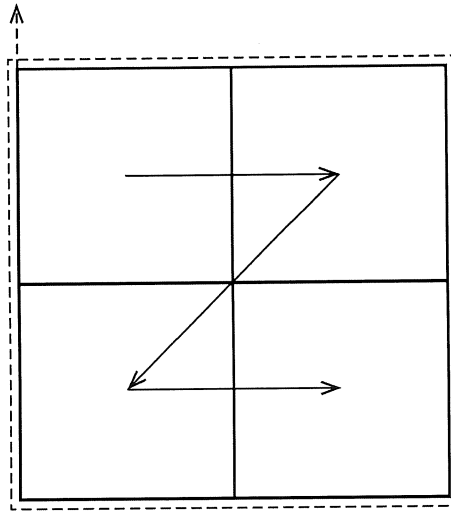


FIG.6A

601

0		1	
2		3	

FIG.6B

602

0			
1			

FIG.6C

603

0			
1			
2			

FIG.6D

604

0		1	

FIG.6E

605

0	1		2

FIG.7

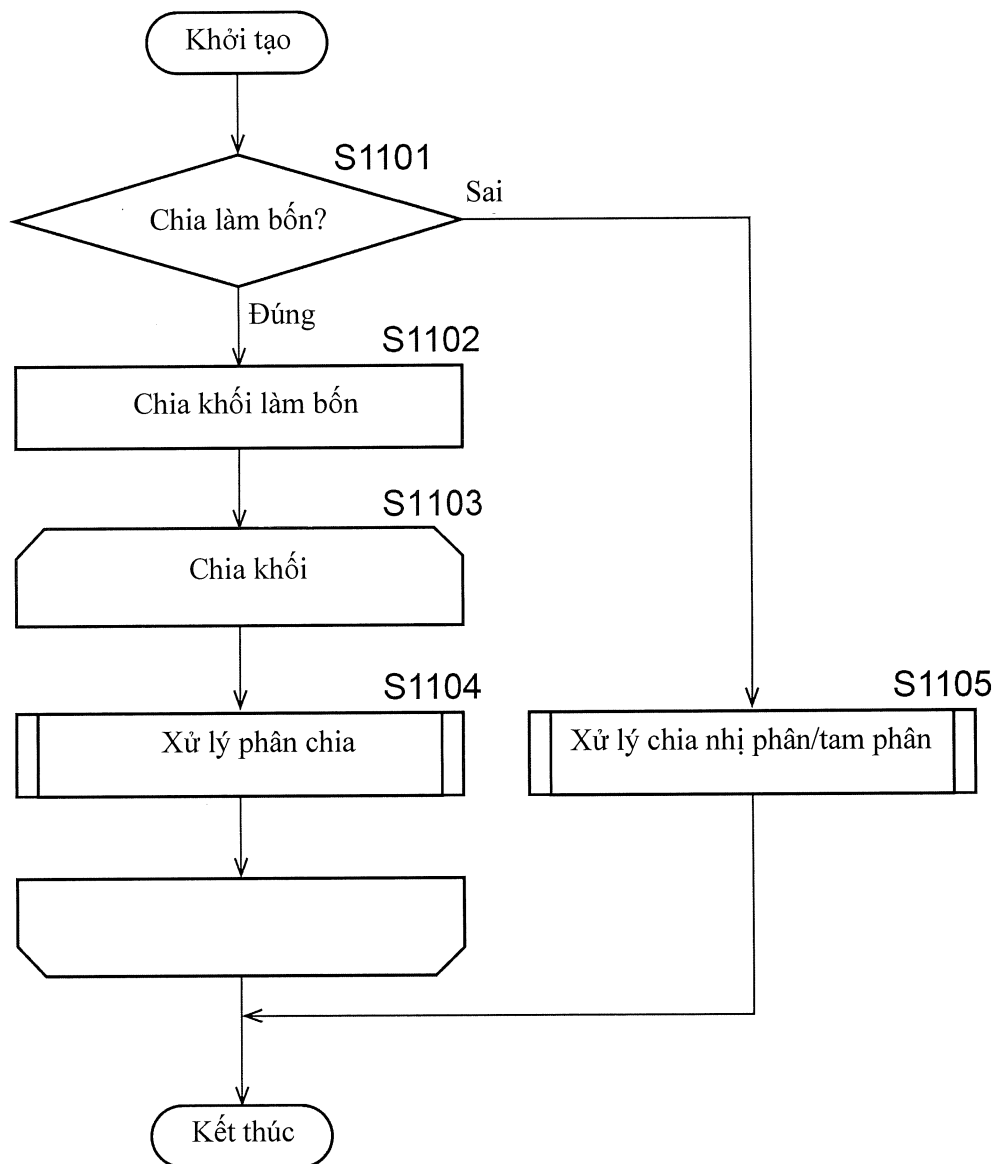


FIG.8

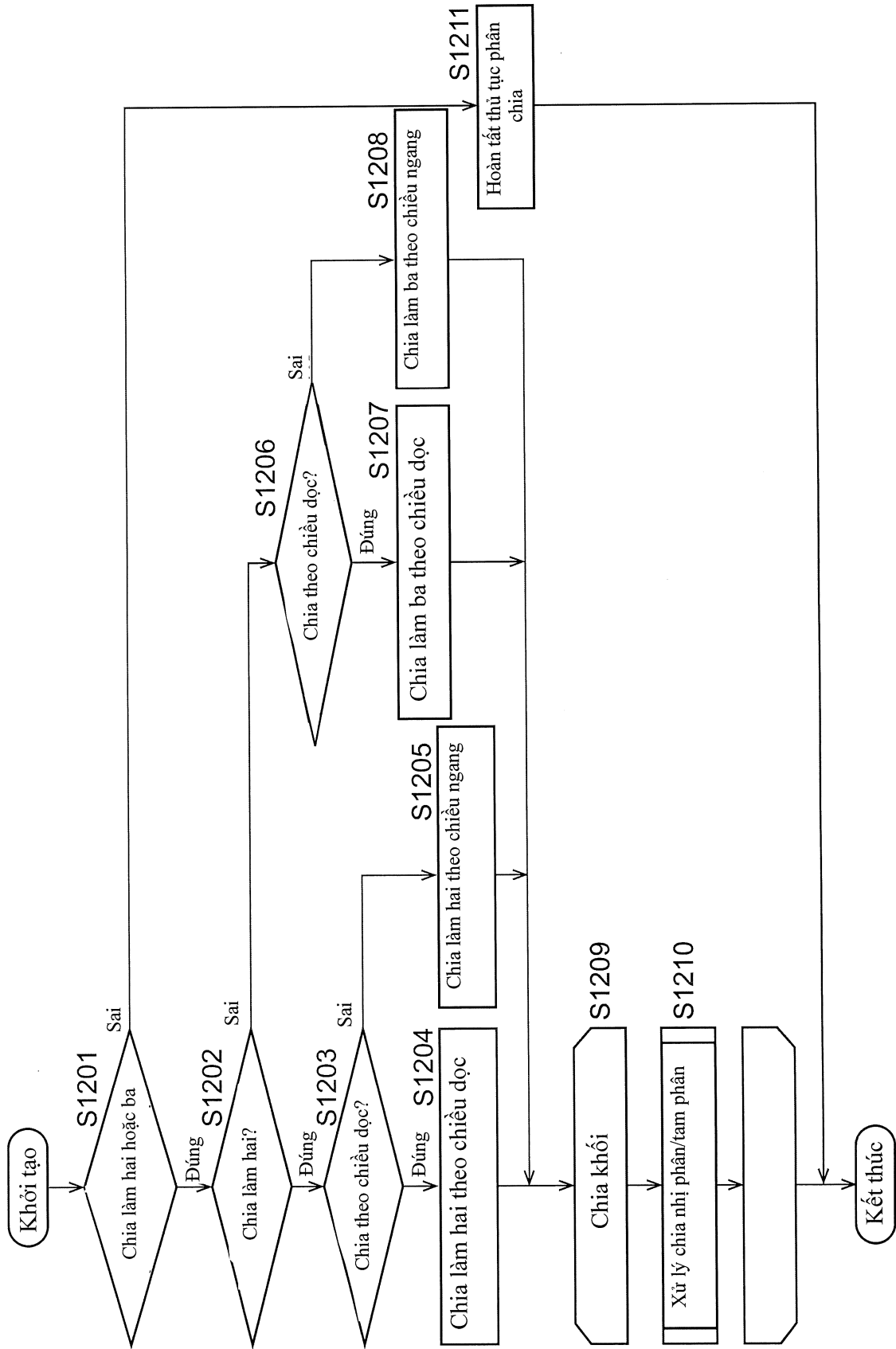


FIG.9

```
coding_quadtree() {  
    qt_split  
    if(qt_split) {  
        coding_quadtree(0)  
        coding_quadtree(1)  
        coding_quadtree(2)  
        coding_quadtree(3)  
    }  
    else {  
        multi_type_tree()  
    }  
}
```

```
multi_type_tree() {  
    mtt_split  
    if(mtt_split) {  
        mtt_split_vertical  
        mtt_split_binary  
        if(mtt_split_binary) {  
            multi_type_tree(0, mtt_split_vertical)  
            multi_type_tree(1, mtt_split_vertical)  
        } else {  
            multi_type_tree(0, mtt_split_vertical)  
            multi_type_tree(1, mtt_split_vertical)  
            multi_type_tree(2, mtt_split_vertical)  
        }  
    } else {  
        // end split  
    }  
}
```

FIG.10A

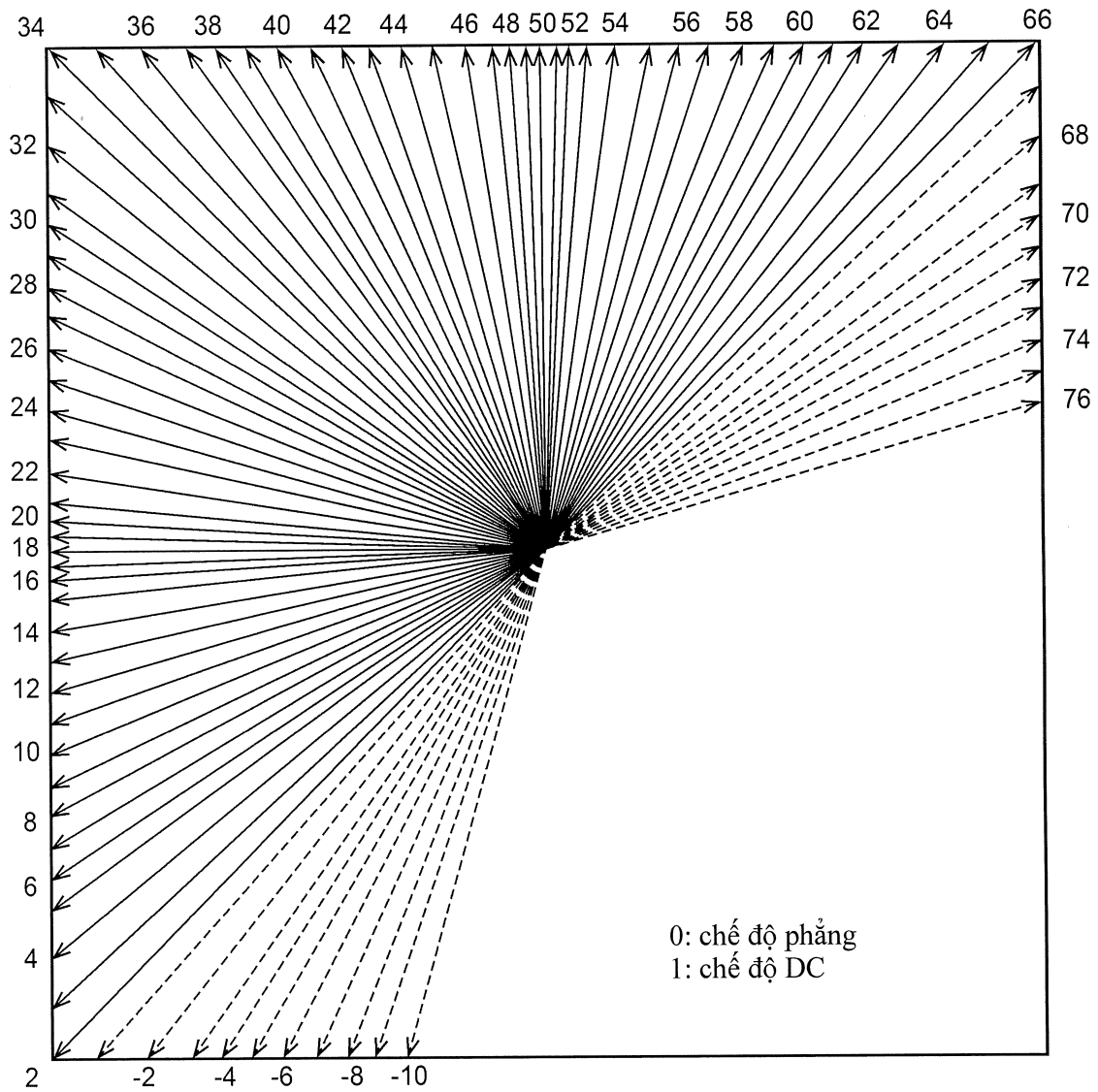


FIG.10B

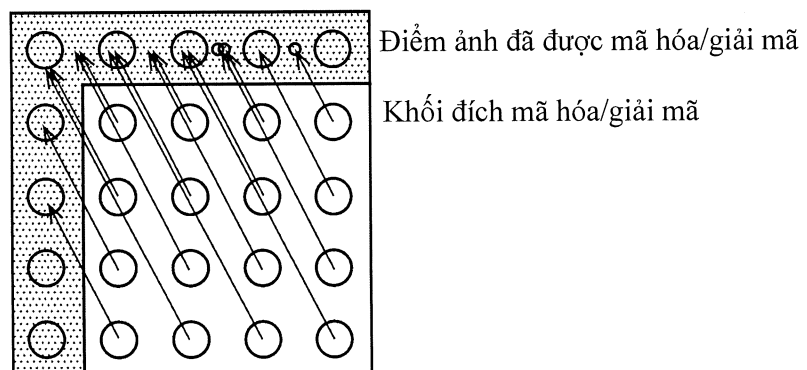


FIG.11

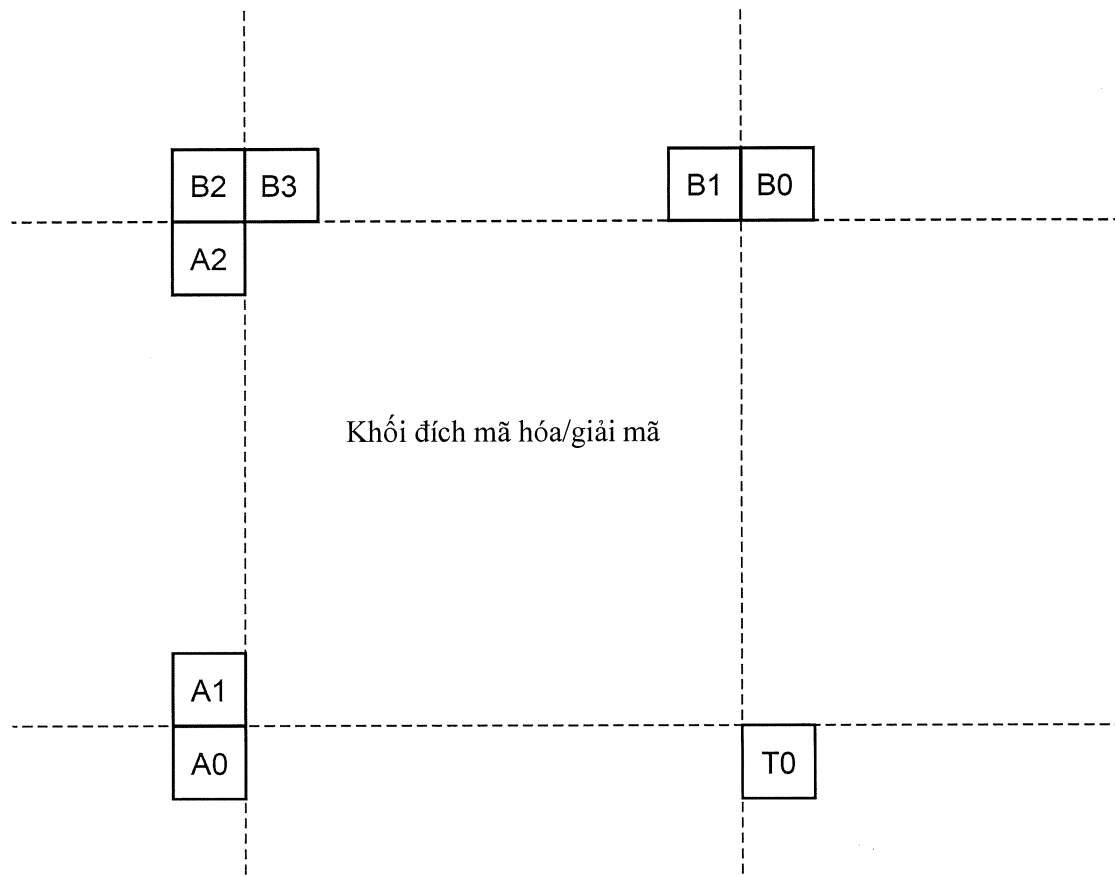


FIG.12A

```

coding_unit() {
    pred_mode_flag
    if( MODE_INTRA ) {
        intra_pred_mode
    }
    else { // MODE_INTER
        merge_flag
        if( merge_flag ) {
            merge_data()
        } else {
            inter_affine_flag
            if( inter_affine_flag ) {
                cu_affine_type_flag
            }
        }
    }
}

```

FIG.12B

```

merge_data() {
    merge_subblock_flag
    if( merge_subblock_flag ) {
        merge_subblock_idx
    } else {
        merge_triangle_flag
        if( merge_triangle_flag ) {
            merge_triangle_split_dir
            merge_triangle_idx0
            merge_triangle_idx1
        }
        else {
            merge_idx
        }
    }
}

```

FIG.13

merge_flag	inter_affine_flag	merge_subblock_flag	merge_triangle_flag	Chế độ được lựa chọn
0	0	N/A	N/A	Chế độ dự đoán liên ảnh
0	1	N/A	N/A	Chế độ affin liên ảnh
1	N/A	0	0	Chế độ hợp nhất
1	N/A	0	0	Chế độ hợp nhất tam giác
1	N/A	1	N/A	Chế độ hợp nhất affin

FIG.14

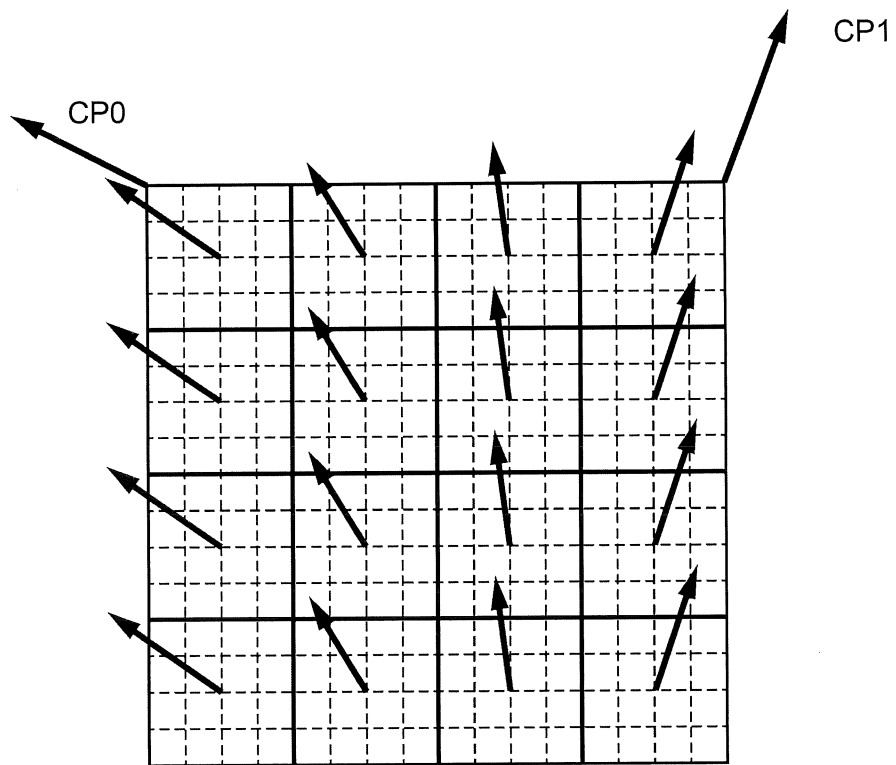


FIG.15

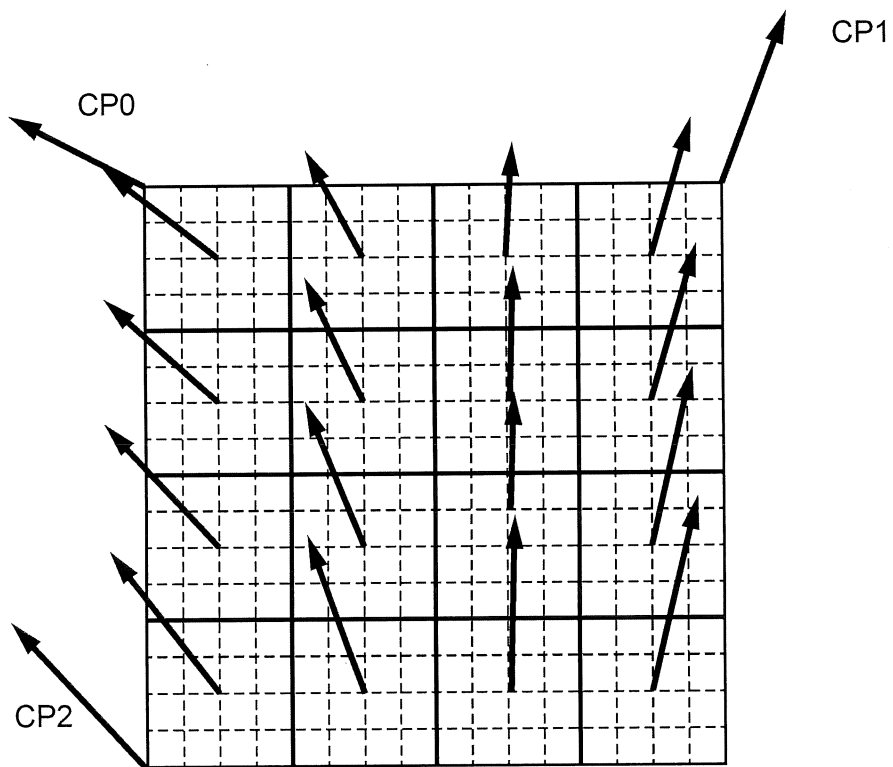


FIG.17

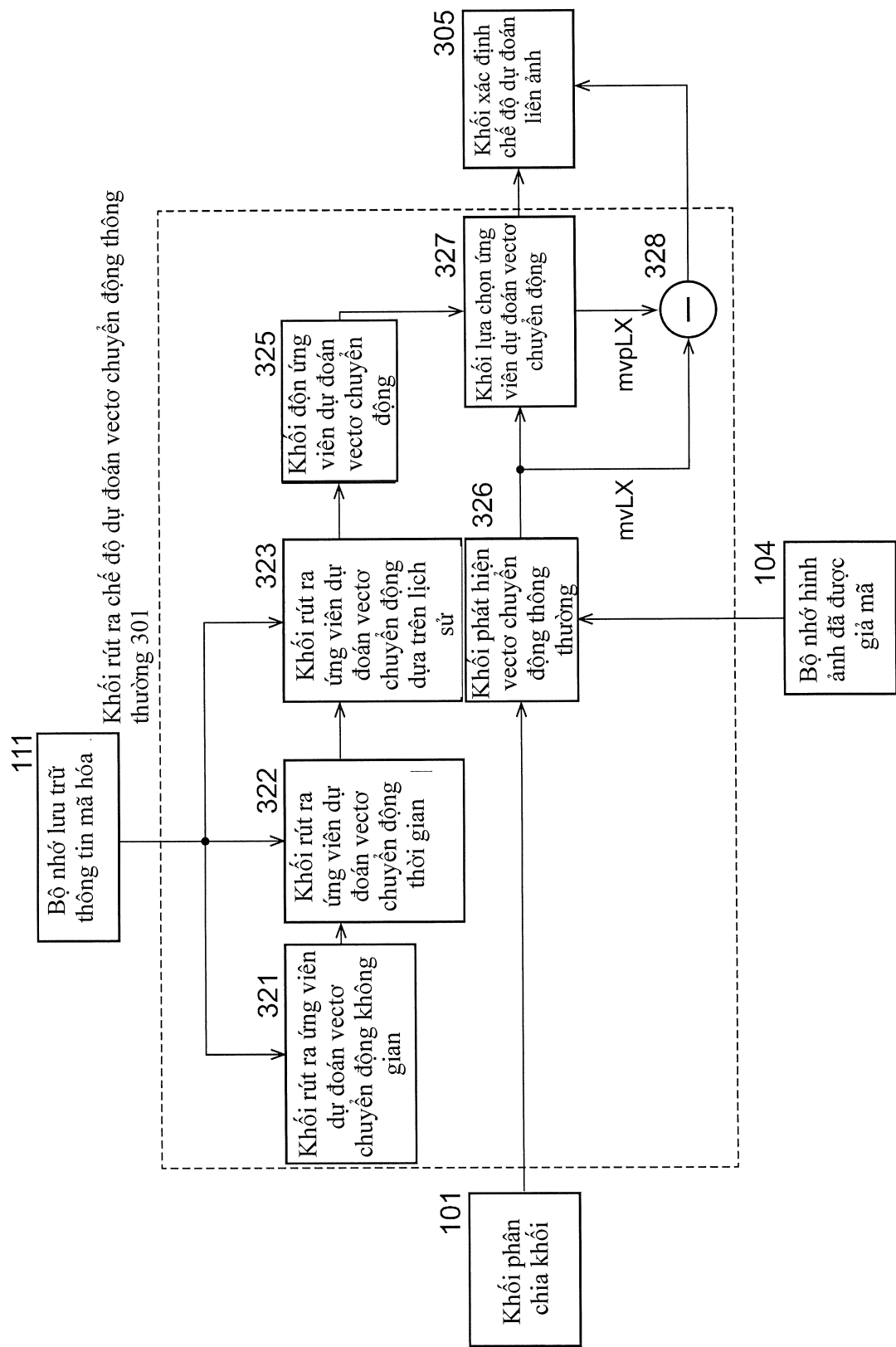


FIG.18

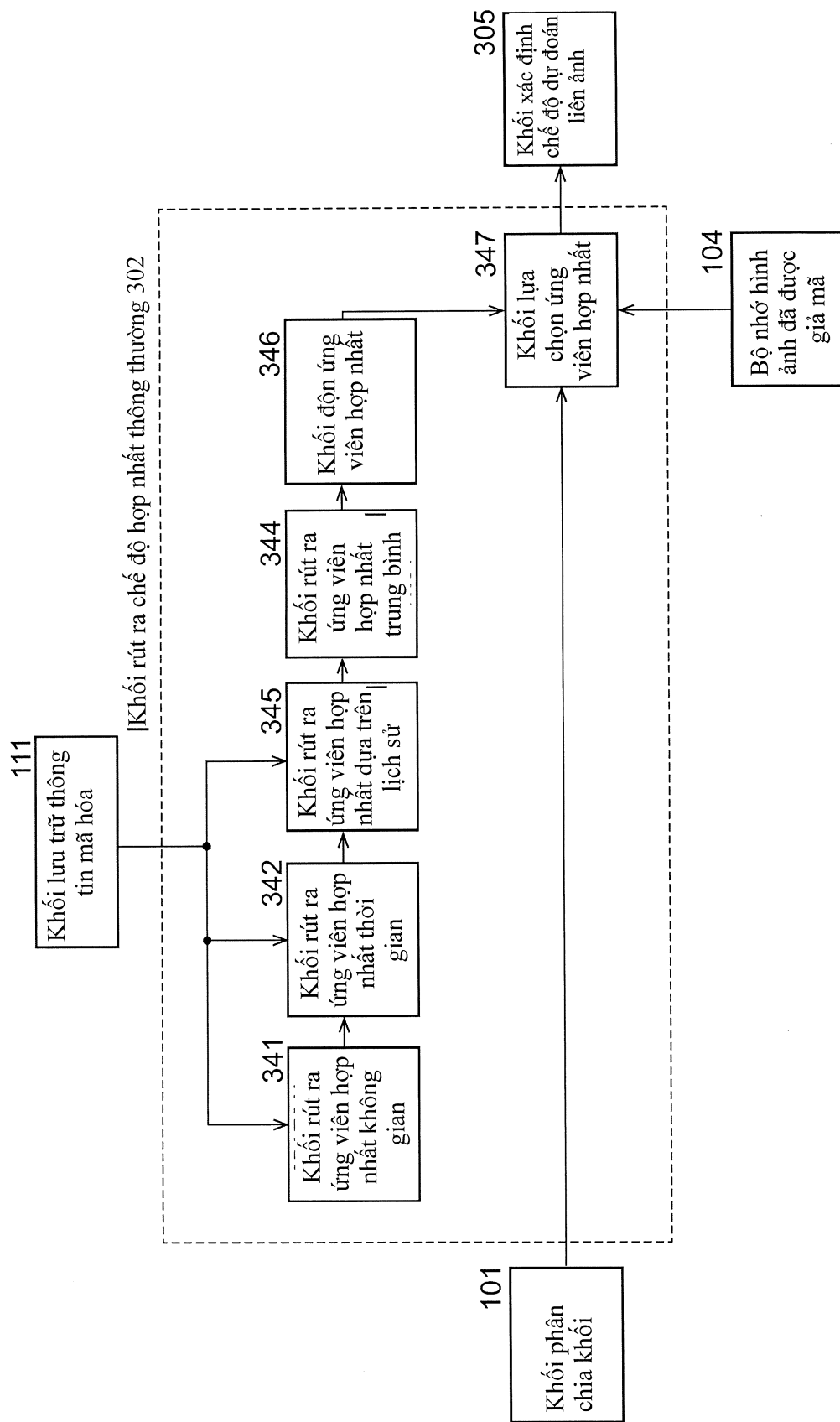


FIG.19

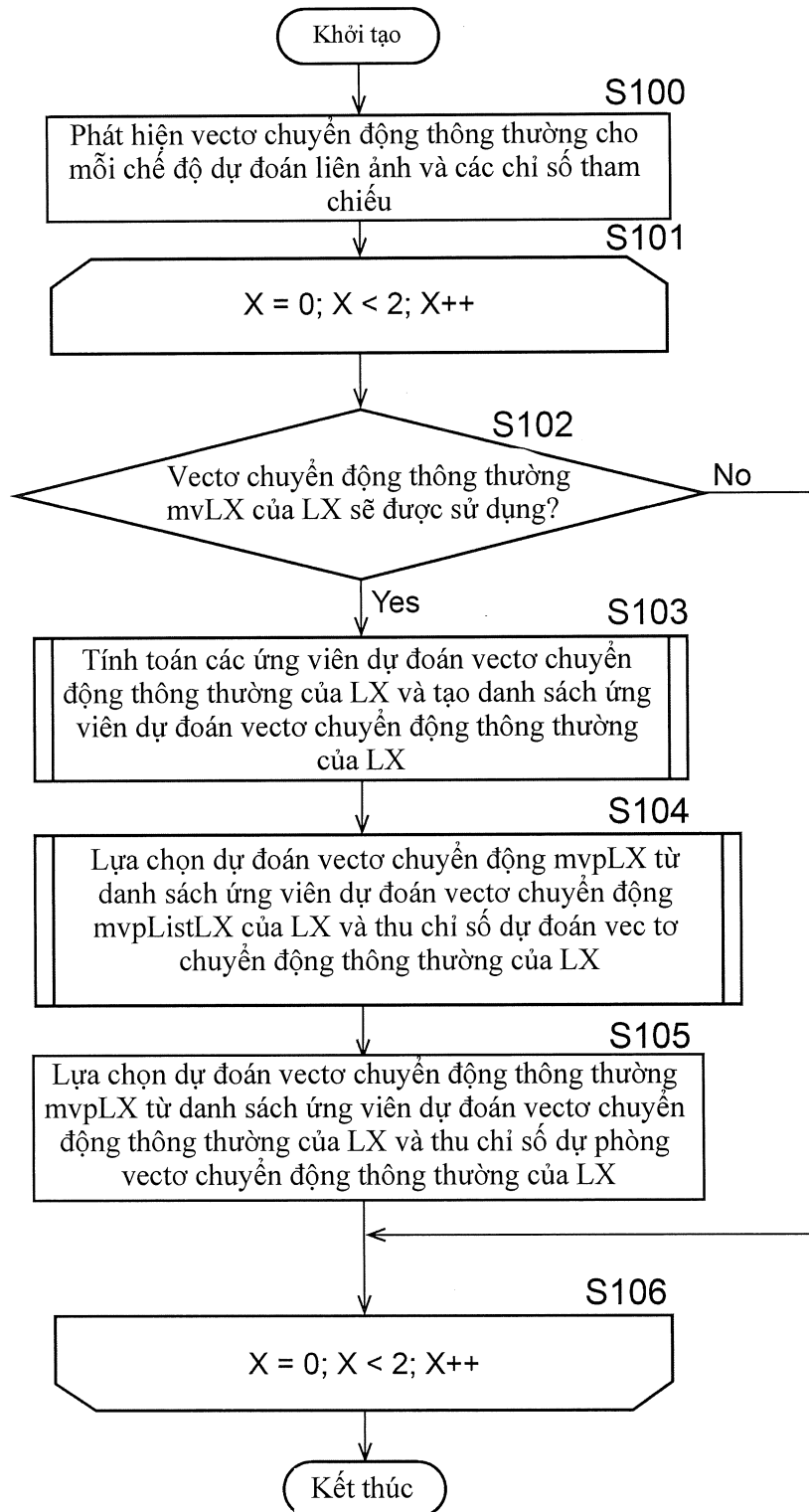


FIG.20

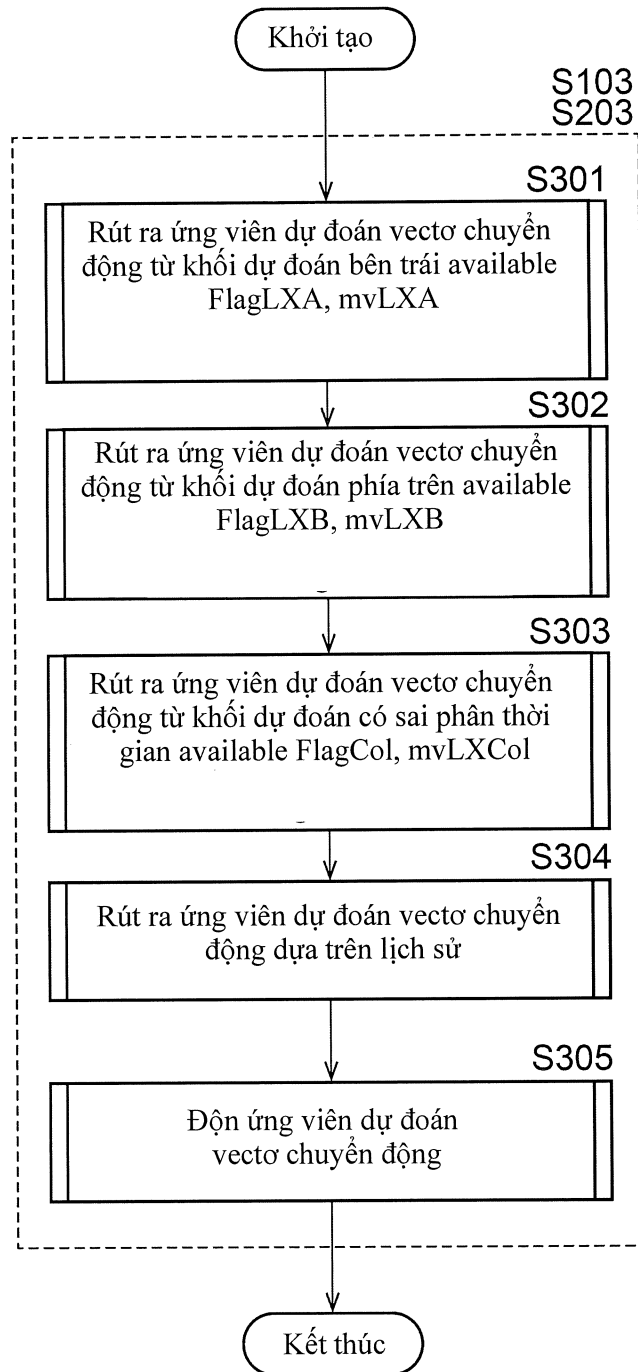


FIG.21

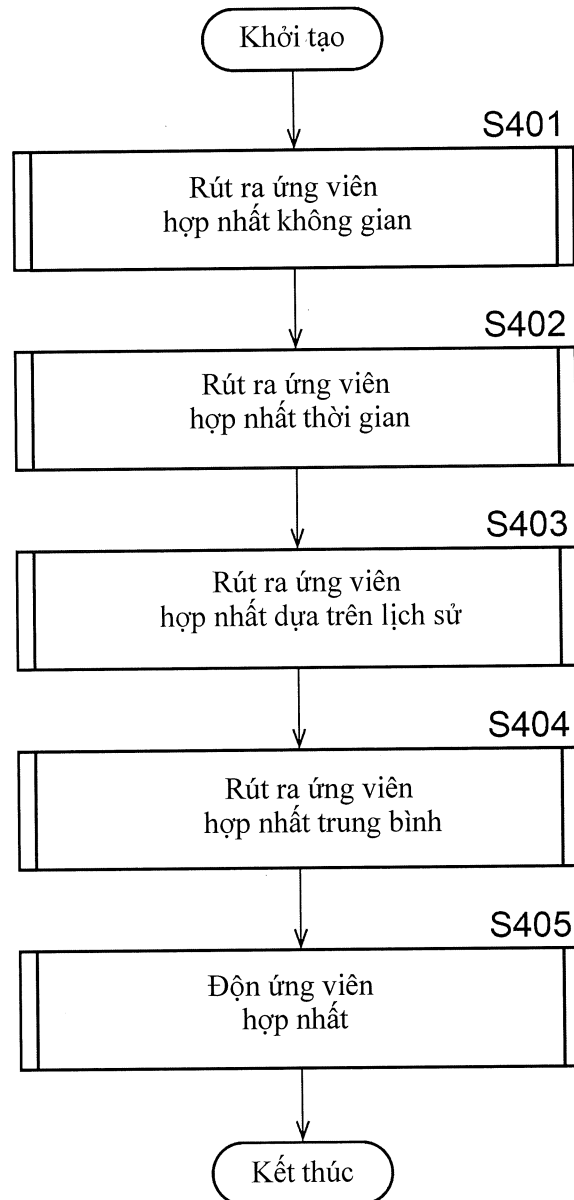


FIG.22

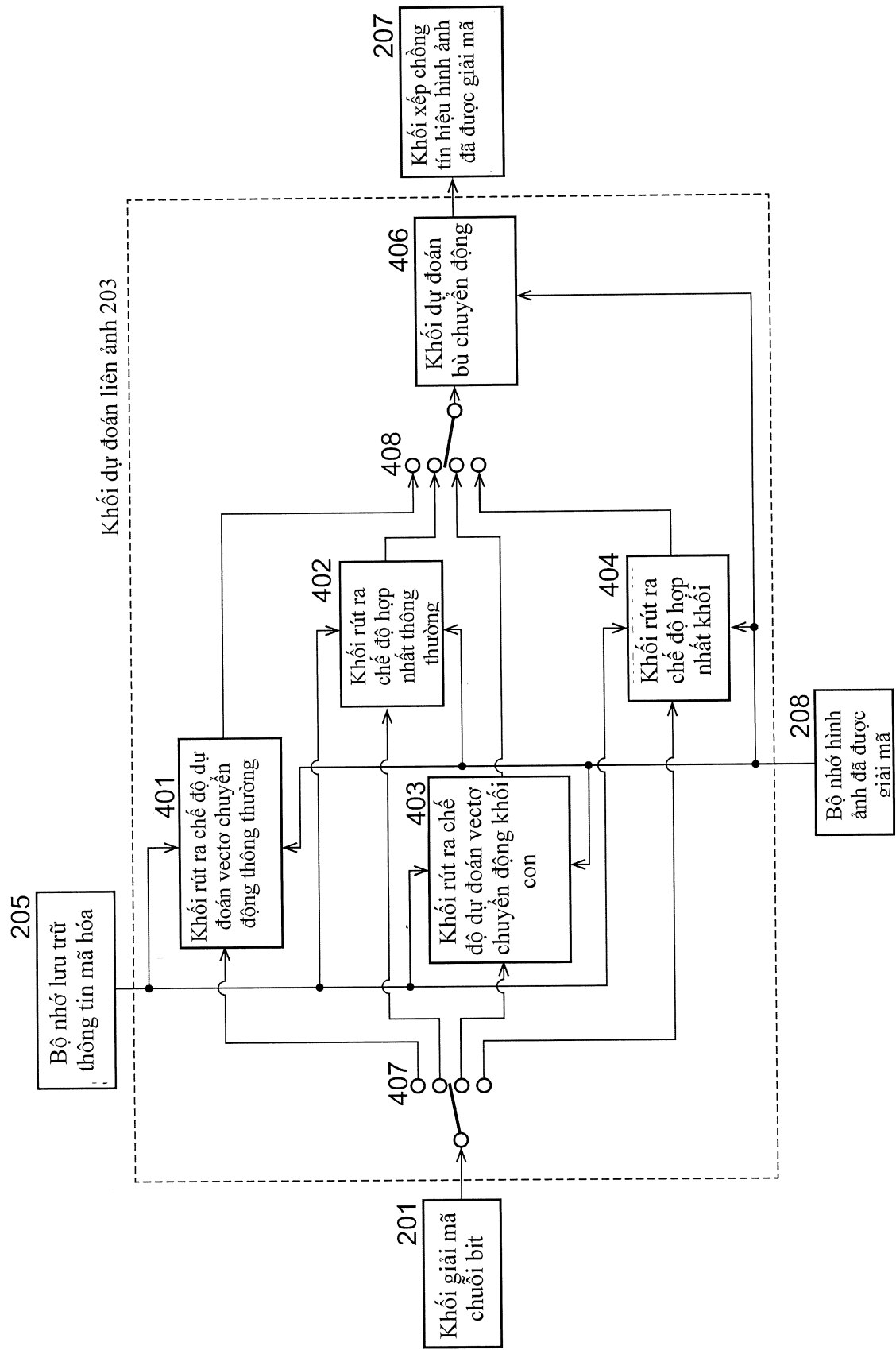


FIG.23

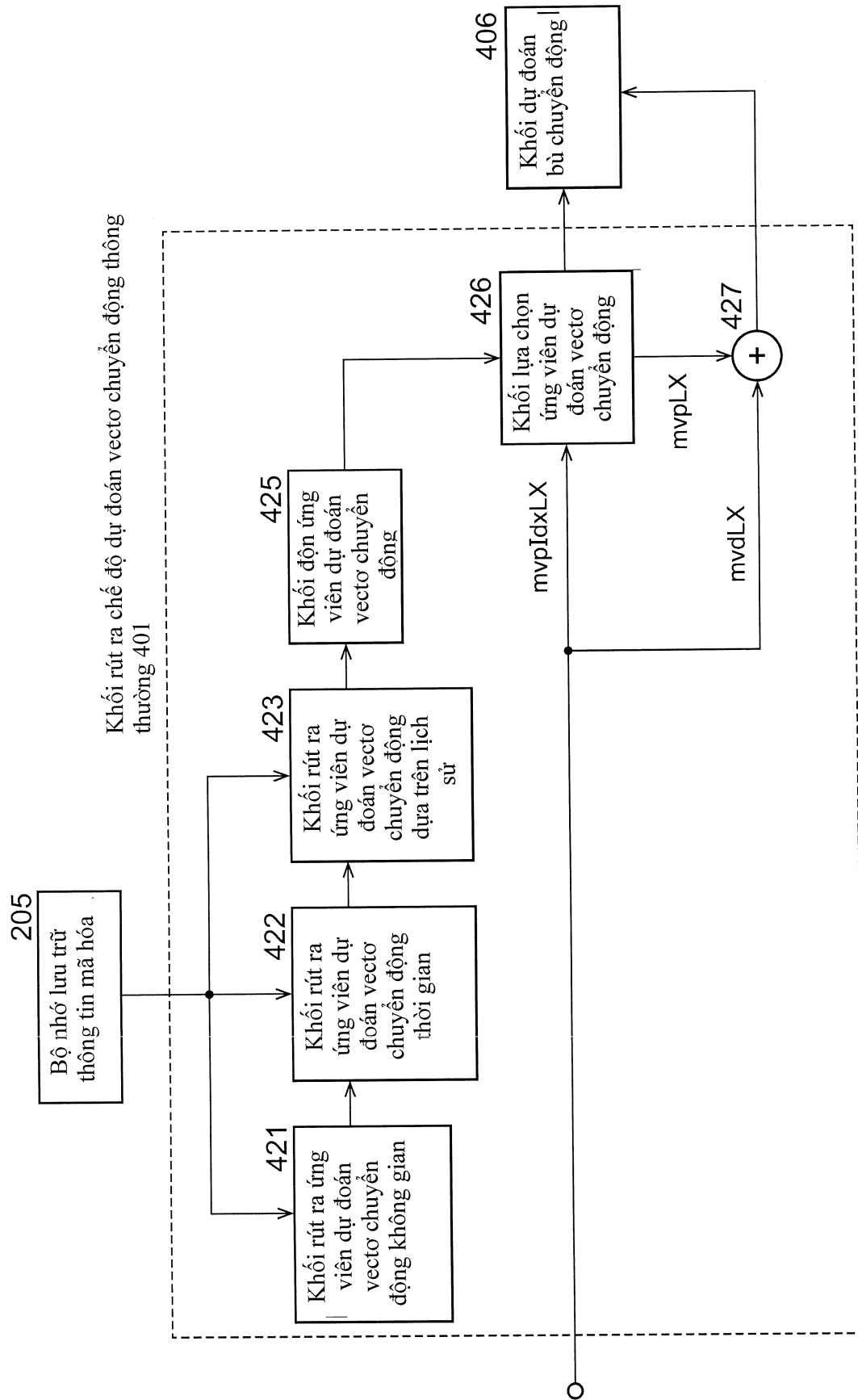


FIG.24

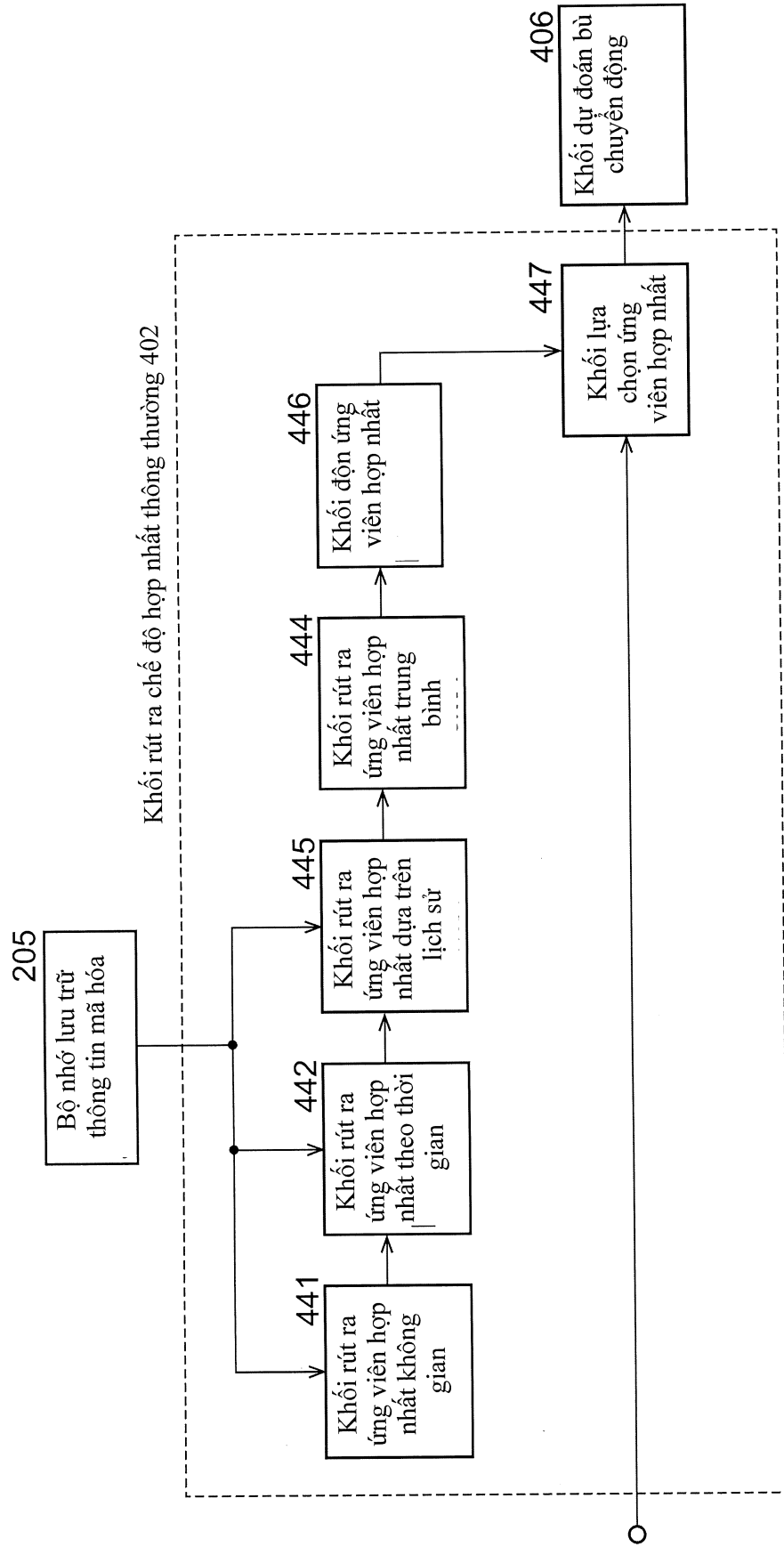


FIG.25

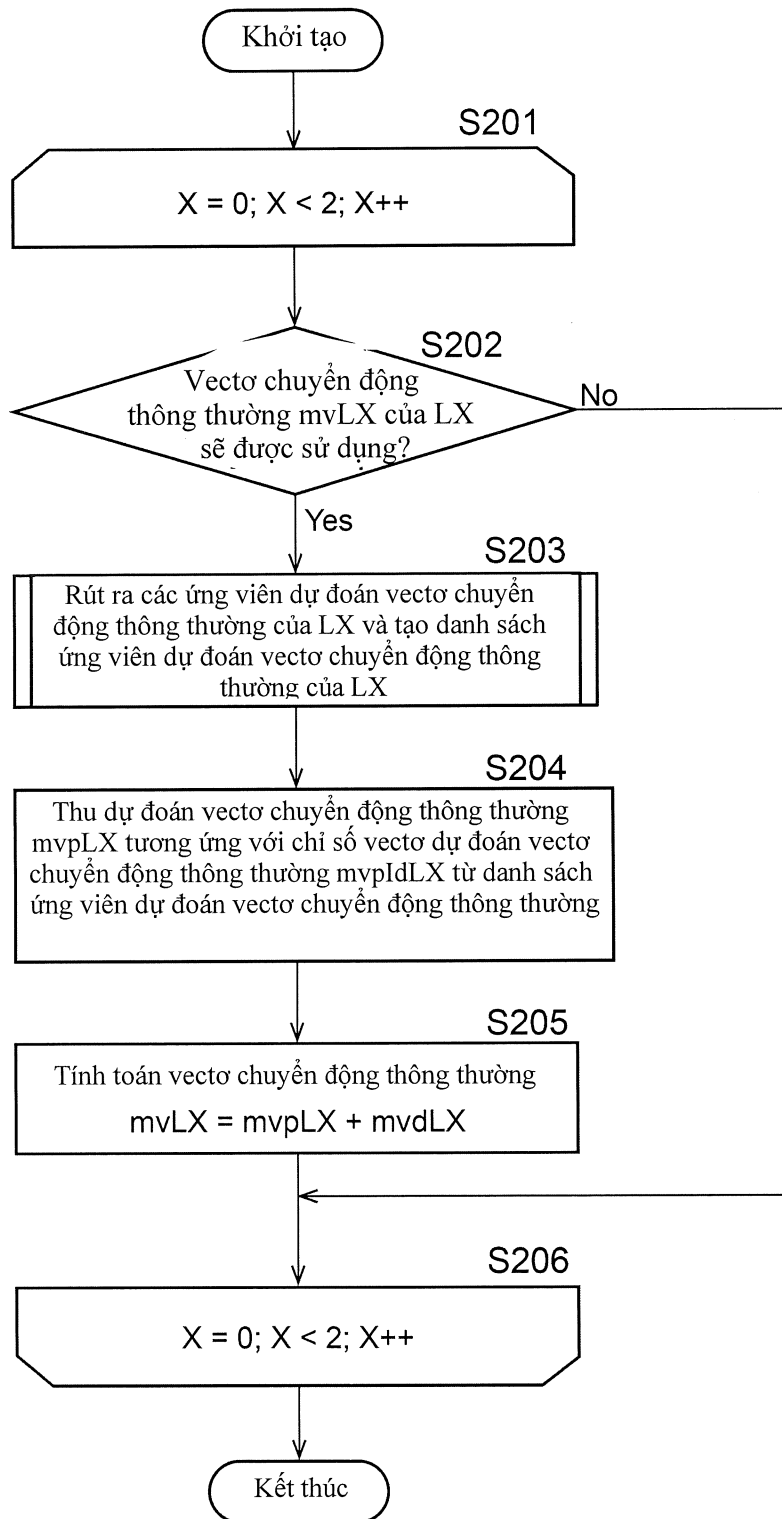


FIG.26

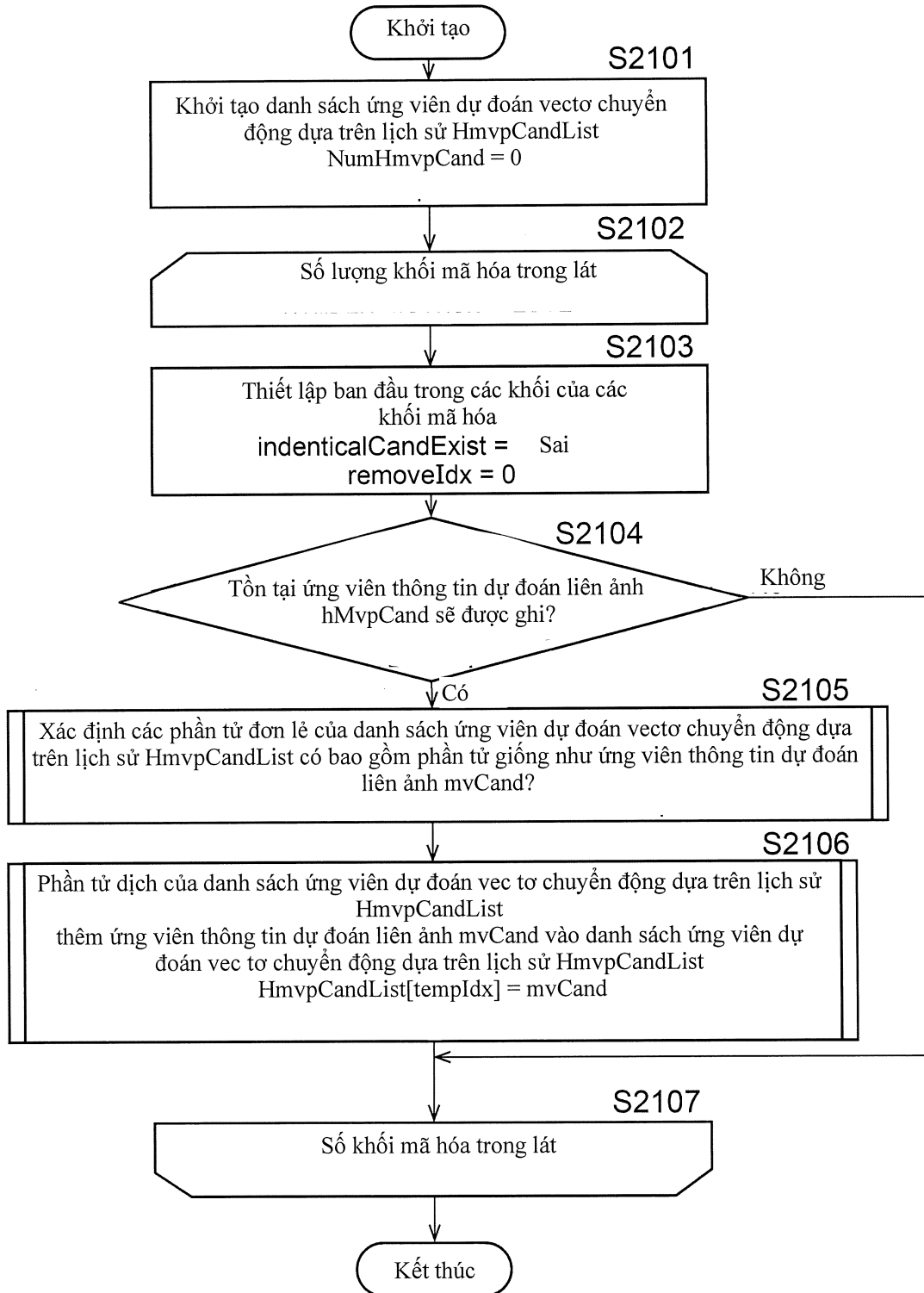


FIG.27

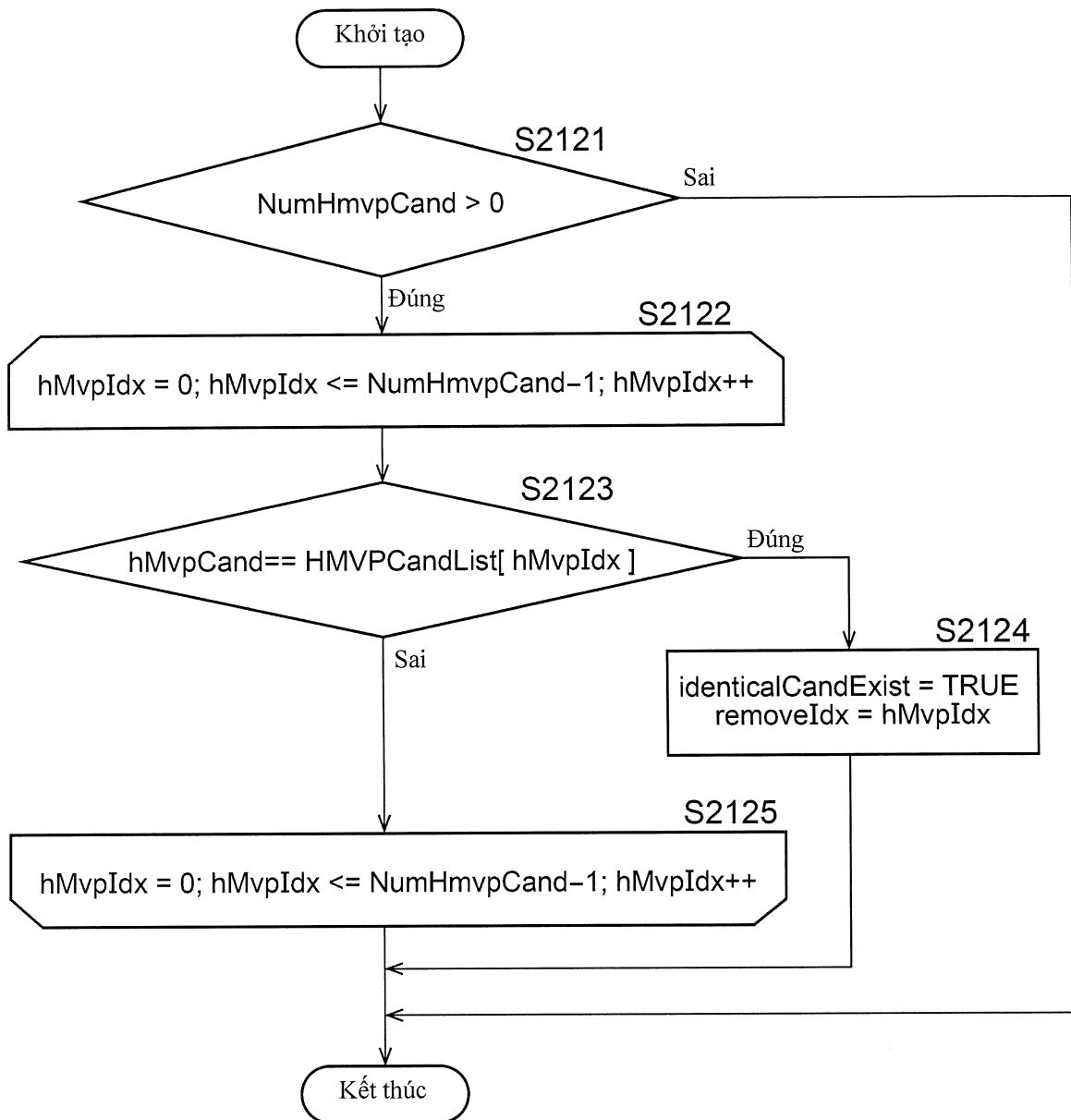


FIG.28

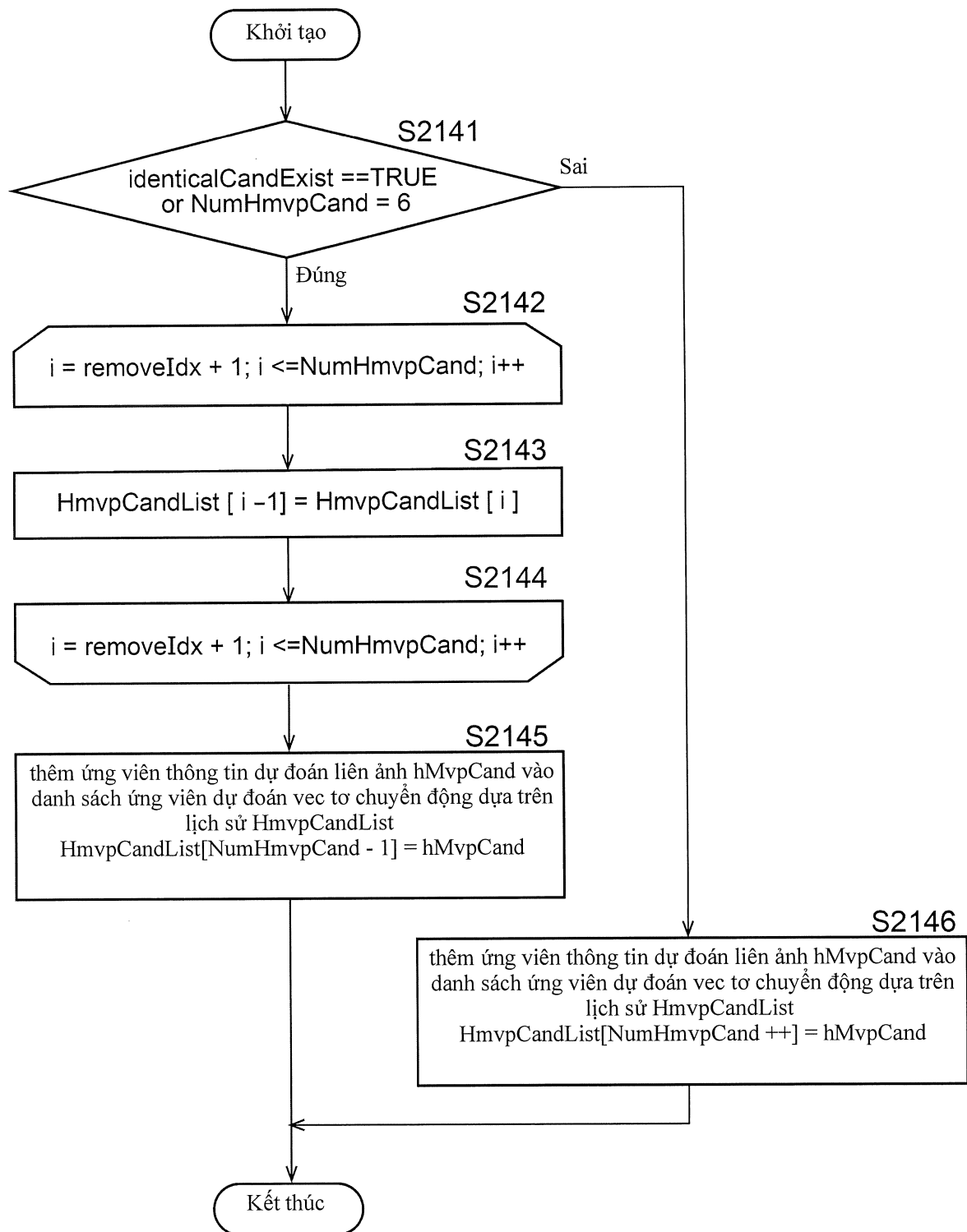


FIG.29

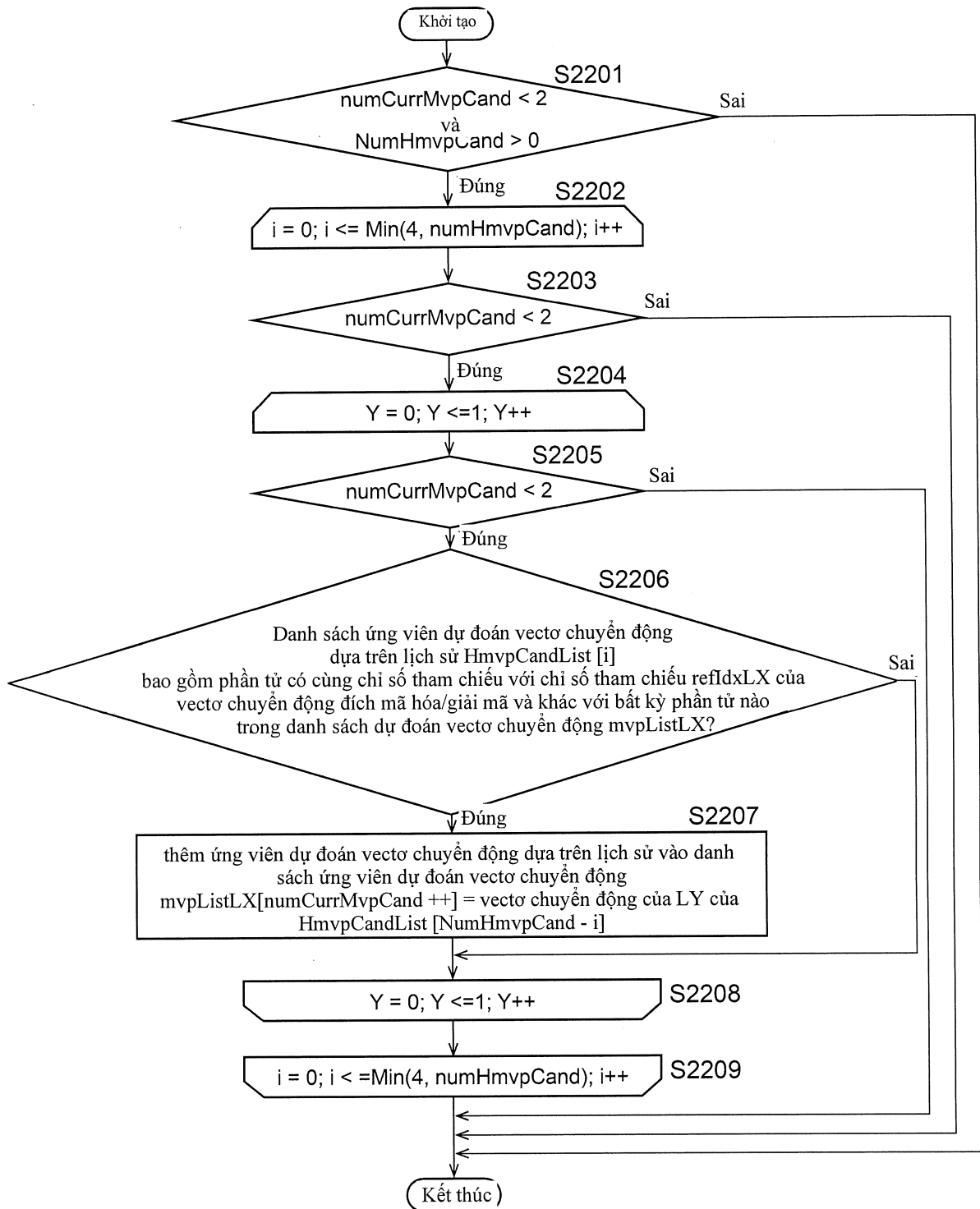


FIG.30

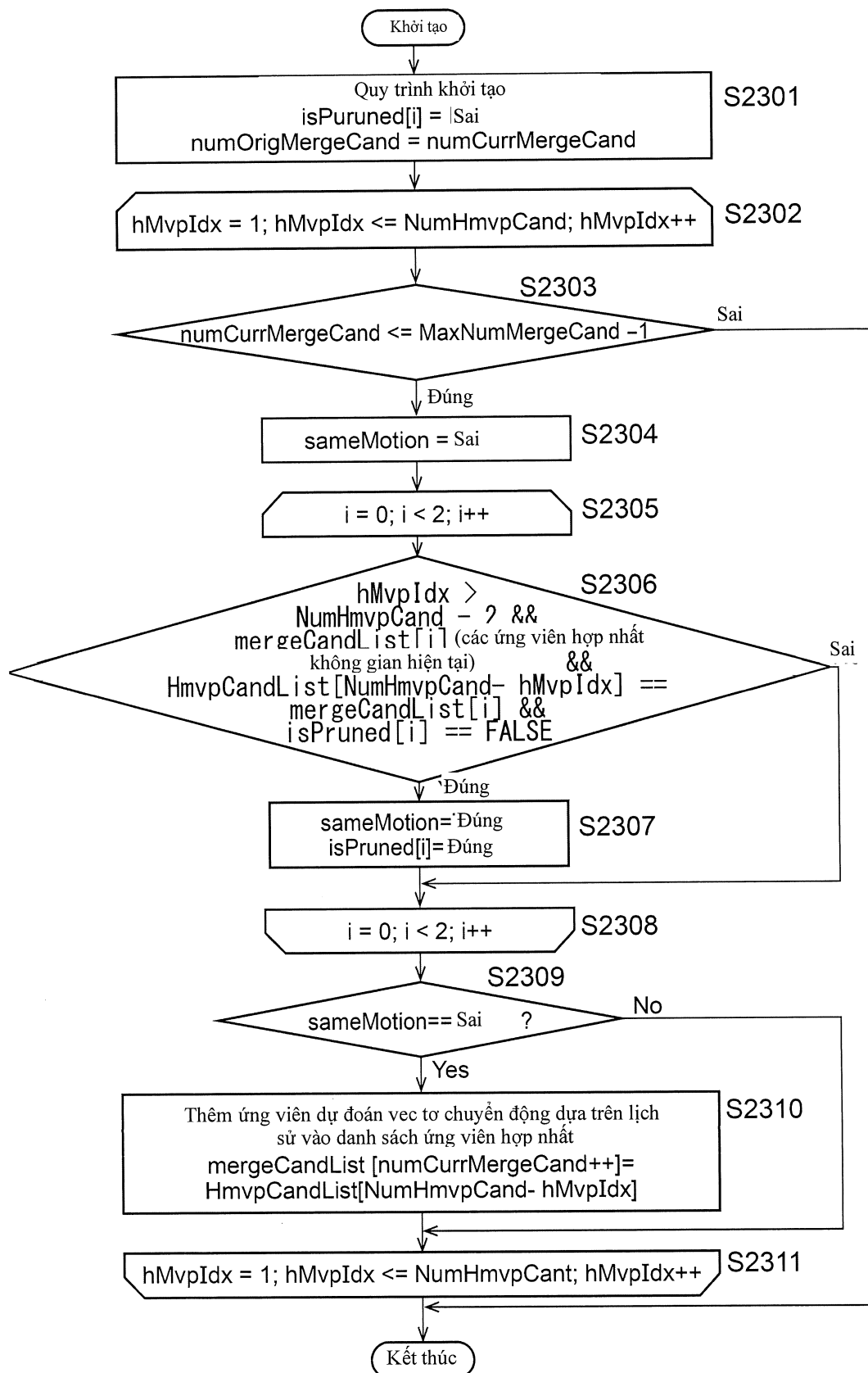


FIG.31A

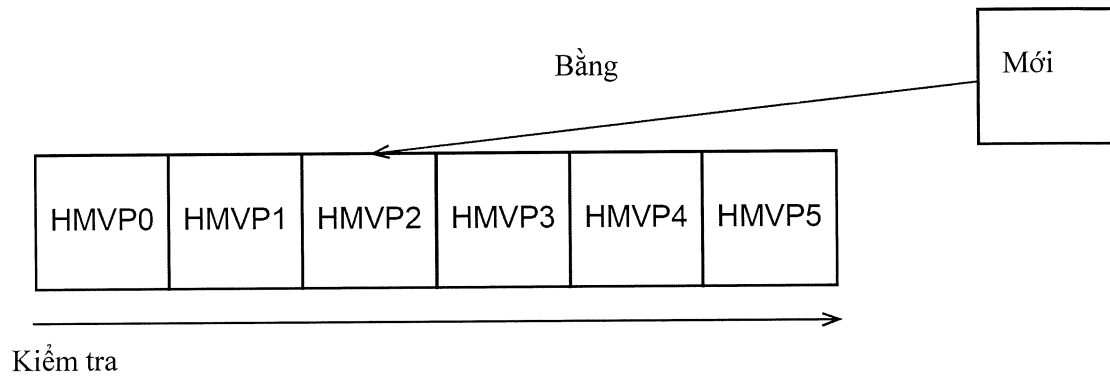


FIG.31B

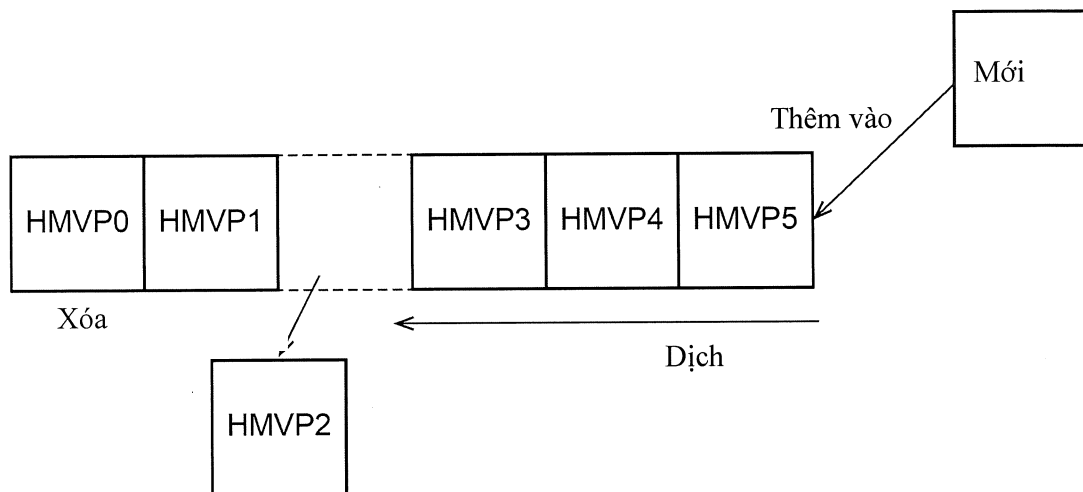


FIG.31C

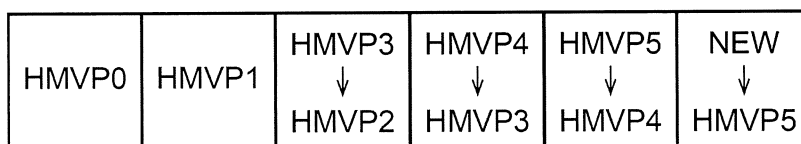


FIG.32

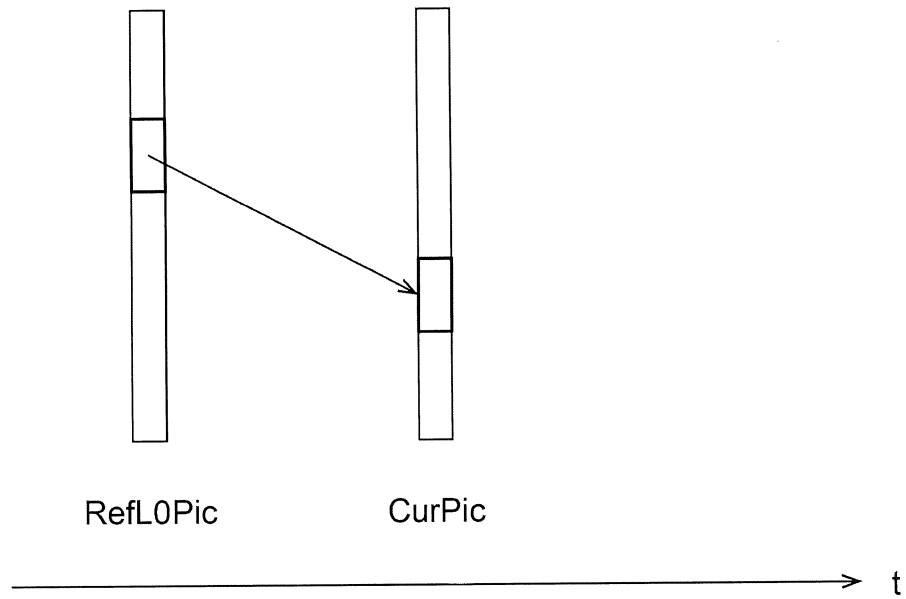


FIG.33

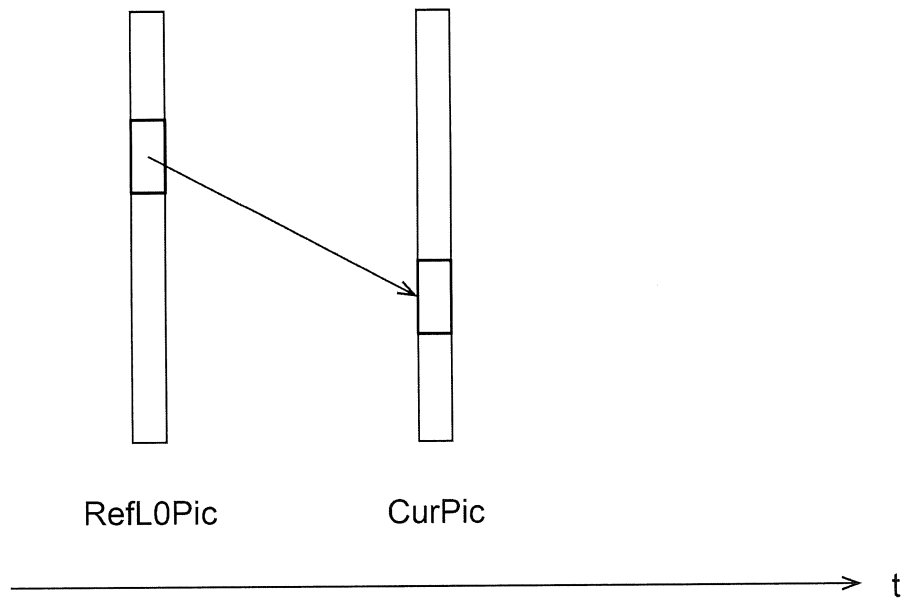


FIG.34

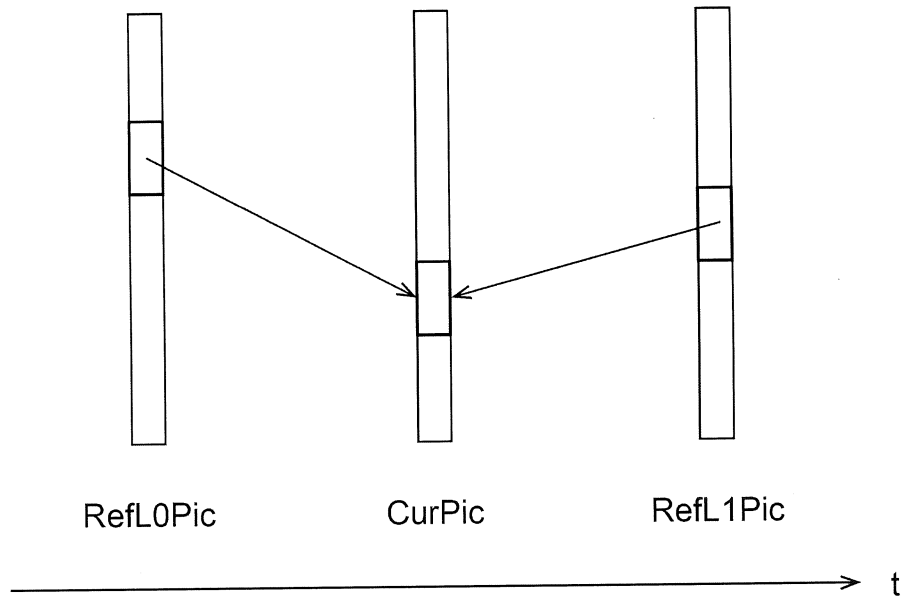


FIG.35

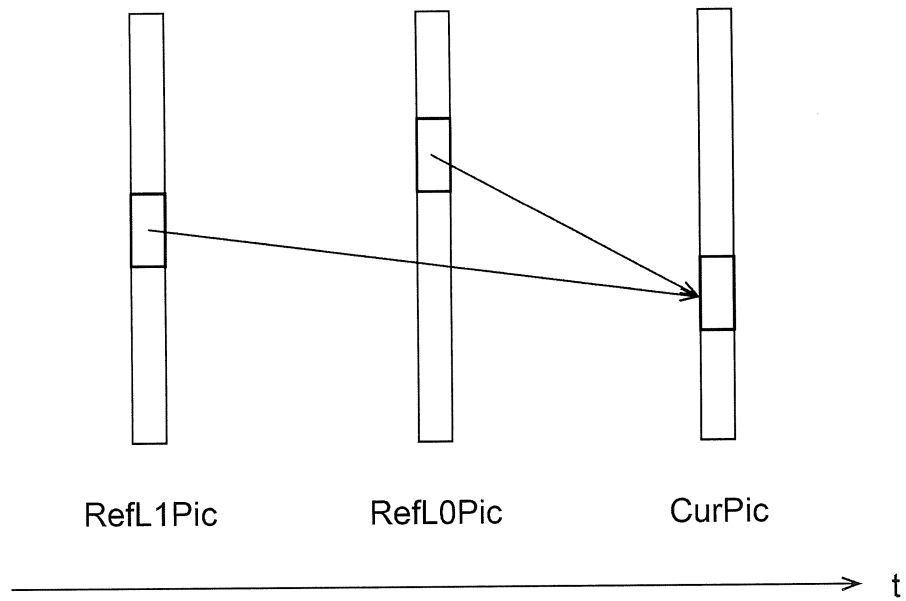


FIG.36

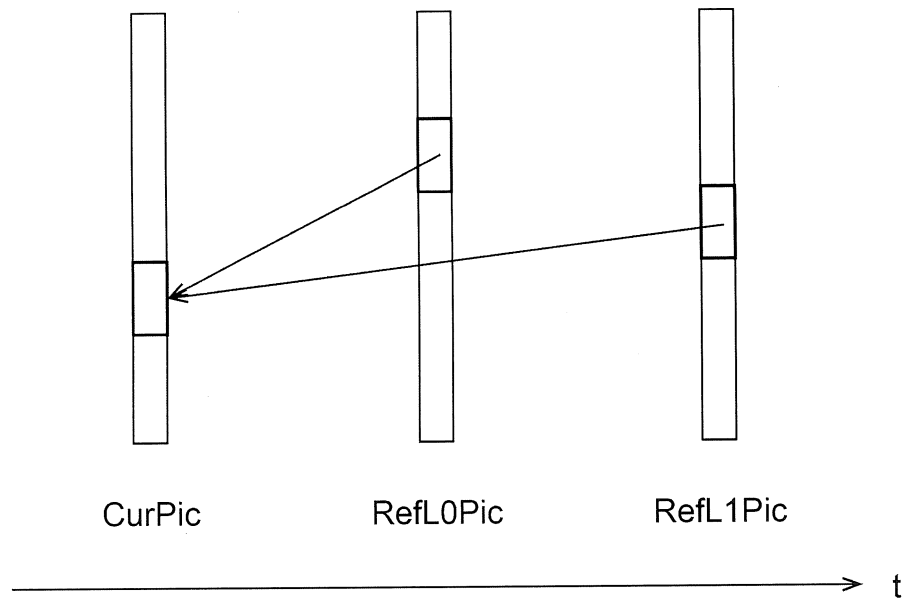


FIG.37

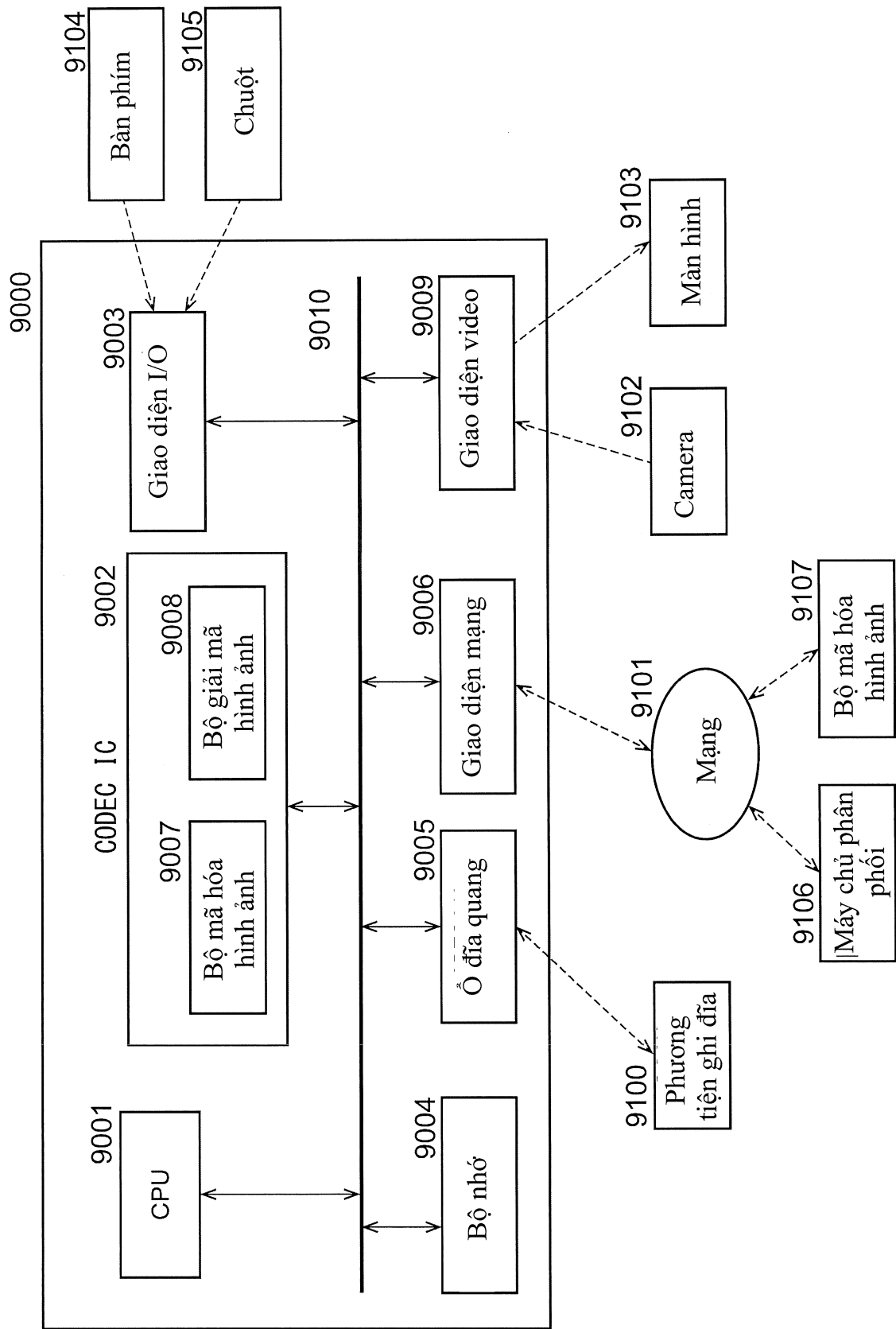


FIG.38

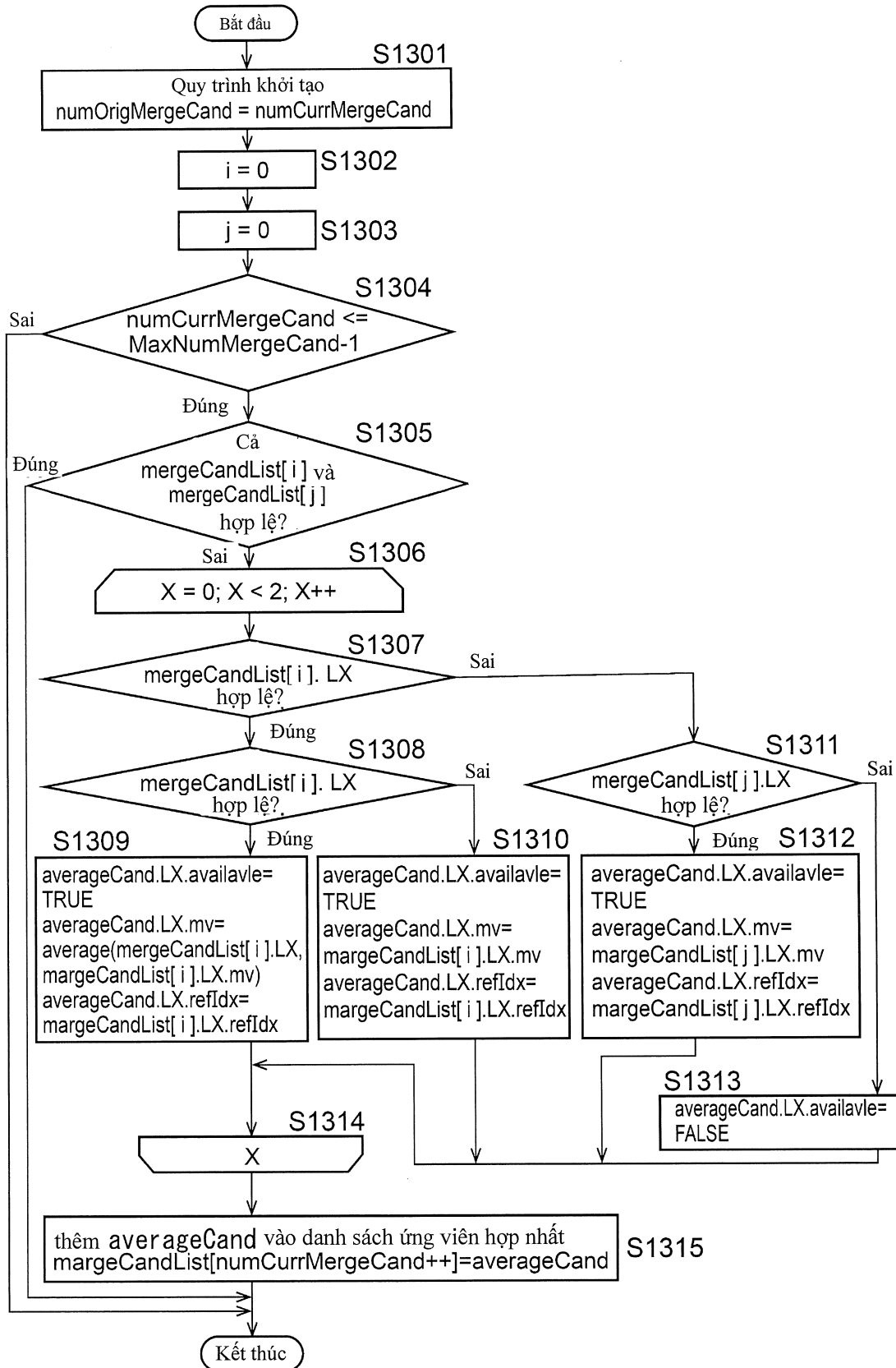


FIG39A

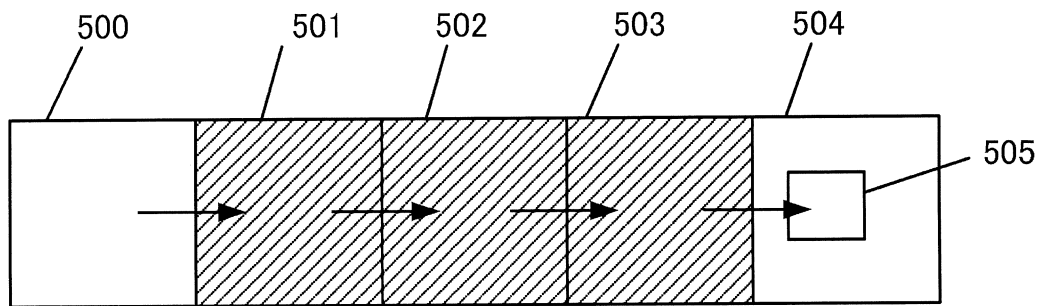


FIG39B

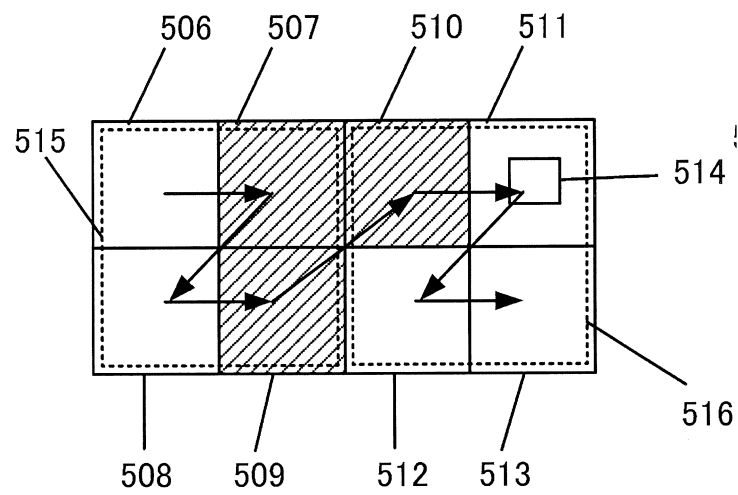
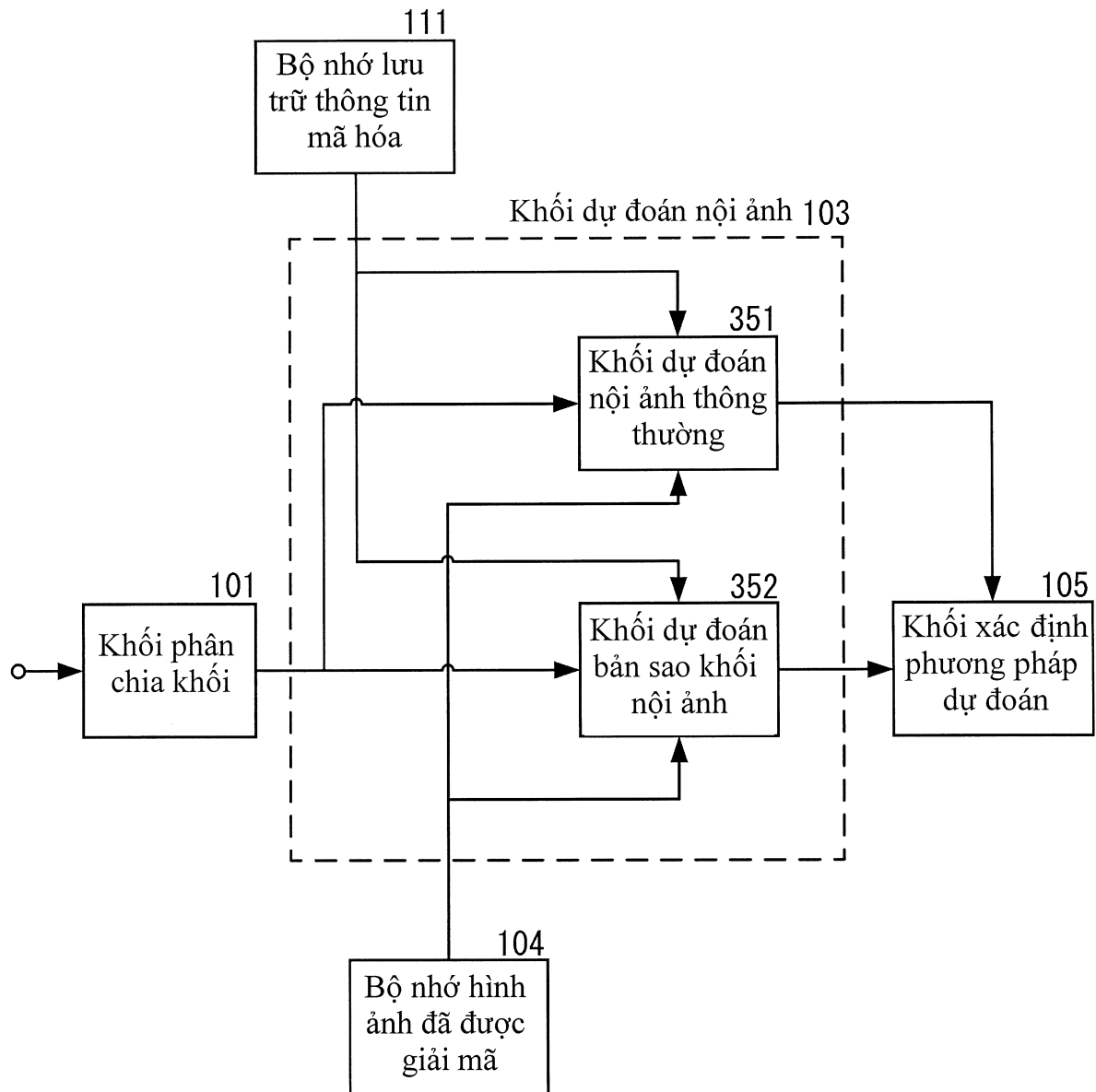


FIG.40



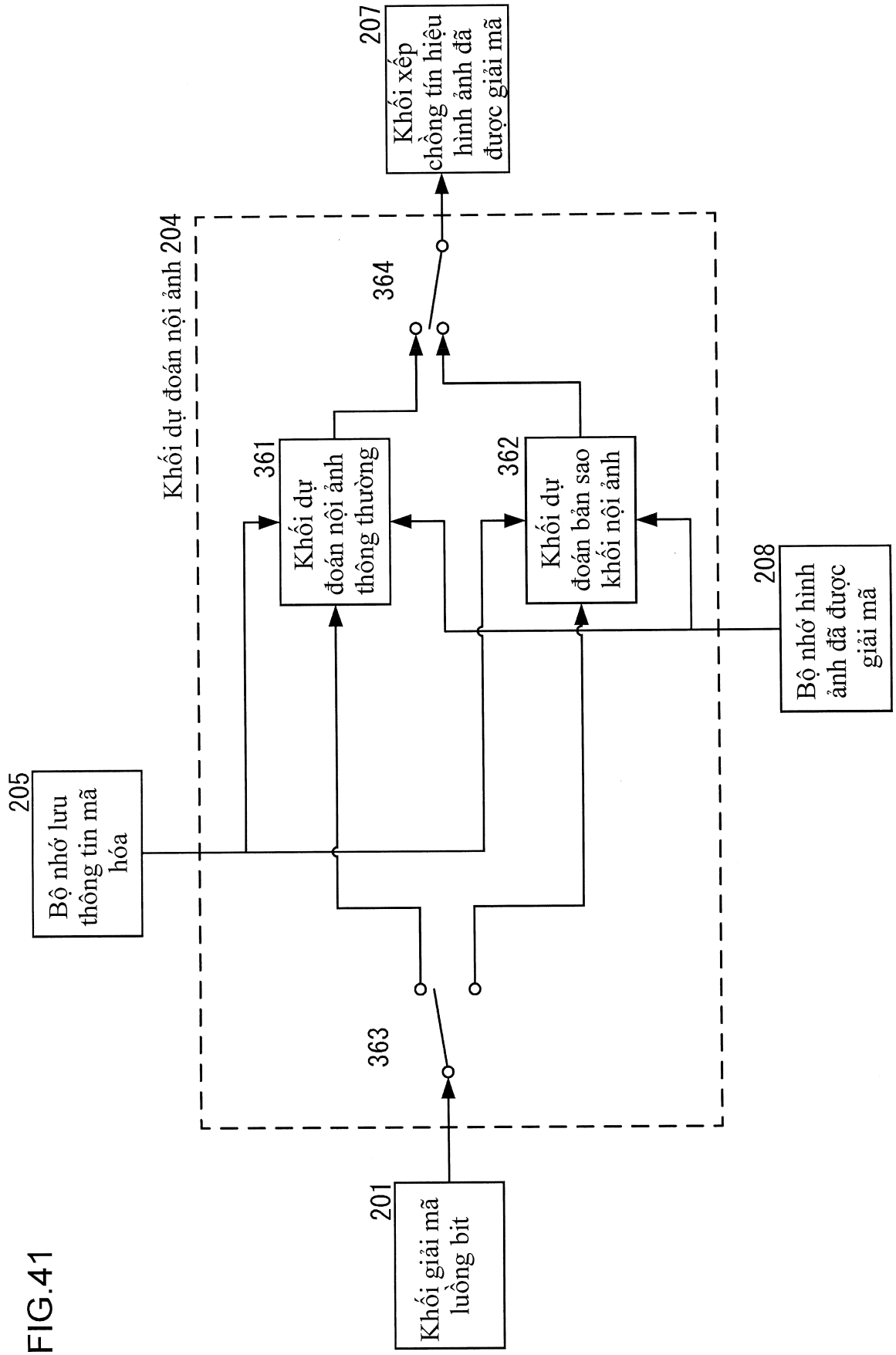


FIG.42

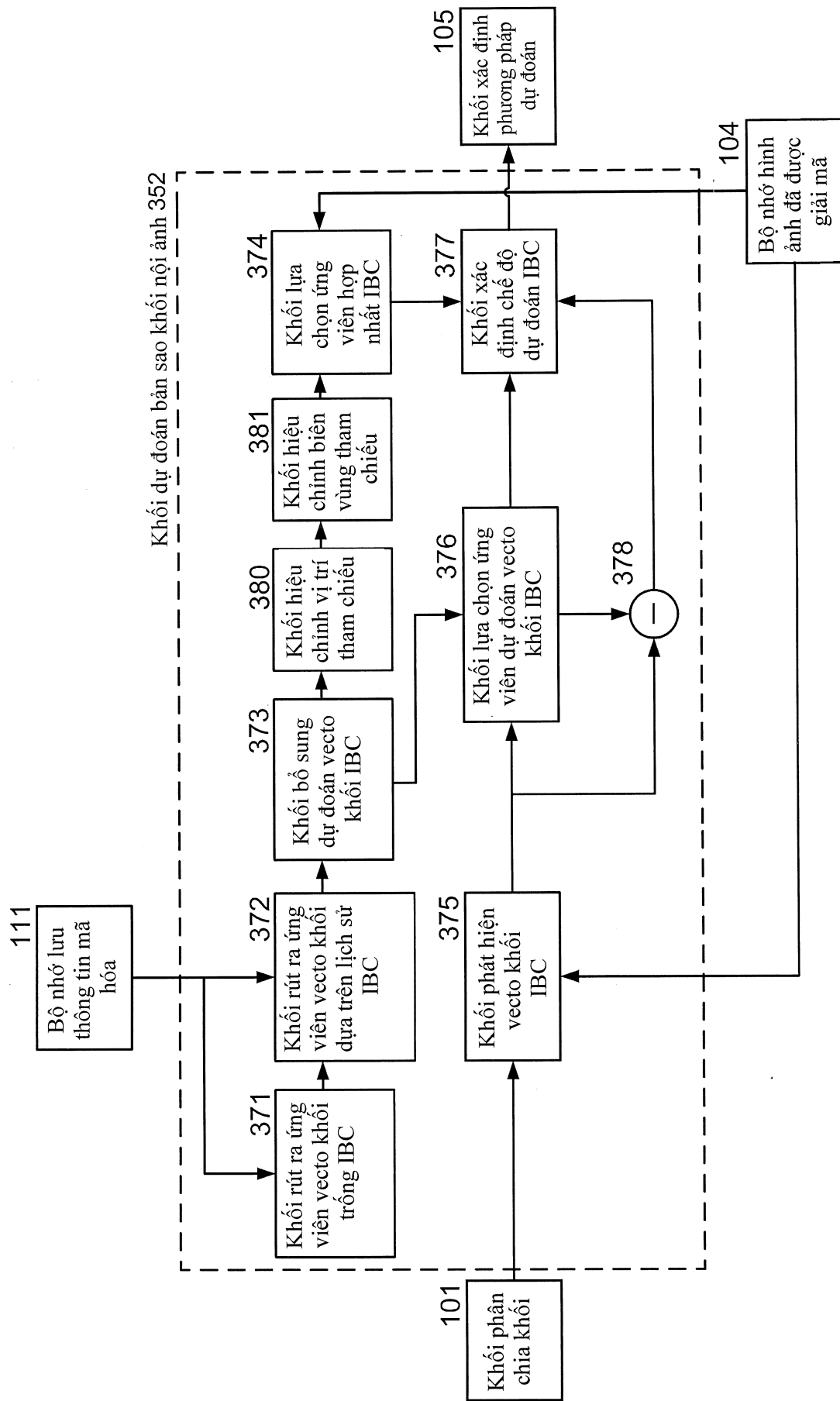


FIG.43

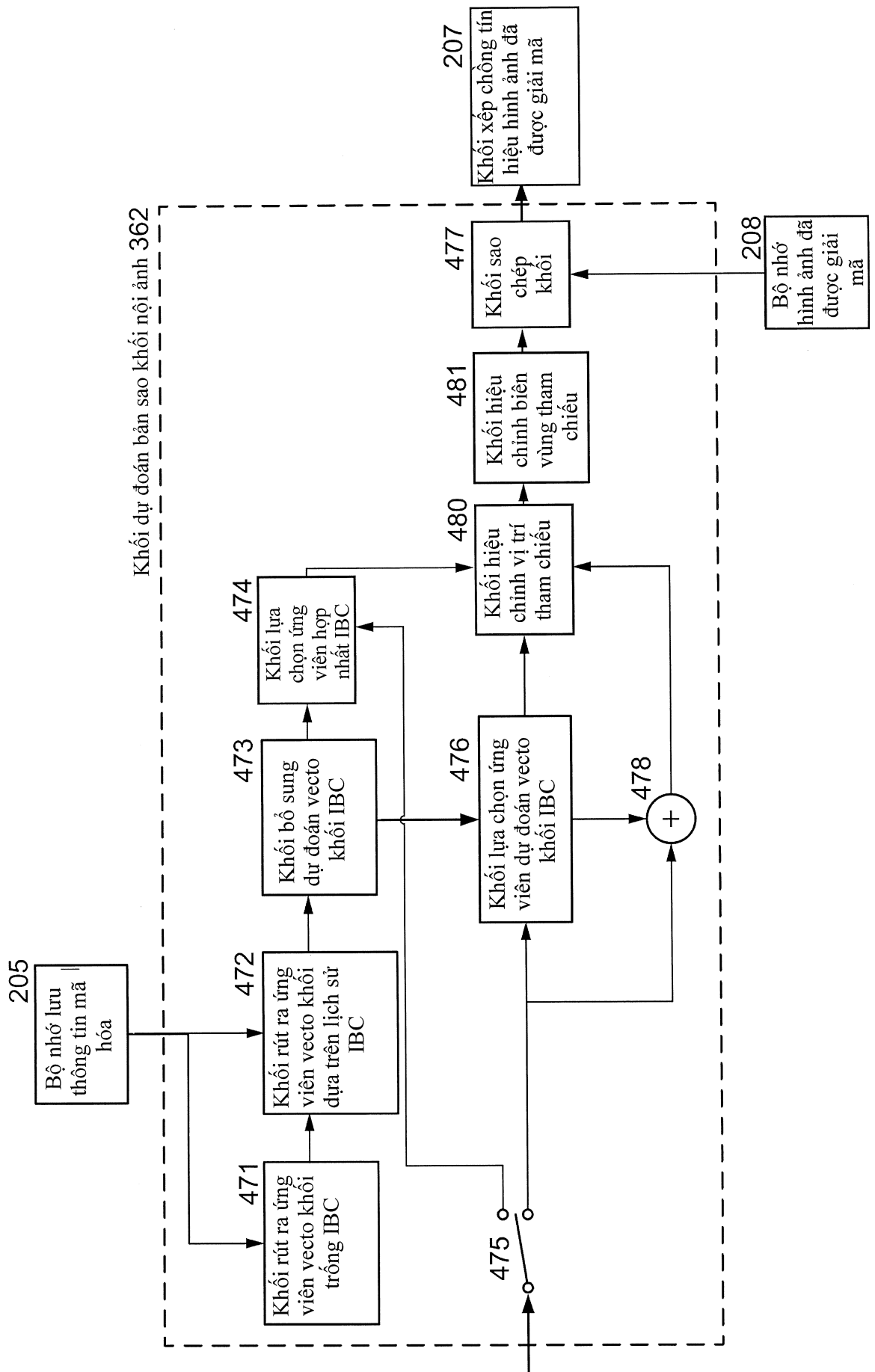


FIG.44

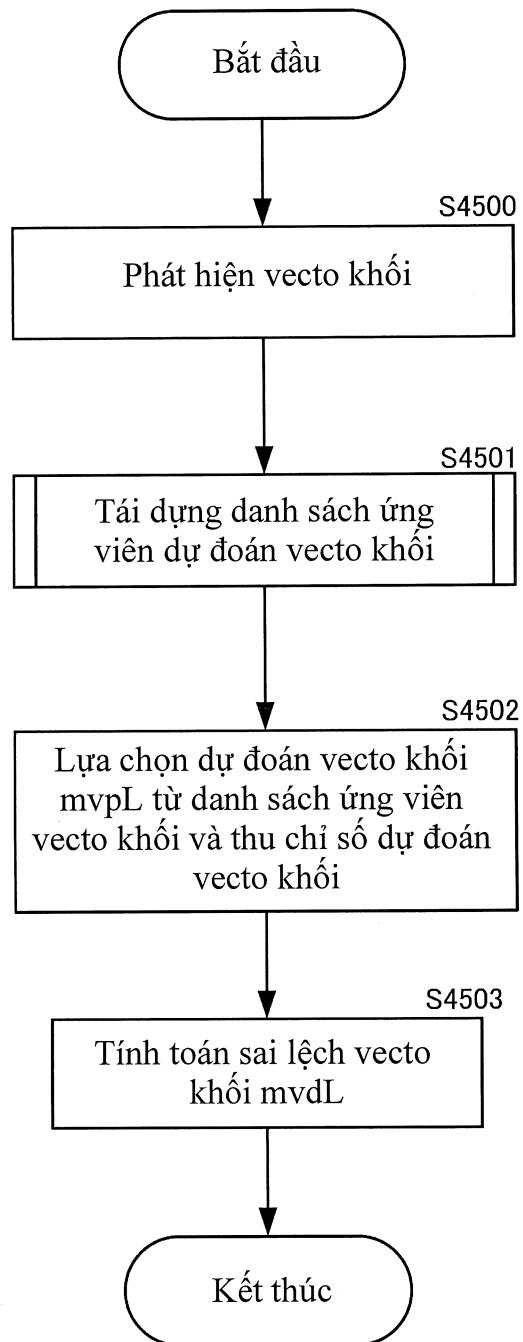


FIG.45

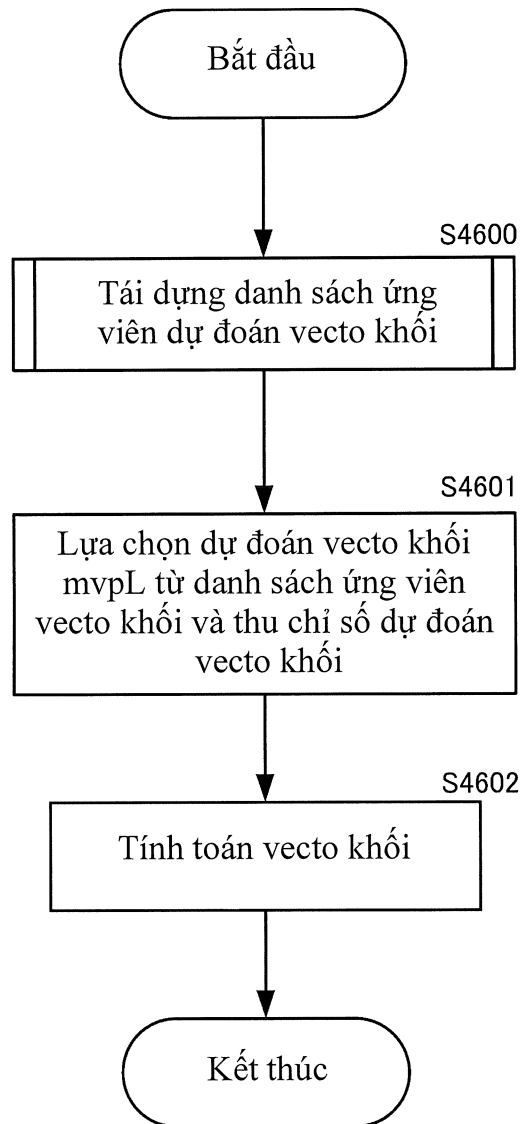


FIG.46

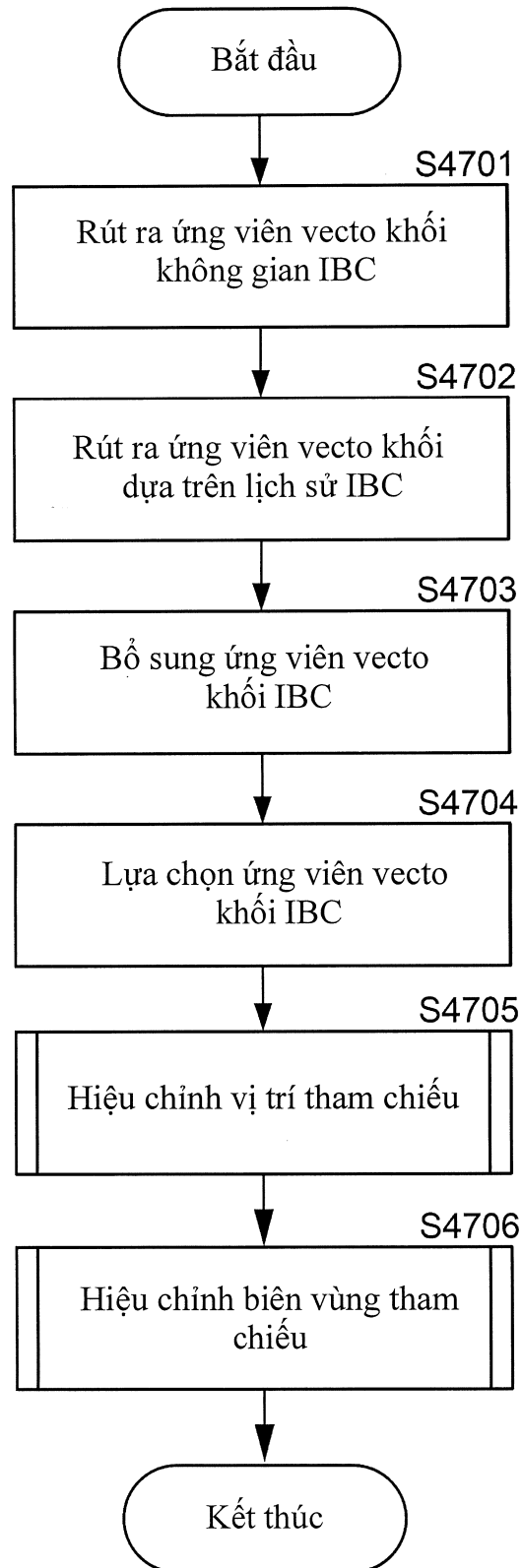


FIG.47

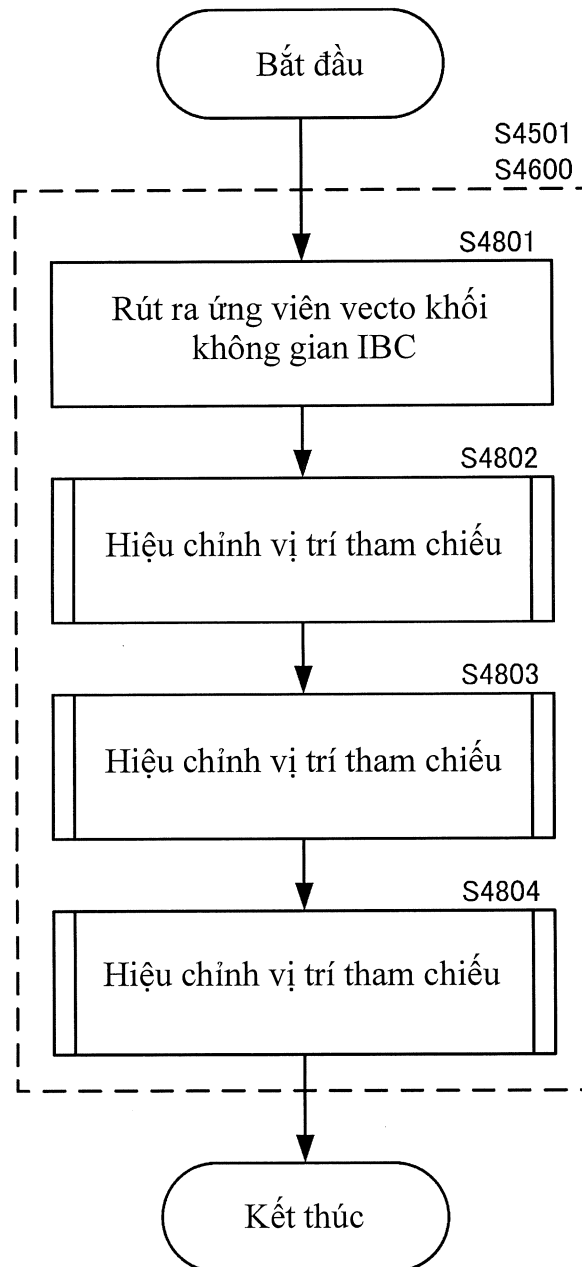


FIG.48

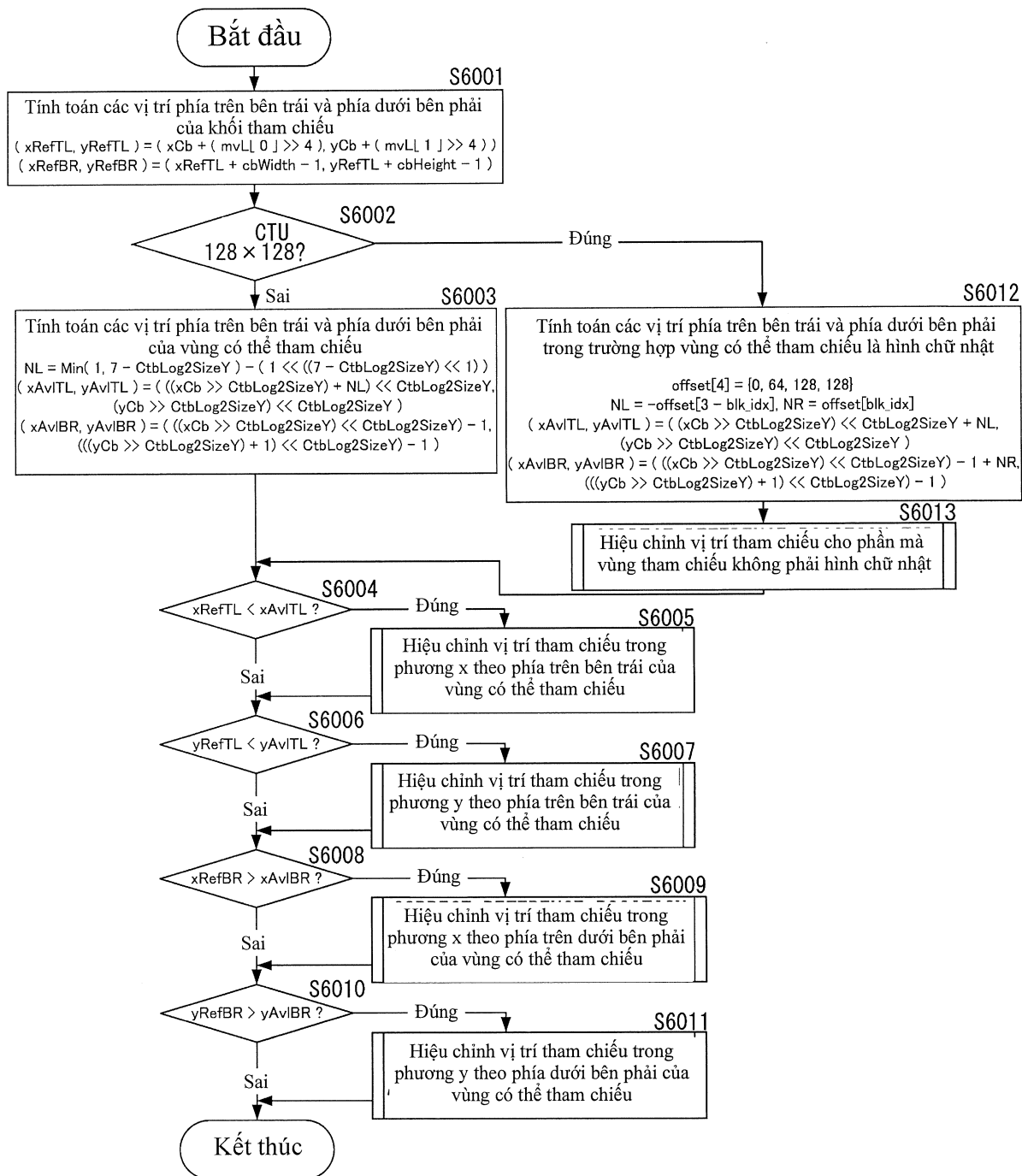


FIG. 49

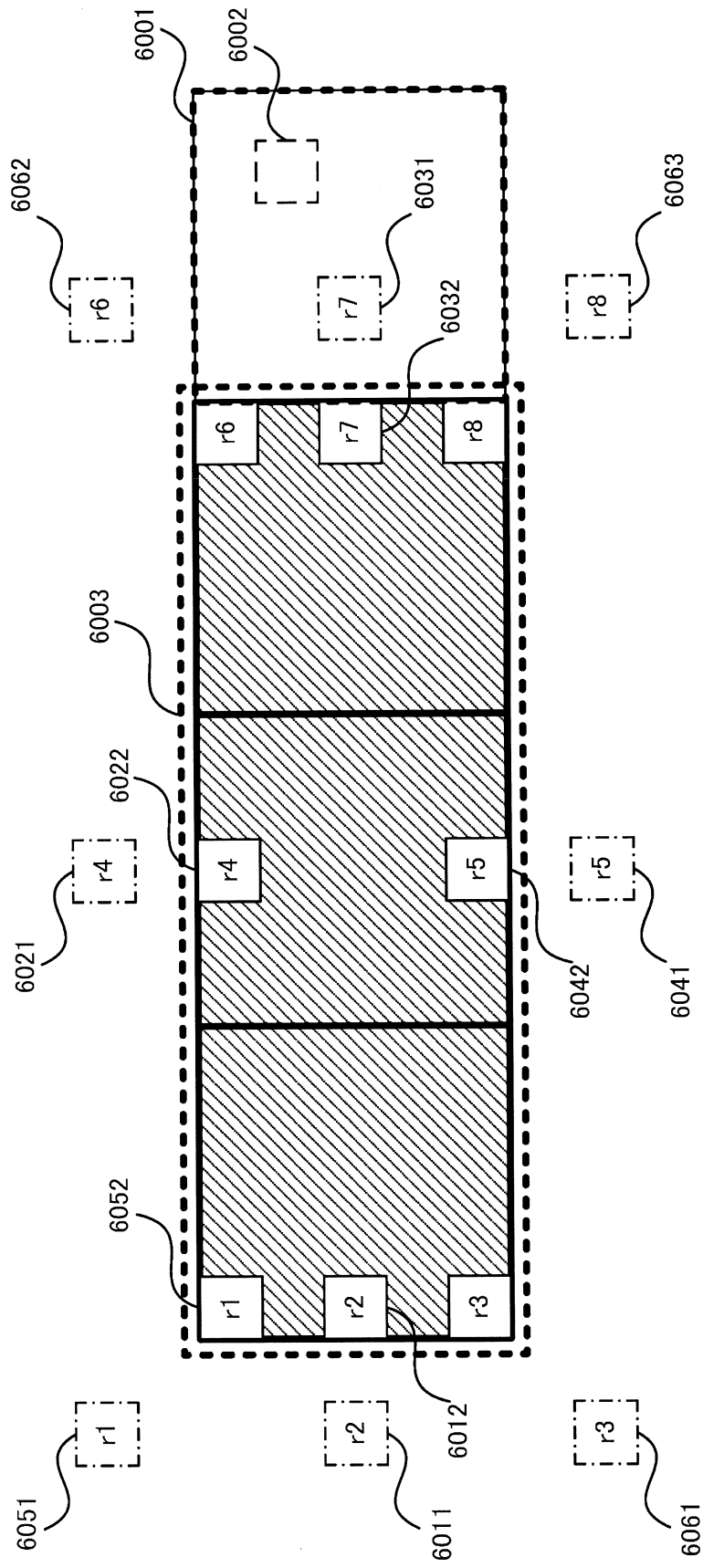


FIG.50A

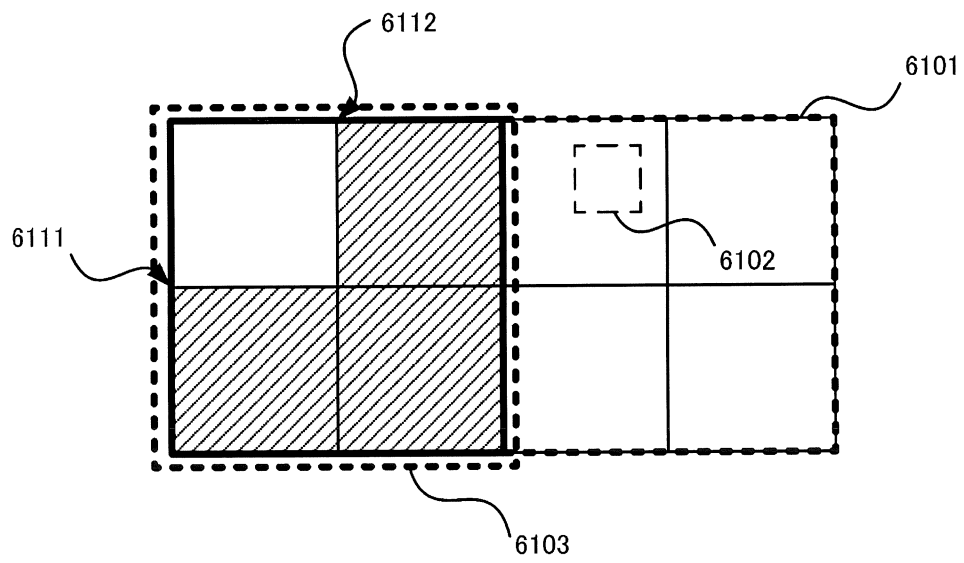


FIG.50B

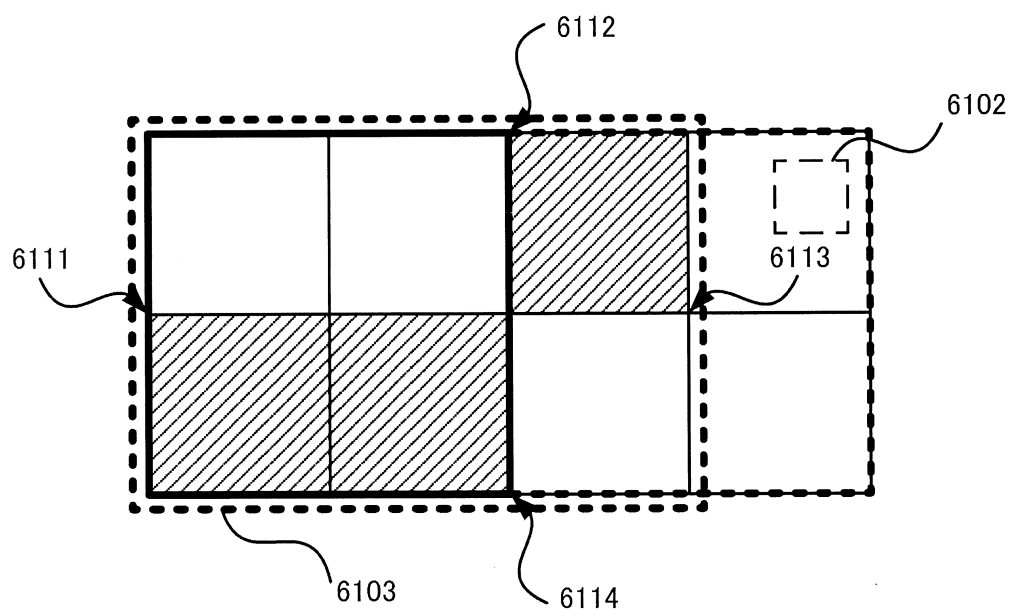


FIG.50C

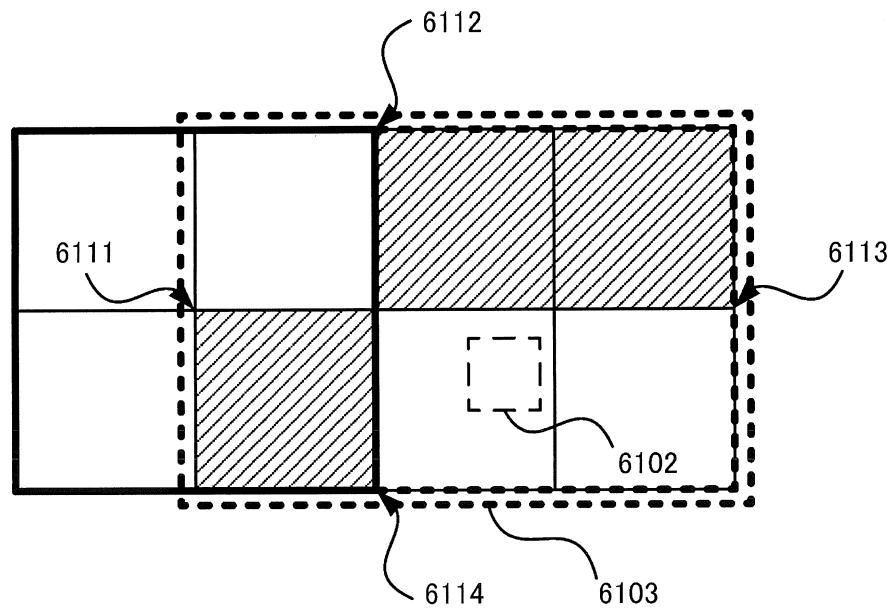


FIG.50D

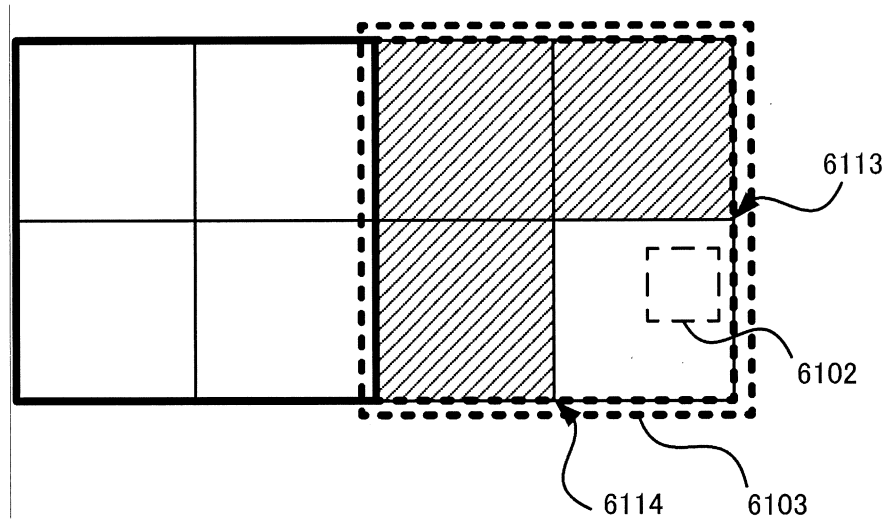


FIG.51

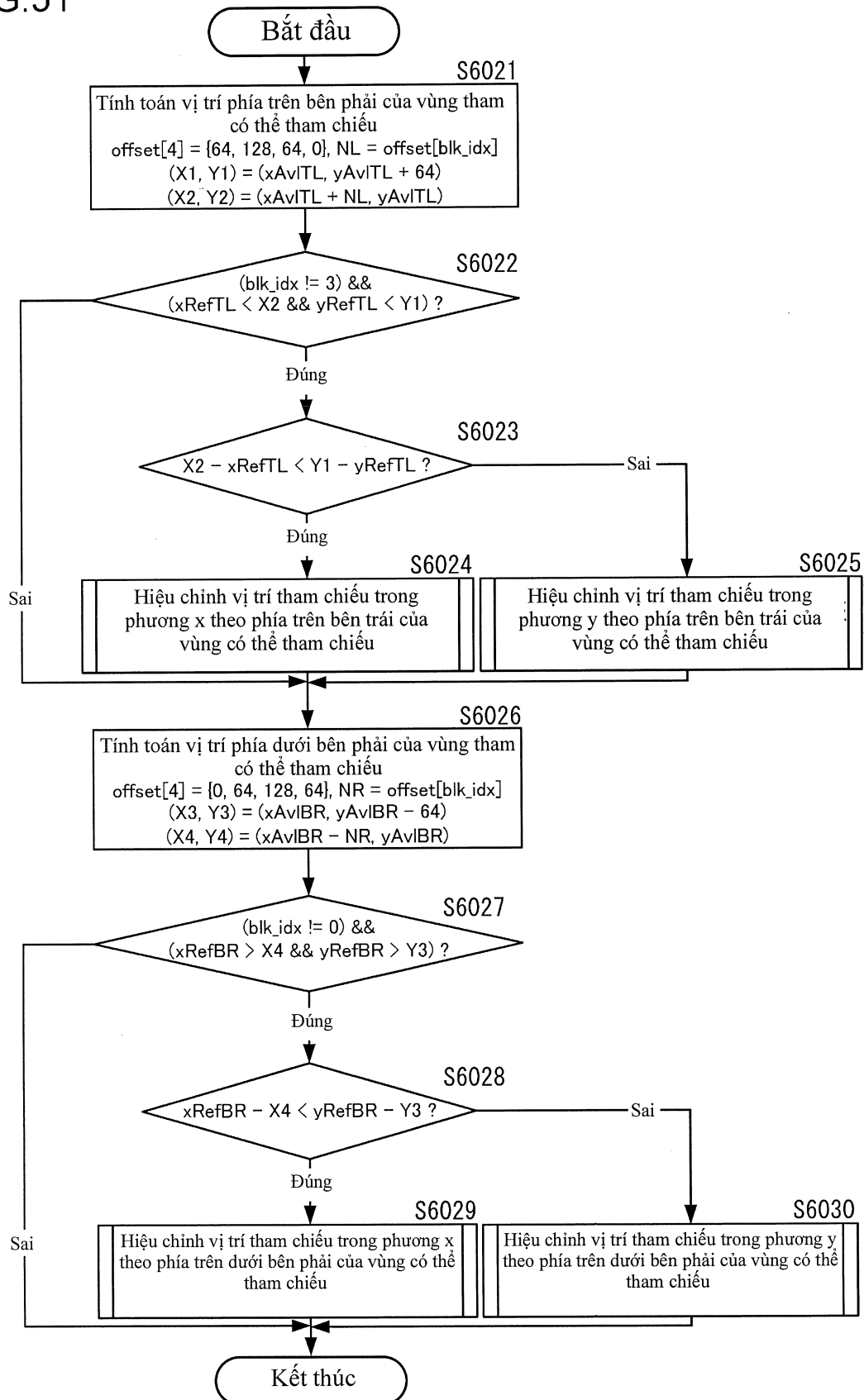


FIG.52A

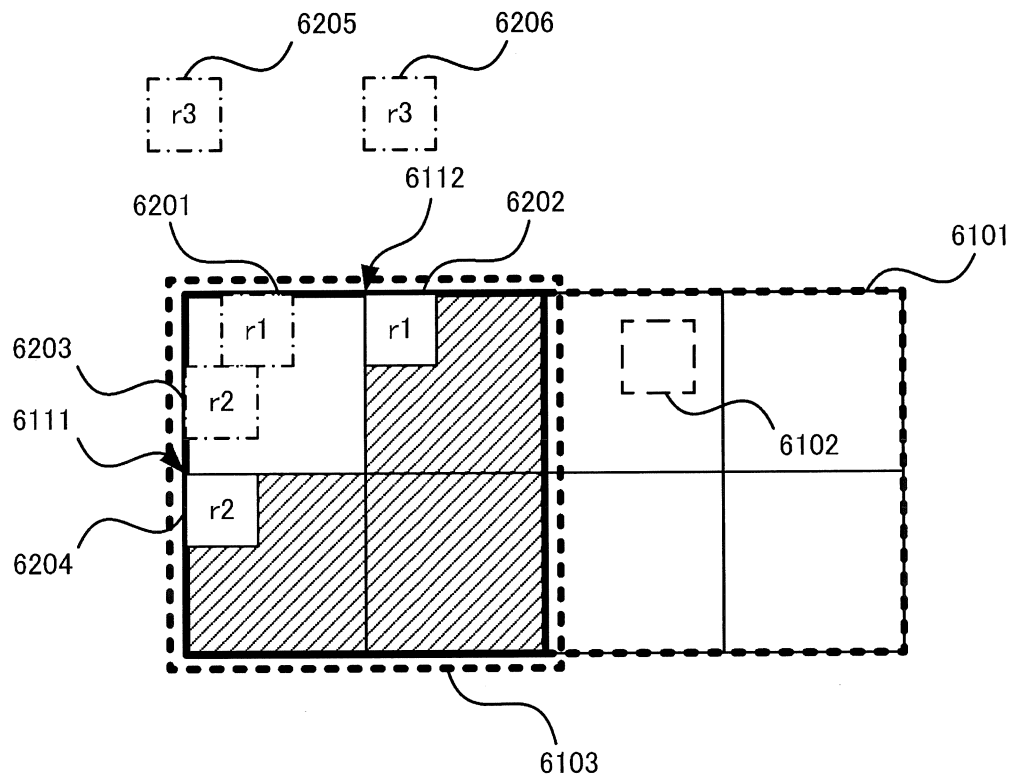


FIG.52B

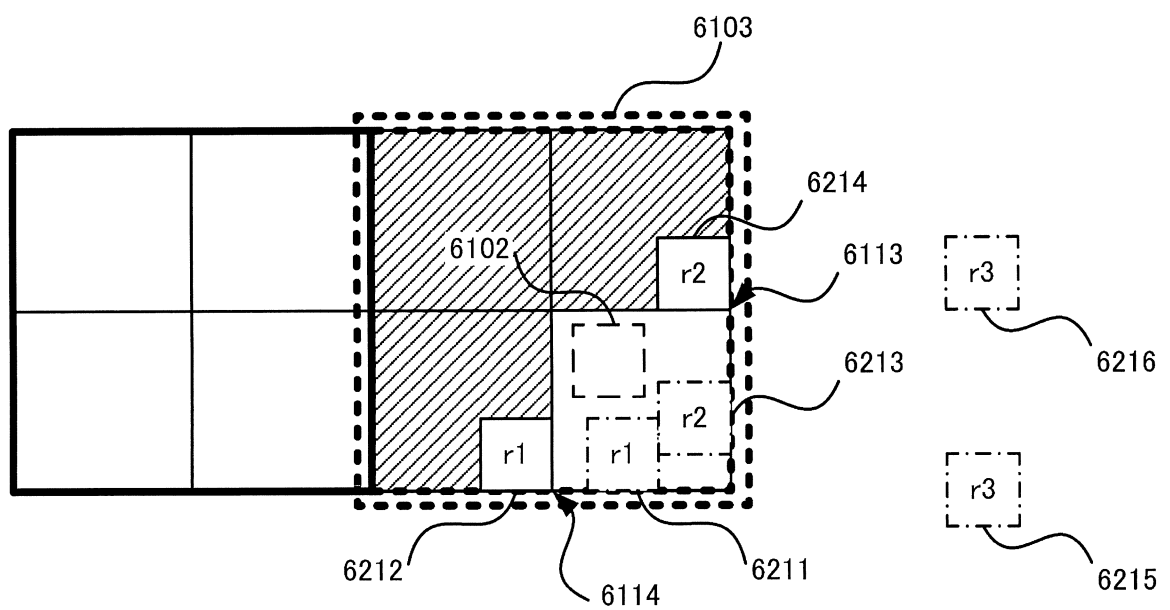


FIG.53

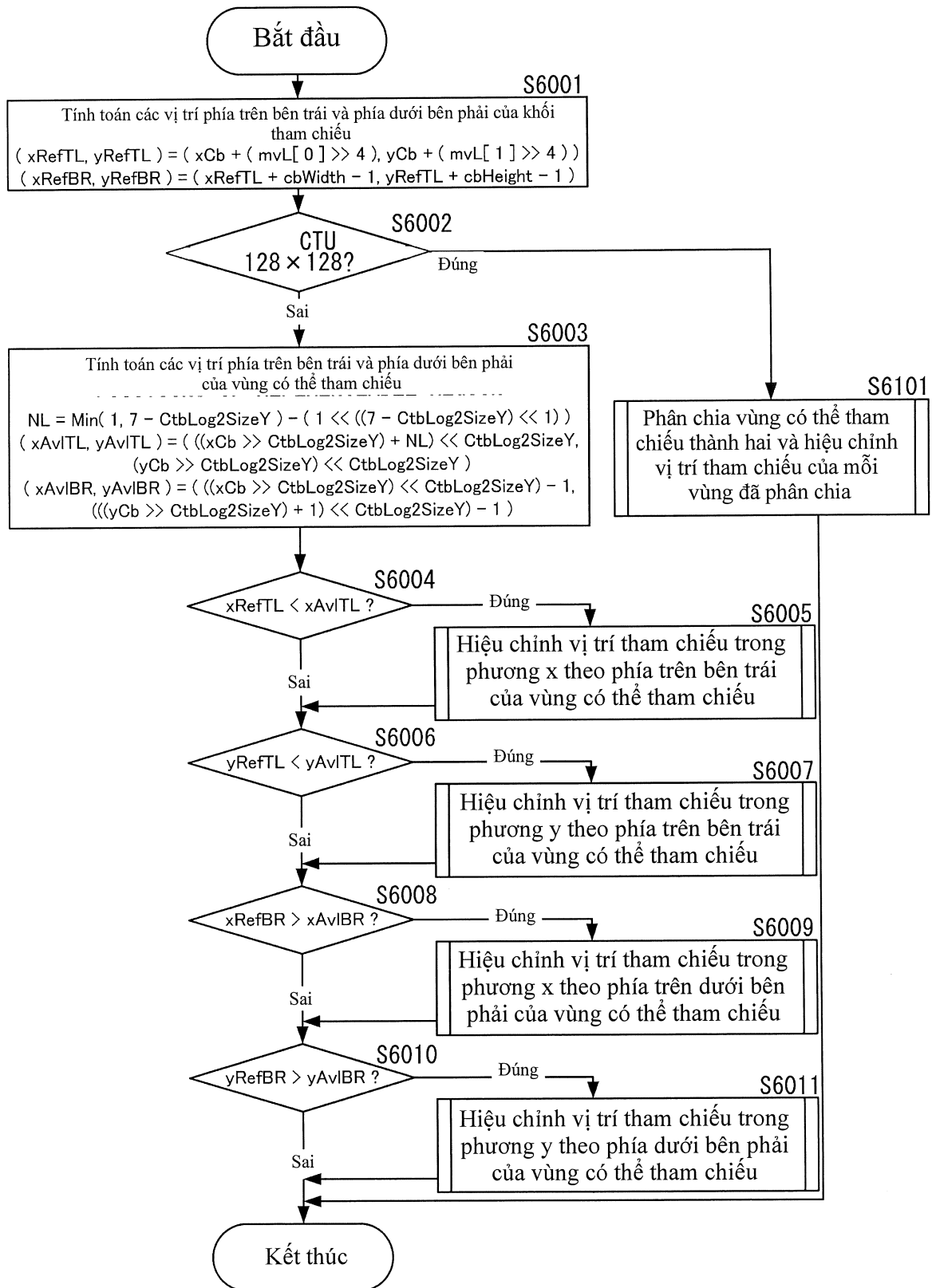


FIG.54A

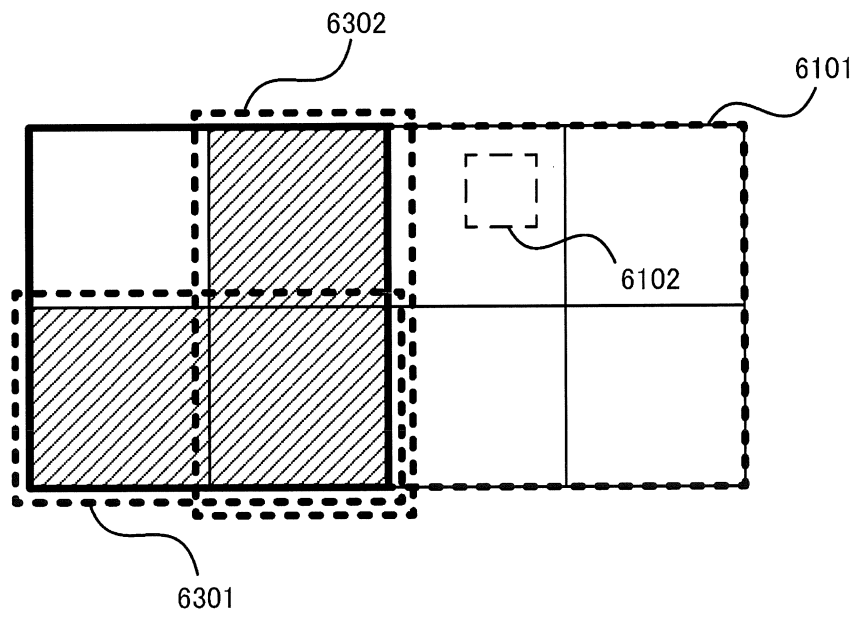


FIG.54B

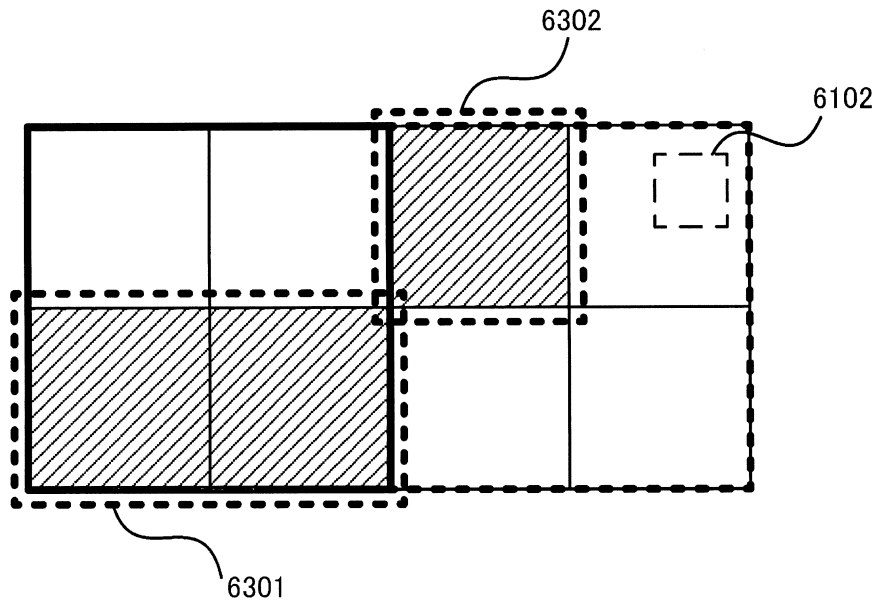


FIG.54C

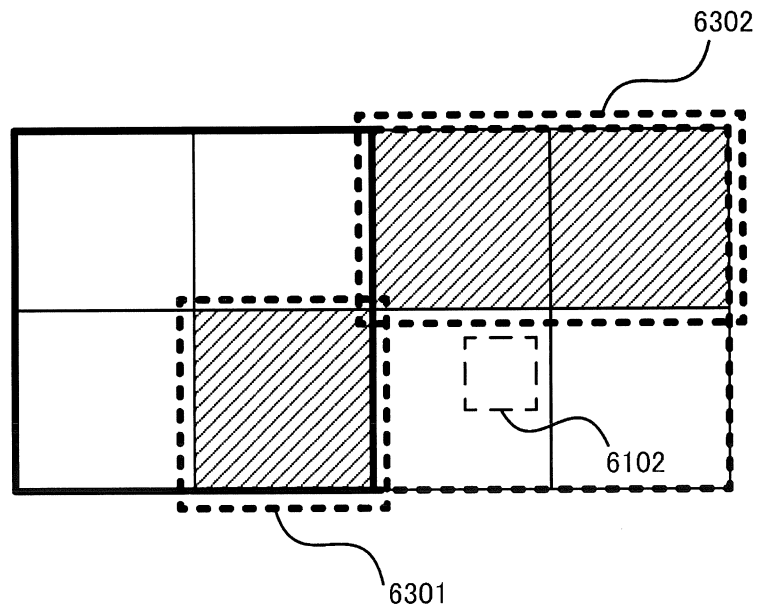


FIG.54D

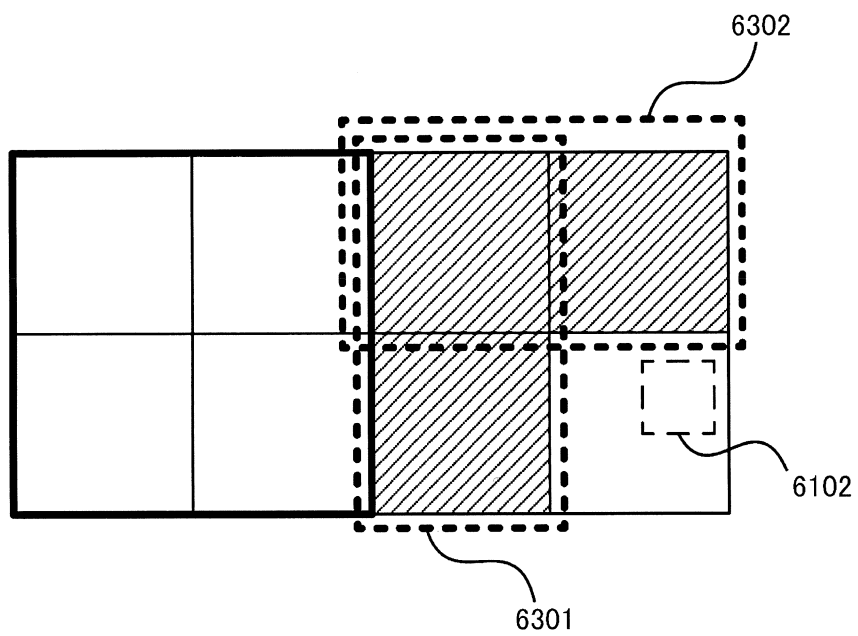


FIG.55

