



(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)



1-0050641

(51)^{2020.01} H04N 19/119; H04N 19/70; H04N
19/176; H04N 19/543; H04N 19/109;
H04N 19/167

(13) B

(21) 1-2022-01854

(22) 26/08/2019

(86) PCT/EP2019/072716 26/08/2019

(87) WO 2021/037337 04/03/2021

(45) 25/08/2025 449

(43) 25/08/2022 413A

(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)

Huawei Administration Building Bantian, Longgang District Shenzhen, Guangdong
518129, China

(72) GAO, Han (CN); ESENLIK, Semih (TR); ALSHINA, Elena Alexandrovna (RU);
WANG, Biao (CN); KOTRA, Anand Meher (IN).

(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ LƯU TRỮ THÔNG TIN CHUYỂN ĐỘNG, THIẾT
BỊ MÃ HÓA/GIẢI MÃ DÒNG VIDEO, VÀ VẬT GHI MÁY TÍNH ĐƯỢC ĐƯỢC

(21)1-2022-01854

(57) Sáng chế đề cập đến phương pháp lưu trữ thông tin chuyển động bao gồm các bước: xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung; và so sánh khoảng cách có ngưỡng để xác định liệu thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu, trong đó thông tin chuyển động thứ ba được dẫn xuất bằng thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai

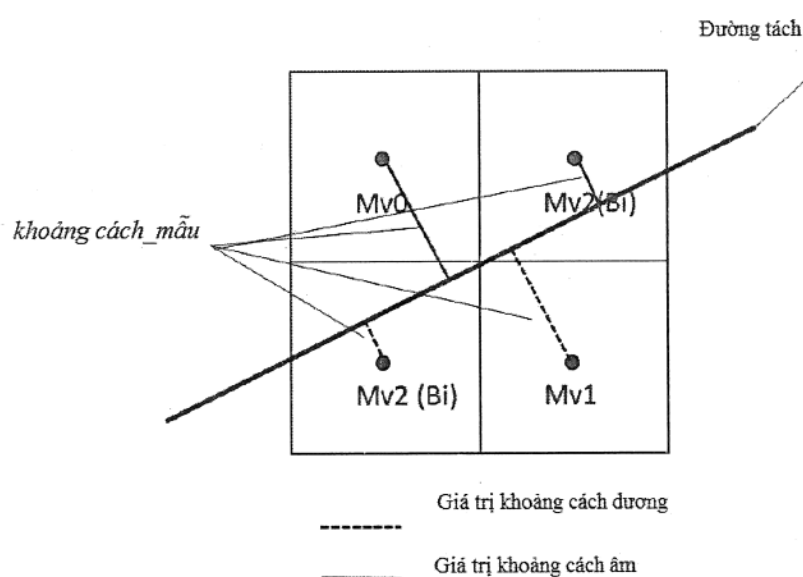


Fig.19

Lĩnh vực kỹ thuật được đề cập

Sáng chế nói chung đề cập đến lĩnh vực tạo mã video và cụ thể hơn đến dự báo ngoài.

Tình trạng kỹ thuật của sáng chế

Kỹ thuật tạo mã video (mã hóa và giải mã video) được sử dụng trong các ứng dụng video số, chẳng hạn phát sóng TV số, truyền video trên các mạng internet và di động, các ứng dụng hội nghị thời gian thực, chẳng hạn, trò chuyện video, hội nghị video, đĩa DVD và Blu-ray, các hệ thống thu thập và chỉnh sửa nội dung video, và các máy quay xách tay của các ứng dụng an ninh.

Lượng dữ liệu video cần để mô tả video cực ngắn có thể là khá lớn, có thể dẫn đến nhiều khó khăn khi dữ liệu cần được phát trực tuyến hoặc được truyền giữa mạng truyền thông có dung lượng băng thông giới hạn. Do vậy, dữ liệu video thường được nén trước khi được truyền giữa các mạng viễn thông ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ do các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở phía nguồn để mã hóa dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó giảm số lượng dữ liệu cần để biểu diễn các ảnh video số. Sau đó, dữ liệu được nén được nhận ở đích bởi thiết bị giải nén video mà giải mã dữ liệu video. Với các tài nguyên mạng hữu hạn và các nhu cầu ngày càng tăng có chất lượng video cao hơn, mong muốn có các kỹ thuật nén và giải nén được cải thiện sẽ cải thiện tỷ lệ nén mà không ảnh hưởng chất lượng ảnh.

Bản chất kỹ thuật của sáng chế

Sáng chế đề cập đến thiết bị và phương pháp mã hóa và giải mã theo các điểm yêu cầu bảo hộ độc lập.

Các mục đích nêu trên và các mục đích khác đạt được nhờ đối tượng của các điểm yêu cầu bảo hộ. Các dạng triển khai khác trở nên rõ ràng trong các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Khía cạnh thứ nhất của sáng chế đề cập đến phương pháp lưu trữ thông tin chuyển động, khối hiện tại bao gồm khối phụ thứ nhất và khối phụ thứ hai, khối phụ thứ nhất và khối phụ thứ hai được tách riêng bởi đường biên chung, khối phụ thứ nhất tương ứng với thông tin chuyển động thứ nhất và khối phụ thứ hai tương ứng với thông tin chuyển động thứ hai, phương pháp bao gồm các bước: xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung; và so sánh khoảng cách có ngưỡng để xác định liệu thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu, trong đó thông tin chuyển động thứ ba được dẫn xuất bằng thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai.

Theo cách triển khai linh hoạt, việc so sánh khoảng cách có ngưỡng để xác định liệu thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu, bao gồm bước: xác định thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu trong trường hợp mà giá trị tuyệt đối của khoảng cách nhỏ hơn hoặc bằng ngưỡng.

Theo cách triển khai linh hoạt, trong trường hợp mà giá trị tuyệt đối của khoảng cách lớn hơn ngưỡng, phương pháp còn bao gồm các bước: xác định thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là dương; hoặc, xác định thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là âm.

Theo cách triển khai linh hoạt, ngưỡng được định trước.

Theo cách triển khai linh hoạt, ngưỡng dựa trên mối quan hệ hình học giữa đường biên chung và phương ngang hoặc phương dọc.

Theo cách triển khai linh hoạt, ngưỡng dựa trên góc giữa phương vuông góc của đường biên chung và phương ngang hoặc phương dọc.

Theo cách triển khai linh hoạt, ngưỡng dựa trên chiều rộng và/hoặc chiều cao của khối hiện tại.

Theo cách triển khai linh hoạt, đường biên chung được xác định bởi tham số góc và tham số khoảng cách, trước khi xác định khoảng cách giữa tập hợp mẫu

trong khối hiện tại và đường biên chung, phương pháp còn bao gồm bước: xác định tham số góc và tham số khoảng cách.

Theo cách triển khai linh hoạt, trong bộ giải mã, xác định tham số góc và tham số khoảng cách, bao gồm: phân tách tham số góc và tham số khoảng cách từ dòng bit; hoặc, phân tách bộ chỉ báo từ dòng bit, và thu được tham số góc và tham số khoảng cách dựa trên bộ chỉ báo.

Theo cách triển khai linh hoạt, sau khi xác định tham số góc và tham số khoảng cách, phương pháp còn bao gồm các bước: xác định tham số tính toán thứ nhất dựa trên tham số góc; tính toán tham số góc tạm thời dựa trên tham số góc; xác định tham số tính toán thứ hai dựa trên tham số góc tạm thời; và tính toán tham số tính toán thứ ba dựa trên tham số góc và tham số khoảng cách.

Theo cách triển khai linh hoạt, tham số tính toán thứ nhất được xác định bằng cách kiểm tra bảng tra cứu định trước theo tham số góc và tham số tính toán thứ hai được xác định bằng cách kiểm tra bảng tra cứu theo tham số góc tạm thời, và trong đó tham số tính toán thứ nhất và tham số tính toán thứ hai lần lượt là giá trị cosin và giá trị sin của cùng góc.

Theo cách triển khai linh hoạt, khoảng cách được tính toán như sau:

$$\text{distFromLine} = (x + K) * P1 + (y + K) * P2 - P3$$

trong đó P1, P2 và P3 lần lượt là tham số tính toán thứ nhất, tham số tính toán thứ hai và tham số tính toán thứ ba, distFromLine là khoảng cách, K là số nguyên không âm, x là tọa độ ngang của vị trí đích trong tập hợp mẫu, y là tọa độ dọc của vị trí đích trong hệ tọa độ chữ nhật, trong đó vị trí của mẫu trên bên trái của khối hiện tại được gán bằng gốc tọa độ, hướng bên phải được gán bằng phương ngang dương và chiều xuống được gán bằng phương dọc dương.

Theo cách triển khai linh hoạt, K bằng 1.

Theo cách triển khai linh hoạt, vị trí đích được định trước trong tập hợp mẫu.

Theo cách triển khai linh hoạt, vị trí định trước là vị trí trên bên trái của tập hợp mẫu, hoặc vị trí tâm của tập hợp mẫu.

Theo cách triển khai linh hoạt, độ chính xác mẫu của khoảng cách cao hơn mẫu số nguyên; một cách tương ứng, khoảng cách được tính toán như sau:

$$\text{distFromLine} = ((x \ll N) + K) * P1 + ((y \ll N) + K) * P2 - P3$$

trong đó $2N$ là nghịch đảo của độ chính xác mẫu của khoảng cách.

Theo cách triển khai linh hoạt, N bằng 1.

Theo cách triển khai linh hoạt, tập hợp mẫu là tập hợp mẫu độ sáng của khối hiện tại.

Theo cách triển khai linh hoạt, trước khi xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung, phương pháp còn bao gồm bước: tách khối hiện tại thành nhiều tập hợp mẫu, trong đó mỗi tập hợp trong nhiều tập hợp mẫu có kích thước giống như tập hợp mẫu; hoặc thiết lập khối hiện tại dưới dạng tập hợp mẫu.

Theo cách triển khai linh hoạt, tập hợp mẫu là mảng mẫu 4×4 .

Theo cách triển khai linh hoạt, mỗi tập hợp trong nhiều tập hợp mẫu tương ứng với mỗi đơn vị lưu trữ thông tin chuyển động, đơn vị lưu trữ thông tin chuyển động được sử dụng để lưu trữ một trong thông tin chuyển động thứ nhất, thông tin chuyển động thứ hai và thông tin chuyển động thứ ba.

Theo cách triển khai linh hoạt, khối phụ thứ nhất và khối phụ thứ hai là phân vùng tam giác, phân vùng hình thang, hoặc phân vùng hình chữ nhật bất đối xứng.

Theo cách triển khai linh hoạt, thông tin chuyển động thứ nhất bao gồm thông tin chuyển động dựa trên danh sách ảnh tham chiếu thứ nhất, thông tin chuyển động thứ hai bao gồm thông tin chuyển động dựa trên danh sách ảnh tham chiếu thứ hai, và trong đó thông tin chuyển động thứ ba bao gồm thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai.

Khía cạnh thứ hai của sáng chế đề cập đến thiết bị giải mã dòng video bao gồm bộ xử lý và bộ nhớ. Bộ nhớ lưu trữ các lệnh khiến bộ xử lý thực hiện phương pháp theo khía cạnh thứ nhất.

Khía cạnh thứ ba của sáng chế đề cập đến thiết bị mã hóa dòng video bao gồm bộ xử lý và bộ nhớ. Bộ nhớ lưu trữ các lệnh khiến bộ xử lý thực hiện phương pháp theo khía cạnh thứ nhất.

Khía cạnh thứ tư của sáng chế đề cập đến vật ghi máy tính đọc được đã lưu trữ các lệnh trên đó mà khi được thực thi khiến một hoặc nhiều bộ xử lý được tạo cấu hình để mã hóa dữ liệu video. Các lệnh khiến một hoặc nhiều bộ xử lý thực hiện phương pháp theo khía cạnh thứ nhất.

Theo khía cạnh thứ năm, chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh thứ nhất khi được thực thi trên máy tính.

Các mô tả chi tiết về một hoặc nhiều phương án thực hiện được nêu trong các hình vẽ đi kèm và phần mô tả dưới đây. Các dấu hiệu, các đối tượng, và các ưu điểm sẽ trở nên rõ ràng trong phần mô tả, các hình vẽ, và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Các phương án thực hiện sáng chế được mô tả chi tiết dựa vào các hình vẽ đi kèm, trong đó:

Fig.1A là sơ đồ khối thể hiện ví dụ về hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.1B là sơ đồ khối thể hiện ví dụ khác về hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.2 là sơ đồ khối thể hiện ví dụ về bộ mã hóa video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.3 là sơ đồ khối thể hiện cấu trúc lầy lăm ví dụ của bộ giải mã video được tạo cấu hình để thực hiện các phương án thực hiện sáng chế;

Fig.4 là sơ đồ khối minh họa ví dụ về thiết bị mã hóa hoặc thiết bị giải mã;

Fig.5 là sơ đồ khối minh họa ví dụ khác về thiết bị mã hóa hoặc thiết bị giải mã;

Fig.6 là sơ đồ thể hiện ví dụ về các vị trí của ứng viên hợp nhất không gian;

Fig.7 là sơ đồ thể hiện ví dụ về các cặp ứng viên được xem xét để kiểm tra dư của các ứng viên hợp nhất không gian;

Fig.8 là sơ đồ thể hiện ví dụ về phép chia tỷ lệ vector chuyển động đối với ứng viên hợp nhất thời gian;

Fig.9 là sơ đồ thể hiện ví dụ về lưu trữ vector chuyển động đối với khối dự báo có các mẫu độ sáng kích thước 8x8;

Fig.10 là sơ đồ thể hiện ví dụ về dự báo ngoài dựa trên phân vùng tam giác;

Fig.11 là sơ đồ thể hiện ví dụ về lựa chọn MV đơn dự báo cho chế độ phân vùng tam giác;

Fig.12 là sơ đồ thể hiện ví dụ về các trọng số được sử dụng trong quá trình kết hợp;

Fig.13 là sơ đồ thể hiện ví dụ về lưu trữ vector chuyển động đối với chế độ phân vùng tam giác;

Fig.14 là sơ đồ thể hiện ví dụ về phân vùng hình học khác nhau;

Fig.15 là sơ đồ thể hiện ví dụ về các tham số phân vùng hình học;

Fig.16 là sơ đồ thể hiện ví dụ về hoạt động kết hợp;

Fig.17 là sơ đồ thể hiện ví dụ về bảng tra cứu đối với hoạt động kết hợp;

Fig.18 là sơ đồ thể hiện ví dụ về ví dụ lưu trữ chuyển động 4x4 có vị trí số nguyên và vị trí nửa pixel;

Fig.19 là sơ đồ thể hiện ví dụ về phương pháp lưu trữ chuyển động;

Fig.20 là sơ đồ khối minh họa ví dụ về phương pháp lưu trữ thông tin chuyển động;

Fig.21 là sơ đồ khối minh họa ví dụ về thiết bị lưu trữ thông tin chuyển động;

Fig.22 là sơ đồ khối thể hiện cấu trúc lấy làm ví dụ của hệ thống cấp nội dung 3100 thực hiện dịch vụ truyền nội dung; và

Fig.23 là sơ đồ khối thể hiện cấu trúc về ví dụ về thiết bị đầu cuối.

Các dấu tham chiếu giống hệt dưới đây đề cập đến các dấu hiệu giống hệt nhau hoặc ít nhất tương đương về mặt chức năng nếu không được chỉ báo tường minh ngược lại.

Mô tả chi tiết sáng chế

Trong phần mô tả sau, tham khảo các hình vẽ đi kèm, tạo thành một phần của sáng chế, và thể hiện, thông qua minh họa, các khía cạnh cụ thể của các phương án thực hiện sáng chế hoặc các khía cạnh cụ thể trong đó các phương án thực hiện sáng chế có thể được sử dụng. Cần hiểu rằng các phương án thực hiện sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các thay đổi cấu trúc hoặc logic không được mô tả trong các hình vẽ. Do vậy, phần mô tả chi

tiết sau không được hiểu theo nghĩa giới hạn, và phạm vi sáng chế được định nghĩa bởi các điểm yêu cầu bảo hộ đi kèm.

Chẳng hạn, cần hiểu rằng sáng chế cùng với phương pháp được mô tả cũng có thể đúng đối với thiết bị hoặc hệ thống tương ứng được tạo cấu hình để thực hiện phương pháp và ngược lại. Chẳng hạn, nếu một hoặc nhiều bước của phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm một hoặc nhiều đơn vị, chẳng hạn, các đơn vị chức năng, để thực hiện một hoặc nhiều bước của phương pháp được mô tả (chẳng hạn, một đơn vị thực hiện một hoặc nhiều bước, hoặc các đơn vị mà mỗi đơn vị thực hiện một hoặc nhiều trong số các bước), thậm chí nếu một hoặc nhiều đơn vị này không được mô tả hoặc minh họa riêng rẽ trong các hình vẽ. Tuy nhiên, chẳng hạn, nếu thiết bị cụ thể được mô tả dựa trên một hoặc nhiều đơn vị, chẳng hạn, các đơn vị chức năng, phương pháp tương ứng có thể bao gồm một bước để thực hiện chức năng của một hoặc nhiều đơn vị (chẳng hạn, một bước thực hiện chức năng của một hoặc nhiều đơn vị, hoặc mỗi bước trong các bước thực hiện chức năng của một hoặc nhiều trong các khối), thậm chí nếu một hoặc nhiều bước này không được mô tả hoặc minh họa tường minh trong các hình vẽ. Ngoài ra, cần hiểu rằng các dấu hiệu của các phương án thực hiện lấy làm ví dụ khác nhau và/hoặc các khía cạnh được mô tả ở đây có thể được kết hợp với nhau, trừ khi có lưu ý khác.

Kỹ thuật tạo mã video thường đề cập đến việc xử lý chuỗi ảnh, mà tạo video hoặc chuỗi video. Thay vì cụm từ “ảnh”, cụm từ “khung” hoặc “ảnh” có thể được sử dụng làm các từ đồng nghĩa trong lĩnh vực tạo mã video. Kỹ thuật tạo mã video (hoặc tạo mã nói chung) bao gồm hai phần mã hóa video và giải mã video. Mã hóa video được thực hiện ở phía nguồn, thường bao gồm bước xử lý (chẳng hạn, bằng cách nén) các ảnh video ban đầu để giảm lượng dữ liệu cần để biểu diễn các ảnh video (để lưu trữ và/hoặc truyền hiệu quả hơn). Giải mã video được thực hiện ở phía đích và thường bao gồm bước xử lý ngược so với bộ mã hóa để tái tạo các ảnh video. Các phương án thực hiện đề cập đến “tạo mã” các ảnh video (hoặc các ảnh nói chung) cần hiểu là liên quan đến “mã hóa” hoặc “giải mã” các ảnh video hoặc các chuỗi video tương ứng. Việc kết hợp của phần mã hóa và phần giải mã cũng được gọi là CODEC (mã hóa và giải mã).

Trong trường hợp tạo mã video không tổn hao, các ảnh video ban đầu có thể được tái tạo, tức là, các ảnh video được tái tạo có cùng chất lượng với các ảnh video ban đầu (giả sử không tổn hao truyền hoặc tổn hao dữ liệu khác trong khi lưu trữ hoặc truyền). Trong trường hợp tạo mã video có tổn hao, kỹ thuật nén khác, chẳng hạn, bằng cách lượng tử, được thực hiện, để giảm lượng dữ liệu biểu diễn các ảnh video, không thể được tái tạo hoàn toàn ở bộ giải mã, tức là, chất lượng của các ảnh video được tái tạo thấp hơn hoặc tệ hơn so với chất lượng của các ảnh video ban đầu.

Vài chuẩn tạo mã video thuộc nhóm “các codec video lai có tổn hao” (tức là, kết hợp dự báo không gian và thời gian trong miền mẫu và mã hóa biến đổi 2D để áp dụng lượng tử trong miền biến đổi). Mỗi ảnh của chuỗi video thường được phân vùng thành tập hợp khối không chồng lấn và việc tạo mã thường được thực hiện ở mức khối. Nói cách khác, ở bộ mã hóa, video thường được xử lý, tức là, được mã hóa, ở mức độ khối (khối video), chẳng hạn, nhờ sử dụng dự báo không gian (trong ảnh) và/hoặc dự báo thời gian (ngoài ảnh) để tạo khối dự báo, lấy khối hiện tại (khối đang được xử lý/cần được xử lý) trừ đi khối dự báo để thu được khối dư, biến đổi khối dư và lượng tử khối dư trong miền biến đổi để giảm lượng dữ liệu cần được truyền (nén), trong đó ở bộ giải mã, bước xử lý đảo so với bộ mã hóa được áp dụng cho khối được mã hóa hoặc được nén để tái tạo khối hiện tại để biểu diễn. Ngoài ra, bộ mã hóa sao chép vòng xử lý bộ giải mã sao cho cả hai sẽ tạo các dự báo giống hệt (chẳng hạn, các dự báo trong và ngoài) và/hoặc các tái tạo để xử lý, tức là, tạo mã, các khối tiếp theo.

Theo các phương án thực hiện sau, hệ thống tạo mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa trên các hình vẽ từ Fig.1 đến Fig.3.

Fig. 1A là sơ đồ khối minh họa hệ thống tạo mã 10 lấy làm ví dụ, chẳng hạn, hệ thống tạo mã video 10 (hoặc gọi tắt là hệ thống tạo mã 10) có thể sử dụng các kỹ thuật của sáng chế. Bộ mã hóa video 20 (hoặc gọi tắt là bộ mã hóa 20) và bộ giải mã video 30 (hoặc gọi tắt là bộ giải mã 30) của hệ thống tạo mã video 10 biểu diễn các ví dụ về các thiết bị có thể được tạo cấu hình để thực hiện các kỹ thuật theo các ví dụ khác nhau được mô tả theo sáng chế.

Như được thể hiện trên Fig.1A, hệ thống tạo mã 10 bao gồm thiết bị nguồn 12 được tạo cấu hình để cấp dữ liệu ảnh được mã hóa 21 chẳng hạn, đến thiết bị đích 14 để giải mã dữ liệu ảnh được mã hóa 13.

Thiết bị nguồn 12 bao gồm bộ mã hóa 20, và có thể còn, tức là, tùy chọn, bao gồm nguồn ảnh 16, bộ tiền xử lý (hoặc khối tiền xử lý) 18, chẳng hạn, bộ tiền xử lý ảnh 18, và giao diện truyền thông hoặc khối truyền thông 22.

Nguồn ảnh 16 có thể bao gồm hoặc là loại bất kỳ của thiết bị chụp ảnh, chẳng hạn camera để chụp ảnh thế giới thực, và/hoặc loại bất kỳ của thiết bị tạo ảnh, chẳng hạn bộ xử lý đồ họa máy tính để tạo ảnh được làm động bằng máy tính, hoặc loại bất kỳ của thiết bị khác để thu được và/hoặc tạo ảnh thế giới thực, ảnh được tạo bằng máy tính (chẳng hạn, nội dung màn hình, ảnh thực tại ảo (virtual reality, VR)) và/hoặc kết hợp bất kỳ của nó (chẳng hạn, ảnh thực tại tăng cường (augmented reality, AR)). Nguồn ảnh có thể là loại bất kỳ của bộ nhớ hoặc bộ lưu trữ mà lưu trữ ảnh bất kỳ trong các ảnh nêu trên.

Để phân biệt với bộ tiền xử lý 18 và quá trình được thực hiện bằng bộ tiền xử lý 18, ảnh hoặc dữ liệu ảnh 17 có thể cũng được gọi là ảnh thô hoặc dữ liệu ảnh thô 17.

Bộ tiền xử lý 18 được tạo cấu hình để nhận dữ liệu ảnh (thô) 17 và để thực hiện tiền xử lý trên dữ liệu ảnh 17 để thu được ảnh được tiền xử lý 19 hoặc dữ liệu ảnh được tiền xử lý 19. Bước tiền xử lý được thực hiện bằng bộ tiền xử lý 18 có thể, chẳng hạn, bao gồm cắt gọn, biến đổi định dạng màu (chẳng hạn, từ RGB sang YCbCr), hiệu chỉnh màu, hoặc khử nhiễu. Có thể hiểu rằng khối tiền xử lý 18 có thể có thể là thành phần tùy chọn.

Bộ mã hóa video 20 được tạo cấu hình để nhận dữ liệu ảnh được tiền xử lý 19 và cấp dữ liệu ảnh được mã hóa 21 (các chi tiết khác sẽ được mô tả dưới đây, chẳng hạn, dựa trên Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 và truyền dữ liệu ảnh được mã hóa 21 (hoặc bất kỳ phiên bản được xử lý khác của nó) trên kênh truyền thông 13 đến thiết bị khác, chẳng hạn, thiết bị đích 14 hoặc bất kỳ thiết bị khác, để lưu trữ hoặc tái tạo trực tiếp.

Thiết bị đích 14 bao gồm bộ giải mã 30 (chẳng hạn, bộ giải mã video 30), và có thể còn, tức là, một cách tùy chọn, bao gồm giao diện truyền thông hoặc khối truyền thông 28, bộ hậu xử lý 32 (hoặc khối hậu xử lý 32) và bộ phận hiển thị 34.

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 (hoặc bất kỳ phiên bản được xử lý khác của nó), chẳng hạn, trực tiếp từ thiết bị nguồn 12 hoặc từ nguồn khác bất kỳ, chẳng hạn, thiết bị lưu trữ, chẳng hạn, thiết bị lưu trữ dữ liệu ảnh được mã hóa, và cấp dữ liệu ảnh được mã hóa 21 đến bộ giải mã 30.

Giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình để truyền hoặc nhận dữ liệu ảnh được mã hóa 21 hoặc dữ liệu được mã hóa 13 qua liên kết truyền thông trực tiếp giữa thiết bị nguồn 12 và thiết bị đích 14, chẳng hạn, kết nối hữu tuyến hoặc không dây trực tiếp, hoặc qua loại mạng bất kỳ, chẳng hạn, mạng hữu tuyến hoặc không dây hoặc kết hợp bất kỳ của nó, hoặc loại bất kỳ trong mạng riêng tư và công cộng, hoặc loại bất kỳ của kết hợp của nó.

Giao diện truyền thông 22 có thể, chẳng hạn, được tạo cấu hình để đóng gói dữ liệu ảnh được mã hóa 21 thành định dạng thích hợp, chẳng hạn, các gói, và/hoặc xử lý dữ liệu ảnh được mã hóa nhờ sử dụng loại bất kỳ của mã hóa truyền hoặc xử lý để truyền trên liên kết truyền thông hoặc mạng truyền thông.

Giao diện truyền thông 28, tạo thành đối trọng của giao diện truyền thông 22, có thể, chẳng hạn, được tạo cấu hình để nhận dữ liệu được truyền và xử lý dữ liệu truyền nhờ sử dụng loại bất kỳ của giải mã truyền tương ứng hoặc xử lý và/hoặc tháo gói để thu được dữ liệu ảnh được mã hóa 21.

Cả giao diện truyền thông 22 lẫn giao diện truyền thông 28 có thể được tạo cấu hình dưới dạng các giao diện truyền thông đơn hướng như được chỉ báo bằng mũi tên cho kênh truyền thông 13 trên Fig.1A trở từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai hướng, và có thể được tạo cấu hình, chẳng hạn, để truyền và nhận thông điệp, chẳng hạn, để thiết lập kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc truyền dữ liệu, chẳng hạn, truyền dữ liệu ảnh được mã hóa.

Bộ giải mã 30 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 và cấp dữ liệu ảnh được giải mã 31 hoặc ảnh được giải mã 31 (các chi tiết khác sẽ được mô tả dưới đây, chẳng hạn, dựa trên Fig. 3 hoặc Fig. 5).

Bộ hậu xử lý 32 của thiết bị đích 14 được tạo cấu hình để xử lý hậu kỳ dữ liệu ảnh được giải mã 31 (cũng được gọi là dữ liệu ảnh được tái tạo), chẳng hạn, ảnh được giải mã 31, để thu được dữ liệu ảnh được xử lý hậu kỳ 33, chẳng hạn, ảnh được xử lý hậu kỳ 33. Xử lý hậu kỳ được thực hiện bằng khối hậu xử lý 32 có thể bao gồm, chẳng hạn, biến đổi định dạng màu (chẳng hạn, từ YCbCr sang RGB), hiệu chỉnh màu, cắt gọn, hoặc lấy mẫu lại, hoặc bất kỳ xử lý khác, chẳng hạn, để chuẩn bị dữ liệu ảnh được giải mã 31 để hiển thị, chẳng hạn, bằng bộ phận hiển thị 34.

Bộ phận hiển thị 34 của thiết bị đích 14 được tạo cấu hình để nhận dữ liệu ảnh được xử lý hậu kỳ 33 để hiển thị ảnh, chẳng hạn, đến người dùng hoặc người xem. Bộ phận hiển thị 34 có thể là hoặc bao gồm loại bộ phận hiển thị bất kỳ để biểu diễn ảnh được tái tạo, chẳng hạn, bộ phận hiển thị hoặc màn hình được tích hợp hoặc bên ngoài. Các bộ phận hiển thị có thể, chẳng hạn, bao gồm màn hình tinh thể lỏng (liquid crystal display, LCD), màn hình đi-ốt phát quang hữu cơ (organic light emitting diode, OLED), màn hình plasma, máy chiếu, màn hình micro LED, tinh thể lỏng trên silic (liquid crystal on silicon, LCoS), bộ xử lý ánh sáng số (digital light processor, DLP) hoặc loại hiển thị khác bất kỳ.

Mặc dù Fig. 1A mô tả thiết bị nguồn 12 và thiết bị đích 14 dưới dạng các thiết bị riêng rẽ, theo các phương án thực hiện, các thiết bị cũng có thể bao gồm cả hai chức năng, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng. Theo các phương án thực hiện này thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng có thể được triển khai nhờ sử dụng phần cứng và/hoặc phần mềm tương tự hoặc bằng phần cứng và/hoặc phần mềm riêng rẽ hoặc kết hợp bất kỳ của nó.

Người có hiểu biết trung bình trong lĩnh vực sẽ hiểu dựa trên phần mô tả, sự tồn tại và tách riêng (chính xác) các chức năng của các khối khác nhau hoặc các chức năng trong thiết bị nguồn 12 và/hoặc thiết bị đích 14 như được thể hiện trên Fig. 1A có thể thay đổi phụ thuộc vào thiết bị và ứng dụng thực.

Bộ mã hóa 20 (chẳng hạn, bộ mã hóa video 20) hoặc bộ giải mã 30 (chẳng hạn, bộ giải mã video 30) hoặc cả bộ mã hóa 20 lẫn bộ giải mã 30 có thể được triển khai qua hệ mạch xử lý như được thể hiện trên Fig.1B, chẳng hạn một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (digital signal processor, DSP), mạch tích hợp ứng dụng cụ thể (application-specific integrated circuit, ASIC), mảng cổng dạng trường lập trình được (field-programmable gate array, FPGA), logic rời rạc, phần cứng, tạo mã video dành riêng hoặc kết hợp bất kỳ của nó. Bộ mã hóa 20 có thể được triển khai qua hệ mạch xử lý 46 để triển khai các môđun khác nhau như được nêu theo bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống phụ mã hóa khác bất kỳ được mô tả ở đây. Bộ giải mã 30 có thể được triển khai qua hệ mạch xử lý 46 để thực hiện các môđun khác nhau như được nêu dựa vào bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống phụ giải mã khác bất kỳ được mô tả ở đây. Hệ mạch xử lý có thể được tạo cấu hình để thực hiện các hoạt động khác nhau như được nêu sau đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được triển khai một phần trong phần mềm, thiết bị có thể lưu trữ các lệnh cho phần mềm trong vật lưu trữ máy tính đọc được thích hợp bất biến và có thể thực thi các lệnh trong phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Một trong bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp dưới dạng một phần của bộ mã hóa/bộ giải mã kết hợp (CODEC) trong một thiết bị, chẳng hạn, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong các loại thiết bị, bao gồm loại bất kỳ của thiết bị cầm tay hoặc tĩnh, chẳng hạn, notebook hoặc máy tính xách tay, điện thoại di động, điện thoại thông minh, tablet hoặc máy tính bảng, camera, máy tính để bàn, hộp tín hiệu truyền hình cáp (set-top box, STB), TV, bộ phận hiển thị, máy phát media số, bảng điều khiển video game, thiết bị phát trực tuyến video (chẳng hạn, máy chủ dịch vụ nội dung hoặc máy chủ truyền nội dung), bộ thu phát thanh, bộ truyền phát thanh, hoặc tương tự và có thể không sử dụng hệ điều hành hoặc sử dụng hệ điều hành bất kỳ. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị để truyền thông không dây. Do vậy, thiết bị nguồn 12 và thiết bị đích 14 có thể là các thiết bị truyền thông không dây.

Trong một số trường hợp, hệ thống tạo mã video 10 được minh họa trên Fig. 1A chỉ là ví dụ và các kỹ thuật của sáng chế có thể áp dụng cho các thiết lập tạo mã video (chẳng hạn, mã hóa video hoặc giải mã video) không nhất thiết bao gồm truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Trong các ví dụ khác, dữ liệu được truy xuất từ bộ nhớ cục bộ, được phát trực tuyến trên mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu đến bộ nhớ, và/hoặc thiết bị giải mã video có thể truy xuất và giải mã dữ liệu từ bộ nhớ. Trong một số ví dụ, bước mã hóa và giải mã được thực hiện bằng các thiết bị không truyền thông với nhau, nhưng chỉ đơn giản mã hóa dữ liệu vào bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Để tiện mô tả, các phương án thực hiện sáng chế được mô tả ở đây, chẳng hạn, nhờ viện dẫn đến chuẩn tạo mã video hiệu suất cao (High-Efficiency Video Coding, HEVC) hoặc đến phần mềm tham chiếu của chuẩn tạo mã video đa dụng (Versatile Video Coding, VVC), chuẩn tạo mã video thế hệ tiếp theo được phát triển bằng nhóm cộng tác chung về tạo mã video (Joint Collaboration Team on video coding, JCT-VC) của nhóm chuyên gia tạo mã video ITU-T (Video Coding Experts Group, VCEG) và nhóm chuyên gia ảnh động (Motion Picture Experts Group, MPEG) ISO/IEC. Người có hiểu biết trung bình trong lĩnh vực sẽ hiểu rằng các phương án thực hiện sáng chế không bị giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

Fig.2 thể hiện sơ đồ khối của bộ mã hóa video 20 lấy làm ví dụ được tạo cấu trúc để thực hiện các kỹ thuật của sáng chế. Trong ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đầu vào 201 (hoặc giao diện đầu vào 201), khối tính toán phần dư 204, khối xử lý biến đổi 206, khối lượng tử 208, khối lượng tử ngược 210, và khối xử lý biến đổi ngược 212, khối tái tạo 214, khối lọc vòng 220, bộ đệm ảnh được giải mã (Decoded Picture buffer, DPB) 230, khối chọn chế độ 260, khối mã hóa entropy 270 và đầu ra 272 (hoặc giao diện đầu ra 272). Khối chọn chế độ 260 có thể bao gồm khối dự báo ngoài 244, khối dự báo trong 254 và khối phân vùng 262. Khối dự báo ngoài 244 có thể bao gồm khối ước tính chuyển động và khối bù chuyển động (không được thể hiện trên hình vẽ). Bộ mã hóa

video 20 như được thể hiện trên Fig.2 có thể cũng được gọi là bộ mã hóa video lai hoặc bộ mã hóa video theo codec video lai.

Khối tính toán phần dư 204, khối xử lý biến đổi 206, khối lượng tử 208, khối chọn chế độ 260 có thể được gọi là tạo đường tín hiệu thuận của bộ mã hóa 20, trong đó khối lượng tử ngược 210, khối xử lý biến đổi ngược 212, khối tái tạo 214, bộ đệm 216, bộ lọc vòng 220, DPB 230, khối dự báo ngoài 244 và khối dự báo trong 254 có thể được gọi là tạo đường tín hiệu ngược của bộ mã hóa video 20, trong đó đường tín hiệu ngược của bộ mã hóa video 20 tương ứng với đường tín hiệu của bộ giải mã (xem bộ giải mã video 30 trên Fig.3). Khối lượng tử ngược 210, khối xử lý biến đổi ngược 212, khối tái tạo 214, bộ lọc vòng 220, DPB 230, khối dự báo ngoài 244 và khối dự báo trong 254 cũng được gọi là tạo “bộ giải mã cài sẵn” của bộ mã hóa video 20.

Ảnh & phân vùng ảnh (ảnh & khối)

Bộ mã hóa 20 có thể được tạo cấu hình để nhận, chẳng hạn, qua đầu vào 201, ảnh 17 (hoặc dữ liệu ảnh 17), chẳng hạn, ảnh của chuỗi ảnh tạo video hoặc chuỗi video. Ảnh hoặc dữ liệu ảnh được nhận cũng có thể là ảnh được tiền xử lý 19 (hoặc dữ liệu ảnh được tiền xử lý 19). Để cho đơn giản, phần mô tả sau đề cập đến ảnh 17. Ảnh 17 có thể cũng được gọi là ảnh hiện tại hoặc ảnh cần được mã hóa (cụ thể là khi tạo mã video để phân biệt ảnh hiện tại với các ảnh khác, chẳng hạn, các ảnh được mã hóa và/hoặc được giải mã của cùng chuỗi video, tức là, chuỗi video cũng bao gồm ảnh hiện tại).

Ảnh (số) là hoặc có thể được xem là mảng hai chiều hoặc ma trận mẫu có các giá trị cường độ. Mẫu trong mảng có thể cũng được gọi là pixel (viết tắt của phần tử ảnh) hoặc pel. Số lượng mẫu theo phương ngang và dọc (hoặc trục) của mảng hoặc ảnh định nghĩa kích thước và/hoặc độ phân giải của ảnh. Để biểu diễn màu, thường là ba thành phần màu được sử dụng, tức là, ảnh có thể được biểu diễn hoặc ba gồm ba mảng mẫu. Ở định dạng RGB hoặc không gian màu, ảnh bao gồm mảng mẫu đỏ, lục và lam tương ứng. Tuy nhiên, trong kỹ thuật tạo mã video, mỗi pixel thường được biểu diễn ở định dạng độ sáng và sắc độ hoặc không gian màu, chẳng hạn, YCbCr, bao gồm thành phần độ sáng được chỉ báo

bằng Y (thay vào đó đôi lúc L cũng được sử dụng) và hai thành phần sắc độ được chỉ báo bằng Cb và Cr. Thành phần độ sáng (hoặc viết tắt là luma) Y biểu diễn độ sáng hoặc cường độ mức xám (chẳng hạn, giống như trong ảnh thang xám), trong khi hai thành phần sắc độ (hoặc viết tắt là chroma) Cb và Cr biểu diễn sắc sai hoặc các thành phần thông tin màu. Do đó, ảnh ở định dạng YCbCr bao gồm mảng mẫu độ sáng của các giá trị mẫu độ sáng (Y), và hai mảng mẫu sắc độ của các giá trị sắc độ (Cb và Cr). Các ảnh ở định dạng RGB có thể được biến đổi thành định dạng YCbCr và ngược lại, quá trình này được biết đến như là biến đổi màu. Nếu ảnh là đơn sắc, ảnh có thể bao gồm chỉ mảng mẫu độ sáng. Do đó, ảnh có thể là, chẳng hạn, mảng mẫu độ sáng ở định dạng đơn sắc hoặc mảng mẫu độ sáng và hai mảng mẫu sắc độ tương ứng ở định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Theo các phương án thực hiện, bộ mã hóa video 20 có thể bao gồm khối phân vùng ảnh (không được mô tả trên Fig.2) được tạo cấu hình để phân vùng ảnh 17 thành các khối ảnh (thường là không chồng lấn) 203. Các khối này có thể cũng được gọi là các khối gốc, các khối lớn (H.264/AVC) hoặc các khối cây mã (coding tree block, CTB) hoặc các đơn vị cây mã (coding tree unit, CTU) (H.265/HEVC và VVC). Khối phân vùng ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối cho tất cả các ảnh của chuỗi video và lưới tương ứng định nghĩa kích thước khối, hoặc thay đổi kích thước khối giữa các ảnh hoặc các tập con hoặc các nhóm ảnh, và phân vùng mỗi ảnh thành các khối tương ứng.

Theo các phương án thực hiện khác, bộ mã hóa video có thể được tạo cấu hình để nhận trực tiếp khối 203 của ảnh 17, chẳng hạn, một, vài hoặc tất cả các khối tạo ảnh 17. Khối ảnh 203 có thể cũng được gọi là khối ảnh hiện tại hoặc khối ảnh cần được tạo mã.

Giống như ảnh 17, khối ảnh 203 lại là hoặc có thể được xem là mảng hai chiều hoặc ma trận mẫu có các giá trị cường độ (các giá trị mẫu), mặc dù có kích thước nhỏ hơn ảnh 17. Nói cách khác, khối 203 có thể bao gồm, chẳng hạn, một mảng mẫu (chẳng hạn, mảng độ sáng trong trường hợp ảnh đơn sắc 17, hoặc mảng độ sáng hoặc sắc độ trong trường hợp ảnh màu) hoặc ba mảng mẫu (chẳng hạn, mảng độ sáng và hai mảng sắc độ trong trường hợp ảnh màu 17) hoặc số

lượng khác bất kỳ và/hoặc loại mảng phụ thuộc vào định dạng màu được áp dụng. Số lượng mẫu theo phương ngang và dọc (hoặc trục) của khối 203 định nghĩa kích thước của khối 203. Do đó, khối có thể, chẳng hạn, là mảng mẫu $M \times N$ (M cột $\times$ N hàng), hoặc mảng hệ số biến đổi $M \times N$.

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig. 2 có thể được tạo cấu hình để mã hóa ảnh 17 theo khối, chẳng hạn, mã hóa và dự báo được thực hiện theo khối 203.

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig. 2 có thể còn được tạo cấu hình để phân vùng và/hoặc mã hóa ảnh nhờ sử dụng các lát (cũng được gọi là các lát video), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát (thường là không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU) hoặc một hoặc nhiều nhóm khối (chẳng hạn, các ô ảnh (H.265/HEVC và VVC) hoặc các khối viên (VVC)).

Theo các phương án thực hiện, bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân vùng và/hoặc mã hóa ảnh nhờ sử dụng các lát/các nhóm ô (cũng được gọi là các nhóm ô video) và/hoặc các ô (cũng được gọi là các ô video), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát/nhóm ô (thường là không chồng lấn), và mỗi lát/nhóm ô có thể bao gồm, chẳng hạn, một hoặc nhiều khối (chẳng hạn, các CTU) hoặc một hoặc nhiều ô, trong đó mỗi ô, chẳng hạn, có thể có hình chữ nhật và có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU), chẳng hạn, các khối hoàn chỉnh hoặc phân số.

Tính toán dư

Khối tính toán phần dư 204 có thể được tạo cấu hình để tính toán khối dư 205 (cũng được gọi là phần dư 205) dựa trên khối ảnh 203 và khối dự báo 265 (các chi tiết khác về khối dự báo 265 được nêu sau đây), chẳng hạn, bằng cách lấy các giá trị mẫu của khối ảnh 203 trừ đi các giá trị mẫu của khối dự báo 265, theo mẫu (từng pixel) để thu được khối dư 205 trong miền mẫu.

Biến đổi

Khối xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng biến đổi, chẳng hạn, biến đổi cosin rời rạc (discrete cosine transform, DCT) hoặc biến đổi sin rời rạc (discrete sine transform, DST), trên các giá trị mẫu của khối dư 205 để thu được các hệ số biến đổi 207 trong miền biến đổi. Các hệ số biến đổi 207 có thể cũng được gọi là các hệ số dư biến đổi và biểu diễn khối dư 205 trong miền biến đổi.

Khối xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng các phép xấp xỉ nguyên của DCT/DST, chẳng hạn các phép biến đổi được xác định cho H.265/HEVC. So với phép biến đổi DCT trực giao, các phép xấp xỉ số nguyên này thường được chia tỷ lệ theo hệ số cụ thể. Để duy trì chuẩn của khối dư được xử lý bằng các phép biến đổi thuận và ngược, các hệ số chia tỷ lệ bổ sung được áp dụng như là một phần của quá trình biến đổi. Các hệ số chia tỷ lệ thường được chọn dựa trên các ràng buộc cụ thể như việc các hệ số chia tỷ lệ là lũy thừa 2 đối với các phép dịch, chiều sâu bit của các hệ số biến đổi, cân bằng giữa độ chính xác và các chi phí thực hiện, v.v.. Các hệ số chia tỷ lệ cụ thể, chẳng hạn, được xác định cho phép biến đổi ngược, chẳng hạn, bằng khối xử lý biến đổi ngược 212 (và phép biến đổi ngược tương ứng, chẳng hạn, bằng khối xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các hệ số chia tỷ lệ tương ứng cho phép biến đổi thuận, chẳng hạn, bởi khối xử lý biến đổi 206, ở bộ mã hóa 20 có thể được xác định tương ứng.

Theo các phương án thực hiện, bộ mã hóa video 20 (tương ứng là khối xử lý biến đổi 206) có thể được tạo cấu hình để xuất ra các tham số biến đổi, chẳng hạn, loại biến đổi hoặc các phép biến đổi, chẳng hạn, trực tiếp hoặc được mã hóa hoặc được nén qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số biến đổi để giải mã.

Lượng tử

Khối lượng tử 208 có thể được tạo cấu hình để lượng tử các hệ số biến đổi 207 để thu được các hệ số lượng tử 209, chẳng hạn, bằng cách áp dụng lượng tử

vô hướng hoặc lượng tử vector. Các hệ số lượng tử 209 có thể cũng được gọi là các hệ số biến đổi lượng tử 209 hoặc các hệ số dư được lượng tử 209.

Quá trình lượng tử có thể giảm chiều sâu bit được liên kết với một số hoặc tất cả các hệ số biến đổi 207. Chẳng hạn, hệ số biến đổi n bit có thể được làm tròn xuống hệ số biến đổi m bit trong khi lượng tử, trong đó n lớn hơn m . Mức độ lượng tử có thể được chỉnh sửa bằng cách điều chỉnh tham số lượng tử (quantization parameter, QP). Chẳng hạn đối với lượng tử vô hướng, phép chia tỷ lệ khác có thể được áp dụng để thu được lượng tử mịn hơn hoặc thô hơn. Kích thước bậc lượng tử nhỏ hơn tương ứng với lượng tử mịn hơn, trong đó các kích thước bậc lượng tử lớn hơn tương ứng với lượng tử thô hơn. Kích thước bậc lượng tử áp dụng được có thể được chỉ báo bằng QP. QP có thể chẳng hạn là chỉ số cho tập hợp định trước của các kích thước bậc lượng tử áp dụng được. Chẳng hạn, các QP nhỏ có thể tương ứng với lượng tử mịn hơn (các kích thước bậc lượng tử nhỏ) và các QP lớn có thể tương ứng với lượng tử thô (các kích thước bậc lượng tử lớn) hoặc ngược lại. Lượng tử có thể bao gồm phép chia cho kích thước bậc lượng tử và lượng tử ngược tương ứng, chẳng hạn, bởi khối lượng tử ngược 210, có thể bao gồm phép nhân với kích thước bậc lượng tử. Theo các phương án thực hiện, một số chuẩn, chẳng hạn, HEVC, có thể được tạo cấu hình để sử dụng QP để xác định kích thước bậc lượng tử. Nói chung, kích thước bậc lượng tử có thể được tính toán dựa trên QP nhờ sử dụng phép xấp xỉ dấu chấm tĩnh của phương trình bao gồm phép chia. Các hệ số chia tỷ lệ bổ sung có thể được đưa vào để lượng tử và lượng tử ngược để khôi phục chuẩn của khối dư, có thể được chỉnh sửa nhờ phép chia tỷ lệ được sử dụng trong phép xấp xỉ dấu chấm tĩnh của phương trình cho kích thước bậc lượng tử và QP. Trong một cách triển khai ví dụ, phép chia tỷ lệ biến đổi ngược và lượng tử ngược có thể được kết hợp. Theo cách khác, các bảng lượng tử được tùy chỉnh có thể được sử dụng và được báo hiệu từ bộ mã hóa đến bộ giải mã, chẳng hạn, trong dòng bit. Lượng tử là phép toán có tổn hao, trong đó tổng hao tăng với việc tăng các kích thước bậc lượng tử.

Theo các phương án thực hiện, bộ mã hóa video 20 (tương ứng là khối lượng tử 208) có thể được tạo cấu hình để xuất ra các QP, chẳng hạn, trực tiếp hoặc

được mã hóa qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và áp dụng các QP để giải mã.

Lượng tử ngược

Khối lượng tử ngược 210 được tạo cấu hình để áp dụng lượng tử ngược của khối lượng tử 208 lên các hệ số lượng tử để thu được các hệ số lượng tử ngược 211, chẳng hạn, bằng cách áp dụng phép đảo của phương pháp lượng tử được áp dụng bởi khối lượng tử 208 dựa trên hoặc sử dụng kích thước bậc lượng tử giống như khối lượng tử 208. Các hệ số lượng tử ngược 211 có thể cũng được gọi là các hệ số dư được lượng tử ngược 211 và tương ứng – mặc dù không giống với các hệ số biến đổi do tổn hao bởi lượng tử - với các hệ số biến đổi 207.

Biến đổi ngược

Khối xử lý biến đổi ngược 212 được tạo cấu hình để áp dụng phép biến đổi ngược của phép biến đổi được áp dụng bởi khối xử lý biến đổi 206, chẳng hạn, DCT ngược hoặc DST ngược hoặc các phép biến đổi ngược khác, để thu được khối dư được tái tạo 213 (hoặc các hệ số lượng tử ngược 213 tương ứng) trong miền mẫu. Khối dư được tái tạo 213 có thể cũng được gọi là khối biến đổi 213.

Tái tạo

Khối tái tạo 214 (chẳng hạn, bộ cộng 214) được tạo cấu hình để bổ sung khối biến đổi 213 (tức là, khối dư được tái tạo 213) vào khối dự báo 265 để thu được khối được tái tạo 215 trong miền mẫu, chẳng hạn, bằng cách bổ sung – theo từng mẫu - các giá trị mẫu của khối dư được tái tạo 213 và các giá trị mẫu của khối dự báo 265.

Lọc

Khối lọc vòng 220 (hoặc gọi tắt là “lọc vòng” 220), được tạo cấu hình để lọc khối được tái tạo 215 để thu được khối được lọc 221, hoặc nói chung, để lọc các mẫu được tái tạo để thu được các giá trị mẫu được lọc. Khối lọc vòng, chẳng hạn, được tạo cấu hình để làm mượt các phép dịch chuyển pixel, hoặc ngược lại cải

thiện chất lượng video. Khối lọc vòng 220 có thể bao gồm một hoặc nhiều bộ lọc vòng chẳng hạn, bộ lọc tách khối (de-blocking filter, DBF), bộ lọc độ lệch mẫu thích ứng (sample-adaptive offset, SAO) hoặc một hoặc nhiều bộ lọc khác, chẳng hạn, bộ lọc vòng thích ứng (adaptive loop filter, ALF), bộ lọc chặn nhiễu (noise suppression filter, NSF), hoặc kết hợp bất kỳ của nó. Trong ví dụ, khối lọc vòng 220 có thể bao gồm DBF, bộ lọc SAO và bộ lọc ALF. Thứ tự của quá trình lọc có thể là DBF, SAO và ALF. Trong ví dụ khác, quá trình được gọi là ánh xạ độ sáng có phép chia tỷ lệ sắc độ (luma mapping with chroma scaling, LMCS) (tức là, bộ tạo dạng lại trong vòng thích ứng) được bổ sung. Quá trình này được thực hiện trước khi tách khối. Trong ví dụ khác, quá trình DBF cũng có thể được áp dụng cho các cạnh trong của khối phụ, chẳng hạn, các cạnh khối phụ afin, các cạnh khối phụ ATMVP, các cạnh biến đổi khối phụ (sub-block transform, SBT) và các cạnh phân vùng phụ bên trong (intra sub-partition, ISP). Mặc dù khối lọc vòng 220 được thể hiện trên Fig.2 như là bộ lọc trong vòng, trong các cấu hình khác, khối lọc vòng 220 có thể được triển khai dưới dạng hậu bộ lọc vòng post loop filter. Khối được lọc 221 có thể cũng được gọi là khối được tái tạo được lọc 221.

Theo các phương án thực hiện, bộ mã hóa video 20 (tương ứng là khối lọc vòng 220) có thể được tạo cấu hình để xuất ra các tham số lọc vòng (chẳng hạn, các tham số lọc SAO hoặc các tham số lọc ALF hoặc các tham số LMCS), chẳng hạn, trực tiếp hoặc được mã hóa qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã 30 có thể nhận và áp dụng các tham số lọc vòng giống nhau hoặc các bộ lọc vòng tương ứng để giải mã.

Bộ đệm ảnh được giải mã

DPB 230 có thể là bộ nhớ lưu trữ các ảnh tham chiếu, hoặc nói chung dữ liệu ảnh tham chiếu, để mã hóa dữ liệu video bằng bộ mã hóa video 20. DPB 230 có thể được tạo bởi bộ nhớ bất kỳ trong các thiết bị nhớ, chẳng hạn bộ nhớ truy nhập ngẫu nhiên động (dynamic random access memory, DRAM), bao gồm DRAM đồng bộ (synchronous DRAM, SDRAM), RAM từ trở (magnetoresistive RAM, MRAM), RAM điện trở (resistive RAM, RRAM), hoặc các loại bộ nhớ khác.

DPB 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. DPB 230 có thể còn được tạo cấu hình để lưu trữ các khối được lọc khác trước đó, chẳng hạn, các khối được lọc và được tái tạo trước đó 221, của cùng ảnh hiện tại hoặc của các ảnh khác nhau, chẳng hạn, các ảnh được tái tạo trước đó, và có thể tạo các ảnh hoàn chỉnh được tái tạo trước đó, tức là, được giải mã, (và các khối tham chiếu và mẫu tương ứng) và/hoặc ảnh hiện tại được tái tạo một phần (và các khối tham chiếu tương ứng và các mẫu), chẳng hạn để dự báo ngoài. DPB 230 có thể cũng được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái tạo không được lọc 215, hoặc nói chung các mẫu được tái tạo không được lọc, chẳng hạn, nếu khối được tái tạo 215 không được lọc bởi khối lọc vòng 220, hoặc bất kỳ phiên bản được xử lý khác nữa của các khối được tái tạo hoặc các mẫu.

Lựa chọn chế độ (Phân vùng & dự báo)

Khối chọn chế độ 260 bao gồm khối phân vùng 262, khối dự báo ngoài 244 và khối dự báo trong 254, và được tạo cấu hình để nhận dữ liệu ảnh gốc, chẳng hạn, khối gốc 203 (khối hiện tại 203 của ảnh hiện tại 17), và dữ liệu ảnh được tái tạo, chẳng hạn, các mẫu được tái tạo được lọc và/hoặc không được lọc hoặc các mẫu của cùng ảnh (hiện tại) và/hoặc từ một hoặc nhiều ảnh được giải mã trước đó, chẳng hạn, từ DPB 230 hoặc các bộ đệm khác (chẳng hạn, bộ đệm dòng, không được thể hiện trên hình vẽ). Dữ liệu ảnh được tái tạo được sử dụng làm dữ liệu ảnh tham chiếu để dự báo, chẳng hạn, dự báo ngoài hoặc dự báo trong, để thu được khối dự báo 265 hoặc bộ dự báo 265.

Khối chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc chọn phân vùng cho chế độ dự báo khối hiện tại (bao gồm không phân vùng) và chế độ dự báo (chẳng hạn, chế độ dự báo trong hoặc ngoài) và tạo khối dự báo tương ứng 265, mà được sử dụng để tính toán khối dư 205 và để tái tạo khối được tái tạo 215.

Theo các phương án thực hiện, khối chọn chế độ 260 có thể được tạo cấu hình để chọn chế độ phân vùng và dự báo (chẳng hạn, từ các chế độ được hỗ trợ hoặc khả dụng cho khối chọn chế độ 260), mà tạo so khớp tốt nhất hoặc nói cách khác, phần dư nhỏ nhất (phần dư nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc

lưu trữ), hoặc phụ tải báo hiệu nhỏ nhất (phụ tải báo hiệu nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Khối chọn chế độ 260 có thể được tạo cấu hình để xác định chế độ phân vùng và dự báo dựa trên tối ưu méo dạng tốc độ (rate distortion optimization, RDO), tức là, lựa chọn chế độ dự báo tạo méo dạng tốc độ nhỏ nhất. Các cụm từ như “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v.. trong ngữ cảnh này không nhất thiết đề cập đến toàn bộ “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. nhưng cũng có thể đề cập đến việc hoàn thành tiêu chí kết thúc hoặc lựa chọn như giá trị lớn hơn hoặc nhỏ hơn ngưỡng hoặc các ràng buộc khác có thể dẫn đến “lựa chọn tối ưu phụ” nhưng giảm độ phức tạp và thời gian xử lý.

Nói cách khác, khối phân vùng 262 có thể được tạo cấu hình để phân vùng ảnh từ chuỗi video thành chuỗi CTU, và CTU 203 có thể còn được phân vùng thành các phân vùng khối nhỏ hơn hoặc các khối phụ (lại tạo các khối), chẳng hạn, sử dụng lặp lại phân vùng cây tứ phân (quad-tree, QT), phân vùng cây nhị phân (binary tree, BT) hoặc phân vùng cây tam phân (triple-tree, TT) hoặc kết hợp bất kỳ của nó, và để thực hiện, chẳng hạn, dự báo cho mỗi trong số các phân vùng khối hoặc các khối phụ, trong đó việc chọn chế độ bao gồm chọn cấu trúc cây của khối được phân vùng 203 và các chế độ dự báo được áp dụng cho mỗi trong số các phân vùng khối hoặc các khối phụ.

Dưới đây, quá trình phân vùng (chẳng hạn, bằng khối phân vùng 260) và dự báo (bằng khối dự báo ngoài 244 và khối dự báo trong 254) được thực hiện bằng bộ mã hóa video 20 lấy làm ví dụ sẽ được giải thích chi tiết hơn.

Phân vùng

Khối phân vùng 262 có thể được tạo cấu hình để phân vùng ảnh từ chuỗi video thành chuỗi CTU, và khối phân vùng 262 có thể phân vùng (hoặc tách) CTU 203 thành các phân vùng nhỏ hơn, chẳng hạn, các khối nhỏ hơn có kích thước hình vuông hoặc chữ nhật. Đối với ảnh có ba mảng mẫu, CTU bao gồm khối mẫu độ sáng $N \times N$ cùng với hai khối mẫu sắc độ tương ứng. Kích thước lớn nhất được phép của khối độ sáng trong CTU được xác định bằng 128×128 theo chuẩn VVC đang phát triển, nhưng có thể được xác định bằng giá trị khác ngoài

128x128 trong tương lai, chẳng hạn, 256x256. Các CTU của ảnh có thể được tạo chùm/nhóm dưới dạng các lát/các nhóm ô, các ô hoặc các khối viên. Ô bao gồm vùng chữ nhật của ảnh, và ô có thể được phân chia thành một hoặc nhiều khối viên. Khối viên bao gồm nhiều hàng CTU trong ô. Ô không được phân vùng thành nhiều khối viên có thể được gọi là khối viên. Tuy nhiên, khối viên là tập con đúng của ô và không được gọi là ô. Có hai chế độ của các nhóm ô được hỗ trợ trong VVC, tức là, chế độ lát quét mảnh/nhóm ô và chế độ lát hình chữ nhật. Trong chế độ nhóm ô quét mảnh, nhóm lát/ô chứa chuỗi ô theo thứ tự quét mảnh ô của ảnh. Trong chế độ lát chữ nhật, lát chứa nhiều khối viên của ảnh tạo chung thành vùng chữ nhật của ảnh. Các khối viên trong lát chữ nhật theo thứ tự quét mảnh khối viên của lát. Các khối nhỏ hơn này (có thể cũng được gọi là các khối phụ) có thể còn được phân vùng thành thậm chí các phân vùng nhỏ hơn. Đây cũng được gọi là phân vùng cây hoặc phân vùng cây phân cấp, trong đó khối gốc, chẳng hạn, ở mức gốc cây 0 (mức phân cấp 0, chiều sâu 0), có thể được phân vùng đệ quy, chẳng hạn, được phân vùng thành hai hoặc nhiều khối của mức cấp thấp hơn tiếp theo, chẳng hạn, các nút ở mức cây 1 (mức phân cấp 1, chiều sâu 1), trong đó các khối này có thể lại được phân vùng thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, chẳng hạn, mức cây 2 (mức phân cấp 2, chiều sâu 2), v.v. cho đến khi kết thúc phân vùng, chẳng hạn, do tiêu chí kết thúc được thỏa mãn, chẳng hạn, đạt đến chiều sâu cây lớn nhất hoặc kích thước khối nhỏ nhất. Các khối không được phân vùng tiếp cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng phân vùng thành hai phân vùng được gọi là BT, cây sử dụng phân vùng thành ba phân vùng được gọi là TT, và cây sử dụng phân vùng thành bốn phân vùng được gọi là QT.

Chẳng hạn, CTU có thể là hoặc bao gồm CTB của các mẫu độ sáng, hai CTB tương ứng của các mẫu sắc độ của ảnh mà có ba mảng mẫu, hoặc CTB của các mẫu của ảnh đơn sắc hoặc ảnh được mã hóa nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu. Một cách tương ứng, CTB có thể là khối mẫu NxN cho giá trị bất kỳ của N sao cho phép phân chia thành phần thành các CTB gọi là phép phân vùng. Đơn vị mã (coding unit, CU) có thể là hoặc bao gồm khối mã của các mẫu độ sáng, hai khối mã tương

ứng của các mẫu sắc độ của ảnh có ba mảng mẫu, hoặc khối mã của các mẫu của ảnh đơn sắc hoặc ảnh được mã hóa nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để mã hóa các mẫu. Một cách tương ứng, CB có thể là khối mẫu $M \times N$ đối với một số giá trị M và N sao cho phép phân chia CTB thành các CB gọi là phép phân vùng.

Theo các phương án thực hiện, chẳng hạn, theo HEVC, CTU có thể được tách thành các CU nhờ sử dụng cấu trúc QT được ký hiệu là cây mã. Quyết định liệu có mã hóa diện tích ảnh nhờ sử dụng dự báo ngoài ảnh (thời gian) hoặc trong ảnh (không gian) được thực hiện ở mức CU lá. Mỗi CU lá có thể còn được tách thành một, hai hoặc bốn PU theo loại tách PU. Bên trong một PU, quá trình dự báo tương tự được áp dụng và thông tin liên quan được truyền đến bộ giải mã trên cơ sở PU. Sau khi thu được khối dư bằng cách áp dụng quá trình dự báo dựa trên loại tách PU, CU lá có thể được phân vùng thành các đơn vị biến đổi (transform unit, TU) theo cấu trúc QT khác tương tự như cây mã cho CU.

Theo các phương án thực hiện, chẳng hạn, theo chuẩn tạo mã video mới nhất đang được phát triển, được gọi là VVC, cây đa loại được lồng với cây tứ phân kết hợp nhờ sử dụng cấu trúc phân đoạn tách nhị phân và tam phân, chẳng hạn được sử dụng để phân vùng CTU. Trong cấu trúc cây mã bên trong CTU, CU có thể có hình vuông hoặc chữ nhật. Chẳng hạn, CTU trước hết được phân vùng bằng cây tứ phân. Sau đó, các nút lá cây tứ phân có thể còn được phân vùng bởi cấu trúc cây đa loại. Có bốn loại tách trong cấu trúc cây đa loại, tách nhị phân dọc (SPLIT_BT_VER), tách nhị phân ngang (SPLIT_BT_HOR), tách tam phân dọc (SPLIT_TT_VER), và tách tam phân ngang (SPLIT_TT_HOR). Các nút lá cây đa loại được gọi là các CU, và trừ khi CU quá lớn đối với chiều dài biến đổi lớn nhất, việc phân đoạn này được sử dụng để xử lý dự báo và biến đổi mà không phân vùng thêm. Điều này nghĩa là, trong phần lớn trường hợp, CU, PU và TU có cùng kích thước khối trong cây tứ phân với cấu trúc CB cây đa loại được lồng. Ngoại lệ xuất hiện khi chiều dài biến đổi lớn nhất được hỗ trợ nhỏ hơn chiều rộng hoặc chiều cao của thành phần màu của CU. VVC phát triển cơ chế báo hiệu duy nhất của thông tin tách phân vùng trong cấu trúc cây tứ phân có cây mã cây đa loại được lồng. Trong cơ chế báo hiệu, CTU được xử lý như là gốc cây

tứ phân và được phân vùng trước bởi cấu trúc QT. Sau đó, mỗi nút lá QT (khi đủ lớn để cho phép nó) được phân vùng tiếp bởi cấu trúc cây đa loại. Trong cấu trúc cây đa loại, cờ thứ nhất (`mtt_split_cu_flag`) được báo hiệu để chỉ báo liệu nút có được phân vùng tiếp hay không; khi nút được phân vùng tiếp, cờ thứ hai (`mtt_split_cu_vertical_flag`) được báo hiệu để chỉ báo hướng tách, và sau đó cờ thứ ba (`mtt_split_cu_binary_flag`) được báo hiệu để chỉ báo liệu phép tách này là tách nhị phân hoặc tách tam phân. Dựa trên các giá trị của `mtt_split_cu_vertical_flag` và `mtt_split_cu_binary_flag`, chế độ tách cây đa loại (`MttSplitMode`) của CU có thể được dẫn xuất bằng bộ giải mã dựa trên quy tắc hoặc bảng định trước. Cần lưu ý, đối với thiết kế cụ thể, chẳng hạn, khối độ sáng 64×64 và thiết kế tạo đường ống sắc độ 32×32 trong các bộ giải mã phân cứng VVC, phép tách TT bị cấm khi chiều rộng hoặc chiều cao của CB độ sáng lớn hơn 64, như được thể hiện trên Fig.6. Phép tách TT cũng bị cấm khi chiều rộng hoặc chiều cao của CB sắc độ lớn hơn 32. Thiết kế tạo đường ống sẽ phân chia ảnh thành các đơn vị dữ liệu đường ống ảo (Virtual pipeline data unit, VPDU) được định nghĩa là các đơn vị không chồng lấn trong ảnh. Trong các bộ giải mã phân cứng, các VPDU liên tiếp được xử lý đồng thời bởi nhiều giai đoạn đường ống. Kích thước VPDU gần như tỷ lệ thuận với kích thước bộ đệm trong phần lớn giai đoạn đường ống, do vậy, quan trọng là duy trì kích thước VPDU nhỏ. Trong phần lớn bộ giải mã phân cứng, kích thước VPDU có thể được gán bằng kích thước khối biến đổi (Transform Block, TB) lớn nhất. Tuy nhiên, trong VVC, phân vùng TT và BT có thể dẫn đến làm tăng kích thước VPDU. Ngoài ra, cần lưu ý rằng, khi phần của khối nút cây lớn hơn đường biên ảnh dưới hoặc bên phải, khối nút cây bị buộc phải tách cho đến khi tất cả các mẫu của mỗi CU được mã hóa được đặt bên trong các đường biên ảnh.

Lấy làm ví dụ, công cụ phân vùng phụ bên trong (Intra Sub-Partition, ISP) có thể phân chia các khối được dự báo trong theo chiều dọc hoặc chiều ngang thành 2 hoặc 4 phân vùng phụ tùy thuộc vào kích thước khối.

Trong một ví dụ, khối chọn chế độ 260 của bộ mã hóa video 20 có thể được tạo cấu hình để thực hiện kết hợp bất kỳ của các kỹ thuật phân vùng được mô tả ở đây.

Như được mô tả trên đây, bộ mã hóa video 20 được tạo cấu hình để xác định hoặc chọn chế độ dự báo tốt nhất hoặc tối ưu từ tập hợp chế độ dự báo (chẳng hạn, được định trước). Tập hợp chế độ dự báo có thể bao gồm, chẳng hạn, các chế độ dự báo trong và/hoặc các chế độ dự báo ngoài.

Dự báo trong

Tập hợp của các chế độ dự báo trong có thể bao gồm 35 chế độ dự báo trong khác nhau, chẳng hạn, các chế độ vô hướng giống như chế độ DC (hoặc trung bình) và chế độ phẳng, hoặc các chế độ định hướng, chẳng hạn, như được định nghĩa trong HEVC, hoặc có thể bao gồm 67 chế độ dự báo trong khác nhau, chẳng hạn, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ phẳng, hoặc các chế độ định hướng, chẳng hạn, như được định nghĩa cho VVC. Lấy làm ví dụ, vài chế độ dự báo trong góc đã biết được thay thế thích ứng bằng các chế độ dự báo trong góc rộng đối với các khối không vuông, chẳng hạn, như được định nghĩa trong VVC. Lấy làm ví dụ khác, để tránh các phép chia để dự báo DC, chỉ cạnh dài hơn được sử dụng để tính toán phép trung bình cho các khối không vuông. Và, các kết quả dự báo trong của chế độ phẳng có thể còn được chỉnh sửa nhờ phương pháp kết hợp dự báo trong phụ thuộc vị trí (position dependent intra prediction combination, PDPC).

Khối dự báo trong 254 được tạo cấu hình để sử dụng các mẫu được tái tạo của các khối lân cận của cùng ảnh hiện tại để tạo khối dự báo trong 265 theo chế độ dự báo trong của tập hợp chế độ dự báo trong.

Khối dự báo trong 254 (nói chung là khối chọn chế độ 260) còn được tạo cấu hình để xuất các tham số dự báo trong (hoặc nói chung thông tin chỉ báo chế độ dự báo trong được chọn cho khối) ra khối mã hóa entropy 270 ở dạng của các phần tử cú pháp 266 để đưa vào dữ liệu ảnh được mã hóa 21, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số dự báo để giải mã.

Dự báo ngoài

Tập hợp các chế độ dự báo ngoài (hoặc khả thi) phụ thuộc vào các ảnh tham chiếu khả dụng (tức là, các ảnh được giải mã ít nhất một phần trước đó, chẳng

hạn, được lưu trữ trong DBP 230) và các tham số dự báo ngoài khác, chẳng hạn, liệu toàn bộ ảnh tham chiếu hoặc chỉ một phần, chẳng hạn, khu vực cửa sổ tìm kiếm quanh khu vực của khối hiện tại, của ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu so khớp tốt nhất, và/hoặc chẳng hạn, liệu nội suy pixel có được áp dụng hay không, chẳng hạn, nội suy nửa/bán pel, $\frac{1}{4}$ pel và/hoặc $\frac{1}{16}$ pel.

Bên cạnh các chế độ dự báo nêu trên, chế độ bỏ qua, chế độ trực tiếp có thể được áp dụng.

Khối dự báo ngoài 244 có thể bao gồm khối ước tính chuyển động (motion estimation, ME) và khối bù chuyển động (motion compensation, MC) (cả hai không được thể hiện trên Fig.2). Khối ước tính chuyển động có thể được tạo cấu hình để nhận khối ảnh 203 (khối ảnh hiện tại 203 của ảnh hiện tại 17) và ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái tạo trước đó, chẳng hạn, các khối được tái tạo của một hoặc nhiều ảnh được giải mã trước đó 231 khác, để ước tính chuyển động. Chẳng hạn, chuỗi video có thể bao gồm the ảnh hiện tại và các ảnh được giải mã trước đó 231, hoặc nói cách khác, ảnh hiện tại và các ảnh được giải mã trước đó 231 có thể là một phần hoặc tạo thành chuỗi ảnh tạo chuỗi video.

Bộ mã hóa 20 có thể, chẳng hạn, được tạo cấu hình để chọn khối tham chiếu từ các khối tham chiếu của các ảnh giống hoặc khác nhau của các ảnh khác và tạo ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu) và/hoặc độ lệch (độ lệch không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện tại làm các tham số dự báo ngoài cho khối ước tính chuyển động. Độ lệch này cũng được gọi là vectơ chuyển động (motion vector, MV).

Khối bù chuyển động được tạo cấu hình để thu được, chẳng hạn, nhận, tham số dự báo ngoài và thực hiện dự báo ngoài dựa trên hoặc sử dụng tham số dự báo ngoài để thu được khối dự báo ngoài 265. Bù chuyển động, được thực hiện bằng khối bù chuyển động, có thể bao gồm việc nạp hoặc tạo khối dự báo dựa trên MV được xác định bằng cách ước tính chuyển động, có thể thực hiện các phép nội suy theo độ chính xác của pixel phụ. Bước lọc nội suy có thể tạo các mẫu pixel bổ sung từ các mẫu pixel đã biết, do vậy có thể tăng số lượng khối dự báo

ứng viên có thể được sử dụng để mã hóa khối ảnh. Khi nhận MV cho PU của khối ảnh hiện tại, khối bù chuyển động có thể định vị khối dự báo mà MV trỏ đến ở một trong các danh sách ảnh tham chiếu.

Khối bù chuyển động có thể cũng tạo các phần tử cú pháp được liên kết với các khối và các lát video để sử dụng bằng bộ giải mã video 30 khi giải mã các khối ảnh của lát video. Bên cạnh các lát và các phần tử cú pháp tương ứng, các nhóm ô và/hoặc các ô và các phần tử cú pháp tương ứng có thể được tạo hoặc được sử dụng.

Mã hóa Entropy

Khối mã hóa entropy 270 được tạo cấu hình để áp dụng, chẳng hạn, thuật toán hoặc phương tiện mã hóa entropy (chẳng hạn, phương tiện mã hóa chiều dài biến thiên (variable length coding, VLC), phương tiện VLC ngữ cảnh thích ứng (context adaptive VLC, CAVLC), phương tiện mã hóa số học, phép biến đổi nhị phân, mã hóa số học nhị phân ngữ cảnh thích ứng (context adaptive binary arithmetic coding, CABAC), mã hóa số học nhị phân ngữ cảnh thích ứng dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, SBAC), mã hóa entropy phân vùng khoảng xác suất (probability interval partitioning entropy, PIPE) hoặc phương pháp hoặc kỹ thuật mã hóa entropy khác) hoặc đi vòng (không nén) trên các hệ số lượng tử 209, tham số dự báo ngoài, tham số dự báo trong, tham số lọc vòng và/hoặc các phần tử cú pháp khác để thu được dữ liệu ảnh được mã hóa 21 có thể được xuất ra qua đầu ra 272, chẳng hạn, ở dạng dòng bit được mã hóa 21, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và sử dụng các tham số để giải mã. Dòng bit được mã hóa 21 có thể được truyền đến bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ để truyền hoặc truy xuất sau bởi bộ giải mã video 30.

Các biến thể cấu trúc khác của bộ mã hóa video 20 có thể được sử dụng để mã hóa dòng video. Chẳng hạn, bộ mã hóa không dựa trên biến đổi 20 có thể lượng tử trực tiếp tín hiệu dư không qua khối xử lý biến đổi 206 đối với các khối hoặc các khung cụ thể. Theo triển khai khác, bộ mã hóa 20 có thể kết hợp khối lượng tử 208 và khối lượng tử ngược 210 thành một khối.

Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện ví dụ về bộ giải mã video 30 được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Bộ giải mã video 30 được tạo cấu hình để nhận dữ liệu ảnh được mã hóa 21 (chẳng hạn, dòng bit được mã hóa 21), chẳng hạn, được mã hóa bằng bộ mã hóa 20, để thu được ảnh được giải mã 331. Dữ liệu ảnh được mã hóa hoặc dòng bit bao gồm thông tin để giải mã dữ liệu ảnh được mã hóa, chẳng hạn, dữ liệu biểu diễn các khối ảnh của lát video được mã hóa (và/hoặc các nhóm ô hoặc các ô) và các phân tử cú pháp liên kết.

Trong ví dụ trên Fig.3, bộ giải mã 30 bao gồm khối giải mã entropy 304, khối lượng tử ngược 310, khối xử lý biến đổi ngược 312, khối tái tạo 314 (chẳng hạn, bộ cộng 314), bộ lọc vòng 320, DBP 330, khối áp dụng chế độ 360, khối dự báo ngoài 344 và khối dự báo trong 354. Khối dự báo ngoài 344 có thể là hoặc bao gồm khối bù chuyển động. Bộ giải mã video 30 có thể, trong một số ví dụ, thực hiện bước giải mã nói chung nghịch đảo với bước mã hóa được mô tả với bộ mã hóa video 100 từ Fig.2.

Như được giải thích với bộ mã hóa 20, khối lượng tử ngược 210, khối xử lý biến đổi ngược 212, khối tái tạo 214, bộ lọc vòng 220, DPB 230, khối dự báo ngoài 344 và khối dự báo trong 354 cũng được gọi là tạo “bộ giải mã được cài sẵn” của bộ mã hóa video 20. Do đó, khối lượng tử ngược 310 có thể có chức năng giống hệt khối lượng tử ngược 110, khối xử lý biến đổi ngược 312 có thể có chức năng giống hệt khối xử lý biến đổi ngược 212, khối tái tạo 314 có thể có chức năng giống hệt khối tái tạo 214, bộ lọc vòng 320 có thể có chức năng giống hệt bộ lọc vòng 220, và DPB 330 có thể có chức năng giống hệt DPB 230. Do vậy, các giải thích đối với các khối và các chức năng tương ứng của bộ mã hóa video 20 áp dụng một cách tương ứng cho các khối và các chức năng tương ứng của bộ giải mã video 30.

Giải mã Entropy

Khối giải mã entropy 304 được tạo cấu hình để phân tách dòng bit 21 (hoặc nói chung dữ liệu ảnh được mã hóa 21) và thực hiện, chẳng hạn, giải mã entropy cho dữ liệu ảnh được mã hóa 21 để thu được, chẳng hạn, các hệ số lượng tử 309

và/hoặc các tham số mã được giải mã (không được thể hiện trên Fig.3), chẳng hạn, bất kỳ hoặc tất cả các tham số dự báo ngoài (chẳng hạn, chỉ số ảnh tham chiếu và MV), tham số dự báo trong (chẳng hạn, chế độ hoặc chỉ số dự báo trong), các tham số biến đổi, các QP, các tham số lọc vòng, và/hoặc các phần tử cú pháp khác. Khối giải mã entropy 304 có thể được tạo cấu hình để áp dụng các thuật toán hoặc các phương tiện giải mã tương ứng với các phương tiện mã hóa như được mô tả với khối mã hóa entropy 270 của bộ mã hóa 20. Khối giải mã entropy 304 có thể còn được tạo cấu hình để tạo các tham số dự báo ngoài, tham số dự báo trong và/hoặc các phần tử cú pháp khác cho khối áp dụng chế độ 360 và các tham số khác cho các đơn vị khác của bộ giải mã 30. Bộ giải mã video 30 có thể nhận các phần tử cú pháp ở mức lát video và/hoặc mức khối video. Bên cạnh các lát và các phần tử cú pháp tương ứng, các nhóm ô và/hoặc các ô và các phần tử cú pháp tương ứng có thể được nhận và/hoặc được sử dụng.

Lượng tử ngược

Khối lượng tử ngược 310 có thể được tạo cấu hình để nhận các QP (hoặc nói chung thông tin liên quan đến lượng tử ngược) và các hệ số lượng tử từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bằng khối giải mã entropy 304) và dựa trên các QP để áp dụng lượng tử ngược trên các hệ số lượng tử được giải mã 309 để thu được các hệ số lượng tử ngược 311, có thể cũng được gọi là các hệ số biến đổi 311. Quá trình lượng tử ngược có thể bao gồm sử dụng QP được xác định bởi bộ mã hóa video 20 cho mỗi khối video trong lát video (hoặc ô hoặc nhóm ô) để xác định mức độ lượng tử và, tương tự, mức độ lượng tử ngược cần được áp dụng.

Biến đổi ngược

Khối xử lý biến đổi ngược 312 có thể được tạo cấu hình để nhận các hệ số lượng tử ngược 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng biến đổi cho các hệ số lượng tử ngược 311 để thu được các khối dư được tái tạo 213 trong miền mẫu. Các khối dư được tái tạo 213 có thể cũng được gọi là các khối biến đổi 313. Phép biến đổi có thể là biến đổi ngược, chẳng hạn, DCT ngược,

DST ngược, phép biến đổi số nguyên ngược, hoặc quá trình biến đổi ngược tương tự về mặt khái niệm. Khối xử lý biến đổi ngược 312 có thể còn được tạo cấu hình để nhận các tham số biến đổi hoặc thông tin tương ứng từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bằng khối giải mã entropy 304) để xác định phép biến đổi được áp dụng cho các hệ số lượng tử ngược 311.

Tái tạo

Khối tái tạo 314 (chẳng hạn, bộ cộng 314) có thể được tạo cấu hình để bổ sung khối dư được tái tạo 313, cho khối dự báo 365 để thu được khối được tái tạo 315 trong miền mẫu, chẳng hạn, bằng cách bổ sung các giá trị mẫu của khối dư được tái tạo 313 và các giá trị mẫu của khối dự báo 365.

Lọc

Khối lọc vòng 320 (hoặc trong vòng mã hoặc sau vòng mã) được tạo cấu hình để lọc khối được tái tạo 315 để thu được khối được lọc 321, chẳng hạn, để làm mượt các phép dịch chuyển pixel, hoặc ngược lại cải thiện chất lượng video. Khối lọc vòng 320 có thể bao gồm một hoặc nhiều bộ lọc vòng chẳng hạn, DBF, bộ lọc SAO hoặc một hoặc nhiều bộ lọc khác, chẳng hạn, ALF, NSF, hoặc kết hợp bất kỳ của nó. Trong ví dụ, khối lọc vòng 220 có thể bao gồm DBF, bộ lọc SAO và ALF. Thứ tự của quá trình lọc có thể là DBF, SAO và ALF. Trong ví dụ khác, quá trình được gọi là LMCS (tức là, bộ tạo dạng lại trong vòng thích ứng) được bổ sung. Quá trình này được thực hiện trước khi tách khối. Trong ví dụ khác, quá trình DBF cũng có thể được áp dụng cho các cạnh trong của khối phụ, chẳng hạn, các cạnh khối phụ afin, các cạnh khối phụ ATMVP, các cạnh SBT và các cạnh ISP. Mặc dù khối lọc vòng 320 được thể hiện trên Fig.3 như là bộ lọc trong vòng, trong các cấu hình khác, khối lọc vòng 320 có thể được triển khai dưới dạng hậu lọc vòng.

Bộ đệm ảnh được giải mã

Sau đó, các khối video được giải mã 321 của ảnh được lưu trữ trong DPB 330, lưu trữ các ảnh được giải mã 331 dưới dạng các ảnh tham chiếu để bù chuyển động tiếp theo cho các ảnh khác và/hoặc để hiển thị đầu ra tương ứng.

Bộ giải mã 30 được tạo cấu hình để xuất ra ảnh được giải mã 311, chẳng hạn, qua đầu ra 312, để hiển thị hoặc trình chiếu đến người dùng.

Dự báo

Khối dự báo ngoài 344 có thể có chức năng giống như khối dự báo ngoài 244 (cụ thể là với khối bù chuyển động) và khối dự báo trong 354 có thể có chức năng giống như khối dự báo ngoài 254, và thực hiện các quyết định tách hoặc phân vùng và dự báo dựa trên các tham số phân vùng và/hoặc dự báo hoặc thông tin tương ứng được nhận từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bằng khối giải mã entropy 304). Khối áp dụng chế độ 360 có thể được tạo cấu hình để thực hiện dự báo (dự báo trong hoặc ngoài) theo khối dựa trên các ảnh được tái tạo, các khối hoặc các mẫu tương ứng (được lọc hoặc không được lọc) để thu được khối dự báo 365.

Khi lát video được mã hóa dưới dạng lát được mã hóa trong (I), khối dự báo trong 354 của khối áp dụng chế độ 360 được tạo cấu hình để tạo khối dự báo 365 cho khối ảnh của lát video hiện tại dựa trên chế độ dự báo trong được báo hiệu và dữ liệu từ các khối được giải mã trước đó của ảnh hiện tại. Khi ảnh video được mã hóa dưới dạng lát được mã hóa ngoài (tức là, B, hoặc P), khối dự báo ngoài 344 (chẳng hạn, khối bù chuyển động) của khối áp dụng chế độ 360 được tạo cấu hình để tạo các khối dự báo 365 cho khối video của lát video hiện tại dựa trên các MV và các phần tử cú pháp khác được nhận từ khối giải mã entropy 304. Để dự báo ngoài, các khối dự báo có thể được tạo từ một trong các ảnh tham chiếu trong một trong các danh sách ảnh tham chiếu. Bộ giải mã video 30 có thể tạo các danh sách khung tham chiếu, List 0 và List 1, nhờ sử dụng các kỹ thuật tạo mặc định dựa trên các ảnh tham chiếu được lưu trữ trong DPB 330. Các kỹ thuật tương tự có thể được áp dụng cho hoặc bằng các phương án thực hiện nhờ sử dụng các nhóm ô (chẳng hạn, các nhóm ô video) và/hoặc các ô (chẳng hạn,

các ô video) bên cạnh các lát (chẳng hạn, các lát video), chẳng hạn, video có thể được mã hóa nhờ sử dụng các nhóm ô và/hoặc các ô I, P hoặc B.

Khối áp dụng chế độ 360 được tạo cấu hình để xác định thông tin dự báo cho khối video của lát video hiện tại bằng cách phân tách các MV hoặc thông tin liên quan hoặc các phần tử cú pháp khác, và sử dụng thông tin dự báo để tạo các khối dự báo cho khối video hiện tại đang được giải mã. Chẳng hạn, khối áp dụng chế độ 360 sử dụng một số phần tử cú pháp được nhận để xác định chế độ dự báo (chẳng hạn, dự báo trong hoặc ngoài) được sử dụng để mã hóa các khối video của lát video, loại lát dự báo ngoài (chẳng hạn, lát B, lát P, hoặc lát GPB), thông tin xây dựng cho một hoặc nhiều danh sách ảnh tham chiếu cho lát, các Mv cho mỗi khối video được mã hóa ngoài của lát, trạng thái dự báo ngoài cho mỗi khối video được mã hóa ngoài của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại. Kỹ thuật tương tự hoặc giống hệt có thể được áp dụng cho hoặc bằng các phương án thực hiện nhờ sử dụng các nhóm ô (chẳng hạn, các nhóm ô video) và/hoặc các ô (chẳng hạn, các ô video) bên cạnh các lát (chẳng hạn, các lát video), chẳng hạn, video có thể được tạo mã nhờ sử dụng các nhóm ô và/hoặc các ô I, P hoặc B.

Theo các phương án thực hiện, bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân vùng và/hoặc giải mã ảnh nhờ sử dụng các lát (cũng được gọi là các lát video), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát (thường là không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (chẳng hạn, các CTU) hoặc một hoặc nhiều nhóm khối (chẳng hạn, các ô (H.265/HEVC và VVC) hoặc các khối viên (VVC)).

Theo các phương án thực hiện, bộ giải mã video 30 như được thể hiện trên Fig. 3 có thể được tạo cấu hình để phân vùng và/hoặc giải mã ảnh nhờ sử dụng các nhóm lát/ô (cũng được gọi là các nhóm ô video) và/hoặc các ô (cũng được gọi là các ô video), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát/nhóm ô (thường là không chồng lấn), và mỗi lát/nhóm ô có thể bao gồm, chẳng hạn, một hoặc nhiều khối (chẳng hạn, CTU) hoặc một hoặc nhiều ô, trong đó mỗi ô, chẳng hạn, có thể có hình chữ nhật và có

thể bao gồm một hoặc nhiều khối (chẳng hạn, CTU), chẳng hạn, các khối hoàn chỉnh hoặc một phần khối.

Các biến thể khác của bộ giải mã video 30 có thể được sử dụng để giải mã dữ liệu ảnh được mã hóa 21. Chẳng hạn, bộ giải mã 30 có thể tạo dòng video đầu ra không qua khối lọc vòng 320. Chẳng hạn, bộ giải mã không dựa trên biến đổi 30 có thể lượng tử ngược tín hiệu dư trực tiếp không qua khối xử lý biến đổi ngược 312 cho các khối hoặc khung cụ thể. Theo cách thực hiện khác, bộ giải mã video 30 có thể kết hợp khối lượng tử ngược 310 và khối xử lý biến đổi ngược 312 thành một khối.

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại có thể còn được xử lý và sau đó được xuất ra bước tiếp theo. Chẳng hạn, sau khi lọc nội suy, dẫn xuất MV hoặc lọc vòng, hoạt động khác, chẳng hạn, cắt xén hoặc dịch, có thể được thực hiện trên kết quả xử lý lọc nội suy, dẫn xuất MV hoặc lọc vòng.

Cần lưu ý rằng các hoạt động khác có thể được áp dụng cho các MV được dẫn xuất của khối hiện tại (bao gồm mà không giới hạn ở các MV điểm điều khiển (control point motion vector, CPMV) của chế độ afin, các MV của khối phụ trong chế độ afin, chế độ phẳng, ATMVP, các MV thời gian, và v.v.). Chẳng hạn, giá trị của MV bị giới hạn ở khoảng định trước theo bit biểu diễn của nó. Nếu bit biểu diễn của MV là bitDepth, thì khoảng này là từ $-2^{(\text{bitDepth}-1)}$ đến $2^{(\text{bitDepth}-1)}-1$, trong đó “^” nghĩa là nâng lũy thừa. Chẳng hạn, nếu bitDepth được gán bằng 16, khoảng này là từ -32768 đến 32767; nếu bitDepth được gán bằng 18, khoảng này là từ -131072 đến 131071. Chẳng hạn, giá trị của MV được dẫn xuất (chẳng hạn, các MV của bốn khối phụ 4x4 trong một khối 8x8) bị giới hạn sao cho hiệu số lớn nhất giữa các phần số nguyên của bốn MV khối phụ 4x4 không lớn hơn N pixel, chẳng hạn không lớn hơn 1 pixel. Ở đây đề xuất hai phương pháp giới hạn MV theo bitDepth.

Fig.4 là sơ đồ thiết bị tạo mã video 400 theo phương án thực hiện sáng chế. Thiết bị tạo mã video 400 thích hợp để thực hiện các phương án thực hiện được bộc lộ như được mô tả ở đây. Theo phương án thực hiện, thiết bị tạo mã video

400 có thể là bộ giải mã, chẳng hạn bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa, chẳng hạn bộ mã hóa video 20 trên Fig.1A.

Thiết bị tạo mã video 400 bao gồm các cổng nạp 410 (hoặc cổng vào 410) và các bộ nhận (Rx) 420 để nhận dữ liệu; bộ xử lý, đơn vị logic, hoặc khối xử lý trung tâm (central processing unit, CPU) 430 để xử lý dữ liệu; các bộ truyền (Tx) 440 và các cổng ra 450 (hoặc các cổng xuất 450) để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị tạo mã video 400 có thể cũng bao gồm các thành phần quang sang điện (optical-to-electrical, OE) và các thành phần điện sang quang (electrical-to-optical, EO) được ghép nối với các cổng nạp 410, các Rx 420, các Tx 440, và các cổng xuất 450 để xuất hoặc nạp các tín hiệu quang hoặc điện.

Bộ xử lý 430 được triển khai bằng phần cứng và phần mềm. Bộ xử lý 430 có thể được triển khai dưới dạng một hoặc nhiều chip CPU, lõi (chẳng hạn, dưới dạng bộ xử lý nhiều lõi), FPGA, ASIC, và DSP. Bộ xử lý 430 đang truyền thông với các cổng vào 410, Rx 420, Tx 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun tạo mã 470. Môđun tạo mã 470 thực hiện các phương án thực hiện được bộc lộ trên đây. Chẳng hạn, môđun tạo mã 470 thực hiện, xử lý, chuẩn bị, hoặc tạo các hoạt động tạo mã khác nhau. Do vậy, việc đưa vào môđun tạo mã 470 cải thiện đáng kể chức năng của thiết bị tạo mã video 400 và biến đổi thiết bị tạo mã video 400 sang trạng thái khác. Theo cách khác, môđun tạo mã 470 được triển khai dưới dạng các lệnh được lưu trữ trong bộ nhớ 460 và được thực thi bằng bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, ổ băng, và ổ trạng thái rắn (solid-state drive, SSD) và có thể được sử dụng dưới dạng thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình này được chọn để thực thi, và để lưu trữ các lệnh và dữ liệu được đọc khi thực thi chương trình. Bộ nhớ 460 có thể là, chẳng hạn, khá biến và/hoặc bất biến và có thể là bộ nhớ chỉ đọc (read-only memory, ROM), bộ nhớ truy nhập ngẫu nhiên (random access memory, RAM), bộ nhớ có nội dung đánh địa chỉ được tam phân (ternary content-addressable memory, TCAM), và/hoặc RAM tĩnh (static random-access memory, SRAM).

Fig. 5 là sơ đồ khối được đơn giản hóa của thiết bị 500 có thể được sử dụng dưới dạng một trong hoặc cả thiết bị nguồn 12 lần thiết bị đích 14 từ Fig.1 theo phương án thực hiện lấy làm ví dụ.

Bộ xử lý 502 trong thiết bị 500 có thể là CPU. Theo cách khác, bộ xử lý 502 có thể là loại khác bất kỳ của thiết bị, hoặc nhiều thiết bị, có thể thao tác hoặc xử lý thông tin hiện có hoặc được phát triển sau này. Mặc dù các cách triển khai được bộc lộ có thể được thực hiện với một bộ xử lý như được thể hiện trên hình vẽ, chẳng hạn, bộ xử lý 502, các ưu điểm về tốc độ và hiệu suất có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong thiết bị 500 có thể là ROM hoặc RAM theo cách thức thực hiện. Loại thích hợp khác bất kỳ của thiết bị lưu trữ có thể được sử dụng làm bộ nhớ 504. Bộ nhớ 504 có thể bao gồm mã và dữ liệu 506 được truy nhập bằng bộ xử lý 502 nhờ sử dụng buýt 512. Bộ nhớ 504 có thể còn bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình cho phép bộ xử lý 502 thực hiện các phương pháp được mô tả ở đây. Chẳng hạn, các chương trình ứng dụng 510 có thể bao gồm các ứng dụng từ 1 đến N, mà còn bao gồm ứng dụng tạo mã video thực hiện các phương pháp được mô tả ở đây.

Thiết bị 500 cũng có thể bao gồm một hoặc nhiều bộ phận xuất ra, chẳng hạn bộ phận hiển thị 518. Bộ phận hiển thị 518 có thể là, trong một ví dụ, bộ phận hiển thị cảm ứng kết hợp bộ phận hiển thị với phần tử cảm ứng có thể hoạt động để cảm nhận đầu vào tiếp xúc. Bộ phận hiển thị 518 có thể được ghép nối với bộ xử lý 502 qua buýt 512.

Mặc dù được mô tả ở đây dưới dạng một buýt, buýt 512 của thiết bị 500 có thể bao gồm nhiều buýt. Ngoài ra, bộ phận lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các thành phần khác của thiết bị 500 hoặc có thể được truy nhập qua mạng và có thể bao gồm một đơn vị tích hợp, chẳng hạn thẻ nhớ hoặc nhiều đơn vị, chẳng hạn nhiều thẻ nhớ. Do vậy, thiết bị 500 có thể được triển khai trong các loại cấu hình.

Một số ví dụ liên quan như sau, và cần lưu ý rằng tài liệu hội nghị JVET JVET-N1002 được sử dụng làm tham chiếu.

I. Dự báo hợp nhất:

Danh sách ứng viên hợp nhất được tạo bằng cách bao gồm năm loại ứng viên theo thứ tự sau:

- 1) Bộ dự báo MV (Motion Vector Predictor, MVP) không gian từ các CU lân cận không gian
- 2) MVP thời gian từ các CU cùng vị trí
- 3) MVP dựa trên lịch sử từ bảng FIFO
- 4) MVP trung bình theo cặp
- 5) Các MV số 0.

Kích thước của danh sách hợp nhất được báo hiệu trong tiêu đề lát và kích thước lớn nhất được phép của danh sách hợp nhất là số nguyên, chẳng hạn số ngày có thể bằng 6. Đối với mỗi mã CU trong chế độ hợp nhất, chỉ số của ứng viên hợp nhất tốt nhất được mã hóa. Quá trình tạo của mỗi phân loại ứng viên hợp nhất được tạo trong phần này.

(1) Dẫn xuất các ứng viên không gian

Lấy làm ví dụ, tối đa bốn ứng viên hợp nhất được chọn trong số các ứng viên được đặt ở các vị trí được mô tả trên Fig.6. Thứ tự dẫn xuất là A0, B0, B1, A1 và B2. Vị trí B2 được xem xét chỉ khi CU bất kỳ của vị trí A0, B0, B1, A1 không khả dụng (chẳng hạn, do nó thuộc lát hoặc ô khác) hoặc được mã hóa trong. Sau khi ứng viên ở vị trí A1 được bổ sung, việc bổ sung các ứng viên còn lại phụ thuộc vào kiểm tra dư mà đảm bảo rằng các ứng viên có cùng thông tin chuyển động bị loại bỏ khỏi danh sách. Theo cách triển khai linh hoạt, chỉ các cặp được liên kết với mũi tên trên Fig.7 được xem xét và ứng viên chỉ được bổ sung vào danh sách nếu ứng viên tương ứng được sử dụng để kiểm tra dư thừa không có cùng thông tin chuyển động.

(2) Dẫn xuất các ứng viên thời gian

Lấy làm ví dụ, chỉ một ứng viên được bổ sung vào danh sách. Cụ thể là, khi dẫn xuất ứng viên hợp nhất thời gian này, MV được chia tỷ lệ được dẫn xuất dựa trên CU cùng vị trí thuộc ảnh tham chiếu cùng vị trí. Danh sách ảnh tham chiếu cần được sử dụng để dẫn xuất CU cùng vị trí được báo hiệu tường minh trong tiêu đề lát. MV được chia tỷ lệ đối với ứng viên hợp nhất thời gian thu được như

được minh họa bằng đường nét đứt trên Fig.8, mà được chia tỷ lệ từ MV của CU cùng vị trí nhờ sử dụng các khoảng cách số đếm thứ tự ảnh (Picture Order Count, POC), tb và td, trong đó tb được định nghĩa là hiệu số POC giữa ảnh tham chiếu của ảnh hiện tại và ảnh hiện tại và td được định nghĩa là hiệu số POC giữa ảnh tham chiếu của ảnh cùng vị trí và ảnh cùng vị trí. Chỉ số ảnh tham chiếu của ứng viên hợp nhất thời gian được gán bằng 0.

Cần lưu ý rằng POC nghĩa là biến được liên kết với mỗi ảnh, nhận diện duy nhất ảnh được liên kết trong số tất cả các ảnh trong chuỗi video được mã hóa (Coded Video Sequence, CVS), và, khi ảnh được liên kết cần được xuất ra từ DPB, chỉ báo vị trí của ảnh được liên kết theo thứ tự đầu ra so với các vị trí thứ tự đầu ra của các ảnh khác trong cùng CVS được xuất ra từ DPB.

(3) Dẫn xuất ứng viên hợp nhất dựa trên lịch sử

Các ứng viên hợp nhất MVP dựa trên lịch sử (history-based MVP, HMVP) được bổ sung vào danh sách hợp nhất sau MVP không gian và TMVP. Thông tin chuyển động của khối được mã hóa trước đó được lưu trữ trong bảng và được sử dụng làm MVP cho CU hiện tại. Bảng có nhiều ứng viên HMVP được duy trì trong quá trình mã hóa/giải mã. Bảng được thiết lập lại khi gặp hàng CTU mới. Bất kỳ khi nào có CU được mã hóa ngoài không khối phụ, thông tin chuyển động liên kết được bổ sung vào mục cuối của bảng làm ứng viên HMVP mới. Vài ứng viên HMVP mới nhất trong bảng này được kiểm tra theo thứ tự và được chèn vào danh sách ứng viên sau ứng viên TMVP. Phép kiểm tra dư được áp dụng lên các ứng viên HMVP cho ứng viên hợp nhất không gian và thời gian.

(4) Dẫn xuất các ứng viên hợp nhất trung bình theo cặp

Các ứng viên trung bình theo cặp được tạo bằng cách tính trung bình các cặp ứng viên định trước trong danh sách ứng viên hợp nhất hiện tại, và các cặp định trước được định nghĩa là $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)\}$, trong đó các số ký hiệu các chỉ số hợp nhất cho danh sách ứng viên hợp nhất. Các MV được tính trung bình được tính toán riêng rẽ cho mỗi danh sách tham chiếu. Nếu cả hai MV khả dụng trong một danh sách, hai MV này được tính trung bình thậm chí khi chúng trỏ đến các ảnh tham chiếu khác nhau; nếu chỉ một MV khả dụng,

sử dụng ngay MV này; nếu không MV nào khả dụng, duy trì danh sách này là không hợp lệ.

(5) Khi danh sách hợp nhất không đầy sau khi các ứng viên hợp nhất trung bình theo cặp được bổ sung, các MVP số 0 được chèn vào cuối cho đến khi đạt số lượng ứng viên hợp nhất lớn nhất.

II. Dự báo kép

Dự báo ngoài ảnh sử dụng tương quan thời gian giữa các ảnh để dẫn xuất dự báo bù chuyển động (motion-compensated prediction, MCP) cho khối mẫu ảnh. Đối với mỗi khối, khối tương ứng trong ảnh được giải mã trước đó có thể được tìm thấy dùng làm bộ dự báo. Vị trí của khối trong ảnh được giải mã trước đó được chỉ báo bằng MV (Δx , Δy) trong đó Δx xác định dịch chuyển ngang và Δy xác định dịch chuyển dọc so với vị trí của khối hiện tại. Các MV (Δx , Δy) có thể có độ chính xác mẫu phân số để chụp chính xác hơn chuyển động của đối tượng nằm dưới. Nội suy được áp dụng trên các ảnh tham chiếu để dẫn xuất tín hiệu dự báo khi MV tương ứng có độ chính xác mẫu phân số. Ảnh được giải mã trước đó được gọi là ảnh tham chiếu và được chỉ báo bằng chỉ số tham chiếu Δt đến danh sách ảnh tham chiếu.

Trong trường hợp dự báo kép (so với dự báo đơn sử dụng một tập hợp dữ liệu chuyển động), hai tập hợp dữ liệu chuyển động (Δx_0 , Δy_0 , Δt_0 và Δx_1 , Δy_1 , Δt_1) được sử dụng để tạo hai MCP (có thể từ các ảnh khác nhau hoặc ảnh giống nhau), mà sau đó được kết hợp để có MCP cuối cùng. Nói chung, nó được dẫn xuất bằng cách tính trung bình. Trong trường hợp dự báo có trọng số, các trọng số khác nhau có thể được áp dụng cho mỗi MCP. Các ảnh tham chiếu có thể được sử dụng trong dự báo kép được lưu trữ trong hai danh sách riêng rẽ, tức là list 0 và list 1.

III. Lưu trữ thông tin chuyển động

Sau khi thực hiện bù chuyển động, thông tin chuyển động (dự báo đơn hoặc dự báo kép) được liên kết với khối dự báo được lưu trữ trong bộ nhớ. Thông tin chuyển động được lưu trữ được sử dụng làm thông tin tham chiếu của khối tương lai (CU/PU), chẳng hạn qua dự báo MV không gian hoặc thời gian.

Trong một số ví dụ, việc lưu trữ thông tin chuyển động được thực hiện nhờ sử dụng mảng hai chiều của các thông tin chuyển động khối phụ hoặc các khối lưu trữ thông tin chuyển động. Độ phân giải không gian của mảng hai chiều có thể khác với độ phân giải không gian của ảnh cần được tạo mã. Trong ví dụ, độ phân giải thấp hơn của đơn vị lưu trữ thông tin chuyển động được sử dụng để tiết kiệm không gian bộ nhớ. Chẳng hạn, các mẫu độ sáng 16×16 tương ứng với một đơn vị lưu trữ thông tin chuyển động, nghĩa là mỗi mẫu trong các mẫu độ sáng 16×16 có cùng thông tin chuyển động và cùng thông tin chuyển động được lưu trữ trong một khối lưu trữ thông tin chuyển động tương ứng với các mẫu độ sáng 16×16 này. Nói cách khác, đơn vị lưu trữ thông tin chuyển động chứa thông tin dự báo ngoài ảnh của khối dự báo được liên kết, bao gồm các mẫu độ sáng 16×16 .

Và cần lưu ý rằng các mẫu độ sáng 16×16 nghĩa là tập hợp mẫu độ sáng vốn là mảng mẫu hai chiều, và cả chiều rộng và chiều cao của mảng mẫu hai chiều bằng 16. Một khối lưu trữ thông tin chuyển động cũng có thể tương ứng với các mẫu độ sáng 8×8 , các mẫu độ sáng 4×4 hoặc các mẫu độ sáng $N \times M$ khác hoặc các mẫu sắc độ, N và M là các số nguyên dương. Và có thể hiểu rằng kích thước nhỏ hơn của khối lưu trữ thông tin chuyển động cho phép dự báo MV tốt hơn, nhưng tiêu tốn nhiều không gian nhớ hơn. Lưu trữ vector chuyển động đối với khối dự báo có các mẫu độ sáng kích thước 8×8 được lấy ví dụ trên Fig.9. Thông tin chuyển động của khối dự báo, được dán nhãn $Mv0$, được lưu trữ trong mỗi đơn vị lưu trữ MV 2×2 .

III. Phân vùng tam giác để dự báo ngoài

Lấy làm ví dụ, chế độ phân vùng tam giác chỉ được áp dụng cho các CU 8×8 hoặc lớn hơn và được mã hóa trong chế độ bỏ qua hoặc hợp nhất. Cờ mức CU được báo hiệu để chỉ báo liệu chế độ phân vùng tam giác có được áp dụng hay không.

CU trong chế độ phân vùng tam giác được tách đều thành hai phân vùng hình tam giác, nhờ sử dụng tách đường chéo hoặc tách phi đường chéo (Fig.10). Mỗi phân vùng tam giác trong CU được dự báo ngoài nhờ sử dụng chuyển động riêng;

chỉ dự báo đơn được phép cho mỗi phân vùng, tức là, mỗi phân vùng có một MV và một chỉ số tham chiếu. Ràng buộc chuyển động dự báo đơn được áp dụng để đảm bảo rằng giống như dự báo kép truyền thống, chỉ hai dự báo bù chuyển động cần cho mỗi CU. Chuyển động dự báo đơn cho mỗi phân vùng được dẫn xuất trực tiếp từ danh sách ứng viên hợp nhất được tạo cho dự báo hợp nhất nêu trên, và lựa chọn chuyển động dự báo đơn từ ứng viên hợp nhất cụ thể trong danh sách là theo thủ tục “tạo danh sách ứng viên dự báo đơn”. Sau khi dự báo mỗi phân vùng trong các phân vùng tam giác, các giá trị mẫu dọc theo cạnh đường chéo hoặc phi đường chéo được điều chỉnh nhờ sử dụng quá trình kết hợp với các trọng số thích ứng. Cuối cùng, trường chuyển động của CU được dự báo nhờ sử dụng chế độ phân vùng tam giác được lưu trữ.

(1) Tạo danh sách ứng viên dự báo đơn

Với chỉ số ứng viên hợp nhất, MV dự báo đơn được dẫn xuất từ danh sách ứng viên hợp nhất được tạo cho dự báo hợp nhất nêu trên, như là ví dụ trên Fig.11. Đối với ứng viên trong danh sách, MV LX của nó với X bằng bit chặn lẻ của giá trị chỉ số ứng viên hợp nhất, được sử dụng làm MV dự báo đơn cho chế độ phân vùng tam giác. Các MV này được đánh dấu là “x” trên Fig.11. Nếu MV LX tương ứng không tồn tại, MV L(1-X) của cùng ứng viên trong danh sách ứng viên dự báo hợp nhất được sử dụng làm MV dự báo đơn cho chế độ phân vùng tam giác.

(2) Kết hợp dọc theo cạnh phân vùng tam giác

Sau khi dự báo mỗi phân vùng tam giác nhờ sử dụng chuyển động riêng, áp dụng kết hợp cho hai tín hiệu dự báo để dẫn xuất các mẫu quanh cạnh đường chéo hoặc phi đường chéo. Các trọng số sau được sử dụng trong quá trình kết hợp: $\{7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8\}$ đối với độ sáng và $\{6/8, 4/8, 2/8\}$ đối với sắc độ, như được thể hiện trên Fig.12.

(3) Lưu trữ trường chuyển động

Lấy làm ví dụ, các MV của CU được mã hóa trong chế độ phân vùng tam giác được lưu trữ trong các đơn vị 4×4 . Tùy thuộc vào vị trí của mỗi đơn vị 4×4 , các MV dự báo đơn hoặc dự báo kép được lưu trữ. Ký hiệu Mv1 và Mv2 làm các MV dự báo đơn lần lượt cho phân vùng 1 và phân vùng 2. Nếu đơn vị 4×4

được đặt trong khu vực không có trọng số được thể hiện trên ví dụ của Fig.12, Mv1 hoặc Mv2 được lưu trữ đối với đơn vị 4x4 đó. Ngược lại (nếu đơn vị 4x4 được đặt trong khu vực có trọng số), MV dự báo kép được lưu trữ. MV dự báo kép được dẫn xuất từ Mv1 và Mv2 theo quá trình sau:

1) Nếu Mv1 và Mv2 là từ các danh sách ảnh tham chiếu khác nhau (một từ L0 và MV còn lại từ L1), thì Mv1 và Mv2 được kết hợp đơn giản để tạo thành MV dự báo kép.

2) Ngược lại, nếu Mv1 và Mv2 là từ cùng danh sách, và không đánh mất tính tổng quát, giả sử chúng đều từ L0. Trong trường hợp này:

Nếu ảnh tham chiếu của Mv2 (hoặc Mv1) xuất hiện trong L1, thì Mv2 đó (hoặc Mv1) được biến đổi thành MV L1 sử dụng ảnh tham chiếu đó trong L1. Sau đó, hai MV được kết hợp để tạo MV dự báo kép;

Ngược lại, thay vì chuyển động dự báo kép, chỉ chuyển động dự báo đơn Mv1 được lưu trữ.

Lấy ví dụ khác, lưu trữ vector chuyển động đối với chế độ phân vùng tam giác được lấy ví dụ trên Fig.13.

IV. Phân vùng hình học cho các khối ngoài

Đối với chế độ phân vùng tam giác, hai khối phụ được tách bằng đường tách, và hướng của đường tách bằng 45° hoặc 135° . Các góc phân vùng khác và các tỷ lệ phân vùng cũng có thể có, được gọi là chế độ phân vùng hình học (Geometrical partitioning, GEO) theo sáng chế, chẳng hạn, như được thể hiện trên Fig.14.

Trong ví dụ, toàn bộ số lượng phân vùng GEO có thể bằng 140. Con số này được cho là tối ưu để cân bằng giữa tính linh hoạt và chi phí bit. Báo hiệu bổ sung cho GEO đòi hỏi báo hiệu cho góc α , và việc dịch chuyển đường tách so với tâm của khối p . α là góc lượng tử giữa 0° và 360° , với việc tách $11,25^\circ$ và p là khoảng cách với 5 giá trị khác nhau. Ý nghĩa α và p được lấy ví dụ trên Fig.15.

Có thể hiểu rằng trong lĩnh vực kỹ thuật tạo mã video, nói chung, quá trình dự báo được thực hiện ở phía bộ mã hóa và được thực hiện ở phía bộ giải mã là giống nhau, ngoại trừ việc thông tin phụ được phân tách từ dòng bit ở phía bộ

giải mã, trong khi thông tin phụ được xác định bởi quy tắc định trước, như phương pháp RDO. Lấy làm ví dụ, các bước sau được áp dụng để thu được các mẫu dự báo trong chế độ GEO ở phía bộ giải mã:

S101: Xác định tỷ lệ phân vùng cho khối hiện tại.

Phần tử cú pháp `geo_partition_idx` được phân tách từ dòng bit, được sử dụng làm chỉ số cho bảng tra cứu lưu trữ các cặp α và ρ . Trong ví dụ, α và ρ lần lượt được biểu diễn bằng chỉ số góc và chỉ số khoảng cách.

S102: Thu được chế độ dự báo thứ nhất cho khối phụ thứ nhất và chế độ dự báo thứ hai cho khối phụ thứ hai.

Nói chung, quá trình dẫn xuất danh sách hợp nhất tương tự được sử dụng cho chế độ phân vùng tam giác được sử dụng để dẫn xuất các MV của mỗi phân vùng (khối phụ) của khối GEO. Mỗi phân vùng được dự báo chỉ bằng cách dự báo đơn. Hai phần tử cú pháp `geo_merge_idx0` và `geo_merge_idx1` lần lượt được phân tách từ dòng bit để chỉ báo các chế độ dự báo của hai khối phụ.

Trong ví dụ, chế độ dự báo thứ nhất không giống hệt chế độ dự báo thứ hai. Trong ví dụ, chế độ dự báo (chế độ dự báo thứ nhất hoặc chế độ dự báo thứ hai) có thể là chế độ dự báo ngoài, thông tin cho chế độ dự báo ngoài có thể bao gồm chỉ số ảnh tham chiếu và MV. Trong ví dụ khác, chế độ dự báo có thể là chế độ dự báo trong, thông tin cho chế độ dự báo trong có thể bao gồm chỉ số chế độ dự báo trong.

S103: Tạo giá trị dự báo thứ nhất cho khối phụ thứ nhất và giá trị dự báo thứ hai cho khối phụ thứ hai, lần lượt sử dụng chế độ dự báo thứ nhất và chế độ dự báo thứ hai.

S104: Thu được các giá trị được kết hợp của các mẫu dự báo theo kết hợp của giá trị dự báo thứ nhất và giá trị dự báo thứ hai, theo phép chia được bộc lộ ở bước S101.

Lấy làm ví dụ, cụ thể hơn, bước S104 bao gồm:

S104-1: Đối với mẫu độ sáng trong khối hiện tại, khoảng cách mẫu (`sample_dist`) được tính toán, khoảng cách mẫu là khoảng cách của mẫu độ sáng đến đường tách của khối hiện tại được xác định ở bước S101.

Trong ví dụ, khoảng cách mẫu `sample_dist` được tính toán theo công thức:

$sample_dist = ((x < 1) + 1) * Dis[angleIdx1] + ((y < 1) + 1) * Dis[angleIdx2] - offset(distanceIdx).$

- $angleIdx1$ và $angleIdx2$ thu được từ dòng bit hoặc được dẫn xuất dựa trên thông tin khác thu được từ dòng bit, chẳng hạn, được dẫn xuất dựa trên phần tử cú pháp $geo_partition_idx$, làm chỉ số góc ở bước S101; $angleIdx1$ và $angleIdx2$ là các tham số lượng giác được lượng tử của đường tách.

- x và y là các khoảng cách ngang và dọc của mẫu với mẫu trên bên trái của khối hiện tại.

- $offset(distanceIdx)$ làm hàm của giá trị chỉ số ($distanceIdx$), giá trị chỉ số thu được từ dòng bit hoặc được dẫn xuất dựa trên thông tin khác thu được từ dòng bit, chẳng hạn, được dẫn xuất dựa trên phần tử cú pháp $geo_partition_idx$, như là chỉ số khoảng cách ở bước S101.

- $Dis[]$ là bảng tra cứu định trước.

Trong ví dụ khác, $sample_dist$ và $sampleWeight1$ có thể thu được theo các phương trình sau:

$$- nCbR = (W > H) ? (W / H) : (H / W)$$

$$- sample_dist = (W > H) ? (Clip3(0, 8, (x / nCbR - y) + 4)) : (Clip3(0, 8, (x - y / nCbR) + 4)),$$

hoặc

$$sample_dist = (W > H) ? (Clip3(0, 8, (H - 1 - x / nCbR - y) + 4)) : (Clip3(0, 8, (W - 1 - x - y / nCbR) + 4)).$$

trong đó W là chiều rộng khối hiện tại, H là chiều cao của khối hiện tại.

Cần lưu ý rằng 2 ví dụ nêu trên thể hiện hai phương pháp tính toán theo số học số nguyên. Các phương án thực hiện sáng chế không bị giới hạn ở các ví dụ tính toán $sample_dist$.

Trong ví dụ, $sample_dist$ có thể là khoảng cách đường vuông góc giữa mẫu được đặt ở (x, y) và đường tách. Trong ví dụ khác, tạo thành đường ngang hoặc đường dọc qua mẫu được đặt ở (x, y) . Đường ngang hoặc đường dọc sẽ có điểm giao với đường tách. $sample_dist$ có thể là khoảng cách giữa mẫu được đặt ở (x, y) và điểm giao.

S104-2: Hoạt động kết hợp được triển khai dưới dạng hàm có $sample_dist$

làm đầu vào, và sampleWeight1 hoặc sampleWeight2 làm đầu ra.

sample_dist được sử dụng để tính toán các hệ số trọng số, các hệ số trọng số được sử dụng đối với giá trị dự báo kết hợp được dẫn xuất bằng cách kết hợp lần lượt giá trị dự báo thứ nhất và giá trị dự báo thứ hai tương ứng với khối phụ thứ nhất và khối phụ thứ hai. Trong ví dụ, các hệ số trọng số được ký hiệu là sampleWeight1 và sampleWeight2, đề cập đến các giá trị trọng số lần lượt tương ứng với giá trị dự báo thứ nhất và giá trị dự báo thứ hai. Trong ví dụ, sampleWeight1 được tính toán theo sample_dist, sampleWeight2 được tính toán theo công thức $\text{sampleWeight2} = T - \text{sampleWeight1}$, trong đó T là hằng số định trước. Trong một ví dụ, hằng số định trước T bằng 8, nghĩa là sampleWeight1 và sampleWeight2 có thể có giá trị nằm trong khoảng giá trị giữa 0 và 8 (bao gồm 0 và 8). Trong ví dụ khác, hằng số định trước T bằng 16.

Theo một ví dụ, hệ số trọng số thứ nhất sampleWeight1 thu được theo sample_dist và bảng tra cứu. Trong ví dụ, bảng tra cứu được sử dụng để lưu trữ các hệ số lọc. Nói cách khác, hoạt động kết hợp được triển khai theo bảng tra cứu. Trong ví dụ này, hàm sample_dist có thể được sử dụng làm giá trị chỉ số cho bảng tra cứu. Hàm nêu trên có thể là phép chia hoặc phép nhân với phép toán số hằng số, phép dịch phải lượng hằng số, lấy giá trị tuyệt đối hoặc hoạt động rút gọn hoặc kết hợp của các giá trị này.

Hoạt động kết hợp được áp dụng cho mẫu theo giá trị chỉ báo trong dòng bit, hoặc mẫu theo giá trị chỉ báo trong dòng bit.

Trong ví dụ, $\text{sampleWeight1} = \text{Clip3}(0, 8, f(\text{sample_dist}))$. Hàm $f()$ có thể là phép chia/phép cộng/phép nhân với phép toán số hằng số, phép dịch phải với lượng hằng số, lấy phép toán số tuyệt đối hoặc phép rút gọn hoặc kết hợp của chúng.

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{Clip3}(0, 8, \text{sample_dist})$,

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{Clip3}(0, 8, \text{sample_dist}+4)$;

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{sample_dist} == 4 ? 4 : \text{sample_dist} < 4 ? 0 : 8$;

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{Clip3}(0, 8, (\text{sample_dist} - 4) * K + \text{sample_dist})$;

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{sample_dist} == 0 ? 4 : \text{sample_dist} < 0 ? 0 : 8$;

Trong ví dụ, sampleWeight1 (hoặc sampleWeight2) = $\text{Clip3}(0, 8, \text{sample_dist} * K + \text{Clip3}(0, 8, \text{sample_dist} + 4))$, K là số nguyên có giá trị lớn hơn 0. Cần lưu ý rằng hoạt động kết hợp trở nên ngắn hơn (do vậy sắc nét hơn) với K tăng. Chẳng hạn, khi $K = 4$, thì hoạt động kết hợp thứ hai trở nên giống như - “ $\text{sampleWeight} = \text{sample_dist} == 0 ? 4 : \text{sample_dist} < 0 ? 0 : 8$ ”, được lấy ví dụ trên Fig.16. Theo một ví dụ, giá trị của bộ chỉ báo trong dòng bit chỉ báo giá trị của K .

Trong ví dụ, hoạt động kết hợp có thể được triển khai dưới dạng bảng tra cứu, chẳng hạn các bảng được lấy ví dụ trên Fig.17. Trong ví dụ, sampleWeight có thể thu được dưới dạng $\text{geoFilter}[\text{idx}]$ trong đó idx thu được dưới dạng hàm của sampleDist và geoFilter là mảng tuyến tính một chiều của các trọng số lọc. Lấy làm ví dụ, $\text{idx} = \min((\text{abs}(\text{sample_dist}) + 8) \gg 4, \text{maxIdx})$, trong đó maxIdx là giá trị lớn nhất mà idx có thể có.

S105: lưu trữ thông tin chế độ dự báo (chẳng hạn, thông tin chuyển động bao gồm các MV hoặc thông tin dự báo trong) của khối hiện tại theo phép chia được bộc lộ ở bước S101 và theo phương pháp kết hợp được bộc lộ ở bước 104.

Cụ thể là, theo phương án thực hiện thứ nhất của quá trình giải mã đối với chế độ dự báo GEO:

Quá trình này được gọi khi giải mã CU với $\text{MergeTriangleFlag}[\text{xCb}][\text{yCb}]$ bằng 1 hoặc với $\text{geo_flag}[\text{xCb}][\text{yCb}]$ bằng 1.

Các đầu vào quá trình này là:

- vị trí độ sáng (xCb , yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của khối mã hiện tại trong các mẫu độ sáng,
- biến cbHeight xác định chiều cao của khối mã hiện tại trong các mẫu độ sáng,

- các MV độ sáng có độ chính xác mẫu phân số $1/16$ mvA và mvB,
- các MV sắc độ mvCA và mvCB,
- các chỉ số tham chiếu refIdxA và refIdxB,
- các cờ danh sách dự báo predListFlagA và predListFlagB.

Đầu ra của quá trình này là:

- mảng $(cbWidth) \times (cbHeight)$ predSamples_L của các mẫu dự báo độ sáng,
- mảng $(cbWidth / 2) \times (cbHeight / 2)$ predSamples_{Cb} của các mẫu dự báo sắc độ đối với thành phần Cb,
- mảng $(cbWidth / 2) \times (cbHeight / 2)$ predSamples_{Cr} của các mẫu dự báo sắc độ đối với thành phần Cr.

Lấy predSamplesLA_L và predSamplesLB_L là các mảng $(cbWidth) \times (cbHeight)$ của các giá trị mẫu độ sáng được dự báo và, predSamplesLAc_b, predSamplesLBc_b, predSamplesLAc_r và predSamplesLBc_r là các mảng $(cbWidth / 2) \times (cbHeight / 2)$ của các giá trị mẫu sắc độ được dự báo.

predSamples_L, predSamples_{Cb} và predSamples_{Cr} được dẫn xuất bởi các bước có thứ tự sau:

1. Với N là một trong A và B, điều kiện sau áp dụng:

- ảnh tham chiếu gồm mảng hai chiều được sắp thứ tự refPicLN_L của các mẫu độ sáng và hai mảng hai chiều được sắp thứ tự refPicLN_{Cb} và refPicLN_{Cr} của các mẫu sắc độ được dẫn xuất b với X được gán bằng predListFlagN và refIdxX được gán bằng refIdxN làm đầu vào.

- Mảng predSamplesLN_L được dẫn xuất với vị trí độ sáng (xCb, yCb), chiều rộng khối mã độ sáng sbWidth được gán bằng cbWidth, chiều cao khối mã độ sáng sbHeight được gán bằng cbHeight, độ lệch MV mvOffset được gán bằng (0, 0), MV mvLX được gán bằng mvN và mảng tham chiếu refPicLX_L được gán bằng refPicLN_L, biến bdofFlag được gán bằng FALSE, và biến cIdx được gán bằng 0 làm đầu vào.

- Mảng predSamplesLN_{Cb} được dẫn xuất với vị trí độ sáng (xCb, yCb), chiều rộng khối mã sbWidth được gán bằng $cbWidth / 2$, chiều cao khối mã sbHeight được gán bằng $cbHeight / 2$, độ lệch MV mvOffset được gán

bằng (0, 0), MV mvLX được gán bằng mvCN, và mảng tham chiếu refPicLXCb được gán bằng refPicLNCb, biến bdoFlag được gán bằng FALSE, và biến cIdx được gán bằng 1 làm đầu vào.

– Mảng predSamplesLNCr được dẫn xuất với vị trí độ sáng (xCb, yCb), chiều rộng khối mã sbWidth được gán bằng cbWidth / 2, chiều cao khối mã sbHeight được gán bằng cbHeight / 2, độ lệch MV mvOffset được gán bằng (0, 0), MV mvLX được gán bằng mvCN, và mảng tham chiếu refPicLXCr được gán bằng refPicLNCr, biến bdoFlag được gán bằng FALSE, và biến cIdx được gán bằng 2 làm đầu vào.

2. Nếu geo_flag[xCb][yCb] bằng 1, các biến liên quan đến phân vùng hình học angleIdx và distanceIdx được thiết lập theo giá trị của geo_partitioning_idx[xCb][yCb] như được xác định trong Bảng 1.

a. Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học được gọi. Đầu vào cho quá trình này là lần lượt ba mảng được sắp thứ tự sampleWeightL, sampleWeightC và motionWeight có kích thước (cbWidth)x(cbHeight), (cbWidth/2)x(cbHeight/2) và (cbWidth/4)x(cbHeight/4), cũng như các biến angleIdx, distanceIdx, cbWidth và cbHeight.

b. Các mẫu dự báo bên trong khối mã độ sáng hiện tại, predSamplesL[xL][yL] với $xL = 0..cbWidth - 1$ và $yL = 0..cbHeight - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học với chiều rộng khối mã nCbW được gán bằng cbWidth, chiều cao khối mã nCbH được gán bằng cbHeight, các mảng mẫu predSamplesLAL, predSamplesLBL, predWeightL và cIdx bằng 0 làm đầu vào.

c. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cb, predSamplesCb[xC][yC] với $xC = 0..cbWidth / 2 - 1$ và $yC = 0..cbHeight / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học với chiều rộng khối mã nCbW được gán bằng cbWidth / 2, chiều cao khối mã nCbH được gán bằng cbHeight / 2, các mảng mẫu predSamplesLACb, predSamplesLBCb, predWeightC và cIdx bằng 1 làm đầu vào.

d. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cr, $\text{predSamplesCr}[xC][yC]$ với $xC = 0..cbWidth / 2 - 1$ và $yC = 0..cbHeight / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học với chiều rộng khối mã $nCbW$ được gán bằng $cbWidth / 2$, chiều cao khối mã $nCbH$ được gán bằng $cbHeight / 2$, các mảng mẫu predSamplesLACr , predSamplesLBCr , predWeightC và $cIdx$ bằng 2 làm đầu vào.

e. Quá trình lưu trữ MV đối với chế độ hợp nhất hình học được gọi với vị trí khối mã độ sáng (xCb, yCb), chiều rộng khối mã độ sáng $cbWidth$, chiều cao khối mã độ sáng $cbHeight$, mảng mẫu motionWeight , các MV độ sáng mvA và mvB , các chỉ số tham chiếu $refIdxA$ và $refIdxB$, và các cờ danh sách dự báo predListFlagA và predListFlagB làm đầu vào.

Quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác

Đầu vào cho quá trình này là:

- biến X biểu diễn danh sách tham chiếu bằng 0 hoặc 1,
- chỉ số tham chiếu $refIdxN$.

Đầu ra của quá trình này là:

- chỉ số tham chiếu $refIdxTemp$.

Biến $refPicPoc$ được dẫn xuất như sau:

$$\text{refPicPoc} = (X == 0) ? \text{RefPicList}[0][refIdxN] : \text{RefPicList}[1][refIdxN]$$

(8-878)

Danh sách ảnh tham chiếu refPicListTemp được dẫn xuất như sau:

$$\text{refPicListTemp} = (X == 0) ? \text{RefPicList}[1] : \text{RefPicList}[0]$$

(8-879)

Biến $refIdxTemp$ được dẫn xuất như sau:

- Biến mapStop được gán bằng FALSE.
- Với biến $refIdx_m$ có $m = 0..NumRefIdxActive[1] - 1$, phương trình sau áp dụng đến khi mapStop bằng FALSE:

$$\text{refIdxTemp} = (\text{refPicListTemp}[refIdx_m] == \text{refPicPoc}) ? refIdx_m : -1$$

(8-880)

mapStop = (refIdxTemp != -1) ? TRUE : FALSE
(8-881)

Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học

Các đầu vào quá trình này là:

- hai biến nCbW và nCbH xác định chiều rộng và chiều cao của khối mã hiện tại,
- hai biến angleIdx và distanceIdx xác định góc và các giá trị tra cứu khoảng cách của chế độ phân vùng hình học hiện tại,
- mảng (nCbW)x(nCbH) sampleWeightL,
- mảng (nCbW/2)x(nCbH/2) sampleWeightC,
- mảng (nCbW/4)x(nCbH/4) motionWeight.

Đầu ra của quá trình này là ba mảng giống nhau sampleWeightL, sampleWeightC và motionWeight của các giá trị trọng số theo mẫu có khoảng từ 0 ... 8.

1. Các trọng số kết hợp độ sáng được lưu trữ trong sampleWeightL đối với chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

Giá trị của các biến sau được thiết lập:

- wIdx được gán bằng $\log_2(\text{nCbW}) - 3$,
- hIdx được gán bằng $\log_2(\text{nCbH}) - 3$,
- stepOffset được gán bằng 64,
- các biến sau được thiết lập theo so sánh ($wIdx \geq hIdx$):
- whRratio = $(wIdx \geq hIdx)?wIdx - hIdx:hIdx - wIdx$ (8-882)
- wIsLargerH = $(wIdx \geq hIdx)?\text{true}:\text{false}$ (8-883)
- scaleStep = $(wIdx \geq hIdx)?(1 \ll hIdx):(1 \ll wIdx)$ (8-884)
- displacementX được gán bằng angleIdx
- displacementY được gán bằng $(\text{displacementX} + 8)\%32$
- angleN được gán bằng:
 - angleN = angleIdx nếu $\text{angleIdx} \geq 0 \ \&\& \ \text{angleIdx} \leq 8$,
 - angleN = $16 - \text{angleIdx}$ nếu $\text{angleIdx} > 8 \ \&\& \ \text{angleIdx} \leq 16$,
 - angleN = $\text{angleIdx} - 16$ nếu $\text{angleIdx} > 16 \ \&\& \ \text{angleIdx} \leq 24$,
 - angleN = $32 - \text{angleIdx}$ ngược lại,

– ngoài ra, angleN được gán bằng giá trị sau, phụ thuộc vào wIsLargerH:

$$\text{angleN} = (\text{wIsLargerH}) ? 8 - \text{angleN} : \text{angleN}$$

– stepDis được gán theo các giá trị của whRatio và angleN như được xác định trong Bảng 2 và ngoài ra,

$$\text{stepSize} = \text{stepDis} + \text{stepOffset}$$

– rho được gán bằng giá trị sau nhờ sử dụng bảng tra cứu được ký hiệu là Dis, được xác định trong Bảng 3:

$$\text{rho} = \text{distanceIdx} * \text{stepSize} * \text{scaleStep} + \text{nCbW} * \text{Dis}[\text{displacementX}] + \text{nCbH} * \text{Dis}[\text{displacementY}]. \quad (8-885)$$

Đối với mỗi vị trí mẫu $x=0 \dots \text{nCbW} - 1$ và $y=0 \dots \text{nCbH} - 1$, giá trị của $\text{sampleWeight}_L[x][y]$ được dẫn xuất theo cách sau:

– khoảng cách distFromLine được tính toán sử dụng bảng tra cứu được ký hiệu là Dis, được xác định trong Bảng 3 như sau:

$$\text{distFromLine} = ((x < 1) + 1) * \text{Dis}[\text{displacementX}] + ((y < 1) + 1) * \text{Dis}[\text{displacementY}] - \text{rho}. \quad (8-886)$$

– Khoảng cách chia tỷ lệ distScaled được dẫn xuất là:

$$\text{distScaled} = \min((\text{abs}(\text{distFromLine}) + 8) \gg 4, 14).$$

(8-887)

Giá trị của $\text{sampleWeight}_L[x][y]$ được thiết lập theo bảng tra cứu trọng số lọc, được ký hiệu là GeoFilter được xác định trong

– Bảng 4 là:

$$\text{sampleWeight}_L[x][y] = \text{GeoFilter}[\text{distScaled}] \quad \text{nếu } \text{distFromLine} \leq 0$$

$$\text{sampleWeight}_L[x][y] = 8 - \text{GeoFilter}[\text{distScaled}] \quad \text{nếu } \text{distFromLine} > 0$$

2. Các trọng số sắc độ được lưu trữ trong sampleWeight_c đối với chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

Đối với mỗi vị trí mẫu $x=0 \dots (\text{nCbW}/2) - 1$ và $y=0 \dots (\text{nCbH}/2) - 1$, giá trị của $\text{sampleWeight}_c[x][y]$ được dẫn xuất theo cách sau:

$$\text{sampleWeight}_c[x][y] = \text{sampleWeight}_L[(x < 1)][(y < 1)]$$

3. Các trọng số chuyển động được lưu trữ trong mảng motionWeight đối với chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

- Các biến sau được thiết lập:

$$\text{threshScaler} = ((\log_2(\text{nCbW}) + \log_2(\text{nCbH})) \gg 1) - 1$$

$$\text{threshLower} = 32 \gg \text{threshScaler}$$

$$\text{threshUpper} = 32 - \text{threshLower}$$

Đối với mỗi vị trí mẫu $x=0\dots(\text{nCbW}/4) - 1$ và $y=0\dots(\text{nCbH}/4) - 1$, giá trị của $\text{motionWeight}[x][y]$ được dẫn xuất theo cách sau:

$$\begin{aligned} \text{Cnt} = & \text{sampleWeight}_L[(x \ll 2)][(y \ll 2)] \\ & + \text{sampleWeight}_L[(x \ll 2) + 3][(y \ll 2)] \\ & + \text{sampleWeight}_L[(x \ll 2)][(y \ll 2) + 3] \\ & + \text{sampleWeight}_L[(x \ll 2) + 3][(y \ll 2) + 3] \end{aligned}$$

Giá trị của $\text{motionWeight}[x][y]$ được gán bằng:

$$\text{motionWeight}[x][y] = 0, \quad \text{nếu } \text{Cnt} \leq \text{threshLower}$$

$$\text{motionWeight}[x][y] = 1, \quad \text{nếu } \text{Cnt} \geq \text{threshUpper}$$

$$\text{motionWeight}[x][y] = 2, \quad \text{ngược lại}$$

Quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học

Các đầu vào quá trình này là:

- hai biến nCbW và nCbH xác định chiều rộng và chiều cao của khối mã hiện tại,
- hai mảng $(\text{nCbW}) \times (\text{nCbH})$ predSamplesLA và predSamplesLB ,
- mảng $(\text{nCbW}) \times (\text{nCbH})$ sampleWeight ,
- biến cIdx xác định chỉ số thành phần màu.

Đầu ra của quá trình này là mảng $(\text{nCbW}) \times (\text{nCbH})$ pbSamples của các giá trị mẫu dự báo.

Biến bitDepth được dẫn xuất như sau:

- Nếu cIdx bằng 0, bitDepth được gán bằng BitDepth_Y .
- Ngược lại, bitDepth được gán bằng BitDepth_C .

Các biến shift1 và offset1 được dẫn xuất như sau:

- Biến shift1 được gán bằng $\text{Max}(5, 17 - \text{bitDepth})$.
- Biến offset1 được gán bằng $1 \ll (\text{shift1} - 1)$.

Các giá trị mẫu dự báo được dẫn xuất như sau:

$$\text{pbSamples}[x][y] =$$

$$\begin{aligned} & \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \\ & (\text{predSamplesLA}[x][y] * \text{sampleWeight}[x][y] \\ & + \text{predSamplesLB}[x][y] * (8 - \text{sampleWeight}[x][y]) + \text{offset1}) \gg s \\ & \text{hft1}) \end{aligned}$$

Lưu trữ MV đối với chế độ hợp nhất phân vùng hình học

Quá trình này được gọi khi giải mã CU với `geo_flag[xCb][yCb]` bằng 1.

Các đầu vào quá trình này là:

- vị trí độ sáng (`xCb`, `yCb`) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến `cbWidth` xác định chiều rộng của khối mã hiện tại trong mẫu độ sáng,
- biến `cbHeight` xác định chiều cao của khối mã hiện tại trong mẫu độ sáng,
- mảng `motionWeight` có kích thước $(\text{cbWidth}/4) \times (\text{cbHeight}/4)$ chứa các chỉ số phân định chuyển động,
- các MV độ sáng có độ chính xác mẫu phân số $1/16$ `mvA` và `mvB`,
- các chỉ số tham chiếu `refIdxA` và `refIdxB`,
- các cờ danh sách dự báo `predListFlagA` và `predListFlagB`.

Các biến `numSbX` và `numSbY` xác định số lượng khối 4×4 trong khối mã hiện tại theo chiều ngang và chiều dọc được gán bằng `numSbX = cbWidth >> 2` và `numSbY = cbHeight >> 2`.

Biến `minSb` được gán bằng `min(numSbX, numSbY)`.

Biến `refIdxTempA` được dẫn xuất bằng cách gọi quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác với X được gán bằng `predListFlagA` và `refIdxN` được gán bằng `refIdxA` làm đầu vào.

Biến `refIdxTempB` được dẫn xuất bằng gọi quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác với X được gán bằng `predListFlagB` và `refIdxN` được gán bằng `refIdxB` làm đầu vào.

Đối với mỗi khối phụ 4×4 ở chỉ số khối phụ (`xSbIdx`, `ySbIdx`) với `xSbIdx = 0..numSbX - 1`, và `ySbIdx = 0..numSbY - 1`, điều kiện sau áp dụng:

- Nếu `motionWeight[xSbIdx][ySbIdx]` bằng 0, phương trình sau áp dụng:

$$\text{predFlagL0} = (\text{predListFlagA} == 0) ? 1 : 0 \quad (8-888)$$

$$\text{predFlagL1} = (\text{predListFlagA} == 0) ? 0 : 1 \quad (8-889)$$

$$\text{refIdxL0} = (\text{predListFlagA} == 0) ? \text{refIdxA} : -1 \quad (8-890)$$

$$\text{refIdxL1} = (\text{predListFlagA} == 0) ? -1 : \text{refIdxA} \quad (8-891)$$

$$\text{mvL0}[0] = (\text{predListFlagA} == 0) ? \text{mvA}[0] : 0 \quad (8-892)$$

$$\text{mvL0}[1] = (\text{predListFlagA} == 0) ? \text{mvA}[1] : 0 \quad (8-893)$$

$$\text{mvL1}[0] = (\text{predListFlagA} == 0) ? 0 : \text{mvA}[0] \quad (8-894)$$

$$\text{mvL1}[1] = (\text{predListFlagA} == 0) ? 0 : \text{mvA}[1] \quad (8-895)$$

– Ngược lại, nếu $\text{motionWeight}[\text{xSbIdx}][\text{ySbIdx}]$ bằng 1, phương trình sau

áp dụng:

$$\text{predFlagL0} = (\text{predListFlagB} == 0) ? 1 : 0 \quad (8-896)$$

$$\text{predFlagL1} = (\text{predListFlagB} == 0) ? 0 : 1 \quad (8-897)$$

$$\text{refIdxL0} = (\text{predListFlagB} == 0) ? \text{refIdxB} : -1 \quad (8-898)$$

$$\text{refIdxL1} = (\text{predListFlagB} == 0) ? -1 : \text{refIdxB} \quad (8-899)$$

$$\text{mvL0}[0] = (\text{predListFlagB} == 0) ? \text{mvB}[0] : 0 \quad (8-900)$$

$$\text{mvL0}[1] = (\text{predListFlagB} == 0) ? \text{mvB}[1] : 0 \quad (8-901)$$

$$\text{mvL1}[0] = (\text{predListFlagB} == 0) ? 0 : \text{mvB}[0] \quad (8-902)$$

$$\text{mvL1}[1] = (\text{predListFlagB} == 0) ? 0 : \text{mvB}[1] \quad (8-903)$$

– Ngược lại ($\text{motionWeight}[\text{xSbIdx}][\text{ySbIdx}]$ bằng 2), phương trình sau áp dụng:

– Nếu $\text{predListFlagA} + \text{predListFlagB}$ bằng 1,

$$\text{predFlagL0} = 1 \quad (8-904)$$

$$\text{predFlagL1} = 1 \quad (8-905)$$

$$\text{refIdxL0} = (\text{predListFlagA} == 0) ? \text{refIdxA} : \text{refIdxB} \quad (8-906)$$

$$\text{refIdxL1} = (\text{predListFlagA} == 0) ? \text{refIdxB} : \text{refIdxA} \quad (8-907)$$

$$\text{mvL0}[0] = (\text{predListFlagA} == 0) ? \text{mvA}[0] : \text{mvB}[0] \quad (8-908)$$

$$\text{mvL0}[1] = (\text{predListFlagA} == 0) ? \text{mvA}[1] : \text{mvB}[1] \quad (8-909)$$

$$\text{mvL1}[0] = (\text{predListFlagA} == 0) ? \text{mvB}[0] : \text{mvA}[0] \quad (8-910)$$

$$\text{mvL1}[1] = (\text{predListFlagA} == 0) ? \text{mvB}[1] : \text{mvA}[1] \quad (8-911)$$

– Nếu $\text{predListFlagA} + \text{predListFlagB}$ bằng 0, phương trình sau áp dụng:

$$\text{predFlagL0} = 1 \quad (8-912)$$

predFlagL1 = (refIdxTempA == -1 && refIdxTempB == -1) ? 0 : 1
(8-913)

refIdxL0 = (refIdxTempB != -1) ? refIdxA : (8-914)

((refIdxTempA != -1) ? refIdxB : refIdxA)

refIdxL1 = (refIdxTempB != -1) ? refIdxTempB : (8-915)

((refIdxTempA != -1) ? refIdxTempA : -1)

mvL0[0] = (refIdxTempB != -1) ? mvA[0] : (8-916)

((refIdxTempA != -1) ? mvB[0]:mvA[0])

mvL0[1] = (refIdxTempB != -1) ? mvA[1] : (8-917)

((refIdxTempA != -1) ? mvB[1]:mvA[1])

mvL1[0] = (refIdxTempB != -1) ? mvB[0] : (8-918)

((refIdxTempA != -1) ? mvA[0]:0)

mvL1[1] = (refIdxTempB != -1) ? mvB[1] : (8-919)

((refIdxTempA != -1) ? mvA[1]:0)

– Nếu predListFlagA + predListFlagB is bằng 2, phương trình sau áp dụng:

predFlagL0 = (refIdxTempA == -1 && refIdxTempB == -1) ? 0 : 1
(8-920)

predFlagL1 = 1 (8-921)

refIdxL0 = (refIdxTempB != -1) ? refIdxTempB : (8-922)

((refIdxTempA != -1) ? refIdxTempA : -1)

refIdxL1 = (refIdxTempB != -1) ? refIdxA : (8-923)

((refIdxTempA != -1) ? refIdxB : refIdxA)

mvL0[0] = (refIdxTempB != -1) ? mvB[0] : (8-924)

((refIdxTempA != -1) ? mvA[0]:0)

mvL0[1] = (refIdxTempB != -1) ? mvB[1] : (8-925)

((refIdxTempA != -1) ? mvA[1]:0)

mvL1[0] = (refIdxTempB != -1) ? mvA[0] : (8-926)

((refIdxTempA != -1) ? mvB[0]:mvA[0])

mvL1[1] = (refIdxTempB != -1) ? mvA[1] : (8-927)

((refIdxTempA != -1) ? mvB[1]:mvA[1])

– Các phép sau được thực hiện với $x = 0..3$ và $y = 0..3$:

$$MvL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = mvL0 \quad (8-928)$$

$$MvL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = mvL1 \quad (8-929)$$

$$RefIdxL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = refIdxL0 \quad (8-930)$$

$$RedIdxL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = refIdxL1 \quad (8-931)$$

$$PredFlagL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = predFlagL0 \quad (8-932)$$

$$PredFlagL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = predFlagL1 \quad (8-933)$$

Bảng 1 – Mô tả các giá trị angleIdx và distanceIdx dựa trên giá trị geo_partition_idx.

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

33	7	0
34	7	1
35	7	2
36	7	3
37	7	4
38	8	1
39	8	2
40	8	3
41	8	4
42	9	0
43	9	1
44	9	2
45	9	3
46	9	4
47	10	0
48	10	1
49	10	2
50	10	3
51	10	4
52	11	0
53	11	1
54	11	2
55	11	3
56	11	4
57	12	1
58	12	2
59	12	3
60	12	4
61	13	0
62	13	1
63	13	2
64	13	3
65	13	4
66	14	0
67	14	1
68	14	2
69	14	3
70	14	4
71	15	0
72	15	1
73	15	2
74	15	3
75	15	4
76	16	1
77	16	2
78	16	3
79	16	4
80	17	1

81	17	2
82	17	3
83	17	4
84	18	1
85	18	2
86	18	3
87	18	4
88	19	1
89	19	2
90	19	3
91	19	4
92	20	1
93	20	2
94	20	3
95	20	4
96	21	1
97	21	2
98	21	3
99	21	4
100	22	1
101	22	2
102	22	3
103	22	4
104	23	1
105	23	2
106	23	3
107	23	4
108	24	1
109	24	2
110	24	3
111	24	4
112	25	1
113	25	2
114	25	3
115	25	4
116	26	1
117	26	2
118	26	3
119	26	4
120	27	1
121	27	2
122	27	3
123	27	4
124	28	1
125	28	2
126	28	3
127	28	4
128	29	1

129	29	2
130	29	3
131	29	4
132	30	1
133	30	2
134	30	3
135	30	4
136	31	1
137	31	2
138	31	3
139	31	4

Bảng 2 – Mô tả khoảng cách bước stepDis theo các giá trị whRatio và angleN.

whRatio	angleN	stepDis
0	0	0
0	1	18
0	2	31
0	3	40
0	4	42
0	5	40
0	6	31
0	7	18
0	8	0
1	0	0
1	1	38
1	2	71
1	3	97
1	4	115
1	5	125
1	6	126
1	7	118
1	8	102
2	0	0
2	1	78
2	2	149
2	3	210
2	4	260
2	5	295
2	6	315
2	7	319
2	8	307
3	0	0
3	1	158
3	2	306
3	3	438

3	4	549
3	5	636
3	6	694
3	7	721
3	8	717
4	0	0
4	1	318
4	2	619
4	3	893
4	4	1129
4	5	1317
4	6	1450
4	7	1524
4	8	1536

Bảng 3 – Bảng tra cứu Dis để dẫn xuất khoảng cách phân vùng hình học.

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dis[idx]	64	63	59	53	45	36	24	12	0	-12	-24	-36	-45	-53	-59	-63
idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Dis[idx]	-64	-63	-59	-53	-45	-36	-24	-12	0	12	24	36	45	53	59	63

Bảng 4 – Bảng tra cứu trọng số lọc GeoFilter để dẫn xuất các trọng số lọc phân vùng hình học

Idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
GeoFilter[idx]	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8

Theo phương án thực hiện thứ hai của quá trình giải mã cho chế độ dự báo GEO, quá trình lưu trữ thông tin chuyển động như sau:

Phân chia phụ khối mã thành K đơn vị lưu trữ chuyển động, kích thước của khối mã là các mẫu độ sáng $M \times N$, trong ví dụ, mỗi đơn vị lưu trữ chuyển động có cùng kích thước của các mẫu độ sáng $(M/K) \times (N/K)$. Trong ví dụ, giá trị K bằng 4, giá trị M bằng 16 và giá trị N bằng 16, trong ví dụ này, mỗi đơn vị lưu trữ chuyển động được liên kết với khối phụ độ sáng có kích thước 4×4 . Vị trí độ sáng trên bên trái (x,y) của khối mã được liên kết với vị trí trên bên trái $(x/K, y/K)$ của khối lưu trữ chuyển động.

Đối với đơn vị lưu trữ chuyển động, quyết định liệu thông tin chuyển động dự báo đơn thứ nhất, thông tin chuyển động dự báo đơn thứ hai, hoặc thông tin chuyển động dự báo kép được lưu trữ, theo trọng số mẫu thứ nhất hoặc trọng số

mẫu thứ hai. Trong ví dụ, quyết định này theo khoảng cách đối với mẫu trong đơn vị lưu trữ chuyển động đến đường biên tách. Mẫu có thể là mẫu số nguyên hoặc mẫu phân số. Đường tách được định nghĩa là đường tách của chế độ phân vùng hình học. Theo một ví dụ, K bằng 4 (chiều rộng và chiều cao của đơn vị lưu trữ chuyển động liên quan đến các mẫu độ sáng). Ví dụ phân chia khối mã thành các đơn vị lưu trữ chuyển động được mô tả trên Fig.13.

Theo một ví dụ, khoảng cách giữa mẫu và đường biên tách có thể được tính toán như sau: Đối với mẫu độ sáng trong khối, khoảng cách mẫu (sample_dist) được tính toán, khoảng cách mẫu là khoảng cách của mẫu độ sáng đến đường tách của khối.

Trong ví dụ, khoảng cách mẫu sample_dist được tính toán theo công thức:

$$\text{sample_dist} = ((x < 1) + 1) * \text{Dis}[\text{angleIdx1}] + ((y < 1) + 1) * \text{Dis}[\text{angleIdx2}] - \text{offset}(\text{distanceIdx}).$$

- Giá trị của angleIdx1 và giá trị của angleIdx2 thu được từ dòng bit hoặc được dẫn xuất/được tính toán dựa trên thông tin khác thu được từ dòng bit, angleIdx1 và angleIdx2 là các tham số lượng giác được lượng tử của đường tách, tham số thứ nhất có loại cosin và tham số thứ hai có loại sin.

- x và y là các tọa độ -x và -y của mẫu với mẫu trên bên trái của khối mã.
- offset(distanceIdx) là giá trị độ lệch vốn là hàm giá trị chỉ số (distanceIdx), giá trị chỉ số thu được từ dòng bit hoặc được dẫn xuất/được tính toán dựa trên thông tin khác thu được từ dòng bit.

- Dis[] là bảng tra cứu. Dis[angleIdx1] mô tả thay đổi trong khoảng cách mẫu (sample_dist) với tăng đơn vị (tăng giá trị 1) ở tọa độ -x của mẫu nêu trên. Dis[angleIdx2] mô tả thay đổi trong khoảng cách mẫu (sample_dist) với việc tăng đơn vị (tăng giá trị 1) ở tọa độ -y của mẫu nêu trên.

Cần lưu ý rằng tọa độ của ví dụ nêu trên là theo vị trí trên bên trái của khối mã hiện tại, không phải theo vị trí trên bên trái của khối mã.

Cần lưu ý rằng ví dụ nêu trên thể hiện phương pháp tính toán theo số học số nguyên. Các phương án thực hiện sáng chế không bị giới hạn ở các ví dụ đó khi tính toán sample_dist.

Cần lưu ý rằng theo ví dụ nêu trên, tọa độ mẫu được dịch trái một bit ((x < 1),

($y \ll 1$)). Trong ví dụ này khoảng cách tính toán có thể được thực hiện lên tới độ chính xác mẫu $\frac{1}{2}$ cho số học số nguyên, chẳng hạn vị trí mẫu ở (0,5, 0,5) có thể được tính toán.

Fig.18 thể hiện ví dụ lưu trữ chuyển động 4x4 có vị trí số nguyên và vị trí nửa pixel (vị trí phân số, đường biên tách và khoảng cách giữa mẫu và đường biên tách (sample_dist))

Trong ví dụ, quyết định liệu có lưu trữ thông tin chuyển động dự báo đơn thứ nhất, được dán nhãn $Mv0$, thông tin chuyển động dự báo đơn thứ hai, được dán nhãn $Mv1$, hoặc thông tin chuyển động kết hợp, chẳng hạn thông tin chuyển động dự báo kép, được dán nhãn $Mv2$, được xem xét, theo các bước sau. Trong ví dụ, quyết định được thực thi đối với đơn vị lưu trữ chuyển động; trong ví dụ khác, quyết định được thực thi đối với ít nhất hai (hoặc tất cả) đơn vị lưu trữ chuyển động được chứa trong khối mã. Tọa độ trên bên trái của đơn vị lưu trữ chuyển động là $(x/K, y/K)$, và chiều rộng và chiều cao của đơn vị lưu trữ chuyển động bằng K theo các mẫu độ sáng.

Nếu khoảng cách được tính toán sample_dist smaller than và/hoặc bằng ngưỡng, thông tin chuyển động kết hợp ($Mv2$) được lưu trữ đối với đơn vị lưu trữ chuyển động. Ngưỡng có thể là số cố định, có thể được dẫn xuất thích ứng dựa trên góc của đường tách, khuôn dạng của khối mã hiện tại hoặc các tham số khác; ngược lại (nếu khoảng cách sample_dist lớn hơn và/hoặc bằng ngưỡng), dấu của sample_dist được kiểm tra:

Nếu mẫu_dist là âm (hoặc dương), $MV0$ (hoặc $MV1$) được lưu trữ cho đơn vị lưu trữ chuyển động hiện tại,

Ngược lại (mẫu_dist là dương (hoặc âm)), $MV1$ (hoặc $MV0$) được lưu trữ cho đơn vị lưu trữ chuyển động hiện tại.

Fig.19 thể hiện một ví dụ lưu trữ chuyển động dựa trên phương pháp được nêu.

So với phương pháp lưu trữ chuyển động khác, phương pháp này chỉ cần tính toán một khoảng cách mẫu của mỗi đơn vị lưu trữ chuyển động, giảm độ phức tạp tính toán.

Theo phương án thực hiện thứ ba, quá trình giải mã đối với chế độ dự báo GEO, như được thể hiện trên Fig.20, như sau:

Đối với kỹ thuật tạo mã video cho phép dự báo kép, hai danh sách khung tham chiếu được sử dụng, chẳng hạn, List0 và List1, hoặc danh sách khung tham chiếu thuận và danh sách khung tham chiếu ngược. Lấy làm ví dụ, thông tin chuyển động bao gồm:

(1) Cờ dự báo L0 (PFL0) và Cờ dự báo L1 (PFL1), trong đó khi cờ dự báo L0 là true, List0 sẽ được sử dụng trong dự báo ngoài, ngược lại (Cờ dự báo L0 là false), List0 sẽ không được sử dụng. Cờ dự báo L1 là tương tự.

(2) Chỉ số tham chiếu L0 (RIL0) và Chỉ số tham chiếu L1 (RIL1), Chỉ số tham chiếu L0 được sử dụng để chỉ báo khung tham chiếu nào (ứng viên) từ List0 được sử dụng làm khung tham chiếu hiện tại. Chỉ số tham chiếu L1 là tương tự. Và theo cách triển khai linh hoạt, chỉ số tham chiếu L0 bằng -1 nghĩa là không khung tham chiếu nào được sử dụng từ List0.

(3) Vector chuyển động L0 (MVL0) và Vector chuyển động L1 (MVL1), lần lượt tương ứng với List0 và List1 và cả hai vector đều có thành phần ngang và thành phần dọc.

Sau khi thông tin chuyển động được xác định cho khối hiện tại, theo cách triển khai linh hoạt, thông tin chuyển động có thể được lưu trữ và được sử dụng làm thông tin tham chiếu cho quá trình dự báo của các khối mã tương lai, chẳng hạn, làm thông tin chuyển động dự báo của khối lân cận trong chế độ dự báo không gian hoặc chế độ dự báo thời gian.

Thông tin chuyển động được lưu trữ trong bộ nhớ, cụ thể hơn, được lưu trữ trong các khối lưu trữ thông tin chuyển động theo phương án thực hiện.

Khối hiện tại có thể được tách thành nhiều tập hợp mẫu, trong đó mỗi tập hợp trong nhiều tập hợp mẫu có kích thước giống như tập hợp mẫu; hoặc thiết lập khối hiện tại dưới dạng tập hợp mẫu. Chẳng hạn, tập hợp mẫu có thể là mảng mẫu 4x4. Kích thước của tập hợp mẫu cũng có thể là 8x8, 16x16, 8x16 và v.v., vốn không bị giới hạn. Mỗi tập hợp trong nhiều tập hợp mẫu tương ứng với mỗi đơn vị lưu trữ thông tin chuyển động, do vậy thông tin chuyển động của tập hợp mẫu sẽ được lưu trữ trong khối lưu trữ thông tin chuyển động tương ứng. Cần

lưu ý rằng có hai trường hợp về “thông tin chuyển động của tập hợp mẫu”. Trường hợp 1: thông tin chuyển động được sử dụng để xác định giá trị dự báo của tập hợp mẫu; Trường hợp 2: thông tin chuyển động được gán cho tập hợp mẫu để lưu trữ.

Cũng theo phương án thực hiện này, tập hợp mẫu là tập hợp mẫu độ sáng, và theo phương án thực hiện khác, tập hợp mẫu có thể là tập hợp mẫu sắc độ.

Chế độ dự báo GEO được nêu trong phần nêu trên. Nói chung, trong chế độ dự báo GEO, khối hiện tại bao gồm khối phụ thứ nhất và khối phụ thứ hai, khối phụ thứ nhất và khối phụ thứ hai được tách riêng bởi đường biên chung, khối phụ thứ nhất tương ứng với thông tin chuyển động thứ nhất và khối phụ thứ hai tương ứng với thông tin chuyển động thứ hai. Khối phụ thứ nhất và khối phụ thứ hai có thể là phân vùng tam giác, phân vùng hình thang, hoặc phân vùng chữ nhật bất đối xứng và v.v., vốn không bị giới hạn. Và có thể hiểu rằng chẳng hạn, trong phân vùng tam giác, mỗi khối phụ là khối tam giác.

Lấy làm ví dụ theo phương án thực hiện này, cả thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai đến từ dự báo đơn. Thông tin chuyển động thứ nhất có thể chỉ bao gồm RIL0, MVL0, nghĩa là $PFL0=1$, $PFL1=0$; hoặc có thể chỉ bao gồm RIL1, MVL1, nghĩa là $PFL0=0$, $PFL1=1$; tương tự, thông tin chuyển động thứ hai có thể chỉ bao gồm RIL0, MVL0, nghĩa là $PFL0=1$, $PFL1=0$; hoặc có thể chỉ bao gồm RIL1, MVL1, nghĩa là $PFL0=0$, $PFL1=1$.

S201: xác định tham số góc và tham số khoảng cách.

Đường biên chung được xác định bởi tham số góc và tham số khoảng cách.

(1) Trong bộ giải mã:

Theo cách triển khai linh hoạt, phân tách trực tiếp tham số góc và tham số khoảng cách từ dòng bit.

Theo cách triển khai linh hoạt khác, phân tách bộ chỉ báo từ dòng bit, và thu được tham số góc và tham số khoảng cách dựa trên bộ chỉ báo. Bộ chỉ báo có thể là chỉ số, tham số góc và tham số khoảng cách có thể được xác định bằng cách kiểm tra bảng tra cứu với chỉ số này. Hoặc, tham số góc và tham số khoảng cách có thể được tính toán bằng hàm, và bộ chỉ báo là đầu vào của hàm.

Theo cách triển khai linh hoạt khác, phân tách hai bộ chỉ báo từ dòng bit, và thu được tham số góc từ một trong hai bộ chỉ báo và tham số khoảng cách từ bộ còn lại trong hai bộ chỉ báo.

Theo phương án thực hiện này, phần tử cú pháp “geo_partition_idx” được phân tách từ dòng bit. Tham số góc “angleIdx” và tham số khoảng cách “distanceIdx” thu được bằng cách tìm kiếm bảng tra cứu, định nghĩa mối quan hệ giữa “geo_partition_idx” và cặp “angleIdx” và “distanceIdx”. Bảng tra cứu, chẳng hạn, có thể là bảng 1.

S202: xác định tham số tính toán thứ nhất dựa trên tham số góc; tính toán tham số góc tạm thời dựa trên tham số góc; xác định tham số tính toán thứ hai dựa trên tham số góc tạm thời; và tính toán tham số tính toán thứ ba dựa trên tham số góc và tham số khoảng cách.

Tham số tính toán thứ nhất được xác định bằng cách kiểm tra bảng tra cứu định trước theo tham số góc và tham số tính toán thứ hai được xác định bằng cách kiểm tra bảng tra cứu theo tham số góc tạm thời, và trong đó tham số tính toán thứ nhất và tham số tính toán thứ hai lần lượt là giá trị cosin và giá trị sin của cùng góc. Có thể hiểu rằng các phép tính cosin và sin có thể được mô phỏng và thay thế bằng phương pháp bảng tra cứu, và độ phức tạp tính toán được giảm.

Theo phương án thực hiện này,

Theo cách triển khai linh hoạt: Tham số tính toán thứ nhất được gán bằng $\text{Dis}[\text{angleIdx}]$, $\text{Dis}[]$ là bảng tra cứu, chẳng hạn, có thể là bảng 3.

Tham số góc tạm thời được gán bằng $(\text{angleIdx} + 8) \% 32$, giả sử rằng TempIdx .

Tham số tính toán thứ hai được gán bằng $\text{Dis}[\text{TempIdx}]$.

Cần lưu ý rằng về mặt lý thuyết, $\text{Dis}[\text{angleIdx}]$ và $\text{Dis}[\text{TempIdx}]$ là giá trị cosin xấp xỉ và giá trị sin của cùng góc, và góc này là mối quan hệ hình học giữa đường biên chung và khối hiện tại.

Theo cách triển khai linh hoạt khác: Tham số tính toán thứ nhất được gán bằng $\text{Dis1}[\text{angleIdx}]$, $\text{Dis1}[]$ là bảng tra cứu, chẳng hạn, có thể là bảng 3.

Tham số tính toán thứ hai được gán bằng $\text{Dis2}[\text{angleIdx}]$, $\text{Dis2}[]$ là bảng tra cứu khác.

Cần lưu ý rằng $\text{Dis1}[\text{angleIdx}]$ và $\text{Dis2}[\text{angleIdx}]$ là giá trị cosin và giá trị sin xấp xỉ của cùng góc, và góc này là mối quan hệ hình học giữa đường biên chung và khối hiện tại.

Tham số tính toán thứ ba thu được như sau:

$wIdx$ được gán bằng $\log_2(nCbW) - 3$, $nCbW$ là chiều rộng của khối hiện tại;

$hIdx$ được gán bằng $\log_2(nCbH) - 3$, $nCbH$ là chiều cao của khối hiện tại;

stepOffset được gán bằng số nguyên dương, chẳng hạn, được gán bằng 64.

So sánh $wIdx$ và $hIdx$:

$\text{whRatio} = (wIdx \geq hIdx) ? wIdx - hIdx : hIdx - wIdx$

$\text{wIsLargerH} = (wIdx \geq hIdx) ? \text{true} : \text{false}$

$\text{scaleStep} = (wIdx \geq hIdx) ? (1 \ll hIdx) : (1 \ll wIdx)$

Có thể hiểu rằng theo cách triển khai linh hoạt khác:

$\text{whRatio} = (wIdx > hIdx) ? wIdx - hIdx : hIdx - wIdx$

$\text{wIsLargerH} = (wIdx > hIdx) ? \text{true} : \text{false}$

$\text{scaleStep} = (wIdx > hIdx) ? (1 \ll hIdx) : (1 \ll wIdx)$

angleN được gán bằng:

$\text{angleN} = \text{angleIdx}$ nếu $\text{angleIdx} \geq 0 \ \&\& \ \text{angleIdx} \leq 8$,

$\text{angleN} = 16 - \text{angleIdx}$ nếu $\text{angleIdx} > 8 \ \&\& \ \text{angleIdx} \leq 16$,

$\text{angleN} = \text{angleIdx} - 16$ nếu $\text{angleIdx} > 16 \ \&\& \ \text{angleIdx} \leq 24$,

$\text{angleN} = 32 - \text{angleIdx}$ ngược lại,

Cần lưu ý rằng theo các triển khai khác nhau, các điểm đầu mút angleIdx 0, 8, 16, 24 có thể thuộc phần phụ khác, chẳng hạn:

Theo cách triển khai linh hoạt khác, angleN được gán bằng:

$\text{angleN} = \text{angleIdx}$ nếu $\text{angleIdx} > 0 \ \&\& \ \text{angleIdx} < 8$,

$\text{angleN} = 16 - \text{angleIdx}$ nếu $\text{angleIdx} \geq 8 \ \&\& \ \text{angleIdx} < 16$,

$\text{angleN} = \text{angleIdx} - 16$ nếu $\text{angleIdx} \geq 16 \ \&\& \ \text{angleIdx} < 24$,

$\text{angleN} = 32 - \text{angleIdx}$ ngược lại,

Ngoài ra, angleN được gán bằng giá trị sau:

$\text{angleN} = (\text{wIsLargerH}) ? 8 - \text{angleN} : \text{angleN}$

stepDis được gán theo các giá trị của whRatio và angleN như được xác định trong bảng tra cứu, chẳng hạn, Bảng 2.

$$\text{stepSize} = \text{stepDis} + \text{stepOffset}$$

Tham số tính toán thứ ba rho được gán bằng giá trị sau sử dụng bảng tra cứu được ký hiệu là Dis, chẳng hạn, Bảng 3.

$$\text{rho} = \text{distanceIdx} * \text{stepSize} * \text{scaleStep} + \text{nCbW} * \text{Dis}[\text{angleIdx}] + \text{nCbH} * \text{Dis}[\text{TempIdx}] .$$

và một cách tương ứng, theo cách triển khai linh hoạt khác, tham số tính toán thứ ba rho có thể được gán bằng:

$$\text{rho} = \text{distanceIdx} * \text{stepSize} * \text{scaleStep} + \text{nCbW} * \text{Dis1}[\text{angleIdx}] + \text{nCbH} * \text{Dis2}[\text{angleIdx}] .$$

Cần lưu ý rằng do sau tham số góc và tham số khoảng cách, tham số tính toán thứ nhất, thứ hai và thứ ba có thể được xác định. Theo cách triển khai linh hoạt, tham số tính toán thứ nhất, thứ hai và thứ ba sẽ chỉ được tính toán một lần dưới dạng các tham số cho tất cả các mẫu trong khối hiện tại. Và bước S201 và S202 có thể được bỏ qua đối với quá trình lưu trữ thông tin chuyển động nếu tham số tính toán thứ nhất, thứ hai và thứ ba của khối hiện tại đã được xác định.

S203: xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung.

khoảng cách được tính toán như sau:

$$\text{distFromLine} = (x + K) * P1 + (y + K) * P2 - P3$$

trong đó P1, P2 và P3 lần lượt là tham số tính toán thứ nhất, tham số tính toán thứ hai và tham số tính toán thứ ba, distFromLine là khoảng cách, K là số nguyên không âm, x là tọa độ ngang của vị trí đích trong tập hợp mẫu, y là tọa độ dọc của vị trí đích trong hệ tọa độ chữ nhật, trong đó vị trí của mẫu trên bên trái của khối hiện tại được gán bằng gốc tọa độ, hướng bên phải được gán bằng phương ngang dương và chiều xuống được gán bằng phương dọc dương.

Theo cách triển khai linh hoạt, K bằng 1.

$$\text{distFromLine} = (x + 1) * P1 + (y + 1) * P2 - P3 .$$

Cần lưu ý rằng khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung, nói chung, đó là khoảng cách giữa vị trí đích trong tập hợp mẫu và đường biên chung.

Vị trí đích trong tập hợp mẫu được định trước. Chẳng hạn, vị trí định trước có thể là vị trí trên bên trái của tập hợp mẫu, hoặc vị trí tâm của tập hợp mẫu, hoặc vị trí khác của tập hợp mẫu.

Cần lưu ý rằng vị trí đích có thể là vị trí của vị trí mẫu số nguyên trong tập hợp mẫu, và vị trí đích cũng có thể là vị trí của vị trí mẫu phân số. Chẳng hạn, vị trí đích có thể là vị trí của tâm trọng lực của khối hiện tại. Cụ thể hơn, nếu khối hiện tại là mảng mẫu 4x4, vị trí đích có thể là vị trí có các tọa độ (1.5, 1.5) trong hệ tọa độ chữ nhật theo phương án thực hiện hiện tại.

Theo cách triển khai linh hoạt, độ chính xác mẫu của khoảng cách cao hơn mẫu số nguyên. Một cách tương ứng, khoảng cách được tính toán như sau:

$$\text{distFromLine} = ((x < N) + K) * P1 + ((y < N) + K) * P2 - P3$$

trong đó 2^N là nghịch đảo của độ chính xác mẫu của khoảng cách, chẳng hạn, khi độ chính xác mẫu của khoảng cách bằng độ chính xác $\frac{1}{2}$ pixel, N bằng 1, khi độ chính xác mẫu của khoảng cách có độ chính xác pixel $\frac{1}{4}$, N bằng 2.

Lấy làm ví dụ, khoảng cách được tính toán như sau:

$$\text{distFromLine} = ((x < 1) + 1) * \text{Dis}[\text{angleIdx}] + ((y < 1) + 1) * \text{Dis}[\text{TempIdx}] - \text{rho}$$

S204: so sánh khoảng cách có ngưỡng để xác định liệu thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu.

Thông tin chuyển động thứ ba được dẫn xuất bằng thông tin chuyển động thứ nhất và thông tin thứ hai. Theo cách triển khai linh hoạt, thông tin chuyển động thứ ba là kết hợp của thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai, giống như thông tin chuyển động để dự báo kép. Cụ thể hơn là, nếu thông tin chuyển động thứ nhất bao gồm RILX_first, MVLX_first, thông tin chuyển động thứ hai bao gồm RILY_second, MVLY_second, trong đó X có thể bằng 0 hoặc 1, và Y có thể bằng 0 hoặc 1, thông tin chuyển động thứ ba có thể bao gồm tất cả RILX_first, MVLX_first, RILY_second, MVLY_second, nghĩa là PFL0=1, PFL1 = 1.

Theo cách triển khai linh hoạt khác thông tin chuyển động thứ ba có thể bằng một trong thông tin chuyển động thứ nhất hoặc thông tin chuyển động thứ hai trong trường hợp mà cả thông tin chuyển động thứ nhất và thông tin chuyển động

thứ hai tương ứng với danh sách tương tự (List0 hoặc List1). Chẳng hạn, nếu PFL0 bằng 1 và PFL1 bằng 0 đối với cả thông tin chuyển động thứ nhất và thứ hai, thông tin chuyển động thứ ba có thể được gán bằng thông tin chuyển động thứ nhất (hoặc thông tin chuyển động thứ hai, theo cách thực hiện khác). Tương tự, nếu PFL0 bằng 0 và PFL1 bằng 1 đối với cả thông tin chuyển động thứ nhất và thứ hai, thông tin chuyển động thứ ba có thể được gán bằng thông tin chuyển động thứ nhất (hoặc thông tin chuyển động thứ hai, theo cách thực hiện khác).

Theo cách triển khai linh hoạt, ngưỡng có thể là giá trị định trước.

Theo cách triển khai linh hoạt khác, ngưỡng dựa trên mối quan hệ hình học giữa đường biên chung và phương ngang, hoặc dựa trên mối quan hệ hình học giữa đường biên chung và phương dọc. Cụ thể hơn là, ngưỡng dựa trên góc giữa phương vuông góc của đường biên chung và phương ngang hoặc phương dọc. Và theo các cách thực hiện khác, mối quan hệ hình học có thể bao gồm mối quan hệ độ dài giữa đường biên chung và đường biên khối của khối hiện tại hoặc khối phụ của khối hiện tại. Theo phương án thực hiện này, ngưỡng có thể thu được bằng bảng tra cứu, chẳng hạn Bảng 5, theo góc được tính toán trên đây.

$\text{Threshold} = \text{RhoMax}[\text{angleN}]$.

Cần lưu ý rằng do angleN có thể được xác định trước bước S204, ngưỡng có thể được xác định trước bước S204 và cho khối hiện tại.

Ngưỡng cũng có thể là

$\text{threshLower} = -\text{RhoMax}[\text{angleN}]$

$\text{threshUpper} = \text{RhoMax}[\text{angleN}]$.

Bảng 5 – Bảng RhoMax để dẫn xuất các trọng số chuyển động phân vùng hình học.

Idx	0	1	2	3	4	5	6	7	8
RhoMax[idx]	1	2	2	3	3	3	2	2	1

Theo cách triển khai linh hoạt khác ngưỡng cũng có thể là

$\text{threshLower} = -K$

$\text{threshUpper} = K$, trong đó K là số dương định trước. Chẳng hạn giá trị của K có thể bằng 1. Trong một ví dụ K có thể thu được theo bộ chỉ báo trong dòng bit.

Theo cách triển khai linh hoạt khác, ngưỡng dựa trên chiều rộng và/hoặc chiều cao của khối hiện tại.

Cần lưu ý rằng ngưỡng cũng có thể được xác định bằng hai hoặc nhiều điều kiện nêu trên. Chẳng hạn, chiều rộng và/hoặc chiều cao của khối hiện tại có thể xác định tỷ lệ, $ratio = 1 < \text{abs}(\log_2(\text{width}) - \log_2(\text{height}))$, và sau đó tỷ lệ này được sử dụng để chia tỷ lệ ngưỡng được xác định bằng góc. Và ngưỡng được chia tỷ lệ được gán bằng ngưỡng cuối cùng.

Các phép so sánh giữa ngưỡng và khoảng cách được thực hiện:

S204-1: Xác định thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu trong trường hợp mà giá trị tuyệt đối của khoảng cách nhỏ hơn hoặc bằng ngưỡng;

S204-2: trong trường hợp mà giá trị tuyệt đối của khoảng cách lớn hơn ngưỡng:

xác định thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là dương (hoặc âm); hoặc,

xác định thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là âm (hoặc dương, theo nhánh nêu trên).

Có thể hiểu rằng theo cách thực hiện khác, điều kiện ở bước S204-1 cũng có thể là giá trị tuyệt đối của khoảng cách nhỏ hơn ngưỡng, do vậy, điều kiện ở bước S204-2 có thể là giá trị tuyệt đối của khoảng cách lớn hơn hoặc bằng ngưỡng.

Theo cách triển khai linh hoạt khác, phép tính toán giá trị tuyệt đối được thay bằng nhiều phần phụ hơn.

Theo phương án thực hiện này, thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu được lưu trữ trong trường hợp mà khoảng cách nhỏ hơn threshLower; Thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu được lưu trữ trong trường hợp mà khoảng cách lớn hơn threshUpper; Ngược lại, thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu được lưu trữ.

Và có thể hiểu rằng một trong thông tin chuyển động thứ nhất, thông tin chuyển động thứ hai và thông tin chuyển động thứ ba được lưu trữ trong đơn vị lưu trữ thông tin chuyển động tương ứng với tập hợp mẫu.

Cần lưu ý rằng khi khối hiện tại có thể được tách thành nhiều tập hợp mẫu, theo cách triển khai linh hoạt, các bước nêu trên theo phương án thực hiện này được sử dụng cho tất cả các tập hợp mẫu. Theo cách triển khai linh hoạt khác, các bước nêu trên theo phương án thực hiện này được sử dụng cho duy nhất một tập hợp mẫu. Theo cách triển khai linh hoạt khác, các bước nêu trên theo phương án thực hiện này được sử dụng cho ít nhất hai tập hợp mẫu. Đối với các tập hợp mẫu mà không được triển khai với các bước nêu trên, theo cách triển khai linh hoạt, thông tin chuyển động mặc định thông tin chuyển động được suy ra theo điều kiện định trước sẽ được lưu trữ.

Cần lưu ý rằng chỉ giá trị khoảng cách cần được tính toán dựa trên môđun phân vùng GEO được định trước (hàm), có tham số môđun có thể được xác định ở mức CU, để xác định lưu trữ thông tin chuyển động đối với phân vùng GEO. Độ phức tạp tính toán bị giới hạn.

Theo phương án thực hiện thứ tư, quá trình giải mã cho chế độ dự báo GEO:

Quá trình này được gọi khi giải mã CU với MergeTriangleFlag[xCb][yCb] bằng 1 hoặc với geo_flag[xCb][yCb] bằng 1.

Các đầu vào quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của khối mã hiện tại trong mẫu độ sáng,
- biến cbHeight xác định chiều cao của khối mã hiện tại trong mẫu độ sáng,
- các MV độ sáng có độ chính xác mẫu phân số 1/16 mvA và mvB,
- các MV sắc độ mvCA và mvCB,
- các chỉ số tham chiếu refIdxA và refIdxB,
- các cờ danh sách dự báo predListFlagA và predListFlagB.

Đầu ra của quá trình này là:

- mảng (cbWidth)x(cbHeight) predSamplesL của các mẫu dự báo độ sáng,

- mảng $(cbWidth / 2) \times (cbHeight / 2)$ $predSamples_{Cb}$ của các mẫu dự báo sắc độ đối với thành phần Cb,

- mảng $(cbWidth / 2) \times (cbHeight / 2)$ $predSamples_{Cr}$ của các mẫu dự báo sắc độ đối với thành phần Cr.

Giả sử $predSamples_{LA_L}$ và $predSamples_{LB_L}$ là các mảng $(cbWidth) \times (cbHeight)$ của các giá trị mẫu độ sáng được dự báo và $predSamples_{LA_{Cb}}$, $predSamples_{LB_{Cb}}$, $predSamples_{LA_{Cr}}$ và $predSamples_{LB_{Cr}}$ là các mảng $(cbWidth / 2) \times (cbHeight / 2)$ của các giá trị mẫu sắc độ được dự báo.

$predSamples_L$, $predSamples_{Cb}$ và $predSamples_{Cr}$ được dẫn xuất bởi các bước có thứ tự sau:

3. Với N là một trong A và B, phương trình sau áp dụng:

- ảnh tham chiếu gồm mảng hai chiều được sắp thứ tự $refPicLN_L$ của các mẫu độ sáng và hai mảng hai chiều được sắp thứ tự $refPicLN_{Cb}$ và $refPicLN_{Cr}$ của các mẫu sắc độ được dẫn xuất với X được gán bằng $predListFlag_N$ và $refIdxX$ được gán bằng $refIdx_N$ làm đầu vào.

- Mảng $predSamplesLN_L$ được dẫn xuất bằng gọi quá trình nội suy mẫu phân số với vị trí độ sáng (xCb, yCb) , chiều rộng khối mã độ sáng $sbWidth$ được gán bằng $cbWidth$, chiều cao khối mã độ sáng $sbHeight$ được gán bằng $cbHeight$, độ lệch MV $mvOffset$ được gán bằng $(0, 0)$, MV $mvLX$ được gán bằng mvN và mảng tham chiếu $refPicLX_L$ được gán bằng $refPicLN_L$, biến $bdofFlag$ được gán bằng FALSE, và biến $cIdx$ được gán bằng 0 làm đầu vào.

- Mảng $predSamplesLN_{Cb}$ được dẫn xuất bằng gọi quá trình nội suy mẫu phân số với vị trí độ sáng (xCb, yCb) , chiều rộng khối mã $sbWidth$ được gán bằng $cbWidth / 2$, chiều cao khối mã $sbHeight$ được gán bằng $cbHeight / 2$, độ lệch MV $mvOffset$ được gán bằng $(0, 0)$, MV $mvLX$ được gán bằng $mvCN$, và mảng tham chiếu $refPicLX_{Cb}$ được gán bằng $refPicLN_{Cb}$, biến $bdofFlag$ được gán bằng FALSE, và biến $cIdx$ được gán bằng 1 làm đầu vào.

- Mảng $predSamplesLN_{Cr}$ được dẫn xuất bằng gọi quá trình nội suy mẫu phân số với vị trí độ sáng (xCb, yCb) , chiều rộng khối mã $sbWidth$ được gán bằng $cbWidth / 2$, chiều cao khối mã $sbHeight$ được gán bằng $cbHeight / 2$, độ lệch MV $mvOffset$ được gán bằng $(0, 0)$, MV $mvLX$ được gán bằng $mvCN$, và

mảng tham chiếu refPicLX_{Cr} được gán bằng refPicLN_{Cr} , biến bdoFlag được gán bằng FALSE, và biến cIdx được gán bằng 2 làm đầu vào.

4. Nếu $\text{MergeTriangleFlag}[\text{xCb}][\text{yCb}]$ bằng 1, hướng phân vùng của biến chế độ tam giác hợp nhất triangleDir được gán bằng $\text{merge_triangle_split_dir}[\text{xCb}][\text{yCb}]$.

a. Các mẫu dự báo bên trong khối mã độ sáng hiện tại, $\text{predSamplesL}[\text{xL}][\text{yL}]$ với $\text{xL} = 0..\text{cbWidth} - 1$ và $\text{yL} = 0..\text{cbHeight} - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất tam giác với chiều rộng khối mã nCbW được gán bằng cbWidth , chiều cao khối mã nCbH được gán bằng cbHeight , các mảng mẫu predSamplesLA_L và predSamplesLB_L , và các biến triangleDir , và cIdx bằng 0 làm đầu vào.

b. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cb , $\text{predSamplesCb}[\text{xC}][\text{yC}]$ với $\text{xC} = 0..\text{cbWidth} / 2 - 1$ và $\text{yC} = 0..\text{cbHeight} / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất tam giác với chiều rộng khối mã nCbW được gán bằng $\text{cbWidth} / 2$, chiều cao khối mã nCbH được gán bằng $\text{cbHeight} / 2$, các mảng mẫu $\text{predSamplesLA}_{Cb}$ và $\text{predSamplesLB}_{Cb}$, và các biến triangleDir , và cIdx bằng 1 làm đầu vào.

c. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cr , $\text{predSamplesCr}[\text{xC}][\text{yC}]$ với $\text{xC} = 0..\text{cbWidth} / 2 - 1$ và $\text{yC} = 0..\text{cbHeight} / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất tam giác với chiều rộng khối mã nCbW được gán bằng $\text{cbWidth} / 2$, chiều cao khối mã nCbH được gán bằng $\text{cbHeight} / 2$, các mảng mẫu $\text{predSamplesLA}_{Cr}$ và $\text{predSamplesLB}_{Cr}$, và các biến triangleDir , và cIdx bằng 2 làm đầu vào.

d. Quá trình lưu trữ MV đối với chế độ tam giác hợp nhất được gọi với vị trí khối mã độ sáng (xCb, yCb) , chiều rộng khối mã độ sáng cbWidth , chiều cao khối mã độ sáng cbHeight , hướng phân vùng triangleDir , các MV độ sáng mvA và mvB , các chỉ số tham chiếu refIdxA và refIdxB , và các cờ danh sách dự báo predListFlagA và predListFlagB làm đầu vào.

5. Nếu $\text{geo_flag}[\text{xCb}][\text{yCb}]$ bằng 1, các biến liên quan đến phân vùng hình học angleIdx và distanceIdx được thiết lập theo giá trị của $\text{geo_partitioning_idx}[\text{xCb}][\text{yCb}]$ như được xác định trong Bảng 1.

a. Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học được gọi. Đầu vào cho quá trình này lần lượt là ba mảng được sắp thứ tự sampleWeightL , sampleWeightC và motionWeight có kích thước $(\text{cbWidth}) \times (\text{cbHeight})$, $(\text{cbWidth}/2) \times (\text{cbHeight}/2)$ và $(\text{cbWidth}/4) \times (\text{cbHeight}/4)$ cũng như các biến angleIdx , distanceIdx , cbWidth và cbHeight .

b. Các mẫu dự báo bên trong khối mã độ sáng hiện tại, $\text{predSamplesL}[\text{xL}][\text{yL}]$ với $\text{xL} = 0..\text{cbWidth} - 1$ và $\text{yL} = 0..\text{cbHeight} - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học với chiều rộng khối mã nCbW được gán bằng cbWidth , chiều cao khối mã nCbH được gán bằng cbHeight , các mảng mẫu predSamplesLAL , predSamplesLBL , predWeightL và cIdx bằng 0 làm đầu vào.

c. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cb , $\text{predSamplesCb}[\text{xC}][\text{yC}]$ với $\text{xC} = 0..\text{cbWidth} / 2 - 1$ và $\text{yC} = 0..\text{cbHeight} / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học với chiều rộng khối mã nCbW được gán bằng $\text{cbWidth} / 2$, chiều cao khối mã nCbH được gán bằng $\text{cbHeight} / 2$, các mảng mẫu predSamplesLACb , predSamplesLBCb , predWeightC và cIdx bằng 1 làm đầu vào.

d. Các mẫu dự báo bên trong khối mã thành phần sắc độ hiện tại Cr , $\text{predSamplesCr}[\text{xC}][\text{yC}]$ với $\text{xC} = 0..\text{cbWidth} / 2 - 1$ và $\text{yC} = 0..\text{cbHeight} / 2 - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học được xác định trong mệnh đề 8.5.8.3 với chiều rộng khối mã nCbW được gán bằng $\text{cbWidth} / 2$, chiều cao khối mã nCbH được gán bằng $\text{cbHeight} / 2$, các mảng mẫu predSamplesLACr , predSamplesLBCr , predWeightC và cIdx bằng 2 làm đầu vào.

e. Quá trình lưu trữ MV đối với chế độ hợp nhất hình học được gọi với vị trí khối mã độ sáng (xCb, yCb) , chiều rộng khối mã độ sáng cbWidth , chiều cao khối mã độ sáng cbHeight , mảng mẫu motionWeight , các MV độ sáng mvA và

mvB, các chỉ số tham chiếu refIdxA và refIdxB, và các cờ danh sách dự báo predListFlagA và predListFlagB làm đầu vào.

Quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác

Đầu vào cho quá trình này là:

- biến X biểu diễn danh sách tham chiếu bằng 0 hoặc 1,
- chỉ số tham chiếu refIdxN.

Đầu ra của quá trình này là:

- chỉ số tham chiếu refIdxTemp.

Biến refPicPoc được dẫn xuất như sau:

$$\text{refPicPoc} = (X == 0) ? \text{RefPicList}[0][\text{refIdxN}] : \text{RefPicList}[1][\text{refIdxN}]$$

(8-878)

danh sách ảnh tham chiếu refPicListTemp được dẫn xuất như sau:

$$\text{refPicListTemp} = (X == 0) ? \text{RefPicList}[1] : \text{RefPicList}[0] \quad (8-879)$$

Biến refIdxTemp được dẫn xuất như sau:

- Biến mapStop được gán bằng FALSE.
- Với biến refIdx_m có $m = 0.. \text{NumRefIdxActive}[1] - 1$, phương trình sau áp dụng cho đến khi mapStop bằng FALSE:

$$\text{refIdxTemp} = (\text{refPicListTemp}[\text{refIdx}_m] == \text{refPicPoc}) ? \text{refIdx}_m : -1 \quad (8-880)$$

$$\text{mapStop} = (\text{refIdxTemp} != -1) ? \text{TRUE} : \text{FALSE} \quad (8-881)$$

Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học

Các đầu vào quá trình này là:

- hai biến nCbW và nCbH xác định chiều rộng và chiều cao của khối mã hiện tại,
- hai biến angleIdx và distanceIdx xác định góc và các giá trị tra cứu khoảng cách của chế độ phân vùng hình học hiện tại,
- mảng (nCbW)x(nCbH) sampleWeight_L,
- mảng (nCbW/2)x(nCbH/2) sampleWeight_C,
- mảng (nCbW/4)x(nCbH/4) motionWeight.

Đầu ra của quá trình này là ba mảng giống nhau sampleWeight_L, sampleWeight_C và motionWeight của các giá trị trọng số theo mẫu có khoảng từ 0 ... 8.

4. Các trọng số kết hợp độ sáng được lưu trữ trong `sampleWeightL` cho chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

Giá trị của các biến sau được thiết lập:

- `wIdx` được gán bằng $\log_2(\text{nCbW}) - 3$,
- `hIdx` được gán bằng $\log_2(\text{nCbH}) - 3$,
- `stepOffset` được gán bằng 64,
- các biến sau được thiết lập theo phép so sánh (`wIdx >= hIdx`):
- $\text{whRatio} = (\text{wIdx} \geq \text{hIdx}) ? \text{wIdx} - \text{hIdx} : \text{hIdx} - \text{wIdx}$

(8-882)

- $\text{wIsLargerH} = (\text{wIdx} \geq \text{hIdx}) ? \text{true} : \text{false}$ (8-883)
- $\text{scaleStep} = (\text{wIdx} \geq \text{hIdx}) ? (1 \ll \text{hIdx}) : (1 \ll \text{wIdx})$

(8-884)

- `displacementX` được gán bằng `angleIdx`
- `displacementY` được gán bằng $(\text{displacementX} + 8) \% 32$
- `angleN` được gán bằng:

$$\begin{aligned} \text{angleN} &= \text{angleIdx} && \text{nếu } \text{angleIdx} \geq 0 \ \&\& \ \text{angleIdx} \leq 8, \\ \text{angleN} &= 16 - \text{angleIdx} && \text{nếu } \text{angleIdx} > 8 \ \&\& \ \text{angleIdx} \leq 16, \\ \text{angleN} &= \text{angleIdx} - 16 && \text{nếu } \text{angleIdx} > 16 \ \&\& \ \text{angleIdx} \leq 24, \\ \text{angleN} &= 32 - \text{angleIdx} && \text{ngược lại,} \end{aligned}$$
- Ngoài ra, `angleN` được gán bằng giá trị sau, phụ thuộc vào

`wIsLargerH`:

$$\text{angleN} = (\text{wIsLargerH}) ? 8 - \text{angleN} : \text{angleN}$$

- `stepDis` được gán theo các giá trị của `whRatio` và `angleN` như được xác định trong Bảng 2 và ngoài ra,

$$\text{stepSize} = \text{stepDis} + \text{stepOffset}$$

- `rho` được gán bằng giá trị sau using bảng tra cứu được ký hiệu là `Dis`, được xác định trong Bảng 3:

$$\text{rho} = \text{distanceIdx} * \text{stepSize} * \text{scaleStep} + \text{nCbW} * \text{Dis}[\text{displacementX}] + \text{nCbH} * \text{Dis}[\text{displacementY}] \quad (8-885)$$

Đối với mỗi vị trí mẫu $x=0 \dots \text{nCbW} - 1$ và $y=0 \dots \text{nCbH} - 1$, giá trị của `sampleWeightL[x][y]` được dẫn xuất theo cách sau:

– khoảng cách $\text{distFromLine}[x][y]$ được tính toán sử dụng bảng tra cứu được ký hiệu là Dis , được xác định trong Bảng 3 như sau:

$$\text{distFromLine}[x][y] = ((x < 1) + 1) * \text{Dis}[\text{displacementX}] + ((y < 1) + 1) * \text{Dis}[\text{displacementY}] - \rho \quad (8-886)$$

– Khoảng cách được chia tỷ lệ distScaled được dẫn xuất là:

$$\text{distScaled} = \min((\text{abs}(\text{distFromLine}[x][y]) + 8) \gg 4, 14). \quad (8-887)$$

Giá trị của $\text{sampleWeight}_L[x][y]$ được gán theo bảng tra cứu trọng số lọc, được ký hiệu là GeoFilter được xác định trong Bảng 4 bằng:

$$\text{sampleWeight}_L[x][y] = \text{GeoFilter}[\text{distScaled}] \quad \text{nếu } \text{distFromLine} \leq 0$$

$$\text{sampleWeight}_L[x][y] = 8 - \text{GeoFilter}[\text{distScaled}] \quad \text{nếu } \text{distFromLine} > 0$$

5. Các trọng số sắc độ được lưu trữ trong sampleWeight_C với chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

Đối với mỗi vị trí mẫu $x=0 \dots (n_{CbW}/2) - 1$ và $y=0 \dots (n_{CbH}/2) - 1$, giá trị của $\text{sampleWeight}_C[x][y]$ được dẫn xuất theo cách sau:

$$\text{sampleWeight}_C[x][y] = \text{sampleWeight}_L[(x < 1)][(y < 1)]$$

6. Các trọng số chuyển động được lưu trữ trong mảng motionWeight for chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

– Các biến sau được thiết lập theo Bảng 5:

$$\text{threshLower} = -\text{RhoMax}[\text{angleN}]$$

$$\text{threshUpper} = \text{RhoMax}[\text{angleN}]$$

Đối với mỗi vị trí mẫu $x=0 \dots (n_{CbW}/4) - 1$ và $y=0 \dots (n_{CbH}/4) - 1$ giá trị của $\text{motionWeight}[x][y]$ được dẫn xuất theo cách sau:

$\text{Cnt} = \text{distFromLine}[x*4 + 1][y*4 + 1]$ Giá trị của $\text{motionWeight}[x][y]$ được gán bằng:

$$\text{motionWeight}[x][y] = 0 \quad \text{nếu } \text{Cnt} \leq \text{threshLower}$$

$$\text{motionWeight}[x][y] = 1 \quad \text{nếu } \text{Cnt} \geq \text{threshUpper}$$

$$\text{motionWeight}[x][y] = 2 \quad \text{ngược lại}$$

Quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất phân vùng hình học

Các đầu vào quá trình này là:

- hai biến $nCbW$ và $nCbH$ xác định chiều rộng và chiều cao của khối mã hiện tại,
- hai mảng $(nCbW) \times (nCbH)$ $predSamplesLA$ và $predSamplesLB$,
- mảng $(nCbW) \times (nCbH)$ $sampleWeight$,
- biến $cIdx$ xác định chỉ số thành phần màu.

Đầu ra của quá trình này là mảng $(nCbW) \times (nCbH)$ $pbSamples$ của các giá trị mẫu dự báo.

Biến $bitDepth$ được dẫn xuất như sau:

- Nếu $cIdx$ bằng 0, $bitDepth$ được gán bằng $BitDepthY$.
- Ngược lại, $bitDepth$ được gán bằng $BitDepthC$.

Các biến $shift1$ và $offset1$ được dẫn xuất như sau:

- Biến $shift1$ được gán bằng $\text{Max}(5, 17 - bitDepth)$.
- Biến $offset1$ được gán bằng $1 \ll (shift1 - 1)$.

Các giá trị mẫu dự báo được dẫn xuất như sau:

$$pbSamples[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, \\ (predSamplesLA[x][y] * sampleWeight[x][y] \\ + predSamplesLB[x][y] * (8 - sampleWeight[x][y]) + offset1) \gg shift1)$$

Lưu trữ MV đối với chế độ hợp nhất phân vùng hình học

Quá trình này được gọi khi giải mã CU với $geo_flag[xCb][yCb]$ bằng 1.

Các đầu vào quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến $cbWidth$ xác định chiều rộng của khối mã hiện tại trong các mẫu độ sáng,
- biến $cbHeight$ xác định chiều cao của khối mã hiện tại trong các mẫu độ sáng,
- mảng $motionWeight$ có kích thước $(cbWidth/4) \times (cbHeight/4)$ chứa các chỉ số phân định chuyển động,
- các MV độ sáng có độ chính xác mẫu phân số $1/16$ mvA và mvB ,

- các chỉ số tham chiếu refIdxA và refIdxB ,
- các cờ danh sách dự báo predListFlagA và predListFlagB .

Các biến numSbX và numSbY xác định số lượng khối 4×4 trong khối mã hiện tại theo chiều ngang và chiều dọc được gán bằng $\text{numSbX} = \text{cbWidth} \gg 2$ và $\text{numSbY} = \text{cbHeight} \gg 2$.

Biến minSb được gán bằng $\min(\text{numSbX}, \text{numSbY})$.

Biến refIdxTempA được dẫn xuất bằng gọi quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác được xác định trong mệnh đề 8.5.7.4 với X được gán bằng predListFlagA và refIdxN được gán bằng refIdxA làm đầu vào.

Biến refIdxTempB được dẫn xuất bằng gọi quá trình ánh xạ ảnh tham chiếu đối với chế độ hợp nhất tam giác được xác định trong clause 8.5.7.4 với X được gán bằng predListFlagB và refIdxN được gán bằng refIdxB làm đầu vào.

Đối với mỗi khối phụ 4×4 ở chỉ số khối phụ $(\text{xSbIdx}, \text{ySbIdx})$ với $\text{xSbIdx} = 0..\text{numSbX} - 1$, và $\text{ySbIdx} = 0..\text{numSbY} - 1$, phương trình sau áp dụng:

- Nếu $\text{motionWeight}[\text{xSbIdx}][\text{ySbIdx}]$ bằng 0, phương trình sau áp dụng:

```

predFlagL0 = (predListFlagA == 0) ? 1 : 0 (8-888)
predFlagL1 = (predListFlagA == 0) ? 0 : 1 (8-889)
refIdxL0 = (predListFlagA == 0) ? refIdxA : -1 (8-890)
refIdxL1 = (predListFlagA == 0) ? -1 : refIdxA (8-891)
mvL0[0] = (predListFlagA == 0) ? mvA[0] : 0 (8-892)
mvL0[1] = (predListFlagA == 0) ? mvA[1] : 0 (8-893)
mvL1[0] = (predListFlagA == 0) ? 0 : mvA[0] (8-894)
mvL1[1] = (predListFlagA == 0) ? 0 : mvA[1] (8-895)

```

- Ngược lại, nếu $\text{motionWeight}[\text{xSbIdx}][\text{ySbIdx}]$ bằng 1, phương trình sau áp dụng:

```

predFlagL0 = (predListFlagB == 0) ? 1 : 0 (8-896)
predFlagL1 = (predListFlagB == 0) ? 0 : 1 (8-897)
refIdxL0 = (predListFlagB == 0) ? refIdxB : -1 (8-898)
refIdxL1 = (predListFlagB == 0) ? -1 : refIdxB (8-899)

```

$mvL0[0] = (predListFlagB == 0) ? mvB[0] : 0$ (8-900)

$mvL0[1] = (predListFlagB == 0) ? mvB[1] : 0$ (8-901)

$mvL1[0] = (predListFlagB == 0) ? 0 : mvB[0]$ (8-902)

$mvL1[1] = (predListFlagB == 0) ? 0 : mvB[1]$ (8-903)

– Ngược lại ($motionWeight[xSbIdx][ySbIdx]$ bằng 2), phương trình sau áp dụng:

– Nếu $predListFlagA + predListFlagB$ bằng 1,

$predFlagL0 = 1$ (8-904)

$predFlagL1 = 1$ (8-905)

$refIdxL0 = (predListFlagA == 0) ? refIdxA : refIdxB$ (8-906)

$refIdxL1 = (predListFlagA == 0) ? refIdxB : refIdxA$ (8-907)

$mvL0[0] = (predListFlagA == 0) ? mvA[0] : mvB[0]$ (8-908)

$mvL0[1] = (predListFlagA == 0) ? mvA[1] : mvB[1]$ (8-909)

$mvL1[0] = (predListFlagA == 0) ? mvB[0] : mvA[0]$ (8-910)

$mvL1[1] = (predListFlagA == 0) ? mvB[1] : mvA[1]$ (8-911)

– Nếu $predListFlagA + predListFlagB$ is bằng 0, phương trình sau áp dụng:

$predFlagL0 = 1$ (8-912)

$predFlagL1 = (refIdxTempA == -1 \ \&\& \ refIdxTempB == -1) ? 0 : 1$
(8-913)

$refIdxL0 = (refIdxTempB != -1) ? refIdxA : ($ (8-914)

$((refIdxTempA != -1) ? refIdxB : refIdxA)$

$refIdxL1 = (refIdxTempB != -1) ? refIdxTempB : ($ (8-915)

$((refIdxTempA != -1) ? refIdxTempA : -1)$

$mvL0[0] = (refIdxTempB != -1) ? mvA[0] : ($ (8-916)

$((refIdxTempA != -1) ? mvB[0] : mvA[0])$

$mvL0[1] = (refIdxTempB != -1) ? mvA[1] : ($ (8-917)

$((refIdxTempA != -1) ? mvB[1] : mvA[1])$

$mvL1[0] = (refIdxTempB != -1) ? mvB[0] : ($ (8-918)

$((refIdxTempA != -1) ? mvA[0] : 0)$

$mvL1[1] = (refIdxTempB \neq -1) ? mvB[1] : (8-919)$

$((refIdxTempA \neq -1) ? mvA[1]:0)$

– Nếu $predListFlagA + predListFlagB$ bằng 2, phương trình sau áp dụng:

$predFlagL0 = (refIdxTempA == -1 \ \&\& \ refIdxTempB == -1) ? 0 : 1 \quad (8-920)$

$predFlagL1 = 1 \quad (8-921)$

$refIdxL0 = (refIdxTempB \neq -1) ? refIdxTempB : (8-922)$

$((refIdxTempA \neq -1) ? refIdxTempA : -1)$

$refIdxL1 = (refIdxTempB \neq -1) ? refIdxA : (8-923)$

$((refIdxTempA \neq -1) ? refIdxB : refIdxA)$

$mvL0[0] = (refIdxTempB \neq -1) ? mvB[0] : (8-924)$

$((refIdxTempA \neq -1) ? mvA[0]:0)$

$mvL0[1] = (refIdxTempB \neq -1) ? mvB[1] : (8-925)$

$((refIdxTempA \neq -1) ? mvA[1]:0)$

$mvL1[0] = (refIdxTempB \neq -1) ? mvA[0] : (8-926)$

$((refIdxTempA \neq -1) ? mvB[0]:mvA[0])$

$mvL1[1] = (refIdxTempB \neq -1) ? mvA[1] : (8-927)$

$((refIdxTempA \neq -1) ? mvB[1]:mvA[1])$

– Các phép gán sau được thực hiện với $x = 0..3$ và $y = 0..3$:

$MvL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = mvL0 \quad (8-928)$

$MvL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = mvL1 \quad (8-929)$

$RefIdxL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = refIdxL0 \quad (8-930)$

$RedIdxL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = refIdxL1 \quad (8-931)$

$PredFlagL0[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = predFlagL0 \quad (8-932)$

$PredFlagL1[(xSbIdx \ll 2) + x][(ySbIdx \ll 2) + y] = predFlagL1 \quad (8-933)$

Theo cách triển khai tùy chọn, bước 6 của “Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học” có thể được triển khai như sau:

6. Các trọng số chuyển động được lưu trữ trong mảng `motionWeight` với chế độ phân vùng hình học hiện tại được dẫn xuất như sau:

– Các biến sau được thiết lập theo Bảng 5:

threshLower = - RhoMax[angleN]

threshUpper = RhoMax[angleN]

Đối với mỗi vị trí mẫu $x=0\dots (nC_bW/4) - 1$ và $y=0\dots (nC_bH/4) - 1$, giá trị của $\text{motionWeight}[x][y]$ được dẫn xuất theo cách sau:

$$\text{Cnt} = ((x < 4) + 3) * \text{Dis}[\text{displacementX}] + ((y < 4) + 3) * \text{Dis}[\text{displacementY}] - \rho < 1$$

Giá trị của $\text{motionWeight}[x][y]$ được gán bằng:

$\text{motionWeight}[x][y] = 0$ if $\text{Cnt} \leq \text{threshLower}$

$\text{motionWeight}[x][y] = 1$ if $\text{Cnt} \geq \text{threshUpper}$

$\text{motionWeight}[x][y] = 2$ ngược lại

Theo triển khai ví dụ nêu trên của bước 6 của “Quá trình dẫn xuất trọng số mẫu đối với chế độ hợp nhất phân vùng hình học”, phương trình “ $\text{Cnt} = ((x < 4) + 3) * \text{Dis}[\text{displacementX}] + ((y < 4) + 3) * \text{Dis}[\text{displacementY}] - \rho < 1$ ” tương ứng với tính toán khoảng cách giữa tâm của trọng lực của tập hợp mẫu và đường thẳng. Tập hợp mẫu trong ví dụ này bao gồm 16 mẫu có các tọa độ sau:

$(x*4, y*4), (x*4+1, y*4), (x*4+2, y*4), (x*4+3, y*4), (x*4, y*4+1),$
 $(x*4+1, y*4+1), (x*4+2, y*4+1), (x*4+3, y*4+1), (x*4, y*4+2), (x*4+1, y*4+2),$
 $(x*4+2, y*4+2), (x*4+3, y*4+2), (x*4, y*4+3), (x*4+1, y*4+3), (x*4+2, y*4+3),$
 $(x*4+3, y*4+3).$

Các tọa độ của tâm của trọng lực là $(x*4 + 1,5, y*4 + 1,5)$. Trong phương trình trên tất cả các số hạng cộng tính của phương trình được nhân với 2 để khiến các tọa độ lệch 1,5 số nguyên (do vậy bằng 3 trong ví dụ này), do số học số nguyên là đặc điểm triển khai mong muốn.

Theo phương án thực hiện thứ năm, thiết bị lưu trữ thông tin chuyển động, như được thể hiện trên Fig.21, như sau:

Thiết bị lưu trữ thông tin chuyển động (3000), trong đó khối hiện tại bao gồm khối phụ thứ nhất và khối phụ thứ hai, khối phụ thứ nhất và khối phụ thứ hai được tách riêng bởi đường biên chung, khối phụ thứ nhất tương ứng với thông tin chuyển động thứ nhất và khối phụ thứ hai tương ứng với thông tin chuyển động thứ hai, thiết bị (3000) bao gồm: môđun tính toán (3001), được tạo cấu hình để xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên

chung; và môđun so sánh (3002), được tạo cấu hình để so sánh khoảng cách với ngưỡng để xác định liệu thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu, trong đó thông tin chuyển động thứ ba được dẫn xuất bằng thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai.

Theo cách triển khai linh hoạt, trong đó môđun so sánh (3002) được tạo cấu hình để:

xác định thông tin chuyển động thứ ba được lưu trữ đối với tập hợp mẫu trong trường hợp mà giá trị tuyệt đối của khoảng cách nhỏ hơn hoặc bằng ngưỡng.

Theo cách triển khai linh hoạt, trong đó trong trường hợp mà giá trị tuyệt đối của khoảng cách lớn hơn ngưỡng, môđun so sánh (3002) còn được tạo cấu hình để: xác định thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là dương; hoặc, xác định thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là âm.

Theo cách triển khai linh hoạt, trong đó ngưỡng được định trước.

Theo cách triển khai linh hoạt, trong đó ngưỡng dựa trên mối quan hệ hình học giữa đường biên chung và phương ngang hoặc phương dọc.

Theo cách triển khai linh hoạt, trong đó ngưỡng dựa trên góc giữa phương vuông góc của đường biên chung và phương ngang hoặc phương dọc.

Theo cách triển khai linh hoạt, trong đó ngưỡng dựa trên chiều rộng và/hoặc chiều cao của khối hiện tại.

Theo cách triển khai linh hoạt, trong đó đường biên chung được xác định bởi tham số góc và tham số khoảng cách, môđun tính toán (3001) còn được tạo cấu hình để: xác định tham số góc và tham số khoảng cách.

Theo cách triển khai linh hoạt, trong đó trong bộ giải mã, môđun tính toán (3001) còn được tạo cấu hình để: phân tách tham số góc và tham số khoảng cách từ dòng bit; hoặc, phân tách bộ chỉ báo từ dòng bit, và obtain tham số góc và tham số khoảng cách dựa trên bộ chỉ báo.

Theo cách triển khai linh hoạt, trong đó môđun tính toán (3001) còn được tạo cấu hình để: xác định tham số tính toán thứ nhất dựa trên tham số góc; tính toán tham số góc tạm thời dựa trên tham số góc; xác định tham số tính toán thứ

hai dựa trên tham số góc tạm thời; và tính toán tham số tính toán thứ ba dựa trên tham số góc và tham số khoảng cách.

Theo cách triển khai linh hoạt, trong đó tham số tính toán thứ nhất được xác định bằng cách kiểm tra bảng tra cứu định trước theo tham số góc và tham số tính toán thứ hai được xác định bằng cách kiểm tra bảng tra cứu theo tham số góc tạm thời, và trong đó tham số tính toán thứ nhất và tham số tính toán thứ hai lần lượt là giá trị cosin và giá trị sin của cùng góc.

Theo cách triển khai linh hoạt, trong đó khoảng cách được tính toán như sau:

$$\text{distFromLine} = (x + K) * P1 + (y + K) * P2 - P3$$

trong đó P1, P2 và P3 lần lượt là tham số tính toán thứ nhất, tham số tính toán thứ hai và tham số tính toán thứ ba, distFromLine là khoảng cách, K là số nguyên không âm, x là tọa độ ngang của vị trí đích trong tập hợp mẫu, y là tọa độ dọc của vị trí đích trong hệ tọa độ chữ nhật, trong đó vị trí của mẫu trên bên trái của khối hiện tại được gán bằng gốc tọa độ, hướng bên phải được gán bằng phương ngang dương và chiều xuống được gán bằng phương dọc dương.

Theo cách triển khai linh hoạt, trong đó K bằng 1.

Theo cách triển khai linh hoạt, trong đó vị trí đích được định trước trong tập hợp mẫu.

Theo cách triển khai linh hoạt, trong đó vị trí định trước là vị trí trên bên trái của tập hợp mẫu, hoặc vị trí tâm của tập hợp mẫu.

Theo cách triển khai linh hoạt, trong đó độ chính xác mẫu của khoảng cách cao hơn mẫu số nguyên;

một cách tương ứng, khoảng cách được tính toán như sau:

$$\text{distFromLine} = ((x \ll N) + K) * P1 + ((y \ll N) + K) * P2 - P3$$

trong đó 2N là nghịch đảo của độ chính xác mẫu của khoảng cách.

Theo cách triển khai linh hoạt, trong đó N bằng 1.

Theo cách triển khai linh hoạt, trong đó tập hợp mẫu là tập hợp mẫu độ sáng của khối hiện tại.

Theo cách triển khai linh hoạt, trong đó thiết bị (3000) còn bao gồm môđun tách (3003), được tạo cấu hình để: tách khối hiện tại thành nhiều tập hợp mẫu,

trong đó mỗi tập hợp trong nhiều tập hợp mẫu có kích thước giống như tập hợp mẫu; hoặc thiết lập khối hiện tại như là tập hợp mẫu.

Theo cách triển khai linh hoạt, trong đó tập hợp mẫu là mảng mẫu 4x4.

Theo cách triển khai linh hoạt, trong đó mỗi tập hợp trong nhiều tập hợp mẫu tương ứng với mỗi đơn vị lưu trữ thông tin chuyển động, đơn vị lưu trữ thông tin chuyển động được sử dụng để lưu trữ một trong thông tin chuyển động thứ nhất, thông tin chuyển động thứ hai và thông tin chuyển động thứ ba.

Theo cách triển khai linh hoạt, trong đó khối phụ thứ nhất và khối phụ thứ hai là phân vùng tam giác, phân vùng hình thang, hoặc phân vùng hình chữ nhật bất đối xứng.

Theo cách triển khai linh hoạt, trong đó thông tin chuyển động thứ nhất bao gồm thông tin chuyển động dựa trên danh sách ảnh tham chiếu thứ nhất, thông tin chuyển động thứ hai bao gồm thông tin chuyển động dựa trên danh sách ảnh tham chiếu thứ hai, và trong đó thông tin chuyển động thứ ba bao gồm thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai.

Theo phương án thực hiện thứ sáu, thiết bị giải mã dòng video bao gồm bộ xử lý và bộ nhớ, trong đó bộ nhớ lưu trữ các lệnh khiến bộ xử lý thực hiện phương pháp theo triển khai bất kỳ của phương án thực hiện thứ ba.

Theo phương án thực hiện thứ bảy, thiết bị mã hóa dòng video bao gồm bộ xử lý và bộ nhớ, trong đó bộ nhớ lưu trữ các lệnh khiến bộ xử lý thực hiện phương pháp theo triển khai bất kỳ của phương án thực hiện thứ ba.

Theo phương án thực hiện thứ tám, vật ghi máy tính đọc được đã lưu trữ các lệnh trên đó mà khi được thực thi khiến một hoặc nhiều bộ xử lý được tạo cấu hình để mã hóa dữ liệu video, trong đó các lệnh khiến một hoặc nhiều bộ xử lý thực hiện phương pháp theo triển khai bất kỳ của phương án thực hiện thứ ba.

Theo phương án thực hiện thứ chín, chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo triển khai bất kỳ của phương án thực hiện thứ ba khi được thực thi trên máy tính.

Theo sáng chế, hân sau là giải thích phương pháp mã hóa cũng như phương pháp giải mã như được thể hiện trên các phương án thực hiện nêu trên, và hệ thống sử dụng chúng.

Fig.22 là sơ đồ khối thể hiện hệ thống cấp nội dung 3100 để thực hiện dịch vụ phân tán nội dung. Hệ thống cấp nội dung 3100 bao gồm thiết bị chụp 3102, thiết bị đầu cuối 3106, và một cách tùy chọn bao gồm thiết bị hiển thị 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 trên liên kết truyền thông 3104. Liên kết truyền thông có thể bao gồm kênh truyền thông 13 được mô tả trên đây. Liên kết truyền thông 3104 bao gồm nhưng không bị giới hạn ở WiFi, Ethernet, cáp, không dây (3G/4G/5G), USB, hoặc loại kết hợp bất kỳ của nó, hoặc tương tự.

Thiết bị chụp 3102 tạo dữ liệu, và có thể mã hóa dữ liệu bằng phương pháp mã hóa như được thể hiện trên các phương án thực hiện nêu trên. Theo cách khác, thiết bị chụp 3102 có thể phân tán dữ liệu đến máy chủ streaming (không được thể hiện trên hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa đến thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm mà không bị giới hạn ở camera, điện thoại thông minh hoặc Pad, máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, thiết bị lắp trên xe, hoặc kết hợp của bất kỳ thiết bị nào trong số đó, hoặc tương tự. Chẳng hạn, thiết bị chụp 3102 có thể bao gồm thiết bị nguồn 12 như được mô tả trên đây. Khi dữ liệu bao gồm video, bộ mã hóa video 20 được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện xử lý mã hóa video. Khi dữ liệu bao gồm audio (tức là, giọng nói), bộ mã hóa audio được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện quá trình mã hóa audio. Đối với một số kịch bản thực tế, thiết bị chụp 3102 phân tán dữ liệu video và audio được mã hóa bằng cách ghép kênh chúng với nhau. Đối với các kịch bản thực tế khác, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Thiết bị chụp 3102 phân tán riêng rẽ dữ liệu audio được mã hóa và dữ liệu video được mã hóa đến thiết bị đầu cuối 3106.

Trong hệ thống cấp nội dung 3100, thiết bị đầu cuối 310 nhận và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng nhận và khôi phục dữ liệu, chẳng hạn điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, đầu ghi video mạng (network video recorder, NVR)/đầu ghi video số (digital video recorder, DVR) 3112, TV 3114, hộp tín

hiệu truyền hình cáp (set top box, STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, thiết bị hỗ trợ số cá nhân (personal digital assistant, PDA) 3122, thiết bị lắp trên xe 3124, hoặc kết hợp bất kỳ trong số chúng, hoặc tương tự có thể giải mã dữ liệu được mã hóa nêu trên. Chẳng hạn, thiết bị đầu cuối 3106 có thể bao gồm thiết bị đích 14 như được mô tả trên đây. Khi dữ liệu được mã hóa bao gồm video, bộ giải mã video 30 được bao gồm trong thiết bị đầu cuối được ưu tiên thực hiện giải mã video. Khi dữ liệu được mã hóa bao gồm audio, bộ giải mã audio được bao gồm trong thiết bị đầu cuối được ưu tiên thực hiện quá trình giải mã audio.

Đối với thiết bị đầu cuối có bộ phận hiển thị, chẳng hạn, điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, NVR/DVR 3112, TV 3114, PDA 3122, hoặc thiết bị lắp trên xe 3124, thiết bị đầu cuối có thể cấp dữ liệu được giải mã cho bộ phận hiển thị của nó. Đối với thiết bị đầu cuối không có bộ phận hiển thị, chẳng hạn, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, bộ phận hiển thị bên ngoài 3126 được tiếp xúc trong đó để nhận và thể hiện dữ liệu được giải mã.

Khi mỗi thiết bị trong hệ thống này thực hiện mã hóa hoặc giải mã, thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh, như được thể hiện trên các phương án thực hiện nêu trên, có thể được sử dụng.

Fig.23 là sơ đồ thể hiện cấu trúc của ví dụ về thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 nhận stream từ thiết bị chụp 3102, khối xử lý giao thức 3202 phân tích giao thức truyền của stream. Giao thức bao gồm nhưng không bị giới hạn ở giao thức streaming thời gian thực (Real Time Streaming Protocol, RTSP), giao thức truyền siêu văn bản (Hyper Text Transfer Protocol, HTTP), giao thức streaming trực tiếp (HTTP Live streaming protocol, HLS), MPEG-DASH, giao thức truyền thời gian thực (Real-time Transport protocol, RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, RTMP), hoặc loại bất kỳ của kết hợp của nó, hoặc tương tự.

Sau khi khối xử lý giao thức 3202 xử lý stream, tệp tin stream được tạo. Tệp tin được xuất ra khỏi phân kênh 3204. Khối phân kênh 3204 có thể tách riêng dữ liệu được ghép kênh thành dữ liệu audio được mã hóa và dữ liệu video được mã

hóa. Như được mô tả nêu trên, đối với một số kịch bản thực tế, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Trong tình huống này, dữ liệu được mã hóa được truyền đến bộ giải mã video 3206 và bộ giải mã audio 3208 mà không qua khối phân kênh 3204.

Qua quá trình phân kênh, dòng cơ bản (elementary stream, ES) video, ES audio, và một cách tùy chọn phụ đề được tạo. Bộ giải mã video 3206, bao gồm bộ giải mã video 30 như được giải thích theo các phương án thực hiện nêu trên, giải mã ES video bằng phương pháp giải mã như được thể hiện trên các phương án thực hiện nêu trên để tạo khung video, và cấp dữ liệu này cho khối đồng bộ 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo khung audio, và nạp dữ liệu này cho khối đồng bộ 3212. Theo cách khác, khung video có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.23) trước khi cấp nó cho khối đồng bộ 3212. Tương tự, khung audio có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.23) trước khi cấp nó cho khối đồng bộ 3212.

Khối đồng bộ 3212 đồng bộ khung video và khung audio, và cấp video/audio cho bộ hiển thị video/audio 3214. Chẳng hạn, khối đồng bộ 3212 đồng bộ hiển thị thông tin video và audio. Thông tin có thể mã hóa trong cú pháp nhờ sử dụng các nhãn thời gian liên quan đến hiển thị audio và dữ liệu hình ảnh được mã hóa và các nhãn thời gian liên quan đến việc truyền chính dữ liệu stream.

Nếu phụ đề được bao gồm trong stream, bộ giải mã phụ đề 3210 giải mã phụ đề, và đồng bộ nó với khung video và khung audio, và cấp video/audio/phụ đề cho bộ phận hiển thị video/audio/phụ đề 3216.

Sáng chế không bị giới hạn ở hệ thống nêu trên, và thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh theo các phương án thực hiện nêu trên có thể được đưa vào hệ thống khác, chẳng hạn, hệ thống xe hơi.

Toán tử toán học

Các toán tử toán học được sử dụng trong bản mô tả giống như các toán tử được sử dụng trong ngôn ngữ lập trình C. Tuy nhiên, các kết quả của các phép phân chia số nguyên và dịch số học được định nghĩa chính xác hơn, và các phép

toán bổ sung được định nghĩa, chẳng hạn phép nâng lũy thừa và phép chia số thực. Các quy ước đánh số và đếm thường bắt đầu từ 0, chẳng hạn, “thứ nhất” tương đương với thứ 0, “thứ hai” tương đương với thứ 1, v.v..

Các toán tử số học

Các toán tử số học dưới đây được định nghĩa như sau:

+	Phép cộng
–	Phép trừ (như là phép toán hai đối số) hoặc âm (như là toán tử tiền tố đơn nguyên)
*	Phép nhân, bao gồm phép nhân ma trận
x^y	Lũy thừa. Xác định x nâng lên lũy thừa y . Trong các ngữ cảnh khác, ký pháp này được sử dụng cho chỉ số trên chứ không nhằm được hiểu như là lũy thừa.
/	Phép chia số nguyên có rút gọn kết quả về 0. Chẳng hạn, $7 / 4$ và $-7 / -4$ được rút gọn thành 1 và $-7 / 4$ và $7 / -4$ được rút gọn thành -1.
÷	Được sử dụng để ký hiệu phép chia trong các phương trình toán học trong đó không nhằm rút gọn hoặc làm tròn.
$\frac{x}{y}$	Được sử dụng để ký hiệu phép chia trong các phương trình toán học trong đó không nhằm rút gọn hoặc làm tròn.
$\sum_{i=x}^y f(i)$	Tổng của $f(i)$ với i có tất cả các giá trị nguyên từ x đến y và bao gồm y .
$x \% y$	Modulus. Phần dư của phép chia x cho y , được định nghĩa chỉ cho các số nguyên x và y với $x \geq 0$ và $y > 0$.

Toán tử logic

Các toán tử logic sau được định nghĩa như sau:

$x \&\& y$ Logic boolean “and” của x và y

$x || y$ Logic boolean “or” của x và y

! Logic boolean “not”

$x ? y : z$ nếu x là TRUE hoặc không bằng 0, nâng lên giá trị của y ; ngược lại, nâng lên giá trị của z .

Các toán tử quan hệ

Các toán tử quan hệ sau được định nghĩa như sau:

> Lớn hơn

$\geq$ Lớn hơn hoặc bằng

$<$ Nhỏ hơn

$\leq$ Nhỏ hơn hoặc bằng

$=$ Bằng

$\neq$ Không bằng

Khi toán tử quan hệ được áp dụng cho phần tử cú pháp hoặc biến đã được gán giá trị “na” (không áp dụng), giá trị “na” được xử lý như là giá trị riêng cho phần tử cú pháp hoặc biến. Giá trị “na” được xem xét không bằng bất kỳ giá trị khác.

Các toán tử theo bit

Các toán tử theo bit sau được định nghĩa như sau:

$\&$ “and” theo bit. Khi hoạt động trên các đối số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit có trọng số bằng 0.

$|$ “or” theo bit. Khi hoạt động trên các đối số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit có trọng số bằng 0.

$\wedge$ “exclusive or” theo bit. Khi hoạt động trên các đối số nguyên, hoạt động trên biểu diễn bù 2 của giá trị nguyên. Khi hoạt động trên đối số nhị phân chứa ít bit hơn đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung nhiều bit có trọng số bằng 0.

$x \gg y$ Dịch phải số học của biểu diễn số nguyên bù 2 của các số nhị phân x sang y . Hàm này được định nghĩa chỉ cho các giá trị số nguyên không âm của y . Các bit được dịch thành các bit có trọng số lớn nhất (most significant bit, MSB) làm kết quả dịch phải có giá trị bằng MSB của x trước phép toán dịch.

$x \ll y$ Dịch trái số học của biểu diễn số nguyên bù 2 của các số nhị phân x sang y . Hàm này được định nghĩa chỉ cho các giá trị số nguyên không âm của y . Các bit được dịch thành các bit có trọng số nhỏ nhất (least significant bit, LSB) làm kết quả dịch trái có giá trị bằng 0.

Các toán tử gán

Các toán tử số học dưới đây được định nghĩa như sau:

= Toán tử gán

++ Tăng, tức là, $x++$ tương đương $x = x + 1$; khi được sử dụng trong chỉ số mảng, nâng lên giá trị của biến trước phép tăng.

-- Giảm, tức là, $x--$ tương đương $x = x - 1$; khi được sử dụng trong chỉ số mảng, nâng lên giá trị của biến trước khi phép giảm.

+= Tăng bởi lượng được xác định, tức là, $x += 3$ tương đương $x = x + 3$, và

$x += (-3)$ tương đương $x = x + (-3)$.

-- Giảm bởi lượng được xác định, tức là, $x -= 3$ tương đương $x = x - 3$, và

$x -= (-3)$ tương đương $x = x - (-3)$.

Ký pháp khoảng

Ký pháp sau được sử dụng để xác định khoảng giá trị:

$x = y..z$ chiếm các giá trị số nguyên bắt đầu từ y đến z , bao gồm cả y và z , với x , y , và z là các số nguyên và z lớn hơn y .

Các hàm số học

Các hàm số học sau được định nghĩa:

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

$\text{Asin}(x)$ là hàm sin đảo lượng giác, hoạt động trên đối số x nằm trong khoảng từ $-1,0$ đến $1,0$, bao gồm hai giá trị đầu mút, có giá trị đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, bao gồm hai giá trị đầu mút, theo các đơn vị radian

$\text{Atan}(x)$ là hàm tang ngược lượng giác, hoạt động trên đối số x , có giá trị đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, bao gồm hai giá trị đầu mút, theo các đơn vị radian

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x == 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{ngược lại} \end{cases}$$

$\text{Ceil}(x)$ số nguyên nhỏ nhất lớn hơn hoặc bằng x .

$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x)$

$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{ngược lại} \end{cases}$$

$\text{Cos}(x)$ hàm cosin lượng giác hoạt động trên đối số x theo các đơn vị radian.

$\text{Floor}(x)$ số nguyên lớn nhất nhỏ hơn hoặc bằng x .

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d/2 \\ c - d & ; \quad a - b > d/2 \\ c & ; \quad \text{ngược lại} \end{cases}$$

$\text{Ln}(x)$ lôgarit tự nhiên của x (lôgarit cơ số e , trong đó e là hằng số cơ số lôgarit tự nhiên 2,718 281 828...).

$\text{Log2}(x)$ lôgarit cơ số 2 của x .

$\text{Log10}(x)$ lôgarit cơ số 10 của x .

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0,5)$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases}$$

$\text{Sin}(x)$ hàm sin lượng giác hoạt động trên đối số x theo các đơn vị radian

$\text{Sqrt}(x) = \sqrt{x}$

$$\text{Swap}(x, y) = (y, x)$$

$\text{Tan}(x)$ hàm tang lượng giác hoạt động trên đối số x theo các đơn vị radian

Thứ tự ưu tiên phép toán

Khi thứ tự ưu tiên trong biểu thức hông được chỉ báo tường minh nhờ sử dụng các dấu ngoặc, các quy tắc sau áp dụng:

- Các phép toán có độ ưu tiên cao hơn được đánh giá trước bất kỳ phép toán có độ ưu tiên thấp hơn.
- Các phép toán có cùng độ ưu tiên được đánh giá tuần tự từ trái sang phải.

Bảng dưới đây xác định độ ưu tiên của các phép toán từ cao nhất đến thấp nhất; vị trí cao hơn trong bảng chỉ báo độ ưu tiên cao hơn.

Đối với các toán tử cũng được sử dụng trong ngôn ngữ lập trình C, thứ tự ưu tiên được sử dụng trong bản mô tả giống như được sử dụng trong ngôn ngữ lập trình C.

Bảng: Ưu tiên phép toán từ cao nhất (trên cùng của bảng) đến thấp nhất (dưới cùng của bảng)

Các phép toán (với các toán hạng x , y , và z)
" $x++$ ", " $x--$ "
" $!x$ ", " $-x$ " (làm toán tử tiền tố đơn nguyên)
x^y
" $x * y$ ", " x / y ", " $x \div y$ ", " $\frac{x}{y}$ ", " $x \% y$ "
" $x + y$ ", " $x - y$ " (làm toán tử hai đối số), " $\sum_{i=x}^y f(i)$ "
" $x << y$ ", " $x >> y$ "
" $x < y$ ", " $x \leq y$ ", " $x > y$ ", " $x \geq y$ "
" $x == y$ ", " $x != y$ "
" $x \& y$ "
" $x y$ "
" $x \&\& y$ "
" $x y$ "
" $x ? y : z$ "
" $x..y$ "
" $x = y$ ", " $x += y$ ", " $x -= y$ "

Mô tả bằng văn bản các phép toán logic

Theo văn bản, mệnh đề của các phép toán logic như sẽ được mô tả bằng toán

học ở dạng dưới đây:

```

    if(condition 0)
        statement 0
    else if(condition 1)
        statement 1
    ...
    else /* nhận xét mang thông tin ở điều kiện còn lại*/
        statement n

```

có thể được mô tả theo cách dưới đây:

...như sau / ... điều kiện sau áp dụng:

- If condition 0, statement 0
- Ngược lại, if condition 1, statement 1
- ...
- Ngược lại (nhận xét mang thông tin ở điều kiện còn lại), statement n

Mỗi mệnh đề “Nếu ... Ngược lại, nếu ... Ngược lại, ...” bằng văn bản được nêu với “...như sau” hoặc “... điều kiện sau áp dụng” tiếp sau ngay bằng “Nếu ...”. Điều kiện cuối cùng của “Nếu ... Ngược lại, nếu ... Ngược lại, ...” luôn là “Ngược lại, ...”. Các mệnh đề đan xen “nếu ... Ngược lại, nếu... Ngược lại, ...” có thể được nhận diện bằng cách so khớp “... như sau” hoặc “... điều kiện sau áp dụng” với kết thúc “Ngược lại, ...”.

Trong văn bản, mệnh đề của các phép lô gic như sẽ được mô tả về mặt toán học ở dạng dưới đây:

```

    if(condition 0a && condition 0b)
        statement 0
    else if(condition 1a || condition 1b)
        statement 1
    ...
    else
        statement n

```

có thể được mô tả theo cách dưới đây:

... như sau/ ... điều kiện sau áp dụng:

- Nếu tất cả các điều kiện sau là đúng, statement 0:
 - condition 0a
 - condition 0b
- Ngược lại, nếu một hoặc nhiều điều kiện sau là đúng, statement 1:
 - condition 1a
 - condition 1b
- ...
- Ngược lại, statement n

Theo văn bản, mệnh đề của các phép toán lô gic như sẽ được mô tả về mặt toán học ở dạng dưới đây:

if(condition 0)

statement 0

if(condition 1)

statement 1

có thể được mô tả theo cách dưới đây:

When condition 0, statement 0

When condition 1, statement 1.

Theo các phương án thực hiện, chẳng hạn, bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, chẳng hạn, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được triển khai trong phần cứng, phần mềm, firmware, hoặc kết hợp bất kỳ của nó. Nếu được triển khai trong phần mềm, các chức năng có thể được lưu trữ trên phương tiện máy tính đọc được hoặc được truyền trên các phương tiện truyền thông dưới dạng một hoặc nhiều lệnh hoặc mã và được thực thi bằng khối xử lý dựa trên phần cứng. Phương tiện máy tính đọc được có thể bao gồm các phương tiện lưu trữ máy tính đọc được, tương ứng với phương tiện hữu hình, chẳng hạn phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm phương tiện bất kỳ để truyền chương trình máy tính từ một điểm sang điểm khác, chẳng hạn, theo giao thức truyền thông. Theo cách này, phương tiện

máy tính đọc được thường có thể tương ứng với (1) các phương tiện lưu trữ máy tính đọc được hữu hình vốn là bất biến hoặc (2) phương tiện truyền thông, chẳng hạn tín hiệu hoặc sóng kênh mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ có thể được truy nhập bằng một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất lệnh, mã và/hoặc các cấu trúc dữ liệu để thực hiện các kỹ thuật được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện máy tính đọc được.

Thông qua ví dụ, và không giới hạn, các phương tiện lưu trữ máy tính đọc được có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc lưu trữ đĩa quang khác, lưu trữ đĩa từ, hoặc các thiết bị lưu trữ từ tính khác, bộ nhớ nhanh, hoặc phương tiện khác bất kỳ có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng lệnh hoặc cấu trúc dữ liệu và có thể được truy nhập bằng máy tính. Bất kỳ kết nối cũng được gọi đúng là phương tiện máy tính đọc được. Chẳng hạn, nếu các lệnh được truyền từ website, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn, đường thuê bao số (digital subscriber line, DSL), hoặc các công nghệ không dây, chẳng hạn, hồng ngoại, vô tuyến, và vi sóng, sau đó, cáp đồng trục, cáp sợi quang, cáp xoắn, DSL, hoặc các công nghệ không dây, chẳng hạn, hồng ngoại, vô tuyến, và vi sóng được bao gồm trong định nghĩa về phương tiện. Tuy nhiên, cần hiểu rằng các phương tiện lưu trữ máy tính đọc được và phương tiện lưu trữ dữ liệu không bao gồm các kết nối, sóng kênh mang, tín hiệu, hoặc các phương tiện khả biến khác, nhưng thay vào đó dành cho các phương tiện lưu trữ hữu hình, bất biến. Disk và disc, như được sử dụng ở đây, bao gồm đĩa nén (compact disc, CD), đĩa laze, đĩa quang, đĩa đa dụng số (digital versatile disc, DVD), đĩa mềm và đĩa Blu-ray, trong đó disk thường tái tạo dữ liệu dạng từ tính, trong khi disc tái tạo dữ liệu dạng quang học bằng tia laze. Các tổ hợp nêu trên cũng cần được bao gồm trong phạm vi phương tiện máy tính đọc được.

Các lệnh có thể được thực thi bằng một hoặc nhiều bộ xử lý, chẳng hạn một hoặc nhiều bộ xử lý tín hiệu số (digital signal processor, DSP), bộ vi xử lý đa năng, ASIC, FPGA, hoặc mạch tích hợp tương đương khác hoặc logic rời rạc. Do đó, cụm từ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến trường hợp

bất kỳ của cấu trúc nêu trên hoặc cấu trúc khác bất kỳ thích hợp để thực hiện các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, chức năng được mô tả ở đây có thể được đặt trong phần cứng và/hoặc các môđun phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được đưa vào codec kết hợp. Các kỹ thuật cũng có thể được triển khai đầy đủ trong một hoặc nhiều mạch hoặc phần tử logic.

Các kỹ thuật của sáng chế có thể được triển khai trong nhiều bộ phận hoặc thiết bị, bao gồm điện thoại cầm tay không dây, mạch tích hợp (integrated circuit, IC) hoặc tập hợp IC (chẳng hạn, chip set). Các thành phần, môđun, hoặc đơn vị khác nhau được mô tả theo sáng chế để nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không nhất thiết yêu cầu thực hiện bằng các đơn vị phần cứng khác nhau. Thay vào đó, như được mô tả trên đây, các đơn vị khác nhau có thể được kết hợp trong đơn vị phần cứng codec hoặc được bố trí bằng tập hợp các đơn vị phần cứng liên tác, bao gồm một hoặc nhiều bộ xử lý như được mô tả trên đây, cùng với phần mềm và/hoặc firmware thích hợp.

YÊU CẦU BẢO HỘ

1. Phương pháp lưu trữ thông tin chuyển động, trong đó khối hiện tại gồm khối phụ thứ nhất và khối phụ thứ hai, khối phụ thứ nhất và khối phụ thứ hai được tách riêng bởi đường biên chung, trong đó đường biên chung là đường tách của chế độ phân vùng hình học, khối phụ thứ nhất tương ứng với thông tin chuyển động thứ nhất và khối phụ thứ hai tương ứng với thông tin chuyển động thứ hai, trong đó thông tin chuyển động thứ nhất bao gồm vector chuyển động đơn dự báo cho khối phụ thứ nhất, thông tin chuyển động thứ hai bao gồm vector chuyển động đơn dự báo cho khối phụ thứ hai, phương pháp bao gồm các bước:

xác định (S203) khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung; trong đó khoảng cách được tính toán bằng công thức sau:

$$distFromLine = ((x \ll N) + K) * P1 + ((y \ll N) + K) * P2 - P3$$

trong đó *distFromLine* là khoảng cách, *x* là tọa độ ngang của vị trí đích trong tập hợp mẫu, *y* là tọa độ dọc của vị trí đích, *K* là số nguyên không âm, *N*=1, và *P1*, *P2* và *P3* lần lượt là tham số tính toán thứ nhất, tham số tính toán thứ hai và tham số tính toán thứ ba, trong đó tham số tính toán thứ nhất được xác định bằng cách kiểm tra bảng tra cứu định trước theo tham số góc, tham số tính toán thứ hai được xác định bằng cách kiểm tra bảng tra cứu theo tham số góc tạm thời, và tham số tính toán thứ ba được xác định dựa trên tham số góc và tham số khoảng cách; trong đó tham số góc được biểu diễn dưới dạng *angleIdx*, và tham số góc tạm thời thu được bởi $(angleIdx + 8) \% 32$, trong đó tham số khoảng cách được biểu diễn dưới dạng *distanceIdx*; *angleIdx* xác định chỉ số góc của chế độ phân vùng hình học, và *khoảng cáchIdx* xác định chỉ số khoảng cách của chế độ phân vùng hình học; và

so sánh (S204) khoảng cách có ngưỡng để xác định liệu có lưu trữ thông tin chuyển động thứ ba đối với tập hợp mẫu, trong đó thông tin chuyển động thứ ba được dẫn xuất bằng cách kết hợp thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai khi thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai là từ các danh sách ảnh tham chiếu khác nhau; và thông tin chuyển động thứ ba bằng thông tin chuyển động thứ nhất hoặc thông tin chuyển động thứ hai khi thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai là

từ cùng danh sách ảnh tham chiếu;

lưu trữ thông tin chuyển động thứ ba đối với tập hợp mẫu trong trường hợp mà giá trị tuyệt đối của khoảng cách nhỏ hơn ngưỡng; trong đó thông tin chuyển động được lưu trữ được sử dụng để xác định giá trị dự báo của tập hợp mẫu, hoặc thông tin chuyển động được lưu trữ được sử dụng làm thông tin tham chiếu cho quá trình dự báo của các khối tạo mã trong tương lai.

2. Phương pháp theo điểm 1, trong đó trong trường hợp mà giá trị tuyệt đối của khoảng cách lớn hơn hoặc bằng ngưỡng, phương pháp còn bao gồm các bước:

xác định thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là dương; hoặc,

xác định thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là âm.

3. Phương pháp theo điểm 1 hoặc 2, trong đó ngưỡng được định trước.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó đường biên chung được xác định bằng tham số góc và tham số khoảng cách, trước khi xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung, phương pháp còn bao gồm bước:

xác định (S201) tham số góc và tham số khoảng cách.

5. Phương pháp theo điểm 4, trong đó khi phương pháp được triển khai trong bộ giải mã, xác định tham số góc và tham số khoảng cách, bao gồm các bước:

phân tách tham số góc và tham số khoảng cách từ dòng bit; hoặc,

phân tách bộ chỉ báo từ dòng bit, và

thu được tham số góc và tham số khoảng cách dựa trên bộ chỉ báo.

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, trong đó khối phụ thứ nhất và khối phụ thứ hai là phân vùng tam giác, phân vùng hình thang, hoặc phân vùng hình chữ nhật bất đối xứng.

7. Thiết bị lưu trữ thông tin chuyển động, trong đó khối hiện tại gồm khối phụ thứ nhất và khối phụ thứ hai, khối phụ thứ nhất và khối phụ thứ hai được tách riêng bởi đường biên chung, trong đó đường biên chung là đường tách của chế độ phân vùng hình học, khối phụ thứ nhất tương ứng với thông tin chuyển động thứ nhất và khối phụ thứ hai tương ứng với thông tin chuyển động thứ hai, trong

đó thông tin chuyển động thứ nhất bao gồm vectơ chuyển động đơn dự báo cho khối phụ thứ nhất, thông tin chuyển động thứ hai bao gồm vectơ chuyển động đơn dự báo cho khối phụ thứ hai, trong đó thiết bị bao gồm:

môđun tính toán (3001), được tạo cấu hình để xác định khoảng cách giữa tập hợp mẫu trong khối hiện tại và đường biên chung; trong đó khoảng cách được tính toán bằng công thức sau:

$$\text{distFromLine} = ((x \ll N) + K) * P1 + ((y \ll N) + K) * P2 - P3$$

trong đó *distFromLine* là khoảng cách, *x* là tọa độ ngang của vị trí đích trong tập hợp mẫu, *y* là tọa độ dọc của vị trí đích, *K* là số nguyên không âm, *N=1*, *P1*, *P2* và *P3* lần lượt là tham số tính toán thứ nhất, tham số tính toán thứ hai và tham số tính toán thứ ba, trong đó tham số tính toán thứ nhất được xác định bằng cách kiểm tra bảng tra cứu định trước theo tham số góc và tham số tính toán thứ hai được xác định bằng cách kiểm tra bảng tra cứu theo tham số góc tạm thời, tham số tính toán thứ ba được xác định dựa trên tham số góc và tham số khoảng cách; trong đó tham số góc là *angleIdx*, tham số góc tạm thời là $(\text{angleIdx} + 8) \% 32$, tham số khoảng cách là *distanceIdx*; *angleIdx* xác định chỉ số góc của chế độ phân vùng hình học, và *distanceIdx* xác định chỉ số khoảng cách của chế độ phân vùng hình học; và

môđun so sánh (3002), được tạo cấu hình để so sánh khoảng cách có ngưỡng để xác định liệu lưu trữ thông tin chuyển động thứ ba đối với tập hợp mẫu, trong đó thông tin chuyển động thứ ba được dẫn xuất bằng cách kết hợp thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai khi thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai là từ các danh sách ảnh tham chiếu khác nhau; và thông tin chuyển động thứ ba bằng thông tin chuyển động thứ nhất or thông tin chuyển động thứ hai khi thông tin chuyển động thứ nhất và thông tin chuyển động thứ hai là từ cùng danh sách ảnh tham chiếu;

môđun lưu trữ (44, 460, 504), được tạo cấu hình để lưu trữ thông tin chuyển động thứ ba đối với tập hợp mẫu trong trường hợp mà giá trị tuyệt đối của khoảng cách nhỏ hơn ngưỡng; trong đó thông tin chuyển động được lưu trữ được sử dụng để xác định giá trị dự báo của tập hợp mẫu, hoặc thông tin chuyển động được lưu trữ được sử dụng làm thông tin tham chiếu cho quá trình dự báo của các khối

tạo mã trong tương lai.

8. Thiết bị theo điểm 7, trong đó trong trường hợp mà giá trị tuyệt đối của khoảng cách lớn hơn hoặc bằng ngưỡng, môđun so sánh (3002) còn được tạo cấu hình để:

xác định thông tin chuyển động thứ nhất được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là dương; hoặc,

xác định thông tin chuyển động thứ hai được lưu trữ đối với tập hợp mẫu trong trường hợp mà khoảng cách là âm.

9. Thiết bị theo điểm 7 hoặc 8, trong đó ngưỡng được định trước.

10. Thiết bị theo điểm bất kỳ trong số các điểm từ 7 đến 9, trong đó đường biên chung được xác định bởi tham số góc và tham số khoảng cách, môđun tính toán (3001) còn được tạo cấu hình để:

xác định tham số góc và tham số khoảng cách.

11. Thiết bị theo điểm 10, trong đó khi thiết bị là bộ giải mã, môđun tính toán (3001) còn được tạo cấu hình để:

phân tách tham số góc và tham số khoảng cách từ dòng bit; hoặc,

phân tách bộ chỉ báo từ dòng bit, và

thu được tham số góc và tham số khoảng cách dựa trên bộ chỉ báo.

12. Thiết bị theo điểm bất kỳ trong số các điểm từ 7 đến 11, trong đó khối phụ thứ nhất và khối phụ thứ hai là phân vùng tam giác, phân vùng hình thang, hoặc phân vùng hình chữ nhật bất đối xứng.

13. Vật ghi máy tính bao gồm mã chương trình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 6 khi được thực thi trên máy tính.

1/13

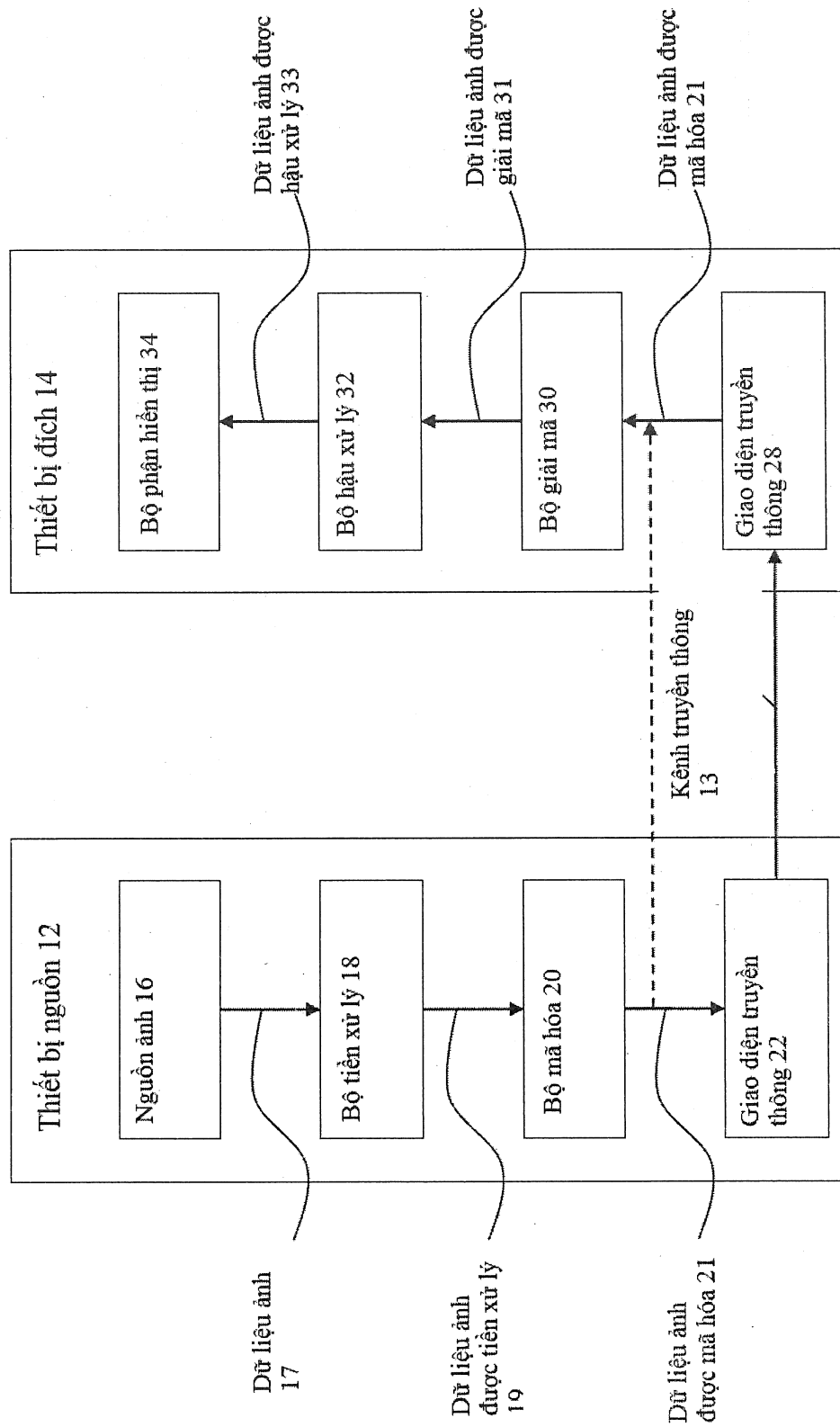


Fig.1A

2/13

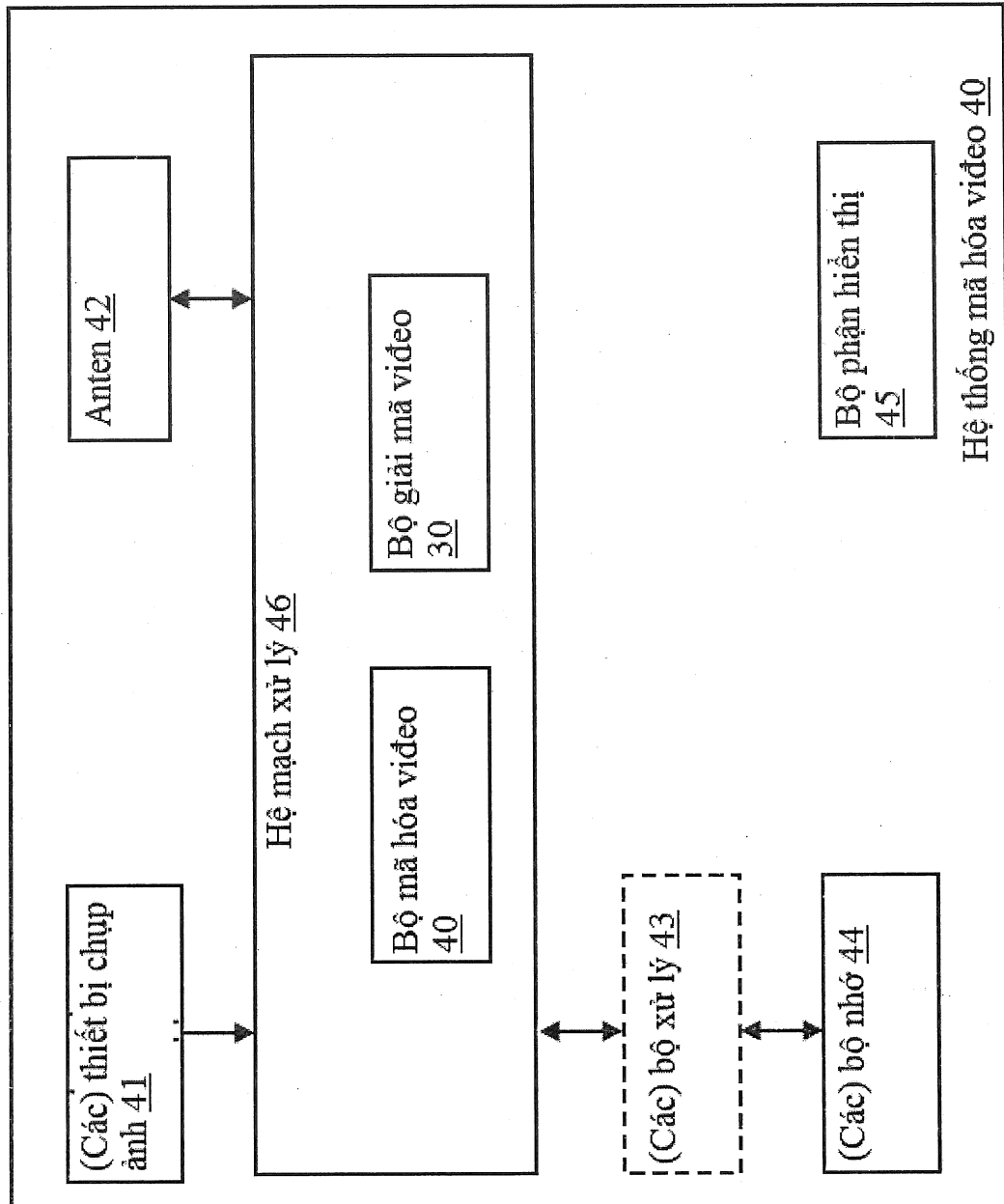


Fig.1B

4/13

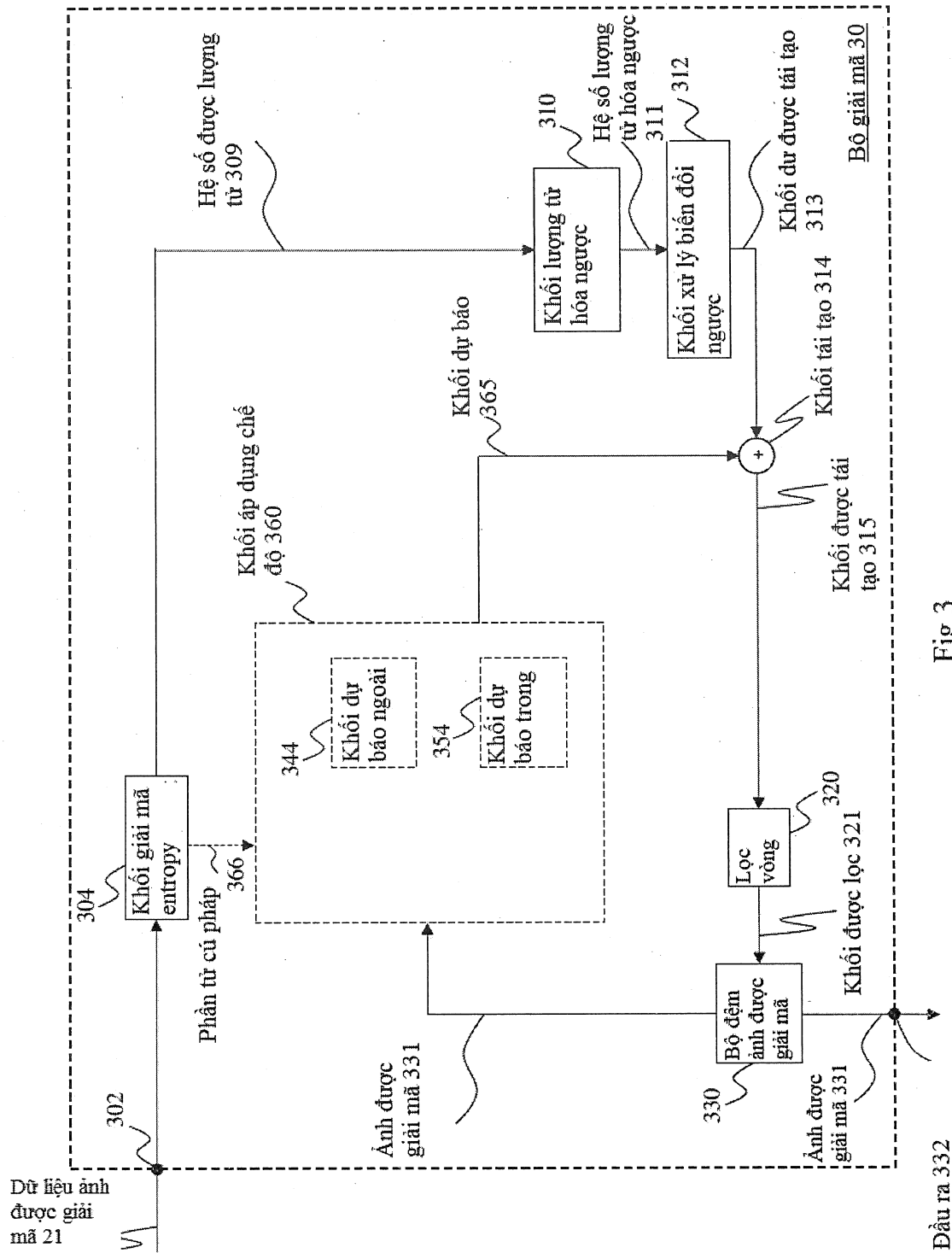


Fig.3

5/13

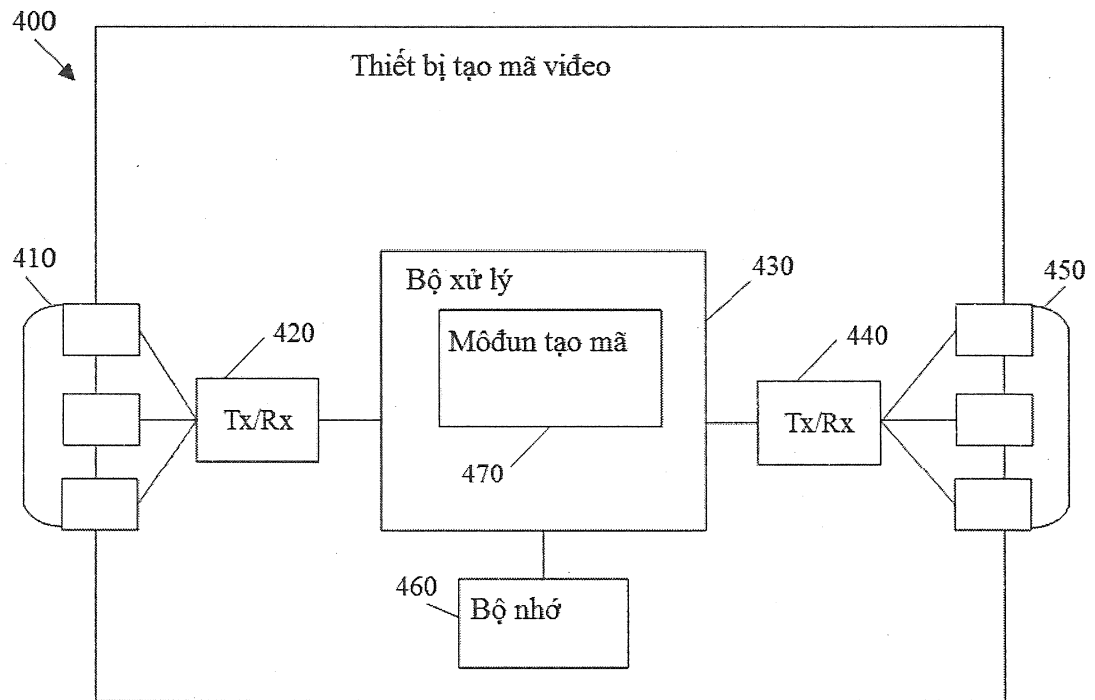


Fig.4

6/13

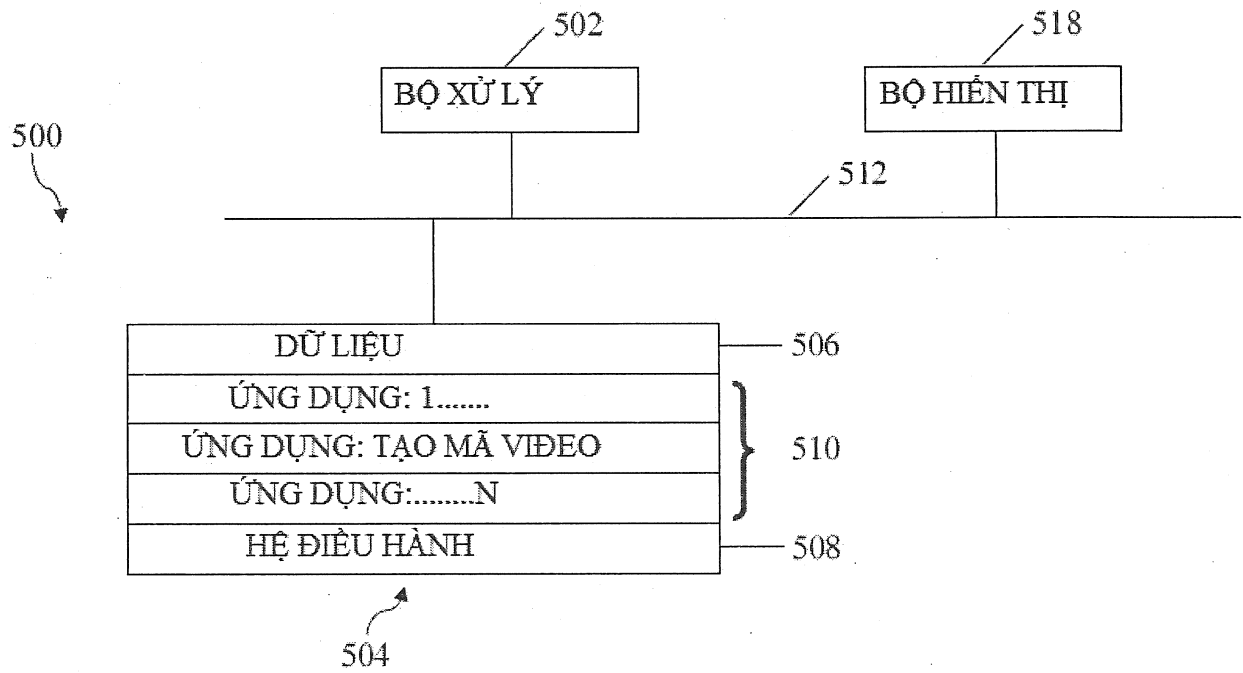


Fig.5

7/13

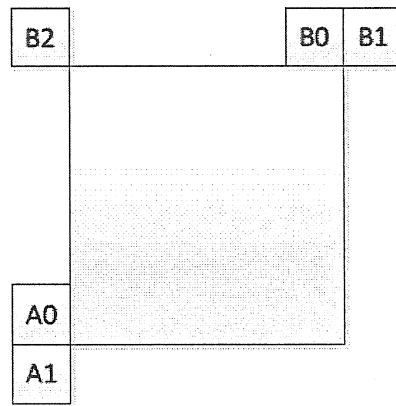


Fig.6

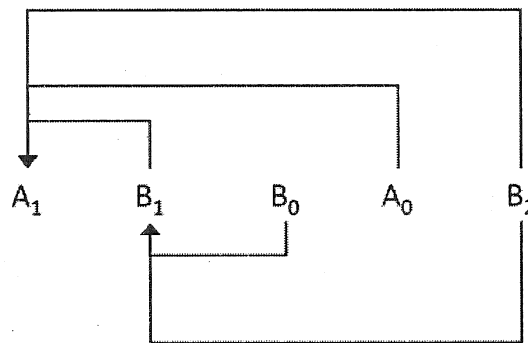


Fig.7

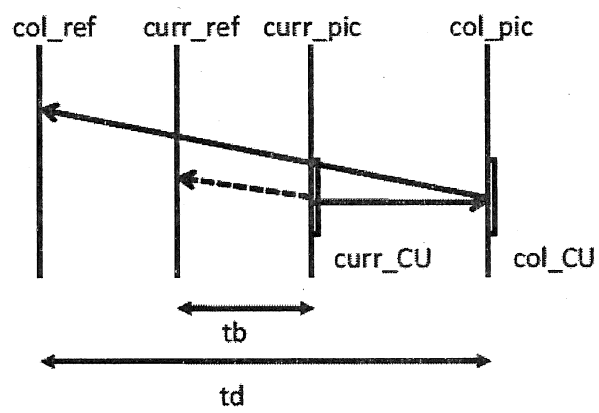


Fig.8

8/13

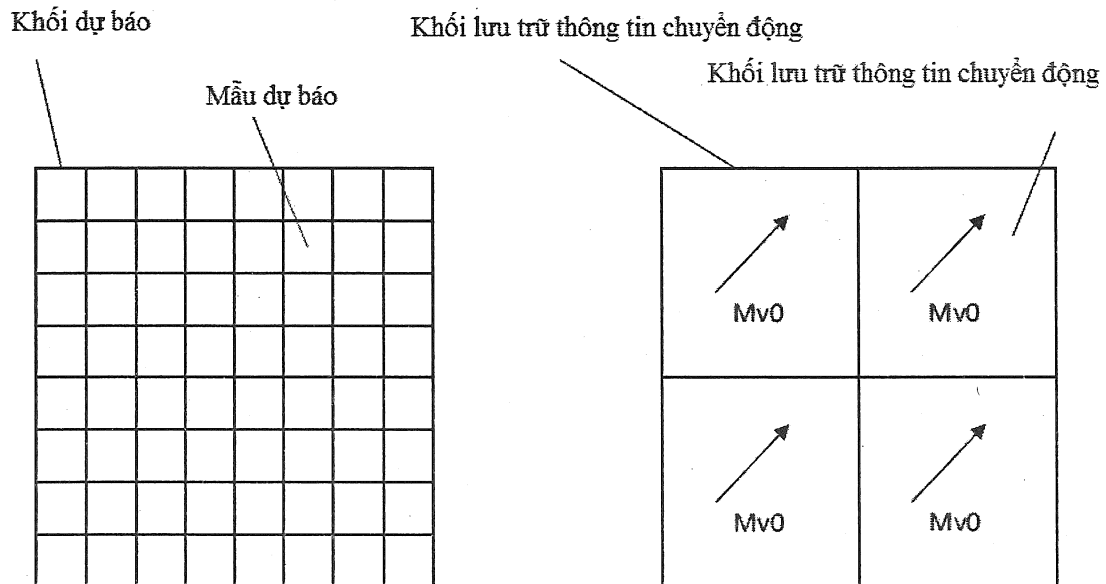


Fig.9

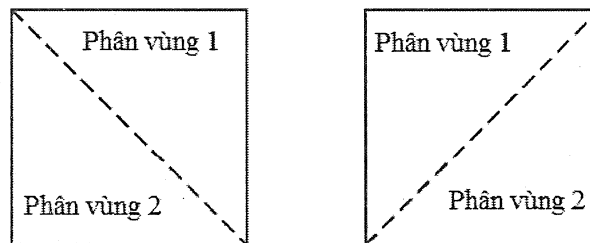


Fig.10

Chỉ số hợp nhất	L0 MV	L1 MV
0	x	
1		x
2	x	
3		x
4	x	

Fig.11

9/13

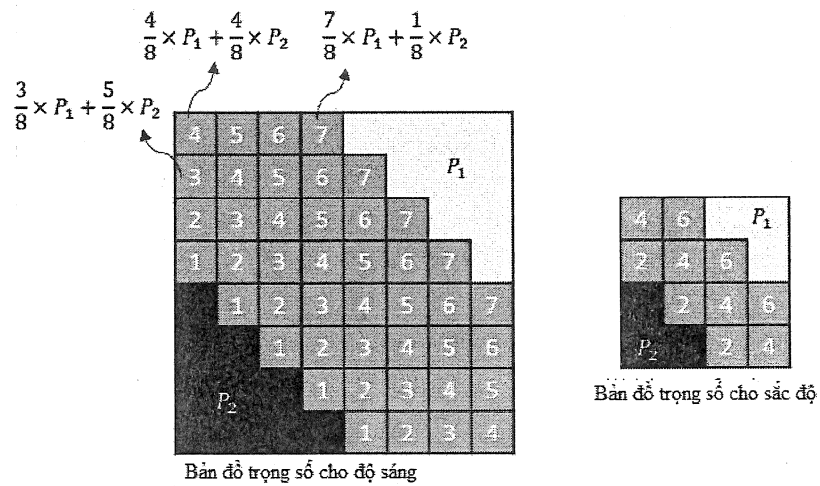


Fig.12

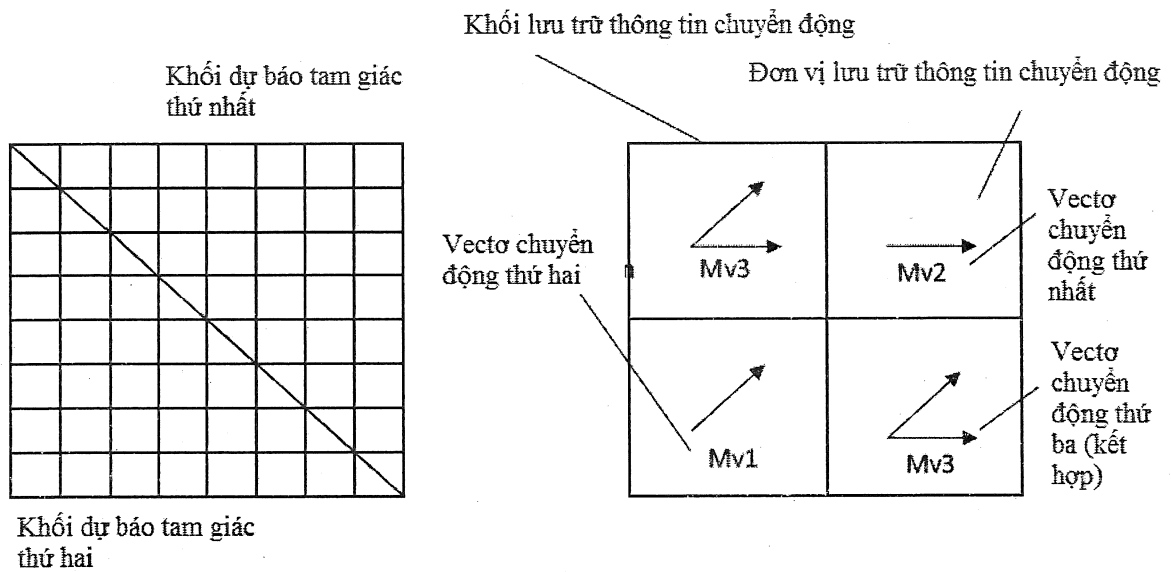


Fig.13

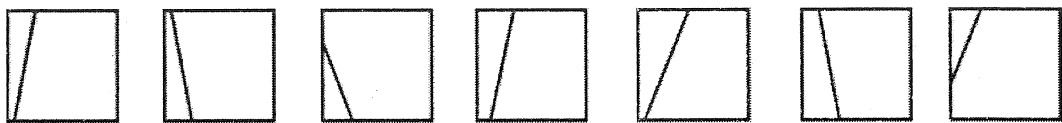


Fig.14

10/13

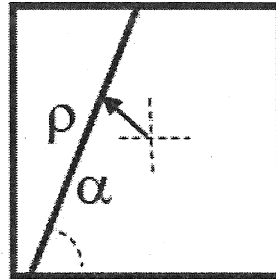


Fig.15

Mẫu ở tọa độ x=7, y=0

4	8	8	8	8	8	8	8
0	4	8	8	8	8	8	8
0	0	4	8	8	8	8	8
0	0	0	4	8	8	8	8
0	0	0	0	4	8	8	8
0	0	0	0	0	4	8	8
0	0	0	0	0	0	4	8
0	0	0	0	0	0	0	4

Giá trị sampleWeight tương ứng với vị trí mẫu

Đường phân tách

Fig.16

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
GeoFilter[idx]	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8

Fig.17

11/13

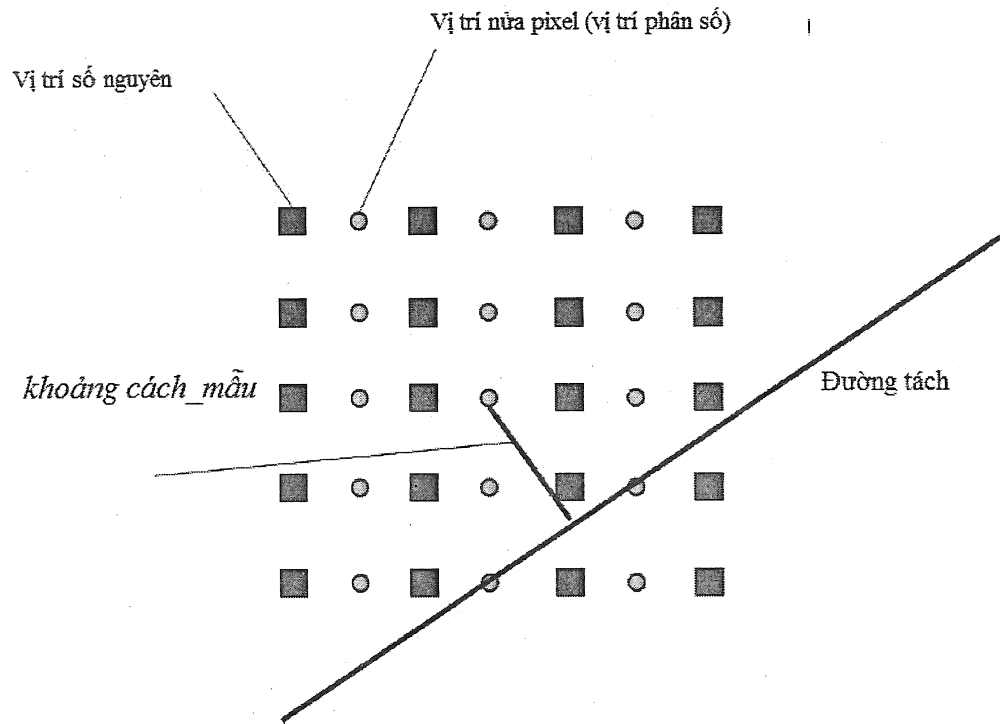


Fig.18

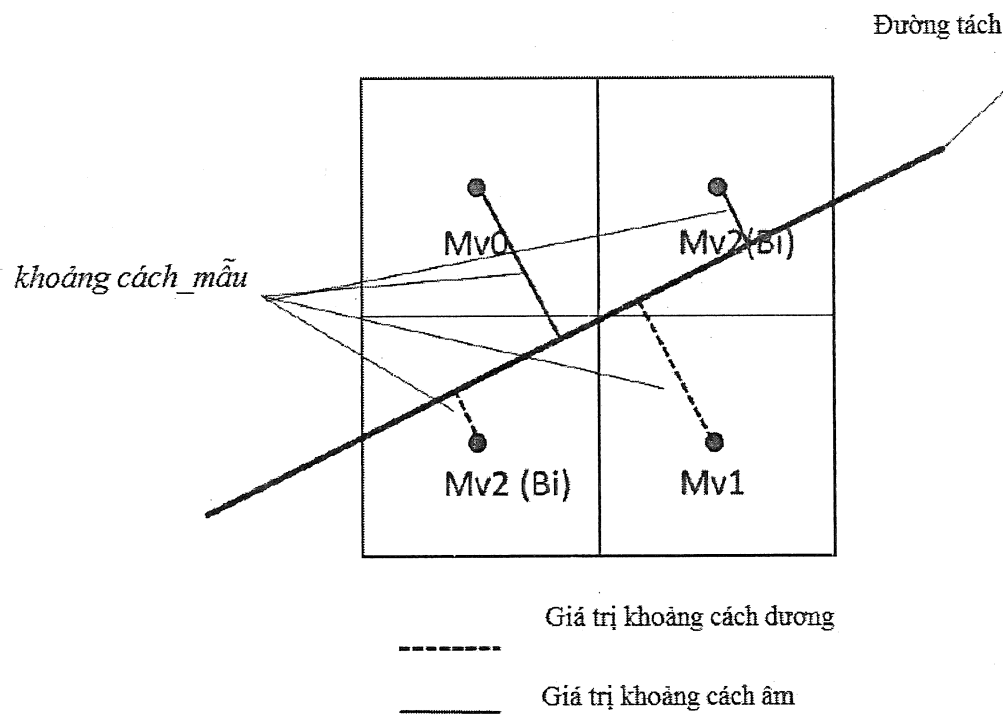


Fig.19

12/13

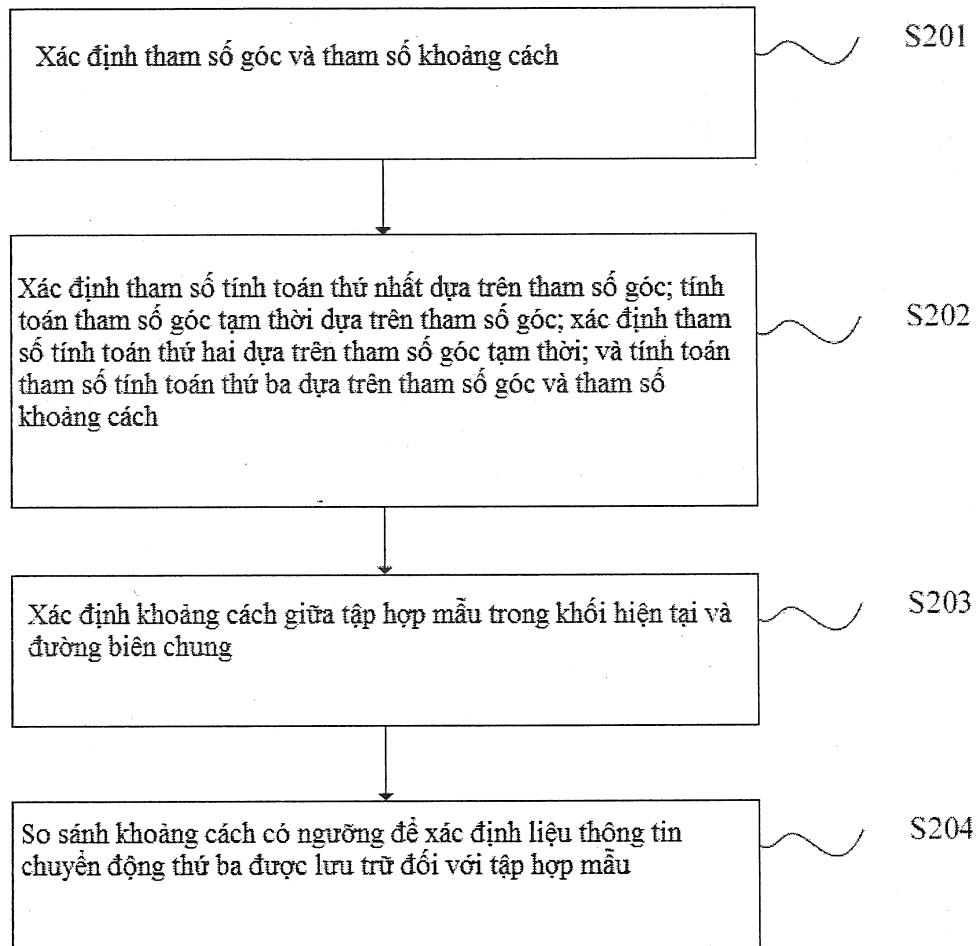


Fig.20

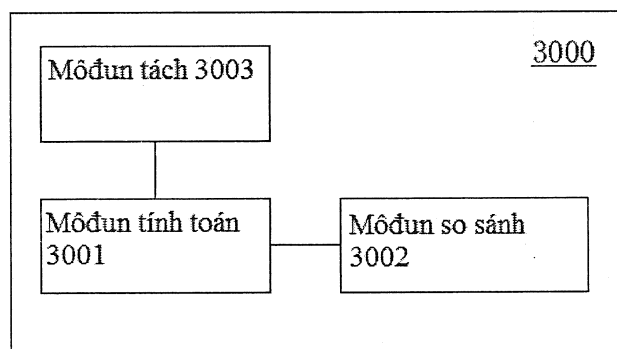


Fig.21

13/13

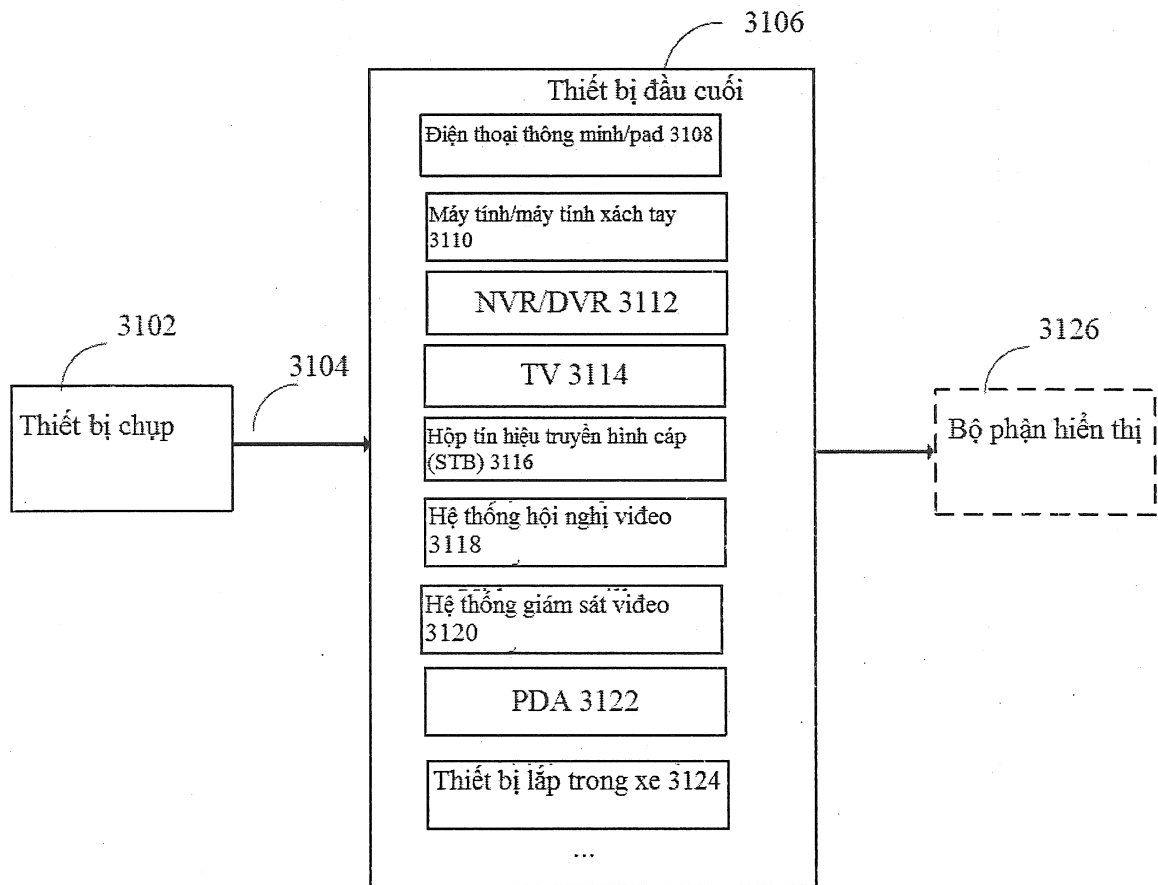


Fig.22

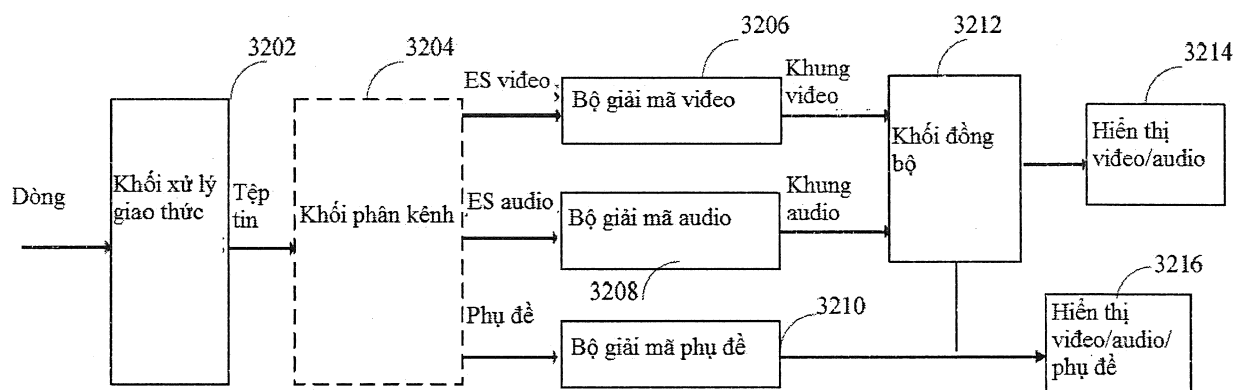


Fig.23