



(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)



1-0053269

(51)^{2020.01} H04N 19/91; H04N 19/176

(13) B

(21) 1-2022-02350

(22) 13/10/2020

(86) PCT/CN2020/120554 13/10/2020

(87) WO 2021/073488 22/04/2021

(30) 62/914,544 13/10/2019 US

(45) 25/11/2025 452

(43) 25/10/2022 415A

(73) 1. BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (CN)

Room B-0035, 2/F, No.3 Building No.30, Shixing Road, Shijingshan District Beijing
100041, China

2. BYTEDANCE INC. (US)

12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California
90066, USA

(72) ZHANG, Kai (CN); ZHANG, Li (CN); LIU, Hongbin (CN).

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

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ VIDEO, VÀ VẬT GHI MÁY TÍNH

(21) 1-2022-02350

(57) Sáng chế đề cập đến phương pháp lấy làm ví dụ xử lý video bao gồm bước xác định, đối với biến đổi giữa ảnh hiện tại của video và biểu diễn dòng bit của video, sử dụng công cụ tạo mã dựa trên điều kiện được liên kết với kích thước của một phần của một trong nhiều ảnh tham chiếu hoặc ảnh hiện tại. Phương pháp cũng bao gồm bước thực hiện biến đổi dựa trên việc xác định.

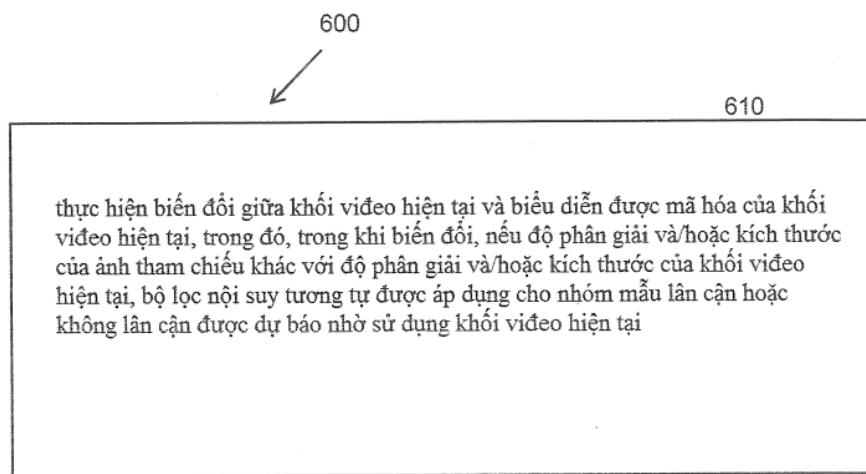


Fig.6

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến kỹ thuật, thiết bị và hệ thống tạo mã video.

Tình trạng kỹ thuật của sáng chế

Hiện tại, đang nỗ lực để cải thiện hiệu năng của các công nghệ codec video hiện tại để cho tỷ lệ nén tốt hơn hoặc tạo các sơ đồ mã hóa và giải mã video mà cho phép độ phức tạp thấp hơn hoặc các cách triển khai song song. Người có hiểu biết trung bình trong lĩnh vực gần đây đã đề xuất vài công cụ và kiểm tra tạo mã video mới hiện tại trong quá trình xác định hiệu quả của chúng.

Bản chất kỹ thuật của sáng chế

Thiết bị, hệ thống và phương pháp đề cập đến mã hóa video số, và cụ thể là, đến việc quản lý các vector chuyển động được nêu. Các phương pháp được mô tả có thể được áp dụng cho các chuẩn tạo mã video hiện có (chẳng hạn, Tạo mã video hiệu suất cao (HEVC) hoặc tạo mã video đa dụng) và các chuẩn tạo mã video hoặc các codec video tương lai.

Theo một khía cạnh đại diện, sáng chế có thể được sử dụng để đề cập phương pháp xử lý video. Phương pháp bao gồm thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, bộ lọc nội suy tương tự được áp dụng cho nhóm mẫu lân cận hoặc không lân cận được dự báo nhờ sử dụng khối video hiện tại.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, trong đó các khối được dự báo

nhờ sử dụng khối video hiện tại chỉ được phép sử dụng thông tin chuyển động có giá trị số nguyên liên quan đến khối hiện tại.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, bộ lọc nội suy được áp dụng để dẫn xuất các khối được dự báo nhờ sử dụng khối video hiện tại, và trong đó bộ lọc nội suy được chọn dựa trên quy tắc.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, áp dụng có chọn lọc bộ lọc tách khối, trong đó độ bền của bộ lọc tách khối được thiết lập theo quy tắc liên quan đến độ phân giải và/hoặc kích thước của ảnh tham chiếu so với độ phân giải và/hoặc kích thước của khối video hiện tại.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, ảnh tham chiếu của khối video hiện tại được lấy mẫu lại theo quy tắc dựa trên các kích thước của khối video hiện tại.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, sử dụng công cụ mã hóa trên khối video hiện tại được kích hoạt hoặc vô hiệu hóa có chọn lọc tùy thuộc độ phân giải/kích thước của ảnh tham chiếu của khối video hiện tại so với độ phân giải/kích thước của khối video hiện tại.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa nhiều khối video và các biểu diễn được mã hóa của nhiều khối video, trong đó,

trong khi biến đổi, cửa sổ tương hợp thứ nhất được định nghĩa đối với khối video thứ nhất và cửa sổ tương hợp thứ hai đối với khối video thứ hai, và trong đó tỷ lệ của chiều rộng và/hoặc chiều cao của cửa sổ tương hợp thứ nhất to cửa sổ tương hợp thứ hai tuân theo quy tắc dựa vào ít nhất dòng bit tương hợp.

Theo khía cạnh đại diện khác, sáng chế có thể được sử dụng để đề cập đến phương pháp xử lý video khác. Phương pháp bao gồm bước thực hiện biến đổi giữa nhiều khối video và các biểu diễn được mã hóa của nhiều khối video, trong đó, trong khi biến đổi, cửa sổ tương hợp thứ nhất được định nghĩa đối với khối video thứ nhất và cửa sổ tương hợp thứ hai đối với khối video thứ hai, và trong đó tỷ lệ của chiều rộng và/hoặc chiều cao của cửa sổ tương hợp thứ nhất trên cửa sổ tương hợp thứ hai tuân theo quy tắc dựa vào ít nhất dòng bit tương hợp.

Ngoài ra, theo khía cạnh đại diện, thiết bị trong hệ thống video bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó được bộc lộ. Các lệnh khi thực thi bằng bộ xử lý, khiến bộ xử lý thực hiện một hoặc nhiều phương pháp được bộc lộ.

Theo một khía cạnh đại diện, sáng chế đề cập đến thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu ở đây.

Theo một khía cạnh đại diện, sáng chế đề cập đến thiết bị mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu ở đây.

Sáng chế cũng đề cập đến sản phẩm chương trình máy tính được lưu trữ trên phương tiện máy tính đọc được bất biến, sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện một hoặc nhiều phương pháp được bộc lộ.

Các khía cạnh và các dấu hiệu nêu trên và khác của sáng chế được mô tả chi tiết hơn trong các hình vẽ, phần mô tả và các điểm yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ thể hiện ví dụ về vector chuyển động khối phụ (VSB) và hiệu số vector chuyển động;

Fig.2 là hình vẽ thể hiện ví dụ về khối video 16x16 được phân chia thành 16 vùng 4x4;

Fig.3A là hình vẽ thể hiện ví dụ về vị trí cụ thể trong mẫu;

Fig.3B là hình vẽ thể hiện ví dụ khác của vị trí cụ thể trong mẫu;

Fig.3C là hình vẽ thể hiện ví dụ khác nữa về vị trí cụ thể trong mẫu;

Fig.4A là hình vẽ thể hiện ví dụ về các vị trí của mẫu hiện tại và vị trí của mẫu tham chiếu;

Fig.4B là hình vẽ thể hiện ví dụ khác của các vị trí của mẫu hiện tại và các vị trí của mẫu tham chiếu;

Fig.5 là sơ đồ khối của ví dụ của nền tảng phần cứng để thực hiện kỹ thuật giải mã hoặc mã hóa phương tiện hình ảnh được nêu trong bản mô tả.

Fig.6 là hình vẽ thể hiện lưu đồ của phương pháp mã hóa video lấy làm ví dụ;

Fig.7 là hình vẽ thể hiện ví dụ của kỹ thuật tinh lọc vector chuyển động phía bộ giải mã;

Fig.8 là hình vẽ thể hiện ví dụ về luồng xếp tầng các quá trình DMVR và BDOF trong VTM5.0. Các hoạt động DMVR SAD và các hoạt động BDOF SAD là khác nhau và không được chia sẻ;

Fig.9 là sơ đồ khối của hệ thống xử lý video ví dụ trong đó các phương án thực hiện có thể được triển khai;

Fig.10 là sơ đồ khối minh họa hệ thống tạo mã video ví dụ;

Fig.11 là sơ đồ khối minh họa bộ mã hóa theo một số phương án thực hiện sáng chế;

Fig.12 là sơ đồ khối minh họa bộ giải mã theo một số phương án thực hiện sáng chế;

Fig.13 là lưu đồ biểu diễn phương pháp xử lý video theo sáng chế;

Fig.14 là lưu đồ biểu diễn phương pháp khác xử lý video theo sáng chế; và

Fig.15 là lưu đồ biểu diễn phương pháp khác xử lý video nữa theo sáng chế.

Mô tả chi tiết sáng chế

1. Mã hóa video trong HEVC/H.265

Các chuẩn mã hóa video đã phát triển chủ yếu qua sự phát triển của các chuẩn ITU-T và ISO/IEC đã biết. ITU-T đã tạo ra các chuẩn H.261 và H.263, ISO/IEC đã tạo ra các chuẩn MPEG-1 và MPEG-4 hình ảnh, và hai tổ chức này đồng thời đã tạo ra các chuẩn H.262/MPEG-2 Video và H.264/MPEG-4 mã hóa video cao cấp (AVC) và H.265/HEVC. Kể từ chuẩn H.262, các chuẩn mã hóa video dựa trên cấu

trúc mã hóa video lai trong đó sử dụng dự báo thời gian cộng mã hóa biến đổi. Để khai thác các công nghệ mã hóa video tương lai ngoài HEVC, nhóm khai thác video chung (JVET) được thành lập bởi VCEG và MPEG đồng thời vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp thuận bởi JVET và được đưa vào phần mềm tham chiếu có tên là mô hình khai thác chung (JEM). Vào tháng 4 2018, nhóm chuyên gia video chung (JVET) giữa VCEG (Q6/16) và ISO/IEC JTC1 SC29/WG11 (MPEG) được thành lập để làm việc trên chuẩn VVC nhằm giảm 50% tốc độ bit so với HEVC.

2. Tổng quan

2.1. Thay đổi độ phân giải thích ứng (ARC)

AVC và HEVC không có khả năng thay đổi độ phân giải mà không phải đưa vào IDR hoặc ảnh điểm truy nhập ngẫu nhiên nội vùng (IRAP); khả năng này có thể được gọi là thay đổi độ phân giải thích ứng (ARC). Có các trường hợp sử dụng hoặc các kịch bản ứng dụng sẽ hưởng lợi từ tính năng ARC, bao gồm các dấu hiệu sau:

- Thích nghi tốc độ trong cuộc gọi video và hội đàm: Để thích ứng video được mã hóa với các điều kiện mạng thay đổi, khi các điều kiện mạng trở nên xấu hơn do vậy băng thông khả dụng trở nên thấp hơn, bộ mã hóa có thể thích nghi với nó bằng cách mã hóa các ảnh có độ phân giải nhỏ hơn. Hiện tại, việc thay đổi độ phân giải ảnh có thể được thực hiện chỉ sau ảnh IRAP; điều này có vài vấn đề. ảnh IRAP có chất lượng hợp lý sẽ lớn hơn nhiều ảnh được mã hóa ngoài và sẽ trở nên phức tạp hơn tương ứng để giải mã: điều này tiêu tốn thời gian và tài nguyên. Đây là vấn đề nếu thay đổi độ phân giải được yêu cầu bởi bộ giải mã đối với các nguyên nhân về tải. Cũng có thể phá hỏng các điều kiện bộ đệm độ trễ thấp, ép buộc đồng bộ lại audio, và độ trễ từ đầu này đến đầu kia của dòng sẽ tăng, ít nhất tạm thời. Điều này có thể khiến trải nghiệm người dùng xấu.

- Các thay đổi loa hoạt động trong hội đàm video đa bên: Để hội đàm video đa bên, thông thường loa hoạt động được thể hiện trong kích thước video lớn hơn so với video cho những người tham gia hội đàm còn lại. Khi loa hoạt động thay đổi, độ phân giải ảnh cho mỗi người tham gia có thể cũng cần được điều chỉnh. Cần có tính

năng ARC trở nên quan trọng hơn khi thay đổi này trong loa hoạt động xuất hiện thường xuyên.

- Bắt đầu nhanh trong streaming: Đối với ứng dụng streaming, thông thường ứng dụng sẽ đệm tới chiều dài cụ thể của ảnh được giải mã trước khi bắt đầu hiển thị. Khởi động dòng bit có độ phân giải nhỏ hơn sẽ cho phép ứng dụng có đủ ảnh trong bộ đệm để bắt đầu hiển thị nhanh hơn.

Chuyển đổi dòng thích ứng trong streaming: Đặc tả truyền trực tuyến thích ứng động trên HTTP (Dynamic Adaptive Streaming over HTTP, DASH) bao gồm tính năng có tên `@mediaStreamStructureId`. Điều này có thể chuyển đổi giữa các biểu diễn khác nhau ở các điểm truy nhập ngẫu nhiên GOP mở có các ảnh dẫn đầu không thể giải mã, chẳng hạn, các ảnh CRA có các ảnh RASL liên kết trong HEVC. Khi hai biểu diễn khác nhau của cùng video có các tốc độ bit khác nhau nhưng có cùng độ phân giải không gian trong khi chúng có cùng giá trị của `@mediaStreamStructureId`, chuyển đổi giữa hai biểu diễn ở ảnh CRA có các ảnh RASL liên kết có thể được thực hiện, và các ảnh RASL được liên kết với các ảnh CRA đang chuyển mạch có thể được giải mã với chất lượng chấp nhận được do vậy cho phép chuyển đổi liên tục. Với ARC, đặc điểm `@mediaStreamStructureId` cũng sẽ có thể sử dụng để chuyển đổi giữa các biểu diễn DASH có các độ phân giải không gian khác nhau.

ARC cũng được biết đến như là biến đổi độ phân giải động.

ARC có thể cũng được xem là trường hợp đặc biệt của lấy mẫu lại ảnh tham chiếu (RPR) chẳng hạn H.263 Annex P.

2.2. Lấy mẫu lại ảnh tham chiếu trong H.263 Annex P

Chế độ này mô tả thuật toán để làm cong ảnh tham chiếu trước khi sử dụng để dự báo. Có thể hữu ích khi lấy mẫu lại ảnh tham chiếu có định dạng nguồn khác với ảnh được dự báo. Cũng có thể được sử dụng để ước tính chuyển động toàn cục, hoặc ước tính chuyển động quay, bằng cách làm cong hình dạng, kích thước, và vị trí của ảnh tham chiếu. Cú pháp bao gồm các tham số làm cong được sử dụng cũng như thuật toán lấy mẫu lại. Mức độ hoạt động đơn giản nhất để chế độ lấy mẫu lại ảnh tham chiếu là lấy mẫu lại hệ số ngàm bằng 4 do chỉ bộ lọc FIR cần được áp dụng cho các quá trình tăng và giảm kích thước mẫu. Trong trường hợp này, không

yêu cầu thêm phụ tải báo hiệu do việc sử dụng của chúng được hiểu khi kích thước của ảnh mới (được chỉ báo trong tiêu đề ảnh) khác với kích thước của ảnh trước đó.

2.3. Cửa sổ tương hợp trong VVC

Cửa sổ tương hợp trong VVC định nghĩa hình chữ nhật. Các mẫu bên trong cửa sổ tương hợp thuộc ảnh liên quan. Các mẫu bên ngoài cửa sổ tương hợp có thể bị loại bỏ khi được xuất ra.

Khi cửa sổ tương hợp được áp dụng, hệ số chia tỷ lệ trong RPR được dẫn xuất dựa trên các cửa sổ tương hợp.

Cú pháp RBSP tập hợp tham số ảnh

pic parameter set rbsp() {	Mô tả
pps_pic_parameter_set_id	<u>ue(v)</u>
pps_seq_parameter_set_id	<u>ue(v)</u>
pic_width_in_luma_samples	<u>ue(v)</u>
pic_height_in_luma_samples	<u>ue(v)</u>
<i>conformance_window_flag</i>	<u>u(1)</u>
<i>if(conformance_window_flag) {</i>	
<i>conf_win_left_offset</i>	<u>ue(v)</u>
<i>conf_win_right_offset</i>	<u>ue(v)</u>
<i>conf_win_top_offset</i>	<u>ue(v)</u>
<i>conf_win_bottom_offset</i>	<u>ue(v)</u>
<i>}</i>	

pic_width_in_luma_samples xác định chiều rộng của mỗi ảnh được giải mã viện dẫn đến PPS theo các đơn vị của mẫu độ sáng. **pic_width_in_luma_samples** sẽ không bằng 0, sẽ là bội số nguyên của $\text{Max}(8, \text{MinCbSizeY})$, và sẽ nhỏ hơn hoặc bằng **pic_width_max_in_luma_samples**.

Khi **subpics_present_flag** bằng 1, giá trị của **pic_width_in_luma_samples** sẽ bằng **pic_width_max_in_luma_samples**.

pic_height_in_luma_samples xác định chiều cao của mỗi ảnh được giải mã viện dẫn đến PPS theo các đơn vị của mẫu độ sáng. **pic_height_in_luma_samples** sẽ không bằng 0 và sẽ là bội số nguyên của $\text{Max}(8, \text{MinCbSizeY})$, và sẽ nhỏ hơn hoặc bằng **pic_height_max_in_luma_samples**.

Khi **subpics_present_flag** bằng 1, giá trị của **pic_height_in_luma_samples** sẽ bằng **pic_height_max_in_luma_samples**.

Giả sử **refPicWidthInLumaSamples** và **refPicHeightInLumaSamples** lần lượt là **pic_width_in_luma_samples** và **pic_height_in_luma_samples**, của ảnh tham chiếu

của ảnh hiện tại viện dẫn đến PPS này. Tương hợp dòng bit yêu cầu rằng tất cả các điều kiện sau được thỏa mãn:

- *pic_width_in_luma_samples * 2 sẽ lớn hơn hoặc bằng refPicWidthInLumaSamples.*
- *pic_height_in_luma_samples * 2 sẽ lớn hơn hoặc bằng refPicHeightInLumaSamples.*
- *pic_width_in_luma_samples sẽ nhỏ hơn hoặc bằng refPicWidthInLumaSamples * 8.*
- *pic_height_in_luma_samples sẽ nhỏ hơn hoặc bằng refPicHeightInLumaSamples * 8.*

conformance_window_flag bằng 1 chỉ báo rằng các tham số độ lệch cửa sổ cắt tương hợp tiếp sau trong SPS. **conformance_window_flag** bằng 0 chỉ báo rằng các tham số độ lệch cửa sổ cắt tương hợp không xuất hiện.

conf_win_left_offset, **conf_win_right_offset**, **conf_win_top_offset**, và **conf_win_bottom_offset** xác định các mẫu của các ảnh trong CVS được xuất ra từ quá trình giải mã, liên quan đến vùng chữ nhật được xác định trong các tọa độ ảnh để xuất ra. Khi **conformance_window_flag** bằng 0, các giá trị của **conf_win_left_offset**, **conf_win_right_offset**, **conf_win_top_offset**, và **conf_win_bottom_offset** được suy ra bằng 0.

Cửa sổ cắt tương hợp chứa các mẫu độ sáng có các tọa độ ảnh ngang từ $\text{SubWidthC} * \text{conf_win_left_offset}$ sang $\text{pic_width_in_luma_samples} - (\text{SubWidthC} * \text{conf_win_right_offset} + 1)$ và các tọa độ ảnh dọc từ $\text{SubHeightC} * \text{conf_win_top_offset}$ sang $\text{pic_height_in_luma_samples} - (\text{SubHeightC} * \text{conf_win_bottom_offset} + 1)$, bao gồm hai giá trị đầu mút.

Giá trị của $\text{SubWidthC} * (\text{conf_win_left_offset} + \text{conf_win_right_offset})$ sẽ nhỏ hơn $\text{pic_width_in_luma_samples}$, và giá trị của $\text{SubHeightC} * (\text{conf_win_top_offset} + \text{conf_win_bottom_offset})$ sẽ nhỏ hơn $\text{pic_height_in_luma_samples}$.

Các biến **PicOutputWidthL** and **PicOutputHeightL** được dẫn xuất như sau:

$$\text{PicOutputWidthL} = \text{pic_width_in_luma_samples} - \text{SubWidthC} * (\text{conf_win_right_offset} + \text{conf_win_left_offset}) \quad (7-43)$$

$$\text{PicOutputHeightL} = \text{pic_height_in_pic_size_units} - \text{SubHeightC} * (\text{conf_win_bottom_offset} + \text{conf_win_top_offset}) \quad (7-44)$$

Khi ChromaArrayType không bằng 0, các mẫu được xác định tương ứng của hai mảng sắc độ là các mẫu có các tọa độ ảnh ($x / \text{SubWidthC}$, $y / \text{SubHeightC}$), trong đó (x , y) là các tọa độ ảnh của mẫu độ sáng được xác định.

Lưu ý – Các tham số độ lệch của số cắt tương hợp chỉ được áp dụng ở đầu ra. Tất cả các quá trình giải mã nội bộ được áp dụng cho kích thước ảnh không được cắt.

Giả sử ppsA và ppsB là bất kỳ hai PPS viện dẫn đến cùng SPS. Tương hợp dòng bit yêu cầu rằng, khi ppsA và ppsB lần lượt có các giá trị giống nhau của pic_width_in_luma_samples và pic_height_in_luma_samples, ppsA và ppsB lần lượt sẽ có các giá trị giống nhau của conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, và conf_win_bottom_offset.

2.4. Lấy mẫu lại ảnh tham chiếu (RPR)

Theo một số phương án thực hiện, ARC cũng được biết đến là lấy mẫu lại ảnh tham chiếu (RPR). Với RPR, TMVP bị vô hiệu hóa nếu ảnh cùng vị trí có độ phân giải khác nhau với ảnh hiện tại. Ngoài ra, kỹ thuật luồng quang hai hướng (BDof) và kỹ thuật tinh lọc MV phía bộ giải mã (DMVR) bị vô hiệu hóa khi ảnh tham chiếu có độ phân giải khác nhau với ảnh hiện tại.

Để xử lý MV thông thường khi ảnh tham chiếu có độ phân giải khác với ảnh hiện tại, phần nội suy được định nghĩa dưới đây:

8.5.6.3 Quá trình nội suy mẫu phân số

8.5.6.3.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (x_{Sb} , y_{Sb}) xác định mẫu trên bên trái của khối phụ mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến sbWidth xác định chiều rộng của khối phụ mã hiện tại,
- biến sbHeight xác định chiều cao của khối phụ mã hiện tại,
- độ lệch vector chuyển động mvOffset,
- vector chuyển động được tinh lọc refMvLX,
- mảng mẫu ảnh tham chiếu được chọn refPicLX,
- chỉ số bộ lọc nội suy nửa mẫu hpelIdx,
- cờ luồng quang hai hướng bdofFlag,

- biến `cIdx` xác định chỉ số thành phần màu sắc của khối hiện tại.

Các đầu ra của quá trình này là:

- mảng $(sbWidth + brdExtSize) \times (sbHeight + brdExtSize)$ `predSamplesLX` của các giá trị mẫu dự báo.

Kích thước mở rộng đường biên khối dự báo `brdExtSize` được dẫn xuất như sau:

$$brdExtSize = (bdoFlag \quad || \quad (inter_affine_flag[xSb][ySb] \quad \&\& \quad sps_affine_prof_enabled_flag)) ? 2 : 0 \quad (8-752)$$

Biến `fRefWidth` được gán bằng `PicOutputWidthL` của ảnh tham chiếu trong mẫu độ sáng.

Biến `fRefHeight` được gán bằng `PicOutputHeightL` của ảnh tham chiếu trong mẫu độ sáng.

Vectơ chuyển động `mvLX` được gán bằng $(refMvLX - mvOffset)$.

- Nếu `cIdx` bằng 0, điều kiện sau áp dụng:

- Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$hori_scale_fp = ((fRefWidth \ll 14) + (PicOutputWidthL \gg 1)) / PicOutputWidthL \quad (8-753)$$

$$vert_scale_fp = ((fRefHeight \ll 14) + (PicOutputHeightL \gg 1)) / PicOutputHeightL \quad (8-754)$$

- Giả sử $(xIntL, yIntL)$ là vị trí độ sáng theo các đơn vị mẫu toàn phần và $(xFracL, yFracL)$ là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu `refPicLX`.

- Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbIntL, ySbIntL)$ được gán bằng $(xSb + (mvLX[0] \gg 4), ySb + (mvLX[1] \gg 4))$.

- Đối với mỗi vị trí mẫu độ sáng $(x_L = 0..sbWidth - 1 + brdExtSize, y_L = 0..sbHeight - 1 + brdExtSize)$ bên trong mảng mẫu độ sáng dự báo `predSamplesLX`, giá trị mẫu độ sáng dự báo tương ứng `predSamplesLX[xL][yL]` được dẫn xuất như sau:

- Giả sử $(refxSb_L, refySb_L)$ và $(refx_L, refy_L)$ là các vị trí độ sáng được trở đến bởi vectơ chuyển động $(refMvLX[0], refMvLX[1])$ theo các đơn vị 1/16 mẫu. Các biến `refxSb_L`, `refx_L`, `refySb_L`, và `refy_L` được dẫn xuất như sau:

$$\text{refxSb}_L = ((\text{xSb} \ll 4) + \text{refMvLX}[0]) * \text{hori_scale_fp} \quad (8-755)$$

$$\begin{aligned} \text{refx}_L &= ((\text{Sign}(\text{refxSb}) * ((\text{Abs}(\text{refxSb}) + 128) \gg 8) \\ &\quad + \text{x}_L * ((\text{hori_scale_fp} + 8) \gg 4)) + 32) \gg 6 \end{aligned} \quad (8-756)$$

$$\text{refySb}_L = ((\text{ySb} \ll 4) + \text{refMvLX}[1]) * \text{vert_scale_fp} \quad (8-757)$$

$$\begin{aligned} \text{refy}_L &= ((\text{Sign}(\text{refySb}) * ((\text{Abs}(\text{refySb}) + 128) \gg 8) + \text{y}_L * \\ &\quad ((\text{vert_scale_fp} + 8) \gg 4)) + 32) \gg 6 \end{aligned} \quad (8-758)$$

– Các biến xInt_L , yInt_L , xFrac_L và yFrac_L được dẫn xuất như sau:

$$\text{xInt}_L = \text{refx}_L \gg 4 \quad (8-759)$$

$$\text{yInt}_L = \text{refy}_L \gg 4 \quad (8-760)$$

$$\text{xFrac}_L = \text{refx}_L \& 15 \quad (8-761)$$

$$\text{yFrac}_L = \text{refy}_L \& 15 \quad (8-762)$$

– Nếu bdoFlag bằng TRUE hoặc ($\text{sps_affine_prof_enabled_flag}$ bằng TRUE và $\text{inter_affine_flag}[\text{xSb}][\text{ySb}]$ bằng TRUE), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 với $(\text{xInt}_L + (\text{xFrac}_L \gg 3) - 1)$, $\text{yInt}_L + (\text{yFrac}_L \gg 3) - 1$ và refPicLX làm đầu vào.

– x_L bằng 0.

– x_L bằng $\text{sbWidth} + 1$.

– y_L bằng 0.

– y_L bằng $\text{sbHeight} + 1$.

– Ngược lại, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(\text{xInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $(\text{yInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $(\text{xFrac}_L, \text{yFrac}_L)$, $(\text{xSbInt}_L, \text{ySbInt}_L)$, refPicLX , hpelIdx , sbWidth , sbHeight and (xSb, ySb) làm đầu vào.

– Ngược lại (cIdx không bằng 0), điều kiện sau áp dụng:

– Giả sử $(\text{xInt}_C, \text{yInt}_C)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(\text{xFrac}_C, \text{yFrac}_C)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu refPicLX .

- Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (x_{SbIntC} , y_{SbIntC}) được gán bằng $((x_{Sb} / SubWidthC) + (mvLX[0] \gg 5), (y_{Sb} / SubHeightC) + (mvLX[1] \gg 5))$.
- Đối với mỗi vị trí mẫu sắc độ ($x_C = 0..sbWidth - 1$, $y_C = 0..sbHeight - 1$) bên trong các mảng mẫu sắc độ dự báo $predSamplesLX$, giá trị mẫu sắc độ dự báo tương ứng $predSamplesLX[x_C][y_C]$ được dẫn xuất như sau:
 - Giả sử $(refxSb_C, refySb_C)$ và $(refx_C, refy_C)$ là các vị trí sắc độ được trở đến bởi vectơ chuyển động $(mvLX[0], mvLX[1])$ theo các đơn vị 1/32 mẫu. Các biến $refxSb_C$, $refySb_C$, $refx_C$ và $refy_C$ được dẫn xuất như sau:

$$refxSb_C = ((x_{Sb} / SubWidthC \ll 5) + mvLX[0]) * hori_scale_fp \quad (8-763)$$

$$refx_C = ((Sign(refxSb_C) * ((Abs(refxSb_C) + 256) \gg 9) + x_C * ((hori_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-764)$$

$$refySb_C = ((y_{Sb} / SubHeightC \ll 5) + mvLX[1]) * vert_scale_fp \quad (8-765)$$

$$refy_C = ((Sign(refySb_C) * ((Abs(refySb_C) + 256) \gg 9) + y_C * ((vert_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-766)$$
 - Các biến $xInt_C$, $yInt_C$, $xFrac_C$ và $yFrac_C$ được dẫn xuất như sau:

$$xInt_C = refx_C \gg 5 \quad (8-767)$$

$$yInt_C = refy_C \gg 5 \quad (8-768)$$

$$xFrac_C = refy_C \& 31 \quad (8-769)$$

$$yFrac_C = refy_C \& 31 \quad (8-770)$$
- Giá trị mẫu dự báo $predSamplesLX[x_C][y_C]$ được dẫn xuất bằng cách gọi quá trình được xác định trong mệnh đề 8.5.6.3.4 với $(xInt_C, yInt_C)$, $(xFrac_C, yFrac_C)$, $(xSbInt_C, ySbInt_C)$, $sbWidth$, $sbHeight$ và $refPicLX$ làm đầu vào.

8.5.6.3.2 Quá trình lọc nội suy mẫu độ sáng

Các đầu vào cho quá trình này là:

- vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_L$, $yInt_L$),
- vị trí độ sáng theo các đơn vị mẫu phân số ($xFrac_L$, $yFrac_L$),
- vị trí độ sáng theo các đơn vị mẫu toàn phần ($xSbInt_L$, $ySbInt_L$) xác định mẫu trên bên trái của khối giới hạn để đệm mẫu tham chiếu so với mẫu độ sáng trên bên trái của ảnh tham chiếu,
- mảng mẫu tham chiếu độ sáng $refPicLX_L$,

- chỉ số bộ lọc nội suy nửa mẫu $hpellfIdx$,
- biến $sbWidth$ xác định chiều rộng của khối phụ hiện tại,
- biến $sbHeight$ xác định chiều cao của khối phụ hiện tại,
- vị trí độ sáng (xSb , ySb) xác định mẫu trên bên trái của khối phụ hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

Đầu ra của quá trình này là giá trị mẫu độ sáng được dự báo $predSampleLXL$

Các biến $shift1$, $shift2$ và $shift3$ được dẫn xuất như sau:

- Biến $shift1$ được gán bằng $\text{Min}(4, \text{BitDepth}_Y - 8)$, biến $shift2$ được gán bằng 6 và biến $shift3$ được gán bằng $\text{Max}(2, 14 - \text{BitDepth}_Y)$.
- Biến $picW$ được gán bằng $pic_width_in_luma_samples$ và biến $picH$ được gán bằng $pic_height_in_luma_samples$.

Các hệ số lọc nội suy độ sáng $f_L[p]$ đối với mỗi vị trí mẫu phân số $1/16$ p bằng $xFrac_L$ hoặc $yFrac_L$ được dẫn xuất như sau:

- Nếu $\text{MotionModelIdx}[xSb][ySb]$ lớn hơn 0, và $sbWidth$ và $sbHeight$ đều bằng 4, các hệ số lọc nội suy độ sáng $f_L[p]$ được xác định trong Bảng 2.
- Ngược lại, các hệ số lọc nội suy độ sáng $f_L[p]$ được xác định trong Bảng 1 tùy thuộc $hpellfIdx$.

Các vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_i$, $yInt_i$) được dẫn xuất như sau với $i = 0..7$:

- Nếu $subpic_treated_as_pic_flag[\text{SubPicIdx}]$ bằng 1, điều kiện sau áp dụng:

$$xInt_i = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos},$$

$$xInt_L + i - 3) \quad (8-771)$$

$$yInt_i = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos},$$

$$yInt_L + i - 3) \quad (8-772)$$

- Ngược lại ($subpic_treated_as_pic_flag[\text{SubPicIdx}]$ bằng 0), điều kiện sau áp dụng:

$$\begin{aligned} xInt_i &= \text{Clip3}(0, \quad picW - 1, \quad sps_ref_wraparound_enabled_flag \quad ? \\ &\quad \text{ClipH}((sps_ref_wraparound_offset_minus1 + 1) * \text{MinCbSizeY}, picW, xInt_L + i - 3) \quad : \\ &\quad (8-773) \end{aligned}$$

$$xInt_L + i - 3)$$

$$yInt_i = \text{Clip3}(0, picH - 1, yInt_L + i - 3) \quad (8-774)$$

Các vị trí độ sáng theo các đơn vị mẫu toàn phần còn được chỉnh sửa như sau với $i = 0..7$:

$$xInt_i = \text{Clip3}(xSbInt_L - 3, xSbInt_L + sbWidth + 4, xInt_i) \quad (8-775)$$

$$yInt_i = \text{Clip3}(ySbInt_L - 3, ySbInt_L + sbHeight + 4, yInt_i) \quad (8-776)$$

Giá trị mẫu độ sáng được dự báo predSampleLXL được dẫn xuất như sau:

- Nếu cả $x\text{Frac}_L$ lẫn $y\text{Frac}_L$ bằng 0, giá trị của predSampleLXL được dẫn xuất như sau:

$$\text{predSampleLXL} = \text{refPicLXL}[xInt_3][yInt_3] \ll \text{shift3} \quad (8-777)$$

- Ngược lại, nếu $x\text{Frac}_L$ không bằng 0 và $y\text{Frac}_L$ bằng 0, giá trị của predSampleLXL được dẫn xuất như sau:

$$\text{predSampleLXL} = \left(\sum_{i=0}^7 f_L[x\text{Frac}_L][i] * \text{refPicLXL}[xInt_i][yInt_3] \right) \gg \text{shift1} \quad (8-778)$$

- Ngược lại, if $x\text{Frac}_L$ bằng 0 and $y\text{Frac}_L$ không bằng 0, giá trị của predSampleLXL được dẫn xuất như sau:

$$\text{predSampleLXL} = \left(\sum_{i=0}^7 f_L[y\text{Frac}_L][i] * \text{refPicLXL}[xInt_3][yInt_i] \right) \gg \text{shift1} \quad (8-779)$$

- Ngược lại, nếu $x\text{Frac}_L$ không bằng 0 và $y\text{Frac}_L$ không bằng 0, giá trị của predSampleLXL được dẫn xuất như sau:

- Mảng mẫu $\text{temp}[n]$ với $n = 0..7$, được dẫn xuất như sau:

$$\text{temp}[n] = \left(\sum_{i=0}^7 f_L[x\text{Frac}_L][i] * \text{refPicLXL}[xInt_i][yInt_n] \right) \gg \text{shift1} \quad (8-780)$$

- Giá trị mẫu độ sáng được dự báo predSampleLXL được dẫn xuất như sau:

$$\text{predSampleLXL} = \left(\sum_{i=0}^7 f_L[y\text{Frac}_L][i] * \text{temp}[i] \right) \gg \text{shift2} \quad (8-781)$$

Bảng 8-11 – Mô tả các hệ số lọc nội suy độ sáng $f_L[p]$ đối với mỗi vị trí mẫu phân số $1/16$ p.

Vị trí mẫu phân số p	các hệ số lọc nội suy							
	$f_L[p][0]$	$f_L[p][1]$	$f_L[p][2]$	$f_L[p][3]$	$f_L[p][4]$	$f_L[p][5]$	$f_L[p][6]$	$f_L[p][7]$
1	0	1	-3	63	4	-2	1	0
2	-1	2	-5	62	8	-3	1	0
3	-1	3	-8	60	13	-4	1	0
4	-1	4	-10	58	17	-5	1	0
5	-1	4	-11	52	26	-8	3	-1
6	-1	3	-9	47	31	-10	4	-1
7	-1	4	-11	45	34	-10	4	-1
8 (hpelIfIdx == 0)	-1	4	-11	40	40	-11	4	-1
8 (hpelIfIdx == 1)	0	3	9	20	20	9	3	0
9	-1	4	-10	34	45	-11	4	-1
10	-1	4	-10	31	47	-9	3	-1
11	-1	3	-8	26	52	-11	4	-1
12	0	1	-5	17	58	-10	4	-1
13	0	1	-4	13	60	-8	3	-1
14	0	1	-3	8	62	-5	2	-1
15	0	1	-2	4	63	-3	1	0

Bảng 8-12 – Mô tả các hệ số lọc nội suy độ sáng $f_L[p]$ đối với mỗi vị trí mẫu phân số $1/16 p$ đối với chế độ chuyển động affin.

Vị trí mẫu phân số p	các hệ số lọc nội suy							
	$f_L[p][0]$	$f_L[p][1]$	$f_L[p][2]$	$f_L[p][3]$	$f_L[p][4]$	$f_L[p][5]$	$f_L[p][6]$	$f_L[p][7]$
1	0	1	-3	63	4	-2	1	0
2	0	1	-5	62	8	-3	1	0
3	0	2	-8	60	13	-4	1	0
4	0	3	-10	58	17	-5	1	0
5	0	3	-11	52	26	-8	2	0
6	0	2	-9	47	31	-10	3	0
7	0	3	-11	45	34	-10	3	0
8	0	3	-11	40	40	-11	3	0
9	0	3	-10	34	45	-11	3	0
10	0	3	-10	31	47	-9	2	0
11	0	2	-8	26	52	-11	3	0
12	0	1	-5	17	58	-10	3	0
13	0	1	-4	13	60	-8	2	0
14	0	1	-3	8	62	-5	1	0
15	0	1	-2	4	63	-3	1	0

8.5.6.3.3 Quá trình nạp mẫu số nguyên độ sáng

Các đầu vào cho quá trình này là:

- vị trí độ sáng theo các đơn vị mẫu toàn phần (x_{IntL} , y_{IntL}),
- mảng mẫu tham chiếu độ sáng $refPicLXL$,

Đầu ra của quá trình này là giá trị mẫu độ sáng được dự báo $predSampleLXL$

Biến shift được gán bằng $\text{Max}(2, 14 - \text{BitDepth}_Y)$.

Biến picW được gán bằng pic_width_in_luma_samples and biến picH được gán bằng pic_height_in_luma_samples.

Các vị trí độ sáng theo các đơn vị mẫu toàn phần (xInt, yInt) được dẫn xuất như sau:

$$xInt = \text{Clip3}(0, picW - 1, sps_ref_wraparound_enabled_flag ? \quad (8-782)$$

$$\text{ClipH}((sps_ref_wraparound_offset_minus1 + 1) * \text{MinCbSizeY}, picW, xInt_L) : xInt_L)$$

$$yInt = \text{Clip3}(0, picH - 1, yInt_L) \quad (8-783)$$

Giá trị mẫu độ sáng được dự báo predSampleLX_L được dẫn xuất như sau:

$$\text{predSampleLX}_L = \text{refPicLX}_L[xInt][yInt] \ll \text{shift3} \quad (8-784)$$

8.5.6.3.4 Quá trình nội suy mẫu sắc độ

Các đầu vào cho quá trình này là:

- vị trí sắc độ theo các đơn vị mẫu toàn phần (xInt_C, yInt_C),
- vị trí sắc độ theo các đơn vị mẫu phân số 1/32 (xFrac_C, yFrac_C),
- vị trí sắc độ theo các đơn vị mẫu toàn phần (xSbInt_C, ySbInt_C) xác định mẫu trên bên trái của khối giới hạn để đệm mẫu tham chiếu so với mẫu sắc độ trên bên trái của ảnh tham chiếu,
- biến sbWidth xác định chiều rộng của khối phụ hiện tại,
- biến sbHeight xác định chiều cao của khối phụ hiện tại,
- mảng mẫu tham chiếu sắc độ refPicLX_C.

Đầu ra của quá trình này là giá trị mẫu sắc độ được dự báo predSampleLX_C

Các biến shift1, shift2 và shift3 được dẫn xuất như sau:

- Biến shift1 được gán bằng $\text{Min}(4, \text{BitDepth}_C - 8)$, biến shift2 được gán bằng 6 và biến shift3 được gán bằng $\text{Max}(2, 14 - \text{BitDepth}_C)$.
- Biến picW_C được gán bằng $\text{pic_width_in_luma_samples} / \text{SubWidth}_C$ and biến picH_C được gán bằng $\text{pic_height_in_luma_samples} / \text{SubHeight}_C$.

Các hệ số lọc nội suy sắc độ fc[p] đối với mỗi vị trí mẫu phân số 1/32 p bằng xFrac_C hoặc yFrac_C được xác định trong Bảng 3.

Biến xOffset được gán bằng $(sps_ref_wraparound_offset_minus1 + 1) * \text{MinCbSizeY} / \text{SubWidth}_C$.

Các vị trí sắc độ theo các đơn vị mẫu toàn phần ($xInt_i$, $yInt_i$) được dẫn xuất như sau với $i = 0..3$:

- Nếu `subpic_treated_as_pic_flag[SubPicIdx]` bằng 1, điều kiện sau áp dụng:

$$xInt_i = Clip3(SubPicLeftBoundaryPos / SubWidthC, SubPicRightBoundaryPos / SubWidthC, xInt_L + i) \quad (8-785)$$

$$yInt_i = Clip3(SubPicTopBoundaryPos / SubHeightC, SubPicBotBoundaryPos / SubHeightC, yInt_L + i) \quad (8-786)$$

- Ngược lại (`subpic_treated_as_pic_flag[SubPicIdx]` bằng 0), điều kiện sau áp dụng:

$$\begin{aligned} xInt_i &= Clip3(0, \text{picWc} - 1, \\ &\text{sps_ref_wraparound_enabled_flag} ? ClipH(xOffset, \text{picWc}, xInt_C + i - 1) : \\ &xInt_C + i - 1) \end{aligned} \quad (8-787)$$

$$yInt_i = Clip3(0, \text{picHc} - 1, yInt_C + i - 1) \quad (8-788)$$

Các vị trí sắc độ theo các đơn vị mẫu toàn phần ($xInt_i$, $yInt_i$) còn được chỉnh sửa như sau với $i = 0..3$:

$$xInt_i = Clip3(xSbInt_C - 1, xSbInt_C + sbWidth + 2, xInt_i) \quad (8-789)$$

$$yInt_i = Clip3(ySbInt_C - 1, ySbInt_C + sbHeight + 2, yInt_i) \quad (8-790)$$

Giá trị mẫu sắc độ được dự báo predSampleLXc được dẫn xuất như sau:

- Nếu cả $xFracc$ lẫn $yFracc$ bằng 0, giá trị của predSampleLXc được dẫn xuất như sau:

$$\text{predSampleLXc} = \text{refPicLXc}[xInt_1][yInt_1] \ll \text{shift3} \quad (8-791)$$

- Ngược lại, nếu $xFracc$ không bằng 0 và $yFracc$ bằng 0, giá trị của predSampleLXc được dẫn xuất như sau:

$$\text{predSampleLXc} = \left(\sum_{i=0}^3 fc[xFracc][i] * \text{refPicLXc}[xInt_i][yInt_1] \right) \gg \text{shift1} \quad (8-792)$$

- Ngược lại, nếu $xFracc$ bằng 0 và $yFracc$ không bằng 0, giá trị của predSampleLXc được dẫn xuất như sau:

$$\text{predSampleLXc} = \left(\sum_{i=0}^3 fc[yFracc][i] * \text{refPicLXc}[xInt_1][yInt_i] \right) \gg \text{shift1} \quad (8-793)$$

- Ngược lại, nếu $xFracc$ không bằng 0 và $yFracc$ không bằng 0, giá trị của predSampleLXc được dẫn xuất như sau:

- Mảng mẫu $\text{temp}[n]$ với $n = 0..3$, được dẫn xuất như sau:

$$\begin{aligned} \text{temp}[n] &= \left(\sum_{i=0}^3 fc[xFracc][i] * \text{refPicLXc}[xInt_i][yInt_n] \right) \gg \text{shift1} \\ &\quad (8-794) \end{aligned}$$

- Giá trị mẫu sắc độ được dự báo predSampleLX_C được dẫn xuất như sau:

$$\begin{aligned} \text{predSampleLX}_C &= (\text{fc}[\text{yFracC}][0] * \text{temp}[0] + \\ &\quad \text{fc}[\text{yFracC}][1] * \text{temp}[1] + \\ &\quad \text{fc}[\text{yFracC}][2] * \text{temp}[2] + \quad (8-795) \\ &\quad \text{fc}[\text{yFracC}][3] * \text{temp}[3]) >> \text{shift2} \end{aligned}$$

Bảng 8-13 – Mô tả các hệ số lọc nội suy sắc độ $\text{fc}[p]$ đối với mỗi vị trí mẫu phân số $1/32 p$.

Vị trí mẫu phân số p	các hệ số lọc nội suy			
	$\text{fc}[p][0]$	$\text{fc}[p][1]$	$\text{fc}[p][2]$	$\text{fc}[p][3]$
1	-1	63	2	0
2	-2	62	4	0
3	-2	60	7	-1
4	-2	58	10	-2
5	-3	57	12	-2
6	-4	56	14	-2
7	-4	55	15	-2
8	-4	54	16	-2
9	-5	53	18	-2
10	-6	52	20	-2
11	-6	49	24	-3
12	-6	46	28	-4
13	-5	44	29	-4
14	-4	42	30	-4
15	-4	39	33	-4
16	-4	36	36	-4
17	-4	33	39	-4
18	-4	30	42	-4
19	-4	29	44	-5
20	-4	28	46	-6
21	-3	24	49	-6
22	-2	20	52	-6
23	-2	18	53	-5
24	-2	16	54	-4
25	-2	15	55	-4
26	-2	14	56	-4
27	-2	12	57	-3
28	-2	10	58	-2
29	-1	7	60	-2
30	0	4	62	-2
31	0	2	63	-1

2.5. Dự báo bù chuyển động afin dựa trên khối phụ được tính lọc

Các kỹ thuật được bộc lộ ở đây bao gồm phương pháp để tính lọc dự báo bù chuyển động afin dựa trên khối phụ có luồng quang. Sau khi bù chuyển động afin

dựa trên khối phụ được thực hiện, mẫu dự báo được tinh lọc bằng cách bổ sung hiệu số được dẫn xuất bởi phương trình luồng quang, mà được gọi là tinh lọc dự báo có luồng quang (PROF). Phương pháp được đề xuất có thể thực hiện dự báo ngoài theo độ hạt mức pixel mà không tăng băng thông truy nhập bộ nhớ.

Để đạt được độ hạt bù chuyển động mịn hơn, phần này đề xuất phương pháp để tinh lọc dự báo bù chuyển động afin dựa trên khối phụ có luồng quang. Sau khi bù chuyển động afin dựa trên khối phụ được thực hiện, mẫu dự báo độ sáng được tinh lọc bằng cách bổ sung hiệu số được dẫn xuất bởi phương trình luồng quang. PROF được đề xuất (tinh lọc dự báo có luồng quang) được mô tả dưới dạng bốn bước sau.

Bước 1) Bù chuyển động afin dựa trên khối phụ được thực hiện để tạo dự báo khối phụ $I(i, j)$.

Bước 2) Các gradien không gian $g_x(i, j)$ và $g_y(i, j)$ của dự báo khối phụ được tính toán ở mỗi vị trí mẫu nhờ sử dụng bộ lọc ba bậc $[-1, 0, 1]$.

$$g_x(i, j) = I(i + 1, j) - I(i - 1, j)$$

$$g_y(i, j) = I(i, j + 1) - I(i, j - 1)$$

Dự báo khối phụ được tăng lên một pixel ở mỗi phía để tính toán gradien. Để giảm băng thông bộ nhớ và độ phức tạp, các pixel ở các đường biên được mở rộng được sao chép từ vị trí pixel số nguyên gần nhất trong ảnh tham chiếu. Do vậy, nội suy bổ sung cho vùng đệm được tránh.

Bước 3) Tinh lọc dự báo độ sáng (ký hiệu ΔI) như được tính toán bằng phương trình luồng quang.

$$\Delta I(i, j) = g_x(i, j) * \Delta v_x(i, j) + g_y(i, j) * \Delta v_y(i, j)$$

trong đó ΔMV (được ký hiệu là $\Delta v(i, j)$) là hiệu số giữa MV pixel được tính toán cho vị trí mẫu (i, j) , được ký hiệu là $v(i, j)$, và MV khối phụ của khối phụ mà có pixel (i, j) , như được thể hiện trên Fig.1.

Do các tham số mô hình afin và vị trí pixel so với tâm của khối phụ không được thay đổi từ khối phụ sang khối phụ, $\Delta v(i, j)$ có thể được tính toán cho khối phụ thứ nhất, và được sử dụng lại cho các khối phụ khác trong cùng CU. Giả sử x và y là độ lệch ngang và dọc từ vị trí pixel đến điểm giữa của khối phụ, $\Delta v(x, y)$ có thể được dẫn xuất bằng phương trình sau,

$$\begin{cases} \Delta v_x(x, y) = c * x + d * y \\ \Delta v_y(x, y) = e * x + f * y \end{cases}$$

Đối với mô hình afin 4 tham số,

$$\begin{cases} c = f = \frac{v_{1x} - v_{0x}}{w} \\ e = -d = \frac{v_{1y} - v_{0y}}{w} \end{cases}$$

Đối với mô hình afin 6 tham số,

$$\begin{cases} c = \frac{v_{1x} - v_{0x}}{w} \\ d = \frac{v_{2x} - v_{0x}}{h} \\ e = \frac{v_{1y} - v_{0y}}{w} \\ f = \frac{v_{2y} - v_{0y}}{h} \end{cases}$$

trong đó (v_{0x}, v_{0y}) , (v_{1x}, v_{1y}) , (v_{2x}, v_{2y}) là các vector chuyển động điểm điều khiển trên bên trái, trên bên phải và dưới bên trái, w và h là chiều rộng và chiều cao của CU.

Bước 4) Cuối cùng, tinh lọc dự báo độ sáng được bổ sung vào dự báo khối phụ $I(i, j)$. Dự báo cuối cùng I' được tạo dưới dạng phương trình sau.

$$I'(i, j) = I(i, j) + \Delta I(i, j)$$

Một số chi tiết được mô tả dưới đây:

a) Cách thức dẫn xuất các gradient cho PROF

Theo một số phương án thực hiện, các gradient được tính toán đối với mỗi khối phụ (khối phụ 4×4 trong VTM-4.0) đối với mỗi danh sách tham chiếu. Đối với mỗi khối phụ, các mẫu số nguyên gần nhất của khối tham chiếu được nạp để đệm bốn đường mẫu phụ bên ngoài.

Giả sử MV đối với khối phụ hiện tại là (MV_x, MV_y) . Sau đó, phần phân số được tính toán là $(FracX, FracY) = (MV_x \& 15, MV_y \& 15)$. Phần nguyên được tính toán là $(IntX, IntY) = (MV_x \gg 4, MV_y \gg 4)$. Các độ lệch $(OffsetX, OffsetY)$ được dẫn xuất là:

$$OffsetX = FracX > 7 ? 1 : 0;$$

$$OffsetY = FracY > 7 ? 1 : 0;$$

Giả sử tọa độ trên bên trái của khối phụ hiện tại là (xCur, yCur) và các kích thước của khối phụ hiện tại là W×H. Sau đó (xCor0, yCor0), (xCor1, yCor1), (xCor2, yCor2) và (xCor3, yCor3) được tính toán là:

$$\begin{aligned}(xCor0, yCor0) &= (xCur + IntX + OffsetX - 1, yCur + IntY + OffsetY - 1); \\(xCor1, yCor1) &= (xCur + IntX + OffsetX - 1, yCur + IntY + OffsetY + H); \\(xCor2, yCor2) &= (xCur + IntX + OffsetX - 1, yCur + IntY + OffsetY); \\(xCor3, yCor3) &= (xCur + IntX + OffsetX + W, yCur + IntY + OffsetY);\end{aligned}$$

Giả sử PredSample[x][y] với $x=0..W-1$, $y=0..H-1$ lưu trữ các mẫu dự báo cho khối phụ. Sau đó, các mẫu đệm được dẫn xuất là:

$$\begin{aligned}\text{PredSample}[x][-1] &= (\text{Ref}(xCor0+x, yCor0) \ll \text{Shift0}) - \text{Rounding}, \text{ với } x=-1..W; \\ \text{PredSample}[x][H] &= (\text{Ref}(xCor1+x, yCor1) \ll \text{Shift0}) - \text{Rounding}, \text{ với } x=-1..W; \\ \text{PredSample}[-1][y] &= (\text{Ref}(xCor2, yCor2+y) \ll \text{Shift0}) - \text{Rounding}, \text{ với } y=0..H-1; \\ \text{PredSample}[W][y] &= (\text{Ref}(xCor3, yCor3+y) \ll \text{Shift0}) - \text{Rounding}, \text{ với } y=0..H-1;\end{aligned}$$

trong đó Rec là ảnh tham chiếu. Rounding là số nguyên, bằng 2^{13} theo phương án thực hiện PROF lấy làm ví dụ. $\text{Shift0} = \text{Max}(2, (14 - \text{BitDepth}))$; PROF thử làm tăng độ chính xác của các gradien, không giống BIO trong VTM-4.0, trong đó các gradien được xuất ra có độ chính xác tương tự làm đầu vào mẫu độ sáng.

Các gradien trong PROF được tính toán dưới đây:

$$\text{Shift1} = \text{Shift0} - 4.$$

$$\text{gradientH}[x][y] = (\text{predSamples}[x+1][y] - \text{predSample}[x-1][y]) \gg \text{Shift1}$$

$$\text{gradientV}[x][y] = (\text{predSample}[x][y+1] - \text{predSample}[x][y-1]) \gg \text{Shift1}$$

Cần lưu ý rằng predSamples[x][y] duy trì độ chính xác sau khi nội suy.

b) Cách thức dẫn xuất Δv cho PROF

Việc dẫn xuất Δv (được ký hiệu là dMvH[posX][posY] và dMvV[posX][posY] với $\text{posX} = 0..W-1$, $\text{posY} = 0..H-1$) có thể được mô tả dưới đây.

Giả sử các kích thước của khối hiện tại là $\text{cbWidth} \times \text{cbHeight}$, số lượng vector chuyển động điểm điều khiển là numCpMv, và các vector chuyển động điểm điều khiển là cpMvLX[cpIdx], với $\text{cpIdx} = 0..\text{numCpMv} - 1$ và X là 0 hoặc 1 biểu diễn hai danh sách tham chiếu.

Các biến log2CbW và log2CbH được dẫn xuất như sau:

$$\text{log2CbW} = \text{Log2}(\text{cbWidth})$$

$$\text{log2CbH} = \text{Log2}(\text{cbHeight})$$

Các biến mvScaleHor, mvScaleVer, dHorX và dVerX được dẫn xuất như sau:

$$\text{mvScaleHor} = \text{cpMvLX}[0][0] \ll 7$$

$$\text{mvScaleVer} = \text{cpMvLX}[0][1] \ll 7$$

$$\text{dHorX} = (\text{cpMvLX}[1][0] - \text{cpMvLX}[0][0]) \ll (7 - \log_2 \text{CbW})$$

$$\text{dVerX} = (\text{cpMvLX}[1][1] - \text{cpMvLX}[0][1]) \ll (7 - \log_2 \text{CbW})$$

Các biến dHorY và dVerY được dẫn xuất như sau:

- Nếu numCpMv bằng 3, điều kiện sau áp dụng:

$$\text{dHorY} = (\text{cpMvLX}[2][0] - \text{cpMvLX}[0][0]) \ll (7 - \log_2 \text{CbH})$$

$$\text{dVerY} = (\text{cpMvLX}[2][1] - \text{cpMvLX}[0][1]) \ll (7 - \log_2 \text{CbH})$$

- Ngược lại (numCpMv bằng 2), điều kiện sau áp dụng:

$$\text{dHorY} = -\text{dVerX}$$

$$\text{dVerY} = \text{dHorX}$$

Biến qHorX, qVerX, qHorY và qVerY được dẫn xuất as:

$$\text{qHorX} = \text{dHorX} \ll 2;$$

$$\text{qVerX} = \text{dVerX} \ll 2;$$

$$\text{qHorY} = \text{dHorY} \ll 2;$$

$$\text{qVerY} = \text{dVerY} \ll 2;$$

dMvH[0][0] và dMvV[0][0] được tính toán as:

$$\text{dMvH}[0][0] = ((\text{dHorX} + \text{dHorY}) \ll 1) - ((\text{qHorX} + \text{qHorY}) \ll 1);$$

$$\text{dMvV}[0][0] = ((\text{dVerX} + \text{dVerY}) \ll 1) - ((\text{qVerX} + \text{qVerY}) \ll 1);$$

dMvH[xPos][0] và dMvV[xPos][0] với xPos từ 1 đến W-1 được dẫn xuất là:

$$\text{dMvH}[\text{xPos}][0] = \text{dMvH}[\text{xPos}-1][0] + \text{qHorX};$$

$$\text{dMvV}[\text{xPos}][0] = \text{dMvV}[\text{xPos}-1][0] + \text{qVerX};$$

Với yPos từ 1 đến H-1, điều kiện sau áp dụng:

$$\text{dMvH}[\text{xPos}][\text{yPos}] = \text{dMvH}[\text{xPos}][\text{yPos}-1] + \text{qHorY} \text{ với } \text{xPos}=0..W-1$$

$$\text{dMvV}[\text{xPos}][\text{yPos}] = \text{dMvV}[\text{xPos}][\text{yPos}-1] + \text{qVerY} \text{ với } \text{xPos}=0..W-1$$

Cuối cùng, dMvH[xPos][yPos] và dMvV[xPos][yPos] với posX = 0..W-1, posY=0..H-1 được dịch phải bằng:

$$\text{dMvH}[\text{xPos}][\text{yPos}] = \text{SatShift}(\text{dMvH}[\text{xPos}][\text{yPos}], 7+2-1);$$

$$\text{dMvV}[\text{xPos}][\text{yPos}] = \text{SatShift}(\text{dMvV}[\text{xPos}][\text{yPos}], 7+2-1);$$

trong đó SatShift(x, n) và Shift(x, n) được định nghĩa là:

$$\text{SatShift}(x, n) = \begin{cases} (x + \text{offset0}) \gg n & \text{if } x \geq 0 \\ -((-x + \text{offset1}) \gg n) & \text{if } x < 0 \end{cases}$$

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n$$

Trong một ví dụ, *offset0* và/hoặc *offset1* được gán bằng $(1 \ll n) \gg 1$.

c) Cách thức dẫn xuất ΔI cho PROF

Đối với vị trí (posX, posY) bên trong khối phụ, $\Delta v(i, j)$ tương ứng của nó được ký hiệu là (dMvH[posX][posY], dMvV[posX][posY]). Các gradient tương ứng được ký hiệu là (gradientH[posX][posY], gradientV[posX][posY]).

Sau đó, $\Delta I(\text{posX}, \text{posY})$ được dẫn xuất như sau.

(dMvH[posX][posY], dMvV[posX][posY]) được rút gọn bằng:

$$\text{dMvH}[\text{posX}][\text{posY}] = \text{Clip3}(-32768, 32767, \text{dMvH}[\text{posX}][\text{posY}]);$$

$$\text{dMvV}[\text{posX}][\text{posY}] = \text{Clip3}(-32768, 32767, \text{dMvV}[\text{posX}][\text{posY}]);$$

$$\Delta I(\text{posX}, \text{posY}) = \text{dMvH}[\text{posX}][\text{posY}] \times \text{gradientH}[\text{posX}][\text{posY}] + \text{dMvV}[\text{posX}][\text{posY}] \times \text{gradientV}[\text{posX}][\text{posY}];$$

$$\Delta I(\text{posX}, \text{posY}) = \text{Shift}(\Delta I(\text{posX}, \text{posY}), 1+1+4);$$

$$\Delta I(\text{posX}, \text{posY}) = \text{Clip3}(-(2^{13}-1), 2^{13}-1, \Delta I(\text{posX}, \text{posY}));$$

d) Cách thức dẫn xuất I' cho PROF

Nếu khối hiện tại không được mã hóa như là dự báo kép hoặc dự báo có trọng số,

$$I'(\text{posX}, \text{posY}) = \text{Shift}((I(\text{posX}, \text{posY}) + \Delta I(\text{posX}, \text{posY})), \text{Shift0}),$$

$$I'(\text{posX}, \text{posY}) = \text{ClipSample}(I'(\text{posX}, \text{posY})),$$

trong đó ClipSample rút gọn giá trị mẫu thành giá trị mẫu đầu ra hợp lệ. Sau đó, $I'(\text{posX}, \text{posY})$ được xuất ra dưới dạng giá trị dự báo ngoài.

Ngược lại (khối hiện tại được mã hóa như là dự báo kép hoặc dự báo có trọng số). $I'(\text{posX}, \text{posY})$ sẽ được lưu trữ và sử dụng để tạo giá trị dự báo ngoài theo các giá trị dự báo khác và/hoặc các giá trị trọng số.

2.6. Tiêu đề lát ví dụ

slice_header() {	Mô tả
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)

if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL nal_unit_type == IDR_N_LP nal_unit_type == CRA_NUT NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	

}	
}	
}	
if(slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1)	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if(sps_ibc_enabled_flag)	
slice_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	

slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

2.7. Tập hợp tham số chuỗi ví dụ

seq parameter set rbsp() {	Mô tả
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gdr_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
min_qp_prime_ts_minus4	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
if(sps_max_sub_layers_minus1 > 0)	
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_ctu_size_minus5	u(2)
log2_min_luma_coding_block_size_minus2	ue(v)

<u>partition_constraints_override_enabled_flag</u>	u(1)
<u>sps_log2_diff_min_qt_min_cb_intra_slice_luma</u>	ue(v)
<u>sps_log2_diff_min_qt_min_cb_inter_slice</u>	ue(v)
<u>sps_max_mtt_hierarchy_depth_inter_slice</u>	ue(v)
<u>sps_max_mtt_hierarchy_depth_intra_slice_luma</u>	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
<u>sps_log2_diff_max_bt_min_qt_intra_slice_luma</u>	ue(v)
<u>sps_log2_diff_max_tt_min_qt_intra_slice_luma</u>	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	
<u>sps_log2_diff_max_bt_min_qt_inter_slice</u>	ue(v)
<u>sps_log2_diff_max_tt_min_qt_inter_slice</u>	ue(v)
}	
if(qtbt dual tree intra flag) {	
<u>sps_log2_diff_min_qt_min_cb_intra_slice_chroma</u>	ue(v)
<u>sps_max_mtt_hierarchy_depth_intra_slice_chroma</u>	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
<u>sps_log2_diff_max_bt_min_qt_intra_slice_chroma</u>	ue(v)
<u>sps_log2_diff_max_tt_min_qt_intra_slice_chroma</u>	ue(v)
}	
}	
<u>sps_max_luma_transform_size_64_flag</u>	u(1)
if(ChromaArrayType != 0) {	
<u>same_qp_table_for_chroma</u>	u(1)
for(i = 0; i < same_qp_table_for_chroma ? 1 : 3; i++) {	
<u>num_points_in_qp_table_minus1[i]</u>	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
<u>delta_qp_in_val_minus1[i][j]</u>	ue(v)
<u>delta_qp_out_val[i][j]</u>	ue(v)
}	
}	
}	
<u>sps_weighted_pred_flag</u>	u(1)
<u>sps_weighted_bipred_flag</u>	u(1)
<u>sps_sao_enabled_flag</u>	u(1)
<u>sps_alf_enabled_flag</u>	u(1)
<u>sps_transform_skip_enabled_flag</u>	u(1)
if(sps_transform_skip_enabled_flag)	
<u>sps_bdpcm_enabled_flag</u>	u(1)
<u>sps_joint_cbr_enabled_flag</u>	u(1)
<u>sps_ref_wraparound_enabled_flag</u>	u(1)
if(sps_ref_wraparound_enabled_flag)	
<u>sps_ref_wraparound_offset_minus1</u>	ue(v)
<u>sps_temporal_mvp_enabled_flag</u>	u(1)
if(sps_temporal_mvp_enabled_flag)	
<u>sps_sbtmvp_enabled_flag</u>	u(1)
<u>sps_amvr_enabled_flag</u>	u(1)
<u>sps_bdof_enabled_flag</u>	u(1)
<u>sps_smvd_enabled_flag</u>	u(1)
<u>sps_dmvr_enabled_flag</u>	u(1)
if(sps_bdof_enabled_flag sps_dmvr_enabled_flag)	
<u>sps_bdof_dmvr_slice_present_flag</u>	u(1)

sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(sps_cclm_enabled_flag && chroma_format_idc == 1)	
sps_cclm_colocated_chroma_flag	u(1)
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
sps_sbt_enabled_flag	u(1)
if(sps_sbt_enabled_flag)	
sps_sbt_max_size_64_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
sps_affine_type_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
}	
if(chroma_format_idc == 3)	
sps_palette_enabled_flag	u(1)
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_triangle_enabled_flag	u(1)
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_scaling_list_enabled_flag	u(1)
hrd_parameters_present_flag	u(1)
if(general_hrd_parameters_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
sub_layer_cpb_parameters_present_flag	u(1)
if(sub_layer_cpb_parameters_present_flag)	
general_hrd_parameters(0, sps_max_sub_layers_minus1)	
Else	

general_hrd_parameters(sps_max_sub_layers_minus1, sps_max_sub_layers_minus1)	
}	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

2.8 Tập hợp tham số ảnh ví dụ

pic_parameter_set_rbsp() {	Mô tả
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
uniform_tile_spacing_flag	u(1)
if(uniform_tile_spacing_flag) {	
tile_cols_width_minus1	ue(v)
tile_rows_height_minus1	ue(v)
} else {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
for(i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
brick_splitting_present_flag	u(1)
if(uniform_tile_spacing_flag && brick_splitting_present_flag)	
num_tiles_in_pic_minus1	ue(v)
for(i = 0; brick_splitting_present_flag && i <= num_tiles_in_pic_minus1 + 1; i++) {	
if(RowHeight[i] > 1)	
brick_split_flag[i]	u(1)
if(brick_split_flag[i]) {	
if(RowHeight[i] > 2)	

uniform_brick_spacing_flag[i]	u(1)
if(uniform_brick_spacing_flag[i])	
brick_height_minus1[i]	ue(v)
else {	
num_brick_rows_minus2[i]	ue(v)
for(j = 0; j <= num_brick_rows_minus2[i]; j++)	
brick_row_height_minus1[i][j]	ue(v)
}	
}	
}	
single_brick_per_slice_flag	u(1)
if(!single_brick_per_slice_flag)	
rect_slice_flag	u(1)
if(rect_slice_flag && !single_brick_per_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
bottom_right_brick_idx_length_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
bottom_right_brick_idx_delta[i]	u(v)
brick_idx_delta_sign_flag[i]	u(1)
}	
}	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	
if(rect_slice_flag) {	
signalled_slice_id_flag	u(1)
if(signalled_slice_id_flag) {	
signalled_slice_id_length_minus1	ue(v)
for(i = 0; i <= num_slices_in_pic_minus1; i++)	
slice_id[i]	u(v)
}	
}	
entropy_coding_sync_enabled_flag	u(1)
if(!single_tile_in_pic_flag entropy_coding_sync_enabled_flag)	
entry_point_offsets_present_flag	u(1)
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
if(sps_transform_skip_enabled_flag)	
log2_transform_skip_max_size_minus2	ue(v)
cu_qp_delta_enabled_flag	u(1)
if(cu_qp_delta_enabled_flag)	
cu_qp_delta_subdiv	ue(v)
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
cu_chroma_qp_offset_enabled_flag	u(1)
if(cu_chroma_qp_offset_enabled_flag) {	

cu_chroma_qp_offset_subdiv	ue(v)
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
joint_cbr_qp_offset_list[i]	se(v)
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
if(constant_slice_header_params_enabled_flag) {	
pps_dep_quant_enabled_idc	u(2)
for(i = 0; i < 2; i++)	
pps_ref_pic_list_sps_idc[i]	u(2)
pps_temporal_mvp_enabled_idc	u(2)
pps_mv_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_five_minus_max_num_subblock_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1	ue(v)
}	
pps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(pps_loop_filter_across_virtual_boundaries_disabled_flag) {	
pps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < pps_num_ver_virtual_boundaries; i++)	
pps_virtual_boundaries_pos_x[i]	u(13)
pps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < pps_num_hor_virtual_boundaries; i++)	
pps_virtual_boundaries_pos_y[i]	u(13)
}	
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

2.9. Tập hợp tham số thích ứng ví dụ

adaptation_parameter_set_rbsp() {	Mô tả
adaptation_parameter_set_id	u(5)

aps_params_type	u(3)
if(aps_params_type == ALF_APS)	
alf_data()	
else if(aps_params_type == LMCS_APS)	
lmcs_data()	
else if(aps_params_type == SCALING_APS)	
scaling_list_data()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

alf_data() {	Mô tả
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	ue(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	u(v)
}	
alf_luma_coeff_signalled_flag	u(1)
if(alf_luma_coeff_signalled_flag) {	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++)	
alf_luma_coeff_flag[sfIdx]	u(1)
}	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++) {	
if(alf_luma_coeff_flag[sfIdx]) {	
for(j = 0; j < 12; j++) {	
alf_luma_coeff_abs[sfIdx][j]	uek(v)
if(alf_luma_coeff_abs[sfIdx][j])	
alf_luma_coeff_sign[sfIdx][j]	u(1)
}	
}	
}	
}	
if(alf_luma_clip_flag) {	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++) {	
if(alf_luma_coeff_flag[sfIdx]) {	
for(j = 0; j < 12; j++)	
alf_luma_clip_idx[sfIdx][j]	u(2)
}	
}	
}	
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_num_alt_filters_minus1	ue(v)
for(altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++) {	
alf_chroma_clip_flag[altIdx]	u(1)

for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs [altIdx][j]	ue(v)
if(alf_chroma_coeff_abs[altIdx][j] > 0)	
alf_chroma_coeff_sign [altIdx][j]	u(1)
}	
if(alf_chroma_clip_flag[altIdx]) {	
for(j = 0; j < 6; j++)	
alf_chroma_clip_idx [altIdx][j]	u(2)
}	
}	
}	

lmcs_data() {	Mô tả
lmcs_min_bin_idx	ue(v)
lmcs_delta_max_bin_idx	ue(v)
lmcs_delta_cw_prec_minus1	ue(v)
for (i = lmcs_min_bin_idx; i <= LmcsMaxBinIdx; i++) {	
lmcs_delta_abs_cw [i]	u(v)
if (lmcs_delta_abs_cw[i] > 0)	
lmcs_delta_sign_cw_flag [i]	u(1)
}	
}	

scaling_list data() {	Mô tả
for(sizeId = 1; sizeId < 7; sizeId++)	
for(matrixId = 0; matrixId < 6; matrixId++) {	
if(! (((sizeId == 1) && (matrixId % 3 == 0)) ((sizeId == 6) && (matrixId % 3 != 0)))) {	
scaling_list_pred_mode_flag [sizeId][matrixId]	u(1)
if(!scaling_list_pred_mode_flag[sizeId][matrixId])	
scaling_list_pred_matrix_id_delta [sizeId][matrixId]	ue(v)
else {	
nextCoef = 8	
coefNum = Min(64, (1 << (sizeId << 1)))	
if(sizeId > 3) {	
scaling_list_dc_coef_minus8 [sizeId - 4][matrixId]	se(v)
nextCoef = scaling_list_dc_coef_minus8[sizeId - 4][matrixId] + 8	
}	
for(i = 0; i < coefNum; i++) {	
x = DiagScanOrder[3][3][i][0]	
y = DiagScanOrder[3][3][i][1]	
if (!(sizeId == 6 && x >= 4 && y >= 4)) {	
scaling_list_delta_coef	se(v)
nextCoef = (nextCoef + scaling_list_delta_coef + 256) % 256	
ScalingList[sizeId][matrixId][i] = nextCoef	
}	
}	
}	
}	
}	
}	
}	

2.10. Tiêu đề ảnh lấy làm ví dụ

Theo một số phương án thực hiện, tiêu đề ảnh được thiết kế để có các thuộc tính sau:

1. ID thời gian và ID lớp của đơn vị NAL tiêu đề ảnh giống như ID thời gian và ID lớp của đơn vị truy nhập lớp mà chứa tiêu đề ảnh.
2. Đơn vị NAL tiêu đề ảnh sẽ đứng trước đơn vị NAL chứa lát thứ nhất của ảnh liên kết. Bước này thiết lập liên kết giữa tiêu đề ảnh và các lát ảnh được liên kết với tiêu đề ảnh không cần có ID tiêu đề ảnh được báo hiệu trong tiêu đề ảnh và được viện dẫn từ tiêu đề lát.

3. Đơn vị NAL tiêu đề ảnh sẽ tuân theo các tập hợp tham số mức ảnh hoặc mức cao hơn chẳng hạn DPS, VPS, SPS, PPS, v.v.. Kết quả là, bước này đòi hỏi các tập hợp tham số đó không được lặp lại /xuất hiện trong ảnh hoặc trong đơn vị truy nhập.

4. Tiêu đề ảnh chứa thông tin về loại ảnh của ảnh liên kết. Loại ảnh có thể được sử dụng để định nghĩa dưới đây (không phải là danh sách vét cạn)

a. Ảnh là ảnh IDR.

b. Ảnh là ảnh CRA.

c. Ảnh là ảnh GDR.

d. Ảnh là ảnh phi IRAP, phi GDR và chỉ chứa các lát I.

e. Ảnh là ảnh phi IRAP, phi GDR và có thể chứa chỉ các lát P và I.

f. Ảnh là ảnh phi IRAP, phi GDR và chứa lát bất kỳ trong các lát B, P, và/hoặc I.

5. Di chuyển báo hiệu của các phần tử cú pháp mức ảnh trong tiêu đề lát đến tiêu đề ảnh.

6. Báo hiệu các phần tử cú pháp không phải mức ảnh trong tiêu đề lát về cơ bản giống như cho tất cả các lát của cùng ảnh trong tiêu đề ảnh. Khi các phần tử cú pháp đó không xuất hiện trong tiêu đề ảnh, chúng có thể được báo hiệu trong tiêu đề lát.

Theo một số phương án thực hiện, khái niệm tiêu đề ảnh bắt buộc được sử dụng cần được truyền một lần trên ảnh dưới dạng đơn vị NAL VCL thứ nhất của ảnh. Cũng đề xuất di chuyển các phần tử cú pháp hiện tại trong tiêu đề lát đến tiêu đề ảnh này. Các phần tử cú pháp mà về mặt chức năng chỉ cần được truyền một lần trên ảnh có thể được di chuyển đến tiêu đề ảnh thay vì được truyền nhiều lần cho ảnh cụ thể, chẳng hạn, các phần tử cú pháp trong tiêu đề lát được truyền một lần trên lát. Việc di chuyển các phần tử cú pháp tiêu đề lát bị giới hạn là giống như trong ảnh

Các phần tử cú pháp đã bị giới hạn là tương tự trong tất cả các lát của ảnh. Khẳng định rằng việc di chuyển các trường này đến tiêu đề ảnh do vậy chúng được báo hiệu chỉ một lần trên ảnh thay vì một lần trên lát tránh truyền bit dư không cần thiết mà không có bất kỳ thay đổi nào đến chức năng của các phần tử cú pháp này.

1. Theo một số phương án thực hiện, có ràng buộc ngữ nghĩa sau:

Nếu xuất hiện, giá trị của mỗi phần tử trong các phần tử cú pháp tiêu đề lát slice_pic_parameter_set_id, non_reference_picture_flag, colour_plane_id, slice_pic_order_cnt_lsb, recovery_poc_cnt, no_output_of_prior_pics_flag, pic_output_flag, và slice_temporal_mvp_enabled_flag sẽ là giống nhau trong tất cả các tiêu đề lát của ảnh được mã hóa. Do vậy, mỗi phần tử trong các phần tử cú pháp này có thể được di chuyển đến tiêu đề ảnh để tránh các bit dư không cần thiết.

recovery_poc_cnt và no_output_of_prior_pics_flag không được di chuyển đến tiêu đề ảnh trong phần này. Sự xuất hiện của nó trong tiêu đề lát phụ thuộc vào việc kiểm tra có điều kiện của tiêu đề lát nal_unit_type, do vậy đề xuất nghiên cứu liệu có mong muốn di chuyển các phần tử cú pháp này đến tiêu đề ảnh.

2. Theo một số phương án thực hiện, có các ràng buộc ngôn ngữ sau:

Nếu xuất hiện, giá trị slice_lmcs_aps_id sẽ giống nhau cho tất cả các lát của ảnh.

Nếu xuất hiện, giá trị slice_scaling_list_aps_id sẽ giống nhau cho tất cả các lát của ảnh. Do vậy, mỗi phần tử trong các phần tử cú pháp này có thể được di chuyển đến tiêu đề ảnh để tránh các bit dư không cần thiết.

Theo một số phương án thực hiện, các phần tử cú pháp hiện tại không bắt buộc giống nhau trong tất cả các lát của ảnh. Đề xuất đánh giá sử dụng có dự báo trước các phần tử cú pháp này để xác định có thể được di chuyển vào tiêu đề ảnh để đơn giản hóa toàn bộ thiết kế VVC như được bảo hộ có ảnh hưởng phức tạp đối với việc xử lý số lượng lớn của các phần tử cú pháp trong mỗi tiêu đề lát.

1. Các phần tử cú pháp sau được đề xuất di chuyển đến tiêu đề ảnh. Hiện tại không có các ràng buộc lên việc phải có các giá trị khác nhau đối với các lát khác nhau nhưng yêu cầu rằng không có/có rất ít lợi ích và tổn hao mã khi truyền chúng trong mỗi tiêu đề lát do việc sử dụng được dự báo sẽ thay đổi ở mức ảnh:

- a. six_minus_max_num_merge_cand
- b. five_minus_max_num_subblock_merge_cand
- c. slice_fpel_mmvd_enabled_flag
- d. slice_disable_bdof_dmvr_flag
- e. max_num_merge_cand_minus_max_num_triangle_cand
- f. slice_six_minus_max_num_ibc_merge_cand

2. Các phần tử cú pháp sau được đề xuất di chuyển đến tiêu đề ảnh. Hiện tại không có các ràng buộc lên việc phải có các giá trị khác nhau đối với các lát khác nhau nhưng yêu cầu rằng không có/có rất ít lợi ích và tổn hao mã khi truyền chúng trong mỗi tiêu đề lát do việc sử dụng được dự báo sẽ thay đổi ở mức ảnh:

- a. `partition_constraints_override_flag`
- b. `slice_log2_diff_min_qt_min_cb_luma`
- c. `slice_max_mtt_hierarchy_depth_luma`
- d. `slice_log2_diff_max_bt_min_qt_luma`
- e. `slice_log2_diff_max_tt_min_qt_luma`
- f. `slice_log2_diff_min_qt_min_cb_chroma`
- g. `slice_max_mtt_hierarchy_depth_chroma`
- h. `slice_log2_diff_max_bt_min_qt_chroma`
- i. `slice_log2_diff_max_tt_min_qt_chroma`

Kiểm tra có điều kiện “`slice_type == I`” được liên kết với một số phần tử cú pháp này đã được loại bỏ với việc dịch chuyển đến tiêu đề ảnh.

3. Các phần tử cú pháp sau được đề xuất di chuyển đến tiêu đề ảnh. Hiện tại không có các ràng buộc lên việc phải có các giá trị khác nhau đối với các lát khác nhau nhưng yêu cầu rằng không có/có rất ít lợi ích và tổn hao mã khi truyền chúng trong mỗi tiêu đề lát do việc sử dụng được dự báo sẽ thay đổi ở mức ảnh:

- a. `mvd_l1_zero_flag`

Kiểm tra có điều kiện “`slice_type == B`” được liên kết với một số phần tử cú pháp này đã được loại bỏ với việc dịch chuyển đến tiêu đề ảnh.

4. Các phần tử cú pháp sau được đề xuất di chuyển đến tiêu đề ảnh. Hiện tại không có các ràng buộc lên việc phải có các giá trị khác nhau đối với các lát khác nhau nhưng yêu cầu rằng không có/có rất ít lợi ích và tổn hao mã khi truyền chúng trong mỗi tiêu đề lát do việc sử dụng được dự báo sẽ thay đổi ở mức ảnh:

- a. `dep_quant_enabled_flag`
- b. `sign_data_hiding_enabled_flag`

2.10.1 Các bảng cú pháp lấy làm ví dụ

7.3.2.8 Cú pháp RBSP tiêu đề ảnh

picture_header_rbsp() {	Mô tả
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i < NumSubPics; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	

partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbt dual tree intra flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
pic_cu_qp_delta_subdiv_intra_slice	ue(v)
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(cu_chroma_qp_offset_enabled_flag) {	
pic_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
pic_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
if(sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
pic_joint_cbr_sign_flag	u(1)

if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
if(!pps_dep_quant_enabled_flag)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	
pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)

}	
if(picture header extension present flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits()	
}	

2.11. DMVR lấy làm ví dụ

Kỹ thuật tinh lọc vectơ chuyển động phía bộ giải mã (DMVR) sử dụng so khớp hai bên (BM) để dẫn xuất thông tin chuyển động của CU hiện tại bằng cách tìm ra so khớp gần nhất giữa hai khối dọc theo quỹ đạo chuyển động của CU hiện tại trong hai ảnh tham chiếu khác nhau. Phương pháp BM tính toán méo dạng giữa hai khối ứng viên trong danh sách ảnh tham chiếu L0 và danh sách L1. Như được minh họa trên Fig.1, SAD giữa các khối màu đỏ dựa trên mỗi ứng viên MV quanh MV ban đầu được tính toán. Ứng viên MV có SAD nhỏ nhất trở thành MV được tinh lọc và được sử dụng để tạo tín hiệu được dự báo kép. Hàm chi phí được sử dụng trong quá trình so khớp là SAD được lấy mẫu phụ theo hàng (sum of absolute difference). Ví dụ trên Fig.7.

Trong VTM5.0, DMVR được sử dụng để tinh lọc các vectơ chuyển động (MV) ở bộ giải mã đối với đơn vị mã (CU) khi CU được mã hóa bằng chế độ hợp nhất/bỏ qua thông thường và dự báo kép, một ảnh tham chiếu đứng trước ảnh hiện tại và ảnh tham chiếu còn lại đứng sau ảnh hiện tại theo thứ tự hiển thị, khoảng cách thời gian giữa ảnh hiện tại và một ảnh tham chiếu là giữa ảnh hiện tại và ảnh tham chiếu còn lại, và dự báo kép có các trọng số CU (BCW) lựa chọn các trọng số bằng nhau. Khi DMVR được áp dụng, một khối mã độ sáng (CB) được phân chia thành vài khối phụ được xử lý độc lập có kích thước $\min(\text{cbWidth}, 16) \times \min(\text{cbHeight}, 16)$. DMVR tinh lọc các MV của mỗi khối phụ bằng cách giảm tối đa SAD giữa các mẫu dự báo L0 và L1 10 bit được lấy mẫu phụ $\frac{1}{2}$ được tạo bằng cách nội suy song tuyến tính. Đối với mỗi khối phụ, tìm kiếm ΔMV số nguyên quanh các MV ban đầu (tức là, các MV của ứng viên hợp nhất/bỏ qua thông thường được chọn) được thực hiện trước nhờ sử dụng SAD, và sau đó dẫn xuất ΔMV theo phân số được thực hiện để thu được các MV cuối cùng.

BDOF tính lọc các mẫu dự báo độ sáng cho CU khi CU được mã hóa với dự báo kép, một ảnh tham chiếu đứng trước ảnh hiện tại và ảnh tham chiếu còn lại đứng sau ảnh hiện tại theo thứ tự hiển thị, và BCW chọn các trọng số bằng nhau. Nội suy 8 bậc được sử dụng để tạo các mẫu dự báo L0 và L1 ban đầu theo các MV đầu vào (chẳng hạn, các MV cuối cùng của DMVR nếu kích hoạt DMVR). Tiếp theo, quá trình kết thúc sớm hai mức được thực hiện. Quá trình kết thúc sớm thứ nhất là ở mức khối phụ, và quá trình kết thúc sớm thứ hai là ở mức khối 4×4 và được kiểm tra khi quá trình kết thúc sớm thứ nhất không xuất hiện. Ở mỗi mức, SAD giữa các mẫu dự báo L0 và L1 14 bit được lấy mẫu toàn phần ở mỗi khối phụ/khối 4x4 được tính toán trước. Nếu SAD nhỏ hơn một ngưỡng, BDOF không được áp dụng cho khối phụ/khối 4x4. Ngược lại, các tham số BDOF được dẫn xuất và được sử dụng để tạo các bộ dự báo mẫu độ sáng cuối cùng đối với mỗi khối 4x4. Trong BDOF, kích thước khối phụ giống như kích thước trong DMVR, tức là, $\min(\text{cbWidth}, 16) \times \min(\text{cbHeight}, 16)$.

Khi CU được mã hóa với chế độ hợp nhất/bỏ qua thông thường, một ảnh tham chiếu đứng trước ảnh hiện tại và ảnh tham chiếu khác đứng sau ảnh hiện tại theo thứ tự hiển thị, khoảng cách thời gian giữa ảnh hiện tại và một ảnh tham chiếu giữa ảnh hiện tại và ảnh tham chiếu còn lại, và BCW lựa chọn các trọng số bằng nhau, DMVR và BDOF đều được áp dụng. Luồng ghép tầng các quá trình DMVR và BDOF được thể hiện trên Fig.8.

Fig.8 thể hiện luồng ghép tầng các quá trình DMVR và BDOF trong VTM5.0. Các hoạt động DMVR SAD và các hoạt động BDOF SAD là khác nhau và không được chia sẻ.

Để giảm độ trễ và các hoạt động trong đường tới hạn này, khi DMVR và BDOF đều được áp dụng, phác thảo VVC mới nhất đã được chỉnh sửa để sử dụng lại khối phụ SAD được tính toán trong DMVR cho quá trình kết thúc sớm khối phụ trong BDOF.

Phép tính toán SAD được định nghĩa như sau:

$$\text{sad} = \sum_{x=0}^{n_{\text{SbW}}-1} \sum_{y=0}^{n_{\text{SbH}}/2-1} \text{abs}(\text{pL0}[x+2+dX][2*y+2+dY] - \text{pL1}[x+2-dX][2*y+2-dY])$$

trong đó hai biến $nSbW$ và $nSbH$ xác định chiều rộng và chiều cao của khối phụ hiện tại, hai mảng $(nSbW + 4) \times (nSbH + 4)$ $pL0$ và $pL1$ lần lượt chứa các mẫu được dự báo cho $L0$ và $L1$, và độ lệch mẫu số nguyên (dX , dY) trong danh sách dự báo $L0$.

Để giảm sai lỗi không chắc chắn của phép tính lọc DMVR, đề xuất ưu tiên MV ban đầu trong quá trình DMVR. SAD giữa các khối tham chiếu được đề cập đến bởi ứng viên MV ban đầu (hoặc gốc) được giảm đi $1/4$ giá trị SAD. Tức là, khi cả dX và dY trong phương trình nêu trên bằng 0, giá trị của sad được chỉnh sửa như sau:

$$sad = sad - (sad >> 2)$$

Khi giá trị SAD nhỏ hơn ngưỡng ($2 \times \text{chiều rộng khối phụ} \times \text{chiều cao khối phụ}$), không cần thực hiện BDOF nữa.

3. Các nhược điểm của các triển khai hiện tại

DMVR và BIO không bao gồm tín hiệu gốc trong khi tính lọc các vector chuyển động, có thể dẫn đến các khối mã có thông tin chuyển động không chính xác. DMVR và BIO đôi lúc cũng sử dụng các vector chuyển động phân số sau các tính lọc chuyển động trong khi các video màn hình thường có các vector chuyển động số nguyên, mà khiến thông tin chuyển động hiện tại thiếu chính xác hơn và khiến hiệu năng mã kém hơn.

Khi RPR được áp dụng theo VVC, RPR (ARC) có thể có các vấn đề sau:

1. Với RPR, các bộ lọc nội suy có thể khác nhau đối với các mẫu liền kề ở khối, vốn không mong muốn khi triển khai SIMD (Một lệnh nhiều dữ liệu).

2. Vùng giới hạn không xem xét RPR.

3. Lưu ý rằng “Các tham số độ lệch cửa sổ cắt tương hợp chỉ được áp dụng ở đầu ra. Tất cả các quá trình giải mã nội bộ được áp dụng cho kích thước ảnh không được cắt.” Tuy nhiên, các tham số đó có thể được sử dụng trong quá trình giải mã khi RPR được áp dụng.

4. Khi dẫn xuất vị trí mẫu tham chiếu, RPR chỉ xem xét tỷ lệ giữa hai cửa sổ tương hợp. Nhưng hiệu số độ lệch trên bên trái giữa hai cửa sổ tương hợp cũng nên được xem xét.

5. Tỷ lệ giữa chiều rộng/chiều cao của ảnh tham chiếu và tỷ lệ của ảnh hiện tại bị ràng buộc trong VVC. Tuy nhiên, tỷ lệ giữa chiều rộng/chiều cao của cửa sổ tương hợp của ảnh tham chiếu và tỷ lệ của cửa sổ tương hợp của ảnh hiện tại không bị ràng buộc.

6. Không phải tất cả các phần tử cú pháp được xử lý thích hợp trong tiêu đề ảnh.

7. Trong VVC hiện tại, đối với chế độ dự báo TPM, GEO, các trọng số kết hợp sắc độ được dẫn xuất bất kể loại vị trí mẫu sắc độ của chuỗi video. Chẳng hạn, trong TPM/GEO, nếu các trọng số sắc độ được dẫn xuất từ các trọng số độ sáng, các trọng số độ sáng có thể cần được giảm kích thước mẫu để so khớp lấy mẫu tín hiệu sắc độ. Việc giảm kích thước mẫu sắc độ thường được áp dụng giả sử loại vị trí mẫu sắc độ 0, được sử dụng rộng rãi trong phần chứa ITU-R BT.601 hoặc ITU-R BT.709. Tuy nhiên, nếu loại vị trí mẫu sắc độ khác nhau được sử dụng, có thể dẫn đến căn chỉnh sai giữa các mẫu sắc độ và mẫu độ sáng bị giảm kích thước mẫu, có thể giảm hiệu năng mã.

8. Lưu ý rằng phép tính SAD/ngưỡng SAD không xem xét tác động chiều sâu bit. Do vậy, đối với chiều sâu bit cao hơn (chẳng hạn, các chuỗi đầu vào 14 hoặc 16 bit), ngưỡng kết thúc sớm có thể quá nhỏ.

9. Đối với trường hợp phi RPR, AMVR với MV có độ chính xác $\frac{1}{2}$ pel (tức là, bộ lọc nội suy tùy chọn/bộ lọc nội suy có thể thay đổi) được áp dụng với bộ lọc bù chuyển động 6 bậc nhưng 8 bậc được áp dụng cho các trường hợp khác (chẳng hạn, $\frac{1}{16}$ pel). Tuy nhiên, đối với trường hợp RPR, bộ lọc nội suy tương tự được áp dụng cho tất cả các trường hợp mà không xem xét độ chính xác MV/mvd. Do vậy, trường hợp báo hiệu $\frac{1}{2}$ pel (bộ lọc nội suy tùy chọn/bộ lọc nội suy có thể thay đổi) đang lãng phí bit.

10. Quyết định tách cây phân vùng được phép hoặc không phụ thuộc vào độ phân giải ảnh được mã hóa thay vì độ phân giải ảnh đầu ra.

11. SMVD/MMVD được áp dụng không xem xét trường hợp RPR. Các phương pháp này dựa trên giả thiết rằng MVD đối xứng được áp dụng cho hai ảnh tham chiếu. Tuy nhiên, khi độ phân giải ảnh đầu ra là khác nhau, giả thiết này không đúng.

12. Ứng viên hợp nhất theo cặp được tạo bằng cách tính trung bình hai MV từ hai ứng viên hợp nhất trong cùng danh sách ảnh tham chiếu. Tuy nhiên, khi hai ảnh tham chiếu được liên kết với hai ứng viên hợp nhất có độ phân giải khác nhau, phép tính trung bình không hợp lý.

4. Các kỹ thuật và các phương án thực hiện lấy làm ví dụ

Các phương án thực hiện được mô tả chi tiết dưới đây cần được xem xét làm các ví dụ để giải thích các khái niệm chung. Các phương án thực hiện này không được hiểu theo nghĩa hẹp. Ngoài ra, các phương án thực hiện này có thể được kết hợp theo cách thức bất kỳ.

Các phương pháp được mô tả dưới đây cũng có thể được áp dụng cho các công nghệ dẫn xuất thông tin chuyển động của bộ giải mã khác bên cạnh DMVR và BIO được nêu dưới đây.

Vector chuyển động được ký hiệu là (mv_x, mv_y) trong đó mv_x là thành phần ngang và mv_y là thành phần dọc.

Theo sáng chế, độ phân giải (hoặc kích thước, hoặc chiều rộng/chiều cao, hoặc kích thước) của ảnh có thể viện dẫn đến độ phân giải (hoặc kích thước, hoặc chiều rộng/chiều cao, hoặc kích thước) của ảnh được mã hóa/giải mã, hoặc có thể viện dẫn đến độ phân giải (hoặc kích thước, hoặc chiều rộng/chiều cao, hoặc kích thước) của cửa sổ tương hợp trong ảnh được mã hóa/giải mã. Trong một ví dụ, độ phân giải (hoặc kích thước, hoặc chiều rộng/chiều cao, hoặc kích thước) của ảnh có thể đề cập đến việc các tham số liên quan đến quá trình RPR (lấy mẫu lại ảnh tham chiếu), chẳng hạn, cửa sổ chia tỷ lệ/cửa sổ độ lệch pha. Trong một ví dụ, độ phân giải (hoặc kích thước, hoặc chiều rộng/chiều cao, hoặc kích thước) của ảnh liên quan đến độ phân giải được liên kết với ảnh đầu ra.

Bù chuyển động trong RPR

1. Khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại, các giá trị được dự báo cho nhóm mẫu (ít nhất hai mẫu) của khối hiện tại có thể được tạo với bộ lọc nội suy ngang và/hoặc dọc tương tự.

- a. Trong một ví dụ, nhóm có thể bao gồm tất cả các mẫu ở vùng của khối.

- i. Chẳng hạn, khối có thể được phân chia thành S hình chữ nhật $M \times N$ không được chồng lấn với nhau. Mỗi hình chữ nhật $M \times N$ là nhóm. Trong ví dụ như được thể hiện trên Fig.2, khối 16×16 có thể được phân chia thành 16 hình chữ nhật 4×4 , mà mỗi hình chữ nhật là nhóm.
 - ii. Chẳng hạn, hàng có N mẫu là nhóm. N là số nguyên không lớn hơn chiều rộng khối. Trong một ví dụ, N bằng 4 hoặc 8 hoặc chiều rộng khối.
 - iii. Chẳng hạn, cột có N mẫu là nhóm. N là số nguyên không lớn hơn chiều cao khối. Trong một ví dụ, N bằng 4 hoặc 8 hoặc chiều cao khối.
 - iv. M và/hoặc N có thể được định trước hoặc được dẫn xuất ngay lập tức, chẳng hạn dựa trên kích thước khối/thông tin được mã hóa hoặc được báo hiệu.
- b. Trong một ví dụ, các mẫu ở nhóm có thể có cùng MV (được ký hiệu là MV được chia sẻ).
 - c. Trong một ví dụ, các mẫu ở nhóm có thể có các MV với thành phần ngang tương tự (được ký hiệu là thành phần ngang chia sẻ).
 - d. Trong một ví dụ, các mẫu ở nhóm có thể có các MV với thành phần dọc tương tự (được ký hiệu là thành phần dọc chia sẻ).
 - e. Trong một ví dụ, các mẫu ở nhóm có thể có các MV với phần phân số tương tự của thành phần ngang (được ký hiệu là thành phần ngang phân số chia sẻ).
 - i. Chẳng hạn, giả sử MV đối với mẫu thứ nhất là $(MV1x, MV1y)$ và MV đối với mẫu thứ hai là $(MV2x, MV2y)$, cần thỏa mãn rằng $MV1x \& (2^M - 1)$ bằng $MV2x \& (2^M - 1)$, trong đó M ký hiệu độ chính xác MV. Chẳng hạn, $M=4$.
 - f. Trong một ví dụ, các mẫu ở nhóm có thể có các MV với phần phân số tương tự của thành phần dọc (được ký hiệu là thành phần dọc phân số chia sẻ).

- i. Chẳng hạn, giả sử MV đối với mẫu thứ nhất là $(MV1x, MV1y)$ và MV đối với mẫu thứ hai là $(MV2x, MV2y)$, cần thỏa mãn rằng $MV1y \& (2^M-1)$ bằng $MV2y \& (2^M-1)$, trong đó M ký hiệu độ chính xác MV. Chẳng hạn, $M=4$.
- g. Trong một ví dụ, đối với mẫu trong nhóm cần được dự báo, vector chuyển động, được ký hiệu là MV_b , có thể được dẫn xuất trước theo các độ phân giải của ảnh hiện tại và ảnh tham chiếu (chẳng hạn $(refx_L, refy_L)$ được dẫn xuất trong 8.5.6.3.1 trong JVET-O2001-v14). Sau đó, MV_b có thể còn được chỉnh sửa (chẳng hạn, được làm tròn/rút gọn/cắt) thành MV' để thỏa mãn các yêu cầu chẳng hạn các đề mục nêu trên, và MV' sẽ được sử dụng để dẫn xuất mẫu dự báo cho mẫu.
 - i. Trong một ví dụ, MV' có phần số nguyên tương tự như MV_b , và phần phân số của MV' được gán bằng thành phần dọc và/hoặc ngang phân số được chia sẻ.
 - ii. Trong một ví dụ, MV' được gán bằng bằng vector có thành phần dọc và/hoặc ngang phân số được chia sẻ, và gần nhất với MV_b .
- h. Vector chuyển động được chia sẻ (và/hoặc thành phần ngang chia sẻ và/hoặc thành phần dọc chia sẻ và/hoặc thành phần dọc phân số chia sẻ và/hoặc thành phần dọc phân số chia sẻ) có thể được gán bằng vector chuyển động (và/hoặc thành phần ngang và/hoặc thành phần dọc và/hoặc thành phần dọc phân số và/hoặc thành phần dọc phân số) của mẫu cụ thể trong nhóm.
 - i. Chẳng hạn, mẫu cụ thể có thể ở góc của nhóm hình chữ nhật, chẳng hạn “A”, “B”, “C” và “D” được thể hiện trên Fig.3A.
 - ii. Chẳng hạn, mẫu cụ thể có thể ở tâm của nhóm hình chữ nhật, chẳng hạn “E”, “F”, “G” và “H” được thể hiện trên Fig.3A.
 - iii. Chẳng hạn, mẫu cụ thể có thể ở cuối của nhóm theo hàng hoặc nhóm theo cột, chẳng hạn “A” và “D” được thể hiện trên Fig.3B và Fig.3C.

- iv. Chẳng hạn, mẫu cụ thể có thể ở giữa của nhóm theo hàng hoặc nhóm theo cột, chẳng hạn “B” và “C” được thể hiện trên Fig.3B và Fig.3C.
- v. Trong một ví dụ, vectơ chuyển động của mẫu cụ thể có thể là MV_b được nêu trong đề mục g.
- i. Vectơ chuyển động được chia sẻ (và/hoặc thành phần ngang chia sẻ và/hoặc thành phần dọc chia sẻ và/hoặc thành phần dọc phân số chia sẻ và/hoặc thành phần dọc phân số chia sẻ) có thể được gán bằng vectơ chuyển động (và/hoặc thành phần ngang và/hoặc thành phần dọc và/hoặc thành phần dọc phân số và/hoặc thành phần dọc phân số) của mẫu ảo được đặt ở vị trí khác so với tất cả các mẫu ở nhóm này.
 - i. Trong một ví dụ, mẫu ảo không trong nhóm, nhưng nó đặt trong vùng bao phủ tất cả các mẫu ở nhóm.
 - 1) Theo cách khác, mẫu ảo được đặt ngoài vùng bao phủ tất cả các mẫu ở nhóm, chẳng hạn, cạnh vị trí dưới bên phải của vùng.
 - ii. Trong một ví dụ, MV của mẫu ảo được dẫn xuất theo cách tương tự như mẫu thực nhưng với các vị trí khác nhau.
 - iii. “V” trên các hình vẽ từ Fig.3A đến Fig.3C thể hiện ba ví dụ về các mẫu ảo.
- j. MV được chia sẻ (và/hoặc thành phần ngang chia sẻ và/hoặc thành phần dọc chia sẻ và/hoặc thành phần dọc phân số chia sẻ và/hoặc thành phần dọc phân số chia sẻ) có thể được gán bằng hàm của các MV (và/hoặc các thành phần ngang và/hoặc các thành phần dọc và/hoặc các thành phần dọc phân số và/hoặc các thành phần dọc phân số) của nhiều mẫu và/hoặc các mẫu ảo.
 - i. Chẳng hạn, MV được chia sẻ (và/hoặc thành phần ngang chia sẻ và/hoặc thành phần dọc chia sẻ và/hoặc thành phần dọc phân số chia sẻ và/hoặc thành phần dọc phân số chia sẻ) có thể được gán bằng trung bình của các MV (và/hoặc các thành phần ngang và/hoặc các thành phần dọc và/hoặc các thành phần dọc

phân số và/hoặc các thành phần dọc phân số) của tất cả hoặc một phần của các mẫu ở nhóm, hoặc của mẫu “E”, “F”, “G”, “H” trên Fig.3A, hoặc của mẫu “E”, “H” trên Fig.3A, hoặc của mẫu “A”, “B”, “C”, “D” trên Fig.3A, hoặc của mẫu “A”, “D” trên Fig.3A, hoặc của mẫu “B”, “C” trên Fig.3B, hoặc của mẫu “A”, “D” trên Fig.3B, hoặc của mẫu “B”, “C” trên Fig.3C, hoặc của mẫu “A”, “D” trên Fig.3C,

2. Đề xuất rằng chỉ các MV số nguyên được phép thực hiện quá trình bù chuyển động để dẫn xuất khối dự báo của khối hiện tại khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
 - a. Trong một ví dụ, vector chuyển động được giải mã cho các mẫu được dự báo được làm tròn xuống các MV số nguyên trước khi được sử dụng.
 - b. Trong một ví dụ, vector chuyển động được giải mã cho các mẫu được dự báo được làm tròn xuống MV số nguyên gần nhất với vector chuyển động được giải mã.
 - c. Trong một ví dụ, vector chuyển động được giải mã cho các mẫu được dự báo được làm tròn xuống MV số nguyên gần nhất với vector chuyển động được giải mã theo chiều ngang.
 - d. Trong một ví dụ, vector chuyển động được giải mã cho các mẫu được dự báo được làm tròn xuống MV số nguyên gần nhất với vector chuyển động được giải mã theo chiều dọc.
3. Các vector chuyển động được sử dụng trong quá trình bù chuyển động cho các mẫu ở khối hiện tại (chẳng hạn, MV được chia sẻ/ thành phần ngang hoặc dọc hoặc phân số được chia sẻ/MV’ được nêu trong các đề mục trên) có thể được lưu trữ trong bộ đệm ảnh được giải mã và được sử dụng để dự báo vector chuyển động của các khối tiếp theo trong các ảnh hiện tại/khác nhau.
 - a. Theo cách khác, các vector chuyển động được sử dụng trong quá trình bù chuyển động cho các mẫu ở khối hiện tại (chẳng hạn, MV được

chia sẻ/ thành phần ngang hoặc dọc hoặc phân số được chia sẻ/MV' được nêu trong các đề mục trên) có thể không được phép được sử dụng để dự báo vector chuyển động của các khối tiếp theo trong các ảnh hiện tại/khác nhau.

- i. Trong một ví dụ, các vector chuyển động được giải mã (chẳng hạn, MV_b trong các đề mục nêu trên) có thể được sử dụng để dự báo vector chuyển động của các khối tiếp theo trong các ảnh hiện tại/khác nhau.
 - b. Trong một ví dụ, các vector chuyển động được sử dụng trong quá trình bù chuyển động cho các mẫu ở khối hiện tại có thể được sử dụng trong quá trình lọc (chẳng hạn, bộ lọc tách khối/SAO/ALF).
 - i. Theo cách khác, các vector chuyển động được giải mã (chẳng hạn, MV_b trong các đề mục nêu trên) có thể được sử dụng trong quá trình lọc.
 - c. Trong một ví dụ, MV này có thể được dẫn xuất ở mức khối phụ và có thể được lưu trữ đối với mỗi khối phụ.
4. Đề xuất rằng các bộ lọc nội suy được sử dụng trong quá trình bù chuyển động để dẫn xuất khối dự báo của khối hiện tại có thể được lựa chọn tùy thuộc liệu độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc liệu chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
- a. Trong một ví dụ, các bộ lọc nội suy có ít bậc hơn có thể được áp dụng khi điều kiện A được thỏa mãn, trong đó điều kiện A phụ thuộc vào các kích thước của ảnh hiện tại và/hoặc ảnh tham chiếu.
 - i. Trong một ví dụ, điều kiện A là độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại.
 - ii. Trong một ví dụ, điều kiện A là chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
 - iii. Trong một ví dụ, điều kiện A là $W1 > a*W2$ và/hoặc $H1 > b*H2$, trong đó ($W1$, $H1$) là chiều rộng và chiều cao của ảnh

tham chiếu và (W2, H2) là chiều rộng và chiều cao của ảnh hiện tại, a và b là hai yếu tố, chẳng hạn $a = b = 1,5$.

- iv. Trong một ví dụ, điều kiện A có thể cũng phụ thuộc vào liệu dự báo kép được sử dụng.
- v. Trong một ví dụ, các bộ lọc 1 bậc được áp dụng. Nói cách khác, pixel số nguyên mà không lọc được xuất ra dưới dạng kết quả nội suy.
- vi. Trong một ví dụ, các bộ lọc song tuyến tính được áp dụng khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại.
- vii. Trong một ví dụ, các bộ lọc 4 bậc hoặc các bộ lọc 6 bậc được áp dụng khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
 - 1) Các bộ lọc 6 bậc có thể cũng được sử dụng để bù chuyển động afin.
 - 2) Các bộ lọc 4 bậc có thể cũng được sử dụng để nội suy cho các mẫu sắc độ.
- b. Trong một ví dụ, các mẫu đệm được sử dụng để thực hiện nội suy khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
- c. Liệu và/hoặc cách thức áp dụng các phương pháp được nêu trong mục 4 có thể phụ thuộc vào các thành phần màu.
 - i. Chẳng hạn, các phương pháp chỉ được áp dụng trên thành phần độ sáng.
- d. Liệu và/hoặc cách thức áp dụng các phương pháp được nêu trong mục 4 có thể phụ thuộc vào hướng lọc nội suy.
 - i. Chẳng hạn, các phương pháp chỉ được áp dụng trên lọc ngang.
 - ii. Chẳng hạn, các phương pháp chỉ được áp dụng trên lọc dọc.

5. Đề xuất rằng quá trình tạo khối dự báo hai giai đoạn được áp dụng khi độ phân giải của ảnh tham chiếu là khác nhau với ảnh hiện tại, hoặc khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu lớn hơn của ảnh hiện tại.
 - a. Trong giai đoạn thứ nhất, khối tham chiếu ảo được tạo bởi tăng kích thước mẫu hoặc giảm kích thước mẫu vùng trong ảnh tham chiếu tùy thuộc chiều rộng và/hoặc chiều cao của ảnh hiện tại và ảnh tham chiếu.
 - b. Trong giai đoạn thứ hai, các mẫu dự báo được tạo từ khối tham chiếu ảo bằng cách áp dụng lọc nội suy, độc lập với chiều rộng và/hoặc chiều cao của ảnh hiện tại và ảnh tham chiếu.
6. Đề xuất rằng việc tính toán tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (x_{SbIntL} , y_{SbIntL}) như theo một số phương án thực hiện có thể được dẫn xuất tùy thuộc chiều rộng và/hoặc chiều cao của ảnh hiện tại và ảnh tham chiếu.
 - a. Trong một ví dụ, các vị trí độ sáng theo các đơn vị mẫu toàn phần được chỉnh sửa như là:

$$x_{Inti} = \text{Clip3}(x_{SbIntL} - Dx, x_{SbIntL} + sbWidth + Ux, x_{Inti}),$$

$$y_{Inti} = \text{Clip3}(y_{SbIntL} - Dy, y_{SbIntL} + sbHeight + Uy, y_{Inti}),$$
 trong đó Dx và/hoặc Dy và/hoặc Ux và/hoặc Uy có thể phụ thuộc vào chiều rộng và/hoặc chiều cao của ảnh hiện tại và ảnh tham chiếu.
 - b. Trong một ví dụ, các vị trí sắc độ theo các đơn vị mẫu toàn phần được chỉnh sửa như là:

$$x_{Inti} = \text{Clip3}(x_{SbIntc} - Dx, x_{SbIntc} + sbWidth + Ux, x_{Inti})$$

$$y_{Inti} = \text{Clip3}(y_{SbIntc} - Dy, y_{SbIntc} + sbHeight + Uy, y_{Inti})$$
 trong đó Dx và/hoặc Dy và/hoặc Ux và/hoặc Uy có thể phụ thuộc vào chiều rộng và/hoặc chiều cao của ảnh hiện tại và ảnh tham chiếu.
7. Thay vì lưu trữ/sử dụng các vector chuyển động cho khối dựa trên cùng độ phân giải ảnh tham chiếu làm ảnh hiện tại, đề xuất sử dụng các vector chuyển động thực với độ lệch độ phân giải được xem xét.

- a. Theo cách khác, ngoài ra, khi sử dụng vector chuyển động để tạo khối dự báo, không cần thay đổi thêm vector chuyển động theo các độ phân giải của ảnh hiện tại và ảnh tham chiếu (chẳng hạn ($refx_L$, $refy_L$)).

Tương tác giữa RPR và các công cụ mã khác

8. Liệu/cách thức áp dụng quá trình lọc (chẳng hạn, bộ lọc tách khối) có thể phụ thuộc vào các độ phân giải của các ảnh tham chiếu và/hoặc độ phân giải của ảnh hiện tại.
 - a. Trong một ví dụ, các thiết lập độ bền đường biên (BS) trong các bộ lọc tách khối có thể xem xét các độ lệch độ phân giải bên cạnh các hiệu số vector chuyển động.
 - i. Trong một ví dụ, hiệu số vector chuyển động được chia tỷ lệ theo các độ phân giải của ảnh hiện tại và các ảnh tham chiếu có thể được sử dụng để xác định độ bền đường biên.
 - b. Trong một ví dụ, độ bền của bộ lọc tách khối đối với đường biên giữa khối A và khối B có thể được thiết lập khác (chẳng hạn, được tăng/giảm) nếu độ phân giải của ít nhất một ảnh tham chiếu của khối A là khác với (hoặc nhỏ hơn hoặc lớn hơn) độ phân giải của ít nhất một ảnh tham chiếu của khối B so với trường hợp mà các độ phân giải giống nhau được sử dụng đối với hai khối.
 - c. Trong một ví dụ, đường biên giữa khối A và khối B được đánh dấu là được lọc (chẳng hạn, BS được gán bằng 2) nếu độ phân giải của ít nhất một ảnh tham chiếu của khối A là khác với (hoặc nhỏ hơn hoặc lớn hơn) độ phân giải của ít nhất một ảnh tham chiếu của khối B.
 - d. Trong một ví dụ, độ bền của bộ lọc tách khối đối với đường biên giữa khối A và khối B có thể được thiết lập khác (chẳng hạn, được tăng/giảm) nếu độ phân giải của ít nhất một ảnh tham chiếu của khối A và/hoặc khối B là khác với (hoặc nhỏ hơn hoặc lớn hơn) độ phân giải của ảnh hiện tại so với trường hợp độ phân giải tương tự được sử dụng của ảnh tham chiếu và ảnh hiện tại.
 - e. Trong một ví dụ, đường biên giữa hai khối được đánh dấu là cần được lọc (chẳng hạn, BS được gán bằng 2) nếu ít nhất một ảnh tham chiếu

của ít nhất một khối trong hai khối có độ phân giải khác với độ phân giải của ảnh hiện tại.

9. Khi có ảnh phụ, dòng bit tương hợp có thể cần thỏa mãn ảnh tham chiếu phải có cùng độ phân giải với ảnh hiện tại.
 - a. Theo cách khác, khi ảnh tham chiếu có độ phân giải khác nhau với ảnh hiện tại, không cần có ảnh phụ trong ảnh hiện tại.
 - b. Theo cách khác, đối với ảnh phụ trong ảnh hiện tại, không được phép sử dụng ảnh tham chiếu mà có độ phân giải khác với ảnh hiện tại.
 - i. Theo cách khác, ngoài ra, việc quản lý ảnh tham chiếu có thể được gọi để loại bỏ các ảnh tham chiếu đó có các độ phân giải khác nhau.

10. Trong một ví dụ, các ảnh phụ (chẳng hạn, cách thức tách một ảnh thành nhiều ảnh phụ) có thể được định nghĩa riêng rẽ đối với các ảnh có các độ phân giải khác nhau.

Trong một ví dụ, ảnh phụ tương ứng trong ảnh tham chiếu có thể được dẫn xuất bằng cách chia tỷ lệ và/hoặc làm lệch ảnh phụ của ảnh hiện tại, nếu ảnh tham chiếu có độ phân giải khác nhau với ảnh hiện tại.

11. PROF (tinh lọc dự báo có luồng quang) có thể được kích hoạt khi ảnh tham chiếu có độ phân giải khác với của ảnh hiện tại.
 - a. Trong một ví dụ, một tập hợp của MV (được ký hiệu là MV_g) có thể được tạo đối với nhóm các mẫu và có thể được sử dụng để bù chuyển động như được mô tả trong đề mục 1. Mặt khác, MV (được ký hiệu là MV_p) có thể được dẫn xuất đối với mỗi mẫu, và hiệu số (chẳng hạn, tương ứng với Δv được sử dụng trong PROF) giữa MV_p và MV_g cùng với các gradien (chẳng hạn, các gradien không gian của các khối được bù chuyển động) có thể được sử dụng để dẫn xuất tinh lọc dự báo.
 - b. Trong một ví dụ, MV_p có thể có độ chính xác khác với MV_g . Chẳng hạn, MV_p có thể có độ chính xác $1/N\text{-pel}$ ($N > 0$), $N = 32, 64$ v.v..
 - c. Trong một ví dụ, MV_g có thể có độ chính xác khác với độ chính xác MV bên trong (chẳng hạn, $1/16\text{-pel}$).

- d. Trong một ví dụ, việc tinh lọc dự báo được bổ sung vào khối dự báo để tạo khối dự báo được tinh lọc.
 - e. Trong một ví dụ, phương pháp này có thể được áp dụng theo mỗi hướng dự báo.
 - f. Trong một ví dụ, phương pháp này có thể được áp dụng chỉ trong trường hợp dự báo đơn.
 - g. Trong một ví dụ, phương pháp này có thể được áp dụng trong dự báo đơn và/hoặc dự báo kép.
 - h. Trong một ví dụ, phương pháp này có thể được áp dụng chỉ khi ảnh tham chiếu có độ phân giải khác với ảnh hiện tại.
12. Đề xuất rằng chỉ một MV có thể được sử dụng cho khối/khối phụ để thực hiện quá trình bù chuyển động để dẫn xuất khối dự báo của khối hiện tại khi độ phân giải của ảnh tham chiếu là khác với độ phân giải của ảnh hiện tại.
- a. Trong một ví dụ, chỉ MV cho khối/khối phụ có thể được định nghĩa là hàm (chẳng hạn, trung bình) của tất cả các MV được liên kết với mỗi mẫu trong khối/khối phụ.
 - b. Trong một ví dụ, chỉ MV cho khối/khối phụ có thể được định nghĩa là MV được chọn được liên kết với mẫu được chọn (chẳng hạn, mẫu trung tâm) trong khối/khối phụ.
 - c. Trong một ví dụ, chỉ một MV có thể được sử dụng khối 4x4 hoặc khối phụ (chẳng hạn, 4x1).
 - d. Trong một ví dụ, BIO có thể còn được áp dụng để bù tổn hao độ chính xác do vector chuyển động dựa trên khối.
13. Chế độ lười không báo hiệu bất kỳ vector chuyển động dựa trên các khối có thể được áp dụng khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu khác với that của ảnh hiện tại.
- a. Trong một ví dụ, không vector chuyển động nào có thể được báo hiệu và quá trình bù chuyển động xấp xỉ trường hợp thay đổi độ phân giải thuần túy của ảnh tĩnh.

- b. Trong một ví dụ, chỉ vector chuyển động ở mức ảnh/ô/khối viên/CTU có thể là tín hiệu và các khối liên quan có thể sử dụng vector chuyển động khi thay đổi độ phân giải.
14. PROF có thể được áp dụng để xấp xỉ bù chuyển động khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu là khác với kích thước của ảnh hiện tại đối với các khối được mã hóa với chế độ dự báo afin và/hoặc chế độ dự báo không afin.
- a. Trong một ví dụ, PROF có thể được kích hoạt khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu và kích thước của ảnh hiện tại là khác nhau.
 - b. Trong một ví dụ, tập hợp chuyển động afin có thể được tạo bằng cách kết hợp chuyển động và phép chia tỷ lệ độ phân giải được chỉ báo và được sử dụng bởi PROF.
15. Dự báo đan xen (chẳng hạn như được nêu trong JVET-K0102) có thể được áp dụng để xấp xỉ bù chuyển động khi chiều rộng và/hoặc chiều cao của ảnh tham chiếu là khác với kích thước của ảnh hiện tại.
- a. Trong một ví dụ, thay đổi độ phân giải (thu phóng) được nêu như là chuyển động afin, và dự báo chuyển động được đan xen có thể được áp dụng.
16. LMCS và/hoặc phép chia tỷ lệ dư sắc độ có thể bị vô hiệu hóa khi chiều rộng và/hoặc chiều cao của ảnh hiện tại là khác với kích thước của ảnh IRAP trong cùng chu kỳ IRAP.
- a. Trong một ví dụ, khi LMCS bị vô hiệu hóa, các cờ mức lát chẳng hạn `slice_lmcs_enabled_flag`, `slice_lmcs_aps_id`, và `slice_chroma_residual_scale_flag` có thể không được báo hiệu và được suy ra bằng 0.
 - b. Trong một ví dụ, khi phép chia tỷ lệ dư sắc độ bị vô hiệu hóa, các cờ mức lát chẳng hạn `slice_chroma_residual_scale_flag` có thể không được báo hiệu và được suy ra bằng 0.

Các ràng buộc ở RPR

17. RPR có thể được áp dụng cho các khối mã có các ràng buộc kích thước khối.

- a. Trong một ví dụ, đối với khối mã $M \times N$, với M là chiều rộng khối và N là chiều cao khối, khi $M * N < T$ hoặc $M * N \leq T$ (chẳng hạn $T = 256$), RPR có thể không được sử dụng.
- b. Trong một ví dụ, khi $M < K$ (hoặc $M \leq K$) (chẳng hạn, $K = 16$) và/hoặc $N < L$ (hoặc $N \leq L$) (chẳng hạn, $L = 16$), RPR có thể không được sử dụng.

18. Tương hợp dòng bit có thể được bổ sung để giới hạn tỷ lệ giữa chiều rộng và/hoặc chiều cao của ảnh tham chiếu hoạt động (hoặc cửa sổ tương hợp của nó) và kích thước của ảnh hiện tại (hoặc cửa sổ tương hợp của nó). Giả sử $refPicW$ và $refPicH$ ký hiệu chiều rộng và chiều cao của ảnh tham chiếu, $curPicW$ và $curPicH$ ký hiệu chiều rộng và chiều cao của ảnh hiện tại,

- a. Trong một ví dụ, khi $(refPicW \div curPicW)$ bằng số nguyên, ảnh tham chiếu có thể được đánh dấu là ảnh tham chiếu hoạt động.
 - i. Theo cách khác, khi $(refPicW \div curPicW)$ bằng số phân số, ảnh tham chiếu có thể được đánh dấu là không khả dụng.
- b. Trong một ví dụ, khi $(refPicW \div curPicW)$ bằng $(X * n)$, trong đó X ký hiệu số phân số chẳng hạn $X = 1/2$, và n ký hiệu số nguyên chẳng hạn $n = 1, 2, 3, 4, \dots$, ảnh tham chiếu có thể được đánh dấu là ảnh tham chiếu hoạt động.
 - i. Trong một ví dụ, khi $(refPicW \div curPicW)$ không bằng $(X * n)$, ảnh tham chiếu có thể được đánh dấu là không khả dụng.

19. Liệu và/hoặc cách thức kích hoạt công cụ mã (chẳng hạn, dự báo kép/toàn bộ chế độ dự báo tam giác (TPM)/quá trình kết hợp trong TPM) đối với khối $M \times N$ có thể phụ thuộc vào các độ phân giải của các ảnh tham chiếu (hoặc các cửa sổ tương hợp) và/hoặc độ phân giải của ảnh hiện tại (hoặc cửa sổ tương hợp của nó).

- a. Trong một ví dụ, $M * N < T$ hoặc $M * N \leq T$ (chẳng hạn, $T = 64$).
- b. Trong một ví dụ, $M < K$ (hoặc $M \leq K$) (chẳng hạn, $K = 16$) và/hoặc $N < L$ (hoặc $N \leq L$) (chẳng hạn, $L = 16$).
- c. Trong một ví dụ, công cụ mã không được phép khi chiều rộng/chiều cao của ít nhất một ảnh tham chiếu là khác nhau với ảnh hiện tại,

- i. Trong một ví dụ, công cụ mã không được phép khi chiều rộng/chiều cao của ít nhất một ảnh tham chiếu của khối lớn hơn của ảnh hiện tại.
- d. Trong một ví dụ, công cụ mã không được phép khi chiều rộng/chiều cao của mỗi ảnh tham chiếu của khối là khác với kích thước của ảnh hiện tại,
 - i. Trong một ví dụ, công cụ mã không được phép khi chiều rộng/chiều cao của mỗi các ảnh tham chiếu lớn hơn của ảnh hiện tại.
- e. Theo cách khác, ngoài ra, khi công cụ mã không được phép, bù chuyển động có thể được tiến hành với một MV như là dự báo đơn.

Cửa sổ tương hợp liên quan

- 20. Các tham số độ lệch cửa sổ cắt tương hợp (chẳng hạn, `conf_win_left_offset`) được báo hiệu có độ chính xác N điểm ảnh thay vì 1-pel trong đó N là số nguyên dương lớn hơn 1.
 - a. Trong một ví dụ, độ lệch thực tế có thể được dẫn xuất như là độ lệch được báo hiệu được nhân với N.
 - b. Trong một ví dụ, N được gán bằng 4 hoặc 8.
- 21. Đề xuất rằng các tham số độ lệch cửa sổ cắt tương hợp không chỉ được áp dụng ở đầu ra. Các quá trình giải mã nội bộ cụ thể có thể phụ thuộc vào kích thước ảnh được cắt (chẳng hạn, độ phân giải của cửa sổ tương hợp trong ảnh).
- 22. Đề xuất rằng các tham số độ lệch cửa sổ cắt tương hợp trong đơn vị video thứ nhất (chẳng hạn, PPS) và trong đơn vị video thứ hai có thể là khác nhau khi chiều rộng và/hoặc chiều cao của ảnh được ký hiệu là (`pic_width_in_luma_samples`, `pic_height_in_luma_samples`) trong đơn vị video thứ nhất và đơn vị video thứ hai là giống nhau.
- 23. Đề xuất rằng các tham số độ lệch cửa sổ cắt tương hợp trong đơn vị video thứ nhất (chẳng hạn, PPS) và trong đơn vị video thứ hai nên giống nhau trong dòng bit tương hợp khi chiều rộng và/hoặc chiều cao của ảnh được ký hiệu

là (pic_width_in_luma_samples, pic_height_in_luma_samples) trong đơn vị video thứ nhất và đơn vị video thứ hai là khác nhau.

- a. Đề xuất rằng các tham số độ lệch của sổ cắt tương hợp trong đơn vị video thứ nhất (chẳng hạn, PPS) và trong đơn vị video thứ hai nên giống nhau trong dòng bit tương hợp bất kể chiều rộng và/hoặc chiều cao của ảnh được ký hiệu là (pic_width_in_luma_samples, pic_height_in_luma_samples) trong đơn vị video thứ nhất và đơn vị video thứ hai là giống nhau hoặc không.

24. Giả sử chiều rộng và chiều cao của cửa sổ tương hợp được định nghĩa trong đơn vị video thứ nhất (chẳng hạn, PPS) được ký hiệu lần lượt là W1 và H1. Chiều rộng và chiều cao của cửa sổ tương hợp được định nghĩa trong đơn vị video thứ hai (chẳng hạn, PPS) được ký hiệu lần lượt là W2 và H2. Vị trí trên bên trái của cửa sổ tương hợp được định nghĩa trong đơn vị video thứ nhất (chẳng hạn, PPS) được ký hiệu là X1 và Y1. Vị trí trên bên trái của cửa sổ tương hợp được định nghĩa trong đơn vị video thứ hai (chẳng hạn, PPS) được ký hiệu là X2 và Y2. Chiều rộng và chiều cao của ảnh được mã hóa/giải mã (chẳng hạn pic_width_in_luma_samples và pic_height_in_luma_samples) được định nghĩa trong đơn vị video thứ nhất (chẳng hạn, PPS) được ký hiệu lần lượt là PW1 và PH1. Chiều rộng và chiều cao của ảnh được mã hóa/giải mã được định nghĩa trong đơn vị video thứ hai (chẳng hạn PPS) được ký hiệu là PW2 và PH2.

- a. Trong một ví dụ, W1/W2 phải bằng X1/X2 trong dòng bit tương hợp.
 - i. Theo cách khác, W1/X1 phải bằng W2/X2 trong dòng bit tương hợp.
 - ii. Theo cách khác, W1 * X2 phải bằng W2 * X1 trong dòng bit tương hợp.
- b. Trong một ví dụ, H1/H2 phải bằng Y1/Y2 trong dòng bit tương hợp.
 - i. Theo cách khác, H1/Y1 phải bằng H2/Y2 trong dòng bit tương hợp.
 - ii. Theo cách khác, H1 * Y2 phải bằng H2 * Y1 trong dòng bit tương hợp.

- c. Trong một ví dụ, $PW1/PW2$ phải bằng $X1/X2$ trong dòng bit tương hợp.
 - i. Theo cách khác, $PW1/X1$ phải bằng $PW2/X2$ trong dòng bit tương hợp.
 - ii. Theo cách khác, $PW1 * X2$ phải bằng $PW2 * X1$ trong dòng bit tương hợp.
- d. Trong một ví dụ, $PH1/PH2$ phải bằng $Y1/Y2$ trong dòng bit tương hợp.
 - i. Theo cách khác, $PH1/Y1$ phải bằng $PH2/Y2$ trong dòng bit tương hợp.
 - ii. Theo cách khác, $PH1 * Y2$ phải bằng $PH2 * Y1$ trong dòng bit tương hợp.
- e. Trong một ví dụ, $PW1/PW2$ phải bằng $W1/W2$ trong dòng bit tương hợp.
 - i. Theo cách khác, $PW1/W1$ phải bằng $PW2/W2$ trong dòng bit tương hợp.
 - ii. Theo cách khác, $PW1 * W2$ phải bằng $PW2 * W1$ trong dòng bit tương hợp.
- f. Trong một ví dụ, $PH1/PH2$ phải bằng $H1/H2$ trong dòng bit tương hợp.
 - i. Theo cách khác, $PH1/H1$ phải bằng $PH2/H2$ trong dòng bit tương hợp.
 - ii. Theo cách khác, $PH1 * H2$ phải bằng $PH2 * H1$ trong dòng bit tương hợp.
- g. Trong dòng bit tương hợp, nếu $PW1$ lớn hơn $PW2$, $W1$ phải lớn hơn $W2$.
- h. Trong dòng bit tương hợp, nếu $PW1$ nhỏ hơn $PW2$, $W1$ phải nhỏ hơn $W2$.
- i. Trong dòng bit tương hợp, $(PW1-PW2) * (W1-W2)$ phải không nhỏ hơn 0.

- j. Trong dòng bit tương hợp, nếu $PH1$ lớn hơn $PH2$, $H1$ phải lớn hơn $H2$.
 - k. Trong dòng bit tương hợp, nếu $PH1$ nhỏ hơn $PH2$, $H1$ phải nhỏ hơn $H2$.
 - l. Trong dòng bit tương hợp, $(PH1 - PH2) * (H1 - H2)$ phải không nhỏ hơn 0.
 - m. Trong dòng bit tương hợp, $W1/W2$ phải không lớn hơn (hoặc nhỏ hơn) $PW1/PW2$ nếu $PW1 \geq PW2$.
 - n. Trong dòng bit tương hợp, $H1/H2$ phải không lớn hơn (hoặc nhỏ hơn) $PH1/PH2$ nếu $PH1 \geq PH2$.
25. Giả sử chiều rộng và chiều cao của cửa sổ tương hợp của ảnh hiện tại được ký hiệu lần lượt là W và H . Chiều rộng và chiều cao của cửa sổ tương hợp của ảnh tham chiếu được ký hiệu lần lượt là W' và H' . Sau đó, ít nhất một ràng buộc dưới đây nên được tuân theo bởi dòng bit tương hợp.
- a. $W * pw \geq W'$; pw là số nguyên chẵn hạn 2.
 - b. $W * pw > W'$; pw là số nguyên chẵn hạn 2.
 - c. $W' * pw' \geq W$; pw' là số nguyên chẵn hạn 8.
 - d. $W' * pw' > W$; pw' là số nguyên chẵn hạn 8.
 - e. $H * ph \geq H'$; ph là số nguyên chẵn hạn 2.
 - f. $H * ph > H'$; ph là số nguyên chẵn hạn 2.
 - g. $H' * ph' \geq H$; ph' là số nguyên chẵn hạn 8.
 - h. $H' * ph' > H$; ph' là số nguyên chẵn hạn 8.
 - i. Trong một ví dụ, pw bằng pw' .
 - j. Trong một ví dụ, ph bằng ph' .
 - k. Trong một ví dụ, pw bằng ph .
 - l. Trong một ví dụ, pw' bằng ph' .
 - m. Trong một ví dụ, các đề mục phụ nêu trên có thể được yêu cầu thỏa mãn bởi dòng bit tương hợp khi W và H lần lượt là chiều rộng và chiều cao của ảnh hiện tại. W' và H' là chiều rộng và chiều cao của ảnh tham chiếu.
26. Đề xuất rằng các tham số cửa sổ tương hợp được báo hiệu một phần.

- a. Trong một ví dụ, mẫu trên bên trái trong cửa sổ tương hợp của ảnh là mẫu tương tự như mẫu trong ảnh.
- b. Chẳng hạn, `conf_win_left_offset` như được định nghĩa trong VVC không được báo hiệu và được suy ra bằng 0.
- c. Chẳng hạn, `conf_win_top_offset` như được định nghĩa trong VVC không được báo hiệu và được suy ra bằng 0.

27. Đề xuất rằng việc dẫn xuất vị trí (chẳng hạn (`refxL`, `refyL`) như được định nghĩa trong VVC) của mẫu tham chiếu có thể phụ thuộc vào vị trí trên bên trái (chẳng hạn (`conf_win_left_offset`, `conf_win_top_offset`) như được định nghĩa trong VVC) của cửa sổ tương hợp của ảnh hiện tại và/hoặc ảnh tham chiếu. Fig.4 thể hiện các ví dụ của các vị trí mẫu được dẫn xuất như trong VVC (a) và trong phương pháp được đề xuất (b). Các hình chữ nhật nét đứt biểu diễn các cửa sổ tương hợp.

- a. Trong một ví dụ, tồn tại lệ thuộc chỉ khi chiều rộng và/hoặc chiều cao của ảnh hiện tại và kích thước của ảnh tham chiếu là khác nhau.
- b. Trong một ví dụ, việc dẫn xuất vị trí ngang (chẳng hạn `refxL` như được định nghĩa trong VVC) của mẫu tham chiếu có thể phụ thuộc vào vị trí bên trái (chẳng hạn `conf_win_left_offset` như được định nghĩa trong VVC) của cửa sổ tương hợp của ảnh hiện tại và/hoặc ảnh tham chiếu.
 - i. Trong một ví dụ, vị trí ngang (được ký hiệu là xSb') của mẫu hiện tại so với vị trí trên bên trái của cửa sổ tương hợp trong ảnh hiện tại được tính toán và được sử dụng để dẫn xuất vị trí của mẫu tham chiếu.

1) Chẳng hạn, $xSb' = xSb - (\text{conf_win_left_offset} \ll \text{Prec})$ được tính toán và được sử dụng để dẫn xuất vị trí của mẫu tham chiếu, trong đó xSb là vị trí ngang của mẫu hiện tại trong ảnh hiện tại. `conf_win_left_offset` là vị trí ngang của mẫu trên bên trái trong cửa sổ tương hợp của ảnh hiện tại. `Prec` là độ chính xác của xSb và xSb' trong đó $(xSb \gg \text{Prec})$ có thể thể hiện tọa độ ngang

thực của mẫu hiện tại so với ảnh hiện tại. Chẳng hạn, $Prec = 0$ hoặc $Prec = 4$.

- ii. Trong một ví dụ, vị trí ngang (được ký hiệu là Rx') của mẫu tham chiếu so với vị trí trên bên trái của cửa sổ tương hợp trong ảnh tham chiếu được tính toán.

1) Việc tính toán Rx' có thể phụ thuộc vào xSb' , và/hoặc vector chuyển động, và/hoặc tỷ lệ lấy mẫu lại.

- iii. Trong một ví dụ, vị trí ngang (được ký hiệu là Rx) của mẫu tham chiếu liên quan trong ảnh tham chiếu được tính toán tùy thuộc Rx' .

1) Chẳng hạn, $Rx = Rx' + (conf_win_left_offset_ref \ll Prec)$ được tính toán, trong đó $conf_win_left_offset_ref$ là vị trí ngang của mẫu trên bên trái trong cửa sổ tương hợp của ảnh tham chiếu. $Prec$ là độ chính xác của Rx và Rx' . Chẳng hạn, $Prec = 0$ hoặc $Prec = 4$.

- iv. Trong một ví dụ, Rx có thể được tính toán trực tiếp tùy thuộc xSb' , và/hoặc vector chuyển động, và/hoặc tỷ lệ lấy mẫu lại. Nói cách khác, hai bước dẫn xuất ở Rx' và Rx được kết hợp thành phép tính một bước.

- v. Liệu và/hoặc cách thức sử dụng vị trí bên trái (chẳng hạn $conf_win_left_offset$ như được định nghĩa trong VVC) của cửa sổ tương hợp của ảnh hiện tại và/hoặc ảnh tham chiếu có thể phụ thuộc vào các thành phần màu và/hoặc định dạng màu.

1) Chẳng hạn, $conf_win_left_offset$ có thể được sửa lại thành $conf_win_left_offset = conf_win_left_offset * SubWidthC$, trong đó $SubWidthC$ định nghĩa bước lấy mẫu ngang của thành phần màu. Chẳng hạn, $SubWidthC$ bằng 1 đối với thành phần độ sáng. $SubWidthC$ bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0 hoặc 4:2:2.

- 2) Chẳng hạn, $\text{conf_win_left_offset}$ có thể được sửa lại thành $\text{conf_win_left_offset} = \text{conf_win_left_offset} / \text{SubWidthC}$, trong đó SubWidthC định nghĩa bước lấy mẫu ngang của thành phần màu. Chẳng hạn, SubWidthC bằng 1 đối với thành phần độ sáng. SubWidthC bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0 hoặc 4:2:2.
- c. Trong một ví dụ, việc dẫn xuất vị trí dọc (chẳng hạn refy_L như được định nghĩa trong VVC) của mẫu tham chiếu có thể phụ thuộc vào vị trí trên (chẳng hạn $\text{conf_win_top_offset}$ như được định nghĩa trong VVC) của cửa sổ tương hợp của ảnh hiện tại và/hoặc ảnh tham chiếu.
- i. Trong một ví dụ, vị trí dọc (được ký hiệu là y_{Sb}') của mẫu hiện tại so với vị trí trên bên trái của cửa sổ tương hợp trong ảnh hiện tại được tính toán và được sử dụng để dẫn xuất vị trí của mẫu tham chiếu.
- 1) Chẳng hạn, $y_{Sb}' = y_{Sb} - (\text{conf_win_top_offset} \ll \text{Prec})$ được tính toán và được sử dụng để dẫn xuất vị trí của mẫu tham chiếu, trong đó y_{Sb} là vị trí dọc của mẫu hiện tại trong ảnh hiện tại. $\text{conf_win_top_offset}$ represents vị trí dọc của mẫu trên bên trái trong cửa sổ tương hợp của ảnh hiện tại. Prec là độ chính xác của y_{Sb} và y_{Sb}' . Chẳng hạn, $\text{Prec} = 0$ hoặc $\text{Prec} = 4$.
- ii. Trong một ví dụ, vị trí dọc (được ký hiệu là Ry') của mẫu tham chiếu so với vị trí trên bên trái của cửa sổ tương hợp trong ảnh tham chiếu được tính toán.
- 1) Việc tính toán Ry' có thể phụ thuộc vào y_{Sb}' , và/hoặc vector chuyển động, và/hoặc tỷ lệ lấy mẫu lại.
- iii. Trong một ví dụ, vị trí dọc (được ký hiệu là Ry) của mẫu tham chiếu liên quan trong ảnh tham chiếu được tính toán tùy thuộc Ry' .

1) Chẳng hạn, $Ry = Ry' + (conf_win_top_offset_ref \ll Prec)$ được tính toán, trong đó $conf_win_top_offset_ref$ represents vị trí dọc của mẫu trên bên trái trong cửa sổ tương hợp của ảnh tham chiếu. $Prec$ là độ chính xác của Ry and Ry' . Chẳng hạn, $Prec = 0$ hoặc $Prec = 4$.

iv. Trong một ví dụ, Ry có thể được tính toán trực tiếp tùy thuộc ySb' , và/hoặc vector chuyển động, và/hoặc tỷ lệ lấy mẫu lại. Nói cách khác, hai bước dẫn xuất ở Ry' và Ry được kết hợp thành phép tính một bước.

v. Liệu và/hoặc cách thức sử dụng vị trí trên (chẳng hạn $conf_win_top_offset$ như được định nghĩa trong VVC) của cửa sổ tương hợp của ảnh hiện tại và/hoặc ảnh tham chiếu có thể phụ thuộc vào các thành phần màu và/hoặc các định dạng màu.

1) Chẳng hạn, $conf_win_top_offset$ có thể được sửa lại thành $conf_win_top_offset = conf_win_top_offset * SubHeightC$, trong đó $SubHeightC$ định nghĩa bước lấy mẫu dọc của thành phần màu. Chẳng hạn, $SubHeightC$ bằng 1 đối với thành phần độ sáng. $SubHeightC$ bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0.

2) Chẳng hạn, $conf_win_top_offset$ có thể được sửa lại thành $conf_win_top_offset = conf_win_top_offset / SubHeightC$, trong đó $SubHeightC$ định nghĩa bước lấy mẫu dọc của thành phần màu. Chẳng hạn, $SubHeightC$ bằng 1 đối với thành phần độ sáng. $SubHeightC$ bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0.

28. Đề xuất rằng phần số nguyên của tọa độ ngang của mẫu tham chiếu có thể được cắt thành $[minW, maxW]$. Giả sử chiều rộng và chiều cao của cửa sổ tương hợp của ảnh tham chiếu được ký hiệu lần lượt là W và H . Chiều rộng và chiều cao của cửa sổ tương hợp của ảnh tham chiếu được ký hiệu là W'

và H' . Vị trí trên bên trái của cửa sổ tương hợp trong ảnh tham chiếu được ký hiệu là $(X0, Y0)$.

- a. Trong một ví dụ, $\min W$ bằng 0.
- b. Trong một ví dụ, $\min W$ bằng $X0$.
- c. Trong một ví dụ, $\max W$ bằng $W-1$.
- d. Trong một ví dụ, $\max W$ bằng $W'-1$.
- e. Trong một ví dụ, $\max W$ bằng $X0+W'-1$.
- f. Trong một ví dụ, $\min W$ và/hoặc $\max W$ có thể được chỉnh sửa dựa trên định dạng màu và/hoặc thành phần màu.
 - i. Chẳng hạn, $\min W$ được chỉnh sửa thành $\min W * \text{SubC}$.
 - ii. Chẳng hạn, $\min W$ được chỉnh sửa thành $\min W / \text{SubC}$.
 - iii. Chẳng hạn, $\max W$ được chỉnh sửa thành $\max W * \text{SubC}$.
 - iv. Chẳng hạn, $\max W$ được chỉnh sửa thành $\max W / \text{SubC}$.
 - v. Trong một ví dụ, SubC bằng 1 đối với thành phần độ sáng.
 - vi. Trong một ví dụ, SubC bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0.
 - vii. Trong một ví dụ, SubC bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:2.
 - viii. Trong một ví dụ, SubC bằng 1 đối với thành phần sắc độ khi định dạng màu là 4:4:4.
- g. Trong một ví dụ, liệu và/hoặc cách thực hiện cắt có thể phụ thuộc vào các kích thước của ảnh hiện tại (hoặc cửa sổ tương hợp trong đó) và các kích thước của ảnh tham chiếu (hoặc cửa sổ tương hợp trong đó).
 - i. Trong một ví dụ, việc cắt được thực hiện chỉ khi các kích thước của ảnh hiện tại (hoặc cửa sổ tương hợp trong đó) và các kích thước của ảnh tham chiếu (hoặc cửa sổ tương hợp trong đó) là khác nhau.

29. Đề xuất rằng phần số nguyên của tọa độ dọc của mẫu tham chiếu có thể được cắt thành $[\min H, \max H]$. Giả sử chiều rộng và chiều cao của cửa sổ tương hợp của ảnh tham chiếu được ký hiệu lần lượt là W và H . Chiều rộng và chiều cao của cửa sổ tương hợp của ảnh tham chiếu được ký hiệu là W' và

H' . Vị trí trên bên trái của cửa sổ tương hợp trong ảnh tham chiếu được ký hiệu là (X_0, Y_0) .

- a. Trong một ví dụ, $\min H$ bằng 0.
- b. Trong một ví dụ, $\min H$ bằng Y_0 .
- c. Trong một ví dụ, $\max H$ bằng $H-1$.
- d. Trong một ví dụ, $\max H$ bằng $H'-1$.
- e. Trong một ví dụ, $\max H$ bằng $Y_0+H'-1$.
- f. Trong một ví dụ, $\min H$ và/hoặc $\max H$ có thể được chỉnh sửa dựa trên định dạng màu và/hoặc thành phần màu.
 - i. Chẳng hạn, $\min H$ được chỉnh sửa thành $\min H * \text{SubC}$.
 - ii. Chẳng hạn, $\min H$ được chỉnh sửa thành $\min H / \text{SubC}$.
 - iii. Chẳng hạn, $\max H$ được chỉnh sửa thành $\max H * \text{SubC}$.
 - iv. Chẳng hạn, $\max H$ được chỉnh sửa thành $\max H / \text{SubC}$.
 - v. Trong một ví dụ, SubC bằng 1 đối với thành phần độ sáng.
 - vi. Trong một ví dụ, SubC bằng 2 đối với thành phần sắc độ khi định dạng màu là 4:2:0.
 - vii. Trong một ví dụ, SubC bằng 1 đối với thành phần sắc độ khi định dạng màu là 4:2:2.
 - viii. Trong một ví dụ, SubC bằng 1 đối với thành phần sắc độ khi định dạng màu là 4:4:4.
- g. Trong một ví dụ, liệu và/hoặc cách thực hiện cắt có thể phụ thuộc vào các kích thước của ảnh hiện tại (hoặc cửa sổ tương hợp trong đó) và các kích thước của ảnh tham chiếu (hoặc cửa sổ tương hợp trong đó).
 - i. Trong một ví dụ, việc cắt được thực hiện chỉ khi các kích thước của ảnh hiện tại (hoặc cửa sổ tương hợp trong đó) và các kích thước của ảnh tham chiếu (hoặc cửa sổ tương hợp trong đó) là khác nhau.

Trong phần mô tả sau, phần tử cú pháp thứ nhất được thừa nhận là “tương ứng” với phần tử cú pháp thứ hai, nếu hai phần tử cú pháp có chức năng tương đương nhưng có thể được báo hiệu ở đơn vị video khác (chẳng hạn VPS/SPS/PPS/tiêu đề lát/tiêu đề ảnh etc.)

30. Đề xuất rằng phần tử cú pháp có thể được báo hiệu trong đơn vị video thứ nhất (chẳng hạn tiêu đề ảnh hoặc PPS) và không phần tử cú pháp tương ứng nào được báo hiệu trong đơn vị video thứ hai ở mức cao hơn (chẳng hạn SPS) hoặc mức thấp hơn (chẳng hạn tiêu đề lát).

- a. Theo cách khác, phần tử cú pháp thứ nhất có thể được báo hiệu trong đơn vị video thứ nhất (chẳng hạn tiêu đề ảnh hoặc PPS) và phần tử cú pháp thứ hai tương ứng có thể được báo hiệu trong đơn vị video thứ hai ở mức thấp hơn (chẳng hạn tiêu đề lát).
 - i. Theo cách khác, bộ chỉ báo có thể được báo hiệu trong đơn vị video thứ hai để thông báo liệu phần tử cú pháp thứ hai được báo hiệu sau đó.
 - ii. Trong một ví dụ, lát được liên kết với đơn vị video thứ hai (chẳng hạn tiêu đề lát) có thể tuân theo chỉ báo này của phần tử cú pháp thứ hai thay cho phần tử thứ nhất nếu phần tử thứ hai được báo hiệu.
 - iii. Bộ chỉ báo được liên kết với phần tử cú pháp thứ nhất có thể được báo hiệu trong đơn vị video thứ nhất để thông báo liệu phần tử cú pháp thứ hai được báo hiệu trong bất kỳ lát nào (hoặc đơn vị video khác) được liên kết với đơn vị video thứ nhất.
- b. Theo cách khác, phần tử cú pháp thứ nhất có thể được báo hiệu trong đơn vị video thứ nhất ở mức cao hơn (chẳng hạn, VPS/SPS/PPS), và phần tử cú pháp thứ hai tương ứng có thể được báo hiệu trong đơn vị video thứ hai (chẳng hạn, tiêu đề ảnh).
 - i. Theo cách khác, bộ chỉ báo có thể được báo hiệu để thông báo liệu phần tử cú pháp thứ hai được báo hiệu sau đó.
 - ii. Trong một ví dụ, ảnh (có thể được phân vùng thành các lát) được liên kết với đơn vị video thứ hai có thể tuân theo chỉ báo này của phần tử cú pháp thứ hai thay cho phần tử thứ nhất nếu phần tử thứ hai được báo hiệu.

- c. Phần tử cú pháp thứ nhất trong tiêu đề ảnh có thể có chức năng tương đương như phần tử cú pháp thứ hai trong tiêu đề lát như được xác định trong phần 2.6 (chẳng hạn, nhưng bị giới hạn ở **slice_temporal_mvp_enabled_flag**, **cabac_init_flag**, **six_minus_max_num_merge_cand**, **five_minus_max_num_subblock_merge_cand**, **slice_fpel_mmvd_enabled_flag**, **slice_disable_bdof_dmvr_flag**, **max_num_merge_cand_minus_max_num_triangle_cand**, **slice_fpel_mmvd_enabled_flag**, **slice_six_minus_max_num_ibc_merge_cand**, **slice_joint_cbr_sign_flag**, **slice_qp_delta**, ...) nhưng điều khiển tất cả các lát của ảnh.
- d. Phần tử cú pháp thứ nhất trong SPS như được xác định trong phần 2.6 có thể có chức năng tương đương như phần tử cú pháp thứ hai trong tiêu đề ảnh (chẳng hạn, nhưng bị giới hạn ở **sps_bdof_dmvr_slice_present_flag**, **sps_mmvd_enabled_flag**, **sps_isp_enabled_flag**, **sps_mrl_enabled_flag**, **sps_mip_enabled_flag**, **sps_cclm_enabled_flag**, **sps_mts_enabled_flag** ...) nhưng điều khiển chỉ ảnh liên kết (có thể được phân vùng thành các lát).
- e. Phần tử cú pháp thứ nhất trong PPS như được xác định trong phần 2.7 có thể có chức năng tương đương như phần tử cú pháp thứ hai trong tiêu đề ảnh (chẳng hạn, nhưng bị giới hạn ở **entropy_coding_sync_enabled_flag**, **entry_point_offsets_present_flag**, **cabac_init_present_flag**, **rpl1_idx_present_flag** ...) nhưng điều khiển chỉ ảnh liên kết (có thể được phân vùng thành các lát).
31. Các phần tử cú pháp được báo hiệu trong tiêu đề ảnh được tách khỏi các phần tử cú pháp khác được báo hiệu hoặc được dẫn xuất trong SPS/VPS/DPS.

32. Chỉ báo kích hoạt/vô hiệu hóa DMVR và BDOF có thể được báo hiệu riêng rẽ trong tiêu đề ảnh, thay vì được điều khiển bằng cùng cờ (chẳng hạn, `pic_disable_bdof_dmvr_flag`).

33. Chỉ báo kích hoạt/vô hiệu hóa PROF/ALF giữa thành phần/dự báo ngoài có phân vùng hình học (GEO) có thể được báo hiệu trong các tiêu đề ảnh.

- a. Theo cách khác, chỉ báo kích hoạt/vô hiệu hóa PROF trong tiêu đề ảnh có thể được báo hiệu có điều kiện theo cờ kích hoạt PROF trong SPS.
- b. Theo cách khác, chỉ báo kích hoạt/vô hiệu hóa ALF giữa thành phần (CCALF) trong tiêu đề ảnh có thể được báo hiệu có điều kiện theo cờ kích hoạt CCALF trong SPS.
- c. Theo cách khác, chỉ báo này kích hoạt/vô hiệu hóa GEO trong tiêu đề ảnh có thể được báo hiệu có điều kiện theo cờ kích hoạt GEO trong SPS.
- d. Theo cách khác, ngoài ra, các chỉ báo kích hoạt/vô hiệu hóa PROF/ALF giữa các thành phần/dự báo ngoài có GEO trong các tiêu đề lát có thể được báo hiệu có điều kiện theo các phân tử cú pháp đó được báo hiệu trong tiêu đề ảnh thay vì SPS.

34. Các chỉ báo các loại dự báo về các lát/khối viên/ô (hoặc các đơn vị video khác nhỏ hơn ảnh) trong cùng ảnh có thể được báo hiệu trong tiêu đề ảnh.

- a. Trong một ví dụ, chỉ báo của liệu tất cả các lát/khối viên/ô (hoặc các đơn vị video khác nhỏ hơn ảnh) đều được mã hóa trong (chẳng hạn, tất cả các lát I) có thể được báo hiệu trong tiêu đề ảnh.
 - i. Theo cách khác, ngoài ra, các loại lát có thể không được báo hiệu trong tiêu đề lát nếu chỉ báo này báo tất cả các lát trong ảnh là các lát I.
- b. Theo cách khác, chỉ báo của liệu ít nhất một trong các lát/khối viên/ô (hoặc các đơn vị video khác nhỏ hơn ảnh) không được mã hóa trong (chẳng hạn, ít nhất một lát không I) có thể được báo hiệu trong tiêu đề ảnh.

- c. Theo cách khác, chỉ báo của liệu tất cả các lát/khối viên/ô (hoặc các đơn vị video khác nhỏ hơn ảnh) đều có cùng loại dự báo (chẳng hạn, các lát I/P/B) có thể được báo hiệu trong tiêu đề ảnh.
 - i. Theo cách khác, ngoài ra, các loại lát có thể không được báo hiệu trong tiêu đề lát.
 - ii. Theo cách khác, ngoài ra, chỉ báo của các công cụ được phép cho các loại dự báo cụ thể (chẳng hạn, DMVR/BDOF/TPM/GEO chỉ được phép cho các lát B; cây kép chỉ được phép cho các lát I) có thể được báo hiệu có điều kiện theo chỉ báo loại dự báo.
 - d. Theo cách khác, ngoài ra, việc báo hiệu các chỉ báo kích hoạt/vô hiệu hóa các công cụ có thể phụ thuộc vào các chỉ báo loại dự báo được nêu trong các đề mục phụ trên đây.
 - i. Theo cách khác, ngoài ra, các chỉ báo kích hoạt/vô hiệu hóa các công cụ có thể được dẫn xuất theo các chỉ báo loại dự báo được nêu trong các đề mục phụ trên đây.
35. Theo sáng chế (đề mục 1 đến đề mục 29), cụm từ “cửa sổ tương hợp” có thể được thay thế bằng các cụm từ khác chẳng hạn “cửa sổ chia tỷ lệ”. Cửa sổ chia tỷ lệ có thể được báo hiệu khác với cửa sổ tương hợp và được sử dụng để dẫn xuất hệ số chia tỷ lệ và/hoặc độ lệch trên bên trái được sử dụng để dẫn xuất vị trí mẫu tham chiếu cho RPR.
- a. Trong một ví dụ, cửa sổ chia tỷ lệ có thể bị ràng buộc bởi cửa sổ tương hợp. Chẳng hạn, trong dòng bit tương hợp, cửa sổ chia tỷ lệ phải được chứa bởi cửa sổ tương hợp.
36. Liệu và/hoặc cách thức báo hiệu kích thước khối lớn nhất được phép cho các khối được mã hóa TS có thể phụ thuộc vào kích thước khối lớn nhất cho các khối được mã hóa biến đổi.
- a. Theo cách khác, kích thước khối lớn nhất cho các khối được mã hóa TS không thể lớn hơn kích thước khối lớn nhất cho các khối được mã hóa biến đổi trong dòng bit tương hợp.

37. Liệu và cách thức báo hiệu chỉ báo kích hoạt mã hóa phần dư Cb-Cr đồng thời (JCCR) (chẳng hạn, `sps_joint_cbr_enabled_flag`) có thể phụ thuộc vào định dạng màu (chẳng hạn, 4:0:0, 4:2:0 v.v.)

- a. Chẳng hạn, chỉ báo kích hoạt phần dư Cb-Cr đồng thời (JCCR) có thể không được báo hiệu nếu định dạng màu là 4:0:0. Thiết kế cú pháp lấy làm ví dụ dưới đây:

<i>if(ChromaArrayType != 0)</i>	
<code>sps_joint_cbr_enabled_flag</code>	u(1)

Loại bộ lọc giảm kích thước mẫu để tạo mặt nạ kết hợp sắc độ trong TPM/GEO

38. Loại bộ lọc giảm kích thước mẫu được sử dụng để dẫn xuất các trọng số kết hợp cho các mẫu sắc độ có thể được báo hiệu ở mức đơn vị video (chẳng hạn SPS/VPS/PPS/Tiêu đề ảnh/Mức ảnh con/lát/tiêu đề lát ô/khối viên/CTY/VPDU).

- a. Trong một ví dụ, cờ mức cao có thể được báo hiệu để chuyển đổi giữa các loại định dạng sắc độ khác nhau của nội dung.
- Trong một ví dụ, cờ mức cao có thể được báo hiệu để chuyển đổi giữa loại định dạng sắc độ 0 and loại định dạng sắc độ 2.
 - Trong một ví dụ, cờ có thể được báo hiệu để xác định liệu các trọng số độ sáng bị giảm kích thước mẫu trên bên trái trong chế độ dự báo TPM/GEO cùng vị trí với các trọng số độ sáng trên bên trái (chẳng hạn, loại vị trí mẫu sắc độ 0).
 - Trong một ví dụ, cờ có thể được báo hiệu để xác định liệu mẫu độ sáng bị giảm kích thước mẫu trên bên trái trong chế độ dự báo TPM/GEO nằm cùng vị trí theo chiều ngang với mẫu độ sáng trên bên trái nhưng được xê dịch theo chiều dọc đi 0,5 đơn vị của mẫu độ sáng so với mẫu độ sáng trên bên trái (chẳng hạn, loại vị trí mẫu sắc độ 2).
- b. Trong một ví dụ, loại bộ lọc giảm kích thước mẫu có thể được báo hiệu đối với định dạng sắc độ 4:2:0 và/hoặc định dạng sắc độ 4:2:2.

- c. Trong một ví dụ, cờ có thể được báo hiệu để xác định loại bộ lọc giảm kích thước mẫu sắc độ được sử dụng để dự báo TPM/GEO.
 - i. Trong một ví dụ, cờ có thể được báo hiệu liệu có sử dụng bộ lọc giảm kích thước mẫu A hoặc bộ lọc giảm kích thước mẫu B để dẫn xuất các trọng số sắc độ trong chế độ dự báo TPM/GEO.
39. Loại bộ lọc giảm kích thước mẫu được sử dụng để dẫn xuất các trọng số kết hợp cho các mẫu sắc độ có thể được dẫn xuất ở mức đơn vị video (chẳng hạn SPS/VPS/PPS/Tiêu đề ảnh/Mức ảnh con/lát/tiêu đề lát/ô/khối viên/CTY/VPDU).
- a. Trong một ví dụ, bảng tra cứu có thể được định nghĩa để xác định mối quan hệ tương ứng giữa loại bộ lọc lấy mẫu phụ sắc độ và các loại định dạng sắc độ của nội dung.
40. Bộ lọc giảm kích thước mẫu xác định có thể được sử dụng để chế độ dự báo TPM/GEO nếu loại vị trí mẫu sắc độ khác nhau.
- a. Trong một ví dụ, các trọng số sắc độ của TPM/GEO có thể được lấy mẫu phụ từ các trọng số độ sáng trên bên trái cùng vị trí nếu có loại vị trí mẫu sắc độ cụ thể (chẳng hạn, loại vị trí mẫu sắc độ 0).
 - b. Trong một ví dụ, nếu có loại vị trí mẫu sắc độ cụ thể (chẳng hạn, loại vị trí mẫu sắc độ 0 hoặc 2), bộ lọc X bậc cụ thể (X là hằng số chẳng hạn $X = 6$ hoặc 5) có thể được sử dụng để lấy mẫu phụ các trọng số sắc độ trong chế độ dự báo TPM/GEO.
41. Trong đơn vị video (chẳng hạn SPS, PPS, tiêu đề ảnh, tiêu đề lát v.v.), phần tử cú pháp thứ nhất (chẳng hạn cờ) có thể được báo hiệu để chỉ báo liệu lựa chọn đa biến đổi (MTS) bị vô hiệu hóa cho tất cả các khối (các lát/các ảnh).
- a. Phần tử cú pháp thứ hai chỉ báo cách thức áp dụng MTS (chẳng hạn kích hoạt MTS/vô hiệu hóa MTS/MTS ngầm/MTS tường minh) trên các khối mã hóa trong (các lát/các ảnh) được báo hiệu có điều kiện on phần tử cú pháp thứ nhất. Chẳng hạn, phần tử cú pháp thứ hai được báo hiệu chỉ khi phần tử cú pháp thứ nhất chỉ báo rằng MTS không bị vô hiệu hóa cho tất cả các khối (các lát/các ảnh).

- b. Phần tử cú pháp thứ ba chỉ báo cách thức áp dụng MTS (chẳng hạn kích hoạt MTS/vô hiệu hóa MTS/MTS ngầm/MTS tường minh) trên các khối mã hóa ngoài (các lát/các ảnh) được báo hiệu có điều kiện on phần tử cú pháp thứ nhất. Chẳng hạn, phần tử cú pháp thứ ba được báo hiệu chỉ khi phần tử cú pháp thứ nhất chỉ báo rằng MTS không bị vô hiệu hóa cho tất cả các khối (các lát/các ảnh).

- c. Thiết kế cú pháp lấy làm ví dụ dưới đây

...	
enable mts flag	
if(enable mts flag){	
mts_control intra	
mts_control inter	
}	
...	

- d. Phần tử cú pháp thứ ba có thể được báo hiệu có điều kiện về liệu biến đổi khối phụ (SBT) được áp dụng hoặc không. Thiết kế cú pháp lấy làm ví dụ dưới đây

...	
if(sps_sbt_enabled_flag)	
mts_control inter	
...	

- e. Thiết kế cú pháp lấy làm ví dụ dưới đây

<u>if(sps_isp_enabled_flag)</u>	<u>-</u>
<u>sps_isp_non_dct2_enabled_flag</u>	<u>u(1)</u>
...	<u>-</u>
<u>sps_intra_mts_selection</u>	<u>ue(v)</u>
<u>sps_inter_mts_selection</u>	<u>ue(v)</u>

Xác định sử dụng công cụ mã hóa X

42. Việc xác định liệu và/hoặc cách thức kích hoạt công cụ mã X có thể phụ thuộc vào chiều rộng và/hoặc chiều cao của ảnh đang xem xét của một hoặc nhiều ảnh tham chiếu và/hoặc ảnh hiện tại.

- a. Chiều rộng và/hoặc chiều cao của ảnh đang xem xét của một hoặc nhiều ảnh tham chiếu và/hoặc ảnh hiện tại có thể được chỉnh sửa để ra quyết định.
- b. Ảnh đang xem xét có thể được định nghĩa bằng cửa sổ tương hợp or cửa sổ chia tỷ lệ.

- i. Ảnh đang xem xét có thể là toàn bộ ảnh.
- c. Trong một ví dụ, liệu và/hoặc cách thức kích hoạt công cụ mã X có thể phụ thuộc vào chiều rộng của ảnh trừ 1 hoặc nhiều độ lệch theo chiều ngang và/hoặc chiều cao của ảnh trừ độ lệch theo chiều dọc.
 - i. Trong một ví dụ, độ lệch ngang có thể được định nghĩa là `scaling_win_left_offset`.
 - ii. Trong một ví dụ, độ lệch dọc có thể được định nghĩa là `scaling_win_top_offset`.
 - iii. Trong một ví dụ, độ lệch ngang có thể được định nghĩa là $(\text{scaling_win_right_offset} + \text{scaling_win_left_offset})$.
 - iv. Trong một ví dụ, độ lệch dọc có thể được định nghĩa là $(\text{scaling_win_bottom_offset} + \text{scaling_win_top_offset})$.
 - v. Trong một ví dụ, độ lệch ngang có thể được định nghĩa là $\text{SubWidthC} * (\text{scaling_win_right_offset} + \text{scaling_win_left_offset})$.
 - vi. Trong một ví dụ, độ lệch dọc có thể được định nghĩa là $\text{SubHeightC} * (\text{scaling_win_bottom_offset} + \text{scaling_win_top_offset})$.
- d. Trong một ví dụ, nếu ít nhất một trong hai ảnh tham chiếu xem xét có độ phân giải khác (chiều rộng hoặc chiều cao) với ảnh hiện tại, công cụ mã X bị vô hiệu hóa.
 - i. Theo cách khác, nếu ít nhất một trong hai ảnh tham chiếu đầu ra có kích thước (chiều rộng hoặc chiều cao) lớn hơn kích thước của ảnh hiện tại, công cụ mã X bị vô hiệu hóa.
- e. Trong một ví dụ, nếu một ảnh tham chiếu đang xem xét cho danh sách ảnh tham chiếu L có độ phân giải khác với ảnh hiện tại, công cụ mã X bị vô hiệu hóa cho danh sách ảnh tham chiếu L.
 - i. Theo cách khác, nếu một ảnh tham chiếu đang xem xét cho danh sách ảnh tham chiếu L có kích thước (chiều rộng hoặc chiều cao) lớn hơn kích thước của ảnh hiện tại, công cụ mã X bị vô hiệu hóa đối với danh sách ảnh tham chiếu L.

- f. Trong một ví dụ, nếu hai ảnh tham chiếu đang xem xét của hai danh sách ảnh tham chiếu có các độ phân giải khác nhau, công cụ mã có thể bị vô hiệu hóa.
 - i. Theo cách khác, các chỉ báo của công cụ mã có thể được báo hiệu có điều kiện theo các độ phân giải.
 - ii. Theo cách khác, báo hiệu các chỉ báo của công cụ mã có thể được bỏ qua.
- g. Trong một ví dụ, nếu hai ảnh tham chiếu đang xem xét của hai ứng viên hợp nhất được sử dụng để dẫn xuất ứng viên hợp nhất theo cặp thứ nhất đối với ít nhất một danh sách ảnh tham chiếu, công cụ mã có thể bị vô hiệu hóa, chẳng hạn, ứng viên hợp nhất theo cặp thứ nhất được đánh dấu là không khả dụng.
 - i. Theo cách khác, nếu hai ảnh tham chiếu đang xem xét của hai ứng viên hợp nhất được sử dụng để dẫn xuất ứng viên hợp nhất theo cặp thứ nhất cho cả hai danh sách ảnh tham chiếu, công cụ mã có thể bị vô hiệu hóa, chẳng hạn, ứng viên hợp nhất theo cặp thứ nhất được đánh dấu là không khả dụng.
- h. Trong một ví dụ, quá trình giải mã của công cụ mã có thể được chỉnh sửa có xem xét kích thước ảnh.
 - i. Trong một ví dụ, việc dẫn xuất MVD đối với danh sách ảnh tham chiếu khác (chẳng hạn, danh sách 1) trong SMVD có thể dựa trên các hiệu số độ phân giải (chẳng hạn, các hệ số chia tỷ lệ) của ít nhất một trong hai ảnh tham chiếu SMVD mục tiêu.
 - ii. Trong một ví dụ, việc dẫn xuất ứng viên hợp nhất theo cặp có thể dựa trên các độ lệch độ phân giải (chẳng hạn, các hệ số chia tỷ lệ) của ít nhất một trong hai ảnh tham chiếu được liên kết với hai ảnh tham chiếu, chẳng hạn, trung bình có trọng số tuyến tính có thể được áp dụng thay vì các trọng số bằng nhau.
- i. Trong một ví dụ, X có thể là:
 - i. DMVR/BDOF/PROF/SMVD/MMVD/các công cụ mã hóa khác tinh lọc chuyển động/dự báo ở phía bộ giải mã.

- ii. TMVP/các công cụ mã hóa khác dựa vào thông tin chuyển động thời gian.
- iii. MTS hoặc các công cụ mã hóa biến đổi khác
- iv. CC-ALF
- v. TPM
- vi. GEO
- vii. Bộ lọc nội suy có thể thay đổi (chẳng hạn, bộ lọc nội suy tùy chọn để bù chuyển động nửa điểm ảnh)
- viii. Quá trình kết hợp trong TPM/GEO/các công cụ mã hóa khác tách một khối thành nhiều phân vùng.
- ix. Công cụ mã hóa dựa vào thông tin được lưu trữ trong ảnh khác với ảnh hiện tại
- x. ứng viên hợp nhất theo cặp (khi các điều kiện cụ thể liên quan đến độ phân giải không được thỏa mãn, ứng viên hợp nhất theo cặp không được tạo)
- xi. Dự báo kép có các trọng số mức CU (BCW).
- xii. Dự báo có trọng số.
- xiii. Dự báo Afin.
- xiv. Độ phân giải MV thích ứng (AMVR)

43. Liệu và/hoặc cách thức báo hiệu sử dụng công cụ mã hóa có thể phụ thuộc vào chiều rộng và/hoặc chiều cao của ảnh đang xem xét của một hoặc nhiều ảnh tham chiếu và/hoặc ảnh hiện tại.

- a. Chiều rộng và/hoặc chiều cao của ảnh đang xem xét của một hoặc nhiều ảnh tham chiếu và/hoặc ảnh hiện tại có thể được chỉnh sửa để ra quyết định.
- b. Ảnh đang xem xét có thể được định nghĩa bởi cửa sổ tương hợp or cửa sổ chia tỷ lệ như được định nghĩa trong JVET-P0590.
- xv. Ảnh đang xem xét có thể là toàn bộ ảnh.
- c. Trong một ví dụ, X có thể là AMVR.
- d. Trong một ví dụ, X có thể là phương pháp hợp nhất có các hiệu số MV (MMVD).

- xvi. Trong một ví dụ, việc tạo các chỉ số tham chiếu MVD đối xứng có thể phụ thuộc vào các độ phân giải ảnh/các chỉ báo của các trường hợp RPR đối với các ảnh tham chiếu khác nhau.
 - e. Trong một ví dụ, X có thể là phương pháp SMVD.
 - f. Trong một ví dụ, X có thể là QT/BT/TT hoặc các loại phân vùng khác.
 - g. Trong một ví dụ, X có thể là dự báo kép có các trọng số mức CU (BCW).
 - h. Trong một ví dụ, X có thể là dự báo có trọng số.
 - i. Trong một ví dụ, X có thể là dự báo Afin.
 - j. Trong một ví dụ, liệu báo hiệu chỉ báo sử dụng độ chính xác vector chuyển động nửa pel/bộ lọc nội suy có thể chuyển đổi có thể phụ thuộc vào thông tin độ phân giải/liệu RPR được kích hoạt cho khối hiện tại.
 - k. Trong một ví dụ, báo hiệu **amvr_precision_idx** có thể phụ thuộc vào thông tin độ phân giải/liệu RPR được kích hoạt cho khối hiện tại.
 - l. Trong một ví dụ, báo hiệu **sym_mvd_flag/mmvd_merge_flag** có thể phụ thuộc vào thông tin độ phân giải/liệu RPR được kích hoạt cho khối hiện tại.
 - m. Dòng bit tương hợp sẽ thỏa mãn việc độ chính xác MV và/hoặc MVD $\frac{1}{2}$ pel (chẳng hạn, bộ lọc nội suy tùy chọn/bộ lọc nội suy có thể thay đổi) không được phép khi chiều rộng và/hoặc chiều cao của ảnh đang xem xét của một hoặc nhiều ảnh tham chiếu khác với kích thước của ảnh đầu ra hiện tại.
44. Đề xuất rằng AMVR có độ chính xác MV và/hoặc MVD $\frac{1}{2}$ pel (hoặc bộ lọc nội suy tùy chọn/bộ lọc nội suy có thể thay đổi) có thể vẫn được kích hoạt cho khối trong RPR.
- a. Theo cách khác, ngoài ra, bộ lọc nội suy khác nhau có thể được áp dụng cho các khối có các độ chính xác $\frac{1}{2}$ pel hoặc độ chính xác khác.
45. Kiểm tra có điều kiện có các độ phân giải giống nhau/khác nhau trong các đề mục nêu trên có thể được thay thế bằng cách bổ sung cờ cho ảnh tham chiếu và kiểm tra các cờ được liên kết với các ảnh tham chiếu.

- a. Trong một ví dụ, quá trình gán cờ là true hoặc false (chẳng hạn, để chỉ báo liệu ảnh tham chiếu là trường hợp RPR hoặc trường hợp phi RPR) có thể được gọi trong quá trình tạo danh sách ảnh tham chiếu.

xvii. Chẳng hạn, điều kiện sau có thể được áp dụng:

fRefWidth được gán bằng PicOutputWidthL của ảnh tham chiếu RefPicList[i][j] trong mẫu độ sáng, trong đó PicOutputWidthL là chiều rộng của ảnh đang xem xét của ảnh tham chiếu.

fRefWidth được gán bằng PicOutputHeightL của ảnh tham chiếu RefPicList[i][j] trong mẫu độ sáng, trong đó PicOutputHeightL là chiều cao của ảnh đang xem xét của ảnh tham chiếu.

RefPicScale[i][j][0] =

((fRefWidth << 14) + (PicOutputWidthL >> 1)) / PicOutputWidthL, trong đó

PicOutputWidthL là chiều rộng của ảnh đang xem xét của ảnh hiện tại.

RefPicScale[i][j][1] =

((fRefHeight << 14) + (PicOutputHeightL >> 1)) / PicOutputHeightL, trong đó PicOutputHeightL là chiều cao của ảnh

đang xem xét của ảnh hiện tại

RefPicIsScaled[i][j] = (RefPicScale[i][j][0] != (1 << 14))

|| (RefPicScale[i][j][1] != (1 << 14))

trong đó RefPicList[i][j] là ảnh tham chiếu thứ j trong danh sách ảnh tham chiếu i.

- b. Trong một ví dụ, khi RefPicIsScaled[0][refIdxL0] không bằng 0 hoặc RefPicIsScaled[1][refIdxL1] không bằng 0, công cụ tạo mã X (chẳng hạn, DMVR/BDOF/SMVD/MMVD/SMVD/PROF/các trường hợp khác được nêu trong các đề mục trên) có thể bị vô hiệu hóa.

- c. Trong một ví dụ, khi cả RefPicIsScaled[0][refIdxL0] và RefPicIsScaled[1][refIdxL1] không bằng 0, công cụ tạo mã X (chẳng

hạn, DMVR/BDOF/SMVD/MMVD/SMVD/PROF/các trường hợp khác được nêu trong các đề mục trên) có thể bị vô hiệu hóa.

- d. Trong một ví dụ, khi `RefPicIsScaled[0][refIdxL0]` không bằng 0, công cụ tạo mã X (chẳng hạn, PROF hoặc các trường hợp khác được nêu trong các đề mục trên) có thể bị vô hiệu hóa cho danh sách ảnh tham chiếu 0.
- e. Trong một ví dụ, khi `RefPicIsScaled[1][refIdxL1]` không bằng 0, công cụ tạo mã X (chẳng hạn, PROF hoặc các trường hợp khác được nêu trong các đề mục trên) có thể bị vô hiệu hóa cho danh sách ảnh tham chiếu 1.

46. SAD và/hoặc ngưỡng được sử dụng bởi BDOF/DMVR có thể phụ thuộc vào chiều sâu bit.

- f. Trong một ví dụ, giá trị SAD được tính toán có thể được dịch trước bởi hàm chiều sâu bit trước khi được sử dụng để được so sánh với ngưỡng.
- g. Trong một ví dụ, giá trị SAD được tính toán có thể được so sánh trực tiếp với ngưỡng được chỉnh sửa có thể phụ thuộc vào hàm chiều sâu bit.

5. Các phương án thực hiện bổ sung

Dưới đây, các thay đổi văn bản được thể hiện trên phông chữ in nghiêng đậm gạch chân.

5.1. Phương án thực hiện về các ràng buộc trên cửa sổ tương hợp

`conf_win_left_offset`, `conf_win_right_offset`, `conf_win_top_offset`, và `conf_win_bottom_offset` xác định các mẫu của các ảnh trong CVS được xuất ra từ quá trình giải mã, liên quan đến vùng chữ nhật được xác định trong các tọa độ ảnh để xuất ra. Khi `conformance_window_flag` bằng 0, các giá trị của `conf_win_left_offset`, `conf_win_right_offset`, `conf_win_top_offset`, và `conf_win_bottom_offset` được suy ra bằng 0.

Cửa sổ cắt tương hợp chứa các mẫu độ sáng có các tọa độ ảnh ngang từ `SubWidthC * conf_win_left_offset` đến `pic_width_in_luma_samples - (SubWidthC * conf_win_right_offset + 1)` và các

tọa độ ảnh dọc từ $\text{SubHeightC} * \text{conf_win_top_offset}$ đến $\text{pic_height_in_luma_samples} - (\text{SubHeightC} * \text{conf_win_bottom_offset} + 1)$, bao gồm hai giá trị đầu mút.

Giá trị của $\text{SubWidthC} * (\text{conf_win_left_offset} + \text{conf_win_right_offset})$ sẽ nhỏ hơn $\text{pic_width_in_luma_samples}$, và giá trị của $\text{SubHeightC} * (\text{conf_win_top_offset} + \text{conf_win_bottom_offset})$ sẽ nhỏ hơn $\text{pic_height_in_luma_samples}$.

Các biến PicOutputWidthL và PicOutputHeightL được dẫn xuất như sau:

$$\text{PicOutputWidthL} = \text{pic_width_in_luma_samples} - \quad (7-43)$$

$$\text{SubWidthC} * (\text{conf_win_right_offset} + \text{conf_win_left_offset})$$

$$\text{PicOutputHeightL} = \text{pic_height_in_pic_size_units} -$$

$$(7-44) \quad \text{SubHeightC} * (\text{conf_win_bottom_offset} +$$

$$\text{conf_win_top_offset})$$

Khi ChromaArrayType không bằng 0, các mẫu được xác định tương ứng của hai mảng sắc độ là các mẫu có các tọa độ ảnh ($x/\text{SubWidthC}$, $y/\text{SubHeightC}$), trong đó (x , y) là các tọa độ ảnh của mẫu độ sáng được xác định.

Giả sử $\text{refPicOutputWidthL}$ và $\text{refPicOutputHeightL}$ lần lượt là PicOutputWidthL và PicOutputHeightL , của ảnh tham chiếu của ảnh hiện tại viên dẫn đến PPS này. Tương hợp dòng bit yêu cầu rằng tất cả các điều kiện sau được thỏa mãn:

- $\text{PicOutputWidthL} * 2$ sẽ lớn hơn hoặc bằng $\text{refPicOutputWidthL}$.
- $\text{PicOutputHeightL} * 2$ sẽ lớn hơn hoặc bằng $\text{refPicOutputHeightL}$.
- PicOutputWidthL sẽ nhỏ hơn hoặc bằng $\text{refPicOutputWidthL} * 8$.
- PicOutputHeightL sẽ nhỏ hơn hoặc bằng $\text{refPicOutputHeightL} * 8$.

5.2. Phương án thực hiện 1 của dẫn xuất vị trí mẫu tham chiếu

8.5.6.3.1 Tổng quát

...

Biến fRefWidth được gán bằng PicOutputWidthL của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefHeight được gán bằng PicOutputHeightL của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefLeftOff$ được gán bằng $conf\ win\ left\ offset$ của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefTopOff$ được gán bằng $conf\ win\ top\ offset$ của ảnh tham chiếu trong mẫu độ sáng.

Vector chuyển động $mvLX$ được gán bằng $(refMvLX - mvOffset)$.

– Nếu $cIdx$ bằng 0, điều kiện sau áp dụng:

– Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$hori_scale_fp = ((fRefWidth \ll 14) + (PicOutputWidthL \gg 1)) / PicOutputWidthL \quad (8-753)$$

$$vert_scale_fp = ((fRefHeight \ll 14) + (PicOutputHeightL \gg 1)) / PicOutputHeightL \quad (8-754)$$

– Giả sử $(xIntL, yIntL)$ là vị trí độ sáng theo các đơn vị mẫu toàn phần và $(xFracL, yFracL)$ là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu $refPicLX$.

– Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbIntL, ySbIntL)$ được gán bằng $(xSb + (mvLX[0] \gg 4), ySb + (mvLX[1] \gg 4))$.

– Đối với mỗi vị trí mẫu độ sáng $(x_L = 0..sbWidth - 1 + brdExtSize, y_L = 0..sbHeight - 1 + brdExtSize)$ bên trong mảng mẫu độ sáng dự báo $predSamplesLX$, giá trị mẫu độ sáng dự báo tương ứng $predSamplesLX[x_L][y_L]$ được dẫn xuất như sau:

– Giả sử $(refxSbL, refySbL)$ và $(refxL, refyL)$ là các vị trí độ sáng được trở đến bởi vector chuyển động $(refMvLX[0], refMvLX[1])$ theo các đơn vị 1/16 mẫu. Các biến $refxSbL, refxL, refySbL$, và $refyL$ được dẫn xuất như sau:

$$refxSbL = (((xSb - \underline{conf\ win\ left\ offset}) \ll 4) + refMvLX[0]) * hori_scale_fp \quad (8-755)$$

$$refxL = ((Sign(refxSb) * ((Abs(refxSb) + 128) \gg 8) + x_L * ((hori_scale_fp + 8) \gg 4)) + 32) \gg 6 \quad (8-756)$$

$$refySbL = (((ySb - \underline{conf\ win\ top\ offset}) \ll 4) + refMvLX[1]) * vert_scale_fp \quad (8-757)$$

$$\text{refy}_L = ((\text{Sign}(\text{refySb}) * ((\text{Abs}(\text{refySb}) + 128) \gg 8) + y_L * ((\text{vert_scale_fp} + 8) \gg 4)) + 32) \gg 6 \quad (8-758)$$

$$\text{refx}_L = \text{refx}_L + (\text{fRefLeftOff} \ll 4).$$

$$\text{refy}_L = \text{refy}_L + (\text{fRefTopOff} \ll 4).$$

- Các biến xInt_L , yInt_L , xFrac_L và yFrac_L được dẫn xuất như sau:

$$\text{xInt}_L = \text{refx}_L \gg 4 \quad (8-759)$$

$$\text{yInt}_L = \text{refy}_L \gg 4 \quad (8-760)$$

$$\text{xFrac}_L = \text{refx}_L \& 15 \quad (8-761)$$

$$\text{yFrac}_L = \text{refy}_L \& 15 \quad (8-762)$$

- Nếu bdoFlag bằng TRUE hoặc ($\text{sps_affine_prof_enabled_flag}$ bằng TRUE và $\text{inter_affine_flag}[\text{xSb}][\text{ySb}]$ bằng TRUE), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 với $(\text{xInt}_L + (\text{xFrac}_L \gg 3) - 1)$, $\text{yInt}_L + (\text{yFrac}_L \gg 3) - 1$ và refPicLX làm đầu vào.

- x_L bằng 0.
- x_L bằng $\text{sbWidth} + 1$.
- y_L bằng 0.
- y_L bằng $\text{sbHeight} + 1$.

- Ngược lại, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(\text{xInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $\text{yInt}_L - (\text{brdExtSize} > 0 ? 1 : 0)$, $(\text{xFrac}_L, \text{yFrac}_L)$, $(\text{xSbInt}_L, \text{ySbInt}_L)$, refPicLX , hpelIdx , sbWidth , sbHeight và (xSb, ySb) làm đầu vào.

- Ngược lại (cIdx không bằng 0), điều kiện sau áp dụng:
 - Giả sử $(\text{xInt}_C, \text{yInt}_C)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(\text{xFrac}_C, \text{yFrac}_C)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu refPicLX .

- Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbIntC, ySbIntC)$ được gán bằng $((xSb / SubWidthC) + (mvLX[0] \gg 5), (ySb / SubHeightC) + (mvLX[1] \gg 5))$.
- Đối với mỗi vị trí mẫu sắc độ $(xC = 0..sbWidth - 1, yC = 0..sbHeight - 1)$ bên trong các mảng mẫu sắc độ dự báo $predSamplesLX$, giá trị mẫu sắc độ dự báo tương ứng $predSamplesLX[xC][yC]$ được dẫn xuất như sau:
 - Giả sử $(refxSbc, refySbc)$ và $(refxC, refyC)$ là các vị trí sắc độ được trở đến bởi vector chuyển động $(mvLX[0], mvLX[1])$ theo các đơn vị $1/32$ mẫu. Các biến $refxSbc, refySbc, refxC$ và $refyC$ được dẫn xuất như sau:

$$refxSbc = (((xSb - conf_win_left_offset) / SubWidthC \ll 5) + mvLX[0]) * hori_scale_fp \quad (8-763)$$

$$refxC = ((Sign(refxSbc) * ((Abs(refxSbc) + 256) \gg 9) + xC * ((hori_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-764)$$

$$refySbc = (((ySb - conf_win_top_offset) / SubHeightC \ll 5) + mvLX[1]) * vert_scale_fp \quad (8-765)$$

$$refyC = ((Sign(refySbc) * ((Abs(refySbc) + 256) \gg 9) + yC * ((vert_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-766)$$

$$- \quad \underline{refxC = refxC + (fRefLeftOff / SubWidthC \ll 5).}$$

$$- \quad \underline{refyC = refyC + (fRefTopOff / SubHeightC \ll 5).}$$

- Các biến $xIntC, yIntC, xFracc$ và $yFracc$ được dẫn xuất như sau:

$$xIntC = refxC \gg 5 \quad (8-767)$$

$$yIntC = refyC \gg 5 \quad (8-768)$$

$$xFracc = refxC \& 31 \quad (8-769)$$

$$yFracc = refyC \& 31 \quad (8-770)$$

5.3. Phương án thực hiện 2 của dẫn xuất vị trí mẫu tham chiếu

8.5.6.3.1 Tổng quát

...

Biến $fRefWidth$ được gán bằng $PicOutputWidthL$ của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefHeight$ được gán bằng $PicOutputHeightL$ của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefLeftOff được gán bằng conf win left offset của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefTopOff được gán bằng conf win top offset của ảnh tham chiếu trong mẫu độ sáng.

Vector chuyển động mvLX được gán bằng (refMvLX – mvOffset).

- Nếu cIdx bằng 0, điều kiện sau áp dụng:
 - Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$\text{hori_scale_fp} = ((\text{fRefWidth} \ll 14) + (\text{PicOutputWidthL} \gg 1)) / \text{PicOutputWidthL} \quad (8-753)$$

$$\text{vert_scale_fp} = ((\text{fRefHeight} \ll 14) + (\text{PicOutputHeightL} \gg 1)) / \text{PicOutputHeightL} \quad (8-754)$$
 - Giả sử (xIntL, yIntL) là vị trí độ sáng theo các đơn vị mẫu toàn phần và (xFracL, yFracL) là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu refPicLX.
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (xSbIntL, ySbIntL) được gán bằng (xSb + (mvLX[0] >> 4), ySb + (mvLX[1] >> 4)).
 - Đối với mỗi vị trí mẫu độ sáng (xL = 0..sbWidth – 1 + brdExtSize, yL = 0..sbHeight – 1 + brdExtSize) bên trong mảng mẫu độ sáng dự báo predSamplesLX, giá trị mẫu độ sáng dự báo tương ứng predSamplesLX[xL][yL] được dẫn xuất như sau:
 - Giả sử (refxSbL, refySbL) và (refxL, refyL) là các vị trí độ sáng được trở đến bởi vector chuyển động (refMvLX[0], refMvLX[1]) theo các đơn vị 1/16 mẫu. Các biến refxSbL, refxL, refySbL, và refyL được dẫn xuất như sau:

$$\text{refxSbL} = (((\text{xSb} - \text{conf win left offset}) \ll 4) + \text{refMvLX}[0]) * \text{hori_scale_fp} \quad (8-755)$$

$$\text{refxL} = ((\text{Sign}(\text{refxSb}) * ((\text{Abs}(\text{refxSb}) + 128) \gg 8) + \text{xL} * ((\text{hori_scale_fp} + 8) \gg 4)) + (\text{fRefLeftOff} \ll 10) + 32) \gg 6 \quad (8-756)$$

$$\text{refySbL} = (((\text{ySb} - \text{conf win top offset}) \ll 4) + \text{refMvLX}[1]) * \text{vert_scale_fp} \quad (8-757)$$

$$\begin{aligned} \text{refy}_L &= ((\text{Sign}(\text{refySb}) * ((\text{Abs}(\text{refySb}) + 128) \gg 8) + y_L * \\ &((\text{vert_scale_fp} + 8) \gg 4)) + \underline{\text{fRefTopOff} \ll 10} + 32) \gg 6 \quad (8-758) \end{aligned}$$

- Các biến xInt_L , yInt_L , xFrac_L và yFrac_L được dẫn xuất như sau:

$$\text{xInt}_L = \text{refx}_L \gg 4 \quad (8-759)$$

$$\text{yInt}_L = \text{refy}_L \gg 4 \quad (8-760)$$

$$\text{xFrac}_L = \text{refx}_L \& 15 \quad (8-761)$$

$$\text{yFrac}_L = \text{refy}_L \& 15 \quad (8-762)$$

- Nếu bdoFlag bằng TRUE hoặc ($\text{sps_affine_prof_enabled_flag}$ bằng TRUE và $\text{inter_affine_flag}[\text{xSb}][\text{ySb}]$ bằng TRUE), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{xL}][\text{yL}]$ được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 với $(\text{xInt}_L + (\text{xFrac}_L \gg 3) - 1)$, $\text{yInt}_L + (\text{yFrac}_L \gg 3) - 1$ và refPicLX làm đầu vào.
 - xL bằng 0.
 - xL bằng $\text{sbWidth} + 1$.
 - yL bằng 0.
 - yL bằng $\text{sbHeight} + 1$.
- Ngược lại, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{xL}][\text{yL}]$ được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(\text{xInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $(\text{yInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $(\text{xFrac}_L, \text{yFrac}_L)$, $(\text{xSbInt}_L, \text{ySbInt}_L)$, refPicLX , hpelIdx , sbWidth , sbHeight và (xSb, ySb) làm đầu vào.
- Ngược lại (cIdx không bằng 0), điều kiện sau áp dụng:
 - Giả sử $(\text{xInt}_C, \text{yInt}_C)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(\text{xFrac}_C, \text{yFrac}_C)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu refPicLX .
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(\text{xSbInt}_C, \text{ySbInt}_C)$ được gán bằng $((\text{xSb} / \text{SubWidth}_C) + (\text{mvLX}[0] \gg 5))$, $(\text{ySb} / \text{SubHeight}_C) + (\text{mvLX}[1] \gg 5))$.

- Đối với mỗi vị trí mẫu sắc độ ($x_C = 0..sbWidth - 1$, $y_C = 0.. sbHeight - 1$) bên trong các mảng mẫu sắc độ dự báo $predSamplesLX$, giá trị mẫu sắc độ dự báo tương ứng $predSamplesLX[x_C][y_C]$ được dẫn xuất như sau:
- Giả sử $(refxSbc, refySbc)$ và $(refxc, refyc)$ là các vị trí sắc độ được trở đến bởi vector chuyển động $(mvLX[0], mvLX[1])$ theo các đơn vị 1/32 mẫu. Các biến $refxSbc$, $refySbc$, $refxc$ và $refyc$ được dẫn xuất như sau:

$$refxSbc = (((xSb - conf\ win\ left\ offset) / SubWidthC << 5) + mvLX[0]) * hori_scale_fp \quad (8-763)$$

$$refxc = ((Sign(refxSbc) * ((Abs(refxSbc) + 256) >> 9) + x_C * ((hori_scale_fp + 8) >> 4) + 16) >> 5) \quad (8-764)$$

$$refySbc = (((ySb - conf\ win\ top\ offset) / SubHeightC << 5) + mvLX[1]) * vert_scale_fp \quad (8-765)$$

$$refyc = ((Sign(refySbc) * ((Abs(refySbc) + 256) >> 9) + y_C * ((vert_scale_fp + 8) >> 4) + 16) >> 5) \quad (8-766)$$

$$-refxc = refxc + (fRefLeftOff / SubWidthC << 5).$$

$$-refyc = refyc + (fRefTopOff / SubHeightC << 5).$$

- Các biến $xIntc$, $yIntc$, $xFracc$ và $yFracc$ được dẫn xuất như sau:

$$xIntc = refxc >> 5 \quad (8-767)$$

$$yIntc = refyc >> 5 \quad (8-768)$$

$$xFracc = refyc \& 31 \quad (8-769)$$

$$yFracc = refyc \& 31 \quad (8-770)$$

5.4. Phương án thực hiện 3 của dẫn xuất vị trí mẫu tham chiếu

8.5.6.3.1 Tổng quát

...

Biến $fRefWidth$ được gán bằng $PicOutputWidthL$ của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefHeight$ được gán bằng $PicOutputHeightL$ của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefLeftOff$ được gán bằng $conf\ win\ left\ offset$ của ảnh tham chiếu trong mẫu độ sáng.

Biến $fRefTopOff$ được gán bằng $conf\ win\ top\ offset$ của ảnh tham chiếu trong mẫu độ sáng.

Vector chuyển động mvLX được gán bằng (refMvLX – mvOffset).

- Nếu cIdx bằng 0, điều kiện sau áp dụng:
 - Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$\text{hori_scale_fp} = ((\text{fRefWidth} \ll 14) + (\text{PicOutputWidthL} \gg 1)) / \text{PicOutputWidthL} \quad (8-753)$$

$$\text{vert_scale_fp} = ((\text{fRefHeight} \ll 14) + (\text{PicOutputHeightL} \gg 1)) / \text{PicOutputHeightL} \quad (8-754)$$
 - Giả sử (xIntL, yIntL) là vị trí độ sáng theo các đơn vị mẫu toàn phần và (xFracL, yFracL) là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu refPicLX.
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (xSbIntL, ySbIntL) được gán bằng (xSb + (mvLX[0] >> 4), ySb + (mvLX[1] >> 4)).
 - Đối với mỗi vị trí mẫu độ sáng (xL = 0..sbWidth – 1 + brdExtSize, yL = 0..sbHeight – 1 + brdExtSize) bên trong mảng mẫu độ sáng dự báo predSamplesLX, giá trị mẫu độ sáng dự báo tương ứng predSamplesLX[xL][yL] được dẫn xuất như sau:
 - Giả sử (refxSbL, refySbL) và (refxL, refyL) là các vị trí độ sáng được trở đến bởi vector chuyển động (refMvLX[0], refMvLX[1]) theo các đơn vị 1/16 mẫu. Các biến refxSbL, refxL, refySbL, và refyL được dẫn xuất như sau:

$$\text{refxSbL} = (((\text{xSb} - \text{conf win left offset}) \ll 4) + \text{refMvLX}[0]) * \text{hori_scale_fp} \quad (8-755)$$

$$\text{refxL} = ((\text{Sign}(\text{refxSb}) * ((\text{Abs}(\text{refxSb}) + 128) \gg 8) + \text{xL} * ((\text{hori_scale_fp} + 8) \gg 4)) + (\text{fRefLeftOff} \ll 10) + 32) \gg 6 \quad (8-756)$$

$$\text{refySbL} = (((\text{ySb} - \text{conf win top offset}) \ll 4) + \text{refMvLX}[1]) * \text{vert_scale_fp} \quad (8-757)$$

$$\text{refyL} = ((\text{Sign}(\text{refySb}) * ((\text{Abs}(\text{refySb}) + 128) \gg 8) + \text{yL} * ((\text{vert_scale_fp} + 8) \gg 4)) + (\text{fRefTopOff} \ll 10) + 32) \gg 6 \quad (8-758)$$

- Các biến xIntL, yIntL, xFracL và yFracL được dẫn xuất như sau:

$$\text{xIntL} = \text{refxL} \gg 4 \quad (8-759)$$

$$\text{yIntL} = \text{refyL} \gg 4 \quad (8-760)$$

$$xFrac_L = refx_L \& 15 \quad (8-761)$$

$$yFrac_L = refy_L \& 15 \quad (8-762)$$

- Nếu `bdofFlag` bằng `TRUE` hoặc (`sps_affine_prof_enabled_flag` bằng `TRUE` và `inter_affine_flag[xSb][ySb]` bằng `TRUE`), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo `predSamplesLX[xL][yL]` được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 with $(xInt_L + (xFrac_L \gg 3) - 1)$, $yInt_L + (yFrac_L \gg 3) - 1$ and `refPicLX` làm đầu vào.
 - x_L bằng 0.
 - x_L bằng `sbWidth + 1`.
 - y_L bằng 0.
 - y_L bằng `sbHeight + 1`.
- Ngược lại, giá trị mẫu độ sáng dự báo `predSamplesLX[xL][yL]` được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(xInt_L - (brdExtSize > 0 ? 1 : 0))$, $yInt_L - (brdExtSize > 0 ? 1 : 0)$, $(xFrac_L, yFrac_L)$, $(xSbInt_L, ySbInt_L)$, `refPicLX`, `hpelIdx`, `sbWidth`, `sbHeight` và (xSb, ySb) làm đầu vào.
- Ngược lại (`cIdx` không bằng 0), điều kiện sau áp dụng:
 - Giả sử $(xInt_C, yInt_C)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(xFrac_C, yFrac_C)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu `refPicLX`.
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbInt_C, ySbInt_C)$ được gán bằng $((xSb / SubWidth_C) + (mvLX[0] \gg 5), (ySb / SubHeight_C) + (mvLX[1] \gg 5))$.
 - Đối với mỗi vị trí mẫu sắc độ ($x_C = 0..sbWidth - 1$, $y_C = 0..sbHeight - 1$) bên trong các mảng mẫu sắc độ dự báo `predSamplesLX`, giá trị mẫu sắc độ dự báo tương ứng `predSamplesLX[xC][yC]` được dẫn xuất như sau:

- Giả sử (refxSb_C, refySb_C) và (refxC, refyC) là các vị trí sắc độ được trở đến bởi vector chuyển động (mvLX[0], mvLX[1]) theo các đơn vị 1/32 mẫu. Các biến refxSb_C, refySb_C, refxC and refyC được dẫn xuất như sau:

$$\text{refxSb}_C = (((xSb - \text{conf win left offset}) / \text{SubWidthC} \ll 5) + \text{mvLX}[0]) * \text{hori_scale_fp} \quad (8-763)$$

$$\begin{aligned} \text{refxC} = & ((\text{Sign}(\text{refxSb}_C) * ((\text{Abs}(\text{refxSb}_C) + 256) \gg 9) \\ & + xC * ((\text{hori_scale_fp} + 8) \gg 4)) + (\text{fRefLeftOff} / \text{SubWidthC} \ll 10) + 16) \gg 5 \end{aligned} \quad (8-764)$$

$$\text{refySb}_C = (((ySb - \text{conf win top offset}) / \text{SubHeightC} \ll 5) + \text{mvLX}[1]) * \text{vert_scale_fp} \quad (8-765)$$

$$\begin{aligned} \text{refyC} = & ((\text{Sign}(\text{refySb}_C) * ((\text{Abs}(\text{refySb}_C) + 256) \gg 9) \\ & + yC * ((\text{vert_scale_fp} + 8) \gg 4)) + (\text{fRefTopOff} / \text{SubHeightC} \ll 10) + 16) \gg 5 \end{aligned} \quad (8-766)$$

- Các biến xInt_C, yInt_C, xFrac_C và yFrac_C được dẫn xuất như sau:

$$xInt_C = \text{refxC} \gg 5 \quad (8-767)$$

$$yInt_C = \text{refyC} \gg 5 \quad (8-768)$$

$$xFrac_C = \text{refxC} \& 31 \quad (8-769)$$

$$yFrac_C = \text{refyC} \& 31 \quad (8-770)$$

5.5. Phương án thực hiện 1 của cắt vị trí mẫu tham chiếu

8.5.6.3.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xSb, ySb) xác định mẫu trên bên trái của khối phụ mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến sbWidth xác định chiều rộng của khối phụ mã hiện tại,
- biến sbHeight xác định chiều cao của khối phụ mã hiện tại,
- độ lệch vector chuyển động mvOffset,
- vector chuyển động được tính lọc refMvLX,
- mảng mẫu ảnh tham chiếu được chọn refPicLX,
- chỉ số bộ lọc nội suy nửa mẫu hpellIdx,
- cờ luồng quang hai hướng bdofFlag,
- biến cIdx xác định chỉ số thành phần màu sắc của khối hiện tại.

Các đầu ra của quá trình này là:

- mảng $(sbWidth + brdExtSize) \times (sbHeight + brdExtSize)$ `predSamplesLX` của các giá trị mẫu dự báo.

Kích thước mở rộng đường biên khỏi dự báo `brdExtSize` được dẫn xuất như sau:

$$brdExtSize = (bdofFlag \mid (inter_affine_flag[xSb][ySb] \&\& \\ sps_affine_prof_enabled_flag)) ? 2 : 0(8-752)$$

Biến `fRefWidth` được gán bằng `PicOutputWidthL` của ảnh tham chiếu trong mẫu độ sáng.

Biến `fRefHeight` được gán bằng `PicOutputHeightL` của ảnh tham chiếu trong mẫu độ sáng.

Vectơ chuyển động `mvLX` được gán bằng $(refMvLX - mvOffset)$.

- Nếu `cIdx` bằng 0, điều kiện sau áp dụng:
 - Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$\begin{aligned} hori_scale_fp &= ((fRefWidth \ll 14) + (PicOutputWidthL \\ &\gg 1)) / PicOutputWidthL \quad (8-753) \\ vert_scale_fp &= ((fRefHeight \ll 14) + (PicOutputHeightL \\ &\gg 1)) / PicOutputHeightL \quad (8-754) \end{aligned}$$
 - Giả sử $(xIntL, yIntL)$ là vị trí độ sáng theo các đơn vị mẫu toàn phần và $(xFracL, yFracL)$ là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu `refPicLX`.
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbIntL, ySbIntL)$ được gán bằng $(xSb + (mvLX[0] \gg 4), ySb + (mvLX[1] \gg 4))$.
 - Đối với mỗi vị trí mẫu độ sáng $(x_L = 0..sbWidth - 1 + brdExtSize, y_L = 0..sbHeight - 1 + brdExtSize)$ bên trong mảng mẫu độ sáng dự báo `predSamplesLX`, giá trị mẫu độ sáng dự báo tương ứng `predSamplesLX[x_L][y_L]` được dẫn xuất như sau:
 - Giả sử $(refxSb_L, refySb_L)$ và $(refx_L, refy_L)$ là các vị trí độ sáng được trở đến bởi vectơ chuyển động $(refMvLX[0], refMvLX[1])$ theo các đơn vị 1/16 mẫu. Các biến `refxSb_L`, `refx_L`, `refySb_L`, và `refy_L` được dẫn xuất như sau:

$$\text{refxSb}_L = ((\text{xSb} \ll 4) + \text{refMvLX}[0]) * \text{hori_scale_fp} \quad (8-755)$$

$$\begin{aligned} \text{refx}_L &= ((\text{Sign}(\text{refxSb}) * ((\text{Abs}(\text{refxSb}) + 128) \gg 8) \\ &\quad + \text{x}_L * ((\text{hori_scale_fp} + 8) \gg 4)) + 32) \gg 6 \end{aligned} \quad (8-756)$$

$$\text{refySb}_L = ((\text{ySb} \ll 4) + \text{refMvLX}[1]) * \text{vert_scale_fp} \quad (8-757)$$

$$\begin{aligned} \text{refy}_L &= ((\text{Sign}(\text{refySb}) * ((\text{Abs}(\text{refySb}) + 128) \gg 8) + \text{y}_L * \\ &\quad ((\text{vert_scale_fp} + 8) \gg 4)) + 32) \gg 6 \end{aligned} \quad (8-758)$$

- Các biến xInt_L , yInt_L , xFrac_L và yFrac_L được dẫn xuất như sau:

$$\text{xInt}_L = \text{Clip3}(0, \text{fRefWidth} - 1, \text{refx}_L \gg 4) \quad (8-759)$$

$$\text{yInt}_L = \text{Clip3}(0, \text{fRefHeight} - 1, \text{refy}_L \gg 4) \quad (8-760)$$

$$\text{xFrac}_L = \text{refx}_L \& 15 \quad (8-761)$$

$$\text{yFrac}_L = \text{refy}_L \& 15 \quad (8-762)$$

- Nếu bdofFlag bằng TRUE hoặc ($\text{sps_affine_prof_enabled_flag}$ bằng TRUE và $\text{inter_affine_flag}[\text{xSb}][\text{ySb}]$ bằng TRUE), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 với $(\text{xInt}_L + (\text{xFrac}_L \gg 3) - 1)$, $\text{yInt}_L + (\text{yFrac}_L \gg 3) - 1$ và refPicLX làm đầu vào.

- x_L bằng 0.
- x_L bằng $\text{sbWidth} + 1$.
- y_L bằng 0.
- y_L bằng $\text{sbHeight} + 1$.

- Ngược lại, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[\text{x}_L][\text{y}_L]$ được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(\text{xInt}_L - (\text{brdExtSize} > 0 ? 1 : 0))$, $\text{yInt}_L - (\text{brdExtSize} > 0 ? 1 : 0)$, $(\text{xFrac}_L, \text{yFrac}_L)$, $(\text{xSbInt}_L, \text{ySbInt}_L)$, refPicLX , hpelIdx , sbWidth , sbHeight và (xSb, ySb) làm đầu vào.

- Ngược lại (cIdx không bằng 0), điều kiện sau áp dụng:
 - Giả sử $(\text{xInt}_C, \text{yInt}_C)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(\text{xFrac}_C, \text{yFrac}_C)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu refPicLX .

- Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (x_{SbIntC} , y_{SbIntC}) được gán bằng $((x_{Sb} / SubWidthC) + (mvLX[0] \gg 5), (y_{Sb} / SubHeightC) + (mvLX[1] \gg 5))$.
- Đối với mỗi vị trí mẫu sắc độ ($x_C = 0..sbWidth - 1$, $y_C = 0..sbHeight - 1$) bên trong các mảng mẫu sắc độ dự báo $predSamplesLX$, giá trị mẫu sắc độ dự báo tương ứng $predSamplesLX[x_C][y_C]$ được dẫn xuất như sau:
 - Giả sử $(refxSb_C, refySb_C)$ và $(refxC, refyC)$ là các vị trí sắc độ được trở đến bởi vector chuyển động $(mvLX[0], mvLX[1])$ theo các đơn vị 1/32 mẫu. Các biến $refxSb_C$, $refySb_C$, $refxC$ và $refyC$ được dẫn xuất như sau:

$$refxSb_C = ((x_{Sb} / SubWidthC \ll 5) + mvLX[0]) * hori_scale_fp \quad (8-763)$$

$$refxC = ((Sign(refxSb_C) * ((Abs(refxSb_C) + 256) \gg 9) + x_C * ((hori_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-764)$$

$$refySb_C = ((y_{Sb} / SubHeightC \ll 5) + mvLX[1]) * vert_scale_fp \quad (8-765)$$

$$refyC = ((Sign(refySb_C) * ((Abs(refySb_C) + 256) \gg 9) + y_C * ((vert_scale_fp + 8) \gg 4)) + 16) \gg 5 \quad (8-766)$$
 - Các biến $xInt_C$, $yInt_C$, $xFrac_C$ and $yFrac_C$ được dẫn xuất như sau:

$$xInt_C = \underline{Clip3(0, fRefWidth / SubWidthC - 1, refxC \gg 5)} \quad (8-767)$$

$$yInt_C = \underline{Clip3(0, fRefHeight / SubHeightC - 1, refyC \gg 5)} \quad (8-768)$$

$$xFrac_C = refxC \& 31 \quad (8-769)$$

$$yFrac_C = refyC \& 31 \quad (8-770)$$
- Giá trị mẫu dự báo $predSamplesLX[x_C][y_C]$ được dẫn xuất bằng cách gọi quá trình được xác định trong mệnh đề 8.5.6.3.4 với $(xInt_C, yInt_C)$, $(xFrac_C, yFrac_C)$, $(xSbIntC, ySbIntC)$, $sbWidth$, $sbHeight$ và **refPicLX** làm đầu vào.

5.6. Phương án thực hiện 2 của cắt vị trí mẫu tham chiếu

8.5.6.3.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí độ sáng (x_{Sb} , y_{Sb}) xác định mẫu trên bên trái của khối phụ mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến $sbWidth$ xác định chiều rộng của khối phụ mã hiện tại,
- biến $sbHeight$ xác định chiều cao của khối phụ mã hiện tại,
- độ lệch vector chuyển động $mvOffset$,

- vectơ chuyển động được tính lọc refMvLX,
- mảng mẫu ảnh tham chiếu được chọn refPicLX,
- chỉ số bộ lọc nội suy nửa mẫu hpellfIdx,
- cờ luồng quang hai hướng bdofFlag,
- biến cIdx xác định chỉ số thành phần màu sắc của khối hiện tại.

Các đầu ra của quá trình này là:

- mảng $(sbWidth + brdExtSize) \times (sbHeight + brdExtSize)$ predSamplesLX của các giá trị mẫu dự báo.

Kích thước mở rộng đường biên khối dự báo brdExtSize được dẫn xuất như sau:

$$brdExtSize = (bdofFlag \mid (inter_affine_flag[xSb][ySb] \&\& \\ sps_affine_prof_enabled_flag)) ? 2 : 0 \quad (8-752)$$

Biến fRefWidth được gán bằng PicOutputWidthL của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefHeight được gán bằng PicOutputHeightL của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefLeftOff được gán bằng conf win left offset của ảnh tham chiếu trong mẫu độ sáng.

Biến fRefTopOff được gán bằng conf win top offset của ảnh tham chiếu trong mẫu độ sáng.

Vectơ chuyển động mvLX được gán bằng $(refMvLX - mvOffset)$.

- Nếu cIdx bằng 0, điều kiện sau áp dụng:
 - Các hệ số chia tỷ lệ và các biểu diễn dấu chấm cố định được định nghĩa là

$$hori_scale_fp = ((fRefWidth \ll 14) + (PicOutputWidthL \\ \gg 1)) / PicOutputWidthL \quad (8-753)$$

$$vert_scale_fp = ((fRefHeight \ll 14) + (PicOutputHeightL \\ \gg 1)) / PicOutputHeightL \quad (8-754)$$
 - Giả sử $(xIntL, yIntL)$ là vị trí độ sáng theo các đơn vị mẫu toàn phần và $(xFracL, yFracL)$ là độ lệch theo đơn vị 1/16 mẫu. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số bên trong các mảng mẫu tham chiếu refPicLX.

- Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu (x_{SbIntL} , y_{SbIntL}) được gán bằng ($x_{Sb} + (mvLX[0] \gg 4)$, $y_{Sb} + (mvLX[1] \gg 4)$).
- Đối với mỗi vị trí mẫu độ sáng ($x_L = 0..sbWidth - 1 + brdExtSize$, $y_L = 0..sbHeight - 1 + brdExtSize$) bên trong mảng mẫu độ sáng dự báo $predSamplesLX$, giá trị mẫu độ sáng dự báo tương ứng $predSamplesLX[x_L][y_L]$ được dẫn xuất như sau:

- Giả sử ($refxSb_L$, $refySb_L$) và ($refx_L$, $refy_L$) là các vị trí độ sáng được trở đến bởi vector chuyển động ($refMvLX[0]$, $refMvLX[1]$) theo các đơn vị 1/16 mẫu. Các biến $refxSb_L$, $refx_L$, $refySb_L$, và $refy_L$ được dẫn xuất như sau:

$$refxSb_L = ((x_{Sb} \ll 4) + refMvLX[0]) * hori_scale_fp \quad (8-755)$$

$$refx_L = ((Sign(refxSb) * ((Abs(refxSb) + 128) \gg 8) + x_L * ((hori_scale_fp + 8) \gg 4)) + 32) \gg 6 \quad (8-756)$$

$$refySb_L = ((y_{Sb} \ll 4) + refMvLX[1]) * vert_scale_fp \quad (8-757)$$

$$refy_L = ((Sign(refySb) * ((Abs(refySb) + 128) \gg 8) + y_L * ((vert_scale_fp + 8) \gg 4)) + 32) \gg 6 \quad (8-758)$$

- Các biến x_{IntL} , y_{IntL} , x_{FracL} và y_{FracL} được dẫn xuất như sau:

$$x_{IntL} = \underline{Clip3(fRefLeftOff, fRefLeftOff + fRefWidth - 1, refx_L \gg 4)} \quad (8-759)$$

$$y_{IntL} = \underline{Clip3(fRefTopOff, fRefTopOff + fRefHeight - 1, refy_L \gg 4)} \quad (8-760)$$

$$x_{FracL} = refx_L \& 15 \quad (8-761)$$

$$y_{FracL} = refy_L \& 15 \quad (8-762)$$

- Nếu $bdoFlag$ bằng TRUE hoặc ($sps_affine_prof_enabled_flag$ bằng TRUE và $inter_affine_flag[x_{Sb}][y_{Sb}]$ bằng TRUE), và một hoặc nhiều điều kiện sau là đúng, giá trị mẫu độ sáng dự báo $predSamplesLX[x_L][y_L]$ được dẫn xuất bằng cách gọi quá trình nạp mẫu số nguyên độ sáng như được xác định trong mệnh đề 8.5.6.3.3 với ($x_{IntL} + (x_{FracL} \gg 3) - 1$, $y_{IntL} + (y_{FracL} \gg 3) - 1$) và $refPicLX$ làm đầu vào.
 - x_L bằng 0.
 - x_L bằng $sbWidth + 1$.
 - y_L bằng 0.
 - y_L bằng $sbHeight + 1$.

- Ngược lại, giá trị mẫu độ sáng dự báo $\text{predSamplesLX}[xL][yL]$ được dẫn xuất bằng cách gọi quá trình lọc nội suy 8 bậc mẫu độ sáng như được xác định trong mệnh đề 8.5.6.3.2 với $(xIntL - (\text{brdExtSize} > 0 ? 1 : 0), yIntL - (\text{brdExtSize} > 0 ? 1 : 0))$, $(xFracL, yFracL)$, $(xSbIntL, ySbIntL)$, refPicLX , hpelIfIdx , sbWidth , sbHeight và (xSb, ySb) làm đầu vào.
- Ngược lại (cIdx không bằng 0), điều kiện sau áp dụng:
 - Giả sử $(xIntC, yIntC)$ là vị trí sắc độ theo các đơn vị mẫu toàn phần và $(xFracC, yFracC)$ là độ lệch theo các đơn vị mẫu 1/32. Các biến này chỉ được sử dụng trong mệnh đề này để xác định các vị trí mẫu phân số tổng quát bên trong các mảng mẫu tham chiếu refPicLX .
 - Tọa độ trên bên trái của khối giới hạn để đệm mẫu tham chiếu $(xSbIntC, ySbIntC)$ được gán bằng $((xSb / \text{SubWidthC}) + (\text{mvLX}[0] \gg 5), (ySb / \text{SubHeightC}) + (\text{mvLX}[1] \gg 5))$.
 - Đối với mỗi vị trí mẫu sắc độ ($x_C = 0..\text{sbWidth} - 1, y_C = 0..\text{sbHeight} - 1$) bên trong các mảng mẫu sắc độ dự báo predSamplesLX , giá trị mẫu sắc độ dự báo tương ứng $\text{predSamplesLX}[x_C][y_C]$ được dẫn xuất như sau:
 - Giả sử $(\text{refxSbc}, \text{refySbc})$ và $(\text{refxC}, \text{refyC})$ là các vị trí sắc độ được trở đến bởi vector chuyển động $(\text{mvLX}[0], \text{mvLX}[1])$ theo các đơn vị 1/32 mẫu. Các biến refxSbc , refySbc , refxC và refyC được dẫn xuất như sau:

$$\text{refxSbc} = ((xSb / \text{SubWidthC} \ll 5) + \text{mvLX}[0]) * \text{hori_scale_fp} \quad (8-763)$$

$$\begin{aligned} \text{refxC} = & ((\text{Sign}(\text{refxSbc}) * ((\text{Abs}(\text{refxSbc}) + 256) \gg 9) \\ & + x_C * ((\text{hori_scale_fp} + 8) \gg 4)) + 16) \gg 5 \end{aligned} \quad (8-764)$$

$$\text{refySbc} = ((ySb / \text{SubHeightC} \ll 5) + \text{mvLX}[1]) * \text{vert_scale_fp} \quad (8-765)$$

$$\begin{aligned} \text{refyC} = & ((\text{Sign}(\text{refySbc}) * ((\text{Abs}(\text{refySbc}) + 256) \gg 9) \\ & + y_C * ((\text{vert_scale_fp} + 8) \gg 4)) + 16) \gg 5 \end{aligned} \quad (8-766)$$
 - Các biến $xIntC$, $yIntC$, $xFracC$ và $yFracC$ được dẫn xuất như sau:

$$\begin{aligned} xIntC = & \text{Clip3}(\text{fRefLeftOff} / \text{SubWidthC}, (\text{fRefWidth} + \text{fRefLeftOff}) / \text{SubWidthC} - 1, \\ & \text{refxC} \gg 5) \quad (8-767) \\ yIntC = & \text{Clip3}(\text{fRefTopOff} / \text{SubHeightC}, (\text{fRefHeight} + \text{fRefTopOff}) / \text{SubHeightC} - 1, \\ & \text{refyC} \gg 5) \quad (8-768) \\ xFracC = & \text{refyC} \& 31 \quad (8-769) \end{aligned}$$

$yFracC = ref_{yc} \& 31$

(8-770)

- Giá trị mẫu dự báo $predSamplesLX[xC][yC]$ được dẫn xuất bằng cách gọi quá trình được xác định trong mệnh đề 8.5.6.3.4 với $(xIntC, yIntC)$, $(xFracC, yFracC)$, $(xSbIntC, ySbIntC)$, $sbWidth$, $sbHeight$ và $refPicLX$ làm đầu vào.

5.7. Phương án thực hiện về việc sử dụng các công cụ mã

5.7.1 Điều khiển bật/tắt BDOF

- Biến $currPic$ xác định ảnh hiện tại và biến $bdofFlag$ được dẫn xuất như sau:
 - Nếu tất cả điều kiện sau là đúng, $bdofFlag$ được gán bằng TRUE.
 - $sps_bdof_enabled_flag$ bằng 1 và $slice_disable_bdof_dmvr_flag$ bằng 0.
 - $predFlagL0[xSbIdx][ySbIdx]$ và $predFlagL1[xSbIdx][ySbIdx]$ đều bằng 1.
 - $DiffPicOrderCnt(currPic, RefPicList[0][refIdxL0])$ bằng $DiffPicOrderCnt(RefPicList[1][refIdxL1], currPic)$.
 - $RefPicList[0][refIdxL0]$ là ảnh tham chiếu ngắn hạn và $RefPicList[1][refIdxL1]$ là ảnh tham chiếu ngắn hạn.
 - $MotionModelIdx[xCb][yCb]$ bằng 0.
 - $merge_subblock_flag[xCb][yCb]$ bằng 0.
 - $sym_mvd_flag[xCb][yCb]$ bằng 0.
 - $ciip_flag[xCb][yCb]$ bằng 0.
 - $BcwIdx[xCb][yCb]$ bằng 0.
 - $luma_weight_10_flag[refIdxL0]$ và $luma_weight_11_flag[refIdxL1]$ đều bằng 0.
 - $cbWidth$ lớn hơn hoặc bằng 8.
 - $cbHeight$ lớn hơn hoặc bằng 8.
 - $cbHeight * cbWidth$ lớn hơn hoặc bằng 128.
 - $RefPicIsScaled[0][refIdxL0]$ bằng 0 và $RefPicIsScaled[1][refIdxL1]$ bằng 0.
 - $cIdx$ bằng 0.
 - Ngược lại, $bdofFlag$ được gán bằng FALSE.

5.7.2 Điều khiển bật/tắt DMVR

- Khi tất cả điều kiện sau là đúng, dmvrFlag được gán bằng 1:
 - sps_dmvr_enabled_flag bằng 1 và slice_disable_bdo_f_dmvr_flag bằng 0
 - general_merge_flag[xCb][yCb] bằng 1
 - cả predFlagL0[0][0] và predFlagL1[0][0] are bằng 1
 - mmvd_merge_flag[xCb][yCb] bằng 0
 - ciip_flag[xCb][yCb] bằng 0
 - DiffPicOrderCnt(currPic, RefPicList[0][refIdxL0]) bằng
DiffPicOrderCnt(RefPicList[1][refIdxL1], currPic)
 - RefPicList[0][refIdxL0] là ảnh tham chiếu ngắn hạn và
RefPicList[1][refIdxL1] là ảnh tham chiếu ngắn hạn.
 - BcwIdx[xCb][yCb] bằng 0
 - Cả luma_weight_l0_flag[refIdxL0] lẫn luma_weight_l1_flag[refIdxL1]
bằng 0
 - cbWidth lớn hơn hoặc bằng 8
 - cbHeight lớn hơn hoặc bằng 8
 - cbHeight*cbWidth lớn hơn hoặc bằng 128
 - RefPicIsScaled[0][refIdxL0] bằng 0 và RefPicIsScaled[1][refIdxL1]
bằng 0.

5.7.3 Điều khiển bật/tắt PROF cho danh sách ảnh tham chiếu X

Biến cbProfFlagLX được dẫn xuất như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, cbProfFlagLX được gán bằng FALSE.
 - sps_affine_prof_enabled_flag bằng 0.
 - fallbackModeTriggered bằng 1.
 - numCpMv bằng 2 và cpMvLX[1][0] bằng cpMvLX[0][0] và
cpMvLX[1][1] bằng cpMvLX[0][1].
 - numCpMv bằng 3 và cpMvLX[1][0] bằng cpMvLX[0][0] và
cpMvLX[1][1] bằng cpMvLX[0][1] và cpMvLX[2][0] bằng cpMvLX[0][0]
và cpMvLX[2][1] bằng cpMvLX[0][1].
 - RefPicIsScaled[X][refIdxLX] bằng 0

- [[pic_width_in_luma_samples của ảnh tham chiếu refPicLX được lần lượt liên kết với refIdxLX không bằng pic_width_in_luma_samples của ảnh hiện tại.
- pic_height_in_luma_samples của ảnh tham chiếu refPicLX được lần lượt liên kết với refIdxLX không bằng pic_height_in_luma_samples của ảnh hiện tại.]]
- Ngược lại, cbProfFlagLX được gán bằng TRUE.

5.7.4 Điều khiển bật/tắt PROF cho danh sách ảnh tham chiếu X (phương án thực hiện thứ hai)

Biến cbProfFlagLX được dẫn xuất như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, cbProfFlagLX được gán bằng FALSE.
 - sps_affine_prof_enabled_flag bằng 0.
 - fallbackModeTriggered bằng 1.
 - numCpMv bằng 2 và cpMvLX[1][0] bằng cpMvLX[0][0] và cpMvLX[1][1] bằng cpMvLX[0][1].
 - numCpMv bằng 3 và cpMvLX[1][0] bằng cpMvLX[0][0] và cpMvLX[1][1] bằng cpMvLX[0][1] và cpMvLX[2][0] bằng cpMvLX[0][0] và cpMvLX[2][1] bằng cpMvLX[0][1].
 - *RefPicIsScaled[0][refIdxL0] và RefPicIsScaled[1][refIdxL1] bằng 0*
 - [[pic_width_in_luma_samples của ảnh tham chiếu refPicLX được lần lượt liên kết với refIdxLX không bằng pic_width_in_luma_samples của ảnh hiện tại.
 - pic_height_in_luma_samples của ảnh tham chiếu refPicLX được lần lượt liên kết với refIdxLX không bằng pic_height_in_luma_samples của ảnh hiện tại.]]
- Ngược lại, cbProfFlagLX được gán bằng TRUE.

6. Các phương án thực hiện lấy làm ví dụ của sáng chế

Fig.5 là sơ đồ khối của thiết bị xử lý video 500. Thiết bị 500 có thể được sử dụng để thực hiện một hoặc nhiều phương pháp được mô tả ở đây. Thiết bị 500 có thể được triển khai trong điện thoại thông minh, máy tính bảng, máy tính, bộ thu

Internet vạn vật (IoT), và v.v.. Thiết bị 500 có thể bao gồm một hoặc nhiều bộ xử lý 502, một hoặc nhiều bộ nhớ 504 và phần cứng xử lý video 506. (các) bộ xử lý 502 có thể được tạo cấu hình để thực hiện một hoặc nhiều phương pháp được nêu trong bản mô tả. Bộ nhớ (các bộ nhớ) 504 có thể được sử dụng để lưu trữ dữ liệu và mã được sử dụng để thực hiện các phương pháp và các kỹ thuật được mô tả ở đây. Phần cứng xử lý video 506 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được nêu trong bản mô tả, và có thể là một phần hoặc toàn bộ là một phần của các bộ xử lý 502 (chẳng hạn, CPU lõi bộ xử lý đồ họa và hệ mạch xử lý tín hiệu khác).

Theo sáng chế, cụm từ “xử lý video” hoặc mã hóa có thể đề cập đến mã hóa video, giải mã video, nén video hoặc giải nén video. Chẳng hạn, các thuật toán nén video có thể được áp dụng trong khi biến đổi từ biểu diễn pixel của video sang biểu diễn dòng bit tương ứng hoặc ngược lại. Biểu diễn dòng bit của khối video hiện tại có thể, chẳng hạn, tương ứng với các bit nằm cùng vị trí hoặc được trải rộng ở các địa điểm khác nhau trong dòng bit, như được định nghĩa bởi cú pháp. Chẳng hạn, khối lớn có thể được mã hóa theo các giá trị dư sai số được mã hóa và được biến đổi và cũng sử dụng các bit trong các tiêu đề và các trường khác trong dòng bit.

Cần hiểu rằng các phương pháp và các kỹ thuật được bộc lộ sẽ tạo thuận lợi các phương án thực hiện của bộ mã hóa và/hoặc bộ giải mã video được đưa vào trong các thiết bị xử lý video chẳng hạn điện thoại thông minh, máy tính xách tay, máy tính để bàn, và thiết bị tương tự bằng cách cho phép sử dụng các kỹ thuật được bộc lộ theo sáng chế.

Fig.6 là lưu đồ của phương pháp lấy làm ví dụ 600 xử lý video. Phương pháp 600 bao gồm, ở bước 610, thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, bộ lọc nội suy tương tự được áp dụng cho nhóm mẫu lân cận hoặc không lân cận được dự báo nhờ sử dụng khối video hiện tại.

Một số phương án thực hiện có thể được mô tả nhờ sử dụng định dạng dựa trên mệnh đề sau.

1. Phương pháp xử lý video, bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, bộ lọc nội suy tương tự được áp dụng cho nhóm mẫu lân cận hoặc không lân cận được dự báo nhờ sử dụng khối video hiện tại.

2. Phương pháp theo mệnh đề 1, trong đó bộ lọc nội suy tương tự là bộ lọc nội suy dọc.

3. Phương pháp theo mệnh đề 1, trong đó bộ lọc nội suy tương tự là bộ lọc nội suy ngang.

4. Phương pháp theo mệnh đề 1, trong đó nhóm các mẫu liên kề hoặc không liên kề bao gồm tất cả các mẫu được đặt trong vùng của khối video hiện tại.

5. Phương pháp theo mệnh đề 4, trong đó các khối video hiện tại được phân chia thành nhiều hình chữ nhật mà mỗi hình chữ nhật có kích thước $M \times N$.

6. Phương pháp theo mệnh đề 5, trong đó M và/hoặc N được xác định trước.

7. Phương pháp theo mệnh đề 5, trong đó M và/hoặc N được dẫn xuất từ các kích thước của khối video hiện tại.

8. Phương pháp theo mệnh đề 5, trong đó M và/hoặc N được báo hiệu trong biểu diễn được mã hóa của khối video hiện tại.

9. Phương pháp theo mệnh đề 1, trong đó nhóm mẫu chia sẻ cùng vector chuyển động.

10. Phương pháp theo mệnh đề 9, trong đó nhóm mẫu chia sẻ cùng thành phần ngang và/hoặc cùng phần phân số của thành phần ngang.

11. Phương pháp theo mệnh đề 9, trong đó nhóm mẫu chia sẻ cùng thành phần dọc và/hoặc cùng phần phân số của thành phần dọc.

12. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 9 đến 11, trong đó cùng vector chuyển động hoặc các thành phần của nó thỏa mãn một hoặc nhiều quy tắc dựa vào ít nhất một trong: độ phân giải của ảnh tham chiếu, kích thước của ảnh tham chiếu, độ phân giải của khối video hiện tại, kích thước của khối video hiện tại, hoặc giá trị chính xác.

13. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 9 đến 11, trong đó cùng vectơ chuyển động hoặc các thành phần của nó tương ứng với thông tin chuyển động của mẫu được đặt trong khối video hiện tại.

14. Phương pháp theo mệnh đề bất kỳ trong các mệnh đề từ 9 đến 11, trong đó cùng vectơ chuyển động hoặc các thành phần của nó được gán bằng thông tin chuyển động của mẫu ảo được đặt bên trong hoặc ngoài nhóm.

15. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, trong đó các khối được dự báo nhờ sử dụng khối video hiện tại chỉ được phép sử dụng thông tin chuyển động có giá trị số nguyên liên quan đến khối hiện tại.

16. Phương pháp theo mệnh đề 15, trong đó thông tin chuyển động có giá trị số nguyên được dẫn xuất bằng cách làm tròn thông tin chuyển động ban đầu của khối video hiện tại.

17. Phương pháp theo mệnh đề 15, trong đó thông tin chuyển động ban đầu của khối video hiện tại theo phương ngang và/hoặc hướng thẳng đứng.

18. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, bộ lọc nội suy được áp dụng để dẫn xuất các khối được dự báo nhờ sử dụng khối video hiện tại, và trong đó bộ lọc nội suy được chọn dựa trên quy tắc.

19. Phương pháp theo mệnh đề 18, liệu quy tắc này liên quan đến độ phân giải và/hoặc kích thước của ảnh tham chiếu so với độ phân giải và/hoặc kích thước của khối video hiện tại.

20. Phương pháp theo mệnh đề 18, trong đó bộ lọc nội suy là bộ lọc nội suy dọc.

21. Phương pháp theo mệnh đề 18, trong đó bộ lọc nội suy là bộ lọc nội suy ngang.

22. Phương pháp theo mệnh đề 18, trong đó bộ lọc nội suy là một trong: bộ lọc 1 bậc, bộ lọc song tuyến tính, bộ lọc 4 bậc, hoặc bộ lọc 6 bậc.

23. Phương pháp theo mệnh đề 22, trong đó bộ lọc nội suy được sử dụng như là một phần của các bước biến đổi khác.

24. Phương pháp theo mệnh đề 18, trong đó bộ lọc nội suy bao gồm sử dụng các mẫu đệm.

25. Phương pháp theo mệnh đề 18, trong đó sử dụng bộ lọc nội suy phụ thuộc vào thành phần màu của mẫu của khối video hiện tại.

26. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu độ phân giải và/hoặc kích thước của ảnh tham chiếu khác với độ phân giải và/hoặc kích thước của khối video hiện tại, áp dụng có chọn lọc bộ lọc tách khối, trong đó độ bền của bộ lọc tách khối được thiết lập theo quy tắc liên quan đến độ phân giải và/hoặc kích thước của ảnh tham chiếu so với độ phân giải và/hoặc kích thước của khối video hiện tại.

27. Phương pháp theo mệnh đề 27, trong đó độ bền của bộ lọc tách khối thay đổi từ một khối video sang khối khác.

28. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, nếu có ảnh phụ của khối video hiện tại, dòng bit tương hợp thỏa mãn quy tắc liên quan đến độ phân giải và/hoặc kích thước của ảnh tham chiếu so với độ phân giải và/hoặc kích thước của khối video hiện tại.

29. Phương pháp theo mệnh đề 28, trong đó phương pháp còn bao gồm bước:

tách khối video hiện tại thành một hoặc nhiều ảnh phụ, trong đó việc tách phụ thuộc ít nhất vào độ phân giải của khối video hiện tại.

30. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, ảnh tham chiếu của khối video hiện tại được lấy mẫu lại theo quy tắc dựa trên các kích thước của khối video hiện tại.

31. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa khối video hiện tại và biểu diễn được mã hóa của khối video hiện tại, trong đó, trong khi biến đổi, sử dụng công cụ mã hóa trên khối

video hiện tại được kích hoạt hoặc vô hiệu hóa có chọn lọc tùy thuộc độ phân giải/kích thước của ảnh tham chiếu của khối video hiện tại so với độ phân giải/kích thước của khối video hiện tại.

32. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó nhóm mẫu được đặt trong cửa sổ tương hợp.

33. Phương pháp theo mệnh đề 32, trong đó cửa sổ tương hợp có hình chữ nhật.

34. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó độ phân giải gắn với độ phân giải của khối video được mã hóa/giải mã hoặc độ phân giải của cửa sổ tương hợp trong khối video được mã hóa/giải mã.

35. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó kích thước gắn với kích thước của khối video được mã hóa/giải mã hoặc kích thước của cửa sổ tương hợp trong khối video được mã hóa/giải mã.

36. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó kích thước gắn với các kích thước của khối video được mã hóa/giải mã hoặc các kích thước của cửa sổ tương hợp trong khối video được mã hóa/giải mã.

37. Phương pháp theo mệnh đề 32, trong đó cửa sổ tương hợp được định nghĩa bởi tập hợp tham số cửa sổ cắt tương hợp.

38. Phương pháp theo mệnh đề 37, trong đó ít nhất một phần tập hợp tham số cửa sổ cắt tương hợp được báo hiệu ngầm hoặc tường minh trong biểu diễn được mã hóa.

39. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó tập hợp tham số cửa sổ cắt tương hợp không được phép báo hiệu trong biểu diễn được mã hóa.

40. Phương pháp theo một hoặc nhiều mệnh đề nêu trên, trong đó vị trí của mẫu tham chiếu được dẫn xuất với mẫu trên bên trái của khối video hiện tại trong cửa sổ tương hợp.

41. Phương pháp xử lý video bao gồm bước:

thực hiện biến đổi giữa nhiều khối video và các biểu diễn được mã hóa của nhiều khối video, trong đó, trong khi biến đổi, cửa sổ tương hợp thứ nhất được định nghĩa đối với khối video thứ nhất và cửa sổ tương hợp thứ hai đối với khối video thứ hai, và trong đó tỷ lệ của chiều rộng và/hoặc chiều cao của cửa sổ tương hợp

thứ nhất trên cửa sổ tương hợp thứ hai tuân theo quy tắc dựa vào ít nhất dòng bit tương hợp.

42. Thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề 1 đến 41.

43. Thiết bị mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu theo một hoặc nhiều mệnh đề 1 đến 41.

44. Sản phẩm chương trình máy tính có mã máy tính được lưu trữ trên đó, mã, khi được thực thi bằng bộ xử lý, khiến bộ xử lý thực thi phương pháp được nêu trong mệnh đề bất kỳ trong các mệnh đề từ 1 đến 41.

45. Phương pháp, thiết bị hoặc hệ thống được nêu trong bản mô tả.

Fig.9 là sơ đồ khối thể hiện hệ thống xử lý video ví dụ 900 trong đó các kỹ thuật khác nhau được bộc lộ ở đây có thể được triển khai. Các cách thực hiện khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 900. Hệ thống 900 có thể bao gồm đầu vào 902 để nhận nội dung video. Nội dung video có thể được nhận ở định dạng thô hoặc không được nén, chẳng hạn, các giá trị pixel đa thành phần 8 hoặc 10 bit, hoặc có thể ở định dạng nén hoặc được mã hóa. Đầu vào 902 có thể là giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện lưu trữ. Các ví dụ của giao diện mạng bao gồm các giao diện hữu tuyến chẳng hạn Ethernet, mạng quang thụ động (PON), v.v. và giao diện không dây chẳng hạn giao diện Wi-Fi hoặc tế bào.

Hệ thống 900 có thể bao gồm thành phần mã 904 có thể thực hiện các phương pháp tạo mã hoặc mã hóa khác nhau được nêu trong bản mô tả. Thành phần mã 904 có thể giảm tốc độ bit trung bình của video từ đầu vào 902 đến đầu ra của thành phần mã 904 to produce biểu diễn được mã hóa của video. Do vậy, các kỹ thuật mã hóa đôi lúc được gọi là các kỹ thuật nén video hoặc chuyển mã video. Đầu ra của thành phần mã 904 có thể được lưu trữ, hoặc được truyền qua kết nối truyền thông, như được biểu diễn bởi thành phần 906. Biểu diễn dòng bit được lưu trữ hoặc được truyền thông (hoặc được mã hóa) của video được nhận ở đầu vào 902 có thể được sử dụng bởi thành phần 908 để tạo các giá trị pixel hoặc video có thể hiển thị được truyền đến giao diện hiển thị 910. Quá trình tạo video người dùng có thể xem từ biểu diễn dòng bit đôi lúc được gọi là giải nén video. Ngoài ra, trong khi các hoạt

động xử lý video cụ thể được gọi là các hoạt động hoặc các công cụ “mã hóa”, cần hiểu rằng các hoạt động hoặc các công cụ mã hóa được sử dụng ở bộ mã hóa và các công cụ hoặc các hoạt động giải mã tương ứng mà đảo ngược kết quả mã hóa sẽ được thực thi bởi bộ giải mã.

Các ví dụ về giao diện buýt ngoại vi hoặc giao diện hiển thị có thể bao gồm buýt nối tiếp đa năng (USB) hoặc giao diện đa phương tiện độ nét cao (HDMI) or Displayport, và v.v.. Các ví dụ về các giao diện lưu trữ bao gồm SATA (serial advanced technology attachment), PCI, giao diện IDE, và tương tự. Các kỹ thuật được nêu trong bản mô tả có thể được triển khai trong các thiết bị điện tử khác nhau chẳng hạn, điện thoại di động, máy tính xách tay, điện thoại thông minh hoặc các thiết bị khác có thể thực hiện xử lý dữ liệu số và/hoặc hiển thị video.

Fig.10 là sơ đồ khối minh họa hệ thống tạo mã video ví dụ 100 có thể sử dụng các kỹ thuật của sáng chế.

Như được thể hiện trên Fig.10, hệ thống mã hóa video 100 có thể bao gồm a thiết bị nguồn 110 và thiết bị đích 120. Thiết bị nguồn 110 tạo dữ liệu video được mã hóa có thể được gọi là thiết bị mã hóa video. Thiết bị đích 120 có thể giải mã dữ liệu video được mã hóa được tạo bởi thiết bị nguồn 110 có thể được gọi là thiết bị giải mã video.

Thiết bị nguồn 110 có thể bao gồm nguồn video 112, bộ mã hóa video 114, và giao diện nhập/xuất (I/O) 116.

Nguồn video 112 có thể bao gồm nguồn chẳng hạn thiết bị ghi video, giao diện để nhận dữ liệu video từ nhà cung cấp nội dung video, và/hoặc hệ thống đồ họa máy tính để tạo dữ liệu video, hoặc kết hợp của các nguồn này. Dữ liệu video có thể bao gồm một hoặc nhiều ảnh. Bộ mã hóa video 114 mã hóa dữ liệu video từ nguồn video 112 để tạo dòng bit. Dòng bit có thể bao gồm chuỗi bit mà tạo biểu diễn được mã hóa của dữ liệu video. Dòng bit có thể bao gồm các ảnh được mã hóa và dữ liệu liên kết. ảnh được mã hóa là biểu diễn được mã hóa của ảnh. Dữ liệu liên kết có thể bao gồm các tập hợp tham số chuỗi, các tập hợp tham số ảnh, và các cấu trúc cú pháp khác. Giao diện I/O 116 có thể bao gồm bộ điều biến/giải điều biến (modem) và/hoặc bộ truyền. Dữ liệu video được mã hóa có thể được truyền trực tiếp đến thiết bị đích 120 qua giao diện I/O 116 qua mạng 130a. Dữ liệu video được mã hóa cũng

có thể được lưu trữ vào phương tiện/máy chủ lưu trữ 130b để truy nhập bởi thiết bị đích 120.

Thiết bị đích 120 có thể bao gồm giao diện I/O 126, bộ giải mã video 124, và bộ phận hiển thị 122.

Giao diện I/O 126 có thể bao gồm bộ thu và/hoặc modem. Giao diện I/O 126 có thể thu được dữ liệu video được mã hóa từ thiết bị nguồn 110 hoặc phương tiện/máy chủ lưu trữ 130b. Bộ giải mã video 124 có thể giải mã dữ liệu video được mã hóa. Bộ phận hiển thị 122 có thể hiển thị dữ liệu video được giải mã đến người dùng. Bộ phận hiển thị 122 có thể được tích hợp với thiết bị đích 120, hoặc có thể nằm ngoài thiết bị đích 120 được tạo cấu hình để giao tiếp với bộ phận hiển thị bên ngoài.

Bộ mã hóa video 114 và bộ giải mã video 124 có thể hoạt động theo chuẩn nén video, chẳng hạn chuẩn HEVC, chuẩn VVC và các chuẩn khác và/hoặc hiện tại khác.

Fig.11 là sơ đồ khối minh họa ví dụ của bộ mã hóa video 200, có thể là bộ mã hóa video 114 trong hệ thống 100 được minh họa trên Fig.10.

Bộ mã hóa video 200 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật của sáng chế. Trong ví dụ trên Fig.9, bộ mã hóa video 200 bao gồm các thành phần chức năng. Các kỹ thuật được mô tả theo sáng chế có thể được chia sẻ trong số các thành phần khác nhau của bộ mã hóa video 200. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả theo sáng chế.

Các thành phần chức năng của bộ mã hóa video 200 có thể bao gồm khối phân vùng 201, khối dự báo 202 có thể bao gồm khối chọn chế độ 203, khối ước tính chuyển động 204, khối bù chuyển động 205 và khối dự báo trong 206, khối tạo phần dư 207, khối biến đổi 208, khối lượng tử 209, khối lượng tử ngược 210, khối biến đổi ngược 211, khối tái tạo 212, bộ đệm 213, và khối mã hóa entropy 214.

Trong các ví dụ khác, bộ mã hóa video 200 có thể bao gồm vài thành phần chức năng khác nhau. Trong ví dụ, khối dự báo 202 có thể bao gồm đơn vị sao chép trong khối (IBC). Khối IBC có thể thực hiện dự báo trong chế độ IBC trong đó ít nhất một ảnh tham chiếu là ảnh trong đó đặt khối video hiện tại.

Ngoài ra, một số thành phần, chẳng hạn khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể được tích hợp mạnh, nhưng được biểu diễn trong ví dụ trên Fig.5 riêng rẽ để giải thích.

Khối phân vùng 201 có thể phân vùng ảnh thành một hoặc nhiều khối video. Bộ mã hóa video 200 và bộ giải mã video 300 có thể hỗ trợ các kích thước khối video khác nhau.

Khối chọn chế độ 203 có thể chọn một trong chế độ mã, trong hoặc ngoài, chẳng hạn, dựa trên các kết quả sai số, và cấp khối được mã hóa trong hoặc ngoài thu được cho khối tạo phần dư 207 để tạo dữ liệu khối dư và với khối tái tạo 212 để tái tạo khối được mã hóa để sử dụng làm ảnh tham chiếu. Trong một số ví dụ, Khối chọn chế độ 203 có thể chọn chế độ kết hợp dự báo trong ngoài (CIIP) trong đó dự báo dựa trên tín hiệu dự báo ngoài và tín hiệu dự báo trong. Khối chọn chế độ 203 cũng có thể chọn giải pháp đối với vector chuyển động (chẳng hạn, độ chính xác pixel con hoặc pixel nguyên) đối với khối trong trường hợp dự báo ngoài.

Để thực hiện dự báo ngoài trên khối video hiện tại, khối ước tính chuyển động 204 có thể tạo thông tin chuyển động cho khối video hiện tại bằng cách so sánh một hoặc nhiều khung tham chiếu từ bộ đệm 213 với khối video hiện tại. Khối bù chuyển động 205 có thể xác định khối video được dự báo cho khối video hiện tại dựa trên thông tin chuyển động và các mẫu được giải mã của các ảnh từ bộ đệm 213 khác ngoài ảnh được liên kết với khối video hiện tại.

Khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể thực hiện các hoạt động khác nhau cho khối video hiện tại, chẳng hạn, tùy thuộc liệu khối video hiện tại ở lát I, lát P, hoặc lát B.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể thực hiện dự báo đơn hướng cho khối video hiện tại, và khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu của danh sách 0 hoặc danh sách 1 đối với khối video tham chiếu cho khối video hiện tại. Sau đó, khối ước tính chuyển động 204 có thể tạo chỉ số tham chiếu mà chỉ báo ảnh tham chiếu trong danh sách 0 hoặc danh sách 1 mà chứa khối video tham chiếu và vector chuyển động mà chỉ báo dịch chuyển không gian giữa khối video hiện tại và khối video tham chiếu. Khối ước tính chuyển động 204 có thể xuất ra chỉ số tham chiếu, bộ chỉ báo hướng dự báo, và vector chuyển động

dưới dạng thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo khối video được dự báo của khối hiện tại dựa trên khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong các ví dụ khác, khối ước tính chuyển động 204 có thể thực hiện dự báo hai hướng cho khối video hiện tại, khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu trong danh sách 0 đối với khối video tham chiếu cho khối video hiện tại và cũng có thể tìm kiếm các ảnh tham chiếu trong danh sách 1 đối với khối video tham chiếu khác cho khối video hiện tại. Khối ước tính chuyển động 204 sau đó có thể tạo các chỉ số tham chiếu mà chỉ báo các ảnh tham chiếu trong danh sách 0 và danh sách 1 chứa các khối video tham chiếu và các vector chuyển động mà chỉ báo các dịch chuyển không gian giữa các khối video tham chiếu và khối video hiện tại. Khối ước tính chuyển động 204 có thể xuất ra các chỉ số tham chiếu và vector chuyển động của khối video hiện tại dưới dạng thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo khối video được dự báo của khối video hiện tại dựa trên các khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể xuất ra toàn bộ tập hợp thông tin chuyển động để xử lý giải mã bộ giải mã.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể không xuất ra toàn bộ tập hợp thông tin chuyển động đối với video hiện tại. Thay vào đó, khối ước tính chuyển động 204 có thể báo hiệu thông tin chuyển động của khối video hiện tại dựa vào thông tin chuyển động của khối video khác. Chẳng hạn, khối ước tính chuyển động 204 có thể xác định rằng thông tin chuyển động của khối video hiện tại hoàn toàn giống như thông tin chuyển động của khối video lân cận.

Trong một ví dụ, khối ước tính chuyển động 204 có thể chỉ báo, trong cấu trúc cú pháp được liên kết với khối video hiện tại, giá trị mà chỉ báo to bộ giải mã video 300 that khối video hiện tại có thông tin chuyển động giống như khối video khác.

Trong ví dụ khác, khối ước tính chuyển động 204 có thể nhận diện, trong cấu trúc cú pháp được liên kết với khối video hiện tại, khối video khác và hiệu số vector chuyển động (MVD). MVD chỉ báo hiệu số giữa vector chuyển động của khối video hiện tại and vector chuyển động của khối video được chỉ báo. Bộ giải mã video 300

có thể sử dụng vector chuyển động của khối video được chỉ báo và MVD để xác định vector chuyển động của khối video hiện tại.

Như nêu trên, bộ mã hóa video 200 có thể báo hiệu dự báo vector chuyển động. Hai ví dụ về kỹ thuật báo hiệu dự báo đó có thể được triển khai bằng bộ mã hóa video 200 bao gồm dự báo vector chuyển động cải tiến (AMVP) và báo hiệu chế độ hợp nhất.

Khối dự báo trong 206 có thể thực hiện dự báo trong trên khối video hiện tại. Khi khối dự báo trong 206 thực hiện dự báo trong trên khối video hiện tại, khối dự báo trong 206 có thể tạo dữ liệu dự báo cho khối video hiện tại dựa trên các mẫu được giải mã của các khối video khác trong cùng ảnh. Dữ liệu dự báo cho khối video hiện tại có thể bao gồm khối video được dự báo và các phần tử cú pháp khác nhau.

Khối tạo phần dư 207 có thể tạo dữ liệu dư cho khối video hiện tại bằng cách lấy khối video hiện tại trừ đi (chẳng hạn, được chỉ báo bởi dấu trừ) (các) khối video được dự báo của khối video hiện tại. Dữ liệu dư của khối video hiện tại có thể bao gồm các khối video dư tương ứng với các thành phần mẫu khác nhau của các mẫu trong khối video hiện tại.

Trong các ví dụ khác, có thể không có dữ liệu dư cho khối video hiện tại cho khối video hiện tại, chẳng hạn ở chế độ bỏ qua, và khối tạo phần dư 207 có thể không thực hiện phép trừ.

Khối xử lý biến đổi 208 có thể tạo một hoặc nhiều khối video có hệ số biến đổi cho khối video hiện tại bằng cách áp dụng một hoặc nhiều phép biến đổi cho khối video dư được liên kết với khối video hiện tại.

Sau khi khối xử lý biến đổi 208 tạo khối video có hệ số biến đổi được liên kết với khối video hiện tại, khối lượng tử 209 có thể lượng tử hóa khối video có hệ số biến đổi được liên kết với khối video hiện tại dựa trên một hoặc nhiều giá trị tham số lượng tử (QP) được liên kết với khối video hiện tại.

Khối lượng tử ngược 210 và khối biến đổi ngược 211 có thể áp dụng lần lượt lượng tử ngược và biến đổi ngược cho khối video có hệ số biến đổi, để tái tạo khối video dư từ khối video có hệ số biến đổi. Khối tái tạo 212 có thể bổ sung khối video dư được tái tạo vào các mẫu tương ứng từ một hoặc nhiều khối video được dự báo

được tạo bởi khối dự báo 202 để tạo khối video được tái tạo được liên kết với khối hiện tại để lưu trữ trong bộ đệm 213.

Sau khi khối tái tạo 212 tái tạo khối video, hoạt động lọc vòng có thể được thực hiện để giảm các giả tượng tạo khối video trong khối video.

Khối mã hóa entropy 214 có thể nhận dữ liệu từ các thành phần chức năng khác của bộ mã hóa video 200. Khi khối mã hóa entropy 214 nhận dữ liệu, khối mã hóa entropy 214 có thể thực hiện một hoặc nhiều hoạt động mã hóa entropy để tạo dữ liệu được mã hóa entropy và xuất dòng bit bao gồm dữ liệu được mã hóa entropy.

Fig.12 là sơ đồ khối minh họa ví dụ của bộ giải mã video 300 có thể là bộ giải mã video 114 in hệ thống 100 được minh họa trên Fig.10.

Bộ giải mã video 300 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật của sáng chế. Trong ví dụ trên Fig.10, bộ giải mã video 300 bao gồm các thành phần chức năng. Các kỹ thuật được mô tả theo sáng chế có thể được chia sẻ giữa các thành phần khác nhau của bộ giải mã video 300. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả theo sáng chế.

Trong ví dụ trên Fig.10, bộ giải mã video 300 bao gồm khối giải mã entropy 301, khối bù chuyển động 302, khối dự báo trong 303, khối lượng tử ngược 304, khối biến đổi ngược 305, và khối tái tạo 306 và bộ đệm 307. Bộ giải mã video 300 có thể, trong một số ví dụ, thực hiện bước giải mã nghịch đảo với bước mã hóa được mô tả với bộ mã hóa video 200 (chẳng hạn, Fig.9).

Khối giải mã entropy 301 có thể truy xuất dòng bit được mã hóa. Dòng bit được mã hóa có thể bao gồm dữ liệu video được mã hóa entropy (chẳng hạn, các khối được mã hóa của dữ liệu video). Khối giải mã entropy 301 có thể giải mã dữ liệu video được mã hóa entropy, và từ dữ liệu video được giải mã entropy, khối bù chuyển động 302 có thể xác định thông tin chuyển động bao gồm các vector chuyển động, độ chính xác vector chuyển động, các chỉ số danh sách ảnh tham chiếu, và thông tin chuyển động khác. Khối bù chuyển động 302 có thể, chẳng hạn, xác định thông tin bằng cách thực hiện AMVP và chế độ hợp nhất.

Khối bù chuyển động 302 có thể tạo các khối được bù chuyển động, có thể thực hiện nội suy dựa trên các bộ lọc nội suy. Các định danh cho các bộ lọc nội suy cần

được sử dụng có độ chính xác pixel con có thể được bao gồm trong phần tử cú pháp.

Khối bù chuyển động 302 có thể sử dụng các bộ lọc nội suy như được sử dụng bằng bộ mã hóa video 20 trong khi mã hóa khối video để tính toán các giá trị được nội suy đối với các pixel số nguyên phụ của khối tham chiếu. Khối bù chuyển động 302 có thể xác định các bộ lọc nội suy được sử dụng bằng bộ mã hóa video 200 theo thông tin cú pháp được nhận và sử dụng các bộ lọc nội suy để tạo các khối dự báo.

Khối bù chuyển động 302 có thể sử dụng một số thông tin cú pháp để xác định các kích thước của các khối được sử dụng để mã hóa (các) khung và/hoặc (các) lát của chuỗi video được mã hóa, thông tin phân vùng mà mô tả cách thức mỗi khối lớn của ảnh của chuỗi video được mã hóa được phân vùng, các chế độ chỉ báo cách mã hóa mỗi phân vùng, một hoặc nhiều khung tham chiếu (và danh sách khung tham chiếu) đối với mỗi khối được mã hóa ngoài, và thông tin khác để giải mã chuỗi video được mã hóa.

Khối dự báo trong 303 có thể sử dụng các chế độ dự báo trong chẳng hạn được nhận trong dòng bit để tạo khối dự báo từ các khối liền kề trong không gian. Khối lượng tử ngược 303 lượng tử ngược, chẳng hạn, bỏ lượng tử, các hệ số của khối video được lượng tử được tạo trong dòng bit và được giải mã bằng khối giải mã entropy 301. Khối biến đổi ngược 303 áp dụng biến đổi ngược.

Khối tái tạo 306 có thể tính tổng các khối dự với các khối dự báo tương ứng được tạo bởi khối bù chuyển động 202 hoặc khối dự báo trong 303 để tạo các khối được giải mã. Nếu cần, bộ lọc tách khối có thể cũng được áp dụng để lọc các khối được giải mã để loại bỏ các giả tượng khối. Sau đó các khối video được giải mã được lưu trữ trong bộ đệm 307, mà tạo các khối tham chiếu để bù chuyển động tiếp theo.

Fig.11 là lưu đồ biểu diễn phương pháp xử lý video theo sáng chế. Phương pháp 1100 bao gồm, ở hoạt động 1110, thực hiện biến đổi giữa video và biểu diễn dòng bit của video. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng khả năng áp dụng của công cụ mã hóa tinh lọc MV phía bộ giải mã và công cụ mã hóa

luồng quang hai hướng cho ảnh video được chỉ báo riêng rẽ trong biểu diễn dòng bit.

Fig.13 là lưu đồ biểu diễn phương pháp 1300 xử lý video theo sáng chế. Phương pháp 1300 bao gồm, ở hoạt động 1310, xác định, đối với biến đổi giữa ảnh hiện tại của video và biểu diễn dòng bit của video, sử dụng công cụ tạo mã dựa trên điều kiện được liên kết với kích thước của một phần của một trong nhiều ảnh tham chiếu hoặc ảnh hiện tại. Phương pháp 1300 cũng bao gồm, ở hoạt động 1320, thực hiện biến đổi dựa trên việc xác định.

Theo một số phương án thực hiện, việc sử dụng bao gồm liệu công cụ mã có được kích hoạt cho ảnh hiện tại. Theo một số phương án thực hiện, phần ảnh bao gồm cửa sổ tương hợp nằm ngoài cửa sổ mà các mẫu sẵn có khi xuất ra video, cửa sổ chia tỷ lệ trong đó các mẫu tùy thuộc vào hoạt động lấy mẫu lại, hoặc toàn bộ ảnh. Theo một số phương án thực hiện, công cụ mã bao gồm công cụ mã hóa tinh lọc MV phía bộ giải mã, công cụ mã hóa luồng quang hai hướng, công cụ mã hóa tinh lọc dự báo có luồng quang (PROF), hoặc bộ lọc nội suy có độ chính xác nửa điểm ảnh để bù chuyển động. Theo một số phương án thực hiện, công cụ mã còn bao gồm công cụ mã hóa dự báo MV thời gian, công cụ mã hóa nhiều tập biến đổi, công cụ mã hóa ALF giữa các thành phần, công cụ mã hóa phân vùng hình học, quá trình kết hợp cho công cụ tạo mã mà tách khỏi thành nhiều phân vùng, công cụ tạo mã yêu cầu thông tin được lưu trữ trong ảnh khác với ảnh hiện tại, công cụ tạo mã tạo ứng viên hợp nhất theo cặp, dự báo kép có các trọng số mức CU, công cụ mã hóa dự báo có trọng số, công cụ mã hóa dự báo afin, hoặc công cụ mã hóa độ phân giải MV thích ứng.

Theo một số phương án thực hiện, công cụ mã bị vô hiệu hóa nếu điều kiện xác định kích thước trong một phần của ít nhất một trong nhiều ảnh tham chiếu khác với kích thước của ảnh hiện tại được thỏa mãn. Theo một số phương án thực hiện, kích thước của phần được chỉnh sửa để xác định. Theo một số phương án thực hiện, việc sử dụng công cụ mã dựa trên ít nhất một trong (1) chiều rộng của ảnh trừ đi một hoặc nhiều độ lệch ngang, hoặc (2) chiều cao của ảnh trừ đi một hoặc nhiều độ lệch dọc. Theo một số phương án thực hiện, một hoặc nhiều độ lệch ngang bao gồm độ lệch bên trái đối với cửa sổ chia tỷ lệ. Theo một số phương án thực hiện, một

hoặc nhiều độ lệch dọc bao gồm độ lệch trên đối với cửa sổ chia tỷ lệ. Theo một số phương án thực hiện, một hoặc nhiều độ lệch ngang bao gồm độ lệch bên phải và độ lệch bên trái cho cửa sổ chia tỷ lệ. Theo một số phương án thực hiện, một hoặc nhiều độ lệch dọc bao gồm độ lệch dưới và độ lệch trên đối với cửa sổ chia tỷ lệ. Theo một số phương án thực hiện, một hoặc nhiều độ lệch ngang bao gồm giá trị, dựa trên, việc chia tỷ lệ tổng của độ lệch bên phải và độ lệch bên trái đối với cửa sổ chia tỷ lệ. Theo một số phương án thực hiện, một hoặc nhiều độ lệch dọc bao gồm giá trị dựa trên việc chia tỷ lệ tổng độ lệch dưới và độ lệch trên đối với cửa sổ chia tỷ lệ.

Theo một số phương án thực hiện, công cụ mã bị vô hiệu hóa nếu điều kiện xác định các kích thước của các phần của ít nhất hai trong nhiều ảnh tham chiếu là khác nhau được thỏa mãn. Theo một số phương án thực hiện, ứng viên hợp nhất theo cặp thứ nhất được đánh dấu là không khả dụng nếu các kích thước của các phần của ít nhất hai trong nhiều ảnh tham chiếu được sử dụng để dẫn xuất ứng viên hợp nhất theo cặp thứ nhất đối với một hoặc hai danh sách ảnh tham chiếu là khác nhau. Theo một số phương án thực hiện, phần tử cú pháp chỉ báo việc sử dụng công cụ mã được báo hiệu có điều kiện trong biểu diễn dòng bit dựa trên các kích thước của các phần. Theo một số phương án thực hiện, phần tử cú pháp chỉ báo việc sử dụng công cụ mã bị bỏ qua trong biểu diễn dòng bit.

Theo một số phương án thực hiện, phần tử cú pháp chỉ báo việc sử dụng công cụ mã được bao gồm có điều kiện trong biểu diễn dòng bit còn dựa trên liệu quá trình lấy mẫu lại ảnh tham chiếu (RPR) được kích hoạt. Theo một số phương án thực hiện, phần tử cú pháp bao gồm một trong `amvr_precision_idx`, `sym_mvd_flag`, hoặc `mmvd_merge_flag`. Theo một số phương án thực hiện, phép biến đổi theo việc sử dụng công cụ mã được chỉnh sửa dựa trên các kích thước của các phần. Theo một số phương án thực hiện, phép biến đổi bao gồm bước dẫn xuất MVD cho danh sách ảnh tham chiếu trong quá trình MVD đối xứng, việc dẫn xuất dựa trên hiệu số của các kích thước của các phần của ít nhất hai trong nhiều ảnh tham chiếu. Theo một số phương án thực hiện, phép biến đổi bao gồm bước dẫn xuất ứng viên hợp nhất theo cặp, và việc dẫn xuất bao gồm áp dụng trung bình có trọng số dựa trên hiệu số của các kích thước của các phần của ít nhất hai trong nhiều ảnh tham chiếu.

Fig.14 là lưu đồ biểu diễn phương pháp 1400 xử lý video theo sáng chế. Phương pháp 1400 bao gồm, ở hoạt động 1410, thực hiện biến đổi giữa khối của video và biểu diễn dòng bit của video. Trong khi biến đổi AMVR có vectơ chuyển động độ chính xác 1/2-pel được kích hoạt cho khối do quá trình lấy mẫu lại ảnh tham chiếu có thể áp dụng cho khối.

Theo một số phương án thực hiện, bộ lọc nội suy khác nhau được sử dụng cho các khối có độ chính xác khác ngoài độ chính xác $\frac{1}{2}$ pel. Theo một số phương án thực hiện, điều kiện được liên kết với kích thước của phần ảnh tham chiếu được chỉ báo bởi cờ được liên kết với ảnh tham chiếu. Theo một số phương án thực hiện, công cụ mã bị vô hiệu hóa nếu cờ chỉ báo rằng ít nhất một ảnh tham chiếu trong danh sách ảnh tham chiếu được chia tỷ lệ. Theo một số phương án thực hiện, cờ bao gồm RefPicIsScaled[0][refIdxL0], trong đó refIdxL0 là chỉ số cho danh sách ảnh tham chiếu thứ nhất. Theo một số phương án thực hiện, cờ bao gồm RefPicIsScaled[0][refIdxL1], trong đó refIdxL1 là chỉ số cho danh sách ảnh tham chiếu thứ hai. Theo một số phương án thực hiện, cờ được xác định trong quá trình tạo danh sách ảnh tham chiếu.

Fig.15 là lưu đồ biểu diễn phương pháp 1500 xử lý video theo sáng chế. Phương pháp 1500 bao gồm, ở hoạt động 1510, xác định, đối với biến đổi giữa khối của video và biểu diễn dòng bit của video, tổng của hiệu số tuyệt đối được sử dụng trong công cụ mã hóa luồng quang hai hướng (BDOF) hoặc công cụ mã hóa DMVR theo cách thức dựa trên chiều sâu bit của khối. Phương pháp 1500 cũng bao gồm, ở hoạt động 1520, thực hiện biến đổi dựa trên việc xác định.

Theo một số phương án thực hiện, tổng của hiệu số tuyệt đối được xê dịch dựa trên hàm của chiều sâu bit trước khi được so sánh với ngưỡng cho công cụ mã hóa BDOF hoặc công cụ mã hóa DMVR. Theo một số phương án thực hiện, tổng của hiệu số tuyệt đối được so sánh với ngưỡng được chỉnh sửa dựa trên hàm của chiều sâu bit.

Theo một số phương án thực hiện, phép biến đổi bao gồm mã hóa video thành biểu diễn dòng bit. Theo một số phương án thực hiện, phép biến đổi bao gồm giải mã biểu diễn dòng bit thành video.

Một số phương án thực hiện sáng chế bao gồm quyết định kích hoạt công cụ hoặc chế độ xử lý video. Trong ví dụ, khi công cụ hoặc chế độ xử lý video được kích hoạt, bộ mã hóa sẽ sử dụng hoặc triển khai công cụ hoặc chế độ khi xử lý khối video, nhưng có thể không nhất thiết chỉnh sửa dòng bit thu được dựa trên việc sử dụng công cụ hoặc chế độ. Tức là, phép biến đổi từ khối video sang biểu diễn dòng bit của video sẽ sử dụng công cụ hoặc chế độ xử lý video khi được kích hoạt dựa trên quyết định này. Trong ví dụ khác, khi công cụ hoặc chế độ xử lý video được kích hoạt, bộ giải mã sẽ xử lý dòng bit với nhận thức rằng dòng bit đã được chỉnh sửa dựa trên công cụ hoặc chế độ xử lý video. Tức là, phép biến đổi từ biểu diễn dòng bit của video sang khối video sẽ được thực hiện nhờ sử dụng công cụ hoặc chế độ xử lý video được kích hoạt dựa trên quyết định.

Một số phương án thực hiện sáng chế bao gồm quyết định để vô hiệu hóa công cụ hoặc chế độ xử lý video. Trong ví dụ, khi công cụ hoặc chế độ xử lý video bị vô hiệu hóa, bộ mã hóa sẽ không sử dụng công cụ hoặc chế độ trong phép biến đổi của khối video sang biểu diễn dòng bit của video. Trong ví dụ khác, khi công cụ hoặc chế độ xử lý video bị vô hiệu hóa, bộ giải mã sẽ xử lý dòng bit với nhận thức rằng dòng bit đã không được chỉnh sửa nhờ sử dụng công cụ hoặc chế độ xử lý video được kích hoạt dựa trên quyết định.

Các giải pháp, các ví dụ, các phương án thực hiện, các mô đun và các hoạt động chức năng được bộc lộ và khác được mô tả trong tài liệu có thể được triển khai trong hệ mạch điện tử số, hoặc trong phần mềm máy tính, firmware, hoặc phần cứng, bao gồm các cấu trúc được bộc lộ trong tài liệu và các tương đương về mặt cấu trúc, hoặc kết hợp của một hoặc nhiều trong số chúng. Các phương án thực hiện được bộc lộ và các phương án khác có thể được triển khai dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, chẳng hạn, một hoặc nhiều mô đun lệnh chương trình máy tính được mã hóa trên phương tiện máy tính đọc được để thực thi bởi, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Phương tiện máy tính đọc được có thể là thiết bị lưu trữ máy đọc được, nền tảng lưu trữ máy đọc được, thiết bị nhớ, hợp chất thực hiện tạo thành tín hiệu được lan truyền máy đọc được, hoặc tổ hợp của một hoặc nhiều trong số chúng. Cụm từ “thiết bị xử lý dữ liệu” bao gồm tất cả các thiết bị, bộ phận, và máy để xử lý dữ liệu, bao gồm thông qua ví dụ, bộ

xử lý lập trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, bên cạnh phần cứng, mã tạo môi trường thực thi cho chương trình máy tính đang chạy, chẳng hạn, mã mà tạo firmware bộ xử lý, xếp chồng giao thức, hệ thống quản trị cơ sở dữ liệu, hệ điều hành, hoặc kết hợp của một hoặc nhiều trong số chúng. Tín hiệu được lan truyền là tín hiệu được nhân tạo, chẳng hạn, tín hiệu điện được tạo bằng máy, quang học, hoặc điện từ, được tạo để mã hóa thông tin để truyền đến bộ phận nhận thích hợp.

Chương trình máy tính (cũng được biết đến như là chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết ở dạng ngôn ngữ lập trình bất kỳ, bao gồm các ngôn ngữ biên dịch hoặc thông dịch, và có thể được triển khai ở dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, trình con, hoặc đơn vị khác thích hợp để sử dụng trong môi trường tính toán. Chương trình máy tính không nhất thiết tương ứng với tệp tin trong hệ thống tệp tin. Chương trình có thể được lưu trữ trong phần tệp tin chứa các chương trình hoặc dữ liệu khác (chẳng hạn, một hoặc nhiều tệp tin được lưu trữ trong tài liệu ngôn ngữ đánh dấu), trong một tệp tin dành cho chương trình đang chạy, hoặc trong nhiều tệp tin phối hợp (chẳng hạn, các tệp tin mà lưu trữ một hoặc nhiều môđun, chương trình phụ, hoặc các phần mã). Chương trình máy tính có thể được khai triển để được thực thi trên một máy tính hoặc trên nhiều máy tính được đặt ở một điểm hoặc được phân tán giữa nhiều địa điểm và được liên kết bằng mạng truyền thông.

Các tiến trình và các luồng logic được mô tả trong tài liệu có thể được thực hiện bằng một hoặc nhiều bộ xử lý lập trình được thực thi một hoặc nhiều chương trình máy tính để thực hiện các chức năng bằng cách hoạt động trên dữ liệu đầu vào và tạo đầu ra. Các tiến trình và các luồng logic cũng có thể được thực hiện bởi, và thiết bị cũng có thể được triển khai dưới dạng, hệ mạch logic chuyên dụng, chẳng hạn, FPGA (mảng cổng dạng trường lập trình được) hoặc ASIC (mạch tích hợp ứng dụng cụ thể).

Các bộ xử lý thích hợp để thực thi chương trình máy tính bao gồm, thông qua ví dụ, cả bộ vi xử lý đa năng và chuyên dụng, và bất kỳ một hoặc nhiều bộ xử lý của loại máy tính số bất kỳ. Nói chung, bộ xử lý sẽ nhận các lệnh và dữ liệu từ bộ nhớ chỉ đọc hoặc bộ nhớ truy nhập ngẫu nhiên hoặc cả hai. Các phần tử cơ bản của

máy tính là bộ xử lý để thực thi các lệnh và một hoặc nhiều thiết bị nhớ để lưu trữ lệnh và dữ liệu. Nói chung, máy tính cũng sẽ bao gồm, hoặc được ghép nối vận hành để nhận dữ liệu từ hoặc truyền dữ liệu đến, hoặc cả hai, một hoặc nhiều bộ phận lưu trữ dữ liệu lớn để lưu trữ dữ liệu, chẳng hạn, đĩa từ, đĩa quang từ, hoặc đĩa quang. Tuy nhiên, máy tính không cần có các thiết bị này. Các phương tiện máy tính đọc được thích hợp để lưu trữ các lệnh chương trình máy tính và dữ liệu bao gồm tất cả các dạng của bộ nhớ bất biến, phương tiện và thiết bị nhớ, bao gồm thông qua ví dụ các thiết bị nhớ bán dẫn, chẳng hạn, EPROM, EEPROM, và các thiết bị nhớ nhanh, đĩa từ, chẳng hạn, đĩa cứng bên trong hoặc đĩa tháo được; đĩa quang từ; và đĩa CD ROM và DVD-ROM. Bộ xử lý và bộ nhớ có thể được bổ sung bằng, hoặc được đưa vào trong, hệ mạch logic chuyên dụng.

Trong khi sáng chế có nhiều đặc tả, các đặc tả này không được hiểu như là các giới hạn ở phạm vi của đối tượng hoặc phần có thể được bảo hộ, nhưng dưới dạng các phần mô tả của các dấu hiệu mà có thể riêng cho các phương án thực hiện cụ thể của các phương án cụ thể. Các dấu hiệu cụ thể được mô tả theo sáng chế trong ngữ cảnh của các phương án thực hiện riêng rẽ cũng có thể được triển khai kết hợp theo một phương án thực hiện. Ngược lại, các dấu hiệu khác nhau được mô tả trong ngữ cảnh của một phương án thực hiện cũng có thể được triển khai theo nhiều phương án thực hiện riêng rẽ hoặc trong tổ hợp phụ thích hợp bất kỳ. Ngoài ra, mặc dù các dấu hiệu có thể được mô tả trên đây như là hoạt động trong các tổ hợp cụ thể và thậm chí được bảo hộ ban đầu, một hoặc nhiều dấu hiệu từ tổ hợp được bảo hộ có thể trong một số trường hợp được thực thi từ tổ hợp này, và tổ hợp được bảo hộ có thể dành cho tổ hợp phụ hoặc biến thể của tổ hợp phụ.

Tương tự, trong khi các hoạt động được mô tả trong các hình vẽ theo thứ tự cụ thể, điều này không được hiểu như là yêu cầu rằng các hoạt động này được thực hiện theo thứ tự cụ thể được thể hiện hoặc theo thứ tự lần lượt, hoặc tất cả các hoạt động được minh họa được thực hiện, để có các kết quả mong muốn. Ngoài ra, việc tách riêng các thành phần khác nhau của hệ thống theo các phương án thực hiện được mô tả theo sáng chế sẽ không được hiểu như là yêu cầu tách riêng theo tất cả các phương án thực hiện.

Chỉ vài cách triển khai và ví dụ được mô tả và các triển khai khác, các tăng cường và các biến thể có thể được thực hiện dựa trên phần được mô tả và được minh họa trên sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý video bao gồm các bước:

xác định, đối với phép biến đổi giữa khối hiện tại của ảnh hiện tại của video và dòng bit của video, liệu có vô hiệu hóa công cụ tạo mã tinh lọc để tinh lọc vector chuyển động hoặc mẫu dự báo đối với khối hiện tại bởi giá trị độ lệch; và

thực hiện biến đổi dựa trên bước xác định,

trong đó công cụ tạo mã tinh lọc bị vô hiệu hóa khi kích thước của ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

2. Phương pháp theo điểm 1, trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ nhất, và

trong đó công cụ tạo mã tinh lọc thứ nhất tinh lọc vector chuyển động được báo hiệu dựa trên ít nhất một vector chuyển động có độ lệch với vector chuyển động được báo hiệu.

3. Phương pháp theo điểm 1, trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ hai, và

trong đó công cụ tạo mã tinh lọc thứ hai nhận tinh lọc vector chuyển động dựa trên ít nhất một giá trị gradient tương ứng với mẫu trong khối tham chiếu của khối hiện tại.

4. Phương pháp theo điểm 1, trong đó chế độ affin được áp dụng cho khối hiện tại và các mẫu dự báo ban đầu của khối phụ của khối hiện tại được tạo,

trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ ba,

trong đó công cụ tạo mã tinh lọc thứ ba được áp dụng cho tạo các mẫu dự báo cuối cùng đối với khối phụ bằng cách dẫn xuất tinh lọc dự báo dựa trên các chênh lệch vector chuyển động dMvH và/hoặc dMvV, và

trong đó dMvH và dMvV chỉ báo các chênh lệch vector chuyển động dọc theo phương ngang và phương dọc.

5. Phương pháp theo điểm 4, trong đó công cụ tạo mã tinh lọc thứ ba bị vô hiệu hóa đối với danh sách tham chiếu thứ nhất của khối hiện tại khi kích thước của ảnh tham

chiều trong danh sách tham chiếu thứ nhất khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách tham chiếu thứ nhất khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại, và

trong đó công cụ tạo mã tĩnh lọc thứ ba bị vô hiệu hóa đối với danh sách ảnh tham chiếu thứ hai của khối hiện tại khi kích thước của ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

6. Phương pháp theo điểm 1, trong đó liệu công cụ tạo mã tĩnh lọc bị vô hiệu hóa được xác định dựa trên các giá trị của một hoặc nhiều cờ lần lượt được liên kết với một hoặc nhiều ảnh tham chiếu.

7. Phương pháp theo điểm 6, trong đó khi kích thước của ảnh tham chiếu trong danh sách tham chiếu thứ nhất of khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách tham chiếu thứ nhất của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại, giá trị của cờ thứ nhất được gán không bằng 0, và

khi kích thước của ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại, giá trị của cờ thứ hai được gán không bằng 0.

8. Phương pháp theo điểm 6, trong đó một hoặc nhiều cờ được dẫn xuất trong quá trình tạo danh sách ảnh tham chiếu.

9. Phương pháp theo điểm 1, trong đó kích thước của cửa sổ chia tỷ lệ được xác định dựa trên ít nhất một của độ lệch bên trái được liên kết với cửa sổ tỷ lệ, độ lệch bên phải được liên kết với cửa sổ tỷ lệ, độ lệch dưới được liên kết với cửa sổ tỷ lệ, hoặc độ lệch trên được liên kết với cửa sổ tỷ lệ.

10. Phương pháp theo điểm 1, trong đó độ phân giải vector chuyển động thích ứng được kích hoạt đối với khối hiện tại khi kích thước của ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu của một hoặc nhiều ảnh

tham chiếu của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

11. Phương pháp theo điểm 1, trong đó phép biến đổi bao gồm bước mã hóa ảnh hiện tại thành dòng bit.

12. Phương pháp theo điểm 1, trong đó phép biến đổi bao gồm bước giải mã ảnh hiện tại từ dòng bit.

13. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó, trong đó các lệnh khi thực thi bằng bộ xử lý, khiến bộ xử lý:

xác định, đối với phép biến đổi giữa khối hiện tại của ảnh hiện tại của video và dòng bit của video, liệu có vô hiệu hóa công cụ tạo mã tinh lọc để tinh lọc vector chuyển động hoặc mẫu dự báo đối với khối hiện tại bởi giá trị độ lệch; và

thực hiện phép biến đổi dựa trên bước xác định,

trong đó công cụ tạo mã tinh lọc bị vô hiệu hóa khi kích thước của ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

14. Thiết bị theo điểm 13, trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ nhất, và

trong đó công cụ tạo mã tinh lọc thứ nhất tinh lọc vector chuyển động được báo hiệu dựa trên ít nhất một vector chuyển động có độ lệch với vector chuyển động được báo hiệu.

15. Thiết bị theo điểm 13, trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ hai, và

trong đó công cụ tạo mã tinh lọc thứ hai nhận tinh lọc vector chuyển động dựa trên ít nhất một giá trị gradien tương ứng với mẫu trong khối tham chiếu của khối hiện tại.

16. Thiết bị theo điểm 13, trong đó chế độ affin được áp dụng cho khối hiện tại và các mẫu dự báo ban đầu của khối phụ của khối hiện tại được tạo,

trong đó công cụ tạo mã tinh lọc bao gồm công cụ tạo mã tinh lọc thứ ba,

trong đó công cụ tạo mã tinh lọc thứ ba được áp dụng cho tạo các mẫu dự báo cuối cùng đối với khối phụ bằng cách dẫn xuất tinh lọc dự báo dựa trên các chênh lệch vector chuyển động dMvH và/or dMvV, và

trong đó dMvH và dMvV chỉ báo các chênh lệch vector chuyển động dọc theo phương ngang và phương dọc.

17. Thiết bị theo điểm 16, trong đó công cụ tạo mã tinh lọc thứ ba bị vô hiệu hóa đối với danh sách tham chiếu thứ nhất của khối hiện tại khi kích thước của ảnh tham chiếu trong danh sách tham chiếu thứ nhất khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách tham chiếu thứ nhất khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại, và

trong đó công cụ tạo mã tinh lọc thứ ba bị vô hiệu hóa đối với danh sách ảnh tham chiếu thứ hai của khối hiện tại khi kích thước của ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu trong danh sách ảnh tham chiếu thứ hai khác với kích thước của cửa sổ chia tỷ lệ in ảnh hiện tại.

18. Thiết bị theo điểm 13, trong đó liệu công cụ tạo mã tinh lọc bị vô hiệu hóa được xác định dựa trên các giá trị của một hoặc nhiều cờ lần lượt được liên kết với một hoặc nhiều ảnh tham chiếu.

19. Bộ ghi máy tính đọc được bất biến lưu trữ các lệnh khiến bộ xử lý:

xác định, đối với phép biến đổi giữa khối hiện tại của ảnh hiện tại của video và dòng bit của video, liệu có vô hiệu hóa công cụ tạo mã tinh lọc để tinh lọc vector chuyển động hoặc mẫu dự báo đối với khối hiện tại bởi giá trị độ lệch; và

thực hiện phép biến đổi dựa trên bước xác định,

trong đó công cụ tạo mã tinh lọc bị vô hiệu hóa khi kích thước của ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

20. Vật ghi máy tính đọc được bất biến lưu trữ dòng bit của video được tạo bởi phương pháp được thực hiện bởi thiết bị xử lý video, trong đó phương pháp bao gồm các bước:

xác định, đối với khối hiện tại của ảnh hiện tại của video, liệu có vô hiệu hóa công cụ tạo mã tinh lọc để tinh lọc vector chuyển động hoặc mẫu dự báo đối với khối hiện tại bởi giá trị độ lệch; và

tạo dòng bit dựa trên bước xác định,

trong đó công cụ tạo mã tinh lọc bị vô hiệu hóa khi kích thước của ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của ảnh hiện tại, hoặc kích thước của cửa sổ chia tỷ lệ trong ảnh tham chiếu của một hoặc nhiều ảnh tham chiếu của khối hiện tại khác với kích thước của cửa sổ chia tỷ lệ trong ảnh hiện tại.

1/17

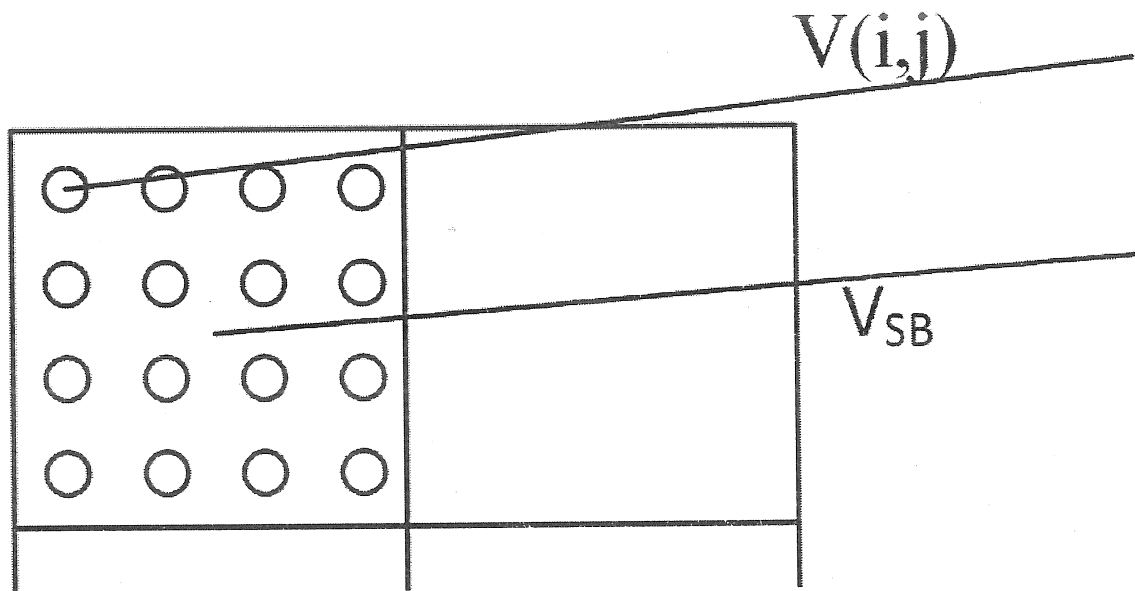


Fig.1

2/17

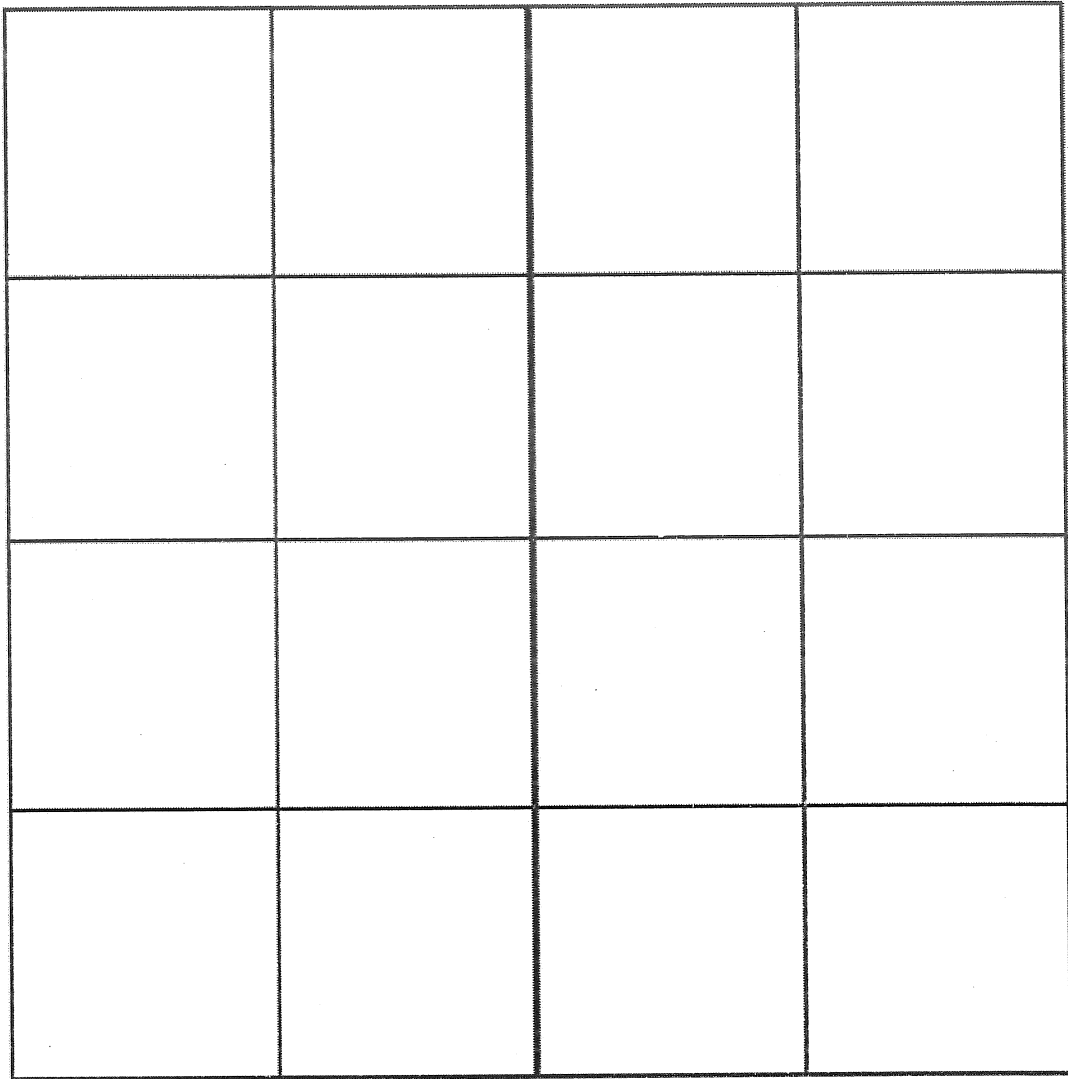


Fig.2

3/17

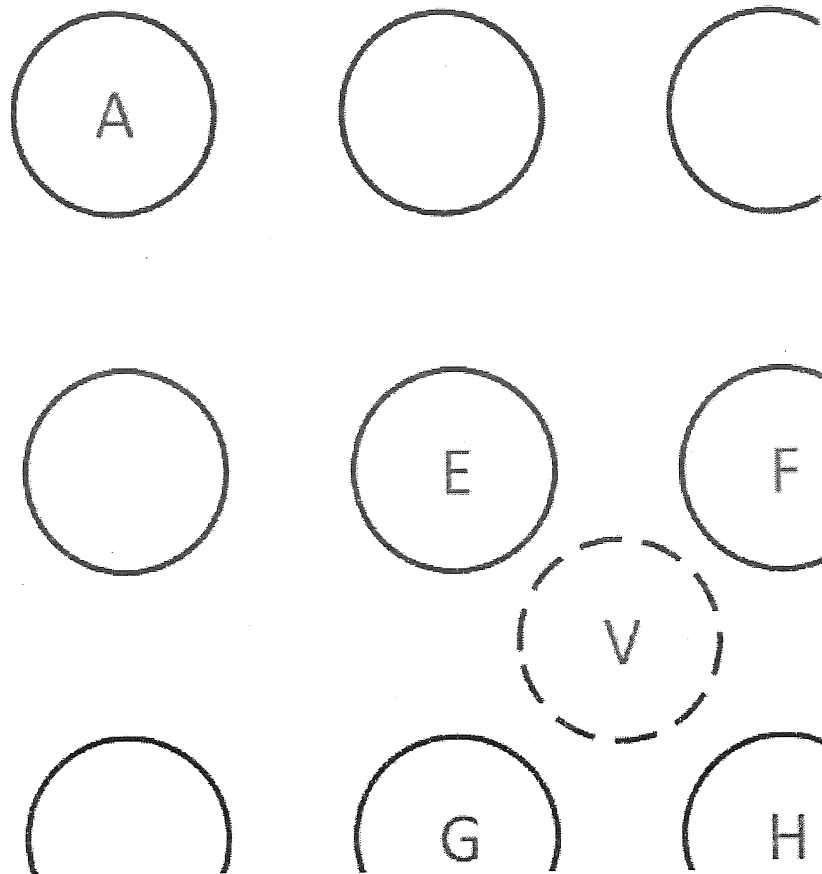


Fig.3A

4/17



Fig.3B

Fig.3C



5/17

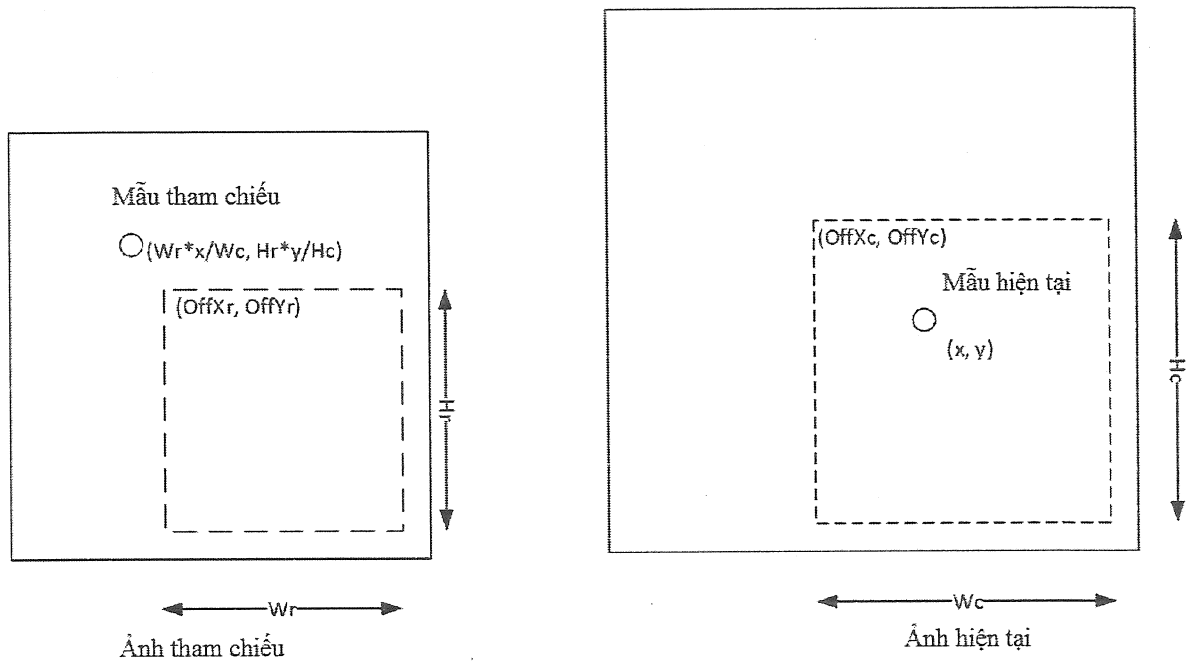


Fig.4A

6/17

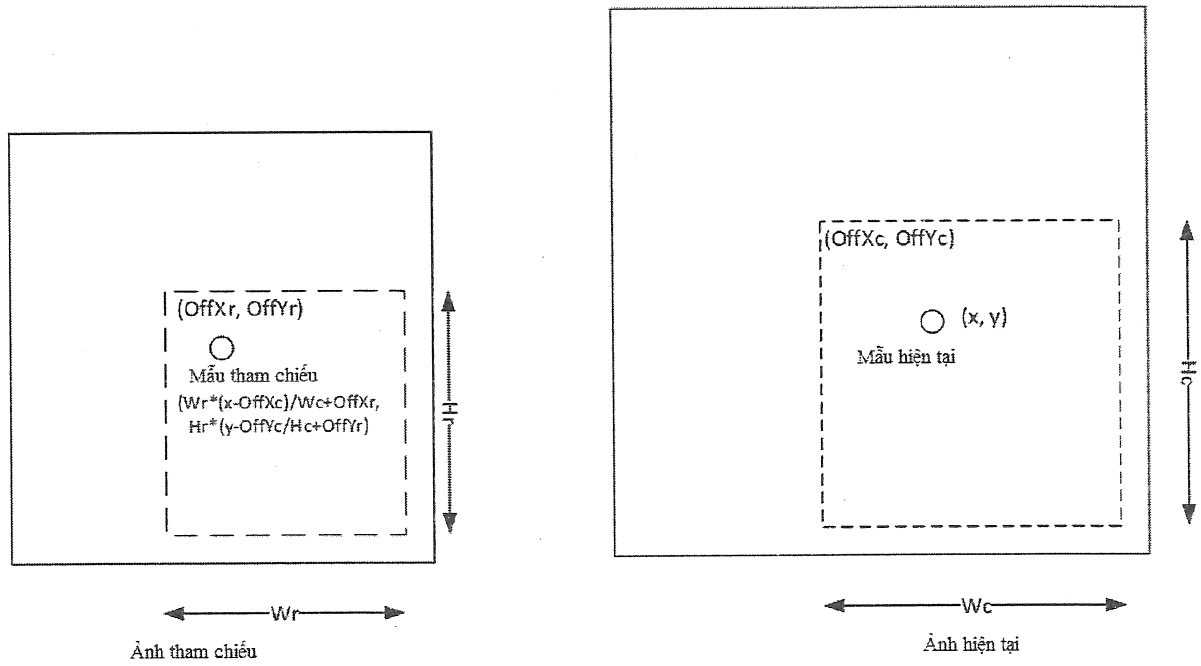


Fig.4B

7/17

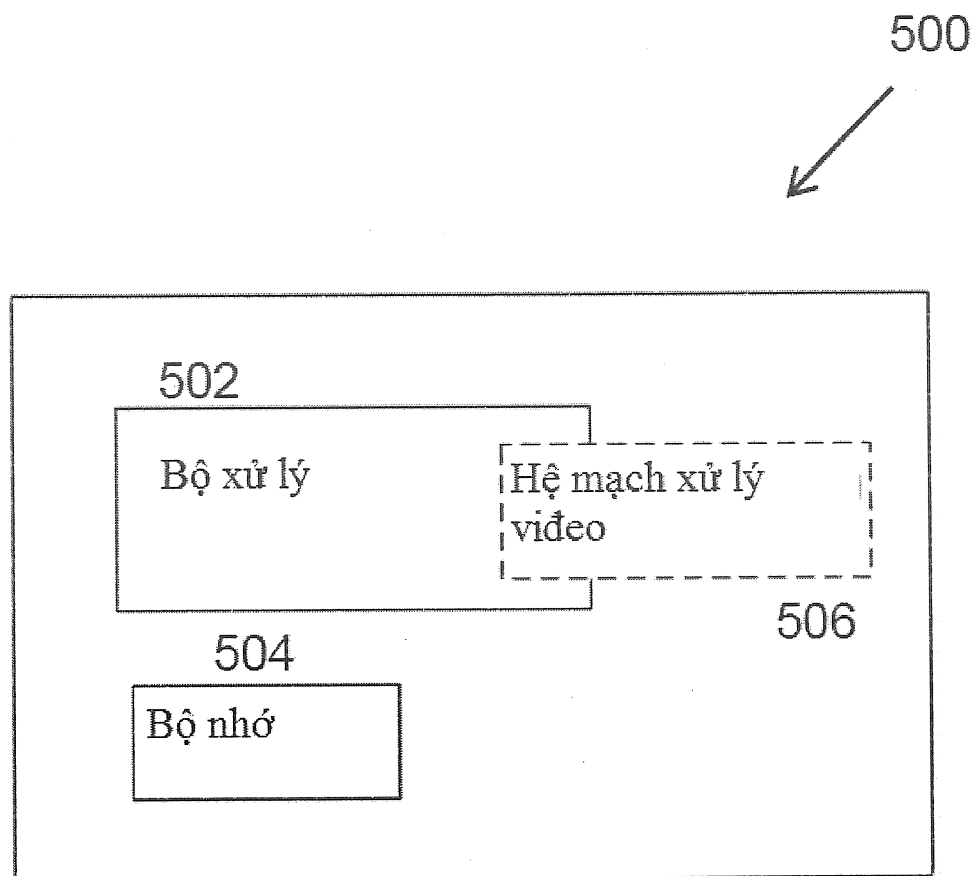


Fig.5

8/17

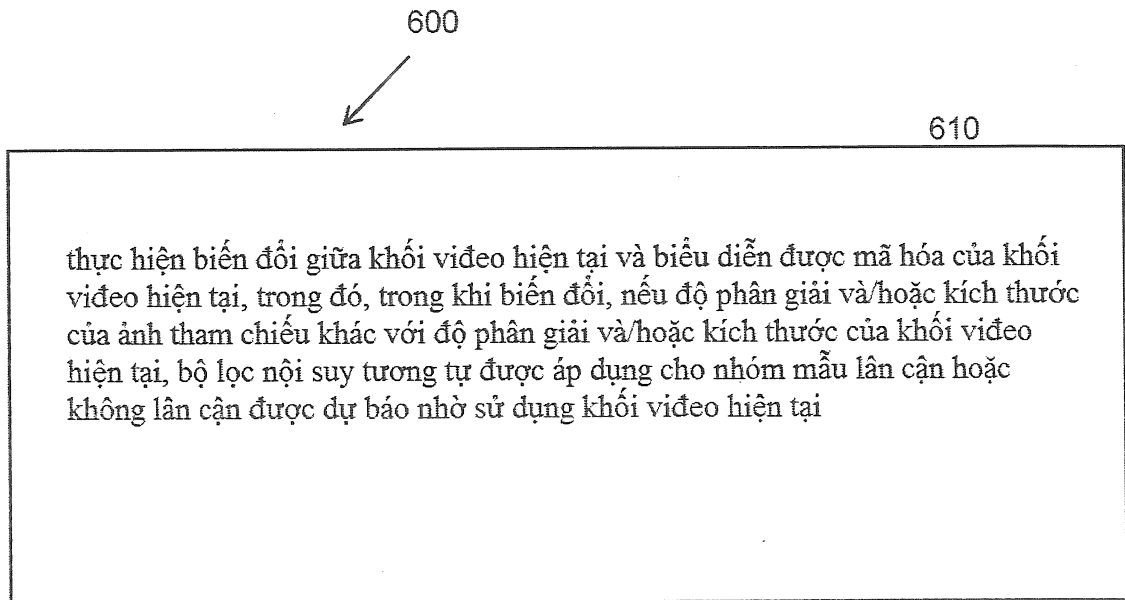


Fig.6

9/17

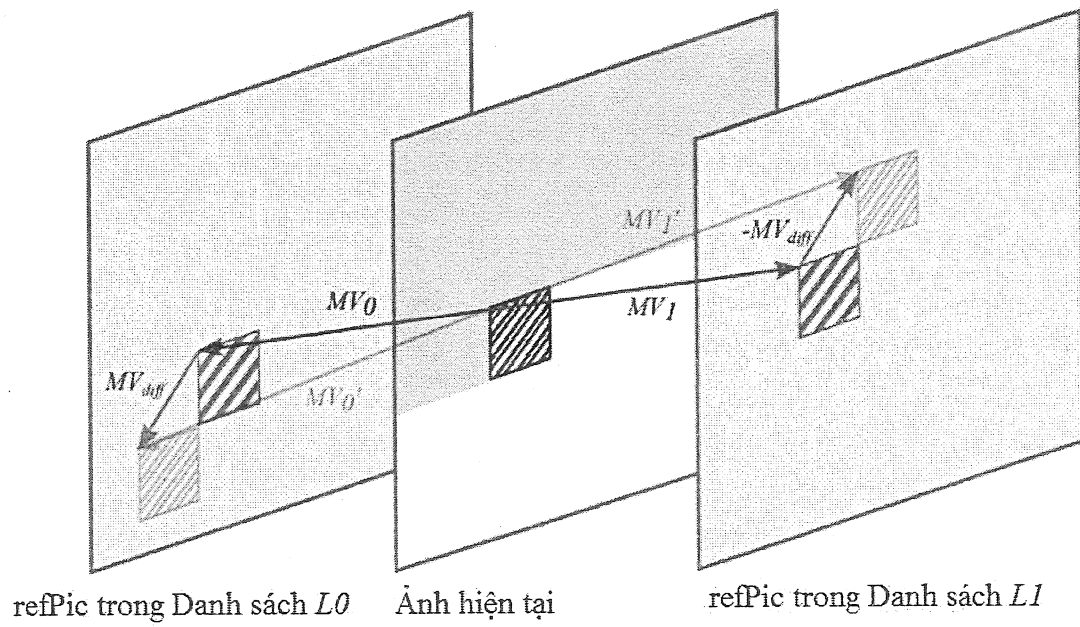


Fig.7

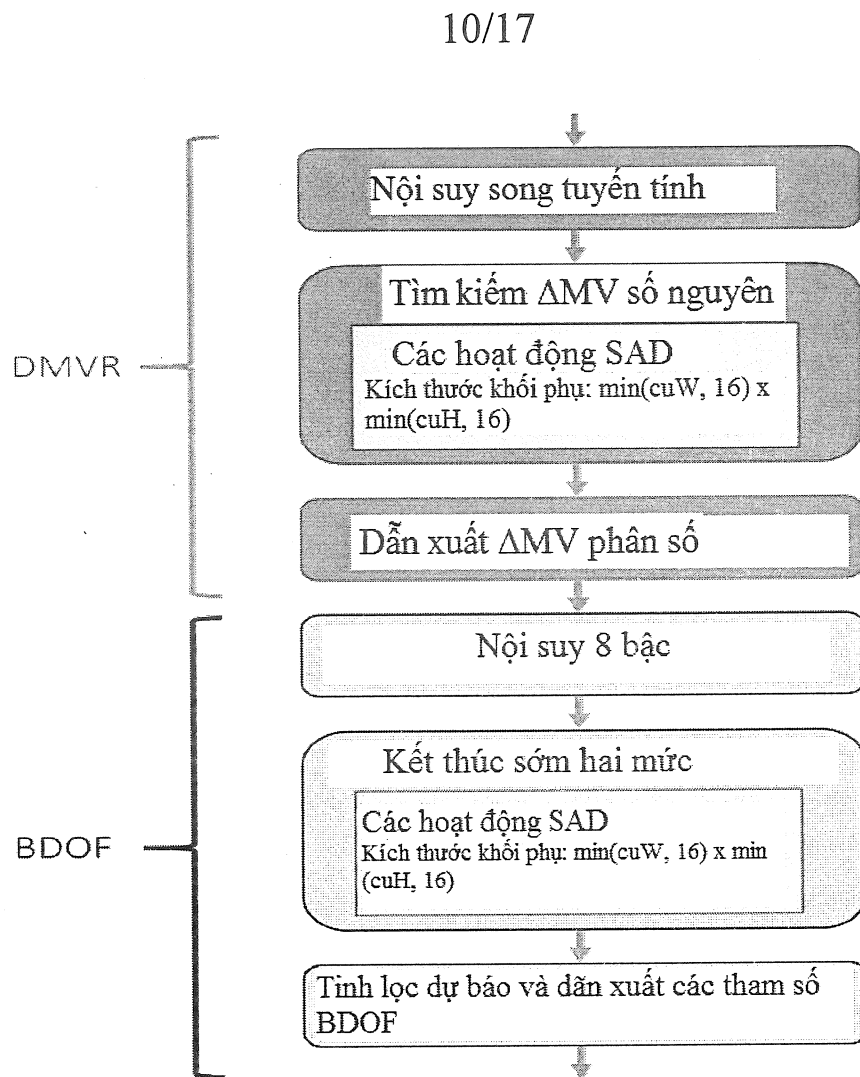


Fig.8

11/17

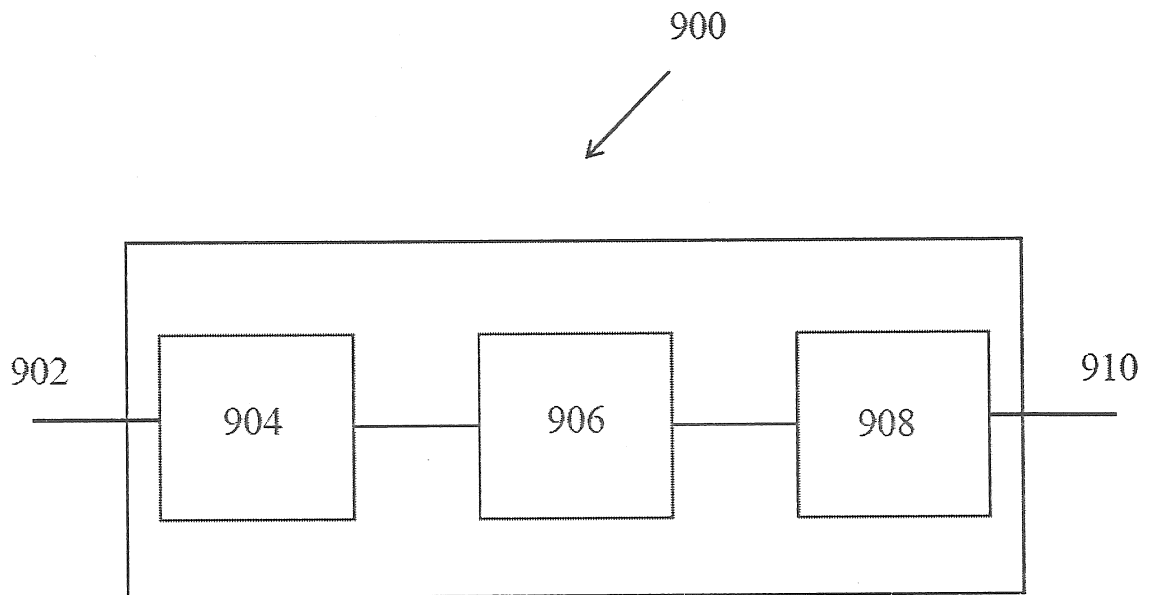


Fig.9

12/17

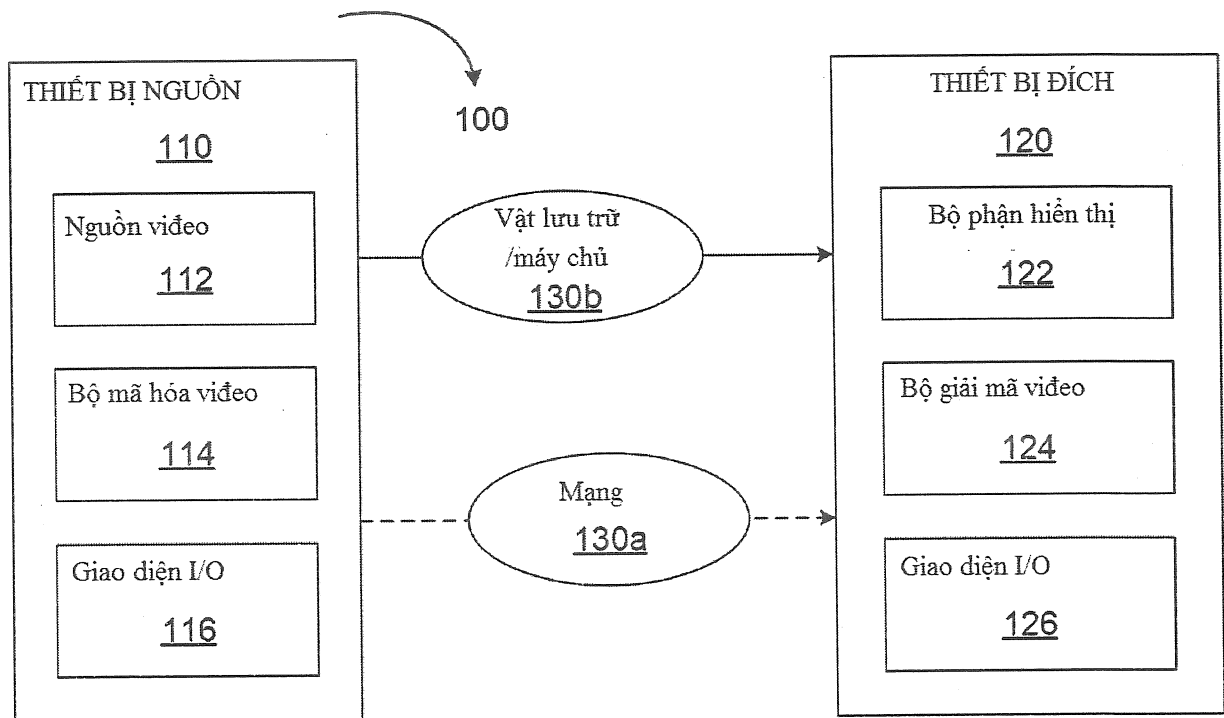


Fig.10

13/17

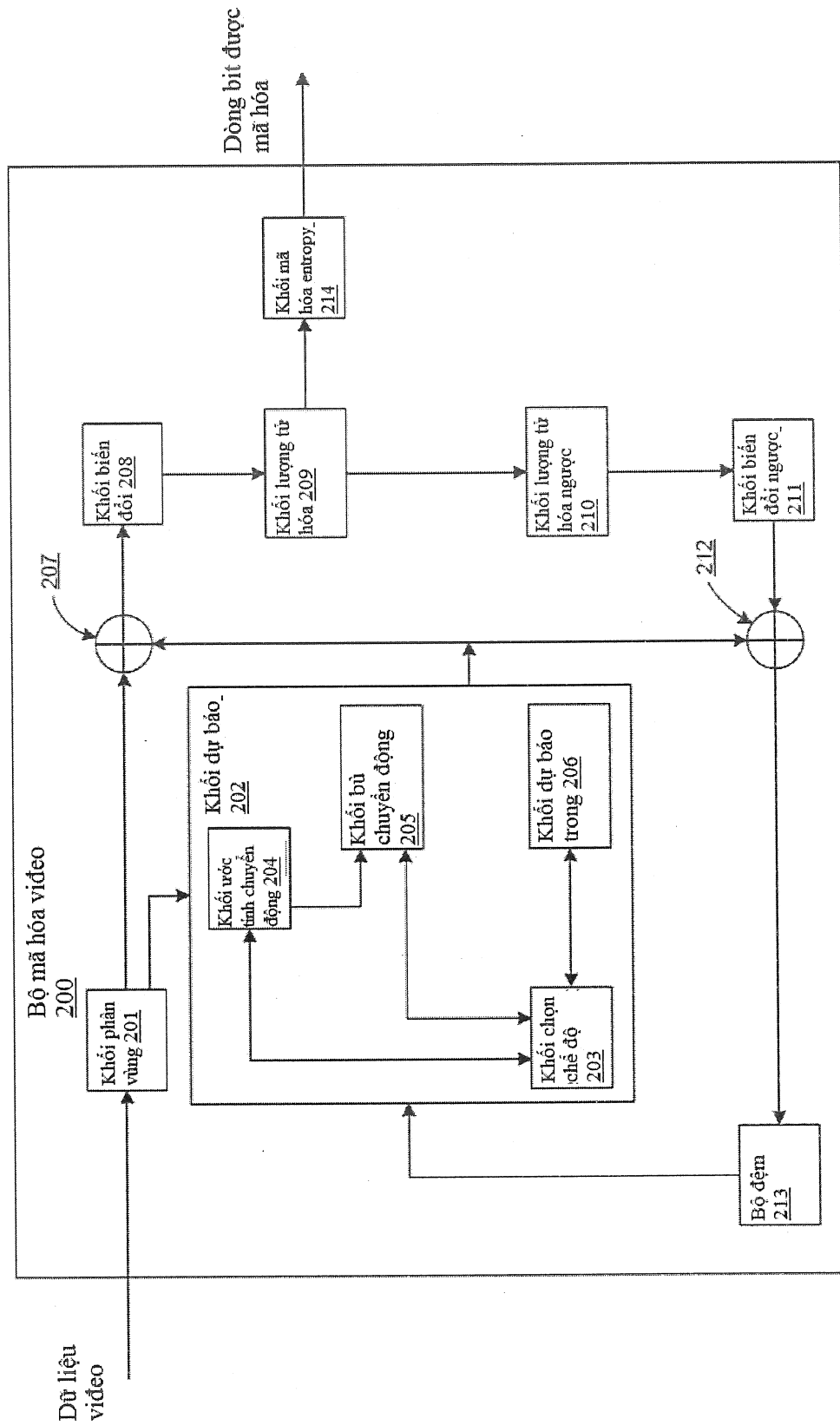


Fig.11

14/17

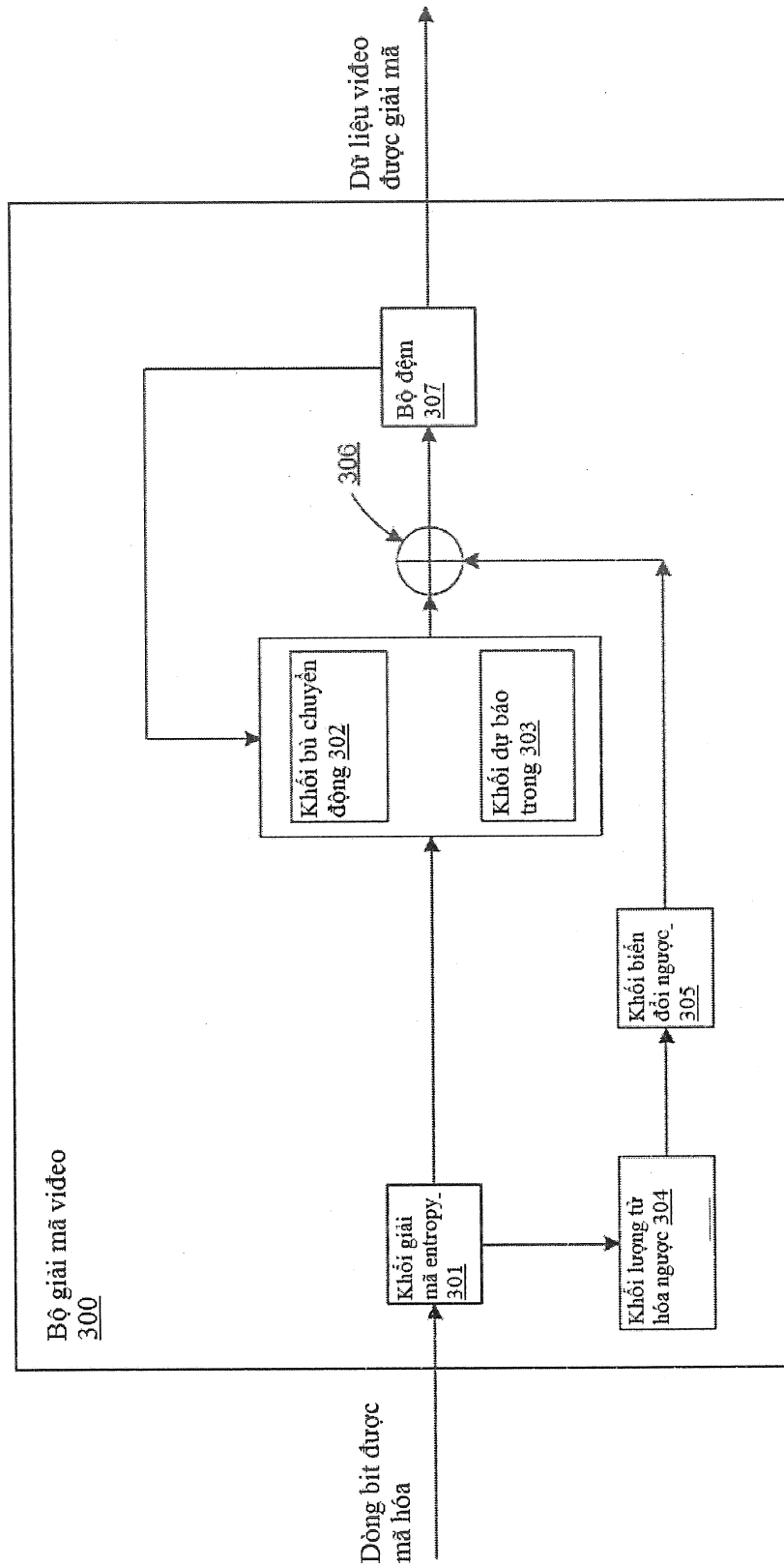


Fig.12

15/17

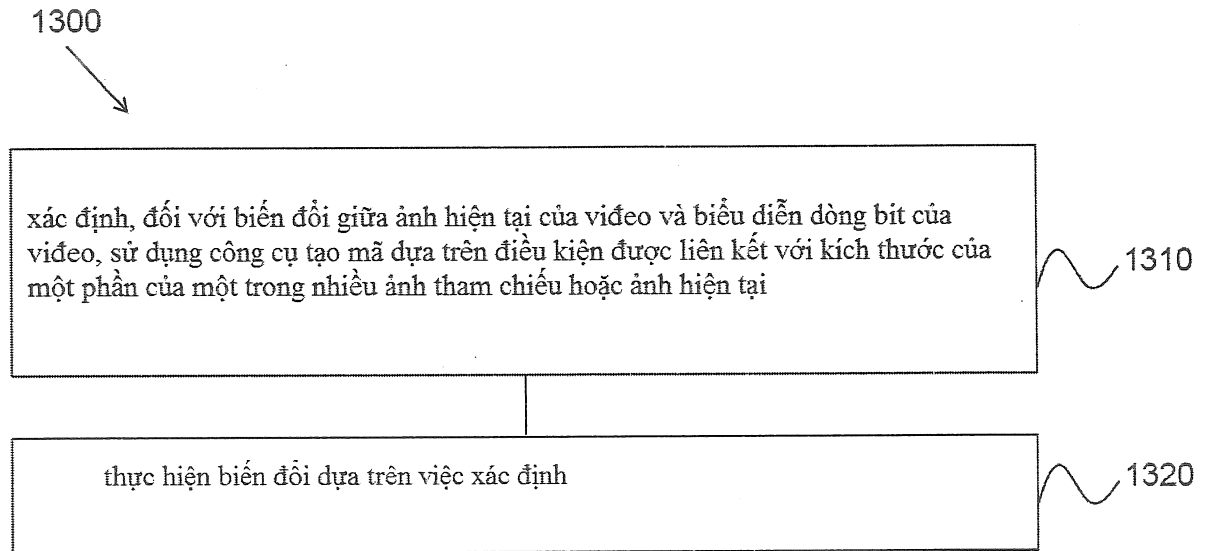


Fig.13

16/17

1400

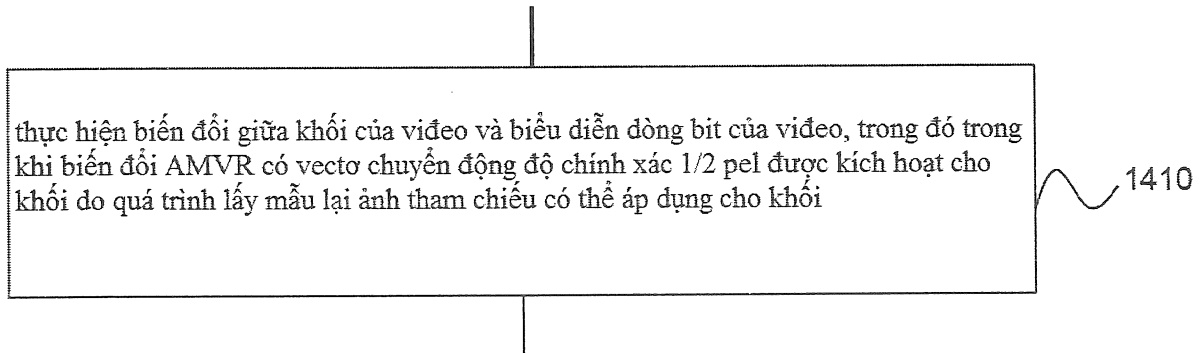


Fig.14

17/17

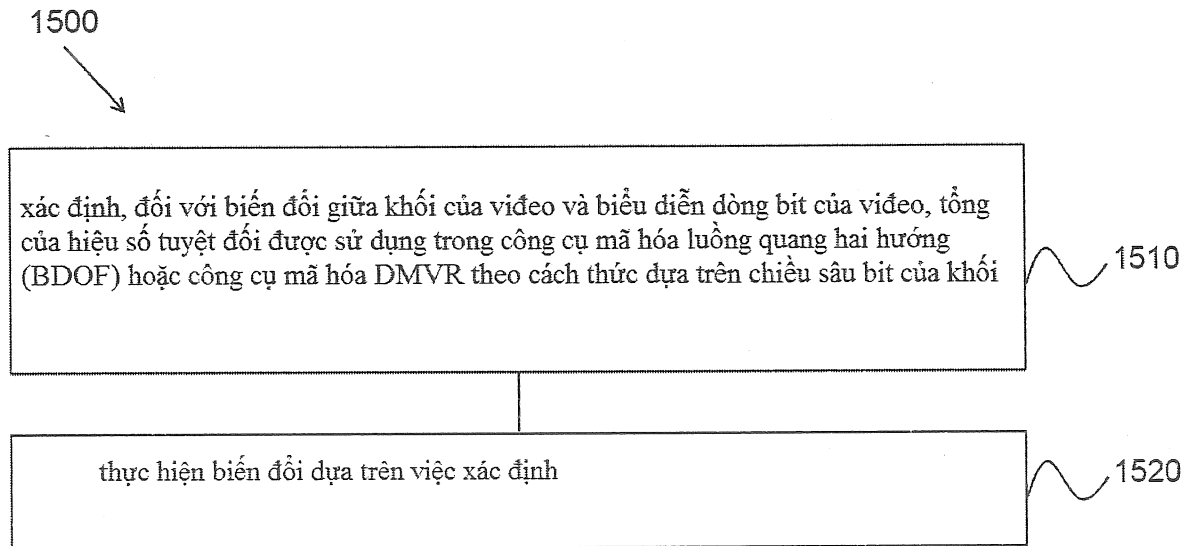


Fig.15