



- (12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0051561
(51)^{2021.01} H04N 19/70; H04N 19/80; H04N 19/174; H04N 19/186 (13) B
-

- (21) 1-2022-05589 (22) 15/01/2021
(86) PCT/CN2021/071969 15/01/2021 (87) WO 2021/155740 12/08/2021
(30) PCT/EP2020/052682 04/02/2020 EP; PCT/EP2020/059031 31/03/2020 EP
(45) 25/09/2025 450 (43) 25/11/2022 416A
(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, China
(72) KOTRA, Anand Meher (IN); ESENLIK, Semih (TR); WANG, Biao (CN); GAO,
Han (CN); ALSHINA, Elena Alexandrovna (RU).
(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)
-

- (54) PHƯƠNG PHÁP TẠO MÃ, BỘ GIẢI MÃ, THIẾT BỊ GIẢI MÃ VÀ VẬT GHI
MÁY TÍNH ĐỌC ĐƯỢC BẤT BIẾN

(21) 1-2022-05589

(57) Sáng chế đề cập đến phương pháp tạo mã được triển khai bởi thiết bị giải mã, phương pháp bao gồm thu được dòng bit cho khối mã; thu được giá trị của cú pháp từ dòng bit; thu được giá trị của tham số điều khiển tách khối từ dòng bit, khi giá trị của cú pháp bằng giá trị định trước.

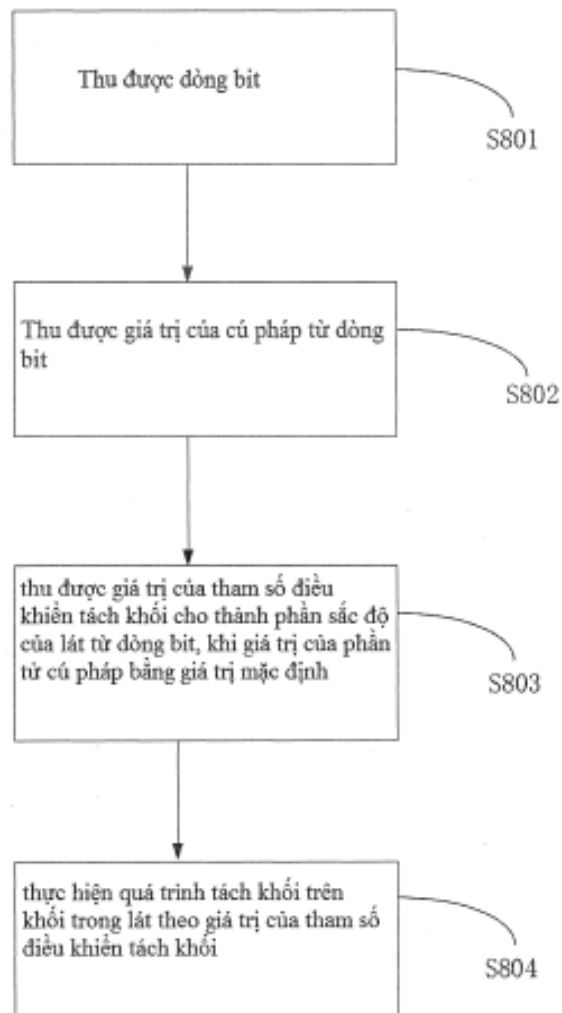


Fig.8

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến lĩnh vực xử lý ảnh và cụ thể hơn đến bảo hiệu cú pháp.

Tình trạng kỹ thuật của sáng chế

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

Lượng dữ liệu video cần để mô tả đoạn video tương đối ngắn có thể là khá lớn, có thể dẫn đến các khó khăn khi dữ liệu được truyền trên mạng hoặc được truyền qua mạng truyền thông có băng thông hữu hạn. Do vậy, dữ liệu video thường được nén trước khi được truyền thông qua mạng viễn thông hiện đại. Kích thước video cũng có thể là vấn đề khi video được lưu trữ trên bộ lưu trữ do tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng tạo nguồn để tạo mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó giảm số lượng dữ liệu cần để biểu diễn các ảnh video số. Sau đó, dữ liệu được nén được nhận ở đích bằng thiết bị giải nén video giải mã dữ liệu video. Với các tài nguyên mạng hữu hạn và nhu cầu ngày càng tăng đối với chất lượng video cao hơn, mong muốn tạo ra các kỹ thuật nén và giải mã được cải tiến mà cải tiến tỷ lệ nén không làm ảnh hưởng đến chất lượng ảnh.

Bản chất kỹ thuật của sáng chế

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

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

Các phương án thực hiện cụ thể được nêu trong các điểm yêu cầu bảo hộ độc lập, với các phương án thực hiện khác trong các điểm yêu cầu bảo hộ phụ thuộc.

Khía cạnh thứ nhất của sáng chế đề cập đến phương pháp tạo mã được triển khai bởi thiết bị giải mã, phương pháp bao gồm các bước: thu được giá trị của phần tử cú pháp từ dòng bit, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã; phân tách giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát từ dòng bit, khi giá trị của phần tử cú pháp bằng giá trị định trước, giá trị định trước là giá trị số nguyên. Trong ví dụ, giá trị định trước không bằng 0. Trong ví dụ, giá trị định trước bằng 1.

Theo một phương án thực hiện, phương pháp còn bao gồm bước: thực hiện quá trình tách khối trên khối trong lát theo giá trị của tham số điều khiển tách khối.

Các phương án thực hiện sáng chế đề cập đến phương tiện báo hiệu của tham số điều khiển tách khối, tham số điều khiển tách khối đối với thành phần sắc độ được báo hiệu có điều kiện. Khi định dạng sắc độ là (4:0:0 hoặc (4:4:4 và chế độ tạo mã mặt phẳng màu riêng rẽ đang được sử dụng)), tham số điều khiển tách khối đối với thành phần sắc độ không được báo hiệu trong dòng bit. Do vậy, hiệu suất sử dụng và giải mã dòng bit đã được cải thiện.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp thu được

từ tập hợp tham số ảnh (picture parameter set, PPS).

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ tiêu đề ảnh (picture header, PH).

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ tiêu đề lát (slice header, SH).

Theo một phương án thực hiện, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp bằng 0.

Theo một phương án thực hiện, trong đó tham số điều khiển tách khối được báo hiệu (chỉ) khi chuỗi video có thành phần màu.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối đối với thành phần độ sáng của lát giống như tham số điều khiển tách khối đối với thành phần sắc độ của lát.

Theo một phương án thực hiện, trong đó phương pháp còn bao gồm bước:

gán giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát bằng giá trị của tham số điều khiển tách khối đối với thành phần độ sáng của lát, khi giá trị của phần tử cú pháp không bằng giá trị định trước.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối là độ lệch tham số tách khối định trước được áp dụng cho thành phần Cb-Cr chung của lát.

Khía cạnh thứ hai của sáng chế đề cập đến thiết bị giải mã video, thiết bị bao gồm: môđun phân tách, được tạo cấu hình để thu được giá trị của phần tử cú pháp từ dòng bit, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã; môđun phân tách được tạo cấu hình để thu được giá trị của tham số điều khiển tách khối

cho thành phần sắc độ của lát từ dòng bit, khi giá trị của phần tử cú pháp bằng giá trị định trước, giá trị định trước là giá trị số nguyên.

Theo một phương án thực hiện, thiết bị giải mã còn bao gồm môđun nhận, được tạo cấu hình để thu được dòng bit.

Theo một phương án thực hiện, thiết bị giải mã còn bao gồm môđun tách khối, được tạo cấu hình để thực hiện quá trình tách khối trên khối trong lát theo giá trị của tham số điều khiển tách khối.

Các phương án thực hiện sáng chế đề cập đến phương tiện báo hiệu của tham số điều khiển tách khối, tham số điều khiển tách khối đối với thành phần sắc độ được báo hiệu có điều kiện. Khi định dạng sắc độ là (4:0:0 hoặc (4:4:4 và chế độ tạo mã mặt phẳng màu riêng rẽ đang được sử dụng)), tham số điều khiển tách khối đối với thành phần sắc độ không được báo hiệu trong dòng bit. Do vậy, hiệu suất sử dụng và giải mã dòng bit đã được cải thiện.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp thu được từ PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ PH.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối thu được từ SH.

Theo một phương án thực hiện, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp bằng 0.

Theo một phương án thực hiện, trong đó tham số điều khiển tách khối được báo hiệu (chỉ) khi chuỗi video có thành phần màu.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối đối với thành phần độ sáng của lát giống như tham số điều khiển tách khối đối với thành phần sắc độ của lát.

Theo một phương án thực hiện, trong đó mô đun phân tách còn được tạo cấu hình để:

gán giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát bằng giá trị của tham số điều khiển tách khối đối với thành phần độ sáng của lát, khi giá trị của phần tử cú pháp không bằng giá trị định trước.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối là độ lệch tham số tách khối định trước được áp dụng cho thành phần Cb-Cr chung của lát.

Khía cạnh thứ ba của sáng chế đề cập đến phương pháp tạo mã được triển khai bởi thiết bị mã hoá, phương pháp bao gồm các bước:

xác định giá trị của phần tử cú pháp cho lát của ảnh được tạo mã, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát; mã hoá giá trị của tham số điều khiển tách khối cho thành phần sắc độ của khối mã thành dòng bit, khi xác định rằng giá trị của phần tử cú pháp bằng giá trị định trước, giá trị định trước là giá trị số nguyên. Trong ví dụ, giá trị định trước không bằng 0. Trong ví dụ, giá trị định trước bằng 1.

Các phương án thực hiện sáng chế đề cập đến phương tiện báo hiệu của tham số điều khiển tách khối, tham số điều khiển tách khối đối với thành phần sắc độ được báo hiệu có điều kiện. Khi định dạng sắc độ là (4:0:0 hoặc (4:4:4 và chế độ tạo mã mặt phẳng màu riêng rẽ đang được sử dụng)), tham số điều khiển tách khối đối với thành phần sắc độ không được báo hiệu trong dòng bit. Do vậy, hiệu suất sử dụng và giải mã dòng bit đã được cải thiện.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được

báo hiệu trong PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong PH.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong SH.

Theo một phương án thực hiện, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp được xác định bằng 0.

Theo một phương án thực hiện, trong đó tham số điều khiển tách khối được báo hiệu (chỉ) khi chuỗi video có thành phần màu.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối đối với thành phần độ sáng của lát giống như tham số điều khiển tách khối đối với thành phần sắc độ của lát.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối là độ lệch tham số tách khối định trước được áp dụng cho thành phần Cb-Cr chung của lát.

Khía cạnh thứ tư của sáng chế đề cập đến thiết bị mã hoá video, thiết bị mã hoá video bao gồm: môđun xác định, được tạo cấu hình để xác định giá trị của phần tử cú pháp cho lát của ảnh được tạo mã, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát; môđun xử lý, được tạo cấu hình để mã hoá giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thành dòng bit, khi xác định rằng giá trị của phần tử cú pháp bằng giá trị định trước, giá trị định trước là giá trị số nguyên. Trong ví dụ, giá trị định trước không bằng 0. Trong ví dụ, giá trị định trước bằng 1.

Các phương án thực hiện sáng chế đề cập đến phương tiện báo hiệu của tham số điều khiển tách khối, tham số điều khiển tách khối đối với thành phần sắc độ được báo hiệu có điều kiện. Khi định dạng sắc độ là (4:0:0 hoặc (4:4:4 và chế độ tạo mã mặt phẳng màu riêng rẽ đang được sử dụng)), tham số điều khiển tách khối đối với thành phần sắc độ không được báo hiệu trong dòng bit. Do vậy, hiệu suất sử dụng và giải mã dòng bit đã được cải thiện.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được báo hiệu trong PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong PPS.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong PH.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối được báo hiệu trong SH.

Theo một phương án thực hiện, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp được xác định bằng 0.

Theo một phương án thực hiện, trong đó tham số điều khiển tách khối được báo hiệu (chỉ) khi chuỗi video có thành phần màu.

Theo một phương án thực hiện, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối đối với thành phần độ sáng của lát giống như tham số điều khiển tách khối đối với thành phần sắc độ của lát.

Theo một phương án thực hiện, trong đó giá trị của tham số điều khiển tách khối là độ lệch tham số tách khối định trước được áp dụng cho thành phần Cb-Cr chung của lát.

Khía cạnh thứ năm của sáng chế đề cập đến bộ giải mã bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ nhất và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ nhất.

Khía cạnh thứ sáu của sáng chế đề cập đến sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh thứ nhất, khía cạnh thứ ba và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ ba khi được thực thi trên máy tính hoặc bộ xử lý.

Khía cạnh thứ bảy của sáng chế đề cập đến bộ giải mã, bao gồm: một hoặc nhiều bộ xử lý; và vật lưu trữ máy tính đọc được bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bằng các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo trường hợp bất kỳ của khía cạnh thứ nhất, khía cạnh thứ ba và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ ba.

Khía cạnh thứ tám của sáng chế đề cập đến vật ghi máy tính đọc được bất biến mang mã chương trình mà, khi được thực thi bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo trường hợp bất kỳ của khía cạnh thứ nhất, khía cạnh thứ ba và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ ba.

Khía cạnh thứ chín của sáng chế đề cập đến bộ mã hoá bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ ba và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ ba.

Khía cạnh thứ mười của sáng chế đề cập đến bộ mã hoá, bao gồm: một hoặc nhiều bộ xử lý; và vật lưu trữ máy tính đọc được bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bằng các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện

phương pháp theo trường hợp bất kỳ của khía cạnh thứ ba và trường hợp bất kỳ của cách thực hiện của khía cạnh thứ ba.

Khía cạnh thứ mười một của sáng chế đề cập đến vật lưu trữ bất biến bao gồm dòng bit được mã hoá/được giải mã bởi phương pháp theo một trong các phương án thực hiện nêu trên.

Khía cạnh thứ mười hai của sáng chế đề cập đến dòng bit được mã hoá đối với tín hiệu video bằng cách bao gồm các phần tử cú pháp, trong đó các phần tử cú pháp bao gồm tham số điều khiển tách khối đối với thành phần sắc độ, được báo hiệu có điều kiện ít nhất dựa trên giá trị của phần tử cú pháp, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã.

Khía cạnh thứ mười ba của sáng chế đề cập đến vật lưu trữ bất biến bao gồm dòng bit được mã hoá được giải mã bởi thiết bị giải mã ảnh, dòng bit được tạo bằng cách phân chia khung của tín hiệu video hoặc tín hiệu ảnh thành các khối, và bao gồm các phần tử cú pháp, trong đó các phần tử cú pháp bao gồm tham số điều khiển tách khối đối với thành phần sắc độ, được báo hiệu có điều kiện ít nhất dựa trên giá trị của phần tử cú pháp, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã.

Phương pháp theo khía cạnh thứ nhất của sáng chế có thể được thực hiện bằng thiết bị theo khía cạnh thứ hai của sáng chế. Các dấu hiệu và các dạng triển khai khác của phương pháp theo khía cạnh thứ nhất của sáng chế tương ứng với các dấu hiệu và các dạng triển khai của thiết bị theo khía cạnh thứ hai của sáng chế.

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

Mô tả vắn tắt các hình vẽ

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

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

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

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

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

Fig.4 là sơ đồ khối minh hoạ ví dụ của thiết bị mã hoá hoặc thiết bị giải mã;

Fig.5 là sơ đồ khối minh hoạ ví dụ khác của thiết bị mã hoá hoặc thiết bị giải mã;

Fig.6 là sơ đồ khối thể hiện cấu trúc ví dụ của hệ thống cung cấp nội dung 3100 thực hiện dịch vụ truyền nội dung;

Fig.7 là sơ đồ khối thể hiện ví dụ cấu trúc của thiết bị đầu cuối;

Fig.8 là lưu đồ thể hiện phương pháp theo phương án thực hiện sáng chế; và

Fig.9 là sơ đồ khối thể hiện thiết bị theo phương án thực hiện sáng chế.

Trong phần sau, các dấu chỉ dẫn giống hệt đề cập đến các dấu hiệu giống hệt hoặc ít nhất tương đương về chức năng nếu không được nêu tường minh theo cách ngược lại.

Mô tả chi tiết sáng chế

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

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

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

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

Vài chuẩn tạo mã video thuộc nhóm “codec video lai có tổn hao” (tức là, kết hợp dự báo không gian và thời gian trong miền mẫu và tạo mã biến đổi 2D để áp dụng lượng tử hóa trong miền biến đổi). Mỗi ảnh của chuỗi video được phân vùng thông thường thành tập hợp các khối không chồng lấn và tạo mã thường được thực hiện ở mức khối. Nói cách khác, ở bộ mã hóa, video thường được xử lý, tức là, được mã hóa, ở mức khối (khối video), chẳng hạn, nhờ sử dụng dự

báo không gian (trong ảnh) và/hoặc dự báo thời gian (ngoài ảnh) để tạo khối dự báo, lấy khối hiện tại (khối đang được xử lý/cần được xử lý) trừ khối dự báo để thu được khối dư, biến đổi khối dư và lượng tử hóa khối dư trong miền biến đổi để giảm lượng dữ liệu cần được truyền (nén), trong đó ở bộ giải mã, xử lý đảo so với bộ mã hóa được áp dụng cho khối được mã hóa hoặc được nén để tái tạo khối hiện tại để biểu diễn. Ngoài ra, bộ mã hóa sao chép vòng xử lý bộ giải mã sao cho chúng đều tạo các dự báo giống hệt (chẳng hạn, các dự báo trong và ngoài) và/hoặc các tái tạo để xử lý, tức là, tạo mã, các khối tiếp theo.

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

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

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

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

Nguồn ảnh 16 có thể bao gồm hoặc là loại bất kỳ của thiết bị chụp ảnh, chẳng hạn, camera để chụp ảnh thế giới thực, và/hoặc loại bất kỳ của thiết bị tạo ảnh, chẳng hạn bộ xử lý đồ họa máy tính để tạo ảnh động bằng máy tính, hoặc loại

bất kỳ của thiết bị khác để thu được và/hoặc cấp ảnh thể giới thực, ảnh được tạo bằng máy tính (chẳng hạn, nội dung màn hình, ảnh thực tại ảo (virtual reality, VR)) và/hoặc kết hợp bất kỳ của chúng (chẳng hạn, ảnh thực tại tăng cường (augmented reality, AR)). Nguồn ảnh có thể là loại bất kỳ của bộ nhớ hoặc bộ lưu trữ sẽ lưu trữ ảnh bất kỳ trong các ảnh nêu trên.

Bên cạnh bộ tiền xử lý 18 và quá trình được thực hiện bằng khối tiền xử lý 18, ảnh hoặc dữ liệu ảnh 17 cũng có thể được gọi là ảnh thô hoặc dữ liệu ảnh thô 17.

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

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

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

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

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

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

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

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

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

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

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

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

Mặc dù Fig.1A mô tả bộ phận nguồn 12 và bộ phận đích 14 dưới dạng các thiết bị riêng rẽ, theo các phương án thực hiện, các thiết bị cũng có thể bao gồm cả hai chức năng, bộ phận nguồn 12 hoặc chức năng tương ứng và bộ phận đích 14 hoặc chức năng tương ứng. Theo các phương án thực hiện này, bộ phận nguồn 12 hoặc chức năng tương ứng và bộ phận đích 14 hoặc chức năng tương ứng có

thể được triển khai nhờ sử dụng phần cứng và/hoặc phần mềm tương tự hoặc bằng phần cứng và/hoặc phần mềm riêng rẽ hoặc kết hợp bất kỳ của chúng.

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

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

bộ mã hóa/bộ giải mã kết hợp (CODEC) trong một thiết bị, chẳng hạn, như được thể hiện theo Fig.1B.

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

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

Để tiện mô tả, các phương án thực hiện sáng chế được mô tả ở đây, chẳng hạn, nhờ tham chiếu đến tạo mã video hiệu suất cao (High-Efficiency Video Coding, HEVC) hoặc đến phần mềm tham chiếu tạo mã video đa dụng (Versatile Video coding, VVC), chuẩn tạo mã video thế hệ tiếp theo được phát triển bởi

nhóm cộng tác chung về tạo mã video (Joint Collaboration Team on Video Coding, JCT-VC) của nhóm chuyên gia tạo mã video ITU-T (Video Coding Experts Group, VCEG) và nhóm chuyên gia ảnh động ISO/IEC (Motion Picture Experts Group, MPEG). Người có hiểu biết trung bình trong lĩnh vực sẽ hiểu rằng các phương án thực hiện sáng chế không bị giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

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

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

ngược 210, khối xử lý biến đổi ngược 212, khối tái tạo 214, bộ lọc vòng 220, DPB 230, khối dự báo ngoài 244 và khối dự báo trong 254 cũng được gọi là tạo “bộ giải mã cài sẵn” của bộ mã hóa video 20.

Ảnh & Phân vùng ảnh (Ảnh & khối)

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

Ảnh (số) là hoặc có thể được xem là mảng hai chiều hoặc ma trận mẫu có các giá trị cường độ. Mẫu trong mảng có thể also được gọi là pixel (viết tắt của phần tử ảnh) hoặc pel. Số lượng mẫu theo chiều ngang và dọc (hoặc trục) của mảng hoặc ảnh định nghĩa kích thước và/hoặc độ phân giải của ảnh. Để biểu diễn màu, thông thường ba thành phần màu được sử dụng, tức là, ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Ở định dạng RGB hoặc không gian màu, ảnh bao gồm mảng mẫu đỏ, lục và lam tương ứng. Tuy nhiên, trong kỹ thuật tạo mã video, mỗi pixel thường được đại diện ở định dạng độ sáng và sắc độ hoặc không gian màu, chẳng hạn, YCbCr, bao gồm thành phần độ sáng được chỉ báo bởi Y (đôi lúc L cũng được sử dụng thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần độ sáng (hoặc gọi tắt là luma) Y biểu diễn độ sáng hoặc cường độ mức xám (chẳng hạn, tương tự trong ảnh thang xám), trong khi hai thành phần sắc độ (hoặc gọi tắt là chroma) Cb và Cr biểu

diễn sắc sai hoặc thành phần thông tin màu. Theo đó, ảnh ở định dạng YCbCr bao gồm mảng mẫu độ sáng của các giá trị mẫu độ sáng (Y), và hai mảng mẫu sắc độ của các giá trị sắc độ (Cb và Cr). Các ảnh ở định dạng RGB có thể được biến đổi thành định dạng YCbCr và ngược lại, quá trình này cũng được biết là biến đổi màu. Nếu ảnh là đơn sắc, ảnh có thể bao gồm chỉ mảng mẫu độ sáng. Theo đó, ảnh có thể là, chẳng hạn, mảng mẫu độ sáng ở định dạng đơn sắc hoặc mảng mẫu độ sáng và hai mảng tương ứng của các mẫu sắc độ ở định dạng màu 4:2:0, 4:2:2, và 4:4:4.

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

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

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

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

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

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

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

Tính toán phần dư

Khối tính toán phần dư 204 có thể được tạo cấu hình để tính toán khối dư 205 (cũng được gọi là phần dư 205) dựa trên khối ảnh 203 và khối dự báo 265

(các chi tiết khác về khối dự báo 265 được nêu dưới đây), chẳng hạn, bằng cách lấy các giá trị mẫu của khối ảnh 203 trừ đi các giá trị mẫu của khối dự báo 265, theo từng mẫu (theo từng pixel) để thu được khối dư 205 trong miền mẫu.

Biến đổi

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

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

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

Lượng tử hóa

Khối lượng tử 208 có thể được tạo cấu hình để lượng tử hóa các hệ số biến đổi 207 để thu được các hệ số được lượng tử 209, chẳng hạn, bằng cách áp dụng lượng tử vô hướng hoặc lượng tử vector. Các hệ số được lượng tử 209 có thể cũng được gọi là các hệ số biến đổi được lượng tử 209 hoặc các hệ số dư được lượng tử 209.

Quá trình lượng tử có thể giảm độ sâu bit được liên kết với một số hoặc tất cả các hệ số biến đổi 207. Chẳng hạn, hệ số biến đổi n bit có thể được làm tròn xuống hệ số biến đổi m bit trong khi lượng tử, trong đó n lớn hơn m . Mức độ lượng tử có thể được chỉnh sửa bằng cách điều chỉnh tham số lượng tử (quantization parameter, QP). Chẳng hạn để lượng tử vô hướng, phép chia tỷ lệ khác có thể được áp dụng để lượng tử mịn hơn hoặc thô hơn. Các kích thước bậc lượng tử nhỏ hơn tương ứng với lượng tử mịn hơn, trong đó các kích thước bậc lượng tử lớn hơn tương ứng với lượng tử thô hơn. Kích thước bậc lượng tử áp dụng được có thể được chỉ báo bởi tham số lượng tử (quantization parameter, QP). QP có thể chẳng hạn là chỉ số cho tập hợp định trước của các kích thước bậc lượng tử có thể áp dụng. Chẳng hạn, các QP nhỏ có thể tương ứng với lượng tử mịn (các kích thước bậc lượng tử nhỏ) và các QP lớn có thể tương ứng với lượng tử thô (các kích thước bậc lượng tử lớn) hoặc ngược lại. Lượng tử có thể bao gồm phép chia cho kích thước bậc lượng tử và/hoặc lượng tử ngược tương ứng, chẳng hạn, bằng khối lượng tử ngược 210, có thể bao gồm phép nhân với

kích thước bậc lượng tử. Các phương án thực hiện theo một số chuẩn, chẳng hạn, HEVC, có thể được tạo cấu hình để sử dụng QP để xác định kích thước bậc lượng tử. Nói chung, kích thước bậc lượng tử có thể được tính toán dựa trên QP nhờ sử dụng phép xấp xỉ dấu chấm tĩnh của phương trình bao gồm phép chia. Các hệ số chia tỷ lệ bổ sung có thể được đề xuất để lượng tử và khử lượng tử để khôi phục chuẩn của khối dư, có thể được chỉnh sửa do tỷ lệ được sử dụng trong phép xấp xỉ dấu chấm tĩnh của phương trình cho kích thước bậc lượng tử và QP. Theo một cách thực hiện ví dụ, phép tỷ lệ của biến đổi ngược và khử lượng tử có thể được kết hợp. Theo cách khác, các bảng lượng tử được tùy chỉnh có thể được sử dụng và được báo hiệu từ bộ mã hóa đến bộ giải mã, chẳng hạn, trong dòng bit. Phép lượng tử là phép tổn hao, trong đó tổn hao tăng khi tăng các kích thước bậc lượng tử.

Theo các phương án thực hiện, bộ mã hóa video 20 (tương ứng là khối lượng tử 208) có thể được tạo cấu hình để xuất ra các QP, chẳng hạn, trực tiếp hoặc được mã hóa qua khối mã hóa entropy 270, sao cho, chẳng hạn, bộ giải mã video 30 có thể nhận và áp dụng các QP để giải mã.

Lượng tử ngược

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

Biến đổi ngược

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

Tái tạo

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

Lọc

Bộ lọc vòng 220 có thể được tạo cấu hình để lọc khối được tái tạo 215 để thu được khối được lọc 221, hoặc nói chung, để lọc các mẫu được tái tạo để thu được các giá trị mẫu được lọc. Bộ lọc vòng, chẳng hạn, được tạo cấu hình để làm mượt các phép dịch chuyển pixel, hoặc ngược lại cải thiện chất lượng video. Bộ lọc vòng 220 có thể bao gồm một hoặc nhiều bộ lọc vòng, chẳng hạn, DBF, lọc độ lệch mẫu thích ứng (sample-adaptive offset, SAO) hoặc một hoặc nhiều bộ lọc khác, chẳng hạn, ALF, bộ lọc chặn nhiễu (noise suppression filter, NSF), hoặc kết hợp bất kỳ của chúng. Trong ví dụ, bộ lọc vòng 220 có thể bao gồm DBF, bộ lọc SAO và ALF. Thứ tự của quá trình lọc có thể là DBF, SAO và ALF. Trong ví dụ khác, quá trình được gọi là ánh xạ độ sáng có chia tỷ lệ sắc độ (luma mapping with chroma scaling, LMCS) (tức là, bộ tạo dạng lại trong vòng thích

ứng) được bổ sung. Quá trình này được thực hiện trước khi tách khối. Trong ví dụ khác, quá trình lọc tách khối cũng có thể được áp dụng cho các mép khối thành phần bên trong, chẳng hạn, các mép khối thành phần afin, các mép khối thành phần ATMVP, các mép biến đổi khối thành phần (sub-block transform, SBT) và các mép phân vùng thành phần bên trong (intra sub-partition, ISP).

Để loại bỏ hiệu quả các giả tượng tạo khối xuất hiện cho “các khối” lớn, VVC sử dụng DBF bước dài hơn. Ở đây, cụm từ “các khối” được sử dụng chung chung và có thể đề cập đến “khối biến đổi (transform block, TB), khối dự báo (prediction block, PB) hoặc CU”. Bộ lọc bước dài hơn được áp dụng cho cả thành phần độ sáng và sắc độ. Bộ lọc bước dài hơn đối với các thành phần độ sáng chỉnh sửa tối đa 7 mẫu cho mỗi dòng mẫu vuông góc và liền kề với mép và được áp dụng cho các khối có kích thước ≥ 32 mẫu theo hướng tách khối, tức là, đối với các mép dọc, chiều rộng khối phải ≥ 32 mẫu và đối với các mép ngang, chiều cao khối phải ≥ 32 mẫu.

Bộ lọc bước dài hơn sắc độ được áp dụng cho các khối sắc độ khi cả hai khối liền kề với mép cụ thể có ≥ 8 mẫu và chỉnh sửa tối đa ba mẫu trên mỗi phía của mép. Do vậy, đối với các mép dọc, chiều rộng khối của cả các khối liền kề với mép phải ≥ 8 mẫu và đối với các mép ngang, chiều cao khối của cả hai khối liền kề với mép phải ≥ 8 mẫu. Mặc dù bộ lọc vòng 220 được thể hiện trên Fig.2 như là bộ lọc trong vòng, trong các cấu hình khác, bộ lọc vòng 220 có thể được triển khai dưới dạng hậu lọc vòng. Khối được lọc 221 cũng có thể được gọi là khối được tái tạo được lọc 221.

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

có thể nhận và áp dụng các tham số lọc vòng tương tự hoặc các bộ lọc vòng tương ứng để giải mã.

Bộ đệm ảnh được giải mã

DPB 230 có thể là bộ nhớ lưu trữ các ảnh tham chiếu, hoặc nói chung là dữ liệu ảnh tham chiếu, để mã hóa dữ liệu video bằng bộ mã hóa video 20. DPB 230 có thể được tạo bằng thiết bị bất kỳ trong các bộ nhớ, chẳng hạn bộ nhớ truy nhập ngẫu nhiên động (dynamic random access memory, DRAM), bao gồm DRAM đồng bộ (synchronous DRAM, SDRAM), RAM từ trở (magnetoresistive RAM, MRAM), RAM điện trở (resistive RAM, RRAM), hoặc các loại bộ nhớ khác. DPB 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. DPB 230 có thể còn được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, chẳng hạn, các khối được lọc và được tái tạo trước đó 221, của ảnh hiện tại tương tự hoặc của các ảnh khác nhau, chẳng hạn, các ảnh được tái tạo trước đó, và có thể tạo các ảnh được tái tạo hoàn chỉnh trước đó, tức là được giải mã, (và các khối tham chiếu và mẫu tương ứng) và/hoặc ảnh hiện tại được tái tạo một phần (và các khối tham chiếu và mẫu tương ứng), chẳng hạn để dự báo ngoài. DPB 230 cũng có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái tạo không được lọc 215, hoặc nói chung, các mẫu được tái tạo không được lọc, chẳng hạn, nếu khối được tái tạo 215 không được lọc bởi khối lọc vòng 220, hoặc phiên bản được xử lý thêm khác bất kỳ của các khối được tái tạo hoặc các mẫu.

Chọn chế độ (Phân vùng & dự báo)

Khối chọn chế độ 260 bao gồm khối phân vùng 262, khối dự báo ngoài 244 và khối dự báo trong 254, và được tạo cấu hình để nhận hoặc thu được dữ liệu ảnh ban đầu, chẳng hạn, khối ban đầu 203 (khối hiện tại 203 của ảnh hiện tại

17), và dữ liệu ảnh được tái tạo, chẳng hạn, các mẫu hoặc các khối được tái tạo được lọc và/hoặc không được lọc của cùng ảnh (hiện tại) và/hoặc từ một hoặc các ảnh được giải mã trước đó, chẳng hạn, từ DPB 230 hoặc các bộ đệm khác (chẳng hạn, bộ đệm dòng, không được thể hiện trên hình vẽ). Dữ liệu ảnh được tái tạo được sử dụng làm dữ liệu ảnh tham chiếu để dự báo, chẳng hạn, dự báo ngoài hoặc dự báo trong, để thu được khối dự báo 265 hoặc bộ dự báo 265.

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

Theo các phương án thực hiện, khối chọn chế độ 260 có thể được tạo cấu hình để chọn phân vùng và chế độ dự báo (chẳng hạn, từ các mẫu được hỗ trợ bởi hoặc khả dụng cho khối chọn chế độ 260), tạo so khớp tốt nhất hoặc nói cách khác phép dư nhỏ nhất (phép dư nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc lưu trữ), hoặc phụ tải báo hiệu nhỏ nhất (phụ tải báo hiệu nhỏ nhất nghĩa là nén tốt hơn để truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Khối chọn chế độ 260 có thể được tạo cấu hình để xác định chế độ phân vùng và dự báo dựa trên tối ưu hóa méo dạng tốc độ (rate distortion optimization, RDO), tức là, chọn chế độ dự báo cho méo dạng tốc độ nhỏ nhất. Các cụm từ, tương tự, “tốt nhất”, “nhỏ nhất”, “tối ưu” v.v. trong ngữ cảnh này không nhất thiết đề cập đến toàn bộ “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. nhưng cũng có thể là hoàn thành tiêu chí kết thúc hoặc lựa chọn tương tự giá trị lớn hơn hoặc nhỏ hơn ngưỡng hoặc các ràng buộc khác có thể dẫn đến “lựa chọn tối ưu phụ” nhưng giảm thời gian xử lý và độ phức tạp.

Nói cách khác, khối phân vùng 262 có thể được tạo cấu hình để phân vùng ảnh từ chuỗi video thành chuỗi CTU, và CTU 203 có thể còn được phân vùng

thành các phân vùng khối nhỏ hơn hoặc các khối phụ (lại tạo thành các khối), chẳng hạn, sử dụng lặp lại phân vùng cây tứ phân (quad-tree, QT), phân vùng cây nhị phân (binary tree, BT) hoặc phân vùng cây tam phân (triple-tree, TT) hoặc kết hợp bất kỳ của chúng, và để thực hiện, chẳng hạn, dự báo cho mỗi khối trong các phân vùng khối hoặc các khối thành phần, trong đó việc chọn chế độ bao gồm chọn cấu trúc cây của khối được phân vùng 203 và các chế độ dự báo được áp dụng cho mỗi trong các phân vùng khối hoặc các khối thành phần.

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

Phân vùng

Khối phân vùng 262 có thể được tạo cấu hình để phân vùng ảnh từ chuỗi video thành chuỗi CTU, và khối phân vùng 262 có thể phân vùng (hoặc tách) CTU 203 thành các phân vùng nhỏ hơn, chẳng hạn, các khối nhỏ hơn có kích thước hình vuông hoặc hình chữ nhật. Đối với ảnh có ba mảng mẫu, CTU gồm khối $N \times N$ của các mẫu độ sáng cùng với hai khối tương ứng của các mẫu sắc độ. Kích thước lớn nhất được phép của khối độ sáng trong CTU được xác định bằng 128×128 trong chuẩn VVC đang phát triển, nhưng có thể được xác định bằng giá trị khác ngoài 128×128 trong tương lai, chẳng hạn, 256×256 . Các CTU của ảnh có thể được tạo cụm/được nhóm dưới dạng các lát/nhóm ô, ô hoặc khối viên. Ô bao phủ vùng chữ nhật của ảnh, và ô có thể được phân chia thành một hoặc nhiều khối viên. Khối viên bao gồm các hàng CTU trong ô. Ô không được phân vùng thành nhiều khối viên có thể được gọi là khối viên. Tuy nhiên, khối viên là tập con đúng của ô và không được gọi là ô. Có hai chế độ của các nhóm ô được hỗ trợ trong VVC, tức là, chế độ nhóm lát/ô quét mảnh và chế độ lát chữ

nhật. Trong chế độ nhóm ô quét mảnh, nhóm lát/ô chứa chuỗi ô khi quét mảnh ô của ảnh. Trong chế độ lát chữ nhật, lát chứa các khối viên của ảnh tạo chung vùng chữ nhật của ảnh. Các khối viên trong lát chữ nhật theo thứ tự quét mảnh khối viên của lát. Các khối nhỏ hơn này (cũng có thể được gọi là các khối thành phần) có thể còn được phân vùng thành các phân vùng thậm chí nhỏ hơn. Đây cũng được gọi là phân vùng cây hoặc phân vùng cây phân cấp, trong đó khối gốc, chẳng hạn, ở mức gốc cây 0 (mức phân cấp 0, chiều sâu 0), có thể được phân vùng đệ quy, chẳng hạn, được phân vùng thành hai hoặc nhiều khối của mức cấp thấp hơn tiếp theo, chẳng hạn, các nút ở mức cây 1 (mức phân cấp 1, chiều sâu 1), trong đó các khối này có thể lại được phân vùng thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, chẳng hạn, mức cây 2 (mức phân cấp 2, chiều sâu 2), v.v. cho đến khi phân vùng kết thúc, chẳng hạn, do tiêu chí phân vùng được hoàn thành, chẳng hạn, chiều sâu cây lớn nhất hoặc kích thước khối nhỏ nhất đạt đến. Các khối không được phân vùng tiếp cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng phân vùng thành hai phân vùng được gọi là BT, cây sử dụng phân vùng thành ba phân vùng được gọi là TT, và cây sử dụng phân vùng thành bốn phân vùng được gọi là QT.

Chẳng hạn, CTU có thể là hoặc bao gồm CTB của các mẫu độ sáng, hai CTB tương ứng của các mẫu sắc độ của ảnh có ba mảng mẫu, hoặc CTB của các mẫu của ảnh đơn sắc hoặc ảnh được tạo mã nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Một cách tương ứng, CTB có thể là khối $N \times N$ của các mẫu đối với giá trị bất kỳ của N sao cho việc phân chia thành phần thành các CTB là phân vùng. CU có thể là hoặc bao gồm CB của các mẫu độ sáng, hai CB tương ứng của các mẫu sắc độ của ảnh mà có ba mảng mẫu, hoặc CB của các mẫu của ảnh đơn sắc hoặc ảnh được tạo mã nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo

mã các mẫu. Một cách tương ứng, CB có thể là khối $M \times N$ của các mẫu đối với vài giá trị của M và N sao cho phân chia CTB thành các CB là phân vùng.

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

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

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

Ngoài ra, cần lưu ý rằng, khi một phần của khối nút cây lớn hơn đường biên ảnh dưới hoặc bên phải, khối nút cây bị buộc phải tách cho đến khi tất cả các mẫu của mỗi CU được tạo mã được đặt bên trong các đường biên ảnh.

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

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

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

Dự báo trong

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

kết hợp dự báo trong phụ thuộc vị trí (position dependent intra prediction combination, PDPC).

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

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

Dự báo ngoài

Tập hợp chế độ dự báo ngoài (hoặc khả thi) phụ thuộc vào các ảnh tham chiếu khả dụng (tức là, các ảnh được giải mã ít nhất một phần trước đó, chẳng hạn, được lưu trữ trong DBP 230) và các tham số dự báo ngoài khác, chẳng hạn, liệu toàn bộ ảnh tham chiếu hoặc chỉ một phần, chẳng hạn, vùng cửa sổ tìm kiếm quanh vùng của khối hiện tại, của ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu so khớp tốt nhất, và/hoặc chẳng hạn, liệu có áp dụng nội suy pixel, chẳng hạn, nội suy nửa pel, $\frac{1}{4}$ pel và/hoặc $\frac{1}{16}$ pel, hay không.

Bên cạnh các chế độ dự báo nêu trên, chế độ skip, chế độ trực tiếp và/hoặc chế độ dự báo ngoài khác có thể được áp dụng.

Chẳng hạn, dự báo hợp nhất mở rộng, danh sách ứng viên hợp nhất của chế độ này được tạo bằng cách bao gồm năm loại ứng viên sau theo thứ tự: MVP không gian từ các CU lân cận không gian, MVP thời gian từ các CU cùng vị trí, MVP dựa trên lịch sử từ bảng FIFO, MVP trung bình theo cặp và các MV số 0.

Và tinh lọc MV phía bộ giải mã dựa trên so khớp hai bên (decoder side motion vector refinement, DMVR) có thể được áp dụng để tăng độ chính xác của các MV của chế độ hợp nhất. Chế độ hợp nhất có MVD (Merge mode with MVD, MMVD), đến từ chế độ hợp nhất có các hiệu số MV. Cờ MMVD được báo hiệu ngay sau khi gửi cờ skip và cờ merge để xác định liệu chế độ MMVD có được sử dụng cho CU hay không. Và phương tiện độ phân giải MV thích ứng (adaptive motion vector resolution, AMVR) mức CU có thể được áp dụng. AMVR cho phép MVD của CU được tạo mã với độ chính xác khác nhau. Tùy thuộc vào chế độ dự báo cho CU hiện tại, các MVD của CU hiện tại có thể được chọn thích ứng. Khi CU được tạo mã trong chế độ hợp nhất, chế độ dự báo trong/ngoài kết hợp (combined inter/intra prediction, CIIP) có thể được áp dụng cho CU hiện tại. Phép tính trung bình có trọng số của các tín hiệu dự báo trong và ngoài được thực hiện để thu được dự báo CIIP. Dự báo bù chuyển động affin, trường chuyển động affin của khối được mô tả bằng thông tin chuyển động của MV hai điểm điều khiển (4 tham số) hoặc ba điểm điều khiển (6 tham số). Dự báo MV thời gian dựa trên khối thành phần (Subblock-based temporal motion vector prediction, SbTMVP), giống như dự báo MV thời gian (temporal motion vector prediction, TMVP) trong HEVC, nhưng dự báo các MV của các CU thành phần trong CU hiện tại. Luồng quang hai hướng (Bi-directional optical flow, BDOF), trước được gọi là BIO, là phiên bản đơn giản hơn đòi hỏi tính toán ít hơn nhiều, đặc biệt liên quan đến các phép nhân và kích thước của số nhân. Chế độ phân vùng tam giác, trong chế độ này, CU được tách đều thành hai phân vùng tam giác, sử dụng tách đường chéo hoặc tách phi đường chéo. Ngoài ra, chế độ dự báo kép được mở rộng ngoài việc tính trung bình đơn giản để cho phép tính trung bình có trọng số hai tín hiệu dự báo.

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

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

Khối bù chuyển động có thể được tạo cấu hình để thu được, chẳng hạn, nhận, tham số dự báo ngoài và để thực hiện dự báo ngoài dựa trên hoặc sử dụng tham số dự báo ngoài để thu được khối dự báo ngoài 265. Bù chuyển động, được thực hiện bằng khối bù chuyển động, có thể bao gồm việc nạp hoặc tạo khối dự báo dựa trên vectơ chuyển động/khối được xác định bằng ước tính chuyển động, có thể thực hiện các phép nội suy theo độ chính xác pixel thành phần. Phép lọc nội suy có thể tạo các mẫu pixel bổ sung từ các mẫu pixel đã biết, do vậy có thể tăng số lượng khối dự báo ứng viên mà có thể được sử dụng để tạo mã khối ảnh. Khi nhận MV cho PU của khối ảnh hiện tại, khối bù chuyển động có thể định vị khối dự báo mà MV trỏ đến ở một trong các danh sách ảnh tham chiếu.

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

Tạo mã entropy

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

Các biến thể cấu trúc khác của bộ mã hóa video 20 có thể được sử dụng để mã hóa dòng video. Chẳng hạn, bộ mã hóa không dựa trên biến đổi 20 có thể lượng tử hóa tín hiệu dư trực tiếp không qua khối xử lý biến đổi 206 đối với các

khối hoặc khung cụ thể. Theo cách thực hiện khác, bộ mã hóa 20 có thể có khối lượng tử 208 và khối lượng tử ngược 210 được kết hợp thành một đơn vị.

Bộ giải mã và phương pháp giải mã

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

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

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

DPB 230. Do vậy, các giải thích được nêu cho các khối và các chức năng tương ứng của bộ mã hóa video 20 áp dụng tương ứng cho các khối và các chức năng tương ứng của bộ giải mã video 30.

Giải mã Entropy

Khối giải mã entropy 304 được tạo cấu hình để phân tách dòng bit 21 (hoặc nói chung là dữ liệu ảnh được mã hóa 21) và thực hiện, chẳng hạn, giải mã entropy thành dữ liệu ảnh được mã hóa 21 để thu được, chẳng hạn, các hệ số được lượng tử 309 và/hoặc các tham số mã được giải mã (không được thể hiện trên Fig.3), chẳng hạn, bất kỳ hoặc tất cả các tham số dự báo ngoài (chẳng hạn, chỉ số ảnh tham chiếu và MV), tham số dự báo trong (chẳng hạn, chế độ hoặc chỉ số dự báo trong), các tham số biến đổi, các QP, các tham số lọc vòng, và/hoặc các phần tử cú pháp khác. Khối giải mã entropy 304 có thể được tạo cấu hình để áp dụng các thuật toán hoặc các phương tiện giải mã tương ứng với các phương tiện mã hóa như được nêu dựa vào khối mã hóa entropy 270 của bộ mã hóa 20. Khối giải mã entropy 304 có thể còn được tạo cấu hình để tạo các tham số dự báo ngoài, tham số dự báo trong và/hoặc các phần tử cú pháp khác cho khối áp dụng chế độ 360 và các tham số khác cho các khối khác của bộ giải mã 30. Bộ giải mã video 30 có thể nhận các phần tử cú pháp ở mức lát video và/hoặc mức khối video. Ngoài ra hoặc bên cạnh các lát và các phần tử cú pháp khác, các nhóm ô và/hoặc các ô và các phần tử cú pháp tương ứng có thể được nhận và/hoặc được sử dụng.

Lượng tử ngược

Khối lượng tử ngược 310 có thể được tạo cấu hình để nhận các QP (hoặc nói chung thông tin liên quan đến lượng tử ngược) và các hệ số được lượng tử từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng

hạn, bằng khối giải mã entropy 304) và áp dụng dựa trên các QP lượng tử ngược trên các hệ số được lượng tử 309 được giải mã để thu được các hệ số được khử lượng tử 311, cũng có thể được gọi là các hệ số biến đổi 311. Quá trình lượng tử ngược có thể bao gồm sử dụng QP được xác định bằng bộ mã hóa video 20 cho mỗi khối video trong lát video (hoặc ô hoặc nhóm ô) để xác định mức độ lượng tử và, tương tự, mức độ lượng tử ngược cần được áp dụng.

Biến đổi ngược

Khối xử lý biến đổi ngược 312 có thể được tạo cấu hình để nhận các hệ số được khử lượng tử 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng biến đổi cho các hệ số được khử lượng tử 311 để thu được các khối dư được tái tạo 213 trong miền mẫu. Các khối dư được tái tạo 213 cũng có thể được gọi là các khối biến đổi 313. Biến đổi này có thể là biến đổi ngược, chẳng hạn, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc quá trình biến đổi ngược tương tự về mặt khái niệm. Khối xử lý biến đổi ngược 312 có thể còn được tạo cấu hình để nhận các tham số biến đổi hoặc thông tin tương ứng từ dữ liệu ảnh được mã hóa 21 (chẳng hạn, bằng cách phân tách và/hoặc giải mã, chẳng hạn, bằng khối giải mã entropy 304) để xác định phép biến đổi cần được áp dụng cho các hệ số được khử lượng tử 311.

Tái tạo

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

Lọc

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

Bộ đệm ảnh được giải mã

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

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

Dự báo

Khối dự báo ngoài 344 có thể có chức năng giống hệt khối dự báo ngoài 244 (cụ thể là với khối bù chuyển động) và khối dự báo trong 354 có thể có chức

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

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

Khối áp dụng chế độ 360 có thể được tạo cấu hình để xác định thông tin dự báo cho khối video của lát video hiện tại bằng cách phân tách các MV hoặc thông

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

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

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

hạn, CTU) hoặc một hoặc nhiều ô, trong đó mỗi ô, chẳng hạn, có thể có dạng chữ nhật và có thể bao gồm một hoặc nhiều khối (chẳng hạn, CTU), chẳng hạn, các khối hoàn chỉnh hoặc bán phần.

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

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

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

N pixel, chẳng hạn không lớn hơn 1 pixel. Hai phương pháp giới hạn MV được nêu ở đây theo bitDepth.

Fig.4 là sơ đồ của thiết bị tạo mã video 400 theo phương án thực hiện sáng chế. Thiết bị tạo mã video 400 thích hợp để thực hiện các phương án thực hiện được bộc lộ như được nêu ở đây. Theo phương án thực hiện, thiết bị tạo mã video 400 có thể là bộ giải mã, chẳng hạn, bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa, chẳng hạn, bộ mã hóa video 20 trên Fig.1A.

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

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

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

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

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

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

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

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

Đối với quá trình lọc tách khối được bộc lộ trong HEVC, hai tham số điều khiển mức cao `beta_offset_div2` và `tc_offset_div2` đã được nêu để điều khiển cường độ tách khối. Các tham số này có thể được truyền ở mức PPS hoặc có thể được ghi đè ở mức tiêu đề lát.

Trong ví dụ, các tham số điều khiển lọc tách khối riêng rẽ cho các thành phần Cb và Cr được đề xuất để tạo mức độ linh hoạt tách khối tốt hơn. Các tham số điều khiển tách khối có thể được báo hiệu trong PPS hoặc PH hoặc SH.

Cú pháp cho các tham số điều khiển tách khối như sau:

Cú pháp PPS:

.....	
pps_chroma_tool_offsets_present_flag	u(1)
if(pps_chroma_tool_offsets_present_flag) {	
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset_present_flag	u(1)
if(pps_joint_cbr_qp_offset_present_flag)	
pps_joint_cbr_qp_offset_value	se(v)

pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
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)
if(pps_joint_cbr_qp_offset_present_flag)	
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)
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
.....	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
.....	

Cú pháp PH:

.....	
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)

if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	
}	
.....	

Cú pháp SH:

....	
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_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)
slice_cb_beta_offset_div2	se(v)
slice_cb_tc_offset_div2	se(v)
slice_cr_beta_offset_div2	se(v)
slice_cr_tc_offset_div2	se(v)
}	
}	
.....	

Ngữ nghĩa của các phần tử cú pháp điều khiển tách khối như sau:

pps_chroma_tool_offsets_present_flag bằng 1 xác định rằng các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ xuất hiện trong PPS RBSP syntax

structure. `pps_chroma_tool_offsets_present_flag` bằng 0 xác định rằng các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ không xuất hiện trong cấu trúc cú pháp PPS RBSP. Khi `ChromaArrayType` bằng 0, giá trị của `pps_chroma_tool_offsets_present_flag` shall be bằng 0.

`pps_beta_offset_div2` và `pps_tc_offset_div2` xác định các độ lệch tham số tách khối mặc định cho β và tC (chia cho 2) được áp dụng cho thành phần độ sáng cho các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khối mặc định được ghi đè bởi các độ lệch tham số tách khối xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát viện dẫn đến PPS. Các giá trị của `pps_beta_offset_div2` và `pps_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_beta_offset_div2` và `pps_tc_offset_div2` đều được suy ra bằng 0.

`pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` xác định các độ lệch tham số tách khối mặc định đối với β và tC (chia cho 2) được áp dụng cho thành phần Cb đối với các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khối mặc định được ghi đè bởi các độ lệch tham số tách khối xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát viện dẫn đến PPS. Các giá trị của `pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` đều được suy ra bằng 0.

`pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` xác định các độ lệch tham số tách khối mặc định đối với β và tC (chia cho 2) được áp dụng cho thành phần Cr cho các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khối mặc định được ghi đè bởi các độ lệch tham số tách khối xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát viện dẫn đến PPS. Các giá trị của `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` phải đều nằm trong khoảng

từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` đều được suy ra bằng 0.

`ph_beta_offset_div2` và `ph_tc_offset_div2` xác định các độ lệch tham số tách khối đối với β và tC (chia cho 2) được áp dụng cho thành phần độ sáng cho các lát được liên kết với PH. Các giá trị của `ph_beta_offset_div2` và `ph_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `ph_beta_offset_div2` và `ph_tc_offset_div2` lần lượt được suy ra bằng `pps_beta_offset_div2` và `pps_tc_offset_div2`.

`ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` xác định các độ lệch tham số tách khối đối với β và tC (chia cho 2) được áp dụng cho thành phần C_b cho các lát được liên kết với PH. Các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` lần lượt được suy ra bằng `pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2`.

`ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` xác định các độ lệch tham số tách khối đối với β và tC (chia cho 2) được áp dụng cho thành phần C_r cho các lát được liên kết với PH. Các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` lần lượt được suy ra bằng `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2`.

`slice_beta_offset_div2` và `slice_tc_offset_div2` xác định các độ lệch tham số tách khối đối với β và tC (chia cho 2) được áp dụng cho thành phần độ sáng cho lát hiện tại. Các giá trị của `slice_beta_offset_div2` và `slice_tc_offset_div2` phải đều

nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `slice_beta_offset_div2` và `slice_tc_offset_div2` được lần lượt suy ra bằng `ph_beta_offset_div2` và `ph_tc_offset_div2`.

`slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` xác định các độ lệch tham số tách khối for β và tC (chia cho 2) được áp dụng cho thành phần Cb cho lát hiện tại. Các giá trị của `slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` được lần lượt suy ra bằng `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2`.

`slice_cb_beta_offset_div2` và `slice_cb_tc_offset_div2` xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần Cr cho lát hiện tại. Các giá trị của `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` được lần lượt suy ra bằng `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2`.

Kỹ thuật tạo mã video đa dụng (Versatile Video Coding, VVC) sử dụng công cụ có tên là tạo mã dư sắc độ chung (Joint Chroma residual coding, JCCR), được báo hiệu trong dòng bit nhờ sử dụng cú pháp **“tu_joint_cbr_residual_flag”**. Công cụ này xác định liệu các mẫu dư cho cả các thành phần sắc độ Cb và Cr được tạo mã dưới dạng một khối biến đổi. Giá trị của **“tu_joint_cbr_residual_flag”** bằng 1 xác định rằng cú pháp đơn vị biến đổi bao gồm các mức hệ số biến đổi cho một khối biến đổi từ đó các mẫu dư cho cả Cb lẫn Cr được dẫn xuất. Công cụ JCCR tận dụng thực tế rằng cả phần dư Cb lẫn Cr dường như tương quan ngược với nhau.

Tuỳ thuộc vào `tu_joint_cbr_residual_flag`, `tu_cbf_cb`, và `tu_cbf_cr`, biến `TuCResMode` được dẫn xuất như sau, trong đó, `tu_cbf_cb` xác định cờ khối được tạo mã của thành phần Cb, `tu_cbf_cr` là cờ khối được tạo mã của thành phần Cr. Và `TuCResMode` chỉ báo chế độ JCCR.

- Nếu `tu_joint_cbr_residual_flag` bằng 0, biến `TuCResMode` được gán bằng 0;
- Ngược lại, nếu `tu_cbf_cb` bằng 1 và `tu_cbf_cr` bằng 0, biến `TuCResMode` được gán bằng 1;
- Ngược lại, nếu `tu_cbf_cb` bằng 1, biến `TuCResMode` được gán bằng 2;
- Ngược lại, biến `TuCResMode` được gán bằng 3.

Mối quan hệ giữa “việc tái tạo các phần dư Cb và Cr” dựa trên các biến `tu_cbf_cb`, và `tu_cbf_cr`, biến `TuCResMode` như được thể hiện trong bảng dưới đây.

<code>tu_cbf_cb</code>	<code>tu_cbf_cr</code>	<code>TuCResMode</code>	Tái tạo các phần dư Cb và Cr
1	0	1	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$
1	1	2	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = \text{CSign} * \text{resJointC}[x][y]$
0	1	3	$\text{resCb}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$ $\text{resCr}[x][y] = \text{resJointC}[x][y]$

Biến `CSgin` là giá trị dấu (+1 hoặc -1) và điều này được báo hiệu trong SH. `resJointC[x][y]` là phần dư thực sự được truyền trong dòng bit.

`resCb[x][y]` chỉ báo các mẫu dư được dẫn xuất cho thành phần sắc độ Cb;

`resCr[x][y]` chỉ báo các mẫu dư được dẫn xuất cho thành phần sắc độ Cr.

Ngoài ra, VVC 6.0 có thể sử dụng các bảng ánh xạ QP sắc độ riêng rẽ cho mỗi thành phần trong các thành phần sắc độ Cb và Cr, và bảng ánh xạ riêng rẽ

cho các phần dư Cb-Cr chung. Khi giá trị của phần tử cú pháp “**same_qp_table_for_chroma**” bằng 1, xác định rằng chỉ một bảng QP sắc độ được báo hiệu và bảng này áp dụng cho Cb, Cr và các phần dư Cb-Cr chung. Khi giá trị của “**same_qp_table_for_chroma**” bằng 0, biểu diễn việc ba bảng ánh xạ QP sắc độ được báo hiệu trong SPS.

Các phần tử cú pháp **num_points_in_qp_table_minus1[i]**, **delta_qp_in_val_minus1[i][j]**, **delta_qp_out_val[i][j]** còn được sử dụng để dẫn xuất các bảng ánh xạ QP sắc độ. Ngữ nghĩa của các phần tử cú pháp này và thủ tục dẫn xuất các bảng ánh xạ QP sắc độ như được thể hiện dưới đây:

num_points_in_qp_table_minus1[i] cộng 1 xác định số lượng điểm được sử dụng để mô tả bảng ánh xạ QP sắc độ thứ i. Giá trị của **num_points_in_qp_table_minus1[i]** phải nằm trong khoảng từ 0 đến $63 + QpBdOffset_c$, bao gồm hai giá trị đầu mút. Khi **num_points_in_qp_table_minus1[0]** không xuất hiện trong dòng bit, giá trị của **num_points_in_qp_table_minus1[0]** được suy ra bằng 0.

delta_qp_in_val_minus1[i][j] xác định giá trị delta được sử dụng để dẫn xuất tọa độ đầu vào của điểm xoay thứ j của bảng ánh xạ QP sắc độ thứ i. Khi **delta_qp_in_val_minus1[0][j]** không xuất hiện trong dòng bit, giá trị của **delta_qp_in_val_minus1[0][j]** được suy ra bằng 0.

delta_qp_out_val[i][j] xác định giá trị delta được sử dụng để dẫn xuất tọa độ đầu ra của điểm xoay thứ j của bảng ánh xạ QP sắc độ thứ i. Khi **delta_qp_out_val[0][j]** không xuất hiện trong dòng bit, giá trị của **delta_qp_out_val[0][j]** được suy ra bằng 0.

Bảng ánh xạ QP sắc độ thứ i **ChromaQpTable[i]** với $i = 0..same_qp_table_for_chroma ? 0 : 2$ được dẫn xuất như sau:

```

qpInVal[i][0] = -QpBdOffsetc + delta_qp_in_val_minus1[i][0]
qpOutVal[i][0] = -QpBdOffsetc + delta_qp_out_val[i][0]
for(j = 1; j <= num_points_in_qp_table_minus1[i]; j++) {
    qpInVal[i][j] = qpInVal[i][j - 1] + delta_qp_in_val_minus1[i][j] + 1
    qpOutVal[i][j] = qpOutVal[i][j - 1] + delta_qp_out_val[i][j]
}
ChromaQpTable[i][qpInVal[i][0]] = qpOutVal[i][0]
for(k = qpInVal[i][0] - 1; k >= -QpBdOffsetc; k--)
    ChromaQpTable[i][k] = Clip3(-QpBdOffsetc, 63, ChromaQpTable[i][k + 1] - 1)    (7-31)
for(j = 0; j < num_points_in_qp_table_minus1[i]; j++) {
    sh = (delta_qp_in_val_minus1[i][j] + 1) >> 1
    for(k = qpInVal[i][j] + 1, m = 1; k <= qpInVal[i][j] + 1; k++, m++)
        ChromaQpTable[i][k] = ChromaQpTable[i][qpInVal[i][j]] +
            (delta_qp_out_val[i][j] + 1) * m + sh) /
            (delta_qp_in_val_minus1[i][j] + 1)
}
for(k = qpInVal[i][num_points_in_qp_table_minus1[i]] + 1; k <= 63; k++)
    ChromaQpTable[i][k] = Clip3(-QpBdOffsetc, 63, ChromaQpTable[i][k - 1] + 1)

```

Khi `same_qp_table_for_chroma` bằng 1, `ChromaQpTable[1][k]` và `ChromaQpTable[2][k]` được gán bằng `ChromaQpTable[0][k]` với $k = -QpBdOffset_c..63$.

Tương hợp dòng bit yêu cầu rằng các giá trị của `qpInVal[i][j]` và `qpOutVal[i][j]` phải nằm trong khoảng từ $-QpBdOffset_c$ đến 63, bao gồm hai giá trị đầu mút với $i = 0..same_qp_table_for_chroma ? 0 : 2$ và $j = 0..num_points_in_qp_table_minus1[i]$.

Lưu ý rằng bảng ánh xạ QP sắc độ cũng có thể được biểu diễn nhờ sử dụng công thức đơn giản lấy giá trị QP độ sáng (QP_i) làm đầu vào và giá trị thành phần màu sắc ($cIdx$) và sau đó xuất ra giá trị QP sắc độ (QP_c) tương ứng. Công thức có thể mô tả mối quan hệ tuyến tính giữa QP độ sáng và QP sắc độ. Chẳng hạn, công thức có thể như sau:

$QP_c = QP_i - x$ trong đó x là hằng số phụ thuộc giá trị thành phần màu sắc ($cIdx$) và x có thể có các giá trị khác nhau cho các chỉ số thành phần màu khác nhau bao gồm thành phần Cb-Cr chung.

8.8.3 Quá trình lọc tách khối

8.8.3.1 Tổng quát

Các đầu vào cho quá trình này là ảnh được tái tạo trước khi tách khối, tức là, mảng $recPicture_L$ và, khi $ChromaArrayType$ không bằng 0, các mảng $recPicture_{Cb}$ và $recPicture_{Cr}$.

Các đầu ra của quá trình này là ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảng $recPicture_L$ và, khi $ChromaArrayType$ không bằng 0, các mảng $recPicture_{Cb}$ và $recPicture_{Cr}$.

Các cạnh dọc trong ảnh được lọc trước. Sau đó, các cạnh ngang trong ảnh được lọc với các mẫu được chỉnh sửa bằng quá trình lọc cạnh dọc làm đầu vào. Các cạnh dọc và ngang trong các CTB của mỗi CTU được xử lý riêng rẽ dựa vào đơn vị mã. Các cạnh dọc của các khối mã trong đơn vị mã được lọc bắt đầu với cạnh ở phía tay trái của các khối mã đi qua các cạnh về phía tay phải của các khối mã theo thứ tự hình học. Các cạnh ngang của các khối mã trong đơn vị mã được lọc bắt đầu với cạnh phía trên của các khối mã đi qua các cạnh phía dưới của các khối mã theo thứ tự hình học.

Lưu ý – Mặc dù quá trình lọc được xác định dựa trên ảnh trong bản mô tả, quá trình lọc có thể được triển khai dựa trên CU có kết quả tương đương, giả sử

bộ giải mã chịu trách nhiệm đúng về thứ tự lệ thuộc xử lý để cho ra các giá trị đầu ra tương tự.

Quá trình lọc tách khối được áp dụng cho tất cả các cạnh khối mã thành phần và các cạnh khối biến đổi của ảnh, ngoại trừ các loại cạnh sau:

- Các cạnh mà nằm ở đường biên của ảnh,
- Các cạnh mà trùng với các đường biên của ảnh thành phần có chỉ số ảnh thành phần `subpicIdx` và `loop_filter_across_subpic_enabled_flag[subpicIdx]` bằng 0,
- Các cạnh mà trùng với các đường biên ảo của ảnh khi `VirtualBoundariesPresentFlag` bằng 1,
- Các cạnh mà trùng với các đường biên ô khi `loop_filter_across_tiles_enabled_flag` bằng 0,
- Các cạnh mà trùng với các đường biên lát khi `loop_filter_across_slices_enabled_flag` bằng 0,
- Các cạnh mà trùng với các đường biên trên hoặc bên trái của các lát có `slice_deblocking_filter_disabled_flag` bằng 1,
- Các cạnh trong các lát có `slice_deblocking_filter_disabled_flag` bằng 1,
- Các cạnh mà không tương ứng với các đường biên lưới mẫu 4×4 của thành phần độ sáng,
- Các cạnh mà không tương ứng với các đường biên lưới mẫu 8×8 của thành phần sắc độ,
- Các cạnh trong thành phần độ sáng mà cả hai phía của cạnh có `intra_bdpcm_luma_flag` bằng 1,
- Các cạnh trong thành phần sắc độ mà cả hai phía của cạnh có `intra_bdpcm_chroma_flag` bằng 1,

- Các cạnh của các khối sắc độ thành phần không phải là các cạnh của đơn vị biến đổi liên kết.

Các loại cạnh, dọc hoặc ngang, được biểu diễn bởi biến `edgeType` như được xác định trong Bảng 42.

Bảng 42 – Tên liên kết với `edgeType`

<code>edgeType</code>	Tên <code>edgeType</code>
0 (vertical edge)	EDGE_VER
1 (horizontal edge)	EDGE_HOR

Khi `slice_deblocking_filter_disabled_flag` của lát hiện tại bằng 0, điều kiện sau áp dụng:

- Biến `treeType` được gán bằng DUAL_TREE_LUMA.
- Các cạnh dọc được lọc bằng cách gọi quá trình lọc tách khối cho một hướng như được xác định trong mệnh đề 8.8.3.2 với biến `treeType`, ảnh được tái tạo trước khi tách khối, tức là, mảng `recPictureL` và biến `edgeType` được gán bằng EDGE_VER làm đầu vào, và ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảng `recPictureL` làm đầu ra.
- Cạnh ngang được lọc bằng cách gọi quá trình lọc tách khối cho một hướng như được xác định trong mệnh đề 8.8.3.2 với biến `treeType`, ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảng `recPictureL` và biến `edgeType` được gán bằng EDGE_HOR làm đầu vào, và ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảng `recPictureL` làm đầu ra.
- Khi `ChromaArrayType` không bằng 0, điều kiện sau áp dụng:
 - Biến `treeType` được gán bằng DUAL_TREE_CHROMA
 - Các cạnh dọc được lọc bằng cách gọi quá trình lọc tách khối cho một hướng như được xác định trong mệnh đề 8.8.3.2 với biến `treeType`, ảnh được tái tạo trước khi tách khối, tức là, mảng `recPictureCb` và `recPictureCr`, và biến

edgeType được gán bằng EDGE_VER làm đầu vào, và ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảngs recPictureCb và recPictureCr làm đầu ra.

– Cạnh ngang được lọc bằng cách gọi quá trình lọc tách khối cho một hướng như được xác định trong clause 8.8.3.2 với biến treeType, ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, các mảng recPictureCb và recPictureCr, và biến edgeType được gán bằng EDGE_HOR làm đầu vào, và ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là, mảngs recPictureCb và recPictureCr làm đầu ra.

8.8.3.2 Quá trình lọc tách khối cho một hướng

Các đầu vào cho quá trình này là:

- biến treeType xác định liệu the luma (DUAL_TREE_LUMA) or chroma components (DUAL_TREE_CHROMA) are currently processed,
- khi treeType bằng DUAL_TREE_LUMA, ảnh được tái tạo trước khi tách khối, tức là, mảng recPictureL,
- khi ChromaArrayType không bằng 0 và treeType bằng DUAL_TREE_CHROMA, các mảng recPictureCb và recPictureCr,
- biến edgeType xác định liệu cạnh dọc (EDGE_VER) hoặc ngang (EDGE_HOR) được lọc.

Các đầu ra của quá trình này là ảnh được tái tạo được chỉnh sửa sau khi tách khối, tức là:

- khi treeType bằng DUAL_TREE_LUMA, mảng recPictureL,
- khi ChromaArrayType không bằng 0 và treeType bằng DUAL_TREE_CHROMA, các mảng recPictureCb và recPictureCr.

Biến firstCompIdx và lastCompIdx được dẫn xuất như sau:

$$\text{firstCompIdx} = (\text{treeType} == \text{DUAL_TREE_CHROMA}) ? 1 : 0 \quad (1244)$$

$\text{lastCompIdx} = (\text{treeType} == \text{DUAL_TREE_LUMA} \mid \mid \text{ChromaArrayType} == 0) ? 0 : 2$ (1245)

Đối với mỗi đơn vị mã và mỗi khối mã trên thành phần màu của đơn vị mã được chỉ báo bởi chỉ số thành phần màu cIdx nằm trong khoảng từ firstCompIdx đến lastCompIdx , bao gồm hai giá trị đầu mút, có chiều rộng khối mã nCbW , chiều cao khối mã nCbH và vị trí của mẫu trên bên trái của khối mã (xCb, yCb) , khi cIdx bằng 0, hoặc khi cIdx không bằng 0 và edgeType bằng EDGE_VER và $\text{xCb} \% 8$ bằng 0, hoặc khi cIdx không bằng 0 và edgeType bằng EDGE_HOR và $\text{yCb} \% 8$ bằng 0, các cạnh được lọc bằng các bước có thứ tự sau:

Biến filterEdgeFlag được dẫn xuất như sau:

- Nếu edgeType bằng EDGE_VER và một hoặc nhiều điều kiện sau là đúng, filterEdgeFlag được gán bằng 0:
 - Đường biên trái của khối mã hiện tại là đường biên trái của ảnh.
 - Đường biên trái của khối mã hiện tại trùng với đường biên trái của ảnh thành phần hiện tại và $\text{loop_filter_across_subpic_enabled_flag}[\text{CurrSubpicIdx}]$ hoặc $\text{loop_filter_across_subpic_enabled_flag}[\text{subpicIdx}]$ bằng 0, trong đó subpicIdx là chỉ số ảnh thành phần của ảnh thành phần mà đường biên trái của khối mã hiện tại trùng với đường biên ảnh thành phần bên phải của ảnh thành phần đó.
 - Đường biên trái của khối mã hiện tại là đường biên trái của ô và $\text{loop_filter_across_tiles_enabled_flag}$ bằng 0.
 - Đường biên trái của khối mã hiện tại là đường biên trái của lát và $\text{loop_filter_across_slices_enabled_flag}$ bằng 0.
 - Đường biên trái của khối mã hiện tại là một trong các đường biên ảo dọc của ảnh và $\text{VirtualBoundariesPresentFlag}$ bằng 1.

- Ngược lại, nếu `edgeType` bằng `EDGE_HOR` và một hoặc nhiều điều kiện sau là đúng, biến `filterEdgeFlag` được gán bằng 0:
 - Đường biên trên của khối mã độ sáng hiện tại là đường biên trên của ảnh.
 - Đường biên trên của khối mã hiện tại trùng với đường biên trên của ảnh thành phần hiện tại và `loop_filter_across_subpic_enabled_flag[CurrSubpicIdx]` hoặc `loop_filter_across_subpic_enabled_flag[subpicIdx]` bằng 0, trong đó `subpicIdx` là chỉ số ảnh thành phần của ảnh thành phần mà đường biên trên của khối mã hiện tại trùng với đường biên ảnh thành phần dưới của ảnh thành phần đó.
 - Đường biên trên của khối mã hiện tại là đường biên trên của ô và `loop_filter_across_tiles_enabled_flag` bằng 0.
 - Đường biên trên của khối mã hiện tại là đường biên trên của lát và `loop_filter_across_slices_enabled_flag` bằng 0.
 - Đường biên trên của khối mã hiện tại là một trong các đường biên ảo ngang của ảnh và `VirtualBoundariesPresentFlag` bằng 1.
 - Ngược lại, `filterEdgeFlag` được gán bằng 1.

Tất cả các thành phần của mảng hai chiều $(nCbW) \times (nCbH)$ `edgeFlags`, `maxFilterLengthQs` và `maxFilterlengthPs` được khởi tạo bằng 0.

Quá trình dẫn xuất của đường biên khối biến đổi được xác định trong mệnh đề 8.8.3.3 được gọi với vị trí (xCb, yCb) , chiều rộng khối mã `nCbW`, chiều cao khối mã `nCbH`, biến `cIdx`, biến `filterEdgeFlag`, mảng `edgeFlags`, các mảng chiều dài lọc lớn nhất `maxFilterLengthPs` và `maxFilterLengthQs`, và biến `edgeType` làm đầu vào, và mảng được chỉnh sửa `edgeFlags`, các mảng chiều dài lọc lớn nhất được chỉnh sửa `maxFilterLengthPs` và `maxFilterLengthQs` làm đầu ra.

Khi $cIdx$ bằng 0, quá trình dẫn xuất của đường biên khối phụ mã thành phần được xác định trong mệnh đề 8.8.3.4 được gọi với vị trí (xCb, yCb) , chiều rộng khối mã $nCbW$, chiều cao khối mã $nCbH$, mảng $edgeFlags$, các mảng chiều dài lọc lớn nhất $maxFilterLengthPs$ và $maxFilterLengthQs$, và biến $edgeType$ làm đầu vào, và mảng được chỉnh sửa $edgeFlags$, các mảng chiều dài lọc lớn nhất được chỉnh sửa $maxFilterLengthPs$ và $maxFilterLengthQs$ làm đầu ra.

Mảng mẫu ảnh $recPicture$ được dẫn xuất như sau:

- Nếu $cIdx$ bằng 0, $recPicture$ được gán bằng mảng mẫu ảnh độ sáng được tái tạo trước khi tách khỏi $recPicture_L$.
- Ngược lại, nếu $cIdx$ bằng 1, $recPicture$ được gán bằng mảng mẫu ảnh sắc độ được tái tạo trước khi tách khỏi $recPicture_{Cb}$.
- Ngược lại ($cIdx$ bằng 2), $recPicture$ được gán bằng mảng mẫu ảnh sắc độ được tái tạo trước khi tách khỏi $recPicture_{Cr}$.

Quá trình dẫn xuất của cường độ lọc đường biên được xác định trong mệnh đề 8.8.3.5 được gọi với mảng mẫu ảnh $recPicture$, vị trí độ sáng (xCb, yCb) , chiều rộng khối mã $nCbW$, chiều cao khối mã $nCbH$, biến $edgeType$, biến $cIdx$, và mảng $edgeFlags$ làm đầu vào, và mảng $(nCbW) \times (nCbH)$ bS làm đầu ra.

Quá trình lọc cạnh cho một hướng được gọi cho khối mã như được xác định trong mệnh đề 8.8.3.6 với biến $edgeType$, biến $cIdx$, ảnh được tái tạo trước khi tách khỏi $recPicture$, vị trí (xCb, yCb) , chiều rộng khối mã $nCbW$, chiều cao khối mã $nCbH$, và các mảng bS , $maxFilterLengthPs$, và $maxFilterLengthQs$, làm đầu vào, và ảnh được tái tạo được chỉnh sửa $recPicture$ làm đầu ra.

8.8.3.3 Quá trình dẫn xuất đường biên khối biến đổi

Các đầu vào cho quá trình này là:

- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,

- biến $nCbW$ xác định chiều rộng của khối mã hiện tại,
- biến $nCbH$ xác định chiều cao của khối mã hiện tại,
- biến $cIdx$ xác định thành phần màu của khối mã hiện tại,
- biến $filterEdgeFlag$,
- mảng hai chiều $(nCbW) \times (nCbH)$ $edgeFlags$,
- các mảng hai chiều $(nCbW) \times (nCbH)$ $maxFilterLengthQs$ và $maxFilterLengthPs$,
- biến $edgeType$ xác định liệu cạnh dọc ($EDGE_VER$) hoặc cạnh ngang ($EDGE_HOR$) được lọc.

Các đầu ra của quá trình này là:

- mảng hai chiều được chỉnh sửa $(nCbW) \times (nCbH)$ $edgeFlags$,
- các mảng hai chiều được chỉnh sửa $(nCbW) \times (nCbH)$ $maxFilterLengthQs$, $maxFilterLengthPs$.

Tùy thuộc vào $edgeType$, các mảng $edgeFlags$, $maxFilterLengthPs$ và $maxFilterLengthQs$ được dẫn xuất như sau:

- Biến $gridSize$ được gán như sau:

$$gridSize = cIdx == 0 ? 4 : 8 \quad (1246)$$

- Nếu $edgeType$ bằng $EDGE_VER$, điều kiện sau áp dụng:
 - Biến $numEdges$ được gán bằng $\text{Max}(1, nCbW / gridSize)$.
- Với $xEdge = 0..numEdges - 1$ và $y = 0..nCbH - 1$, điều kiện sau áp dụng:
- Vị trí ngang x bên trong khối mã hiện tại được gán bằng $xEdge * gridSize$.
- Giá trị của $edgeFlags[x][y]$ được dẫn xuất như sau:
- Nếu $VirtualBoundariesPresentFlag$ bằng 1 và $(xCb + x)$ bằng $VirtualBoundariesPosX[n]$ với mọi $n = 0..NumVerVirtualBoundaries - 1$, $edgeFlags[x][y]$ được gán bằng 0.

- Ngược lại, nếu x bằng 0, $edgeFlags[x][y]$ được gán bằng $filterEdgeFlag$.
- Ngược lại, nếu vị trí $(xCb + x, yCb + y)$ nằm ở cạnh của khối biến đổi, $edgeFlags[x][y]$ được gán bằng 1.
- Khi $edgeFlags[x][y]$ bằng 1, điều kiện sau áp dụng:
- Nếu $cIdx$ bằng 0, điều kiện sau áp dụng:
- Giá trị của $maxFilterLengthQs[x][y]$ được dẫn xuất như sau:
- Nếu chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc nhỏ hơn 4 hoặc chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x - 1, yCb + y)$ bằng hoặc nhỏ hơn 4, $maxFilterLengthQs[x][y]$ được gán bằng 1.
- Ngược lại, nếu chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc lớn hơn 32, $maxFilterLengthQs[x][y]$ được gán bằng 7.
- Ngược lại, $maxFilterLengthQs[x][y]$ được gán bằng 3.
- Giá trị của $maxFilterLengthPs[x][y]$ được dẫn xuất như sau:
- Nếu chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc nhỏ hơn 4 hoặc chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x - 1, yCb + y)$ bằng hoặc nhỏ hơn 4, $maxFilterLengthPs[x][y]$ được gán bằng 1.
- Ngược lại, nếu chiều rộng ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x - 1, yCb + y)$ bằng hoặc lớn hơn 32, $maxFilterLengthPs[x][y]$ được gán bằng 7.
- Ngược lại, $maxFilterLengthPs[x][y]$ được gán bằng 3.
- Ngược lại ($cIdx$ không bằng 0), các giá trị của $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được dẫn xuất như sau:

– Nếu chiều rộng trong các mẫu sắc độ của khối biến đổi ở vị trí sắc độ $(xCb + x, yCb + y)$ và chiều rộng ở vị trí sắc độ $(xCb + x - 1, yCb + y)$ đều bằng hoặc lớn hơn 8, $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được gán bằng 3.

– Ngược lại, $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được gán bằng 1.

– Ngược lại (edgeType bằng EDGE_HOR), điều kiện sau áp dụng:

– Biến numEdges được gán bằng $\text{Max}(1, nCbH / gridSize)$.

– Với $yEdge = 0..numEdges - 1$ và $x = 0..nCbW - 1$, điều kiện sau áp dụng:

– Vị trí dọc y bên trong khối mã hiện tại được gán bằng $yEdge * gridSize$.

– Giá trị của $edgeFlags[x][y]$ được dẫn xuất như sau:

– Nếu VirtualBoundariesPresentFlag bằng 1 và $(yCb + y)$ bằng VirtualBoundariesPosY[n] với mọi $n = 0..NumHorVirtualBoundaries - 1$, $edgeFlags[x][y]$ được gán bằng 0.

– Ngược lại, nếu y bằng 0, $edgeFlags[x][y]$ được gán bằng filterEdgeFlag.

– Ngược lại, nếu vị trí $(xCb + x, yCb + y)$ nằm ở cạnh của khối biến đổi, $edgeFlags[x][y]$ được gán bằng 1.

– Khi $edgeFlags[x][y]$ bằng 1, điều kiện sau áp dụng:

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

– Giá trị của $maxFilterLengthQs[x][y]$ được dẫn xuất như sau:

– Nếu chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc nhỏ hơn 4 or chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y - 1)$ bằng hoặc nhỏ hơn 4, $maxFilterLengthQs[x][y]$ được gán bằng 1.

- Ngược lại, nếu chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc lớn hơn 32, $maxFilterLengthQs[x][y]$ được gán bằng 7.
- Ngược lại, $maxFilterLengthQs[x][y]$ được gán bằng 3.
- Giá trị của $maxFilterLengthPs[x][y]$ được dẫn xuất như sau:
- Nếu chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y)$ bằng hoặc nhỏ hơn 4 hoặc chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y - 1)$ bằng hoặc nhỏ hơn 4, $maxFilterLengthPs[x][y]$ được gán bằng 1.
- Ngược lại, nếu chiều cao ở các mẫu độ sáng của khối biến đổi ở vị trí độ sáng $(xCb + x, yCb + y - 1)$ bằng hoặc lớn hơn 32, $maxFilterLengthPs[x][y]$ được gán bằng 7.
- Ngược lại, $maxFilterLengthPs[x][y]$ được gán bằng 3.
- Ngược lại ($cIdx$ không bằng 0), các giá trị của $maxFilterLengthPs[x][y]$ and $maxFilterLengthQs[x][y]$ được dẫn xuất như sau:
- Nếu chiều cao trong các mẫu sắc độ của khối biến đổi ở vị trí sắc độ $(xCb + x, yCb + y)$ và chiều cao trong các mẫu sắc độ của khối biến đổi ở vị trí sắc độ $(xCb + x, yCb + y - 1)$ đều bằng hoặc lớn hơn 8, điều kiện sau áp dụng:
- Nếu $(yCb + y) \% CtbHeightC$ lớn hơn 0, tức là, cạnh ngang không chồng lấn với đường biên CTB sắc độ trên, cả $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được gán bằng 3
- Ngược lại $((yCb + y) \% CtbHeightC$ bằng 0, tức là, cạnh ngang chồng lấn với đường biên CTB sắc độ trên), $maxFilterLengthPs[x][y]$ được gán bằng 1 và $maxFilterLengthQs[x][y]$ được gán bằng 3.
- Ngược lại, $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được gán bằng 1.

8.8.3.4 Quá trình dẫn xuất của đường biên khối phụ mã thành phần

Các đầu vào cho quá trình này là:

- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,
- biến $nCbW$ xác định chiều rộng của khối mã hiện tại,
- biến $nCbH$ xác định chiều cao của khối mã hiện tại,
- mảng hai chiều $(nCbW) \times (nCbH)$ `edgeFlags`,
- các mảng hai chiều $(nCbW) \times (nCbH)$ `maxFilterLengthQs` và `maxFilterLengthPs`,
- biến `edgeType` xác định liệu cạnh dọc (`EDGE_VER`) hoặc cạnh ngang (`EDGE_HOR`) được lọc.

Các đầu ra của quá trình này là:

- mảng hai chiều $(nCbW) \times (nCbH)$ được chỉnh sửa `edgeFlags`,
- các mảng hai chiều $(nCbW) \times (nCbH)$ được chỉnh sửa `maxFilterLengthQs` và `maxFilterLengthPs`.

Số lượng khối mã thành phần theo chiều ngang `numSbX` và theo chiều dọc `numSbY` được dẫn xuất như sau:

- Nếu `inter_affine_flag[xCb][yCb]` bằng 1 hoặc `merge_subblock_flag[xCb][yCb]` bằng 1, `numSbX` và `numSbY` lần lượt được gán bằng `NumSbX[xCb][yCb]` và `NumSbY[xCb][yCb]`.
- Ngược lại, `numSbX` và `numSbY` đều được gán bằng 1.

Tuỳ thuộc vào giá trị của `edgeType` điều kiện sau áp dụng:

- Nếu `edgeType` bằng `EDGE_VER`, điều kiện sau áp dụng:
- Biến `sbW` được gán bằng `Max(8, nCbW / numSbX)`.
- Mảng `edgeTbFlags` được gán bằng `edgeFlags`.
- Với $xEdge = 0..\min((nCbW / 8) - 1, numSbX - 1)$, $y = 0..nCbH - 1$:

- Vị trí ngang x bên trong khối mã hiện tại được gán bằng $xEdge * sbW$.
- Giá trị của $edgeFlags[x][y]$ được dẫn xuất như sau:
- Nếu $VirtualBoundariesPresentFlag$ bằng 1 và x bằng $VirtualBoundariesPosX[n]$ với mọi $n = 0..NumVerVirtualBoundaries - 1$, điều kiện sau áp dụng:

$$edgeFlags[x][y] = 0 \quad (1247)$$

- Ngược lại, điều kiện sau áp dụng:

$$edgeFlags[x][y] = 2 \quad (1248)$$

- Khi $edgeFlags[x][y]$ bằng 1 hoặc 2, các giá trị của $maxFilterLengthPs[x][y]$ và $maxFilterLengthQs[x][y]$ được chỉnh sửa như sau:

- Nếu x bằng 0, điều kiện sau áp dụng:

- Khi $numSbX$ lớn hơn 1, điều kiện sau áp dụng:

$$maxFilterLengthQs[x][y] = \text{Min}(5, maxFilterLengthQs[x][y]) \quad (1249)$$

- Khi $inter_affine_flag[xCb - 1][yCb + y]$ bằng 1 hoặc $merge_subblock_flag[xCb - 1][yCb + y]$ bằng 1, điều kiện sau áp dụng:

$$maxFilterLengthPs[x][y] = \text{Min}(5, maxFilterLengthPs[x][y]) \quad (1250)$$

- Ngược lại, nếu $edgeTbFlags[x][y]$ bằng 1, điều kiện sau áp dụng:

$$maxFilterLengthPs[x][y] = \text{Min}(5, maxFilterLengthPs[x][y]) \quad (1251)$$

$$maxFilterLengthQs[x][y] = \text{Min}(5, maxFilterLengthQs[x][y]) \quad (1252)$$

- Ngược lại, nếu một hoặc nhiều điều kiện sau là đúng:

- $(x + 4)$ lớn hơn hoặc bằng $nCbW$,

- $edgeTbFlags[x - 4][y]$ bằng 1,

- $edgeTbFlags[x + 4][y]$ bằng 1,

điều kiện sau áp dụng:

$$maxFilterLengthPs[x][y] = 1 \quad (1253)$$

$$maxFilterLengthQs[x][y] = 1 \quad (1254)$$

– Ngược lại, nếu một hoặc nhiều điều kiện sau là đúng:

- xEdge bằng 1,
- xEdge bằng $(nCbW / 8) - 1$,
- edgeTbFlags[x - sbW][y] bằng 1,
- edgeTbFlags[x + sbW][y] bằng 1,

điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = 2 \quad (1255)$$

$$\text{maxFilterLengthQs}[x][y] = 2 \quad (1256)$$

– Ngược lại, điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = 3 \quad (1257)$$

$$\text{maxFilterLengthQs}[x][y] = 3 \quad (1258)$$

– Ngược lại, nếu edgeType bằng EDGE_HOR, điều kiện sau áp dụng:

- Biên sbH được gán bằng $\text{Max}(8, nCbH / \text{numSbY})$.
- Mảng edgeTbFlags được gán bằng edgeFlags.
- Với $yEdge = 0..\text{min}((nCbH / 8) - 1, \text{numSbY} - 1)$, $x = 0..nCbW - 1$:
 - Vị trí dọc y bên trong khối mã hiện tại được gán bằng $yEdge * sbH$.
 - Giá trị của edgeFlags[x][y] được dẫn xuất như sau:
 - Nếu VirtualBoundariesPresentFlag bằng 1 và y bằng VirtualBoundariesPosY[n] với mọi $n = 0..\text{NumHorVirtualBoundaries} - 1$, điều kiện sau áp dụng:

$$\text{edgeFlags}[x][y] = 0 \quad (1259)$$

– Ngược lại, điều kiện sau áp dụng:

$$\text{edgeFlags}[x][y] = 2 \quad (1260)$$

- Khi edgeFlags[x][y] bằng 1 hoặc 2, các giá trị của $\text{maxFilterLengthPs}[x][y]$ và $\text{maxFilterLengthQs}[x][y]$ được chỉnh sửa như sau:
 - Nếu y bằng 0, điều kiện sau áp dụng:

- Khi numSbY lớn hơn 1, điều kiện sau áp dụng:

$$\text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) \quad (1261)$$

- Khi $\text{inter_affine_flag}[x\text{Cb} + x][y\text{Cb} - 1]$ bằng 1 hoặc

$\text{merge_subblock_flag}[x\text{Cb} + x][y\text{Cb} - 1]$ bằng 1, điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) \quad (1262)$$

- Ngược lại, nếu $\text{edgeTbFlags}[x][y]$ bằng 1, điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) \quad (1263)$$

$$\text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) \quad (1264)$$

- Ngược lại, nếu một hoặc nhiều điều kiện sau là đúng:

- $(y + 4)$ lớn hơn hoặc bằng $n\text{CbH}$,

- $\text{edgeTbFlags}[x][y - 4]$ bằng 1,

- $\text{edgeTbFlags}[x][y + 4]$ bằng 1,

điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = 1 \quad (1265)$$

$$\text{maxFilterLengthQs}[x][y] = 1 \quad (1266)$$

- Ngược lại, nếu một hoặc nhiều điều kiện sau là đúng:

- $y\text{Edge}$ bằng 1,

- $y\text{Edge}$ bằng $(n\text{CbH} / 8) - 1$,

- $\text{edgeTbFlags}[x][y - sbH]$ bằng 1,

- $\text{edgeTbFlags}[x][y + sbH]$ bằng 1,

điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = 2 \quad (1267)$$

$$\text{maxFilterLengthQs}[x][y] = 2 \quad (1268)$$

- Ngược lại, điều kiện sau áp dụng:

$$\text{maxFilterLengthPs}[x][y] = 3 \quad (1269)$$

$$\text{maxFilterLengthQs}[x][y] = 3 \quad (1270)$$

8.8.3.5 Quá trình dẫn xuất của cường độ lọc đường biên

Các đầu vào cho quá trình này là:

- mảng mẫu ảnh recPicture,
- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,
- biến nCbW xác định chiều rộng của khối mã hiện tại,
- biến nCbH xác định chiều cao của khối mã hiện tại,
- biến edgeType xác định liệu cạnh dọc (EDGE_VER) hoặc ngang (EDGE_HOR) được lọc,
- biến cIdx xác định thành phần màu của khối mã hiện tại,
- mảng hai chiều (nCbW)x(nCbH) edgeFlags.

Đầu ra của quá trình này là mảng hai chiều (nCbW)x(nCbH) bS xác định cường độ lọc đường biên.

Biến xD_i, yD_j, xN và yN được dẫn xuất như sau:

- Biến gridSize được gán như sau:

$$\text{gridSize} = \text{cIdx} == 0 ? 4 : 8 \quad (1271)$$

- Nếu edgeType bằng EDGE_VER,

$$\text{xDi} = (i * \text{gridSize}) \quad (1272)$$

$$\text{yDj} = \text{cIdx} == 0 ? (j << 2) : (j << 1) \quad (1273)$$

$$\text{xN} \text{ được gán bằng } \text{Max}(0, (\text{nCbW} / \text{gridSize}) - 1) \quad (1274)$$

$$\text{yN} = \text{cIdx} == 0 ? (\text{nCbH} / 4) - 1 : (\text{nCbH} / 2) - 1 \quad (1275)$$

- Ngược lại (edgeType bằng EDGE_HOR),

$$\text{xDi} = \text{cIdx} == 0 ? (i << 2) : (i << 1) \quad (1276)$$

$$\text{yDj} = j * \text{gridSize} \quad (1277)$$

$$\text{xN} = \text{cIdx} == 0 ? (\text{nCbW} / 4) - 1 : (\text{nCbW} / 2) - 1 \quad (1278)$$

$$\text{yN} = \text{Max}(0, (\text{nCbH} / \text{gridSize}) - 1) \quad (1279)$$

Với x_{D_i} có $i = 0..xN$ và y_{D_j} với $j = 0..yN$, điều kiện sau áp dụng:

- Nếu $\text{edgeFlags}[x_{D_i}][y_{D_j}]$ bằng 0, biến $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 0.
- Ngược lại, điều kiện sau áp dụng:
 - Các giá trị mẫu p_0 và q_0 được dẫn xuất như sau:
 - Nếu edgeType bằng EDGE_VER , p_0 được gán bằng $\text{recPicture}[x_{Cb} + x_{D_i} - 1][y_{Cb} + y_{D_j}]$ và q_0 được gán bằng $\text{recPicture}[x_{Cb} + x_{D_i}][y_{Cb} + y_{D_j}]$.
 - Ngược lại (edgeType bằng EDGE_HOR), p_0 được gán bằng $\text{recPicture}[x_{Cb} + x_{D_i}][y_{Cb} + y_{D_j} - 1]$ và q_0 được gán bằng $\text{recPicture}[x_{Cb} + x_{D_i}][y_{Cb} + y_{D_j}]$.
 - Biến $\text{bS}[x_{D_i}][y_{D_j}]$ được dẫn xuất như sau:
 - Nếu cIdx bằng 0 và cả hai mẫu p_0 và q_0 nằm trong khối mã có $\text{intra_bdpcm_luma_flag}$ bằng 1, $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 0.
 - Ngược lại, nếu cIdx lớn hơn 0 và cả hai mẫu p_0 và q_0 nằm trong khối mã có $\text{intra_bdpcm_chroma_flag}$ bằng 1, $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 0.
 - Ngược lại, nếu mẫu p_0 hoặc q_0 nằm trong khối mã của đơn vị mã được tạo mã có chế độ dự báo trong, $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 2.
 - Ngược lại, nếu cạnh khối cũng là cạnh khối mã và mẫu p_0 hoặc q_0 nằm trong khối mã có ciip_flag bằng 1, $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 2.
 - Ngược lại, nếu cạnh khối cũng là cạnh khối biến đổi và mẫu p_0 hoặc q_0 nằm trong khối biến chưa một hoặc nhiều mức hệ số biến đổi khác 0, $\text{bS}[x_{D_i}][y_{D_j}]$ được gán bằng 1.
 - Ngược lại, nếu chế độ dự báo của khối mã thành phần chứa mẫu p_0 khác với chế độ dự báo của khối mã thành phần chứa mẫu q_0 (tức là, một trong khối

mã thành phần được tạo mã trong chế độ dự báo IBC và khối còn lại được tạo mã trong chế độ dự báo ngoài), $bS[xD_i][yD_j]$ được gán bằng 1.

– Ngược lại, nếu $cIdx$ bằng 0, $edgeFlags[xD_i][yD_j]$ bằng 2, và một hoặc nhiều điều kiện sau là đúng, $bS[xD_i][yD_j]$ được gán bằng 1:

– Khối mã thành phần chứa mẫu p_0 và khối mã thành phần chứa mẫu q_0 đều được tạo mã trong chế độ dự báo IBC, và hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của các vector khối được sử dụng khi dự báo hai khối mã thành phần lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng.

– Để dự báo khối mã thành phần chứa mẫu p_0 ảnh tham chiếu khác nhau hoặc số lượng vector chuyển động khác nhau được sử dụng thay vì để dự báo khối mã thành phần chứa mẫu q_0 .

Lưu ý 1 – Việc xác định liệu các ảnh tham chiếu được sử dụng cho hai khối mã thành phần giống nhau hoặc khác nhau chỉ dựa trên các ảnh nào được tạo tham chiếu, không dựa vào việc liệu có tạo dự báo nhờ sử dụng chỉ số vào danh sách ảnh tham chiếu 0 hoặc chỉ số vào danh sách ảnh tham chiếu 1, và cũng dựa vào việc liệu vị trí chỉ số trong danh sách ảnh tham chiếu là khác nhau.

Lưu ý 2 – Số lượng vector chuyển động được sử dụng để dự báo khối mã thành phần có mẫu trên bên trái bao phủ (xSb, ySb) , bằng $PredFlagL0[xSb][ySb] + PredFlagL1[xSb][ySb]$.

– Một vector chuyển động được sử dụng để dự báo khối mã thành phần chứa mẫu p_0 và một vector chuyển động được sử dụng để dự báo khối mã thành phần chứa mẫu q_0 , và hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của các vector chuyển động được sử dụng lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng.

– Hai vector chuyển động và hai ảnh tham chiếu khác nhau được sử dụng để dự báo khối mã thành phần chứa mẫu p_0 , hai vector chuyển động cho hai ảnh

tham chiếu tương tự được sử dụng để dự báo khối mã thành phần chứa mẫu q_0 và hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của hai vector chuyển động được sử dụng khi dự báo hai khối mã thành phần đối với ảnh tham chiếu tương tự lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng.

- Hai vector chuyển động đối với ảnh tham chiếu tương tự được sử dụng để dự báo khối mã thành phần chứa mẫu p_0 , hai vector chuyển động đối với ảnh tham chiếu tương tự được sử dụng để dự báo khối mã thành phần chứa mẫu q_0 và cả hai điều kiện sau là đúng:

- Hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của các vector chuyển động danh sách 0 được sử dụng khi dự báo hai khối mã thành phần lớn hơn hoặc bằng 8 trong 1/16 mẫu độ sáng, hoặc hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của các vector chuyển động danh sách 1 được sử dụng khi dự báo hai khối mã thành phần lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng.

- Hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của vector chuyển động danh sách 0 được sử dụng trong dự báo khối mã thành phần chứa mẫu p_0 và vector chuyển động danh sách 1 được sử dụng trong dự báo khối mã thành phần chứa mẫu q_0 lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng, hoặc hiệu số tuyệt đối giữa thành phần ngang hoặc dọc của vector chuyển động danh sách 1 được sử dụng trong dự báo khối mã thành phần chứa mẫu p_0 và vector chuyển động danh sách 0 được sử dụng trong dự báo khối mã thành phần chứa mẫu q_0 lớn hơn hoặc bằng 8 theo các đơn vị 1/16 mẫu độ sáng.

- Ngược lại, biến $bS[xD_i][yD_j]$ được gán bằng 0.

8.8.3.6 Quá trình lọc cạnh cho một hướng

Các đầu vào cho quá trình này là:

- biến `edgeType` xác định liệu các cạnh dọc (`EDGE_VER`) hoặc các cạnh ngang (`EDGE_HOR`) đang được xử lý,

- biến $cIdx$ xác định thành phần màu hiện tại,
- ảnh được tái tạo trước khi tách khối $recPicture$,
- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,

- biến $nCbW$ xác định chiều rộng của khối mã hiện tại,
- biến $nCbH$ xác định chiều cao của khối mã hiện tại,
- mảng bS xác định cường độ đường biên,
- các mảng $maxFilterLengthPs$ và $maxFilterLengthQs$.

Đầu ra của quá trình này là ảnh được tái tạo được chỉnh sửa sau khi tách khối $recPicture$.

Đối với quá trình lọc cạnh, điều kiện sau áp dụng:

- Biến $gridSize$ được gán như sau:

$$gridSize = cIdx == 0 ? 4 : 8 \quad (1280)$$

- Biếns $subW$, $subH$, xN , yN được dẫn xuất như sau:

$$subW = cIdx == 0 ? 1 : SubWidthC \quad (1281)$$

$$subH = cIdx == 0 ? 1 : SubHeightC \quad (1282)$$

$$xN = edgeType == EDGE_VER ? \text{Max}(0, (nCbW / gridSize) - 1) : (nCbW / 4 / subW) - 1 \quad (1283)$$

$$yN = edgeType == EDGE_VER ? (nCbH / 4 / subH) - 1 : \text{Max}(0, (nCbH / gridSize) - 1) \quad (1284)$$

- Các biến xD_k với $k = 0..xN$ và yD_m với $m = 0..yN$ được dẫn xuất như sau:

$$xD_k = edgeType == EDGE_VER ? (k * gridSize) : (k \ll (2 / subW)) \quad (1285)$$

$$yD_m = edgeType == EDGE_VER ? (m \ll (2 / subH)) : (m * gridSize) \quad (1286)$$

- Với xD_k có $k = 0..xN$ và yD_m có $m = 0..yN$, điều kiện sau áp dụng:
- Khi $bS[xD_k][yD_m]$ lớn hơn 0, các bước có thứ tự sau áp dụng:

– Nếu $cIdx$ bằng 0, quá trình lọc đối với các cạnh trong khối mã độ sáng của đơn vị mã hiện tại gồm các bước có thứ tự sau:

1. Quá trình quyết định cho các cạnh khối độ sáng như được xác định trong mệnh đề 8.8.3.6.1 được gọi với mảng mẫu ảnh độ sáng $recPicture$, vị trí của khối mã độ sáng (xCb, yCb) , vị trí độ sáng của khối (xBl, yBl) được gán bằng (xD_k, yD_m) , hướng cạnh $edgeType$, cường độ lọc đường biên $bS[xD_k][yD_m]$, các chiều dài lọc lớn nhất $maxFilterLengthP$ được gán bằng $maxFilterLengthPs[xD_k][yD_m]$ và $maxFilterLengthQ$ được gán bằng $maxFilterLengthQs[xD_k][yD_m]$ làm đầu vào, và các quyết định dE , dEp và dEq , các chiều dài lọc lớn nhất được chỉnh sửa $maxFilterLengthP$ và $maxFilterLengthQ$, và biến tc làm đầu ra.

2. Quá trình lọc for block edges như được xác định trong mệnh đề 8.8.3.6.2 được gọi với mảng mẫu ảnh độ sáng $recPicture$, vị trí của khối mã độ sáng (xCb, yCb) , vị trí độ sáng của khối (xBl, yBl) được gán bằng (xD_k, yD_m) , hướng cạnh $edgeType$, các quyết định dE , dEp và dEq , các chiều dài lọc lớn nhất $maxFilterLengthP$ và $maxFilterLengthQ$, và biến tc làm đầu vào, và mảng mẫu ảnh độ sáng được chỉnh sửa $recPicture$ làm đầu ra.

– Ngược lại ($cIdx$ không bằng 0), quá trình lọc đối với các cạnh trong khối mã sắc độ của đơn vị mã hiện tại được xác định bởi $cIdx$ gồm các bước có thứ tự sau:

1. Quá trình quyết định đối với các cạnh khối sắc độ như được xác định trong mệnh đề 8.8.3.6.3 được gọi với mảng mẫu ảnh sắc độ $recPicture$, vị trí của khối mã sắc độ (xCb, yCb) , vị trí của khối sắc độ (xBl, yBl) được gán bằng (xD_k, yD_m) , hướng cạnh $edgeType$, biến $cIdx$, cường độ lọc đường biên $bS[xD_k][yD_m]$, các chiều dài lọc lớn nhất $maxFilterLengthP$ được gán bằng $maxFilterLengthPs[xD_k][yD_m]$ và các chiều dài lọc lớn nhất $maxFilterLengthQ$

được gán bằng $\text{maxFilterLengthQs}[\text{xD}_k][\text{yD}_m]$ làm đầu vào, và các chiều dài lọc lớn nhất được chỉnh sửa maxFilterLengthP và maxFilterLengthQ , và biến t_c làm đầu ra.

2. Khi maxFilterLengthQ lớn hơn 0, quá trình lọc đối với các cạnh khối sắc độ như được xác định trong mệnh đề 8.8.3.6.4 được gọi với mảng mẫu ảnh sắc độ recPicture , vị trí của khối mã sắc độ (xCb, yCb) , vị trí sắc độ của khối (xBl, yBl) được gán bằng $(\text{xD}_k, \text{yD}_m)$, hướng cạnh edgeType , biến t_c , các chiều dài lọc lớn nhất maxFilterLengthP và maxFilterLengthQ làm đầu vào, và mảng mẫu ảnh sắc độ được chỉnh sửa recPicture làm đầu ra.

8.8.3.6.1 Quá trình quyết định cho các cạnh khối độ sáng

Các đầu vào cho quá trình này là:

- mảng mẫu ảnh recPicture ,
- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,
- vị trí (xBl, yBl) xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của khối mã hiện tại,
- biến edgeType xác định liệu cạnh dọc (EDGE_VER) hoặc cạnh ngang (EDGE_HOR) được lọc,
- biến bS xác định cường độ lọc đường biên,
- biến maxFilterLengthP xác định chiều dài lọc lớn nhất,
- biến maxFilterLengthQ xác định chiều dài lọc lớn nhất.

Các đầu ra của quá trình này là:

- các biến dE , dEp và dEq chứa các quyết định,
- các biến chiều dài lọc được chỉnh sửa maxFilterLengthP và maxFilterLengthQ ,
- biến t_c .

Các giá trị mẫu $p_{i,k}$ và $q_{j,k}$ với $i = 0..Max(2, maxFilterLengthP)$, $j = 0..Max(2, maxFilterLengthQ)$ và $k = 0$ và 3 được dẫn xuất như sau:

- Nếu edgeType bằng EDGE_VER, điều kiện sau áp dụng:

$$q_{j,k} = recPicture[xCb + xBl + j][yCb + yBl + k] \quad (1287)$$

$$p_{i,k} = recPicture[xCb + xBl - i - 1][yCb + yBl + k] \quad (1288)$$

- Ngược lại (edgeType bằng EDGE_HOR), điều kiện sau áp dụng:

$$q_{j,k} = recPicture[xCb + xBl + k][yCb + yBl + j] \quad (1289)$$

$$p_{i,k} = recPicture[xCb + xBl + k][yCb + yBl - i - 1] \quad (1290)$$

Biến qpOffset được dẫn xuất như sau:

- Nếu sps_ladf_enabled_flag bằng 1, điều kiện sau áp dụng:
- Biến lumaLevel of the reconstructed luma level được dẫn xuất như sau:

$$lumaLevel = ((p_{0,0} + p_{0,3} + q_{0,0} + q_{0,3}) >> 2), \quad (1291)$$

- Biến qpOffset được gán bằng sps_ladf_lowest_interval_qp_offset và được chỉnh sửa như sau:

```
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {
    if(lumaLevel > SpsLadfIntervalLowerBound[i + 1])
        qpOffset = sps_ladf_qp_offset[i]    (1292)
    else
        break
}
```

- Ngược lại, qpOffset được gán bằng 0.

Các biến Q_{pQ} và Q_{pP} được gán bằng các giá trị Q_{pY} của các đơn vị mã lần lượt bao gồm các khối mã chứa mẫu $q_{0,0}$ và $p_{0,0}$.

Biến qP được dẫn xuất như sau:

$$qP = ((Q_{pQ} + Q_{pP} + 1) >> 1) + qpOffset \quad (1293)$$

Giá trị của biến β' được xác định như được xác định trong Bảng 43 dựa trên tham số lượng tử Q được dẫn xuất như sau:

$$Q = \text{Clip3}(0, 63, qP + (\text{slice_beta_offset_div2} \ll 1)) \quad (1294)$$

trong đó $\text{slice_beta_offset_div2}$ là giá trị của phần tử cú pháp $\text{slice_beta_offset_div2}$ đối với lát chứa mẫu $q_{0,0}$.

Biến β được dẫn xuất như sau:

$$\beta = \beta' * (1 \ll (\text{BitDepth} - 8)) \quad (1295)$$

Giá trị của biến t_c' được xác định như trong Bảng 43 dựa trên tham số lượng tử Q được dẫn xuất như sau:

$$Q = \text{Clip3}(0, 65, qP + 2 * (bS - 1) + (\text{slice_tc_offset_div2} \ll 1)) \quad (1296)$$

trong đó $\text{slice_tc_offset_div2}$ là giá trị của phần tử cú pháp $\text{slice_tc_offset_div2}$ đối với lát chứa mẫu $q_{0,0}$.

Biến t_c được dẫn xuất như sau:

$$t_c = \text{BitDepth} < 10 ? (t_c' + 2) \gg (10 - \text{BitDepth}) : t_c' * (1 \ll (\text{BitDepth} - 10)) \quad (1297)$$

Các bước có thứ tự sau áp dụng:

Các biến dp_0 , dp_3 , dq_0 và dq_3 được dẫn xuất như sau:

$$dp_0 = \text{Abs}(p_{2,0} - 2 * p_{1,0} + p_{0,0}) \quad (1298)$$

$$dp_3 = \text{Abs}(p_{2,3} - 2 * p_{1,3} + p_{0,3}) \quad (1299)$$

$$dq_0 = \text{Abs}(q_{2,0} - 2 * q_{1,0} + q_{0,0}) \quad (1300)$$

$$dq_3 = \text{Abs}(q_{2,3} - 2 * q_{1,3} + q_{0,3}) \quad (1301)$$

Khi maxFilterLengthP và maxFilterLengthQ đều bằng hoặc lớn hơn 3 biểns sp_0 , sq_0 , spq_0 , sp_3 , sq_3 và spq_3 được dẫn xuất như sau:

$$sp_0 = \text{Abs}(p_{3,0} - p_{0,0}) \quad (1302)$$

$$sq_0 = \text{Abs}(q_{0,0} - q_{3,0}) \quad (1303)$$

$$spq_0 = \text{Abs}(p_{0,0} - q_{0,0}) \quad (1304)$$

$$sp_3 = \text{Abs}(p_{3,3} - p_{0,3}) \quad (1305)$$

$$sq3 = \text{Abs}(q_{0,3} - q_{3,3}) \quad (1306)$$

$$spq3 = \text{Abs}(p_{0,3} - q_{0,3}) \quad (1307)$$

1. Các biến sidePisLargeBlk và sideQisLargeBlk được gán bằng 0.

Khi maxFilterLengthP lớn hơn 3, sidePisLargeBlk được gán bằng 1.

Khi maxFilterLengthQ lớn hơn 3, sideQisLargeBlk được gán bằng 1.

Khi edgeType bằng EDGE_HOR và $(yCb + yBl) \% CtbSizeY$ bằng 0, sidePisLargeBlk được gán bằng 0.

Các biến dSam0 và dSam3 được khởi tạo bằng 0.

Khi sidePisLargeBlk hoặc sideQisLargeBlk lớn hơn 0, điều kiện sau áp dụng:

a. Các biến dp0L, dp3L được dẫn xuất và maxFilterLengthP được chỉnh sửa

như sau:

– Nếu sidePisLargeBlk bằng 1, điều kiện sau áp dụng:

$$dp0L = (dp0 + \text{Abs}(p_{5,0} - 2 * p_{4,0} + p_{3,0}) + 1) \gg 1 \quad (1308)$$

$$dp3L = (dp3 + \text{Abs}(p_{5,3} - 2 * p_{4,3} + p_{3,3}) + 1) \gg 1 \quad (1309)$$

– Ngược lại, điều kiện sau áp dụng:

$$dp0L = dp0 \quad (1310)$$

$$dp3L = dp3 \quad (1311)$$

$$\text{maxFilterLengthP} = 3 \quad (1312)$$

b. Các biến dq0L và dq3L được dẫn xuất như sau:

– Nếu sideQisLargeBlk bằng 1, điều kiện sau áp dụng:

$$dq0L = (dq0 + \text{Abs}(q_{5,0} - 2 * q_{4,0} + q_{3,0}) + 1) \gg 1 \quad (1313)$$

$$dq3L = (dq3 + \text{Abs}(q_{5,3} - 2 * q_{4,3} + q_{3,3}) + 1) \gg 1 \quad (1314)$$

– Ngược lại, điều kiện sau áp dụng:

$$dq0L = dq0 \quad (1315)$$

$$dq3L = dq3 \quad (1316)$$

c. Biếns sp0L and sp3L được dẫn xuất như sau:

- Nếu maxFilterLengthP bằng 7, điều kiện sau áp dụng:

$$sp0L = sp0 + Abs(p_{7,0} - p_{6,0} - p_{5,0} + p_{4,0}) \quad (1317)$$

$$sp3L = sp3 + Abs(p_{7,3} - p_{6,3} - p_{5,3} + p_{4,3}) \quad (1318)$$

- Ngược lại, điều kiện sau áp dụng:

$$sp0L = sp0 \quad (1319)$$

$$sp3L = sp3 \quad (1320)$$

d. Các biến sq0L và sq3L được dẫn xuất như sau:

- Nếu maxFilterLengthQ bằng 7, điều kiện sau áp dụng:

$$sq0L = sq0 + Abs(q_{4,0} - q_{5,0} - q_{6,0} + q_{7,0}) \quad (1321)$$

$$sq3L = sq3 + Abs(q_{4,3} - q_{5,3} - q_{6,3} + q_{7,3}) \quad (1322)$$

- Ngược lại, điều kiện sau áp dụng:

$$sq0L = sq0 \quad (1323)$$

$$sq3L = sq3 \quad (1324)$$

e. Các biến dpq0L, dpq3L, và dL được dẫn xuất như sau:

$$dpq0L = dp0L + dq0L \quad (1325)$$

$$dpq3L = dp3L + dq3L \quad (1326)$$

$$dL = dpq0L + dpq3L \quad (1327)$$

f. Khi dL nhỏ hơn β , các bước có thứ tự sau áp dụng:

- Biến dpq được gán bằng $2 * dpq0L$.

Biến sp được gán bằng sp0L, biến sq được gán bằng sq0L và biến spq được gán bằng spq0.

Các biến p_0 p_3 q_0 và q_3 trước hết được khởi tạo bằng 0 và sau đó được chỉnh sửa theo sidePisLargeBlk và sideQisLargeBlk như sau:

- Khi sidePisLargeBlk bằng 1, điều kiện sau áp dụng:

$$p_3 = p_{3,0} \quad (1328)$$

$$p_0 = p_{\text{maxFilterLengthP},0} \quad (1329)$$

- Khi sideQisLargeBlk bằng 1, điều kiện sau áp dụng:

$$q_3 = q_{3,0} \quad (1330)$$

$$q_0 = q_{\maxFilterLengthQ,0} \quad (1331)$$

Đối với vị trí mẫu $(xCb + xBl, yCb + yBl)$, quá trình quyết định cho mẫu độ sáng như được xác định trong mệnh đề 8.8.3.6.5 được gọi với các giá trị mẫu p_0 , p_3 , q_0 , q_3 , các biến dpq , sp , sq , spq , sidePisLargeBlk, sideQisLargeBlk, β và tc làm đầu vào, và đầu ra được gán cho quyết định dSam0.

Biến dpq được gán bằng $2 * dpq3L$.

Biến sp được gán bằng $sp3L$, biến sq được gán bằng $sq3L$ và biến spq được gán bằng $spq3$.

Các biến p_0 , p_3 , q_0 và q_3 trước hết được khởi tạo bằng 0 và sau đó được chỉnh sửa theo sidePisLargeBlk và sideQisLargeBlk như sau:

- Khi sidePisLargeBlk bằng 1, điều kiện sau áp dụng:

$$p_3 = p_{3,3} \quad (1332)$$

$$p_0 = p_{\maxFilterLengthP,3} \quad (1333)$$

- Khi sideQisLargeBlk bằng 1, điều kiện sau áp dụng:

$$q_3 = q_{3,3} \quad (1334)$$

$$q_0 = q_{\maxFilterLengthQ,3} \quad (1335)$$

Khi edgeType bằng EDGE_VER đối với vị trí mẫu $(xCb + xBl, yCb + yBl + 3)$ hoặc khi edgeType bằng EDGE_HOR đối với vị trí mẫu $(xCb + xBl + 3, yCb + yBl)$, quá trình quyết định đối với mẫu độ sáng như được xác định trong mệnh đề 8.8.3.6.5 được gọi với các giá trị mẫu p_0 , p_3 , q_0 , q_3 , các biến dpq , sp , sq , spq , sidePisLargeBlk, sideQisLargeBlk, β and tc làm đầu vào, và đầu ra được gán cho quyết định dSam3.

2. Các biến dE, dEp và dEq được dẫn xuất như sau:

– Nếu dSam0 và dSam3 đều bằng 1, biến dE được gán bằng 3, dEp được gán bằng 1, và dEq được gán bằng 1.

– Ngược lại, các bước có thứ tự sau áp dụng:

Các biến dpq0, dpq3, dp, dq và d được dẫn xuất như sau:

$$dpq0 = dp0 + dq0 \quad (1336)$$

$$dpq3 = dp3 + dq3 \quad (1337)$$

$$dp = dp0 + dp3 \quad (1338)$$

$$dq = dq0 + dq3 \quad (1339)$$

$$d = dpq0 + dpq3 \quad (1340)$$

Các biến dE, dEp, dEq, sidePisLargeBlk và sideQisLargeBlk được gán bằng 0.

Khi d nhỏ hơn β và cả maxFilterLengthP lẫn maxFilterLengthQ lớn hơn 2, các bước có thứ tự sau áp dụng:

Biến dpq được gán bằng $2 * dpq0$.

Biến sp được gán bằng sp0, biến sq được gán bằng sq0 và biến spq được gán bằng spq0.

Đối với vị trí mẫu ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl}$), quá trình quyết định đối với mẫu độ sáng như được xác định trong mệnh đề 8.8.3.6.5 được gọi với các biến p_0 , p_3 , q_0 , q_3 đều được gán bằng 0, các biến dpq, sp, sq, spq, sidePisLargeBlk, sideQisLargeBlk, β và t_c làm đầu vào, và đầu ra được gán cho quyết định dSam0.

Biến dpq được gán bằng $2 * dpq3$.

Biến sp được gán bằng sp3, biến sq được gán bằng sq3 và biến spq được gán bằng spq3.

Khi edgeType bằng EDGE_VER đối với vị trí mẫu ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl} + 3$) hoặc khi edgeType bằng EDGE_HOR đối với vị trí mẫu ($x_{Cb} + x_{Bl} + 3$, $y_{Cb} + y_{Bl}$), quá trình quyết định đối với mẫu như được xác

định trong mệnh đề 8.8.3.6.5 được gọi với các biến p_0, p_3, q_0, q_3 đều được gán bằng 0, các biến $dpq, sp, sq, spq, sidePisLargeBlk, sideQisLargeBlk, \beta$ và tc làm đầu vào, và đầu ra được gán cho quyết định $dSam3$.

Khi d nhỏ hơn β , các bước có thứ tự sau áp dụng:

Biến dE được gán bằng 1.

Khi $dSam0$ bằng 1 và $dSam3$ bằng 1, biến dE được gán bằng 2 và cả $maxFilterLengthP$ và $maxFilterLengthQ$ được gán bằng 3.

Khi $maxFilterLengthP$ lớn hơn 1, và $maxFilterLengthQ$ lớn hơn 1, và dp nhỏ hơn $(\beta + (\beta \gg 1)) \gg 3$, biến dEp được gán bằng 1.

Khi $maxFilterLengthP$ lớn hơn 1, và $maxFilterLengthQ$ lớn hơn 1, và dq nhỏ hơn $(\beta + (\beta \gg 1)) \gg 3$, biến dEq được gán bằng 1.

Khi dE bằng 1, $maxFilterLengthP$ được gán bằng $1 + dEp$ và $maxFilterLengthQ$ được gán bằng $1 + dEq$.

Bảng 43 – Dẫn xuất các biến ngưỡng β' và tc' từ đầu vào Q

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
tc'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
β'	7	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28
tc'	0	3	4	4	4	4	5	5	5	5	7	7	8	9	10	10	11
Q	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
β'	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
tc'	13	14	15	17	19	21	24	25	29	33	36	41	45	51	57	64	71
Q	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65		
β'	64	66	68	70	72	74	76	78	80	82	84	86	88	-	-		
tc'	80	89	100	112	125	141	157	177	198	222	250	280	314	352	395		

8.8.3.6.2 Quá trình lọc cho các cạnh khối độ sáng

Các đầu vào cho quá trình này là:

- mảng mẫu ảnh `recPicture`,

- vị trí (xCb, yCb) xác định mẫu trên bên trái của khối mã hiện tại so với mẫu trên bên trái của ảnh hiện tại,
- vị trí (xBl, yBl) xác định mẫu trên bên trái của khối hiện tại so với mẫu trên bên trái của khối mã hiện tại,
- biến edgeType xác định liệu cạnh dọc (EDGE_VER) hoặc ngang (EDGE_HOR) được lọc,
- các biến dE, dEp và dEq có các quyết định,
- các biến maxFilterLengthP và maxFilterLengthQ có các chiều dài lọc lớn nhất,
- biến tc.

Đầu ra của quá trình này là mảng mẫu ảnh được chỉnh sửa recPicture.

Tuỳ thuộc vào giá trị của edgeType, điều kiện sau áp dụng:

- Nếu edgeType bằng EDGE_VER, các bước có thứ tự sau áp dụng:

Các giá trị mẫu $p_{i,k}$ và $q_{j,k}$ với $i = 0..maxFilterLengthP$, $j = 0..maxFilterLengthQ$ và $k = 0..3$ được dẫn xuất như sau:

$$q_{j,k} = recPicture[xCb + xBl + j][yCb + yBl + k] \quad (1341)$$

$$p_{i,k} = recPicture[xCb + xBl - i - 1][yCb + yBl + k] \quad (1342)$$

Khi dE không bằng 0 và dE không bằng 3, đối với mỗi vị trí mẫu $(xCb + xBl, yCb + yBl + k)$, $k = 0..3$, các bước có thứ tự sau áp dụng:

Quá trình lọc đối với mẫu độ sáng sử dụng bộ lọc ngắn như được xác định trong mệnh đề 8.8.3.6.6 được gọi với các biến maxFilterLengthP, maxFilterLengthQ, các giá trị mẫu $p_{i,k}$, $q_{j,k}$ với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, quyết định dE, các biến dEp và dEq và biến tc làm đầu vào, và số lượng mẫu được lọc nDp và nDq từ mỗi phía của đường biên khối và các giá trị mẫu được lọc p_i' và q_j' làm đầu ra.

Khi nDp lớn hơn 0, các giá trị mẫu được lọc p_i' với $i = 0..nDp - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl - i - 1][yCb + yBl + k] = p_i' \quad (1343)$$

Khi nDq lớn hơn 0, các giá trị mẫu được lọc q_j' với $j = 0..nDq - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + j][yCb + yBl + k] = q_j' \quad (1344)$$

Khi dE bằng 3, đối với mỗi vị trí mẫu $(xCb + xBl, yCb + yBl + k)$, $k = 0..3$, các bước có thứ tự sau áp dụng:

Quá trình lọc đối với mẫu độ sáng sử dụng các bộ lọc dài như được xác định trong mệnh đề 8.8.3.6.7 được gọi với các biến $maxFilterLengthP$, $maxFilterLengthQ$, các giá trị mẫu $p_{i,k}$, $q_{j,k}$ với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, và tc làm đầu vào và các giá trị mẫu được lọc p_i' và q_j' làm đầu ra.

Các giá trị mẫu được lọc p_i' với $i = 0..maxFilterLengthP - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl - i - 1][yCb + yBl + k] = p_i' \quad (1345)$$

Các giá trị mẫu được lọc q_j' với $j = 0..maxFilterLengthQ - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + j][yCb + yBl + k] = q_j' \quad (1346)$$

– Ngược lại ($edgeType$ bằng $EDGE_HOR$), các bước có thứ tự sau áp dụng:

1. Các giá trị mẫu $p_{i,k}$ và $q_{j,k}$ với $i = 0..maxFilterLengthP$, $j = 0..maxFilterLengthQ$ và $k = 0..3$ được dẫn xuất như sau:

$$q_{j,k} = recPicture[xCb + xBl + k][yCb + yBl + j] \quad (1347)$$

$$p_{i,k} = recPicture[xCb + xBl + k][yCb + yBl - i - 1] \quad (1348)$$

2. Khi dE không bằng 0 và dE không bằng 3, đối với mỗi vị trí mẫu $(xCb + xBl + k, yCb + yBl)$, $k = 0..3$, các bước có thứ tự sau áp dụng:

Quá trình lọc cho mẫu độ sáng nhờ sử dụng các bộ lọc ngắn như được xác định trong mệnh đề 8.8.3.6.6 được gọi với các biến $maxFilterLengthP$, $maxFilterLengthQ$, các giá trị mẫu $p_{i,k}$, $q_{i,k}$ với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, quyết định dE , các biến dEp và dEq , và biến tc làm đầu vào, và số lượng mẫu được lọc nDp và nDq từ mỗi phía của đường biên khối và các giá trị mẫu được lọc p_i' và q_j' làm đầu ra.

Khi nDp lớn hơn 0, các giá trị mẫu được lọc p_i' với $i = 0..nDp - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + k][yCb + yBl - i - 1] = p_i' \quad (1349)$$

Khi nDq lớn hơn 0, các giá trị mẫu được lọc q_j' với $j = 0..nDq - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + k][yCb + yBl + j] = q_j' \quad (1350)$$

3. Khi dE bằng 3, đối với mỗi vị trí mẫu $(xCb + xBl + k, yCb + yBl)$, $k = 0..3$, các bước có thứ tự sau áp dụng:

Quá trình lọc cho mẫu độ sáng sử dụng các bộ lọc dài như được xác định trong mệnh đề 8.8.3.6.7 được gọi với các biến $maxFilterLengthP$, $maxFilterLengthQ$, các giá trị mẫu $p_{i,k}$, $q_{j,k}$ với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, và biến tc làm đầu vào, và các giá trị mẫu được lọc p_i' và q_j' làm đầu ra.

Các giá trị mẫu được lọc p_i' với $i = 0..maxFilterLengthP - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + k][yCb + yBl - i - 1] = p_i' \quad (1351)$$

Các giá trị mẫu được lọc q_j' với $j = 0..maxFilterLengthQ - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$\text{recPicture}[\text{xCb} + \text{xBl} + \text{k}][\text{yCb} + \text{yBl} + \text{j}] = \text{q}_i' \quad (1352)$$

8.8.3.6.3 Quá trình quyết định đối với các cạnh khối sắc độ

Quá trình này chỉ được gọi khi ChromaArrayType không bằng 0.

Các đầu vào cho quá trình này là:

- mảng mẫu ảnh sắc độ recPicture,
- vị trí sắc độ (xCb, yCb) xác định mẫu trên bên trái của khối mã sắc độ hiện tại so với mẫu sắc độ trên bên trái của ảnh hiện tại,
- vị trí sắc độ (xBl, yBl) xác định mẫu trên bên trái của khối sắc độ hiện tại so với mẫu trên bên trái của khối mã sắc độ hiện tại,
- biến edgeType xác định liệu cạnh dọc (EDGE_VER) hoặc ngang (EDGE_HOR) được lọc,
- biến cIdx xác định chỉ số thành phần màu,
- biến bS xác định cường độ lọc đường biên,
- biến maxFilterLengthP xác định chiều dài lọc lớn nhất,
- biến maxFilterLengthQ xác định chiều dài lọc lớn nhất.

Các đầu ra của quá trình này là:

- các biến chiều dài lọc được chỉnh sửa maxFilterLengthP và maxFilterLengthQ,
- biến tc.

Biến maxK được dẫn xuất như sau:

- Nếu edgeType bằng EDGE_VER, điều kiện sau áp dụng:

$$\text{maxK} = (\text{SubHeightC} == 1) ? 3 : 1 \quad (1353)$$

- Ngược lại (edgeType bằng EDGE_HOR), điều kiện sau áp dụng:

$$\text{maxK} = (\text{SubWidthC} == 1) ? 3 : 1 \quad (1354)$$

Các giá trị $p_{i,k}$ và $q_{i,k}$ với $i = 0..\text{maxFilterLengthP}$, $j = 0..\text{maxFilterLengthQ}$ và $k = 0..\text{maxK}$ được dẫn xuất như sau:

- Nếu edgeType bằng EDGE_VER, điều kiện sau áp dụng:

$$q_{j,k} = \text{recPicture}[xCb + xBl + j][yCb + yBl + k] \quad (1355)$$

$$p_{i,k} = \text{recPicture}[xCb + xBl - i - 1][yCb + yBl + k] \quad (1356)$$

$$\text{subSampleC} = \text{SubHeightC} \quad (1357)$$

- Ngược lại (edgeType bằng EDGE_HOR), điều kiện sau áp dụng:

$$q_{j,k} = \text{recPicture}[xCb + xBl + k][yCb + yBl + j] \quad (1358)$$

$$p_{i,k} = \text{recPicture}[xCb + xBl + k][yCb + yBl - i - 1] \quad (1359)$$

$$\text{subSampleC} = \text{SubWidthC} \quad (1360)$$

Biến Q_{pP} được dẫn xuất như sau:

- Vị trí độ sáng (xTb_P , yTb_P) được gán là vị trí mẫu độ sáng trên bên trái của khối biến đổi chứa mẫu $p_{0,0}$, so với mẫu độ sáng trên bên trái của ảnh.

- Nếu $TuCResMode[xTb_P][yTb_P]$ bằng 2, Q_{pP} được gán bằng $Q_{p'CbCr}$ của khối biến đổi chứa mẫu $p_{0,0}$.

- Ngược lại, nếu $cIdx$ bằng 1, Q_{pP} được gán bằng $Q_{p'Cb}$ của khối biến đổi chứa mẫu $p_{0,0}$.

- Ngược lại, Q_{pP} được gán bằng $Q_{p'Cr}$ of khối biến đổi chứa mẫu $p_{0,0}$.

Biến Q_{pQ} được dẫn xuất như sau:

- Vị trí độ sáng (xTb_Q , yTb_Q) được gán bằng vị trí mẫu độ sáng trên bên trái của khối biến đổi chứa mẫu $q_{0,0}$, so với mẫu độ sáng trên bên trái của ảnh.

- Nếu $TuCResMode[xTb_Q][yTb_Q]$ bằng 2, Q_{pQ} được gán bằng $Q_{p'CbCr}$ của khối biến đổi chứa mẫu $q_{0,0}$.

- Ngược lại, nếu $cIdx$ bằng 1, Q_{pQ} được gán bằng $Q_{p'Cb}$ của khối biến đổi chứa mẫu $q_{0,0}$.

- Ngược lại, Q_{pQ} được gán bằng $Q_{p'Cr}$ của khối biến đổi chứa mẫu $q_{0,0}$.

Biến Q_{pC} được dẫn xuất như sau:

$$Q_{pC} = (Q_{pQ} - Q_{pBdOffset} + Q_{pP} - Q_{pBdOffset} + 1) \gg 1 \quad (1361)$$

Giá trị của biến β' được xác định như được xác định trong Bảng 43 dựa trên tham số lượng tử Q được dẫn xuất như sau:

$$\text{sliceBetaOffsetDiv2} = (\text{cIdx} == 1 ? \text{slice_cb_beta_offset_div2} : \text{slice_cr_beta_offset_div2})$$

$$Q = \text{Clip3}(0, 63, Q_{pc} + (\text{sliceBetaOffsetDiv2} \ll 1)) \quad (1362)$$

trong đó $\text{slice_cb_beta_offset_div2}$ và $\text{slice_cr_beta_offset_div2}$ lần lượt là các giá trị của các phần tử cú pháp $\text{slice_cb_beta_offset_div2}$ và $\text{slice_cr_beta_offset_div2}$, cho lát chứa mẫu $q_{0,0}$.

Biến β được dẫn xuất như sau:

$$\beta = \beta' * (1 \ll (\text{BitDepth} - 8)) \quad (1363)$$

Giá trị của biến tc' được xác định như được xác định trong Bảng 43 dựa trên tham số lượng tử sắc độ Q được dẫn xuất như sau:

$$\text{sliceTcOffsetDiv2} = (\text{cIdx} == 1 ? \text{slice_cb_tc_offset_div2} : \text{slice_cr_beta_offset_div2})$$

$$Q = \text{Clip3}(0, 65, Q_{pc} + 2 * (bS - 1) + (\text{sliceTcOffsetDiv2} \ll 1)) \quad (1364)$$

trong đó $\text{slice_cb_tc_offset_div2}$ và $\text{slice_cr_beta_offset_div2}$ lần lượt là các giá trị của các phần tử cú pháp $\text{slice_cb_tc_offset_div2}$ và $\text{slice_cr_beta_offset_div2}$, cho lát chứa mẫu $q_{0,0}$.

Biến tc được dẫn xuất như sau:

$$tc = (\text{BitDepth} < 10) ? (tc' + 2) \gg (10 - \text{BitDepth}) : tc' * (1 \ll (\text{BitDepth} - 10)) \quad (1365)$$

Khi cả maxFilterLengthP lẫn maxFilterLengthQ bằng 1 và bS không bằng 2, maxFilterLengthP và maxFilterLengthQ đều được gán bằng 0.

Khi maxFilterLengthQ bằng 3, các bước có thứ tự sau áp dụng:

1. Các biến $n1$ được dẫn xuất như sau:

$$n1 = \text{subSampleC} == 2 ? 1 : 3 \quad (1366)$$

2. Khi maxFilterLengthP bằng 1, các mẫu $p_{3,0}$ và $p_{2,0}$ đều được gán bằng $p_{1,0}$

và các mẫu $p_{3,n1}$, $p_{2,n1}$ đều được gán bằng $p_{1,n1}$.

3. Các biến $dpq0$, $dpq1$, dp , dq và d được dẫn xuất như sau:

$$dp0 = \text{Abs}(p_{2,0} - 2 * p_{1,0} + p_{0,0}) \quad (1367)$$

$$dp1 = \text{Abs}(p_{2,n1} - 2 * p_{1,n1} + p_{0,n1}) \quad (1368)$$

$$dq0 = \text{Abs}(q_{2,0} - 2 * q_{1,0} + q_{0,0}) \quad (1369)$$

$$dq1 = \text{Abs}(q_{2,n1} - 2 * q_{1,n1} + q_{0,n1}) \quad (1370)$$

$$dpq0 = dp0 + dq0 \quad (1371)$$

$$dpq1 = dp1 + dq1 \quad (1372)$$

$$dp = dp0 + dp1 \quad (1373)$$

$$dq = dq0 + dq1 \quad (1374)$$

$$d = dpq0 + dpq1 \quad (1375)$$

4. Các biến $dSam0$ và $dSam1$ đều được gán bằng 0.

5. Khi d nhỏ hơn β , các bước có thứ tự sau áp dụng:

a. Biến dpq được gán bằng $2 * dpq0$.

b. Biến $dSam0$ được dẫn xuất bằng cách gọi quá trình quyết định cho mẫu sắc độ như được xác định trong mệnh đề 8.8.3.6.8 đối với vị trí mẫu $(xCb + xBl, yCb + yBl)$ với các giá trị mẫu $p_{0,0}$, $p_{3,0}$, $q_{0,0}$, và $q_{3,0}$, các biến dpq , β và t_c làm đầu vào, và đầu ra được gán cho quyết định $dSam0$.

c. Biến dpq được gán bằng $2 * dpq1$.

d. Biến $dSam1$ được chỉnh sửa như sau:

– Nếu $edgeType$ bằng $EDGE_VER$, đối với vị trí mẫu $(xCb + xBl, yCb + yBl + n1)$, quá trình quyết định cho mẫu sắc độ như được xác định trong mệnh đề 8.8.3.6.8 được gọi với các giá trị mẫu $p_{0,n1}$, $p_{3,n1}$, $q_{0,n1}$, và $q_{3,n1}$, các biến dpq , β và t_c làm đầu vào, và đầu ra được gán cho quyết định $dSam1$.

– Ngược lại ($edgeType$ bằng $EDGE_HOR$), đối với vị trí mẫu $(xCb + xBl + n1, yCb + yBl)$, quá trình quyết định cho mẫu sắc độ như được xác

định trong mệnh đề 8.8.3.6.8 được gọi với các giá trị mẫu $p_{0,n1}$, $p_{3,n1}$, $q_{0,n1}$ và $q_{3,n1}$, các biến dpq , β và t_c làm đầu vào, và đầu ra được gán cho quyết định $dSam1$.

6. Khi $dSam0$ bằng 0 hoặc $dSam1$ bằng 0, $maxFilterLengthP$ và $maxFilterLengthQ$ đều được gán bằng 1.

8.8.3.6.4 Quá trình lọc đối với các cạnh khối sắc độ

Quá trình này chỉ được gọi khi $ChromaArrayType$ không bằng 0.

Các đầu vào cho quá trình này là:

- mảng mẫu ảnh sắc độ $recPicture$,
- vị trí sắc độ (xCb , yCb) xác định mẫu trên bên trái của khối mã sắc độ hiện tại so với mẫu sắc độ trên bên trái của ảnh hiện tại,
- vị trí sắc độ ($xB1$, $yB1$) xác định mẫu trên bên trái của khối sắc độ hiện tại so với mẫu trên bên trái của khối mã sắc độ hiện tại,
- biến $edgeType$ xác định liệu cạnh dọc ($EDGE_VER$) hoặc ngang ($EDGE_HOR$) được lọc,
- biến $maxFilterLengthP$ xác định chiều dài lọc lớn nhất,
- biến $maxFilterLengthQ$ xác định chiều dài lọc lớn nhất,
- biến tC .

Đầu ra của quá trình này là mảng mẫu ảnh sắc độ được chỉnh sửa $recPicture$.

Biến $maxK$ được dẫn xuất như sau:

- Nếu $edgeType$ bằng $EDGE_VER$, điều kiện sau áp dụng:

$$maxK = (SubHeightC == 1) ? 3 : 1 \quad (1376)$$

- Ngược lại ($edgeType$ bằng $EDGE_HOR$), điều kiện sau áp dụng:

$$maxK = (SubWidthC == 1) ? 3 : 1 \quad (1377)$$

Các giá trị p_i với $i = 0..maxFilterLengthP$, q_j với $j = 0..maxFilterLengthQ$, và $k = 0..maxK$ được dẫn xuất như sau:

- If edgeType bằng EDGE_VER, điều kiện sau áp dụng:

$$q_{j,k} = \text{recPicture}[xCb + xBl + j][yCb + yBl + k] \quad (1378)$$

$$p_{i,k} = \text{recPicture}[xCb + xBl - i - 1][yCb + yBl + k] \quad (1379)$$

- Ngược lại (edgeType bằng EDGE_HOR), điều kiện sau áp dụng:

$$q_{j,k} = \text{recPicture}[xCb + xBl + k][yCb + yBl + j] \quad (1380)$$

$$p_{i,k} = \text{recPicture}[xCb + xBl + k][yCb + yBl - i - 1] \quad (1381)$$

Phụ thuộc vào giá trị của edgeType, điều kiện sau áp dụng:

- Nếu edgeType bằng EDGE_VER, đối với mỗi vị trí mẫu $(xCb + xBl, yCb + yBl + k)$, $k = 0..maxK$, các bước có thứ tự sau áp dụng:

1. Quá trình lọc cho mẫu sắc độ như được xác định trong mệnh đề 8.8.3.6.9 được gọi với các biến maxFilterLengthP và maxFilterLengthQ, các giá trị mẫu $p_{i,k}$, $q_{j,k}$ với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, và biến tc làm đầu vào, và các giá trị mẫu được lọc p_i' và q_j' với $i = 0..maxFilterLengthP - 1$ và $j = 0..maxFilterLengthQ - 1$ làm đầu ra.

2. Các giá trị mẫu được lọc p_i' và q_j' với $i = 0..maxFilterLengthP - 1$ và $j = 0..maxFilterLengthQ - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu recPicture như sau:

$$\text{recPicture}[xCb + xBl + j][yCb + yBl + k] = q_j' \quad (1382)$$

$$\text{recPicture}[xCb + xBl - i - 1][yCb + yBl + k] = p_i' \quad (1383)$$

- Ngược lại (edgeType bằng EDGE_HOR), đối với mỗi vị trí mẫu $(xCb + xBl + k, yCb + yBl)$, $k = 0..maxK$, các bước có thứ tự sau áp dụng:

1. Quá trình lọc cho mẫu sắc độ như được xác định trong mệnh đề 8.8.3.6.9 được gọi với biến maxFilterLengthP và maxFilterLengthQ, các giá trị mẫu $p_{i,k}$, $q_{j,k}$, với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$, và biến tc làm đầu vào, và các giá trị mẫu được lọc p_i' và q_j' với $i = 0..maxFilterLengthP - 1$ và $j = 0..maxFilterLengthQ - 1$ làm đầu ra.

2. Các giá trị mẫu được lọc p_i' và q_j' với $i = 0..maxFilterLengthP - 1$ và $j = 0..maxFilterLengthQ - 1$ thay thế các mẫu tương ứng bên trong mảng mẫu $recPicture$ như sau:

$$recPicture[xCb + xBl + k][yCb + yBl + j] = q_j' \quad (1384)$$

$$recPicture[xCb + xBl + k][yCb + yBl - i - 1] = p_i' \quad (1385)$$

8.8.3.6.5 Quá trình quyết định đối với mẫu độ sáng

Các đầu vào cho quá trình này là:

- các giá trị mẫu p_0, p_3, q_0 và q_3 ,
- các biến $dpq, sp, sq, spq, sidePisLargeBlk, sideQisLargeBlk, \beta$ và tc .

Đầu ra của quá trình này là biến $dSam$ chứa quyết định.

Các biến sp và sq được chỉnh sửa như sau:

- Khi $sidePisLargeBlk$ bằng 1, điều kiện sau áp dụng:

$$sp = (sp + Abs(p_3 - p_0) + 1) \gg 1 \quad (1386)$$

- Khi $sideQisLargeBlk$ bằng 1, điều kiện sau áp dụng:

$$sq = (sq + Abs(q_3 - q_0) + 1) \gg 1 \quad (1387)$$

Các biến $sThr1$ và $sThr2$ được dẫn xuất như sau:

- Nếu $sidePisLargeBlk$ bằng 1 hoặc $sideQisLargeBlk$ bằng 1, điều kiện

sau áp dụng:

$$sThr1 = 3 * \beta \gg 5 \quad (1388)$$

$$sThr2 = \beta \gg 4 \quad (1389)$$

- Ngược lại, điều kiện sau áp dụng:

$$sThr1 = \beta \gg 3 \quad (1390)$$

$$sThr2 = \beta \gg 2 \quad (1391)$$

Biến $dSam$ được xác định như sau:

- Nếu tất cả các điều kiện sau là đúng, $dSam$ được gán bằng 1:
- dpq nhỏ hơn $sThr2$,

- $sp + sq$ nhỏ hơn $sThr1$,
- spq nhỏ hơn $(5 * tc + 1) \ggg 1$.
- Ngược lại, $dSam$ được gán bằng 0.

8.8.3.6.6 Quá trình lọc cho mẫu độ sáng sử dụng các bộ lọc ngắn

Các đầu vào cho quá trình này là:

- Các biến $maxFilterLengthP$ và $maxFilterLengthQ$,
- các giá trị mẫu p_i và q_j với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$,
- biến dE ,
- các biến dEp và dEq lần lượt chứa các quyết định để lọc các mẫu $p1$ và $q1$,
- biến tc .

Các đầu ra của quá trình này là:

- số lượng mẫu được lọc nDp và nDq ,
- các giá trị mẫu được lọc p_i' và q_j' với $i = 0..nDp - 1, j = 0..nDq - 1$.

Phụ thuộc vào giá trị của dE , điều kiện sau áp dụng:

- Nếu biến dE bằng 2, nDp và nDq đều được gán bằng 3 và bộ lọc mạnh sau áp dụng:

$$p_0' = \text{Clip3}(p_0 - 3 * tc, p_0 + 3 * tc, (p_2 + 2 * p_1 + 2 * p_0 + 2 * q_0 + q_1 + 4) \ggg 3) \quad (1392)$$

$$p_1' = \text{Clip3}(p_1 - 2 * tc, p_1 + 2 * tc, (p_2 + p_1 + p_0 + q_0 + 2) \ggg 2) \quad (1393)$$

$$p_2' = \text{Clip3}(p_2 - 1 * tc, p_2 + 1 * tc, (2 * p_3 + 3 * p_2 + p_1 + p_0 + q_0 + 4) \ggg 3) \quad (1394)$$

$$q_0' = \text{Clip3}(q_0 - 3 * tc, q_0 + 3 * tc, (p_1 + 2 * p_0 + 2 * q_0 + 2 * q_1 + q_2 + 4) \ggg 3) \quad (1395)$$

$$q_1' = \text{Clip3}(q_1 - 2 * tc, q_1 + 2 * tc, (p_0 + q_0 + q_1 + q_2 + 2) \ggg 2) \quad (1396)$$

$$q_2' = \text{Clip3}(q_2 - 1 * tc, q_2 + 1 * tc, (p_0 + q_0 + q_1 + 3 * q_2 + 2 * q_3 + 4) \ggg 3)$$

(1397)

- Ngược lại, nDp và nDq được đều gán bằng 0 và lọc yếu sau áp dụng:
- Điều kiện sau áp dụng:

$$\Delta = (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) \gg 4 \quad (1398)$$

- Khi Abs(Δ) nhỏ hơn $t_c * 10$, các bước có thứ tự sau áp dụng:
- Các giá trị mẫu được lọc p_0' và q_0' được xác định như sau:

$$\Delta = \text{Clip3}(-t_c, t_c, \Delta) \quad (1399)$$

$$p_0' = \text{Clip1}(p_0 + \Delta) \quad (1400)$$

$$q_0' = \text{Clip1}(q_0 - \Delta) \quad (1401)$$

- Khi dEp bằng 1, giá trị mẫu được lọc p_1' được xác định như sau:

$$\Delta_p = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, (((p_2 + p_0 + 1) \gg 1) - p_1 + \Delta) \gg 1) \quad (1402)$$

$$p_1' = \text{Clip1}(p_1 + \Delta_p) \quad (1403)$$

- Khi dEq bằng 1, giá trị mẫu được lọc q_1' được xác định như sau:

$$\Delta_q = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, (((q_2 + q_0 + 1) \gg 1) - q_1 - \Delta) \gg 1) \quad (1404)$$

$$q_1' = \text{Clip1}(q_1 + \Delta_q) \quad (1405)$$

- nDp được gán bằng dEp + 1 và nDq được gán bằng dEq + 1.

Khi nDp lớn hơn 0 và pred_mode_plt_flag của đơn vị mã bao gồm khối mã chứa mẫu p_0 bằng 1, nDp được gán bằng 0

Khi nDq lớn hơn 0 và pred_mode_plt_flag của đơn vị mã bao gồm khối mã chứa mẫu q_0 bằng 1, nDq được gán bằng 0:

8.8.3.6.7 Quá trình lọc cho mẫu độ sáng sử dụng các bộ lọc dài

Các đầu vào cho quá trình này là:

- Các biến maxFilterLengthP và maxFilterLengthQ,

– các giá trị mẫu p_i và q_j với $i = 0..maxFilterLengthP$ và $j = 0..maxFilterLengthQ$,

– biến tc.

Các đầu ra của quá trình này là:

– các giá trị mẫu được lọc p'_i và q'_j với $i = 0..maxFilterLengthP - 1$, $j = 0..maxFilterLengthQ - 1$.

Biến refMiddle được dẫn xuất như sau:

– Nếu maxFilterLengthP bằng maxFilterLengthQ và maxFilterLengthP bằng 5, điều kiện sau áp dụng:

$$refMiddle = (p_4 + p_3 + 2 * (p_2 + p_1 + p_0 + q_0 + q_1 + q_2) + q_3 + q_4 + 8) \gg 4 \quad (1406)$$

– Ngược lại, nếu maxFilterLengthP bằng maxFilterLengthQ và maxFilterLengthP không bằng 5, điều kiện sau áp dụng:

$$refMiddle = (p_6 + p_5 + p_4 + p_3 + p_2 + p_1 + 2 * (p_0 + q_0) + q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4 \quad (1407)$$

- Ngược lại, nếu một trong các điều kiện sau đúng,
- maxFilterLengthQ bằng 7 và maxFilterLengthP bằng 5,
- maxFilterLengthQ bằng 5 và maxFilterLengthP bằng 7,

phương trình sau áp dụng:

$$refMiddle = (p_5 + p_4 + p_3 + p_2 + 2 * (p_1 + p_0 + q_0 + q_1) + q_2 + q_3 + q_4 + q_5 + 8) \gg 4 \quad (1408)$$

- Ngược lại, nếu một trong các điều kiện sau đúng,
- maxFilterLengthQ bằng 5 và maxFilterLengthP bằng 3,
- maxFilterLengthQ bằng 3 và maxFilterLengthP bằng 5,

phương trình sau áp dụng:

$$refMiddle = (p_3 + p_2 + p_1 + p_0 + q_0 + q_1 + q_2 + q_3 + 4) \gg 3 \quad (1409)$$

– Ngược lại, nếu maxFilterLengthQ bằng 7 và maxFilterLengthP bằng 3, điều kiện sau áp dụng:

$$\text{refMiddle} = (2 * (p_2 + p_1 + p_0 + q_0) + p_0 + p_1 + q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4 \quad (1410)$$

– Ngược lại, điều kiện sau áp dụng:

$$\text{refMiddle} = (p_6 + p_5 + p_4 + p_3 + p_2 + p_1 + 2 * (q_2 + q_1 + q_0 + p_0) + q_0 + q_1 + 8) \gg 4 \quad (1411)$$

Các biến refP và refQ được dẫn xuất như sau:

$$\text{refP} = (p_{\text{maxFilterLengthP}} + p_{\text{maxFilterLengthP}-1} + 1) \gg 1 \quad (1412)$$

$$\text{refQ} = (q_{\text{maxFilterLengthQ}} + q_{\text{maxFilterLengthQ}-1} + 1) \gg 1 \quad (1413)$$

Các biến f_i và tcPD_i được định nghĩa như sau:

– Nếu maxFilterLengthP bằng 7, điều kiện sau áp dụng:

$$f_{0..6} = \{59, 50, 41, 32, 23, 14, 5\} \quad (1414)$$

$$\text{tcPD}_{0..6} = \{6, 5, 4, 3, 2, 1, 1\} \quad (1415)$$

– Ngược lại, nếu maxFilterLengthP bằng 5, phương trình sau áp dụng:

$$f_{0..4} = \{58, 45, 32, 19, 6\} \quad (1416)$$

$$\text{tcPD}_{0..4} = \{6, 5, 4, 3, 2\} \quad (1417)$$

– Ngược lại, phương trình sau áp dụng:

$$f_{0..2} = \{53, 32, 11\} \quad (1418)$$

$$\text{tcPD}_{0..2} = \{6, 4, 2\} \quad (1419)$$

Các biến g_j và tcQD_j được định nghĩa như sau:

– Nếu maxFilterLengthQ bằng 7, phương trình sau áp dụng:

$$g_{0..6} = \{59, 50, 41, 32, 23, 14, 5\} \quad (1420)$$

$$\text{tcQD}_{0..6} = \{6, 5, 4, 3, 2, 1, 1\} \quad (1421)$$

– Ngược lại, nếu maxFilterLengthQ bằng 5, phương trình sau áp dụng:

$$g_{0..4} = \{58, 45, 32, 19, 6\} \quad (1422)$$

$$\text{tcQD}_{0..4} = \{6, 5, 4, 3, 2\} \quad (1423)$$

– Ngược lại, phương trình sau áp dụng:

$$g_{0..2} = \{53, 32, 11\} \quad (1424)$$

$$\text{tcQD}_{0..2} = \{6, 4, 2\} \quad (1425)$$

Các giá trị mẫu được lọc p_i' và q_j' với $i = 0..maxFilterLengthP - 1$ và $j = 0..maxFilterLengthQ - 1$ được dẫn xuất như sau:

$$p_i' = \text{Clip3}(p_i - (t_c * t_{cPD_i}) \gg 1, p_i + (t_c * t_{cPD_i}) \gg 1, (\text{refMiddle} * f_i + \text{refP} * (64 - f_i) + 32) \gg 6) \quad (1426)$$

$$q_j' = \text{Clip3}(q_j - (t_c * t_{cQD_j}) \gg 1, q_j + (t_c * t_{cQD_j}) \gg 1, (\text{refMiddle} * g_j + \text{refQ} * (64 - g_j) + 32) \gg 6) \quad (1427)$$

Khi `pred_mode_plt_flag` của đơn vị mã bao gồm khối mã chứa mẫu p_i bằng 1, giá trị mẫu được lọc, p_i' được thay thế bằng giá trị mẫu đầu vào tương ứng p_i với $i = 0..maxFilterLengthP - 1$.

Khi `pred_mode_plt_flag` của đơn vị mã bao gồm khối mã chứa mẫu q_j bằng 1, giá trị mẫu được lọc, q_j' được thay thế bằng giá trị mẫu đầu vào tương ứng q_j với $j = 0..maxFilterLengthQ - 1$.

8.8.3.6.8 Quá trình quyết định cho mẫu sắc độ

Các đầu vào cho quá trình này là:

- các giá trị mẫu p_0, p_3, q_0 và q_3 ,
- Các biến dpq, β và t_c .

Đầu ra của quá trình này là biến `dSam` chứa quyết định.

Biến `dSam` được xác định như sau:

- Nếu tất cả các điều kiện sau là đúng, `dSam` được gán bằng 1:
- dpq nhỏ hơn $(\beta \gg 2)$,
- $\text{Abs}(p_3 - p_0) + \text{Abs}(q_0 - q_3)$ nhỏ hơn $(\beta \gg 3)$,
- $\text{Abs}(p_0 - q_0)$ nhỏ hơn $(5 * t_c + 1) \gg 1$.
- Ngược lại, `dSam` được gán bằng 0.

8.8.3.6.9 Quá trình lọc cho mẫu sắc độ

Quá trình này chỉ được gọi khi `ChromaArrayType` không bằng 0.

Các đầu vào cho quá trình này là:

- Các biến maxFilterLengthP và maxFilterLengthQ ,
- các giá trị mẫu sắc độ p_i và q_j với $i = 0..\text{maxFilterLengthP} - 1$ và $j = 0..\text{maxFilterLengthQ} - 1$,
- biến t_c .

Các đầu ra của quá trình này là các giá trị mẫu được lọc p_i' và q_j' với $i = 0..\text{maxFilterLengthP} - 1$ và $j = 0..\text{maxFilterLengthQ} - 1$.

Các giá trị mẫu được lọc p_i' và q_j' với $i = 0..\text{maxFilterLengthP} - 1$ và $j = 0..\text{maxFilterLengthQ} - 1$ được dẫn xuất như sau:

- Nếu cả maxFilterLengthP lẫn maxFilterLengthQ bằng 3, áp dụng bộ lọc mạnh sau:

$$p_0' = \text{Clip3}(p_0 - t_c, p_0 + t_c, (p_3 + p_2 + p_1 + 2 * p_0 + q_0 + q_1 + q_2 + 4) >> 3) \quad (1428)$$

$$p_1' = \text{Clip3}(p_1 - t_c, p_1 + t_c, (2 * p_3 + p_2 + 2 * p_1 + p_0 + q_0 + q_1 + 4) >> 3) \quad (1429)$$

$$p_2' = \text{Clip3}(p_2 - t_c, p_2 + t_c, (3 * p_3 + 2 * p_2 + p_1 + p_0 + q_0 + 4) >> 3) \quad (1430)$$

$$q_0' = \text{Clip3}(q_0 - t_c, q_0 + t_c, (p_2 + p_1 + p_0 + 2 * q_0 + q_1 + q_2 + q_3 + 4) >> 3) \quad (1431)$$

$$q_1' = \text{Clip3}(q_1 - t_c, q_1 + t_c, (p_1 + p_0 + q_0 + 2 * q_1 + q_2 + 2 * q_3 + 4) >> 3) \quad (1432)$$

$$q_2' = \text{Clip3}(q_2 - t_c, q_2 + t_c, (p_0 + q_0 + q_1 + 2 * q_2 + 3 * q_3 + 4) >> 3) \quad (1433)$$

- Ngược lại, nếu biến maxFilterLengthP bằng 1 và maxFilterLengthQ bằng 3, áp dụng bộ lọc sau:

$$p_0' = \text{Clip3}(p_0 - t_c, p_0 + t_c, (3 * p_1 + 2 * p_0 + q_0 + q_1 + q_2 + 4) >> 3) \quad (1434)$$

$$q_0' = \text{Clip3}(q_0 - t_c, q_0 + t_c, (2 * p_1 + p_0 + 2 * q_0 + q_1 + q_2 + q_3 + 4) >> 3) \quad (1435)$$

$$q_1' = \text{Clip3}(q_1 - t_c, q_1 + t_c, (p_1 + p_0 + q_0 + 2 * q_1 + q_2 + 2 * q_3 + 4) >> 3) \quad (1436)$$

$$q_2' = \text{Clip3}(q_2 - t_c, q_2 + t_c, (p_0 + q_0 + q_1 + 2 * q_2 + 3 * q_3 + 4) >> 3) \quad (1437)$$

- Ngược lại, áp dụng bộ lọc yếu sau:

$$\Delta = \text{Clip3}(-t_c, t_c, (((q_0 - p_0) << 2) + p_1 - q_1 + 4) >> 3)) \quad (1438)$$

$$p_0' = \text{Clip1}(p_0 + \Delta) \quad (1439)$$

$$q_0' = \text{Clip1}(q_0 - \Delta) \quad (1440)$$

Khi `pred_mode_plt_flag` của đơn vị mã bao gồm khối mã chứa mẫu p_i bằng 1, giá trị mẫu được lọc, p_i' được thay thế bằng giá trị mẫu đầu vào tương ứng p_i với $i = 0..\text{maxFilterLengthP} - 1$.

Khi `pred_mode_plt_flag` của đơn vị mã bao gồm khối mã chứa mẫu q_i bằng 1, giá trị mẫu được lọc, q_i' được thay thế bằng giá trị mẫu đầu vào tương ứng q_i với $i = 0..\text{maxFilterLengthQ} - 1$.

Hiện tại, báo hiệu các tham số điều khiển tách khối Cb và Cr được thực hiện thậm chí nếu định dạng màu của chuỗi được mã hoá là 4:0:0, chẳng hạn, khi giá trị của `ChromaArrayType` bằng 0. Khi chuỗi đầu vào không có thành phần màu, không có yêu cầu khác thực hiện tách khối cho các thành phần màu.

Trong một số trường hợp, các tham số điều khiển tách khối có thể giống như giữa các thành phần độ sáng lẫn sắc độ trong các trường hợp cụ thể.

Đối với các khối được tạo mã nhờ sử dụng cơ chế Cb-Cr chung, có thể thể hiện các đặc tính sai số lượng tử khác nhau và do vậy có thể hưởng lợi từ việc có các tham số điều khiển tách khối riêng rẽ (các độ lệch Beta và Tc) được báo hiệu.

Phương án thực hiện 1

Theo phương án thực hiện, các độ lệch Beta và Tc tách khối Cb và Cr (để đơn giản, gọi là các tham số điều khiển tách khối theo sáng chế) (chỉ) được báo hiệu khi giá trị của `ChromaArrayType` không bằng 0.

Cú pháp theo phương án thực hiện được thể hiện như sau:

Cú pháp PPS:

.....	
<code>pps_chroma_tool_offsets_present_flag</code>	<code>u(1)</code>
<code>if(pps_chroma_tool_offsets_present_flag) {</code>	
<code>pps_cb_qp_offset</code>	<code>se(v)</code>

pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset_present_flag	u(1)
if(pps_joint_cbr_qp_offset_present_flag)	
pps_joint_cbr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
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)
if(pps_joint_cbr_qp_offset_present_flag)	
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)
if(ChromaArrayType != 0) {	
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
}	
.....	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)

.....	
-------	--

Hoặc

Cú pháp PH:

if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
if(ChromaArrayType != 0) {	
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	
}	
}	

Hoặc

Cú pháp tiêu đề lát:

if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_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(ChromaArrayType != 0) {	
slice_cb_beta_offset_div2	se(v)

<code>slice_cb_tc_offset_div2</code>	<code>se(v)</code>
<code>slice_cr_beta_offset_div2</code>	<code>se(v)</code>
<code>slice_cr_tc_offset_div2</code>	<code>se(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	

Trong một số ví dụ, giá trị của `ChromaArrayType` có thể không được nhận ở mức cú pháp PPS, do `ChromaArrayType` được báo hiệu trong mức tập hợp tham số chuỗi (Sequence Parameter set level, SPS). Báo hiệu có điều kiện dựa trên `ChromaArrayType` trong PPS có thể tạo ra sự lệ thuộc phân tách giữa SPS và PPS. Do vậy, một giải pháp tùy chọn là di chuyển các tham số điều khiển tách khối hiện có từ mức PPS sang mức SPS. Theo cách này, tránh được sự lệ thuộc phân tách giữa SPS và PPS.

Theo cách khác, các tham số điều khiển tách khối có thể được báo hiệu có điều kiện dựa trên phần tử cú pháp đã có gọi là **`pps_chroma_tool_offsets_present_flag`**.

Cú pháp cải biến của PPS được thể hiện dưới đây:

Cú pháp tập hợp tham số ảnh (Picture parameter set, PPS):

<code>.....</code>	
<code>pps_chroma_tool_offsets_present_flag</code>	<code>u(1)</code>
<code>if(pps_chroma_tool_offsets_present_flag) {</code>	
<code> pps_cb_qp_offset</code>	<code>se(v)</code>
<code> pps_cr_qp_offset</code>	<code>se(v)</code>
<code> pps_joint_cbr_qp_offset_present_flag</code>	<code>u(1)</code>
<code> if(pps_joint_cbr_qp_offset_present_flag)</code>	
<code> pps_joint_cbr_qp_offset_value</code>	<code>se(v)</code>
<code> pps_slice_chroma_qp_offsets_present_flag</code>	<code>u(1)</code>
<code> pps_cu_chroma_qp_offset_list_enabled_flag</code>	<code>u(1)</code>
<code>}</code>	
<code>if(pps_cu_chroma_qp_offset_list_enabled_flag) {</code>	

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)
if(pps_joint_cbr_qp_offset_present_flag)	
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)
if(pps_chroma_tool_offsets_present_flag) {	
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
}	
.....	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
.....	

Giải pháp nêu trên có ưu điểm là không tạo lệ thuộc phân tách bất kỳ giữa SPS và PPS, và báo hiệu các tham số điều khiển tách khối cho các thành phần Cb và Cr (chỉ) khi giá trị của ChromaArrayType không bằng 0.

Giải pháp tùy chọn khác sẽ không báo hiệu có điều kiện tham số điều khiển tách khối ở mức PPS (dựa trên ChromaArrayType), mà báo hiệu chúng có điều kiện ở mức PH và SH (dựa trên ChromaArrayType).

Theo phương án thực hiện, định nghĩa về các cú pháp này trong bảng có thể đề cập đến các phần mô tả nêu trên.

Phương án thực hiện 2

Theo phương án thực hiện, phần tử cú pháp mới được báo hiệu. Cú pháp này được sử dụng để chỉ báo liệu các tham số điều khiển tách khối độ sáng và sắc độ có giống nhau hay không. Khi các giá trị của các tham số điều khiển tách khối độ sáng và sắc độ là khác nhau, các tham số điều khiển tách khối Cb và Cr còn được báo hiệu. Phương án thực hiện loại bỏ báo hiệu dư khi các tham số tách khối giống nhau giữa các thành phần độ sáng và sắc độ.

Cú pháp và ngữ nghĩa theo phương án thực hiện được thể hiện dưới đây:

Cú pháp PPS:

.....	
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)
pps_chroma_offsets_same_as_luma	u(1)
if(!pps_chroma_offsets_same_as_luma) {	
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	

}	
}	
.....	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
.....	

Hoặc cú pháp PH:

if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_chroma_offsets_same_as_luma	u(1)
if(!ph_chroma_offsets_same_as_luma) {	
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	
}	
}	

Hoặc cú pháp SH:

if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)

<code>slice_tc_offset_div2</code>	<code>se(v)</code>
<code>slice_chroma_offsets_same_as_luma</code>	<code>u(1)</code>
<code>if(!slice_chroma_offsets_same_as_luma) {</code>	
<code>slice_cb_beta_offset_div2</code>	<code>se(v)</code>
<code>slice_cb_tc_offset_div2</code>	<code>se(v)</code>
<code>slice_cr_beta_offset_div2</code>	<code>se(v)</code>
<code>slice_cr_tc_offset_div2</code>	<code>se(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	

Ngữ nghĩa của các phần tử cú pháp mới như sau:

`slice_chroma_offsets_same_as_luma` bằng 0 xác định rằng các phần tử cú pháp `slice_cb_beta_offset_div2`, `slice_cb_tc_offset_div2`, `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` còn được báo hiệu trong SH. `slice_chroma_offsets_same_as_luma` bằng 1 xác định rằng các giá trị của syntax elements `slice_cb_beta_offset_div2`, `slice_cb_tc_offset_div2`, `slice_cr_beta_offset_div2` và `slice_cr_tc_offset_div2` còn không được báo hiệu và được suy ra lần lượt giống như `slice_beta_offset_div2` và `slice_tc_offset_div2`.

`ph_chroma_offsets_same_as_luma` bằng 0 xác định rằng các phần tử cú pháp `ph_cb_beta_offset_div2`, `ph_cb_tc_offset_div2`, `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` còn được báo hiệu trong tiêu đề ảnh. `ph_chroma_offsets_same_as_luma` bằng 1 xác định rằng các giá trị của các phần tử cú pháp `ph_cb_beta_offset_div2`, `ph_cb_tc_offset_div2`, `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` không được báo hiệu còn được suy ra lần lượt giống như `ph_beta_offset_div2` và `ph_tc_offset_div2`.

`pps_chroma_offsets_same_as_luma` bằng 0 xác định rằng các phần tử cú pháp `pps_cb_beta_offset_div2`, `pps_cb_tc_offset_div2`,

pps_cr_beta_offset_div2 và pps_cr_tc_offset_div2 còn được báo hiệu trong PPS. pps_chroma_offsets_same_as_luma bằng 1 xác định rằng các giá trị của các phần tử cú pháp pps_cb_beta_offset_div2, pps_cb_tc_offset_div2, pps_cr_beta_offset_div2 và pps_cr_tc_offset_div2 không được báo hiệu và còn được suy ra lần lượt giống như pps_beta_offset_div2 và pps_tc_offset_div2.

Định nghĩa của các cú pháp khác trong bảng có thể đề cập đến các phần mô tả nêu trên.

Phương án thực hiện 3

Theo phương án thực hiện, các tham số độ lệch beta và tc riêng rẽ được nêu cho các khối được tạo mã Cb-Cr đồng thời.

Cú pháp như sau:

Cú pháp PPS:

.....	
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)
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
pps_cbr_beta_offset_div2	se(v)
pps_cbr_tc_offset_div2	se(v)
}	
}	
.....	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)

.....	
-------	--

Cú pháp PH:

if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
ph_cbr_beta_offset_div2	se(v)
ph_cbr_tc_offset_div2	se(v)
}	
}	
}	

Cú pháp SH:

if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_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)
slice_cb_beta_offset_div2	se(v)
slice_cb_tc_offset_div2	se(v)
slice_cr_beta_offset_div2	se(v)
slice_cr_tc_offset_div2	se(v)
slice_cbr_beta_offset_div2	se(v)

<code>slice_cbr_tc_offset_div2</code>	<code>se(v)</code>
<code>}</code>	
<code>}</code>	

Ngữ nghĩa của các phần tử cú pháp mới như sau:

`pps_cbr_beta_offset_div2` và `pps_cbr_tc_offset_div2` xác định các độ lệch tham số tách khối mặc định cho β và tC (chia cho 2) được áp dụng cho thành phần Cb-Cr chung cho các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khối mặc định được ghi đè bởi các độ lệch tham số tách khối xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của lats viện dẫn đến PPS. Các giá trị của `pps_cbr_beta_offset_div2` và `pps_cbr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_cbr_beta_offset_div2` và `pps_cbr_tc_offset_div2` đều được suy ra bằng 0.

`ph_cbr_beta_offset_div2` và `ph_cbr_tc_offset_div2` xác định các độ lệch tham số tách khối cho β và tC (chia cho 2) được áp dụng cho thành phần Cb-Cr chung cho các lát được liên kết với PH. Các giá trị của `ph_cbr_beta_offset_div2` và `ph_cbr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `ph_cbr_beta_offset_div2` và `ph_cbr_tc_offset_div2` được suy ra bằng `pps_cbr_beta_offset_div2` và `pps_cbr_tc_offset_div2`,

`slice_cbr_beta_offset_div2` và `slice_cbr_tc_offset_div2` xác định các độ lệch tham số tách khối cho β và tC (chia cho 2) được áp dụng cho thành phần Cb-Cr chung cho lát hiện tại. Các giá trị của `slice_cbr_beta_offset_div2` và `slice_cbr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của

slice_cbr_beta_offset_div2 và slice_cbr_tc_offset_div2 lần lượt được suy ra bằng ph_cbr_beta_offset_div2 và ph_cbr_tc_offset_div2.

Các thay đổi được yêu cầu khi dẫn xuất giá trị QP để tách khỏi sắc độ như sau:

Giá trị của biến β' được xác định như được xác định trong Bảng 43 dựa trên tham số lượng tử Q được dẫn xuất như sau:

$$\begin{aligned} \text{sliceBetaOffsetDiv2} &= ((\text{TuCResMode}[\text{xTb}_Q][\text{yTb}_Q] == 2) \parallel \\ &(\text{TuCResMode}[\text{xTb}_p][\text{yTb}_p] == 2) ? \text{slice_cbr_beta_offset_div2} : (\text{cIdx} == 1 ? \\ &\text{slice_cb_beta_offset_div2} : \text{slice_cr_beta_offset_div2})) \\ Q &= \text{Clip3}(0, 63, Q_p_C + (\text{sliceBetaOffsetDiv2} \ll 1)) \quad (1355) \end{aligned}$$

trong đó slice_cb_beta_offset_div2, slice_cr_beta_offset_div2 và slice_cbr_beta_offset_div2 là các giá trị của các phần tử cú pháp slice_cb_beta_offset_div2, slice_cr_beta_offset_div2 và slice_cbr_beta_offset_div2, lần lượt đối với lát chứa mẫu $q_{0,0}$

Giá trị của biến t_c' được xác định như được xác định trong Bảng 43 dựa trên tham số lượng tử sắc độ Q được dẫn xuất như sau:

$$\begin{aligned} \text{sliceTcOffsetDiv2} &= ((\text{TuCResMode}[\text{xTb}_Q][\text{yTb}_Q] \parallel \\ &\text{TuCResMode}[\text{xTb}_p][\text{yTb}_p] == 2) ? \text{slice_cbr_tc_offset_div2} : (\text{cIdx} == 1 ? \\ &\text{slice_cb_tc_offset_div2} : \text{slice_cr_tc_offset_div2})) \\ Q &= \text{Clip3}(0, 65, Q_p_C + 2 * (bS - 1) + (\text{sliceTcOffsetDiv2} \ll 1)) \quad (1357) \end{aligned}$$

trong đó slice_cb_tc_offset_div2, slice_cr_beta_offset_div2 và slice_cr_tc_offset_div2 lần lượt là các giá trị của các phần tử cú pháp slice_cb_tc_offset_div2, slice_cr_tc_offset_div2 và slice_cbr_tc_offset_div2, cho lát chứa mẫu $q_{0,0}$.

Định nghĩa về các cú pháp khác trong bảng có thể đề cập đến các phần mô tả nêu trên.

Theo phương án thực hiện như được thể hiện trên Fig.8, phương pháp tạo mã được triển khai bởi thiết bị giải mã bao gồm các bước:

S801: thu được dòng bit.

Dòng bit có thể thu được theo mạng không dây hoặc mạng hữu tuyến. Dòng bit có thể được truyền từ website, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn, đường thuê bao số (digital subscriber line, DSL), hoặc công nghệ không dây chẳng hạn, hồng ngoại, vô tuyến, vi sóng, WiFi, Bluetooth, LTE hoặc 5G.

Theo phương án thực hiện, dòng bit là chuỗi bit, ở dạng dòng đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) hoặc dòng byte, tạo biểu diễn của chuỗi đơn vị truy nhập (access unit, AU) tạo một hoặc nhiều chuỗi video được tạo mã (coded video sequence, CVS).

Trong một số phương án thực hiện, đối với quá trình giải mã, phía bộ giải mã đọc dòng bit và dẫn xuất các ảnh được giải mã từ dòng bit; đối với quá trình mã hoá, phía bộ mã hoá tạo dòng bit.

Thông thường, dòng bit sẽ bao gồm các phần tử cú pháp được tạo bởi cấu trúc cú pháp. phần tử cú pháp: Phần tử dữ liệu được biểu diễn trong dòng bit.

cấu trúc cú pháp: 0 hoặc nhiều phần tử cú pháp đồng thời xuất hiện trong dòng bit theo thứ tự cụ thể.

Trong ví dụ cụ thể, các định dạng dòng bit xác định mối quan hệ giữa dòng đơn vị NAL và dòng byte, mỗi dòng trong đó được gọi là dòng bit.

Dòng bit có thể ở một trong hai định dạng: định dạng dòng đơn vị NAL hoặc định dạng dòng byte. Định dạng dòng đơn vị NAL về cơ bản là loại “cơ bản” hơn. Định dạng dòng đơn vị NAL bao gồm chuỗi cấu trúc cú pháp được gọi là đơn vị NAL. Chuỗi này được sắp thứ tự theo thứ tự giải mã. Có các ràng buộc

đặt dựa trên thứ tự giải mã (và các nội dung) của các đơn vị NAL trong dòng đơn vị NAL.

Định dạng dòng byte có thể được tạo từ định dạng dòng đơn vị NAL bằng cách sắp xếp thứ tự các đơn vị NAL theo thứ tự giải mã và tạo tiền tố mỗi đơn vị NAL có tiền tố mã bắt đầu và 0 hoặc nhiều byte bằng 0 để tạo thành dòng byte. Định dạng dòng đơn vị NAL có thể được trích xuất từ định dạng dòng byte bằng cách tìm kiếm vị trí của mẫu hình tiền tố mã bắt đầu duy nhất trong dòng byte này.

Mệnh đề này xác định mối quan hệ giữa các ảnh nguồn và được giải mã được tạo qua dòng bit.

Nguồn video được biểu diễn bởi dòng bit là chuỗi ảnh theo thứ tự giải mã.

Các ảnh nguồn và được giải mã mà mỗi ảnh được bao gồm một hoặc nhiều mảng mẫu:

- Chỉ độ sáng (Y) (đơn sắc).
- Độ sáng và hai sắc độ (YCbCr hoặc YCgCo).
- Lục, lam, và đỏ (GBR, cũng được biết là RGB).
- Mảng biểu diễn các mẫu màu đơn sắc hoặc ba màu cơ bản (chẳng hạn, YZX, cũng được biết đến là XYZ).

Các biến và các số hạng được liên kết với các mảng này được gọi là độ sáng (hoặc L hoặc Y) và sắc độ, trong đó hai mảng sắc độ được gọi là Cb và Cr; bất kể phương pháp biểu diễn màu thực đang sử dụng. Phương pháp biểu diễn màu thực đang sử dụng có thể được chỉ báo trong cú pháp được xác định trong các tham số VUI như được xác định trong ITU-T H.264 | ISO/IEC 23002-7.

S802: thu được giá trị của phần tử cú pháp từ dòng bit.

Theo phương án thực hiện, giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khỏi đối với thành phần sắc độ của lát của ảnh được tạo mã.

Chẳng hạn, giá trị của phần tử cú pháp chỉ báo liệu các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ xuất hiện trong cú pháp tải tin chuỗi byte thô (raw byte sequence payload, RBSP) của tập hợp tham số ảnh (picture parameter set, PPS) hoặc không.

Trong ví dụ, cú pháp được biểu diễn theo `pps_chroma_tool_offsets_present_flag`. `pps_chroma_tool_offsets_present_flag` bằng 1 xác định rằng các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ xuất hiện trong cấu trúc cú pháp PPS RBSP và các phần tử cú pháp độ lệch α và β tách khối sắc độ có thể xuất hiện trong các cấu trúc cú pháp PH hoặc các SH của các ảnh viện dẫn đến PPS. `pps_chroma_tool_offsets_present_flag` bằng 0 xác định rằng các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ không xuất hiện trong cấu trúc cú pháp PPS RBSP và các phần tử cú pháp độ lệch α và β tách khối sắc độ không xuất hiện trong các cấu trúc cú pháp PH hoặc các SH của các ảnh viện dẫn đến PPS. Khi `sps_chroma_format_idc` bằng 0, giá trị của `pps_chroma_tool_offsets_present_flag` phải bằng 0.

Trong ví dụ, giá trị của phần tử cú pháp thu được trong PPS.

Trong ví dụ, khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp bằng 0.

Trong ví dụ, giá trị của cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối cho thành phần độ sáng của khối mã giống như tham số điều khiển tách khối cho thành phần sắc độ của khối.

S803: thu được giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát từ dòng bit, khi giá trị của phần tử cú pháp bằng giá trị định trước.

Giá trị định trước là giá trị số nguyên. Trong ví dụ, giá trị định trước không bằng 0. Trong ví dụ, giá trị định trước bằng 1.

Trong ví dụ, giá trị của tham số điều khiển tách khối cho thành phần sắc độ của khối mã thu được trong PPS.

Trong ví dụ, giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thu được trong PH.

Trong ví dụ, giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thu được trong SH.

Trong ví dụ, tham số điều khiển tách khối đối với thành phần sắc độ của lát được báo hiệu khi chuỗi video có thành phần màu.

Trong ví dụ, ở mức PPS, tham số điều khiển tách khối đối với thành phần sắc độ của lát được biểu diễn theo `pps_cb_beta_offset_div2`, `pps_cb_tc_offset_div2`, `pps_cr_beta_offset_div2` hoặc `pps_cr_tc_offset_div2`.

Có thể hiểu rằng, theo một số cách thực hiện, chỉ có một tham số điều khiển tách khối đối với thành phần sắc độ hoặc kết hợp bất kỳ của các tham số điều khiển tách khối này. Chẳng hạn, tất cả bốn tham số điều khiển tách khối này được báo hiệu có điều kiện theo giá trị của `pps_chroma_tool_offsets_present_flag`.

<code>if(pps_chroma_tool_offsets_present_flag) {</code>	
<code>pps_cb_beta_offset_div2</code>	<code>se(v)</code>
<code>pps_cb_tc_offset_div2</code>	<code>se(v)</code>
<code>pps_cr_beta_offset_div2</code>	<code>se(v)</code>
<code>pps_cr_tc_offset_div2</code>	<code>se(v)</code>
<code>}</code>	

`pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` xác định các độ lệch tham số tách khối mặc định với β và tC (chia cho 2) được áp dụng cho thành phần Cb đối với các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khối mặc định được ghi đè bởi các độ lệch tham số tách khối xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của lát viện dẫn đến PPS. Các giá trị của

`pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2` lần lượt được suy ra bằng `pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2`.

`pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` xác định các độ lệch tham số tách khỏi mặc định cho β và tC (chia cho 2) được áp dụng cho thành phần Cr cho các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khỏi mặc định được ghi đè bởi các độ lệch tham số tách khỏi xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của lát viện dẫn đến PPS. Các giá trị của `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2` lần lượt được suy ra bằng `pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2`.

`pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2` xác định các độ lệch tham số tách khỏi mặc định cho β và tC (chia cho 2) được áp dụng cho thành phần độ sáng cho các lát viện dẫn đến PPS, trừ khi các độ lệch tham số tách khỏi mặc định được ghi đè bởi các độ lệch tham số tách khỏi xuất hiện trong các tiêu đề ảnh hoặc các tiêu đề lát của các lát viện dẫn đến PPS. Các giá trị của `pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12 , bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2` đều được suy ra bằng 0.

Trong ví dụ, ở mức PH, tham số điều khiển tách khỏi đối với thành phần sắc độ của lát được biểu diễn theo `ph_cb_beta_offset_div2`, `ph_cb_tc_offset_div2`, `ph_cr_beta_offset_div2` hoặc `ph_cr_tc_offset_div2`. Có thể hiểu rằng, theo một số cách thực hiện, chỉ có một tham số điều khiển tách khỏi đối với thành phần

sắc độ hoặc tổ hợp bất kỳ của các tham số điều khiển tách khối này. Chẳng hạn, tất cả bốn tham số điều khiển tách khối này được báo hiệu có điều kiện theo giá trị của `pps_chroma_tool_offsets_present_flag`.

<code>if(pps_chroma_tool_offsets_present_flag) {</code>	
<code> ph_cb_beta_offset_div2</code>	<code>se(v)</code>
<code> ph_cb_tc_offset_div2</code>	<code>se(v)</code>
<code> ph_cr_beta_offset_div2</code>	<code>se(v)</code>
<code> ph_cr_tc_offset_div2</code>	<code>se(v)</code>
<code>}</code>	

`ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần Cb cho các lát trong ảnh hiện tại. Các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút.

Khi không xuất hiện, các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` được suy ra như sau:

Nếu `pps_chroma_tool_offsets_present_flag` bằng 1, các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` lần lượt được suy ra bằng `pps_cb_beta_offset_div2` và `pps_cb_tc_offset_div2`.

Ngược lại (`pps_chroma_tool_offsets_present_flag` bằng 0), các giá trị của `ph_cb_beta_offset_div2` và `ph_cb_tc_offset_div2` lần lượt được suy ra bằng `ph_luma_beta_offset_div2` và `ph_luma_tc_offset_div2`.

`ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần Cr cho các lát trong ảnh hiện tại. Các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút.

Khi không xuất hiện, các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` được suy ra như sau:

Nếu `pps_chroma_tool_offsets_present_flag` bằng 1, các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` lần lượt được suy ra bằng `pps_cr_beta_offset_div2` và `pps_cr_tc_offset_div2`.

Ngược lại (`pps_chroma_tool_offsets_present_flag` bằng 0), các giá trị của `ph_cr_beta_offset_div2` và `ph_cr_tc_offset_div2` lần lượt được suy ra bằng `ph_luma_beta_offset_div2` và `ph_luma_tc_offset_div2`.

`ph_luma_beta_offset_div2` và `ph_luma_tc_offset_div2` xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần độ sáng cho các lát trong ảnh hiện tại. Các giá trị của `ph_luma_beta_offset_div2` và `ph_luma_tc_offset_div2` phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của `ph_luma_beta_offset_div2` và `ph_luma_tc_offset_div2` lần lượt được suy ra bằng `pps_luma_beta_offset_div2` và `pps_luma_tc_offset_div2`.

Trong ví dụ, ở mức SH, tham số điều khiển tách khối đối với thành phần sắc độ của lát được biểu diễn theo `sh_cb_beta_offset_div2`, `sh_cb_tc_offset_div2`, `sh_cr_beta_offset_div2` hoặc `sh_cr_tc_offset_div2`.

Có thể hiểu rằng, theo một số phương án thực hiện, chỉ có một tham số điều khiển tách khối đối với thành phần sắc độ hoặc kết hợp bất kỳ của các tham số điều khiển tách khối này. Chẳng hạn, tất cả bốn tham số điều khiển tách khối này được báo hiệu có điều kiện theo giá trị của `pps_chroma_tool_offsets_present_flag`.

<code>if(pps_chroma_tool_offsets_present_flag) {</code>	
<code>sh_cb_beta_offset_div2</code>	<code>se(v)</code>
<code>sh_cb_tc_offset_div2</code>	<code>se(v)</code>
<code>sh_cr_beta_offset_div2</code>	<code>se(v)</code>

sh_cr_tc_offset_div2	se(v)
}	

sh_cb_beta_offset_div2 và sh_cb_tc_offset_div2 xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần Cb cho lát hiện tại. Các giá trị của sh_cb_beta_offset_div2 và sh_cb_tc_offset_div2 phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút.

Khi không xuất hiện, các giá trị của sh_cb_beta_offset_div2 và sh_cb_tc_offset_div2 được suy ra như sau:

Nếu pps_chroma_tool_offsets_present_flag bằng 1, các giá trị của sh_cb_beta_offset_div2 và sh_cb_tc_offset_div2 lần lượt được suy ra bằng ph_cb_beta_offset_div2 và ph_cb_tc_offset_div2.

Ngược lại (pps_chroma_tool_offsets_present_flag bằng 0), các giá trị của sh_cb_beta_offset_div2 và sh_cb_tc_offset_div2 lần lượt được suy ra bằng sh_luma_beta_offset_div2 và sh_luma_tc_offset_div2.

sh_cr_beta_offset_div2 và sh_cr_tc_offset_div2 xác định các độ lệch tham số tách khối với β và tC (chia cho 2) được áp dụng cho thành phần Cr cho lát hiện tại. Các giá trị của sh_cr_beta_offset_div2 và sh_cr_tc_offset_div2 phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút.

Khi không xuất hiện, các giá trị của sh_cr_beta_offset_div2 và sh_cr_tc_offset_div2 được suy ra như sau:

Nếu pps_chroma_tool_offsets_present_flag bằng 1, các giá trị của sh_cr_beta_offset_div2 và sh_cr_tc_offset_div2 lần lượt được suy ra bằng ph_cr_beta_offset_div2 và ph_cr_tc_offset_div2.

Ngược lại (pps_chroma_tool_offsets_present_flag bằng 0), các giá trị của sh_cr_beta_offset_div2 và sh_cr_tc_offset_div2 lần lượt được suy ra bằng sh_luma_beta_offset_div2 và sh_luma_tc_offset_div2.

sh_luma_beta_offset_div2 và sh_luma_tc_offset_div2 xác định các độ lệch tham số tách khối với β và tc (chia cho 2) được áp dụng cho thành phần độ sáng cho lát hiện tại. Các giá trị của sh_luma_beta_offset_div2 và sh_luma_tc_offset_div2 phải đều nằm trong khoảng từ -12 đến 12, bao gồm hai giá trị đầu mút. Khi không xuất hiện, các giá trị của sh_luma_beta_offset_div2 và sh_luma_tc_offset_div2 lần lượt được suy ra bằng ph_luma_beta_offset_div2 và ph_luma_tc_offset_div2.

Theo phương án thực hiện, phương pháp còn bao gồm bước: gán giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát bằng giá trị của tham số điều khiển tách khối đối với thành phần độ sáng của lát, khi giá trị của phần tử cú pháp không bằng giá trị định trước.

S804: thực hiện quá trình tách khối trên khối trong lát theo giá trị của tham số điều khiển tách khối.

Nói chung, đối với quá trình lọc tách khối, các đầu vào cho quá trình lọc tách khối là ảnh được tái tạo trước khi tách khối, chẳng hạn, mảng recPictureL và, khi sps_chroma_format_idc không bằng 0, các mảng recPictureCb và recPictureCr.

Các đầu ra của quá trình này là ảnh được tái tạo được chỉnh sửa sau khi tách khối, mảng recPictureL và, khi sps_chroma_format_idc không bằng 0, các mảng recPictureCb và recPictureCr.

Các cạnh dọc trong ảnh được lọc trước. Sau đó, các cạnh ngang trong ảnh được lọc với các mẫu được chỉnh sửa bằng quá trình lọc cạnh dọc làm đầu vào. Các cạnh dọc và ngang trong các CTB của mỗi CTU được xử lý riêng rẽ dựa vào đơn vị mã. Các cạnh dọc của các khối mã trong đơn vị mã được lọc bắt đầu với cạnh ở phía tay trái của các khối mã đi qua các cạnh về phía tay phải của các khối mã theo thứ tự hình học. Các cạnh ngang của các khối mã trong đơn vị mã

được lọc bắt đầu với cạnh phía trên của các khối mã đi qua các cạnh phía dưới của các khối mã theo thứ tự hình học.

Chi tiết về quá trình tách khối có thể viện dẫn đến mô tả trên đây.

Theo phương án thực hiện như được thể hiện trên Fig.11, thiết bị giải mã video 900 được bộc lộ, thiết bị 900 bao gồm:

môđun nhận 901, được tạo cấu hình để thu được dòng bit;

môđun phân tách 902 được tạo cấu hình để thu được giá trị của phần tử cú pháp từ dòng bit, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã, (chẳng hạn, giá trị của phần tử cú pháp chỉ báo liệu các phần tử cú pháp liên quan đến các độ lệch công cụ sắc độ xuất hiện trong cấu trúc RBSP PPS hay không); môđun phân tách 902 được tạo cấu hình để thu được giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát từ dòng bit, khi giá trị của phần tử cú pháp bằng giá trị định trước, giá trị định trước là giá trị số nguyên.

Dòng bit có thể thu được theo mạng không dây hoặc mạng hữu tuyến. Dòng bit có thể được truyền từ website, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn, DSL, hoặc công nghệ không dây, chẳng hạn, hồng ngoại, vô tuyến, vi sóng, WiFi, Bluetooth, LTE hoặc 5G.

Theo phương án thực hiện, dòng bit là chuỗi bit, ở dạng dòng đơn vị NAL hoặc dòng byte, tạo biểu diễn của chuỗi AU tạo một hoặc nhiều CVS.

Trong một số phương án thực hiện, đối với quá trình giải mã, phía bộ giải mã đọc dòng bit và dẫn xuất các ảnh được giải mã từ dòng bit; đối với quá trình mã hoá, phía bộ mã hoá tạo dòng bit.

Thông thường, dòng bit sẽ bao gồm các phần tử cú pháp được tạo bởi cấu trúc cú pháp. phần tử cú pháp: Phần tử dữ liệu được biểu diễn trong dòng bit.

cấu trúc cú pháp: 0 hoặc nhiều phần tử cú pháp đồng thời xuất hiện trong dòng bit theo thứ tự cụ thể.

Trong ví dụ cụ thể, các định dạng dòng bit xác định mối quan hệ giữa dòng đơn vị NAL và dòng byte, trong số đó được gọi là dòng bit.

Dòng bit có thể ở một trong hai định dạng: định dạng dòng đơn vị NAL hoặc định dạng dòng byte. Định dạng dòng đơn vị NAL về mặt khái niệm là loại “cơ bản” hơn. Định dạng dòng đơn vị NAL bao gồm chuỗi cấu trúc cú pháp gọi là các đơn vị NAL. Chuỗi này được sắp theo thứ tự giải mã. Có các ràng buộc bắt buộc theo thứ tự giải mã (và nội dung) của các đơn vị NAL trong dòng đơn vị NAL.

Định dạng dòng byte có thể được tạo từ định dạng dòng đơn vị NAL bằng cách sắp thứ tự các đơn vị NAL theo thứ tự giải mã và tạo tiền tố mỗi đơn vị NAL có tiền tố mã khởi động và 0 hoặc nhiều byte bằng 0 để tạo dòng byte. Định dạng dòng đơn vị NAL có thể được trích xuất từ định dạng dòng byte bằng cách tìm kiếm vị trí của mẫu hình tiền tố mã khởi động duy nhất trong dòng byte này.

Mệnh đề này xác định mối quan hệ giữa các ảnh nguồn và được giải mã được tạo qua dòng bit.

Nguồn video được biểu diễn bởi dòng bit là chuỗi ảnh theo thứ tự giải mã.

Các ảnh nguồn và được giải mã mà mỗi ảnh bao gồm một hoặc nhiều mảng mẫu:

- Chỉ độ sáng (Y) (đơn sắc).
- Độ sáng và hai sắc độ (YCbCr hoặc YCgCo).
- Lục, lam, và đỏ (GBR, cũng được biết đến là RGB).
- Các mảng biểu diễn các mẫu đơn sắc và ba màu cơ bản chưa xác định khác (chẳng hạn, YZX, cũng được biết đến là XYZ).

Các biến và các số hạng được liên kết với các mảng này được gọi là độ sáng (hoặc L hoặc Y) và sắc độ, trong đó hai mảng sắc độ được gọi là Cb và Cr; bất kể phương pháp biểu diễn màu thực đang sử dụng. Phương pháp biểu diễn màu thực đang sử dụng có thể được chỉ báo trong cú pháp được xác định trong các tham số VUI như được xác định trong ITU-T H.265 | ISO/IEC 23002-7.

Các chi tiết khác đối với mô đun nhận 901 và mô đun phân tách 902 có thể viện dẫn đến các ví dụ và các cách thực hiện phương pháp trên.

Ví dụ 1. Phương pháp tạo mã được triển khai bởi thiết bị giải mã, phương pháp bao gồm các bước:

- thu được dòng bit đối với khối mã;

- thu được giá trị của cú pháp từ dòng bit;

- thu được giá trị của tham số điều khiển tách khối từ dòng bit, khi giá trị của cú pháp bằng giá trị định trước (Trong ví dụ, giá trị định trước không bằng 0).

Ví dụ 2. Phương pháp theo ví dụ 1, trong đó giá trị của cú pháp thu được ở mức tập hợp tham số chuỗi (sequence parameter set, SPS).

Ví dụ 3. Phương pháp theo ví dụ 1, trong đó giá trị của cú pháp thu được theo PPS.

Ví dụ 4. Phương pháp theo ví dụ 1, trong đó giá trị của cú pháp thu được theo PH.

Ví dụ 5. Phương pháp theo ví dụ 1, trong đó giá trị của cú pháp thu được theo SH.

Ví dụ 6. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 5, trong đó giá trị của tham số điều khiển tách khối thu được theo PPS.

Ví dụ 7. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 5, trong đó giá trị của tham số điều khiển tách khối thu được theo PH.

Ví dụ 8. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 5, trong đó giá trị của tham số điều khiển tách khối thu được theo SH.

Ví dụ 9. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 5, trong đó giá trị của tham số điều khiển tách khối thu được ở mức SPS.

Ví dụ 10. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 9, trong đó giá trị của cú pháp được sử dụng để chỉ báo không có thành phần màu trong chuỗi video.

Ví dụ 11. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 10, trong đó tham số điều khiển tách khối được báo hiệu chỉ khi chuỗi video có thành phần màu.

Ví dụ 12. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 9, trong đó giá trị của cú pháp được sử dụng để chỉ báo liệu tham số điều khiển tách khối cho thành phần độ sáng của khối giống như tham số điều khiển tách khối cho thành phần sắc độ của khối hay không.

Ví dụ 13. Phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 12, trong đó giá trị của tham số điều khiển tách khối là độ lệch tham số tách khối định trước được áp dụng cho thành phần Cb-Cr chung của khối.

Ví dụ 14. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ từ 1 đến 13.

Ví dụ 15. Sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ trước đó khi được thực thi trên máy tính hoặc bộ xử lý.

Ví dụ 16. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

vật lưu trữ máy tính đọc được bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bằng các bộ xử lý, trong đó chương trình, khi được

thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ trước đó.

Ví dụ 17. Vật ghi máy tính đọc được bất biến mang mã chương trình mà, khi được thực thi bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo ví dụ bất kỳ trong các ví dụ trước đó.

Dưới đây là giải thích việc áp dụng phương pháp mã hóa cũng như phương pháp giải mã như được thể hiện theo các phương án thực hiện nêu trên, và hệ thống sử dụng chúng.

Fig.6 là sơ đồ khối thể hiện hệ thống cung cấp nội dung 3100 để thực hiện dịch vụ phân tán nội dung. Hệ thống cung cấp nội dung 3100 bao gồm thiết bị chụp 3102, thiết bị đầu cuối 3106, và một cách tùy chọn bao gồm bộ phận hiển thị 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 trên liên kết truyền thông 3104. Liên kết truyền thông có thể bao gồm kênh truyền thông 13 nêu trên. Liên kết truyền thông 3104 bao gồm nhưng không bị giới hạn ở WiFi, Ethernet, cáp, không dây (3G/4G/5G), USB, hoặc loại tổ hợp bất kỳ của nó, hoặc tương tự.

Thiết bị chụp 3102 tạo dữ liệu, và có thể mã hóa dữ liệu bằng phương pháp mã hóa như được thể hiện theo các phương án thực hiện nêu trên. Theo cách khác, thiết bị chụp 3102 có thể phân tán dữ liệu đến máy chủ streaming (không được thể hiện trên các hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa đến thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm nhưng không bị giới hạn ở camera, điện thoại thông minh hoặc Pad, máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, thiết bị lắp trong xe, hoặc kết hợp của chúng, hoặc tương tự. Chẳng hạn, thiết bị chụp 3102 có thể bao gồm bộ phận nguồn 12 như nêu trên. Khi dữ liệu bao gồm video, bộ mã hóa video 20 được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện quá trình mã hóa video.

Khi dữ liệu bao gồm audio (tức là, giọng nói), bộ mã hóa audio được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện quá trình mã hóa audio. Đối với một số kịch bản thực tế, thiết bị chụp 3102 phân tán dữ liệu audio và video được mã hóa bằng cách ghép kênh chúng với nhau. Đối với các kịch bản thực tế khác, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Thiết bị chụp 3102 phân tán riêng rẽ dữ liệu audio được mã hóa và dữ liệu video được mã hóa đến thiết bị đầu cuối 3106.

Trong hệ thống cung cấp nội dung 3100, thiết bị đầu cuối 310 nhận và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng nhận và khôi phục dữ liệu, chẳng hạn điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, bộ ghi video mạng (network video recorder, NVR)/bộ ghi video số (digital video recorder, DVR) 3112, TV 3114, hộp tín hiệu truyền hình cáp (set top box, STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, thiết bị hỗ trợ số cá nhân (personal digital assistant, PDA) 3122, thiết bị lắp trong xe 3124, hoặc kết hợp của chúng, hoặc tương tự có thể giải mã dữ liệu được mã hóa nêu trên. Chẳng hạn, thiết bị đầu cuối 3106 có thể bao gồm bộ phận đích 14 như nêu trên. Khi dữ liệu được mã hóa bao gồm video, bộ giải mã video 30 được bao gồm trong thiết bị đầu cuối được ưu tiên để thực hiện giải mã video. Khi dữ liệu được mã hóa bao gồm audio, bộ giải mã audio được bao gồm trong thiết bị đầu cuối được ưu tiên để thực hiện quá trình giải mã audio.

Đối với thiết bị đầu cuối có bộ phận hiển thị, chẳng hạn, điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, NVR/DVR 3112, TV 3114, PDA 3122, hoặc thiết bị lắp trong xe 3124, thiết bị đầu cuối có thể cấp dữ liệu được giải mã cho bộ phận hiển thị. Đối với thiết bị đầu cuối không có bộ

phần hiển thị, chẳng hạn, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, bộ phận hiển thị bên ngoài 3126 được tiếp xúc trong đó để nhận và thể hiện dữ liệu được giải mã.

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

Fig.11 là sơ đồ thể hiện cấu trúc của ví dụ của thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 nhận stream từ thiết bị chụp 3102, khối xử lý giao thức 3202 phân tích giao thức truyền của stream. Giao thức bao gồm nhưng không bị giới hạn ở giao thức streaming thời gian thực (Real Time Streaming Protocol, RTSP), giao thức truyền siêu văn bản (Hyper Text Transfer Protocol, HTTP), giao thức truyền video trực tuyến HTTP (HTTP Live streaming protocol, HLS), MPEG-DASH, giao thức vận tải thời gian thực (Real-time Transport protocol, RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, RTMP), hoặc loại tổ hợp bất kỳ của nó, hoặc tương tự.

Sau khi khối xử lý giao thức 3202 xử lý stream, tệp stream được tạo. Tệp tin được xuất ra khỏi phân kênh 3204. Khối phân kênh 3204 có thể tách riêng dữ liệu được ghép kênh thành dữ liệu audio được mã hóa và dữ liệu video được mã hóa. Như nêu trên, đối với một số kịch bản thực tế, chẳng hạn trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được ghép kênh. Trong tình huống thực, dữ liệu được mã hóa được truyền đến bộ giải mã video 3206 và bộ giải mã audio 3208 mà không qua khối phân kênh 3204.

Qua quá trình phân kênh, dòng cơ bản video (elementary stream, ES), audio ES, và một cách tùy chọn phụ đề được tạo. Bộ giải mã video 3206, bao gồm bộ giải mã video 30 như được giải thích theo các phương án thực hiện nêu trên, giải mã ES video bằng phương pháp giải mã như được thể hiện theo các phương án

thực hiện nêu trên để tạo khung video, và cấp dữ liệu này cho khối đồng bộ 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo khung audio, và cấp dữ liệu này cho khối đồng bộ 3212. Theo cách khác, khung video có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.Y) trước khi cấp nó cho khối đồng bộ 3212. Tương tự, khung audio có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.Y) trước khi cấp nó cho khối đồng bộ 3212.

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

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

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

Các toán tử toán học

Các toán tử toán học được sử dụng theo sáng chế giống như các toán tử được sử dụng trong ngôn ngữ lập trình C. Tuy nhiên, các kết quả của các phép chia số nguyên và dịch số học được định nghĩa chính xác hơn, và các phép bổ sung được định nghĩa, chẳng hạn lũy thừa và phép chia số thực. Các quy ước đánh số và đếm thường bắt đầu từ 0, chẳng hạn, “thứ nhất” tương đương với thứ 0, “thứ hai” tương đương với thứ nhất, v.v..

Các phép toán số học

Các phép toán số học được định nghĩa như sau:

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

Các toán tử logic

Các toán tử logic sau được định nghĩa như sau:

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

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

! Logic Boolean “not”

$x ? y : z$ nếu x ĐÚNG hoặc không bằng 0, tính giá trị y ; ngược lại, tính giá trị z .

Các toán tử quan hệ

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

> Lớn hơn

>= Lớn hơn hoặc bằng

< Nhỏ hơn

<= Nhỏ hơn hoặc bằng

= = Bằng

!= Không bằng

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

Các phép toán theo bit

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

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

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

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

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

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

Các toán tử gán

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

= Toán tử gán

++ Tăng, tức là, $x++$ tương đương với $x = x + 1$; Khi được sử dụng trong chỉ số mảng, chuyển sang giá trị của biến trước phép tăng.

-- Giảm, tức là, $x--$ tương đương với $x = x - 1$; Khi được sử dụng trong chỉ số mảng, chuyển sang giá trị của biến trước phép giảm.

+= Tăng lên lượng cụ thể, tức là, $x += 3$ tương đương với $x = x + 3$, và $x += (-3)$ tương đương với $x = x + (-3)$.

-= Giảm đi lượng cụ thể, tức là, $x -= 3$ tương đương với $x = x - 3$, và $x -= (-3)$ tương đương với $x = x - (-3)$.

Ký pháp khoảng

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

$x = y..zx$ chiếm các giá trị số nguyên bắt đầu từ y đến z , có hai giá trị đầu mút, với x , y , và z là các số nguyên và z lớn hơn y .

Các hàm toán học

Các hàm toán học sau được định nghĩa:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Thứ tự ưu tiên phép toán

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

- Các phép toán có độ ưu tiên cao hơn được tính trước các phép toán có độ ưu tiên thấp hơn.
- Các phép toán có cùng độ ưu tiên được tính lần lượt từ trái sang phải.

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

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

Bảng: Độ ưu tiên phép toán từ cao nhất (ở đầu bảng) đến thấp nhất (cuối bảng)

Phép toán (có các toán hạng x, y, và z)
"x++", "x--"
"!x", "-x" (như là toán tử tiền tố đơn nguyên)
x^y
"x * y", "x/y", "x ÷ y", $\frac{x}{y}$, "x % y"
"x + y", "x - y" (như là phép toán hai đối số), " $\sum_{i=x}^y f(i)$ "
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Phần mô tả bằng văn bản các phép toán logic

Trong phần văn bản, mệnh đề của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau:

if(điều kiện 0)

mệnh đề 0

else if(điều kiện 1)

mệnh đề 1

...

else /* nhận xét mang thông tin về điều kiện còn lại */

mệnh đề n

có thể được mô tả theo cách nhau:

... như sau/... mệnh đề sau áp dụng:

- If điều kiện 0, mệnh đề 0
- Ngược lại, if điều kiện 1, mệnh đề 1
- ...
- Ngược lại (nhận xét mang thông tin về điều kiện còn lại),

mệnh đề n

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

Theo văn bản, mệnh đề của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau:

if(điều kiện 0a && điều kiện 0b)

mệnh đề 0

else if(điều kiện 1a || điều kiện 1b)

mệnh đề 1

...

else

mệnh đề n

có thể được mô tả theo cách sau:

... như sau/... mệnh đề sau áp dụng:

- nếu tất cả các điều kiện sau là đúng, mệnh đề 0:
 - điều kiện 0a
 - điều kiện 0b
- Ngược lại, nếu một hoặc nhiều trong số các điều kiện sau là đúng, mệnh đề 1:
 - điều kiện 1a
 - điều kiện 1b
- ...
- Ngược lại, mệnh đề n

Theo văn bản, mệnh đề của các phép toán logic như sẽ được mô tả về mặt toán học ở dạng sau:

if(điều kiện 0)

mệnh đề 0

if(điều kiện 1)

mệnh đề 1

có thể được mô tả theo cách sau:

Khi điều kiện 0, mệnh đề 0

Khi điều kiện 1, mệnh đề 1

Mặc dù các phương án thực hiện sáng chế đã được mô tả chủ yếu dựa trên kỹ thuật tạo mã video, cần lưu ý rằng theo các phương án thực hiện, hệ thống tạo mã 10, bộ mã hóa 20 và bộ giải mã 30 (và tương ứng là hệ thống 10) và các phương án thực hiện khác được mô tả ở đây cũng có thể được tạo cấu hình để

xử lý hoặc tạo mã ảnh tĩnh, tức là, xử lý hoặc tạo mã ảnh riêng rẽ độc lập với ảnh đứng trước hoặc sau như trong quá trình tạo mã video. Nói chung, chỉ các khối dự báo ngoài 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp tạo mã xử lý ảnh bị giới hạn ở một ảnh 17. Tất cả các chức năng khác (cũng được gọi là các công cụ hoặc công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng ngang bằng để xử lý ảnh tĩnh, chẳng hạn, khối tính toán dư 204/304, khối biến đổi 206, khối lượng tử 208, khối lượng tử ngược 210/310, khối biến đổi (ngược) 212/312, khối phân vùng 262/362, khối dự báo trong 254/354, và/hoặc khối lọc vòng 220, 320, và khối tạo mã entropy 270 và khối giải mã entropy 304.

Theo các phương án thực hiện, chẳng hạn, bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, chẳng hạn, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được triển khai trong phần cứng, phần mềm, firmware, hoặc kết hợp bất kỳ của chúng. Nếu được triển khai bằng phần mềm, các chức năng có thể được lưu trữ trên vật máy tính đọc được hoặc được truyền trên phương tiện truyền thông dưới dạng một hoặc nhiều lệnh hoặc mã và được thực thi bằng khối xử lý dựa trên phần cứng. Phương tiện máy tính đọc được có thể bao gồm phương tiện lưu trữ máy tính đọc được, tương ứng với phương tiện hữu hình, chẳng hạn phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm phương tiện bất kỳ để truyền chương trình máy tính từ một điểm sang điểm khác, chẳng hạn, theo giao thức truyền thông. Theo cách này, phương tiện máy tính đọc được thường có thể tương ứng với (1) phương tiện lưu trữ máy tính đọc được hữu hình vốn bất biến hoặc (2) phương tiện truyền thông, chẳng hạn, tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ có thể được truy nhập bằng một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất lệnh, mã và/hoặc cấu trúc dữ liệu để thực hiện các kỹ thuật

được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện máy tính đọc được.

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

Các lệnh có thể được thực thi bằng một hoặc nhiều bộ xử lý, chẳng hạn một hoặc nhiều DSP, bộ vi xử lý đa năng, mạch tích hợp ứng dụng cụ thể (application specific integrated circuit, ASIC), mảng cổng dạng trường lập trình được (field programmable logic array, FPGA), hoặc mạch logic tích hợp hoặc rời rạc tương

đương khác. Theo đó, cụm từ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến cấu trúc bất kỳ nêu trên hoặc cấu trúc khác bất kỳ thích hợp để thực hiện các kỹ thuật được nêu ở đây. Ngoài ra, theo một số khía cạnh, chức năng được mô tả ở đây có thể được bố trí trong phần cứng và/hoặc các môđun phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được đưa vào codec kết hợp. Các kỹ thuật cũng có thể được triển khai đầy đủ trong một hoặc nhiều mạch hoặc các phần tử logic.

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

YÊU CẦU BẢO HỘ

1. Phương pháp tạo mã được triển khai bởi thiết bị giải mã, trong đó phương pháp bao gồm các bước:

thu được giá trị của phần tử cú pháp từ dòng bit, trong đó giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối đối với thành phần sắc độ của lát của ảnh được tạo mã, phần tử cú pháp chỉ báo liệu có thành phần màu trong trong chuỗi video;

phân tách giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát từ dòng bit, khi giá trị của phần tử cú pháp bằng giá trị định trước, trong đó giá trị định trước là giá trị số nguyên;

thực hiện quá trình tách khối trên khối trong lát theo giá trị của tham số điều khiển tách khối.

2. Phương pháp theo điểm 1, trong đó giá trị của phần tử cú pháp thu được từ tập hợp tham số ảnh (picture parameter set, PPS).

3. Phương pháp theo điểm 1 hoặc 2, trong đó giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thu được từ PPS.

4. Phương pháp theo điểm 1 hoặc 2, trong đó giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thu được từ tiêu đề ảnh (picture header, PH).

5. Phương pháp theo điểm 1 hoặc 2, trong đó giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thu được từ tiêu đề lát (slice header, SH).

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp bằng 0.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, trong đó tham số điều khiển tách khối đối với thành phần sắc độ của lát được báo hiệu khi chuỗi video có thành phần màu.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối đối với thành phần độ sáng của lát giống như tham số điều khiển tách khối đối với thành phần sắc độ của lát.

9. Phương pháp theo điểm 8, trong đó phương pháp còn bao gồm bước:

gán giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát bằng giá trị của tham số điều khiển tách khối cho thành phần độ sáng của lát, khi giá trị của phần tử cú pháp không bằng giá trị định trước.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 1, 5 hoặc 9, trong đó giá trị của tham số điều khiển tách khối là các độ lệch tham số tách khối được định trước được áp dụng cho thành phần Cb-Cr chung của lát.

11. Phương pháp tạo mã được triển khai bởi thiết bị mã hóa, trong đó phương pháp bao gồm các bước:

xác định giá trị của phần tử cú pháp của lát của ảnh được tạo mã, trong đó, giá trị của phần tử cú pháp liên quan đến tham số điều khiển tách khối cho thành phần sắc độ của lát của ảnh được tạo mã, phần tử cú pháp chỉ báo liệu có thành phần màu trong chuỗi video;

mã hóa giá trị của tham số điều khiển tách khối đối với thành phần sắc độ của lát thành dòng bit khi giá trị của phần tử cú pháp bằng giá trị định trước, trong đó giá trị định trước là giá trị số nguyên.

12. Phương pháp theo điểm 11, trong đó giá trị của phần tử cú pháp thu được từ tập hợp tham số ảnh (picture parameter set, PPS).

13. Phương pháp theo điểm 12, trong đó giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát thu được từ PPS.

14. Phương pháp theo điểm 11, trong đó giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát thu được từ PH.

15. Phương pháp theo điểm 11, trong đó giá trị của tham số điều khiển tách khối cho thành phần sắc độ của lát thu được từ SH.

16. Phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 15, trong đó khi không có thành phần màu trong chuỗi video, giá trị của phần tử cú pháp bằng 0.

17. Phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 15, trong đó tham số điều khiển tách khối cho thành phần sắc độ của lát được báo hiệu khi chuỗi video có thành phần màu.

18. Phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 15, trong đó giá trị của phần tử cú pháp được sử dụng để xác định liệu tham số điều khiển tách khối cho thành phần độ sáng của lát giống như tham số điều khiển tách khối cho thành phần sắc độ của lát.

19. Phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 15, trong đó giá trị của tham số điều khiển tách khối là các độ lệch tham số tách khối định trước mà được áp dụng cho thành phần Cb-Cr chung của lát.

20. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

21. Bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 19.

22. Bộ giải mã bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10.

23. Bộ mã hóa bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 19.

24. Vật ghi máy tính đọc được mang mã chương trình mà, khi được thực thi bởi máy tính, khiến máy tính thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10 hoặc thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 11 đến 19.

1/10

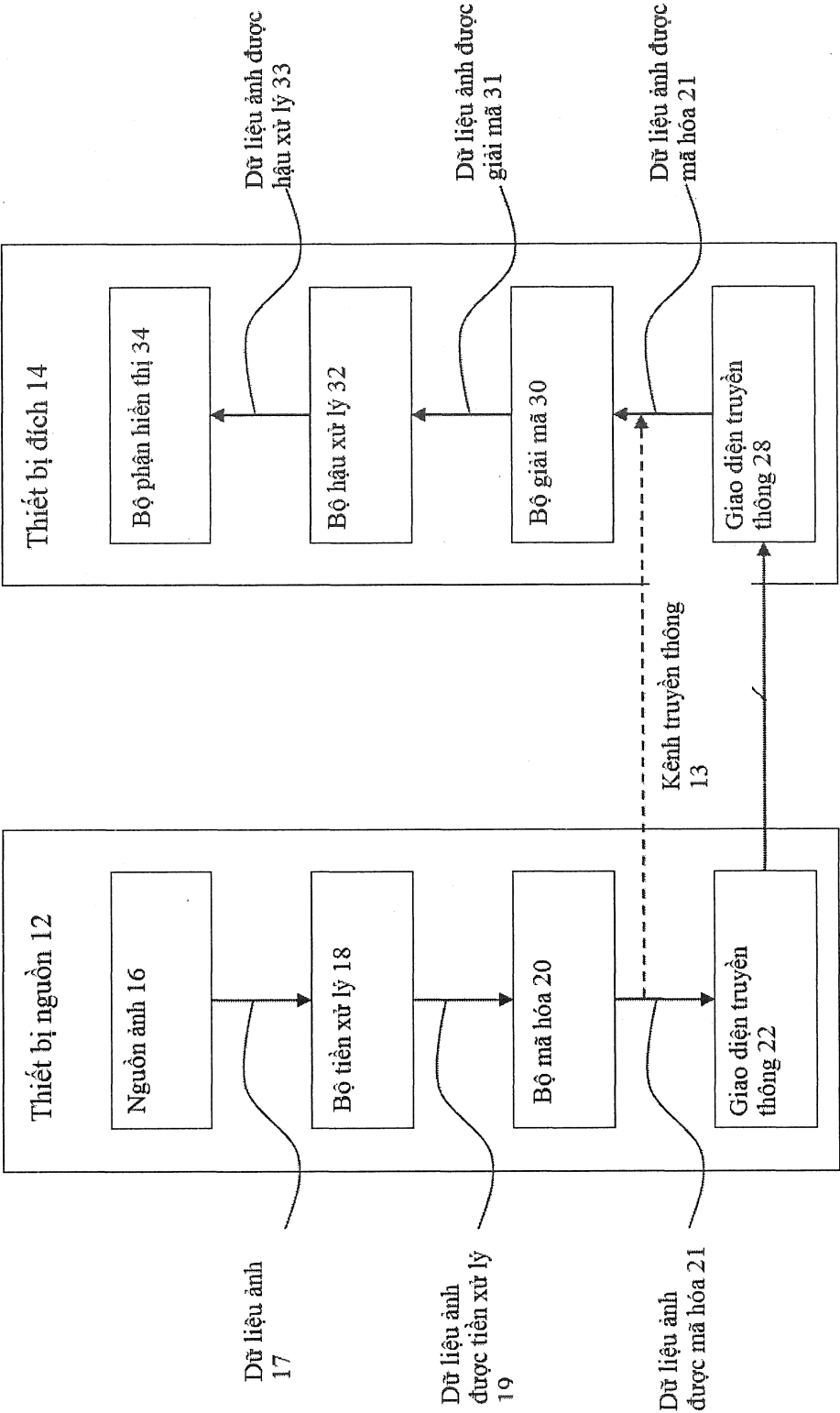


Fig.1A

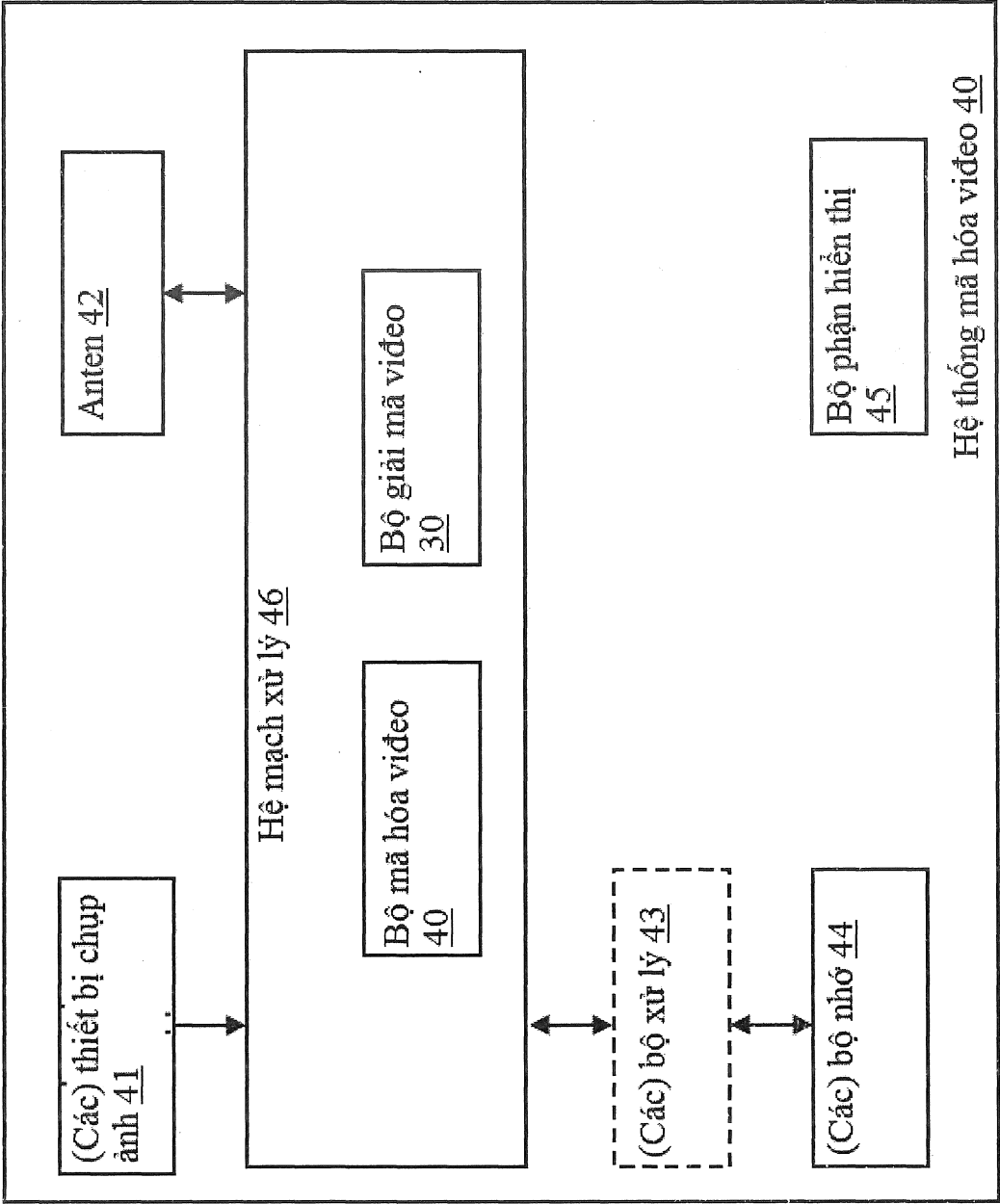


Fig.1B

3/10

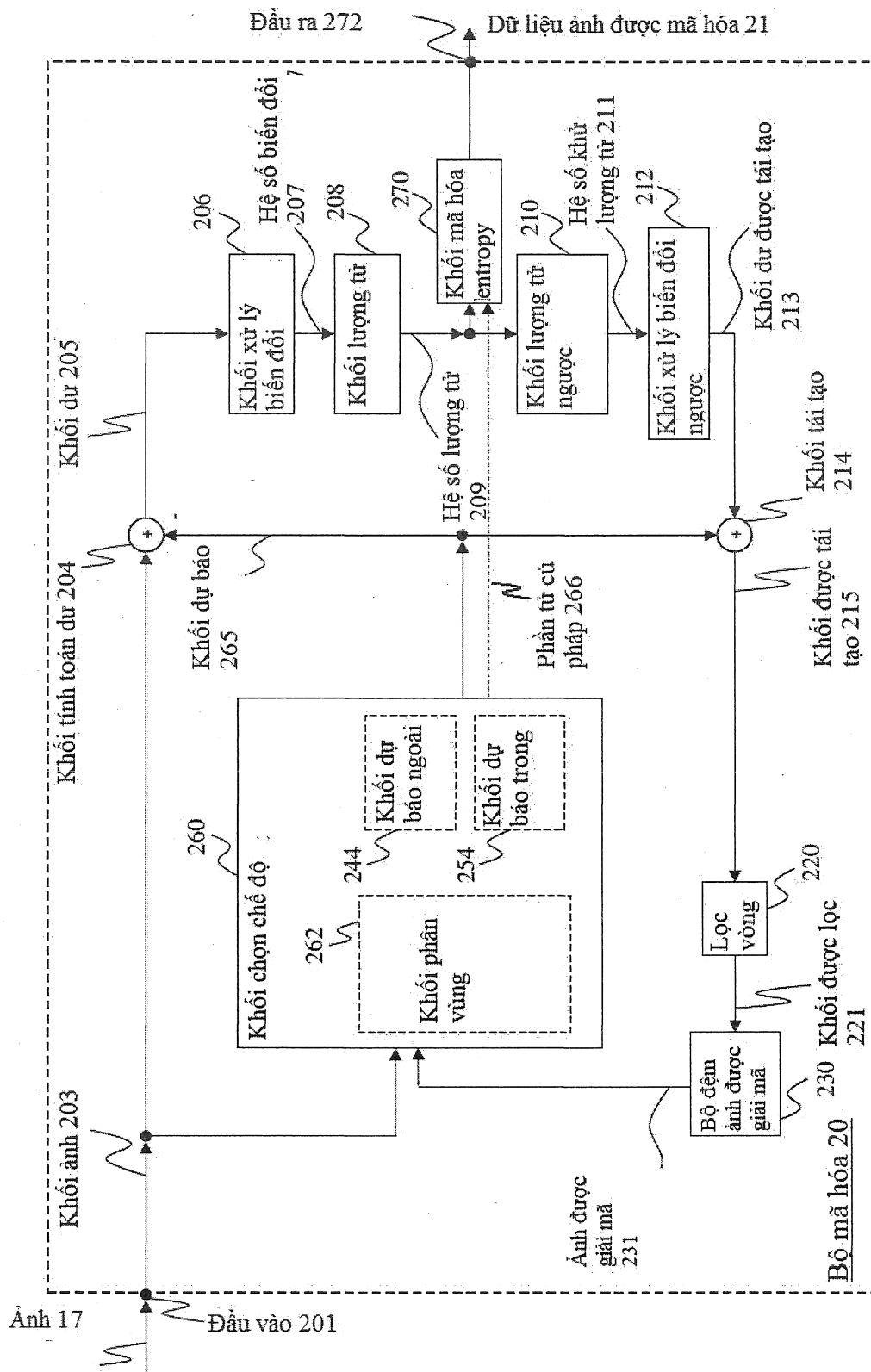


Fig.2

5/10

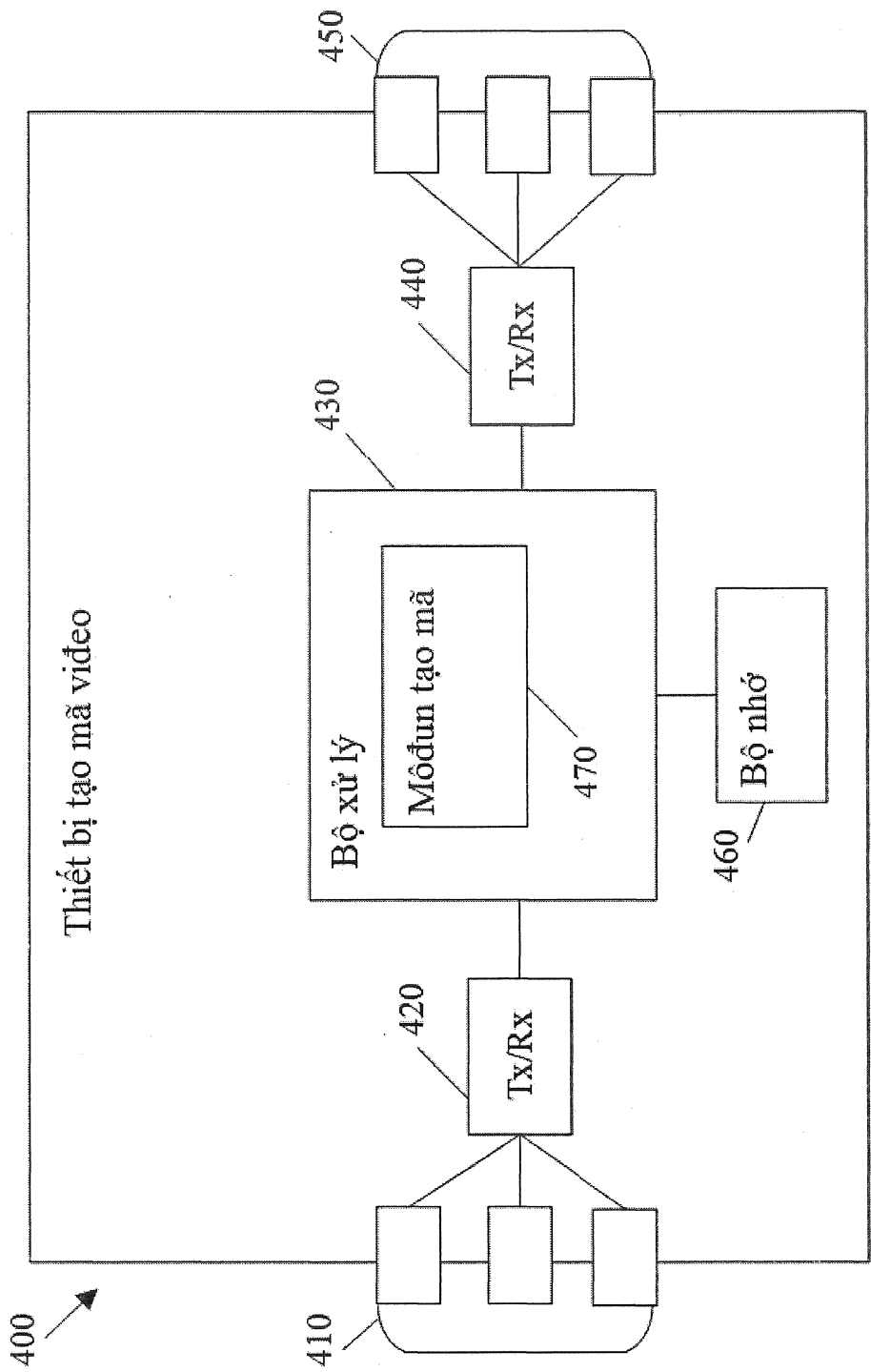


Fig.4

6/10

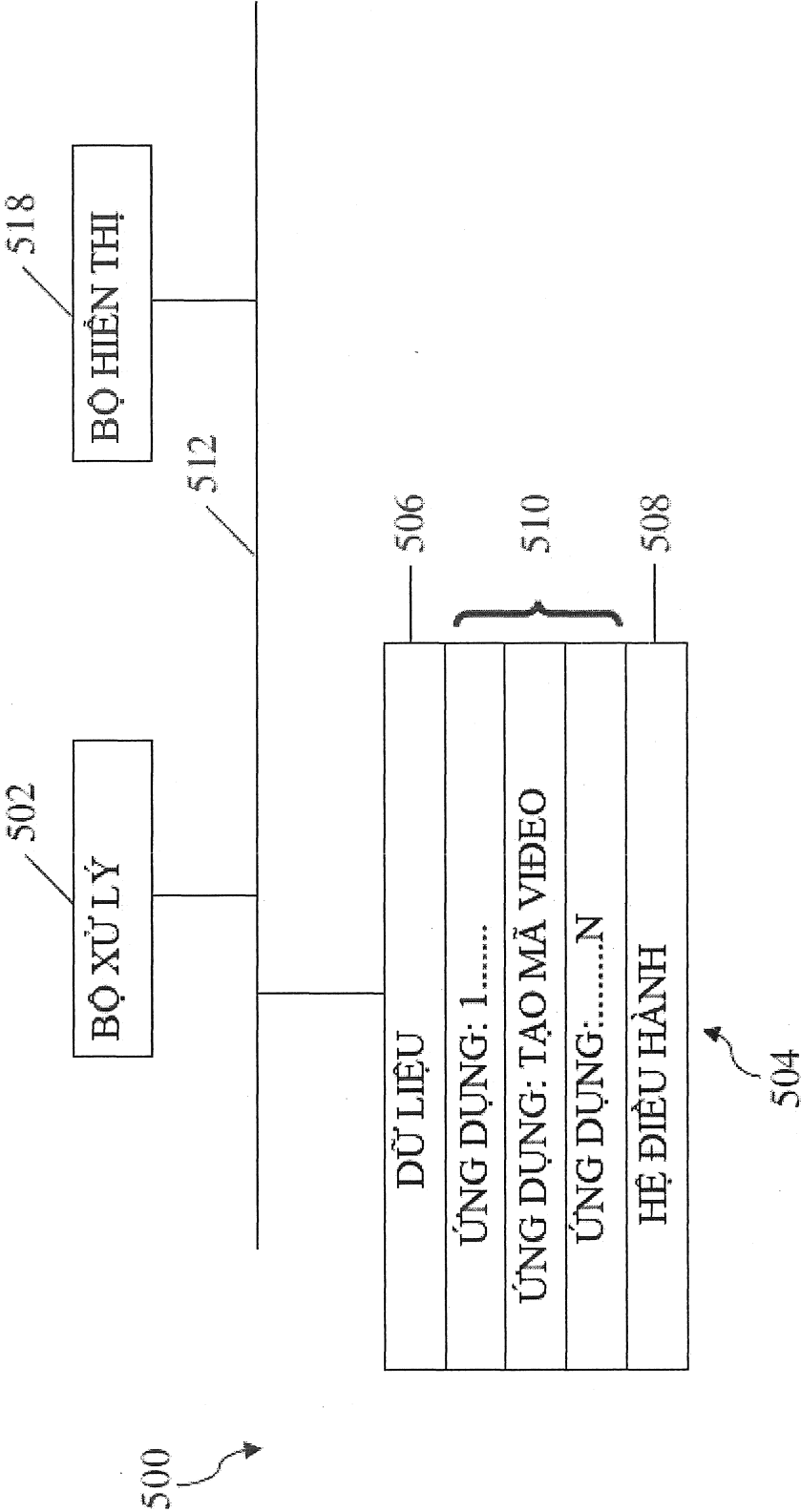


Fig.5

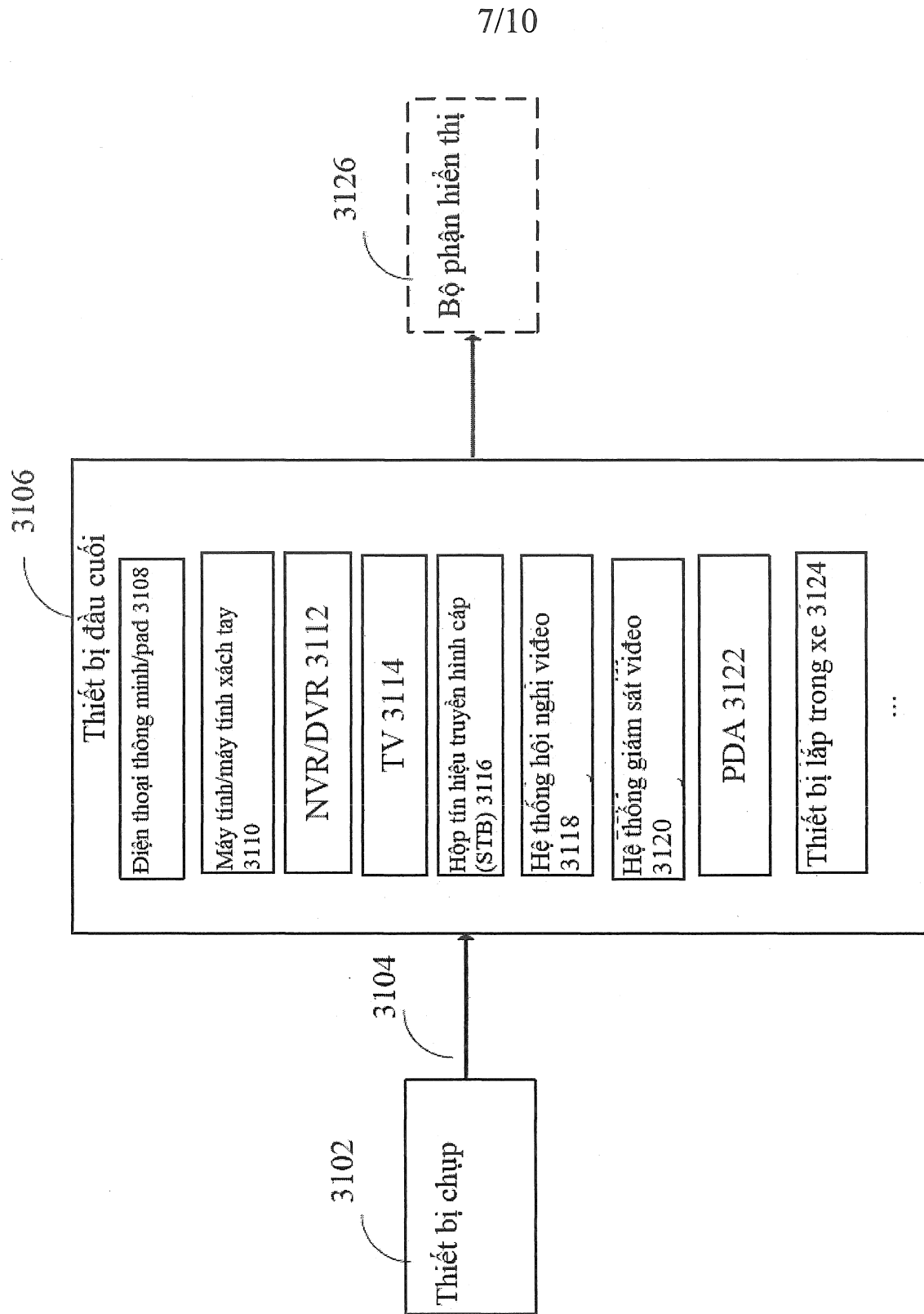


Fig.6

8/10

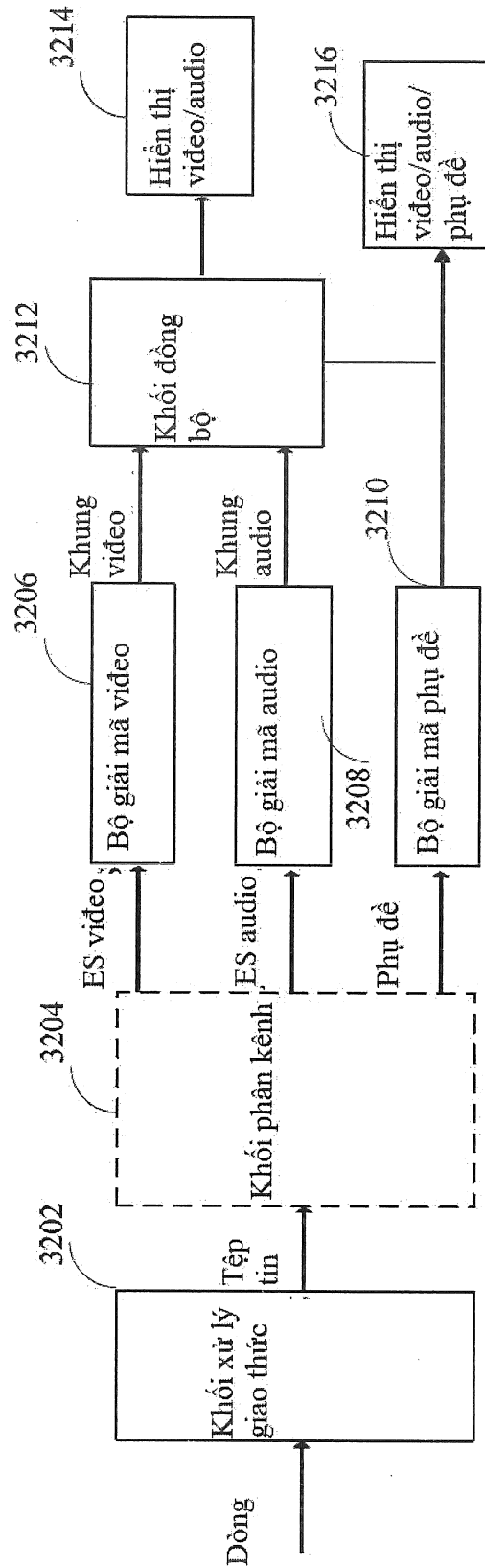


Fig.7

9/10

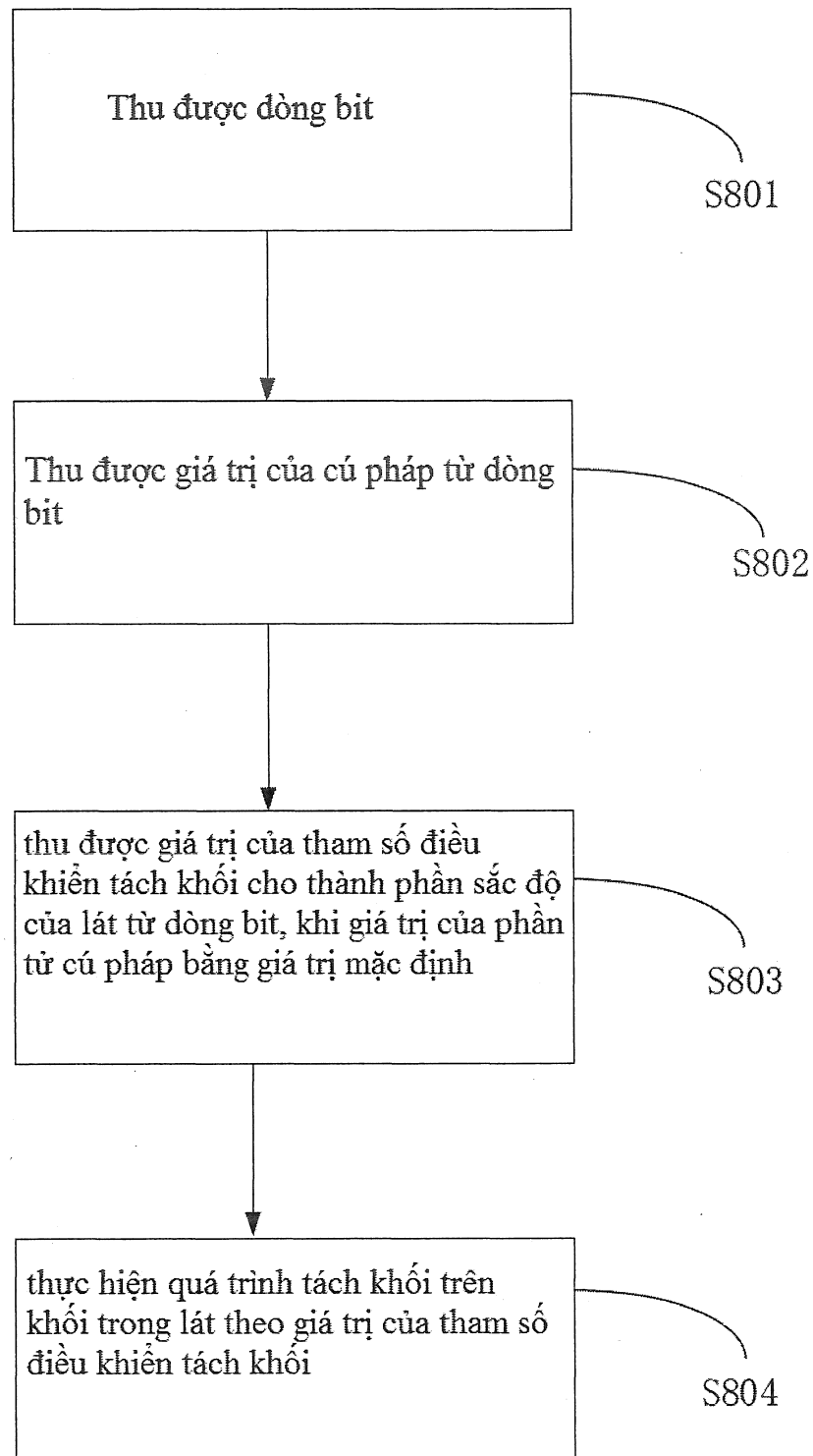


Fig.8

10/10

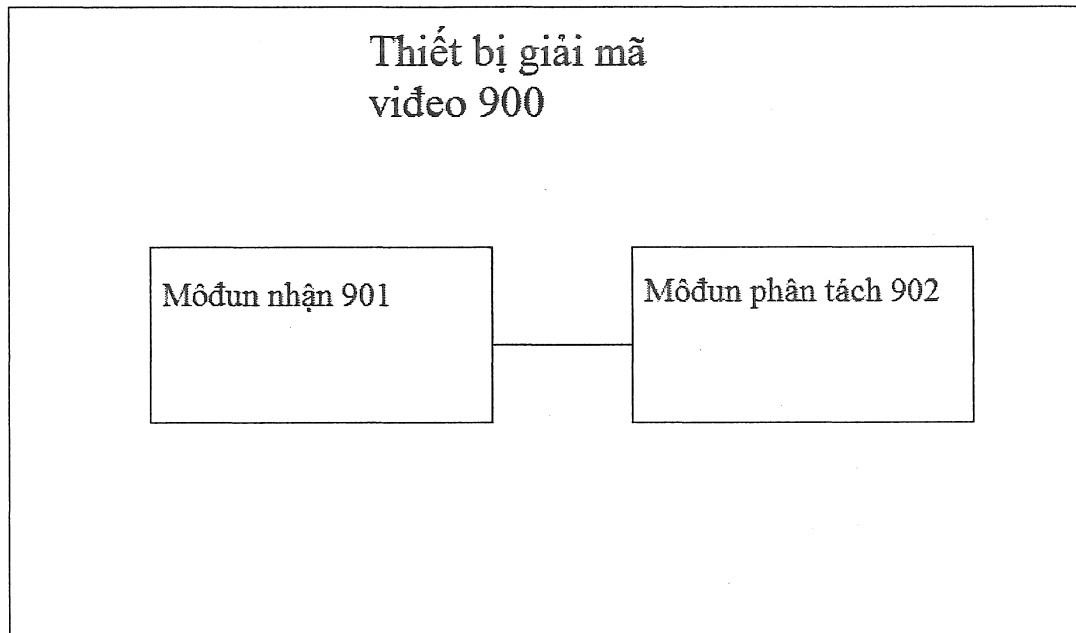


Fig.9