(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0053349**

(51)[2021.01] **H04N 19/52**

(13) **B**

(21) 1-2022-05098
(22) 13/01/2021
(86) PCT/RU2021/050007 13/01/2021
(87) WO 2021/045659 11/03/2021
(30) 62/961,159 14/01/2020 US
(45) 25/11/2025 452
(43) 25/10/2022 415A

(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building Bantian, Longgang District Shenzhen, Guangdong 518129, China

(72) FILIPPOV, Alexey Konstantinovich (RU); RUFITSKIY, Vasily Alexeevich (RU); ALSHINA, Elena Alexandrovna (RU).

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

(54) PHƯƠNG PHÁP MÃ HÓA, THIẾT BỊ GIẢI MÃ VIĐEO, BỘ MÃ HÓA, VÀ VẬT GHI MÁY TÍNH ĐỌC ĐƯỢC

(21)1-2022-05098

(57)Sáng chế đề cập đến phương pháp thu thập số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học để giải mã viđeo và thiết bị giải mã viđeo, trong đó phương pháp bao gồm các bước: thu thập dòng bit cho chuỗi viđeo; thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên dự báo vectơ chuyển động hợp nhất (merging motion vector prediction, MVP); thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu có thể bù chuyển động dựa trên phân vùng hình học cho chuỗi viđeo hay không; và phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

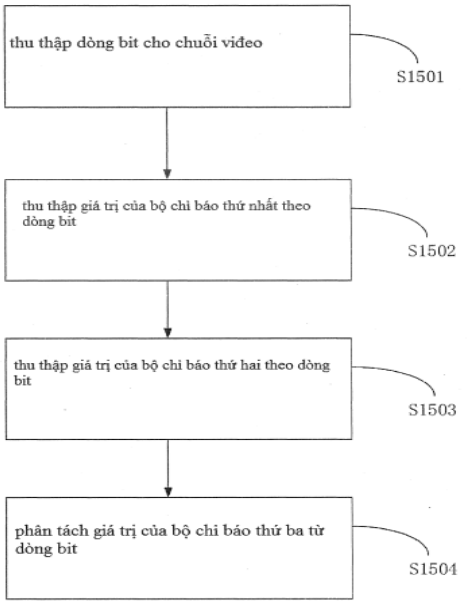

Fig.15

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến lĩnh vực tạo mã ảnh động và cụ thể hơn đến báo hiệu số lượng ứng viên chế độ hợp nhất.

Tình trạng kỹ thuật của sáng chế

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

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

Bản chất kỹ thuật của sáng chế

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

Các mục đích 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 là rõ ràng từ 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 đi kèm, 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 thu thập số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học để giải mã viđeo, phương pháp bao gồm các bước:

thu thập dòng bit cho chuỗi viđeo; thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên dự báo vectơ chuyển động (motion vector prediction, MVP) hợp nhất; thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

Các phương án thực hiện sáng chế bộc lộ phương tiện báo hiệu của bộ chỉ báo số lượng ứng viên chế độ hợp nhất. Số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được báo hiệu có điều kiện. Do vậy, hiệu suaất giải mã và sử dụng dòng bit đã được cải thiện.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Theo một cách thực hiện, trong đó ngưỡng bằng 2.

Theo một cách thực hiện, trong đó giá trị định trước bằng 1.

Theo một cách thực hiện, trong đó bước thu thập giá trị của bộ chỉ báo thứ hai được thực hiện sau bước thu thập giá trị của bộ chỉ báo thứ nhất.

Theo một cách thực hiện, bộ chỉ báo thứ nhất thu được theo phần tử cú pháp được tạo mã trong dòng bit.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được phân tách từ tập hợp tham số chuỗi (sequence parameter set, SPS), của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng. Chẳng hạn, phân tách phần tử cú pháp trong SPS của dòng bit để thu được giá trị của bộ chỉ báo thứ hai.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit. Chẳng hạn, phân tách phần tử cú pháp trong SPS của dòng bit để thu được giá trị của bộ chỉ báo thứ hai.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ ba thu được từ SPS của dòng bit. Chẳng hạn, phân tách phần tử cú pháp trong SPS của dòng bit để thu được giá trị của bộ chỉ báo thứ hai.

Khía cạnh thứ hai của sáng chế đề cập đến thiết bị giải mã viđeo, thiết bị giải mã viđeo bao gồm: môđun nhận, được tạo cấu hình để thu được dòng bit cho chuỗi viđeo; môđun thu thập, được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất biểu diễn số lượng lớn nhất của các ứng viên MVP hợp nhất; môđun thu thập được tạo cấu hình để thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; môđun phân tách, được tạo cấu hình để phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

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

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.

Theo một cách thực hiện, trong đó môđun thu thập được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo một cách thực hiện, trong đó môđun thu thập được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Theo một cách thực hiện, trong đó ngưỡng bằng 2.

Theo một cách thực hiện, trong đó giá trị định trước bằng 1.

Theo một cách thực hiện, trong đó bước thu thập giá trị của bộ chỉ báo thứ hai được thực hiện sau bước thu thập giá trị của bộ chỉ báo thứ nhất.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được phân tách từ SPS của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ ba thu được từ SPS của dòng bit.

Một cách thực hiện đề cập đến phương pháp thu thập số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học để giải mã viđeo, trong đó phương pháp bao gồm các bước:

thu thập dòng bit cho chuỗi viđeo; thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên MVP hợp nhất; và chỉ nếu giá trị thu được của bộ chỉ báo thứ nhất bằng hoặc lớn hơn ngưỡng: thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; và phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, chỉ khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và giá trị của bộ chỉ báo

thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

Khía cạnh thứ ba của sáng chế đề cập đến phương pháp mã hóa số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học, phương pháp bao gồm các bước:

xác định giá trị của bộ chỉ báo thứ nhất, trong đó bộ chỉ báo thứ nhất biểu diễn số lượng lớn nhất của các ứng viên MVP hợp nhất; xác định giá trị của bộ chỉ báo thứ hai, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; mã hóa giá trị của bộ chỉ báo thứ ba thành dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

Các phương án thực hiện sáng chế đề cập đến phương tiện báo hiệu bộ chỉ báo số lượng ứng viên chế độ hợp nhất. Số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được báo hiệu có điều kiện. Do vậy, hiệu suất sử dụng và giải mã dòng bit đã được cải thiện.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Theo một cách thực hiện, trong đó ngưỡng bằng 2.

Theo một cách thực hiện, trong đó giá trị định trước bằng 1.

Theo một cách thực hiện, trong đó bước xác định giá trị của bộ chỉ báo thứ hai được thực hiện sau bước xác định giá trị của bộ chỉ báo thứ nhất.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được mã hóa trong SPS của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được mã hóa trong SPS của dòng bit.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ ba được mã hóa trong SPS của dòng bit.

Khía cạnh thứ tư của sáng chế đề cập đến thiết bị mã hóa viđeo, thiết bị mã hóa viđeo bao gồm: môđun xác định, được tạo cấu hình để xác định giá trị của bộ chỉ báo thứ nhất, trong đó bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên MVP hợp nhất; môđun xác định được tạo cấu hình để xác định giá trị của bộ chỉ báo thứ hai, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; môđun mã hóa, được tạo cấu hình để mã hóa giá trị của bộ chỉ báo thứ ba thành dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

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

Theo một cách thực hiện, trong đó môđun xác định được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo một cách thực hiện, trong đó môđun xác định được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Theo một cách thực hiện, trong đó ngưỡng bằng 2.

Theo một cách thực hiện, trong đó giá trị định trước bằng 1.

Theo một cách thực hiện, trong đó bước xác định giá trị của bộ chỉ báo thứ hai được thực hiện sau bước xác định giá trị của bộ chỉ báo thứ nhất.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được mã hóa trong SPS của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ hai được mã hóa trong SPS của dòng bit.

Theo một cách thực hiện, trong đó giá trị của bộ chỉ báo thứ ba được mã hóa trong SPS của dòng bit.

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 cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ nhất.

Khía cạnh thứ sáu của sáng chế đề cập đến bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ ba và cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ ba.

Khía cạnh thứ bảy 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 cách thực hiện bất kỳ trong các 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 tám của sáng chế đề cập đến bộ giải mã, bao gồm: một hoặc nhiều bộ xử lý; và vật ghi máy tính đọc được bất biến 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 để thực hiện phương pháp theo trường hợp bất kỳ của khía cạnh thứ nhất, khía cạnh thứ ba và cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ ba.

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

thứ ba và cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ ba.

Khía cạnh thứ mười của sáng chế đề cập đến bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ ba và cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ ba.

Khía cạnh thứ mười một của sáng chế đề cập đến bộ mã hóa, bao gồm: một hoặc nhiều bộ xử lý; và vật ghi máy tính đọc được bất biến 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 để thực hiện phương pháp theo trường hợp bất kỳ của khía cạnh thứ ba và cách thực hiện bất kỳ trong các cách thực hiện của khía cạnh thứ ba.

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

Khía cạnh thứ mười ba của sáng chế đề cập đến dòng bit được mã hóa cho tín hiệu viđeo bằng cách bao gồm các phần tử cú pháp, trong đó các phần tử cú pháp bao gồm bộ chỉ báo thứ hai (chẳng hạn sps_geo_enabled_flag), và trong đó bộ chỉ báo thứ ba sps_max_num_merge_cand_minus_max_num_geo_cand được báo hiệu có điều kiện ít nhất dựa trên giá trị của sps_geo_enabled_flag.

Khía cạnh thứ mười bốn của sáng chế đề cập đến vật ghi bất biến bao gồm dòng bit được mã hóa được giải mã bằng thiết bị giải mã ảnh, dòng bit được tạo bằng cách phân chia khung của tín hiệu viđeo hoặc tín hiệu ảnh thành các khối, và bao gồm các phần tử cú pháp, trong đó các phần tử cú pháp bao gồm bộ chỉ báo thứ ba (chẳng hạn sps_max_num_merge_cand_minus_max_num_geo_cand) theo điểm bất kỳ trong các điểm nêu trên.

Khía cạnh thứ mười lăm của sáng chế đề cập đến phương pháp giải mã viđeo, phương pháp bao gồm bước:

thu thập dòng bit cho chuỗi viđeo; thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất biểu diễn số lượng lớn nhất của các ứng viên MVP hợp nhất; thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit,

trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất;

tạo danh sách các ứng viên hợp nhất cho khối tạo mã (coding block, CB) hiện tại, theo các vectơ chuyển động (motion vector, MV) của các khối lân cận của CB hiện tại;

thu thập chỉ số hợp nhất theo giá trị của bộ chỉ báo thứ ba;

thu thập vectơ chuyển động của CB hiện tại theo chỉ số hợp nhất và danh sách ứng viên hợp nhất;

tái tạo CB hiện tại theo MV của CB hiện tại.

Khía cạnh thứ mười sáu của sáng chế đề cập đến thiết bị giải mã viđeo, thiết bị giải mã viđeo bao gồm: môđun nhận, được tạo cấu hình để thu được dòng bit cho chuỗi viđeo; môđun thu thập, được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên MVP hợp nhất; môđun thu thập được tạo cấu hình để thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; môđun phân tách, được tạo cấu hình để phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất;

Môđun tạo danh sách ứng viên hợp nhất, được tạo cấu hình để tạo danh sách ứng viên hợp nhất cho CB hiện tại, theo các MV của các khối lân cận của CB hiện tại;

môđun thu thập được tạo cấu hình để thu thập chỉ số hợp nhất theo giá trị của bộ chỉ báo thứ ba;

môđun thu thập MV, được tạo cấu hình để thu được MV của CB hiện tại theo chỉ số hợp nhất và danh sách ứng viên hợp nhất;

môđun tái tạo pixel, được tạo cấu hình để tái tạo CB hiện tại theo MV của CB hiện tại.

Các chi tiết hoặc các ví dụ về khía cạnh thứ mười lăm của sáng chế và khía cạnh thứ mười sáu của sáng chế có thể viện dẫn đến các ví dụ nêu trên được bộc lộ trong khía cạnh thứ nhất đến khía cạnh thứ mười bốn của sáng chế.

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

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

Mô tả vắn tắt các hình vẽ

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

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

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

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

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

Fig.6 là lưu đồ ra quyết định phía bộ mã hóa dự báo có trọng số và ước tính tham số;

Fig.7 là hình vẽ minh họa ví dụ của chế độ dự báo tam giác;

Fig.8 là hình vẽ minh họa ví dụ của chế độ dự báo hình học;

Fig.9 là hình vẽ minh họa ví dụ khác của chế độ dự báo hình học;

Fig.10 là sơ đồ khối thể hiện cấu trúc ví dụ của hệ thống cấp nội dung 3100 thực hiện dịch vụ truyền nội dung;

Fig.11 là sơ đồ khối thể hiện cấu trúc của ví dụ của thiết bị đầu cuối;

Fig.12 là sơ đồ khối minh họa ví dụ của phương pháp dự báo ngoài theo sáng chế;

Fig.13 là sơ đồ khối minh họa ví dụ của thiết bị để dự báo ngoài theo sáng chế;

Fig.14 là sơ đồ khối minh họa ví dụ khác của thiết bị để dự báo ngoài theo sáng chế;

Fig.15 là lưu đồ thể hiện phương án thực hiện phương pháp theo sáng chế;

Fig.16 là sơ đồ khối thể hiện thiết bị phương án thực hiện theo sáng chế.

Các dấu chỉ dẫn giống hệt dưới đây đề cập đến các dấu hiệu giống hệt hoặc ít nhất tương đương về mặt chức năng nếu không có chỉ báo 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 lôgic 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 viđeo thường đề cập đến xử lý chuỗi ảnh, tạo viđeo hoặc chuỗi viđeo. 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 viđeo. Kỹ thuật tạo mã viđeo (hoặc tạo mã nói chung) bao gồm hai phần mã hóa viđeo và giải mã viđeo. Kỹ thuật mã hóa viđeo được thực hiện ở phía nguồn, thường bao gồm bước xử lý (chẳng hạn, bằng cách nén) các ảnh viđeo ban đầu để giảm lượng dữ liệu cần để biểu diễn các ảnh viđeo (để lưu trữ và/hoặc truyền hiệu quả hơn). Giải mã viđeo được thực hiện ở phía đích và thường bao gồm xử lý ngược so với bộ mã hóa để tái tạo các ảnh viđeo. Các phương án thực hiện đề cập đến "tạo mã" các ảnh viđeo (hoặc ảnh nói chung) cần được hiểu là liên quan đến "mã hóa" hoặc "giải mã" các ảnh viđeo hoặc các chuỗi viđeo 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 viđeo không tổn hao, các ảnh viđeo gốc có thể được tái tạo, tức là, các ảnh viđeo được tái tạo có cùng chất lượng với các ảnh viđeo gốc (giả sử không tổn hao truyền hoặc tổn hao dữ liệu khác trong khi lưu trữ hoặc truyền). Trong trường hợp tạo mã viđeo 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 viđeo, không thể được tái tạo hoàn toàn ở bộ giải mã, tức là, chất lượng của các ảnh viđeo được tái tạo thấp hơn hoặc kém hơn so với chất lượng của các ảnh viđeo ban đầu.

Vài chuẩn tạo mã viđeo thuộc nhóm "codec viđeo 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 viđeo 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, viđeo thường được xử lý, tức là, được mã hóa, ở mức khối (khối viđeo), 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 viđeo 10, bộ mã hóa viđeo 20 và bộ giải mã viđeo 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 viđeo 10 (hoặc gọi tắt là hệ thống tạo mã 10) có thể sử dụng các kỹ thuật của sáng chế. Bộ mã hóa viđeo 20 (hoặc gọi tắt là bộ mã hóa 20) và bộ giải mã viđeo 30 (hoặc gọi tắt là bộ giải mã 30) của hệ thống tạo mã viđeo 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 có thể 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 thế giới thực, ảnh được tạo bằng máy tính (chẳng hạn, nội dung màn hình, ảnh thực tại ảo (virtual reality, VR)) và/hoặc kết hợp bất kỳ của 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 gọn, 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 viđeo 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 có thể bao gồm bộ giải mã 30 (chẳng hạn, bộ giải mã viđeo 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 có thể 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 có thể được tạo cấu hình để truyền hoặc nhận dữ liệu ảnh được mã hóa 21 hoặc dữ liệu được mã hóa 13 qua liên kết truyền thông trực tiếp giữa 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 trỏ 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 điôt 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 viđeo 20) hoặc bộ giải mã 30 (chẳng hạn, bộ giải mã viđeo 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), lôgic rời rạc, phần cứng, tạo mã viđeo 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 viđeo 20 và bộ giải mã viđeo 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 viđeo, thiết bị viđeo 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 and 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 viđeo 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 viđeo (chẳng hạn, mã hóa viđeo hoặc giải mã viđeo) 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 viđeo có thể mã hóa và lưu dữ liệu vào bộ nhớ, và/hoặc thiết bị giải mã viđeo 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 viđeo hiệu suất cao (High-Efficiency Video Coding, HEVC) hoặc đến phần mềm tham chiếu tạo mã viđeo đa dụng (Versatile Video coding, VVC), chuẩn tạo mã vi đeo thế hệ tiếp theo được phát triển bởi nhóm cộng tác chung về tạo mã viđeo (Joint Collaboration Team on Video Coding, JCT-VC) của nhóm chuyên gia tạo mã viđeo 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 viđeo 20 ví dụ có thể được tạo cấu hình để thực hiện các phương án thực hiện sáng chế. Trong ví dụ trên Fig.2, bộ mã hóa viđeo 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 vong 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 viđeo 20 như được thể hiện theo Fig.2 có thể cũng được gọi là bộ mã hóa viđeo lai hoặc bộ mã hóa viđeo theo codec viđeo 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 viđeo 20, trong đó đường tín hiệu ngược của bộ mã hóa viđeo 20 tương ứng với đường

tín hiệu của bộ giải mã (xem bộ giải mã viđeo 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 viđeo 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 viđeo hoặc chuỗi viđeo. 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 viđeo 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 viđeo, tức là, chuỗi viđeo 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 RBG hoặc không gian màu, ảnh bao gồm mảng mẫu đỏ, lục và lam tương ứng. Tuy nhiên, trong kỹ thuật tạo mã viđeo, 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 viđeo 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 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 viđeo 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 viđeo 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 MxN (M cột x N hàng), hoặc mảng hệ số biến đổi MxN.

Theo các phương án thực hiện, bộ mã hóa viđeo 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 viđeo 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 viđeo), 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.

Theo các phương án thực hiện, bộ mã hóa viđeo 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 nhóm ô (cũng được gọi là các nhóm ô viđeo) và/hoặc các ô (cũng được gọi là các ô viđeo), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều nhóm ô (thường là không chồng lấn), và mỗi 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 viđeo 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 viđeo 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 viđeo 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 vectơ. 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 viđeo 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 viđeo 30 có thể nhận và áp dụng các QP để giải mã.

Lượng tử ngược

Khối lượng tử ngược 210 có thể 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 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 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 viđeo. 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, bộ lọc song phương, ALF, bộ lọc làm nét, bộ lọc làm mượt hoặc bộ lọc cộng tác, hoặc kết hợp bất kỳ của các bộ lọc. 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 viđeo 20 (bộ lọc vòng 220) có thể được tạo cấu hình để xuất ra các tham số lọc vòng (chẳng hạn, thông tin SAO), 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 viđeo bằng bộ mã hóa viđeo 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 khối 203 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 viđeo 20 ví dụ sẽ được giải thích chi tiết hơn.

Phân vùng

Khối phân vùng 262 có thể phân vùng (hoặc tách) khối hiện tại 203 thành các phân vùng nhỏ hơn, chẳng hạn, các khối nhỏ hơn có kích thước hình vuông hoặc hình chữ nhật. 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 đê 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.

Như nêu trên, cụm từ "khối" như được sử dụng ở đây có thể là phần, cụ thể là phần hình vuông hoặc hình chữ nhật, của ảnh. Chẳng hạn, dựa vào HEVC và VVC, khối có thể là hoặc tương ứng với đơn vị cây mã (coding tree unit, CTU), đơn vị mã (coding unit, CU), đơn vị dự báo (prediction unit, PU), và đơn vị biến đổi (transform unit, TU) và/hoặc với các khối tương ứng, chẳng hạn khối cây mã (coding tree block, CTB), khối mã (coding block, CB), khối biến đổi (transform block, TB) hoặc khối dự báo (prediction block, PB).

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 NxN 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 MxN 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 bằng cách áp dụng quá trình dự báo dựa trên loại tách PU, CU lá có thể được phân vùng thành các đơn vị biến đổi (transform unit, TU) theo cấu trúc QT khác giống như cây mã cho CU.

Theo các phương án thực hiện, chẳng hạn, theo chuẩn tạo mã viđeo mới nhất đang được phát triển, được gọi là tạo mã viđeo đa dụng (Versatile Video Coding, VVC), phân vùng cây tứ phân và cây nhị phân kết hợp (Quad-tree and binary tree, QTBT) chẳng hạn được sử dụng để phân vùng khối mã (coding block, CB). Trong cấu trúc khối QTBT, CU có thể có dạng hình vuông hoặc chữ nhật. Chẳng hạn, đơn vị cây mã (coding tree unit, CTU) trước hết được phân vùng bởi cấu trúc QT. Các nút lá QT được phân vùng tiếp bởi cấu trúc BT hoặc cây tam phân. Các nút lá cây phân vùng được gọi là các CU, và việc phân đoạn được sử dụng để xử lý dự báo và biến đổi mà không phân vùng tiếp. Điều này nghĩa là CU, PU và TU có cùng kích thước khối trong cấu trúc CB QTBT. Một cách song song, đa phân vùng, chẳng hạn, phân vùng TT có thể được sử dụng cùng với cấu trúc khối QTBT.

Trong một ví dụ, khối chọn chế độ 260 của bộ mã hóa viđeo 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 viđeo 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.

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 viđeo 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, ¼ pel, 1/16 pel, hay không.

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

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 viđeo 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 viđeo.

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 viđeo để sử dụng bởi bộ giải mã viđeo 30 khi giải mã các khối ảnh của lát viđeo. 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 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 viđeo 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 viđeo 30, hoặc được lưu trữ trong bộ nhớ để truyền hoặc truy xuất sau này bằng bộ giải mã viđeo 30.

Các biến thể cấu trúc khác của bộ mã hóa viđeo 20 có thể được sử dụng để mã hóa dòng viđeo. 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 viđeo 30 được tạo cấu hình để thực hiện các phương án thực hiện sáng chế. Bộ giải mã viđeo 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 viđeo 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 viđeo 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 viđeo 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 viđeo 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 viđeo 20 áp dụng tương ứng cho các khối và các chức năng tương ứng của bộ giải mã viđeo 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 viđeo 30 có thể nhận các phần tử cú pháp ở mức lát viđeo và/hoặc mức khối viđeo. 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 viđeo 20 cho mỗi khối viđeo trong lát viđeo (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 viđeo. 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, bộ lọc song phương, ALF, bộ lọc làm nét, bộ lọc làm mượt hoặc bộ lọc cộng tác, hoặc kết hợp bất kỳ của các bộ lọc. 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 viđeo 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 viđeo 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 viđeo 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 viđeo 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 viđeo của lát viđeo 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 viđeo 30 có thể tạo các danh sách ảnh tham chiếu, List 0 và List 1, sử dụng các kỹ thuật tạo mặc định 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 viđeo) và/hoặc các ô (chẳng hạn, các ô viđeo) bên cạnh hoặc theo cách khác với các lát (chẳng hạn, các lát viđeo), chẳng hạn, viđeo 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 được tạo cấu hình để xác định thông tin dự báo cho khối viđeo của lát viđeo 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 viđeo 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 viđeo của lát viđeo, 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 viđeo được mã hóa ngoài của lát, trạng thái dự báo ngoài cho mỗi khối viđeo được tạo mã ngoài của lát, và thông tin khác để giải mã các khối viđeo trong lát viđeo 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 viđeo)

và/hoặc các ô (chẳng hạn, các ô viđeo) bên cạnh hoặc theo cách khác với các lát (chẳng hạn, các lát viđeo), chẳng hạn, viđeo 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 viđeo 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 viđeo), 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).

Theo các phương án thực hiện, bộ giải mã viđeo 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 nhóm ô (cũng được gọi là các nhóm ô viđeo) và/hoặc các ô (cũng được gọi là các ô viđeo), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều nhóm ô (thường là không chồng lấn), và mỗi 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 viđeo 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 viđeo 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 viđeo 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^(bitDepth-1) đến 2^(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.

Phương pháp 1: loại bỏ MSB tràn bằng các phép toán sau

$$ux = (mvx + 2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

$$mvx = (ux >= 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

$$uy = (mvy + 2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

$$mvy = (uy >= 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

trong đó mvx là thành phần ngang của MV của khối ảnh hoặc khối thành phần, mvy là thành phần dọc của MV của khối ảnh hoặc khối thành phần, và ux và uy chỉ báo giá trị trung gian;

Chẳng hạn, nếu giá trị của mvx là -32769, sau khi áp dụng công thức (1) và (2), giá trị thu được bằng 32767. Trong hệ thống máy tính, các số thập phân được lưu trữ dưới dạng bù 2. Bù 2 của -32769 bằng 1,0111,1111,1111,1111 (17 bit), sau đó MSB bị loại bỏ, do vậy bù 2 thu được bằng 0111,1111,1111,1111 (số thập phân là 32767), giống như đầu ra bằng cách áp dụng công thức (1) và (2).

$$ux = (mvpx + mvdx + 2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

$$mvx = (ux >= 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

$$uy = (mvpy + mvdy + 2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

$$mvy = (uy >= 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (8)$$

Các phép toán có thể được áp dụng trong khi cộng mvp và mvd, như được thể hiện trên công thức từ (5) đến (8).

Phương pháp 2: loại bỏ MSB tràn bằng cách rút gọn giá trị

$vx = Clip3(-2^{bitDepth-1}, 2^{bitDepth-1} -1, vx)$

$vy = Clip3(-2^{bitDepth-1}, 2^{bitDepth-1} -1, vy)$

trong đó vx là thành phần ngang của MV của khối ảnh hoặc khối thành phần, vy là thành phần dọc của MV của khối ảnh hoặc khối thành phần; x, y và z một cách lần lượt tương ứng với ba giá trị đầu vào của quá trình rút gọn MV, và định nghĩa về hàm Clip3 như sau:

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

Fig.4 là sơ đồ của thiết bị tạo mã viđeo 400 theo phương án thực hiện sáng chế. Thiết bị tạo mã viđeo 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 viđeo 400 có thể là bộ giải mã, chẳng hạn, bộ giải mã viđeo 30 trên Fig.1A hoặc bộ mã hóa, chẳng hạn, bộ mã hóa viđeo 20 trên Fig.1A.

Thiết bị tạo mã viđeo 400 bao gồm cổng vào 410 và bộ nhận (Rx) 420 để nhận dữ liệu; bộ vi xử lý, đơn vị lôgic, 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 viđeo 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 viđeo 400 và biến đổi thiết bị tạo mã viđeo 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 nhập ngẫu nhiên (random access memory, RAM), bộ nhớ có nội dung đánh địa chỉ được tam phân (ternary content-addressable memory, TCAM), và/hoặc RAM tĩnh (static random-access memory, SRAM).

Fig.5 là sơ đồ khối 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 nhập bởi bộ xử lý 502 nhờ sử dụng buýt 512. Bộ nhớ 504 có thể còn bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình cho phép bộ xử lý 502 thực hiện các phương pháp được 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 viđeo 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 buýt 512.

Mặc dù được mô tả ở đây dưới dạng một buýt, buýt 512 của thiết bị 500 có thể bao gồm nhiều buýt. Ngoài ra, bộ 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.

Chế độ phân vùng tam giác (Triangular partitioning mode, TPM) và chế độ phân vùng chuyển động hình học (GEO) cũng lần lượt được biết đến là chế độ hợp nhất tam giác và chế độ hợp nhất hình học, là các kỹ thuật phân vùng kích hoạt các đường biên không ngang và không dọc giữa các phân vùng dự báo, trong đó đơn vị dự báo PU1 và đơn vị dự báo PU1 được kết hợp trong vùng nhờ sử dụng thủ tục tính trung bình có trọng số của các tập con của các mẫu liên quan đến các thành phần màu khác nhau. TPM kích hoạt các đường biên giữa các phân vùng dự báo dọc theo các đường chéo khối chữ nhật, trong đó các đường biên theo GEO có thể được đặt ở các vị trí tùy ý. Trong vùng mà thủ tục tính trung bình có trọng số được áp dụng cho, các số nguyên trong các hình vuông ký hiệu các trọng số W_{PU1} được áp dụng cho thành phần độ sáng của đơn vị dự báo PU1. Trong ví dụ, các trọng số W_{PU2} được áp dụng cho thành phần độ sáng của đơn vị dự báo PU2 được tính toán như sau:

$W_{PU2} = 8 - W_{PU1}$.

Các trọng số được áp dụng cho các thành phần sắc độ của các khối dự báo tương ứng có thể khác với các trọng số được áp dụng cho các thành phần độ sáng của các khối dự báo tương ứng.

Các chi tiết về cú pháp cho TPM xuất hiện trong Bảng 1, trong đó 4 phần tử cú pháp được sử dụng để báo hiệu thông tin về TPM:

MergeTriangleFlag là cờ nhận diện liệu TPM có được chọn hay không ("0" nghĩa là TPM không được chọn; ngược lại, TPM được chọn);

merge_triangle_split_dir là cờ hướng tách cho TPM ("0" nghĩa là hướng tách từ góc trên bên trái đến góc dưới bên phải; ngược lại, hướng tách là từ góc trên bên phải đến góc dưới bên trái);

merge_triangle_idx0 và merge_triangle_idx1 là các chỉ số của các ứng viên hợp nhất 0 và 1 được sử dụng cho TPM.

Bảng 1. Cú pháp dữ liệu hợp nhất bao gồm cú pháp cho TPM

merge_data(x0, y0, cbWidth, cbHeight) {	**Mô tả**
if (CuPredMode[x0][y0] == MODE_IBC) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag || cbWidth * cbHeight != 32)	
regular_merge_flag[x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] = = 1){	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag && cbWidth * cbHeight != 32)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] = = 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
merge_triangle_idx1[x0][y0]	ae(v)
}	
}	
}	
}	

}	
}	

Trong ví dụ, TPM được mô tả trong kiến nghị sau: R-L. Liao and C.S. Lim "CE10.3.1.b: Triangular prediction unit mode," contribution JVET-L0124 to the 12th JVET meeting, Macao, China, October 2018. GEO được giải thích trong tài liệu sau: S. Esenlik, H. Gao, A. Filippov, V. Rufitskiy, A. M. Kotra, B. Wang, E. Alshina, M. Bläser, và J. Sauer, "Non-CE4: Geometrical partitioning for inter blocks," contribution JVET-O0489 to the 15th JVET meeting, Gothenburg, Sweden, July 2019.

Cách bộc lộ để cân bằng TPM và/hoặc GEO với WP là vô hiệu hóa chúng khi WP được áp dụng. Cách thực hiện thứ nhất được thể hiện trong Bảng 2, liệu giá trị của biến weightedPredFlag có bằng 0 cho CU có được kiểm tra.

Biến weightedPredFlag được dẫn xuất như sau:

– Nếu slice_type bằng P, weightedPredFlag được gán bằng pps_weighted_pred_flag.

– Ngược lại (slice_type bằng B), weightedPredFlag được gán bằng pps_weighted_bipred_flag.

Quá trình dự báo có trọng số có thể được chuyển đổi ở mức ảnh và mức lát, lần lượt sử dụng các phần tử cú pháp pps_weighted_pred_flag và sps_weighted_pred_flag.

Như được bộc lộ trên đây, biến **weightedPredFlag** chỉ báo liệu dự báo có trọng số mức lát có được sử dụng hay không, khi thu thập các mẫu được dự báo ngoài của lát.

Bảng 2. Cú pháp dữ liệu hợp nhất được bộc lộ để cân bằng TPM với WP

merge_data(x0, y0, cbWidth, cbHeight, chType) {	**Mô tả**
if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) || (sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B)))	
regular_merge_flag[x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1){	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
} else {	
if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && weightedPredFlag == 0 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && MaxNumTriangleMergeCand > 1) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
merge_triangle_idx1[x0][y0]	ae(v)
}	
}	
}	
}	
}	

ciip_flag[x0][y0] xác định liệu dự báo trong ảnh và hợp nhất ngoài ảnh kết hợp được áp dụng cho CU hiện tại. Các chỉ số mảng x0, y0 xác định vị trí (x0, y0) của mẫu độ sáng trên bên trái của CB được xem xét so với mẫu độ sáng trên bên trái của ảnh.

Khi ciip_flag[x0][y0] không xuất hiện, nó được suy ra như sau:

– Nếu tất cả các điều kiện sau là đúng, ciip_flag[x0][y0] được suy ra bằng 1:

– sps_ciip_enabled_flag bằng 1.

– general_merge_flag[x0][y0] bằng 1.

– merge_subblock_flag[x0][y0] bằng 0.

– regular_merge_flag[x0][y0] bằng 0.

– cbWidth nhỏ hơn 128.

– cbHeight nhỏ hơn 128.

– cbWidth * cbHeight lớn hơn hoặc bằng 64.

– Ngược lại, ciip_flag[x0][y0] được suy ra bằng 0.

Khi ciip_flag[x0][y0] bằng 1, biến IntraPredModeY[x][y] với x = x0..x0 + cbWidth − 1 và y = y0..y0 + cbHeight − 1 được gán bằng INTRA_PLANAR.

Biến MergeTriangleFlag[x0][y0], xác định liệu bù chuyển động dựa trên hình tam giác có được sử dụng để tạo các mẫu dự báo của CU hiện tại, khi giải mã lát B, được dẫn xuất như sau:

– Nếu tất cả các điều kiện sau là đúng, MergeTriangleFlag[x0][y0] được gán bằng 1:

– sps_triangle_enabled_flag bằng 1.

– slice_type bằng B.

– general_merge_flag[x0][y0] bằng 1.

– MaxNumTriangleMergeCand lớn hơn hoặc bằng 2.

– cbWidth * cbHeight lớn hơn hoặc bằng 64.

– regular_merge_flag[x0][y0] bằng 0.

– merge_subblock_flag[x0][y0] bằng 0.

– ciip_flag[x0][y0] bằng 0.

– weightedPredFlag bằng 0.

– Ngược lại, MergeTriangleFlag[x0][y0] được gán bằng 0.

Cách thực hiện thứ hai xuất hiện trong Bảng 3. Nếu weightedPredFlag bằng 1, phần tử cú pháp **max_num_merge_cand_minus_max_num_triangle_cand** không xuất hiện và được suy ra với giá trị sao cho MaxNumTriangleMergeCand nhỏ hơn 2.

Bảng 3. Cú pháp tiêu đề lát chung được bộc lộ để cân bằng TPM với WP

slice_header() {	**Mô tả**
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag || NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag = = 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL || nal_unit_type == IDR_N_LP || nal_unit_type == CRA_NUT || NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 &&	
(i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	

if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if (slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) || (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	

slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && slice_type == B && !weightedPredFlag && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if (sps_ibc_enabled_flag)	
slice_**six_minus_max_num_ibc_merge_cand**	ue(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if (!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)

if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Cụ thể, các ngữ nghĩa sau có thể được sử dụng cho cách thực hiện thứ hai:

max_num_merge_cand_minus_max_num_triangle_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất tam giác được hỗ trợ trong lát được trừ đi khỏi MaxNumMergeCand.

Khi max_num_merge_cand_minus_max_num_triangle_cand không xuất hiện, và sps_triangle_enabled_flag bằng 1, slice_type bằng B, weightedPredFlag bằng 0, và MaxNumMergeCand lớn hơn hoặc bằng 2, max_num_merge_cand_minus_max_num_triangle_cand được suy ra bằng pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1 + 1.

Khi max_num_merge_cand_minus_max_num_triangle_cand không xuất hiện, và sps_triangle_enabled_flag bằng 1, slice_type bằng B, weightedPredFlag bằng 1, và MaxNumMergeCand lớn hơn hoặc bằng 2, max_num_merge_cand_minus_max_num_triangle_cand được suy ra bằng MaxNumMergeCand or MaxNumMergeCand-1.

Số lượng lớn nhất của các ứng viên chế độ hợp nhất tam giác, MaxNumTriangleMergeCand được dẫn xuất như sau:

MaxNumTriangleMergeCand = MaxNumMergeCand – max_num_merge_cand_minus_max_num_triangle_cand

Khi max_num_merge_cand_minus_max_num_triangle_cand xuất hiện, giá trị của MaxNumTriangleMergeCand phải nằm trong khoảng từ 2 đến MaxNumMergeCand, bao gồm hai giá trị đầu mút.

Khi max_num_merge_cand_minus_max_num_triangle_cand không xuất hiện, và (sps_triangle_enabled_flag bằng 0 hoặc MaxNumMergeCand nhỏ hơn 2), MaxNumTriangleMergeCand được gán bằng 0.

Khi MaxNumTriangleMergeCand bằng 0, chế độ hợp nhất tam giác không được phép cho lát hiện tại.

Các cơ chế được bộc lộ có thể áp dụng cho không chỉ TPM và GEO, mà cả các chế độ phân vùng và dự báo phi chữ nhật khác, chẳng hạn, dự báo trong – ngoài kết hợp với các phân vùng tam giác.

Do TPM và GEO chỉ được áp dụng trong lát B, biến weightedPredFlag theo các phương án thực hiện nêu trên có thể được thay thế trực tiếp bằng biến pps_weighted_bipred_flag.

Cách thực hiện thứ ba được thể hiện trong Bảng 6, liệu giá trị của biến **weightedPredFlag** có bằng 0 hay không đối với CU được kiểm tra.

Biến weightedPredFlag được dẫn xuất như sau:

– Nếu tất cả các điều kiện sau là đúng, weightedPredFlag được gán bằng 0

luma_weight_l0_flag[i] bằng 0 với i từ 0 đến NumRefIdxActive[0]

luma_weight_l1_flag[i] bằng 0 với i từ 0 đến NumRefIdxActive[1]

chroma_weight_l0_flag[i] bằng 0 với i từ 0 đến NumRefIdxActive[0]

chroma _weight_l0_flag[i] bằng 0 với i từ 0 đến NumRefIdxActive[1]

– Ngược lại, weightedPredFlag được gán bằng 1.

Quá trình dẫn xuất weightedPredFlag nghĩa là: nếu tất cả các cờ có trọng số cho các thành phần độ sáng và sắc độ, và đối với tất cả chỉ số tham chiếu của lát hiện tại bằng 0, dự báo có trọng số bị vô hiệu hóa trong lát hiện tại; ngược lại, dự báo có trọng số có thể được sử dụng cho lát hiện tại.

Như nêu trên, biến **weightedPredFlag** chỉ báo liệu dự báo có trọng số mức lát cần được sử dụng khi thu thập các mẫu được dự báo ngoài của lát.

Cách thực hiện thứ tư được thể hiện trong Bảng 2, với weightedPredFlag được thay thế bằng slice_weighted_pred_flag, được báo hiệu trong tiêu đề lát như được thể hiện trên Bảng 4.

Như nêu trên, cú pháp slice_weighted_pred_flag chỉ báo liệu dự báo có trọng số mức lát cần được sử dụng khi thu thập các mẫu được dự báo ngoài của lát.

Bảng 4. Cú pháp tiêu đề lát chung được bộc lộ để báo hiệu cờ dự báo có trọng số mức lát

slice_header() {	**Mô tả**
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag || NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag = = 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL || nal_unit_type == IDR_N_LP || nal_unit_type == CRA_NUT || NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)

delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if (slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) || (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
slice_weighted_pred_flag	u(1)
if (slice_weighted_pred_flag)	
pred_weight_table()	

if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if (sps_ibc_enabled_flag)	
slice **six_minus_max_num_ibc_merge_cand**	ue(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if (!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)

slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Cụ thể là, các ngữ nghĩa sau có thể được sử dụng cho cách thực hiện thứ 4:

slice_weighted_pred_flag bằng 0 xác định rằng dự báo có trọng số không được áp dụng cho lát hiện tại. slice_weighted_pred_flag bằng 1 xác định rằng dự báo có trọng số được áp dụng cho lát hiện tại. Khi không xuất hiện, giá trị của slice_weighted_pred_flag được suy ra bằng 0.

Cách thực hiện thứ 5 là vô hiệu hóa TPM ở mức khối nhờ ràng buộc tương hợp. Trong trường hợp của khối được tạo mã TPM, các hệ số trọng số cho thành phần độ sáng và sắc độ của các ảnh tham chiếu đối với bộ dự báo ngoài P_0 710 và P_1 720 (như được thể hiện trên Fig.7) sẽ không xuất hiện.

Để biết chi tiết, refIdxA và predListFlagA xác định chỉ số tham chiếu và danh sách ảnh tham chiếu của bộ dự báo ngoài P0; refIdxB và predListFlagB xác định chỉ số tham chiếu và danh sách ảnh tham chiếu của bộ dự báo ngoài P1.

Biến lumaWeightedFlag và chromaWeightedFlag được dẫn xuất như sau:

lumaWeightedFlagA = predListFlagA ? luma_weight_l1_flag[refIdxA] : luma_weight_l0_flag[refIdxA]

lumaWeightedFlagB = predListFlagB ? luma_weight_l1_flag[refIdxB] : luma_weight_l0_flag[refIdxB]

chromaWeightedFlagA = predListFlagA ? chroma_weight_l1_flag[refIdxA] : chroma weight_l0_flag[refIdxA]

chromaWeightedFlagB = predListFlagB ? chroma_weight_l1_flag[refIdxB] : chroma weight_l0_flag[refIdxB]

lumaWeightedFlag = lumaWeightedFlagA || lumaWeightedFlagB

chromaWeightedFlag = chromaWeightedFlagA || chromaWeightedFlagB

Tương hợp dòng bit yêu cầu rằng lumaWeightedFlag và chromaWeightedFlag phải bằng 0.

Cách thực hiện thứ sáu là vô hiệu hóa quá trình dự báo mẫu có trọng số kết hợp cho khối được tạo mã TPM khi sử dụng dự báo có trọng số tường minh.

Fig.7 và Fig.8 lần lượt minh họa các ví dụ cho TPM và GEO. Cần lưu ý rằng các phương án thực hiện cho TPM cũng có thể được thực hiện cho chế độ GEO.

Trong trường hợp của khối được tạo mã TPM, nếu các hệ số trọng số cho thành phần độ sáng hoặc sắc độ của ảnh tham chiếu cho bộ dự báo ngoài P_0 710 hoặc P_1 720 xuất hiện, quá trình có trọng số theo các tham số WP (các tham số WP 730 $\{w_0, O_0\}$ và các tham số WP 740 $\{w_1, O_1\}$ cho P_0 và P_1, một cách lần lượt) được sử dụng để tạo khối dự báo ngoài; ngược lại, quá trình có trọng số theo tham số có trọng số kết hợp được sử dụng để tạo bộ dự báo ngoài cho khối 750. Như được thể hiện trên Fig. 9, bộ dự báo ngoài 901 yêu cầu hai khối dự báo P0 911 và P1 912 có diện tích chồng lấn 921 trong đó các trọng số khác 0 được áp dụng cho cả các khối 911 và 912 để kết hợp một phần các bộ dự báo P0 911 và P1 912. Các khối liền kề với khối 901 được ký hiệu là 931, 932, 933, 934, 935, và 936 trên Fig.9. Fig.8 minh họa khác biệt giữa các chế độ hợp nhất TPM và GEO. Trong trường hợp của chế độ hợp nhất GEO, diện tích chồng lấn giữa các bộ dự báo 851 và 852 có thể được đặt không chỉ dọc các đường chéo của khối được dự báo ngoài 850. Các bộ dự báo P0 851 và P1 852 có thể được nhận bằng cách sao chép các khối 810 và 820 ngoài các ảnh khác có hoặc không áp dụng các trọng số và các độ lệch $\{w_0, O_0\}$ 830 và $\{w_1, O_1\}$ 840 cho các khối 810 và 820, một cách lần lượt.

Trong ví dụ, refIdxA và predListFlagA xác định chỉ số tham chiếu và danh sách ảnh tham chiếu của bộ dự báo ngoài P0; refIdxB và predListFlagB xác định chỉ số tham chiếu và danh sách ảnh tham chiếu của bộ dự báo ngoài P1.

Biến lumaWeightedFlag và chromaWeightedFlag được dẫn xuất như sau:

lumaWeightedFlagA = predListFlagA ? luma_weight_l1_flag[refIdxA] : luma_weight_l0_flag[refIdxA]

lumaWeightedFlagB = predListFlagB ? luma_weight_l1_flag[refIdxB] : luma_weight_l0_flag[refIdxB]

chromaWeightedFlagA = predListFlagA ? chroma_weight_l1_flag[refIdxA] : chroma weight_l0_flag[refIdxA]

chromaWeightedFlagB = predListFlagB ? chroma_weight_l1_flag[refIdxB] : chroma weight_l0_flag[refIdxB]

lumaWeightedFlag = lumaWeightedFlagA | | lumaWeightedFlagB

chromaWeightedFlag = chromaWeightedFlagA | | chromaWeightedFlagB

Sau đó, nếu lumaWeightedFlag true, quá trình có trọng số tường minh được gọi; nếu lumaWeightedFlag false, quá trình có trọng số kết hợp được gọi. Thành phần sắc độ được quyết định bằng chromaWeightedFlag.

Đối với cách thực hiện khác, cờ có trọng số cho tất cả các thành phần được xem xét đồng thời. Nếu một trong lumaWeightedFlag hoặc chromaWeightedFlag true, quá trình có trọng số tường minh được gọi; nếu cả lumaWeightedFlag và chromaWeightedFalg false, quá trình có trọng số kết hợp được gọi.

Quá trình có trọng số tường minh đối với khối chữ nhật được dự báo nhờ sử dụng cơ chế dự báo kép, được thực hiện như được nêu dưới đây.

Đầu vào cho quá trình này là:

– Hai biến nCbW và nCbH xác định chiều rộng và chiều cao của CB hiện tại,
– Hai mảng (nCbW)x(nCbH) predSamplesA và predSamplesB,
– Các cờ danh sách dự báo, predListFlagA và predListFlagB,
– Các chỉ số tham chiếu, refIdxA và refIdxB,
– biến cIdx xác định chỉ số thành phần màu,
– chiều sâu bit mẫu, bitDepth.

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

Biến shift1 được gán bằng Max(2, 14 − bitDepth).

Các biến log2Wd, o0, o1, w0 và w1 được dẫn xuất như saus:

– Nếu cIdx bằng 0 cho các mẫu độ sáng, phương trình sau áp dụng:

log2Wd = luma_log2_weight_denom + shift1

w0 = predListFlagA ? LumaWeightL1[refIdxA] : LumaWeightL0[refIdxA]

w1 = predListFlagB ? LumaWeightL1[refIdxB] : LumaWeightL0[refIdxB]

o0 = (predListFlagA ? luma_offset_l1[refIdxA] : luma_offset_l0[refIdxA]) << ($BitDepth_Y$ − 8)

o1 = (predListFlagB ? luma_offset_l1[refIdxB] : luma_offset_l0[refIdxB]) << ($BitDepth_Y$ − 8)

– Ngược lại (cIdx không bằng 0 cho các mẫu sắc độ), phương trình sau áp dụng:

log2Wd = ChromaLog2WeightDenom + shift1

w0 = predListFlagA ? ChromaWeightL1[refIdxA][cIdx − 1] : ChromaWeightL0[refIdxA][cIdx − 1]

w1 = predListFlagA ? ChromaWeightL1[refIdxB][cIdx − 1] : ChromaWeightL0[refIdxB][cIdx − 1]

o0 = (predListFlagA ? ChromaOffsetL1[refIdxA][cIdx − 1] : ChromaOffsetL0[refIdxA][cIdx − 1]) << ($BitDepth_C$ − 8)

o1 = (predListFlagB ? ChromaOffsetL1[refIdxB][cIdx − 1] : ChromaOffsetL0[refIdxB][cIdx − 1]) << ($BitDepth_C$ − 8)

Mẫu dự báo pbSamples[x][y] với x = 0..nCbW − 1 và y = 0..nCbH − 1 được dẫn xuất như sau:

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1,(predSamplesA[x][y] * w0 + predSamplesB[x][y] * w1 +((o0 + o1 + 1) << log2Wd)) >> (log2Wd + 1))

Các tham số của dự báo có trọng số mức lát có thể được biểu diễn dưới dạng tập hợp biến, được gán cho mỗi phần tử của danh sách ảnh tham chiếu. Chỉ số của phần tử được ký hiệu thêm là "*i*". Các tham số này có thể bao gồm:

- LumaWeightL0[*i*]

- luma_offset_l0[*i*] là độ lệch cộng sinh được áp dụng cho giá trị dự báo độ sáng để dự báo danh sách 0 nhờ sử dụng RefPicList[0][i]. Giá trị của luma_offset_l0[i] phải nằm trong khoảng từ −128 đến 127, bao gồm hai giá trị

đầu mút. Khi luma_weight_l0_flag[i] bằng 0, luma_offset_l0[i] được suy ra bằng 0.

Biến LumaWeightL0[i] được dẫn xuất bằng (1 << luma_log2_weight_denom) + delta_luma_weight_l0[i]. Khi luma_weight_l0_flag[i] bằng 1, giá trị của delta_luma_weight_l0[i] phải nằm trong khoảng từ −128 đến 127, bao gồm hai giá trị đầu mút. Khi luma_weight_l0_flag[i] bằng 0, LumaWeightL0[i] được suy ra bằng $2^{luma\_log2\_weight\_denom}$.

Quá trình có trọng số kết hợp đối với khối chữ nhật được dự báo nhờ sử dụng cơ chế dự báo kép, quá trình sau được thực hiện như được nêu dưới đây.

Đầu vào cho quá trình này là:

– Hai biến nCbW và nCbH xác định chiều rộng và chiều cao của CB hiện tại,

– Hai mảng (nCbW)x(nCbH) predSamplesLA và predSamplesLB,

– biến triangleDir xác định hướng phân vùng,

– biến cIdx xác định chỉ số thành phần màu.

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

Biến nCbR được dẫn xuất như sau:

nCbR = (nCbW > nCbH) ? (nCbW/nCbH) : (nCbH/nCbW)

Biến bitDepth được dẫn xuất như sau:

– Nếu cIdx bằng 0, bitDepth được gán bằng $BitDepth_Y$.

– Ngược lại, bitDepth được gán bằng $BitDepth_C$.

Các biến shift1 và offset1 được dẫn xuất như saus:

– Biến shift1 được gán bằng Max(5, 17 − bitDepth).

– Biến offset1 được gán bằng 1 << (shift1 − 1).

Tùy thuộc vào các giá trị của triangleDir, wS và cIdx, các mẫu dự báo pbSamples[x][y] với x = 0..nCbW − 1 và y = 0..nCbH − 1 được dẫn xuất như saus:

– Biến wIdx được dẫn xuất như sau:

– Nếu cIdx bằng 0 và triangleDir bằng 0, phương trình sau áp dụng:

wIdx = (nCbW > nCbH) ? (Clip3(0, 8, (x/nCbR − y) + 4))
: (Clip3(0, 8, (x − y/nCbR) + 4))

– Ngược lại, nếu cIdx bằng 0 và triangleDir bằng 1, phương trình sau áp dụng:

wIdx = (nCbW > nCbH) ? (Clip3(0, 8, (nCbH − 1 − x/nCbR − y) + 4))
(Clip3(0, 8, (nCbW − 1 − x − y/nCbR) + 4))

– Ngược lại, nếu cIdx lớn hơn 0 và triangleDir bằng 0, phương trình sau áp dụng:

wIdx = (nCbW > nCbH) ? (Clip3(0, 4, (x/nCbR − y) + 2))
: (Clip3(0, 4, (x − y/nCbR) + 2))

– Ngược lại (nếu cIdx lớn hơn 0 và triangleDir bằng 1), phương trình sau áp dụng:

wIdx = (nCbW > nCbH) ? (Clip3(0, 4, (nCbH − 1 − x/nCbR − y) + 2))
(Clip3(0, 4, (nCbW − 1 − x − y/nCbR) + 2))

– Biến wValue xác định trọng số của mẫu dự báo được dẫn xuất nhờ sử dụng wIdx và cIdx như sau:

wValue = (cIdx = = 0) ? Clip3(0, 8, wIdx) : Clip3(0, 8, wIdx * 2)

– Các giá trị mẫu dự báo được dẫn xuất như saus:

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesLA[x][y] * wValue +
predSamplesLB[x][y] * (8 − wValue) + offset1) >> shift1)

Đối với chế độ hình học, quá trình có trọng số kết hợp đối với khối chữ nhật được dự báo nhờ sử dụng cơ chế dự báo kép, quá trình sau được thực hiện như được nêu dưới đây.

Đầu vào cho quá trình này là:

– Hai biến nCbW và nCbH xác định chiều rộng và chiều cao của CB hiện tại,

– Hai mảng (nCbW)x(nCbH) predSamplesLA và predSamplesLB,

– biến angleIdx xác định chỉ số gọc của phân vùng hình học,

– biến distanceIdx xác định khoảng cách idx của phân vùng hình học,

– biến cIdx xác định chỉ số thành phần màu.

Đầu ra của quá trình này là mảng (nCbW)x(nCbH) pbSamples của các giá trị mẫu dự báo và biến partIdx.

Biến bitDepth được dẫn xuất như sau:

– Nếu cIdx bằng 0, bitDepth được gán bằng $BitDepth_Y$.

– Ngược lại, bitDepth được gán bằng $BitDepth_C$.

Các biến shift1 và offset1 được dẫn xuất như saus:

– Biến shift1 được gán bằng Max(5, 17 − bitDepth).

– Biến offset1 được gán bằng 1 << (shift1 − 1).

Mảng trọng số $sampleWeight_L[x][y]$ cho độ sáng và $sampleWeight_C[x][y]$ cho sắc độ với x = 0..nCbW − 1 và y = 0..nCbH − 1 được dẫn xuất như saus:

Giá trị của các biến sau được thiết lập:

– hwRatio được gán bằng nCbH/nCbW

– displacementX được gán bằng angleIdx

– displacementY được gán bằng (displacementX + 8)%32

– partIdx được gán bằng angleIdx >=13 && angleIdx <=27 ? 1 : 0

– rho được gán bằng giá trị sau nhờ sử dụng các bảng tra cứu được ký hiệu là Dis, được xác định trong bảng 8-12:

rho = (Dis[displacementX]<< 8) + (Dis[displacementY] << 8)

Nếu một trong các điều kiện sau là đúng, biến shiftHor được gán bằng 0:

angleIdx % 16 bằng 8,

angleIdx % 16 không bằng 0 và hwRatio ≥ 1

Ngược lại, shiftHor được gán bằng 1.

Nếuf shiftHor bằng 0, offsetX và offsetY được dẫn xuất như sau:

offsetX = (256 − nCbW) >> 1,

offsetY = (256 − nCbH) >> 1 + angleIdx < 16 ? (distanceIdx * nCbH) >> 3 : −((distanceIdx * nCbH) >> 3)

Ngược lại, nếu shiftHor bằng 1, offsetX và offsetY được dẫn xuất như saus:

offsetX = (256 − nCbW) >> 1 + angleIdx < 16 ? (distanceIdx * nCbW) >> 3 : −((distanceIdx * nCbW) >> 3)

offsetY = (256 − nCbH) >> 1

Biến weightIdx và weightIdxAbs được tính toán nhờ sử dụng bảng tra cứu 9 với x = 0..nCbW − 1 và y = 0..nCbH − 1 như sau:

weightIdx = (((x + offsetX)<<1) + 1)*Dis[displacementX]
+ (((y + offsetY)<<1) + 1))*Dis[displacementY] − rho.

weightIdxAbs = Clip3(0, 26, abs(weightIdx)).

Giá trị của $sampleWeight_L[x][y]$ với x = 0..nCbW – 1 và y = 0..nCbH – 1 được gán theo Bảng 10 được ký hiệu là GeoFilter:

$sampleWeight_L[x][y]$ = weightIdx <= 0 ? GeoFilter[weightIdxAbs] : 8 – GeoFilter[weightIdxAbs]

Giá trị $sampleWeight_C[x][y]$ với x = 0..nCbW – 1 và y = 0..nCbH – 1 được gán như sau:

$sampleWeight_C[x][y]$ = $sampleWeight_L$[(x<< (SubWidthC – 1))][(y<< (SubHeightC – 1))]

Lưu ý – Giá trị của mẫu $sampleWeight_L[x][y]$ cũng có thể được dẫn xuất từ $sampleWeight_L$[x -shiftX][y-shiftY]. Nếu angleIdx lớn hơn 4 và nhỏ hơn 12, hoặc angleIdx lớn hơn 20 và nhỏ hơn 24, shiftX là tang của góc tách và shiftY bằng 1, ngược lại shiftX bằng 1 của góc tách và shiftY là cotang của góc tách. Nếu giá trị tang (resp. cotangent) bằng vô cực, shiftX bằng 1 (resp. 0) hoặc shift Y bằng 0 (reps. 1).

Các giá trị mẫu dự báo được dẫn xuất như sau với X được ký hiệu là L hoặc C với cIdx bằng 0 hoặc không bằng 0 :

pbSamples[x][y] = partIdx ?

Clip3(0, (1 << bitDepth) – 1, (predSamplesLA[x][y] * (8 – $sampleWeight_X$[x][y]) + predSamplesLB[x][y] * $sampleWeight_X$ [x][y] + offset1) >> shift1)

: Clip3(0, (1 << bitDepth) – 1, (predSamplesLA[x][y] * $sampleWeight_X$ [x][y] + predSamplesLB[x][y] * (8 – $sampleWeight_X$[x][y]) + offset1) >> shift1)

Bảng 5 – Bảng tra cứu Dis để dẫn xuất khoảng cách phân vùng hình học.

idx	0	1	2	4	6	7	8	9	10	12	14	15
Dis[idx]	8	8	8	8	4	2	0	-2	-4	-8	-8	-8
idx	16	17	18	20	22	23	24	25	26	28	30	31
Dis[idx]	-8	-8	-8	-8	-4	-2	0	2	4	8	8	8

Bảng 6- Bảng tra cứu trọng số lọc GeoFilter để dẫn xuất các trọng số lọc phân vùng hình học.

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
GeoFilter[idx]	4	4	4	4	5	5	5	5	5	5	5	6	6	6
idx	14	15	16	17	18	19	20	21	22	23	24	25	26	
GeoFilter[idx]	6	6	6	6	7	7	7	7	7	7	7	7	8	

Trong đặc tả VVC Draft 7 (document JVET-P2001-vE: B. Bross, J. Chen, S. Liu, Y.- K. Wang, "Versatile Video Coding (Draft 7)," tài liệu đầu ra JVET-P2001 of the 16th JVET meeting, Geneva, Switzerland; tài liệu này có trong tệp tin JVET-P2001-v14: http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/16_Geneva/wg11/JVET-P2001-v14.zip), khái niệm tiêu đề ảnh (PH) được đưa vào bằng cách di chuyển một phần của các phần tử cú pháp ra khỏi tiêu đề lát (slice header, SH) đến PH để giảm phụ tải báo hiệu gây ra bằng cách gán các giá trị bằng nhau hoặc giống nhau cho các phần tử cú pháp giống nhau trong mỗi SH được liên kết với PH. Như được nêu trong Bảng 7, các phần tử cú pháp để điều khiển số lượng lớn nhất của các ứng viên hợp nhất cho chế độ hợp nhất TPM được báo hiệu trong PH, trong đó các tham số dự báo có trọng số vẫn ở SH như được thể hiện trên Bảng 8 và Bảng 10. Ngữ nghĩa của các phần tử cú pháp được sử dụng trong Bảng 8 và Bảng 9 được mô tả dưới đây.

Bảng 7 – Cú pháp RBSP tiêu đề ảnh

picture_header_rbsp() {	**Mô tả**
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rbsp_trailing_bits()	
}	

Ngữ nghĩa RBSP tiêu đề ảnh

PH chứa thông tin chung cho tất cả các lát của ảnh được tạo mã được liên kết với PH.

non_reference_picture_flag bằng 1 xác định ảnh được liên kết với PH chưa bao giờ được sử dụng làm ảnh tham chiếu. non_reference_picture_flag bằng 0 xác định ảnh được liên kết với PH có thể hoặc không thể được sử dụng làm ảnh tham chiếu.

gdr_pic_flag bằng 1 xác định ảnh được liên kết với PH là ảnh làm mới giải mã dần dần (gradual decoding refresh, GDR). gdr_pic_flag bằng 0 xác định rằng ảnh được liên kết với PH không phải là ảnh GDR.

no_output_of_prior_pics_flag ảnh hưởng đầu ra của các ảnh được giải mã trước đó trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB) sau khi giải mã ảnh bắt đầu cuối viđeo lớp được tạo mã (coded layer video sequence start, CLVSS) không phải là ảnh thứ nhất trong dòng bit.

recovery_poc_cnt xác định điểm khôi phục của các ảnh được giải mã theo thứ tự đầu ra. Nếu ảnh hiện tại là ảnh GDR được liên kết với PH và có ảnh picA đi theo ảnh GDR hiện tại theo thứ tự giải mã trong chuỗi viđeo lớp được tạo mã (coded layer video sequence, CLVS) và có PicOrderCntVal bằng PicOrderCntVal của ảnh GDR hiện tại cộng giá trị của recovery_poc_cnt, ảnh picA được gọi là ảnh điểm khôi phục. Ngược lại, ảnh thứ nhất theo thứ tự đầu ra có PicOrderCntVal lớn hơn PicOrderCntVal của ảnh hiện tại cộng giá trị của recovery_poc_cnt được gọi là ảnh điểm khôi phục. Ảnh điểm khôi phục sẽ không đứng trước ảnh GDR hiện tại theo thứ tự giải mã. Giá trị của recovery_poc_cnt phải nằm trong khoảng từ 0 đến MaxPicOrderCntLsb − 1, bao gồm hai giá trị đầu mút.

Lưu ý 1 – Khi gdr_enabled_flag bằng 1 và PicOrderCntVal của ảnh hiện tại lớn hơn hoặc bằng RpPicOrderCntVal của ảnh GDR được liên kết, các ảnh được giải mã hiện tại và tiếp theo trong thứ tự đầu ra được so khớp chính xác với các ảnh tương ứng được tạo bằng cách bắt đầu quá trình giải mã từ ảnh điểm truy nhập ngẫu nhiên nội vùng (intra random access point, IRAP) trước đó, khi xuất hiện, đứng trước ảnh GDR liên kết theo thứ tự giải mã.

ph_pic_parameter_set_id xác định giá trị của pps_pic_parameter_set_id với PPS đang sử dụng. Giá trị của ph_pic_parameter_set_id phải nằm trong khoảng từ 0 đến 63, bao gồm hai giá trị đầu mút.

Tương hợp dòng bit yêu cầu rằng giá trị của TemporalId của PH phải lớn hơn hoặc bằng giá trị của TemporalId của PPS có pps_pic_parameter_set_id bằng ph_pic_parameter_set_id.

sps_poc_msb_flag bằng 1 xác định rằng phần tử cú pháp ph_poc_msb_cycle_present_flag xuất hiện trong các PH viện dẫn đến SPS. sps_poc_msb_flag bằng 0 xác định rằng phần tử cú pháp ph_poc_msb_cycle_present_flag không xuất hiện trong các PH viện dẫn đến SPS.

ph_poc_msb_present_flag bằng 1 xác định rằng phần tử cú pháp poc_msb_val xuất hiện trong PH. ph_poc_msb_present_flag bằng 0 xác định rằng phần tử cú pháp poc_msb_val không xuất hiện trong PH. Khi vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]] bằng 0 và có ảnh trong đơn vị truy nhập (Access Unit, AU) hiện tại trong lớp tham chiếu của lớp hiện tại, giá trị của ph_poc_msb_present_flag phải bằng 0.

poc_msb_val xác định giá trị bit có trọng số lớn nhất (most significant bit, MSB) của số đếm thứ tự ảnh (picture order count, POC) của ảnh hiện tại. Chiều dài của phần tử cú pháp poc_msb_val bằng poc_msb_len_minus1 + 1 bit.

sps_triangle_enabled_flag xác định liệu bù chuyển động dựa trên hình tam giác có thể được sử dụng cho dự báo ngoài. sps_triangle_enabled_flag bằng 0 xác định rằng cú pháp phải bị giới hạn sao cho không phép bù chuyển động dựa trên hình tam giác nào được sử dụng trong CLVS, và merge_triangle_split_dir, merge_triangle_idx0, và merge_triangle_idx1 không xuất hiện trong cú pháp CU của CLVS. sps_triangle_enabled_flag bằng 1 xác định rằng bù chuyển động dựa trên hình tam giác có thể được sử dụng trong CLVS.

pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 bằng 0 xác định rằng pic_max_num_merge_cand_minus_max_num_triangle_cand xuất hiện trong các PH của các lát viện dẫn đến PPS. pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 lớn hơn 0 xác định rằng pic_max_num_merge_cand_minus_max_num_triangle_cand không xuất hiện trong các PH viện dẫn đến PPS. Giá trị của

pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 phải nằm trong khoảng từ 0 đến MaxNumMergeCand − 1.

pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 bằng 0 xác định rằng pic_max_num_merge_cand_minus_max_num_triangle_cand xuất hiện trong các PH của các lát viện dẫn đến PPS. pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 lớn hơn 0 xác định rằng pic_max_num_merge_cand_minus_max_num_triangle_cand không xuất hiện trong các PH viện dẫn đến PPS. Giá trị của pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 phải nằm trong khoảng từ 0 đến MaxNumMergeCand − 1.

pic_six_minus_max_num_merge_cand xác định số lượng lớn nhất của các ứng viên MVP hợp nhất được hỗ trợ trong các lát được liên kết với PH được trừ đi khỏi 6. Số lượng lớn nhất của các ứng viên MVP hợp nhất, MaxNumMergeCand được dẫn xuất như sau:

MaxNumMergeCand = 6 − picsix_minus_max_num_merge_cand

Giá trị của MaxNumMergeCand phải nằm trong khoảng từ 1 đến 6, bao gồm hai giá trị đầu mút. Khi không xuất hiện, giá trị của pic_six_minus_max_num_merge_cand được suy ra bằng pps_six_minus_max_num_merge_cand_plus1 − 1.

Bảng 8 – Cú pháp tiêu đề lát chung

slice_header() {	**Mô tả**
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag || NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag &&((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 &&	
(i == 0 || (i == 1 && rpl1_idx_present_flag)))	

slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(pic_rpl_present_flag || ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(slice_type != I) {	
...	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
}	
...	
byte_alignment()	
}	

Ngữ nghĩa tiêu đề lát chung

Khi xuất hiện, giá trị của phần tử cú pháp tiêu đề lát slice_pic_order_cnt_lsb sẽ giống nhau trong tất cả các tiêu đề lát của ảnh được tạo mã.

Biến CuQpDeltaVal, xác định khác biệt giữa tham số lượng tử độ sáng cho CU chứa cu_qp_delta_abs và dự báo của nó, được gán bằng 0. Các biến $CuQpOffset_{Cb}$, $CuQpOffset_{Cr}$, và $CuQpOffset_{CbCr}$, xác định các giá trị cần được sử dụng khi xác định các giá trị tương ứng của các tham số lượng tử Qp'_{Cb}, Qp'_{Cr}, và Qp'_{CbCr} cho CU chứa cu_chroma_qp_offset_flag, đều được gán bằng 0.

slice_pic_order_cnt_lsb xác định modulo POC MaxPicOrderCntLsb cho ảnh hiện tại. Chiều dài của phần tử cú pháp slice_pic_order_cnt_lsb là log2_max_pic_order_cnt_lsb_minus4 + 4 bit. Giá trị của

slice_pic_order_cnt_lsb phải nằm trong khoảng từ 0 đến MaxPicOrderCntLsb − 1, bao gồm hai đầu mút.

Khi ảnh hiện tại là ảnh GDR, biến RpPicOrderCntVal được dẫn xuất như sau:

RpPicOrderCntVal = PicOrderCntVal + recovery_poc_cnt.

slice_subpic_id xác định định danh ảnh thành phần của ảnh thành phần chứa lát. Nếu slice_subpic_id xuất hiện, giá trị của biến SubPicIdx được dẫn xuất sao cho SubpicIdList[SubPicIdx] bằng slice_subpic_id. Ngược lại (slice_subpic_id không xuất hiện), biến SubPicIdx được dẫn xuất bằng 0. Chiều dài của slice_subpic_id, in bits, được dẫn xuất như sau:

– Nếu sps_subpic_id_signalling_present_flag bằng 1, chiều dài của slice_subpic_id bằng sps_subpic_id_len_minus1 + 1.

– Ngược lại, nếu ph_subpic_id_signalling_present_flag bằng 1, chiều dài của slice_subpic_id bằng ph_subpic_id_len_minus1 + 1.

– Ngược lại, nếu pps_subpic_id_signalling_present_flag bằng 1, chiều dài của slice_subpic_id bằng pps_subpic_id_len_minus1 + 1.

– Ngược lại, chiều dài của slice_subpic_id bằng Ceil(Log2 (sps_num_subpics_minus1 + 1)).

slice_address xác định địa chỉ lát của lát. Khi không xuất hiện, giá trị của slice_address được suy ra bằng 0.

Nếu rect_slice_flag bằng 0, phương trình sau áp dụng:

– Địa chỉ lát là chỉ số ô quét mành.

– Chiều dài của slice_address bằng Ceil(Log2 (NumTilesInPic)) bit.

– Giá trị của slice_address phải nằm trong khoảng từ 0 đến NumTilesInPic − 1, bao gồm hai giá trị đầu mút.

Ngược lại (rect_slice_flag bằng 1), phương trình sau áp dụng:

– Địa chỉ lát là chỉ số lát của lát trong ảnh thành phần thứ SubPicIdx.

– Chiều dài của slice_address bằng Ceil(Log2(NumSlicesInSubpic[SubPicIdx])) bit.

– Giá trị của slice_address phải nằm trong khoảng từ 0 đến NumSlicesInSubpic[SubPicIdx] − 1, bao gồm hai giá trị đầu mút.

Tương hợp dòng bit yêu cầu rằng áp dụng các ràng buộc sau:

– Nếu rect_slice_flag bằng 0 hoặc subpics_present_flag bằng 0, giá trị của slice_address sẽ không bằng giá trị của slice_address của đơn vị NAL có lát được tạo mã khác bất kỳ của ảnh được tạo mã tương tự.

– Ngược lại, cặp giá trị slice_subpic_id và slice_address sẽ không bằng cặp giá trị slice_subpic_id và slice_address của bất kỳ đơn vị NAL lát được tạo mã khác của cùng ảnh được tạo mã.

– Khi rect_slice_flag bằng 0, các lát của ảnh sẽ theo thứ tự tăng của các giá trị slice_address của nó.

– Các hình dạng của các lát của ảnh phải sao cho mỗi CTU, khi được giải mã, sẽ phải có toàn bộ đường biên bên trái và toàn bộ đường biên trên gồm đường biên ảnh hoặc gồm các đường biên của (các) CTU được giải mã trước đó.

num_tiles_in_slice_minus1 cộng 1, khi xuất hiện, xác định số lượng ô trong lát. Giá trị của num_tiles_in_slice_minus1 phải nằm trong khoảng từ 0 đến NumTilesInPic − 1, bao gồm hai giá trị đầu mút.

Biến NumCtuInCurrSlice, xác định số lượng CTU trong lát hiện tại, và danh sách CtbAddrInCurrSlice[i], với i nằm trong khoảng từ 0 đến NumCtuInCurrSlice − 1, bao gồm hai giá trị đầu mút, xác định địa chỉ quét mành ảnh của CTB thứ i trong lát, được dẫn xuất như saus:

```
if(rect_slice_flag) {
    picLevelSliceIdx = SliceSubpicToPicIdx[SubPicIdx][slice_address]
    NumCtuInCurrSlice = NumCtuInSlice[picLevelSliceIdx]
    for(i = 0; i < NumCtuInCurrSlice; i++)
        CtbAddrInCurrSlice[i] = CtbAddrInSlice[picLevelSliceIdx][i]
} else {
    NumCtuInCurrSlice = 0
    for(tileIdx = slice_address; tileIdx <= slice_address + num_tiles_in_slice_minus1[i]; tileIdx++) {
        tileX = tileIdx % NumTileColumns
        tileY = tileIdx/NumTileColumns
        for(ctbY = tileRowBd[tileY]; ctbY < tileRowBd[tileY + 1]; ctbY++) {
            for(ctbX = tileColBd[tileX]; ctbX < tileColBd[tileX + 1]; ctbX++) {
                CtbAddrInCurrSlice[NumCtuInCurrSlice] = ctbY * PicWidthInCtb + ctbX
                NumCtuInCurrSlice++
```

```
            }
        }
    }
}
```

Các biến SubPicLeftBoundaryPos, SubPicTopBoundaryPos, SubPicRightBoundaryPos, và SubPicBotBoundaryPos được dẫn xuất như saus:

```
if(subpic_treated_as_pic_flag[SubPicIdx]) {
    SubPicLeftBoundaryPos = subpic_ctu_top_left_x[SubPicIdx] * CtbSizeY
    SubPicRightBoundaryPos = Min(pic_width_max_in_luma_samples − 1,
    (subpic_ctu_top_left_x[SubPicIdx] + subpic_width_minus1[SubPicIdx] + 1) * CtbSizeY − 1)
SubPicTopBoundaryPos = subpic_ctu_top_left_y[SubPicIdx] *CtbSizeY SubPicBotBoundaryPos =
Min(pic_height_max_in_luma_samples − 1,
    (subpic_ctu_top_left_y[SubPicIdx] + subpic_height_minus1[SubPicIdx] + 1) * CtbSizeY − 1
)
}
```

slice_type xác định loại mã của lát theo Bảng 9.

Bảng 9 – Liên kết tên với slice_type

slice_type	Tên của slice_type
0	B (B slice)
1	P (P slice)
2	I (I slice)

slice_rpl_sps_flag[i] bằng 1 xác định rằng danh sách ảnh tham chiếu i của lát hiện tại được dẫn xuất dựa trên một trong các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) với listIdx bằng i trong SPS. slice_rpl_sps_flag[i] bằng 0 xác định rằng danh sách ảnh tham chiếu i của lát hiện tại được dẫn xuất dựa trên cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) với listIdx bằng i được bao gồm trực tiếp trong các tiêu đề lát của ảnh hiện tại.

Khi slice_rpl_sps_flag[i] không xuất hiện, phương trình sau áp dụng:

– Nếu pic_rpl_present_flag bằng 1, giá trị của slice_rpl_sps_flag[i] được suy ra bằng pic_rpl_sps_flag[i].

– Ngược lại, nếu num_ref_pic_lists_in_sps[i] bằng 0, giá trị của ref_pic_list_sps_flag[i] được suy ra bằng 0.

– Ngược lại, nếu num_ref_pic_lists_in_sps[i] lớn hơn 0 và nếu rpl1_idx_present_flag bằng 0, giá trị của slice_rpl_sps_flag[1] được suy ra bằng slice_rpl_sps_flag[0].

slice_rpl_idx[i] xác định chỉ số, vào danh sách của các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) với listIdx bằng i được bao gồm trong SPS, của cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) với listIdx bằng i được sử dụng để dẫn xuất danh sách ảnh tham chiếu i của ảnh hiện tại. Phần tử cú pháp slice_rpl_idx[i] được biểu diễn bởi Ceil(Log2(num_ref_pic_lists_in_sps[i])) bit. Khi không xuất hiện, giá trị của slice_rpl_idx[i] được suy ra bằng 0. Giá trị của slice_rpl_idx[i] phải nằm trong khoảng từ 0 đến num_ref_pic_lists_in_sps[i] − 1, bao gồm hai giá trị đầu mút. Khi slice_rpl_sps_flag[i] bằng 1 và num_ref_pic_lists_in_sps[i] bằng 1, giá trị của slice_rpl_idx[i] được suy ra bằng 0. Khi slice_rpl_sps_flag[i] bằng 1 và rpl1_idx_present_flag bằng 0, giá trị của slice_rpl_idx[1] được suy ra bằng slice_rpl_idx[0].

Biến RplsIdx[i] được dẫn xuất như sau:

```
if(pic_rpl_present_flag)
    RplsIdx[i] = PicRplsIdx[i]
else
    RplsIdx[i] = slice_rpl_sps_flag[i] ? slice_rpl_idx[i] : num_ref_pic_lists_in_sps[i]
```

slice_poc_lsb_lt[i][j] xác định giá trị của POC modulo MaxPicOrderCntLsb của mục LTRP thứ j trong danh sách ảnh tham chiếu thứ i. Chiều dài của phần tử cú pháp slice_poc_lsb_lt[i][j] bằng log2_max_pic_order_cnt_lsb_minus4 + 4 bit.

Biến PocLsbLt[i][j] được dẫn xuất như sau:

```
if(pic_rpl_present_flag)
    PocLsbLt[i][j] = PicPocLsbLt[i][j]
else
    PocLsbLt[i][j] = ltrp_in_slice_header_flag[i][RplsIdx[i]] ?
            slice_poc_lsb_lt[i][j] : rpls_poc_lsb_lt[listIdx][RplsIdx[i]][j]
```

slice_delta_poc_msb_present_flag[i][j] bằng 1 xác định rằng slice_delta_poc_msb_cycle_lt[i][j] xuất hiện.

slice_delta_poc_msb_present_flag[i][j] bằng 0 xác định rằng slice_delta_poc_msb_cycle_lt[i][j] không xuất hiện.

Giả sử prevTid0Pic là ảnh đứng trước trong thứ tự giải mã có nuh_layer_id giống như ảnh hiện tại, có TemporalId bằng 0, và không phải là ảnh đứng đầu bỏ qua truy nhập ngẫu nhiên (Random Access Skipped Leading, RASL) hặc ảnh đứng đầu có thể giải mã truy nhập ngẫn nhiên (Random Access Decodable Leading, RADL). Giả sử setOfPrevPocVals là tập hợp chứa các tham số sau:

– PicOrderCntVal của prevTid0Pic,

– PicOrderCntVal của mỗi ảnh được viện dẫn bởi các mục trong RefPicList[0] hoặc RefPicList[1] của prevTid0Pic và có nuh_layer_id giống như ảnh hiện tại,

– PicOrderCntVal của mỗi ảnh theo sau prevTid0Pic trong thứ tự giải mã, có nuh_layer_id giống như ảnh hiện tại, và đứng trước ảnh hiện tại trong thứ tự giải mã.

Khi pic_rpl_present_flag bằng 0 và có nhiều hơn một giá trị trong setOfPrevPocVals mà giá trị modulo MaxPicOrderCntLsb bằng PocLsbLt[i][j], giá trị của slice_delta_poc_msb_present_flag[i][j] phải bằng 1.

slice_delta_poc_msb_cycle_lt[i][j] xác định giá trị của biến FullPocLt[i][j] như sau:

```
if(pic_rpl_present_flag)
    FullPocLt[i][j] = PicFullPocLt[i][j]
else {
    if(j == 0)
      DeltaPocMsbCycleLt[i][j] = delta_poc_msb_cycle_lt[i][j]
    else
      DeltaPocMsbCycleLt[i][j] = delta_poc_msb_cycle_lt[i][j] +
DeltaPocMsbCycleLt[i][j − 1]
    FullPocLt[i][j] = PicOrderCntVal − DeltaPocMsbCycleLt[i][j] * MaxPicOrderCntLsb
−(PicOrderCntVal & (MaxPicOrderCntLsb − 1)) + PocLsbLt[i][j]
}
```

Giá trị của slice_delta_poc_msb_cycle_lt[i][j] phải nằm trong khoảng từ 0 đếm $2^{(32 - \text{log2\_max\_pic\_order\_cnt\_lsb\_minus4} - 4)}$, bao gồm hai giá trị đầu mút. Khi không xuất hiện, giá trị của slice_delta_poc_msb_cycle_lt[i][j] được suy ra bằng 0.

num_ref_idx_active_override_flag bằng 1 xác định rằng phần tử cú pháp num_ref_idx_active_minus1[0] xuất hiện đối với các lát P và B và phần tử cú pháp num_ref_idx_active_minus1[1] xuất hiện cho các lát B. num_ref_idx_active_override_flag bằng 0 xác định rằng các phần tử cú pháp num_ref_idx_active_minus1[0] và num_ref_idx_active_minus1[1] không xuất hiện. Khi không xuất hiện, giá trị của num_ref_idx_active_override_flag được suy ra bằng 1.

num_ref_idx_active_minus1[i] được sử dụng để dẫn xuất biến NumRefIdxActive[i] như được xác định bằng phương trình 145. Giá trị của num_ref_idx_active_minus1[i] phải nằm trong khoảng từ 0 đến 14, bao gồm hai giá trị đầu mút.

Với i bằng 0 hoặc 1, khi lát hiện tại là lát B, num_ref_idx_active_override_flag bằng 1, và num_ref_idx_active_minus1[i] không xuất hiện, num_ref_idx_active_minus1[i] được suy ra bằng 0.

Khi lát hiện tại là lát P, num_ref_idx_active_override_flag bằng 1, và num_ref_idx_active_minus1[0] không xuất hiện, num_ref_idx_active_minus1[0] được suy ra bằng 0.

Biến NumRefIdxActive[i] được dẫn xuất như sau:

```
for(i = 0; i < 2; i++) {
    if(slice_type == B || (slice_type == P && i == 0)) {
        if(num_ref_idx_active_override_flag)
            NumRefIdxActive[i] = num_ref_idx_active_minus1[i] + 1          (145)
        else {
            if(num_ref_entries[i][RplsIdx[i]]                                    >=
num_ref_idx_default_active_minus1[i] + 1)
                NumRefIdxActive[i] = num_ref_idx_default_active_minus1[i] + 1
            else
                NumRefIdxActive[i] = num_ref_entries[i][RplsIdx[i]]
        }
    } else /* slice_type == I || (slice_type == P && i == 1) */
        NumRefIdxActive[i] = 0
}
```

Giá trị của NumRefIdxActive[i] − 1 xác định chỉ số tham chiếu lớn nhất cho danh sách ảnh tham chiếu i có thể được sử dụng để giải mã lát. Khi giá trị của NumRefIdxActive[i] bằng 0, không chỉ số tham chiếu nào cho danh sách ảnh tham chiếu i có thể được sử dụng để giải mã lát.

Khi lát hiện tại là lát P, giá trị của NumRefIdxActive[0] phải lớn hơn 0.

Khi lát hiện tại là lát B, cả NumRefIdxActive[0] và NumRefIdxActive[1] phải lớn hơn 0.

Cú pháp các tham số dự báo có trọng số

pred_weight_table() {	**Mô tả**
luma_log2_weight_denom	ue(v)
if(ChromaArrayType != 0)	
delta_chroma_log2_weight_denom	se(v)
for(i = 0; i < NumRefIdxActive[0]; i++)	
luma_weight_l0_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[0]; i++)	
chroma_weight_l0_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[0]; i++) {	
if(luma_weight_l0_flag[i]) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if(chroma_weight_l0_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
if(slice_type == B) {	
for(i = 0; i < NumRefIdxActive[1]; i++)	
luma_weight_l1_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[1]; i++)	
chroma_weight_l1_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[1]; i++) {	
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_weight_l1_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	

Ngữ nghĩa các tham số dự báo có trọng số

luma_log2_weight_denom là lôgarit cơ số 2 của mẫu số cho tất cả các hệ số trọng số độ sáng. Giá trị của luma_log2_weight_denom phải nằm trong khoảng từ 0 đến 7, bao gồm hai giá trị đầu mút.

delta_chroma_log2_weight_denom là hiệu số của lôgarit cơ số 2 của mẫu số cho tất cả các hệ số trọng số sắc độ. Khi delta_chroma_log2_weight_denom không xuất hiện, được suy ra bằng 0.

Biến ChromaLog2WeightDenom được dẫn xuất bằng luma_log2_weight_denom + delta_chroma_log2_weight_denom và giá trị phải nằm trong khoảng từ 0 đến 7, bao gồm hai giá trị đầu mút.

luma_weight_l0_flag[i] bằng 1 xác định rằng các hệ số trọng số cho thành phần độ sáng dự báo danh sách 0 sử dụng RefPicList[0][i] xuất hiện. luma_weight_l0_flag[i] bằng 0 xác định rằng these các hệ số trọng số không xuất hiện.

chroma_weight_l0_flag[i] bằng 1 xác định rằng các hệ số trọng số cho các giá trị dự báo sắc độ dự báo danh sách 0 sử dụng RefPicList[0][i] xuất hiện. **chroma_weight_l0_flag[i]** bằng 0 xác định rằng các hệ số trọng số này không xuất hiện. Khi chroma_weight_l0_flag[i] không xuất hiện, được suy ra bằng 0.

delta_luma_weight_l0[i] là hiệu số của hệ số trọng số được áp dụng cho giá tị dự báo độ sáng để dự báo danh sách 0 nhờ sử dụng RefPicList[0][i].

Biến LumaWeightL0[i] được dẫn xuất bằng (1 << luma_log2_weight_denom) + delta_luma_weight_l0[i]. Khi luma_weight_l0_flag[i] bằng 1, giá trị của delta_luma_weight_l0[i] phải nằm trong khoảng từ −128 đến 127, bao gồm hai giá trị đầu mút. Khi luma_weight_l0_flag[i] bằng 0, LumaWeightL0[i] được suy ra bằng 2luma_log2_weight_denom.

luma_offset_l0[i] là độ lệch cộng sinh được áp dụng cho giá trị dự báo độ sáng để dự báo danh sách 0 nhờ sử dụng RefPicList[0][i]. Giá trị của luma_offset_l0[i] phải nằm trong khoảng từ −128 đến 127, bao gồm hai giá trị đầu mút. Khi luma_weight_l0_flag[i] bằng 0, luma_offset_l0[i] được suy ra bằng 0.

delta_chroma_weight_l0[i][j] là hiệu số của hệ số trọng số được áp dụng cho các giá trị dự báo sắc độ để dự báo danh sách 0 nhờ sử dụng RefPicList[0][i] với j bằng 0 với Cb và j bằng 1 với Cr.

Biến ChromaWeightL0[i][j] được dẫn xuất bằng (1 << ChromaLog2WeightDenom) + delta_chroma_weight_l0[i][j]. Khi chroma_weight_l0_flag[i] bằng 1, giá trị của delta_chroma_weight_l0[i][j] phải nằm trong khoảng từ −128 đến 127, bao gồm hai giá trị đầu mút. Khi chroma_weight_l0_flag[i] bằng 0, ChromaWeightL0[i][j] được suy ra bằng 2ChromaLog2WeightDenom.

delta_chroma_offset_l0[i][j] là hiệu số của độ lệch cộng sinh được áp dụng cho các giá trị dự báo sắc độ để dự báo danh sách 0 nhờ sử dụng RefPicList[0][i] với j bằng 0 đối với Cb và j bằng 1 đối với Cr.

Biến ChromaOffsetL0[i][j] được dẫn xuất như sau:

ChromaOffsetL0[i][j] = Clip3(−128, 127,
(128 + delta_chroma_offset_l0[i][j] −
((128 * ChromaWeightL0[i][j]) >> ChromaLog2WeightDenom)))

Giá trị của delta_chroma_offset_l0[i][j] phải nằm trong khoảng từ −4 * 128 đến 4 * 127, bao gồm hai giá trị đầu mút. Khi chroma_weight_l0_flag[i] bằng 0, ChromaOffsetL0[i][j] được suy ra bằng 0.

luma_weight_l1_flag[i], chroma_weight_l1_flag[i], delta_luma_weight_l1[i], luma_offset_l1[i], delta_chroma_weight_l1[i][j], và delta_chroma_offset_l1[i][j] có cùng ngữ nghĩa với luma_weight_l0_flag[i], chroma_weight_l0_flag[i], delta_luma_weight_l0[i], luma_offset_l0[i], delta_chroma_weight_l0[i][j] và delta_chroma_offset_l0[i][j], một cách lần lượt, với l0, L0, danh sách 0 và List0 được thay thế bằng l1, L1, list 1 và List1, một cách lần lượt.

Biến sumWeightL0Flags được dẫn xuất bằng tổng của luma_weight_l0_flag[i] + 2 * chroma_weight_l0_flag[i], với i = 0..NumRefIdxActive[0] − 1.

Khi slice_type bằng B, biến sumWeightL1Flags được dẫn xuất bằng tổng của luma_weight_l1_flag[i] + 2 * chroma_weight_l1_flag[i], với i = 0..NumRefIdxActive[1] − 1.

Tương hợp dòng bit yêu cầu rằng, khi slice_type bằng P, sumWeightL0Flags nhỏ hơn hoặc bằng 24 và khi slice_type bằng B, tổng của sumWeightL0Flags và sumWeightL1Flags nhỏ hơn hoặc bằng 24.

Ngữ nghĩa cấu trúc danh sách ảnh tham chiếu

Cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) có thể xuất hiện trong SPS hoặc trong tiêu đề lát. Tùy thuộc liệu cấu trúc cú pháp có được bao gồm trong tiêu đề lát hoặc SPS, phương trình sau áp dụng:

– Nếu xuất hiện trong tiêu đề lát, cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) xác định danh sách ảnh tham chiếu listIdx của ảnh hiện tại (ảnh chứa lát).

– Ngược lại (xuất hiện trong SPS), cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) xác định ứng viên cho danh sách ảnh tham chiếu listIdx, và cụm từ "ảnh hiện tại" trong ngữ nghĩa này được xác định trong phần còn lại của mệnh đề này đề cập đến mỗi ảnh mà 1) có một hoặc nhiều lát chứa ref_pic_list_idx[listIdx] bằng chỉ số vào danh sách của các cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) được bao gồm trong SPS, và 2) trong chuỗi viđeo được tạo mã (Coded Video Sequence, CVS) đề cập đến SPS.

num_ref_entries[listIdx][rplsIdx] xác định số lượng mục trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx). Giá trị của num_ref_entries[listIdx][rplsIdx] phải nằm trong khoảng từ 0 đến MaxDecPicBuffMinus1 + 14, bao gồm hai giá trị đầu mút.

ltrp_in_slice_header_flag[listIdx][rplsIdx] bằng 0 xác định rằng các POC LSB của các mục LTRP trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) xuất hiện trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx). ltrp_in_slice_header_flag[listIdx][rplsIdx] bằng 1 xác định rằng các POC LSB của các mục ảnh tham chiếu dài hạn (Long-Term Reference Picture, LTRP) trong

cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) không xuất hiện trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx).

inter_layer_ref_pic_flag[listIdx][rplsIdx][i] bằng 1 xác định rằng mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) là mục ảnh tham chiếu ngoài lớp (Inter-Layer Reference Picture, ILRP). inter_layer_ref_pic_flag[listIdx][rplsIdx][i] bằng 0 xác định rằng mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) không phải là mục ILRP. Khi không xuất hiện, giá trị của inter_layer_ref_pic_flag[listIdx][rplsIdx][i] được suy ra bằng 0.

st_ref_pic_flag[listIdx][rplsIdx][i] bằng 1 xác định rằng mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) là mục STRP. st_ref_pic_flag[listIdx][rplsIdx][i] bằng 0 xác định rằng mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) là mục LTRP. Khi inter_layer_ref_pic_flag[listIdx][rplsIdx][i] bằng 0 và st_ref_pic_flag[listIdx][rplsIdx][i] không xuất hiện, giá trị của st_ref_pic_flag[listIdx][rplsIdx][i] được suy ra bằng 1.

Biến NumLtrpEntries[listIdx][rplsIdx] được dẫn xuất như sau:

```
for(i = 0, NumLtrpEntries[listIdx][rplsIdx] = 0; i < num_ref_entries[listIdx][rplsIdx]; i++)
    if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i] && !st_ref_pic_flag[listIdx][rplsIdx
][i])
        NumLtrpEntries[listIdx][rplsIdx]++
```

abs_delta_poc_st[listIdx][rplsIdx][i] xác định giá trị của biến AbsDeltaPocSt[listIdx][rplsIdx][i] như sau:

```
if(sps_weighted_pred_flag || sps_weighted_bipred_flag)
    AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i]
else
    AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i] + 1
```

Giá trị của abs_delta_poc_st[listIdx][rplsIdx][i] phải nằm trong khoảng từ 0 đến $2^{15} - 1$, bao gồm hai giá trị đầu mút.

strp_entry_sign_flag[listIdx][rplsIdx][i] bằng 1 xác định rằng mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) có giá trị lớn hơn hoặc bằng 0. strp_entry_sign_flag[listIdx][rplsIdx][i] bằng 0 xác định rằng mục thứ i trong

cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx) có giá trị nhỏ hơn 0. Khi không xuất hiện, giá trị của strp_entry_sign_flag[listIdx][rplsIdx][i] được suy ra bằng 1.

Danh sách DeltaPocValSt[listIdx][rplsIdx] được dẫn xuất như sau:

for(i = 0; i < num_ref_entries[listIdx][rplsIdx]; i++)

if(!inter_layer_ref_pic_flag[listIdx][rplsIdx][i] && st_ref_pic_flag[listIdx][rplsIdx][i])

DeltaPocValSt[listIdx][rplsIdx][i] = (strp_entry_sign_flag[listIdx][rplsIdx][i]) ? AbsDeltaPocSt[listIdx][rplsIdx][i] : 0 − AbsDeltaPocSt[listIdx][rplsIdx][i]

rpls_poc_lsb_lt[listIdx][rplsIdx][i] xác định giá trị của modulo POC MaxPicOrderCntLsb của ảnh được viện dẫn bởi mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx). Chiều dài của phần tử cú pháp rpls_poc_lsb_lt[listIdx][rplsIdx][i] bằng log2_max_pic_order_cnt_lsb_minus4 + 4 bit.

ilrp_idx[listIdx][rplsIdx][i] xác định chỉ số, đến danh sách của các lớp tham chiếu trực tiếp, của ILRP của mục thứ i trong cấu trúc cú pháp ref_pic_list_struct(listIdx, rplsIdx). Giá trị của ilrp_idx[listIdx][rplsIdx][i] phải nằm trong khoảng từ 0 đến NumDirectRefLayers[GeneralLayerIdx[nuh_layer_id]] − 1, bao gồm hai giá trị đầu mút.

Do vậy, các cơ chế khác nhau có thể được sử dụng để có thể điều khiển các chế độ hợp nhất GEO/TPM phụ thuộc liệu WP được áp dụng cho các ảnh tham chiếu trong đó các khối tham chiếu P0 và P1 được lấy từ đó, tức là:

- Di chuyển các tham số WP được liệt kê trong Bảng 14 từ SH sang PH;
- Di chuyển lại các tham số GEO/TPM từ PH sang SH;
- Thay đổi các ngữ nghĩa của MaxNumTriangleMergeCand, tức là, bằng cách gán MaxNumTriangleMergeCand bằng 0 hoặc 1 cho các lát này khi các ảnh tham chiếu có WP có thể được sử dụng (chẳng hạn, trong đó ít nhất một trong các cờ lumaWeightedFlag bằng true).

Đối với chế độ hợp nhất TPM, các khối tham chiếu lấy làm ví dụ P0 và P1 được ký hiệu là 710 và 720 trên Fig. 7, một cách lần lượt. Đối với chế độ hợp

nhất GEO, các khối tham chiếu lấy làm ví dụ P0 và P1 được ký hiệu là 810 và 820 trên Fig.8, một cách lần lượt.

Do vậy, các cơ chế khác nhau có thể được sử dụng để có thể điều khiển các chế độ hợp nhất GEO/TPM phụ thuộc vào việc liệu WP được áp dụng cho các ảnh tham chiếu trong đó các khối tham chiếu P0 và P1 được lấy từ đó, tức là:

- Di chuyển các tham số WP được liệt kê trong Bảng 14 từ SH đến PH;
- Di chuyển lại các tham số GEO/TPM từ PH sang SH;
- Thay đổi các ngữ nghĩa của MaxNumTriangleMergeCand, tức là, bằng cách gán MaxNumTriangleMergeCand bằng 0 hoặc 1 cho các lát này khi các ảnh tham chiếu có WP có thể được sử dụng (chẳng hạn, trong đó ít nhất một trong các cờ lumaWeightedFlag bằng true).

Đối với chế độ hợp nhất TPM, các khối tham chiếu lấy làm ví dụ P0 và P1 được ký hiệu là 710 và 720 trên Fig.7, một cách lần lượt. Đối với chế độ hợp nhất GEO, các khối tham chiếu lấy làm ví dụ P0 và P1 được ký hiệu là 810 và 820 trên Fig.8, một cách lần lượt.

Theo phương án thực hiện, khi các tham số WP và việc kích hoạt các chế độ phi chữ nhật (chẳng hạn, GEO và TPM) được báo hiệu trong tiêu đề ảnh, cú pháp sau có thể được sử dụng, như được thể hiện trên bảng dưới đây:

Bảng Cú pháp tiêu đề ảnh RBSP

picture_header_rbsp() {	**Mô tả**
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
...	
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	

if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
...	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
...	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 && WPDisabled)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rbsp_trailing_bits()	
}	

....

Biến WPDisabled được gán bằng 1 khi tất cả các giá trị của luma_weight_l0_flag[i], chroma_weight_l0_flag[i], luma_weight_l1_flag[j] và chroma_weight_l1_flag[j] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[0]; và giá trị của j=0.. NumRefIdxActive[1]; ngược lại, giá trị của WPDisabled được gán bằng 0.

Khi biến WPDisabled được gán bằng 0, giá trị của pic_max_num_merge_cand_minus_max_num_triangle_cand được gán bằng MaxNumMergeCand.

Trong ví dụ, báo hiệu các tham số WP và việc kích hoạt các chế độ phi chữ nhật (chẳng hạn, GEO và TPM) được thực hiện trong tiêu đề lát. Cú pháp lấy làm ví dụ được nêu trong bảng dưới đây:

slice_header() {	**Mô tả**
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag || NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	

num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag = = 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL || nal_unit_type == IDR_N_LP || nal_unit_type == CRA_NUT || NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type = = I && qtbtt_dual_tree_intra_flag) {	

slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if (slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) || (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
slice_weighted_pred_flag	u(1)
if (slice_weighted_pred_flag)	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1 && WPDisabled) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if (sps_ibc_enabled_flag)	
slice **six_minus_max_num_ibc_merge_cand**	ue(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_qp_offset	se(v)

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

for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

....

Biến WPDisabled được gán bằng 1 khi tất cả các giá trị của luma_weight_l0_flag[i], chroma_weight_l0_flag[i], luma_weight_l1_flag[j] and chroma_weight_l1_flag[j] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[0]; và giá trị của j=0.. NumRefIdxActive[1]; ngược lại, giá trị của WPDisabled được gán bằng 0.

Khi biến WPDisabled được gán bằng 0, giá trị của max_num_merge_cand_minus_max_num_triangle_cand được gán bằng MaxNumMergeCand

Theo phương án thực hiện được bộc lộ trên đây, các tham số dự báo có trọng số có thể được báo hiệu trong tiêu đề ảnh hoặc trong tiêu đề lát.

Trong ví dụ, việc xác định liệu TPM hoặc GEO được kích hoạt được thực hiện khi xem xét danh sách ảnh tham chiếu mà khối có thể sử dụng cho dự báo có trọng số phi chữ nhật. Khi danh sách hợp nhất cho khối chứa các phần tử từ chỉ một danh sách ảnh tham chiếu k, giá trị của biến WPDisabled [k] xác định liệu chế độ hợp nhất này có được kích hoạt hay không.

Trong ví dụ, danh sách hợp nhất cho chế độ dự báo ngoài phi chữ nhật được tạo theo cách sao cho nó chứa chỉ các phần tử mà dự báo có trọng số không được kích hoạt.

Phần mô tả minh họa ví dụ này:

Đầu vào cho quá trình này là:

– vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của CB độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

– biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

– biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng.

Các đầu ra của quá trình này như sau, với X bằng 0 hoặc 1:

– các cờ khả dụng $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ và $availableFlagB_2$ của các CU lân cận,

– các chỉ số tham chiếu $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ và $refIdxLXB_2$ của các CU lân cận,

– các cờ sử dụng danh sách dự báo $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ và $predFlagLXB_2$ của các CU lân cận,

– các vectơ chuyển động có độ chính xác mẫu phân số 1/16 $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ và $mvLXB_2$ của các CU lân cận,

– các chỉ số lọc nội suy nửa mẫu $hpelIfIdxA_0$, $hpelIfIdxA_1$, $hpelIfIdxB_0$, $hpelIfIdxB_1$, và $hpelIfIdxB_2$,

– các chỉ số trọng số dự báo kép $bcwIdxA_0$, $bcwIdxA_1$, $bcwIdxB_0$, $bcwIdxB_1$, và $bcwIdxB_2$.

Để dẫn xuất $availableFlagB_1$, $refIdxLXB_1$, $predFlagLXB_1$, $mvLXB_1$, $hpelIfIdxB_1$ và $bcwIdxB_1$ phương trình sau áp dụng:

– Vị trí độ sáng ($xNbB_1$, $yNbB_1$) bên trong khối mã độ sáng lân cận được gán bằng (xCb + cbWidth − 1, yCb − 1).

– Quá trình dẫn xuất đối với tính khả dụng khối lân cận như được nêu trong mệnh đề 6.4.4 được gọi với vị trí độ sáng hiện tại (xCurr, yCurr) được gán bằng (xCb, yCb), vị trí độ sáng lân cận ($xNbB_1$, $yNbB_1$), checkPredModeY được gán bằng TRUE, và cIdx được gán bằng 0 làm đầu vào, và đầu ra được gán bằng cờ tính khả dụng của khối $availableB_1$.

– Các biến $availableFlagB_1$, $refIdxLXB_1$, $predFlagLXB_1$, $mvLXB_1$, $hpelIfIdxB_1$ và $bcwIdxB_1$ được dẫn xuất như saus:

– Nếu $availableB_1$ bằng FALSE, $availableFlagB_1$ được gán bằng 0, cả hai thành phần của $mvLXB_1$ được gán bằng 0, $refIdxLXB_1$ được gán bằng −1 và $predFlagLXB_1$ được gán bằng 0, với X bằng 0 hoặc 1, $hpelIfIdxB_1$ được gán bằng 0, và $bcwIdxB_1$ được gán bằng 0.

– Ngược lại, $availableFlagB_1$ được gán bằng 1 và thực hiện các phép gán sau:

$mvLXB_1 = MvLX[xNbB_1][yNbB_1]$ (501)

$refIdxLXB_1 = RefIdxLX[xNbB_1][yNbB_1]$ (502)

$predFlagLXB_1 = PredFlagLX[xNbB_1][yNbB_1]$

(503)

$hpelIfIdxB_1 = HpelIfIdx[xNbB_1][yNbB_1]$ (504)

$bcwIdxB_1 = BcwIdx[xNbB_1][yNbB_1]$ (505)

Để dẫn xuất $availableFlagA_1$, $refIdxLXA_1$, $predFlagLXA_1$, $mvLXA_1$, $hpelIfIdxA_1$ và $bcwIdxA_1$ phương trình sau áp dụng:

– Vị trí độ sáng ($xNbA_1$, $yNbA_1$) bên trong CB độ sáng lân cận được gán bằng (xCb − 1, yCb + cbHeight − 1).

– Quá trình dẫn xuất đối với tính khả dụng khối lân cận như được nêu trong mệnh đề 6.4.4 được gọi với vị trí độ sáng hiện tại (xCurr, yCurr) được gán bằng (xCb, yCb). vị trí độ sáng lân cận ($xNbA_1$, $yNbA_1$), checkPredModeY được gán bằng TRUE, và cIdx được gán bằng 0 làm các đầu vào, và đầu ra được gán bằng cờ tính khả dụng của khối $availableA_1$.

– Các biến $availableFlagA_1$, $refIdxLXA_1$, $predFlagLXA_1$, $mvLXA_1$, $hpelIfIdxA_1$ và $bcwIdxA_1$ được dẫn xuất như sau:

– Nếu một hoặc nhiều điều kiện sau là đúng, $availableFlagA_1$ được gán bằng 0, cả hai thành phần của $mvLXA_1$ được gán bằng 0, $refIdxLXA_1$ được gán bằng −1 và $predFlagLXA_1$ được gán bằng 0, với X bằng 0 hoặc 1, $hpelIfIdxA_1$ được gán bằng 0, và $bcwIdxA_1$ được gán bằng 0:

– $availableA_1$ bằng FALSE.

– $availableB_1$ bằng TRUE và các vị trí độ sáng ($xNbA_1$, $yNbA_1$) và ($xNbB_1$, $yNbB_1$) có các MV giống nhau và các chỉ số tham chiếu giống nhau.

– $WPDisabledX[RefIdxLX[xNbA_1][yNbA_1]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– $WPDisabledX[RefIdxLX[xNbB_1][yNbB_1]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– Ngược lại, $availableFlagA_1$ được gán bằng 1 và thực hiện các phép gán sau:

$mvLXA_1 = MvLX[xNbA_1][yNbA_1]$ (506)

$refIdxLXA_1 = RefIdxLX[xNbA_1][yNbA_1]$ (507)

$predFlagLXA_1 = PredFlagLX[xNbA_1][yNbA_1]$ (508)

$hpelIfIdxA_1 = HpelIfIdx[xNbA_1][yNbA_1]$ (509)

$bcwIdxA_1 = BcwIdx[xNbA_1][yNbA_1]$ (510)

Để dẫn xuất $availableFlagB_0$, $refIdxLXB_0$, $predFlagLXB_0$, $mvLXB_0$, $hpelIfIdxB_0$ và $bcwIdxB_0$ phương trình sau áp dụng:

– Vị trí độ sáng ($xNbB_0$, $yNbB_0$) bên trong CB độ sáng lân cận được gán bằng (xCb + cbWidth, yCb − 1).

– Quá trình dẫn xuất đối với tính khả dụng khối lân cận như được nêu trong mệnh đề 6.4.4 được gọi với vị trí độ sáng hiện tại (xCurr, yCurr) được gán bằng (xCb, yCb), vị trí độ sáng lân cận ($xNbB_0$, $yNbB_0$), checkPredModeY được gán bằng TRUE, và cIdx được gán bằng 0 làm các đầu vào, và đầu ra được gán bằng cờ tính khả dụng của khối $availableB_0$.

– Biến $availableFlagB_0$, $refIdxLXB_0$, $predFlagLXB_0$, $mvLXB_0$, $hpelIfIdxB_0$ và $bcwIdxB_0$ được dẫn xuất như saus:

– Nếu một hoặc nhiều điều kiện sau là đúng, $availableFlagB_0$ được gán bằng 0, cả hai thành phần của $mvLXB_0$ được gán bằng 0, $refIdxLXB_0$ được gán bằng −1 và $predFlagLXB_0$ được gán bằng 0, với X bằng 0 hoặc 1, $hpelIfIdxB_0$ được gán bằng 0, và $bcwIdxB_0$ được gán bằng 0:

– $availableB_0$ bằng FALSE.

– $availableB_1$ bằng TRUE và các vị trí độ sáng ($xNbB_1$, $yNbB_1$) và ($xNbB_0$, $yNbB_0$) có các MV giống nhau và các chỉ số tham chiếu giống nhau.

– $WPDisabledX[RefIdxLX[xNbB_0][yNbB_0]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– $WPDisabledX[RefIdxLX[xNbB_1][yNbB_1]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– Ngược lại, $availableFlagB_0$ được gán bằng 1 và thực hiện các phép gán sau:

$mvLXB_0 = MvLX[xNbB_0][yNbB_0]$ (511)

$refIdxLXB_0 = RefIdxLX[xNbB_0][yNbB_0]$ (512)

$predFlagLXB_0 = PredFlagLX[xNbB_0][yNbB_0]$ (513)

$hpelIfIdxB_0 = HpelIfIdx[xNbB_0][yNbB_0]$ (514)

$bcwIdxB_0 = BcwIdx[xNbB_0][yNbB_0]$ (515)

Để dẫn xuất $availableFlagA_0$, $refIdxLXA_0$, $predFlagLXA_0$, $mvLXA_0$, $hpelIfIdxA_0$ và $bcwIdxA_0$ phương trình sau áp dụng:

– Vị trí độ sáng ($xNbA_0$, $yNbA_0$) bên trong CB độ sáng lân cận được gán bằng (xCb − 1, yCb + cbWidth).

– Quá trình dẫn xuất đối với tính khả dụng khối lân cận như được nêu trong mệnh đề 6.4.4 được gọi với vị trí độ sáng hiện tại (xCurr, yCurr) được gán bằng (xCb, yCb). vị trí độ sáng lân cận ($xNbA_0$, $yNbA_0$), checkPredModeY được gán bằng TRUE, và cIdx được gán bằng 0 làm các đầu vào, và đầu ra được gán bằng cờ tính khả dụng của khối $availableA_0$.

– Các biến $availableFlagA_0$, $refIdxLXA_0$, $predFlagLXA_0$, $mvLXA_0$, $hpelIfIdxA_0$ và $bcwIdxA_0$ được dẫn xuất như sau:

– Nếu một hoặc nhiều điều kiện sau là đúng, $availableFlagA_0$ được gán bằng 0, cả hai thành phần của $mvLXA_0$ được gán bằng 0, $refIdxLXA_0$ được gán bằng −1 và $predFlagLXA_0$ được gán bằng 0, với X bằng 0 hoặc 1, $hpelIfIdxA_0$ được gán bằng 0, và $bcwIdxA_0$ được gán bằng 0:

– $availableA_0$ bằng FALSE.

– $availableA_1$ bằng TRUE và các vị trí độ sáng ($xNbA_1$, $yNbA_1$) và ($xNbA_0$, $yNbA_0$) có các MV giống nhau và các chỉ số tham chiếu giống nhau.

– $WPDisabledX[RefIdxLX[xNbA_0][yNbA_0]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– $WPDisabledX[RefIdxLX[xNbA_1][yNbA_1]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– Ngược lại, $availableFlagA_0$ được gán bằng 1 và thực hiện các phép gán sau:

$mvLXA_0 = MvLX[xNbA_0][yNbA_0]$ (516)

$refIdxLXA_0 = RefIdxLX[xNbA_0][yNbA_0]$ (517)

$predFlagLXA_0 = PredFlagLX[xNbA_0][yNbA_0]$ (518)

$hpelIfIdxA_0 = HpelIfIdx[xNbA_0][yNbA_0]$ (519)

$bcwIdxA_0 = BcwIdx[xNbA_0][yNbA_0]$ (520)

Để dẫn xuất $availableFlagB_2$, $refIdxLXB_2$, $predFlagLXB_2$, $mvLXB_2$, $hpelIfIdxB_2$ và $bcwIdxB_2$ phương trình sau áp dụng:

– Vị trí độ sáng ($xNbB_2$, $yNbB_2$) bên trong CB độ sáng lân cận được gán bằng (xCb − 1, yCb − 1).

– Quá trình dẫn xuất đối với tính khả dụng khối lân cận như được nêu trong mệnh đề 6.4.4 được gọi với vị trí độ sáng hiện tại (xCurr, yCurr) được gán bằng (xCb, yCb), vị trí độ sáng lân cận ($xNbB_2$, $yNbB_2$), checkPredModeY được gán bằng TRUE, và cIdx được gán bằng 0 làm các đầu vào, và đầu ra được gán bằng cờ tính khả dụng của khối $availableB_2$.

– Các biến $availableFlagB_2$, $refIdxLXB_2$, $predFlagLXB_2$, $mvLXB_2$, $hpelIfIdxB_2$ và $bcwIdxB_2$ được dẫn xuất như saus:

– Nếu một hoặc nhiều điều kiện sau là đúng, $availableFlagB_2$ được gán bằng 0, cả hai thành phần của $mvLXB_2$ được gán bằng 0, $refIdxLXB_2$ được gán bằng −1 và $predFlagLXB_2$ được gán bằng 0, với X bằng 0 hoặc 1, $hpelIfIdxB_2$ được gán bằng 0, và $bcwIdxB_2$ được gán bằng 0:

– $availableB_2$ bằng FALSE.

– $availableA_1$ bằng TRUE và các vị trí độ sáng ($xNbA_1$, $yNbA_1$) và ($xNbB_2$, $yNbB_2$) có các MV giống nhau và các chỉ số tham chiếu giống nhau.

– $availableB_1$ bằng TRUE và các vị trí độ sáng ($xNbB_1$, $yNbB_1$) và ($xNbB_2$, $yNbB_2$) có các MV giống nhau và các chỉ số tham chiếu giống nhau.

– $availableFlagA_0 + availableFlagA_1 + availableFlagB_0 + availableFlagB_1$ bằng 4.

– $WPDisabledX[RefIdxLX[xNbB_1][yNbB_1]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– $WPDisabledX[RefIdxLX[xNbB_2][yNbB_2]]$ được gán bằng 0 và chế độ hợp nhất là phi chữ nhật (chẳng hạn, cờ tam giác được gán bằng 1 cho khối ở vị trí độ sáng hiện tại (xCurr, yCurr))

– Ngược lại, $availableFlagB_2$ được gán bằng 1 và thực hiện các phép gán sau:

$mvLXB_2 = MvLX[xNbB_2][yNbB_2]$ (521)

$refIdxLXB_2 = RefIdxLX[xNbB_2][yNbB_2]$ (522)

$predFlagLXB_2 = PredFlagLX[xNbB_2][yNbB_2]$ (523)

$hpelIfIdxB_2 = HpelIfIdx[xNbB_2][yNbB_2]$ (524)

$bcwIdxB_2 = BcwIdx[xNbB_2][yNbB_2]$ (525)

Trong các ví dụ được bộc lộ trên đây, định nghĩa biến sau được sử dụng:

Biến WPDisabled0[i] được gán bằng 1 khi tất cả các giá trị của luma_weight_l0_flag[i] and chroma_weight_l0_flag[i] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[0] . Ngược lại, giá trị của WPDisabled0[i] được gán bằng 0.

Biến WPDisabled1[i] được gán bằng 1 khi tất cả các giá trị của luma_weight_l1_flag[i] and chroma_weight_l1_flag[i] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[1] . Ngược lại, giá trị của WPDisabled1 [1] được gán bằng 0.

Trong ví dụ khác, biến SliceMaxNumTriangleMergeCand được định nghĩa ở tiêu đề lát theo một trong các phương trình sau:

- SliceMaxNumTriangleMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 0 : MaxNumTriangleMergeCand;
- SliceMaxNumTriangleMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 1 : MaxNumTriangleMergeCand;
- SliceMaxNumTriangleMergeCand = slice_weighted_pred_flag ? 0 : MaxNumTriangleMergeCand;

Hoặc

- SliceMaxNumTriangleMergeCand = slice_weighted_pred_flag ? 1 : MaxNumTriangleMergeCand

Giá trị của SliceMaxNumTriangleMergeCand được sử dụng tiếp khi phân tách thông tin hợp nhất ở mức khối. Cú pháp lấy làm ví dụ được nêu trong bảng dưới đây:

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Mô tả
if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) || (sps_triangle_enabled_flag && SliceMaxNumTriangleMergeCand > 1 && slice_type == B)))	
regular_merge_flag[x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1){	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
} else {	
if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && SliceMaxNumTriangleMergeCand > 1 && weightedPredFlag == 0 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && SliceMaxNumTriangleMergeCand > 1) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
merge_triangle_idx1[x0][y0]	ae(v)
}	
}	
}	
}	
}	

Đối với các trường hợp của chế độ dự báo ngoài phi chữ nhật là chế độ GEO, các ví dụ sau còn được mô tả.

Các cơ chế khác nhau có thể được sử dụng để có thể điều khiển các chế độ hợp nhất GEO/TPM, tùy thuộc liệu WP được áp dụng cho các ảnh tham chiếu trong đó các khối tham chiếu P0 và P1 được lấy từ đó, tức là:

- Di chuyển các tham số WP được liệt kê trong Bảng 14 từ SH đến PH;
- Di chuyển lại các tham số GEO từ PH về SH;
- Thay đổi các ngữ nghĩa của MaxNumGeoMergeCand, chẳng hạn, bằng cách gán MaxNumGeoMergeCand bằng 0 hoặc 1 cho các lát này khi các ảnh tham chiếu có WP có thể được sử dụng (chẳng hạn, trong đó ít nhất một trong các cờ lumaWeightedFlag bằng true).

Đối với chế độ hợp nhất GEO, các khối tham chiếu lấy làm ví dụ P0 và P1 được ký hiệu là 810 và 820 trên Fig.8, một cách lần lượt.

Trong ví dụ, khi các tham số WP và việc kích hoạt các chế độ phi chữ nhật (chẳng hạn, GEO và TPM) được báo hiệu trong tiêu đề ảnh, cú pháp sau có thể được sử dụng, như được thể hiện trên bảng dưới đây:

Bảng – Cú pháp RBSP tiêu đề ảnh

picture_header_rbsp() {	**Mô tả**
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
…	
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	

ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
...	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
...	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 && WPDisabled)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rbsp_trailing_bits()	
}	

....

Biến WPDisabled được gán bằng 1 khi tất cả các giá trị của luma_weight_l0_flag[i], chroma_weight_l0_flag[i], luma_weight_l1_flag[j] và chroma_weight_l1_flag[j] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[0]; và giá trị của j=0.. NumRefIdxActive[1]; ngược lại, giá trị của WPDisabled được gán bằng 0.

Khi biến WPDisabled được gán bằng 0, giá trị của pic_max_num_merge_cand_minus_max_num_geo_cand được gán bằng MaxNumMergeCand.

Trong ví dụ khác, pic_max_num_merge_cand_minus_max_num_geo_cand được gán bằng MaxNumMergeCand – 1.

Trong ví dụ, báo hiệu các tham số WP và việc kích hoạt các chế độ phi chữ nhật (chẳng hạn, GEO và TPM) được thực hiện trong tiêu đề lát. Cú pháp lấy làm ví dụ được nêu trong bảng dưới đây:

slice_header() {	**Mô tả**
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag || NumTilesInPic > 1)	
slice_address	u(v)

if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag &&((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(pic_rpl_present_flag || ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(pic_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) || (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if (slice_weighted_pred_flag)	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	**ue(v)**
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	

five_minus_max_num_subblock_merge_cand	**ue(v)**
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	**u(1)**
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	**u(1)**
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1 && WPDisabled) {	
max_num_merge_cand_minus_max_num_geo_cand	**ue(v)**
}	
...	
}	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !pic_sao_enabled_present_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !pic_deblocking_filter_override_present_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(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	

if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Biến WPDisabled được gán bằng 1 khi tất cả các giá trị của luma_weight_l0_flag[i], chroma_weight_l0_flag[i], luma_weight_l1_flag[j] và chroma_weight_l1_flag[j] được gán bằng 0, giá trị của i =0 .. NumRefIdxActive[0]; và giá trị của j=0.. NumRefIdxActive[1]; ngược lại, giá trị của WPDisabled được gán bằng 0.

Khi biến WPDisabled được gán bằng 0, giá trị của max_num_merge_cand_minus_max_num_geo_cand được gán bằng MaxNumMergeCand.

Theo phương án thực hiện khác, khi biến WPDisabled được gán bằng 0, giá trị của max_num_merge_cand_minus_max_num_geo_cand được gán bằng MaxNumMergeCand – 1.

Trong các ví dụ nêu trên, các tham số dự báo có trọng số có thể được báo hiệu trong tiêu đề ảnh hoặc trong tiêu đề lát.

Theo các phương án thực hiện khác, biến SliceMaxNumGeoMergeCand được định nghĩa ở tiêu đề lát theo một trong các phương trình sau:

- SliceMaxNumGeoMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 0 : MaxNumGeoMergeCand ;
- SliceMaxNumGeoMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 1 : MaxNumGeoMergeCand ;
- SliceMaxNumGeoMergeCand = slice_weighted_pred_flag ? 0 : MaxNumGeoMergeCand ;

Hoặc SliceMaxNumGeoMergeCand = slice_weighted_pred_flag ? 1 : MaxNumGeoMergeCand

Các phương án thực hiện khác nhau sử dụng các trường hợp khác nhau được liệt kê trên đây.

Giá trị của biến SliceMaxNumGeoMergeCand còn được sử dụng khi phân tách thông tin hợp nhất ở mức khối. Cú pháp lấy làm ví dụ được nêu trong bảng dưới đây:

7.3.9.7 Cú pháp dữ liệu hợp nhất

merge_data(x0, y0, cbWidth, cbHeight, chType) {	**Mô tả**
if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) || (sps_geo_enabled_flag && SliceMaxNumGeoMergeCand > 1 && cbWidth>=8 && cbHeight >=8 && slice_type == B))	
regular_merge_flag[x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && sps_geo_enabled_flag && SliceMaxNumGeoMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < 128 && cbHeight < 128)	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && SliceMaxNumGeoMergeCand > 1) {	
merge_geo_partition_idx[x0][y0]	ae(v)
merge_geo_idx0[x0][y0]	ae(v)
if(SliceMaxNumGeoMergeCand > 2)	
merge_geo_idx1[x0][y0]	ae(v)
}	
}	
}	
}	

}	

Ngữ nghĩa tiêu đề ảnh liên quan như sau:

pic_max_num_merge_cand_minus_max_num_geo_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học được hỗ trợ trong các lát được liên kết với tiêu đề ảnh được trừ đi khỏi MaxNumMergeCand.

Khi pic_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện, và sps_geo_enabled_flag bằng 1 and MaxNumMergeCand lớn hơn hoặc bằng 2, pic_max_num_merge_cand_minus_max_num_geo_cand được suy ra bằng pps_max_num_merge_cand_minus_max_num_geo_cand_plus1 − 1.

Số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học, MaxNumGeoMergeCand được dẫn xuất như sau:

MaxNumGeoMergeCand = MaxNumMergeCand − pic_max_num_merge_cand_minus_max_num_geo_cand

Khi pic_max_num_merge_cand_minus_max_num_geo_cand xuất hiện, giá trị của MaxNumGeoMergeCand phải nằm trong khoảng từ 2 đến MaxNumMergeCand, bao gồm hai giá trị đầu mút.

Khi pic_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện, và (sps_geo_enabled_flag bằng 0 hoặc MaxNumMergeCand nhỏ hơn 2), MaxNumGeoMergeCand được gán bằng 0.

Khi MaxNumGeoMergeCand bằng 0, chế độ hợp nhất hình học không được phép cho các lát được liên kết với PH.

Trong các ví dụ sau, vài khía cạnh liên quan báo hiệu được xem xét. Tức là, các khía cạnhnày như sau:

- các phần tử cú pháp liên quan đến số lượng ứng viên cho chế độ hợp nhất () được báo hiệu trong SPS, làm các cách thực hiện cụ thể có thể dẫn xuất số lượng ứng viên chế độ hợp nhất phi chữ nhật (MaxNumGeoMergeCand) ở mức SPS;
- PH có thể được báo hiệu trong SH, khi ảnh bao gồm chỉ một lát;
- Định nghĩa cơ chế ghi đè tham số PH/SH với điều kiện như sau:

Các cờ PPS xác định liệu phần tử cú pháp của công cụ mã liên quan có xuất hiện trong PH hoặc SH (nhưng không phải cả hai).

Cụ thể là, bảng danh sách ảnh tham chiếu và dự báo có trọng số có thể sử dụng cơ chế này

- Trong bảng có trọng số dự báo, loại dữ liệu thứ năm có thể được báo hiệu trong PH hoặc SH (giống như ALF, tách khối, RPL, và SAO);
- khi dự báo có trọng số có được kích hoạt cho ảnh, tất cả các lát ảnh phải có cùng các danh sách ảnh tham chiếu;
- các phần tử cú pháp trong và ngoài được báo hiệu có điều kiện nếu chỉ các loại lát cụ thể được sử dụng trong ảnh được liên kết với PH.

Cụ thể là, hai cờ, pic_inter_slice_present_flag và pic_intra_slice_present_flag được đưa vào.

Trong ví dụ, các phần tử cú pháp liên quan đến số lượng ứng viên cho chế độ hợp nhất () được báo hiệu trong SPS, khiến các cách thực hiện cụ thể có thể dẫn xuất số lượng ứng viên chế độ hợp nhất phi chữ nhật (MaxNumGeoMergeCand) ở mức SPS. Khía cạnh này có thể được thực hiện bằng quá trình mã hóa hoặc giải mã dựa trên cú pháp sau.

7.3.2.3 Cú pháp RBSP SPS

seq_parameter_set_rbsp() {	**Mô tả**
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
sps_seq_parameter_set_id	u(4)
...	
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
sps_affine_type_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
if(chroma_format_idc == 3) {	
sps_palette_enabled_flag	u(1)
sps_act_enabled_flag	u(1)
}	
sps_bcw_enabled_flag	u(1)

sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_geo_enabled_flag	u(1)
sps_six_minus_max_num_merge_cand_plus1	ue(v)
if (sps_geo_enabled_flag)	
sps_max_num_merge_cand_minus_max_num_geo_cand_plus1	ue(v)
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
...	

Cú pháp nêu trên có các ngữ nghĩa sau.

sps_six_minus_max_num_merge_cand_plus1 bằng 0 xác định rằng pic_six_minus_max_num_merge_cand xuất hiện trong các PH đề cập đến PPS. sps_six_minus_max_num_merge_cand_plus1 lớn hơn 0 xác định rằng pic_six_minus_max_num_merge_cand không xuất hiện trong các PH viện dẫn đến PPS. Giá trị của sps_six_minus_max_num_merge_cand_plus1 phải nằm trong khoảng từ 0 đến 6, bao gồm hai giá trị đầu mút.

sps_max_num_merge_cand_minus_max_num_geo_cand_plus1 bằng 0 xác định rằng pic_max_num_merge_cand_minus_max_num_geo_cand xuất hiện trong các PH của các lát viện dẫn đến PPS. sps_max_num_merge_cand_minus_max_num_geo_cand_plus1 lớn hơn 0 xác định rằng pic_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện trong các PH viện dẫn đến PPS. Giá trị của sps_max_num_merge_cand_minus_max_num_geo_cand_plus1 phải nằm trong khoảng từ 0 đến MaxNumMergeCand − 1.

Ngữ nghĩa của các phần tử tương ứng của PH như sau:

pic_six_minus_max_num_merge_cand xác định số lượng lớn nhất của các ứng viên MVP hợp nhất được hỗ trợ trong các lát được liên kết với PH được trừ đi khỏi 6. Số lượng lớn nhất của các ứng viên MVP hợp nhất, MaxNumMergeCand được dẫn xuất như sau:

MaxNumMergeCand = 6 – pic_six_minus_max_num_merge_cand

Giá trị của MaxNumMergeCand phải nằm trong khoảng từ 1 đến 6, bao gồm hai giá trị đầu mút. Khi không xuất hiện, giá trị của

pic_six_minus_max_num_merge_cand được suy ra bằng sps_six_minus_max_num_merge_cand_plus1 − 1.

pic_max_num_merge_cand_minus_max_num_geo_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học được hỗ trợ trong các lát được liên kết với tiêu đề ảnh được trừ đi khỏi MaxNumMergeCand.

Khi sps_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện, và sps_geo_enabled_flag bằng 1 and MaxNumMergeCand lớn hơn hoặc bằng 2, pic_max_num_merge_cand_minus_max_num_geo_cand được suy ra bằng sps_max_num_merge_cand_minus_max_num_geo_cand_plus1 − 1.

Số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học, MaxNumGeoMergeCand được dẫn xuất như sau:

MaxNumGeoMergeCand = MaxNumMergeCand − pic_max_num_merge_cand_minus_max_num_geo_cand

Khi pic_max_num_merge_cand_minus_max_num_geo_cand xuất hiện, giá trị của MaxNumGeoMergeCand phải nằm trong khoảng từ 2 đếnMaxNumMergeCand, bao gồm hai giá trị đầu mút.

Khi pic_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện, và (sps_geo_enabled_flag bằng 0 hoặc MaxNumMergeCand nhỏ hơn 2), MaxNumGeoMergeCand được gán bằng 0.

Khi MaxNumGeoMergeCand bằng 0, chế độ hợp nhất hình học không được phép cho các lát được liên kết với PH.

Theo cách khác, max_num_merge_cand_minus_max_num_geo_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học được hỗ trợ trong SPS được trừ đi khỏi MaxNumMergeCand.

Khi sps_geo_enabled_flag bằng 1 và MaxNumMergeCand lớn hơn hoặc bằng 3, số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học, MaxNumGeoMergeCand được dẫn xuất như sau:

MaxNumGeoMergeCand = MaxNumMergeCand − max_num_merge_cand_minus_max_num_geo_cand

Nếu giá trị của sps_geo_enabled_flag bằng 1, giá trị của MaxNumGeoMergeCand phải nằm trong khoảng từ 2 đến MaxNumMergeCand, bao gồm hai giá trị đầu mút.

Ngược lại khi sps_geo_enabled_flag bằng 1 và MaxNumMergeCand bằng 2, MaxNumGeoMergeCand được gán bằng 2.

Ngược lại, MaxNumGeoMergeCand được gán bằng 0.

Cú pháp và ngữ nghĩa tùy chọn cho ví dụ này như sau:

...	
sps_geo_enabled_flag	u(1)
sps_six_minus_max_num_merge_cand	ue(v)
if (sps_geo_enabled_flag)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

sps_six_minus_max_num_merge_cand xác định số lượng lớn nhất của các ứng viên MVP hợp nhất được hỗ trợ trong các lát được liên kết với the PH được trừ đi khỏi 6. Số lượng lớn nhất của các ứng viên MVP hợp nhất, MaxNumMergeCand được dẫn xuất như sau:

MaxNumMergeCand = 6 – sps_six_minus_max_num_merge_cand

Giá trị của MaxNumMergeCand phải nằm trong khoảng từ 1 đến 6, bao gồm hai giá trị đầu mút.

sps_max_num_merge_cand_minus_max_num_geo_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học được hỗ trợ trong các lát được liên kết với tiêu đề ảnh được trừ đi khỏi MaxNumMergeCand.

Số lượng lớn nhất của các ứng viên chế độ hợp nhất hình học, MaxNumGeoMergeCand được dẫn xuất như sau:

MaxNumGeoMergeCand = MaxNumMergeCand – sps_max_num_merge_cand_minus_max_num_geo_cand

Khi sps_max_num_merge_cand_minus_max_num_geo_cand xuất hiện, giá trị của MaxNumGeoMergeCand phải nằm trong khoảng từ 2 đến MaxNumMergeCand, bao gồm hai giá trị đầu mút.

Khi sps_max_num_merge_cand_minus_max_num_geo_cand không xuất hiện, và (sps_geo_enabled_flag bằng 0 hoặc MaxNumMergeCand nhỏ hơn 2), MaxNumGeoMergeCand được gán bằng 0.

Khi MaxNumGeoMergeCand bằng 0, chế độ hợp nhất hình học không được phép.

Đối với các ví dụ nêu trên và đối với các định nghĩa cú pháp tùy chọn, phép kiểm tra được thực hiện trên liệu dự báo có trọng số được kích hoạt. Phép kiểm tra này ảnh hưởng dẫn xuất biến MaxNumGeoMergeCand, và giá trị của MaxNumGeoMergeCand được gán bằng 0 ở một trong các trường hợp sau:

- khi với giá trị của i = 0 .. NumRefIdxActive[0] và giá trị của j = 0 .. NumRefIdxActive[1] tất cả các giá trị của luma_weight_l0_flag[i], chroma_weight_l0_flag[i], luma_weight_l1_flag[j] và chroma_weight_l1_flag[j] được gán bằng 0 hoặc không xuất hiện;
- khi cờ trong SPS hoặc PPS chỉ báo sự xuất hiện của dự báo có trọng số hai hướng (pps_weighted_bipred_flag);
- khi sự có mặt của dự báo có trọng số hai hướng được chỉ báo trong tiêu đề ảnh (PH) hoặc tiêu đề lát (SH).

Cờ mức SPS chỉ báo sự có mặt của các tham số dự báo có trọng số có thể được báo hiệu như sau:

sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_wp_enabled_flag	
if (!sps_wp_enabled_flag)	
sps_geo_enabled_flag	u(1)
sps_six_minus_max_num_merge_cand_plus1	ue(v)
if (sps_geo_enabled_flag)	
sps_max_num_merge_cand_minus_max_num_geo_cand_plus1	ue(v)
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
...	

Phần tử cú pháp "sps_wp_enabled_flag" xác định liệu dự báo có trọng số có thể được kích hoạt ở mức thấp hơn (PPS, PH hoặc SH). Cách thực hiện lấy làm ví dụ được nêu dưới đây:

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_cbcr_qp_offset_present_flag)	
joint_cbcr_qp_offset_list[i]	se(v)

}	
}	
if (sps_wp_enabled_flag) {	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
}	

Trong bảng trên, pps_weighted_pred_flag và pps_weighted_bipred_flag là các cờ trong dòng bit chỉ báo liệu dự báo có trọng số có được kích hoạt cho các khối được dự báo đơn và dự báo kép.

Trong ví dụ, trong đó các cờ dự báo có trọng số được xác định trong tiêu đề ảnh, chẳng hạn, dưới dạng pic_weighted_pred_flag và pic_weighted_bipred_flag, lệ thuộc sau vào sps_wp_enabled_flag có thể được xác định trong cú pháp dòng bit:

...
if (sps_wp_enabled_flag) {
pic_weighted_pred_flag
pic_weighted_bipred_flag
}
...

Trong ví dụ, trong đó các cờ dự báo có trọng số được xác định trong tiêu đề lát, chẳng hạn, dưới dạng weighted_pred_flag và weighted_bipred_flag, lệ thuộc sau vào sps_wp_enabled_flag có thể được xác định trong cú pháp dòng bit:

...
if (sps_wp_enabled_flag) {
weighted_pred_flag
weighted_bipred_flag
}
...

Trong ví dụ, các danh sách ảnh tham chiếu có thể được chỉ báo trong PPS hoặc trong PH hoặc SH (nhưng không phải cả hai). Trong một số ví dụ, báo hiệu của danh sách ảnh tham chiếu lệ thuộc vào các phần tử cú pháp chỉ báo sự có mặt của dự báo có trọng số (chẳng hạn, pps_weighted_pred_flag và pps_weighted_bipred_flag). Do vậy, việc lệ thuộc vào liệu danh sách ảnh tham chiếu được chỉ báo trong PPS, PH hoặc SH, các tham số dự báo có trọng số được báo hiệu trước danh sách ảnh tham chiếu một cách tương ứng trong PPS, PH hoặc SH.

Cú pháp sau có thể được xác định cho phương án thực hiện:

Cú pháp tập hợp tham số ảnh

pic_parameter_set_rbsp() {	**Mô tả**
…	
rpl_present_in_ph_flag	u(1)
sao_present_in_ph_flag	u(1)
alf_present_in_ph_flag	u(1)
…	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
if(pps_weighted_pred_flag || pps_weighted_bipred_flag || rpl_present_in_ph_flag)	
weighted_pred_table_present_in_ph_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_present_in_ph_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
…	
}	

…

rpl_present_in_ph_flag bằng 1 xác định báo hiệu danh sách ảnh tham chiếu không xuất hiện trong các tiêu đề lát viện dẫn đến PPS mà có thể xuất hiện trong các PH viện dẫn đến PPS. rpl_present_in_ph_flag bằng 0 xác định báo hiệu danh sách ảnh tham chiếu không xuất hiện trong các PH viện dẫn đến PPS mà có thể xuất hiện trong các tiêu đề lát viện dẫn đến PPS.

sao_present_in_ph_flag bằng 1 xác định các phần tử cú pháp để có thể sử dụng SAO không xuất hiện trong các tiêu đề lát viện dẫn đến PPS mà có thể xuất hiện trong các PH viện dẫn đến PPS. sao_present_in_ph_flag bằng 0 xác định các phần tử cú pháp để có thể sử dụng SAO không xuất hiện trong các PH viện dẫn đến PPS mà có thể xuất hiện trong các tiêu đề lát viện dẫn đến PPS.

alf_present_in_ph_flag bằng 1 xác định các phần tử cú pháp để có thể sử dụng ALF không xuất hiện trong các tiêu đề lát viện dẫn đến PPS mà có thể xuất hiện trong các PH viện dẫn đến PPS. alf_present_in_ph_flag bằng 0 xác định các

phần tử cú pháp để có thể sử dụng ALF không xuất hiện trong các PH viện dẫn đến PPS mà có thể xuất hiện trong các tiêu đề lát viện dẫn đến PPS.

...

weighted_pred_table_present_in_ph_flag bằng 1 xác định rằng bảng dự báo có trọng số không xuất hiện trong các tiêu đề lát viện dẫn đến PPS mà có thể xuất hiện trong các PH viện dẫn đến PPS. weighted_pred_table_present_in_ph_flag bằng 0 xác định rằng bảng dự báo có trọng số không xuất hiện trong các PH viện dẫn đến PPS mà có thể xuất hiện trong tiêu đề láts viện dẫn đến PPS. Khi không xuất hiện, giá trị của weighted_pred_table_present_in_ph_flag được suy ra bằng 0.

...

deblocking_filter_override_enabled_flag bằng 1 xác định rằng ghi chén lọc tách khối có thể xuất hiện trong các PH hoặc trong các tiêu dề lát viện dẫn đến PPS. deblocking_filter_override_enabled_flag bằng 0 xác định rằng ghi chèn lọc tách khối không xuất hiện trong các PH hoặc trong các tiêu đề lát viện dẫn đến PPS. Khi không xuất hiện, giá trị của deblocking_filter_override_enabled_flag được suy ra bằng 0.

deblocking_filter_override_present_in_ph_flag bằng 1 xác định rằng ghi chèn lọc tách khối không xuất hiện trong các tiêu đề lát viện dẫn đến PPS mà có thể xuất hiện trong các PH viện dẫn đến PPS. deblocking_filter_override_present_in_ph_flag bằng 0 xác định rằng ghi chèn lọc tách khối không xuất hiện trong các PH viện dẫn đến PPS mà có thể xuất hiện trong các tiêu đề lát viện dẫn đến PPS.

...

picture_header_rbsp() {	**Mô tả**
...	
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	
if(rpl_present_in_ph_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	

pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
...	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag)	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
...	
}	

slice_header() {	**Mô tả**
...	
if(!rpl_present_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	

for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 &&	
(i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(rpl_present_in_ph_flag || ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
…	
if(slice_type != I) {	
…	
if(((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B)) && !weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
}	
if(sps_sao_enabled_flag && !sao_present_in_ph_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !alf_present_in_ph_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	

slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !deblocking_filter_override_present_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)
}	
}	
...	
}	

Cú pháp tùy chọn cho tiêu đề ảnh như sau:

picture_header_rbsp() {	**Mô tả**
...	
if(rpl_present_in_ph_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	
pic_alf_enabled_flag	u(1)

if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
...	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag)	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
...	
}	

Trong ví dụ khác, báo hiệu các phần tử tiêu đề ảnh và tiêu đề lát có thể được kết hợp trong một quá trình.

Ví dụ này đưa vào cờ ("picture_header_in_slice_header_flag") mà chỉ báo liệu các tiêu đề ảnh và lát có được kết hợp hay không. Cú pháp cho dòng bit theo ví dụ này như sau:

Cú pháp RBSP tiêu đề ảnh

picture_header_rbsp() {	**Mô tả**
picture_header_structure()	
}	

Cú pháp cấu trúc tiêu đề ảnh

picture_header_structure() {	**Mô tả**
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
...	
}	

Cú pháp tiêu đề lát chung

slice_header() {	**Mô tả**
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag || NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag &&((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
…	
}	

Ngữ nghĩa cho picture_header_in_slice_header_flag và các ràng buộc dòng bit liên quan như sau:

picture_header_in_slice_header_flag bằng 1 xác định rằng tiêu đề ảnh cấu trúc cú pháp xuất hiện trong tiêu đề lát. picture_header_in_slice_header_flag bằng 0 xác định rằng tiêu đề ảnh cấu trúc cú pháp không xuất hiện trong tiêu đề lát.

Tương hợp dòng bit yêu cầu rằng giá trị của picture_header_in_slice_header_flag là giống nhau cho tất cả các lát của CLVS.

Khi picture_header_in_slice_header_flag bằng 1, tương hợp dòng bit yêu cầu rằng không đơn vị NAL nào có loại đơn vị NAL bằng PH_NUT xuất hiện trong CLVS.

Khi picture_header_in_slice_header_flag bằng 0, tương hợp dòng bit yêu cầu rằng đơn vị NAL có loại đơn vị NAL bằng PH_NUT xuất hiện trong PU, đứng trước đơn vị VCL NAL thứ nhất của PU.

Kết hợp theo các khía cạnh của các ví dụ này như sau.

Khi picture_header_in_slice_header_flag bằng 0, các cờ xác định liệu phần tử cú pháp của công cụ mã liên quan xuất hiện trong PH hoặc SH (nhưng không phải cả hai);

Ngược lại (khi picture_header_in_slice_header_flag bằng 1), các cờ này được suy ra bằng 0 chỉ báo báo hiệu tham số công cụ ở mức lát.

Kết hợp tùy chọn như sau:

Khi picture_header_in_slice_header_flag bằng 0, các cờ xác định liệu phần tử cú pháp của công cụ mã liên quan xuất hiện trong PH hoặc SH (nhưng không phải cả hai);

Ngược lại (khi picture_header_in_slice_header_flag bằng 1), các cờ này được suy ra bằng 0 chỉ báo báo hiệu tham số công cụ ở mức tiêu đề ảnh.

Kết hợp này có cú pháp sau:

Cú pháp tập hợp tham số ảnh

pic_parameter_set_rbsp() {	**Mô tả**
picture_header_in_slice_header_flag	u(1)
…	
if (**picture_header_in_slice_header_flag**)	
{	
rpl_present_in_ph_flag	u(1)
sao_present_in_ph_flag	u(1)
alf_present_in_ph_flag	u(1)
}	
…	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
if(pps_weighted_pred_flag || pps_weighted_bipred_flag || rpl_present_in_ph_flag)	
weighted_pred_table_present_in_ph_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_present_in_ph_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
…	
}	

Trong ví dụ này, phép kiểm tra liệu dự báo có trọng số được kích hoạt được thực hiện bằng cách chỉ báo số lượng mục trong danh sách ảnh tham chiếu được tạo tham chiếu với dự báo có trọng số.

Cú pháp và ngữ nghĩa trong ví dụ này được định nghĩa như sau:

pred_weight_table() {	**Mô tả**
luma_log2_weight_denom	ue(v)
if(ChromaArrayType != 0)	
delta_chroma_log2_weight_denom	se(v)
num_l0_weighted_ref_pics	ue(v)
for(i = 0; i < num_l0_weighted_ref_pics; i++)	
luma_weight_l0_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[0]; i++)	
chroma_weight_l0_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[0]; i++) {	
if(luma_weight_l0_flag[i]) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if(chroma_weight_l0_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
num_l1_weighted_ref_pics	ue(v)
for(i = 0; i < num_l1_weighted_ref_pics; i++)	
luma_weight_l1_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[1]; i++)	
chroma_weight_l1_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[1]; i++) {	
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_weight_l1_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	

...

num_l0_weighted_ref_pics xác định số lượng ảnh tham chiếu trong danh sách ảnh tham chiếu 0 có trọng số. Giá trị của num_l0_weighted_ref_pics sẽ nằm trong khoảng từ 0 đến MaxDecPicBuffMinus1 + 14, bao gồm hai giá trị đầu mút.

Tương hợp dòng bit yêu cầu rằng khi xuất hiện, giá trị của num_l0_weighted_ref_pics sẽ không nhỏ hơn số lượng ảnh tham chiếu hoạt động cho L0 của bất kỳ lát nào trong ảnh được liên kết với tiêu đề ảnh.

num_l1_weighted_ref_pics xác định số lượng ảnh tham chiếu trong danh sách ảnh tham chiếu 1 có trọng số. Giá trị của num_l1_weighted_ref_pics sẽ nằm trong khoảng từ 0 đến MaxDecPicBuffMinus1 + 14, bao gồm hai giá trị đầu mút.

Tương hợp dòng bit yêu cầu rằng khi xuất hiện, giá trị của num_l1_weighted_ref_pics sẽ không nhỏ hơn số lượng ảnh tham chiếu hoạt động cho L1 của bất kỳ lát nào trong ảnh được liên kết với tiêu đề ảnh.

...

MaxNumGeoMergeCand được gán bằng 0 khi num_l0_weighted_ref_pics hoặc num_l1_weighted_ref_pics khác 0. Cú pháp sau là ví dụ cách thức lệ thuộc này có thể được sử dụng:

if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_geo_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_geo_cand_plus1 && **num_l0_weighted_ref_pics==0 && num_l1_weighted_ref_pics==0)**	
pic_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	

Ngữ nghĩa của pic_max_num_merge_cand_minus_max_num_geo_cand theo phương án thực hiện giống như đối với các phương án thực hiện trước đó.

Trong ví dụ, các phần tử cú pháp trong và ngoài được báo hiệu có điều kiện nếu chỉ các loại lát cụ thể được sử dụng trong ảnh được liên kết với PH.

Cú pháp cho ví dụ này dưới đây:

picture_header_rbsp() {	**Mô tả**
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	

ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_virtual_boundaries_present_flag) {	
ph_virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(pic_intra_slice_present_flag) {	
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)

pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
pic_cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(pic_inter_slice_present_flag) {	
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	

if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
...	

7.3.7.1 Cú pháp tiêu đề lát chung

slice_header() {	**Mô tả**
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag || NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
if(pic_inter_slice_present_flag)	
slice_type	ue(v)
...	

7.4.3.6 Ngữ nghĩa RBSP tiêu đề ảnh

pic_inter_slice_present_flag bằng 1 xác định rằng một hoặc nhiều lát có slice_type bằng 0 (B) hoặc 1 (P) có thể xuất hiện trong ảnh được liên kết với PH. pic_inter_slice_present_flag bằng 0 xác định rằng không lát nào với slice_type bằng 0 (B) hoặc 1 (P) có thể xuất hiện trong ảnh được liên kết với PH.

pic_intra_slice_present_flag bằng 1 xác định rằng một hoặc nhiều lát có slice_type bằng 2 (I) có thể xuất hiện trong ảnh được liên kết với PH. pic_intra_slice_present_flag bằng 0 xác định rằng không lát nào với slice_type bằng 2 (I) có thể xuất hiện trong ảnh được liên kết với PH. Khi không xuất hiện, giá trị của pic_intra_slice_only_flag được suy ra bằng 1.

Lưu ý –: Các giá trị của both pic_inter_slice_present_flag và pic_intra_slice_present_flag được gán bằng 1 trong tiêu đề ảnh được liên kết với ảnh chứa một hoặc nhiều ảnh thành phần chứa (các) lát được tạo mã trong có thể được hợp nhất với một hoặc nhiều ảnh thành phần chứa (các) lát được tạo mã ngoài.

7.4.8.1 Ngữ nghĩa tiêu đề lát chung

slice_type xác định loại mã của lát theo Bảng 7-5.

Bảng 7-5 – Liên kết tên với slice_type

slice_type	Tên slice_type
0	B (lát B)
1	P (lát P)
2	I (lát I)

Khi nal_unit_type là giá trị của nal_unit_type trong khoảng từ IDR_W_RADL đến CRA_NUT, bao gồm hai giá trị đầu mút, và ảnh hiện tại là ảnh thứ nhất trong AU, slice_type phải bằng 2.

Khi không xuất hiện, giá trị của slice_type được suy ra bằng 2.

Khi pic_intra_slice_present_flag bằng 0, giá trị của slice_type phải nằm trong khoảng từ 0 đến 1, bao gồm hai giá trị đầu mút.

Ví dụ này có thể được kết hợp với báo hiệu pred_weight_table() trong tiêu đề ảnh. Báo hiệu pred_weight_table() trong tiêu đề ảnh được bộc lộ trong các ví dụ trước đó.

Cú pháp lấy làm ví dụ như sau:

picture_header_rbsp() {	**Mô tả**
...	
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

Khi chỉ báo xuất hiện pred_weight_table() trong tiêu đề ảnh, cú pháp sau có thể được sử dụng.

picture_header_rbsp() {	**Mô tả**
...	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && pic_inter_slice_present_flag)	
pred_weight_table()	
...	

Các ví dụ tùy chọn có thể sử dụng cú pháp sau:

picture_header_rbsp() {	**Mô tả**
...	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	

if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

Các ví dụ tùy chọn có thể sử dụng cú pháp sau:

picture_header_rbsp() {	**Mô tả**
...	
pic_inter_bipred_slice_present_flag	**u(1)**
if (!pic_inter_bipred_slice_present_flag)	
pic_inter_slice_present_flag	**u(1)**
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

Trong cú pháp nêu trên, pic_inter_bipred_slice_present_flag chỉ báo sẹ xuất hiện của tất cả các loại lát, I, B và P viện dẫn đến tiêu đề ảnh.

Khi pic_inter_bipred_slice_present_flag bằng 0, ảnh bao gồm chỉ các lát loại I hoặc B.

Trong trường hợp này, các chế độ phi chữ nhật bị vô hiệu hóa.

Ví dụ bộc lộ kết hợp các ví dụ nêu trên. Cú pháp lấy làm ví dụ được mô tả như sau:

picture_header_rbsp() {	**Mô tả**
...	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
~~**pic_rpl_present_flag**~~	~~u(1)~~
~~if(pic_rpl_present_flag) {~~	
if(rpl_present_in_ph_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)

pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
…	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag && pic_inter_slice_present_flag)	
pred_weight_table()	
…	
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {	
~~**pic_sao_enabled_present_flag**~~	~~u(1)~~
~~if(pic_sao_enabled_present_flag) {~~	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
~~}~~	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	
~~**pic_alf_enabled_present_flag**~~	~~u(1)~~
~~if(pic_alf_enabled_present_flag) {~~	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
~~}~~	
}	
…	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag) ~~{~~	
~~**pic_deblocking_filter_override_present_flag**~~	~~u(1)~~
~~if(pic_deblocking_filter_override_present_flag) {~~	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
~~}~~	
~~}~~	
…	

}	

Trong ví dụ, lựa chọn chế độ phi chữ nhật (chẳng hạn, GEO) viện dẫn đến ảnh không có hệ số dự báo có trọng số được phép.

Trong ví dụ này, ngữ nghĩa được định nghĩa như sau:

7.4.10.7 Ngữ nghĩa dữ liệu hợp nhất

...

Biến MergeGeoFlag[x0][y0], xác định liệu bù chuyển động dựa trên hình dạng hình học được sử dụng để tạo các mẫu dự báo của CU hiện tại, khi giải mã lát B, được dẫn xuất như sau:

– Nếu tất cả các điều kiện sau là đúng, MergeGeoFlag[x0][y0] được gán bằng 1:

– sps_geo_enabled_flag bằng 1.
– slice_type bằng B.
– general_merge_flag[x0][y0] bằng 1.
– MaxNumGeoMergeCand lớn hơn hoặc bằng 2.
– cbWidth lớn hơn hoặc bằng 8
– cbHeight lớn hơn hoặc bằng 8
– cbWidth nhỏ hơn 8*cbHeight
– cbHeight nhỏ hơn 8*cbWidth
– regular_merge_flag[x0][y0] bằng 0.
– merge_subblock_flag[x0][y0] bằng 0.
– ciip_flag[x0][y0] bằng 0.
– Ngược lại, MergeGeoFlag[x0][y0] được gán bằng 0.

Tương hợp dòng bit yêu cầu rằng nếu một trong các cờ có trọng số tường minh độ sáng hoặc sắc độ của CU là true, MergeGeoFlag[x0][y0] phải bằng 0.

Trong ví dụ, một phần của bản mô tả VVC được giải thích như sau:

8.5.7 Quá trình giải mã cho các khối ngoài hình học

8.5.7.1 Tổng quát

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

Đầu vào cho quá trình này là:

– vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái CB hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

– biến cbWidth xác định chiều rộng của CB hiện tại trong các mẫu độ sáng,

– biến cbHeight xác định chiều cao của CB hiện tại trong các mẫu độ sáng,

– các vectơ chuyển động độ sáng có độ chính xác mẫu phân số 1/16 mvA và mvB,

– các vectơ chuyển động sắc độ mvCA và mvCB,

– các chỉ số tham chiếu refIdxA và refIdxB,

– các cờ danh sách dự báo predListFlagA và predListFlagB.

...

Giả sử $predSamplesLA_L$ và $predSamplesLB_L$ là các mảng (cbWidth)x(cbHeight) của các giá trị mẫu độ sáng được dự báo và, $predSamplesLA_{Cb}$, $predSamplesLB_{Cb}$, $predSamplesLA_{Cr}$ và $predSamplesLB_{Cr}$ là các mảng (cbWidth/SubWidthC)x(cbHeight/SubHeightC) của các giá trị mẫu sắc độ được dự báo.

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

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

...

2. Góc phân vùng và khoảng cách của biến chế độ hình học hợp nhất angleIdx và distanceIdx được gán theo giá trị của merge_geo_partition_idx[xCb][yCb] như được xác định trong Bảng 36.

3. Biến explictWeightedFlag được dẫn xuất như sau:

lumaWeightedFlagA = predListFlagA ? luma_weight_l1_flag[refIdxA] : luma_weight_l0_flag[refIdxA]

lumaWeightedFlagB = predListFlagB ? luma_weight_l1_flag[refIdxB] : luma_weight_l0_flag[refIdxB]

chromaWeightedFlagA = predListFlagA ? chroma_weight_l1_flag[refIdxA] : chroma weight_l0_flag[refIdxA]

chromaWeightedFlagB = predListFlagB ? chroma_weight_l1_flag[refIdxB] : chroma weight_l0_flag[refIdxB]

weightedFlag = lumaWeightedFlagA || lumaWeightedFlagB || chromaWeightedFlagA || chromaWeightedFlagB

4. Các mẫu dự báo bên trong CB độ sáng hiện tại, $predSamples_L[x_L][y_L]$ với $x_L = 0..cbWidth - 1$ và $y_L = 0..cbHeight - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất GEO được xác định trong mệnh đề 8.5.7.2 nếu weightedFlag bằng 0, và quá trình dự báo mẫu có trọng số tường minh trong mệnh đề 8.5.6.6.3 nếu weightedFlag bằng 1 có chiều rộng CB nCbW được gán bằng cbWidth, chiều cao CB nCbH được gán bằng cbHeight, các mảng mẫu $predSamplesLA_L$ và $predSamplesLB_L$, và biến angleIdx và distanceIdx, và cIdx bằng 0 làm các đầu vào.

5. Các mẫu dự báo bên trong khối mã Cb thành phần sắc độ hiện tại, $predSamples_{Cb}[x_C][y_C]$ với $x_C = 0..cbWidth/SubWidthC - 1$ và $y_C = 0..cbHeight/SubHeightC - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất GEO được xác định trong mệnh đề 8.5.7.2 nếu weightedFlag bằng 0, và quá trình dự báo mẫu có trọng số tường minh trong mệnh đề 8.5.6.6.3 nếu weightedFlag bằng 1 có chiều rộng CB nCbW được gán bằng cbWidth/SubWidthC, chiều cao CB nCbH được gán bằng cbHeight/SubHeightC, các mảng mẫu $predSamplesLA_{Cb}$ và $predSamplesLB_{Cb}$, và các biến angleIdx và distanceIdx, và cIdx bằng 1 làm các đầu vào.

6. Các mẫu dự báo bên trong khối mã Cr thành phần sắc độ hiện tại, $predSamples_{Cr}[x_C][y_C]$ với $x_C = 0..cbWidth/SubWidthC - 1$ và $y_C = 0..cbHeight/SubHeightC - 1$, được dẫn xuất bằng cách gọi quá trình dự báo mẫu có trọng số đối với chế độ hợp nhất GEO được xác định trong mệnh đề 8.5.7.2 nếu weightedFlag bằng 0, và quá trình dự báo mẫu có trọng số tường minh trong mệnh đề 8.5.6.6.3 nếu weightedFlag bằng 1 có chiều rộng CB nCbW được gán bằng cbWidth/SubWidthC, chiều cao CB nCbH được gán bằng cbHeight/SubHeightC, các mảng mẫu $predSamplesLA_{Cr}$ và $predSamplesLB_{Cr}$, và các biến angleIdx và distanceIdx, và cIdx bằng 2 làm các đầu vào.

7. Quá trình lưu trữ MV cho chế độ hình học hợp nhất được xác định trong mệnh đề 8.5.7.3 được gọi với vị trí CB độ sáng (xCb, yCb), chiều rộng CB độ sáng cbWidth, chiều cao CB độ sáng cbHeight, hướng phân vùng angleIdx và

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

Bảng 36 – Mô tả giá trị angleIdx và distanceIdx dựa trên merge_geo_partition_idx.

merge_geo_partition_idx	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
angleIdx	0	0	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
distanceIdx	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2
merge_geo_partition_idx	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**
angleIdx	4	6	6	8	8	8	8	9	9	9	9	10	10	10	10	11	11
distanceIdx	3	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
merge_geo_partition_idx	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
angleIdx	11	11	12	12	13	13	13	14	14	14	15	15	15	16	16	16	18
distanceIdx	2	3	1	3	1	2	3	1	2	3	1	2	3	1	2	3	1
merge_geo_partition_idx	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**				
angleIdx	18	20	20	20	21	21	21	22	22	22	23	23	23				
distanceIdx	3	1	2	3	1	2	3	1	2	3	1	2	3				

8.5.6.6.3 Quá trình dự báo mẫu có trọng số tường minh

Đầu vào cho quá trình này là:

– Hai biến nCbW và nCbH xác định chiều rộng và chiều cao của CB hiện tại,

– Hai mảng (nCbW)x(nCbH) predSamplesL0 và predSamplesL1,

– các cờ sử dụng danh sách dự báo, predFlagL0 và predFlagL1,

– các chỉ số tham chiếu, refIdxL0 và refIdxL1,

– biến cIdx xác định chỉ số thành phần màu,

– chiều sâu bit mẫu, bitDepth.

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

Biến shift1 được gán bằng Max(2, 14 – bitDepth).

Các biến log2Wd, o0, o1, w0 và w1 được dẫn xuất như sau:

– Nếu cIdx bằng 0 cho các mẫu độ sáng, phương trình sau áp dụng:

log2Wd = luma_log2_weight_denom + shift1 (1010)

w0 = LumaWeightL0[refIdxL0] (1011)

w1 = LumaWeightL1[refIdxL1] (1012)

o0 = luma_offset_l0[refIdxL0] << (bitDepth − 8) (1013)

o1 = luma_offset_l1[refIdxL1] << (bitDepth − 8) (1014)

– Ngược lại (cIdx không bằng 0 cho các mẫu sắc độ), phương trình sau áp dụng:

log2Wd = ChromaLog2WeightDenom + shift1 (1015)

w0 = ChromaWeightL0[refIdxL0][cIdx − 1] (1016)

w1 = ChromaWeightL1[refIdxL1][cIdx − 1] (1017)

o0 = ChromaOffsetL0[refIdxL0][cIdx − 1] << (bitDepth − 8) (1018)

o1 = ChromaOffsetL1[refIdxL1][cIdx − 1] << (bitDepth − 8) (1019)

Mẫu dự báo pbSamples[x][y] với x = 0..nCbW − 1 và y = 0..nCbH − 1 được dẫn xuất như sau:

– Nếu predFlagL0 bằng 1 và predFlagL1 bằng 0, các giá trị mẫu dự báo được dẫn xuất như sau:

if(log2Wd >= 1)

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1,

((predSamplesL0[x][y] * w0 + $2^{log2Wd-1}$) >> log2Wd) + o0) (1020)

else

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, predSamplesL0[x][y] * w0 + o0)

– Ngược lại, nếu predFlagL0 bằng 0 và predFlagL1 bằng 1, các giá trị mẫu dự báo được dẫn xuất như saus:

if(log2Wd >= 1)

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1,

((predSamplesL1[x][y] * w1 + $2^{log2Wd-1}$) >> log2Wd) + o1) (1021)

else

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, predSamplesL1[x][y] * w1 + o1)

– Ngược lại (predFlagL0 bằng 1 và predFlagL1 bằng 1), các giá trị mẫu dự báo được dẫn xuất như sau:

pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1,
(predSamplesL0[x][y] * w0 + predSamplesL1[x][y] * w1 +
((o0 + o1 + 1) << log2Wd)) >> (log2Wd + 1)) (1022)

Ví dụ này bộc lộ cú pháp của tham số dữ liệu hợp nhất bao gồm kiểm tra biến mà chỉ báo sự có mặt của chế độ hợp nhất phi chữ nhật (chẳng hạn, chế độ GEO). Ví dụ cú pháp được nêu dưới đây:

merge_data(x0, y0, cbWidth, cbHeight, chType) {	**Mô tả**
if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(cbWidth < 128 && cbHeight < 128 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64) || (sps_geo_enabled_flag && MaxNumGeoMergeCand > 1 && cbWidth>=8 && cbHeight >=8 && cbWidth < 8*cbHeight && cbHeight < 8*cbWidth && slice_type == B)))	
regular_merge_flag[x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && sps_geo_enabled_flag && MaxNumGeoMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < 8*cbHeight && cbHeight < 8*cbWidth && cbWidth < 128 && cbHeight < 128)	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && MaxNumGeoMergeCand > 1) {	
merge_geo_partition_idx[x0][y0]	ae(v)
merge_geo_idx0[x0][y0]	ae(v)
if(MaxNumGeoMergeCand > 2)	
merge_geo_idx1[x0][y0]	ae(v)

}	
}	
}	
}	
}	

Biến MaxNumGeoMergeCand được dẫn xuất theo ví dụ bất kỳ trong các ví dụ trước đó.

Biến tùy chọn SliceMaxNumGeoMergeCand được dẫn xuất từ biến MaxNumGeoMergeCand có thể được sử dụng. Giá trị của MaxNumGeoMergeCand thu được ở các mức báo hiệu cao hơn (chẳng hạn, PH, PPS hoặc SPS).

Trong ví dụ, SliceMaxNumGeoMergeCand được dẫn xuất dựa trên giá trị của MaxNumGeoMergeCand và các phép kiểm tra bổ sung được thực hiện cho lát.

Chẳng hạn, SliceMaxNumGeoMergeCand = (num_l0_weighted_ref_pics>0 || num_l1_weighted_ref_pics>0) ? 0 : MaxNumGeoMergeCand.

Trong ví dụ khác, biểu thức sau được sử dụng để xác định giá trị MaxNumGeoMergeCand:

SliceMaxNumGeoMergeCand = (!pic_inter_slice_present_flag) ? 0: MaxNumGeoMergeCand.

Trong ví dụ,

Bảng cú pháp sau được định nghĩa:

picture_header_rbsp() {	**Mô tả**
...	
pic_inter_bipred_slice_present_flag	
if (!pic_inter_bipred_slice_present_flag)	
pic_inter_slice_present_flag	
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag || pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

Biến MaxNumGeoMergeCand được dẫn xuất như sau:

SliceMaxNumGeoMergeCand = (!pic_inter_bipred_slice_present_flag) ? 0: MaxNumGeoMergeCand.

Phương pháp chỉ báo số lượng ứng viên hợp nhất cho các chế độ chữ nhật và phi chữ nhật được bộc lộ. Số lượng ứng viên hợp nhất cho các chế độ chữ nhật và phi chữ nhật là tương thuộc, và có thể không cần chỉ báo số lượng ứng viên hợp nhất cho các chế độ phi chữ nhật trong trường hợp khi được chỉ báo rằng số lượng ứng viên hợp nhất cho các chế độ chữ nhật thấp hơn ngưỡng.

Cụ thể là đối với các chế độ hợp nhất TPM hoặc hình học, cần có ít nhất hai ứng viên cho chế độ hợp nhất, do khối được dự báo nhờ sử dụng bất kỳ chế độ nào trong các chế độ hợp nhất phi chữ nhật này đòi hỏi hai bộ dự báo ngoài có các MV khác nhau được xác định cho chúng. Theo phương án thực hiện, khi số lượng ứng viên chế độ hợp nhất được chỉ báo trong SPS, cú pháp sau có thể được sử dụng:

7.3.2.3 Cú pháp RBSP tập hợp tham số chuỗi

seq_parameter_set_rbsp() {	**Mô tả**
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
sps_seq_parameter_set_id	u(4)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
ref_pic_resampling_enabled_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_log2_ctu_size_minus5	u(2)
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_present_flag	u(1)
if(sps_subpic_id_present_flag) {	
sps_subpic_id_signalling_present_flag	u(1)

if(sps_subpic_id_signalling_present_flag) {	
sps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
bit_depth_minus8	ue(v)
min_qp_prime_ts_minus4	ue(v)
sps_weighted_pred_flag	u(1)
sps_weighted_bipred_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	u(4)
sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
dpb_parameters(0, sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
sps_joint_cbcr_enabled_flag	u(1)

if(ChromaArrayType != 0) {	
same_qp_table_for_chroma	u(1)
numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbcr_enabled_flag ? 3 : 2)	
for(i = 0; i < numQpTables; i++) {	
qp_table_start_minus26[i]	se(v)
num_points_in_qp_table_minus1[i]	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
delta_qp_in_val_minus1[i][j]	ue(v)
delta_qp_diff_val[i][j]	ue(v)
}	
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag)	
sps_bdpcm_enabled_flag	u(1)
if(sps_bdpcm_enabled_flag && chroma_format_idc == 3)	
sps_bdpcm_chroma_enabled_flag	u(1)
sps_ref_wraparound_enabled_flag	u(1)
if(sps_ref_wraparound_enabled_flag)	
sps_ref_wraparound_offset_minus1	ue(v)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
if(sps_bdof_enabled_flag)	
sps_bdof_pic_present_flag	u(1)
sps_smvd_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
if(sps_dmvr_enabled_flag)	
sps_dmvr_pic_present_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(chroma_format_idc == 1) {	
sps_chroma_horizontal_collocated_flag	u(1)
sps_chroma_vertical_collocated_flag	u(1)
}	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
sps_six_minus_max_num_merge_cand	ue(v)
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	

sps_five_minus_max_num_subblock_merge_cand	ue(v)
sps_affine_type_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
if(chroma_format_idc == 3) {	
sps_act_enabled_flag	u(1)
sps_palette_enabled_flag	u(1)
}	
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
if (sps_ibc_enabled_flag)	
sps_six_minus_max_num_ibc_merge_cand	ue(v)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_scaling_list_enabled_flag	u(1)
sps_virtual_boundaries_present_flag	u(1)
if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)
if(sps_general_hrd_params_present_flag) {	
general_hrd_parameters()	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_cpb_params_present_flag	u(1)
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 : sps_max_sublayers_minus1	
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)	
}	
}	

field_seq_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters() /* Specified in ITU-T H.SEI | ISO/IEC 23002-7 */	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Theo phương án thực hiện sáng chế, các bước sau được thực hiện để chỉ báo số lượng ứng viên chế độ hợp nhất trong SPS:

– Chỉ báo số lượng ứng viên chế độ hợp nhất cho các chế độ thông thường (MaxNumMergeCand);

– Chỉ báo liệu các chế độ phi chữ nhật được kích hoạt bởi cờ kích hoạt hợp nhất phi chữ nhật (sps_geo_enabled_flag); và

– Nếu giá trị của cờ kích hoạt hợp nhất phi chữ nhật khác 0 và khi số lượng ứng viên chế độ hợp nhất đối với các chế độ hợp nhất thông thường lớn hơn ngưỡng thứ nhất, chỉ báo số lượng chế độ phi chữ nhật (sps_max_num_merge_cand_minus_max_num_geo_cand).

trong đó chỉ báo cờ kích hoạt hợp nhất phi chữ nhật được thực hiện khi số lượng ứng viên chế độ hợp nhất đối với các chế độ thông thường lớn hơn giá trị ngưỡng thứ hai, chẳng hạn, 1.

Theo phương án thực hiện 1, chuỗi bước này được thể hiện dưới dạng phần sau của cú pháp SPS trong mô tả VVC:

…	
if (MaxNumMergeCand > 1)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
…	

Theo phương án thực hiện, thực hiện hai bước kiểm tra tuần tự, và bước kiểm tra thứ hai phụ thuộc vào giá trị của cờ được báo hiệu hoặc không theo kết quả kiểm tra thứ nhất.

Phương án thực hiện 2 thực hiện bước kiểm tra thứ hai khác khi so sánh với quá trình được mô tả cho Phương án thực hiện 1. Cụ thể là, phương án thực hiện

1 sử dụng điều kiện "lớn hơn" thay cho "lớn hơn hoặc bằng". Chuỗi bước này được thể hiện dưới dạng phần sau của cú pháp SPS trong mô tả VVC:

...	
if (MaxNumMergeCand > 1)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

Phương án thực hiện 3 khác với Phương án thực hiện 1 rằng bước kiểm tra thứ hai không được thực hiện khi phép kiểm tra thứ nhất thu được giá trị false, giá trị cờ kích hoạt hợp nhất phi chữ nhật (**sps_geo_enabled_flag**) được xác định sau khi quá trình dẫn xuất giá trị MaxNumMergeCand từ phần tử cú pháp sps_six_minus_max_num_merge_cand được hoàn thành chính là ưu điểm kỹ thuật, do giá trị của sps_geo_enabled_flag không được viện dẫn đối với một số giá trị của MaxNumMergeCand và do vậy có thể được bỏ qua khi xử lý trong quá trình phân tách. Chuỗi bước này được thực hiện theo Phương án thực hiện 3 được thể hiện dưới dạng phần sau của cú pháp SPS của đặc tả VVC:

...	
if (MaxNumMergeCand > 1) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

Phương án thực hiện 4 là kết hợp theo các khía cạnh của phương án thực hiện 2 và phương án thực hiện 3. Chuỗi các bước được thực hiện theo phương án thực hiện 4 được thể hiện dưới dạng phần sau của cú pháp SPS của đặc tả VVC:

...	
if (MaxNumMergeCand > 1) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

Các phương án thực hiện từ 5 đến 8 bộc lộ các dạng khác nhau của các bước kiểm tra thứ nhất và thứ hai. Các phương án thực hiện này có thể được giải thích như sau:

Phương án thực hiện 5

...	
if (MaxNumMergeCand >= 2)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

Phương án thực hiện 6

...	
if (MaxNumMergeCand >= 2)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

Phương án thực hiện 7

...	
if (MaxNumMergeCand >= 2) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

Phương án thực hiện 8

...	
if (MaxNumMergeCand >= 2) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

Theo phương án thực hiện như được thể hiện trên Fig. 15, phương pháp thu thập số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học để giải mã viđeo được bộc lộ, phương pháp bao gồm bước:

S1501: thu thập dòng bit cho chuỗi viđeo.

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, cặp 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 thành biểu diễn của chuỗi khối truy nhập (access unit, AU) tạo một hoặc nhiều chuỗi viđeo được tạo mã (coded video sequence, CVS).

Theo 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 hóa, phía bộ mã hóa 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 của 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 chúng 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 trên "cơ bản". Định dạng dòng đơn vị NAL bao gồm chuỗi cấu trúc cú pháp được gọi là các đơn vị NAL. Chuỗi này được sắp thứ tự theo thứ tự giải mã. Có các ràng buộc được áp lên 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 thứ tự các đơn vị NAL theo thứ tự giải mã và tạo tiền tố mỗi đơn vị NAL với tiền tố mã bắt đầu và 0 hoặc nhiều byte có giá trị 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 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 viđeo này được biểu diễn bằng 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 màu ba màu hoặc đơn sắc không xác định khác (chẳng hạn, YZX, cũng được biết 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 được sử dụng. Phương pháp biểu diễn màu thực đang được 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.SEI | ISO/IEC 23002-7.

S1502: thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit.

Bộ chỉ báo thứ nhất biểu diễn số lượng lớn nhất của các ứng viên dự báo vectơ chuyển động hợp nhất (merging motion vector prediction, MVP).

Trong ví dụ, bộ chỉ báo thứ nhất được biểu diễn theo biến MaxNumMergeCand.

Trong ví dụ, số lượng lớn nhất của các ứng viên MVP hợp nhất, MaxNumMergeCand, được dẫn xuất như sau:

MaxNumMergeCand = 6 − sps_six_minus_max_num_merge_cand.

Trong đó sps_six_minus_max_num_merge_cand xác định số lượng lớn nhất của các ứng viên MVP được hỗ trợ trong SPS được trừ đi khỏi 6. Giá trị của sps_six_minus_max_num_merge_cand sẽ nằm trong khoảng từ 0 đến 5, bao gồm hai giá trị đầu mút.

Trong ví dụ, sps_six_minus_max_num_merge_cand được phân tách từ cấu trúc cú pháp RBSP tập hợp tham số chuỗi trong dòng bit.

S1503: thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit.

Bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo.

Trong ví dụ, bộ chỉ báo thứ hai được biểu diễn theo sps_geo_enabled_flag (sps_gpm_enabled_flag). sps_geo_enabled_flag bằng 1 xác định rằng bù chuyển động dựa trên phân vùng hình học có được kích hoạt cho CLVS và merge_gpm_partition_idx, merge_gpm_idx0, và merge_gpm_idx1 có thể xuất hiện trong cú pháp CU của CLVS. sps_geo_enabled_flag bằng 0 xác định rằng bù chuyển động dựa trên phân vùng hình học bị vô hiệu hóa cho CLVS và merge_gpm_partition_idx, merge_gpm_idx0, và merge_gpm_idx1 không xuất hiện trong cú pháp CU của CLVS. Khi không xuất hiện, giá trị của sps_geo_enabled_flag được suy ra bằng 0.

Theo một cách thực hiện, bước thu thập giá trị của bộ chỉ báo thứ hai được thực hiện sau bước thu thập giá trị của bộ chỉ báo thứ nhất.

Theo một cách thực hiện, giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit.

Theo một cách thực hiện, giá trị của bộ chỉ báo thứ hai được phân tách từ SPS của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng. Ngưỡng là giá trị nguyên, trong ví dụ, ngưỡng bằng 2.

Chẳng hạn, giá trị của bộ chỉ báo thứ hai sps_gpm_enabled_flag thu được theo,

Cú pháp RBSP SPS

if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)

S1504: phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit.

Theo một cách thực hiện, phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

Ngưỡng là giá trị nguyên, giá trị định trước là giá trị nguyên. Trong ví dụ, ngưỡng bằng 2.

Trong ví dụ, giá trị định trước bằng 1.

Trong ví dụ, giá trị của bộ chỉ báo thứ ba thu được từ SPS của dòng bit

Trong ví dụ, bộ chỉ báo thứ ba được biểu diễn theo sps_max_num_merge_cand_minus_max_num_geo_cand (sps_max_num_merge_cand_minus_max_num_gpm_cand).

Chẳng hạn, giá trị của bộ chỉ báo thứ ba sps_max_num_merge_cand_minus_max_num_gpm_cand thu được theo,

Cú pháp RBSP SPS

if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm bước: gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Trong ví dụ, sps_max_num_merge_cand_minus_max_num_gpm_cand xác định số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được hỗ trợ trong SPS được trừ đi khỏi MaxNumMergeCand. Giá trị của sps_max_num_merge_cand_minus_max_num_gpm_cand sẽ nằm trong khoảng từ 0 đến MaxNumMergeCand – 2 bao gồm hai giá trị đầu mút.

Số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học, MaxNumGpmMergeCand (MaxNumGeoMergeCand), được dẫn xuất như sau:

```
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)
 MaxNumGpmMergeCand = MaxNumMergeCand –
            sps_max_num_merge_cand_minus_max_num_gpm_cand
else if(sps_gpm_enabled_flag && MaxNumMergeCand == 2)
 MaxNumGpmMergeCand = 2
```

else

MaxNumGpmMergeCand = 0.

Phương án thực hiện như được thể hiện trên Fig.16 đề cập đến thiết bị giải mã viđeo 1600, thiết bị giải mã viđeo bao gồm: môđun nhận 1601, được tạo cấu hình để thu được dòng bit cho chuỗi viđeo; môđun thu thập 1602, được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ nhất theo dòng bit, trong đó bộ chỉ báo thứ nhất biểu diễn số lượng lớn nhất của các ứng viên MVP; môđun thu thập 1602 được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ hai theo dòng bit, trong đó bộ chỉ báo thứ hai biểu diễn liệu bù chuyển động dựa trên phân vùng hình học được kích hoạt cho chuỗi viđeo; môđun phân tách 1603, được tạo cấu hình để phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất.

Theo phương án thực hiện, môđun thu thập 1602 được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2, khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng and khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

Theo phương án thực hiện, môđun thu thập 1602 được tạo cấu hình để gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0, khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

Theo phương án thực hiện, ngưỡng bằng 2.

Theo phương án thực hiện, giá trị định trước bằng 1.

Theo phương án thực hiện, bước thu thập giá trị của bộ chỉ báo thứ hai được thực hiện sau bước thu thập giá trị của bộ chỉ báo thứ nhất.

Theo phương án thực hiện, giá trị của bộ chỉ báo thứ hai được phân tách từ SPS của dòng bit, khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

Theo phương án thực hiện, giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit.

Theo phương án thực hiện, giá trị của bộ chỉ báo thứ ba thu được từ SPS của dòng bit.

Các chi tiết khác đối với môđun nhận 1601, môđun thu thập 1602 và môđun phân tách 1603 có thể đề cập đến các ví dụ và các cách thực hiện phương pháp nêu trên.

Ví dụ 1.Phương pháp tạo mã viđeo bao gồm báo hiệu số lượng ứng viên chế độ hợp nhất, phương pháp bao gồm:

– Chỉ báo số lượng ứng viên chế độ hợp nhất đối với các chế độ thông thường (MaxNumMergeCand);

– Chỉ báo liệu các chế độ phi chữ nhật được kích hoạt bởi cờ kích hoạt hợp nhất phi chữ nhật (sps_geo_enabled_flag); và

– Nếu giá trị cờ kích hoạt hợp nhất phi chữ nhật khác 0 và khi số lượng ứng viên chế độ hợp nhất đối với các chế độ hợp nhất thông thường lớn hơn ngưỡng thứ nhất, chỉ báo số lượng chế độ phi chữ nhật (sps_max_num_merge_cand_minus_max_num_geo_cand),

trong đó chỉ báo cờ kích hoạt hợp nhất phi chữ nhật được thực hiện khi số lượng ứng viên chế độ hợp nhất đối với các chế độ thông thường lớn hơn giá trị ngưỡng thứ hai (1).

Ví dụ 2. Phương pháp theo ví dụ 1, trong đó giá trị cờ kích hoạt hợp nhất phi chữ nhật được xác định sau khi quá trình dẫn xuất giá trị MaxNumMergeCand từ phần tử cú pháp sps_six_minus_max_num_merge_cand hoàn thành.

Ví dụ 3. Phương pháp theo ví dụ bất kỳ trong các ví dụ trước đó trong đó việc kiểm tra ngưỡng là phép so sánh liệu số lượng ứng viên chế độ hợp nhất đối với các chế độ hợp nhất thông thường lớn hơn 2.

Ví dụ 4. Phương pháp theo ví dụ 1 hoặc ví dụ 2, trong đó việc kiểm tra ngưỡng thứ nhất là phép so sánh liệu số lượng ứng viên chế độ hợp nhất đối với các chế độ hợp nhất thông thường lớn hơn hoặc bằng 3.

Trong ví dụ, phương pháp dự báo ngoài được bộc lộ, bao gồm: xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; thu thập một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối; và thu thập giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham

số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Trong ví dụ, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Trong ví dụ, xác định chế độ dự báo ngoài phi chữ nhật được phép, bao gồm: chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Trong ví dụ, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề ảnh của ảnh.

Trong ví dụ, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề lát của lát.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng hình học (GEO).

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Trong ví dụ, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Trong ví dụ, thiết bị dự báo ngoài được bộc lộ, bao gồm: bộ nhớ bất biến có các lệnh mà bộ xử lý có thể thực thi được lưu trữ trên đó; và bộ xử lý, được ghép

nối với bộ nhớ, được tạo cấu hình để thực thi các lệnh bộ xử lý thực thi được để dễ thực hiện ví dụ bất kỳ trong các ví dụ phương pháp.

Trong ví dụ, dòng bit dự báo ngoài được bộc lộ, bao gồm: thông tin chỉ báo xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; và một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối, trong đó giá trị dự báo của khối hiện tại thu được dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Trong ví dụ, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Trong ví dụ, thông tin chỉ báo bao gồm số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Trong ví dụ, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề ảnh của ảnh.

Trong ví dụ, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề lát của lát.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng hình học (GEO).

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Trong ví dụ, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Trong ví dụ, thiết bị dự báo ngoài được bộc lộ, bao gồm: môđun xác định, được tạo cấu hình để xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; môđun thu thập, được tạo cấu hình để thu được một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối; và môđun dự báo, được tạo cấu hình để thu được giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Trong ví dụ, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Trong ví dụ, môđun xác định được tạo cấu hình cụ thể để: chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Trong ví dụ, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề ảnh của ảnh.

Trong ví dụ, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề lát của lát.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Trong ví dụ, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng hình học (GEO).

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Trong ví dụ, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Trong ví dụ, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của

khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Các phương án thực hiện đề cập đến phương pháp mã hóa và/hoặc giải mã hiệu quả nhờ sử dụng thông tin liên quan tín hiệu trong các tiêu đề lát chỉ đối với các lát cho phép hoặc kích hoạt dự báo ngoài hai chiều, chẳng hạn, trong các lát dự báo hai chiều (B), cũng được gọi là các lát B.

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.10 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 viđeo, 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 viđeo, bộ mã hóa viđeo 20 được bao gồm trong thiết bị chụp 3102 có thể thực sự thực hiện quá trình mã hóa viđeo. 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 viđeo 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 viđeo, dữ liệu audio được mã hóa và dữ liệu viđeo được mã hóa không được ghép kênh. Thiết bị chụp 3102 phân tán riêng rẽ dữ liệu audio được mã hóa và dữ liệu viđeo 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 viđeo mạng (network video recorder, NVR)/bộ ghi viđeo số (digital video recorder, DVR) 3112, TV 3114, hộp tín hiệu truyền hình cáp (set top box, STB) 3116, hệ thống hội nghị viđeo 3118, hệ thống giám sát viđeo 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 viđeo 30 được bao gồm trong thiết bị đầu cuối được ưu tiên để thực hiện giải mã viđeo. 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 phận hiển thị, chẳng hạn, STB 3116, hệ thống hội nghị viđeo 3118, hoặc hệ thống giám sát viđeo 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 viđeo 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 khối 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 viđeo được mã hóa. Như nêu trên, đối với một số kịch bản thực tế, chẳng hạn trong hệ thống hội nghị viđeo, dữ liệu audio được mã hóa và dữ liệu viđeo được mã hóa không được ghép kênh. Trong tình huống thực, dữ liệu được mã hóa được truyền đến bộ giải mã viđeo 3206 và bộ giải mã audio 3208 mà không qua khối phân kênh 3204.

Qua quá trình phân kênh, dòng cơ bản viđeo (elementary stream, ES), audio ES, và một cách tùy chọn phụ đề được tạo. Bộ giải mã viđeo 3206, bao gồm bộ giải mã viđeo 30 như được giải thích theo các phương án thực hiện nêu trên, giải mã ES viđeo bằng phương pháp giải mã như được thể hiện theo các phương án thực hiện nêu trên để tạo khung viđeo, 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 viđeo có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.11) 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.11) trước khi cấp nó cho khối đồng bộ 3212.

Khối đồng bộ 3212 đồng bộ khung viđeo và khung audio, và cấp viđeo/audio cho bộ phận hiển thị viđeo/audio 3214. Chẳng hạn, khối đồng bộ 3212 đồng bộ hiển thị thông tin viđeo 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 viđeo và khung audio, và cấp viđeo/audio/phụ đề cho bộ hiển thị viđeo/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 ôtô.

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, "the first" tương đương với thứ 0, "the second" 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 >= 0 và y > 0.

Các toán tử lôgic

Các toán tử lôgic sau được định nghĩa như sau:

x && y Lôgic Boolean "and" của x và y

x || y Lôgic Boolean "or" của x và y

! Lôgic 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 >> 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 << 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:

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

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 −π÷2 đến π÷2, bao gồm hai giá trị đầu mút, theo các đơn vị rađian

Atan(x) hàm lượng giác tang ngược, hoạt động trên đối số x, có giá trị đầu ra nằm trong khoảng từ −π÷2 đến π÷2, bao gồm hai giá trị đầu mút, theo các đơn vị rađian

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

Ceil(x) số nguyên nhỏ nhất lớn hơn hoặc bằng x.

$\mathrm{Clip1_Y}(x) = \mathrm{Clip3}(0, (1 << \mathrm{BitDepth_Y}) - 1, x)$

$\mathrm{Clip1_C}(x) = \mathrm{Clip3}(0, (1 << \mathrm{BitDepth_C}) - 1, x)$

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

Cos(x) hàm lượng giác côsin hoạt động trên đối số x theo các đơn vị rađian.

Floor(x)số nguyên lớn nhất nhỏ hơn hoặc bằng x.

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

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...).

Log2(x)lôgarit cơ số 2 của x.

Log10(x) lôgarit cơ số 10 của x.

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

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

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$

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

Sin(x) hàm lượng giác sin hoạt động trên đối số x theo các đơn vị rađian

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

Swap(x, y) = (y, x)

Tan(x) hàm lượng giác tang hoạt động trên đối số x theo các đơn vị rađian

Thứ tự ưu tiên phép toán

Khi thứ tự ưu tiên trong biểu thước 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 tiính 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 lôgic

Trong phần văn bản, mệnh đề của các phép toán lôgic 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 lôgic 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 lôgic 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

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 dễ 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 lôgic 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 lôgic.

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.

Các phương án thực hiện khác sáng chế được nêu dưới đây. Cần lưu ý rằng việc đánh số được sử dụng trong phần sau không nhất thiết tuân theo việc đánh số được sử dụng trong các phần trước.

Phương án thực hiện 1: Phương pháp dự báo ngoài khối ảnh, trong đó việc báo hiệu các tham số dự báo có trọng số và kích hoạt dự báo ngoài phi chữ nhật được thực hiện cho nhóm khối được dự báo, phương pháp bao gồm bước: thu thập các tham số chế độ dự báo ngoài cho khối, trong đó việc thu thập bao gồm kiểm tra liệu chế độ dự báo ngoài phi chữ nhật có được kích hoạt cho nhóm khối bao gồm khối được dự báo; và thu thập các tham số dự báo có trọng số được liên kết với khối và các tham số chế độ dự báo ngoài đối với khối theo ảnh tham chiếu được chỉ báo cho khối và các tham số dự báo có trọng số được xác định cho nhóm khối.

Phương án thực hiện 2: Phương pháp phương án thực hiện 1, trong đó việc kích hoạt dự báo ngoài phi chữ nhật được thực hiện bằng cách chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Phương án thực hiện 3: Phương pháp phương án thực hiện 1 hoặc 2, trong đó dự báo ngoài phi chữ nhật được suy ra là bị vô hiệu hóa khi các tham số dự báo

có trọng số xác định dự báo có trọng số được kích hoạt cho ít nhất một chỉ số tham chiếu.

Phương án thực hiện 4: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 3, trong đó nhóm khối là ảnh và cả hai tham số dự báo có trọng số và việc kích hoạt các tham số chế độ dự báo ngoài phi chữ nhật được chỉ báo trong tiêu đề ảnh.

Phương án thực hiện 5: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 4, trong đó nhóm khối là lát và cả hai tham số dự báo có trọng số và việc kích hoạt các tham số chế độ dự báo ngoài phi chữ nhật được chỉ báo ở tiêu đề lát.

Phương án thực hiện 6: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 5, trong đó các tham số chế độ dự báo ngoài bao gồm chỉ số tham chiếu được sử dụng để xác định ảnh tham chiếu và thông tin vectơ chuyển động được sử dụng để xác định vị trí của khối tham chiếu trong ảnh tham chiếu.

Phương án thực hiện 7: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 6, trong đó chế độ hợp nhất phi chữ nhật là chế độ phân vùng tam giác.

Phương án thực hiện 8: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 7, trong đó chế độ hợp nhất phi chữ nhật là chế độ GEO.

Phương án thực hiện 9: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 8, trong đó dự báo có trọng số là cơ chế bù độ sáng mức lát (chẳng hạn, dự báo có trọng số toàn cục).

Phương án thực hiện 10: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 9, trong đó dự báo có trọng số là cơ chế bù độ sáng mức khối, chẳng hạn bù độ sáng cục bộ (local illumination compensation, LIC).

Phương án thực hiện 11: Phương pháp theo phương án bất kỳ trong các phương án thực hiện từ 1 đến 10, trong đó các tham số dự báo có trọng số bao gồm: tập hợp cờ chỉ báo liệu dự báo có trọng số có được áp dụng cho các thành phần độ sáng và sắc độ của khối được dự báo; Các tham số mô hình tuyến tính \alpha và \betta xác định biến đổi tuyến tính của các giá trị của khối được dự báo.

Khía cạnh thứ nhất của sáng chế, như được thể hiện trên Fig.12, đề cập đến phương pháp dự báo ngoài 1200, bao gồm: S1201: xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; S1202: thu thập một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối; và S1203: thu thập giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Theo cách thực hiện khả thi, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Theo cách thực hiện khả thi, việc xác định chế độ dự báo ngoài phi chữ nhật được phép, bao gồm: chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Theo cách thực hiện khả thi, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề ảnh của ảnh.

Theo cách thực hiện khả thi, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề lát của lát.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng hình học (GEO).

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Theo khía cạnh thứ hai của sáng chế, thiết bị 1300 để dự báo ngoài, như được thể hiện trên Fig.13, bao gồm: bộ nhớ bất biến 1301 có các lệnh mà bộ xử lý có thể thực thi được lưu trữ trên đó; và bộ xử lý 1302, được ghép nối với bộ nhớ 1301 được tạo cấu hình để thực thi các lệnh bộ xử lý thực thi được để dễ thực hiện một trong các phương án thực hiện khả thi theo khía cạnh thứ nhất của sáng chế.

Theo khía cạnh thứ ba của sáng chế, dòng bit để dự báo ngoài, bao gồm: thông tin chỉ báo để xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; và một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối, trong đó giá trị dự báo của khối hiện tại thu được dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Theo cách thực hiện khả thi, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Theo cách thực hiện khả thi, thông tin chỉ báo bao gồm số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Theo cách thực hiện khả thi, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề ảnh của ảnh.

Theo cách thực hiện khả thi, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề lát của lát.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng GEO.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Khía cạnh thứ tư của sáng chế đề cập đến, như được thể hiện trên Fig.14, thiết bị dự báo ngoài 1400, bao gồm: môđun xác định 1401, được tạo cấu hình để xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; môđun thu thập 1402, được tạo cấu hình để thu được một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối; và môđun dự báo 1403, được tạo cấu hình để thu được giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Theo cách thực hiện khả thi, thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Theo cách thực hiện khả thi, môđun xác định 1401 được tạo cấu hình cụ thể để: chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Theo cách thực hiện khả thi, nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề ảnh của ảnh.

Theo cách thực hiện khả thi, nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề lát của lát.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Theo cách thực hiện khả thi, chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng GEO.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Theo cách thực hiện khả thi, các tham số dự báo có trọng số bao gồm: các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Các phương pháp theo giải pháp kỹ thuật đã biết có thể được tóm tắt trong danh sách các khía cạnh sau:

Khía cạnh 1. Phương pháp dự báo ngoài bao gồm các bước:

xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối;

thu thập một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối; và

thu thập giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Khía cạnh 2. Phương pháp theo khía cạnh 1, trong đó thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu, và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Khía cạnh 3. Phương pháp theo khía cạnh 1 hoặc 2, trong đó chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Khía cạnh 4. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 3, trong đó xác định chế độ dự báo ngoài phi chữ nhật được phép, bao gồm:

Chỉ báo số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Khía cạnh 5. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 4, trong đó nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề ảnh của ảnh.

Khía cạnh 6. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 4, trong đó nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và chỉ báo thông tin để xác định chế độ dự báo ngoài phi chữ nhật được phép nằm trong tiêu đề lát của lát.

Khía cạnh 7. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 6, trong đó chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Khía cạnh 8. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 6, trong đó chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng GEO.

Khía cạnh 8a. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8, trong đó các phần tử cú pháp liên quan đến số lượng ứng viên cho chế độ hợp nhất (thông tin chỉ báo để xác định dự báo ngoài phi chữ nhật) được báo hiệu trong SPS

Khía cạnh 8b. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8a, trong đó tiêu đề ảnh được báo hiệu trong tiêu đề lát khi ảnh bao gồm chỉ một lát.

Khía cạnh 8c. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 to 8b, trong đó tiêu đề ảnh được báo hiệu trong tiêu đề lát khi ảnh bao gồm chỉ một lát.

Khía cạnh 8d. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8c, trong đó tập hợp tham số ảnh (picture parameter set, PPS) bao gồm cờ, giá trị của cờ định nghĩa liệu các tham số dự báo có trọng số có xuất hiệu trong tiêu đề ảnh hoặc trong tiêu đề lát.

Khía cạnh 8e. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8d, trong đó cờ trong tiêu đề ảnh chỉ báo liệu lát thuộc loại không nội vùng có xuất hiện hay không và liệu các tham số chế độ dự báo ngoài có được báo hiệu cho lát này hay không.

Khía cạnh 9. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8, trong đó các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Khía cạnh 10. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 8, trong đó các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Khía cạnh 11. Phương pháp theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 10, trong đó các tham số dự báo có trọng số bao gồm:

các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và

các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

Khía cạnh 12. Thiết bị để dự báo ngoài, bao gồm:

bộ nhớ bất biến có các lệnh mà bộ xử lý có thể thực thi được lưu trữ trên đó; và

bộ xử lý, được ghép nối với bộ nhớ, được tạo cấu hình để thực thi các lệnh bộ xử lý thực thi được để dễ thực hiện theo khía cạnh bất kỳ trong các khía cạnh từ 1 đến 11.

Khía cạnh 13. Dòng bit để dự báo ngoài, bao gồm:

thông tin chỉ báo để xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối; và

một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối, trong đó giá trị dự báo của khối hiện tại thu được dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số, trong đó một trong số tham số chế độ dự báo ngoài chỉ báo thông tin ảnh tham chiếu cho khối hiện tại, và trong đó nhóm khối bao gồm khối hiện tại.

Khía cạnh 14. Dòng bit theo khía cạnh 13, trong đó thông tin ảnh tham chiếu bao gồm liệu dự báo có trọng số có được kích hoạt cho chỉ số ảnh tham chiếu,

và trong đó chế độ dự báo ngoài phi chữ nhật bị vô hiệu hóa nếu dự báo có trọng số được kích hoạt.

Khía cạnh 15. Dòng bit theo khía cạnh 13 hoặc 14, trong đó chế độ dự báo ngoài phi chữ nhật được kích hoạt nếu dự báo có trọng số bị vô hiệu hóa.

Khía cạnh 16. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 15, trong đó thông tin chỉ báo bao gồm số lượng lớn nhất của các ứng viên hợp nhất tam giác (MaxNumTriangleMergeCand) lớn hơn 1.

Khía cạnh 17. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 16, trong đó nhóm khối bao gồm ảnh, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề ảnh của ảnh.

Khía cạnh 18. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 17, trong đó nhóm khối bao gồm lát, và trong đó các tham số dự báo có trọng số và thông tin chỉ báo nằm trong tiêu đề lát của lát.

Khía cạnh 19. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 18, trong đó chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng tam giác.

Khía cạnh 20. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 19, trong đó chế độ dự báo ngoài phi chữ nhật là chế độ phân vùng hình học (GEO).

Khía cạnh 21. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 20, trong đó các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức lát.

Khía cạnh 22. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 20, trong đó các tham số dự báo có trọng số được sử dụng cho bù độ sáng mức khối.

Khía cạnh 23. Dòng bit theo khía cạnh bất kỳ trong các khía cạnh từ 13 đến 22, trong đó các tham số dự báo có trọng số bao gồm:

các cờ chỉ báo liệu dự báo có trọng số được áp dụng cho các thành phần độ sáng và/hoặc sắc độ của khối dự báo; và

các tham số mô hình tuyến tính xác định biến đổi tuyến tính của giá trị của khối dự báo.

YÊU CẦU BẢO HỘ

1. Phương pháp mã hóa số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học, trong đó phương pháp bao gồm các bước:

thu được giá trị của bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên dự báo vectơ chuyển động (motion vector prediction, MVP) hợp nhất;

thu được giá trị của bộ chỉ báo thứ hai đại diện liệu có thể bù chuyển động dựa trên phân vùng hình học cho chuỗi viđeo; và

mã hóa giá trị của bộ chỉ báo thứ ba thành dòng bit khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất; trong đó giá trị của bộ chỉ báo thứ ba được mã hóa thành tập hợp tham số chuỗi (sequence parameter set, SPS) của dòng bit; trong đó số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học thu được dựa trên giá trị của bộ chỉ báo thứ nhất và giá trị của bộ chỉ báo thứ ba.

2. Phương pháp theo điểm 1, trong đó ngưỡng bằng 2.

3.Phương pháp theo điểm 1, trong đó phương pháp còn bao gồm bước:

gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 2 khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

4. Phương pháp theo điểm 1, trong đó phương pháp còn bao gồm bước:

gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0 khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

5. Phương pháp theo điểm 1, trong đó giá trị định trước bằng 1.

6. Phương pháp theo điểm 1, trong đó bước thu được giá trị của bộ chỉ báo thứ hai được thực hiện sau bước thu được giá trị của bộ chỉ báo thứ nhất.

7. Phương pháp theo điểm 6, trong đó giá trị của bộ chỉ báo thứ hai được phân tích từ tập hợp tham số chuỗi (sequence parameter set, SPS) của dòng bit khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7, trong đó giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit.

9. Thiết bị mã hóa viđeo, trong đó thiết bị bao gồm:

môđun nhận, được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ nhất là số lượng lớn nhất của các ứng viên MVP;

môđun nhận còn được tạo cấu hình để thu được giá trị của bộ chỉ báo thứ hai đại diện liệu có thể bù chuyển động dựa trên phân vùng hình học cho chuỗi viđeo hay khong; và

môđun mã hóa, được tạo cấu hình để mã hóa giá trị của bộ chỉ báo thứ ba thành dòng bit khi giá trị của bộ chỉ báo thứ nhất lớn hơn ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước, trong đó bộ chỉ báo thứ ba là số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học được trừ đi khỏi giá trị của bộ chỉ báo thứ nhất; trong đó giá trị của bộ chỉ báo thứ ba được mã hóa thành SPS của dòng bit; trong đó số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học thu được dựa trên giá trị của bộ chỉ báo thứ nhất và giá trị của bộ chỉ báo thứ ba.

10. Thiết bị theo điểm 9, trong đó môđun nhận còn được tạo cấu hình để:

gán giá trị của số lượng lớn nhất của các ứng viện chế độ hợp nhất phân vùng hình học bằng 2 khi giá trị của bộ chỉ báo thứ nhất bằng ngưỡng và khi giá trị của bộ chỉ báo thứ hai bằng giá trị định trước.

11. Thiết bị theo điểm 9, trong đó môđun nhận còn được tạo cấu hình để:

gán giá trị của số lượng lớn nhất của các ứng viên chế độ hợp nhất phân vùng hình học bằng 0 khi giá trị của bộ chỉ báo thứ nhất nhỏ hơn ngưỡng hoặc khi giá trị của bộ chỉ báo thứ hai không bằng giá trị định trước.

12. Thiết bị theo điểm 9, trong đó ngưỡng bằng 2.

13. Thiết bị theo điểm 9, trong đó giá trị định trước bằng 1.

14. Thiết bị theo điểm 9, trong đó thu được giá trị của bộ chỉ báo thứ hai bởi môđun nhận được thực hiện sau khi thu được giá trị của bộ chỉ báo thứ nhất bởi môđun nhận.

15. Thiết bị theo điểm 14, trong đó giá trị của bộ chỉ báo thứ hai được phân ths từ SPS của dòng bit khi giá trị của bộ chỉ báo thứ nhất lớn hơn hoặc bằng ngưỡng.

16. Thiết bị theo điểm bất kỳ trong số các điểm từ 9 đến 15, trong đó giá trị của bộ chỉ báo thứ hai thu được từ SPS của dòng bit.

17. Bộ mã hóa bao gồm:

một hoặc nhiều bộ xử lý; và

vật ghi máy tính đọc được bất biến 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 8.

18. Vật ghi máy tính đọc được bất biến 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 8.

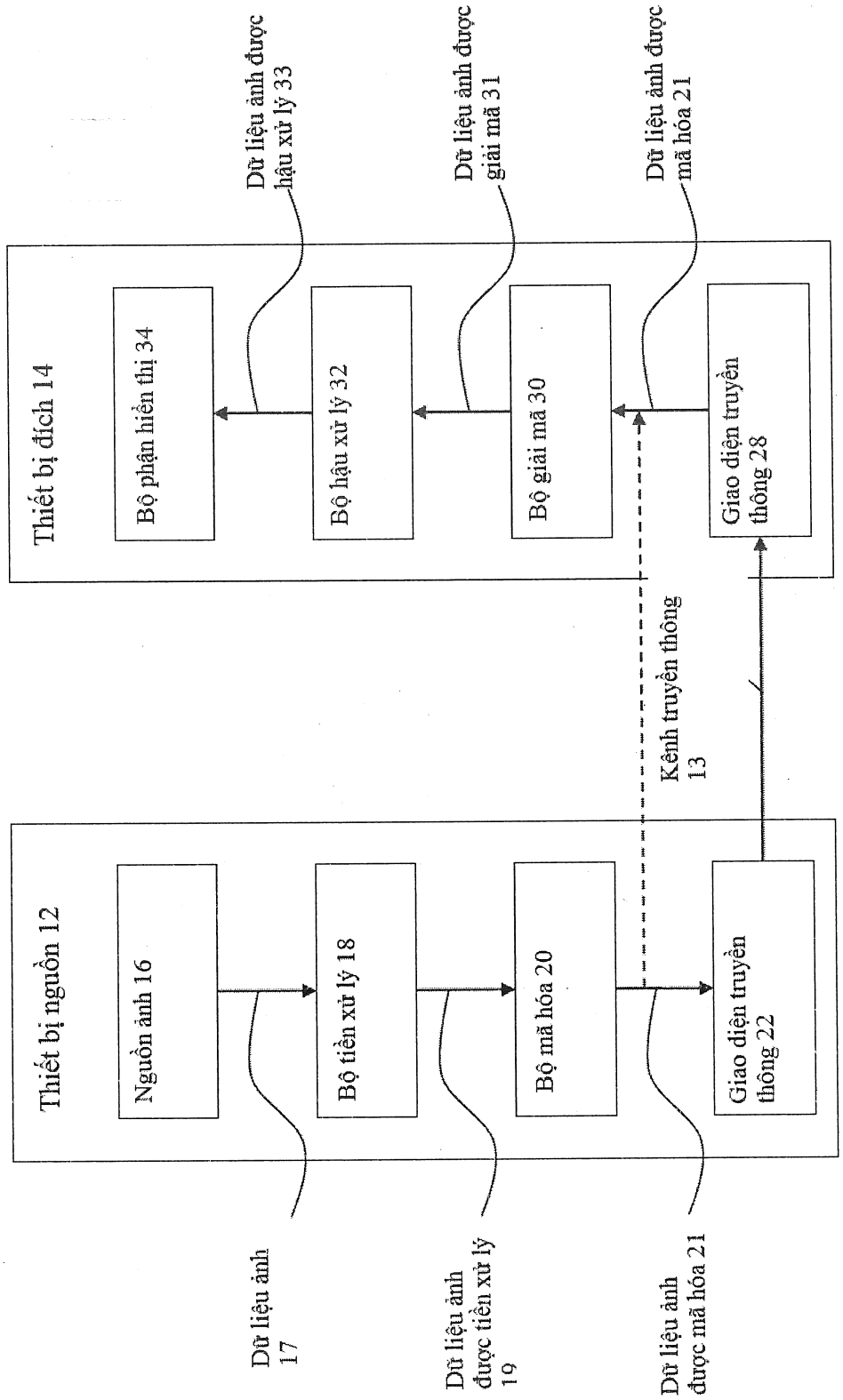
Thiết bị đích 14
Bộ phận hiển thị 34
Bộ hậu xử lý 32
Bộ giải mã 30
Giao diện truyền thông 28
Dữ liệu ảnh được hậu xử lý 33
Dữ liệu ảnh được giải mã 31
Dữ liệu ảnh được mã hóa 21
Kênh truyền thông 13
Thiết bị nguồn 12
Nguồn ảnh 16
Bộ tiền xử lý 18
Bộ mã hóa 20
Giao diện truyền thông 22
Dữ liệu ảnh 17
Dữ liệu ảnh được tiền xử lý 19
Dữ liệu ảnh được mã hóa 21

Fig.1A

10

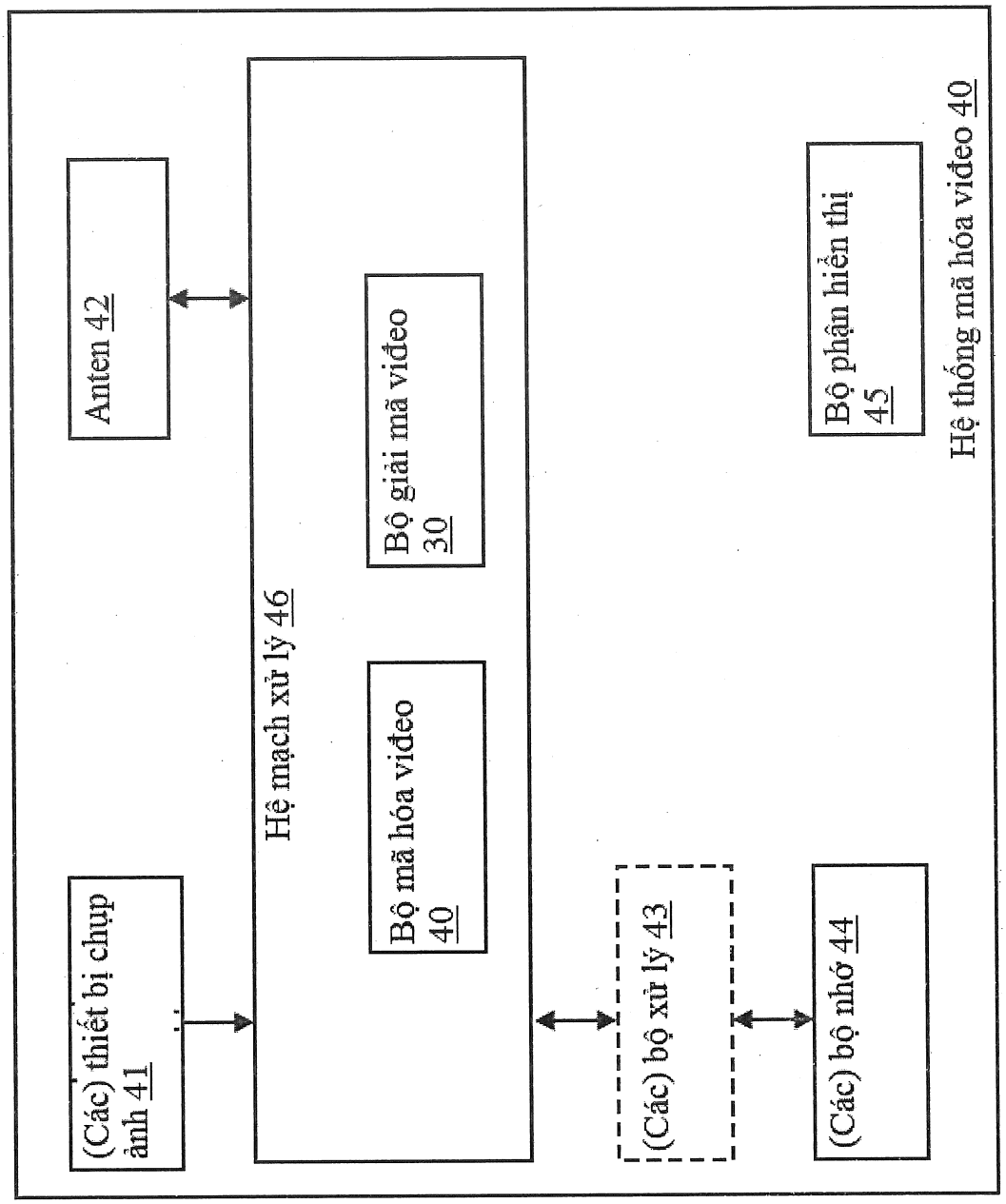
(Các) thiết bị chụp ảnh 41
Anten 42
Hệ mạch xử lý 46
Bộ mã hóa video 40
Bộ giải mã video 30
(Các) bộ xử lý 43
(Các) bộ nhớ 44
Bộ phận hiển thị 45
Hệ thống mã hóa video 40

Fig.1B

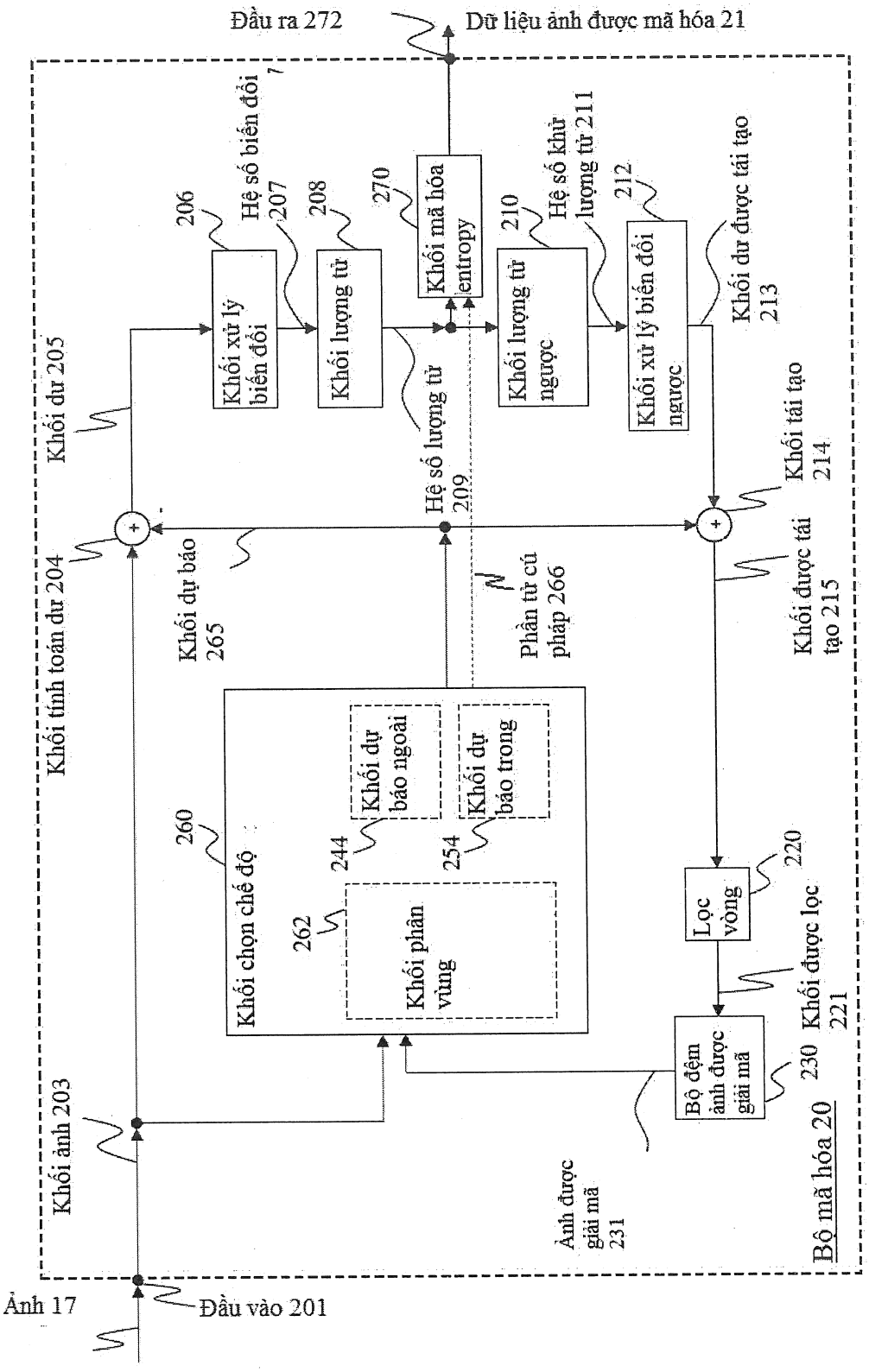
Đầu ra 272
Dữ liệu ảnh được mã hóa 21
Hệ số biến đổi 207
206
Khối xử lý biến đổi
208
Khối lượng tử
270
Khối mã hóa entropy
210
Khối lượng tử ngược
Hệ số khử lượng tử 211
212
Khối xử lý biến đổi ngược
Khối dư được tái tạo 213
Khối dư 205
Hệ số lượng tử 209
Khối tái tạo 214
Khối tính toán dư 204
Khối dự báo 265
Phần tử cú pháp 266
Khối được tái tạo 215
260
Khối chọn chế độ
244
Khối dự báo ngoài
254
Khối dự báo trong
262
Khối phân vùng
220
Lọc vòng
Khối được lọc 221
230
Bộ đệm ảnh được giải mã
Khối ảnh 203
Ảnh được giải mã 231
Bộ mã hóa 20
Ảnh 17
Đầu vào 201

Fig.2

4/17

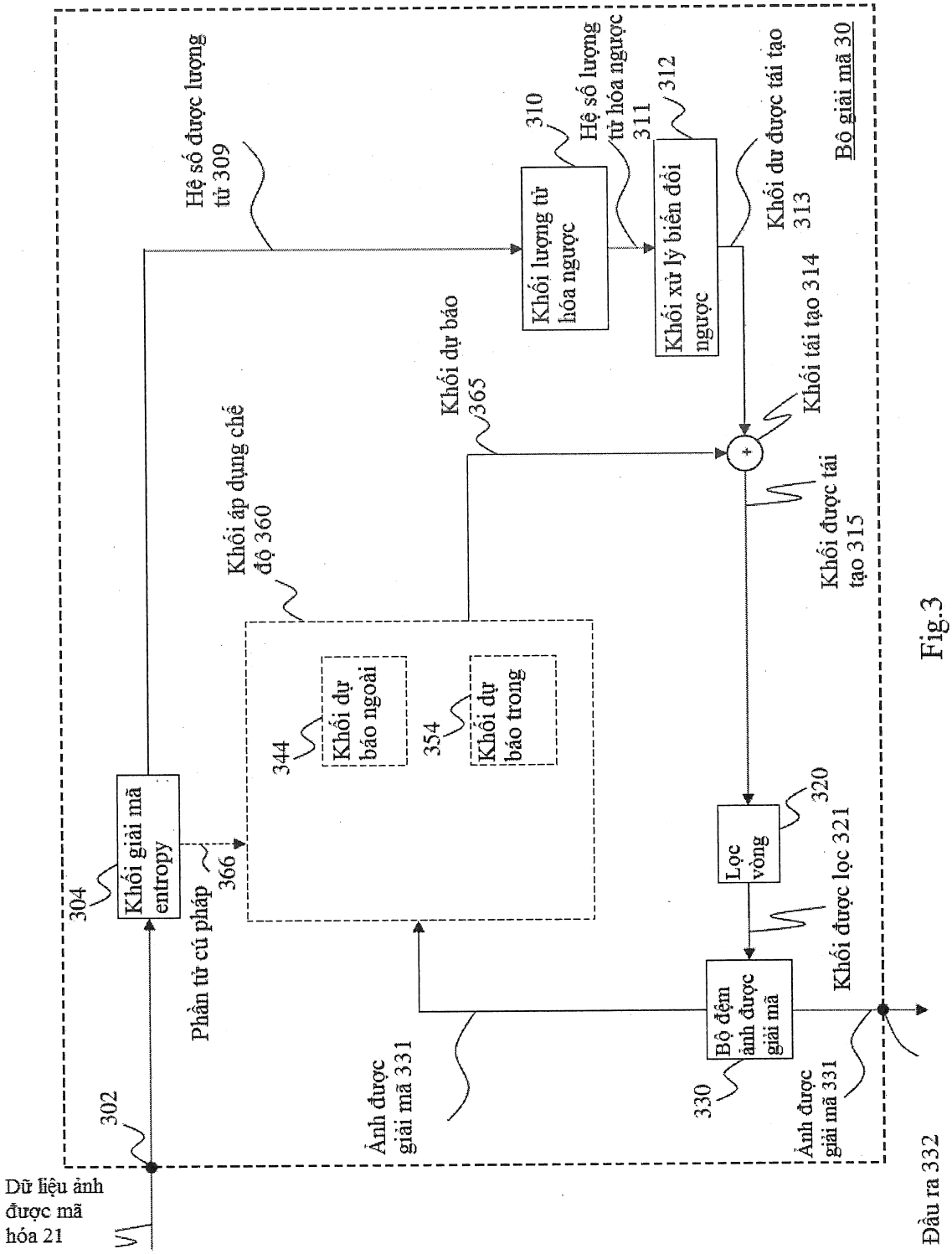
Dữ liệu ảnh được mã hóa 21
302
Khối giải mã entropy
304
Phần tử cú pháp
366
Hệ số được lượng tử 309
Khối áp dụng chế độ 360
Khối dự báo ngoài
344
Khối dự báo trong
354
Khối dự báo
365
Khối lượng tử hóa ngược
310
Hệ số lượng tử hóa ngược
311
Khối xử lý biến đổi ngược
312
Khối dư được tái tạo
313
Khối tái tạo 314
Khối được tái tạo 315
Lọc vòng
320
Khối được lọc 321
Bộ đệm ảnh được giải mã
330
Ảnh được giải mã 331
Ảnh được giải mã 331
Bộ giải mã 30
Đầu ra 332

Fig.3

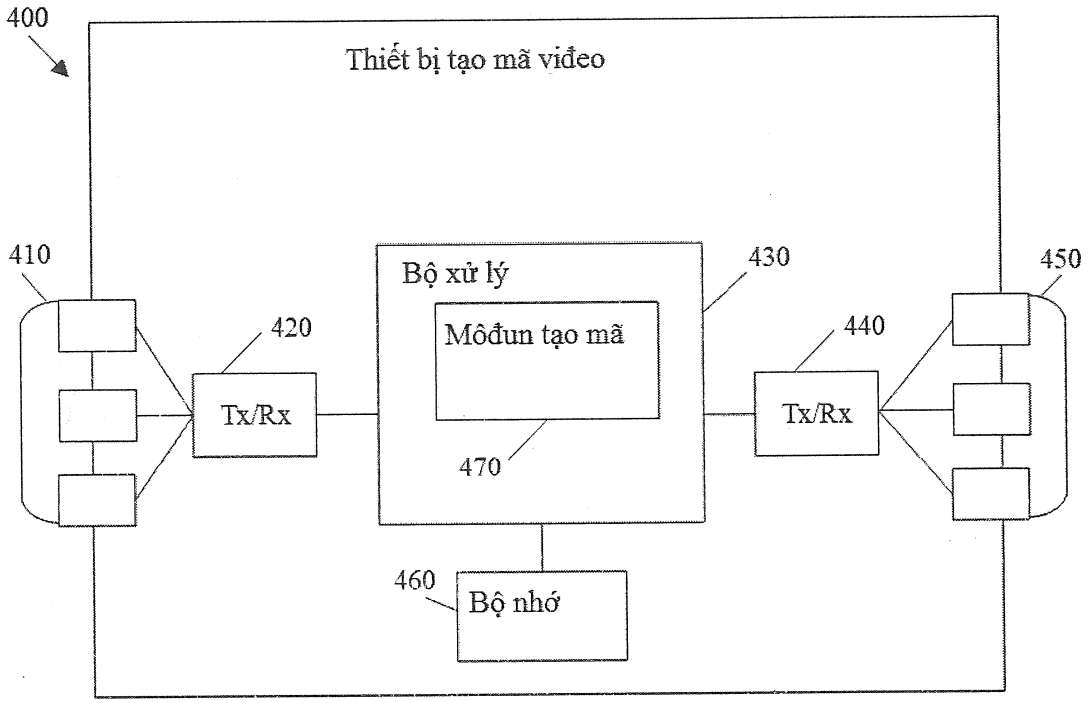
400
Thiết bị tạo mã viđeo
410
420
Tx/Rx
Bộ xử lý
Môđun tạo mã
430
470
440
Tx/Rx
450
460
Bộ nhớ

Fig.4

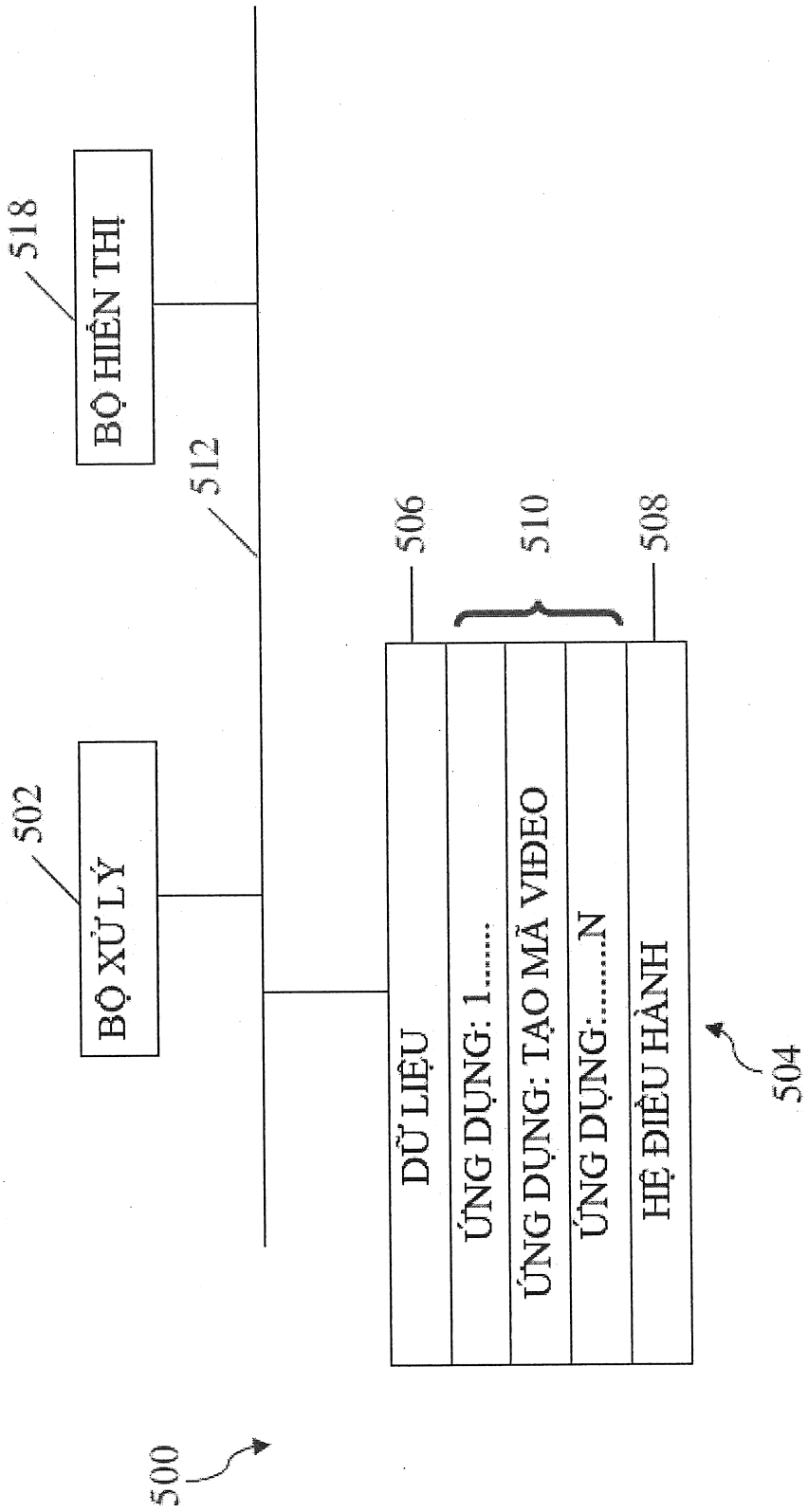
518
BỘ HIỂN THỊ
512
502
BỘ XỬ LÝ
506
510
508
DỮ LIỆU
ỨNG DỤNG: 1.....
ỨNG DỤNG: TẠO MÃ VIĐEO
ỨNG DỤNG:.....N
HỆ ĐIỀU HÀNH
504
500

Fig.5

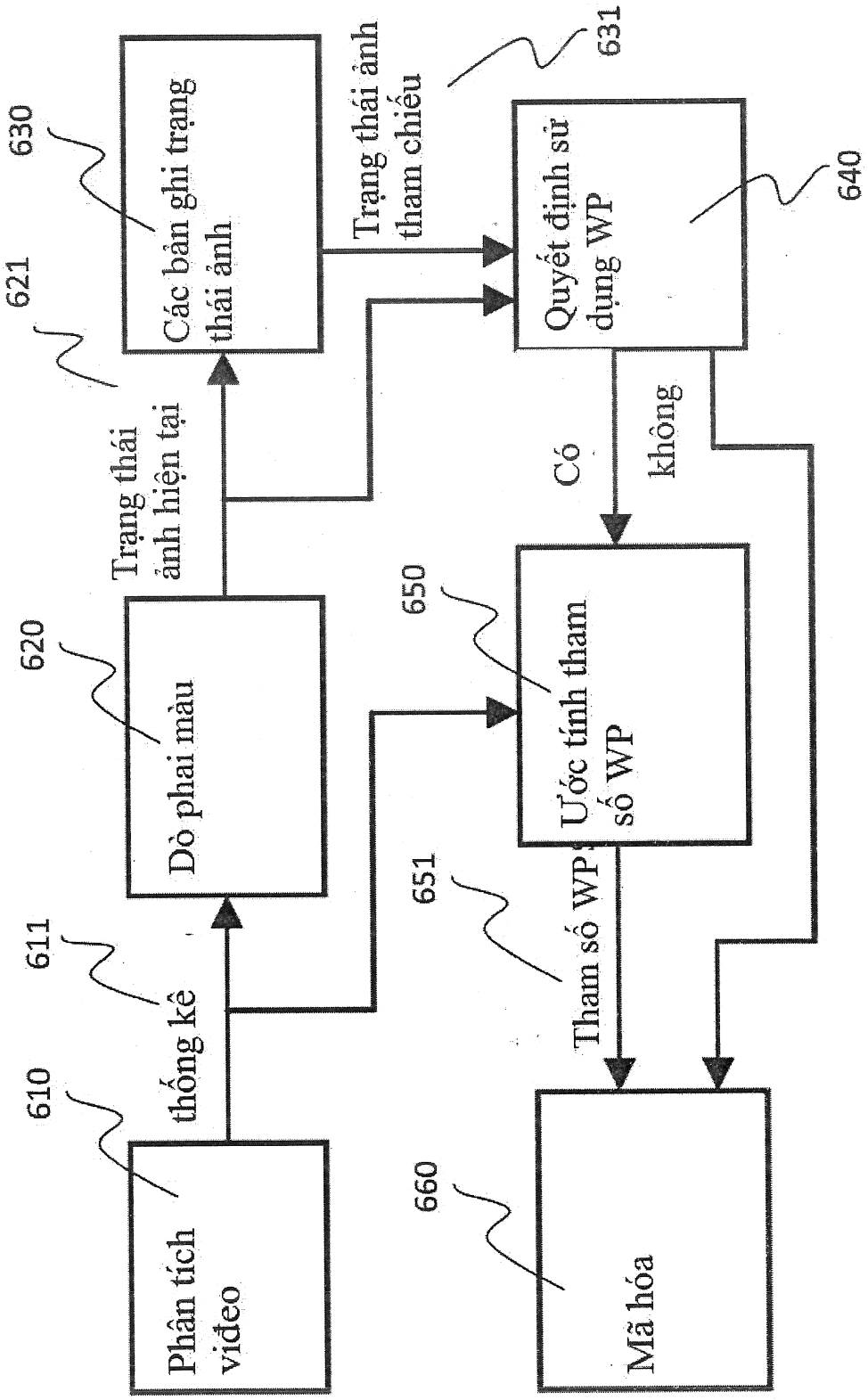
610
Phân tích video
611
thống kê
620
Dò phai màu
621
Trạng thái ảnh hiện tại
630
Các bản ghi trạng thái ảnh
631
Trạng thái ảnh tham chiếu
640
Quyết định sử dụng WP
Có
không
650
Ước tính tham số WP
651
Tham số WP
660
Mã hóa

Fig.6

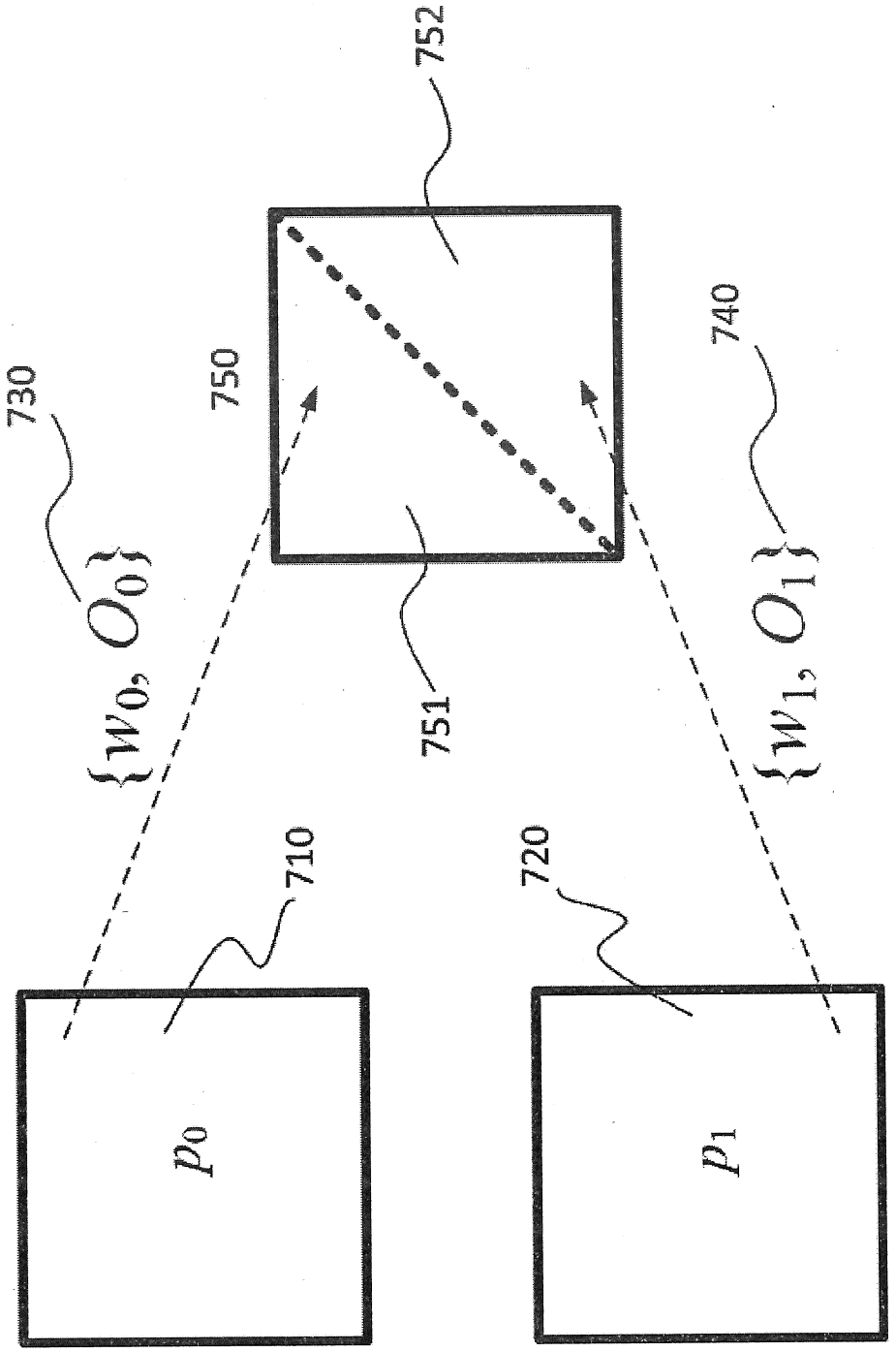

Fig.7

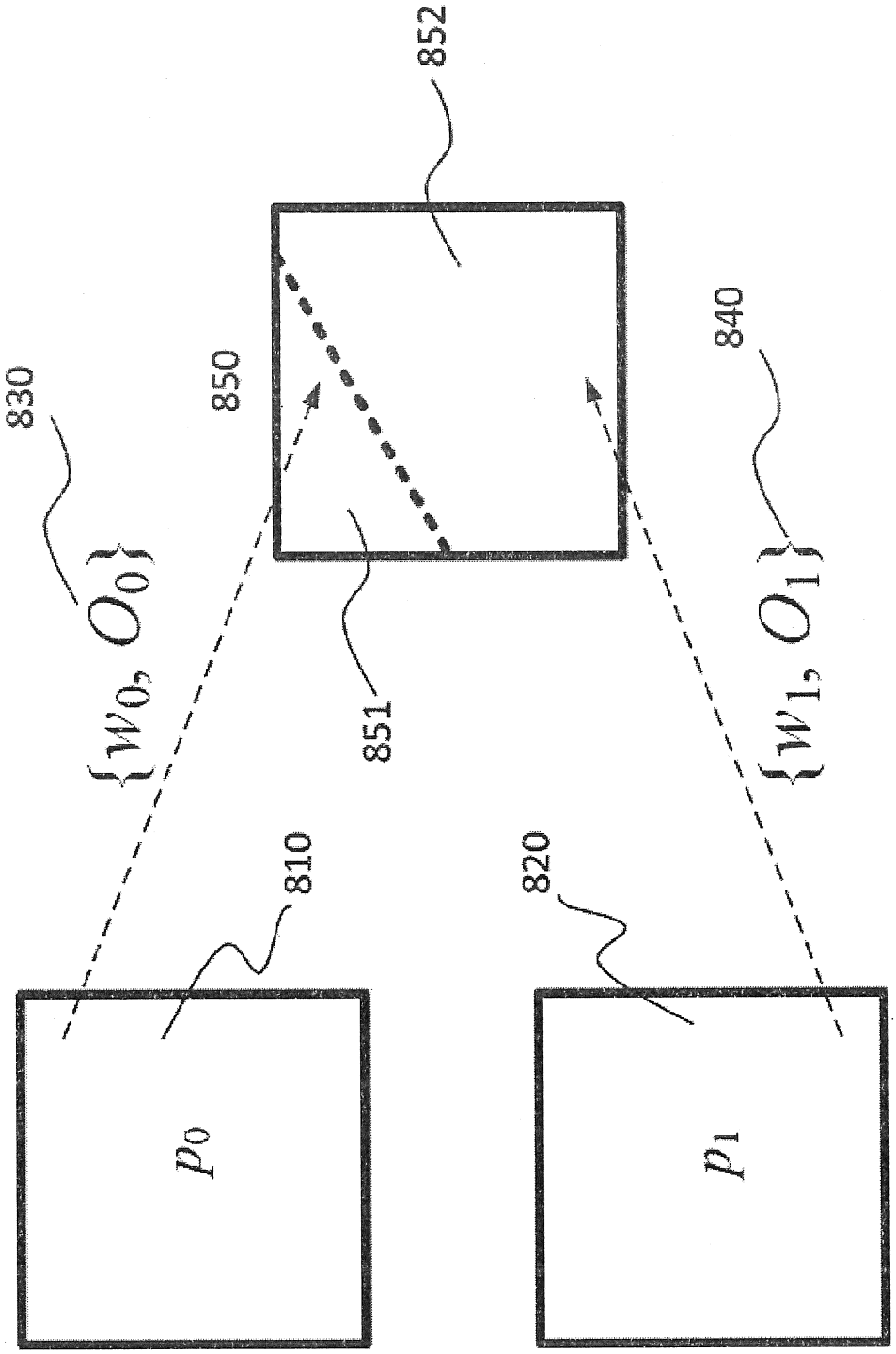

Fig.8

10/17

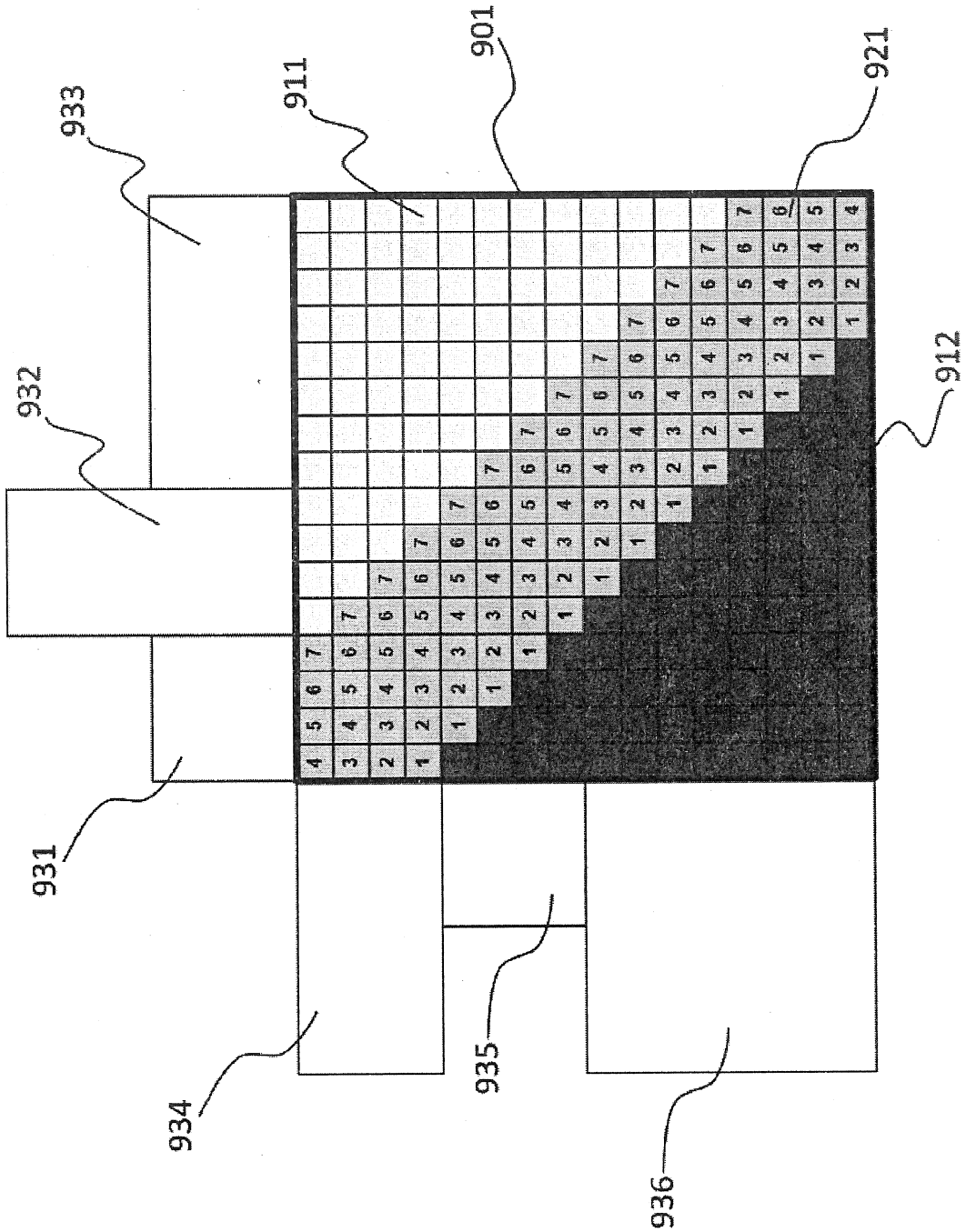

Fig.9

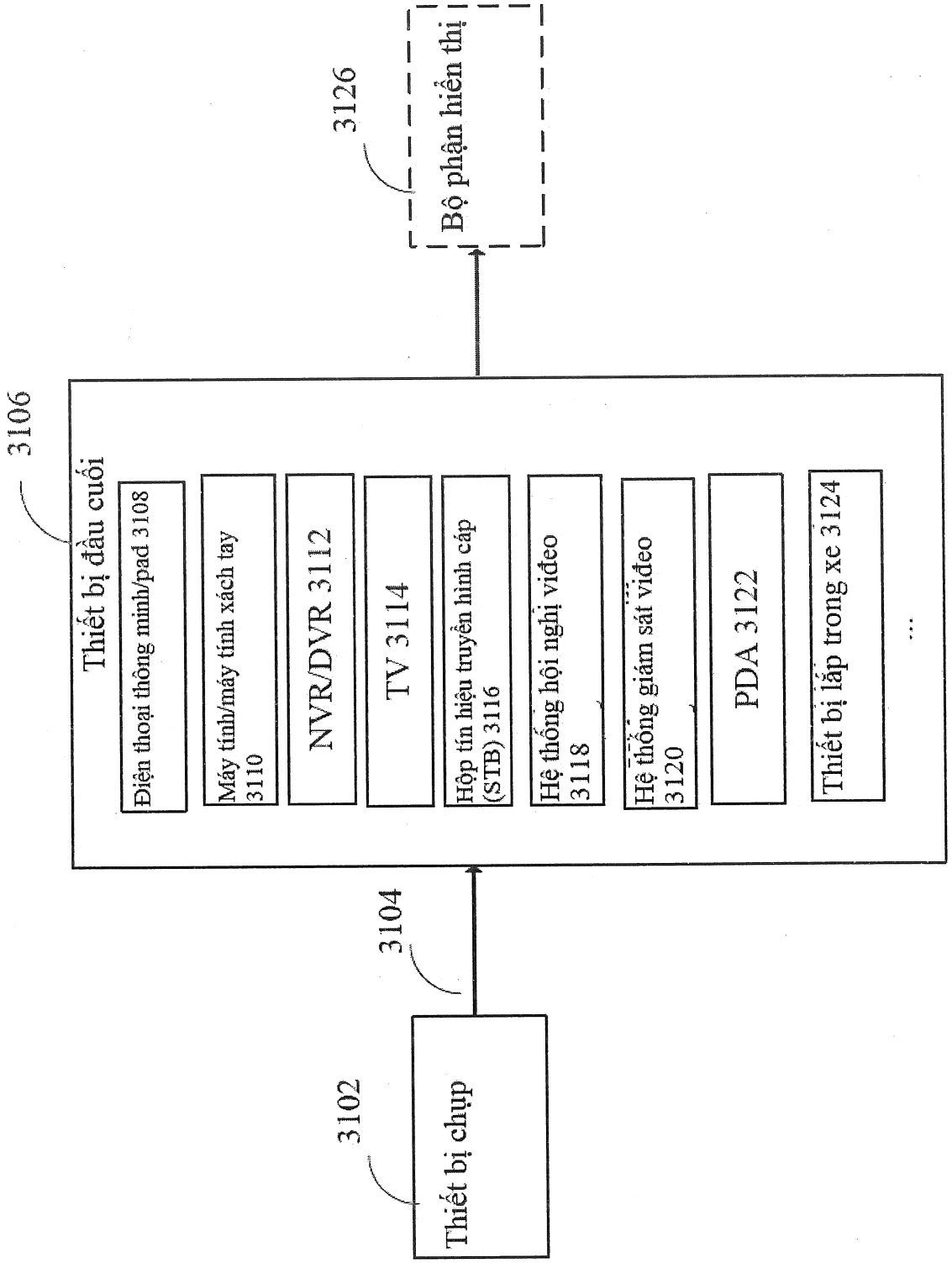
3102
Thiết bị chụp
3104
3106
Thiết bị đầu cuối
Điện thoại thông minh/pad 3108
Máy tính/máy tính xách tay 3110
NVR/DVR 3112
TV 3114
Hộp tín hiệu truyền hình cáp (STB) 3116
Hệ thống hội nghị viđeo 3118
Hệ thống giám sát viđeo 3120
PDA 3122
Thiết bị lắp trong xe 3124
...
3126
Bộ phận hiển thị

Fig.10

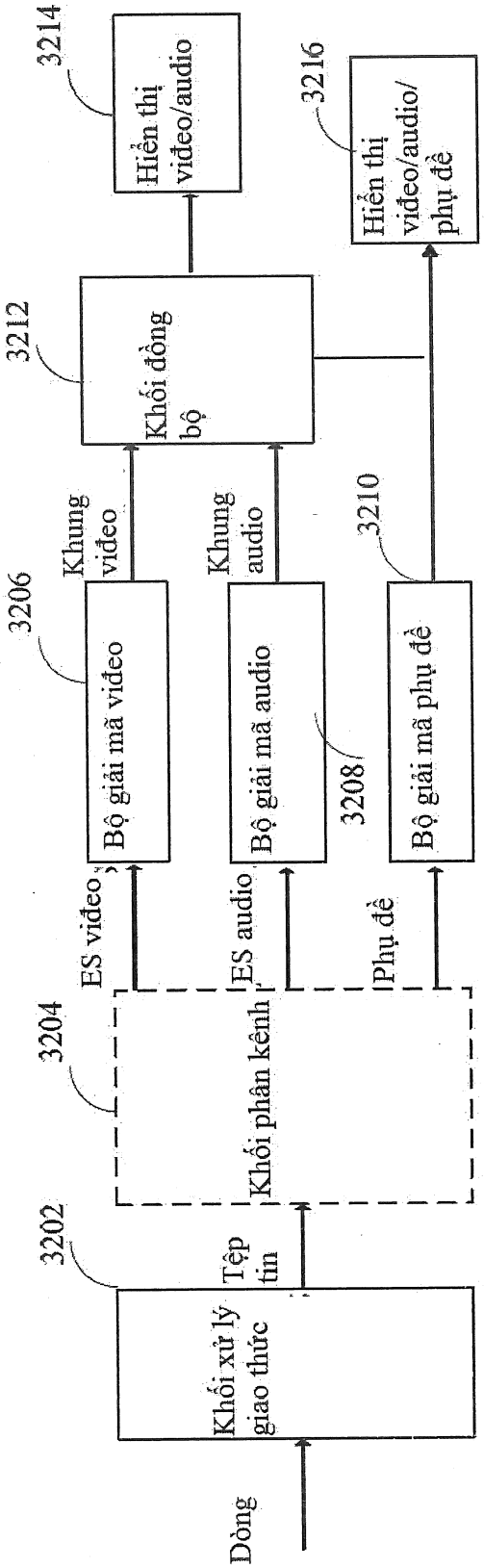
Dòng
3202
Khối xử lý giao thức
Tệp tin
3204
Khối phân kênh
ES video
ES audio
Phụ đề
3206
Bộ giải mã video
Bộ giải mã audio
3208
Bộ giải mã phụ đề
3210
Khung video
Khung audio
3212
Khối đồng bộ
3214
Hiển thị video/audio
3216
Hiển thị video/audio/ phụ đề

Fig.11

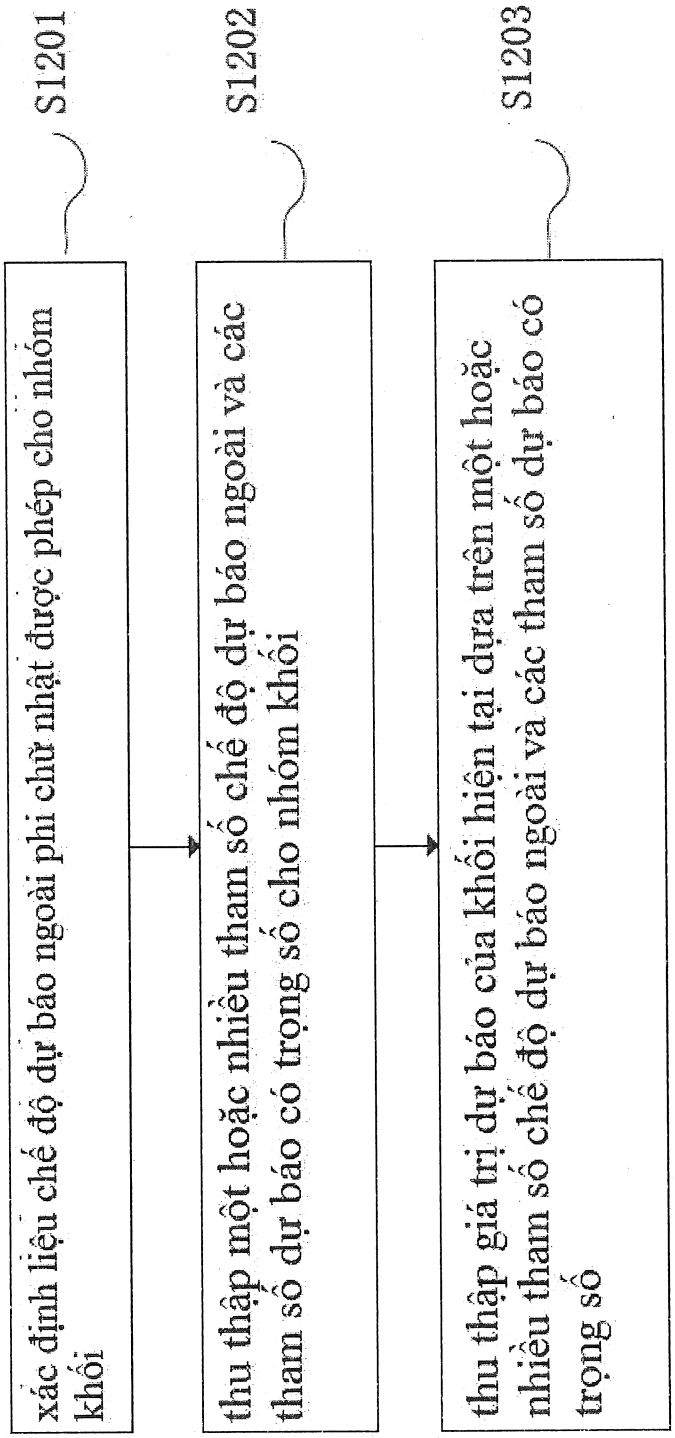
S1201
xác định liệu chế độ dự báo ngoài phi chữ nhật được phép cho nhóm khối
S1202
thu thập một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số cho nhóm khối
S1203
thu thập giá trị dự báo của khối hiện tại dựa trên một hoặc nhiều tham số chế độ dự báo ngoài và các tham số dự báo có trọng số

Fig.12

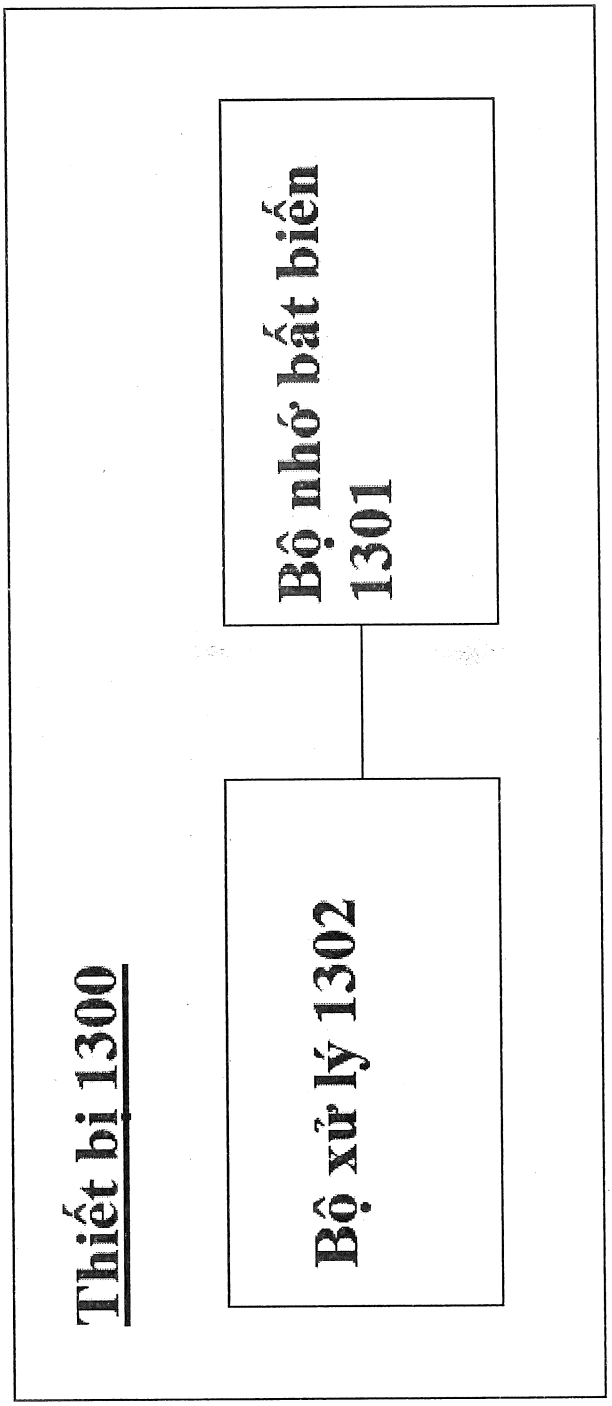

Fig.13

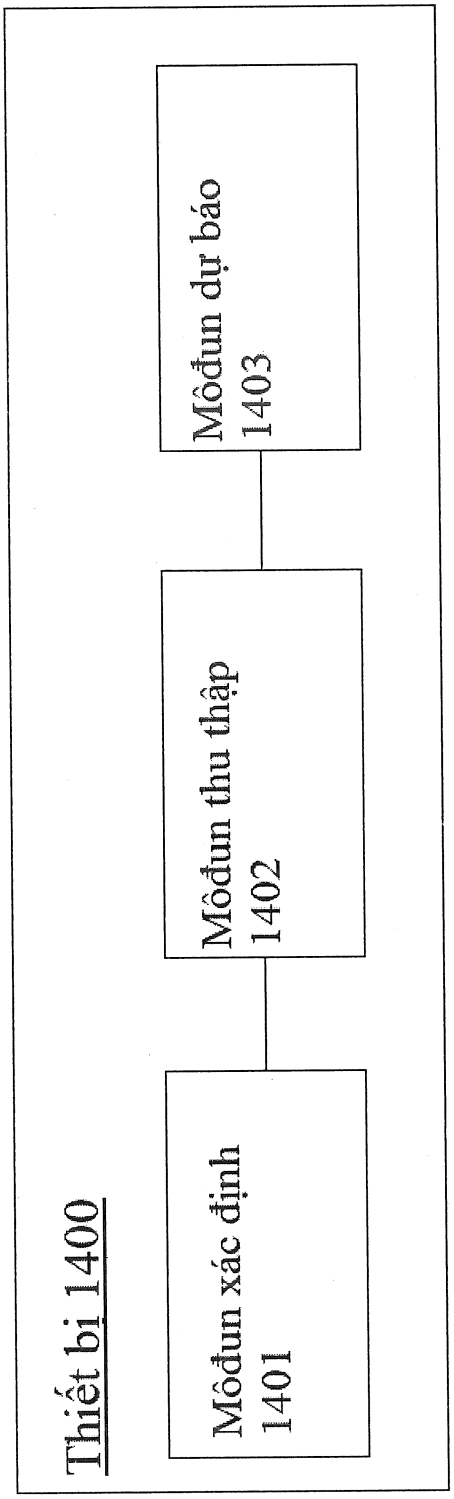
Thiết bị 1400
Môđun xác định
1401
Môđun thu thập
1402
Môđun dự báo
1403

Fig.14

16/17

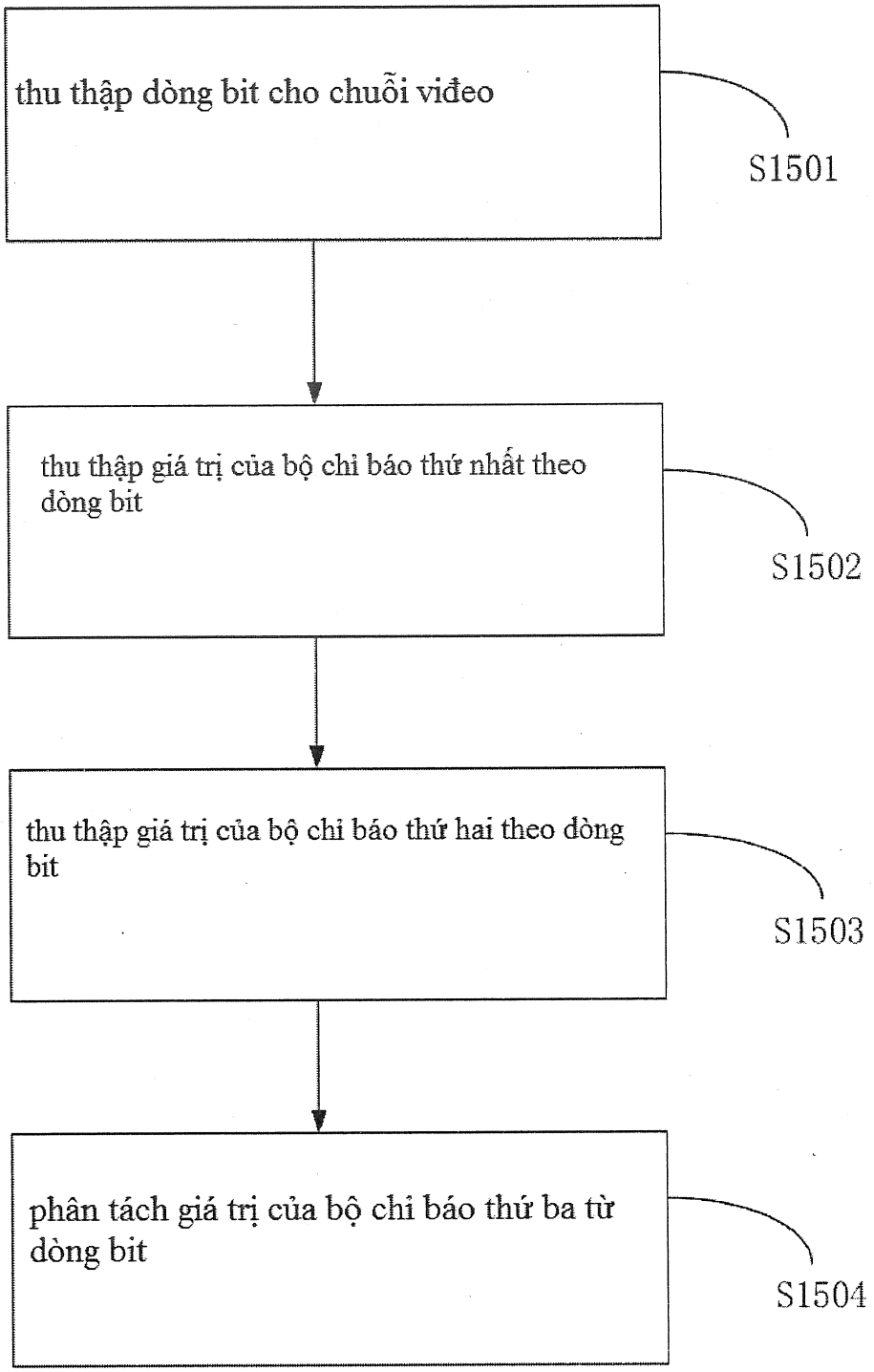
thu thập dòng bit cho chuỗi viđeo
S1501
thu thập giá trị của bộ chỉ báo thứ nhất theo dòng bit
S1502
thu thập giá trị của bộ chỉ báo thứ hai theo dòng bit
S1503
phân tách giá trị của bộ chỉ báo thứ ba từ dòng bit
S1504

Fig.15

17/17

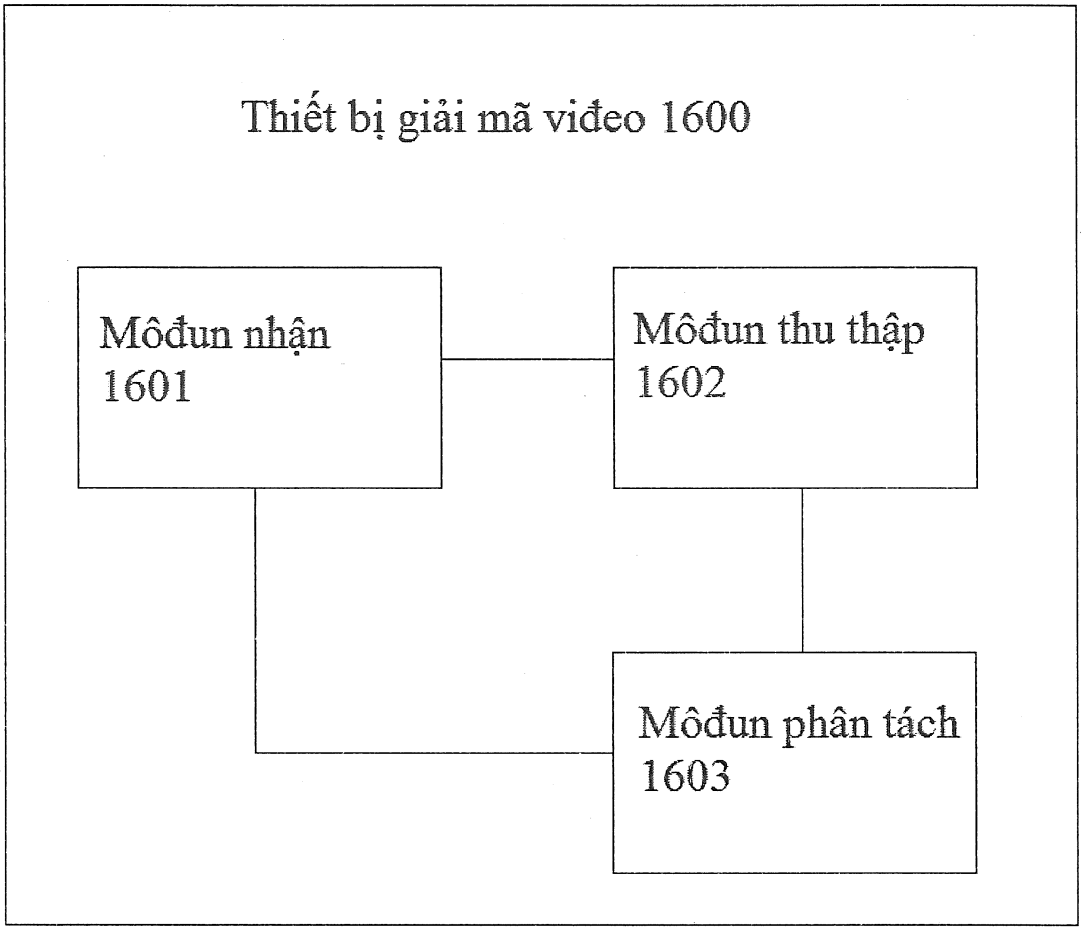
Thiết bị giải mã viđeo 1600
Môđun nhận 1601
Môđun thu thập 1602
Môđun phân tách 1603

Fig.16