(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) **1-0053404**

(51)[2020.01] **H04N 19/119**; H04N 19/70; H04N 19/176; H04N 19/46; H04N 19/169; H04N 19/174

(13) **B**

(21) 1-2022-02118
(22) 09/10/2020
(86) PCT/CN2020/119932 09/10/2020
(87) WO 2021/063421 08/04/2021
(30) PCT/CN2019/109809 02/10/2019 CN
(45) 25/11/2025 452
(43) 25/10/2022 415A
(73) 1. BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (CN)
Room B-0035, 2/F, No. 3 Building No. 30, Shixing Road, Shijingshan District Beijing 100041, China
2. BYTEDANCE INC. (US)
12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California 90066, USA
(72) ZHANG, Kai (CN); DENG, Zhipin (CN); LIU, Hongbin (CN); ZHANG, Li (CN); XU, Jizheng (CN).
(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DỮ LIỆU VIĐEO, VÀ VẬT GHI MÁY TÍNH

(21) 1-2022-02118

(57) Sáng chế đề cập đến phương pháp lấy làm ví dụ xử lý viđeo bao gồm thực hiện biến đổi giữa ảnh của viđeo bao gồm một hoặc nhiều ảnh phụ và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng thông tin về ảnh phụ được bao gồm trong biểu diễn dòng bit dựa trên ít nhất một trong: (1) một hoặc nhiều vị trí góc của ảnh phụ, hoặc (2) kích thước của ảnh phụ. Quy tắc định dạng còn xác định rằng thông tin về ảnh phụ được đặt sau thông tin về khối cây mã trong biểu diễn dòng bit.

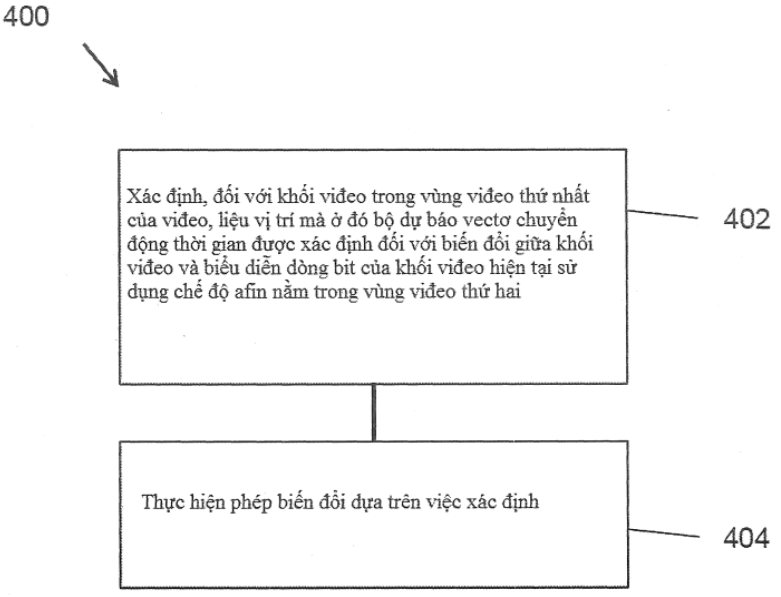

Fig.4

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các công nghệ mã hóa và giải mã viđeo và ảnh.

Tình trạng kỹ thuật của sáng chế

Viđeo số sử dụng lượng băng thông lớn nhất mạng Internet và các mạng truyền thông số khác. Do số lượng thiết bị người dùng được kết nối có thể nhận và hiển thị viđeo tăng, mong đợi rằng nhu cầu sử dụng băng thông cho viđeo số sẽ tiếp tục tăng.

Bản chất kỹ thuật của sáng chế

Các kỹ thuật được nêu có thể được sử dụng bởi bộ giải mã hoặc bộ mã hóa viđeo hoặc ảnh theo các phương án thực hiện mà trong đó mã hóa hoặc giải mã dựa trên ảnh phụ được thực hiện.

Trong một khía cạnh ví dụ, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit được yêu cầu để tuân theo quy tắc định dạng mà xác định rằng mỗi ảnh được mã hóa dưới dạng một hoặc nhiều lát, và trong đó quy tắc định dạng chặn các mẫu ở ảnh không được bao phủ bởi lát bất kỳ trong một hoặc nhiều lát.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm bước xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, cách thức báo hiệu thông tin của một hoặc nhiều lát trong ảnh theo quy tắc được liên kết với số lượng ô hoặc số lượng khối viên trong ảnh. Phương pháp cũng bao gồm bước thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo theo quy tắc. ảnh được mã hóa trong biểu diễn dòng bit dưới dạng một hoặc nhiều lát và quy tắc này xác định liệu hoặc cách thức địa chỉ của lát của ảnh được bao gồm trong biểu diễn dòng bit.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm bước xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, liệu phần tử cú pháp chỉ báo hoạt động lọc truy nhập các mẫu giữa nhiều khối viên trong ảnh được kích hoạt được bao gồm trong biểu diễn dòng bit dựa trên số lượng ô hoặc số lượng khối viên trong ảnh. Phương pháp cũng bao gồm bước thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, trong đó ảnh bao gồm một hoặc nhiều ảnh phụ, và trong đó số lượng của một hoặc nhiều ảnh phụ được chỉ báo bởi phần tử cú pháp in biểu diễn dòng bit.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm thực hiện biến đổi giữa ảnh của viđeo bao gồm một hoặc nhiều ảnh phụ và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng thông tin về ảnh phụ được bao gồm trong biểu diễn dòng bit dựa trên ít nhất một trong: (1) một hoặc nhiều vị trí góc của ảnh phụ, hoặc (2) kích thước của ảnh phụ.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm bước xác định rằng công cụ lấy mẫu lại ảnh tham chiếu được kích hoạt đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo do ảnh được phân chia thành một hoặc nhiều ảnh phụ. Phương pháp cũng bao gồm bước thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm bước thực hiện biến đổi giữa video bao gồm ảnh viđeo bao gồm một hoặc nhiều ảnh phụ bao gồm một hoặc nhiều lát và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng,

đối với ảnh phụ và lát, trong trường hợp mà chỉ số nhận diện ảnh phụ được bao gồm trong tiêu đề của lát, trường địa chỉ đối với lát chỉ báo địa chỉ của lát trong ảnh phụ.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo. Phương pháp bao gồm bước xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó bộ dự báo vectơ chuyển động thời gian được xác định đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại sử dụng chế độ afin nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo khác. Phương pháp bao gồm bước xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó mẫu số nguyên trong ảnh tham chiếu được nạp đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai, trong đó ảnh tham chiếu không được sử dụng trong quá trình nội suy trong khi biến đổi; và thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo khác. Phương pháp bao gồm bước xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó giá trị mẫu độ sáng được tái tạo được nạp đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo khác. Phương pháp bao gồm bước xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó phép kiểm tra liên quan đến tách, dẫn xuất chiều sâu hoặc báo hiệu cờ tách cho khối viđeo được thực hiện trong quá trình biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo khác. Phương pháp bao gồm thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh viđeo bao gồm một hoặc nhiều khối viđeo, và biểu diễn được mã hóa của viđeo, trong đó biểu diễn được mã hóa tuân theo yêu cầu cú pháp mã hóa

rằng phép biến đổi không sẽ dụng mã hóa/giải mã ảnh phụ và công cụ mã hóa/giải mã biến đổi độ phân giải động hoặc công cụ lấy mẫu lại ảnh tham chiếu trong đơn vị viđeo.

Trong khía cạnh ví dụ khác, sáng chế đề cập đến phương pháp xử lý viđeo khác. Phương pháp bao gồm thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh viđeo bao gồm một hoặc nhiều khối viđeo, và biểu diễn được mã hóa của viđeo, trong đó biểu diễn được mã hóa tuân theo yêu cầu cú pháp mã hóa mà phần tử cú pháp thứ nhất subpic_grid_idx[i][j] không lớn hơn phần tử cú pháp thứ hai max_subpics_minus1.

Theo khía cạnh ví dụ khác nữa, phương pháp nêu trên có thể được triển khai bằng thiết bị mã hóa viđeo bao gồm bộ xử lý.

Theo khía cạnh ví dụ khác nữa, phương pháp nêu trên có thể được triển khai bằng thiết bị giải mã viđeo bao gồm bộ xử lý.

Theo khía cạnh ví dụ khác nữa, các phương pháp này có thể được triển khai ở dạng lệnh bộ xử lý thực thi được và được lưu trữ trên vật ghi máy tính đọc được.

Các khía cạnh này, và các khía cạnh khác còn được nêu trong bản mô tả.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ thể hiện ví dụ về ràng buộc vùng trong dự báo MV thời gian (TMVP) và TMVP khối phụ;

Fig.2 là hình vẽ thể hiện ví dụ về phương tiện ước tính chuyển động phân cấp;

Fig.3 là sơ đồ khối của ví dụ của nền tảng phần cứng được sử dụng để thực hiện các kỹ thuật được nêu trong bản mô tả;

Fig.4 là lưu đồ của phương pháp lấy làm ví dụ xử lý viđeo;

Fig.5 là hình vẽ thể hiện ví dụ về ảnh với các CTU độ sáng 18 x 12 được phân vùng thành12 ô và 3 lát quét mành (mang tính thông tin);

Fig.6 là hình vẽ thể hiện ví dụ về ảnh có các CTU độ sáng 18 x 12 được phân vùng thành 24 ô và 9 lát chữ nhật (mang tính thông tin);

Fig.7 là hình vẽ thể hiện ví dụ về ảnh được phân vùng thành 4 ô, 11 khối viên, và 4 lát chữ nhật (mang tính thông tin);

Fig.8 là sơ đồ khối thể hiện hệ thống xử lý viđeo ví dụ trong đó các kỹ thuật khác nhau được bộc lộ ở đây có thể được triển khai;

Fig.9 là sơ đồ khối minh họa hệ thống tạo mã viđeo ví dụ;

Fig.10 là sơ đồ khối minh họa bộ mã hóa theo một số phương án thực hiện sáng chế;

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

Fig.12 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế;

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

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

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

Fig.16 là lưu đồ biểu diễn phương pháp khác xử lý viđeo theo sáng chế;

Fig.17 là lưu đồ biểu diễn phương pháp khác xử lý viđeo theo sáng chế;

Fig.18 là lưu đồ biểu diễn phương pháp khác xử lý viđeo theo sáng chế; và

Fig.19 là lưu đồ biểu diễn phương pháp khác nữa xử lý viđeo theo sáng chế.

Mô tả chi tiết sáng chế

Sáng chế đề cập đến các kỹ thuật khác nhau có thể được sử dụng bằng bộ giải mã ảnh hoặc dòng bit viđeo để cải thiện chất lượng của viđeo hoặc ảnh số được nén hoặc được giải mã. Để ngắn gọn, cụm từ "viđeo" được sử dụng ở đây để bao gồm cả chuỗi ảnh (thường được gọi là viđeo) và các ảnh riêng rẽ. Ngoài ra, bộ mã hóa viđeo có thể cũng thực hiện các kỹ thuật này trong quá trình mã hóa để tái tạo các khung được giải mã được sử dụng để mã hóa tiếp.

Các đề mục từng phần được sử dụng trong sáng chế để dễ hiểu và không giới hạn các phương án thực hiện và các kỹ thuật ở các phần tương ứng này. Thực tế, các phương án thực hiện từ một phần có thể được kết hợp với các phương án thực hiện từ các phần khác.

1. Tóm tắt

Sáng chế đề cập đến đến các công nghệ tạo mã viđeo. Cụ thể là, đề cập đến mã hóa bảng màu có sử dụng biểu diễn dựa trên các màu cơ sở trong tạo mã viđeo. Có thể được áp dụng cho chuẩn tạo mã viđeo hiện tại giống như HEVC, hoặc chuẩn (tạo mã viđeo đa dụng) đang được hoàn thiện. Cũng có thể áp dụng cho các chuẩn mã hóa viđeo hoặc codec viđeo trong tương lai.

2. Thảo luận ban đầu

Các chuẩn mã hóa viđeo đã phát triển chủ yếu qua sự phát triển của các chuẩn ITU-T và ISO/IEC đã biết. ITU-T đã tạo ra các chuẩn H.261 và H.263, ISO/IEC đã tạo ra các chuẩn MPEG-1 và MPEG-4 hình ảnh, và hai tổ chức này đồng thời đã tạo ra các chuẩn H.262/MPEG-2 Viđeo và H.264/MPEG-4 mã hóa viđeo cao cấp (AVC) và H.265/HEVC [1,2]. Kể từ chuẩn H.262, các chuẩn mã hóa viđeo dựa trên cấu trúc mã hóa viđeo lai trong đó sử dụng dự báo thời gian cộng mã hóa biến đổi. Để khai thác các công nghệ mã hóa viđeo tương lai ngoài HEVC, nhóm khai thác viđeo chung (JVET) được thành lập bởi VCEG và MPEG đồng thời vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp thuận bởi JVET và được đưa vào phần mềm tham chiếu có tên là mô hình khai thác chung (JEM). Vào tháng 4 2018, nhóm chuyên gia viđeo chung (JVET) giữa VCEG (Q6/16) và ISO/IEC JTC1 SC29/WG11 (MPEG) được thành lập để làm việc trên chuẩn VVC nhằm giảm 50% tốc độ bit so với HEVC.

2.1 Ràng buộc vùng trong TMVP và TMVP khối phụ trong VVC

Fig.1 minh họa ràng buộc vùng lấy làm ví dụ trong TMVP và TMVP khối phụ. Trong TMVP và TMVP khối phụ, yêu cầu rằng MV thời gian có thể chỉ được lấy từ CTU cùng vị trí cộng với cột khối 4×4 như được thể hiện trên Fig.1.

2.2 Ảnh phụ ví dụ

Theo một số phương án thực hiện, các kỹ thuật mã hóa dựa trên ảnh phụ dựa trên cách thức tạo ô linh hoạt có thể được triển khai. Tóm tắt về các kỹ thuật mã hóa dựa trên ảnh phụ bao gồm các bước sau:

(1) Ảnh có thể được phân chia thành các ảnh phụ.

(2) Chỉ báo sự tồn tại của các ảnh phụ được chỉ báo trong SPS, cũng với thông tin mức chuỗi khác của các ảnh phụ.

(3) Liệu ảnh phụ có được xử lý như là ảnh trong quá trình giải mã (loại trừ các hoạt động lọc trong vòng) có thể được điều khiển bằng dòng bit.

(4) Liệu lọc trong vòng giữa các đường biên ảnh phụ có bị vô hiệu hóa có thể được điều khiển bằng dòng bit đối với mỗi ảnh phụ. Các quá trình DBF, SAO, và ALF được cập nhật để điều khiển các hoạt động lọc trong vòng giữa các đường biên ảnh phụ.

(5) Để cho đơn giản, như là điểm bắt đầu, chiều rộng, chiều cao ảnh phụ, độ lệch ngang, và độ lệch dọc được báo hiệu theo các đơn vị của mẫu độ sáng trong SPS. Các đường biên ảnh phụ bị ràng buộc phải là các đường biên lát.

(6) Xử lý ảnh phụ như là ảnh trong quá trình giải mã (không bao gồm các hoạt động lọc trong vòng) được xác định bằng cách cập nhật nhẹ cú pháp coding_tree_unit(), và cập nhật đối với quá trình giải mã dưới đây:

- Quá trình dẫn xuất để dự báo MV độ sáng thời gian (cải tiến)
- Quá trình nội suy song tuyến tính của mẫu độ sáng
- Quá trình lọc nội suy 8 bậc mẫu độ sáng
- Quá trình nội suy mẫu sắc độ

(7) Các ID ảnh phụ được báo hiệu tường minh trong SPS và được bao gồm trong các tiêu đề nhóm ô để có thể trích xuất các chuỗi ảnh phụ không cần thay đổi các đơn vị VCL NAL.

(8) Các tập hợp ảnh phụ đầu ra (OSPS) được đề xuất để xác định các điểm tương hợp và trích xuất quy chuẩn cho các ảnh phụ và các tập hợp của ảnh phụ.

2.3 Các ảnh phụ lấy làm ví dụ trong VVC

Cú pháp RBSP của 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)
...	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	

subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
}	

subpics_present_flag bằng 1 chỉ báo rằng các tham số ảnh phụ xuất hiện trong cờ hiện tại trong cú pháp RSSP SPS. subpics_present_flag bằng 0 chỉ báo rằng các tham số ảnh phụ không xuất hiện trong cờ hiện tại trong cú pháp RSSP SPS.

Lưu ý 2 – Khi dòng bit là kết quả của quá trình trích xuất dòng bit phụ và chỉ chứa tập con của các ảnh phụ của dòng bit đầu vào với quá trình trích xuất dòng bit phụ, có thể yêu cầu thiết lập giá trị của subpics_present_flag bằng 1 trong RBSP của các SPS.

max_subpics_minus1 cộng 1 xác định số lượng lớn nhất của các ảnh phụ có thể xuất hiện trong CVS. max_subpics_minus1 phải nằm trong khoảng từ 0 đến 254. Giá trị 255 được dành riêng để sử dụng sau này bởi ITU-T | ISO/IEC.

subpic_grid_col_width_minus1 cộng 1 xác định chiều rộng của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của 4 mẫu. Chiều dài của phần tử cú pháp là Ceil(Log2(pic_width_max_in_luma_samples / 4)) bit.

Biến NumSubPicGridCols được dẫn xuất như sau:

NumSubPicGridCols =
(pic_width_max_in_luma_samples + subpic_grid_col_width_minus1 * 4 + 3) /
(subpic_grid_col_width_minus1 * 4 + 4) (7-5)

subpic_grid_row_height_minus1 cộng 1 xác định chiều cao của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của 4 mẫu. Chiều dài của phần tử cú pháp là Ceil(Log2(pic_height_max_in_luma_samples / 4)) bit.

Biến NumSubPicGridRows được dẫn xuất như sau:

NumSubPicGridRows =
(pic_height_max_in_luma_samples + subpic_grid_row_height_minus1 * 4 + 3) /
(subpic_grid_row_height_minus1 * 4 + 4) (7-6)

subpic_grid_idx[i][j] xác định chỉ số ảnh phụ của vị trí lưới (i, j). Chiều dài của phần tử cú pháp bằng Ceil(Log2(max_subpics_minus1 + 1)) bit.

Các biến SubPicTop[subpic_grid_idx[i][j]], SubPicLeft[subpic_grid_idx[i][j]], SubPicWidth[subpic_grid_idx [i][j]], SubPicHeight[subpic_grid_idx[i][j]], và NumSubPics được dẫn xuất như sau:

```
NumSubPics = 0
for(i = 0; i. < NumSubPicGridRows; i++) {
        for(j = 0; j < NumSubPicGridCols; j++) {
            if (i = = 0)
                SubPicTop[subpic_grid_idx[i][j]] = 0
            else if(subpic_grid_idx[i][j] != subpic_grid_idx[i – 1][j]) {
                SubPicTop[subpic_grid_idx[i][j]] = i

        SubPicHeight[subpic_grid_idx[i – 1][j]] =
i – SubPicTop[subpic_grid_idx[i – 1][j]]
            }
            if (j = = 0)
                SubPicLeft[subpic_grid_idx[i][j]] = 0
            else if (subpic_grid_idx[i][j] != subpic_grid_idx[i][j – 1]) {
                SubPicLeft[subpic_grid_idx[i][j]] = j
        SubPicWidth[subpic_grid_idx[i][j]] =
j – SubPicLeft[subpic_grid_idx[i][j – 1]]
            }
            if (i = = NumSubPicGridRows – 1)
        SubPicHeight[subpic_grid_idx[i][j]] =
i – SubPicTop[subpic_grid_idx[i – 1][j]] + 1
            if (j = = NumSubPicGridRows – 1)
        SubPicWidth[subpic_grid_idx[i][j]] =
j – SubPicLeft[subpic_grid_idx[i][j – 1]] + 1
            if(subpic_grid_idx[i][j] > NumSubPics)
                NumSubPics = subpic_grid_idx[i][j]
        }
}
```

subpic_treated_as_pic_flag[i] bằng 1 xác định rằng ảnh phụ thứ i của mỗi ảnh được tạo mã trong CVS được xử lý như là ảnh trong quá trình giải mã không bao gồm các hoạt động lọc trong vòng. subpic_treated_as_pic_flag[i] bằng 0 xác định rằng ảnh phụ thứ i của mỗi ảnh được tạo mã trong CVS không được xử lý như là ảnh trong quá trình giải mã không bao gồm các hoạt động lọc trong vòng. Khi không có mặt, giá trị của subpic_treated_as_pic_flag[i] được suy ra bằng 0.

loop_filter_across_subpic_enabled_flag[i] bằng 1 xác định rằng các hoạt động lọc trong vòng có thể được thực hiện giữa các đường biên của ảnh phụ thứ i trong mỗi ảnh được tạo mã trong CVS. loop_filter_across_subpic_enabled_flag[i] bằng 0 xác định rằng các hoạt động lọc trong vòng không được thực hiện giữa các đường biên của ảnh phụ thứ i trong mỗi ảnh được tạo mã trong CVS. Khi không có mặt, giá trị của loop_filter_across_subpic_enabled_pic_flag[i] được suy ra bằng 1.

Tương hợp dòng bit yêu cầu rằng thỏa mãn các ràng buộc sau:

- Đối với hai ảnh phụ bất kỳ subpicA và subpicB, khi chỉ số của subpicA nhỏ hơn chỉ số của subpicB, bất kỳ đơn vị NAL được mã hóa nào của subPicA phải đứng sau bất kỳ đơn vị NAL được mã hóa nào của subPicB theo thứ tự giải mã.
- Các hình dạng của các ảnh phụ sẽ phải sao cho mỗi ảnh phụ, khi được giải mã, phải có toàn bộ đường biên trái và toàn bộ đường biên trên bao gồm các đường biên ảnh hoặc bao gồm các đường biên của các ảnh phụ được giải mã trước đó.

Danh sách CtbToSubPicIdx[ctbAddrRs] với ctbAddrRs nằm trong khoảng từ 0 đến PicSizeInCtbsY − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ địa chỉ CTB theo thứ tự quét mành ảnh đến chỉ số ảnh phụ, được dẫn xuất như sau:

```
for(ctbAddrRs = 0; ctbAddrRs < PicSizeInCtbsY; ctbAddrRs++) {
    posX = ctbAddrRs % PicWidthInCtbsY * CtbSizeY
    posY = ctbAddrRs / PicWidthInCtbsY * CtbSizeY
    CtbToSubPicIdx[ctbAddrRs] = −1
```

```
for(i = 0; CtbToSubPicIdx[ctbAddrRs] < 0  && i < NumSubPics; i++) {
  if((posX >= SubPicLeft[i] * (subpic_grid_col_width_minus1 + 1) * 4)
&&
  (posX < (SubPicLeft[i] + SubPicWidth[i]) *
          (subpic_grid_col_width_minus1 + 1) * 4)  &&
          (posY >= SubPicTop[i] *
          (subpic_grid_row_height_minus1 + 1) * 4)  &&
          (posY < (SubPicTop[i] + SubPicHeight[i]) *
            (subpic_grid_row_height_minus1 + 1) * 4))
        CtbToSubPicIdx[ctbAddrRs] = i
  }
}
```

num_bricks_in_slice_minus1, nếu xuất hiện, xác định số lượng khối viên trong lát trừ 1. Giá trị của num_bricks_in_slice_minus1 phải nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút. Khi rect_slice_flag bằng 0 và single_brick_per_slice_flag bằng 1, giá trị của num_bricks_in_slice_minus1 được suy ra bằng 0. Khi single_brick_per_slice_flag bằng 1, giá trị của num_bricks_in_slice_minus1 được suy ra bằng 0.

Biến NumBricksInCurrSlice, xác định số lượng khối viên trong lát hiện tại, và SliceBrickIdx[i], xác định chỉ số khối viên của khối viên thứ i trong lát hiện tại, được dẫn xuất như sau:

```
if(rect_slice_flag) {
  sliceIdx = 0
  while(slice_address != slice_id[sliceIdx])
    sliceIdx++
  NumBricksInCurrSlice = NumBricksInSlice[sliceIdx]
  brickIdx = TopLeftBrickIdx[sliceIdx]
  for(bIdx = 0; brickIdx <= BottomRightBrickIdx[sliceIdx]; brickIdx++)
(7-92)
    if(BricksToSliceMap[brickIdx] == sliceIdx)
      SliceBrickIdx[bIdx++] = brickIdx
} else {
```

```
        NumBricksInCurrSlice = num_bricks_in_slice_minus1 + 1
        SliceBrickIdx[0] = slice_address
        for(i = 1; i < NumBricksInCurrSlice; i++)
            SliceBrickIdx[i] = SliceBrickIdx[i − 1] + 1
    }
```

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

```
SubPicIdx = CtbToSubPicIdx[CtbAddrBsToRs[FirstCtbAddrBs[SliceBrickIdx[0]]]]
if(subpic_treated_as_pic_flag[SubPicIdx]) {
SubPicLeftBoundaryPos =
SubPicLeft[SubPicIdx] * (subpic_grid_col_width_minus1 + 1) * 4
SubPicRightBoundaryPos = (SubPicLeft[SubPicIdx] + SubPicWidth[SubPicIdx]) *
                        (subpic_grid_col_width_minus1 + 1) * 4
SubPicTopBoundaryPos =
SubPicTop[SubPicIdx] * (subpic_grid_row_height_minus1 + 1)* 4
SubPicBotBoundaryPos = (SubPicTop[SubPicIdx] + SubPicHeight[SubPicIdx]) *
                        (subpic_grid_row_height_minus1 + 1) * 4
}
...
```

Quá trình dẫn xuất đối với dự báo MV độ sáng thời gian

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của khối mã độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- biến cbWidth xác định chiều rộng của khối mã hiện tại trong mẫu độ sáng,
- biến cbHeight xác định chiều cao của khối mã hiện tại trong mẫu độ sáng,
- chỉ số tham chiếu refIdxLX, với X bằng 0 hoặc 1.

Các đầu ra của quá trình này là:

- dự báo vectơ chuyển động mvLXCol có độ chính xác mẫu phân số 1/16,
- cờ khả dụng availableFlagLXCol.

Biến currCb xác định khối mã độ sáng hiện tại ở vị trí độ sáng (xCb, yCb).

Các biến mvLXCol và availableFlagLXCol được dẫn xuất như sau:

– Nếu slice_temporal_mvp_enabled_flag bằng 0 hoặc (cbWidth * cbHeight) nhỏ hơn hoặc bằng 32, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.
– Ngược lại (slice_temporal_mvp_enabled_flag bằng 1), các bước có thứ tự sau áp dụng:
1. MV cùng vị trí dưới bên phải và các vị trí mẫu đường biên dưới và bên phải được dẫn xuất như sau:

xColBr = xCb + cbWidth (8-421)

yColBr = yCb + cbHeight(8-422)

rightBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ?
SubPicRightBoundaryPos : pic_width_in_luma_samples – 1
(8-423)

botBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ?
SubPicBotBoundaryPos : pic_height_in_luma_samples – 1
(8-424)

– Nếu yCb >> CtbLog2SizeY bằng yColBr >> CtbLog2SizeY, yColBr nhỏ hơn hoặc bằng botBoundaryPos và xColBr nhỏ hơn hoặc bằng rightBoundaryPos, điều kiện sau áp dụng:
– Biến colCb xác định khối mã độ sáng bao phủ vị trí được chỉnh sửa được tạo bởi ((xColBr >> 3) << 3, (yColBr >> 3) << 3) bên trong ảnh cùng vị trí được xác định bởi ColPic.
– Vị trí độ sáng (xColCb, yColCb) được gán bằng mẫu trên bên trái của khối mã độ sáng cùng vị trí được xác định bởi colCb so với mẫu độ sáng trên bên trái của ảnh cùng vị trí được xác định bởi ColPic.
– Quá trình dẫn xuất đối với các MV cùng vị trí như được xác định trong mệnh đề 8.5.2.12 được gọi với currCb, colCb, (xColCb, yColCb), refIdxLX và sbFlag được gán bằng 0 làm đầu vào, và đầu ra được gán cho mvLXCol và availableFlagLXCol.

Ngược lại, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.

...

Quá trình nội suy song tuyến tính mẫu độ sáng

Các đầu vào cho quá trình này là:

– vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_L$, $yInt_L$),

– vị trí độ sáng theo các đơn vị mẫu phân số ($xFrac_L$, $yFrac_L$),

– mảng mẫu tham chiếu độ sáng $refPicLX_L$.

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

Các biến shift1, shift2, shift3, shift4, offset1, offset2 và offset3 được dẫn xuất như sau:

$shift1 = BitDepth_Y - 6$ (8-453)

$offset1 = 1 << (shift1 - 1)$ (8-454)

$shift2 = 4$ (8-455)

$offset2 = 1 << (shift2 - 1)$ (8-456)

$shift3 = 10 - BitDepth_Y$ (8-457)

$shift4 = BitDepth_Y - 10$ (8-458)

$offset4 = 1 << (shift4 - 1)$ (8-459)

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

Các hệ số lọc nội suy độ sáng $fb_L[p]$ đối với mỗi vị trí mẫu phân số 1/16 p bằng $xFrac_L$ hoặc $yFrac_L$ được xác định trong Bảng 8-10.

Các vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_i$, $yInt_i$) được dẫn xuất như sau với i = 0..1:

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

$xInt_i = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i)$ (8-460)

$yInt_i = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i)$ (8-461)

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

$xInt_i$ = Clip3(0, picW − 1, sps_ref_wraparound_enabled_flag ?

ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, ($xInt_L$ + i)) :

(8-462)

$xInt_L$ + i)

$yInt_i$ = Clip3(0, picH − 1, $yInt_L$ + i) (8-463)

...

Quá trình dẫn xuất đối với các ứng viên hợp nhất thời gian dựa trên khối phụ

Các đầu vào cho quá trình này là:

– vị trí độ sáng (xCb, yCb) của mẫu trên bên trái của khối mã độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
– biến cbWidth xác định chiều rộng của khối mã hiện tại trong mẫu độ sáng,
– biến cbHeight xác định chiều cao của khối mã hiện tại trong mẫu độ sáng.
– cờ khả dụng $availableFlagA_1$ của CU lân cận,
– chỉ số tham chiếu $refIdxLXA_1$ của CU lân cận với X bằng 0 hoặc 1,
– cờ sử dụng danh sách dự báo $predFlagLXA_1$ của CU lân cận với X bằng 0 hoặc 1,
– vectơ chuyển động có độ chính xác mẫu phân số 1/16 $mvLXA_1$ của CU lân cận với X bằng 0 hoặc 1.

Các đầu ra của quá trình này là:

– cờ khả dụng availableFlagSbCol,
– số lượng của các khối mã phụ độ sáng theo chiều ngang numSbX và theo chiều dọc numSbY,
– các chỉ số tham chiếu refIdxL0SbCol và refIdxL1SbCol,
– các MV độ sáng có độ chính xác mẫu phân số 1/16 mvL0SbCol[xSbIdx][ySbIdx] và mvL1SbCol[xSbIdx][ySbIdx] với xSbIdx = 0..numSbX − 1, ySbIdx = 0 .. numSbY − 1,

– các cờ sử dụng danh sách dự báo predFlagL0SbCol[xSbIdx][ySbIdx] và predFlagL1SbCol[xSbIdx][ySbIdx] với xSbIdx = 0..numSbX − 1, ySbIdx = 0 .. numSbY − 1.

Cờ khả dụng availableFlagSbCol được dẫn xuất như sau.

– Nếu một hoặc nhiều điều kiện sau là đúng, availableFlagSbCol được gán bằng 0.
  – slice_temporal_mvp_enabled_flag bằng 0.
  – sps_sbtmvp_enabled_flag bằng 0.
  – cbWidth nhỏ hơn 8.
  – cbHeight nhỏ hơn 8.
– Ngược lại, các bước có thứ tự sau áp dụng:
  1. Vị trí (xCtb, yCtb) của mẫu trên bên trái của CTB độ sáng mà chứa khối mã hiện tại và vị trí (xCtr, yCtr) của mẫu trung tâm dưới bên phải của khối mã độ sáng hiện tại được dẫn xuất như sau:

     xCtb = (xCb >> CtuLog2Size) << CtuLog2Size

     (8-542)

     yCtb = (yCb >> CtuLog2Size) << CtuLog2Size

     (8-543)

     xCtr = xCb + (cbWidth / 2)

     (8-544)

     yCtr = yCb + (cbHeight / 2)

     (8-545)
  2. Vị trí độ sáng (xColCtrCb, yColCtrCb) được gán bằng mẫu trên bên trái của khối mã độ sáng cùng vị trí bao phủ vị trí được tạo bởi (xCtr, yCtr) bên trong ColPic so với mẫu độ sáng trên bên trái của ảnh cùng vị trí được xác định bởi ColPic.
  3. Quá trình dẫn xuất đối với dữ liệu cơ sở hợp nhất thời gian dựa trên khối phụ như được xác định trong mệnh đề 8.5.5.4 được gọi với vị trí (xCtb, yCtb), vị trí (xColCtrCb, yColCtrCb), cờ khả dụng availableFlagA_1, và cờ sử dụng danh sách dự báo predFlagLXA_1, và chỉ số tham chiếu refIdxLXA_1, và vectơ chuyển động mvLXA_1, với X bằng

0 và 1 làm đầu vào và các vectơ chuyển động ctrMvLX, và các cờ sử dụng danh sách dự báo ctrPredFlagLX của khối cùng vị trí, với X bằng 0 và 1, và MV thời gian tempMv làm đầu ra.

4. Biến availableFlagSbCol được dẫn xuất như sau:
   - Nếu cả ctrPredFlagL0 và ctrPredFlagL1 bằng 0, availableFlagSbCol được gán bằng 0.
   - Ngược lại, availableFlagSbCol được gán bằng 1.

Khi availableFlagSbCol bằng 1, điều kiện sau áp dụng:

- Các biến numSbX, numSbY, sbWidth, sbHeight và refIdxLXSbCol được dẫn xuất như sau:

  numSbX = cbWidth >> 3 (8-546)

  numSbY = cbHeight >> 3 (8-547)

  sbWidth = cbWidth / numSbX(8-548)

  sbHeight = cbHeight / numSbY(8-549)

  refIdxLXSbCol = 0 (8-550)

- Với xSbIdx = 0..numSbX − 1 và ySbIdx = 0 .. numSbY − 1, các vectơ chuyển động mvLXSbCol[xSbIdx][ySbIdx] và cờ sử dụng các danh sách dự báo predFlagLXSbCol[xSbIdx][ySbIdx] được dẫn xuất như sau:
  - Vị trí độ sáng (xSb, ySb) xác định mẫu trên bên trái của khối phụ mã hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại được dẫn xuất như sau:

    xSb = xCb + xSbIdx * sbWidth + sbWidth / 2

    (8-551)

    ySb = yCb + ySbIdx * sbHeight + sbHeight / 2

    (8-552)
  - Vị trí (xColSb, yColSb) của khối phụ đồng bị trí bên trong ColPic được dẫn xuất như sau.
    - Điều kiện sau áp dụng:

yColSb = Clip3(yCtb,

Min(CurPicHeightInSamplesY − 1, yCtb + (1 << CtbLog2SizeY) − 1),

(8-553)

ySb + (tempMv[1] >> 4))

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

xColSb = Clip3(xCtb,

Min(SubPicRightBoundaryPos, xCtb + (1 << CtbLog2SizeY) + 3),

(8-554)

xSb + (tempMv[0] >> 4))

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

xColSb =

Clip3(xCtb,Min(CurPicWidthInSamplesY − 1, xCtb + (1 << CtbLog2 SizeY) + 3),

(8-555)

xSb + (tempMv[0] >> 4))

...

Quá trình dẫn xuất đối với dữ liệu chuyển động cơ sở hợp nhất thời gian dựa trên khối phụ

Các đầu vào cho quá trình này là:

– vị trí (xCtb, yCtb) của mẫu trên bên trái của CTB độ sáng mà chứa khối mã hiện tại,

– vị trí (xColCtrCb, yColCtrCb) của mẫu trên bên trái của khối mã độ sáng cùng vị trí bao phủ mẫu trung tâm dưới bên phải.

– cờ khả dụng availableFlagA_1 của CU lân cận,

– chỉ số tham chiếu refIdxLXA_1 của CU lân cận,

– cờ sử dụng danh sách dự báo predFlagLXA_1 của CU lân cận,

– vectơ chuyển động có độ chính xác mẫu phân số 1/16 mvLXA_1 của CU lân cận.

Các đầu ra của quá trình này là:

– các vectơ chuyển động ctrMvL0 và ctrMvL1,

– các cờ sử dụng danh sách dự báo ctrPredFlagL0 và ctrPredFlagL1,

– MV thời gian tempMv.

Biến tempMv được thiết lập như sau:

tempMv[0] = 0 (8-558)

tempMv[1] = 0 (8-559)

Biến currPic xác định ảnh hiện tại.

Khi availableFlagA_1 bằng TRUE, điều kiện sau áp dụng:

- Nếu tất cả điều kiện sau là đúng, tempMv được gán bằng mvL0A_1:
  - predFlagL0A_1 bằng 1,
  - DiffPicOrderCnt(ColPic, RefPicList[0][refIdxL0A_1]) bằng 0,
- Ngược lại, nếu tất cả điều kiện sau là đúng, tempMv được gán bằng mvL1A_1:
  - slice_type bằng B,
  - predFlagL1A_1 bằng 1,
  - DiffPicOrderCnt(ColPic, RefPicList[1][refIdxL1A_1]) bằng 0.

Vị trí (xColCb, yColCb) của khối đồng vị trí bên trong ColPic được dẫn xuất như sau.

- Điều kiện sau áp dụng:

  yColCb = Clip3(yCtb,
  Min(CurPicHeightInSamplesY − 1, yCtb + (1 << CtbLog2SizeY) − 1),
  (8-560)
  yColCtrCb + (tempMv[1] >> 4))

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

  xColCb = Clip3(xCtb,
  Min(SubPicRightBoundaryPos, xCtb + (1 << CtbLog2SizeY) + 3),
  (8-561)
  xColCtrCb + (tempMv[0] >> 4))

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

xColCb = Clip3(xCtb,
Min(CurPicWidthInSamplesY − 1, xCtb + (1 << CtbLog2SizeY) + 3),

(8-562)

xColCtrCb + (tempMv[0] >> 4))

...

Quá trình lọc nội suy mẫu độ sáng

Các đầu vào cho quá trình này là:

- vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_L$, $yInt_L$),
- vị trí độ sáng theo các đơn vị mẫu phân số ($xFrac_L$, $yFrac_L$),
- vị trí độ sáng theo các đơn vị mẫu toàn phần ($xSbInt_L$, $ySbInt_L$) xác định mẫu trên bên trái của khối giới hạn để đệm mẫu tham chiếu so với mẫu độ sáng trên bên trái của ảnh tham chiếu,
- mảng mẫu tham chiếu độ sáng $refPicLX_L$,
- chỉ số bộ lọc nội suy nửa mẫu hpelIfIdx,
- biến sbWidth xác định chiều rộng của khối phụ hiện tại,
- biến sbHeight xác định chiều cao của khối phụ hiện tại,
- vị trí độ sáng (xSb, ySb) xác định mẫu trên bên trái của khối phụ hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,

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

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

- Biến shift1 được gán bằng Min(4, $BitDepth_Y$ − 8), biến shift2 được gán bằng 6 và biến shift3 được gán bằng Max(2, 14 − $BitDepth_Y$).
- Biến picW được gán bằng pic_width_in_luma_samples và biến picH được gán bằng pic_height_in_luma_samples.

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

- Nếu MotionModelIdc[xSb][ySb] lớn hơn 0, và sbWidth và sbHeight đều bằng 4, các hệ số lọc nội suy độ sáng $f_L[p]$ được xác định trong Bảng 8-12.
- Ngược lại, các hệ số lọc nội suy độ sáng $f_L[p]$ được xác định trong Bảng 8-11 tùy thuộc hpelIfIdx.

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

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

$xInt_i = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i - 3)$ (8-771)

$yInt_i = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i - 3)$ (8-772)

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

$xInt_i = Clip3(0, picW - 1, sps\_ref\_wraparound\_enabled\_flag ?$
$ClipH((sps\_ref\_wraparound\_offset\_minus1 + 1) * MinCbSizeY, picW, xInt_L + i - 3) :$ (8-773)
$xInt_L + i - 3)$

$yInt_i = Clip3(0, picH - 1, yInt_L + i - 3)$ (8-774)

...

Quá trình nội suy mẫu sắc độ

Các đầu vào cho quá trình này là:

– vị trí sắc độ theo các đơn vị mẫu toàn phần ($xInt_C$, $yInt_C$),

– vị trí sắc độ theo các đơn vị mẫu phân số 1/32 ($xFrac_C$, $yFrac_C$),

– vị trí sắc độ theo các đơn vị mẫu toàn phần (xSbIntC, ySbIntC) xác định mẫu trên bên trái của khối giới hạn để đệm mẫu tham chiếu so với mẫu sắc độ trên bên trái của ảnh tham chiếu,

– biến sbWidth xác định chiều rộng của khối phụ hiện tại,

– biến sbHeight xác định chiều cao của khối phụ hiện tại,

– mảng mẫu tham chiếu sắc độ $refPicLX_C$.

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

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

– Biến shift1 được gán bằng Min(4, $BitDepth_C$ - 8), biến shift2 được gán bằng 6 và biến shift3 được gán bằng Max(2, 14 - $BitDepth_C$).

– Biến $picW_C$ được gán bằng pic_width_in_luma_samples / SubWidthC và biến $picH_C$ được gán bằng pic_height_in_luma_samples / SubHeightC.

Các hệ số lọc nội suy sắc độ $f_C[p]$ đối với mỗi vị trí mẫu phân số 1/32 p bằng $xFrac_C$ hoặc $yFrac_C$ được xác định trong Bảng 8-13.

Biến xOffset được gán bằng (sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY) / SubWidthC.

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

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

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

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

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

$xInt_i$ = Clip3(0, $picW_C$ − 1,
sps_ref_wraparound_enabled_flag ? ClipH(xOffset, $picW_C$, $xInt_C$ + i − 1) : (8-787)
$xInt_C$ + i − 1)

$yInt_i$ = Clip3(0, $picH_C$ − 1, $yInt_C$ + i − 1) (8-788)

2.4 Ví dụ về bộ lọc thời gian dựa trên GOP riêng bộ mã hóa

Theo một số phương án thực hiện, bộ lọc thời gian riêng cho bộ mã hóa có thể được triển khai. Việc lọc được thực hiện ở phía bộ mã hóa như là bước tiền xử lý. Các ảnh nguồn trước và sau ảnh được chọn để mã hóa được đọc và phương pháp bù chuyển động dựa trên khối so với ảnh được chọn được áp dụng trên các ảnh nguồn đó. Các mẫu ở ảnh được chọn được lọc theo thời gian nhờ sử dụng các giá trị mẫu sau khi bù chuyển động.

Toàn bộ độ bền bộ lọc được thiết lập tùy thuộc lớp phụ thời gian của ảnh được chọn cũng như QP. Chỉ các ảnh ở các lớp phụ thời gian 0 và 1 được lọc và các ảnh thuộc lớp 0 được lọc bởi bộ lọc mạnh hơn các ảnh của lớp 1. Độ bền bộ lọc theo mẫu được điều chỉnh tùy thuộc hiệu số giữa giá trị mẫu trong ảnh được chọn và các mẫu cùng vị trí ở các ảnh được bù chuyển động sao cho các độ lệch nhỏ giữa ảnh được bù chuyển động và ảnh được chọn được lọc mạnh hơn các độ lệch lớn hơn.

Bộ lọc thời gian dựa trên GOP

Bộ lọc thời gian được đưa vào trực tiếp sau khi đọc ảnh và trước khi mã hóa. Dưới đây là các bước được mô tả chi tiết hơn.

Hoạt động 1: Các ảnh được đọc bằng bộ mã hóa

Hoạt động 2: Nếu ảnh đủ thấp trong phân cấp mã hóa, nó được lọc trước khi mã hóa. Ngược lại, ảnh được mã hóa mà không lọc. Các ảnh RA có POC % 8 == 0 được lọc cũng như các ảnh LD có POC % 4 == 0. Các ảnh AI không bao giờ được lọc.

Toàn bộ độ bền bộ lọc, s_o, được thiết lập theo phương trình dưới đây cho RA.

$$s_o(n) = \begin{cases} 1.5, & n \bmod 16 = 0 \\ 0.95, & n \bmod 16 \neq 0 \end{cases}$$

trong đó n là số lượng ảnh được đọc.

Với trường hợp LD, $s_o(n) = 0{,}95$ được sử dụng.

Hoạt động 3: Hai ảnh trước và/hoặc sau ảnh được chọn (được gọi dưới đây là ảnh ban đầu) được đọc. Trong các trường hợp mép ảnh, chẳng hạn, nếu nó là ảnh thứ nhất và gần với ảnh cuối, chỉ các ảnh khả dụng được đọc.

Hoạt động 4: Chuyển động của các ảnh được đọc trước và sau, so với ảnh ban đầu được ước tính theo khối ảnh 8x8.

Phương tiện ước tính chuyển động phân cấp được sử dụng và các lớp L0, L1 và L2, được minh họa trên Fig.2. Các ảnh được lấy mẫu phụ được tạo bằng cách tính trung bình mỗi khối 2x2 cho tất cả các ảnh được đọc và ảnh ban đầu, chẳng hạn L1 trên Fig.1. L2 được dẫn xuất từ L1 nhờ sử dụng cùng phương pháp lấy mẫu phụ.

Fig.2 thể hiện các ví dụ của các lớp khác nhau của phương tiện ước tính chuyển động phân cấp. L0 là độ phân giải ban đầu. L1 là phiên bản được lấy mẫu phụ của L0. L2 là phiên bản được lấy mẫu phụ của L1.

Trước hết, ước tính chuyển động được thực hiện đối với mỗi khối 16x16 trong L2. Hiệu số bình phương được tính toán đối với mỗi MV được chọn và vectơ chuyển động tương ứng với hiệu số nhỏ nhất được chọn. Sau đó, MV được chọn được sử dụng làm giá trị ban đầu khi ước tính chuyển động trong L1. Sau đó, quá trình tương tự được thực hiện để ước tính chuyển động trong L0. Như là bước cuối cùng, chuyển động pixel phụ được ước tính đối với mỗi khối 8x8 nhờ sử dụng bộ lọc nội suy trên L0.

Bộ lọc nội suy 6 bậc VTM có thể được sử dụng:

```
0:   0,   0,  64,   0,   0,   0
1:   1,  -3,  64,   4,  -2,   0
2:   1,  -6,  62,   9,  -3,   1
3:   2,  -8,  60,  14,  -5,   1
4:   2,  -9,  57,  19,  -7,   2
5:   3, -10,  53,  24,  -8,   2
6:   3, -11,  50,  29,  -9,   2
7:   3, -11,  44,  35, -10,   3
8:   1,  -7,  38,  38,  -7,   1
9:   3, -10,  35,  44, -11,   3
10:  2,  -9,  29,  50, -11,   3
11:  2,  -8,  24,  53, -10,   3
12:  2,  -7,  19,  57,  -9,   2
13:  1,  -5,  14,  60,  -8,   2
14:  1,  -3,   9,  62,  -6,   1
15:  0,  -2,   4,  64,  -3,   1
```

Hoạt động 5: Bù chuyển động được áp dụng trên các ảnh trước và sau ảnh ban đầu theo chuyển động so khớp tốt nhất đối với mỗi khối, chẳng hạn, sao cho các tọa độ mẫu của ảnh ban đầu trong mỗi khối có các tọa độ so khớp tốt nhất trong các ảnh tham chiếu.

Hoạt động 6: Các mẫu của từng ảnh được xử lý đối với các kênh độ sáng và sắc độ như được mô tả trong các bước sau.

Hoạt động 7: Giá trị mẫu mới, I_n, được tính toán nhờ sử dụng công thức sau.

$$I_n = \frac{I_o + \sum_{i=0}^{3} w_r(i,a) I_r(i)}{1 + \sum_{i=0}^{3} w_r(i,a)}$$

Trong đó I_o là giá trị mẫu của mẫu ban đàu, $I_r(i)$ là cường độ của mẫu tương ứng của ảnh được bù chuyển động i và $w_r(i,a)$ là trọng số của ảnh được bù chuyển động i khi số lượng ảnh được bù chuyển động khả dụng là a.

Trong kênh độ sáng, các trọng số, $w_r(i,a)$, được định nghĩa như sau:

$$w_r(i,a) = s_l s_o(n) s_r(i,a) e^{-\frac{\Delta I(i)^2}{2\sigma_l (QP)^2}}$$

Trong đó

$$s_l = 0.4$$

$$s_r(i,2) = \begin{cases} 1.2, & i = 0 \\ 1.0, & i = 1 \end{cases}$$

$$s_r(i,4) = \begin{cases} 0.60, & i = 0 \\ 0.85, & i = 1 \\ 0.85, & \mathrm{i} = 2 \\ 0.60, & i = 3 \end{cases}$$

Đối với tất cả các trường hợp khác của i, và a: $s_r(i,a) = 0{,}3$

$$\sigma_l(QP) = 3 * (QP - 10)$$

$$\Delta I(i) = I_r(i) - I_o$$

Đối với các kênh sắc độ, các trọng số, $w_r(i,a)$, được định nghĩa như sau:

$$w_r(i,a) = s_c s_o(n) s_r(i,a) e^{-\frac{\Delta I(i)^2}{2{\sigma_c}^2}}$$

Trong đó $s_c = 0{,}55$ và $\sigma_c = 30$

Hoạt động 8: Bộ lọc được áp dụng cho mẫu hiện tại. Giá trị mẫu thu được được lưu trữ riêng rẽ.

Hoạt động 9: Ảnh được lọc sẽ được mã hóa.

2.5 Các phân vùng ảnh ví dụ (ô, khối viên, lát)

Theo một số phương án thực hiện, ảnh được phân chia thành một hoặc nhiều hàng ô và một hoặc nhiều cột ô. Ô là chuỗi CTU bao phủ vùng chữ nhật của ảnh.

Ô được phân chia thành một hoặc nhiều khối viên, mỗi khối viên trong số đó bao gồm số lượng hàng CTU trong ô.

Ô không được phân vùng thành nhiều khối viên cũng được gọi là khối viên. Tuy nhiên, khối viên là tập con thực của ô không được gọi là ô.

Lát chứa số lượng ô của ảnh hoặc số lượng khối viên của ô.

Ảnh phụ chứa một hoặc nhiều lát bao phủ chung vùng chữ nhật của ảnh.

Hai chế độ của lát được hỗ trợ, tức là chế độ lát quét mành và chế độ lát chữ nhật. Trong chế độ lát quét mành, lát chứa chuỗi ô theo thứ tự quét mành ô của ảnh. Trong chế độ lát chữ nhật, lát chứa số lượng khối viên của ảnh tạo chung thành vùng chữ nhật của ảnh. Các khối viên trong lát chữ nhật theo thứ tự quét mành khối viên của lát.

Fig.5 thể hiện ví dụ về phân vùng lát quét mành của ảnh, trong đó ảnh được phân chia thành 12 ô và 3 lát quét mành.

Fig.6 thể hiện ví dụ về phân vùng lát chữ nhật của ảnh, trong đó ảnh được phân chia thành 24 ô (6 cột ô và 4 hàng ô) và 9 lát chữ nhật.

Fig.7 thể hiện ví dụ về ảnh được phân vùng thành các ô, các khối viên, và lát chữ nhật, trong đó ảnh được phân chia thành 4 ô (2 cột ô và 2 hàng ô), 11 khối viên (ô trên bên trái chứa 1 khối viên, ô trên bên phải chứa 5 khối viên, ô dưới bên trái chứa 2 khối viên, và ô dưới bên phải chứa 3 khối viên), và 4 lát chữ nhật.

Cú pháp RBSP của tập hợp tham số ảnh

pic_parameter_set_rbsp() {	Mô tả
...	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
uniform_tile_spacing_flag	u(1)
if(uniform_tile_spacing_flag) {	
tile_cols_width_minus1	ue(v)
tile_rows_height_minus1	ue(v)
} else {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
for(i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
brick_splitting_present_flag	u(1)
if(uniform_tile_spacing_flag && brick_splitting_present_flag)	
num_tiles_in_pic_minus1	ue(v)
for(i = 0; brick_splitting_present_flag && i <= num_tiles_in_pic_minus1 + 1; i++) {	
if(RowHeight[i] > 1)	
brick_split_flag[i]	u(1)
if(brick_split_flag[i]) {	
if(RowHeight[i] > 2)	
uniform_brick_spacing_flag[i]	u(1)
if(uniform_brick_spacing_flag[i])	
brick_height_minus1[i]	ue(v)
else {	
num_brick_rows_minus2[i]	ue(v)
for(j = 0; j <= num_brick_rows_minus2[i]; j++)	
brick_row_height_minus1[i][j]	ue(v)
}	
}	
}	
single_brick_per_slice_flag	u(1)
if(!single_brick_per_slice_flag)	
rect_slice_flag	u(1)
if(rect_slice_flag && !single_brick_per_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
bottom_right_brick_idx_length_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
bottom_right_brick_idx_delta[i]	u(v)
brick_idx_delta_sign_flag[i]	u(1)
}	

}	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	
if(rect_slice_flag) {	
signalled_slice_id_flag	u(1)
if(signalled_slice_id_flag) {	
signalled_slice_id_length_minus1	ue(v)
for(i = 0; i <= num_slices_in_pic_minus1; i++)	
slice_id[i]	u(v)
}	
}	
...	

slice_header() {	Descriptor
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)
...	

single_tile_in_pic_flag bằng 1 xác định rằng chỉ có một ô trong mỗi ảnh đề cập đến PPS. single_tile_in_pic_flag bằng 0 xác định rằng có nhiều hơn một ô trong mỗi ảnh đề cập đến PPS.

Lưu ý – Không có tách khối viên tiếp trong ô, toàn bộ ô được gọi là khối viên. Khi ảnh chứa chỉ một ô mà không tách khối viên tiếp, được gọi là một khối viên.

Tương hợp dòng bit yêu cầu rằng giá trị của single_tile_in_pic_flag sẽ là tương tự cho tất cả các PPS được viện dẫn bởi các ảnh được mã hóa trong CVS.

uniform_tile_spacing_flag bằng 1 xác định rằng các đường biên cột ô và các đường biên hàng ô tương tự được phân phối đồng nhất giữa ảnh và được báo hiệu nhờ sử dụng các phần tử cú pháp tile_cols_width_minus1 và tile_rows_height_minus1. uniform_tile_spacing_flag bằng 0 xác định rằng các đường biên cột ô và các đường biên hàng ô tương tự có thể hoặc có thể không được phân phối đồng nhất giữa ảnh và được báo hiệu nhờ sử dụng các phần tử cú pháp num_tile_columns_minus1 và num_tile_rows_minus1 và danh sách các cặp phần tử cú pháp tile_column_width_minus1[i] và

tile_row_height_minus1[i]. Khi không có mặt, giá trị của uniform_tile_spacing_flag được suy ra bằng 1.

tile_cols_width_minus1 cộng 1 xác định chiều rộng của cột ô không bao gồm cột ô cực phải của ảnh theo các đơn vị của các CTB khi uniform_tile_spacing_flag bằng 1. Giá trị tile_cols_width_minus1 phải nằm trong khoảng từ 0 đến PicWidthInCtbsY − 1, bao gồm hai giá trị đầu mút. Khi không có mặt, giá trị tile_cols_width_minus1 được suy ra bằng PicWidthInCtbsY − 1.

tile_rows_height_minus1 cộng 1 xác định chiều cao của hàng ô không bao gồm hàng ô dưới của ảnh theo các đơn vị của các CTB khi uniform_tile_spacing_flag bằng 1. Giá trị tile_rows_height_minus1 phải nằm trong khoảng từ 0 đến PicHeightInCtbsY − 1, bao gồm hai giá trị đầu mút. Khi không có mặt, giá trị tile_rows_height_minus1 được suy ra bằng PicHeightInCtbsY − 1.

num_tile_columns_minus1 cộng 1 xác định số lượng cột ô phân vùng ảnh khi uniform_tile_spacing_flag bằng 0. Giá trị của num_tile_columns_minus1 phải nằm trong khoảng từ 0 đến PicWidthInCtbsY − 1, bao gồm hai giá trị đầu mút. Nếu single_tile_in_pic_flag bằng 1, giá trị của num_tile_columns_minus1 được suy ra bằng 0. Ngược lại, khi uniform_tile_spacing_flag bằng 1, giá trị của num_tile_columns_minus1 được suy ra như được xác định trong mệnh đề 6.5.1.

num_tile_rows_minus1 cộng 1 xác định số lượng hàng ô phân vùng ảnh khi uniform_tile_spacing_flag bằng 0. Giá trị của num_tile_rows_minus1 phải nằm trong khoảng từ 0 to PicHeightInCtbsY − 1, bao gồm hai giá trị đầu mút. Nếu single_tile_in_pic_flag bằng 1, giá trị của num_tile_rows_minus1 được suy ra bằng 0. Ngược lại, khi uniform_tile_spacing_flag bằng 1, giá trị của num_tile_rows_minus1 được suy ra như được xác định trong mệnh đề 6.5.1.

Biến NumTilesInPic được gán bằng (num_tile_columns_minus1 + 1) * (num_tile_rows_minus1 + 1).

Khi single_tile_in_pic_flag bằng 0, NumTilesInPic sẽ lớn hơn 1.

tile_column_width_minus1[i] cộng 1 xác định chiều rộng của cột ô thứ I theo các đơn vị của các CTB.

tile_row_height_minus1[i] cột 1 xác định chiều cao của hàng ô thứ i theo các đơn vị của các CTB.

brick_splitting_present_flag bằng 1 xác định rằng một hoặc nhiều ô của ảnh đề cập đến PPS có thể được phân chia thành hai hoặc nhiều khối viên. brick_splitting_present_flag bằng 0 xác định rằng không ô nào của ảnh đề cập đến PPS được phân chia thành hai hoặc nhiều khối viên.

num_tiles_in_pic_minus1 cộng 1 xác định số lượng ô trong mỗi ảnh đề cập đến PPS. Giá trị của num_tiles_in_pic_minus1 sẽ bằng NumTilesInPic – 1. Khi không có mặt, giá trị của num_tiles_in_pic_minus1 được suy ra bằng NumTilesInPic − 1.

brick_split_flag[i] bằng 1 xác định rằng ô thứ i được phân chia thành hai hoặc nhiều khối viên. brick_split_flag[i] bằng 0 xác định rằng ô thứ i không được phân chia thành hai hoặc nhiều khối viên. Khi không có mặt, giá trị của brick_split_flag[i] được suy ra bằng 0. Theo một số phương án thực hiện, lệ thuộc phân tách PPS vào SPS được đưa vào bằng cách bổ sung điều kiện cú pháp "if(RowHeight[i] > 1)" (chẳng hạn, tương tự đối với uniform_brick_spacing_flag[i]).

uniform_brick_spacing_flag[i] bằng 1 xác định rằng các đường biên khối viên ngang được phân phối đồng nhất giữa ô thứ i và được báo hiệu nhờ sử dụng phần tử cú pháp brick_height_minus1[i]. uniform_brick_spacing_flag[i] bằng 0 xác định rằng các đường biên khối viên ngang có thể hoặc có thể không được phân phối đồng đều giữa ô thứ i và được báo hiệu nhờ sử dụng phần tử cú pháp num_brick_rows_minus2[i] và danh sách phần tử cú pháp brick_row_height_minus1[i][j]. Khi không có mặt, giá trị của uniform_brick_spacing_flag[i] được suy ra bằng 1.

brick_height_minus1[i] cộng 1 xác định chiều cao của các hàng khối viên không bao gồm khối viên dưới cùng ở ô thứ i theo các đơn vị của các CTB khi uniform_brick_spacing_flag[i] bằng 1. Nếu xuất hiện, giá trị của brick_height_minus1 phải nằm trong khoảng từ 0 đến RowHeight[i] − 2, bao gồm hai giá trị đầu mút. Khi không có mặt, giá trị của brick_height_minus1[i] được suy ra bằng RowHeight[i] − 1.

num_brick_rows_minus2[i] plus 2 xác định số lượng khối viên phân vùng ô thứ i khi uniform_brick_spacing_flag[i] bằng 0. Nếu xuất hiện, giá trị của num_brick_rows_minus2[i] phải nằm trong khoảng từ 0 đến RowHeight[i] − 2, bao gồm hai giá trị đầu mút. Nếu brick_split_flag[i] bằng 0, giá trị của num_brick_rows_minus2[i] được suy ra bằng −1. Ngược lại, khi uniform_brick_spacing_flag[i] bằng 1, giá trị của num_brick_rows_minus2[i] được suy ra như được xác định trong 6.5.1.

brick_row_height_minus1[i][j] plus 1 xác định chiều cao của khối viên thứ j ở ô thứ i theo các đơn vị của các CTB khi uniform_tile_spacing_flag bằng 0.

Các biến sau được dẫn xuất, và, khi uniform_tile_spacing_flag bằng 1, các giá trị của num_tile_columns_minus1 và num_tile_rows_minus1 được suy ra, và, đối với mỗi i nằm trong khoảng từ 0 đến NumTilesInPic − 1, bao gồm hai giá trị đầu mút, khi uniform_brick_spacing_flag[i] bằng 1, giá trị của num_brick_rows_minus2[i] được suy ra, bằng cách gọi quá trình biến đổi quét mành CTB và khối viên như được xác định trong mệnh đề 6.5.1:

- danh sách RowHeight[j] với j nằm trong khoảng từ 0 đến num_tile_rows_minus1, bao gồm hai giá trị đầu mút, xác định chiều cao của hàng ô thứ j theo các đơn vị của các CTB,
- danh sách CtbAddrRsToBs[ctbAddrRs] với ctbAddrRs nằm trong khoảng từ 0 đến PicSizeInCtbsY − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ địa chỉ CTB trong quét mành CTB của ảnh đến địa chỉ CTB trong quá trình quét khối viên,
- danh sách CtbAddrBsToRs[ctbAddrBs] với ctbAddrBs nằm trong khoảng từ 0 đến PicSizeInCtbsY − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ địa chỉ CTB trong quá trình quét khối viên đến địa chỉ CTB trong quá trình quét mành CTB của ảnh,
- danh sách BrickId[ctbAddrBs] với ctbAddrBs nằm trong khoảng từ 0 đến PicSizeInCtbsY − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ địa chỉ CTB trong quá trình quét khối viên đến ID khối viên,

- danh sách NumCtusInBrick[brickIdx] với brickIdx nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ chỉ số khối viên đến số lượng CTU trong khối viên,
- danh sách FirstCtbAddrBs[brickIdx] với brickIdx nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút, xác định phép biến đổi từ ID khối viên đến địa chỉ CTB trong quá trình quét khối viên của CTB thứ nhất trong khối viên.

single_brick_per_slice_flag bằng 1 xác định rằng mỗi lát đề cập đến PPS này bao gồm một khối viên. single_brick_per_slice_flag bằng 0 xác định rằng lát đề cập đến PPS này có thể bao gồm nhiều hơn một khối viên. Khi không có mặt, giá trị của single_brick_per_slice_flag được suy ra bằng 1.

rect_slice_flag bằng 0 xác định rằng các khối viên trong mỗi lát theo thứ tự quét mành và thông tin lát không được báo hiệu trong PPS. rect_slice_flag bằng 1 xác định rằng các khối viên trong mỗi lát bao phủ vùng chữ nhật của ảnh và thông tin lát được báo hiệu trong PPS. Khi brick_splitting_present_flag bằng 1, giá trị của rect_slice_flag sẽ bằng 1. Khi không có mặt, rect_slice_flag được suy ra bằng 1.

num_slices_in_pic_minus1 cộng 1 xác định số lượng lát trong mỗi ảnh đề cập đến PPS. Giá trị của num_slices_in_pic_minus1 phải nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút. Khi không có mặt và single_brick_per_slice_flag bằng 1, giá trị của num_slices_in_pic_minus1 được suy ra bằng NumBricksInPic − 1.

bottom_right_brick_idx_length_minus1 cộng 1 xác định số lượng của bits được sử dụng để biểu diễn phần tử cú pháp bottom_right_brick_idx_delta[i]. Giá trị của bottom_right_brick_idx_length_minus1 phải nằm trong khoảng từ 0 đến Ceil(Log2(NumBricksInPic)) − 1, bao gồm hai giá trị đầu mút.

bottom_right_brick_idx_delta[i] khi i lớn hơn 0 xác định hiệu số giữa chỉ số khối viên của khối viên được đặt ở góc dưới bên phải của lát thứ i và chỉ số khối viên của góc dưới bên phải của lát thứ (i − 1). bottom_right_brick_idx_delta[0] xác định chỉ số khối viên của góc dưới bên phải của lát thứ 0. Khi single_brick_per_slice_flag bằng 1, giá trị của bottom_right_brick_idx_delta[i]

được suy ra bằng 1. Giá trị của BottomRightBrickIdx[num_slices_in_pic_minus1] được suy ra bằng NumBricksInPic – 1. Chiều dài của phần tử cú pháp bottom_right_brick_idx_delta[i] là bottom_right_brick_idx_length_minus1 + 1 bit.

brick_idx_delta_sign_flag[i] bằng 1 chỉ báo là dấu dương cho bottom_right_brick_idx_delta[i]. sign_bottom_right_brick_idx_delta[i] bằng 0 chỉ báo dấu âm cho bottom_right_brick_idx_delta[i].

Tương hợp dòng bit yêu cầu rằng lát phải bao gồm số lượng của các ô hoàn chỉnh và chỉ chuỗi liên tiếp của các khối viên hoàn chỉnh của một ô.

Biến TopLeftBrickIdx[i], BottomRightBrickIdx[i], NumBricksInSlice[i] và BricksToSliceMap[j], xác định chỉ số khối viên của khối viên được đặt ở góc trên bên trái của lát thứ i, chỉ số khối viên của khối viên được đặt ở góc dưới bên phải của lát thứ i, số lượng khối viên trong lát thứ i và việc ánh xạ các khối viên đến các lát, được dẫn xuất như sau:

```
for(j = 0; i == 0 && j < NumBricksInPic; j++)
        BricksToSliceMap[j] = −1
NumBricksInSlice[i] = 0
BottomRightBrickIdx[i] = bottom_right_brick_idx_delta[i]] +((i == 0) ? 0 :
                (brick_idx_delta_sign_flag[i] ? BottomRightBrickIdx[i − 1] :
−BottomRightBrickIdx[i−1])
for(j = BottomRightBrickIdx[i]; j >= 0; j− −) {
        if(BrickColBd[j] <= BrickColBd[BottomRightBrickIdx[i]] &&        (7-43)
                BrickRowBd[j] <= BrickRowBd[BottomRightBrickIdx[i]] &&
                                                BricksToSliceMap[j] == −1) {
          TopLeftBrickIdx[i] = j
          NumBricksInSlice[i]++
          BricksToSliceMap[j] = i
        }
}
```

Ngữ nghĩa học tiêu đề lát chung

Nếu xuất hiện, giá trị của mỗi phần tử trong các phần tử cú pháp tiêu đề lát slice_pic_parameter_set_id, non_reference_picture_flag, colour_plane_id, slice_pic_order_cnt_lsb, recovery_poc_cnt, no_output_of_prior_pics_flag,

pic_output_flag, và slice_temporal_mvp_enabled_flag sẽ là giống nhau trong tất cả các tiêu đề lát của ảnh được mã hóa.

Biến CuQpDeltaVal, xác định hiệu số giữa tham số lượng tử độ sáng đối với CU chứa cu_qp_delta_abs và chế độ 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 thiết lập bằng 0.

slice_pic_parameter_set_id xác định giá trị của pps_pic_parameter_set_id đối với PPS đang được sử dụng. Giá trị slice_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 ảnh hiện tại sẽ lớn hơn hoặc bằng giá trị của TemporalId của PPS có pps_pic_parameter_set_id bằng slice_pic_parameter_set_id.

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

Nếu rect_slice_flag bằng 0, điều kiện sau áp dụng:

- Địa chỉ lát là ID khối viên như được xác định bởi phương trình (7-59).
- Chiều dài slice_address là Ceil(Log2 (NumBricksInPic)) bits.
- Giá trị slice_address phải nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút.

Ngược lại (rect_slice_flag bằng 1), điều kiện sau áp dụng:

- Địa chỉ lát là ID lát của lát.
- Chiều dài slice_address là signalled_slice_id_length_minus1 + 1 bit.
- Nếu signalled_slice_id_flag bằng 0, giá trị slice_address phải nằm trong khoảng từ 0 đến num_slices_in_pic_minus1, bao gồm hai giá trị đầu mút. Ngược lại, giá trị slice_address phải nằm trong khoảng từ 0 to $2^{(signalled\_slice\_id\_length\_minus1 + 1)} - 1$, bao gồm hai giá trị đầu mút.

Tương hợp dòng bit yêu cầu rằng thỏa mãn các ràng buộc sau:

- Giá trị slice_address sẽ không bằng giá trị slice_address của bất kỳ đơn vị NAL lát được mã hóa khác bất kỳ của cùng ảnh được mã hóa.

- Khi rect_slice_flag bằng 0, các lát của ảnh sẽ theo thứ tự tăng dần của các giá trị slice_address của nó.
- Các hình dạng của các lát của ảnh sẽ sao cho mỗi khối viên, khi được giải mã, sẽ có toàn bộ đường biên trái và toàn bộ đường biên trên bao gồm đường biên ảnh hoặc bao gồm các đường biên của (các) khối viên được giải mã trước đó.

num_bricks_in_slice_minus1, nếu xuất hiện, xác định số lượng khối viên trong lát trừ 1. Giá trị của num_bricks_in_slice_minus1 phải nằm trong khoảng từ 0 đến NumBricksInPic − 1, bao gồm hai giá trị đầu mút. Khi rect_slice_flag bằng 0 và single_brick_per_slice_flag bằng 1, giá trị của num_bricks_in_slice_minus1 được suy ra bằng 0. Khi single_brick_per_slice_flag bằng 1, giá trị của num_bricks_in_slice_minus1 được suy ra bằng 0.

Biến NumBricksInCurrSlice, xác định số lượng khối viên trong lát hiện tại, và SliceBrickIdx[i], xác định chỉ số khối viên của khối viên thứ i trong lát hiện tại, được dẫn xuất như sau:

```
if(rect_slice_flag) {
    sliceIdx = 0
    while(slice_address != slice_id[sliceIdx])
        sliceIdx++
    NumBricksInCurrSlice = NumBricksInSlice[sliceIdx]
    brickIdx = TopLeftBrickIdx[sliceIdx]
    for(bIdx = 0; brickIdx <= BottomRightBrickIdx[sliceIdx]; brickIdx++)        (7-92)
        if(BricksToSliceMap[brickIdx] == sliceIdx)
            SliceBrickIdx[bIdx++] = brickIdx
} else {
    NumBricksInCurrSlice = num_bricks_in_slice_minus1 + 1
    SliceBrickIdx[0] = slice_address
    for(i = 1; i < NumBricksInCurrSlice; i++)
        SliceBrickIdx[i] = SliceBrickIdx[i − 1] + 1
}
```

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

```
SubPicIdx = CtbToSubPicIdx[CtbAddrBsToRs[FirstCtbAddrBs[SliceBrickIdx[0]]]]
if(subpic_treated_as_pic_flag[SubPicIdx]) {
```

```
    SubPicLeftBoundaryPos =
SubPicLeft[SubPicIdx] * (subpic_grid_col_width_minus1 + 1) * 4
    SubPicRightBoundaryPos = (SubPicLeft[SubPicIdx] + SubPicWidth[SubPicIdx]) *
                (subpic_grid_col_width_minus1 + 1) * 4        (7-93)
    SubPicTopBoundaryPos =
SubPicTop[SubPicIdx] * (subpic_grid_row_height_minus1 + 1)* 4
    SubPicBotBoundaryPos = (SubPicTop[SubPicIdx] + SubPicHeight[SubPicIdx]) *
                (subpic_grid_row_height_minus1 + 1) * 4
}
```

3. Các ví dụ về các vấn dề kỹ thuật được giải quyết bằng các phương án thực hiện được bộc lộ

(1) Có một số thiết kế có thể vi phạm ràng buộc ảnh phụ.

A. TMVP trong các ứng viên được tạo afin có thể nạp MV trong ảnh cùng vị trí nằm ngoài khoảng của ảnh phụ hiện tại.

B. Khi dẫn xuất các građien trong luồng quang học hai hướng (Bi-Directional Optical Flow, BDOF) và luồng quang học tinh lọc dự báo (PROF), hai hàng mở rộng và hai cột mở rộng của các mẫu tham chiếu số nguyên được yêu cầu nạp. Các mẫu tham chiếu này có thể nằm ngoài khoảng của ảnh phụ hiện tại.

C. Khi dẫn xuất hệ số chia tỷ lệ dư sắc độ trong LMCS, mẫu độ sáng được tái tạo được truy nhập có thể nằm ngoài khoảng của ảnh phụ hiện tại.

D. Khối lân cận có thể nằm ngoài khoảng của ảnh phụ hiện tại, khi dẫn xuất chế độ dự báo trong độ sáng, các mẫu tham chiếu để dự báo trong, các mẫu tham chiếu cho CCLM, tính khả dụng khối lân cận đối với các ứng viên lân cận không gian để hợp nhất/AMVP/CIIP/IBC/LMCS, các QP, quá trình khởi tạo CABAC, dẫn xuất ctxInc nhờ sử dụng các phần tử cú pháp phía trên và bên trái, và ctxIncfor phần tử cú pháp mtt_split_cu_vertical_flag. Biểu diễn ảnh phụ có thể dẫn đến ảnh phụ với các CTU không hoàn chính. Các phân vùng CTU và quá trình tách CU có thể cần xem xét các CTU không hoàn chỉnh.

(2) Các phần tử cú pháp được báo hiệu liên quan đến ảnh phụ có thể lớn tùy ý, có thể gây ra vấn đề tràn.

(3) Việc biểu diễn các ảnh phụ có thể dẫn đến các ảnh phụ không chữ nhật.

(4) Hiện tại ảnh phụ và lưới ảnh phụ được định nghĩa theo các đơn vị của 4 mẫu. Và chiều dài của syntax element phụ thuộc vào chiều cao ảnh chia cho 4. Tuy nhiên, do pic_width_in_luma_samples và pic_height_in_luma_samples hiện tại sẽ là bội số nguyên của Max(8, MinCbSizeY), lưới ảnh phụ có thể cần được định nghĩa theo các đơn vị của 8 mẫu.

(5) Cú pháp SPS, pic_width_max_in_luma_samples và pic_height_max_in_luma_samples có thể cần bị giới hạn để không nhỏ hơn 8.

(6) Tương tác giữa lấy mẫu lại ảnh tham chiếu/tính khả mở và ảnh phụ không được xem xét trong thiết kế hiện tại.

(7) Trong lọc thời gian, các mẫu giữa các ảnh phụ khác nhau có thể được yêu cầu.

(8) Khi báo hiệu các lát, thông tin có thể được suy ra mà không báo hiệu trong một số trường hợp.

(9) Có thể là tất cả các lát được định nghĩa không thể bao phủ toàn bộ ảnh hoặc ảnh phụ.

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

Liệt kê chi tiết dưới đây cần được xem xét làm các ví dụ để giải thích các khái niệm chung. Các mục này không nên được hiểu theo nghĩa hẹp. Ngoài ra, các mục này có thể được kết hợp theo cách thức bất kỳ. Sau đây, bộ lọc thời gian được sử dụng để biểu diễn các bộ lọc yêu cầu các mẫu ở các ảnh khác. Max(x, y) trả về giá trị lớn hơn của x và y. Min(x, y) trả về giá trị nhỏ hơn trong x và y.

1. Vị trí (tên là vị trí RB) mà ở đó bộ dự báo MV thời gian được nạp trong ảnh để tạo các ứng viên chuyển động afin (chẳng hạn, ứng viên hợp nhất afin được tạo) phải nằm trong ảnh phụ được yêu cầu, giả sử tọa độ góc trên bên trái của ảnh phụ được yêu cầu là (xTL, yTL) và tọa độ dưới bên phải của ảnh phụ được yêu cầu là (xBR, yBR).
    a. Trong một ví dụ, ảnh phụ được yêu cầu là ảnh phụ bao phủ khối hiện tại.

b. Trong một ví dụ, nếu vị trí RB có tọa độ (x, y) nằm ngoài ảnh phụ được yêu cầu, bộ dự báo MV thời gian được xem là không khả dụng.
    i. Trong một ví dụ, vị trí RB nằm ngoài ảnh phụ được yêu cầu nếu x > xBR.
    ii. Trong một ví dụ, vị trí RB nằm ngoài ảnh phụ được yêu cầu nếu y > yBR.
    iii. Trong một ví dụ, vị trí RB nằm ngoài ảnh phụ được yêu cầu nếu x < xTL.
    iv. Trong một ví dụ, vị trí RB nằm ngoài ảnh phụ được yêu cầu nếu y < yTL.
c. Trong một ví dụ, vị trí RB, nếu nằm ngoài ảnh phụ được yêu cầu, RB thay thế được sử dụng.
    i. Theo cách khác, ngoài ra, vị trí thay thế phải nằm trong ảnh phụ được yêu cầu.
d. Trong một ví dụ, vị trí RB được rút gọn để nằm ảnh phụ được yêu cầu.
    i. Trong một ví dụ, x được rút gọn thành x = Min(x, xBR).
    ii. Trong một ví dụ, y được rút gọn thành y = Min(y, yBR).
    iii. Trong một ví dụ, x được rút gọn thành x = Max(x, xTL).
    iv. Trong một ví dụ, y được rút gọn thành y = Max(y, yTL).
e. Trong một ví dụ, vị trí RB có thể là vị trí dưới bên phải bên trong khối tương ứng của khối hiện tại trong ảnh cùng vị trí.
f. Phương pháp được đề xuất có thể được sử dụng trong các công cụ mã khác đòi hỏi truy nhập thông tin chuyển động từ ảnh khác với ảnh hiện tại.
g. Trong một ví dụ, liệu các phương pháp nêu trên có được áp dụng (chẳng hạn, vị trí RB phải nằm trong ảnh phụ được yêu cầu (chẳng hạn thực hiện như được nêu trong mục 1.a và/hoặc 1.b)) có thể phụ thuộc vào một hoặc nhiều phần tử cú pháp được báo hiệu trong VPS/DPS/SPS/PPS/APS/tiêu đề lát/tiêu đề nhóm ô.

Chẳng hạn, phần tử cú pháp có thể là subpic_treated_as_pic_flag[SubPicIdx], trong đó SubPicIdx là chỉ số ảnh phụ của ảnh phụ bao phủ khối hiện tại.

2. Vị trí (đặt tên là vị trí S) mà ở đó mẫu số nguyên được nạp trong tham chiếu không được sử dụng trong quá trình nội suy phải nằm trong ảnh phụ được yêu cầu, giả sử tọa độ góc trên bên trái của ảnh phụ được yêu cầu là (xTL, yTL) và tọa độ góc dưới bên phải của ảnh phụ được yêu cầu là (xBR, yBR).
    a. Trong một ví dụ, ảnh phụ được yêu cầu là ảnh phụ bao phủ khối hiện tại.
    b. Trong một ví dụ, nếu vị trí S có tọa độ (x, y) nằm ngoài ảnh phụ được yêu cầu, mẫu tham chiếu được xem là không khả dụng.
        i. Trong một ví dụ, vị trí S nằm ngoài ảnh phụ được yêu cầu nếu $x > xBR$.
        ii. Trong một ví dụ, vị trí S nằm ngoài ảnh phụ được yêu cầu nếu $y > yBR$.
        iii. Trong một ví dụ, vị trí S nằm ngoài ảnh phụ được yêu cầu nếu $x < xTL$.
        iv. Trong một ví dụ, vị trí S nằm ngoài ảnh phụ được yêu cầu nếu $y < yTL$.
    c. Trong một ví dụ, vị trí S được rút gọn để nằm trong ảnh phụ được yêu cầu.
        i. Trong một ví dụ, x được rút gọn thành $x = Min(x, xBR)$.
        ii. Trong một ví dụ, y được rút gọn thành $y = Min(y, yBR)$.
        iii. Trong một ví dụ, x được rút gọn thành $x = Max(x, xTL)$.
        iv. Trong một ví dụ, y được rút gọn thành $y = Max(y, yTL)$.
    d. Trong một ví dụ, liệu vị trí S phải nằm trong ảnh phụ được yêu cầu (chẳng hạn thực hiện như được nêu ở điểm 2.a và/hoặc 2.b) có thể phụ thuộc vào một hoặc nhiều phần tử cú pháp được báo hiệu trong VPS/DPS/SPS/PPS/APS/tiêu đề lát/tiêu đề nhóm ô. Chẳng hạn, phần tử cú pháp có thể là

subpic_treated_as_pic_flag[SubPicIdx], trong đó SubPicIdx là chỉ số ảnh phụ của ảnh phụ bao phủ khối hiện tại.

e. Trong một ví dụ, mẫu số nguyên được nạp được sử dụng để tạo các gradien trong BDOF và/hoặc PORF.

3. Vị trí (tên là vị trí R) ở đó giá trị mẫu độ sáng được tái tạo được nạp có thể nằm trong ảnh phụ được yêu cầu, giả sử tọa độ góc trên bên trái của ảnh phụ được yêu cầu là (xTL, yTL) và tọa độ góc dưới bên phải của ảnh phụ được yêu cầu là (xBR, yBR).
   a. Trong một ví dụ, ảnh phụ được yêu cầu là ảnh phụ bao phủ khối hiện tại.
   b. Trong một ví dụ, nếu vị trí R có tọa độ (x, y) nằm ngoài ảnh phụ được yêu cầu, mẫu tham chiếu được xem là không khả dụng.
      i. Trong một ví dụ, vị trí R nằm ngoài ảnh phụ được yêu cầu nếu $x > xBR$.
      ii. Trong một ví dụ, vị trí R nằm ngoài ảnh phụ được yêu cầu nếu $y > yBR$.
      iii. Trong một ví dụ, vị trí R nằm ngoài ảnh phụ được yêu cầu nếu $x < xTL$.
      iv. Trong một ví dụ, vị trí R nằm ngoài ảnh phụ được yêu cầu nếu $y < yTL$.
   c. Trong một ví dụ, vị trí R được rút gọn để nằm trong ảnh phụ được yêu cầu.
      i. Trong một ví dụ, x được rút gọn thành $x = Min(x, xBR)$.
      ii. Trong một ví dụ, y được rút gọn thành $y = Min(y, yBR)$.
      iii. Trong một ví dụ, x được rút gọn thành $x = Max(x, xTL)$.
      iv. Trong một ví dụ, y được rút gọn thành $y = Max(y, yTL)$.
   d. Trong một ví dụ, liệu vị trí R phải nằm trong ảnh phụ được yêu cầu (chẳng hạn thực hiện như được nêu trong mục 4.a và/hoặc 4.b) có thể phụ thuộc vào một hoặc nhiều phần tử cú pháp được báo hiệu in VPS/DPS/SPS/PPS/APS/tiêu đề lát/tiêu đề nhóm ô. Chẳng hạn, phần tử cú pháp có thể là

subpic_treated_as_pic_flag[SubPicIdx], trong đó SubPicIdx là chỉ số ảnh phụ của ảnh phụ bao phủ khối hiện tại.

e. Trong một ví dụ, mẫu độ sáng được nạp được sử dụng để dẫn xuất hệ số chia tỷ lệ đối với (các) thành phần sắc độ trong LMCS.

4. Vị trí (tên là vị trí N) mà ở đó phép kiểm tra đường biên ảnh để tách BT/TT/QT, dẫn xuất chiều sâu BT/TT/QT, và/hoặc cờ tách CU báo hiệu CU phải nằm trong ảnh phụ được yêu cầu, giả sử tọa độ góc trên bên trái của ảnh phụ được yêu cầu là (xTL, yTL) và tọa độ góc dưới bên phải của ảnh phụ được yêu cầu là (xBR, yBR).
    a. Trong một ví dụ, ảnh phụ được yêu cầu là ảnh phụ bao phủ khối hiện tại.
    b. Trong một ví dụ, nếu vị trí N có tọa độ (x, y) nằm ngoài ảnh phụ được yêu cầu, mẫu tham chiếu được xem là không khả dụng.
        i. Trong một ví dụ, vị trí N nằm ngoài ảnh phụ được yêu cầu nếu x > xBR.
        ii. Trong một ví dụ, vị trí N nằm ngoài ảnh phụ được yêu cầu nếu y > yBR.
        iii. Trong một ví dụ, vị trí N nằm ngoài ảnh phụ được yêu cầu nếu x < xTL.
        iv. Trong một ví dụ, vị trí N nằm ngoài ảnh phụ được yêu cầu nếu y < yTL.
    c. Trong một ví dụ, vị trí N được rút gọn để nằm trong ảnh phụ được yêu cầu.
        i. Trong một ví dụ, x được rút gọn thành x = Min(x, xBR).
        ii. Trong một ví dụ, y được rút gọn thành y = Min(y, yBR).
        iii. Trong một ví dụ, x được rút gọn thành x = Max(x, xTL).
    d. Trong một ví dụ, y được rút gọn thành y = Max(y, yTL).Trong một ví dụ, liệu vị trí N phải nằm trong ảnh phụ được yêu cầu (chẳng hạn thực hiện như được nêu trong 5.a và/hoặc 5.b) có thể phụ thuộc vào một hoặc nhiều phần tử cú pháp được báo hiệu trong VPS/DPS/SPS/PPS/APS/tiêu đề lát/tiêu đề nhóm ô. Chẳng

hạn, phần tử cú pháp có thể là subpic_treated_as_pic_flag[SubPicIdx], trong đó SubPicIdx là chỉ số ảnh phụ của ảnh phụ bao phủ khối hiện tại.

5. Bảng dự báo MV dựa trên lịch sử (HMVP) có thể được thiết lập lại trước khi giải mã ảnh phụ mới trong một ảnh.
    a. Trong một ví dụ, bảng HMVP được sử dụng để mã hóa IBC có thể được thiết lập lại
    b. Trong một ví dụ, bảng HMVP được sử dụng để mã hóa ngoài có thể được thiết lập lại
    c. Trong một ví dụ, bảng HMVP được sử dụng để mã hóa trong có thể được thiết lập lại
6. Các phần tử cú pháp ảnh phụ có thể được định nghĩa theo các đơn vị của N (chẳng hạn N = 8, 32, và v.v.) mẫu.
    a. Trong một ví dụ, chiều rộng của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của N mẫu.
    b. Trong một ví dụ, chiều cao của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của N mẫu.
    c. Trong một ví dụ, N được gán bằng chiều rộng và/hoặc chiều cao củaCTU.
7. Phần tử cú pháp của chiều rộng ảnh và chiều cao ảnh có thể bị giới hạn để không nhỏ hơn K (K>=8).
    a. Trong một ví dụ, chiều rộng ảnh có thể cần bị giới hạn để không nhỏ hơn 8.
    b. Trong một ví dụ, chiều cao ảnh có thể cần bị giới hạn để không nhỏ hơn 8.
8. Dòng bit tương hợp phải thỏa mãn rằng mã hóa ảnh phụ và biến đổi độ phân giải thích ứng (ARC)/biến đổi độ phân giải động (DRC)/ lấy mẫu lại ảnh tham chiếu (RPR) không được phép kích hoạt cho một đơn vị viđeo (chẳng hạn, chuỗi).
    a. Trong một ví dụ, việc báo hiệu kích hoạt mã hóa ảnh phụ có thể theo các điều kiện không cho phép ARC/DRC/RPR.

i. Trong một ví dụ, khi ảnh phụ được kích hoạt, chẳng hạn **subpics_present_flag** bằng 1, **pic_width_in_luma_samples** cho tất cả các ảnh mà SPS cho nó đang hoạt động bằng max_width_in_luma_samples.

b. Theo cách khác, mã hóa ảnh phụ và ARC/DRC/RPR có thể đều được kích hoạt cho một đơn vị viđeo (chẳng hạn, chuỗi).

i. Trong một ví dụ, dòng bit tương hợp sẽ thỏa mãn rằng ảnh phụ bị giảm kích thước mẫu do RC/DRC/RPR vẫn sẽ có dạng K CTU theo chiều rộng và M CTU theo chiều cao trong đó K và M đều là các số nguyên.

ii. Trong một ví dụ, dòng bit tương hợp sẽ thỏa mãn rằng đối với các ảnh phụ không được đặt ở các đường biên ảnh (chẳng hạn, đường biên phải và/hoặc đường biên dưới), ảnh phụ bị giảm kích thước mẫu do RC/DRC/RPR vẫn phải ở dạng K CTU theo chiều rộng và M CTU theo chiều cao trong đó K và M đều là các số nguyên.

iii. Trong một ví dụ, các kích thước CTU có thể được thay đổi thích ứng dựa trên độ phân giải của ảnh.

1) Trong một ví dụ, kích thước CTU lớn nhất có thể được báo hiệu trong SPS. Đối với mỗi ảnh có độ phân giải nhỏ hơn, kích thước CTU có thể được thay đổi tương ứng dựa trên độ phân giải được giảm.

2) Trong một ví dụ, kích thước CTU có thể được báo hiệu trong SPS và PPS, và/hoặc mức ảnh phụ.

9. Phần tử cú pháp subpic_grid_col_width_minus1 và subpic_grid_row_height_minus1 có thể bị ràng buộc.

a. Trong một ví dụ, subpic_grid_col_width_minus1 phải không lớn hơn (hoặc phải nhỏ hơn) T1.

b. Trong một ví dụ, subpic_grid_row_height_minus1 phải không lớn hơn (hoặc phải nhỏ hơn) than T2.

c. Trong một ví dụ, trong dòng bit tương hợp, subpic_grid_col_width_minus1 và/hoặc subpic_grid_row_height_minus1 must follow ràng buộc chẳng hạn đề mục 3.a hoặc 3.b.

d. Trong một ví dụ, T1 trong 3.a và/hoặc T2 trong 3.b có thể phụ thuộc vào các tiểu sử/các mức/các tầng của chuẩn mã hóa viđeo.

e. Trong một ví dụ, T1 trong 3.a có thể phụ thuộc vào chiều rộng ảnh.

i. Chẳng hạn, T1 bằng pic_width_max_in_luma_samples/4 hoặc pic_width_max_in_luma_samples/4+Off. Off có thể bằng 1, 2, -1, -2, v.v..

f. Trong một ví dụ, T2 trong 3.b có thể phụ thuộc vào chiều rộng ảnh.

i. Chẳng hạn, T2 bằng pic_height_max_in_luma_samples/4 hoặc pic_height_max_in_luma_samples /4-1+Off. Off có thể bằng 1, 2, -1, -2, v.v..

10. Giới hạn rằng đường biên giữa hai ảnh phụ phải là đường biên giữa hai CTU.

a. Nói cách khác, CTU không thể được bao phủ bởi nhiều hơn một ảnh phụ.

b. Trong một ví dụ, đơn vị của subpic_grid_col_width_minus1 có thể là chiều rộng CTU (chẳng hạn 32, 64, 128), thay vì 4 như trong VVC. Chiều rộng lưới ảnh phụ cần bằng (subpic_grid_col_width_minus1+1) * chiều rộng CTU.

c. Trong một ví dụ, đơn vị của subpic_grid_col_height_minus1 có thể là chiều cao CTU (chẳng hạn 32, 64, 128), thay vì 4 như trong VVC. Chiều cao lưới ảnh phụ nên bằng (subpic_grid_col_height_minus1+1) * chiều cao CTU.

d. Trong một ví dụ, trong dòng bit tương hợp, ràng buộc phải được thỏa mãn nếu cách thức của ảnh phụ được áp dụng.

11. Ràng buộc rằng chiều rộng của ảnh phụ phải là hình vuông.
    a. Trong một ví dụ, trong dòng bit tương hợp, ràng buộc phải được thỏa mãn nếu cách thức của ảnh phụ được áp dụng.
    b. ảnh phụ có thể chỉ chứa các lát chữ nhật. Chẳng hạn, trong dòng bit tương hợp, ràng buộc phải được thỏa mãn nếu cách thức của ảnh phụ được áp dụng.
12. Ràng buộc rằng hai ảnh phụ không thể bị chồng lấn.
    a. Trong một ví dụ, trong dòng bit tương hợp, ràng buộc phải được thỏa mãn nếu cách thức của ảnh phụ được áp dụng.
    b. Theo cách khác, hai ảnh phụ có thể được chồng lấn với nhau.
13. Ràng buộc rằng bất kỳ vị trí trong ảnh phải được bao phủ bởi một và chỉ một ảnh phụ.
    a. Trong một ví dụ, trong dòng bit tương hợp, ràng buộc phải được thỏa mãn nếu cách thức của ảnh phụ được áp dụng.
    b. Theo cách khác, một mẫu có thể không thuộc bất kỳ ảnh phụ.
    c. Theo cách khác, một mẫu có thể thuộc nhiều hơn một ảnh phụ.
14. Có thể yêu cầu rằng các ảnh phụ được định nghĩa trong SPS được ánh xạ đến mỗi độ phân giải xuất hiện trong cùng chuỗi nên tuân theo vị trí và/hoặc kích thước bị ràng buộc nêu trên.
    a. Trong một ví dụ, chiều rộng và chiều cao của ảnh phụ được định nghĩa trong SPS được ánh xạ đến độ phân giải xuất hiện trong cùng chuỗi, nên gấp nhiều lần số nguyên của N (chẳng hạn 8, 16, 32) mẫu độ sáng.
    b. Trong một ví dụ, các ảnh phụ có thể được định nghĩa cho lớp cụ thể và có thể được ánh xạ đến các lớp khác.
        i. Chẳng hạn, các ảnh phụ có thể được định nghĩa đối với lớp có có độ phân giải cao nhất trong chuỗi.
        ii. Chẳng hạn, các ảnh phụ có thể được định nghĩa đối với lớp có độ phân giải thấp nhất trong chuỗi.

iii. Các ảnh phụ được định nghĩa cho lớp nào có thể được báo hiệu trong SPS/VPS/PPS/tiêu đề lát.

c. Trong một ví dụ, khi các ảnh phụ và các độ phân giải khác nhau đều được áp dụng, tất cả các độ phân giải (chẳng hạn, chiều rộng và/hoặc chiều cao) có thể là bội số nguyên có độ phân giải cụ thể.

d. Trong một ví dụ, chiều rộng và/hoặc chiều cao của ảnh phụ được định nghĩa trong SPS có thể là nhiều lần số nguyên (chẳng hạn, M) có kích thước CTU.

e. Theo cách khác, các ảnh phụ và các độ phân giải khác nhau trong chuỗi có thể không được phép đồng thời.

15. Các ảnh phụ có thể chỉ áp dụng cho (các) lớp cụ thể

a. Trong một ví dụ, các ảnh phụ được định nghĩa trong SPS có thể chỉ áp dụng cho lớp có độ phân giải cao nhất trong chuỗi.

b. Trong một ví dụ, các ảnh phụ được định nghĩa trong SPS có thể chỉ áp dụng cho lớp có ID thời gian thấp nhất trong chuỗi.

c. (Các) lớp nào mà các ảnh phụ có thể được áp dụng cho có thể được chỉ báo bởi một hoặc nhiều phần tử cú pháp trong SPS/VPS/PPS.

d. (Các) lớp nào mà ảnh phụ không thể được áp dụng cho có thể được chỉ báo bởi một hoặc nhiều phần tử cú pháp trong SPS/VPS/PPS.

16. Trong một ví dụ, vị trí và/hoặc các chiều của ảnh phụ có thể được báo hiệu mà không sử dụng subpic_grid_idx.

a. Trong một ví dụ, vị trí trên bên trái của ảnh phụ có thể được báo hiệu.

b. Trong một ví dụ, vị trí dưới bên phải của ảnh phụ có thể được báo hiệu.

c. Trong một ví dụ, chiều rộng của ảnh phụ có thể được báo hiệu.

d. Trong một ví dụ, chiều cao của ảnh phụ có thể được báo hiệu.

17. Đối với bộ lọc thời gian, khi thực hiện lọc thời gian mẫu, chỉ các mẫu trong cùng ảnh phụ mà có mẫu hiện tại có thể được sử dụng. Các mẫu được yêu cầu có thể nằm trong cùng ảnh mà có mẫu hiện tại hoặc trong các ảnh khác.
18. Trong một ví dụ, liệu và/hoặc cách thức áp dụng phương pháp phân vùng (chẳng hạn QT, BT ngang, BT dọc, TT ngang, TT dọc, hoặc không tách, v.v.) có thể phụ thuộc vào liệu khối hiện tại (hoặc phân vùng) giữa một hoặc nhiều đường biên của ảnh phụ.
    a. Trong một ví dụ, phương pháp xử lý đường biên ảnh để phân vùng trong VVC có thể cũng được áp dụng khi đường biên ảnh được thay thế bằng đường biên ảnh phụ.
    b. Trong một ví dụ, liệu có phân tách phần tử cú pháp (chẳng hạn, cờ) biểu diễn phương pháp phân vùng (chẳng hạn QT, BT ngang, BT dọc, TT ngang, TT dọc, hoặc không tách, v.v.) có thể phụ thuộc vào liệu khối hiện tại (hoặc phân vùng) giao với một hoặc nhiều đường biên của ảnh phụ.
19. Thay vì tách một ảnh thành nhiều ảnh phụ với tạo mã độc lập của mỗi ảnh phụ, đề xuất tách ảnh thành ít nhất hai tập hợp vùng con, với tập hợp thứ nhất bao gồm nhiều ảnh phụ và tập hợp thứ hai bao gồm tất cả các mẫu còn lại.
    a. Trong một ví dụ, mẫu trong tập hợp thứ hai không nằm trong bất kỳ ảnh phụ nào.
    b. Theo cách khác, ngoài ra, tập hợp thứ hai có thể được mã hóa/giải mã dựa trên thông tin của tập hợp thứ nhất.
    c. Trong một ví dụ, giá trị mặc định có thể được sử dụng để đánh dấu liệu mẫu/MxK vùng con có thuộc tập hợp thứ hai hay không.
        i. Trong một ví dụ, giá trị mặc định có thể được gán bằng (**max_subpics_minus1** + K) trong đó K là số nguyên lớn hơn 1.

ii. Giá trị mặc định có thể được gán cho subpic_grid_idx[i][j] để chỉ báo rằng lưới thuộc tập hợp thứ hai.

20. Đề xuất rằng phần tử cú pháp subpic_grid_idx[i][j] không thể lớn hơn max_subpics_minus1.
    a. Chẳng hạn, ràng buộc rằng trong dòng bit tương hợp, subpic_grid_idx[i][j] không thể lớn hơn max_subpics_minus1.
    b. Chẳng hạn, từ mã để mã hóa subpic_grid_idx[i][j] không thể lớn hơn max_subpics_minus1.
21. Đề xuất rằng, số nguyên bất kỳ từ 0 đến max_subpics_minus1 phải bằng ít nhất một subpic_grid_idx[i][j].
22. Bộ đệm ảo IBC có thể được thiết lập lại trước khi giải mã ảnh phụ mới trong một ảnh.
    a. Trong một ví dụ, tất cả các mẫu trong bộ đệm ảo IBC có thể được thiết lập lại bằng -1.
23. Danh sách mục bảng màu có thể được thiết lập lại trước khi giải mã ảnh phụ mới trong một ảnh.
    a. Trong một ví dụ, PredictorPaletteSize có thể được gán bằng 0 trước khi giải mã ảnh phụ mới trong một ảnh.
24. Liệu có báo hiệu thông tin của các lát (chẳng hạn số lượng lát và/hoặc các khoảng của các lát) có thể phụ thuộc vào số lượng ô và/hoặc số lượng khối viên.
    a. Trong một ví dụ, nếu số lượng khối viên trong ảnh bằng 1, num_slices_in_pic_minus1 không được báo hiệu và được suy ra bằng 0.
    b. Trong một ví dụ, nếu số lượng khối viên trong ảnh bằng 1, thông tin của các lát (chẳng hạn, số lượng lát và/hoặc các khoảng của các lát) có thể không được báo hiệu.
    c. Trong một ví dụ, nếu số lượng khối viên trong ảnh bằng 1, số lượng lát có thể được suy ra bằng 1. Và lát bao phủ toàn bộ ảnh. Trong một ví dụ, nếu số lượng khối viên trong ảnh bằng 1,

single_brick_per_slice_flag không được báo hiệu và được suy ra bằng một.

i. Theo cách khác, nếu số lượng khối viên trong ảnh bằng 1, single_brick_per_slice_flag phải bằng 1.

d. Thiết kế cú pháp lấy làm ví dụ dưới đây:

pic_parameter_set_rbsp() {	Mô tả
...	
if(NumBricksInPic > 1){	
single_brick_per_slice_flag	u(1)
if(!single_brick_per_slice_flag)	
rect_slice_flag	u(1)
if(rect_slice_flag && !single_brick_per_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
bottom_right_brick_idx_length_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
bottom_right_brick_idx_delta[i]	u(v)
brick_idx_delta_sign_flag[i]	u(1)
}	
}	
}	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	
...	

25. Liệu báo hiệu slice_address có thể được tách khỏi việc liệu các lát được báo hiệu là các hình chữ nhật (chẳng hạn liệu rect_slice_flag bằng 0 hoặc 1).

a. Thiết kế cú pháp lấy làm ví dụ dưới đây:

if([[rect_slice_flag ||]] NumBricksInPic > 1)	
slice_address	u(v)

26. Liệu báo hiệu slice_address có thể phụ thuộc vào số lượng lát khi các lát được báo hiệu là các hình chữ nhật.

if((rect_slice_flag && num_slices_in_pic_minus1> 0) || (!rect_slice_flag && NumBricksInPic > 1))	
slice_address	u(v)

27. Liệu báo hiệu num_bricks_in_slice_minus1 có thể phụ thuộc vào slice_address và/hoặc số lượng khối viên trong ảnh.

a. Thiết kế cú pháp lấy làm ví dụ dưới đây:

if(!rect_slice_flag && !single_brick_per_slice_flag ***&& slice_address < NumBricksInPic -1***)	
num_bricks_in_slice_minus1	ue(v)

28. Liệu báo hiệu loop_filter_across_bricks_enabled_flag có thể phụ thuộc vào số lượng ô và/hoặc số lượng khối viên.

a. Trong một ví dụ, loop_filter_across_bricks_enabled_flag không được báo hiệu nếu số lượng khối viên nhỏ hơn 2.

b. Thiết kế cú pháp lấy làm ví dụ dưới đây:

pic_parameter_set_rbsp() {	Mô tả
...	
if(NumBricksInPic > 1)	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
...	

29. Tương hợp dòng bit yêu cầu rằng tất cả các lát của ảnh phải bao phủ toàn bộ ảnh.

    a. Yêu cầu này phải được thỏa mãn khi các lát được báo hiệu là các hình chữ nhật (chẳng hạn, rect_slice_flag bằng 1).

30. Tương hợp dòng bit yêu cầu rằng tất cả các lát của ảnh phụ phải bao phủ toàn bộ ảnh phụ.

    a. Yêu cầu này phải được thỏa mãn khi các lát được báo hiệu là các hình chữ nhật (chẳng hạn, rect_slice_flag bằng 1).

31. Tương hợp dòng bit yêu cầu rằng lát không thể được chồng lấn với nhiều hơn một ảnh phụ.

32. Tương hợp dòng bit yêu cầu rằng ô không thể được chồng lấn với nhiều hơn một ảnh phụ.

33. Tương hợp dòng bit yêu cầu rằng khối viên không thể được chồng lấn với nhiều hơn một ảnh phụ.

    Trong phần mô tả sau, khối đơn vị cơ bản (BUB) có các chiều CW × CH là vùng chữ nhật. Chẳng hạn, BUB có thể là khối cây mã (CTB).

34. Trong một ví dụ, số lượng ảnh phụ (được ký hiệu là N) có thể được báo hiệu.

    a. Tương hợp dòng bit yêu cầu rằng có ít nhất hai ảnh phụ trong ảnh nếu các ảnh phụ được sử dụng (chẳng hạn, subpics_present_flag bằng 1).

    b. Theo cách khác, N trừ d (tức là, N-d) có thể được báo hiệu, trong đó d là số nguyên chẳng hạn 0, 1, hoặc 2.

c. Chẳng hạn, N-d có thể được mã hóa với mã hóa chiều dài cố định, chẳng hạn, u(x).
    i. Trong một ví dụ, x có thể là số cố định, chẳng hạn 8.
    ii. Trong một ví dụ, x hoặc x-dx có thể được báo hiệu trước khi N-d được báo hiệu, trong đó dx là số nguyên chẳng hạn 0, 1 hoặc 2. x được báo hiệu có thể không lớn hơn giá trị lớn nhất trong dòng bit tương hợp.
    iii. Trong một ví dụ, x có thể được dẫn xuất ngay lập tức.
        1) Chẳng hạn, x có thể được dẫn xuất dưới dạng hàm của tổng số lượng (được ký hiệu là M) BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..
        2) M có thể được dẫn xuất như là M = Ceiling(W/CW) × Ceiling(H/ CH), trong đó W và H là chiều rộng và chiều cao của ảnh, và CW và CH là chiều rộng và chiều cao của BUB.

d. Chẳng hạn, N-d có thể được mã hóa với mã đơn phân hoặc mã đơn phân rút gọn.

e. Trong một ví dụ, giá trị lớn nhất được phép của N-d có thể là số cố định.
    i. Theo cách khác, giá trị lớn nhất được phép của N-d có thể được dẫn xuất dưới dạng hàm tổng số lượng (được ký hiệu là M) BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..

35. Trong một ví dụ, ảnh phụ có thể được báo hiệu bằng các chỉ báo của một hoặc nhiều vị trí được chọn (chẳng hạn, vị trí trên bên trái/trên bên phải/dưới bên trái/dưới bên phải) và/hoặc chiều rộng và/hoặc chiều cao của nó.

a. Trong một ví dụ, vị trí trên bên trái của ảnh phụ có thể được báo hiệu theo độ hạt của BUB với các kích thước CW × CH.

i. Chẳng hạn, chỉ số cột (được ký hiệu là Col) liên quan đến các BUB của BUB trên bên trái của ảnh phụ có thể được báo hiệu.

1) Chẳng hạn, Col - d có thể được báo hiệu, trong đó d là số nguyên chẳng hạn 0, 1, hoặc 2.

a) Theo cách khác, d có thể bằng Col của ảnh phụ được mã hóa trước đó, được cộng vào d1, trong đó d1 là số nguyên chẳng hạn -1,0, hoặc 1.

b) Dấu của Col -d có thể được báo hiệu.

ii. Chẳng hạn, chỉ số hàng (được ký hiệu là Row) liên quan đến BUB của BUB trên bên trái của ảnh phụ có thể được báo hiệu.

1) Chẳng hạn, Row - d có thể được báo hiệu, trong đó d là số nguyên chẳng hạn 0, 1, hoặc 2.

a) Theo cách khác, d có thể bằng hàng ảnh phụ được mã hóa trước đó, được cộng với d1, trong đó d1 là số nguyên chẳng hạn -1,0, hoặc 1.

b) Dấu của Row -d có thể được báo hiệu.

iii. Chỉ số hàng/cột (được ký hiệu là Row) nêu trên có thể được biểu diễn theo đơn vị CTB, chẳng hạn, tọa độ x hoặc y so với vị trí trên bên trái của ảnh có thể được chia cho kích thước CTB và được báo hiệu.

iv. Trong một ví dụ, liệu báo hiệu vị trí của ảnh phụ có thể phụ thuộc vào chỉ số ảnh phụ.

1) Trong một ví dụ, đối với ảnh phụ thứ nhất trong ảnh, vị trí trên bên trái có thể không được báo hiệu.

a) Theo cách khác, ngoài ra, vị trí trên bên trái có thể được suy ra, chẳng hạn, bằng (0, 0).

2) Trong một ví dụ, đối với ảnh phụ cuối cùng trong ảnh, vị trí trên bên trái có thể không được báo hiệu.

a) Vị trí trên bên trái có thể được suy ra tùy thuộc thông tin của các ảnh phụ được báo hiệu trước đó.

b. Trong một ví dụ, các chỉ báo của chiều rộng/chiều cao/ vị trí được chọn của ảnh phụ có thể được báo hiệu với đơn phân rút gọn/nhị phân rút gọn/đơn phân/chiều dài cố định/mã hóa EG thứ K (chẳng hạn, K=0, 1, 2, 3).

c. Trong một ví dụ, chiều rộng của ảnh phụ có thể được báo hiệu theo độ hạt của BUB với các kích thước CW × CH.

i. Chẳng hạn, số lượng cột của BUB trong ảnh phụ (được ký hiệu là W) có thể được báo hiệu.

ii. Chẳng hạn, W-d có thể được báo hiệu, trong đó d là số nguyên chẳng hạn 0, 1, hoặc 2.

1) Theo cách khác, d có thể bằng W của ảnh phụ được mã hóa trước đó, được cộng với d1, trong đó d1 là số nguyên chẳng hạn -1,0, hoặc 1.

2) Dấu của W-d có thể được báo hiệu.

d. Trong một ví dụ, chiều cao của ảnh phụ có thể được báo hiệu theo độ hạt của BUB với các kích thước CW × CH.

i. Chẳng hạn, số lượng hàng của các BUB trong ảnh phụ (được ký hiệu là H) có thể được báo hiệu.

ii. Chẳng hạn, H-d có thể được báo hiệu, trong đó d là số nguyên chẳng hạn 0, 1, hoặc 2.

1) Theo cách khác, d có thể bằng H của ảnh phụ được mã hóa trước đó, được cộng với d1, trong đó d1 là số nguyên chẳng hạn -1,0, hoặc 1.

2) Dấu của H-d có thể được báo hiệu.

e. Trong một ví dụ, Col-d có thể được mã hóa với mã hóa chiều dài cố định, chẳng hạn u(x).
    i. Trong một ví dụ, x có thể là số cố định chẳng hạn 8.
    ii. Trong một ví dụ, x hoặc x-dx có thể được báo hiệu trước khi Col-d được báo hiệu, trong đó dx là số nguyên chẳng hạn 0, 1 hoặc 2. x được báo hiệu có thể không lớn hơn giá trị lớn nhất trong dòng bit tương hợp.
    iii. Trong một ví dụ, x có thể được dẫn xuất ngay lập tức.
        1) Chẳng hạn, x có thể được dẫn xuất dưới dạng hàm của tổng số (được ký hiệu là M) cột BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..
        2) M có thể được dẫn xuất dưới dạng M = Ceiling(W/CW), trong đó W là chiều rộng của ảnh, và CW là chiều rộng của BUB.

f. Trong một ví dụ, Row-d có thể được mã hóa với mã hóa chiều dài cố định chẳng hạn u(x).
    i. Trong một ví dụ, x có thể là số cố định chẳng hạn 8.
    ii. Trong một ví dụ, x hoặc x-dx có thể được báo hiệu trước khi Row-d được báo hiệu, trong đó dx là số nguyên chẳng hạn 0, 1 hoặc 2. x được báo hiệu có thể không lớn hơn giá trị lớn nhất trong dòng bit tương hợp.
    iii. Trong một ví dụ, x có thể được dẫn xuất ngay lập tức.
        1) Chẳng hạn, x có thể được dẫn xuất dưới dạng hàm của tổng số (được ký hiệu là M) hàng BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..

2) M có thể được dẫn xuất dưới dạng M = Ceiling(H/CH), trong đó H là chiều cao của ảnh, và CH là chiều cao của BUB.

g. Trong một ví dụ, W-d có thể được mã hóa với mã hóa chiều dài cố định chẳng hạn u(x).

i. Trong một ví dụ, x có thể là số cố định chẳng hạn 8.

ii. Trong một ví dụ, x hoặc x-dx có thể được báo hiệu trước khi W-d được báo hiệu, trong đó dx là số nguyên chẳng hạn 0, 1 hoặc 2. x được báo hiệu có thể không lớn hơn giá trị lớn nhất trong dòng bit tương hợp.

iii. Trong một ví dụ, x có thể được dẫn xuất ngay lập tức.

1) Chẳng hạn, x có thể được dẫn xuất dưới dạng hàm của tổng số lượng (được ký hiệu là M) cột BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..

2) M có thể được dẫn xuất dưới dạng M = Ceiling(W/CW), trong đó W là chiều rộng của ảnh, và CW là chiều rộng của BUB.

h. Trong một ví dụ, H-d có thể được mã hóa với mã hóa chiều dài cố định chẳng hạn u(x).

i. Trong một ví dụ, x có thể là số cố định chẳng hạn 8.

ii. Trong một ví dụ, x hoặc x-dx có thể được báo hiệu trước khi H-d được báo hiệu, trong đó dx là số nguyên chẳng hạn 0, 1 hoặc 2. x được báo hiệu có thể không lớn hơn giá trị lớn nhất trong dòng bit tương hợp.

iii. Trong một ví dụ, x có thể được dẫn xuất ngay lập tức.

1) Chẳng hạn, x có thể được dẫn xuất dưới dạng hàm của tổng số (được ký hiệu là M) hàng BUB trong ảnh. Chẳng hạn, x = Ceil(log2(M+d0))+d1, trong

đó d0 và d1 là hai số nguyên, chẳng hạn -2, -1, 0, 1, 2, v.v..

2) M có thể được dẫn xuất dưới dạng M = Ceiling(H/CH), trong đó H là chiều cao của ảnh, và CH là chiều cao của BUB.

i. Col-d và/hoặc Row-d có thể được báo hiệu cho tất cả các ảnh phụ.

i. Theo cách khác, Col-d và/hoặc Row-d có thể không được báo hiệu cho tất cả các ảnh phụ.

1) Col-d và/hoặc Row-d có thể không được báo hiệu nếu số lượng ảnh phụ nhỏ hơn 2. (bằng 1).

2) Chẳng hạn, Col-d và/hoặc Row-d có thể không được báo hiệu cho ảnh phụ thứ nhất (chẳng hạn, với chỉ số ảnh phụ (hoặc ID ảnh phụ) bằng 0).

a) Khi chúng không được báo hiệu, chúng có thể được suy ra bằng 0.

3) Chẳng hạn, Col-d và/hoặc Row-d có thể không được báo hiệu cho ảnh phụ cuối cùng (chẳng hạn, với chỉ số ảnh phụ (hoặc ID ảnh phụ) bằng NumSubPics-1).

a) Khi chúng không được báo hiệu, chúng có thể được suy ra tùy thuộc các vị trí và các kích thước của các ảnh phụ đã được báo hiệu.

j. W-d và/hoặc H-d có thể được báo hiệu cho tất cả các ảnh phụ.

i. Theo cách khác, W-d và/hoặc H-d có thể không được báo hiệu cho tất cả các ảnh phụ.

1) W-d và/hoặc H-d có thể không được báo hiệu nếu số lượng của các ảnh phụ nhỏ hơn 2. (bằng 1).

2) Chẳng hạn, W-d và/hoặc H-d có thể không được báo hiệu cho ảnh phụ cuối cùng (chẳng hạn, với

chỉ số ảnh phụ (hoặc ID ảnh phụ) bằng NumSubPics-1).

a) Khi chúng không được báo hiệu, chúng có thể được suy ra tùy thuộc các vị trí và các kích thước của các ảnh phụ đã được báo hiệu.

k. Trong các đề mục nêu trên, BUB có thể là CTB.

36. Trong một ví dụ, thông tin của các ảnh phụ cần được báo hiệu sau khi thông tin có kích thước CTB (chẳng hạn, log2_ctu_size_minus5) đã được báo hiệu.

37. subpic_treated_as_pic_flag[i] có thể không được báo hiệu đối với mỗi các ảnh phụ. Thay vào đó, một subpic_treated_as_pic_flag được báo hiệu để điều khiển liệu ảnh phụ được xử lý dưới dạng ảnh cho tất cả các ảnh phụ.

38. loop_filter_across_subpic_enabled_flag [i] có thể không được báo hiệu đối với mỗi các ảnh phụ. Thay vào đó, một loop_filter_across_subpic_enabled_flag được báo hiệu để điều khiển liệu các bộ lọc vòng có thể được áp dụng giữa các ảnh phụ cho tất cả các ảnh phụ.

39. subpic_treated_as_pic_flag[i] (subpic_treated_as_pic_flag) và/hoặc loop_filter_across_subpic_enabled_flag[i] (loop_filter_across_subpic_enabled_flag) có thể được báo hiệu có điều kiện.

a. Trong một ví dụ, subpic_treated_as_pic_flag[i] và/hoặc loop_filter_across_subpic_enabled_flag[i] có thể không được báo hiệu nếu số lượng của các ảnh phụ nhỏ hơn 2. (bằng 1).

40. RPR có thể được áp dụng khi các ảnh phụ được sử dụng.

a. Trong một ví dụ, hệ số chia tỷ lệ trong RPR có thể bị ràng buộc là tập hợp giới hạn khi các ảnh phụ được sử dụng, chẳng hạn {1:1, 1:2 và/hoặc 2:1}, hoặc {1:1, 1:2 và/hoặc 2:1, 1:4 và/hoặc 4:1 }, {1:1, 1:2 và/hoặc 2:1, 1:4 và/hoặc 4:1, 1:8 và/hoặc 8:1}.

b. Trong một ví dụ, kích thước CTB của ảnh A và kích thước CTB của ảnh B có thể là khác nhau nếu độ phân giải của ảnh A và ảnh B là khác nhau.

c. Trong một ví dụ, giả sử ảnh phụ SA có các kích thước SAW×SAH nằm trong ảnh A và ảnh phụ SB có các kích thước SBW×SBH nằm trong ảnh B, SA tương ứng với SB, và các hệ số chia tỷ lệ giữa ảnh A và ảnh B là Rw và Rh dọc theo các hướng ngang và dọc, sau đó.

i. SAW/SBW hoặc SBW/SAW phải bằng Rw.

ii. SAH/SBH hoặc SBH/SAH phải bằng Rh.

41. Khi các ảnh phụ được sử dụng (chẳng hạn, sub_pics_present_flag là true), chỉ số ảnh phụ (hoặc ID ảnh phụ) có thể được báo hiệu trong tiêu đề lát, và địa chỉ lát bị gián đoạn dưới dạng địa chỉ trong ảnh phụ thay vì toàn bộ ảnh.

5. Các phương án thực hiện

Dưới đây các phương án thực hiện, các đoạn văn bản mới được thêm vào được in nghiêng đậm và các đoạn văn bản bị xóa được đánh dấu bằng "[[]]".

5.1 Phương án thực hiện 1: ràng buộc ảnh phụ trên các ứng viên hợp nhất được tạo afin

8.5.5.6 Quá trình dẫn xuất đối với các ứng viên hợp nhất MV điểm điều khiển afin được tạo

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của khối mã độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- hai biến cbWidth và cbHeight xác định chiều rộng và chiều cao của khối mã độ sáng hiện tại,
- các cờ tính khả dụng $availableA_0$, $availableA_1$, $availableA_2$, $availableB_0$, $availableB_1$, $availableB_2$, $availableB_3$,
- các vị trí mẫu $(xNbA_0, yNbA_0)$, $(xNbA_1, yNbA_1)$, $(xNbA_2, yNbA_2)$, $(xNbB_0, yNbB_0)$, $(xNbB_1, yNbB_1)$, $(xNbB_2, yNbB_2)$ và $(xNbB_3, yNbB_3)$.

Đầu ra của quá trình này là:

- cờ tính khả dụng của các ứng viên hợp nhất CPMV afin được tạo availableFlagConstK, với K = 1..6,
- các chỉ số tham chiếu refIdxLXConstK, với K = 1..6, X bằng 0 hoặc 1,
- các cờ sử dụng danh sách dự báo predFlagLXConstK, với K = 1..6, X bằng 0 hoặc 1,
- các chỉ số mô hình chuyển động afin motionModelIdcConstK, với K = 1..6,
- các chỉ số trọng số dự báo kép bcwIdxConstK, với K = 1..6,
- các CPMV afin được tạo cpMvLXConstK[cpIdx] với cpIdx = 0..2, K = 1..6 và X bằng 0 hoặc 1.

...

CPMV (dưới bên phải cùng vị trí) thứ tư cpMvLXCorner[3], chỉ số tham chiếu refIdxLXCorner[3], cờ sử dụng danh sách dự báo predFlagLXCorner[3] và cờ tính khả dụng availableFlagCorner[3] với X bằng 0 và 1 được dẫn xuất như sau:

- Các chỉ số tham chiếu đối với ứng viên hợp nhất thời gian, refIdxLXCorner[3], với X bằng 0 hoặc 1, được gán bằng 0.
- Các biến mvLXCol và availableFlagLXCol, với X bằng 0 hoặc 1, được dẫn xuất như sau:
  - Nếu slice_temporal_mvp_enabled_flag bằng 0, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.
  - Ngược lại (slice_temporal_mvp_enabled_flag bằng 1), điều kiện sau áp dụng:

    xColBr = xCb + cbWidth (8-601)

    yColBr = yCb + cbHeight (8-602)

rightBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos : pic_width_in_luma_samples − 1

botBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos : pic_height_in_luma_samples − 1

  - Nếu yCb >> CtbLog2SizeY bằng yColBr >> CtbLog2SizeY, ***yColBr nhỏ hơn hoặc bằng botBoundaryPos and xColBr nhỏ hơn hoặc bằng rightBoundaryPos, điều kiện sau áp dụng***:

- Biến colCb xác định khối mã độ sáng bao phủ vị trí được chỉnh sửa được tạo bởi ((xColBr >> 3) << 3, (yColBr >> 3) << 3) bên trong ảnh cùng vị trí được xác định bởi ColPic.
- Vị trí độ sáng (xColCb, yColCb) được gán bằng mẫu trên bên trái của khối mã độ sáng cùng vị trí được xác định bởi colCb so với mẫu độ sáng trên bên trái của ảnh cùng vị trí được xác định bởi ColPic.
- Quá trình dẫn xuất đối với các MV cùng vị trí như được xác định trong mệnh đề 8.5.2.12 được gọi với currCb, colCb, (xColCb, yColCb), refIdxLXCorner[3] và sbFlag set bằng 0 làm đầu vào, và đầu ra được gán cho mvLXCol và availableFlagLXCol.

– Ngược lại, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.

...

5.2 Phương án thực hiện 2: ràng buộc ảnh phụ với các ứng viên hợp nhất được tạo afin

8.5.5.6 Quá trình dẫn xuất đối với các ứng viên hợp nhất MV điểm điều khiển afin được tạo

Các đầu vào cho quá trình này là:

- vị trí độ sáng (xCb, yCb) xác định mẫu trên bên trái của khối mã độ sáng hiện tại so với mẫu độ sáng trên bên trái của ảnh hiện tại,
- hai biến cbWidth và cbHeight xác định chiều rộng và chiều cao của khối mã độ sáng hiện tại,
- các cờ tính khả dụng $availableA_0$, $availableA_1$, $availableA_2$, $availableB_0$, $availableB_1$, $availableB_2$, $availableB_3$,
- các vị trí mẫu ($xNbA_0$, $yNbA_0$), ($xNbA_1$, $yNbA_1$), ($xNbA_2$, $yNbA_2$), ($xNbB_0$, $yNbB_0$), ($xNbB_1$, $yNbB_1$), ($xNbB_2$, $yNbB_2$) và ($xNbB_3$, $yNbB_3$).

Đầu ra của quá trình này là:

- cờ tính khả dụng các ứng viên hợp nhất CPMV afin được tạo availableFlagConstK, với K = 1..6,

- các chỉ số tham chiếu refIdxLXConstK, với K = 1..6, X bằng 0 hoặc 1,
- các cờ sử dụng danh sách dự báo predFlagLXConstK, với K = 1..6, X bằng 0 hoặc 1,
- các chỉ số mô hình chuyển động afin motionModelIdcConstK, với K = 1..6,
- các chỉ số trọng số dự báo kép bcwIdxConstK, với K = 1..6,
- các CPMV afin được tạo cpMvLXConstK[cpIdx] với cpIdx = 0..2, K = 1..6 và X being 0 or 1.

...

CPMV thứ tư (dưới bên phải cùng vị trí) cpMvLXCorner[3], chỉ số tham chiếu refIdxLXCorner[3], cờ sử dụng danh sách dự báo predFlagLXCorner[3] và cờ tính khả dụng availableFlagCorner[3] với X bằng 0 và 1 được dẫn xuất như sau:

- Các chỉ số tham chiếu đối với ứng viên hợp nhất thời gian, refIdxLXCorner[3], với X bằng 0 hoặc 1, được gán bằng 0.
- Các biến mvLXCol và availableFlagLXCol, with X being 0 or 1, được dẫn xuất như sau:
  - Nếu slice_temporal_mvp_enabled_flag bằng 0, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.
  - Ngược lại (slice_temporal_mvp_enabled_flag bằng 1), điều kiện sau áp dụng:

xColBr = xCb + cbWidth (8-601)

yColBr = yCb + cbHeight (8-602)

rightBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos : pic_width_in_luma_samples − 1

botBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos : pic_height_in_luma_samples − 1

xColBr = Min(rightBoundaryPos , xColBr)

yColBr = Min (botBoundaryPos , yColBr)

  - Nếu yCb >> CtbLog2SizeY bằng yColBr >> CtbLog2SizeY, [[yColBr nhỏ hơn pic_height_in_luma_samples and xColBr nhỏ hơn pic_width_in_luma_samples, điều kiện sau áp dụng]]:

- Biến colCb xác định khối mã độ sáng bao phủ vị trí được chỉnh sửa được tạo bởi ((xColBr >> 3) << 3, (yColBr >> 3) << 3) bên trong ảnh cùng vị trí được xác định bởi ColPic.
- Vị trí độ sáng (xColCb, yColCb) được gán bằng mẫu trên bên trái của khối mã độ sáng cùng vị trí được xác định bởi colCb so với mẫu độ sáng trên bên trái của ảnh cùng vị trí được xác định bởi ColPic.
- Quá trình dẫn xuất đối với các MV cùng vị trí như được xác định trong mệnh đề 8.5.2.12 được gọi với currCb, colCb, (xColCb, yColCb), refIdxLXCorner[3] và sbFlag set bằng 0 làm đầu vào, và đầu ra được gán cho mvLXCol và availableFlagLXCol.

– Ngược lại, cả hai thành phần của mvLXCol được gán bằng 0 và availableFlagLXCol được gán bằng 0.

...

5.3 Phương án thực hiện 3: Nạp các mẫu số nguyên theo ràng buộc ảnh phụ

8.5.6.3.3 Quá trình nạp mẫu số nguyên độ sáng

Các đầu vào cho quá trình này là:

– vị trí độ sáng theo các đơn vị mẫu toàn phần ($xInt_L$, $yInt_L$),

– mảng mẫu tham chiếu độ sáng $refPicLX_L$,

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

Biến shift được gán bằng Max(2, 14 − $BitDepth_Y$).

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

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

– Nếu subpic treated as pic flag[SubPicIdx] bằng 1, điều kiện sau áp dụng:

$$xInt = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt)$$

$$yInt = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt)$$

– Ngược lại:

xInt = Clip3(0, picW − 1, sps_ref_wraparound_enabled_flag ?

(8-782)

ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW , $xInt_L$) : $xInt_L$)

yInt = Clip3(0, picH − 1, $yInt_L$)

(8-783)

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

$predSampleLX_L = refPicLX_L[xInt][yInt] << shift3$ (8-784)

5.4 Phương án thực hiện 4: dẫn xuất biến invAvgLuma trong phép chia tỷ lệ dư sắc độ của LMCS

8.7.5.3 Tái tạo ảnh với quá trình chia tỷ lệ dư sắc độ phụ thuộc độ sáng cho các mẫu sắc độ

Các đầu vào cho quá trình này là:

- vị trí sắc độ (xCurr, yCurr) của mẫu sắc độ trên bên trái của khối biến đổi sắc độ hiện tại so với mẫu sắc độ trên bên trái của ảnh hiện tại,
- biến nCurrSw xác định chiều rộng khối biến đổi sắc độ,
- biến nCurrSh xác định chiều cao khối biến đổi sắc độ,
- biến tuCbfChroma xác định cờ khối được mã hóa của khối biến độ sắc độ hiện tại,
- mảng (nCurrSw)x(nCurrSh) predSamples xác định các mẫu dự báo sắc độ của khối hiện tại,
- mảng (nCurrSw)x(nCurrSh) resSamples xác định các mẫu dư sắc độ của khối hiện tại,

Đầu ra của quá trình này là mảng mẫu ảnh sắc độ được tái tạo recSamples.

Biến sizeY được gán bằng Min(CtbSizeY, 64).

Mảng mẫu ảnh sắc độ được tái tạo recSamples được dẫn xuất như sau với i = 0..nCurrSw − 1, j = 0..nCurrSh − 1:

- …
- Ngược lại, điều kiện sau áp dụng:
  - …

- Biến currPic xác định mảng của mẫu độ sáng được tái tạo trong ảnh hiện tại.
- Để dẫn xuất biến varScale các bước có thứ tự sau áp dụng:
  1. Biến invAvgLuma được dẫn xuất như sau:
     - Mảng recLuma[i] với i=0..(2 * sizeY − 1) và biến cnt được dẫn xuất như sau:
       - Biến cnt được gán bằng 0.
       - ***BiếnrightBoundaryPos and botBoundaryPos được dẫn xuất như sau:***

         rightBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos : pic_width_in_luma_samples − 1

         botBoundaryPos = subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos : pic_height_in_luma_samples − 1
       - Khi availL bằng TRUE, mảng recLuma[i] với i = 0..sizeY − 1 được gán bằng

         currPic[xCuCb − 1][Min(yCuCb + i, [[pic_height_in_luma_samples – 1]] ***botBoundaryPos***)] với i = 0..sizeY − 1, và cnt được gán bằng sizeY
       - Khi availT bằng TRUE, mảng recLuma[cnt + i] với i = 0..sizeY – 1 được gán bằng

         currPic[Min(xCuCb + i, [[pic_width_in_luma_samples − 1]] ***rightBoundaryPos***)][yCuCb − 1] với i = 0..sizeY − 1, và cnt được gán bằng (cnt + sizeY)
     - Biến invAvgLuma được dẫn xuất như sau:
       - Nếu cnt lớn hơn 0, điều kiện sau áp dụng:

         $invAvgLuma = Clip1_Y((\sum_{k=0}^{cnt-1} recLuma[k] + (cnt >> 1)) >> Log2(cnt))$

         (8-1013)
       - Ngược lại (cnt bằng 0), điều kiện sau áp dụng:

         $invAvgLuma = 1 << (BitDepth_Y - 1)$

         (8-1014)

5.5 Phương án thực hiện 5: ví dụ định nghĩa phần tử ảnh phụ theo đơn vị N (chẳng hạn N=8 hoặc 32) khác ngoài 4 mẫu

7.4.3.3 Ngữ nghĩa RBSP tập hợp tham số chuỗi

subpic_grid_col_width_minus1 cộng 1 xác định chiều rộng của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của ~~4~~ *N* mẫu. Chiều dài của phần tử cú pháp là Ceil(Log2(pic_width_max_in_luma_samples / ~~4~~ *N*)) bit.

Biến NumSubPicGridCols được dẫn xuất như sau:

NumSubPicGridCols =
(pic_width_max_in_luma_samples + subpic_grid_col_width_minus1 *[[4+ 3]] *N* +*N-1*) /
(subpic_grid_col_width_minus1 * [[4+ 3]] *N* + *N-1*) (7-5)

subpic_grid_row_height_minus1 plus 1 xác định chiều cao của mỗi phần tử của lưới định danh ảnh phụ theo các đơn vị của 4 mẫu. Chiều dài của phần tử cú pháp là Ceil(Log2(pic_height_max_in_luma_samples / ~~4~~ *N*)) bit.

Biến NumSubPicGridRows được dẫn xuất như sau:

NumSubPicGridRows =
(pic_height_max_in_luma_samples + subpic_grid_row_height_minus1 * ~~4~~ *N* + *N-1*) /
(subpic_grid_row_height_minus1 *[[4+ 3] *N* + *N-1*)

7.4.7.1 Ngữ nghĩa tiêu đề lát chung

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

SubPicIdx =
CtbToSubPicIdx[CtbAddrBsToRs[FirstCtbAddrBs[SliceBrickIdx[0]]]]
if(subpic_treated_as_pic_flag[SubPicIdx]) {
SubPicLeftBoundaryPos =
SubPicLeft[SubPicIdx] * (subpic_grid_col_width_minus1 + 1) * ~~4~~ *N*
SubPicRightBoundaryPos =
(SubPicLeft[SubPicIdx] + SubPicWidth[SubPicIdx]) *
(subpic_grid_col_width_minus1 + 1) * ~~4~~ *N* (7-93)
SubPicTopBoundaryPos =

SubPicTop[SubPicIdx] * (subpic_grid_row_height_minus1 + 1)* ***4 N***

SubPicBotBoundaryPos =

(SubPicTop[SubPicIdx] + SubPicHeight[SubPicIdx]) *

(subpic_grid_row_height_minus1 + 1) * ***4N***

}

5.6 Phương án thực hiện 6: giới hạn chiều rộng ảnh và chiều cao ảnh bằng hoặc lớn hơn 8

7.4.3.3 Ngữ nghĩa RBSP tập hợp tham số chuỗi

pic_width_max_in_luma_samples xác định chiều rộng lớn nhất, theo các đơn vị của mẫu độ sáng, của mỗi ảnh được giải mã đề cập đến SPS. pic_width_max_in_luma_samples sẽ không bằng 0 và sẽ là bội số nguyên của [[MinCbSizeY]] ***Max(8, MinCbSizeY)***.

pic_height_max_in_luma_samples xác định chiều cao lớn nhất, theo các đơn vị của mẫu độ sáng, của mỗi ảnh được giải mã đề cập đến SPS. pic_height_max_in_luma_samples sẽ không bằng 0 và sẽ là bội số nguyên của [[MinCbSizeY]] ***Max(8, MinCbSizeY)***.

5.7 Phương án thực hiện 7: kiểm tra đường biên ảnh phụ để tách BT/TT/QT, dẫn xuất chiều sâu BT/TT/QT, và/hoặc cờ tách CU báo hiệu

6.4.2 Quá trình tách nhị phân được phép

Biến allowBtSplit được dẫn xuất như sau:

- …
- Ngược lại, nếu tất cả điều kiện sau là đúng, allowBtSplit được gán bằng FALSE
  - btSplit bằng SPLIT_BT_VER
  - y0 + cbHeight lớn hơn [[pic_height_in_luma_samples]]***subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples***.
- Ngược lại, nếu tất cả điều kiện sau là đúng, allowBtSplit được gán bằng FALSE
  - btSplit bằng SPLIT_BT_VER
  - cbHeight lớn hơn MaxTbSizeY

- x0 + cbWidth lớn hơn [[pic_width_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples***

– Ngược lại, nếu tất cả điều kiện sau là đúng, allowBtSplit được gán bằng FALSE
  - btSplit bằng SPLIT_BT_HOR
  - cbWidth lớn hơn MaxTbSizeY
  - y0 + cbHeight lớn hơn [[pic_height_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicBotBoundaryPos + 1: pic_height_in_luma_samples.***

– Ngược lại, nếu tất cả điều kiện sau là đúng, allowBtSplit được gán bằng FALSE
  - x0 + cbWidth lớn hơn [[pic_width_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples***
  - y0 + cbHeight lớn hơn [[pic_height_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicBotBoundaryPos + 1: pic_height_in_luma_samples***.
  - cbWidth lớn hơn minQtSize

– Ngược lại, nếu tất cả điều kiện sau là đúng, allowBtSplit được gán bằng FALSE
  - btSplit bằng SPLIT_BT_HOR
  - x0 + cbWidth lớn hơn [[pic_width_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples***
  - y0 + cbHeight nhỏ hơn hoặc bằng [[pic_height_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx]*** ? ***SubPicBotBoundaryPos + 1: pic_height_in_luma_samples.***

6.4.2 Quá trình tách tam phân được phép

Biến allowTtSplit được dẫn xuất như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, allowTtSplit được gán bằng FALSE:
  - cbSize nhỏ hơn hoặc bằng 2 * MinTtSizeY
  - cbWidth lớn hơn Min(MaxTbSizeY, maxTtSize)
  - cbHeight lớn hơn Min(MaxTbSizeY, maxTtSize)
  - mttDepth lớn hơn hoặc bằng maxMttDepth
  - x0 + cbWidth lớn hơn [[pic_width_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples***
  - y0 + cbHeight lớn hơn [[pic_height_in_luma_samples]] ***subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples***.
  - treeType bằng DUAL_TREE_CHROMA và (cbWidth / SubWidthC) * (cbHeight / SubHeightC) nhỏ hơn hoặc bằng 32
  - treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA
- Ngược lại, allowTtSplit được gán bằng TRUE.

7.3.8.2 Cú pháp đơn vị cây mã

dual_tree_implicit_qt_split(x0, y0, cbSize, cqtDepth) {	**Mô tả**
…	
if(x1 < [[pic_width_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples)***)	
dual_tree_implicit_qt_split(x1, y0, cbSize / 2, cqtDepth + 1)	
if(y1 < [[pic_height_in_luma_samples]] (***subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples***))	
dual_tree_implicit_qt_split(x0, y1, cbSize / 2, cqtDepth + 1)	
if(x1 < [[pic_width_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples)*** && y1 < [[pic_height_in_luma_samples]] (***subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples***))	
dual_tree_implicit_qt_split(x1, y1, cbSize / 2, cqtDepth + 1)	
} else {	
…	
}	
}	

7.3.8.4 Cú pháp cây mã

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Mô tả
if((allowSplitBtVer || allowSplitBtHor || allowSplitTtVer || allowSplitTtHor || allowSplitQT) && (x0 + cbWidth <= [[pic_width_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples)*** && (y0 + cbHeight <= [[pic_height_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples)***))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= cu_qp_delta_subdiv) {	
…	
depthOffset += (y0 + cbHeight > [[pic_height_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples)***) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
coding_tree(x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < [[pic_height_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples)***)	
coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
…	
if(x1 < [[pic_width_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples)***)	
coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)	
if(y1 < [[pic_height_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples)***)	
coding_tree(x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)	
if(y1 < [[pic_height_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicBotBoundaryPos + 1: pic_height_in_luma_samples)*** && x1 < [[pic_width_in_luma_samples]] ***(subpic_treated_as_pic_flag[SubPicIdx] ? SubPicRightBoundaryPos + 1 : pic_width_in_luma_samples)***)	
coding_tree(x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	

5.8 Phương án thực hiện 8: ví dụ định nghĩa các ảnh phụ

seq_parameter_set_rbsp() {	Mô tả
sps_decoding_parameter_set_id	u(4)
…	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
[[subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}]]	

bit_depth_luma_minus8	ue(v)
...	
log2_ctu_size_minus5	u(2)
...	
subpics_present_flag	*u(1)*
if(subpics_present_flag) {	
num_subpics_minus1	*u(8)*
for(i = 0; i <= num_subpics_minus1; i++) {	
subpic_ctb_addr_x[i]	*u(8)*
subpic_ctb_addr_y[i]	*u(8)*
subpic_ctb_width_minus1[i]	*u(8)*
subpic_ctb_height_minus1[i]	*u(8)*
subpic_treated_as_pic_flag[i]	*u(1)*
loop_filter_across_subpic_enabled_flag[i]	*u(1)*
}	
...	

5.9 Phương án thực hiện 9: ví dụ định nghĩa các ảnh phụ

seq_parameter_set_rbsp() {	Mô tả
sps_decoding_parameter_set_id	u(4)
...	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
[[subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}]]	
bit_depth_luma_minus8	ue(v)
...	
log2_ctu_size_minus5	u(2)
...	
subpics_present_flag	*u(1)*
if(subpics_present_flag) {	
num_subpics_minus1	*ue(v)*
for(i = 0; i <= num_subpics_minus1; i++) {	
subpic_ctb_addr_x[i]	*u(8)*
subpic_ctb_addr_y[i]	*u(8)*
subpic_ctb_width_minus1[i]	*u(8)*
subpic_ctb_height_minus1[i]	*u(8)*
subpic_treated_as_pic_flag[i]	*u(1)*
loop_filter_across_subpic_enabled_flag[i]	*u(1)*
}	
...	

5.10 Phương án thực hiện 10: ví dụ định nghĩa các ảnh phụ

seq_parameter_set_rbsp() {	Mô tả
sps_decoding_parameter_set_id	u(4)
...	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
[[subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}]]	
...	
log2_ctu_size_minus5	u(2)
...	
subpics_present_flag	***u(1)***
if(subpics_present_flag) {	
num_subpics_minus2	***u(v)***
subpic_addr_x_length_minus1	***ue(v)***
subpic_addr_y_length_minus1	***ue(v)***
for(i = 0; i < NumSubPics; i++) {	
subpic_ctb_addr_x[i]	***u(v)***
subpic_ctb_addr_y[i]	***u(v)***
subpic_ctb_width_minus1[i]	***u(v)***
subpic_ctb_height_minus1[i]	***u(v)***
subpic_treated_as_pic_flag[i]	***u(1)***
loop_filter_across_subpic_enabled_flag[i]	***u(1)***
}	
...	

5.11 Phương án thực hiện 11: ví dụ định nghĩa các ảnh phụ

seq_parameter_set_rbsp() {	Mô tả
sps_decoding_parameter_set_id	u(4)
...	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
[[subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}]]	
...	

log2_ctu_size_minus5	u(2)
...	
subpics_present_flag	*u(1)*
if(subpics_present_flag) {	
num_subpics_minus2	*u(v)*
subpic_addr_x_length_minus1	*ue(v)*
subpic_addr_y_length_minus1	*ue(v)*
for(i = 0; i < NumSubPics; i++) {	
if(i < NumSubPics – 1){	
subpic_ctb_addr x[i]	*u(v)*
subpic_ctb_addr y[i]	*u(v)*
subpic_ctb_width_minus1[i]	*u(v)*
subpic_ctb_height_minus1[i]	*u(v)*
}	
subpic_treated_as_pic_flag[i]	*u(1)*
loop_filter_across_subpic_enabled_flag[i]	*u(1)*
}	
...	

NumSubPics= num_subpics_minus2+2.

Fig.3 là sơ đồ khối của thiết bị xử lý viđeo 300. Thiết bị 300 có thể được sử dụng để thực hiện một hoặc nhiều phương pháp được mô tả ở đây. Thiết bị 300 có thể được triển khai trong điện thoại thông minh, máy tính bảng, máy tính, bộ thu Internet vạn vật (IoT), và v.v.. Thiết bị 300 có thể bao gồm một hoặc nhiều bộ xử lý 302, một hoặc nhiều bộ nhớ 304 và phần cứng xử lý viđeo 306. (các) bộ xử lý 302 có thể được tạo cấu hình để thực hiện một hoặc nhiều phương pháp được nêu trong bản mô tả. Bộ nhớ (các bộ nhớ) 304 có thể được sử dụng để lưu trữ dữ liệu và mã được sử dụng để thực hiện các phương pháp và các kỹ thuật được mô tả ở đây. Phần cứng xử lý viđeo 306 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được nêu trong bản mô tả.

Fig.4 là lưu đồ đối với phương pháp 400 xử lý viđeo. Phương pháp 400 bao gồm các bước xác định (402), đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó bộ dự báo vectơ chuyển động thời gian được xác định đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại sử dụng chế độ afin nằm trong vùng viđeo thứ hai, và thực hiện (404) phép biến đổi dựa trên việc xác định.

Các giải pháp sau có thể được triển khai dưới dạng các giải pháp được ưu tiên theo một số phương án thực hiện.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 1).

1. Phương pháp xử lý viđeo, bao gồm các bước: xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó bộ dự báo vectơ chuyển động thời gian được xác định đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại sử dụng chế độ afin nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

2. Phương pháp theo giải pháp 1, trong đó khối viđeo được bao phủ bởi vùng thứ nhất và vùng thứ hai.

3. Phương pháp theo giải pháp 1 hoặc 2, trong đó, trong trường hợp mà vị trí của bộ dự báo MV thời gian nằm ngoài vùng viđeo thứ hai, sau đó bộ dự báo MV thời gian được đánh dấu là không khả dụng và không được sử dụng trong phép biến đổi.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 2).

4. Phương pháp xử lý viđeo, bao gồm bước: xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó mẫu số nguyên trong ảnh tham chiếu được nạp đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai, trong đó ảnh tham chiếu không được sử dụng trong quá trình nội suy trong khi biến đổi; và thực hiện biến đổi dựa trên việc xác định.

5. Phương pháp theo giải pháp 4, trong đó khối viđeo được bao phủ bởi vùng thứ nhất và vùng thứ hai.

6. Phương pháp theo giải pháp 4 hoặc 5, trong đó, trong trường hợp mà vị trí của mẫu nằm ngoài vùng viđeo thứ hai, sau đó mẫu được đánh dấu là không khả dụng và không được sử dụng trong phép biến đổi.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 3).

7. Phương pháp xử lý viđeo, bao gồm: xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó giá trị mẫu độ sáng được tái tạo được nạp đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

8. Phương pháp theo giải pháp 7, trong đó mẫu độ sáng được bao phủ bởi vùng thứ nhất và vùng thứ hai.

9. Phương pháp theo giải pháp 7 hoặc 8, trong đó, trong trường hợp mà vị trí của mẫu độ sáng nằm ngoài vùng viđeo thứ hai, sau đó mẫu độ sáng được đánh dấu là không khả dụng và không được sử dụng trong phép biến đổi.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 4).

10. Phương pháp xử lý viđeo, bao gồm các bước: xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó phép kiểm tra liên quan đến tách, dẫn xuất chiều sâu hoặc báo hiệu cờ tách cho khối viđeo được thực hiện trong quá trình biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại nằm trong vùng viđeo thứ hai; và thực hiện biến đổi dựa trên việc xác định.

11. Phương pháp theo giải pháp 10, trong đó vị trí được bao phủ bởi vùng thứ nhất và vùng thứ hai.

12. Phương pháp theo giải pháp 10 hoặc 11, trong đó, trong trường hợp mà vị trí nằm ngoài vùng viđeo thứ hai, sau đó mẫu độ sáng được đánh dấu là không khả dụng và không được sử dụng trong phép biến đổi.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 8).

13. Phương pháp xử lý viđeo, bao gồm: thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh viđeo bao gồm một hoặc nhiều khối viđeo, và biểu diễn được mã hóa của viđeo, trong đó biểu diễn được mã hóa tuân theo yêu cầu cú pháp mã hóa mà phép biến đổi không sử dụng mã hóa/giải mã ảnh phụ và công cụ mã hóa/giải mã biến đổi độ phân giải động hoặc công cụ lấy mẫu lại ảnh tham chiếu trong đơn vị viđeo.

14. Phương pháp theo giải pháp 13, trong đó đơn vị viđeo tương ứng với chuỗi của một hoặc nhiều ảnh viđeo.

15. Phương pháp theo giải pháp 13 hoặc 14, trong đó công cụ mã hóa/giải mã biến đổi độ phân giải động bao gồm công cụ mã hóa/giải mã biến đổi độ phân giải thích ứng.

16. Phương pháp theo giải pháp 13 hoặc 14, trong đó công cụ mã hóa/giải mã biến đổi độ phân giải động bao gồm công cụ mã hóa/giải mã biến đổi độ phân giải động.

17. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 13 đến 16, trong đó biểu diễn được mã hóa chỉ báo rằng đơn vị viđeo tuân theo yêu cầu cú pháp mã hóa.

18. Phương pháp theo giải pháp 17, trong đó biểu diễn được mã hóa chỉ báo rằng đơn vị viđeo sử dụng mã hóa ảnh phụ.

19. Phương pháp theo giải pháp 17, trong đó biểu diễn được mã hóa chỉ báo rằng đơn vị viđeo sử dụng công cụ mã hóa/giải mã biến đổi độ phân giải động hoặc công cụ lấy mẫu lại ảnh tham chiếu.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 10).

20. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 19, trong đó vùng viđeo thứ hai bao gồm ảnh phụ viđeo và trong đó các đường biên của vùng viđeo thứ hai và vùng viđeo khác cũng là đường biên giữa hai CTU.

21. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 19, trong đó vùng viđeo thứ hai bao gồm ảnh phụ viđeo và trong đó các đường biên của vùng viđeo thứ hai và vùng viđeo khác cũng là đường biên giữa hai CTU.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 11).

22. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 21, trong đó vùng viđeo thứ nhất và vùng viđeo thứ hai có hình chữ nhật.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 12).

23. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 22, trong đó vùng viđeo thứ nhất và vùng viđeo thứ hai không chồng lấn.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 13).

24. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 23, trong đó ảnh viđeo được phân chia thành các vùng viđeo sao cho pixel trong ảnh viđeo được bao phủ bởi một và chỉ một vùng viđeo.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 15).

25. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 24, trong đó ảnh viđeo được tách thành vùng viđeo thứ nhất và vùng viđeo thứ hai do ảnh viđeo nằm trong lớp cụ thể của chuỗi viđeo.

Các giải pháp sau có thể được triển khai cùng với các kỹ thuật bổ sung được mô tả trong các mục được liệt kê trong phần trước đó (chẳng hạn, mục 10).

26. Phương pháp xử lý viđeo, bao gồm: thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh viđeo bao gồm một hoặc nhiều khối viđeo, và biểu diễn được mã hóa của viđeo, trong đó biểu diễn được mã hóa tuân theo yêu cầu cú pháp mã hóa rằng phần tử cú pháp thứ nhất subpic_grid_idx[i][j] không lớn hơn phần tử cú pháp thứ hai max_subpics_minus1.

27. Phương pháp theo giải pháp 26, trong đó từ mã biểu diễn phần tử cú pháp thứ nhất không lớn hơn từ mã biểu diễn phần tử cú pháp thứ hai.

28. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 27, trong đó vùng viđeo thứ nhất bao gồm ảnh phụ viđeo.

29. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 28, trong đó vùng viđeo thứ hai bao gồm ảnh phụ viđeo.

30. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 29, trong đó phép biến đổi bao gồm mã hóa viđeo thành biểu diễn được mã hóa.

31. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 đến 29, trong đó phép biến đổi bao gồm giải mã biểu diễn được mã hóa để tạo các giá trị pixel của viđeo.

32. Thiết bị giải mã viđeo bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều theo giải pháp từ 1 đến 31.

33. Thiết bị mã hóa viđeo bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều theo giải pháp từ 1 đến 31.

34. Sản phẩm chương trình máy tính có mã máy tính được lưu trữ trên đó, mã, khi được thực thi bằng bộ xử lý, khiến bộ xử lý thực thi phương pháp được nêu trong giải pháp bất kỳ trong các giải pháp từ 1 đến 31.

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

Fig.8 là sơ đồ khối thể hiện hệ thống xử lý viđeo ví dụ 800 trong đó các kỹ thuật khác nhau được bộc lộ ở đây có thể được triển khai. Các cách thực hiện khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 800. Hệ thống 800 có thể bao gồm đầu vào 802 để nhận nội dung viđeo. Nội dung viđeo có thể được nhận ở định dạng thô hoặc không được nén, chẳng hạn, các giá trị pixel đa thành phần 8 hoặc 10 bit, hoặc có thể ở định dạng nén hoặc được mã hóa. Đầu vào 802 có thể là giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện lưu trữ. Các ví dụ của giao diện mạng bao gồm các giao diện hữu tuyến chẳng hạn Ethernet, mạng quang thụ động (PON), v.v. và giao diện không dây chẳng hạn giao diện Wi-Fi hoặc tế bào.

Hệ thống 800 có thể bao gồm thành phần mã 804 có thể thực hiện các phương pháp tạo mã hoặc mã hóa khác nhau được nêu trong bản mô tả. Thành phần mã 804 có thể giảm tốc độ bit trung bình của viđeo từ đầu vào 802 đến đầu ra của thành phần mã 804 để tạo biểu diễn được mã hóa của viđeo. Do vậy, các kỹ thuật mã hóa đôi lúc được gọi là các kỹ thuật nén viđeo hoặc chuyển mã viđeo. Đầu ra của thành phần mã 804 có thể được lưu trữ, hoặc được truyền qua kết nối truyền thông, như được biểu diễn bởi thành phần 806. Biểu diễn dòng bit được lưu trữ hoặc được truyền thông (hoặc được mã hóa) của viđeo được nhận ở đầu vào 802 có thể được sử dụng bởi thành phần 808 để tạo các giá trị pixel hoặc viđeo có thể hiển thị được truyền đến giao diện hiển thị 810. Quá trình tạo viđeo người dùng có thể xem từ biểu diễn dòng bit đôi lúc được gọi là giải nén viđeo. Ngoài ra, trong khi các hoạt động xử lý viđeo cụ thể được gọi là các hoạt động hoặc các công cụ "mã hóa", cần hiểu rằng các hoạt động hoặc các công cụ mã hóa được sử dụng ở bộ mã hóa và các công cụ hoặc các hoạt động giải mã tương ứng mà đảo ngược kết quả mã hóa sẽ được thực thi bởi bộ giải mã.

Các ví dụ về giao diện buýt ngoại vi hoặc giao diện hiển thị có thể bao gồm buýt nối tiếp đa năng (USB) hoặc giao diện đa phương tiện độ nét cao (HDMI)

hoặc Displayport, và v.v.. Các ví dụ về các giao diện lưu trữ bao gồm SATA (serial advanced technology attachment), PCI, giao diện IDE, và tương tự. Các kỹ thuật được nêu trong bản mô tả có thể được triển khai trong các thiết bị điện tử khác nhau chẳng hạn, điện thoại di động, máy tính xách tay, điện thoại thông minh hoặc các thiết bị khác có thể thực hiện xử lý dữ liệu số và/hoặc hiển thị viđeo.

Fig.9 là sơ đồ khối minh họa hệ thống tạo mã viđeo ví dụ 100 có thể sử dụng các kỹ thuật của sáng chế.

Như được thể hiện trên Fig.9, hệ thống mã hóa viđeo 100 có thể bao gồm thiết bị nguồn 110 và thiết bị đích 120. Thiết bị nguồn 110 tạo dữ liệu viđeo được mã hóa có thể được gọi là thiết bị mã hóa viđeo. Thiết bị đích 120 có thể giải mã dữ liệu viđeo được mã hóa được tạo bởi thiết bị nguồn 110 có thể được gọi là thiết bị giải mã viđeo.

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

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

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

Giao diện I/O 126 có thể bao gồm bộ thu và/hoặc modem. Giao diện I/O 126

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

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

Fig.10 là sơ đồ khối minh họa ví dụ của bộ mã hóa viđeo 200, có thể là bộ mã hóa viđeo 114 in hệ thống 100 được minh họa trên Fig.9.

Bộ mã hóa viđeo 200 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật của sáng chế. Trong ví dụ trên Fig.10, bộ mã hóa viđeo 200 bao gồm các thành phần chức năng. Các kỹ thuật được mô tả theo sáng chế có thể được chia sẻ trong số các thành phần khác nhau của bộ mã hóa viđeo 200. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả theo sáng chế.

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

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

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

Khối phân vùng 201 có thể phân vùng ảnh thành một hoặc nhiều khối viđeo.

Bộ mã hóa viđeo 200 và bộ giải mã viđeo 300 có thể hỗ trợ các kích thước khối viđeo khác nhau.

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

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

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

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

viđeo hiện tại.

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

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

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

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

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

mã viđeo 300 có thể sử dụng vectơ chuyển động của khối viđeo được chỉ báo và MVD để xác định vectơ chuyển động của khối viđeo hiện tại.

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

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

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

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

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

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

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

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

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

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

Fig.11 là sơ đồ khối minh họa ví dụ của bộ giải mã viđeo 300 có thể là bộ giải mã viđeo 114 trong hệ thống 100 được minh họa trên Fig.9.

Bộ giải mã viđeo 300 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật của sáng chế. Trong ví dụ trên Fig.11, bộ giải mã viđeo 300 bao gồm các thành phần chức năng. Các kỹ thuật được mô tả theo sáng chế có thể được chia sẻ giữa các thành phần khác nhau của bộ giải mã viđeo 300. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả theo sáng chế.

Trong ví dụ trên Fig.11, bộ giải mã viđeo 300 bao gồm khối giải mã entropy 301, khối bù chuyển động 302, khối dự báo trong 303, khối lượng tử ngược 304, khối biến đổi ngược 305, và khối tái tạo 306 và bộ đệm 307. Bộ giải mã viđeo 300 có thể, trong một số ví dụ, thực hiện bước giải mã nghịch đảo với bước mã hóa được mô tả với bộ mã hóa viđeo 200 (chẳng hạn, Fig.10).

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

Khối bù chuyển động 302 có thể tạo các khối được bù chuyển động, có thể

thực hiện nội suy dựa trên các bộ lọc nội suy. Các định danh cho các bộ lọc nội suy cần được sử dụng có độ chính xác pixel con có thể được bao gồm trong các phần tử cú pháp.

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

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

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

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

Fig.12 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1200 bao gồm, ở hoạt động 1210, thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit được yêu

cầu để tuân theo quy tắc định dạng mà xác định rằng mỗi ảnh được mã hóa dưới dạng một hoặc nhiều lát. Quy tắc định dạng chặn các mẫu ở ảnh không được bao phủ bởi lát bất kỳ trong một hoặc nhiều lát.

Theo một số phương án thực hiện, ảnh bao gồm nhiều ảnh phụ, và quy tắc định dạng còn xác định rằng nhiều lát của ảnh phụ của ảnh bao phủ toàn bộ ảnh phụ. Theo một số phương án thực hiện, biểu diễn dòng bit bao gồm phần tử cú pháp chỉ báo rằng nhiều lát có hình chữ nhật. Theo một số phương án thực hiện, lát chồng lấn với nhiều nhất một ảnh phụ của ảnh. Theo một số phương án thực hiện, ô của viđeo chồng lấn với nhiều nhất một ảnh phụ của ảnh. Theo một số phương án thực hiện, khối viên của viđeo chồng lấn với nhiều nhất một ảnh phụ của ảnh.

Fig.13 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1300 bao gồm, ở hoạt động 1310, xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, cách thức báo hiệu thông tin của một hoặc nhiều lát trong ảnh theo quy tắc được liên kết với số lượng ô hoặc số lượng khối viên trong ảnh. Phương pháp 1300 bao gồm, ở hoạt động 1320, thực hiện biến đổi dựa trên việc xác định.

Theo một số phương án thực hiện, thông tin của một hoặc nhiều lát của ảnh bao gồm ít nhất số lượng của một hoặc nhiều lát hoặc khoảng của một hoặc nhiều lát. Theo một số phương án thực hiện, quy tắc này xác định rằng, nếu số lượng khối viên trong ảnh bằng một, thông tin của một hoặc nhiều lát trong ảnh bị loại trừ trong biểu diễn dòng bit. Theo một số phương án thực hiện, số lượng của một hoặc nhiều lát trong ảnh được xem là bằng 1. Theo một số phương án thực hiện, lát bao phủ toàn bộ ảnh. Theo một số phương án thực hiện, số lượng khối viên trên lát được xem là bằng 1.

Fig.14 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1400 bao gồm, ở hoạt động 1410, thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo theo quy tắc. ảnh được mã hóa trong biểu diễn dòng bit dưới dạng một hoặc nhiều lát. Quy tắc này xác định liệu hoặc cách thức địa chỉ của lát của ảnh được bao gồm trong biểu diễn dòng bit.

Theo một số phương án thực hiện, quy tắc này xác định rằng việc báo hiệu

địa chỉ của lát độc lập với việc báo hiệu liệu lát có hình chữ nhật hay không. Theo một số phương án thực hiện, quy tắc này xác định rằng việc báo hiệu địa chỉ của lát phụ thuộc vào số lượng lát trong ảnh nếu lát dưới dạng hình chữ nhật. Theo một số phương án thực hiện, liệu có báo hiệu số lượng khối viên trong lát ít nhất một phần dựa trên địa chỉ của lát. Theo một số phương án thực hiện, liệu có báo hiệu số lượng khối viên trong lát hay không còn dựa trên số lượng khối viên trong ảnh.

Fig.15 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1500 bao gồm, ở hoạt động 1510, xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, liệu phần tử cú pháp chỉ báo hoạt động lọc truy nhập các mẫu giữa nhiều khối viên trong ảnh được kích hoạt được bao gồm trong biểu diễn dòng bit dựa trên số lượng ô hoặc số lượng khối viên trong ảnh. Phương pháp 1500 bao gồm, ở hoạt động 1520, thực hiện biến đổi dựa trên việc xác định. Theo một số phương án thực hiện, phần tử cú pháp bị loại trừ trong biểu diễn dòng bit nếu số lượng khối viên trong ảnh nhỏ hơn 2.

Fig.16 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1600 bao gồm, ở hoạt động 1610, thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo. Ảnh bao gồm một hoặc nhiều ảnh phụ, và số lượng của một hoặc nhiều ảnh phụ được chỉ báo bởi phần tử cú pháp trong biểu diễn dòng bit.

Theo một số phương án thực hiện, số lượng nhiều ảnh phụ được biểu diễn dưới dạng N, và phần tử cú pháp bao gồm giá trị của (N-d), N và d là các số nguyên. Theo một số phương án thực hiện, d bằng 0, 1, hoặc 2. Theo một số phương án thực hiện, giá trị của (N-d) được mã hóa nhờ sử dụng phương tiện tạo mã đơn nguyên hoặc phương tiện tạo mã đơn nguyên rút gọn. Theo một số phương án thực hiện, giá trị của (N-d) được mã hóa có chiều dài cố định nhờ sử dụng phương tiện mã hóa có chiều dài cố định. Theo một số phương án thực hiện, chiều dài cố định bằng 8 bit. Theo một số phương án thực hiện, chiều dài cố định được biểu diễn dưới dạng x-dx, x và dx là các số nguyên dương. x bằng hoặc nhỏ hơn giá trị lớn nhất được xác định dựa trên quy tắc tương hợp, và dx bằng 0, 1, hoặc 2. Theo một số phương án thực hiện, x-dx được báo hiệu. Theo

một số phương án thực hiện, chiều dài cố định được xác định dựa trên số lượng của các khối đơn vị cơ bản trong ảnh. Theo một số phương án thực hiện, chiều dài cố định được biểu diễn dưới dạng x, số lượng của các khối đơn vị cơ bản được biểu diễn dưới dạng M, và x = Ceil(log2(M+d0)) + d1, d0 và d1 là các số nguyên.

Theo một số phương án thực hiện, giá trị lớn nhất của (N-d) được tiền xác định. Theo một số phương án thực hiện, giá trị lớn nhất của (N-d) được xác định dựa trên số lượng của các khối đơn vị cơ bản trong ảnh. Theo một số phương án thực hiện, số lượng của các khối đơn vị cơ bản trong ảnh được biểu diễn dưới dạng M, M là số nguyên, và giá trị lớn nhất của N-d là M-d. Theo một số phương án thực hiện, khối đơn vị cơ bản bao gồm lát. Theo một số phương án thực hiện, khối đơn vị cơ bản bao gồm đơn vị cây mã (CTU). Theo một số phương án thực hiện, ít nhất một trong d0 hoặc d1 bằng -2, -1, 0, 1, hoặc 2.

Fig.17 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1700 bao gồm, ở hoạt động 1710, thực hiện biến đổi giữa ảnh của viđeo bao gồm một hoặc nhiều ảnh phụ và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng thông tin về ảnh phụ được bao gồm trong biểu diễn dòng bit dựa trên ít nhất một trong: (1) một hoặc nhiều vị trí góc của ảnh phụ, hoặc (2) kích thước của ảnh phụ.

Theo một số phương án thực hiện, quy tắc định dạng còn xác định rằng thông tin về ảnh phụ được đặt sau thông tin về kích thước của khối cây mã trong biểu diễn dòng bit. Theo một số phương án thực hiện, một hoặc nhiều vị trí góc của ảnh phụ được báo hiệu trong biểu diễn dòng bit sử dụng độ hạt của khối trong ảnh phụ. Theo một số phương án thực hiện, khối bao gồm khối đơn vị cơ bản hoặc khối cây mã.

Theo một số phương án thực hiện, chỉ số của khối được sử dụng để chỉ báo vị trí góc của ảnh phụ. Theo một số phương án thực hiện, chỉ số bao gồm chỉ số cột hoặc chỉ số hàng. Theo một số phương án thực hiện, chỉ số của khối được biểu diễn dưới dạng I, và giá trị của (I - d) được báo hiệu trong biểu diễn dòng bit, d bằng 0, 1, hoặc 2. Theo một số phương án thực hiện, d được xác định dựa trên chỉ số của ảnh phụ được mã hóa trước đó và số nguyên khác d1, d1 bằng -

1, 0, hoặc 1. Theo một số phương án thực hiện, dấu của (I-d) được báo hiệu trong biểu diễn dòng bit.

Theo một số phương án thực hiện, liệu vị trí của ảnh phụ được báo hiệu trong biểu diễn dòng bit dựa trên chỉ số của ảnh phụ. Theo một số phương án thực hiện, vị trí của ảnh phụ bị bỏ qua trong biểu diễn dòng bit nếu ảnh phụ là ảnh phụ thứ nhất trong ảnh. Theo một số phương án thực hiện, vị trí trên bên trái của ảnh phụ được xác định là (0, 0). Theo một số phương án thực hiện, vị trí của ảnh phụ bị bỏ qua trong biểu diễn dòng bit nếu ảnh phụ là ảnh phụ cuối cùng trong ảnh. Theo một số phương án thực hiện, vị trí trên bên trái của ảnh phụ được xác định dựa trên thông tin về ảnh phụ được biến đổi trước đó. Theo một số phương án thực hiện, thông tin về ảnh phụ bao gồm vị trí trên bên trái và kích thước của ảnh phụ. Thông tin được mã hóa nhờ sử dụng phương tiện mã hóa đơn nguyên rút gọn, phương tiện mã hóa nhị phân rút gọn, phương tiện tạo mã đơn nguyên, phương tiện mã hóa có chiều dài cố định, hoặc phương tiện mã hóa EG thứ K.

Theo một số phương án thực hiện, kích thước của ảnh phụ được báo hiệu trong biểu diễn dòng bit sử dụng độ hạt của khối trong ảnh phụ. Theo một số phương án thực hiện, kích thước bao gồm chiều rộng hoặc chiều cao của ảnh phụ. Theo một số phương án thực hiện, kích thước được biểu diễn dưới dạng số lượng cột hoặc số lượng hàng khối trong ảnh phụ. Theo một số phương án thực hiện, số lượng cột hoặc số lượng hàng khối được biểu diễn dưới dạng I, và giá trị của (I - d) được báo hiệu trong biểu diễn dòng bit, d bằng 0, 1, hoặc 2. Theo một số phương án thực hiện, d được xác định dựa trên kích thước của khối ảnh phụ được mã hóa trước đó và số nguyên khác d1, d1 bằng -1, 0, hoặc 1. Theo một số phương án thực hiện, dấu của (I - d) được báo hiệu trong biểu diễn dòng bit.

Theo một số phương án thực hiện, giá trị của (I-d) được mã hóa có chiều dài cố định nhờ sử dụng phương tiện mã hóa có chiều dài cố định. Theo một số phương án thực hiện, chiều dài cố định bằng 8 bit. Theo một số phương án thực hiện, chiều dài cố định được biểu diễn dưới dạng x-dx, x và dx là các số nguyên dương. x bằng hoặc nhỏ hơn giá trị lớn nhất được xác định dựa trên quy tắc tương hợp, và dx bằng 0, 1, hoặc 2. Theo một số phương án thực hiện, chiều dài cố

định được xác định dựa trên số lượng của các khối đơn vị cơ bản trong ảnh. Theo một số phương án thực hiện, chiều dài cố định được biểu diễn dưới dạng x, số lượng khối đơn vị cơ bản được biểu diễn dưới dạng M, và x = Ceil(log2(M+d0)) + d1, d0 và d1 là các số nguyên. Theo một số phương án thực hiện, ít nhất một trong d0 hoặc d1 bằng -2, -1, 0, 1, hoặc 2. Theo một số phương án thực hiện, (I - d) được báo hiệu cho tất cả các ảnh phụ của ảnh. Theo một số phương án thực hiện, (I - d) được báo hiệu cho tập con của các ảnh phụ của ảnh. Theo một số phương án thực hiện, giá trị của (I-d) bị bỏ qua trong biểu diễn dòng bit nếu số lượng của các ảnh phụ bằng 1. Theo một số phương án thực hiện, giá trị của (I-d) bị bỏ qua trong biểu diễn dòng bit nếu ảnh phụ là ảnh phụ thứ nhất của ảnh. Theo một số phương án thực hiện, giá trị của (I-d) được xác định bằng 0. Theo một số phương án thực hiện, giá trị của (I-d) bị bỏ qua trong biểu diễn dòng bit nếu ảnh phụ là ảnh phụ cuối cùng của ảnh. Theo một số phương án thực hiện, giá trị của (I-d) được xác định dựa trên thông tin về ảnh phụ được biến đổi trước đó.

Theo một số phương án thực hiện, phần tử cú pháp đơn được bao gồm có điều kiện trong biểu diễn dòng bit để chỉ báo liệu tất cả các ảnh phụ của ảnh được xử lý dưới dạng ảnh. Theo một số phương án thực hiện, phần tử cú pháp đơn được bao gồm có điều kiện trong biểu diễn dòng bit để chỉ báo liệu bộ lọc vòng có thể áp dụng giữa các ảnh phụ của ảnh cho tất cả các ảnh phụ. Theo một số phương án thực hiện, phần tử cú pháp đơn bị bỏ qua trong biểu diễn dòng bit nếu số lượng ảnh phụ của ảnh bằng 1.

Fig.18 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1800 bao gồm, ở hoạt động 1810, xác định rằng công cụ lấy mẫu lại ảnh tham chiếu được kích hoạt đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo do ảnh được phân chia thành một hoặc nhiều ảnh phụ. Phương pháp 1800 cũng bao gồm, ở hoạt động 1820, thực hiện biến đổi dựa trên việc xác định.

Theo một số phương án thực hiện, hệ số chia tỷ lệ được sử dụng trong công cụ mã hóa lấy mẫu lại ảnh tham chiếu được xác định dựa trên tập hợp hệ số. Theo một số phương án thực hiện, tập hợp hệ số bao gồm ít nhất một trong: {1:1, 1:2}, {1:1, 2:1}, {1:1, 1:2, 1:4}, {1:1, 1:2, 4:1}, {1:1, 2:1, 1:4}, {1:1, 2:1, 4:1},

{1:1, 1:2, 1:4, 1:8}, {1:1, 1:2, 1:4, 8:1}, {1:1, 1:2, 4:1, 1:8}, {1:1, 1:2, 4:1, 8:1}, {1:1, 2:1, 1:4, 1:8}, {1:1, 2:1, 1:4, 8:1}, {1:1, 2:1, 4:1, 1:8}, hoặc {1:1, 2:1, 4:1, 8:1}. Theo một số phương án thực hiện, kích thước của CTB của ảnh khác với kích thước của CTB của ảnh thứ hai nếu độ phân giải của ảnh khác với độ phân giải của ảnh thứ hai. Theo một số phương án thực hiện, ảnh phụ của ảnh có kích thước của SAW×SAH và ảnh phụ của ảnh thứ hai của viđeo có kích thước của SBW×SBH. Các hệ số chia tỷ lệ giữa ảnh và ảnh thứ hai là Rw và Rh dọc theo phương ngang và hướng thẳng đứng. SAW/SBW hoặc SBW/SAW bằng Rw, và SAH/SBH hoặc SBH/SAH bằng Rh.

Fig.19 là lưu đồ biểu diễn phương pháp xử lý viđeo theo sáng chế. Phương pháp 1900 bao gồm, ở hoạt động 1910, thực hiện biến đổi giữa video bao gồm ảnh viđeo bao gồm một hoặc nhiều ảnh phụ bao gồm một hoặc nhiều lát và biểu diễn dòng bit của viđeo. Biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng, đối với ảnh phụ và lát, trong trường hợp mà chỉ số nhận diện ảnh phụ được bao gồm trong tiêu đề của lát, trường địa chỉ đối với lát chỉ báo địa chỉ của lát trong ảnh phụ.

Theo một số phương án thực hiện, phép biến đổi tạo viđeo từ biểu diễn dòng bit. Theo một số phương án thực hiện, phép biến đổi tạo biểu diễn dòng bit từ viđeo.

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

Một số phương án thực hiện sáng chế bao gồm quyết định để vô hiệu hóa

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

Các giải pháp, các ví dụ, các phương án thực hiện, các môđun và các hoạt động chức năng được bộc lộ và khác được mô tả trong tài liệu có thể được triển khai trong hệ mạch điện tử số, hoặc trong phần mềm máy tính, firmware, hoặc phần cứng, bao gồm các cấu trúc được bộc lộ trong tài liệu và các tương đương về mặt cấu trúc, hoặc kết hợp của một hoặc nhiều trong số chúng. Các phương án thực hiện được bộc lộ và các phương án khác có thể được triển khai dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, chẳng hạn, một hoặc nhiều mô đun lệnh chương trình máy tính được mã hóa trên phương tiện máy tính đọc được để thực thi bởi, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Phương tiện máy tính đọc được có thể là thiết bị lưu trữ máy đọc được, nền tảng lưu trữ máy đọc được, thiết bị nhớ, hợp chất thực hiện tạo thành tín hiệu được lan truyền máy đọc được, hoặc tổ hợp của một hoặc nhiều trong số chúng. Cụm từ "thiết bị xử lý dữ liệu" bao gồm tất cả các thiết bị, bộ phận, và máy để xử lý dữ liệu, bao gồm thông qua ví dụ, bộ xử lý lập trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, bên cạnh phần cứng, mã tạo môi trường thực thi cho chương trình máy tính đang chạy, chẳng hạn, mã mà tạo firmware bộ xử lý, xếp chồng giao thức, hệ thống quản trị cơ sở dữ liệu, hệ điều hành, hoặc kết hợp của một hoặc nhiều trong số chúng. Tín hiệu được lan truyền là tín hiệu được nhân tạo, chẳng hạn, tín hiệu điện được tạo bằng máy, quang học, hoặc điện từ, được tạo để mã hóa thông tin để truyền đến bộ phận nhận thích hợp.

Chương trình máy tính (cũng được biết đến như là chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết ở dạng ngôn ngữ lập trình bất kỳ, bao gồm các ngôn ngữ biên dịch hoặc thông dịch, và có thể được triển khai ở dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, trình con, hoặc đơn vị khác thích hợp để sử dụng trong

môi trường tính toán. Chương trình máy tính không nhất thiết tương ứng với tệp tin trong hệ thống tệp tin. Chương trình có thể được lưu trữ trong phần tệp tin chứa các chương trình hoặc dữ liệu khác (chẳng hạn, một hoặc nhiều tệp tin được lưu trữ trong tài liệu ngôn ngữ đánh dấu), trong một tệp tin dành cho chương trình đang chạy, hoặc trong nhiều tệp tin phối hợp (chẳng hạn, các tệp tin mà lưu trữ một hoặc nhiều môđun, chương trình phụ, hoặc các phần mã). Chương trình máy tính có thể được khai triển để được thực thi trên một máy tính hoặc trên nhiều máy tính được đặt ở một điểm hoặc được phân tán giữa nhiều địa điểm và được liên kết bằng mạng truyền thông.

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

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

dụng.

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

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

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

YÊU CẦU BẢO HỘ

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

phân vùng, để biến đổi giữa ảnh của viđeo và dòng bit của viđeo, ảnh thành một hoặc nhiều ảnh phụ; và

thực hiện biến đổi ít nhất dựa trên việc phân vùng,

trong đó dòng bit bao gồm phần cú pháp thứ nhất chỉ báo số lượng của một hoặc nhiều ảnh phụ,

trong đó phần cú pháp thứ nhất được tạo mã sử dụng phương tiện có bộ mô tả ue(v),

trong đó ảnh phụ của một hoặc nhiều ảnh phụ được chỉ báo bằng cách bao gồm phần tử cú pháp thứ hai chỉ báo vị trí nằm ngang của khối cây tạo mã (CTB) trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ ba chỉ báo vị trí thẳng đứng của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ tư chỉ báo chiều rộng của ảnh phụ theo các đơn vị CtbSizeY, và phần tử cú pháp thứ năm chỉ báo chiều cao của ảnh phụ theo các đơn vị CtbSizeY trong dòng bit, và

trong đó CtbSizeY ký hiệu kích thước của CTB.

2. Phương pháp theo điểm 1, trong đó số lượng của một hoặc nhiều ảnh phụ được ký hiệu là N, giá trị của phần cú pháp thứ nhất được ký hiệu là (N-1), và N là số nguyên.

3. Phương pháp theo điểm 1, trong đó giá trị lớn nhất được phép của phần cú pháp thứ nhất đối với ảnh là giá trị cố định.

4. Phương pháp theo điểm 1, trong đó khi ảnh phụ là ảnh phụ thứ nhất trong ảnh, phần tử cú pháp thứ hai và phần tử cú pháp thứ ba không được bao gồm trong dòng bit, và các giá trị của phần tử cú pháp thứ hai và phần tử cú pháp thứ ba được suy ra bằng 0.

5. Phương pháp theo điểm 1, trong đó phần tử cú pháp thứ hai được tạo mã bằng tạo mã độ dài cố định,

trong đó độ dài của phần tử cú pháp thứ hai được dẫn xuất dựa trên Ceil(Log2(V1)), và

trong đó V1 được xác định dựa trên biến liên quan đến chiều rộng của ảnh và CtbSizeY, và

trong đó Ceil(x) ký hiệu số nguyên nhỏ nhất lớn hơn hoặc bằng x.

6. Phương pháp theo điểm 1, trong đó phần tử cú pháp thứ ba được tạo mã bằng tạo mã độ dài cố định,

trong đó độ dài của phần tử cú pháp thứ ba được dẫn xuất dựa trên Ceil(Log2(V2)),

trong đó V2 được xác định dựa trên biến liên quan đến chiều cao của ảnh và CtbSizeY, và

trong đó Ceil(x) ký hiệu số nguyên nhỏ nhất lớn hơn hoặc bằng x.

7. Phương pháp theo điểm 1, trong đó phần tử cú pháp thứ tư được tạo mã bằng tạo mã độ dài cố định,

trong đó độ dài của phần tử cú pháp thứ tư được dẫn xuất dựa trên Ceil(Log2(V1)),

trong đó V1 được xác định dựa trên biến liên quan đến chiều rộng của ảnh và CtbSizeY, và

trong đó Ceil(x) ký hiệu số nguyên nhỏ nhất lớn hơn hoặc bằng x.

8. Phương pháp theo điểm 1, trong đó phần tử cú pháp thứ năm được tạo mã bằng tạo mã độ dài cố định,

trong đó độ dài của phần tử cú pháp thứ năm được dẫn xuất dựa trên Ceil(Log2(V2)),

trong đó V2 được xác định dựa trên biến liên quan đến chiều cao của ảnh và CtbSizeY, và

trong đó Ceil(x) ký hiệu số nguyên nhỏ nhất lớn hơn hoặc bằng x.

9. Phương pháp theo điểm 1, trong đó phần cú pháp thứ nhất, phần tử cú pháp thứ hai, phần tử cú pháp thứ ba, phần tử cú pháp thứ tư, và phần tử cú pháp thứ năm được bao gồm trong dòng bit sau khi phần tử cú pháp thứ sáu được bao gồm, và

trong đó giá trị của phần tử cú pháp thứ sáu cộng 5 liên quan đến kích thước đơn vị cây tạo mã.

10. Phương pháp theo điểm 9, trong đó cờ subpicture-treated-as-picture chỉ báo liệu ảnh phụ được xử lý dưới dạng ảnh trong quá trình loại bỏ các hoạt động lọc trong vòng được bao gồm trong dòng bit sau phần cú pháp thứ nhất, phần tử cú pháp thứ hai, phần tử cú pháp thứ ba, phần tử cú pháp thứ tư, và phần tử cú pháp thứ năm được bao gồm.

11. Phương pháp theo điểm 9, trong đó cờ loop-filter-across-subpicture-enabled chỉ báo liệu các hoạt động lọc trong vòng có thể được thực hiện giữa các đường biên của ảnh phụ được bao gồm trong dòng bit sau khi phần cú pháp thứ nhất, phần tử cú pháp thứ hai, phần tử cú pháp thứ ba, phần tử cú pháp thứ tư, và phần tử cú pháp thứ năm được bao gồm.

12. Phương pháp theo điểm 1, trong đó phương tiện có bộ mô tả ue(v) là phương tiện tạo mã Exp-Golomb bậc thứ 0.

13. Phương pháp theo điểm 1, trong đó phép biến đổi bao gồm bước mã hóa ảnh thành dòng bit.

14. Phương pháp theo điểm 1, trong đó phép biến đổi bao gồm bước giải mã ảnh từ dòng bit.

15. Thiết bị xử lý dữ liệu viđeo bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó, trong đó các lệnh khi thực thi bằng bộ xử lý, khiến bộ xử lý:

phân vùng, để biến đổi giữa ảnh của viđeo và dòng bit của viđeo, ảnh thành một hoặc nhiều ảnh phụ; và

thực hiện phép biến đổi ít nhất dựa trên the phân vùng,

trong đó dòng bit bao gồm phần cú pháp thứ nhất chỉ báo a số lượng của một hoặc nhiều ảnh phụ,

trong đó phần cú pháp thứ nhất được tạo mã sử dụng phương tiện có bộ mô tả ue(v),

trong đó ảnh phụ của một hoặc nhiều ảnh phụ được chỉ báo bằng cách bao gồm phần tử cú pháp thứ hai chỉ báo vị trí nằm ngang của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ ba chỉ báo vị trí thẳng đứng của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ tư chỉ báo chiều rộng của ảnh phụ theo các đơn vị CtbSizeY, và phần tử cú pháp

thứ năm chỉ báo chiều cao của ảnh phụ theo các đơn vị CtbSizeY trong dòng bit, và

trong đó CtbSizeY ký hiệu kích thước của CTB.

16. Thiết bị theo điểm 15, trong đó số lượng của một hoặc nhiều ảnh phụ được ký hiệu là N, giá trị của phần cú pháp thứ nhất được ký hiệu là (N-1), và N là số nguyên.

17. Vật ghi máy tính đọc được bất biến lưu trữ các lệnh khiến bộ xử lý:

phân vùng, để biến đổi giữa ảnh của viđeo và dòng bit của viđeo, ảnh thành một hoặc nhiều ảnh phụ; và

thực hiện phép biến đổi ít nhất dựa trên bước phân vùng,

trong đó dòng bit bao gồm phần cú pháp thứ nhất chỉ báo số lượng của một hoặc nhiều ảnh phụ,

trong đó phần cú pháp thứ nhất được tạo mã sử dụng phương tiện có bộ mô tả ue(v),

trong đó ảnh phụ của một hoặc nhiều ảnh phụ được chỉ báo bằng cách bao gồm phần tử cú pháp thứ hai chỉ báo vị trí nằm ngang của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ ba chỉ báo vị trí thẳng đứng của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ tư chỉ báo chiều rộng của ảnh phụ theo các đơn vị CtbSizeY, và phần tử cú pháp thứ năm chỉ báo chiều cao của ảnh phụ theo các đơn vị CtbSizeY trong dòng bit, và

trong đó CtbSizeY ký hiệu kích thước của CTB.

18. Vật ghi máy tính đọc được bất biến lưu trữ dòng bit của video được tạo bỉ phương pháp được thực hiện bởi thiết bị xử lý viđeo, trong đó phương pháp bao gồm các bước:

phân vùng, đối với ảnh của viđeo, ảnh thành một hoặc nhiều ảnh phụ; và

tạo dòng bit ít nhất dựa trên phân vùng,

trong đó dòng bit bao gồm phần cú pháp thứ nhất chỉ báo số lượng của một hoặc nhiều ảnh phụ,

trong đó phần cú pháp thứ nhất được tạo mã sử dụng phương tiện có bộ mô tả ue(v),

trong đó ảnh phụ của một hoặc nhiều ảnh phụ được chỉ báo bằng cách bao gồm phần tử cú pháp thứ hai chỉ báo vị trí nằm ngang của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ ba chỉ báo vị trí thẳng đứng của CTB trên bên trái của ảnh phụ theo đơn vị CtbSizeY, phần tử cú pháp thứ tư chỉ báo chiều rộng của ảnh phụ theo các đơn vị CtbSizeY, và phần tử cú pháp thứ năm chỉ báo chiều cao của ảnh phụ theo các đơn vị CtbSizeY in dòng bit, và

trong đó CtbSizeY ký hiệu kích thước của CTB.

19. Vật ghi máy tính đọc được bất biến theo điểm 17, trong đó số lượng của một hoặc nhiều ảnh phụ được ký hiệu là N, giá trị của phần cú pháp thứ nhất được ký hiệu là (N-1), và N là số nguyên.

20. Vật ghi máy tính đọc được bất biến theo điểm 18, trong đó số lượng của một hoặc nhiều ảnh phụ được ký hiệu là N, giá trị của phần cú pháp thứ nhất được ký hiệu là (N-1), và N là số nguyên.

1/19

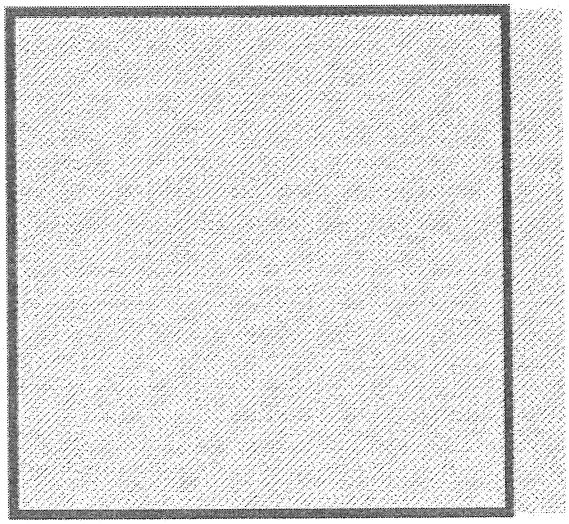

Fig.1

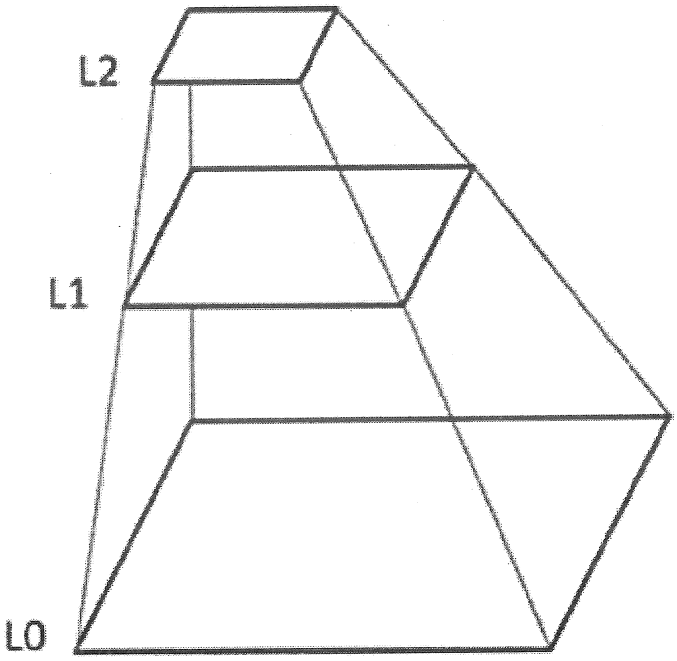

Fig.2

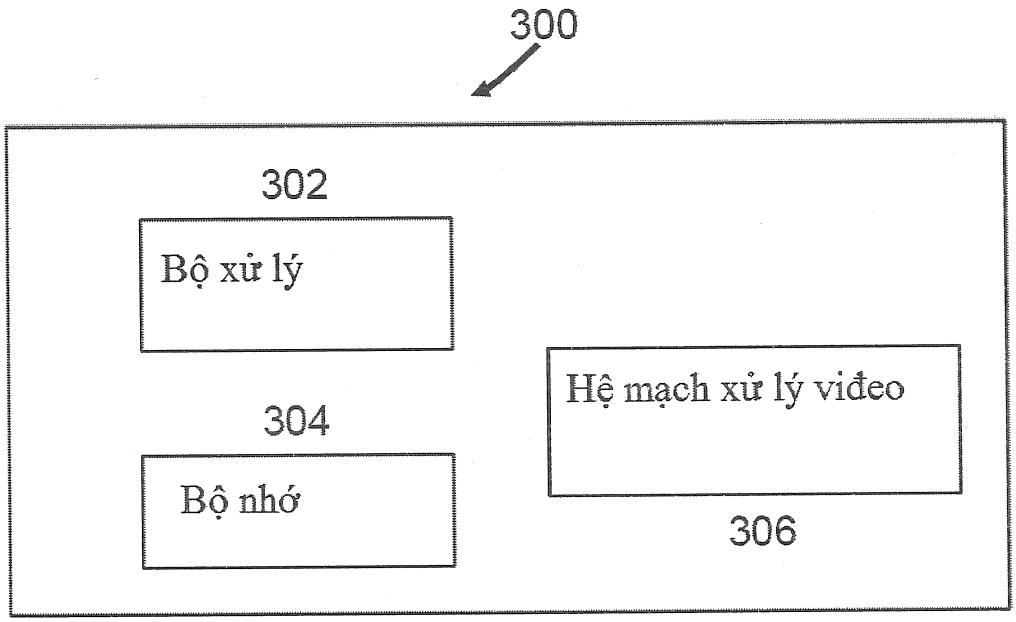

Fig.3

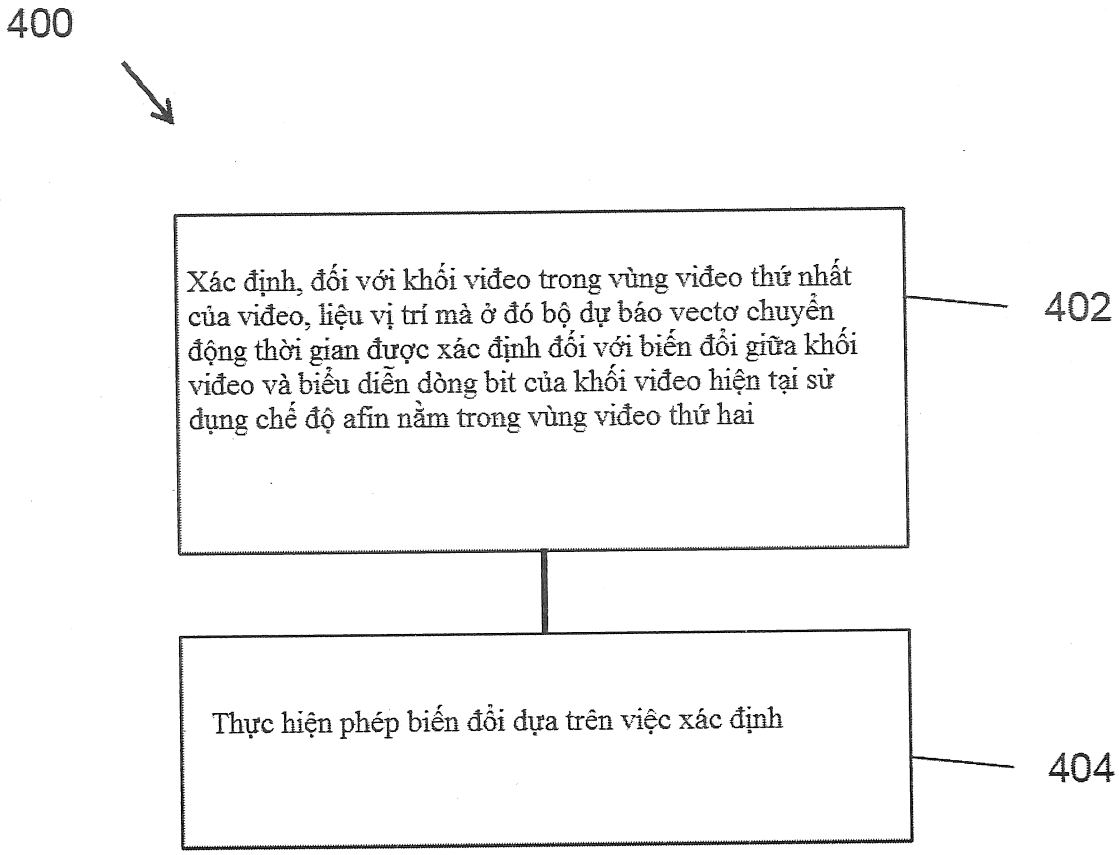
400
Xác định, đối với khối viđeo trong vùng viđeo thứ nhất của viđeo, liệu vị trí mà ở đó bộ dự báo vectơ chuyển động thời gian được xác định đối với biến đổi giữa khối viđeo và biểu diễn dòng bit của khối viđeo hiện tại sử dụng chế độ afin nằm trong vùng viđeo thứ hai
402
Thực hiện phép biến đổi dựa trên việc xác định
404

Fig.4

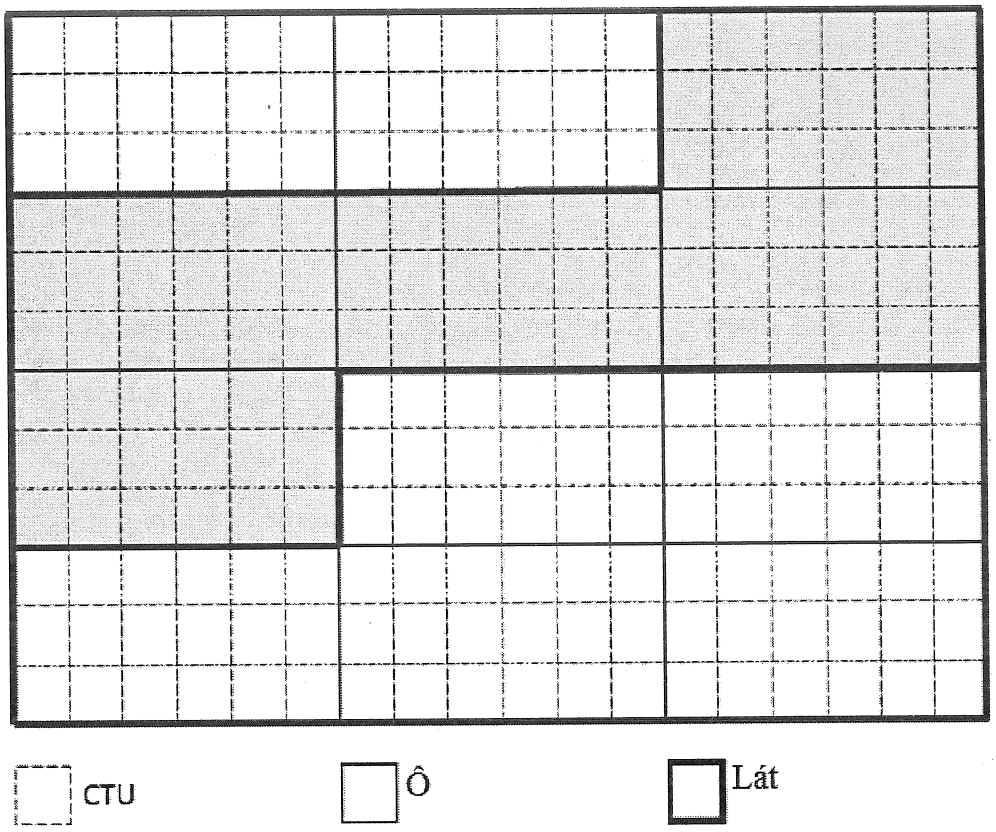

Fig.5

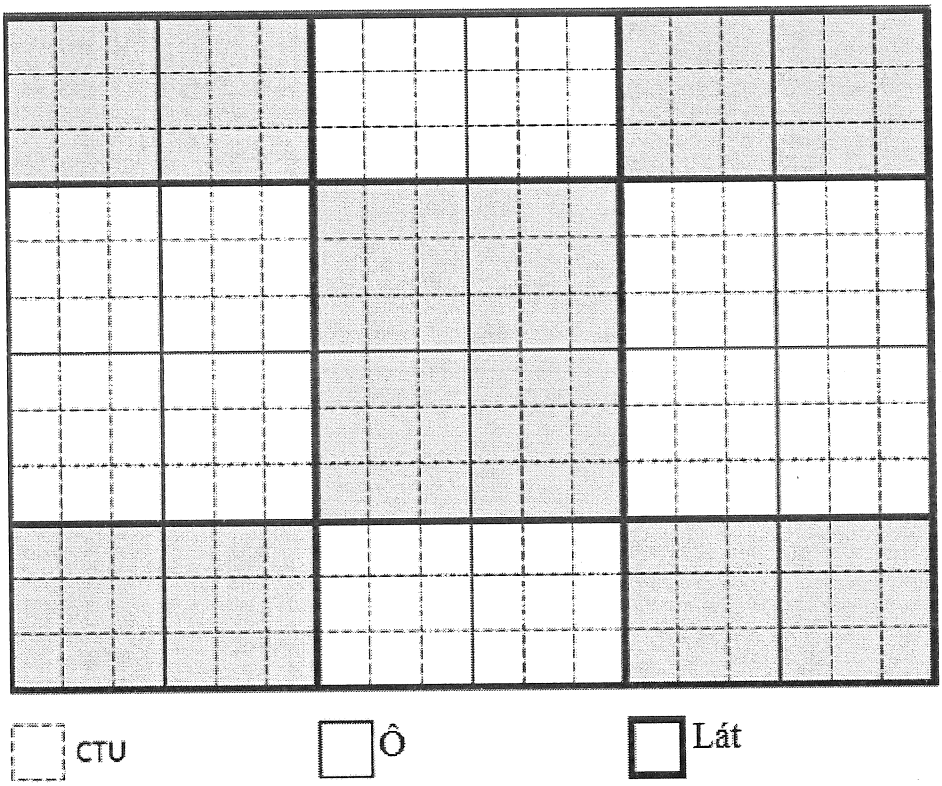

Fig.6

7/19

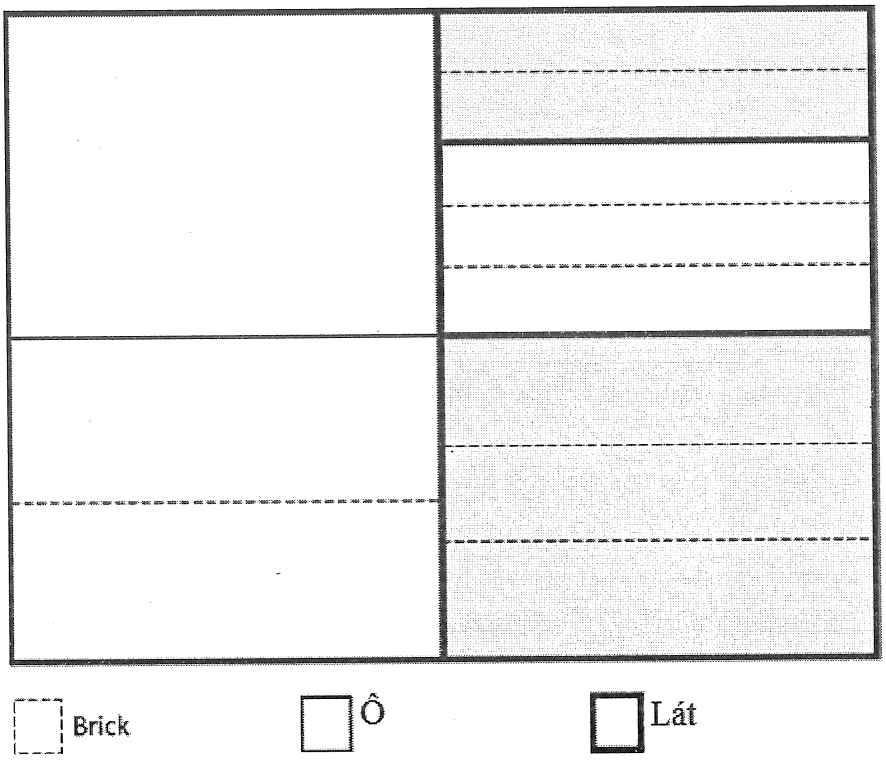

Fig.7

8/19

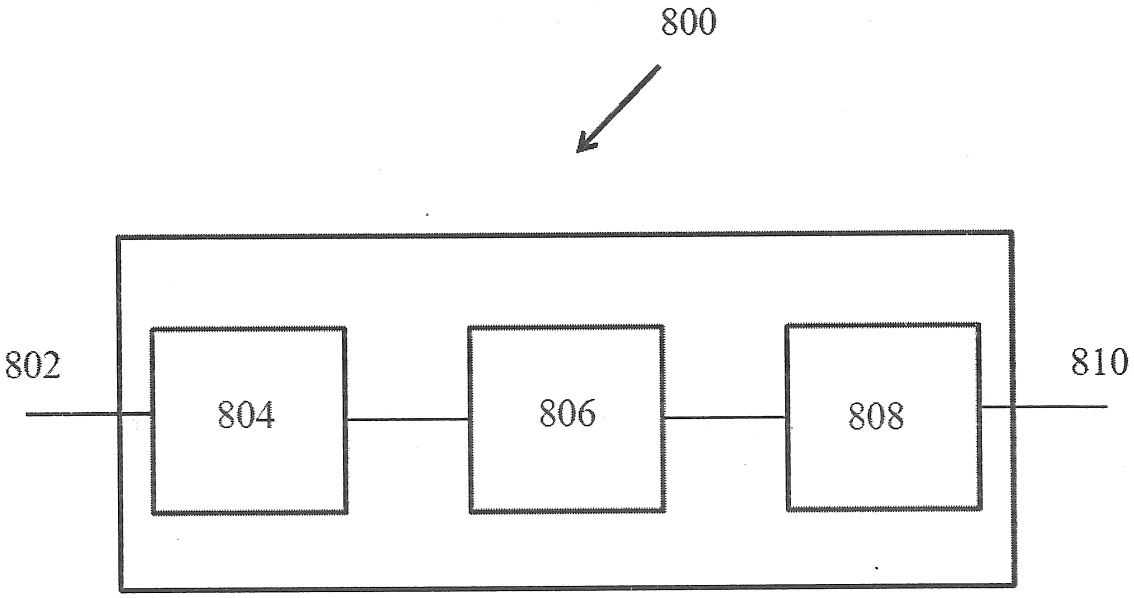

Fig.8

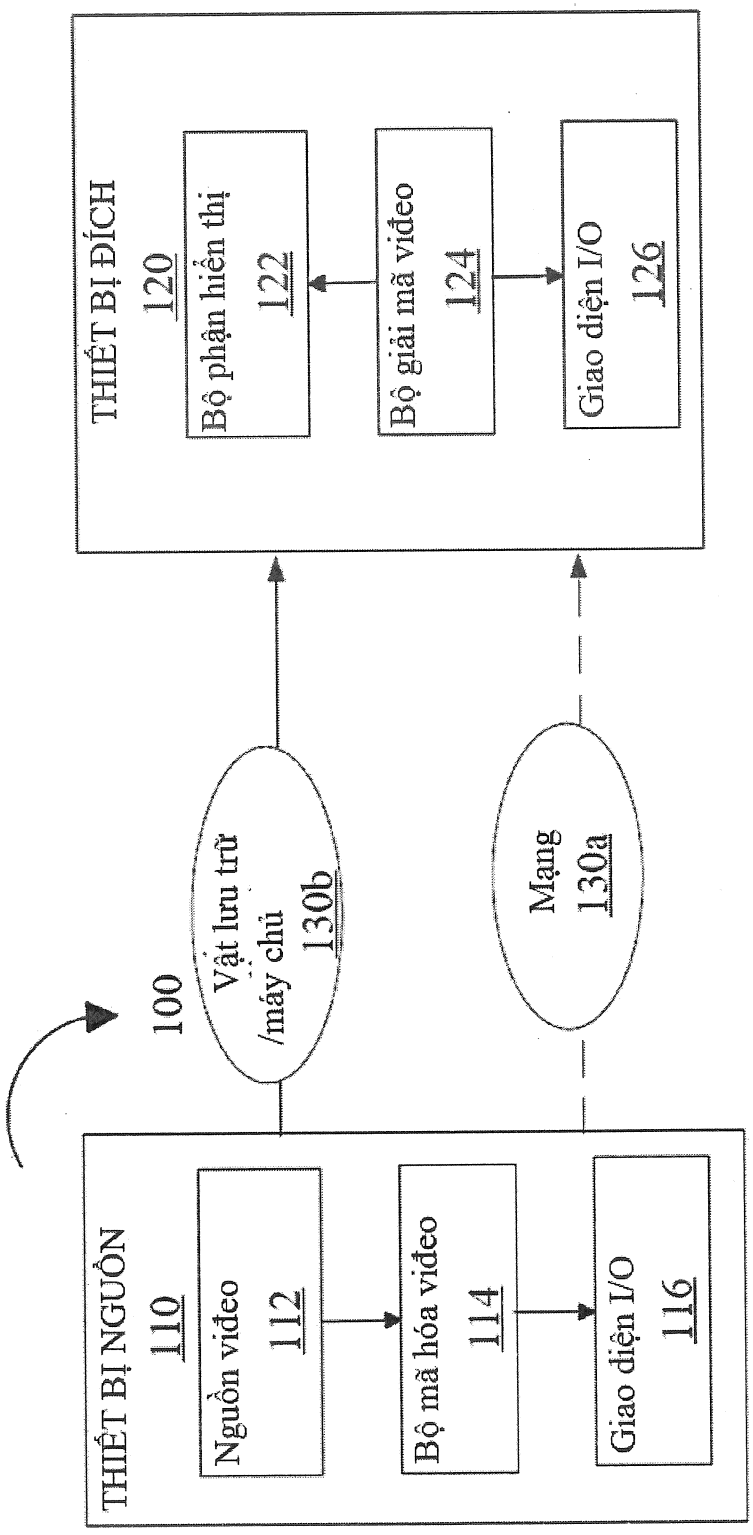
THIẾT BỊ ĐÍCH
120
Bộ phận hiển thị
122
Bộ giải mã video
124
Giao diện I/O
126
100
Vật lưu trữ
/máy chủ
130b
Mạng
130a
THIẾT BỊ NGUỒN
110
Nguồn video
112
Bộ mã hóa video
114
Giao diện I/O
116

Fig.9

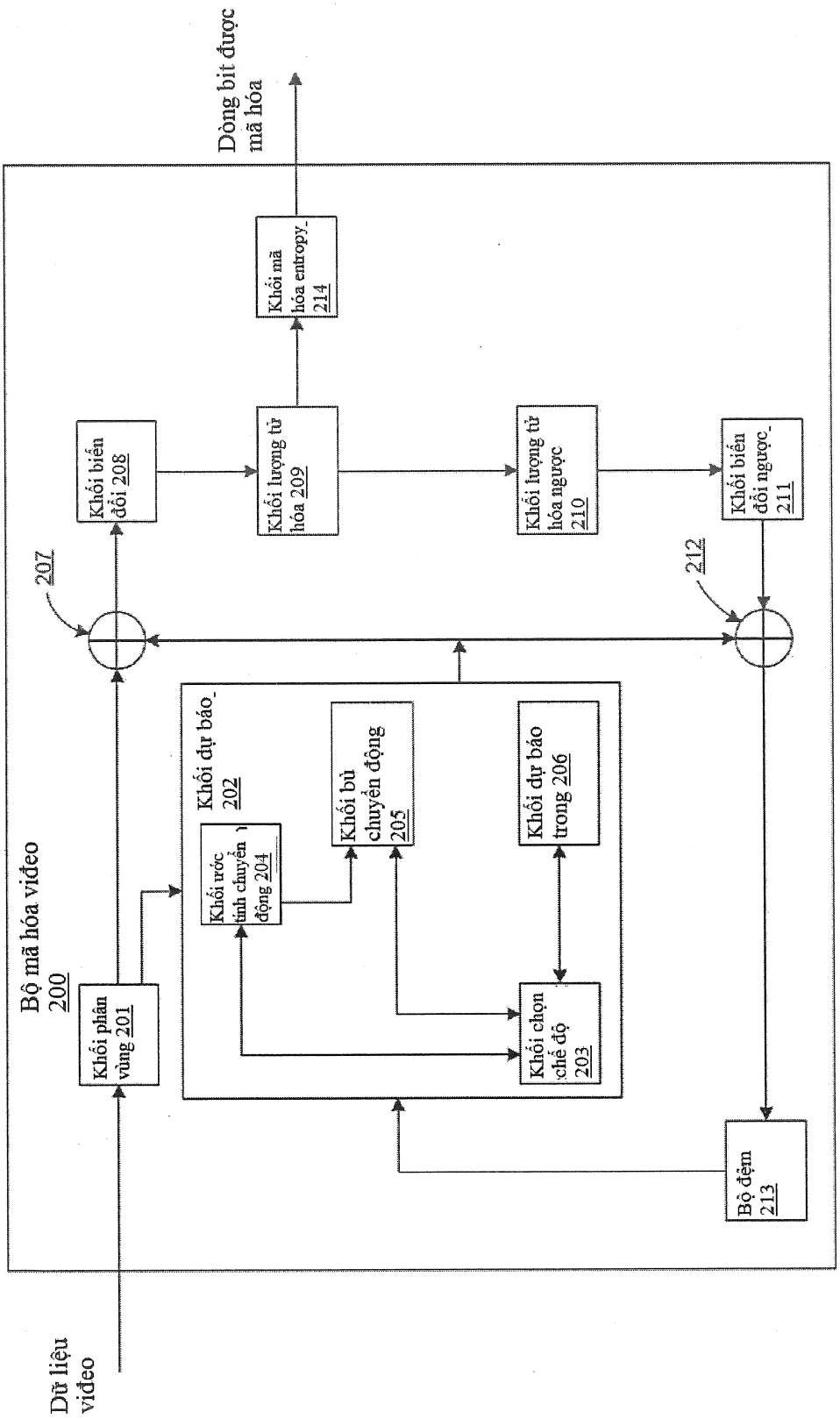
Dòng bit được mã hóa
Khối mã hóa entropy 214
Khối biến đổi 208
Khối lượng tử hóa 209
Khối lượng tử hóa ngược 210
Khối biến đổi ngược 211
207
212
Khối dự báo 202
Khối ước tính chuyển động 204
Khối bù chuyển động 205
Khối dự báo trong 206
Khối chọn chế độ 203
Bộ mã hóa video 200
Khối phân vùng 201
Bộ đệm 213
Dữ liệu video

Fig.10

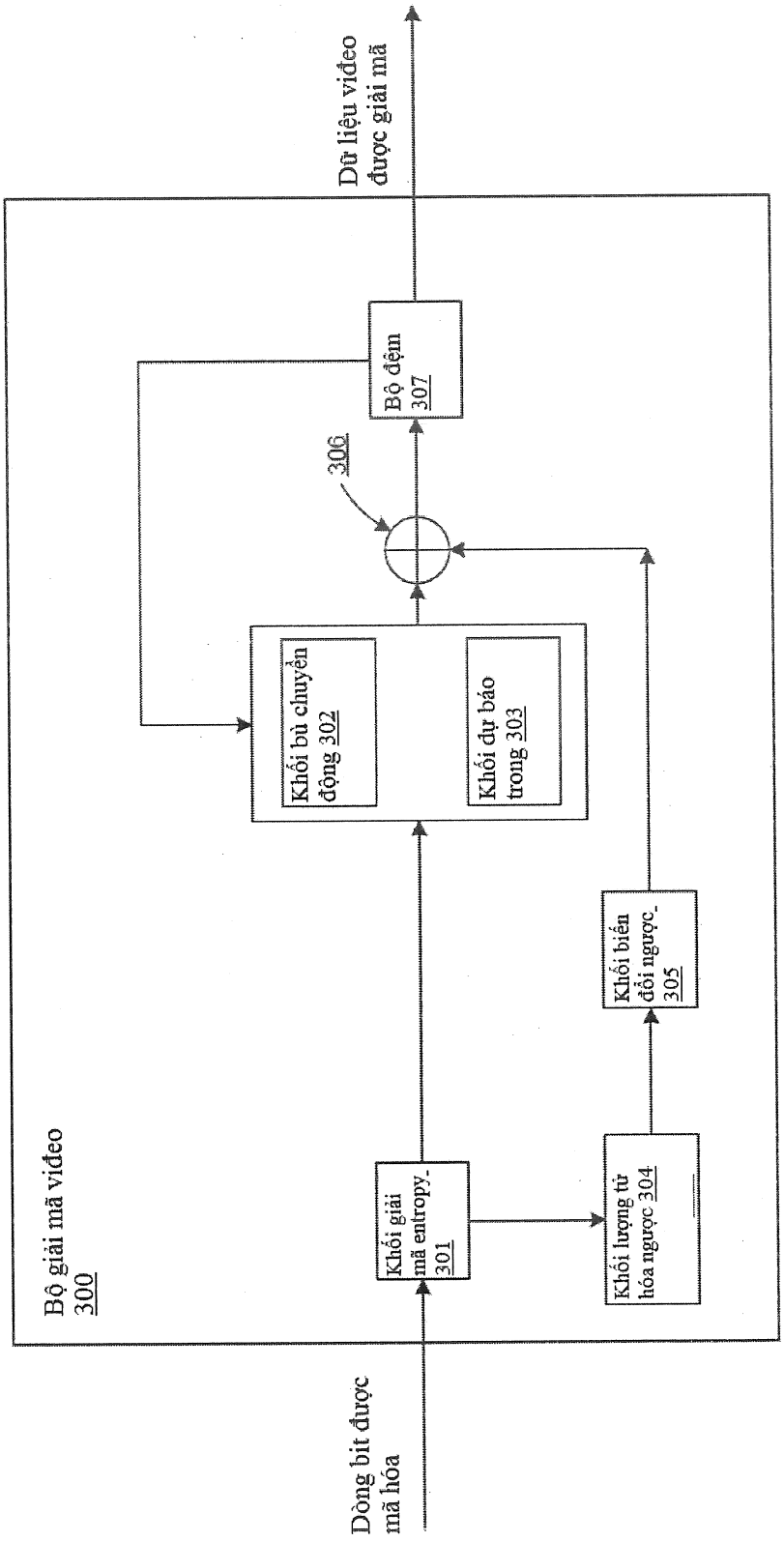
Dòng bit được mã hóa
Bộ giải mã video 300
Khối giải mã entropy- 301
Khối lượng tử hóa ngược 304
Khối biến đổi ngược- 305
Khối bù chuyển động 302
Khối dự báo trong 303
306
Bộ đệm 307
Dữ liệu video được giải mã

Fig.11

12/19

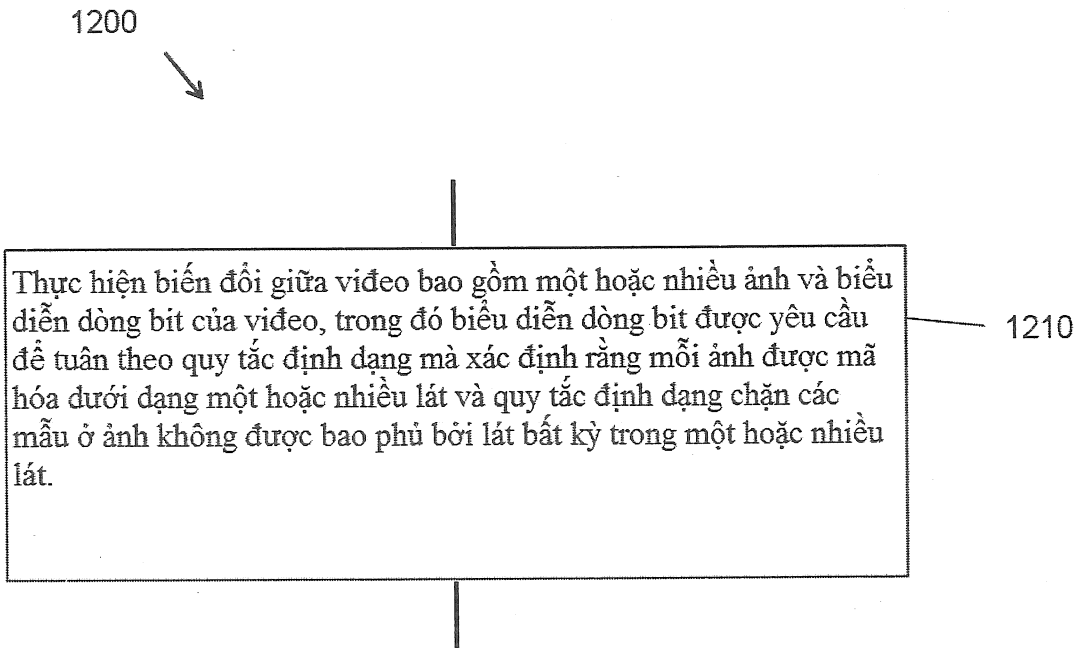
1200
Thực hiện biến đổi giữa viđeo bao gồm một hoặc nhiều ảnh và biểu diễn dòng bit của viđeo, trong đó biểu diễn dòng bit được yêu cầu để tuân theo quy tắc định dạng mà xác định rằng mỗi ảnh được mã hóa dưới dạng một hoặc nhiều lát và quy tắc định dạng chặn các mẫu ở ảnh không được bao phủ bởi lát bất kỳ trong một hoặc nhiều lát.
1210

Fig.12

13/19

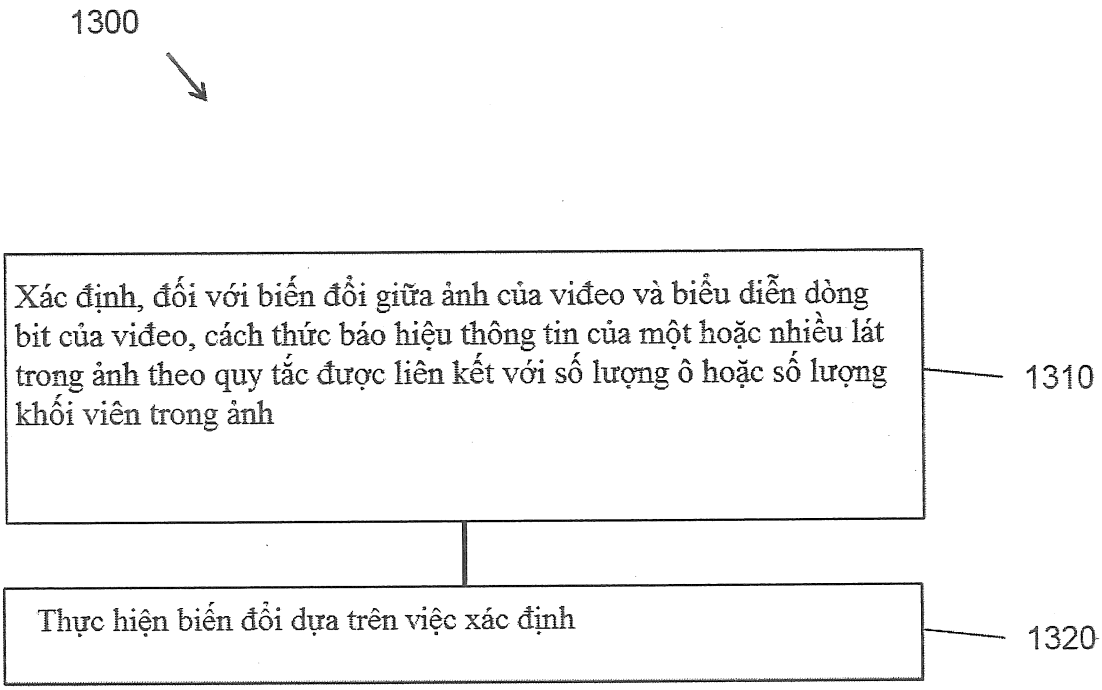
1300
Xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, cách thức báo hiệu thông tin của một hoặc nhiều lát trong ảnh theo quy tắc được liên kết với số lượng ô hoặc số lượng khối viên trong ảnh
1310
Thực hiện biến đổi dựa trên việc xác định
1320

Fig.13

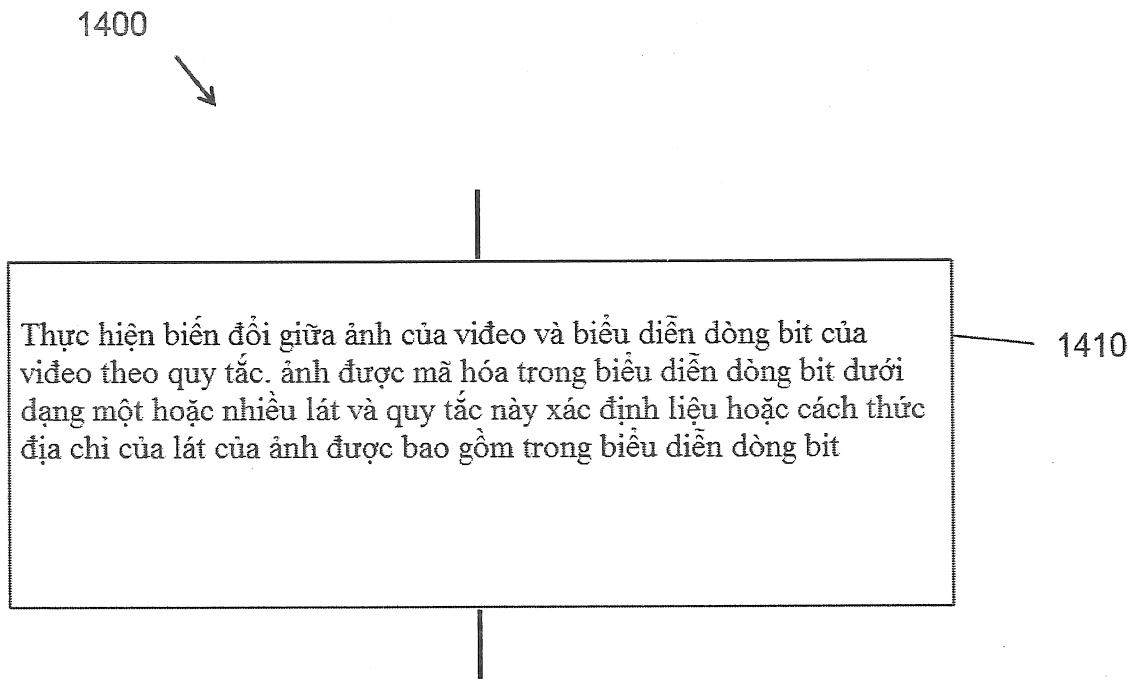
1400
Thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo theo quy tắc. ảnh được mã hóa trong biểu diễn dòng bit dưới dạng một hoặc nhiều lát và quy tắc này xác định liệu hoặc cách thức địa chỉ của lát của ảnh được bao gồm trong biểu diễn dòng bit
1410

Fig.14

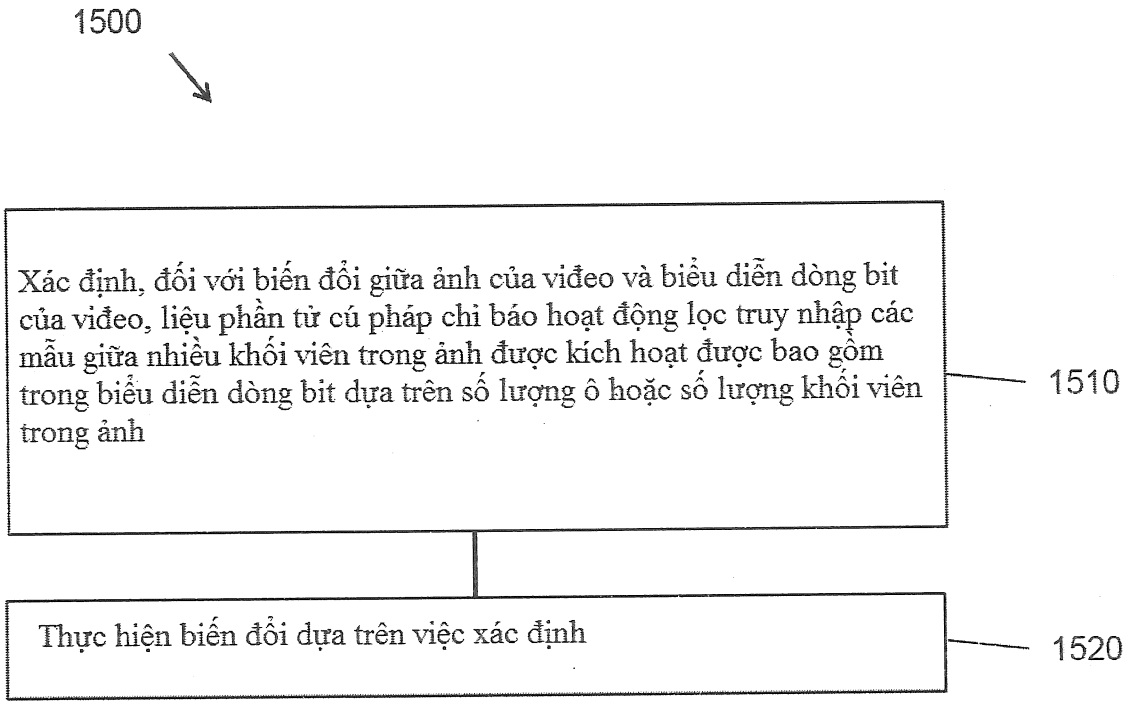
1500
Xác định, đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, liệu phần tử cú pháp chỉ báo hoạt động lọc truy nhập các mẫu giữa nhiều khối viên trong ảnh được kích hoạt được bao gồm trong biểu diễn dòng bit dựa trên số lượng ô hoặc số lượng khối viên trong ảnh
1510
Thực hiện biến đổi dựa trên việc xác định
1520

Fig.15

16/19

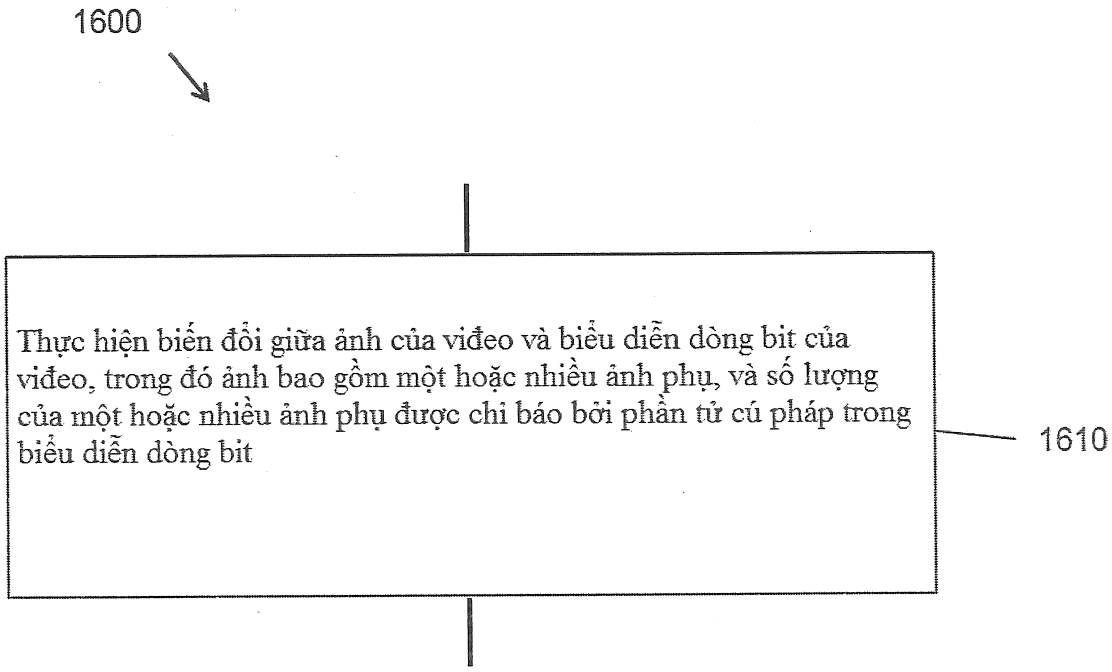
1600
Thực hiện biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo, trong đó ảnh bao gồm một hoặc nhiều ảnh phụ, và số lượng của một hoặc nhiều ảnh phụ được chỉ báo bởi phần tử cú pháp trong biểu diễn dòng bit
1610

Fig.16

17/19

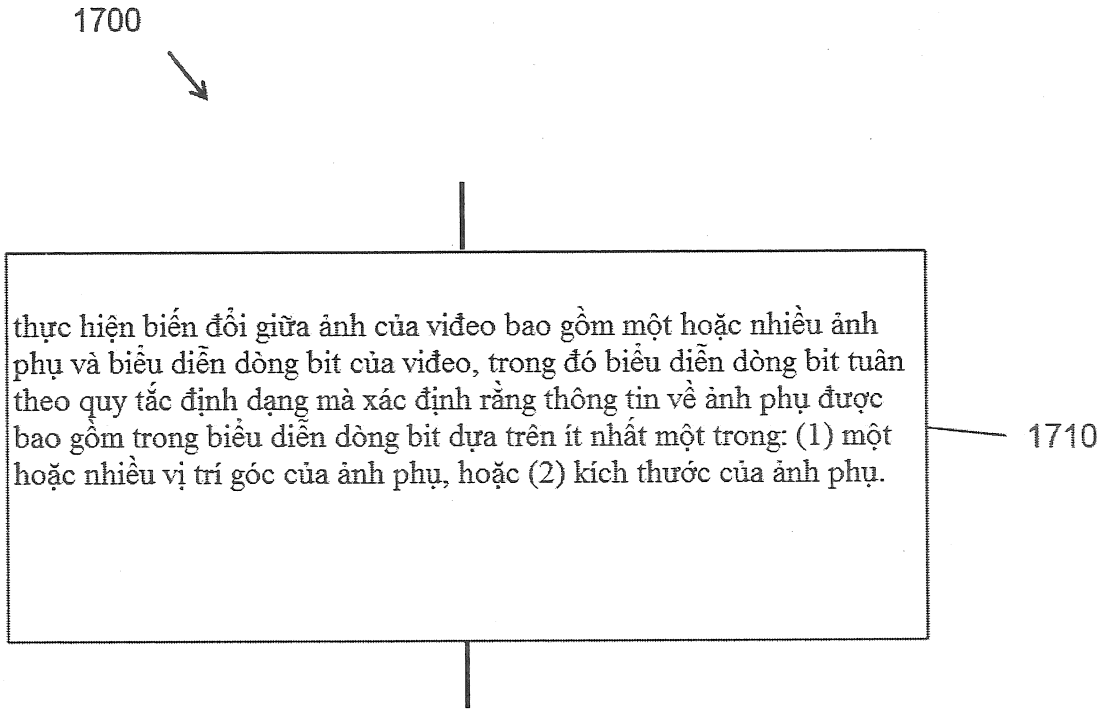
1700
thực hiện biến đổi giữa ảnh của viđeo bao gồm một hoặc nhiều ảnh phụ và biểu diễn dòng bit của viđeo, trong đó biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng thông tin về ảnh phụ được bao gồm trong biểu diễn dòng bit dựa trên ít nhất một trong: (1) một hoặc nhiều vị trí góc của ảnh phụ, hoặc (2) kích thước của ảnh phụ.
1710

Fig.17

18/19

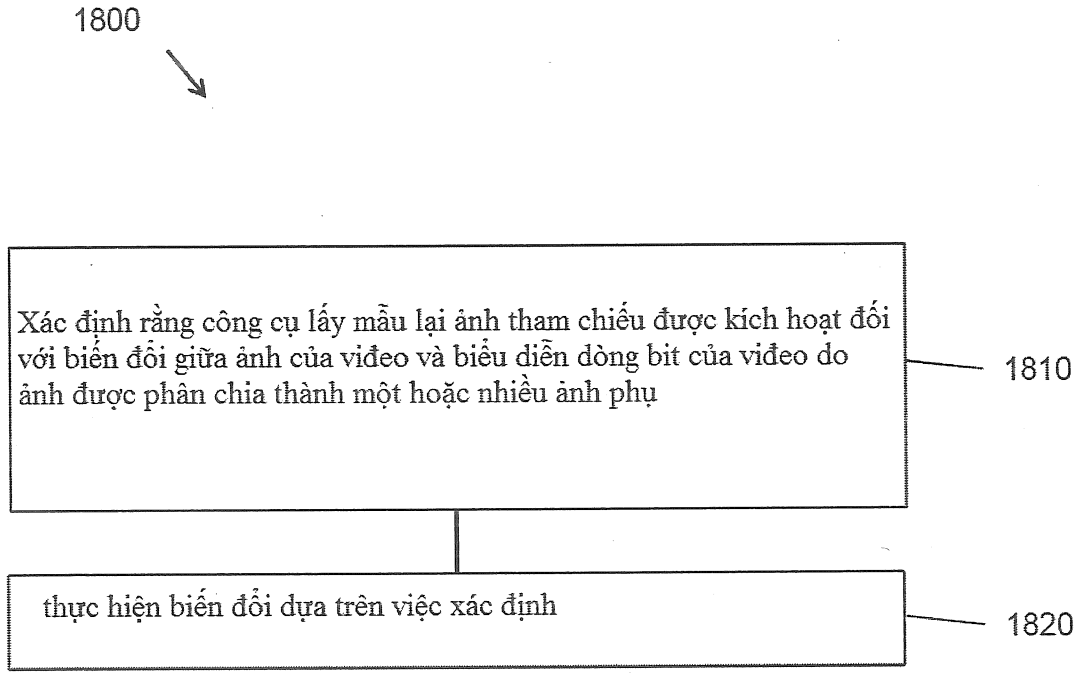
1800
Xác định rằng công cụ lấy mẫu lại ảnh tham chiếu được kích hoạt đối với biến đổi giữa ảnh của viđeo và biểu diễn dòng bit của viđeo do ảnh được phân chia thành một hoặc nhiều ảnh phụ
1810
thực hiện biến đổi dựa trên việc xác định
1820

Fig.18

19/19

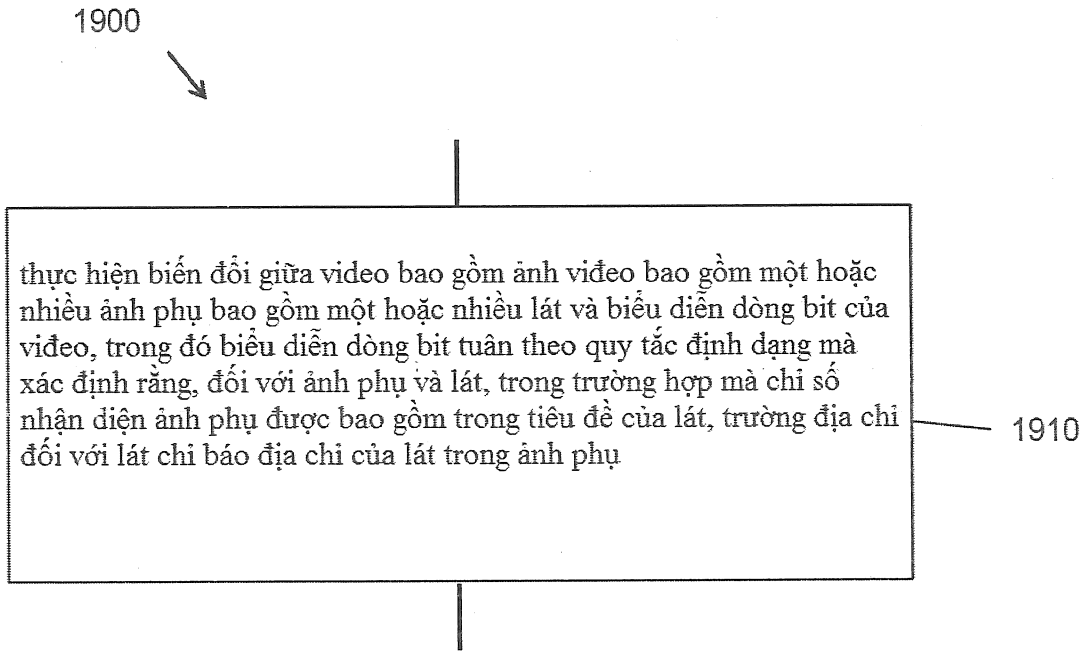
1900
thực hiện biến đổi giữa video bao gồm ảnh viđeo bao gồm một hoặc nhiều ảnh phụ bao gồm một hoặc nhiều lát và biểu diễn dòng bit của viđeo, trong đó biểu diễn dòng bit tuân theo quy tắc định dạng mà xác định rằng, đối với ảnh phụ và lát, trong trường hợp mà chỉ số nhận diện ảnh phụ được bao gồm trong tiêu đề của lát, trường địa chỉ đối với lát chỉ báo địa chỉ của lát trong ảnh phụ
1910

Fig.19