



(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)



1-0048240

(51)^{2006.01} H04N 7/12; H04N 7/46; H04N 7/32

(13) B

(21) 1-2021-06218

(22) 11/03/2020

(86) PCT/US2020/022185 11/03/2020

(87) WO2020/185962 17/09/2020

(30) 62/816,722 11/03/2019 US; 62/871,020 05/07/2019 US

(45) 25/07/2025 448

(43) 27/12/2021 405A

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

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

(72) WANG, Ye-Kui (US); HENDRY, Fnu (ID); CHEN, Jianle (CN).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ LUỒNG BIT VIDEO ĐÃ ĐƯỢC MÃ HÓA, THIẾT BỊ
GIẢI MÃ VÀ HỆ THỐNG TẠO MÃ

(21) 1-2021-06218

(57) Sáng chế đề cập đến bộ mã hóa, bộ giải mã, thiết bị và hệ thống tạo mã, phương tiện để tạo mã và phương pháp giải mã luồng bit video đã được tạo mã được thực hiện bởi bộ giải mã video. Phương pháp bao gồm các bước: bộ giải mã video xác định xem liệu giá trị cho cờ thứ nhất có được cung cấp bởi đầu vào bên ngoài hay không; thiết đặt cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng giá trị của cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài; giải mã ảnh GDR; và lưu trữ ảnh GDR trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB).

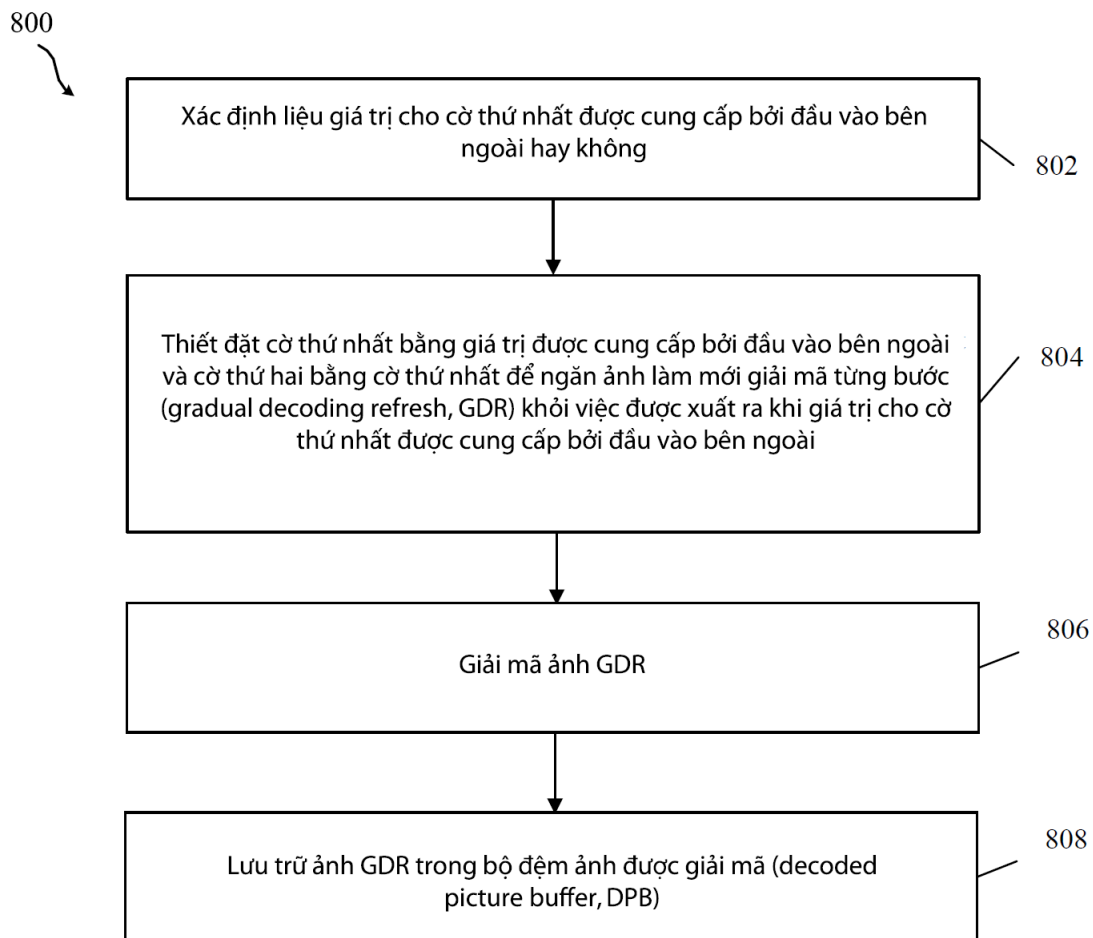


FIG. 8

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế mô tả các kỹ thuật hỗ trợ việc làm mới giải mã từng bước trong việc tạo mã video. Cụ thể hơn, sáng chế cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng các ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP).

Tình trạng kỹ thuật của sáng chế

Lượng dữ liệu video cần để thể hiện ngay cả một video tương đối ngắn cũng có thể là lớn, điều này có thể dẫn đến những khó khăn khi dữ liệu này cần được tạo luồng hoặc được truyền thông qua mạng truyền thông có dung lượng băng thông bị giới hạn. Do đó, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông hiện đại ngày nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng tại nguồn để tạo mã dữ liệu video trước khi phát hoặc lưu trữ, nhờ đó làm giảm lượng dữ liệu cần để biểu diễn các hình ảnh video kỹ thuật số. Sau đó, dữ liệu được nén này được nhận tại đích đến bởi thiết bị giải nén video mà giải mã dữ liệu video đó. Với các tài nguyên mạng bị giới hạn và các nhu cầu ngày càng tăng về chất lượng video cao hơn, thì mong muốn là có các kỹ thuật nén và giải nén được cải thiện để cải thiện tỷ số nén mà chỉ phải hy sinh ít hoặc không phải hy sinh chất lượng hình ảnh.

US 2016/112712 A1 bộc lộ các công nghệ để hỗ trợ sự truy cập ngẫu nhiên trong các luồng video nén.

"SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video: High efficiency video coding H.265", SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, CH, (20150401), pages 1 - 610, URL: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.265-

201504-I!!!PDF-E&type=items, (20151028), bộ lộ HandleCraAsBlaFlag mà giá trị có thể được cung cấp bởi phương tiện bên ngoài.

WO 2013/163569 A1 bộ lộ các công nghệ để cung cấp sự chỉ báo, cho mỗi ảnh CRA, mà chỉ báo xem các tập hợp thông số chứa trong các đơn vị truy cập trước đó có cần thiết để thực hiện truy cập ngẫu nhiên từ ảnh hay không.

Bản chất kỹ thuật của sáng chế

Sáng chế này được trình bày bằng tập hợp các yêu cầu bảo hộ kèm theo. Dưới đây, các phần mô tả và hình vẽ tham chiếu đến các phương án thực hiện, các ví dụ hoặc các cách thực hiện, mà không được bao quát bởi các điểm yêu cầu bảo hộ không được trình bày như các phương án thực hiện sáng chế, mà được xem là các ví dụ minh họa hữu ích cho việc hiểu sáng chế này.

Khía cạnh thứ nhất đề cập đến phương pháp giải mã luồng bit video đã được tạo mã được thực hiện bởi bộ giải mã video. Phương pháp bao gồm các bước: xác định, bởi bộ giải mã video, xem liệu giá trị cho cờ thứ nhất có được cung cấp bởi đầu vào bên ngoài hay không; thiết đặt, bởi bộ giải mã video, cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) trong luồng bit video đã được tạo mã khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài; giải mã, bởi bộ giải mã video, ảnh GDR; và lưu trữ, bởi bộ giải mã video, ảnh GDR trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB).

Phương pháp cung cấp các kỹ thuật mà cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. đầu vào bên ngoài có thể là, ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh

GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khởi việc được xuất ra ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Trong dạng thực hiện thứ nhất của phương pháp theo khía cạnh thứ nhất như vậy, cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài.

Trong dạng thực hiện thứ hai của phương pháp theo khía cạnh thứ nhất như vậy hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, đầu vào bên ngoài là giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video, và giá trị của cờ thứ nhất được cung cấp bởi người dùng của bộ giải mã video sử dụng đầu vào bên ngoài.

Trong dạng thực hiện thứ ba của phương pháp theo khía cạnh thứ nhất như vậy hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, cờ thứ nhất được chỉ định là `HandleGdrAsCvsStartFlag`.

Trong dạng thực hiện thứ tư của phương pháp theo khía cạnh thứ nhất như vậy hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, giá trị của cờ thứ nhất và cờ thứ hai được thiết đặt là một (1) để ngăn ảnh GDR và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra.

Trong dạng thực hiện thứ năm của phương pháp theo khía cạnh thứ nhất như vậy hoặc dạng thực hiện trước đó bất kỳ của khía cạnh thứ nhất, giá trị của cờ thứ nhất được thiết đặt là không (0) khi giá trị cho cờ thứ nhất không được cung cấp bởi đầu vào bên ngoài.

Khía cạnh thứ hai đề cập đến thiết bị giải mã. Thiết bị giải mã chứa bộ nhận được tạo cấu hình để nhận luồng bit video đã được tạo mã; bộ nhớ được ghép nối với bộ nhận, bộ nhớ lưu trữ các lệnh; và bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh để làm cho thiết bị giải mã: xác định xem liệu giá trị cho cờ

thứ nhất có được cung cấp bởi đầu vào bên ngoài hay không; thiết đặt cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài; giải mã ảnh GDR; và lưu trữ ảnh GDR trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB).

Thiết bị giải mã cung cấp các kỹ thuật mà cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. đầu vào bên ngoài có thể be, ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Việc thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khỏi việc được xuất ra ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Theo dạng thực hiện thứ nhất của thiết bị giải mã theo khía cạnh thứ hai như vậy, cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài.

Theo dạng thực hiện thứ hai của thiết bị giải mã theo khía cạnh thứ hai như vậy, đầu vào bên ngoài là giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video, và giá trị của cờ thứ nhất được cung cấp bởi người dùng của bộ giải mã video sử dụng đầu vào bên ngoài.

Theo dạng thực hiện thứ ba của thiết bị giải mã theo khía cạnh thứ hai như vậy, cờ thứ nhất được chỉ định là `HandleGdrAsCvsStartFlag`.

Theo dạng thực hiện thứ tư của thiết bị giải mã theo khía cạnh thứ hai như vậy, giá trị của cờ thứ nhất và cờ thứ hai được thiết đặt là một (1) để ngăn ảnh GDR và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra.

Theo dạng thực hiện thứ năm của thiết bị giải mã theo khía cạnh thứ hai như vậy, giá trị của cờ thứ nhất được thiết đặt là không (0) khi giá trị cho cờ thứ nhất không được cung cấp bởi đầu vào bên ngoài.

Khía cạnh thứ ba đề cập đến thiết bị tạo mã. Thiết bị tạo mã chứa bộ nhận được tạo cấu hình để nhận ảnh để mã hóa hoặc để nhận luồng bit để giải mã; bộ truyền được ghép nối với bộ nhận, bộ truyền được tạo cấu hình để truyền luồng bit đến bộ giải mã hoặc để truyền hình ảnh được giải mã đến bộ phận hiển thị; bộ nhớ được ghép nối với ít nhất một bộ trong số bộ nhận hoặc bộ truyền, bộ nhớ được tạo cấu hình để lưu trữ các lệnh; và bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong bộ nhớ để thực hiện phương pháp bất kỳ trong số các phương pháp được bộc lộ ở đây.

Thiết bị tạo mã cung cấp các kỹ thuật mà cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. Đầu vào bên ngoài có thể là, ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Việc thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khỏi việc được xuất ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề

thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Theo dạng thực hiện thứ nhất của thiết bị tạo mã theo khía cạnh thứ ba như vậy, thì còn bao gồm bộ hiển thị được tạo cấu hình để hiển thị hình ảnh.

Khía cạnh thứ tư đề cập đến hệ thống. Hệ thống bao gồm bộ mã hóa; và bộ giải mã giao tiếp với bộ mã hóa, trong đó bộ mã hóa hoặc bộ giải mã bao gồm thiết bị giải mã, thiết bị mã hóa, hoặc thiết bị tạo mã được bộc lộ ở đây.

Hệ thống cung cấp các kỹ thuật mà cho phép việc làm mới nội tạng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. đầu vào bên ngoài có thể là, ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Việc thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khỏi việc được xuất ra ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Khía cạnh thứ năm đề cập đến phương tiện để tạo mã. Phương tiện để tạo mã chứa phương tiện nhận được tạo cấu hình để nhận ảnh để mã hóa hoặc để nhận luồng bit để giải mã; phương tiện truyền được ghép nối với phương tiện nhận, phương tiện truyền được tạo cấu hình để truyền luồng bit đến phương tiện giải mã hoặc để truyền hình ảnh được giải mã đến phương tiện hiển thị; phương tiện lưu trữ được ghép nối với ít nhất một phương tiện trong số phương tiện nhận hoặc phương tiện truyền, phương tiện lưu trữ được tạo cấu hình để lưu trữ các lệnh; và phương tiện xử lý được ghép nối với phương tiện lưu trữ, phương tiện xử lý được tạo cấu hình để thực hiện các lệnh được lưu trữ trong phương

tiện lưu trữ để thực hiện phương pháp bất kỳ trong số các phương pháp được bộc lộ ở đây.

Phương tiện để tạo mã cung cấp các kỹ thuật mà cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. đầu vào bên ngoài có thể là, ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Việc thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khỏi việc được xuất ra ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Mô tả vắn tắt các hình vẽ

Để hiểu hoàn chỉnh hơn về sáng chế, tham chiếu sẽ được thực hiện tới phần mô tả vắn tắt sau đây, kết hợp với các hình vẽ kèm theo và phần mô tả chi tiết, trong đó các số chỉ dẫn giống nhau biểu thị các bộ phận giống nhau.

Fig.1 là sơ đồ khối minh họa hệ thống tạo mã làm ví dụ mà có thể sử dụng các kỹ thuật GDR.

Fig.2 là sơ đồ khối minh họa bộ mã hóa video làm ví dụ mà có thể thực hiện các kỹ thuật GDR.

Fig.3 là sơ đồ khối minh họa ví dụ về bộ giải mã video mà có thể thực hiện các kỹ thuật GDR.

Fig.4 là sự biểu diễn của mối quan hệ giữa ảnh IRAP so với các ảnh dẫn và các ảnh theo sau theo thứ tự giải mã và thứ tự trình bày.

Fig.5 minh họa kỹ thuật làm mới giải mã từng bước.

Fig.6 là hình vẽ sơ lược minh họa tìm kiếm chuyển động không mong muốn.

Fig.7 minh họa luồng bit video được tạo cấu hình để thực hiện kỹ thuật làm mới giải mã từng bước theo phương án của sáng chế.

Fig.8 thể hiện một phương án của phương pháp giải mã luồng bit video đã được tạo mã.

Fig.9 là hình vẽ dạng giản lược của thiết bị mã hóa video.

Fig.10 là hình vẽ dạng giản lược của một phương án của phương tiện để tạo mã.

Mô tả chi tiết sáng chế

Fig.1 là sơ đồ khối thể hiện hệ thống tạo mã làm ví dụ 10 mà có thể sử dụng các kỹ thuật tạo mã video như được mô tả ở đây. Như được minh họa trên Fig.1, hệ thống tạo mã 10 bao gồm thiết bị nguồn 12 mà tạo ra dữ liệu video được mã hóa để được giải mã ở thời gian sau bởi thiết bị đích 14. Cụ thể là, thiết bị nguồn 12 có thể cung cấp dữ liệu video cho thiết bị đích 14 thông qua phương tiện đọc được bằng máy tính 16. Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong số nhiều thiết bị, bao gồm máy tính để bàn, máy tính ghi chép (ví dụ, các máy tính xách tay), máy tính bảng, hộp giải mã tín hiệu truyền hình, điện thoại cầm tay như điện thoại được gọi là “thông minh”, máy tính bảng (pad) được gọi là “thông minh”, tivi, máy ghi hình, thiết bị hiển thị, máy chơi đa phương tiện số, máy chơi video game, thiết bị phát luồng video, hoặc dạng tương tự. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị cho việc truyền thông không dây.

Thiết bị đích 14 có thể nhận dữ liệu video được mã hóa sẽ được giải mã thông qua phương tiện đọc được bằng máy tính 16. Phương tiện đọc được bằng máy tính 16 có thể bao gồm loại bất kỳ của phương tiện hoặc thiết bị có khả năng di chuyển dữ liệu video được mã hóa từ thiết bị nguồn 12 tới thiết bị đích 14. Theo một ví dụ, phương tiện đọc được bằng máy tính 16 có thể bao gồm phương tiện truyền thông để cho phép thiết bị nguồn 12 truyền dữ liệu video được mã hóa một cách trực tiếp tới thiết bị đích 14 theo thời gian thực. Dữ liệu video được mã hóa có thể được điều biến theo tiêu chuẩn truyền thông như giao thức truyền thông không dây, và được truyền đến thiết bị đích 14. Phương tiện truyền thông có thể bao gồm phương tiện truyền thông không dây hoặc có dây bất kỳ

như phổ tần số vô tuyến (Radio Frequency, RF) hoặc một hoặc nhiều đường truyền vật lý. Phương tiện truyền thông có thể tạo ra một phần của mạng trên cơ sở gói, như mạng cục bộ, mạng diện rộng, hoặc mạng toàn cầu như Internet. Phương tiện truyền thông có thể bao gồm các bộ định tuyến, các bộ chuyển, các trạm gốc, hoặc thiết bị khác bất kỳ mà có thể hữu ích để tạo điều kiện thuận lợi cho việc truyền thông từ thiết bị nguồn 12 đến thiết bị đích 14.

Theo một số ví dụ, dữ liệu được mã hóa có thể được xuất từ giao diện đầu ra 22 đến thiết bị lưu trữ. Tương tự, dữ liệu được mã hóa có thể được truy cập từ thiết bị lưu trữ bởi giao diện đầu vào. Thiết bị lưu trữ có thể bao gồm bất kỳ trong số nhiều phương tiện lưu trữ dữ liệu được truy cập cục bộ hoặc được phân bố như ổ cứng, các đĩa Blu-ray, các đĩa video kỹ thuật số (digital video disk, DVD), các bộ nhớ chỉ đọc dạng đĩa nén (Compact Disc Read-Only Memory, CD-ROM), bộ nhớ tác động nhanh (flash memory), bộ nhớ khả biến hoặc bất khả biến, hoặc phương tiện lưu trữ số thích hợp khác bất kỳ để lưu trữ dữ liệu video được mã hóa. Theo ví dụ khác, thiết bị lưu trữ có thể tương ứng với máy chủ tệp hoặc thiết bị lưu trữ trung gian khác mà có thể lưu trữ video được mã hóa được tạo ra bởi thiết bị nguồn 12. Thiết bị đích 14 có thể truy cập được lưu trữ dữ liệu video từ thiết bị lưu trữ thông qua việc tạo luồng hoặc tải xuống. Máy chủ tệp có thể là loại bất kỳ trong số máy chủ có khả năng lưu trữ dữ liệu video được mã hóa và truyền dữ liệu video được mã hóa đó đến thiết bị đích 14. Các máy chủ tệp làm ví dụ bao gồm máy chủ web (ví dụ, cho trang web), máy chủ giao thức truyền tệp (FTP: file transfer protocol), các thiết bị lưu trữ được gắn trên mạng (NAS: network attached storage), hoặc ổ đĩa cục bộ. Thiết bị đích 14 có thể truy cập dữ liệu video được mã hóa qua kết nối dữ liệu tiêu chuẩn bất kỳ bao gồm kết nối internet. Kết nối này có thể bao gồm kênh không dây (ví dụ, kết nối Wi-Fi), kết nối có dây (ví dụ, tuyến thuê bao kỹ thuật số (DSL: digital subscriber line)), modem cáp, v.v.), hoặc sự kết hợp của cả hai loại trên vốn thích hợp cho truy cập dữ liệu video được mã hóa được lưu trữ trên máy chủ tệp. Việc truyền của dữ liệu video được mã hóa từ thiết bị lưu trữ có thể là việc truyền tạo luồng, việc truyền tải xuống, hoặc dạng kết hợp của chúng.

Các kỹ thuật của sáng chế không nhất thiết bị giới hạn ở các ứng dụng hoặc các thiết lập không dây. Các kỹ thuật có thể được áp dụng cho việc tạo mã video trong việc hỗ trợ ứng dụng bất kỳ trong số nhiều ứng dụng đa phương tiện, như truyền quảng bá truyền

hình qua vô tuyến, truyền truyền hình cáp, truyền truyền hình vệ tinh, truyền tạo luồng video qua Internet, như tạo luồng thích ứng động qua HTTP (dynamic adaptive streaming over HTTP, DASH), video kỹ thuật số mà được mã hóa lên trên phương tiện lưu trữ dữ liệu, việc giải mã video kỹ thuật số được lưu trữ trên phương tiện lưu trữ dữ liệu, hoặc các ứng dụng khác. Theo một số ví dụ, hệ thống tạo mã 10 có thể được tạo cấu hình để hỗ trợ việc truyền video một chiều hoặc hai chiều để hỗ trợ các ứng dụng như việc tạo luồng video, phát lại video, truyền quảng bá video, và/hoặc điện thoại video.

Theo ví dụ trên Fig.1, thiết bị nguồn 12 bao gồm nguồn video 18, bộ mã hóa video 20, và giao diện đầu ra 22. Thiết bị đích 14 chứa giao diện đầu vào 28, bộ giải mã video 30, và thiết bị hiển thị 32. Theo sáng chế, bộ mã hóa video 20 của thiết bị nguồn 12 và/hoặc bộ giải mã video 30 của thiết bị đích 14 có thể được tạo cấu hình để áp dụng các kỹ thuật để tạo mã video. Theo các ví dụ khác, thiết bị nguồn và thiết bị đích có thể chứa các thành phần hoặc các bố trí khác. Ví dụ, thiết bị nguồn 12 có thể nhận dữ liệu video từ nguồn video bên ngoài, như camera bên ngoài. Tương tự, thiết bị đích 14 có thể ghép nối với thiết bị hiển thị bên ngoài, thay vì bao gồm thiết bị hiển thị được tích hợp.

Hệ thống tạo mã 10 được minh họa trên Fig.1 chỉ là một ví dụ. Các kỹ thuật để tạo mã video có thể được thực hiện bởi thiết bị mã hóa và/hoặc giải mã video kỹ thuật số bất kỳ. Mặc dù các kỹ thuật của sáng chế nói chung được thực hiện bởi thiết bị mã hóa video, các kỹ thuật cũng có thể được thực hiện bởi bộ mã hóa/bộ giải mã video, thường được gọi là “CODEC”. Ngoài ra, các kỹ thuật của sáng chế cũng có thể được thực hiện bởi bộ tiền xử lý video. Bộ mã hóa và/hoặc bộ giải mã video có thể là đơn vị xử lý đồ họa (graphics processing unit, GPU) hoặc thiết bị tương tự.

Thiết bị nguồn 12 và thiết bị đích 14 chỉ là các ví dụ về các thiết bị tạo mã này trong đó thiết bị nguồn 12 sinh ra dữ liệu video được tạo mã để truyền đến thiết bị đích 14. Theo một số ví dụ, thiết bị nguồn 12 và thiết bị đích 14 có thể hoạt động theo cách về cơ bản là đối xứng sao cho mỗi thiết bị trong số các thiết bị nguồn 12 và đích 14 chứa các thành phần mã hóa và giải mã video. Do đó, hệ thống tạo mã 10 có thể hỗ trợ việc truyền video một chiều hoặc hai chiều giữa các thiết bị video 12, 14, ví dụ, cho việc tạo luồng video, phát lại video, truyền quảng bá video, hoặc điện thoại video.

Nguồn video 18 của thiết bị nguồn 12 có thể chứa thiết bị quay video, như camera video, bản lưu trữ video chứa video được quay trước đó, và/hoặc giao diện nạp video để

nhận video từ bộ cung cấp nội dung video. Theo phương án thay thế khác, nguồn video 18 có thể sinh ra dữ liệu dựa trên đồ họa máy tính làm nguồn video, hoặc dạng kết hợp của video trực tiếp, video được lưu trữ, và video được sinh ra bởi máy tính.

Trong một số trường hợp, khi nguồn video 18 là camera video, thiết bị nguồn 12 và thiết bị đích 14 có thể tạo ra các điện thoại gọi là điện thoại ghi hình hoặc các điện thoại video. Tuy nhiên, như được mô tả ở trên, các kỹ thuật được mô tả trong sáng chế có thể áp dụng được cho việc tạo mã video nói chung, và có thể được áp dụng cho các ứng dụng không dây và/hoặc có dây. Trong mỗi trường hợp, video được quay, được quay trước, hoặc được sinh ra bởi máy tính có thể được mã hóa bởi bộ mã hóa video 20. Sau đó, thông tin video được mã hóa có thể được xuất ra bởi giao diện đầu ra 22 đến phương tiện đọc được bằng máy tính 16.

Phương tiện đọc được bằng máy tính 16 có thể chứa phương tiện chuyển tiếp, như phương tiện truyền quảng bá không dây hoặc phương tiện truyền mạng có dây, hoặc phương tiện lưu trữ (nghĩa là, phương tiện lưu trữ bất biến), như đĩa cứng, ổ flash, đĩa nén, đĩa video kỹ thuật số, đĩa Blu-ray, hoặc phương tiện đọc được bằng máy tính khác. Theo một số ví dụ, máy chủ mạng (không được minh họa trên hình vẽ) có thể nhận dữ liệu video được mã hóa từ thiết bị nguồn 12 và cung cấp dữ liệu video được mã hóa đến thiết bị đích 14, ví dụ, thông qua hoạt động truyền mạng. Tương tự, thiết bị điện toán của cơ sở sản xuất phương tiện, như cơ sở dán nhãn đĩa, có thể nhận dữ liệu video được mã hóa từ thiết bị nguồn 12 và sản xuất đĩa chứa dữ liệu video được mã hóa. Do đó, phương tiện đọc được bằng máy tính 16 có thể được hiểu là bao gồm một hoặc nhiều phương tiện đọc được bằng máy tính có các dạng khác nhau, theo các ví dụ khác nhau.

Giao diện đầu vào 28 của thiết bị đích 14 nhận thông tin từ phương tiện đọc được bằng máy tính 16. Thông tin của phương tiện đọc được bằng máy tính 16 có thể chứa thông tin cú pháp được xác định bởi bộ mã hóa video 20, mà cũng được sử dụng bởi bộ giải mã video 30, vốn chứa các phần tử cú pháp mô tả các đặc tính và/hoặc việc xử lý của các khối và các đơn vị được tạo mã khác, ví dụ, nhóm của các ảnh (group of pictures, GOPs). Thiết bị hiển thị 32 hiển thị dữ liệu video được giải mã cho người dùng, và có thể bao gồm thiết bị bất kỳ trong nhiều thiết bị hiển thị như ống tia catot (cathode ray tube, CRT), màn hình tinh thể lỏng (liquid crystal display, LCD), màn hình plasma, màn hình

điốt phát sáng hữu cơ (organic light emitting diode, OLED), hoặc loại khác của thiết bị hiển thị.

Bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo tiêu chuẩn tạo mã video, như tiêu chuẩn việc tạo mã video hiệu suất cao (High Efficiency Video Coding, HEVC) hiện đang được phát triển, và có thể phù hợp với mô hình thử nghiệm HEVC (HEVC Test Model, HM). Theo cách thay thế, bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo tiêu chuẩn độc quyền hoặc tiêu chuẩn công nghiệp khác, như tiêu chuẩn lĩnh vực chuẩn hóa viễn thông của hiệp hội viễn thông quốc tế (International Telecommunications Union Telecommunication Standardization Sector, ITU-T) H.264, được gọi theo cách khác là nhóm các chuyên gia về ảnh động (Moving Picture Expert Group, MPEG)-4, phần 10, tạo mã video nâng cao (Advanced Video Coding, AVC), H.265/HEVC, hoặc các mở rộng của các tiêu chuẩn này. Tuy nhiên, các kỹ thuật của sáng chế, không bị giới hạn ở tiêu chuẩn tạo mã cụ thể bất kỳ. Các ví dụ khác về các tiêu chuẩn tạo mã video chứa MPEG-2 và ITU-T H.263. Mặc dù không được minh họa trên Fig.1, theo một số khía cạnh, mỗi bộ trong số bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp với bộ mã hóa và bộ giải mã âm thanh, và có thể chứa các cụm bộ dồn kênh-bộ khử dồn kênh (multiplexer-demultiplexer, MUX-DEMUX) thích hợp, hoặc phần cứng và phần mềm khác, để xử lý việc mã hóa của cả âm thanh và video trong luồng dữ liệu chung hoặc các luồng dữ liệu riêng biệt. Nếu có thể áp dụng, các đơn vị MUX-DEMUX có thể phù hợp với giao thức dồn kênh ITU H.223, hoặc các giao thức khác như giao thức gói thông tin người dùng (user datagram protocol, UDP).

Mỗi bộ trong số bộ mã hóa 20 và bộ giải mã 30 có thể được thực hiện như là hệ mạch bất kỳ trong số nhiều hệ mạch bộ mã hóa thích hợp, như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (digital signal processor, DSP), mạch tích hợp chuyên dụng (application specific integrated circuit, ASIC), mảng cổng lập trình được cổng dạng trường (field programmable gate array, FPGA), bộ logic rời rạc, phần mềm, phần cứng, phần sụn hoặc các dạng kết hợp bất kỳ của chúng. Khi các kỹ thuật được thực hiện một phần trong phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong phương tiện đọc được bởi máy tính bất biến phù hợp và thực hiện các lệnh trong phần cứng sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Mỗi bộ trong số bộ mã hóa video 20 và bộ giải mã video 30 có thể được chứa trong một hoặc nhiều bộ mã

hóa hoặc bộ giải mã, mỗi bộ đều có thể được tích hợp như là một phần của bộ mã hóa/bộ giải mã kết hợp (CODEC) trong thiết bị tương ứng. Thiết bị chứa bộ mã hóa video 20 và/hoặc bộ giải mã video 30 có thể bao gồm mạch tích hợp, bộ vi xử lý, và/hoặc thiết bị truyền thông không dây, như điện thoại di động.

Fig.2 là sơ đồ khối minh họa ví dụ về bộ mã hóa video 20 mà có thể thực hiện các kỹ thuật tạo mã video. Bộ mã hóa video 20 có thể thực hiện việc tạo mã nội và liên thông của các khối video nằm trong các lát video. Việc tạo mã nội dựa vào dự đoán theo không gian để làm giảm hoặc loại bỏ độ dư không gian trong video nằm trong khung video hoặc ảnh cho trước. Việc tạo mã liên thông dựa vào dự đoán theo thời gian để làm giảm hoặc loại bỏ độ dư thời gian trong video nằm trong các khung hoặc các ảnh liên kế của chuỗi video. Chế độ nội (chế độ I) có thể đề cập đến chế độ bất kỳ trong một số chế độ tạo mã dựa trên không gian. Các chế độ liên thông, như dự đoán một chiều (được biết đến là dự đoán đơn) (chế độ P) hoặc dự đoán hai chiều (được biết đến là dự đoán đôi) (chế độ B), có thể đề cập đến chế độ bất kỳ trong một số chế độ tạo mã dựa trên thời gian.

Như được minh họa trên Fig.2, bộ mã hóa video 20 nhận khối video hiện tại nằm trong khung video để được mã hóa. Theo ví dụ trên Fig.2, bộ mã hóa video 20 chứa đơn vị chọn chế độ 40, bộ nhớ khung tham chiếu 64, bộ cộng 50, đơn vị xử lý biến đổi 52, đơn vị lượng tử hóa 54, và đơn vị tạo mã entropy 56. Theo đó, đơn vị chọn chế độ 40 chứa đơn vị bù chuyển động 44, đơn vị ước tính chuyển động 42, đơn vị dự đoán trong ảnh (được biết đến là dự đoán trong ảnh) 46, và đơn vị phân mảnh 48. Đối với việc tái tạo khối video, bộ mã hóa video 20 cũng chứa đơn vị lượng tử hóa ngược 58, đơn vị biến đổi ngược 60, và bộ cộng 62. Bộ lọc khử khối (không được minh họa trên Fig.2) cũng có thể được chứa để lọc các đường bao khối để loại bỏ các thành phần giả tính khối ra khỏi video được tái tạo. Nếu cần, bộ lọc khử khối thường sẽ lọc đầu ra của bộ cộng 62. Các bộ lọc bổ sung (trong vòng lặp hoặc sau vòng lặp) cũng có thể được sử dụng ngoài bộ lọc khử khối. Các bộ lọc này không được thể hiện cho đơn giản, nhưng nếu muốn, có thể lọc đầu ra của bộ cộng 50 (như bộ lọc trong vòng lặp).

Trong quy trình mã hóa, bộ mã hóa video 20 nhận lát hoặc khung video cần được tạo mã. Khung hoặc lát có thể được chia thành nhiều khối video. Đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44 thực hiện việc tạo mã dự đoán liên ảnh của khối video được nhận liên quan đến một hoặc nhiều khối trong một hoặc nhiều khung tham

chiều để cung cấp dự đoán theo thời gian. Theo cách thay thế, đơn vị dự đoán trong ảnh 46 có thể thực hiện việc tạo mã dự đoán trong ảnh của khối video được nhận liên quan đến một hoặc nhiều khối lân cận trong cùng khung hoặc lát như khối cần được tạo mã để cung cấp dự đoán theo không gian. Bộ mã hóa video 20 có thể thực hiện nhiều bước mã hóa, ví dụ, để chọn chế độ tạo mã thích hợp cho mỗi khối của dữ liệu video.

Ngoài ra, đơn vị phân mảnh 48 có thể phân mảnh các khối của dữ liệu video thành các khối con, dựa trên sự đánh giá của các sơ đồ phân mảnh trước đó trong các bước tạo mã trước đó. Ví dụ, ban đầu, đơn vị phân mảnh 48 có thể phân mảnh khung hoặc lát thành các đơn vị tạo mã lớn nhất (largest coding unit, LCU), và phân mảnh mỗi LCU thành các đơn vị tạo mã con (sub-CU) dựa trên sự phân tích tốc độ - độ méo (rate-distortion) (ví dụ, sự tối ưu hóa tốc độ - độ méo). Đơn vị chọn chế độ 40 có thể còn tạo ra cấu trúc dữ liệu cây tứ phân biểu thị sự phân mảnh của LCU thành các CU con. Các CU nút lá của cây tứ phân có thể chứa một hoặc nhiều đơn vị dự đoán (prediction unit, PU) và một hoặc nhiều đơn vị biến đổi (transform unit, TU).

Sáng chế sử dụng thuật ngữ “khối” để đề cập đến loại bất kỳ trong số CU, PU, hoặc TU trong ngữ cảnh của HEVC, hoặc các cấu trúc dữ liệu tương tự trong ngữ cảnh của các tiêu chuẩn khác (ví dụ, các khối macro và các khối con của chúng trong H.264/AVC). CU bao gồm nút tạo mã, các PU, và các TU được liên kết với nút tạo mã. Kích thước của CU tương ứng với kích thước của nút tạo mã và là dạng hình vuông. Kích thước của CU có thể nằm trong khoảng từ các điểm ảnh 8x8 lên đến kích thước của khối cây với tối đa các điểm ảnh 64x64 hoặc lớn hơn. Mỗi CU có thể chứa một hoặc nhiều PU và một hoặc nhiều TU. Ví dụ, dữ liệu cú pháp được kết hợp với CU có thể mô tả việc phân mảnh của CU thành một hoặc nhiều PU. Các chế độ phân mảnh có thể khác nhau giữa việc CU được bỏ qua hoặc được mã hóa theo chế độ trực tiếp, được mã hóa theo chế độ dự đoán trong ảnh, hoặc được mã hóa theo chế độ dự đoán liên ảnh (còn được biết đến là dự đoán liên ảnh). Các PU có thể được phân mảnh để có dạng không phải là hình vuông. Ví dụ, dữ liệu cú pháp được kết hợp với CU cũng có thể mô tả việc phân mảnh của CU thành một hoặc nhiều TU theo cây tứ phân. TU có thể có dạng hình vuông hoặc không phải là hình vuông (ví dụ, hình chữ nhật).

Đơn vị chọn chế độ 40 có thể chọn một chế độ trong số các chế độ tạo mã, nội hoặc liên thông, ví dụ, dựa trên các kết quả lỗi, và cung cấp khối được tạo mã nội hoặc liên

thông thu được cho bộ cộng 50 để sinh ra dữ liệu khối dư và cho bộ cộng 62 để tái tạo khối được mã hóa để sử dụng làm khung tham chiếu. Đơn vị chọn chế độ 40 cũng cung cấp các phân tử cú pháp, như các vectơ chuyển động, các bộ chỉ báo chế độ nội, thông tin phân mảnh, và thông tin cú pháp khác, cho đơn vị tạo mã entropy 56.

Đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44 có thể là được tích hợp cao độ, nhưng được minh họa riêng biệt cho các mục đích về khái niệm. Việc ước tính chuyển động, được thực hiện bởi đơn vị ước tính chuyển động 42, là quá trình tạo ra các vectơ chuyển động, để ước tính chuyển động cho các khối video. Ví dụ, vectơ chuyển động có thể chỉ báo sự dịch chuyển của PU của khối video nằm trong khung video hoặc ảnh hiện tại tương đối với khối dự đoán nằm trong khung tham chiếu (hoặc đơn vị được tạo mã khác) tương đối với khối hiện tại được tạo mã nằm trong khung hiện tại (hoặc đơn vị được tạo mã khác). Khối dự đoán là khối mà được tìm thấy là khớp sát khối cần được tạo mã, về độ chênh lệch điểm ảnh, mà có thể được xác định bởi tổng của độ chênh lệch tuyệt đối (sum of absolute difference, SAD), tổng của độ chênh lệch bình phương (sum of square difference, SSD), hoặc các số đo độ chênh lệch khác. Theo một số ví dụ, bộ mã hóa video 20 có thể tính toán các giá trị cho các vị trí điểm ảnh số nguyên con của các ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 64. Ví dụ, bộ mã hóa video 20 có thể nội suy các giá trị của các vị trí điểm ảnh một phần tư, các vị trí điểm ảnh một phần tám, hoặc các vị trí điểm ảnh phân số khác của hình ảnh tham chiếu. Do đó, đơn vị ước tính chuyển động 42 có thể thực hiện tìm kiếm chuyển động tương đối với các vị trí điểm ảnh đủ và các vị trí điểm ảnh phân số và xuất ra vectơ chuyển động với độ chính xác điểm ảnh phân số.

Đơn vị ước tính chuyển động 42 tính toán vectơ chuyển động cho PU của khối video trong lát được tạo mã nội bằng cách so sánh vị trí của PU với vị trí của khối dự đoán của ảnh tham chiếu. Ảnh tham chiếu có thể được chọn từ danh sách ảnh tham chiếu thứ nhất (danh sách 0) hoặc danh sách ảnh tham chiếu thứ hai (danh sách 1), mỗi danh sách trong số chúng xác định một hoặc nhiều ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 64. Đơn vị ước tính chuyển động 42 gửi vectơ chuyển động được tính toán đến đơn vị tạo mã entropy 56 và đơn vị bù chuyển động 44.

Việc bù chuyển động, được thực hiện bởi đơn vị bù chuyển động 44, có thể liên quan đến sự tìm nạp hoặc sinh ra khối dự đoán dựa trên vectơ chuyển động được xác định

bởi đơn vị ước tính chuyển động 42. Ngoài ra, đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44 có thể được tích hợp theo chức năng, theo một số ví dụ. Ngay khi nhận vector chuyển động cho PU của khối video hiện tại, đơn vị bù chuyển động 44 có thể định vị trí khối dự đoán mà vector chuyển động trỏ vào đó trong một danh sách trong số các danh sách ảnh tham chiếu. Bộ cộng 50 tạo ra khối video dư bằng cách trừ đi các giá trị điểm ảnh của khối dự đoán từ các giá trị điểm ảnh của khối video hiện tại được tạo mã, tạo ra các giá trị chênh lệch điểm ảnh, như được bàn luận dưới đây. Nói chung, đơn vị ước tính chuyển động 42 thực hiện việc ước tính chuyển động so với các thành phần độ chói, và đơn vị bù chuyển động 44 sử dụng các vector chuyển động được tính toán dựa trên các thành phần độ chói cho cả các thành phần sắc độ và các thành phần độ chói. Đơn vị chọn chế độ 40 cũng có thể sinh ra các phần tử cú pháp được kết hợp với các khối video và lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối video của lát video.

Đơn vị dự đoán trong ảnh 46 có thể dự đoán trong ảnh khối hiện tại, theo phương án thay thế cho việc dự đoán liên ảnh được thực hiện bởi đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44, như được mô tả ở trên. Cụ thể là, đơn vị dự đoán trong ảnh 46 có thể xác định chế độ dự đoán trong ảnh để sử dụng cho việc mã hóa khối hiện tại. Theo một số ví dụ, đơn vị dự đoán trong ảnh 46 có thể mã hóa khối hiện tại nhờ sử dụng các chế độ dự đoán trong ảnh khác nhau, ví dụ, trong các bước mã hóa riêng biệt, và đơn vị dự đoán trong ảnh 46 (hoặc theo một số ví dụ là đơn vị chọn chế độ 40) có thể chọn chế độ dự đoán trong ảnh thích hợp để sử dụng từ các chế độ được thử nghiệm.

Ví dụ, đơn vị dự đoán trong ảnh 46 có thể tính toán các giá trị nhờ sử dụng phân tích tốc độ - độ méo cho các chế độ dự đoán trong ảnh được thử nghiệm khác nhau, và chọn chế độ dự đoán trong ảnh có các đặc tính tốc độ - độ méo tốt nhất trong số các chế độ được thử nghiệm. Phân tích tốc độ - độ méo nói chung xác định lượng độ méo (hoặc lỗi) giữa khối được mã hóa và khối gốc không được mã hóa mà được mã hóa để tạo ra khối được mã hóa, cũng như tốc độ bit (nghĩa là, số lượng bit) được sử dụng để tạo ra khối được mã hóa. Đơn vị dự đoán trong ảnh 46 có thể tính toán các tỷ số từ các độ méo và các tốc độ cho các khối được mã hóa khác nhau để xác định chế độ dự đoán trong ảnh nào hiển thị giá trị tốc độ - độ méo tốt nhất cho khối.

Ngoài ra, đơn vị dự đoán trong ảnh 46 có thể được tạo cấu hình để mã hóa các khối chiều sâu của ánh xạ chiều sâu nhờ sử dụng chế độ mô hình hóa chiều sâu (depth modeling mode, DMM). Đơn vị chọn chế độ 40 có thể xác định xem liệu chế độ DMM khả dụng có tạo ra các kết quả tạo mã tốt hơn chế độ dự đoán trong ảnh và các chế độ DMM khác hay không, ví dụ, bằng cách sử dụng sự tối ưu hóa tốc độ - độ méo (rate-distortion optimization, RDO). Dữ liệu cho hình ảnh kết cấu tương ứng với ánh xạ chiều sâu có thể được lưu trữ trong bộ nhớ khung tham chiếu 64. Đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44 cũng có thể được tạo cấu hình để dự đoán liên ảnh các khối chiều sâu của ánh xạ chiều sâu.

Sau khi chọn chế độ dự đoán trong ảnh cho khối (ví dụ, chế độ dự đoán trong ảnh thông thường hoặc một chế độ trong số các chế độ DMM), đơn vị dự đoán trong ảnh 46 có thể cung cấp thông tin biểu thị chế độ dự đoán trong ảnh được chọn cho khối đến đơn vị tạo mã entropy 56. Đơn vị tạo mã entropy 56 có thể mã hóa thông tin biểu thị chế độ dự đoán trong ảnh được chọn. Bộ mã hóa video 20 có thể chứa trong dữ liệu cấu hình luồng bit được truyền, mà có thể bao gồm nhiều bảng chỉ số chế độ dự đoán trong ảnh và nhiều bảng chỉ số chế độ dự đoán trong ảnh được sửa đổi (còn được gọi là các bảng ánh xạ từ mã), các định nghĩa của các ngữ cảnh mã hóa cho các khối khác nhau, và các chỉ báo của chế độ dự đoán trong ảnh khả thi nhất, bảng chỉ số chế độ dự đoán trong ảnh, và bảng chỉ số chế độ dự đoán trong ảnh được sửa đổi để sử dụng cho mỗi ngữ cảnh trong số các ngữ cảnh.

Bộ mã hóa video 20 tạo ra khối video dư bằng cách trừ dữ liệu dự đoán từ đơn vị chọn chế độ 40 khỏi khối video gốc đang được tạo mã. Bộ cộng 50 thể hiện thành phần hoặc các thành phần mà thực hiện phép trừ này.

Đơn vị xử lý biến đổi 52 áp dụng việc biến đổi, như biến đổi cosin rời rạc (discrete cosine transform, DCT) hoặc biến đổi tương tự về khái niệm, cho khối dư, tạo ra khối video bao gồm các giá trị hệ số biến đổi dư. Đơn vị xử lý biến đổi 52 có thể thực hiện các biến đổi khác mà tương tự về nguyên lý với DCT. Các biến đổi sóng nhỏ, các biến đổi nguyên, các biến đổi băng phụ hoặc các loại biến đổi khác cũng có thể được sử dụng.

Đơn vị xử lý biến đổi 52 áp dụng biến đổi cho khối dư, tạo ra khối của các hệ số biến đổi dư. Việc biến đổi có thể chuyển đổi thông tin biến đổi dư từ miền giá trị điểm ảnh thành miền biến đổi, như miền tần số. Đơn vị xử lý biến đổi 52 có thể gửi các hệ số

biến đổi thu được đến đơn vị lượng tử hóa 54. Đơn vị lượng tử hóa 54 lượng tử hóa các hệ số biến đổi để tiếp tục làm giảm tốc độ bit. Quy trình lượng tử hóa có thể làm giảm chiều sâu bit được kết hợp với một số hoặc tất cả các hệ số. Mức độ lượng tử hóa có thể được sửa đổi bằng cách điều chỉnh thông số lượng tử hóa. Theo một số ví dụ, đơn vị lượng tử hóa 54 sau đó có thể thực hiện việc quét ma trận chứa các hệ số biến đổi được lượng tử hóa. Theo cách thay thế, đơn vị tạo mã entropy 56 có thể thực hiện việc quét.

Sau khi lượng tử hóa, đơn vị tạo mã entropy 56 tạo mã entropy các hệ số biến đổi được lượng tử hóa. Ví dụ, đơn vị tạo mã entropy 56 có thể thực hiện việc tạo mã chiều dài biến thiên thích ứng ngữ cảnh (context adaptive variable length coding, CAVLC), việc tạo mã số nhị phân thích ứng ngữ cảnh (context adaptive binary arithmetic coding, CABAC), việc tạo mã số nhị phân thích ứng ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, SBAC), việc tạo mã phân chia khoảng khả năng (probability interval partitioning entropy, PIPE), hoặc kỹ thuật tạo mã entropy khác. Trong trường hợp tạo mã entropy dựa trên ngữ cảnh, ngữ cảnh có thể được dựa trên các khối lân cận. Sau khi tạo mã entropy bởi đơn vị tạo mã entropy 56, luồng bit được mã hóa có thể được truyền đến thiết bị khác (ví dụ, bộ giải mã video 30) hoặc đặt được cho hoạt động truyền hoặc truy tìm về sau.

Đơn vị lượng tử hóa ngược 58 và đơn vị biến đổi ngược 60 áp dụng lượng tử hóa ngược và biến đổi ngược, một cách tương ứng, để tái tạo khối dư trong miền điểm ảnh, ví dụ, để sử dụng sau làm khối tham chiếu. Đơn vị bù chuyển động 44 có thể tính toán khối tham chiếu bằng cách bổ sung khối dư cho khối dự đoán của một khung trong số các khung của bộ nhớ khung tham chiếu 64. Đơn vị bù chuyển động 44 cũng có thể áp dụng một hoặc nhiều bộ lọc nội suy cho khối dư được tái tạo, để tính toán các giá trị điểm ảnh số nguyên phụ để sử dụng trong ước tính chuyển động. Bộ cộng 62 cộng khối dư được tái tạo vào khối dự đoán được bù chuyển động được tạo ra bởi đơn vị bù chuyển động 44 để tạo ra khối video được tái tạo để lưu trữ trong bộ nhớ khung tham chiếu 64. Khối video được tái tạo có thể được sử dụng bởi đơn vị ước tính chuyển động 42 và đơn vị bù chuyển động 44 làm khối tham chiếu để tạo mã liên thông khối trong khung video tiếp theo.

Fig.3 là sơ đồ khối minh họa ví dụ về bộ giải mã video 30 mà có thể thực hiện các kỹ thuật tạo mã video. Theo ví dụ trên Fig.3, bộ giải mã video 30 chứa đơn vị giải mã entropy 70, đơn vị bù chuyển động 72, đơn vị dự đoán trong ảnh 74, đơn vị lượng tử hóa

ngược 76, đơn vị biến đổi ngược 78, bộ nhớ khung tham chiếu 82, và bộ cộng 80. Theo một số ví dụ, bộ giải mã video 30 có thể thực hiện bước giải mã nói chung nghịch đảo với bước mã hóa mà đã được mô tả đối với bộ mã hóa video 20 (xem Fig.2). Đơn vị bù chuyển động 72 có thể sinh ra dữ liệu dự đoán dựa trên các vector chuyển động được nhận từ đơn vị giải mã entropy 70, trong khi đơn vị dự đoán trong ảnh 74 có thể sinh ra dữ liệu dự đoán dựa trên các bộ chỉ báo chế độ dự đoán trong ảnh được nhận từ đơn vị giải mã entropy 70.

Trong quá trình giải mã, bộ giải mã video 30 nhận luồng bit video đã được mã hóa mà thể hiện các khối video của lát video được mã hóa và các phần tử cú pháp được kết hợp, từ bộ mã hóa video 20. Đơn vị giải mã entropy 70 của bộ giải mã video 30 giải mã entropy luồng bit để sinh ra các hệ số được lượng tử hóa, các vector chuyển động hoặc các bộ chỉ báo chế độ dự đoán trong ảnh, và các phần tử cú pháp khác. Đơn vị giải mã entropy 70 chuyển tiếp các vector chuyển động và các phần tử cú pháp khác đến đơn vị bù chuyển động 72. Bộ giải mã video 30 có thể nhận các phần tử cú pháp ở mức lát video và/hoặc mức khối video.

Khi lát video được tạo mã dưới dạng lát được tạo mã nội (I), đơn vị dự đoán trong ảnh 74 có thể sinh ra dữ liệu dự đoán cho khối video của lát video hiện tại dựa trên chế độ dự đoán trong ảnh được tạo tín hiệu và dữ liệu từ các khối được giải mã trước đó của khung hoặc ảnh hiện tại. Khi khung video được tạo mã dưới dạng lát được tạo mã liên thông (ví dụ, B, P, hoặc GPB), đơn vị bù chuyển động 72 tạo ra các khối dự đoán cho khối video của lát video hiện tại dựa trên các vector chuyển động và các phần tử cú pháp khác được nhận từ đơn vị giải mã entropy 70. Các khối dự đoán có thể được tạo ra từ một ảnh trong số các ảnh tham chiếu trong một danh sách trong số các danh sách ảnh tham chiếu. Bộ giải mã video 30 có thể xây dựng các danh sách khung tham chiếu, danh sách 0 và danh sách 1, sử dụng các kỹ thuật xây dựng mặc định dựa vào các ảnh tham chiếu được lưu trữ trong bộ nhớ khung tham chiếu 82.

Đơn vị bù chuyển động 72 xác định dự đoán thông tin cho khối video của lát video hiện tại bằng cách phân tích các vector chuyển động và các phần tử cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán cho khối video hiện tại được giải mã. Ví dụ, đơn vị bù chuyển động 72 sử dụng một số phần tử cú pháp được nhận để xác định chế độ dự đoán (ví dụ, dự đoán trong ảnh hoặc liên thông) được sử dụng để tạo mã các

khối video của lát video, loại lát dự đoán liên ảnh (ví dụ, lát B, lát P hoặc lát GPB), thông tin xây dựng cho một hoặc nhiều trong số các danh sách ảnh tham chiếu cho lát, các vector chuyển động cho mỗi khối video được mã hóa liên thông của lát, trạng thái dự đoán liên ảnh cho mỗi khối video được tạo mã liên thông của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại.

Đơn vị bù chuyển động 72 cũng có thể thực hiện việc nội suy dựa trên các bộ lọc nội suy. Đơn vị bù chuyển động 72 có thể sử dụng các bộ lọc nội suy như được sử dụng bởi bộ mã hóa video 20 trong quá trình mã hóa các khối video để tính toán các giá trị nội suy cho các điểm ảnh số nguyên con của các khối tham chiếu. Trong trường hợp này, đơn vị bù chuyển động 72 có thể xác định các bộ lọc nội suy sử dụng bởi bộ mã hóa video 20 từ các phần tử cú pháp nhận được và sử dụng các bộ lọc nội suy để sản xuất các khối dự đoán.

Dữ liệu cho hình ảnh kết cấu tương ứng với ánh xạ chiều sâu có thể được lưu trữ trong bộ nhớ khung tham chiếu 82. Đơn vị bù chuyển động 72 cũng có thể được tạo cấu hình để dự đoán liên ảnh các khối chiều sâu của ánh xạ chiều sâu.

Theo một phương án, bộ giải mã video 30 chứa giao diện người dùng (user interface, UI) 84. Giao diện người dùng 84 được tạo cấu hình để nhận đầu vào từ người dùng của bộ giải mã video 30 (ví dụ, người quản lý mạng). Qua giao diện người dùng 84, người dùng có thể quản lý hoặc thay đổi các thiết đặt trên bộ giải mã video 30. Ví dụ, người dùng có thể nhập hoặc cung cấp giá trị cho thông số (ví dụ, cờ) để điều khiển cấu hình và/hoặc hoạt động của bộ giải mã video 30 theo sở thích của người dùng. Giao diện người dùng 84 có thể là, ví dụ, giao diện đồ họa người dùng (graphical user interface, GUI) mà cho phép người dùng tương tác với bộ giải mã video 30 qua các biểu tượng đồ họa [https://en.wikipedia.org/wiki/Icon_\(computing\)](https://en.wikipedia.org/wiki/Icon_(computing)), các trình đơn xổ xuống, các hộp đánh dấu, và v.v. Trong một số trường hợp, giao diện người dùng 84 có thể nhận thông tin từ người dùng qua bàn phím, chuột, hoặc thiết bị ngoại vi khác. Theo một phương án, người dùng có thể truy cập giao diện người dùng 84 qua điện thoại thông minh, thiết bị máy tính bảng, máy tính cá nhân được đặt xa khỏi bộ giải mã video 30, và v.v. Như được sử dụng ở đây, giao diện người dùng 84 có thể được gọi là đầu vào bên ngoài hoặc phương tiện bên ngoài.

Lưu ý các điều trên, các kỹ thuật nén video thực hiện việc dự đoán theo không gian (nội) và/hoặc việc dự đoán theo thời gian (liên ảnh) để làm giảm hoặc loại bỏ sự dư có trong các chuỗi video. Đối với việc tạo mã video dựa trên khối, lát video (nghĩa là, ảnh video hoặc một phần của ảnh video) có thể được phân mảnh thành các khối video, mà cũng có thể được gọi là các khối cây, các khối cây tạo mã (coding tree block, CTB), các đơn vị cây tạo mã (coding tree unit, CTU), các đơn vị tạo mã (coding unit, CU) và/hoặc các nút tạo mã. Các khối video trong lát được tạo mã nội (I) của ảnh được mã hóa nhờ sử dụng việc dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh. Các khối video trong lát được tạo mã liên thông (P hoặc B) của ảnh có thể sử dụng việc dự đoán theo không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng ảnh hoặc việc dự đoán theo thời gian đối với các mẫu tham chiếu trong các ảnh tham chiếu khác. Các ảnh có thể được gọi là các khung, và các ảnh tham chiếu có thể được gọi là các khung tham chiếu.

Dự đoán theo không gian hoặc thời gian tạo ra khối dự đoán cho khối cần được tạo mã. Dữ liệu dư thể hiện các khác biệt điểm ảnh giữa khối gốc cần được tạo mã và khối dự đoán. Khối được tạo mã liên thông được mã hóa theo vector chuyển động mà chỉ đến khối của các mẫu tham chiếu đang tạo ra khối dự đoán, và dữ liệu dư biểu thị sự chênh lệch giữa khối được tạo mã và khối dự đoán. Khối được tạo mã nội được mã hóa theo chế độ tạo mã nội và dữ liệu dư. Với việc nén thêm nữa, dữ liệu dư có thể được biến đổi từ miền điểm ảnh tới miền biến đổi, tạo ra các hệ số biến đổi dư, mà sau đó có thể được lượng tử hóa. Các hệ số biến đổi được lượng tử hóa, được bố trí ban đầu vào mảng hai chiều, có thể được quét để tạo ra vector một chiều của các hệ số biến đổi, và việc tạo mã entropy có thể được áp dụng để đạt được độ nén cao hơn nữa.

Việc nén hình ảnh và video đã trải qua sự phát triển nhanh, dẫn đến các tiêu chuẩn tạo mã khác nhau. Các tiêu chuẩn tạo mã video này chứa ITU-T H.261, ISO/IEC MPEG-1 phần 2, ITU-T H.262 hoặc ISO/IEC MPEG-2 phần 2, ITU-T H.263, ISO/IEC MPEG-4 phần 2, tạo mã video nâng cao (Advanced Video Coding, AVC), còn được biết đến là ITU-T H.264 hoặc ISO/IEC MPEG-4 phần 10, và tạo mã video hiệu suất cao (High Efficiency Video Coding, HEVC), còn được biết đến là ITU-T H.265 hoặc MPEG-H phần 2. AVC chứa các tiêu chuẩn mở rộng như tạo mã video khả biến (Scalable Video Coding, SVC), tạo mã video đa điểm quan sát (Multiview Video Coding, MVC) và tạo

mã video đa điểm quan sát cộng với chiều sâu (Multiview Video Coding plus Depth, MVC+D), và 3D AVC (3D-AVC). HEVC chứa các mở rộng như HEVC khả biến (Scalable HEVC, SHVC), HEVC đa điểm quan sát (Multiview HEVC, MV-HEVC), và 3D HEVC (3D-HEVC).

Cũng có tiêu chuẩn tạo mã video mới, được gọi là tạo mã video vạn năng (Versatile Video Coding, VVC), được phát triển bởi nhóm chuyên gia video liên kết (joint video experts team, JVET) của ITU-T và ISO/IEC. Trong khi tiêu chuẩn VVC có một vài dự thảo làm việc, cụ thể là một dự thảo làm việc (Working Draft, WD) của VVC, đó là B. Bross, J. Chen, và S. Liu, “Versatile Video Coding (Draft 4),” JVET-M1001-v5, hội thảo JVET thứ 13, tháng 1 năm 2019 (VVC dự thảo 4) được tham khảo ở đây.

Phần mô tả của các kỹ thuật được bộc lộ ở đây dựa trên tiêu chuẩn tạo mã video đang được phát triển tạo mã video vạn năng (VVC) bởi nhóm chuyên gia liên kết (JVET) của ITU-T và ISO/IEC. Tuy nhiên, các kỹ thuật cũng áp dụng cho các phần mô tả codec video khác.

Fig.4 là sự biểu diễn 400 của mối quan hệ giữa ảnh IRAP 402 so với các ảnh dẫn 404 và các ảnh theo sau 406 theo thứ tự giải mã 408 và thứ tự trình bày 410. Theo một phương án, ảnh IRAP 402 được gọi là ảnh truy cập ngẫu nhiên sạch (clean random access, CRA) hoặc là ảnh làm mới tức thời bộ giải mã (instantaneous decoder refresh, IDR) với ảnh truy cập ngẫu nhiên giải mã được (RADL). Trong HEVC, tất cả các ảnh IDR, các ảnh CRA, và các ảnh truy cập liên kết bị phá vỡ (Broken Link Access, BLA) được xem là các ảnh IRAP 402. Đối với VVC, trong hội thảo JVET thứ 12 vào tháng 10 năm 2018, được đồng ý là có cả các ảnh IDR và CRA như là các ảnh IRAP.

Như được minh họa trên Fig.4, các ảnh dẫn 404 (ví dụ, các ảnh 2 và 3) theo sau ảnh IRAP 402 theo thứ tự giải mã 408, nhưng ở trước ảnh IRAP 402 theo thứ tự trình bày 410. Ảnh theo sau 406 theo sau ảnh IRAP 402 theo cả thứ tự giải mã 408 và thứ tự trình bày 410. Trong khi hai ảnh dẫn 404 và một ảnh theo sau 406 được minh họa trên Fig.4, những người có trình độ trung bình trong lĩnh vực sẽ hiểu rằng nhiều hơn hoặc nhỏ hơn các ảnh dẫn 404 và/hoặc các ảnh theo sau 406 có thể có mặt theo thứ tự giải mã 408 và thứ tự trình bày 410 trong các ứng dụng thực tế.

Các ảnh dẫn 404 trên Fig.4 đã được chia thành hai loại, đó là dẫn được bỏ qua truy cập ngẫu nhiên (random access skipped leading, RASL) và RADL. Khi việc giải mã bắt

đầu với ảnh IRAP 402 (ví dụ, ảnh 1), ảnh RADL (ví dụ, ảnh 3) có thể được giải mã đúng; tuy nhiên, ảnh RASL (ví dụ, ảnh 2) không thể được giải mã đúng. Do đó, ảnh RASL được loại bỏ. Vì sự phân biệt giữa các ảnh RADL và RASL, loại của ảnh dẫn được kết hợp với ảnh IRAP nên được xác định là RADL hoặc RASL cho việc tạo mã hiệu quả và đúng. Trong HEVC, khi các ảnh RASL và RADL có mặt, bị hạn chế đối với các ảnh RASL và RADL mà được kết hợp với cùng ảnh IRAP, các ảnh RASL sẽ ở trước các ảnh RADL theo thứ tự trình bày 410.

Ảnh IRAP 402 tạo ra hai chức năng/lợi ích quan trọng sau. Trước tiên, sự có mặt của ảnh IRAP 402 chỉ báo rằng quy trình giải mã có thể bắt đầu từ ảnh đó. Chức năng này cho phép dấu hiệu truy cập ngẫu nhiên trong đó quy trình giải mã bắt đầu ở vị trí đó trong luồng bit, không nhất thiết ở đầu của luồng bit, miễn là ảnh IRAP 402 có mặt ở vị trí đó. Thứ hai, sự có mặt của ảnh IRAP 402 làm mới quy trình giải mã sao cho ảnh được tạo mã bắt đầu ở ảnh IRAP 402, ngoại trừ các ảnh RASL, được tạo mã mà không có sự tham chiếu bất kỳ đến các ảnh trước đó. Do đó, việc có ảnh IRAP 402 có mặt trong luồng bit sẽ ngăn lỗi bất kỳ mà có thể xảy ra trong quá trình giải mã các ảnh được tạo mã trước ảnh IRAP 402 để truyền đến ảnh IRAP 402 và các ảnh này mà theo sau ảnh IRAP 402 theo thứ tự giải mã 408.

Trong khi các ảnh IRAP 402 tạo ra các chức năng quan trọng, chúng đi kèm với bất lợi đối với hiệu quả nén. Sự có mặt của ảnh IRAP 402 gây ra đột biến trong tốc độ bit. Bất lợi này đối với hiệu quả nén là do hai lý do. Trước tiên, ảnh IRAP 402 là ảnh được dự đoán trong ảnh, bản thân ảnh sẽ yêu cầu tương đối nhiều bit hơn để biểu diễn khi so với các ảnh khác (ví dụ, các ảnh dẫn 404, các ảnh theo sau 406) mà là các ảnh được dự đoán liên ảnh. Thứ hai, vì sự có mặt của ảnh IRAP 402 phá vỡ việc dự đoán theo thời gian (điều này là vì bộ giải mã sẽ làm mới quy trình giải mã, trong đó một hoạt động trong số các hoạt động của quy trình giải mã cho điều này là để loại bỏ các ảnh tham chiếu trước đó trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB)), ảnh IRAP 402 làm cho việc tạo mã của các ảnh mà theo sau ảnh IRAP 402 theo thứ tự giải mã 408 ít hiệu quả hơn (nghĩa là, cần nhiều bit hơn để biểu diễn) vì chúng có ít các ảnh tham chiếu hơn cho việc tạo mã dự đoán liên ảnh của chúng.

Trong số các loại ảnh mà được xem là các ảnh IRAP 402, ảnh IDR trong HEVC có sự tạo tín hiệu và dẫn ra khác khi so với các loại ảnh khác. Một số khác nhau là như sau.

Đối với sự tạo tín hiệu và dẫn ra của giá trị đếm thứ tự ảnh (picture order count, POC) của ảnh IDR, phần bit quan trọng nhất (most significant bit, MSB) của POC không được dẫn ra từ ảnh chính trước đó nhưng được thiết đặt đơn giản là bằng 0.

Đối với thông tin tạo tín hiệu cần cho việc quản lý ảnh tham chiếu, phần đầu lát của ảnh IDR không chứa thông tin cần được tạo tín hiệu để hỗ trợ việc quản lý ảnh tham chiếu. Đối với các loại ảnh khác (nghĩa là, CRA, theo sau, truy cập lớp con thời gian (temporal sub-layer access, TSA), v.v.), thông tin như bộ ảnh tham chiếu (reference picture set, RPS) được mô tả bên dưới hoặc các dạng khác của thông tin tương tự (ví dụ, các danh sách ảnh tham chiếu) cần cho quy trình đánh dấu các ảnh tham chiếu (nghĩa là, quy trình để xác định trạng thái của các ảnh tham chiếu trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB), được sử dụng cho việc tham chiếu và không được sử dụng cho việc tham chiếu). Tuy nhiên, đối với ảnh IDR, thông tin này không cần được tạo tín hiệu vì sự có mặt của IDR chỉ báo rằng quy trình giải mã sẽ đánh dấu một cách đơn giản tất cả các ảnh tham chiếu trong DPB là không được sử dụng cho việc tham chiếu.

Trong HEVC và VVC, mỗi ảnh trong số các ảnh IRAP 402 và các ảnh dẫn 404 có thể được chứa nằm trong đơn vị lớp trừu tượng mạng (NAL) đơn. Tập hợp của các đơn vị NAL có thể được gọi là đơn vị truy cập. Các ảnh IRAP 402 và các ảnh dẫn 404 được cung cấp các loại đơn vị NAL khác nhau sao cho chúng có thể được xác định dễ dàng bởi các ứng dụng mức hệ thống. Ví dụ, bộ nối video cần hiểu các loại ảnh được tạo mã mà không phải hiểu quá chi tiết về phần tử cú pháp trong được tạo mã luồng bit, cụ thể là để nhận biết các ảnh IRAP 402 từ các ảnh không phải là IRAP và để nhận biết các ảnh dẫn 404, chứa xác định các ảnh RASL và RADL, từ các ảnh theo sau 406. Các ảnh theo sau 406 là các ảnh này mà được kết hợp với ảnh IRAP 402 và theo sau ảnh IRAP 402 theo thứ tự trình bày 410. Ảnh có thể theo sau ảnh IRAP 402 cụ thể theo thứ tự giải mã 408 và ở trước ảnh IRAP 402 khác bất kỳ theo thứ tự giải mã 408. Đối với điều này, việc cung cấp cho các ảnh IRAP 402 và các ảnh dẫn 404 loại đơn vị NAL của riêng chúng giúp ích cho các ứng dụng này.

Đối với HEVC, các loại đơn vị NAL cho các ảnh IRAP chứa các thành phần sau:

BLA với ảnh dẫn (BLA_W_LP): Đơn vị NAL của ảnh truy cập liên kết bị phá vỡ (Broken Link Access, BLA) mà có thể được theo sau bởi một hoặc nhiều ảnh dẫn theo thứ tự giải mã.

BLA với RADL (BLA_W_RADL): Đơn vị NAL của ảnh BLA mà có thể được theo sau bởi một hoặc nhiều ảnh RADL nhưng không có ảnh RASL theo thứ tự giải mã.

BLA với không có ảnh dẫn (BLA_N_LP): Đơn vị NAL của ảnh BLA mà không được theo sau bởi ảnh dẫn theo thứ tự giải mã.

IDR với RADL (IDR_W_RADL): Đơn vị NAL của ảnh IDR mà có thể được theo sau bởi một hoặc nhiều ảnh RADL nhưng không có ảnh RASL theo thứ tự giải mã.

IDR với không có ảnh dẫn (IDR_N_LP): Đơn vị NAL của ảnh IDR mà không được theo sau bởi ảnh dẫn theo thứ tự giải mã.

CRA: Đơn vị NAL của ảnh truy cập ngẫu nhiên sạch (clean random access, CRA) mà có thể được theo sau bởi các ảnh dẫn (nghĩa là, hoặc các ảnh RASL hoặc các ảnh RADL hoặc cả hai).

RADL: Đơn vị NAL của ảnh RADL.

RASL: Đơn vị NAL của ảnh RASL.

Đối với VVC, loại đơn vị NAL cho các ảnh IRAP 402 và các ảnh dẫn 404 là như sau:

IDR với RADL (IDR_W_RADL): Đơn vị NAL của ảnh IDR mà có thể được theo sau bởi một hoặc nhiều ảnh RADL nhưng không có ảnh RASL theo thứ tự giải mã.

IDR với không có ảnh dẫn (IDR_N_LP): Đơn vị NAL của ảnh IDR mà không được theo sau bởi ảnh dẫn theo thứ tự giải mã.

CRA: Đơn vị NAL của ảnh truy cập ngẫu nhiên sạch (clean random access, CRA) mà có thể được theo sau bởi các ảnh dẫn (nghĩa là, hoặc các ảnh RASL hoặc các ảnh RADL hoặc cả hai).

RADL: Đơn vị NAL của ảnh RADL.

RASL: Đơn vị NAL của ảnh RASL.

Vệć làm mới nội tạng dần/Việc làm mới giải mã từng bước được bàn luận bên dưới.

Đối với các ứng dụng độ trễ thấp, mong muốn tránh việc tạo mã ảnh như là ảnh IRAP (ví dụ, ảnh IRAP 402) do yêu cầu tốc độ bit tương đối lớn của nó so với các ảnh không phải là IRAP (nghĩa là, P-/B-), mà do đó gây ra nhiều thời gian chờ/độ trễ hơn.

Tuy nhiên, việc hoàn toàn tránh việc sử dụng IRAP có thể không phải là có thể trong tất cả các ứng dụng độ trễ thấp. Ví dụ, đối với các ứng dụng đàm thoại như hội nghị từ xa nhiều bên, cần thiết để cung cấp các điểm thường xuyên trong đó người dùng mới có thể tham gia hội nghị từ xa.

Để cung cấp sự truy cập đến luồng bit mà cho phép người dùng mới tham gia ứng dụng hội nghị từ xa nhiều bên, một chiến lược có thể là sử dụng kỹ thuật làm mới nội tạng dần (progressive intra refresh, PIR) thay vì sử dụng các ảnh IRAP để tránh việc có đỉnh trong tốc độ bit. PIR có thể còn được gọi là việc làm mới giải mã từng bước (GDR). Thuật ngữ PIR và GDR có thể được sử dụng thay thế nhau trong sáng chế.

Fig.5 minh họa kỹ thuật làm mới giải mã từng bước (GDR) 500. Như được minh họa, kỹ thuật GDR 500 được thể hiện bằng cách sử dụng ảnh GDR 502, một hoặc nhiều ảnh theo sau 504, và ảnh điểm khôi phục 506 nằm trong chuỗi video được tạo mã 508 của luồng bit. Theo một phương án, ảnh GDR 502, các ảnh theo sau 504, và ảnh điểm khôi phục 506 có thể xác định thời gian GDR (GDR period) trong CVS 508. CVS 508 là một loạt các ảnh (hoặc các phần của chúng) bắt đầu với ảnh GDR 502 và chứa tất cả các ảnh (hoặc các phần của chúng) cho đến, nhưng không chứa, ảnh GDR tiếp theo hoặc cho đến cuối của luồng bit. Thời gian GDR là một loạt các ảnh bắt đầu với ảnh GDR 502 và chứa tất cả các ảnh cho đến và chứa ảnh điểm khôi phục 506.

Như được minh họa trên Fig.5, kỹ thuật GDR 500 hoặc nguyên lý hoạt động qua một loạt các ảnh bắt đầu với ảnh GDR 502 và kết thúc với ảnh điểm khôi phục 506. Ảnh GDR 502 chứa vùng được làm mới/sạch 510 chứa các khối mà tất cả các khối này đã được tạo mã bằng cách sử dụng việc dự đoán trong ảnh (nghĩa là, các khối được dự đoán trong ảnh) và vùng không được làm mới/bẩn 512 chứa các khối mà tất cả các khối này đã được tạo mã bằng cách sử dụng việc dự đoán liên ảnh (nghĩa là, các khối được dự đoán liên ảnh).

Ảnh theo sau 504 ngay liền kề với ảnh GDR 502 chứa vùng được làm mới/sạch 510 có phần thứ nhất 510A được tạo mã bằng cách sử dụng việc dự đoán trong ảnh và phần thứ hai 510B được tạo mã bằng cách sử dụng việc dự đoán liên ảnh. Phần thứ hai 510B được tạo mã bằng cách tham chiếu vùng được làm mới/sạch 510 của, ví dụ, ảnh ở trước nằm trong thời gian GDR của CVS 508. Như được minh họa, vùng được làm mới/sạch 510 của các ảnh theo sau 504 mở rộng as quy trình tạo mã di chuyển hoặc tiến triển theo

hướng nhất quán (ví dụ, từ trái sang phải), mà tương ứng làm co vùng không được làm mới/bản 512. Cuối cùng, ảnh điểm khôi phục 506, mà chỉ chứa vùng được làm mới/sạch 510, thu được từ quy trình tạo mã. Đáng lưu ý là, và như sẽ được bàn luận thêm bên dưới, phần thứ hai 510B của vùng được làm mới/sạch 510, mà được tạo mã là các khối được dự đoán liên ảnh, có thể chỉ đề cập đến vùng được làm mới / vùng sạch 510 nội tham chiếu.

Trong HEVC, kỹ thuật GDR 500 trên Fig.5 được hỗ trợ theo cách không quy chuẩn sử dụng thông điệp thông tin tăng cường bổ sung (Supplemental Enhancement Information, SEI) điểm khôi phục và thông điệp SEI thông tin làm mới vùng. Hai thông điệp SEI này không xác định cách GDR được thực hiện. Thay vào đó, hai thông điệp SEI cung cấp một cách đơn giản cơ chế để chỉ báo các ảnh đầu tiên và cuối cùng trong thời gian GDR (nghĩa là, được cung cấp bởi thông điệp SEI điểm khôi phục) và vùng mà được làm mới (nghĩa là, được cung cấp bởi thông điệp SEI thông tin làm mới vùng).

Trong thực tế, kỹ thuật GDR 500 được thực hiện bằng cách sử dụng hai kỹ thuật cùng nhau. Hai kỹ thuật này là dự đoán trong ảnh bắt buộc (constraint intra prediction, CIP) và các bắt buộc bộ mã hóa cho các vectơ chuyển động. CIP có thể được sử dụng cho các mục đích GDR, cụ thể là để tạo mã vùng mà được tạo mã chỉ là các khối được dự đoán trong ảnh (ví dụ, phần thứ nhất 510A của vùng được làm mới/sạch 510) vì CIP cho phép vùng không sử dụng các mẫu từ vùng không được làm mới (ví dụ, vùng không được làm mới/bản 512) để được sử dụng cho việc tham chiếu. Tuy nhiên, việc sử dụng CIP gây ra sự suy giảm hiệu suất tạo mã nghiêm trọng vì bắt buộc đối với các khối nội phải được áp dụng không chỉ cho các khối nội trong vùng được làm mới, mà còn cho tất cả các khối nội nội. Các bắt buộc bộ mã hóa cho các vectơ chuyển động hạn chế bộ mã hóa khỏi việc sử dụng các mẫu bất kỳ trong các ảnh tham chiếu mà nằm bên ngoài vùng được làm mới. Bắt buộc này gây ra tìm kiếm chuyển động không tối ưu.

Fig.6 là hình vẽ sơ lược minh họa tìm kiếm chuyển động không mong muốn 600 khi sử dụng sự hạn chế của bộ mã hóa để hỗ trợ GDR. Như được minh họa, tìm kiếm chuyển động 600 thể hiện ảnh hiện tại 602 và ảnh tham chiếu 604. Mỗi ảnh trong số ảnh hiện tại 602 và ảnh tham chiếu 604 chứa vùng được làm mới 606 được tạo mã với việc dự đoán trong ảnh, vùng được làm mới 608 được tạo mã với việc dự đoán liên ảnh, và vùng không được làm mới 608. Vùng được làm mới 604, vùng được làm mới 606, và

vùng không được làm mới 608 tương tự với phần thứ nhất 510A của vùng được làm mới/sạch 510, phần thứ hai 510B của vùng được làm mới/sạch 510, và vùng không được làm mới/bản 512 trên Fig.5.

Trong quy trình tìm kiếm chuyển động, bộ mã hóa bắt buộc hoặc bị ngăn khỏi việc chọn vector chuyển động 610 bất kỳ mà dẫn đến một số mẫu của khối tham chiếu 612 nằm bên ngoài vùng được làm mới 606. Điều này xảy ra ngay cả khi khối tham chiếu 612 cung cấp tiêu chuẩn chi phí tốc độ-độ méo tốt nhất khi dự đoán khối hiện tại 614 nội hiện tại 602. Do đó, Fig.6 minh họa lí do cho sự không tối ưu trong tìm kiếm chuyển động 600 khi sử dụng sự hạn chế của bộ mã hóa để hỗ trợ GDR.

Các tài liệu JVET JVET-K0212 và JVET-L0160 mô tả việc thực hiện của GDR dựa trên việc sử dụng CIP và phương pháp bắt buộc bộ mã hóa. Việc thực hiện có thể được tóm tắt như sau: chế độ dự đoán trong ảnh bị bắt buộc trên đơn vị tạo mã trên cơ sở cột, việc dự đoán trong ảnh bắt buộc được cho phép để đảm bảo việc tái tạo của CU nội, các vector chuyển động bị bắt buộc để chỉ đến nằm trong vùng được làm mới trong khi tính đến giới hạn bổ sung để tránh việc phát tán lỗi cho các bộ lọc (ví dụ là 6 điểm ảnh), và loại bỏ các ảnh tham chiếu trước khi lặp lại cột nội.

Tài liệu JVET JVET-M0529 đề xuất phương pháp chỉ báo rằng ảnh là đầu tiên và cuối cùng trong thời gian GDR theo cách quy chuẩn. Ý tưởng được đề xuất hoạt động như sau.

Xác định đơn vị NAL mới với việc chỉ báo điểm khôi phục loại đơn vị NAL là đơn vị NAL lớp tạo mã không phải là video (VCL). Phần tải của đơn vị NAL chứa phần tử cú pháp để chỉ rõ thông tin mà có thể được sử dụng để dẫn ra giá trị POC của ảnh cuối cùng trong thời gian GDR. Đơn vị truy cập mà chứa đơn vị NAL không VCL với việc chỉ báo điểm khôi phục loại được gọi là đơn vị truy cập (access point, AU) bắt đầu điểm khôi phục (recovery point begin, RBP) và ảnh trong đơn vị truy cập RBP được gọi là ảnh RBP. Quy trình giải mã có thể bắt đầu từ RBP AU. Khi giải mã bắt đầu từ RBP AU, tất cả các ảnh trong thời gian GDR, ngoại trừ ảnh cuối cùng, không được xuất ra.

Một số vấn đề với thiết kế GDR hiện có được bàn luận.

Các thiết kế/phương pháp hiện có để hỗ trợ GDR có ít nhất các vấn đề sau.

Phương pháp để xác định theo cách quy chuẩn GDR trong JVET-M0529 có các vấn đề sau. Phương pháp được đề xuất không mô tả cách GDR được thực hiện. Thay vào đó,

phương pháp được đề xuất chỉ cung cấp một số tạo tín hiệu để chỉ báo các ảnh đầu tiên và cuối cùng trong thời gian GDR. Để chỉ báo các ảnh đầu tiên và cuối cùng trong thời gian GDR, đơn vị NAL không VCL mới được cần đến. Đây là sự dư khi thông tin được chứa trong đơn vị NAL chỉ báo điểm khôi phục (recovery point indication, RPI) có thể được chứa đơn giản trong phần đầu nhóm phiên của ảnh thứ nhất trong thời gian GDR. Đồng thời, phương pháp được đề xuất không thể mô tả vùng nào trong các ảnh trong thời gian GDR là vùng được làm mới và vùng không được làm mới.

Phương pháp GDR được mô tả trong JVET-K0212 và JVET-L0160 có các vấn đề sau. Trước tiên, việc sử dụng CIP. Cần thiết để tạo mã vùng được làm mới với việc sự đoán nội với một số bắt buộc để ngăn các mẫu bất kỳ khỏi vùng không được làm mới được sử dụng cho việc tham khảo theo không gian. Khi CIP được sử dụng, việc tạo mã là dựa vào ảnh, vốn có nghĩa là tất cả các khối nội nội cũng phải tạo mã là các khối nội CIP. Do vậy, điều này gây ra sự suy giảm hiệu suất. Hơn nữa, việc sử dụng bắt buộc bộ mã hóa để giới hạn tìm kiếm chuyển động ngăn bộ mã hóa khỏi chọn vector chuyển động tốt nhất khi các mẫu của khối tham chiếu được kết hợp với vector chuyển động không hoàn toàn nằm trong vùng được làm mới nội tham chiếu. Đồng thời, vùng được làm mới mà được tạo mã chỉ với việc sự đoán nội không phải là kích thước CTU. Thay vào đó, vùng được làm mới instead có thể go nhỏ hơn kích thước CTU, xuống đến kích thước CU tối thiểu. Việc này làm cho việc thực hiện phức tạp một cách không cần thiết vì nó có thể cần sự chỉ báo ở mức khối

Các kỹ thuật được bộc lộ ở đây để hỗ trợ việc làm mới giải mã từng bước (GDR) trong việc tạo mã video. Các kỹ thuật được bộc lộ cho phép việc làm mới nội tăng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh điểm truy cập ngẫu nhiên nội (intra random access point, IRAP). Cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. Đầu vào bên ngoài có thể là , ví dụ, đầu vào được nhận từ người dùng (ví dụ, người quản lý mạng) bằng giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video 30. Việc thiết đặt cờ thứ nhất và cờ thứ hai theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến

bộ phận hiển thị. Nghĩa là, giá trị của cờ thứ nhất và cờ thứ hai kiểm soát liệu dữ liệu bản tiềm tàng từ ảnh GDR được xuất ra hay không hoặc liệu bộ giải mã video sẽ chờ đồng bộ hóa hoàn toàn để bắt đầu hiển thị dữ liệu hay không. Bằng cách có khả năng hạn chế dữ liệu bản khỏi việc được xuất ra ra, bộ tạo mã/bộ giải mã (còn được biết đến là, “codec”) trong việc tạo mã video được cải thiện liên quan đến các codec hiện tại. Đối với vấn đề thực hành, quy trình tạo mã video được cải thiện tạo cho người dùng trải nghiệm người dùng tốt hơn khi các video được gửi, nhận và/hoặc xem.

Để giải quyết một hoặc nhiều vấn đề được bàn luận ở trên, sáng chế bộc lộ các khía cạnh sau. Mỗi khía cạnh trong số các khía cạnh có thể được áp dụng riêng lẻ, và một số trong các khía cạnh này có thể được áp dụng theo cách kết hợp.

1) Đơn vị VCL NAL với loại GDR_NUT được xác định.

a. Các ảnh với loại đơn vị NAL GDR_NUT được gọi là ảnh GDR, nghĩa là, ảnh thứ nhất trong thời gian GDR.

b. Ảnh GDR có temporalID bằng 0.

c. Đơn vị truy cập chứa ảnh GDR được gọi là đơn vị truy cập GDR. Như được lưu ý ở trên, đơn vị truy cập là bộ các đơn vị NAL. Mỗi đơn vị NAL có thể chứa ảnh đơn.

2) Chuỗi video được tạo mã (coded video sequence, CVS) có thể bắt đầu với đơn vị truy cập GDR.

3) Đơn vị truy cập GDR là đơn vị truy cập thứ nhất trong CVS khi một trong các điều sau là đúng:

a. Đơn vị truy cập GDR là đơn vị truy cập thứ nhất trong luồng bit.

b. Đơn vị truy cập GDR ngay theo sau đơn vị truy cập kết thúc chuỗi (end-of-sequence, EOS).

c. Đơn vị truy cập GDR ngay theo sau đơn vị truy cập kết thúc luồng bit (end-of-bitstream, EOB).

d. Cờ bộ giải mã mà được gọi là NoIncorrectPicOutputFlag được kết hợp với ảnh GDR và giá trị của cờ được thiết đặt bằng 1 (nghĩa là, đúng) bởi thực thể bên ngoài bộ giải mã.

4) Khi ảnh GDR là đơn vị truy cập thứ nhất trong CVS, thì sau đây áp dụng:

a. Tất cả các ảnh tham chiếu trong DPB được đánh dấu là “không được sử dụng cho việc tham chiếu”.

b. POC MSB của ảnh được thiết đặt bằng 0.

c. Ảnh GDR và tất cả các ảnh mà theo sau ảnh GDR theo thứ tự xuất ra cho đến ảnh cuối cùng trong thời gian GDR, ngoại trừ ảnh cuối cùng trong thời gian GDR, không được xuất ra (nghĩa là, được đánh dấu là “không cần xuất ra”).

5) Cờ để chỉ rõ liệu GDR được cho phép được tạo tín hiệu trong tập hợp thông số mức chuỗi (ví dụ, trong SPS) hay không.

a. Cờ có thể được chỉ định là `gdr_enabled_flag`.

b. Khi cờ bằng 1, thì ảnh GDR có thể có mặt trong CVS. Ngược lại, khi cờ bằng 0, GDR không được cho phép sao cho ảnh GDR không có mặt trong CVS.

6) Thông tin mà có thể được sử dụng để dẫn ra giá trị POC của ảnh cuối cùng trong thời gian GDR được tạo tín hiệu trong phần đầu nhóm phiên của ảnh GDR.

a. Thông tin được tạo tín hiệu là POC delta giữa ảnh cuối cùng trong thời gian GDR và ảnh GDR. Thông tin có thể được tạo tín hiệu sử dụng phần tử cú pháp được chỉ định là `recovery_point_cnt`.

b. Sự có mặt của phần tử cú pháp `recovery_point_cnt` trong phần đầu nhóm phiên có thể có điều kiện dựa trên giá trị của cờ `gdr_enabled` và loại đơn vị NAL của ảnh, nghĩa là, cờ có mặt chỉ khi `gdr_enabled_flag` bằng 1 và `nal_unit_type` của đơn vị NAL chứa nhóm phiên là `GDR_NUT`.

7) Cờ để chỉ rõ việc liệu nhóm phiên là một phần của vùng được làm mới hay không được tạo tín hiệu trong phần đầu nhóm phiên.

a. Cờ có thể được chỉ định là `refreshed_region_flag`.

b. Sự có mặt của cờ có thể có điều kiện dựa trên giá trị của `gdr_enabled_flag` và việc liệu ảnh chứa nhóm phiên nằm trong thời gian GDR hay không. Do đó, cờ có mặt chỉ khi tất cả các điều sau là đúng:

i. Giá trị của `gdr_enabled_flag` bằng 1.

ii. POC của ảnh hiện tại bằng hoặc lớn hơn giá trị POC của ảnh GDR cuối cùng (khi ảnh hiện tại là ảnh GDR, ảnh GDR cuối cùng là ảnh hiện tại) và nhỏ hơn POC của ảnh cuối cùng trong thời gian GDR.

c. Khi cờ không có mặt trong phần đầu nhóm phiên, thì giá trị của cờ được suy ra là bằng 1.

8) Tất cả các nhóm phiên có refreshed_region_flag bằng 1 bao phủ vùng được kết nối. Tương tự, tất cả các nhóm phiên có refreshed_region_flag bằng 0 cũng bao phủ vùng được kết nối. 9) Nhóm phiên với refreshed_region_flag có thể là loại I (nghĩa là, nhóm phiên nội) hoặc B- hoặc P- (nghĩa là, nhóm phiên liên thông).

10) Mỗi ảnh bắt đầu từ ảnh GDR đến ảnh cuối cùng trong thời gian GDR chứa ít nhất một nhóm phiên với refreshed_region_flag bằng 1.

11) Ảnh GDR chứa ít nhất một nhóm phiên với refreshed_region_flag bằng 1 và tile_group_type bằng I (nghĩa là, nhóm phiên nội).

12) Khi gdr_enabled_flag bằng 1, thì thông tin của nhóm phiên hình chữ nhật, nghĩa là, số lượng nhóm phiên và các địa chỉ của chúng, được cho phép để được tạo tín hiệu trong tập hợp thông số ảnh (picture parameter set, PPS) hoặc trong phần đầu nhóm phiên. Để làm điều này, cờ được tạo tín hiệu trong PPS để chỉ rõ liệu thông tin nhóm phiên hình chữ nhật có mặt trong PPS hay không. Cờ này có thể được gọi là rect_tile_group_info_in_pps_flag. Cờ này có thể bị bắt buộc bằng 1 khi gdr_enabled_flag bằng 1.

a. Theo giải pháp khác, thay vì tạo tín hiệu liệu thông tin nhóm phiên hình chữ nhật có mặt trong PPS hay không, cờ chung hơn có thể được tạo tín hiệu trong PPS để chỉ rõ liệu nhóm phiên thông tin (nghĩa là, các loại bất kỳ của nhóm phiên – như nhóm phiên hình chữ nhật, nhóm phiên quét mảnh) có mặt trong PPS.

13) Khi nhóm phiên thông tin không có mặt trong PPS, thì nó có thể bị bắt buộc thêm rằng việc tạo tín hiệu của thông tin nhận dạng (identifier, ID) nhóm phiên tường minh không có mặt. Thông tin ID nhóm phiên tường minh chứa: signaled_tile_group_id_flag, signaled_tile_group_id_length_minus1, và tile_group_id[i].

14) Cờ được tạo tín hiệu để chỉ rõ liệu hoạt động lọc vòng qua các biên giữa vùng được làm mới và vùng không được làm mới nội được cho phép hay không.

a. Cờ này có thể được tạo tín hiệu trong PPS và được gọi là loop_filter_across_refreshed_region_enabled_flag.

b. Sự có mặt của loop_filter_across_refreshed_region_enabled_flag có thể có điều kiện dựa trên giá trị của loop_filter_across_tile_enabled_flag. Khi loop_filter_across_tile_enabled_flag bằng 0, thì

`loop_filter_across_refreshed_region_enabled_flag` có thể không có mặt và giá trị của nó được suy ra là bằng 0.

c. Theo giải pháp khác, cờ có thể được tạo tín hiệu trong phần đầu nhóm phiên và sự có mặt của nó có thể là điều kiện dựa trên giá trị của `refreshed_region_flag`, nghĩa là, cờ có mặt chỉ khi giá trị của `refreshed_region_flag` bằng 1.

15) Khi nó được chỉ báo rằng nhóm phiên là vùng được làm mới và nó được chỉ báo rằng bộ lọc vòng qua vùng được làm mới không được cho phép, thì sau đây áp dụng:

a. Việc khử khối của gờ ở đường biên của nhóm phiên không được thực hiện khi nhóm phiên lân cận mà có chung gờ là nhóm phiên không được làm mới.

b. Quy trình độ lệch thích ứng mẫu (sample adaptive offset, SAO) cho khối ở biên của nhóm phiên không sử dụng các mẫu bất kỳ từ bên ngoài biên vùng được làm mới.

c. Quy trình lọc vòng thích ứng (adaptive loop filtering, ALF) cho khối ở biên của nhóm phiên không sử dụng các mẫu bất kỳ từ bên ngoài biên vùng được làm mới.

16) Khi `gdr_enabled_flag` bằng 1, mỗi ảnh được kết hợp với các biến để xác định các biên của vùng được làm mới nội. Các biến này có thể được gọi là:

a. `PicRefreshedLeftBoundaryPos` cho vị trí biên bên trái của vùng được làm mới nội.

b. `PicRefreshedRightBoundaryPos` cho vị trí biên bên phải của vùng được làm mới nội.

c. `PicRefreshedTopBoundaryPos` cho vị trí biên trên cùng của vùng được làm mới nội.

d. `PicRefreshedBotBoundaryPos` cho vị trí biên dưới cùng của vùng được làm mới nội.

17) Các biên của vùng được làm mới nội có thể được dẫn ra. Các biên của vùng được làm mới của ảnh được cập nhật bởi bộ giải mã sau khi phần đầu nhóm phiên được phân tích và giá trị của `refreshed_region_flag` của nhóm phiên bằng 1.

18) Theo giải pháp khác với giải pháp 17), các biên của vùng được làm mới nội được tạo tín hiệu tường minh trong mỗi nhóm phiên của ảnh.

a. Cờ có thể được tạo tín hiệu để chỉ báo liệu ảnh mà nhóm phiên thuộc về có chứa vùng không được làm mới hay không. Khi nó được chỉ rõ rằng ảnh không chứa vùng

không được làm mới, thì không có thông tin các biên được làm mới được tạo tín hiệu và có thể đơn giản được suy ra là bằng ảnh các biên.

19) Đối với ảnh hiện tại, các biên của vùng được làm mới được sử dụng trong quy trình lọc trong vòng lặp như sau:

a. Đối với quy trình khử khối, để xác định gờ của vùng được làm mới để quyết định liệu gờ cần được khử khối hay không.

b. Đối với quy trình SAO, để xác định đường biên của vùng làm mới sao cho quy trình cắt có thể được áp dụng để tránh sử dụng các mẫu từ vùng không được làm mới khi bộ lọc vòng qua vùng được làm mới không được cho phép.

c. Đối với quy trình ALF, để xác định đường biên của vùng làm mới sao cho quy trình cắt có thể được áp dụng để tránh sử dụng các mẫu từ vùng không được làm mới khi bộ lọc vòng qua vùng được làm mới không được cho phép.

20) Đối với quy trình bù chuyển động, thông tin về các biên của vùng được làm mới, cụ thể là các biên của vùng được làm mới nội tham chiếu, được sử dụng như sau: khi khối hiện tại nội hiện tại trong nhóm phiên với refreshed_region_flag bằng 1 và khối tham chiếu nội tham chiếu mà chứa vùng không được làm mới, thì dưới đây áp dụng:

a. Vector chuyển động từ khối hiện tại đến ảnh tham chiếu đó được cắt bởi các biên của vùng được làm mới nội tham chiếu đó.

b. Đối với bộ lọc nội suy phân số cho các mẫu nội tham chiếu đó, nó được cắt bởi các biên của vùng được làm mới nội tham chiếu đó.

Mô tả chi tiết sáng chế của các phương án của sáng chế được cung cấp. Phần mô tả liên quan đến văn bản cơ sở, mà là tài liệu JVET JVET-M1001-v5. Nghĩa là, chỉ delta được mô tả, trong khi các văn bản trong văn bản cơ sở mà không được đề cập bên dưới áp dụng như bản thân các văn bản đó. Văn bản được sửa đổi so với văn bản cơ sở được in nghiêng.

Các định nghĩa được cung cấp.

3.1 ảnh truy cập ngẫu nhiên sạch (clean random access, CRA): Ảnh IRAP mà cho nó mỗi đơn vị VCL NAL có nal_unit_type bằng CRA_NUT.

Lưu ý – Ảnh CRA không đề cập đến các ảnh bất kỳ ngoài bản thân nó cho việc dự đoán liên ảnh trong quy trình giải mã của nó, và có thể là ảnh thứ nhất trong luồng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong luồng bit. Ảnh CRA có thể có RADL

được kết hợp hoặc các ảnh RASL. Khi ảnh CRA có *NoIncorrectPicOutputFlag* bằng 1, thì các ảnh RASL được kết hợp không được xuất ra bởi bộ giải mã, vì các ảnh RASL có thể là không giải mã được, khi chúng có thể chứa các tham chiếu đến các ảnh mà không có mặt trong luồng bit.

3.2 chuỗi video được tạo mã (coded video sequence, CVS): Chuỗi của các đơn vị truy cập mà bao gồm, theo thứ tự giải mã, đơn vị truy cập IRAP với *NoIncorrectPicOutputFlag* bằng 1 hoặc đơn vị truy cập GDR với *NoIncorrectPicOutputFlag* bằng 1, được theo sau bởi 0 hoặc nhiều đơn vị truy cập hơn mà không phải là các đơn vị truy cập IRAP với *NoIncorrectPicOutputFlag* bằng 1 hoặc đơn vị truy cập GDR với *NoIncorrectPicOutputFlag* bằng 1, chứa tất cả các đơn vị truy cập tiếp theo lên đến nhưng không chứa đơn vị truy cập tiếp theo bất kỳ mà là đơn vị truy cập IRAP với *NoIncorrectPicOutputFlag* bằng 1 hoặc đơn vị truy cập GDR với *NoIncorrectPicOutputFlag* bằng 1.

Lưu ý 1 – Đơn vị truy cập IRAP có thể là đơn vị truy cập IDR hoặc đơn vị truy cập CRA. Giá trị của *NoIncorrectPicOutputFlag* bằng 1 cho mỗi đơn vị truy cập IDR và mỗi đơn vị truy cập CRA mà là đơn vị truy cập thứ nhất trong luồng bit theo thứ tự giải mã, là đơn vị truy cập thứ nhất mà theo sau đơn vị NAL kết thúc chuỗi theo thứ tự giải mã, hoặc có *HandleCraAsCvsStartFlag* bằng 1.

Lưu ý 2 – Giá trị của *NoIncorrectPicOutputFlag* bằng 1 cho mỗi đơn vị truy cập GDR mà là đơn vị truy cập thứ nhất trong luồng bit theo thứ tự giải mã, là đơn vị truy cập thứ nhất mà theo sau đơn vị NAL kết thúc chuỗi theo thứ tự giải mã, hoặc có *HandleGdrAsCvsStartFlag* bằng 1.

3.3 đơn vị truy cập làm mới giải mã từng bước (gradual decoding refresh, GDR): Đơn vị truy cập trong đó ảnh được tạo mã là ảnh GDR.

3.4 ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR): Ảnh mà cho nó mỗi đơn vị VCL NAL có *nal_unit_type* bằng *GDR_NUT*.

3.5 ảnh dẫn được bỏ qua truy cập ngẫu nhiên (random access skipped leading, RASL): Ảnh được tạo mã mà cho nó mỗi đơn vị VCL NAL có *nal_unit_type* bằng *RASL_NUT*.

Lưu ý – Tất cả các ảnh RASL là các ảnh dẫn của ảnh CRA được kết hợp. Khi ảnh CRA được kết hợp có *NoIncorrectPicOutputFlag* bằng 1, thì ảnh RASL không được xuất

ra và có thể không giải mã được đúng, vì ảnh RASL có thể chứa các tham chiếu đến các ảnh mà không có mặt trong luồng bit. Các ảnh RASL không được sử dụng làm các ảnh tham chiếu cho quy trình giải mã của các ảnh không phải là RASL. Khi có mặt, tất cả các ảnh RASL ở trước, theo thứ tự giải mã, tất cả các ảnh theo sau của cùng ảnh CRA được kết hợp.

Các ngữ nghĩa và cú pháp phân tải chuỗi byte thô (raw byte sequence payload, RBSP) tập hợp thông số chuỗi

seq_parameter_set_rbsp() {	Descriptor (Bộ mô tả)
...	
<i>gdr_enabled_flag</i>	<i>u(1)</i>
...	

gdr_enabled_flag bằng 1 chỉ rõ các ảnh GDR có thể có mặt trong chuỗi video được tạo mã. *gdr_enabled_flag* bằng 0 chỉ rõ rằng các ảnh GDR không có mặt trong chuỗi video được tạo mã.

Các ngữ nghĩa và cú pháp RBSP tập hợp thông số ảnh.

pic_parameter_set_rbsp() {	Descriptor (Bộ mô tả)
...	
<i>single_tile_in_pic_flag</i>	<i>u(1)</i>
if(!single_tile_in_pic_flag) {	
...	<i>ue(v)</i>
<i>single_tile_per_tile_group_flag</i>	<i>u(1)</i>
if(!single_tile_per_tile_group_flag)	
<i>rect_tile_group_flag</i>	<i>u(1)</i>
if(rect_tile_group_flag)	
<i>rect_tile_group_info_in_pps_flag</i>	<i>u(1)</i>
if(rect_tile_group_flag && !single_tile_per_tile_group_flag && <i>rect_tile_group_info_in_pps_flag</i>) {	
<i>num_tile_groups_in_pic_minus1</i>	<i>ue(v)</i>
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
<i>top_left_tile_idx[i]</i>	<i>u(v)</i>
<i>bottom_right_tile_idx[i]</i>	<i>u(v)</i>

}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
if(loop_filter_across_tiles_enabled_flag)	
loop_filter_across_tile_groups_enabled_flag	u(1)
if(loop_filter_across_tile_groups_enabled_flag)	
loop_filter_across_refreshed_region_enabled_flag	u(1)
}	
...	

rect_tile_group_info_in_pps_flag bằng 1 chỉ rõ rằng thông tin nhóm phiên hình chữ nhật được tạo tín hiệu trong PPS. *rect_tile_group_info_in_pps_flag* bằng 0 chỉ rõ rằng thông tin nhóm phiên hình chữ nhật không được tạo tín hiệu trong PPS.

Nó là yêu cầu về sự hợp cách luồng bit mà giá trị của *rect_tile_group_info_in_pps_flag* sẽ bằng 0 khi giá trị của *gdr_enabled_flag* trong SPS hoạt động bằng 0.

loop_filter_across_refreshed_region_enabled_flag bằng 1 chỉ rõ rằng các hoạt động lọc trong vòng lặp có thể được thực hiện qua các biên của nhóm phiên với *refreshed_region_flag* bằng 1 trong các ảnh tham chiếu đến PPS. *loop_filter_across_refreshed_region_enabled_flag* bằng 0 chỉ rõ rằng các hoạt động lọc trong vòng lặp không được thực hiện qua các biên của nhóm phiên với *refreshed_region_flag* bằng 1 trong các ảnh tham chiếu đến PPS. Các hoạt động lọc trong vòng lặp bao gồm các hoạt động của bộ lọc khử khối, bộ lọc dịch vị thích ứng mẫu, và bộ lọc vòng thích ứng. Khi không có mặt, thì giá trị của *loop_filter_across_refreshed_region_enabled_flag* được suy ra là bằng 0.

Việc *signalled_tile_group_id_flag* bằng 1 chỉ rõ rằng ID nhóm phiên cho mỗi nhóm phiên được tạo tín hiệu. Việc *signalled_tile_group_index_flag* bằng 0 chỉ rõ rằng các ID nhóm phiên không được tạo tín hiệu. Khi không có mặt, thì giá trị của *signalled_tile_group_index_flag* được suy ra là bằng 0.

signalled_tile_group_id_length_minus1 plus 1 chỉ rõ số lượng bit được sử dụng để biểu diễn phần tử cú pháp *tile_group_id[i]* khi có mặt, và phần tử cú pháp *tile_group_address* trong phần đầu các nhóm phiên. Giá trị của

signalled_tile_group_index_length_minus1 sẽ nằm trong khoảng từ 0 đến 15, không loại trừ. Khi không có mặt, thì giá trị của signalled_tile_group_index_length_minus1 được suy ra như sau:

Nếu $rect_tile_group_info_in_pps_flag$ bằng 1, thì
 $Ceil(Log2(num_tile_groups_in_pic_minus1 + 1)) - 1$.

Ngược lại, thì $Ceil(Log2(NumTilesInPic)) - 1$

Các ngữ nghĩa và cú pháp phần đầu nhóm phiên chung.

tile_group_header() {	Descriptor (Bộ mô tả)
...	
if(rect_tile_group_flag NumTilesInPic > 1)	
tile_group_address	u(v)
if(!rect_tile_group_info_in_pps_flag && !single_tile_per_tile_group_flag) {	
if(rect_tile_group_flag)	
bottom_right_tile_id	u(v)
else	
num_tiles_in_tile_group_minus1	ue(v)
}	
tile_group_type	ue(v)
tile_group_pic_order_cnt_lsb	u(v)
if(gdr_enabled_flag) {	
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	se(v)
if(PicOrderCntVal >= LastGDRPocVal && PicOrderCntVal < RecoveryPointPocVal)	
refreshed_region_flag	u(1)
}	
...	)

tile_group_address chỉ rõ địa chỉ phiên của phiên thứ nhất trong nhóm phiên. Khi không có mặt, thì giá trị của tile_group_address được suy ra là bằng 0.

Nếu rect_tile_group_flag bằng 0, thì áp dụng như sau:

tile_group_address là ID phiên như được chỉ rõ bởi phương trình 6-7.

Chiều dài của tile_group_address là $Ceil(Log2(NumTilesInPic))$ bit.

Giá trị của `tile_group_address` sẽ trong khoảng từ 0 đến `NumTilesInPic - 1`, không loại trừ.

Nếu không thì nếu `rect_tile_group_flag` bằng 1 và `rect_tile_group_info_in_pps` bằng 0, thì áp dụng như sau:

`tile_group_address` là chỉ số phiên của phiên ở góc trên cùng bên trái của nhóm phiên thứ i .

Chiều dài của `tile_group_address` là `signalled_tile_group_index_length_minus1 + 1` bits.

Nếu `signalled_tile_group_id_flag` bằng 0, thì giá trị của `tile_group_address` sẽ trong khoảng từ 0 đến `NumTilesInPic - 1`, không loại trừ. Ngược lại, giá trị của `tile_group_address` sẽ trong khoảng từ 0 đến $2^{(\text{signalled_tile_group_index_length_minus1} + 1)} - 1$, không loại trừ.

Ngược lại (`rect_tile_group_flag` bằng 1 và `rect_tile_group_info_in_pps` bằng 1), thì áp dụng như sau:

`tile_group_address` là ID nhóm phiên của nhóm phiên.

Chiều dài của `tile_group_address` là `signalled_tile_group_index_length_minus1 + 1` bits.

Nếu `signalled_tile_group_id_flag` bằng 0, thì giá trị của `tile_group_address` sẽ trong khoảng từ 0 đến `num_tile_groups_in_pic_minus1`, không loại trừ. Ngược lại, giá trị của `tile_group_address` sẽ trong khoảng từ 0 đến $2^{(\text{signalled_tile_group_index_length_minus1} + 1)} - 1$, không loại trừ.

`bottom_right_tile_id` chỉ rõ chỉ số phiên của phiên nằm ở góc dưới cùng bên phải của nhóm phiên. Khi `single_tile_per_tile_group_flag` bằng 1, thì `bottom_right_tile_id` được suy ra là bằng `tile_group_address`. Chiều dài của phần tử cú pháp `bottom_right_tile_id` là $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bits.

Biến `NumTilesInCurrTileGroup`, mà chỉ rõ số lượng của các phiên trong nhóm phiên hiện tại, `TopLeftTileIdx` mà chỉ rõ chỉ số phiên của phiên trên cùng bên trái của nhóm phiên, `BottomRightTileIdx` mà chỉ rõ chỉ số phiên của phiên dưới cùng bên phải của nhóm phiên, và `TgTileIdx[ i ]`, mà chỉ rõ chỉ số phiên của phiên thứ i trong nhóm phiên hiện tại, được dẫn ra như sau:

```
if( rect_tile_group_flag ) {
```

```

if ( tile_group_info_in_pps ) {
    tileGroupIdx = 0
    while( tile_group_address != rect_tile_group_id[ tileGroupIdx ] )
        tileGroupIdx++
    tileIdx = top_left_tile_idx[ tileGroupIdx ]
    BottomRightTileIdx = bottom_right_tile_idx[ tileGroupIdx ]
} else {
    tileIdx = tile_group_address
    BottomRightTileIdx = bottom_right_tile_id
}
TopLeftTileIdx = tileIdx
deltaTileIdx = BottomRightTileIdx - TopLeftTileIdx
NumTileRowsInTileGroupMinus1 = deltaTileIdx /
( num_tile_columns_minus1 + 1 ) (7-35)
NumTileColumnsInTileGroupMinus1 = deltaTileIdx %
( num_tile_columns_minus1 + 1 )
NumTilesInCurrTileGroup = ( NumTileRowsInTileGroupMinus1 + 1 ) *
( NumTileColumnsInTileGroupMinus1 + 1 )
for( j = 0, tIdx = 0; j < NumTileRowsInTileGroupMinus1 + 1;
    j++, tileIdx +=
num_tile_columns_minus1 + 1 )
    for( i = 0, currTileIdx = tileIdx; i <
NumTileColumnsInTileGroupMinus1 + 1;
        i++, currTileIdx++, tIdx++ )
        TgTileIdx[ tIdx ] = currTileIdx
} else {
    NumTilesInCurrTileGroup = num_tiles_in_tile_group_minus1 + 1
    TgTileIdx[ 0 ] = tile_group_address
    for( i = 1; i < NumTilesInCurrTileGroup; i++ )
        TgTileIdx[ i ] = TgTileIdx[ i - 1 ] + 1
}

```

recovery_poc_cnt chỉ rõ điểm khôi phục của các ảnh được giải mã theo thứ tự xuất ra. Nếu có ảnh *picA* mà theo sau ảnh hiện tại (nghĩa là, ảnh GDR) theo thứ tự giải mã trong CVS và ảnh đó có *PicOrderCntVal* bằng *PicOrderCntVal* của ảnh hiện tại cộng với giá trị của *recovery_poc_cnt*, thì ảnh *picA* được gọi là ảnh điểm khôi phục. Ngược lại, ảnh thứ nhất theo thứ tự xuất ra mà có *PicOrderCntVal* lớn hơn *PicOrderCntVal* của ảnh hiện tại cộng với giá trị của *recovery_poc_cnt* được gọi là ảnh điểm khôi phục. Ảnh điểm khôi phục sẽ không ở trước ảnh hiện tại theo thứ tự giải mã. Tất cả các ảnh được giải mã theo thứ tự xuất ra được chỉ báo là đúng hoặc gần đúng về nội dung bắt đầu ở vị trí thứ tự xuất của ảnh điểm khôi phục. Giá trị của *recovery_poc_cnt* sẽ nằm trong khoảng từ $-MaxPicOrderCntLsb / 2$ đến $MaxPicOrderCntLsb / 2 - 1$, không loại trừ.

Giá trị *RecoveryPointPocVal* được dẫn ra như sau:

$$RecoveryPointPocVal = PicOrderCntVal + recovery_poc_cnt$$

refreshed_region_flag bằng 1 chỉ rõ rằng việc giải mã của nhóm phiên tạo ra các giá trị các mẫu được tái tạo đúng bất kể liệu giá trị của *NoIncorrectPicOutputFlag* của GDR được kết hợp hay không. **refreshed_region_flag** bằng 0 chỉ rõ rằng việc giải mã của nhóm phiên có thể tạo ra các giá trị các mẫu được tái tạo không đúng khi nó bắt đầu từ GDR được kết hợp. với *NoIncorrectPicOutputFlag* bằng 1. Khi không có mặt, thì giá trị của **refreshed_region_flag** được suy ra là bằng 1.

Lưu ý x – bản thân ảnh hiện tại có thể là ảnh GDR với *NoIncorrectPicOutputFlag* bằng 1.

Các biên được làm mới nhóm phiên được dẫn ra như sau:

$$tileColIdx = TopLeftTileIdx \% (num_tile_columns_minus1 + 1)$$

$$tileRowIdx = TopLeftTileIdx / (num_tile_columns_minus1 + 1)$$

$$TGRefreshedLeftBoundary = ColBd[tileColIdx] << CtbLog2SizeY$$

$$TGRefreshedTopBoundary = RowBd[tileRowIdx] << CtbLog2SizeY$$

$$tileColIdx = BottomRightTileIdx \% (num_tile_columns_minus1 + 1)$$

$$tileRowIdx = BottomRightTileIdx / (num_tile_columns_minus1 + 1)$$

$$TGRefreshedRightBoundary = ((ColBd[tileColIdx] + ColWidth[tileColIdx]) << CtbLog2SizeY) - 1$$

$$TGRefreshedRightBoundary = TGRefreshedRightBoundary > pic_width_in_luma_samples ?$$

$pic_width_in_luma_samples$:

$TGRefreshedRightBoundary$

$TGRefreshedBotBoundary =$ (

$(RowBd[tileRowIdx] + RowHeight[tileRowIdx]) << CtbLog2SizeY) - 1$

$TGRefreshedBotBoundary = TGRefreshedBotBoundary >$

$pic_height_in_luma_samples ?$

$pic_height_in_luma_samples$:

$TGRefreshedBotBoundary$

Các ngữ nghĩa phần đầu đơn vị NAL.

Bảng 7-1 – các mã loại đơn vị NAL và các lớp loại đơn vị NAL

nal_unit_type	Tên của nal_unit_type	Nội dung của đơn vị NAL và cấu trúc cú pháp RBSP	Đơn vị NAL Lớp loại
0	TRAIL_NUT	nhóm phiên được tạo mã của ảnh theo sau không phải là STSA $tile_group_layer_rbsp()$	VCL
1	STSA_NUT	Nhóm phiên được tạo mã của ảnh STSA $tile_group_layer_rbsp()$	VCL
2	RASL_NUT	Nhóm phiên được tạo mã của ảnh RASL $tile_group_layer_rbsp()$	VCL
3	RADL_NUT	Nhóm phiên được tạo mã của ảnh RADL $tile_group_layer_rbsp()$	VCL
4..7	RSV_VCL_4.. RSV_VCL_7	Các loại đơn vị VCL NAL không phải là IRAP được bảo lưu	VCL
8 9	IDR_W_RADL IDR_N_LP	Nhóm phiên được tạo mã của ảnh IDR $tile_group_layer_rbsp()$	VCL
10	CRA_NUT	Nhóm phiên được tạo mã của ảnh CRA $tile_group_layer_rbsp()$	VCL
11 12	RSV_IRAP_VC L11 RSV_IRAP_VC L12	Các loại đơn vị VCL NAL IRAP được bảo lưu	VCL
13	GDR_NUT	nhóm phiên được tạo mã của ảnh GDR $tile_group_layer_rbsp()$	VCL
14..15	RSV_VCL14.. RSV_VCL15	Các loại đơn vị VCL NAL không phải là IRAP được bảo lưu	VCL

16	SPS_NUT	Tập hợp thông số chuỗi seq_parameter_set_rbsp()	không phải VCL
17	PPS_NUT	Tập hợp thông số ảnh pic_parameter_set_rbsp()	không phải VCL
18	APS_NUT	Tập hợp thông số thích ứng adaptation_parameter_set_rbsp()	không phải VCL
19	AUD_NUT	Dấu phân cách đơn vị truy cập access_unit_delimiter_rbsp()	không phải VCL
20	EOS_NUT	Kết thúc chuỗi end_of_seq_rbsp()	không phải VCL
21	EOB_NUT	Kết thúc luồng bit end_of_bitstream_rbsp()	không phải VCL
22, 23	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin tăng cường bổ sung sei_rbsp()	không phải VCL
24..27	RSV_NVCL24. RSV_NVCL27	Các loại đơn vị NAL không VCL được bảo lưu	không phải VCL
28..31	UNSPEC28.. UNSPEC31	Các loại đơn vị NAL không VCL không được chỉ rõ	không phải VCL

...

Khi nal_unit_type bằng GDR_NUT, thì nhóm phiên được tạo mã thuộc về ảnh GDR, TemporalId sẽ bằng 0.

Thứ tự của các đơn vị truy cập và sự kết hợp với các CVS được bàn luận.

Luồng bit tuân theo bản mô tả này (nghĩa là, tài liệu JVET JVET-M1001-v5) chứa một hoặc nhiều CVS.

CVS chứa một hoặc nhiều đơn vị truy cập. Thứ tự của các đơn vị NAL và các ảnh được tạo mã và sự kết hợp của chúng với các đơn vị truy cập được mô tả trong điều 7.4.2.4.4.

Đơn vị truy cập thứ nhất của CVS là một đơn vị trong số các đơn vị sau:

- đơn vị truy cập IRAP với NoBrokenPictureOutputFlag bằng 1.
- đơn vị truy cập GDR với NoIncorrectPicOutputFlag bằng 1.

Nó là yêu cầu về sự hợp cách luồng bit mà, khi có mặt, đơn vị truy cập tiếp theo sau khi đơn vị truy cập mà chứa đơn vị NAL kết thúc chuỗi hoặc đơn vị NAL kết thúc luồng bit sẽ là một đơn vị trong số các đơn vị sau:

- đơn vị truy cập IRAP, mà có thể là đơn vị truy cập IDR hoặc đơn vị truy cập CRA.*
- đơn vị truy cập GDR.*

8.1.1 Quy trình giải mã cho ảnh được tạo mã được bàn luận.

...

Khi ảnh hiện tại là ảnh IRAP, thì áp dụng như sau:

– Khi ảnh hiện tại là ảnh IDR, ảnh thứ nhất trong luồng bit theo thứ tự giải mã, hoặc ảnh thứ nhất mà theo sau đơn vị NAL kết thúc chuỗi theo thứ tự giải mã, biến *NoIncorrectPicOutputFlag* được thiết đặt bằng 1.

– Ngược lại, khi một số phương tiện bên ngoài không được chỉ rõ trong bản mô tả này (ví dụ, đầu vào người dùng) có sẵn để thiết đặt biến *HandleCraAsCvsStartFlag* thành giá trị cho ảnh hiện tại, biến *HandleCraAsCvsStartFlag* được thiết đặt bằng giá trị được cung cấp bởi phương tiện bên ngoài và biến *NoIncorrectPicOutputFlag* được thiết đặt bằng *HandleCraAsCvsStartFlag*.

– Ngược lại, biến *HandleCraAsCvsStartFlag* được thiết đặt bằng 0 và biến *NoIncorrectPicOutputFlag* được thiết đặt bằng 0.

Khi ảnh hiện tại là ảnh GDR, thì áp dụng như sau:

– Khi ảnh hiện tại là ảnh GDR, ảnh thứ nhất trong luồng bit theo thứ tự giải mã, hoặc ảnh thứ nhất mà theo sau đơn vị NAL kết thúc chuỗi theo thứ tự giải mã, biến *NoIncorrectPicOutputFlag* được thiết đặt bằng 1.

– Ngược lại, khi một số phương tiện bên ngoài không được chỉ rõ trong bản mô tả này có sẵn để thiết đặt biến *HandleGdrAsCvsStartFlag* thành giá trị cho ảnh hiện tại, biến *HandleGdrAsCvsStartFlag* được thiết đặt bằng giá trị được cung cấp bởi phương tiện bên ngoài và biến *NoIncorrectPicOutputFlag* được thiết đặt bằng *HandleGdrAsCvsStartFlag*.

– Ngược lại, biến *HandleGdrAsCvsStartFlag* được thiết đặt bằng 0 và biến *NoIncorrectPicOutputFlag* được thiết đặt bằng 0.

...

Quy trình giải mã hoạt động như sau đối với ảnh hiện tại *CurrPic*:

1. Việc giải mã của các đơn vị NAL được chỉ rõ trong điều 8.2.
2. Các quy trình trong điều 8.3 chỉ rõ các quy trình giải mã sau sử dụng các phần tử cú pháp trong lớp phân đầu nhóm phiên trở lên:
 - Các biến và các chức năng liên quan đến đếm thứ tự ảnh được dẫn ra như được chỉ rõ trong điều 8.3.1. Điều này chỉ cần được gọi cho nhóm ảnh xếp lớp thứ nhất của ảnh.
 - Ở bắt đầu của quy trình giải mã cho mỗi nhóm phiên của ảnh không IDR, quy trình giải mã cho việc xây dựng các danh sách ảnh tham chiếu được chỉ rõ trong điều 8.3.2 được gọi cho việc dẫn ra của danh sách ảnh tham chiếu 0 (`RefPicList[ 0 ]`) và danh sách ảnh tham chiếu 1 (`RefPicList[ 1 ]`).
 - Quy trình giải mã cho việc đánh dấu ảnh tham chiếu trong điều 8.3.3 được gọi, trong đó các ảnh tham chiếu có thể được đánh dấu là “không được sử dụng cho việc tham chiếu” hoặc “được sử dụng cho việc tham chiếu dài hạn”. Điều này được gọi chỉ cho nhóm phiên thứ nhất của ảnh.
 - *PicOutputFlag* được thiết đặt như sau:
 - Khi một trong các điều kiện sau là đúng, *PictureOutputFlag* được thiết đặt bằng 0:
 - ảnh hiện tại là ảnh RASL và *NoIncorrectPicOutputFlag* của ảnh IRAP được kết hợp bằng 1.
 - *gdr_enabled_flag* bằng 1 và ảnh hiện tại là ảnh GDR với *NoIncorrectPicOutputFlag* bằng 1.
 - *gdr_enabled_flag* bằng 1 và ảnh hiện tại chứa một hoặc nhiều nhóm phiên với *refreshed_region_flag* bằng 0 và *NoBrokenPictureOutputFlag* của ảnh GDR được kết hợp bằng 1.
 - Ngược lại, *PicOutputFlag* được thiết đặt bằng 1.
3. Các quy trình giải mã được gọi cho các đơn vị cây tạo mã, tỷ lệ, biến đổi, lọc trong vòng lặp, v.v.
4. Sau khi tất cả các nhóm phiên của ảnh hiện tại đã được giải mã, ảnh được giải mã hiện tại được đánh dấu là “được sử dụng cho việc tham chiếu ngắn hạn”.
 Quy trình giải mã đối với số đếm thứ tự ảnh được bàn luận.
 Đầu ra của quy trình này là *PicOrderCntVal*, số đếm thứ tự ảnh của ảnh hiện tại.

Mỗi ảnh được tạo mã được liên kết với biến số đếm thứ tự ảnh, được biểu thị là `PicOrderCntVal`.

Khi ảnh hiện tại không phải là ảnh IRAP với `NoIncorrectPicOutputFlag` bằng 1 hoặc ảnh GDR với `NoIncorrectPicOutputFlag` bằng 1, các biến `prevPicOrderCntLsb` và `prevPicOrderCntMsb` được dẫn ra như sau:

–Để `prevTid0Pic` là ảnh trước đó theo thứ tự giải mã mà có `TemporalId` bằng 0 và không phải là ảnh RASL hoặc RADL.

–Biến `prevPicOrderCntLsb` được thiết đặt bằng `tile_group_pic_order_cnt_lsb` của `prevTid0Pic`.

–Biến `prevPicOrderCntMsb` được thiết đặt bằng `PicOrderCntMsb` của `prevTid0Pic`.

Biến `PicOrderCntMsb` của ảnh hiện tại được dẫn ra như sau:

–Khi ảnh hiện tại là ảnh IRAP với `NoIncorrectPicOutputFlag` bằng 1 hoặc ảnh GDR với `NoIncorrectPicOutputFlag` bằng 1, thì `PicOrderCntMsb` được thiết đặt bằng 0.

–Ngược lại, `PicOrderCntMsb` được dẫn ra như sau:

```
if( ( tile_group_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - tile_group_pic_order_cnt_lsb ) >= (
        MaxPicOrderCntLsb / 2 ) ) )
```

`PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb` (8-

1)

```
else if( (tile_group_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
```

```
    ( ( tile_group_pic_order_cnt_lsb - prevPicOrderCntLsb ) > (
        MaxPicOrderCntLsb / 2 ) ) )
```

`PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb`

else

`PicOrderCntMsb = prevPicOrderCntMsb`

`PicOrderCntVal` được dẫn ra như sau:

`PicOrderCntVal = PicOrderCntMsb + tile_group_pic_order_cnt_lsb` (8-

2)

Lưu ý 1 – Tất cả các ảnh IRAP với `NoIncorrectPicOutputFlag` bằng 1 sẽ có `PicOrderCntVal` bằng `tile_group_pic_order_cnt_lsb` vì đối với các ảnh IRAP với `NoIncorrectPicOutputFlag` bằng 1, thì `PicOrderCntMsb` được thiết đặt bằng 0.

Lưu ý 1 – Tất cả các ảnh GDR với NoIncorrectPicOutputFlag bằng 1 sẽ có PicOrderCntVal bằng tile_group_pic_order_cnt_lsb vì đối các ảnh GDR với NoIncorrectPicOutputFlag bằng 1, thì PicOrderCntMsb được thiết đặt bằng 0.

Giá trị của PicOrderCntVal sẽ nằm trong khoảng từ -2^{31} đến $2^{31} - 1$, tuyến không loại.

Khi ảnh hiện tại là ảnh GDR, giá trị của LastGDRPocVal được thiết đặt là bằng PicOrderCntVal.

Quy trình giải mã cho vị trí biên được làm mới ảnh được bàn luận.

Quy trình này được gọi chỉ khi gdr_enabled_flag bằng 1.

Quy trình này được gọi sau khi việc phân tích phân đầu nhóm phiên được hoàn tất.

Đầu ra của quy trình này là PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, PicRefreshedTopBoundaryPos, và PicRefreshedBotBoundaryPos, các vị trí biên của vùng được làm mới của ảnh hiện tại.

Mỗi ảnh được tạo mã được kết hợp với tập hợp các biến vị trí biên vùng được làm mới, được biểu thị là PicOrderCntVal.

PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, PicRefreshedTopBoundaryPos, và PicRefreshedBotBoundaryPos được dẫn ra như sau:

Nếu nhóm phiên là nhóm phiên được nhận thứ nhất của ảnh hiện tại với refreshed_region_flag bằng 1, thì áp dụng dưới đây:

PicRefreshedLeftBoundaryPos = TGRefreshedLeftBoundary

PicRefreshedRightBoundaryPos = TGRefreshedRightBoundary

PicRefreshedTopBoundaryPos = TGRefreshedTopBoundary

PicRefreshedBotBoundaryPos = TileGroupBotBoundary

Nếu không, nếu refreshed_region_flag bằng 1, thì áp dụng như sau:

PicRefreshedLeftBoundaryPos = TGRefreshedLeftBoundary <

PicRefreshedLeftBoundaryPos ?

TGRefreshedLeftBoundary : PicRefreshedLeftBoundaryPos

PicRefreshedRightBoundaryPos = TGRefreshedRightBoundary >

PicRefreshedRightBoundaryPos ?

TGRefreshedRightBoundary : PicRefreshedRightBoundaryPos

$PicRefreshedTopBoundaryPos = TGR refreshedTopBoundary < PicRefreshedTopBoundaryPos ?$

$TGR refreshedTopBoundary : RefreshedRegionTopBoundaryPos$

$PicRefreshedBotBoundaryPos = TileGroupBotBoundary > PicRefreshedBotBoundaryPos ?$

$TileGroupBotBoundary : PicRefreshedBotBoundaryPos$

Quy trình giải mã cho việc xây dựng các danh sách ảnh tham chiếu được bàn luận.

...

Nó là yêu cầu về sự hợp cách luồng bit mà cho mỗi ảnh hiện tại mà không phải là ảnh IRAP với $NoIncorrectPicOutputFlag$ bằng 1 hoặc ảnh GDR với $NoIncorrectPicOutputFlag$ bằng 1, giá trị của $maxPicOrderCnt - minPicOrderCnt$ sẽ nhỏ hơn $MaxPicOrderCntLsb / 2$.

...

Quy trình giải mã cho việc đánh dấu ảnh tham chiếu

...

Nếu ảnh hiện tại là ảnh IRAP với $NoIncorrectPicOutputFlag$ bằng 1 hoặc ảnh GDR với $NoIncorrectPicOutputFlag$ bằng 1, tất cả các ảnh tham chiếu hiện trong DPB (nếu có) được đánh dấu là “không được sử dụng cho việc tham chiếu”.

...

Quy trình dẫn ra cho việc dự đoán vector chuyển động độ chói theo thời gian được bàn luận.

...

Biến $currCb$ chỉ rõ khối tạo mã độ chói hiện tại ở vị trí độ chói (xCb, yCb).

Các biến $mvLXCol$ và $availableFlagLXCol$ được dẫn ra như sau:

–Nếu $tile_group_temporal_mvp_enabled_flag$ bằng 0, thì cả hai thành phần của $mvLXCol$ được thiết đặt bằng 0 và $availableFlagLXCol$ được thiết đặt bằng 0.

–Ngược lại ($tile_group_temporal_mvp_enabled_flag$ bằng 1), thì áp dụng các bước được xếp thứ tự sau đây:

1. Vector chuyển động cùng vị trí phía dưới bên phải được suy ra như dưới đây:

$$xColBr = xCb + cbWidth \quad (8-414)$$

$$yColBr = yCb + cbHeight \quad (8-415)$$

$$leftBoundaryPos = gdr_enabled_flag ?$$

PicRefreshedLeftBoundaryPos của ảnh được tham chiếu đến bởi RefPicList[X][refIdxLX] :

$$0 \quad (8-415)$$

$$topBoundaryPos = gdr_enabled_flag ?$$

PicRefreshedTopBoundaryPos của ảnh được tham chiếu đến bởi RefPicList[X][refIdxLX] :

$$0 \quad (8-415)$$

$$rightBoundaryPos = gdr_enabled_flag ?$$

PicRefreshedRightBoundaryPos của ảnh được tham chiếu đến bởi RefPicList[X][refIdxLX] :

$$pic_width_in_luma_samples \quad (8-415)$$

$$botBoundaryPos = gdr_enabled_flag ?$$

PicRefreshedBotBoundaryPos của ảnh được tham chiếu đến bởi RefPicList[X][refIdxLX] :

$$pic_height_in_luma_samples \quad (8-415)$$

– Nếu $yCb \gg CtbLog2SizeY$ bằng $yColBr \gg CtbLog2SizeY$, $yColBr$ nằm trong khoảng từ $topBoundaryPos$ đến $botBoundaryPos$, không loại trừ, và $xColBr$ nằm trong khoảng từ $leftBoundaryPos$ đến $rightBoundaryPos$, không loại trừ, thì áp dụng như sau:

– Biến $colCb$ chỉ rõ khối tạo mã độ chói bao phủ vị trí được sửa đổi được cho bởi ($(xColBr \gg 3) \ll 3, (yColBr \gg 3) \ll 3$) bên nội nằm cùng vị trí được chỉ rõ bởi $ColPic$.

– Vị trí độ chói ($xColCb, yColCb$) được thiết đặt bằng mẫu trên cùng bên trái của khối tạo mã độ chói nằm cùng vị trí được chỉ rõ bởi $colCb$ so với mẫu độ chói trên cùng bên trái của ảnh nằm cùng vị trí được chỉ rõ bởi $ColPic$.

– Quy trình dẫn ra cho các vector chuyển động nằm cùng vị trí như được chỉ rõ trong điều 8.5.2.12 được gọi ra với $currCb, colCb, (xColCb, yColCb)$, $refIdxLX$ và $sbFlag$ được thiết đặt bằng 0 làm các đầu vào, và đầu ra được gán cho $mvLXCol$ và $availableFlagLXCol$.

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

2. ...

Quy trình nội suy song tuyến tính mẫu độ chói được bàn luận.

Các đầu vào cho quy trình này là:

–vị trí độ chói trong các đơn vị mẫu đầy đủ ($xInt_L$, $yInt_L$),

–vị trí độ chói trong các đơn vị mẫu phân số ($xFrac_L$, $yFrac_L$),

–mảng mẫu tham chiếu độ chói refPicLXL.

–các biên vùng được làm mới của ảnh tham chiếu *PicRefreshedLeftBoundaryPos*, *PicRefreshedTopBoundaryPos*, *PicRefreshedRightBoundaryPos*, và *PicRefreshedBotBoundaryPos*.

...

Các vị trí độ chói trong các đơn vị mẫu đầy đủ ($xInt_i$, $yInt_i$) được dẫn ra như sau đối với $i = 0..1$:

–Nếu *gdr_enabled_flag* bằng 1, thì áp dụng như sau:

$xInt_i = Clip3(PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, xInt_L + i)$ (8-458)

$yInt_i = Clip3(PicRefreshedTopBoundaryPos, PicRefreshedBotBoundaryPos, yInt_L + i)$ (8-458)

–Ngược lại (*gdr_enabled_flag* bằng 0), thì áp dụng như sau:

$xInt_i = sps_ref_wraparound_enabled_flag ?$

$ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, (xInt_L + i))$: (8-459)

$Clip3(0, picW - 1, xInt_L + i)$

$yInt_i = Clip3(0, picH - 1, yInt_L + i)$ (8-460)

...

Quy trình lọc nội suy 8 điểm nội mẫu độ chói được bàn luận.

Các đầu vào cho quy trình này là:

–vị trí độ chói trong các đơn vị mẫu đầy đủ ($xInt_L$, $yInt_L$),

–vị trí độ chói trong các đơn vị mẫu phân số ($xFrac_L$, $yFrac_L$),

–mảng mẫu tham chiếu độ chói refPicLXL,

– danh sách $\text{padVal}[\text{dir}]$ với $\text{dir} = 0, 1$ chỉ rõ lượng và hướng đệm mẫu tham chiếu.
 – các biên vùng được làm mới của ảnh tham chiếu $\text{PicRefreshedLeftBoundaryPos}$,
 $\text{PicRefreshedTopBoundaryPos}$, $\text{PicRefreshedRightBoundaryPos}$, và
 $\text{PicRefreshedBotBoundaryPos}$.

...

Các vị trí độ chói trong các đơn vị mẫu đầy đủ (x_{Int_i} , y_{Int_i}) được dẫn ra như sau đối với $i = 0..7$:

– Nếu gdr_enabled_flag bằng 1, thì áp dụng như sau:

$x_{\text{Int}_i} = \text{Clip3}(\text{PicRefreshedLeftBoundaryPos}, \text{PicRefreshedRightBoundaryPos},$
 $x_{\text{Int}_L} + i - 3) \quad (8-830)$

$y_{\text{Int}_i} = \text{Clip3}(\text{PicRefreshedTopBoundaryPos}, \text{PicRefreshedBotBoundaryPos}, y_{\text{Int}_L}$
 $+ i - 3) \quad (8-830)$

– Ngược lại (gdr_enabled_flag bằng 0), thì áp dụng như sau:

$x_{\text{Int}_i} = \text{sps_ref_wraparound_enabled_flag} ?$

$\text{ClipH}((\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY}, \text{picW}, x_{\text{Int}_L} +$
 $i - 3) : \quad (8-831)$

$\text{Clip3}(0, \text{picW} - 1, x_{\text{Int}_L} + i - 3)$

$y_{\text{Int}_i} = \text{Clip3}(0, \text{picH} - 1, y_{\text{Int}_L} + i - 3) \quad (8-832)$

Quy trình nội suy mẫu sắc độ được bàn luận.

Các đầu vào cho quy trình này là:

– vị trí sắc độ trong các đơn vị mẫu đầy đủ (x_{Int_c} , y_{Int_c}),

– vị trí sắc độ trong các đơn vị mẫu phân số $1/32$ (x_{Frac_c} , y_{Frac_c}),

– mảng mẫu tham chiếu sắc độ refPicLXC .

– các biên vùng được làm mới của ảnh tham chiếu $\text{PicRefreshedLeftBoundaryPos}$,
 $\text{PicRefreshedTopBoundaryPos}$, $\text{PicRefreshedRightBoundaryPos}$, và
 $\text{PicRefreshedBotBoundaryPos}$.

...

Biến x_{Offset} được thiết đặt bằng $(\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY} / \text{SubWidthC}$.

Các vị trí sắc độ trong các đơn vị mẫu đầy đủ (x_{Int_i} , y_{Int_i}) được dẫn ra như sau đối với $i = 0..3$:

–Nếu *gdr_enabled_flag* bằng 1, thì áp dụng như sau:

$$xInt_i = Clip3(PicRefreshedLeftBoundaryPos / SubWidthC, \\ PicRefreshedRightBoundaryPos / SubWidthC, xInt_L + i) \quad (8-844)$$

$$yInt_i = Clip3(PicRefreshedTopBoundaryPos / SubHeightC, \\ PicRefreshedBotBoundaryPos / SubHeightC, yInt_L + i) \quad (8-844)$$

–Ngược lại (*gdr_enabled_flag* bằng 0), thì áp dụng như sau:

$$xInt_i = sps_ref_wraparound_enabled_flag ? ClipH(xOffset, picW_C, xInt_C + i - 1) : \\ (8-845)$$

$$Clip3(0, picW_C - 1, xInt_C + i - 1)$$

$$yInt_i = Clip3(0, picH_C - 1, yInt_C + i - 1) \quad (8-846)$$

Quy trình lọc khử khối được bàn luận.

Quy trình chung.

...

Quy trình lọc khử khối được áp dụng cho tất cả các gờ khối con tạo mã và các gờ khối biến đổi của ảnh, ngoại trừ các loại gờ sau:

–Các gờ ở biên của ảnh,

–Các gờ trùng với biên phía trên của nhóm phiên *tgA* khi tất cả các điều kiện sau được thỏa mãn:

–*gdr_enabled_flag* bằng 1

–*loop_filter_across_refreshed_region_enabled_flag* bằng 0

–các gờ trùng với biên phía dưới của nhóm phiên *tgB* và giá trị của *refreshed_region_flag* của *tgB* khác với giá trị của *refreshed_region_flag* của *tgA*

–Các gờ trùng với biên bên trái của nhóm phiên *tgA* khi tất cả các điều kiện sau được thỏa mãn:

–*gdr_enabled_flag* bằng 1

–*loop_filter_across_refreshed_region_enabled_flag* bằng 0

–các gờ trùng với biên bên phải củanhóm phiên *tgB* và giá trị của *refreshed_region_flag* của *tgB* khác với giá trị của *refreshed_region_flag* của *tgA*

–Các gờ trùng với các biên phiên khi *loop_filter_across_tiles_enabled_flag* bằng 0,

–Các gờ trùng với các biên phía trên hoặc bên trái của các nhóm phiên với `tile_group_loop_filter_across_tile_groups_enabled_flag` bằng 0 hoặc `tile_group_deblocking_filter_disabled_flag` bằng 1,

–Các gờ nằm trong các nhóm phiên với `tile_group_deblocking_filter_disabled_flag` bằng 1,

–Các gờ không tương ứng với các biên lưới mẫu 8x8 của thành phần được xem xét,

–Các gờ nằm trong các thành phần sắc độ mà cho nó cả hai bên của gờ sử dụng việc dự đoán liên ảnh,

–Các gờ của các khối biến đổi sắc độ mà không phải là các gờ của đơn vị biến đổi được kết hợp.

–Các gờ qua các khối biến đổi độ chói của đơn vị tạo mã mà có giá trị `IntraSubPartitionsSplit` không bằng `ISP_NO_SPLIT`.

Quy trình lọc khử khối cho một hướng được bàn luận.

...

Đối với mỗi đơn vị tạo mã với chiều rộng khối tạo mã $\log_2 CbW$, chiều cao khối tạo mã $\log_2 CbH$ và vị trí của mẫu trên cùng bên trái của khối tạo mã (x_{Cb} , y_{Cb}), khi `edgeType` bằng `EDGE_VER` và $x_{Cb} \% 8$ bằng 0 hoặc khi `edgeType` bằng `EDGE_HOR` và $y_{Cb} \% 8$ bằng 0, các gờ được lọc bởi các bước được sắp xếp thứ tự như sau:

1. Chiều rộng khối tạo mã n_{CbW} được thiết đặt bằng $1 \ll \log_2 CbW$ và chiều cao khối tạo mã n_{CbH} được thiết đặt bằng $1 \ll \log_2 CbH$

2. Biến `filterEdgeFlag` được dẫn ra như sau:

–Nếu `edgeType` bằng `EDGE_VER` và một hoặc nhiều điều kiện sau là đúng, `filterEdgeFlag` được thiết đặt bằng 0:

–Biên bên trái của khối tạo mã hiện tại là biên bên trái của ảnh.

–Biên bên trái của khối tạo mã hiện tại là biên bên trái của phiên và `loop_filter_across_tiles_enabled_flag` bằng 0.

–Biên bên trái của khối tạo mã hiện tại là biên bên trái của nhóm phiên và `tile_group_loop_filter_across_tile_groups_enabled_flag` bằng 0.

–*biên bên trái của khối tạo mã hiện tại là biên bên trái của nhóm phiên hiện tại và tất cả các điều kiện sau được thỏa mãn:*

–*`gdr_enabled_flag` bằng 1*

–*loop_filter_across_refreshed_region_enabled_flag* bằng 0.

–*nhóm phiến mà có chung biên với biên bên trái của nhóm phiến hiện tại có mặt và giá trị của refreshed_region_flag của nó khác với giá trị của refreshed_region_flag của nhóm phiến hiện tại.*

–Ngược lại, nếu *edgeType* bằng *EDGE_HOR* và một hoặc nhiều điều kiện sau là đúng, thì biến *filterEdgeFlag* được thiết đặt bằng 0:

–Biên trên cùng của khối tạo mã độ chói hiện tại là biên trên cùng của ảnh.

–Biên trên cùng của khối tạo mã hiện tại là biên trên cùng của phiến và *loop_filter_across_tiles_enabled_flag* bằng 0.

–Biên trên cùng của khối tạo mã hiện tại là biên trên cùng của nhóm phiến và *tile_group_loop_filter_across_tile_groups_enabled_flag* bằng 0.

–*Biên trên cùng của khối tạo mã hiện tại là biên trên cùng của nhóm phiến hiện tại và tất cả các điều kiện sau được thỏa mãn:*

–*gdr_enabled_flag* bằng 1

–*loop_filter_across_refreshed_region_enabled_flag* bằng 0.

–*nhóm phiến that shared boundary với Biên trên cùng của nhóm phiến hiện tại có mặt và Giá trị của its refreshed_region_flag khác với giá trị của refreshed_region_flag của nhóm phiến hiện tại.*

–Ngược lại, *filterEdgeFlag* được thiết đặt bằng 1.

Điều chỉnh cú pháp khi các phiến được tích hợp.

3. Tất cả các phần tử của mảng $(nCbW) \times (nCbH)$ hai chiều *edgeFlags* được khởi tạo để bằng không.

Quy trình sửa đổi CTB cho SAO được bàn luận.

...

Đối với tất cả các vị trí mẫu (xSi, ySj) và (xYi, yYj) với $i = 0..nCtbSw - 1$ và $j = 0..nCtbSh - 1$, tùy thuộc vào các giá trị của *pcm_loop_filter_disabled_flag*, *pcm_flag[xYi][yYj]* và *cu_transquant_bypass_flag* của đơn vị tạo mã mà chứa khối tạo mã bao phủ *recPicture[xSi][ySj]*, thì áp dụng như sau:

–....

Sửa đổi các phần được làm nổi bật chưa được giải quyết về đường vòng biến đổi/lượng tử hóa tương lai.

–Ngược lại, nếu SaoTypeIdx[cIdx][rx][ry] bằng 2, thì áp dụng các bước được xếp thứ tự sau đây:

1. Các giá trị của hPos[k] và vPos[k] cho $k = 0..1$ được chỉ rõ trong bảng 8-18 dựa trên SaoEoClass[cIdx][rx][ry].

2. Biến edgeIdx được dẫn ra như sau:

–Các vị trí mẫu được sửa đổi ($xS_{ik'}$, $yS_{jk'}$) và ($xY_{ik'}$, $yY_{jk'}$) được dẫn ra như sau:

$$(xS_{ik'}, yS_{jk'}) = (xS_i + hPos[k], yS_j + vPos[k]) \quad (8-1128)$$

$$(xY_{ik'}, yY_{jk'}) = (cIdx == 0) ? (xS_{ik'}, yS_{jk'}) : (xS_{ik'} * SubWidthC, yS_{jk'} * SubHeightC) \quad (8-1129)$$

–Nếu một hoặc nhiều điều kiện sau cho tất cả các vị trí mẫu ($xS_{ik'}$, $yS_{jk'}$) và ($xY_{ik'}$, $yY_{jk'}$) với $k = 0..1$ là đúng, edgeIdx được thiết đặt bằng 0:

–Mẫu ở vị trí ($xS_{ik'}$, $yS_{jk'}$) là bên ngoài ảnh các biên.

–*gdr_enabled_flag* bằng 1, *loop_filter_across_refreshed_region_enabled_flag* bằng 0, *refreshed_region_flag* của nhóm phiên hiện tại bằng 1, và *refreshed_region_flag* của nhóm phiên chứa mẫu ở vị trí ($xS_{ik'}$, $yS_{jk'}$) bằng 0.

–Mẫu ở vị trí ($xS_{ik'}$, $yS_{jk'}$) thuộc về nhóm phiên khác và một điều kiện trong số hai điều kiện là đúng:

–MinTbAddrZs[$xY_{ik'} \gg \text{MinTbLog2SizeY}$][$yY_{jk'} \gg \text{MinTbLog2SizeY}$] nhỏ hơn MinTbAddrZs[$xY_i \gg \text{MinTbLog2SizeY}$][$yY_j \gg \text{MinTbLog2SizeY}$] và *tile_group_loop_filter_across_tile_groups_enabled_flag* trong nhóm phiên, mà mẫu *recPicture*[xS_i][yS_j] thuộc về, bằng 0.

–MinTbAddrZs[$xY_i \gg \text{MinTbLog2SizeY}$][$yY_j \gg \text{MinTbLog2SizeY}$] nhỏ hơn MinTbAddrZs[$xY_{ik'} \gg \text{MinTbLog2SizeY}$][$yY_{jk'} \gg \text{MinTbLog2SizeY}$] và *tile_group_loop_filter_across_tile_groups_enabled_flag* trong nhóm phiên, mà mẫu *recPicture*[$xS_{ik'}$][$yS_{jk'}$] thuộc về, bằng 0.

–*loop_filter_across_tiles_enabled_flag* bằng 0 và mẫu ở vị trí ($xS_{ik'}$, $yS_{jk'}$) thuộc về phiên khác.

Sửa đổi các phần được làm nổi bật khi các phiên mà không có các nhóm phiên được kết hợp

–Ngược lại, edgeIdx được dẫn ra như sau:

–Áp dụng như sau:

$$\begin{aligned} \text{edgeIdx} = & 2 + \text{Sign}(\text{recPicture}[\text{xSi}][\text{ySj}] - \text{recPicture}[\text{xSi} + \text{hPos}[0]][\text{ySj} + \\ & \text{vPos}[0]]) + \\ & \text{Sign}(\text{recPicture}[\text{xSi}][\text{ySj}] - \text{recPicture}[\text{xSi} + \text{hPos}[1]][\text{ySj} + \text{vPos}[1]]) \\ & (8-1130) \end{aligned}$$

– Khi edgeIdx bằng 0, 1, hoặc 2, edgeIdx được sửa đổi như sau:

$$\text{edgeIdx} = (\text{edgeIdx} == 2) ? 0 : (\text{edgeIdx} + 1) \quad (8-1131)$$

3. Mảng mẫu ảnh được sửa đổi saoPicture[xSi][ySj] được dẫn ra như sau:

$$\begin{aligned} \text{saoPicture}[\text{xSi}][\text{ySj}] = & \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \text{recPicture}[\text{xSi}][\text{ySj}] + \\ & \text{SaoOffsetVal}[\text{cIdx}][\text{rx}][\text{ry}][\text{edgeIdx}]) \quad (8-1132) \end{aligned}$$

Quy trình lọc khối cây tạo mã cho các mẫu độ chói cho ALF được bàn luận.

...

Đối với việc dẫn ra của các mẫu độ chói được tái tạo mà được lọc alfPictureL[x][y], thì mỗi mẫu độ chói được tái tạo bên trong khối cây tạo mã độ chói hiện tại recPictureL[x][y] được lọc như sau với x, y = 0..CtbSizeY – 1:

– ...

– Các vị trí (h_x, v_y) cho mỗi mẫu trong số các mẫu độ chói (x, y) tương ứng bên trong mảng recPicture đã cho của các mẫu độ chói được dẫn ra như sau:

– Nếu gdr_enabled_flag bằng 1, $\text{loop_filter_across_refreshed_region_enabled_flag}$ bằng 0, $\text{refreshed_region_flag}$ của nhóm phiên tgA chứa mẫu độ chói ở vị trí (x, y) bằng 1, thì áp dụng như sau:

– Nếu vị trí (h_x, v_y) nằm trong nhóm phiên tgB khác và $\text{refreshed_region_flag}$ của tgB bằng 0, thì các biến leftBoundary, rightBoundary, topBoundary, và botBoundary lần lượt được thiết đặt bằng TGRrefreshedLeftBoundary, TGRrefreshedRightBoundary, TGRrefreshedTopBoundary, và TGRrefreshedBotBoundary.

– Ngược lại, các biến leftBoundary, rightBoundary, topBoundary, và botBoundary lần lượt được thiết đặt bằng PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, PicRefreshedTopBoundaryPos, và PicRefreshedBotBoundaryPos.

$$h_x = \text{Clip3}(\text{leftBoundary}, \text{rightBoundary}, \text{xCtb} + x) \quad (8-1140)$$

$$v_y = \text{Clip3}(\text{topBoundary}, \text{botBoundary}, \text{yCtb} + y) \quad (8-1141)$$

– Ngược lại, thì áp dụng như sau:

$$h_x = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x_{\text{Ctb}} + x) \quad (8-1140)$$

$$v_y = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y_{\text{Ctb}} + y) \quad (8-1141)$$

–...

Quy trình dẫn ra cho chỉ số chuyển vị và chỉ số lọc ALF cho các mẫu độ chói được bàn luận.

...

Các vị trí (h_x , v_y) cho mỗi mẫu trong số các mẫu độ chói (x , y) tương ứng bên trong mảng `recPicture` đã cho của các mẫu độ chói được dẫn ra như sau:

–Nếu gdr_enabled_flag bằng 1, $\text{loop_filter_across_refreshed_region_enabled_flag}$ bằng 0, $\text{refreshed_region_flag}$ của nhóm phiên `tgA` chứa mẫu độ chói ở vị trí (x , y) bằng 1, thì áp dụng như sau:

–Nếu vị trí (h_x , v_y) nằm trong nhóm phiên `tgB` khác và $\text{refreshed_region_flag}$ của `tgB` bằng 0, thì các biến `leftBoundary`, `rightBoundary`, `topBoundary`, và `botBoundary` lần lượt được thiết đặt bằng `TGRefreshedLeftBoundary`, `TGRefreshedRightBoundary`, `TGRefreshedTopBoundary`, và `TGRefreshedBotBoundary`.

–Ngược lại, các biến `leftBoundary`, `rightBoundary`, `topBoundary`, và `botBoundary` lần lượt được thiết đặt bằng `PicRefreshedLeftBoundaryPos`, `PicRefreshedRightBoundaryPos`, `PicRefreshedTopBoundaryPos`, và `PicRefreshedBotBoundaryPos`.

$$h_x = \text{Clip3}(\text{leftBoundary}, \text{rightBoundary}, x) \quad (8-1140)$$

$$v_y = \text{Clip3}(\text{topBoundary}, \text{botBoundary}, y) \quad (8-1141)$$

–Ngược lại, thì áp dụng như sau:

$$h_x = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x) \quad (8-1145)$$

$$v_y = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y) \quad (8-1146)$$

Quy trình lọc khối cây tạo mã cho các mẫu sắc độ được bàn luận.

...

Đối với việc dẫn ra của các mẫu sắc độ được tái tạo mà được lọc `alfPicture[x][y]`, mỗi mẫu sắc độ được tái tạo bên trong khối cây tạo mã sắc độ hiện tại `recPicture[x][y]` được lọc như sau với $x, y = 0..ctbSizeC - 1$:

–Các vị trí (h_x , v_y) cho mỗi mẫu trong số các mẫu sắc độ (x , y) tương ứng bên trong mảng `recPicture` đã cho của các mẫu sắc độ được dẫn ra như sau:

–Nếu $gdr_enabled_flag$ bằng 1, $loop_filter_across_refreshed_region_enabled_flag$ bằng 0, $refreshed_region_flag$ của nhóm phiên tgA chứa mẫu độ chói ở vị trí (x, y) bằng 1, thì áp dụng như sau:

–Nếu vị trí (h_x, v_y) nằm trong nhóm phiên tgB khác và $refreshed_region_flag$ của tgB bằng 0, thì các biến $leftBoundary$, $rightBoundary$, $topBoundary$, và $botBoundary$ lần lượt được thiết đặt bằng $TGRefreshedLeftBoundary$, $TGRefreshedRightBoundary$, $TGRefreshedTopBoundary$, và $TGRefreshedBotBoundary$.

–Ngược lại, các biến $leftBoundary$, $rightBoundary$, $topBoundary$, và $botBoundary$ lần lượt được thiết đặt bằng $PicRefreshedLeftBoundaryPos$, $PicRefreshedRightBoundaryPos$, $PicRefreshedTopBoundaryPos$, và $PicRefreshedBotBoundaryPos$.

$$h_x = Clip3(leftBoundary / SubWidthC , rightBoundary / SubWidthC , xCtbC + x) \quad (8-1140)$$

$$v_y = Clip3(topBoundary / SubWidthC , botBoundary / SubWidthC , yCtbC + y) \quad (8-1141)$$

–Ngược lại, thì áp dụng như sau:

$$h_x = Clip3(0, pic_width_in_luma_samples / SubWidthC - 1, xCtbC + x) \quad (8-1177)$$

$$v_y = Clip3(0, pic_height_in_luma_samples / SubHeightC - 1, yCtbC + y) \quad (8-1178)$$

Fig.7 minh họa luồng bit video 750 được tạo cấu hình để thực hiện kỹ thuật làm mới giải mã từng bước (GDR) 700 theo phương án của sáng chế. Kỹ thuật GDR 700 có thể là tương tự với kỹ thuật GDR 500 trên Fig.5. Như được sử dụng ở đây luồng bit video 750 có thể còn được gọi là luồng bit video đã được tạo mã, luồng bit, hoặc các biến thể của chúng. Như được minh họa trên Fig.7, luồng bit 750 bao gồm tập hợp thông số chuỗi (SPS) 752, tập hợp thông số ảnh (PPS) 754, phần đầu lát 756, và dữ liệu hình ảnh 758.

SPS 752 chứa dữ liệu mà là chung cho tất cả các ảnh trong chuỗi các hình ảnh (sequence of pictures, SOP). Ngược lại, PPS 754 chứa dữ liệu mà chung cho toàn bộ ảnh. Phần đầu lát 756 chứa thông tin về lát hiện tại như, ví dụ, loại lát, mà các ảnh tham chiếu của chúng sẽ được sử dụng, và v.v. SPS 752 và PPS 754 có thể được gọi chung là tập hợp thông số. SPS 752, PPS 754, và phần đầu lát 756 là các loại của các đơn vị lớp trừu tượng

mạng (network abstraction layer, NAL). Đơn vị NAL là cấu trúc cú pháp chứa sự chỉ báo của loại dữ liệu để theo sau (ví dụ, dữ liệu video được tạo mã). Các đơn vị NAL được phân loại thành các đơn vị lớp tạo mã video (VCL) và NAL không VCL. Các đơn vị NAL VCL chứa dữ liệu mà thể hiện các giá trị của các mẫu trong video các ảnh, và các đơn vị NAL không VCL chứa thông tin bổ sung được kết hợp bất kỳ như các tập hợp thông số (dữ liệu phần đầu quan trọng mà có thể áp dụng cho số lượng lớn của các đơn vị NAL VCL) và thông tin tăng cường bổ sung (thông tin định thời và dữ liệu bổ sung khác mà có thể tăng cường khả năng sử dụng của tín hiệu video được giải mã nhưng không cần thiết để giải mã các giá trị của các mẫu trong video các ảnh). Những người có trình độ trung bình trong lĩnh vực sẽ hiểu rằng luồng bit 750 có thể chứa các thông số và thông tin khác trong các ứng dụng thực tế.

Dữ liệu hình ảnh 758 trên Fig.7 bao gồm dữ liệu được kết hợp với các ảnh hoặc video được mã hóa hoặc được giải mã. Dữ liệu hình ảnh 758 có thể được gọi đơn giản là phần tải hoặc dữ liệu được mang trong luồng bit 750. Theo một phương án, dữ liệu hình ảnh 758 bao gồm CVS 708 (hoặc CLVS) chứa ảnh GDR 702, một hoặc nhiều ảnh theo sau 704, và ảnh điểm khôi phục 706. Theo một phương án, các ảnh theo sau 704 có thể được xem là một dạng của ảnh GDR do chúng ở trước ảnh điểm khôi phục 706 trong thời gian GDR.

Theo một phương án, ảnh GDR 702, các ảnh theo sau 704, và ảnh điểm khôi phục 706 có thể xác định thời gian GDR (GDR period) trong CVS 708. Theo một phương án, thứ tự giải mã bắt đầu với ảnh GDR 702, tiếp tục với các ảnh theo sau 704, và sau đó tiếp đến ảnh khôi phục 706.

Khi giá trị (ví dụ, một (1)) được nhận bởi bộ giải mã video 30 qua giao diện người dùng 84, cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi giao diện người dùng (ví dụ, đầu vào bên ngoài) và cờ thứ hai được thiết đặt bằng cờ thứ nhất để ngăn ảnh GDR 702 và các ảnh theo sau 704 bất kỳ giữa ảnh GDR 702 và ảnh điểm khôi phục 706 theo thứ tự xuất ra (ví dụ, thứ tự trình bày 410) khỏi việc được xuất ra. Ngược lại, khi không có giá trị được nhận bởi bộ tạo mã video 30 qua giao diện người dùng 84, cờ thứ nhất và cờ thứ hai được thiết đặt bằng giá trị khác (ví dụ, không (0)). Theo một phương án, chỉ ảnh GDR 702 được ngăn khỏi việc được xuất ra khi cờ thứ nhất được thiết đặt bằng giá trị được cung cấp bởi giao diện người dùng.

CVS 708 là chuỗi video được tạo mã cho mọi chuỗi video lớp được tạo mã (coded layer video sequence, CLVS) trong luồng bit video 750. Đáng lưu ý là, CVS và CLVS là giống nhau khi luồng bit video 750 chứa lớp đơn. CVS và CLVS chỉ khác nhau khi luồng bit video 750 chứa nhiều lớp.

Như được minh họa trên Fig.7, kỹ thuật GDR 700 hoặc nguyên lý hoạt động qua một loạt các ảnh bắt đầu với ảnh GDR 702 và kết thúc với ảnh điểm khôi phục 706. Kỹ thuật GDR 700, ảnh GDR 702, các ảnh theo sau 704, và ảnh điểm khôi phục 706 là tương tự với kỹ thuật GDR 500, ảnh GDR 502, các ảnh theo sau 504, và ảnh điểm khôi phục 506 trên Fig.5. Do đó, để ngắn gọn, cách trong đó kỹ thuật GDR 700 được thực hiện không lặp lại đối với Fig.7.

Như được minh họa trên Fig.7, mỗi ảnh trong số ảnh GDR 702, các ảnh theo sau 704, và ảnh điểm khôi phục 706 trong CVS 708 được chứa nằm trong đơn vị VCL NAL 730 của riêng chúng Tập hợp của các đơn vị NAL VCL 730 trong CVS 708 có thể được gọi là đơn vị truy cập.

Đơn vị NAL 730 chứa ảnh GDR 702 trong CVS 708 có loại đơn vị GDR NAL (GDR_NUT). Nghĩa là, theo một phương án, đơn vị NAL 730 chứa ảnh GDR 702 trong CVS 708 có loại đơn vị NAL duy nhất của riêng nó so với các ảnh theo sau 704 và ảnh điểm khôi phục 706. Theo một phương án, GDR_NUT cho phép luồng bit 750 bắt đầu với ảnh GDR 702 thay vì luồng bit 750 phải bắt đầu với ảnh IRAP. Việc chỉ định đơn vị VCL NAL 730 của ảnh GDR 702 là GDR_NUT có thể chỉ báo cho, ví dụ, bộ giải mã mà đơn vị VCL NAL ban đầu 730 trong CVS 708 chứa ảnh GDR 702.

Theo một phương án, ảnh GDR 702 là ảnh ban đầu trong CVS 708. Theo một phương án, ảnh GDR 702 là ảnh ban đầu trong thời gian GDR. Theo một phương án, ảnh GDR 702 có bộ nhận dạng (identifier, ID) theo thời gian bằng không. ID theo thời gian là giá trị hoặc số mà xác định vị trí hoặc thứ tự của ảnh so với các ảnh khác. Theo một phương án, đơn vị truy cập chứa đơn vị VCL NAL 730 có GDR_NUT được chỉ định là đơn vị truy cập GDR. Theo một phương án, ảnh GDR 702 là lát được tạo mã của ảnh GDR khác (ví dụ, lớn hơn). Nghĩa là, ảnh GDR 702 có thể là một phần của ảnh GDR lớn hơn.

Fig.8 là phương án của phương pháp 800 giải mã luồng bit video đã được tạo mã được thực hiện bởi bộ giải mã video (ví dụ, bộ giải mã video 30). Phương pháp 800 có

thể được thực hiện sau khi luồng bit được giải mã được nhận trực tiếp hoặc gián tiếp từ bộ mã hóa video (ví dụ, bộ mã hóa video 20). Phương pháp 800 cải thiện quy trình giải mã vì phương pháp cho phép việc làm mới nội tạng dần để cho phép việc truy cập ngẫu nhiên mà không phải sử dụng ảnh IRAP. Giá trị của cờ được thiết đặt bởi người dùng của bộ giải mã video qua giao diện người dùng (ví dụ, giao diện người dùng 84 trên Fig.3 hoặc một số phương tiện bên ngoài khác) để ngăn ảnh GDR khỏi việc được xuất ra ra. Theo một phương án, giá trị của cờ được thiết đặt bởi người dùng của bộ giải mã video qua giao diện người dùng (ví dụ, giao diện người dùng 84 trên Fig.3 hoặc một số phương tiện bên ngoài khác) để ngăn ảnh GDR và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra. Việc thiết đặt cờ theo cách này ngăn dữ liệu bản tiềm tàng khỏi việc được xuất ra đến bộ phận hiển thị, và cho phép bộ giải mã video hoạt động theo sở thích của người dùng. Do đó, là vấn đề thực hành, hiệu suất của phương tiện mã hóa được cải thiện, mà dẫn đến trải nghiệm người dùng tốt hơn.

Trong khối 802, bộ giải mã video xác định xem liệu giá trị cho cờ thứ nhất có được cung cấp bởi đầu vào bên ngoài (ví dụ, giao diện người dùng 84 trên Fig.3 hoặc một số phương tiện bên ngoài khác) hay không. Theo một phương án, đầu vào bên ngoài là giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video. Theo một phương án, người dùng của bộ giải mã video cung cấp giá trị của cờ thứ nhất sử dụng đầu vào bên ngoài. Theo một phương án, cờ thứ nhất được chỉ định là `HandleGdrAsCvsStartFlag`.

Trong khối 804, bộ giải mã video thiết đặt cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) (ví dụ, ảnh GDR 702) và các ảnh theo sau 704 bất kỳ giữa ảnh GDR 702 và ảnh điểm khôi phục 706 theo thứ tự xuất ra khỏi việc được xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài. Theo một phương án, giá trị của cờ thứ nhất được thiết đặt là một (1) để ngăn ảnh GDR và các ảnh theo sau bất kỳ giữa ảnh GDR và ảnh điểm khôi phục theo thứ tự xuất ra khỏi việc được xuất ra. Theo một phương án, giá trị của cờ thứ nhất được thiết đặt là không (0) khi giá trị cho cờ thứ nhất không được cung cấp bởi đầu vào bên ngoài.

Theo một phương án ảnh GDR là ảnh ban đầu trong CVS của luồng bit video đã được tạo mã. Theo một phương án, ảnh GDR là ảnh ban đầu trong lớp của luồng bit video

đã được tạo mã. Theo một phương án, lớp là CLVS của CVS của luồng bit video đã được tạo mã.

Trong khối 806, bộ giải mã video giải mã ảnh GDR. Sau đó, các ảnh theo sau và các ảnh điểm khôi phục được giải mã lần lượt. Trong khối 808, bộ giải mã video lưu trữ ảnh GDR trong bộ đệm ảnh được giải mã (decoded picture buffer, DPB). Theo một phương án khi ảnh GDR đã không bị giới hạn khỏi việc được xuất ra bởi thiết đặt của cờ thứ nhất và cờ thứ hai, hình ảnh được sinh ra dựa trên ảnh GDR có thể được hiển thị cho người dùng của thiết bị điện tử (ví dụ, điện thoại thông minh, máy tính bảng, máy tính xách tay, máy tính cá nhân, v.v.).

Fig.9 là hình vẽ dạng giản lược của thiết bị tạo mã video 900 (ví dụ, bộ mã hóa video 20 hoặc bộ giải mã video 30) theo một phương án của sáng chế. Thiết bị tạo mã video 900 là thích hợp để thực hiện các phương án được bộc lộ như được mô tả ở đây. Thiết bị tạo mã video 900 bao gồm các cổng vào 910 và các đơn vị nhận (Rx) 920 để nhận dữ liệu; bộ xử lý, đơn vị logic, hoặc bộ xử lý trung tâm (central processing unit, CPU) 930 để xử lý dữ liệu; các đơn vị truyền (Tx) 940 và các cổng ra 950 để truyền dữ liệu; và bộ nhớ 960 để lưu trữ dữ liệu. Thiết bị tạo mã video 900 cũng có thể bao gồm các thành phần quang sang điện (optical-to-electrical, OE) và các thành phần điện sang quang (electrical-to-optical, EO) được ghép nối với các cổng vào 910, các đơn vị nhận 920, các đơn vị truyền 940, và các cổng ra 950 cho lấy ra hoặc lấy vào các tín hiệu quang hoặc điện.

Bộ xử lý 930 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 930 có thể được thực hiện như là một hoặc nhiều chip CPU, lõi (ví dụ, như là bộ xử lý đa lõi), mảng cổng lập trình được dạng trường (field-programmable gate array, FPGA), mạch tích hợp chuyên dụng (application specific integrated circuit, ASIC), và bộ xử lý tín hiệu số (digital signal processor, DSP). Bộ xử lý 930 giao tiếp với các cổng vào 910, các đơn vị bộ thu 920, các đơn vị bộ phát 940, các cổng ra 950, và bộ nhớ 960. Bộ xử lý 930 bao gồm môđun tạo mã 970. Môđun tạo mã 970 này thực hiện các phương án được bộc lộ đã mô tả trên đây. Ví dụ, môđun tạo mã 970 thực hiện, xử lý, chuẩn bị hoặc cung cấp các chức năng codec khác nhau. Do đó, sự có mặt của môđun tạo mã 970 mang lại sự cải thiện quan trọng cho chức năng của thiết bị tạo mã video 900 và tạo hiệu quả biến đổi thiết bị tạo mã video 900 sang trạng thái khác. Theo cách thay thế, môđun tạo mã 970 được thực

hiện dưới dạng các lệnh được lưu trữ trong bộ nhớ 960 và được thực hiện bởi bộ xử lý 930.

Thiết bị tạo mã video 900 có thể còn bao gồm các thiết bị đầu vào và/hoặc đầu ra (Input/Output, I/O) 980 để truyền thông dữ liệu đến và từ người dùng. Các thiết bị I/O 980 có thể bao gồm các thiết bị đầu ra như bộ hiển thị để hiển thị dữ liệu video, các loa để xuất dữ liệu audio, v.v. Các thiết bị I/O 980 cũng có thể bao gồm các thiết bị đầu vào, như bàn phím, chuột, bẻ xoay, v.v., và/hoặc các giao diện tương ứng để tương tác với các thiết bị đầu ra này. Theo một phương án, các thiết bị I/O 980 là phương tiện bên ngoài được sử dụng bởi người dùng của thiết bị tạo mã video 900 để nhập giá trị của cờ thứ nhất.

Bộ nhớ 960 bao gồm một hoặc nhiều đĩa, ổ băng và ổ trạng thái rắn và có thể được sử dụng thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình này được chọn để thực hiện, và để lưu trữ các lệnh và dữ liệu được đọc trong suốt quá trình thực hiện chương trình. Bộ nhớ 960 có thể là khả biến và/hoặc bất khả biến và có thể là bộ nhớ chỉ đọc (read-only memory, ROM), bộ nhớ truy cập ngẫu nhiên (random access memory, RAM), bộ nhớ khả lập địa chỉ nội dung bậc ba (ternary content-addressable memory, TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (static random-access memory, SRAM).

Fig.10 là hình vẽ dạng giản lược của một phương án của phương tiện để tạo mã 1000. Theo một phương án, phương tiện để tạo mã 1000 được thực hiện trong thiết bị tạo mã video 1002 (ví dụ, bộ mã hóa video 20 hoặc bộ giải mã video 30). Thiết bị tạo mã video 1002 bao gồm phương tiện nhận 1001. Phương tiện nhận 1001 được tạo cấu hình để nhận ảnh để mã hóa hoặc nhận luồng bit để giải mã. Thiết bị tạo mã video 1002 bao gồm phương tiện truyền 1007 được ghép nối với phương tiện nhận 1001. Phương tiện truyền 1007 được tạo cấu hình để truyền luồng bit đến bộ giải mã hoặc truyền hình ảnh được giải mã đến phương tiện hiển thị (ví dụ, một thiết bị trong số các thiết bị I/O 1080).

Thiết bị tạo mã video 1002 chứa phương tiện lưu trữ 1003. Phương tiện lưu trữ 1003 được lắp với ít nhất một phương tiện trong số phương tiện nhận 1001 hoặc phương tiện truyền 1007. Phương tiện lưu trữ 1003 được tạo cấu hình để lưu trữ các lệnh. Thiết bị tạo mã video 1002 cũng chứa phương tiện xử lý 1005. Phương tiện xử lý 1005 được lắp với phương tiện lưu trữ 1003. Phương tiện xử lý 1005 được tạo cấu hình để thực hiện các

lệnh được lưu trữ trong phương tiện lưu trữ 1003 để thực hiện các phương pháp được bộc lộ ở đây.

Cũng nên hiểu được rằng các bước của các phương pháp làm ví dụ mà được đề cập đến ở đây không nhất thiết phải được thực hiện theo thứ tự được mô tả, và thứ tự của các bước của các phương pháp này nên được hiểu chỉ là ví dụ. Tương tự, các bước bổ sung có thể được bao gồm trong các phương pháp này, và các bước nhất định có thể được bỏ qua hoặc được kết hợp, trong các phương pháp nhất quán với các phương án khác nhau của sáng chế.

Mặc dù một vài phương án đã được đề xuất theo sáng chế, nhưng nên hiểu được rằng các hệ thống và các phương pháp đã được bộc lộ có thể được thực hiện theo nhiều dạng cụ thể khác mà không nằm ngoài nguyên lý hoặc phạm vi của sáng chế. Các ví dụ sẽ được xem là minh họa sáng chế và không làm giới hạn sáng chế, và mục đích không bị giới hạn ở các phần chi tiết nêu ra ở đây. Ví dụ, các phần tử hoặc các thành phần khác nhau có thể được kết hợp hoặc tích hợp trong hệ thống khác hoặc các dấu hiệu nhất định có thể được bỏ qua, hoặc không được thực hiện.

Ngoài ra, các kỹ thuật, các hệ thống, các hệ thống con, và các phương pháp được mô tả và được minh họa trong các phương án khác nhau là rời rạc hoặc tách biệt có thể được kết hợp hoặc được tích hợp với các hệ thống, các môđun, các kỹ thuật, hoặc các phương pháp khác mà không nằm ngoài phạm vi của sáng chế. Các mục khác được thể hiện hoặc bàn luận như được ghép nối hoặc được ghép nối trực tiếp hoặc giao tiếp với nhau có thể được ghép nối hoặc giao tiếp gián tiếp thông qua một số giao diện, thiết bị hoặc thành phần trung gian, cho dù là về mặt điện, cơ học hay cách khác. Các ví dụ khác về các thay đổi, các thay thế, và các sửa đổi có thể thực hiện được bởi người có hiểu biết trung bình trong lĩnh vực và có thể được tạo ra mà không nằm ngoài nguyên lý và phạm vi được bộc lộ ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã luồng bit video đã được tạo mã được thực hiện bởi bộ giải mã video, phương pháp này bao gồm các bước:

xác định, bởi bộ giải mã video, xem liệu giá trị cho cờ thứ nhất có được cung cấp bởi đầu vào bên ngoài hay không;

thiết lập, bởi bộ giải mã video, cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) được tạo mã không bị xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài;

giải mã, bởi bộ giải mã video, ảnh GDR được tạo mã.

2. Phương pháp theo điểm 1, phương pháp này còn bao gồm bước thiết lập cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) được tạo mã và các ảnh theo sau bất kỳ giữa ảnh GDR được tạo mã và ảnh điểm khôi phục theo thứ tự xuất ra không bị xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài.

3. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 2, trong đó đầu vào bên ngoài là giao diện đồ họa người dùng (graphic user interface, GUI) của bộ giải mã video, trong đó giá trị của cờ thứ nhất được cung cấp bởi người sử dụng của bộ giải mã video sử dụng đầu vào bên ngoài.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó cờ thứ nhất được chỉ định là `HandleGdrAsCvsStartFlag`.

5. Phương pháp theo điểm 2, trong đó giá trị của cờ thứ nhất và cờ thứ hai được thiết lập là một, 1, để ngăn ảnh GDR được tạo mã và các ảnh theo sau bất kỳ giữa ảnh GDR được tạo mã và ảnh điểm khôi phục theo thứ tự xuất ra không bị xuất ra.

6. Phương pháp theo điểm 1, trong đó giá trị của cờ thứ nhất được thiết lập là không, 0, khi giá trị cho cờ thứ nhất không được cung cấp bởi đầu vào bên ngoài.

7. Thiết bị giải mã, bao gồm:

bộ nhận được tạo cấu hình để nhận dòng bit video được tạo mã;

bộ nhớ được ghép nối với bộ nhận, bộ nhớ để lưu trữ các lệnh; và

bộ xử lý được ghép nối với bộ nhớ, bộ xử lý được tạo cấu hình để thực thi các lệnh khiến cho thiết bị giải mã:

xác định xem liệu giá trị cho cờ thứ nhất có được cung cấp bởi đầu vào bên ngoài hay không;

thiết lập cờ thứ nhất bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) được tạo mã không bị xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài;

giải mã ảnh GDR được tạo mã.

8. Thiết bị giải mã theo điểm 7, trong đó cờ thứ nhất được thiết lập bằng giá trị được cung cấp bởi đầu vào bên ngoài và cờ thứ hai bằng cờ thứ nhất để ngăn ảnh làm mới giải mã từng bước (gradual decoding refresh, GDR) được tạo mã và các ảnh theo sau bất kỳ giữa ảnh GDR được tạo mã và ảnh điểm khôi phục theo thứ tự xuất ra không bị xuất ra khi giá trị cho cờ thứ nhất được cung cấp bởi đầu vào bên ngoài.

9. Thiết bị giải mã theo điểm bất kỳ trong số các điểm từ 7 đến 8, trong đó đầu vào bên ngoài là giao diện đồ họa người dùng (graphic user interface, GUI) của thiết bị giải mã, và trong đó giá trị của cờ thứ nhất được cung cấp bởi người dùng của thiết bị giải mã sử dụng đầu vào bên ngoài.

10. Thiết bị giải mã theo điểm bất kỳ trong số các điểm từ 7 đến 9, trong đó cờ thứ nhất được chỉ định là `HandleGdrAsCvsStartFlag`.

11. Thiết bị giải mã theo điểm 8, trong đó giá trị của cờ thứ nhất và cờ thứ hai được thiết lập là một, 1, để ngăn ảnh GDR được tạo mã và các ảnh theo sau bất kỳ giữa ảnh GDR được tạo mã và ảnh điểm khôi phục theo thứ tự xuất ra không bị xuất ra.

12. Thiết bị giải mã theo điểm 7, trong đó giá trị của cờ thứ nhất được thiết lập là không, 0, khi giá trị cho cờ thứ nhất không được cung cấp bởi đầu vào bên ngoài.

13. Hệ thống tạo mã, bao gồm:

bộ mã hóa; và

bộ giải mã kết nối với bộ mã hóa, trong đó bộ giải mã bao gồm thiết bị giải mã theo theo điểm bất kỳ trong số các điểm từ 7 đến 12.

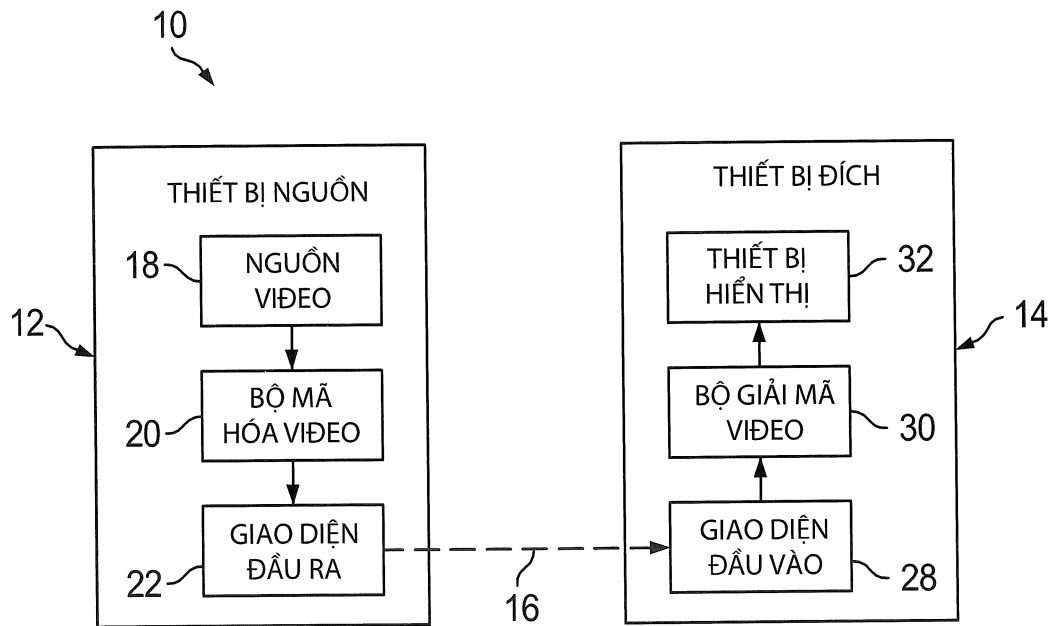


FIG. 1

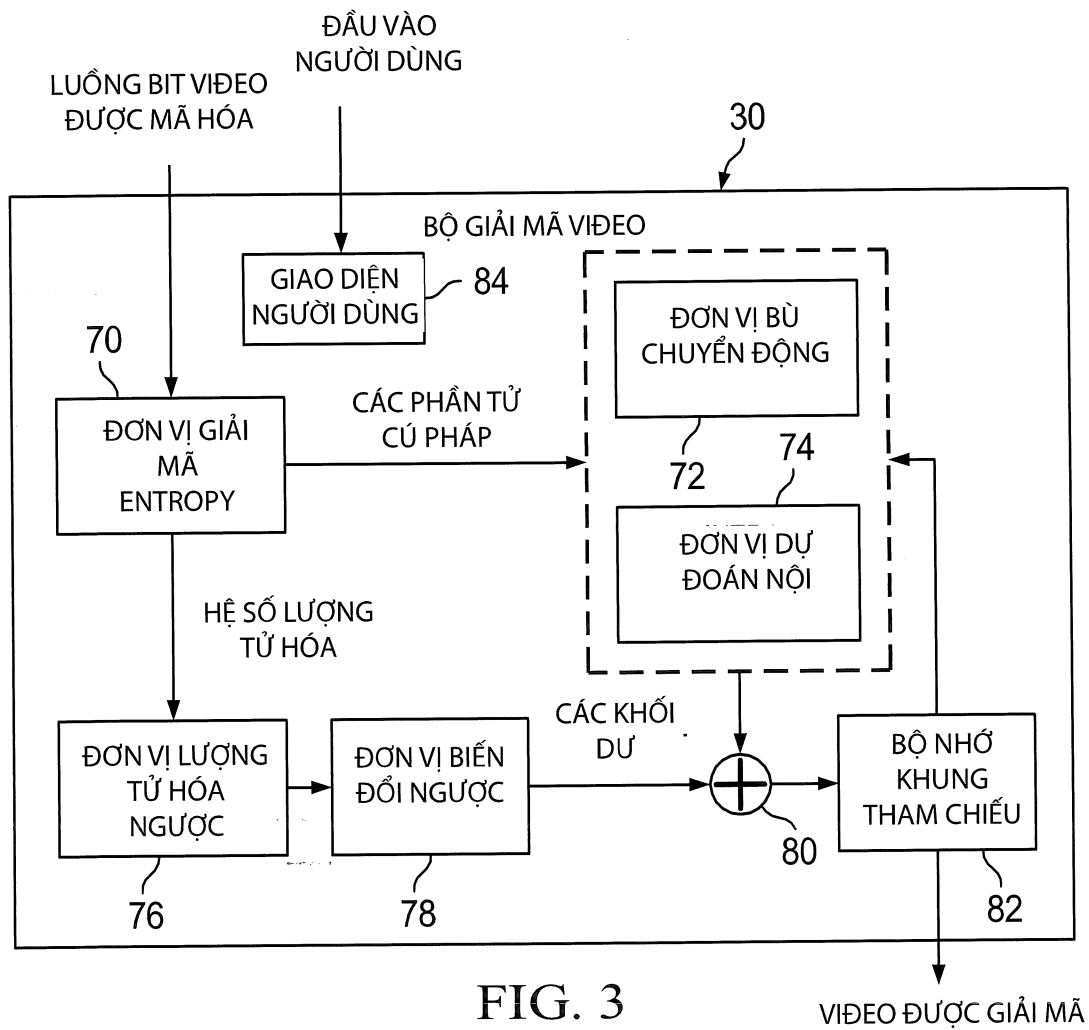


FIG. 3

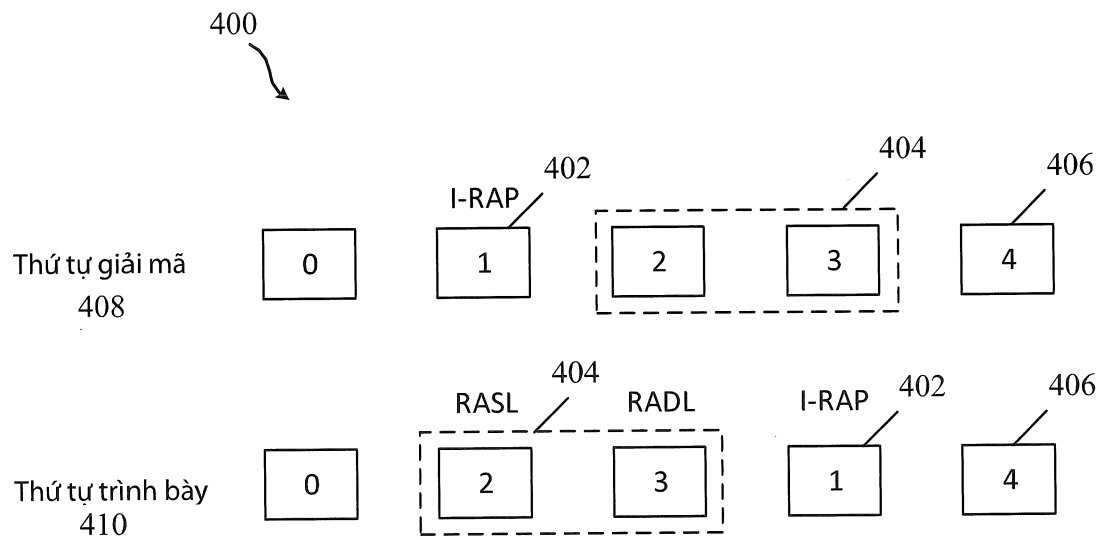


FIG. 4

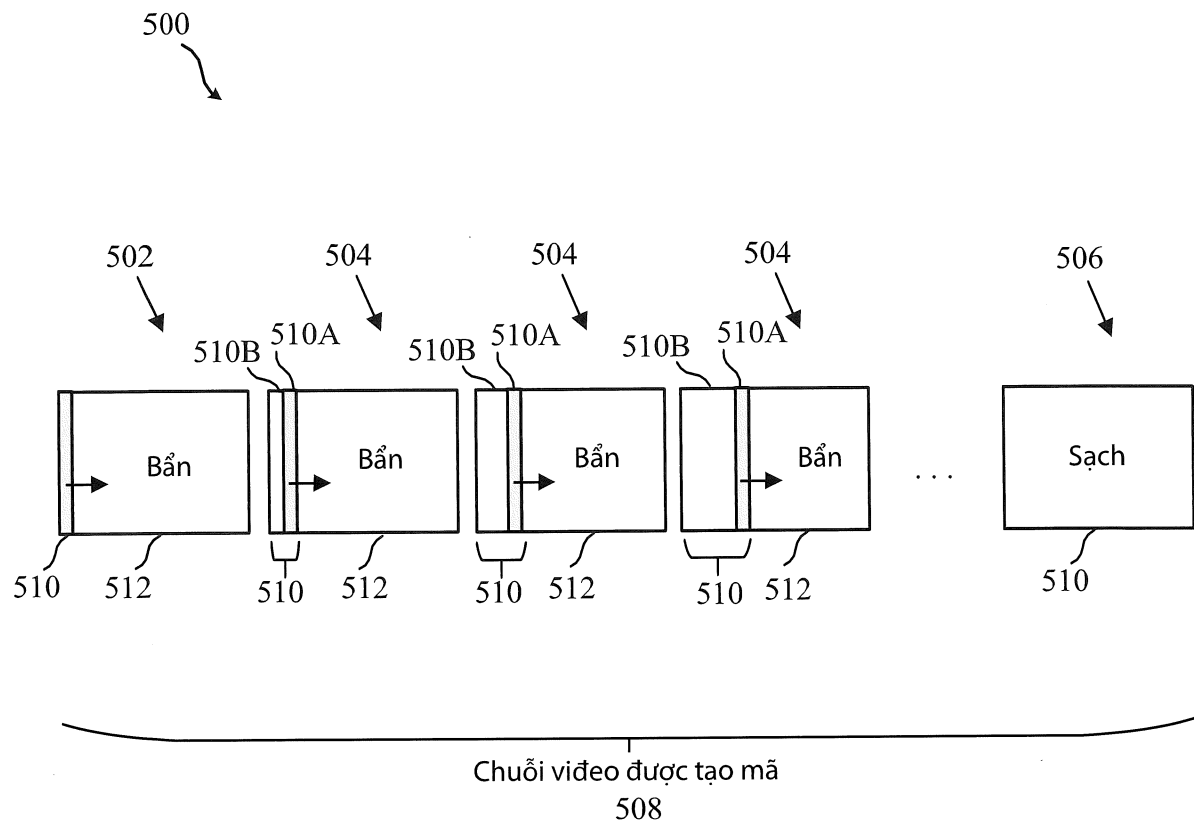


FIG. 5

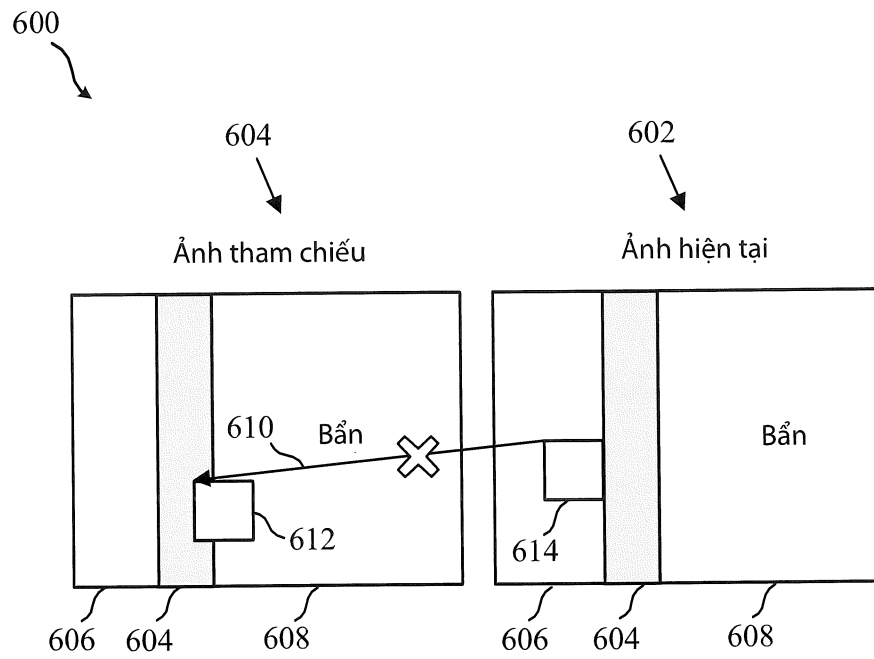


FIG. 6

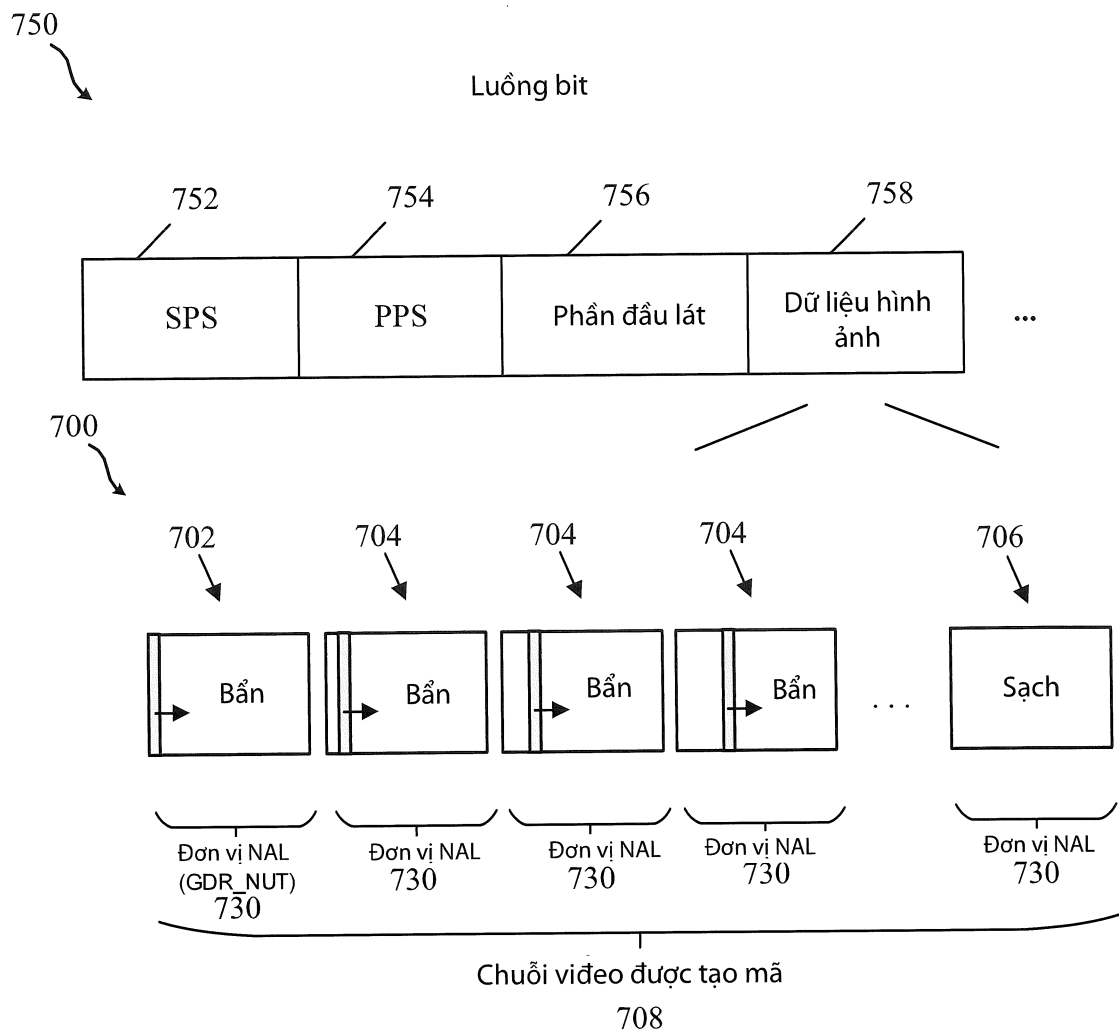


FIG. 7

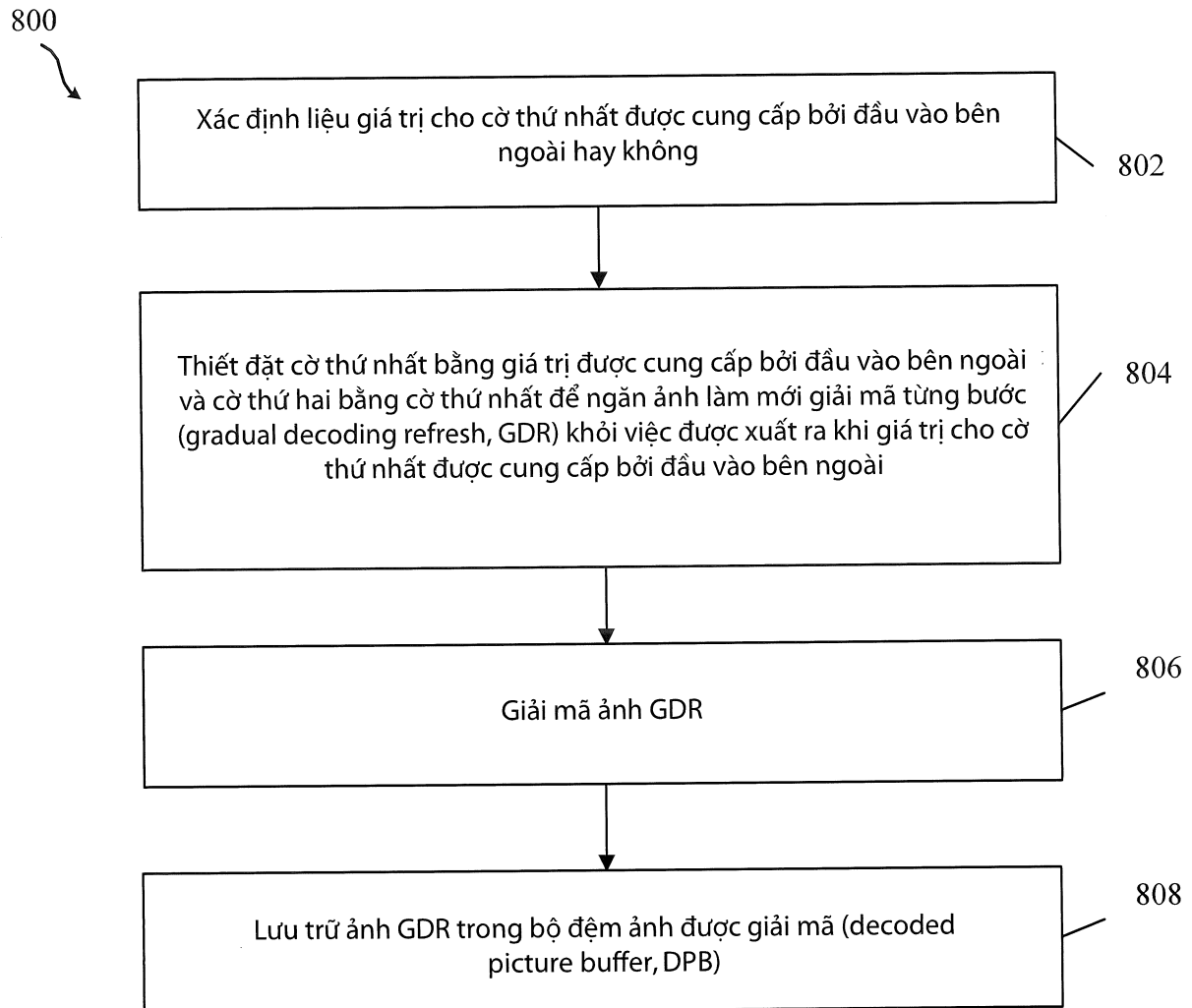


FIG. 8

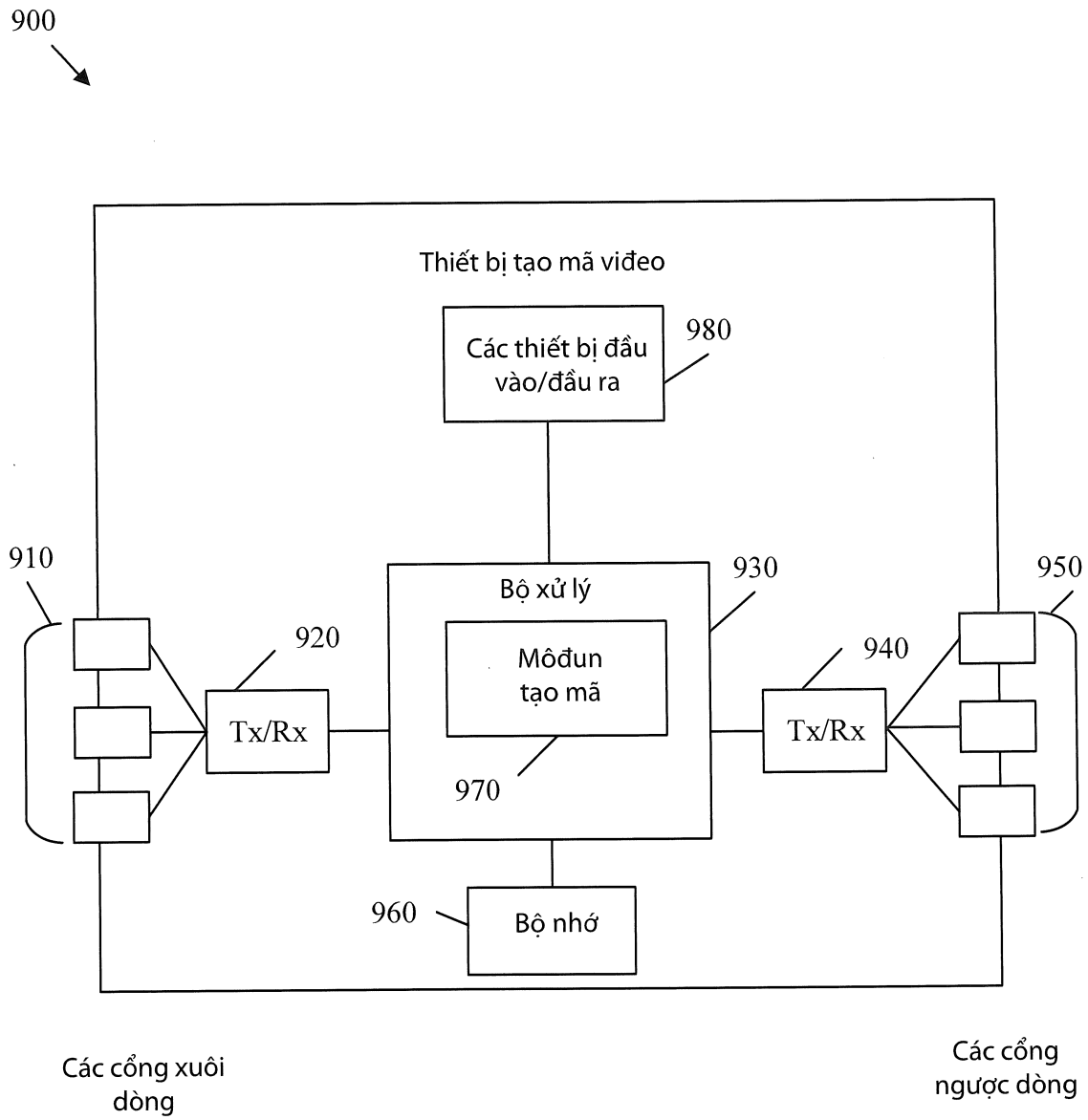


FIG. 9

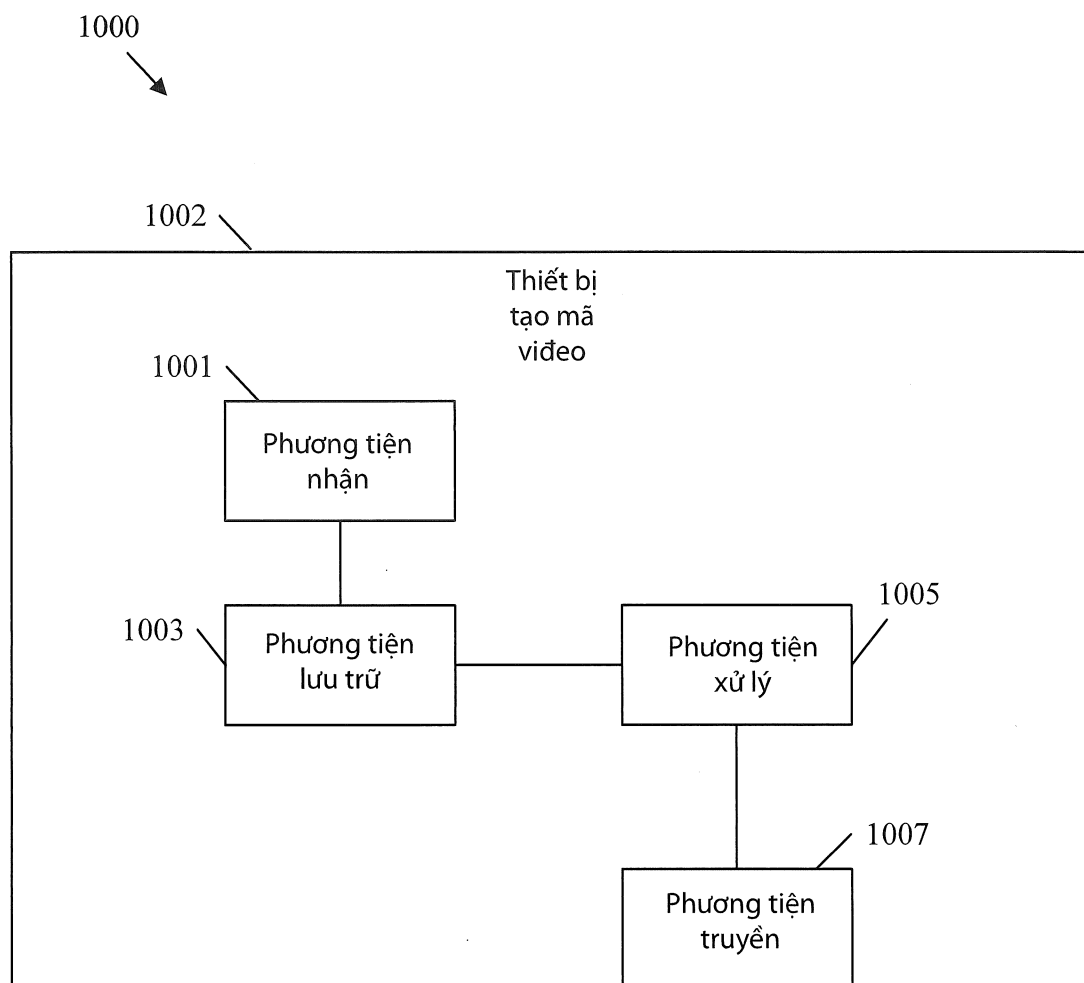


FIG. 10