(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) **1-0052138**

(51)[2021.01] **H04N 19/82**; H04N 19/132; H04N 19/176; H04N 19/70; H04N 19/117; H04N 19/14

(13) **B**

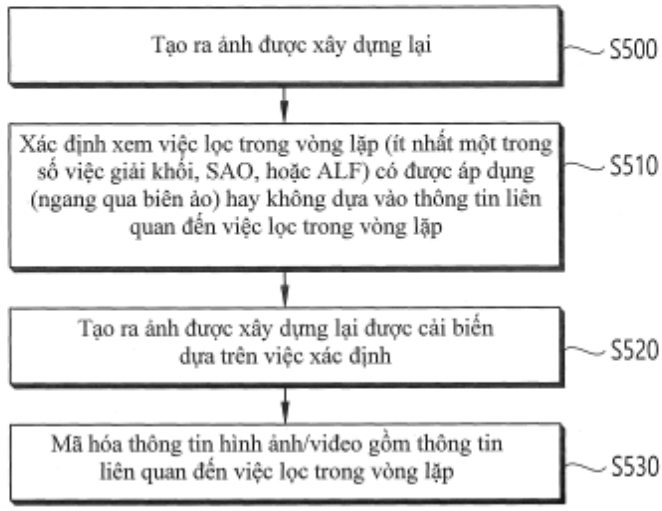
(21) 1-2022-03685
(22) 17/11/2020
(86) PCT/KR2020/016140 17/11/2020
(87) WO2021/101203 27/05/2021
(30) 62/937,230 18/11/2019 US
(45) 27/10/2025 451
(43) 25/08/2022 413A
(73) LG ELECTRONICS INC. (KR)
128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Korea
(72) HENDRY, Hendry (ID); PALURI, Seethal (IN); KIM, Seunghwan (KR).
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ HÌNH ẢNH, PHƯƠNG PHÁP MÃ HÓA HÌNH ẢNH VÀ PHƯƠNG PHÁP TRUYỀN DỮ LIỆU CHO HÌNH ẢNH

(21) 1-2022-03685

(57) Sáng chế đề cập đến phương pháp giải mã hình ảnh, phương pháp mã hóa hình ảnh, phương tiện lưu trữ đọc được bởi máy tính không chuyển tiếp và phương pháp truyền dữ liệu cho hình ảnh. Theo các phương án của sáng chế, thông tin để thực hiện việc lọc trong vòng lặp ngang qua các biên ảo có thể được báo hiệu một cách hiệu quả. Ví dụ, việc lọc trong vòng lặp có thể được thực hiện trên cơ sở việc báo hiệu của thông tin mà liên quan đến việc liệu việc lọc trong vòng lặp ngang qua các biên ảo có được cho phép hay không.

FIG. 5



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và thiết bị lập mã hình ảnh dựa trên việc lọc.

Tình trạng kỹ thuật của sáng chế

Hiện tại, nhu cầu về hình ảnh/viđeo độ phân giải cao, chất lượng cao như là hình ảnh/viđeo 4K hoặc 8K hoặc độ phân giải cực cao (Ultra High Definition, UHD) khác đang gia tăng trong các lĩnh vực khác nhau. Vì dữ liệu hình ảnh/viđeo có độ phân giải cao và chất lượng cao, nên lượng thông tin hoặc các bit cần được truyền gia tăng tương đối với dữ liệu hình ảnh/viđeo sẵn có, và vì thế, việc truyền dữ liệu hình ảnh nhờ sử dụng phương tiện như là đường băng rộng có dây/không dây sẵn có hoặc phương tiện lưu trữ sẵn có hoặc lưu trữ dữ liệu hình ảnh/viđeo nhờ sử dụng phương tiện lưu trữ sẵn có làm tăng chi phí truyền và chi phí lưu trữ.

Bên cạnh đó, sự quan tâm và nhu cầu cho phương tiện nhập vai như là nội dung thực tế ảo (Virtual Reality, VR) và thực tế nhân tạo (Artificial Reality, AR) hoặc ảnh ba chiều gần đây đã tăng lên và việc phát rộng cho hình ảnh/viđeo đang có các đặc tính khác với các hình ảnh ảo như là các hình ảnh trò chơi đã gia tăng.

Theo đó, cần có công nghệ nén hình ảnh/viđeo hiệu quả cao để nén, truyền, lưu trữ, và tái sản xuất một cách hiệu quả thông tin của hình ảnh/viđeo độ phân giải cao, chất lượng cao có các đặc tính khác nhau như được mô tả ở trên.

Cụ thể là, quy trình lọc trong vòng lặp được thực hiện để làm tăng chất lượng trực quan chủ quan/khách quan, và có thảo luận về sơ đồ làm tăng hiệu quả báo hiệu của thông tin để thực hiện việc lọc trong vòng lặp dựa trên các biên ảo.

Bản chất kỹ thuật của sáng chế

Theo một phương án của sáng chế, phương pháp và thiết bị làm tăng hiệu quả lập mã hình ảnh được đề xuất.

Theo một phương án của sáng chế, phương pháp và thiết bị ứng dụng lọc hiệu quả được đề xuất.

Theo một phương án của sáng chế, phương pháp và thiết bị để áp dụng một cách hiệu quả việc giải khối, dịch chuyển thích ứng mẫu (sample adaptive offset, SAO), và lọc vòng lặp thích ứng (adaptive loop filter, ALF) được đề xuất.

Theo một phương án của sáng chế, việc lọc trong vòng lặp có thể được thực hiện dựa trên các biên ảo.

Theo một phương án của sáng chế, tập thông số trình tự (sequence parameter set, SPS) có thể gồm cờ được cho phép các biên ảo SPS chỉ ra xem việc lọc trong vòng lặp có được thực hiện ngang qua các biên ảo hay không.

Theo một phương án của sáng chế, việc lọc trong vòng lặp có thể được thực hiện ngang qua các biên ảo, dựa trên cờ được cho phép các biên ảo SPS.

Theo một phương án của sáng chế, thiết bị mã hóa để thực hiện việc mã hóa viđeo/hình ảnh được đề xuất.

Theo một phương án của sáng chế, có đề xuất phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính trong đó thông tin viđeo/hình ảnh được mã hóa, được tạo ra theo phương pháp mã hóa viđeo/hình ảnh được bộc lộ trong ít nhất một trong các phương án của sáng chế, được lưu trữ.

Theo một phương án của sáng chế, có đề xuất phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính trong đó thông tin được mã hóa hoặc thông tin viđeo/hình ảnh được mã hóa, khiến thực hiện phương pháp giải mã viđeo/hình ảnh được bộc lộ trong ít nhất một trong các phương án của sáng chế bởi thiết bị giải mã, được lưu trữ.

Theo một phương án của sáng chế, hiệu quả nén hình ảnh/viđeo tổng cộng có thể được cải thiện.

Theo một phương án của sáng chế, chất lượng trực quan chủ động/bị động có thể được cải thiện thông qua việc lọc hiệu quả.

Quy trình lọc trong vòng lặp dựa trên các biên ảo theo một phương án của sáng chế có thể tiết kiệm tài nguyên phần cứng.

Theo một phương án của sáng chế, quy trình lọc trong vòng lặp dựa trên các biên ảo có thể được thực hiện một cách hiệu quả, và hiệu quả lọc có thể được cải thiện.

Theo một phương án của sáng chế, thông tin về việc lọc trong vòng lặp dựa trên

các biên ảo có thể được báo hiệu một cách hiệu quả.

Mô tả vắn tắt các hình vẽ

Fig.1 thể hiện sơ lược ví dụ của hệ thống lập mã viđeo/hình ảnh mà cho nó các phương án của sáng chế có thể được áp dụng.

Fig.2 là hình vẽ minh họa sơ lược cấu hình của thiết bị mã hóa viđeo/hình ảnh mà cho nó các phương án của sáng chế có thể được áp dụng.

Fig.3 là hình vẽ minh họa sơ lược cấu hình của thiết bị giải mã viđeo/hình ảnh mà cho nó các phương án của sáng chế có thể được áp dụng.

Fig.4 thể hiện ví dụ kiến trúc phân cấp cho viđeo/hình ảnh được lập mã.

Fig.5 là lưu đồ minh họa phương pháp mã hóa dựa trên việc lọc trong thiết bị mã hóa.

Fig.6 là lưu đồ minh họa phương pháp giải mã dựa trên việc lọc trong thiết bị giải mã.

Fig.7 và Fig.8 thể hiện một cách sơ lược ví dụ của phương pháp mã hóa viđeo/hình ảnh và các thành phần liên quan theo (các) phương án của sáng chế.

Fig.9 và Fig.10 thể hiện một cách sơ lược ví dụ của phương pháp giải mã hình ảnh/viđeo và các thành phần liên quan theo (các) phương án của sáng chế.

Fig.11 thể hiện ví dụ của hệ thống phát luồng nội dung cho nó (các) phương án được bộc lộ trong sáng chế có thể được áp dụng.

Mô tả chi tiết sáng chế

Sáng chế có thể được sửa đổi theo các dạng khác nhau, và các phương án cụ thể của sáng chế sẽ được mô tả và được minh họa trên các hình vẽ. Tuy nhiên, các phương án này không nhằm để giới hạn sáng chế. Các thuật ngữ được sử dụng trong phần mô tả sau đây được sử dụng chỉ đơn thuần để mô tả các phương án cụ thể, chứ không nhằm giới hạn sáng chế. Sự trình bày dạng số ít sẽ gồm sự trình bày dạng số nhiều, miễn là nó được đọc khác nhau một cách rõ ràng. Các từ ngữ như "gồm" và "có" được nhằm để chỉ ra rằng các dấu hiệu, các số lượng, các bước, các hoạt động, các phần tử, các thành phần, hoặc các tổ hợp của chúng, mà được sử dụng trong phần mô tả sau đây, là tồn tại, và vì

thế, nên được hiểu rằng khả năng tồn tại hoặc bổ sung của một hoặc nhiều dấu hiệu, số lượng, bước, hoạt động, phần tử, thành phần khác, hoặc những tổ hợp của chúng, là không bị loại trừ.

Ngoài ra, mỗi cấu hình của các hình vẽ được mô tả trong sáng chế là minh họa độc lập để giải thích các chức năng như các dấu hiệu mà khác nhau, và không có nghĩa là mỗi cấu hình được thực hiện bởi phần cứng khác nhau hoặc phần mềm khác nhau tương hỗ. Ví dụ, hai hoặc nhiều cấu hình có thể được kết hợp để tạo thành một cấu hình, và một cấu hình cũng có thể được chia thành nhiều cấu hình. Không trệch khỏi bản chất của sáng chế, các phương án trong đó các cấu hình được kết hợp và/hoặc được phân tách được gồm trong phạm vi của các điểm yêu cầu bảo hộ.

Ở dưới đây, các ví dụ về các phương án này sẽ được mô tả chi tiết với sự tham khảo đến các hình vẽ kèm theo. Bên cạnh đó, các số chỉ dẫn giống nhau được dùng để chỉ các thành phần giống nhau trong suốt các hình vẽ, và các phần mô tả giống nhau của các thành phần giống nhau này sẽ được lược bỏ.

Sáng chế liên quan đến việc lập mã viđeo/hình ảnh. Ví dụ, các phương pháp/các phương án được bộc lộ trong tài liệu này có thể liên quan tới tiêu chuẩn lập mã vi đeo vạn năng (versatile video coding, VVC) (ITU-T Rec. H.266), tiêu chuẩn lập mã viđeo/hình ảnh thế hệ tiếp theo sau VVC, hoặc các tiêu chuẩn liên quan đến việc lập mã viđeo khác (ví dụ, tiêu chuẩn lập mã viđeo hiệu quả cao (high efficiency video coding, HEVC) (ITU-T Rec. H.265), tiêu chuẩn lập mã viđeo thiết yếu (essential video coding, EVC), tiêu chuẩn AVS2, và tương tự).

Sáng chế đề xuất các phương án khác nhau của việc lập mã viđeo/hình ảnh, và các phương án ở trên có thể được thực hiện trong sự tổ hợp với nhau trừ khi được chỉ rõ theo cách khác.

Trong sáng chế, viđeo có thể đề cập đến một loạt các hình ảnh theo thời gian. Ảnh đề cập chung đến đơn vị biểu diễn một hình ảnh tại một khung thời gian cụ thể, và lát/phiến đề cập đến đơn vị cấu thành một phần của ảnh về việc lập mã. Lát/phiến có thể gồm một hoặc nhiều đơn vị cây lập mã (Coding Tree Unit, CTU). Một ảnh có thể gồm có một hoặc nhiều lát/phiến. Một ảnh có thể gồm một hoặc nhiều nhóm phiến. Một nhóm phiến có thể gồm một hoặc nhiều phiến.

Điểm ảnh (pixel) hoặc pel có thể có nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc hình ảnh). Ngoài ra, 'mẫu' có thể được sử dụng như là thuật ngữ tương ứng với điểm ảnh. Mẫu có thể biểu diễn chung cho điểm ảnh hoặc giá trị của điểm ảnh, và có thể chỉ biểu diễn điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ.

Đơn vị có thể biểu diễn đơn vị cơ bản của việc xử lý hình ảnh. Đơn vị có thể gồm ít nhất một thành phần trong số vùng cụ thể của ảnh và thông tin liên quan đến vùng. Một đơn vị có thể gồm một khối độ chói và hai khối sắc độ (ví dụ cb, cr). Đơn vị có thể được sử dụng theo kiểu thay thế cho nhau theo nghĩa như là khối hoặc khu vực trong một số trường hợp. Trong trường hợp chung, các khối M×N có thể gồm các mẫu (hoặc các mảng mẫu) hoặc tập hợp (hoặc các mảng) của các hệ số biến đổi của M cột và N hàng. Theo cách thay thế, mẫu có thể có nghĩa là giá trị điểm ảnh trong miền không gian, hoặc khi giá trị điểm ảnh này được chuyển đổi thành miền tần số, thì nó có thể có nghĩa là hệ số biến đổi trong miền tần số.

Trong bản mô tả này, thuật ngữ "/" và "," nên được hiểu là để chỉ ra "và/hoặc." Ví dụ, cách thể hiện "A/B" có thể nghĩa là "A và/hoặc B." Ngoài ra, "A, B" có thể nghĩa là "A và/hoặc B." Ngoài ra, "A/B/C" có thể nghĩa là "ít nhất một trong số A, B, và/hoặc C." Ngoài ra, "A/B/C" có thể nghĩa là "ít nhất một trong số A, B, và/hoặc C."

Ngoài ra, trong sáng chế, thuật ngữ "hoặc" nên được hiểu là để chỉ ra "và/hoặc." Ví dụ, việc diễn đạt "A hoặc B" có thể bao gồm 1) chỉ A, 2) chỉ B, và/hoặc 3) cả A và B. Nói cách khác, thuật ngữ "hoặc" trong bản mô tả này nên được hiểu là để chỉ ra "ngoài ra hoặc mặt khác."

Theo bản mô tả này, "ít nhất một trong số A và B" có thể nghĩa là "chỉ A", "chỉ B", hoặc "cả A và B". Ngoài ra, trong bản mô tả này, việc diễn đạt "ít nhất một trong số A hoặc B" hoặc "ít nhất một trong số A và/hoặc B" có thể được diễn giải giống như "ít nhất một trong số A và B".

Ngoài ra, trong bản mô tả này, "ít nhất một trong số A, B và C" có thể nghĩa là "chỉ A", "chỉ B", "chỉ C", hoặc "kết hợp bất kỳ của A, B và C". Ngoài ra, "ít nhất một trong số A, B hoặc C" hoặc "ít nhất một trong số A, B và/hoặc C" có thể nghĩa là "ít nhất một trong số A, B và C".

Ngoài ra, dấu ngoặc đơn được sử dụng trong bản mô tả này có thể nghĩa là "ví dụ". Cụ thể là, trong trường hợp mà "việc dự đoán (việc dự đoán nội)" được diễn đạt, nó có thể được chỉ ra rằng "việc dự đoán nội" được đề xuất như ví dụ của "việc dự đoán". Nói cách khác, thuật ngữ "việc dự đoán" trong bản mô tả này không bị giới hạn ở "việc dự đoán nội", và nó có thể được chỉ ra rằng "việc dự đoán nội" được đề xuất như ví dụ của "việc dự đoán". Ngoài ra, ngay cả trong trường hợp mà "việc dự đoán (nghĩa là, việc dự đoán nội)" được diễn đạt, nó có thể được chỉ ra rằng "việc dự đoán nội" được đề xuất như ví dụ của "việc dự đoán".

Trong bản mô tả này, các dấu hiệu kỹ thuật được mô tả riêng lẻ trong một hình vẽ có thể được thực hiện một cách riêng lẻ, hoặc có thể được thực hiện đồng thời.

Fig.1 minh họa ví dụ của hệ thống lập mã viđeo/hình ảnh mà cho nó phần bộc lộ của sáng chế có thể được áp dụng.

Tham khảo đến Fig.1, hệ thống lập mã viđeo/hình ảnh có thể gồm thiết bị nguồn và thiết bị nhận. Thiết bị nguồn có thể truyền thông tin viđeo/hình ảnh đã được lập mã hoặc dữ liệu đến thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng trong dạng tệp hoặc phát luồng.

Thiết bị nguồn có thể gồm nguồn viđeo, thiết bị mã hóa, và bộ truyền. Thiết bị nhận có thể gồm bộ nhận, thiết bị giải mã, và bộ kết xuất. Thiết bị mã hóa có thể được gọi là thiết bị mã hóa viđeo/hình ảnh, và thiết bị giải mã có thể được gọi là thiết bị giải mã viđeo/hình ảnh. Bộ truyền có thể được gồm trong thiết bị mã hóa. Bộ nhận có thể được gồm trong thiết bị giải mã. Bộ kết xuất có thể gồm bộ phận hiển thị, bộ phận hiển thị có thể được tạo cấu hình là thiết bị riêng biệt hoặc thành phần ngoài.

Nguồn viđeo có thể giành được viđeo/hình ảnh thông qua quy trình chụp, tổng hợp, hoặc tạo ra viđeo/hình ảnh. Nguồn viđeo có thể gồm thiết bị chụp viđeo/hình ảnh và/hoặc thiết bị tạo ra viđeo/hình ảnh. Thiết bị chụp viđeo/hình ảnh có thể gồm, ví dụ, một hoặc nhiều camêra, kho lưu trữ viđeo/hình ảnh gồm các viđeo/hình ảnh đã được chụp trước đó, và tương tự. Thiết bị tạo ra viđeo/hình ảnh có thể gồm, ví dụ, máy tính, máy tính bảng, và điện thoại thông minh, và có thể tạo ra các viđeo/hình ảnh (theo cách điện). Ví dụ, viđeo/hình ảnh ảo có thể được tạo ra thông qua máy tính hoặc tương tự. Trong trường hợp này, quy trình chụp viđeo/hình ảnh có thể được thay thế bởi quy trình

tạo ra dữ liệu liên quan.

Thiết bị mã hóa có thể mã hóa viđeo/hình ảnh đầu vào. Thiết bị mã hóa có thể thực hiện chuỗi các quy trình chẳng hạn như dự đoán, biến đổi, và lượng tử hóa cho hiệu quả nén và lập mã. Dữ liệu được mã hóa (thông tin viđeo/hình ảnh được mã hóa) có thể được xuất ra trong dạng luồng bit.

Bộ truyền có thể truyền hình ảnh/thông tin hình ảnh được mã hóa hoặc dữ liệu được xuất trong dạng luồng bit đến bộ nhận của thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng trong dạng tệp hoặc phát luồng. Phương tiện lưu trữ kỹ thuật số có thể gồm các phương tiện lưu trữ khác nhau như là USB, SD, CD, DVD, Blu-ray, HDD, SSD, và tương tự. Bộ truyền có thể gồm phần tử để tạo ra tệp phương tiện thông qua định dạng tệp được xác định trước và có thể gồm phần tử để truyền thông qua mạng phát rộng/truyền thông. Bộ nhận có thể nhận/trích xuất luồng bit và truyền luồng bit nhận được đến thiết bị giải mã.

Thiết bị giải mã có thể giải mã viđeo/hình ảnh bằng cách thực hiện một loạt các quy trình như là khử lượng tử hóa, biến đổi ngược, và dự đoán tương ứng với hoạt động của thiết bị mã hóa.

Bộ kết xuất có thể kết xuất viđeo/hình ảnh được giải mã. Viđeo/hình ảnh được kết xuất có thể được hiển thị qua bộ phận hiển thị.

Fig.2 là sơ đồ minh họa sơ lược cấu hình của thiết bị mã hóa viđeo/hình ảnh mà cho nó phần bộc lộ của sáng chế có thể được áp dụng. Ở dưới đây, phần được gọi là thiết bị mã hóa viđeo có thể gồm thiết bị mã hóa hình ảnh.

Tham khảo đến Fig.2, thiết bị mã hóa 200 có thể gồm và được tạo cấu hình với bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entrôpi 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể gồm bộ dự đoán liên 221 và bộ dự đoán nội 222. Bộ xử lý phần dư 230 có thể gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ khử lượng tử hóa 234, và bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ xây dựng lại hoặc bộ tạo khối được xây dựng lại. Bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entrôpi 240, bộ cộng 250, và bộ lọc 260, mà đã được mô tả ở trên, có thể được tạo cấu hình bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ

chip bộ mã hóa hoặc các bộ xử lý) theo một phương án. Ngoài ra, bộ nhớ 270 có thể gồm bộ đệm ảnh được giải mã (decoded picture buffer, DPB), và cũng có thể được tạo cấu hình bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 270 như là thành phần trong/ngoài.

Bộ phân vùng hình ảnh 210 có thể chia hình ảnh đầu vào (hoặc, ảnh, khung) được nhập vào thiết bị mã hóa 200 thành một hoặc nhiều đơn vị xử lý. Như một ví dụ, đơn vị xử lý có thể được gọi là đơn vị lập mã (coding unit, CU). Trong trường hợp này, đơn vị lập mã có thể được chia theo cách đệ quy theo cấu trúc cây tứ phân cây nhị phân cây tam phân (Quad-tree binary-tree ternary-tree, QTBTTT) từ đơn vị cây lập mã (coding tree unit, CTU) hoặc đơn vị lập mã lớn nhất (largest coding unit, LCU). Ví dụ, một đơn vị lập mã có thể được chia thành nhiều đơn vị lập mã có độ sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân. Trong trường hợp này, ví dụ, cấu trúc cây tứ phân được áp dụng đầu tiên và cấu trúc cây nhị phân và/hoặc cấu trúc cây tam phân có thể được áp dụng sau đó. Mặt khác, cấu trúc cây nhị phân cũng có thể được áp dụng trước tiên. Quy trình lập mã theo sáng chế có thể được thực hiện dựa trên đơn vị lập mã cuối cùng mà không được chia tiếp nữa. Trong trường hợp này, dựa trên hiệu quả lập mã theo các đặc tính hình ảnh hoặc tương tự, đơn vị lập mã tối đa có thể được sử dụng trực tiếp như đơn vị lập mã cuối cùng, hoặc nếu cần thiết, đơn vị lập mã có thể được chia theo cách đệ quy thành các đơn vị lập mã có độ sâu sâu hơn, sao cho đơn vị lập mã có kích thước tối ưu có thể được sử dụng làm đơn vị lập mã cuối cùng. Ở đây, quy trình lập mã có thể gồm quy trình chẳng hạn như dự đoán, biến đổi, và xây dựng lại sẽ được mô tả sau. Theo một ví dụ khác, đơn vị xử lý có thể còn gồm đơn vị dự đoán (prediction unit, PU) hoặc đơn vị biến đổi (transform unit, TU). Trong trường hợp này, mỗi trong số đơn vị dự đoán và đơn vị biến đổi có thể được chia tách hoặc được phân vùng từ đơn vị lập mã cuối cùng được đề cập ở trên. Đơn vị dự đoán có thể là đơn vị của việc dự đoán mẫu, và đơn vị biến đổi có thể là đơn vị để tạo ra hệ số biến đổi và/hoặc đơn vị để tạo ra tín hiệu phần dư từ hệ số biến đổi.

Đơn vị có thể được sử dụng thay thế lẫn nhau với thuật ngữ chẳng hạn như khối hoặc khu vực trong một số trường hợp. Thông thường, khối MxN có thể biểu diễn các mẫu được cấu thành từ M cột và N hàng hoặc nhóm các hệ số biến đổi. Mẫu có thể biểu diễn chung điểm ảnh hoặc giá trị của điểm ảnh, và có thể cũng biểu diễn chỉ điểm ảnh/giá

trị điểm ảnh của thành phần độ chói, và cũng biểu diễn chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ. Mẫu có thể được sử dụng làm thuật ngữ tương ứng với điểm ảnh hoặc pel cấu hình một ảnh (hoặc hình ảnh).

Bộ trừ 231 có thể tạo ra tín hiệu phần dư (khối phần dư, các mẫu phần dư, hoặc mảng mẫu phần dư) bằng việc trừ tín hiệu dự đoán (khối được dự đoán, các mẫu dự đoán, hoặc mảng mẫu dự đoán) được xuất từ bộ dự đoán 220 từ tín hiệu hình ảnh đầu vào (khối gốc, các mẫu gốc, hoặc mảng mẫu gốc), và tín hiệu phần dư được tạo ra được truyền tới bộ biến đổi 232. Bộ dự đoán 220 có thể thực hiện việc dự đoán cho khối mục tiêu xử lý (sau đây, được gọi là "khối hiện tại") và tạo ra khối được dự đoán gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán 220 có thể xác định liệu việc dự đoán nội hay việc dự đoán liên được áp dụng trên khối hiện tại hay trong đơn vị CU. Như được mô tả ở dưới trong phần mô tả của mỗi chế độ dự đoán, bộ dự đoán có thể tạo ra các loại thông tin khác nhau liên quan đến việc dự đoán, như thông tin chế độ dự đoán, và chuyển thông tin được tạo ra đến bộ mã hóa entrôpi 240. Thông tin về việc dự đoán có thể được mã hóa trong bộ mã hóa entrôpi 240 và được xuất ra trong dạng luồng bit.

Bộ dự đoán nội 222 có thể dự đoán khối hiện tại với tham chiếu tới các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt lân cận với khối hiện tại, hoặc cũng có thể được đặt xa khỏi khối hiện tại theo chế độ dự đoán. Các chế độ dự đoán trong việc dự đoán nội có thể gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Chế độ không có hướng có thể gồm, ví dụ, chế độ DC hoặc chế độ phẳng. Chế độ có hướng có thể gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng theo độ chi tiết của hướng dự đoán. Tuy nhiên, đây là minh họa và các chế độ dự đoán có hướng mà lớn hơn hoặc nhỏ hơn số lượng nêu trên có thể được sử dụng theo việc thiết đặt. Bộ dự đoán nội 222 cũng có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên 221 có thể tạo ra khối được dự đoán của khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vectơ chuyển động trên ảnh tham chiếu. Tại thời điểm này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động có thể được dự đoán trong các đơn vị của khối, khối con, hoặc mẫu dựa trên sự tương liên của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm vectơ chuyển động và

chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin hướng dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, hoặc tương tự). Trong trường hợp việc dự đoán liên, khối lân cận có thể gồm khối lân cận theo không gian có mặt trong ảnh hiện tại và khối lân cận theo thời gian có mặt trong ảnh tham chiếu. Ảnh tham chiếu gồm khối tham chiếu và ảnh tham chiếu gồm khối lân cận theo thời gian cũng có thể là giống với nhau, và cũng có thể là khác nhau. Khối lân cận theo thời gian có thể được gọi tên chẳng hạn như khối tham chiếu được đặt cùng một chỗ, CU được đặt cùng một chỗ (collocated CU, colCU), hoặc tương tự, và ảnh tham chiếu gồm khối lân cận theo thời gian cũng có thể được gọi là ảnh được đặt cùng một chỗ (collocated picture, colPic). Ví dụ, bộ dự đoán liên 221 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận, và tạo ra thông tin chỉ ra ứng viên nào được sử dụng để dẫn xuất vectơ chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và ví dụ, trong trường hợp chế độ bỏ qua và chế độ hợp nhất, bộ dự đoán liên 221 có thể sử dụng thông tin chuyển động của khối lân cận như thông tin chuyển động của khối hiện tại. Trong trường hợp chế độ bỏ qua, tín hiệu phần dư có thể không được truyền không giống chế độ hợp nhất. Chế độ dự đoán vectơ chuyển động (motion vector prediction, MVP) có thể chỉ ra vectơ chuyển động của khối hiện tại bằng việc sử dụng vectơ chuyển động của khối lân cận làm bộ dự đoán vectơ chuyển động, và báo hiệu sự chênh lệch vectơ chuyển động.

Bộ dự đoán 220 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau được mô tả dưới đây. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc dự đoán nội hoặc việc dự đoán liên để dự đoán một khối mà cũng có thể áp dụng một cách đồng thời cả việc dự đoán nội và việc dự đoán liên. Việc này có thể được gọi là việc dự đoán nội và liên được kết hợp (inter and intra prediction, CIIP). Ngoài ra, bộ dự đoán có thể thực hiện sao chép khối nội (intra block copy, IBC) cho việc dự đoán của khối. Sao chép khối nội có thể được sử dụng cho việc lập mã hình ảnh động/hình ảnh nội dung của trò chơi hoặc tương tự, ví dụ, lập mã nội dung màn (screen content coding, SCC). IBC về cơ bản thực hiện việc dự đoán trong ảnh hiện tại mà có thể được thực hiện tương tự như việc dự đoán liên mà trong đó khối tham chiếu được dẫn xuất trong ảnh hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật dự đoán liên được mô tả

trong sáng chế.

Tín hiệu dự đoán được tạo ra thông qua bộ dự đoán liên 221 và/hoặc bộ dự đoán nội 222 có thể được sử dụng để tạo ra tín hiệu được xây dựng lại hoặc để tạo ra tín hiệu dư. Bộ biến đổi 232 có thể tạo ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu dư. Ví dụ, kỹ thuật biến đổi có thể gồm ít nhất một kỹ thuật trong số biến đổi côsin rời rạc (Discrete Cosine Transform, DCT), biến đổi sin rời rạc (Discrete Sine Transform, DST), biến đổi dựa trên đồ thị (Graph-Based Transform, GBT), hoặc biến đổi phi tuyến tùy theo điều kiện (Conditionally Non-Linear Transform, CNT). Ở đây, GBT nghĩa là việc biến đổi thu nhận được từ đồ thị khi thông tin quan hệ giữa các điểm ảnh được biểu diễn bởi đồ thị. CNT đề cập đến việc biến đổi thu được dựa trên tín hiệu dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được xây dựng lại trước đó. Bên cạnh đó, quy trình biến đổi có thể được áp dụng cho các khối điểm ảnh hình vuông có cùng kích thước hoặc có thể được áp dụng cho các khối có kích thước khác hình vuông.

Bộ lượng tử hóa 233 có thể lượng tử hóa các hệ số biến đổi và truyền chúng đến bộ mã hóa entrôpi 240 và bộ mã hóa entrôpi 240 có thể mã hóa tín hiệu được lượng tử hóa (thông tin về các hệ số biến đổi được lượng tử hóa) và xuất ra luồng bit. Thông tin về các hệ số biến đổi được lượng tử hóa có thể được gọi là thông tin phần dư. Bộ lượng tử hóa 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa loại khối thành dạng vectơ một chiều dựa trên thứ tự quét hệ số, và tạo ra thông tin về các hệ số biến đổi được lượng tử hóa dựa trên các hệ số biến đổi được lượng tử hóa trong dạng vectơ một chiều. Bộ mã hóa entrôpi 240 có thể thực hiện các phương pháp mã hóa khác nhau, ví dụ, như Golomb hàm số mũ, lập mã chiều dài biến đổi thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding, CAVLC), lập mã số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC), và tương tự. Bộ mã hóa entrôpi 240 có thể mã hóa thông tin cần thiết cho việc xây dựng lại viđeo/hình ảnh cùng với hoặc tách biệt với các hệ số biến đổi được lượng tử hóa (ví dụ, các giá trị của các phần tử cú pháp và tương tự). Thông tin được mã hóa (ví dụ, thông tin viđeo/hình ảnh được mã hóa) có thể được truyền hoặc được lưu trữ trong đơn vị của lớp trừu tượng mạng (network abstraction layer, NAL) dưới dạng luồng bit. Thông tin viđeo/hình ảnh có thể còn gồm thông tin về các tập thông số khác nhau, như tập thông số thích ứng (adaptation parameter set, APS), tập thông số ảnh (picture parameter set, PPS), tập thông số trình tự

(sequence parameter set, SPS), hoặc tập thông số viđeo (video parameter set, VPS). Bên cạnh đó, thông tin viđeo/hình ảnh có thể còn gồm thông tin ràng buộc chung. Theo sáng chế, thông tin và/hoặc các phần tử cú pháp được báo hiệu/được truyền được mô tả sau có thể được mã hóa qua quy trình mã hóa được mô tả ở trên, và được gồm trong luồng bit. Luồng bit có thể được truyền thông qua mạng, hoặc có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Ở đây, mạng có thể gồm mạng phát rộng và/hoặc mạng truyền thông, và phương tiện lưu trữ kỹ thuật số có thể gồm các phương tiện lưu trữ khác nhau, như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và tương tự. Bộ truyền (không được minh họa) truyền tín hiệu được xuất ra từ bộ mã hóa entrôpi 240 và/hoặc đơn vị lưu trữ (không được minh họa) lưu trữ tín hiệu có thể được tạo cấu hình dưới dạng thành phần trong/ngoài của thiết bị mã hóa 200, và theo cách thay thế, bộ truyền có thể được gồm trong bộ mã hóa entrôpi 240.

Các hệ số biến đổi được lượng tử hóa mà được xuất ra từ bộ lượng tử hóa 233 có thể được sử dụng để tạo ra tín hiệu dự đoán. Ví dụ, tín hiệu dư (khối phần dư hoặc các mẫu phần dư) có thể được xây dựng lại bằng cách áp dụng việc khử lượng tử hóa và biến đổi ngược cho các hệ số biến đổi được lượng tử hóa qua bộ khử lượng tử hóa 234 và bộ biến đổi ngược 235. Bộ cộng 250 cộng tín hiệu phần dư được xây dựng lại vào tín hiệu dự đoán được xuất ra từ bộ dự đoán 220 để tạo ra tín hiệu được xây dựng lại (ảnh được xây dựng lại, khối được xây dựng lại, các mẫu được xây dựng lại, hoặc mảng mẫu được xây dựng lại). Nếu không có phần dư cho khối mục tiêu xử lý, chẳng hạn như trường hợp mà chế độ bỏ qua được áp dụng, khối được dự đoán có thể được sử dụng làm khối được xây dựng lại. Tín hiệu được xây dựng lại được tạo ra có thể được sử dụng cho việc dự đoán nội của khối mục tiêu xử lý tiếp theo trong ảnh hiện tại, và có thể được sử dụng cho việc dự đoán liên của ảnh tiếp theo qua việc lọc như được mô tả dưới đây.

Trong khi đó, việc ánh xạ độ chói với việc định cỡ sắc độ (luma mapping with chroma scaling, LMCS) có thể được áp dụng trong suốt quy trình mã hóa và/hoặc xây dựng lại ảnh.

Bộ lọc 260 có thể cải thiện chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 260 có thể tạo ra ảnh được xây dựng lại mà được sửa đổi bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được xây dựng lại và lưu trữ ảnh được xây dựng lại mà sửa được đổi trong bộ nhớ

270, cụ thể là, trong DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch chuyển thích ứng mẫu (SAO), bộ lọc vòng lặp thích ứng, bộ lọc song phương, và tương tự. Bộ lọc 260 có thể tạo ra các loại thông tin khác nhau liên quan tới việc lọc và chuyển thông tin được tạo ra đến bộ mã hóa entrôpi 290 như sẽ được mô tả sau trong phần mô tả của mỗi phương pháp lọc. Thông tin liên quan đến việc lọc có thể được mã hóa bởi bộ mã hóa entrôpi 290 và được xuất ra trong dạng luồng bit.

Ảnh được xây dựng lại mà sửa được đổi sẽ được truyền đến bộ nhớ 270 có thể được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên 221. Khi việc dự đoán liên được áp dụng thông qua thiết bị mã hóa, thì sự không khớp của việc dự đoán giữa thiết bị mã hóa 200 và thiết bị giải mã có thể được tránh được và hiệu quả mã hóa có thể được cải thiện.

DPB của bộ nhớ 270 có thể lưu trữ ảnh được xây dựng lại được cải biến cho việc sử dụng làm ảnh tham chiếu trong bộ dự đoán liên 221. Bộ nhớ 270 có thể lưu trữ thông tin chuyển động của khối mà thông tin chuyển động trong ảnh hiện tại được dẫn xuất (hoặc được mã hóa) từ đó và/hoặc thông tin chuyển động của các khối trong ảnh, đã được xây dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ dự đoán liên 221 để được sử dụng như là thông tin chuyển động của khối lân cận theo không gian hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ nhớ 270 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong ảnh hiện tại và có thể truyền các mẫu được xây dựng lại đến bộ dự đoán nội 222.

Fig.3 là sơ đồ giải thích sơ lược cấu hình của thiết bị giải mã viđeo/hình ảnh mà cho nó phần bộc lộ của sáng chế có thể được áp dụng.

Tham khảo đến Fig.3, thiết bị giải mã 300 có thể gồm và được tạo cấu hình với bộ giải mã entrôpi 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, bộ lọc 350, và bộ nhớ 360. Bộ dự đoán 330 có thể gồm bộ dự đoán liên 331 và bộ dự đoán nội 332. Bộ xử lý phần dư 320 có thể gồm bộ khử lượng tử hóa 321 và bộ biến đổi ngược 322. Bộ giải mã entrôpi 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, mà đã được mô tả ở trên, có thể được tạo cấu hình bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip bộ giải mã hoặc các bộ xử lý) theo một phương án. Ngoài ra, bộ nhớ 360 có thể gồm bộ đệm ảnh được giải mã (DPB), và có thể được tạo cấu hình bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 360

như là thành phần trong/ngoài.

Khi luồng bit gồm thông tin viđeo/hình ảnh được nhập, thiết bị giải mã 300 có thể xây dựng lại hình ảnh tương ứng với quy trình trong đó thông tin viđeo/hình ảnh được xử lý trong thiết bị mã hóa được minh họa trên Fig.2. Ví dụ, thiết bị giải mã 300 có thể dẫn xuất các đơn vị/các khối dựa trên thông tin liên quan đến việc phân tách khối thu nhận được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã sử dụng đơn vị xử lý được áp dụng cho thiết bị mã hóa. Do đó, đơn vị xử lý để giải mã có thể là, ví dụ, đơn vị lập mã, và đơn vị lập mã có thể được chia tách theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây lập mã hoặc đơn vị lập mã tối đa. Một hoặc nhiều đơn vị biến đổi có thể được dẫn xuất từ đơn vị lập mã. Ngoài ra, tín hiệu hình ảnh được xây dựng lại được giải mã và được xuất ra thông qua thiết bị giải mã 300 có thể được tái tạo thông qua thiết bị tái tạo.

Thiết bị giải mã 300 có thể nhận tín hiệu được xuất ra từ thiết bị mã hóa trên Fig.2 trong dạng luồng bit, và tín hiệu nhận được có thể được giải mã thông qua bộ giải mã entrôpi 310. Ví dụ, bộ giải mã entrôpi 310 có thể phân tích cú pháp luồng bit để dẫn xuất thông tin (ví dụ, thông tin viđeo/hình ảnh) cần thiết cho việc xây dựng lại hình ảnh (hoặc xây dựng lại ảnh). Thông tin viđeo/hình ảnh có thể còn gồm thông tin về các tập thông số khác nhau như tập thông số thích ứng (APS), tập thông số ảnh (PPS), tập thông số trình tự (SPS), hoặc tập thông số viđeo (VPS). Bên cạnh đó, thông tin viđeo/hình ảnh có thể còn gồm thông tin ràng buộc chung. Thiết bị giải mã còn có thể giải mã ảnh dựa trên thông tin về tập thông số và/hoặc thông tin ràng buộc chung. Thông tin được báo hiệu/nhận được và/hoặc các phần tử cú pháp được mô tả sau trong sáng chế có thể được giải mã có thể giải mã quy trình giải mã và thu nhận được từ luồng bit. Ví dụ, bộ giải mã entrôpi 310 giải mã thông tin trong luồng bit dựa trên phương pháp lập mã như là lập mã Golomb hàm mũ, CAVLC, hoặc CABAC, và xuất ra ra các phần tử cú pháp cần thiết cho việc xây dựng lại hình ảnh và các giá trị được lượng tử hóa của các hệ số biến đổi cho phần dư. Cụ thể hơn là, phương pháp giải mã entrôpi CABAC có thể nhận ngăn tương ứng với từng phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh nhờ sử dụng thông tin phần tử cú pháp mục tiêu giải mã, thông tin giải mã của khối mục tiêu giải mã hoặc thông tin về ký hiệu/ngăn được giải mã trong giai đoạn trước đó, và thực hiện việc giải mã số học trên ngăn bằng cách dự đoán khả năng xuất hiện của ngăn theo

mô hình ngữ cảnh được xác định, và tạo ra ký hiệu tương ứng với giá trị của mỗi phần tử cú pháp. Trong trường hợp này, phương pháp giải mã entrôpi CABAC có thể cập nhật mô hình ngữ cảnh bằng cách sử dụng thông tin của ký hiệu/ngăn được giải mã cho mô hình ngữ cảnh của ký hiệu/ngăn tiếp theo sau khi xác định mô hình ngữ cảnh. Thông tin liên quan đến việc dự đoán trong số thông tin được giải mã bởi bộ giải mã entrôpi 310 có thể được cung cấp cho bộ dự đoán 330, và thông tin về phần dư trên đó việc giải mã entrôpi đã được thực hiện trong bộ giải mã entrôpi 310, nghĩa là, các hệ số biến đổi được lượng tử hóa và thông tin thông số liên quan, có thể được nhập vào bộ khử lượng tử hóa 321. Bên cạnh đó, thông tin về việc lọc trong số thông tin được giải mã bởi bộ giải mã entrôpi 310 có thể được cung cấp cho bộ lọc 350. Trong lúc đó, bộ nhận (không được minh họa) để nhận tín hiệu được xuất ra từ thiết bị mã hóa có thể còn được tạo cấu hình làm phần tử trong/ngoài của thiết bị giải mã 300, hoặc bộ nhận có thể là phần tử cấu thành của bộ giải mã entrôpi 310. Trong lúc đó, thiết bị giải mã theo sáng chế có thể được gọi là thiết bị giải mã viđeo/hình ảnh/ảnh, và thiết bị giải mã có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin viđeo/hình ảnh/ảnh) và bộ giải mã mẫu (bộ giải mã mẫu viđeo/hình ảnh/ảnh). Bộ giải mã thông tin có thể gồm bộ giải mã entrôpi 310, và bộ giải mã mẫu có thể gồm ít nhất một trong số bộ khử lượng tử hóa 321, bộ biến đổi ngược 322, bộ dự đoán 330, bộ cộng 340, bộ lọc 350, và bộ nhớ 360.

Bộ khử lượng tử hóa 321 có thể khử lượng tử hóa các hệ số biến đổi được lượng tử hóa để xuất ra các hệ số biến đổi. Bộ khử lượng tử hóa 321 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa trong dạng khối hai chiều. Trong trường hợp này, việc sắp xếp lại có thể được thực hiện dựa trên thứ tự quét hệ số được thực hiện bởi thiết bị mã hóa. Bộ khử lượng tử hóa 321 có thể thực hiện việc khử lượng tử hóa cho các hệ số biến đổi được lượng tử hóa sử dụng thông số lượng tử hóa (ví dụ, thông tin kích thước bước lượng tử hóa), và thu nhận các hệ số biến đổi.

Bộ biến đổi ngược 322 biến đổi ngược các hệ số biến đổi để thu nhận tín hiệu phần dư (khối phần dư, mảng mẫu phần dư).

Bộ dự đoán 330 có thể thực hiện việc dự đoán của khối hiện tại, và tạo ra khối được dự đoán gồm các mẫu dự đoán của khối hiện tại. Bộ dự đoán có thể xác định xem việc dự đoán nội được áp dụng hay việc dự đoán liên được áp dụng cho khối hiện tại dựa trên thông tin về việc dự đoán được xuất từ bộ giải mã entrôpi 310, và xác định chế

độ dự đoán nội/liên cụ thể.

Bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau được mô tả dưới đây. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc dự đoán nội hoặc dự đoán liên để dự đoán một khối mà còn có thể áp dụng một cách đồng thời việc dự đoán nội và dự đoán liên. Việc này có thể được gọi là việc dự đoán nội và liên được kết hợp (CIIP). Ngoài ra, bộ dự đoán có thể thực hiện sao chép khối nội (IBC) cho việc dự đoán của khối. Sao chép khối nội có thể được sử dụng cho việc lập mã hình ảnh động/hình ảnh nội dung của trò chơi hoặc tương tự, ví dụ, lập mã nội dung màn (SCC). IBC về cơ bản thực hiện việc dự đoán trong ảnh hiện tại mà có thể được thực hiện tương tự như việc dự đoán liên mà trong đó khối tham chiếu được dẫn xuất trong ảnh hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật dự đoán liên được mô tả trong sáng chế.

Bộ dự đoán nội 332 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt trong vùng lân cận của khối hiện tại, hoặc có thể được đặt cách xa khỏi khối hiện tại theo chế độ dự đoán. Trong việc dự đoán nội, các chế độ dự đoán có thể gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Bộ dự đoán nội 332 có thể xác định chế độ dự đoán cần được áp dụng cho khối hiện tại bằng việc sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên 331 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi véctơ chuyển động trên ảnh tham chiếu. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thì thông tin chuyển động có thể được dự đoán trong đơn vị của các khối, các khối con, hoặc các mẫu dựa trên sự tương liên của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm vectơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin về hướng dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, và tương tự). Trong trường hợp dự đoán liên, khối lân cận có thể gồm khối lân cận theo không gian tồn tại trong ảnh hiện tại và khối lân cận theo thời gian có mặt trong ảnh tham chiếu. Ví dụ, bộ dự đoán liên 331 có thể xây dựng danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và dẫn xuất vectơ chuyển động của khối hiện tại và/hoặc chỉ

số ảnh tham chiếu dựa trên thông tin chọn ứng viên nhận được. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể gồm thông tin chỉ ra chế độ dự đoán liên cho khối hiện tại.

Bộ cộng 340 có thể tạo ra tín hiệu được xây dựng lại (ảnh được xây dựng lại, khối được xây dựng lại, hoặc mảng mẫu được xây dựng lại) bằng cách cộng tín hiệu dư thu nhận được vào tín hiệu dự đoán (khối được dự đoán hoặc mảng mẫu được dự đoán) được xuất ra từ bộ dự đoán 330. Nếu không có phần dư cho khối mục tiêu xử lý, chẳng hạn như trường hợp mà chế độ bỏ qua được áp dụng, khối được dự đoán có thể được sử dụng làm khối được xây dựng lại.

Bộ cộng 340 có thể được gọi là bộ xây dựng lại hoặc bộ tạo khối được xây dựng lại. Tín hiệu được xây dựng lại được tạo ra có thể được sử dụng cho việc dự đoán nội của khối tiếp theo cần được xử lý trong ảnh hiện tại, và như được mô tả sau đây, có thể cũng được xuất ra qua việc lọc hoặc cũng có thể được sử dụng cho việc dự đoán liên của ảnh tiếp theo.

Trong khi đó, việc ánh xạ độ chói với việc định cỡ sắc độ (LMCS) cũng có thể được áp dụng trong quy trình giải mã ảnh.

Bộ lọc 350 có thể cải thiện chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 350 có thể tạo ra ảnh được xây dựng lại mà được sửa đổi bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được xây dựng lại và lưu trữ ảnh được xây dựng lại mà sửa được đổi trong bộ nhớ 360, cụ thể là, trong DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch chuyển thích ứng mẫu, bộ lọc vòng lặp thích ứng, bộ lọc song phương, và tương tự.

Ảnh được xây dựng lại (được sửa đổi) mà được lưu trữ trong DPB của bộ nhớ 360 có thể được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên 331. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối mà thông tin chuyển động trong ảnh hiện tại được dẫn xuất (hoặc được giải mã) từ đó và/hoặc thông tin chuyển động của các khối trong ảnh hiện đã được xây dựng lại. Thông tin chuyển động được lưu trữ có thể được chuyển đến bộ dự đoán liên 331 để được tận dụng làm thông tin chuyển động của khối lân cận theo không gian hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ

nhớ 360 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong ảnh hiện tại và truyền các mẫu được xây dựng lại đến bộ dự đoán nội 332.

Trong bản mô tả này, các phương án được mô tả trong bộ dự đoán 330, bộ khử lượng tử hóa 321, bộ biến đổi ngược 322, và bộ lọc 350 của thiết bị giải mã 300 cũng có thể được áp dụng theo cùng cách hoặc tương ứng với bộ dự đoán 220, bộ khử lượng tử hóa 234, bộ biến đổi ngược 235, và bộ lọc 260 của thiết bị mã hóa 200.

Trong khi đó, như được mô tả ở trên, trong khi thực hiện lập mã viđeo, việc dự đoán được thực hiện để cải thiện hiệu quả nén. Thông qua việc này, khối được dự đoán gồm các mẫu dự đoán cho khối hiện tại như khối cần được lập mã (tức là, khối mục tiêu lập mã) có thể được tạo ra. Ở đây, khối được dự đoán gồm các mẫu dự đoán trong miền không gian (hoặc miền điểm ảnh). Khối được dự đoán được dẫn xuất theo cách giống như trong thiết bị mã hóa và thiết bị giải mã, và thiết bị mã hóa có thể báo hiệu thông tin (thông tin phần dư) về phần dư giữa khối gốc và khối được dự đoán, thay vì giá trị mẫu gốc của khối gốc, đến thiết bị giải mã, nhờ đó làm tăng hiệu quả lập mã hình ảnh. Thiết bị giải mã có thể dẫn xuất khối phần dư gồm các mẫu phần dư dựa trên thông tin phần dư, bổ sung khối phần dư và khối được dự đoán để tạo ra các khối được xây dựng lại gồm các mẫu được xây dựng lại, và tạo ra ảnh được xây dựng lại gồm các khối được xây dựng lại.

Thông tin phần dư có thể được tạo ra thông qua quy trình biến đổi và lượng tử hóa. Ví dụ, thiết bị mã hóa có thể dẫn xuất khối phần dư giữa khối gốc và khối được dự đoán, thực hiện quy trình biến đổi trên các mẫu phần dư (mảng mẫu phần dư) được gồm trong khối phần dư để dẫn xuất các hệ số biến đổi, thực hiện quy trình lượng tử hóa trên các hệ số biến đổi để dẫn xuất các hệ số biến đổi được lượng tử hóa, và báo hiệu thông tin phần dư liên quan đến thiết bị giải mã (thông qua luồng bit). Ở đây, thông tin phần dư có thể gồm thông tin giá trị của các hệ số biến đổi được lượng tử hóa, thông tin vị trí, kỹ thuật biến đổi, nhân biến đổi, thông số lượng tử hóa, và tương tự. Thiết bị giải mã có thể thực hiện quy trình khử lượng tử hóa/biến đổi ngược dựa trên thông tin phần dư và dẫn xuất các mẫu phần dư (hoặc các khối phần dư). Thiết bị giải mã có thể tạo ra ảnh được xây dựng lại dựa trên khối được dự đoán và khối phần dư. Ngoài ra, đối với sự tham chiếu cho việc dự đoán liên của ảnh sau đó, thiết bị mã hóa cũng có thể khử lượng tử hóa/biến đổi ngược các hệ số biến đổi được lượng tử hóa để dẫn xuất khối phần dư

và tạo ra ảnh được xây dựng lại dựa trên đó.

Trong sáng chế, ít nhất một hoạt động trong số việc lượng tử hóa/việc khử lượng tử hóa và/hoặc việc biến đổi/việc biến đổi ngược có thể được lược bỏ. Khi việc lượng tử hóa/việc khử lượng tử hóa được lược bỏ, thì hệ số biến đổi được lượng tử hóa có thể được gọi là hệ số biến đổi. Khi việc biến đổi/biến đổi ngược được lược bỏ, thì hệ số biến đổi có thể được gọi là hệ số hoặc hệ số dư, hoặc có thể vẫn được gọi là hệ số biến đổi để trình bày đồng nhất.

Trong sáng chế, hệ số biến đổi được lượng tử hóa và hệ số biến đổi có thể được gọi là hệ số biến đổi và hệ số biến đổi được định cỡ, một cách tương ứng. Trong trường hợp này, thông tin phần dư có thể gồm thông tin về hệ số biến đổi (các hệ số biến đổi), và thông tin về hệ số biến đổi (các hệ số biến đổi) có thể được báo hiệu thông qua cú pháp lập mã phần dư. Các hệ số biến đổi có thể được dẫn xuất dựa trên thông tin phần dư (hoặc thông tin về hệ số biến đổi (các hệ số biến đổi)), và các hệ số biến đổi được định cỡ có thể được dẫn xuất thông qua việc biến đổi ngược (định cỡ) trên các hệ số biến đổi. Các mẫu phần dư có thể được dẫn xuất dựa trên việc biến đổi ngược (biến đổi) của các hệ số biến đổi được định cỡ. Điều này cũng có thể được áp dụng/được trình bày trong các phần khác của sáng chế.

Bộ dự đoán của thiết bị mã hóa/thiết bị giải mã có thể dẫn xuất các mẫu dự đoán bằng cách thực hiện việc dự đoán nội trong các đơn vị của các khối. Việc dự đoán liên có thể là việc dự đoán được dẫn xuất theo cách mà phụ thuộc vào các phần tử dữ liệu (ví dụ, các giá trị mẫu hoặc thông tin chuyển động v.v.) của (các) ảnh khác với ảnh hiện tại. Khi việc dự đoán liên được áp dụng cho khối hiện tại, dựa trên khối tham chiếu (các mảnh mẫu tham chiếu) được cụ thể hóa bởi vectơ chuyển động trên ảnh tham chiếu được trỏ tới bởi chỉ số ảnh tham chiếu, khối được dự đoán (các mảng mẫu dự đoán) cho khối hiện tại có thể được dẫn xuất. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động của khối hiện tại có thể được dự đoán trong các đơn vị của các khối, các khối con, hoặc các mẫu dựa trên sự tương liên giữa thông tin chuyển động giữa các khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm vectơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin loại dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, v.v.). Khi việc dự đoán liên được áp dụng, các khối lân cận có thể gồm khối

lân cận theo không gian tồn tại trong ảnh hiện tại và khối lân cận theo thời gian tồn tại trong ảnh tham chiếu. Ảnh tham chiếu gồm khối tham chiếu và ảnh tham chiếu gồm khối lân cận theo thời gian có thể là giống nhau hoặc khác nhau. Khối lân cận theo thời gian có thể được gọi là khối tham chiếu được đặt cùng một chỗ, CU được đặt cùng một chỗ (colCU), v.v., và ảnh tham chiếu gồm khối lân cận theo thời gian có thể được gọi là ảnh được đặt cùng một chỗ (colPic). Ví dụ, danh sách ứng viên thông tin chuyển động có thể được xây dựng dựa trên các khối lân cận của khối hiện tại và cờ hoặc thông tin chỉ số chỉ ra ứng viên nào được lựa chọn (được sử dụng) để dẫn xuất vectơ chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại có thể được báo hiệu. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau. Ví dụ, trong chế độ bỏ qua và chế độ hợp nhất, thông tin chuyển động của khối hiện tại có thể giống với thông tin chuyển động của khối lân cận được lựa chọn. Trong chế độ bỏ qua, không giống với chế độ hợp nhất, tín hiệu dư có thể không được truyền. Trong trường hợp chế độ dự đoán vectơ chuyển động (MVP), vectơ chuyển động của khối lân cận được lựa chọn có thể được sử dụng như bộ dự đoán vectơ chuyển động, và sự chênh lệch vectơ chuyển động có thể được báo hiệu. Trong trường hợp này, vectơ chuyển động của khối hiện tại có thể được dẫn xuất nhờ sử dụng tổng của bộ dự đoán vectơ chuyển động và sự chênh lệch vectơ chuyển động.

Thông tin chuyển động có thể gồm thông tin chuyển động L0 và/hoặc thông tin chuyển động L1 theo loại dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, v.v.). Vectơ chuyển động theo hướng L0 có thể được gọi là vectơ chuyển động L0 hoặc MVL0, và vectơ chuyển động theo hướng L1 có thể được gọi là vectơ chuyển động L1 hoặc MVL1. Việc dự đoán dựa trên vectơ chuyển động L0 có thể được gọi là việc dự đoán L0, việc dự đoán dựa trên vectơ chuyển động L1 có thể được gọi là việc dự đoán L1, và việc dự đoán dựa trên cả vectơ chuyển động L0 và vectơ chuyển động L1 có thể được gọi là việc dự đoán đôi. Ở đây, vectơ chuyển động L0 có thể chỉ ra vectơ chuyển động được kết hợp với danh sách ảnh tham chiếu L0 (L0), và vectơ chuyển động L1 có thể chỉ ra vectơ chuyển động được kết hợp với danh sách ảnh tham chiếu L1 (L1). Danh sách ảnh tham chiếu L0 có thể gồm các ảnh mà sớm hơn ảnh hiện tại theo thứ tự xuất ra như các ảnh tham chiếu, và danh sách ảnh tham chiếu L1 có thể gồm các ảnh mà muộn hơn ảnh hiện tại theo thứ tự xuất ra. Các ảnh trước đó có thể được gọi là các ảnh

(tham chiếu) chuyển tiếp, và các ảnh tiếp sau đó có thể được gọi là các ảnh (tham chiếu) ngược. Danh sách ảnh tham chiếu L0 có thể còn gồm các ảnh mà tiếp sau ảnh hiện tại trong thứ tự xuất như các ảnh tham chiếu. Trong trường hợp này, các ảnh trước đó có thể được định chỉ số đầu tiên, và các ảnh sau đó có thể được định chỉ số tiếp theo trong danh sách ảnh tham chiếu L0. Danh sách ảnh tham chiếu L1 có thể còn gồm các ảnh phía trước ảnh hiện tại trong thứ tự xuất như các ảnh tham chiếu. Trong trường hợp này, các ảnh tiếp sau đó có thể được định chỉ số trước tiên trong danh sách ảnh tham chiếu L1 và các ảnh trước đó có thể được định chỉ số tiếp theo. Ở đây, thứ tự xuất ra có thể tương ứng với thứ tự số đếm thứ tự ảnh (Picture Order Count, POC).

Fig.4 thể hiện ví dụ cấu trúc phân cấp cho hình ảnh/viđeo được lập mã.

Tham khảo đến Fig.4, hình ảnh/viđeo được lập mã được chia thành lớp lập mã viđeo (video coding layer, VCL) mà xử lý quy trình giải mã hình ảnh/viđeo và chính nó, hệ thống phụ mà truyền và lưu trữ thông tin được lập mã, và lớp trừu tượng mạng (NAL) mà tồn tại giữa VCL và các hệ thống phụ và chịu trách nhiệm cho các chức năng thích ứng mạng.

VCL có thể tạo ra dữ liệu VCL gồm dữ liệu hình ảnh được nén (dữ liệu lát), hoặc tạo ra các tập thông số gồm tập thông số ảnh (Picture Parameter Set: PPS), tập thông số trình tự (Sequence Parameter Set: SPS), tập thông số viđeo (Video Parameter Set: VPS) v.v. hoặc tin nhắn thông tin tăng cường bổ sung (supplemental enhancement information, SEI) cần thiết bổ sung cho quy trình giải mã của hình ảnh.

Trong NAL, đơn vị NAL có thể được tạo ra bằng việc bổ sung thông tin phần đầu (phần đầu đơn vị NAL) vào phụ tải trình tự bit thô (raw byte sequence payload, RBSP) được tạo ra trong VCL. Trong trường hợp này, RBSP tham chiếu đến dữ liệu lát, các tập thông số, các tin nhắn SEI, v.v., được tạo ra trong VCL. Phần đầu đơn vị NAL có thể gồm thông tin loại đơn vị NAL được chỉ rõ theo dữ liệu RBSP được gồm trong đơn vị NAL tương ứng.

Như được thể hiện trên hình vẽ, đơn vị NAL có thể được chia thành đơn vị NAL VCL và đơn vị NAL không VCL theo RBSP được tạo ra trong VCL. Đơn vị NAL VCL có thể nghĩa là đơn vị NAL gồm thông tin (dữ liệu được phân lát) về hình ảnh, và đơn vị NAL không VCL có thể nghĩa là đơn vị NAL chứa thông tin (tập thông số hoặc tin

nhắn SEI) cần thiết để giải mã hình ảnh.

Đơn vị NAL VCL và đơn vị NAL không VCL được đề cập ở trên có thể được truyền qua mạng bằng việc gắn thông tin phần đầu theo tiêu chuẩn dữ liệu của hệ thống con. Ví dụ, đơn vị NAL có thể được biến đổi thành dạng dữ liệu của tiêu chuẩn được xác định trước chẳng hạn như định dạng tệp H.266/VVC, giao thức vận chuyển theo thời gian thực (Real-time Transport Protocol, RTP), luồng vận chuyển (Transport Stream, TS), v.v. và được truyền qua các mạng khác nhau.

Như được mô tả ở trên, trong đơn vị NAL, loại đơn vị NAL có thể được cụ thể hóa theo cấu trúc dữ liệu RBSP được gồm trong đơn vị NAL tương ứng, và thông tin về loại đơn vị NAL này có thể được lưu trữ và được báo hiệu trong phần đầu đơn vị NAL.

Ví dụ, đơn vị NAL có thể được phân loại sơ lược thành loại đơn vị NAL VCL và loại đơn vị NAL không VCL phụ thuộc vào việc liệu đơn vị NAL có gồm thông tin về hình ảnh (dữ liệu lát) hay không. Loại đơn vị NAL VCL có thể được phân loại theo đặc tính và loại của ảnh được gồm trong đơn vị NAL VCL, và loại đơn vị NAL không VCL có thể được phân loại theo loại của tập thông số.

Phần sau đây là ví dụ của loại đơn vị NAL được chỉ rõ theo loại của tập thông số được gồm trong loại đơn vị NAL không VCL.

- Đơn vị NAL tập thông số thích ứng (APS, Adaptation Parameter Set): Loại đối với đơn vị NAL gồm APS

- Đơn vị NAL tập thông số giải mã (DPS, Decoding Parameter Set): Loại đối với đơn vị NAL gồm DPS

- Đơn vị NAL tập thông số viđeo (VPS, Video Parameter Set): Loại đối với đơn vị NAL gồm VPS

- Đơn vị NAL tập thông số trình tự (SPS, Sequence Parameter Set): Loại đối với đơn vị NAL gồm SPS

- Đơn vị NAL tập thông số ảnh (PPS, Picture Parameter Set): Loại đối với đơn vị NAL gồm PPS

- Đơn vị NAL phần đầu ảnh (PH, Picture header): Loại đối với đơn vị NAL gồm PH

Các loại đơn vị NAL được mô tả ở trên có thông tin cú pháp cho loại đơn vị NAL, và thông tin cú pháp có thể được lưu trữ và được báo hiệu trong phần đầu đơn vị NAL. Ví dụ, thông tin cú pháp có thể là nal_unit_type, và các loại đơn vị NAL có thể được chỉ rõ bởi giá trị nal_unit_type.

Trong lúc đó, như được mô tả ở trên, một ảnh có thể gồm nhiều lát, và một lát có thể gồm phần đầu lát và dữ liệu lát. Trong trường hợp này, một phần đầu ảnh có thể còn được bổ sung cho nhiều lát (phần đầu lát và tập dữ liệu lát) trong một ảnh. Phần đầu ảnh (cú pháp phần đầu ảnh) có thể gồm thông tin/các thông số có thể áp dụng chung cho ảnh. Trong sáng chế, lát có thể được trộn lẫn hoặc được thay thế với nhóm phiến. Ngoài ra, trong sáng chế, phần đầu lát có thể được trộn lẫn hoặc được thay thế với phần đầu nhóm loại.

Phần đầu lát (cú pháp phần đầu lát hoặc thông tin phần đầu lát) có thể gồm thông tin/các thông số thường có thể áp dụng cho lát. APS (cú pháp APS) hoặc PPS (cú pháp PPS) có thể gồm thông tin/các thông số thường có thể áp dụng cho một hoặc nhiều lát hoặc ảnh. SPS (cú pháp SPS) có thể gồm thông tin/các thông số thường có thể áp dụng cho một hoặc nhiều trình tự. VPS (cú pháp VPS) có thể gồm thông tin/các thông số thường có thể áp dụng cho nhiều lớp. DPS (cú pháp DPS) có thể gồm thông tin/các thông số thường có thể áp dụng cho toàn bộ viđeo. DPS có thể gồm thông tin/các thông số liên quan tới sự móc nối của chuỗi viđeo được lập mã (coded video sequence, CVS). Trong tài liệu này, cú pháp mức cao (high level syntax, HLS) có thể gồm ít nhất một trong số cú pháp APS, cú pháp PPS, cú pháp SPS, cú pháp VPS, cú pháp DPS, cú pháp phần đầu ảnh, và cú pháp phần đầu lát.

Trong tài liệu này, thông tin hình ảnh/viđeo được mã hóa trong thiết bị mã hóa và được báo hiệu dưới dạng luồng bit tới thiết bị giải mã có thể gồm, cũng như thông tin liên quan đến việc phân vùng ảnh trong ảnh, thông tin dự đoán nội/liên, thông tin phần dư, thông tin lọc trong vòng lặp, v.v. thông tin được gồm trong phần đầu lát, thông tin được gồm trong phần đầu ảnh, thông tin được gồm trong APS, thông tin được gồm trong PPS, thông tin được gồm trong SPS, thông tin được gồm trong VPS, và/hoặc thông tin được gồm trong DPS. Ngoài ra, thông tin hình ảnh/viđeo có thể còn gồm thông tin của phần đầu đơn vị NAL.

Trong lúc đó, để bù sự chênh lệch giữa ảnh gốc và hình ảnh được xây dựng lại

do lỗi xảy ra trong quy trình mã hóa nén như là lượng tử hóa, quy trình lọc trong vòng lặp có thể được thực hiện trên các mẫu được xây dựng lại hoặc các ảnh được xây dựng lại như được mô tả ở trên. Như được mô tả ở trên, việc lọc trong vòng lặp có thể được thực hiện bởi bộ lọc của thiết bị mã hóa và bộ lọc của thiết bị giải mã, và bộ lọc giải khối, SAO, và/hoặc bộ lọc vòng lặp thích ứng (adaptive loop filter, ALF) có thể được áp dụng. Ví dụ, quy trình ALF có thể được thực hiện sau quy trình lọc giải khối và/hoặc quy trình SAO được hoàn tất. Tuy nhiên, ngay cả trong trường hợp này, quy trình lọc giải khối và/hoặc quy trình SAO có thể được bỏ qua.

Sau đây, việc xây dựng lại ảnh và việc lọc sẽ được mô tả chi tiết. Trong lập mã hình ảnh/viđeo, khối được xây dựng lại có thể được tạo ra dựa trên việc dự đoán nội/việc dự đoán liên trong mỗi đơn vị khối, và ảnh được xây dựng lại gồm các khối được xây dựng lại có thể được tạo ra. Khi ảnh/lát hiện tại à ảnh/lát I, các khối được gồm trong ảnh/lát hiện tại có thể được xây dựng lại dựa trên chỉ việc dự đoán nội. Cùng lúc đó, khi ảnh/lát hiện tại à ảnh/lát P hoặc B, các khối được gồm trong ảnh/lát hiện tại có thể được xây dựng lại dựa trên việc dự đoán nội hoặc việc dự đoán liên. Trong trường hợp này, việc dự đoán nội có thể được áp dụng cho một số khối trong ảnh/lát hiện tại, và việc dự đoán liên có thể được áp dụng cho các khối còn lại.

Dự đoán nội có thể thể hiện việc dự đoán để tạo ra các mẫu dự đoán cho khối hiện tại dựa trên các mẫu tham chiếu trong ảnh (sau đây, ảnh hiện tại) mà khối hiện tại thuộc về. Trong trường hợp mà việc dự đoán nội được áp dụng cho khối hiện tại, thì các mẫu tham chiếu lân cận cần để được sử dụng cho việc dự đoán nội của khối hiện tại có thể được dẫn xuất. Các mẫu tham chiếu lân cận của khối hiện tại có thể gồm mẫu liền kề với biên trái của khối hiện tại có kích thước là nW×nH, tổng cộng 2×nH mẫu lân cận phần dưới cùng bên trái, mẫu liền kề với biên trên cùng của khối hiện tại, tổng cộng 2×nW mẫu lân cận phần trên cùng bên phải, và một mẫu lân cận phần trên cùng bên trái của khối hiện tại. Mặt khác, các mẫu tham chiếu lân cận của khối hiện tại có thể gồm mẫu lân cận trên cùng của nhiều cột và mẫu lân cận bên trái của nhiều hàng. Mặt khác, các mẫu tham chiếu lân cận của khối hiện tại có thể gồm tổng cộng nH mẫu liền kề với biên bên phải của khối hiện tại có kích thước là nW×nH, tổng cộng nH mẫu liền kề với biên bên phải của khối hiện tại, tổng cộng nW mẫu liền kề với biên dưới cùng của khối hiện tại, và một mẫu lân cận phần dưới cùng bên phải của khối hiện tại.

Tuy nhiên, một số mẫu tham chiếu lân cận của khối hiện tại có thể vẫn chưa được giải mã hoặc có thể không có sẵn. Trong trường hợp này, bộ giải mã có thể tạo cấu hình các mẫu tham chiếu lân cận cần được sử dụng cho việc dự đoán thông qua việc thay thế các mẫu sẵn có cho các mẫu không sẵn có. Theo cách thay thế, các mẫu tham chiếu lân cận cần được sử dụng để dự đoán có thể được tạo cấu hình thông qua phép nội suy của các mẫu khả dụng.

Khi các mẫu tham chiếu lân cận được dẫn xuất, có hai trường hợp, mà là, trường hợp (i) trong đó mẫu dự đoán có thể được dẫn xuất dựa trên trung bình hoặc phép nội suy của các mẫu tham chiếu lân cận của khối hiện tại, và trường hợp (ii) trong đó mẫu dự đoán có thể được dẫn xuất dựa trên mẫu tham chiếu có mặt theo hướng (dự đoán) cụ thể cho mẫu dự đoán trong số các mẫu tham chiếu lân cận của khối hiện tại. Trường hợp (i) có thể được gọi là chế độ không có hướng hoặc chế độ không có góc và trường hợp (ii) có thể được gọi là chế độ có hướng hoặc chế độ có góc. Ngoài ra, mẫu dự đoán cũng có thể được tạo ra thông qua phép nội suy giữa mẫu lân cận thứ nhất và mẫu lân cận thứ hai được đặt theo hướng ngược lại với hướng dự đoán của chế độ dự đoán nội của khối hiện tại dựa trên mẫu dự đoán của khối hiện tại trong số các mẫu tham chiếu lân cận. Trường hợp ở trên có thể được gọi là việc dự đoán nội nội suy tuyến tính (Linear Interpolation Intra Prediction, LIP). Bên cạnh đó, các mẫu dự đoán sắc độ cũng có thể được tạo ra dựa trên các mẫu độ chói nhờ sử dụng mô hình tuyến tính. Trường hợp này có thể được gọi là chế độ LM. Ngoài ra, mẫu dự đoán tạm thời của khối hiện tại có thể được dẫn xuất dựa trên các mẫu tham chiếu lân cận được lọc. Ít nhất một mẫu tham chiếu, mà được dẫn xuất theo chế độ dự đoán nội trong số các mẫu tham chiếu lân cận sẵn có, nghĩa là, các mẫu tham chiếu lân cận không được lọc, và mẫu dự đoán tạm thời có thể được lấy tổng đặt trọng số để dẫn xuất mẫu dự đoán của khối hiện tại. Trường hợp ở trên có thể được gọi là việc dự đoán nội phụ thuộc sự định vị (Position Dependent Intra Prediction, PDPC). Bên cạnh đó, đường mẫu tham chiếu có độ chính xác dự đoán cao nhất trong số các đường mẫu nhiều tham chiếu lân cận của khối hiện tại có thể được lựa chọn để dẫn xuất mẫu dự đoán bằng cách sử dụng mẫu tham chiếu được đặt theo hướng dự đoán trên đường tương ứng, và đường mẫu tham chiếu được sử dụng tại đây có thể được chỉ ra (được báo hiệu) cho thiết bị giải mã, nhờ đó thực hiện việc mã hóa dự đoán nội. Trường hợp ở trên có thể được gọi là việc dự đoán nội đường nhiều tham

chiếu (Multi-Reference Line, MRL) hoặc việc dự đoán nội dựa trên MRL. Bên cạnh đó, việc dự đoán nội có thể được thực hiện dựa trên cùng một chế độ dự đoán nội bằng cách phân chia khối hiện tại thành các phân vùng con dọc hoặc ngang, và các mẫu tham chiếu lân cận có thể được dẫn xuất và được sử dụng trong các đơn vị phân vùng con. Nghĩa là, trong trường hợp này, chế độ dự đoán nội cho khối hiện tại được áp dụng ngang bằng cho các phân vùng con, và hiệu quả dự đoán nội có thể được cải thiện trong một số trường hợp bằng việc dẫn xuất và sử dụng các mẫu tham chiếu lân cận trong các đơn vị là các phân vùng con. Phương pháp dự đoán như vậy có thể được gọi là các phân vùng con trong ảnh (Intra Sub-Partition, ISP) hoặc việc dự đoán nội dựa trên ISP. Các phương pháp dự đoán nội được đề cập ở trên có thể được gọi là loại dự đoán nội riêng biệt với chế độ dự đoán nội trong các phần 1.2. Loại dự đoán nội có thể được gọi theo các thuật ngữ khác nhau như là kỹ thuật dự đoán nội hoặc chế độ dự đoán nội bổ sung hoặc tương tự. Ví dụ, loại dự đoán nội (hoặc chế độ dự đoán nội bổ sung hoặc tương tự) có thể gồm ít nhất một trong số LIP, PDPC, MRL, và ISP được đề cập ở trên. Phương pháp dự đoán nội chung ngoại trừ loại dự đoán nội cụ thể như là LIP, PDPC, MRL, hoặc ISP có thể được gọi là loại dự đoán nội thông thường. Loại dự đoán nội thông thường có thể được áp dụng chung khi loại dự đoán nội cụ thể không được áp dụng, và việc dự đoán có thể được thực hiện dựa trên chế độ dự đoán nội được đề cập ở trên. Trong khi đó, một cách tùy chọn, việc lọc sau xử lý có thể được thực hiện trên mẫu dự đoán được dẫn xuất.

Cụ thể là, quy trình dự đoán nội có thể gồm thao tác xác định chế độ/loại dự đoán nội, thao tác dẫn xuất mẫu tham chiếu lân cận, và thao tác dẫn xuất mẫu dự đoán dựa trên chế độ/loại dự đoán nội. Ngoài ra, một cách tùy chọn, thao tác lọc sau xử lý có thể được thực hiện trên mẫu dự đoán được dẫn xuất.

Ảnh được xây dựng lại được cải biến có thể được tạo ra thông qua quy trình lọc trong vòng lặp, và ảnh được xây dựng lại được cải biến có thể được xuất ra dưới dạng ảnh được giải mã trong thiết bị giải mã và cũng có thể được lưu trữ trong bộ nhớ hoặc bộ đệm ảnh được giải mã của thiết bị mã hóa/thiết bị giải mã và được sử dụng như ảnh tham chiếu trong quy trình dự đoán liên khi ảnh được mã hóa/được giải mã ở thời điểm sau đó. Quy trình lọc trong vòng lặp có thể gồm quy trình lọc giải khối, quy trình dịch chuyển thích ứng mẫu (SAO), và/hoặc quy trình bộ lọc vòng lặp thích ứng (ALF) như được mô tả ở trên. Trong trường hợp này, một hoặc một số trong số quy trình lọc giải

khối, quy trình SAO, quy trình ALF, và quy trình lọc song phương có thể được áp dụng lần lượt, hoặc tất cả các quy trình có thể được áp dụng lần lượt. Ví dụ, quy trình SAO có thể được thực hiện sau khi quy trình lọc giải khối được áp dụng cho ảnh được xây dựng lại. Mặt khác, ví dụ, quy trình ALF có thể được thực hiện sau khi quy trình lọc giải khối được áp dụng cho ảnh được xây dựng lại. Việc này cũng có thể được thực hiện ngang bằng trong thiết bị mã hóa.

Việc lọc giải khối là kỹ thuật lọc mà loại bỏ sự biến dạng xảy ra ở các biên giữa các khối trong ảnh được xây dựng lại. Quy trình lọc giải khối có thể, ví dụ, dẫn xuất biên mục tiêu trong ảnh được xây dựng lại, xác định độ bền biên (bS) cho biên mục tiêu, và thực hiện việc lọc giải khối trên biên mục tiêu, dựa trên bS. bS có thể được xác định dựa trên chế độ dự đoán, sự chênh lệch vectơ chuyển động, xem các ảnh tham chiếu có giống hệt hay không, xem hệ số đáng kể khác không có tồn tại hay không, v.v., của hai khối liền kề với biên mục tiêu.

SAO là phương pháp trong đó sự chênh lệch phần bù giữa ảnh được xây dựng lại và ảnh gốc được bù trên cơ sở mẫu. Ví dụ, SAO có thể được áp dụng dựa trên loại chẳng hạn như phần bù dải, phần bù mép, hoặc tương tự. Theo SAO, các mẫu có thể được phân loại vào các danh mục khác nhau theo mỗi loại SAO, và giá trị phần bù có thể được bổ sung vào mỗi mẫu dựa trên danh mục. Thông tin lọc cho SAO có thể gồm thông tin về việc liệu SAO có được áp dụng hay không, thông tin loại SAO, và thông tin giá trị phần bù SAO, hoặc tương tự. SAO có thể được áp dụng cho ảnh được xây dựng lại sau khi việc lọc giải khối được áp dụng.

Lọc vòng lặp thích ứng (Adaptive Loop Filter, ALF) là kỹ thuật để lọc ảnh được xây dựng lại trên cơ sở mẫu, dựa trên các hệ số bộ lọc theo hình dạng bộ lọc. Thiết bị mã hóa có thể xác định xem ALF có được áp dụng hay không, hình dạng ALF và/hoặc hệ số lọc ALF hoặc tương tự bằng việc so sánh ảnh được xây dựng lại và ảnh gốc và có thể báo hiệu kết quả xác định tới thiết bị giải mã. Nghĩa là, thông tin lọc cho ALF có thể gồm thông tin về việc liệu ALF có được áp dụng hay không, thông tin hình dạng bộ lọc ALF, thông tin hệ số lọc ALF, hoặc tương tự. ALF có thể được áp dụng cho ảnh được xây dựng lại sau khi việc lọc giải khối được áp dụng.

Fig.5 là lưu đồ minh họa phương pháp mã hóa dựa trên việc lọc trong thiết bị mã hóa. Phương pháp của Fig.5 có thể gồm các bước S500 đến S530.

Trong bước S500, thiết bị mã hóa có thể tạo ra ảnh được xây dựng lại. Bước S500 có thể được thực hiện dựa trên quy trình tạo ra ảnh được xây dựng lại được đề cập ở trên (hoặc các mẫu được xây dựng lại).

Trong bước S510, thiết bị mã hóa có thể xác định xem việc lọc trong vòng lặp có được áp dụng (ngang qua biên ảo) hay không dựa trên thông tin liên quan đến việc lọc trong vòng lặp. Ở đây, việc lọc trong vòng lặp có thể gồm ít nhất một trong số việc lọc giải khối được đề cập ở trên, SAO, và ALF.

Trong bước S520, thiết bị mã hóa có thể tạo ra ảnh được xây dựng lại được cải biến (các mẫu được xây dựng lại được cải biến) dựa trên việc xác định của bước S510. Ở đây, ảnh được xây dựng lại được cải biến (các mẫu được xây dựng lại được cải biến) có thể là ảnh được xây dựng lại được lọc (các mẫu được xây dựng lại được lọc).

Trong bước S530, thiết bị mã hóa có thể mã hóa thông tin hình ảnh/viđeo gồm thông tin liên quan đến việc lọc trong vòng lặp, dựa trên quy trình lọc trong vòng lặp.

Fig.6 là lưu đồ minh họa phương pháp giải mã dựa trên việc lọc trong thiết bị giải mã. Phương pháp của Fig.6 có thể gồm các bước S600 đến S630.

Trong bước S600, thiết bị giải mã có thể thu nhận thông tin hình ảnh/viđeo gồm thông tin liên quan đến việc lọc trong vòng lặp từ luồng bit. Ở đây, luồng bit có thể dựa trên thông tin hình ảnh/viđeo được mã hóa được truyền từ thiết bị mã hóa.

Trong bước S610, thiết bị giải mã có thể tạo ra ảnh được xây dựng lại. Bước S610 có thể được thực hiện dựa trên ảnh được xây dựng lại được đề cập ở trên (hoặc các mẫu được xây dựng lại).

Trong bước S620, thiết bị giải mã có thể xác định xem việc lọc trong vòng lặp có được áp dụng (ngang qua biên ảo) hay không dựa trên thông tin liên quan đến việc lọc trong vòng lặp. Ở đây, việc lọc trong vòng lặp có thể gồm ít nhất một trong số việc lọc giải khối được đề cập ở trên, SAO, và ALF.

Trong bước S630, thiết bị giải mã có thể tạo ra ảnh được xây dựng lại được cải biến (các mẫu được xây dựng lại được cải biến) dựa trên việc xác định của bước S620. Ở đây, ảnh được xây dựng lại được cải biến (các mẫu được xây dựng lại được cải biến) có thể là ảnh được xây dựng lại được lọc (các mẫu được xây dựng lại được lọc).

Như được mô tả ở trên, quy trình lọc trong vòng lặp có thể được áp dụng cho ảnh được xây dựng lại. Trong trường hợp này, biên ảo có thể được xác định để cải thiện thêm chất lượng trực quan chủ quan/khách quan của ảnh được xây dựng lại, và quy trình lọc trong vòng lặp có thể được áp dụng ngang qua biên ảo. Biên ảo có thể gồm, ví dụ, mép không liên tục chẳng hạn như hình ảnh 360 độ, hình ảnh VR, biên, ảnh trong ảnh (Picture In Picture, PIP), hoặc tương tự. Ví dụ, biên ảo có thể có mặt ở vị trí được xác định trước, và sự có mặt và/hoặc vị trí của nó có thể được báo hiệu. Ví dụ, biên ảo có thể được đặt ở đường mẫu thứ tư bên trên của hàng CTU (cụ thể là, ví dụ, phía trên mẫu thứ tư bên trên của hàng CTU). Như một ví dụ khác, thông tin về sự có mặt và/hoặc vị trí của biên ảo có thể được báo hiệu thông qua HLS. HLS có thể gồm SPS, PPS, phần đầu ảnh, phần đầu lát, hoặc tương tự như được mô tả ở trên.

Sau đây, việc báo hiệu cú pháp mức cao và ngữ nghĩa sẽ được mô tả theo các phương án của sáng chế.

Một phương án của sáng chế có thể gồm phương pháp điều khiển các bộ lọc vòng lặp. Phương pháp điều khiển các bộ lọc vòng lặp này có thể được áp dụng cho ảnh được xây dựng lại. Các bộ lọc trong vòng lặp (các bộ lọc vòng lặp) có thể được sử dụng để giải mã các luồng bit được mã hóa. Các bộ lọc vòng lặp có thể gồm việc giải khối được đề cập ở trên, SAO, và ALF. SPS có thể gồm các cờ liên quan đến mỗi trong số việc giải khối, SAO, và ALF. Các cờ có thể chỉ ra xem mỗi trong số các công cụ là sẵn có cho việc lập mã của trình tự viđeo lớp được lập mã (coded layer video sequence, CLVS) hay trình tự viđeo được lập mã (CVS) tham chiếu đến SPS.

Khi các bộ lọc vòng lặp là sẵn có cho CVS, việc áp dụng của các bộ lọc vòng lặp có thể được điều khiển không ngang qua các biên cụ thể. Ví dụ, việc liệu các bộ lọc vòng lặp có ngang qua các biên ảnh con hay không có thể được điều khiển. Ngoài ra, việc liệu các bộ lọc vòng lặp có ngang qua các biên phiến hay không có thể được điều khiển. Ngoài ra, việc liệu các bộ lọc vòng lặp có ngang qua các biên ảo hay không có thể được điều khiển. Ở đây, các biên ảo có thể được xác định trên các CTU dựa trên sự sẵn có của bộ đệm đường.

Liên quan đến việc liệu quy trình lọc trong vòng lặp có được thực hiện ngang qua biên ảo hay không, thông tin liên quan đến việc lọc trong vòng lặp có thể gồm ít nhất một trong số cờ được cho phép các biên ảo SPS (cờ được cho phép các biên ảo

trong SPS), cờ biểu diễn các biên ảo SPS, cờ biểu diễn các biên ảo phần đầu ảnh, cờ biểu diễn các biên ảo phần đầu ảnh SPS, và thông tin về vị trí các biên ảo.

Trong các phương án được gồm trong sáng chế, thông tin về vị trí các biên ảo có thể gồm thông tin về tọa độ x của biên ảo dọc và/hoặc thông tin về tọa độ y của biên ảo ngang. Cụ thể là, thông tin về vị trí các biên ảo có thể gồm thông tin về tọa độ x của biên ảo dọc và/hoặc thông tin về tọa độ y của biên ảo ngang trong các đơn vị của các mẫu độ chói. Ngoài ra, thông tin về vị trí các biên ảo có thể gồm thông tin về số lượng của các mẩu thông tin (các phần tử cú pháp) trên tọa độ x của biên ảo dọc mà có mặt trong SPS. Ngoài ra, thông tin về vị trí các biên ảo có thể gồm thông tin về số lượng của các mẩu thông tin (các phần tử cú pháp) trên tọa độ y của biên ảo ngang mà có mặt trong SPS. Mặt khác, thông tin về vị trí các biên ảo có thể gồm thông tin về số lượng của các mẩu thông tin (các phần tử cú pháp) trên tọa độ x của biên ảo dọc mà có mặt trong phần đầu ảnh. Ngoài ra, thông tin về vị trí các biên ảo có thể gồm thông tin về số lượng của các mẩu thông tin (các phần tử cú pháp) trên tọa độ y của biên ảo ngang mà có mặt trong phần đầu ảnh.

Các bảng dưới đây thể hiện cú pháp ví dụ và các ngữ nghĩa của SPS theo phương án này.

[Bảng 1]

seq_parameter_set_rbsp( ) {	Descriptor
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
...	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
...	
}	

[Bảng 2]

subpics_present_flag equal to 1 specifies that subpicture parameters are present in in the SPS RBSP syntax. subpics_present_flag equal to 0 specifies that subpicture parameters are not present in the SPS RBSP syntax.

sps_num_subpics_minus1 plus 1 specifies the number of subpictures. sps_num_subpics_minus1 shall be in the range of 0 to 254. When not present, the value of sps_num_subpics_minus1 is inferred to be equal to 0.

subpic_ctu_top_left_x[i] specifies horizontal position of top left CTU of i-th subpicture in unit of CtbSizeY. The length of the syntax element is Ceil(Log2(pic_width_max_in_luma_samples / CtbSizeY)) bits. When not present, the value of subpic_ctu_top_left_x[i] is inferred to be equal to 0.

subpic_ctu_top_left_y[i] specifies vertical position of top left CTU of i-th subpicture in unit of CtbSizeY. The length of the syntax element is Ceil(Log2(pic_height_max_in_luma_samples / CtbSizeY)) bits. When not present, the value of subpic_ctu_top_left_y[i] is inferred to be equal to 0.

subpic_width_minus1[i] plus 1 specifies the width of the i-th subpicture in units of CtbSizeY. The length of the syntax element is Ceil(Log2(pic_width_max_in_luma_samples / CtbSizeY)) bits. When not present, the value of subpic_width_minus1[i] is inferred to be equal to Ceil(pic_width_max_in_luma_samples / CtbSizeY) − 1.

subpic_height_minus1[i] plus 1 specifies the height of the i-th subpicture in units of CtbSizeY. The length of the syntax element is Ceil(Log2(pic_height_max_in_luma_samples / CtbSizeY)) bits. When not present, the value of subpic_height_minus1[i] is inferred to be equal to Ceil(pic_height_max_in_luma_samples / CtbSizeY) − 1.

subpic_treated_as_pic_flag[i] equal to 1 specifies that the i-th subpicture of each coded picture in the CLVS is treated as a picture in the decoding process excluding in-loop filtering operations. subpic_treated_as_pic_flag[i] equal to 0 specifies that the i-th subpicture of each coded picture in the CLVS is not treated as a picture in the decoding process excluding in-loop filtering operations. When not present, the value of subpic_treated_as_pic_flag[i] is inferred to be equal to 0.

loop_filter_across_subpic_enabled_flag[i] equal to 1 specifies that in-loop filtering operations may be performed across the boundaries of the i-th subpicture in each coded picture in the CLVS. loop_filter_across_subpic_enabled_flag[i] equal to 0 specifies that in-loop filtering operations are not performed across the boundaries of the i-th subpicture in each coded picture in the CLVS. When not present, the value of loop_filter_across_subpic_enabled_pic_flag[i] is inferred to be equal to 1.

sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

sps_sao_enabled_flag equal to 1 specifies that the sample adaptive offset process is applied to the reconstructed picture after the deblocking filter process. sps_sao_enabled_flag equal to 0 specifies that the sample adaptive offset process is not applied to the reconstructed picture after the deblocking filter process.

sps_alf_enabled_flag equal to 0 specifies that the adaptive loop filter is disabled. sps_alf_enabled_flag equal to 1 specifies that the adaptive loop filter is enabled.

sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.

sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. The value of sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.

sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.

sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. The value of sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Các bảng dưới đây thể hiện cú pháp ví dụ và các ngữ nghĩa của tập thông số ảnh (PPS) theo phương án này.

[Bảng 3]

pic_parameter_set_rbsp( ) {	Descriptor
...	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
...	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
...	
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
...	
}	

[Bảng 4]

no_pic_partition_flag equal to 1 specifies that no picture partitioning applied to each picture referring to the PPS. no_pic_partition_flag equal to 0 specifies each picture referring to the PPS may be partitioned into more than one tile or slice.

loop_filter_across_tiles_enabled_flag equal to 1 specifies that in-loop filtering operations may be performed across tile boundaries in pictures referring to the PPS. loop_filter_across_tiles_enabled_flag equal to 0 specifies that in-loop filtering operations are not performed across tile boundaries in pictures referring to the PPS. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

loop_filter_across_slices_enabled_flag equal to 1 specifies that in-loop filtering operations may be performed across slice boundaries in pictures referring to the PPS. loop_filter_across_slice_enabled_flag equal to 0 specifies that in-loop filtering operations are not performed across slice boundaries in pictures referring to the PPS. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

deblocking_filter_control_present_flag equal to 1 specifies the presence of deblocking filter control syntax elements in the PPS. deblocking_filter_control_present_flag equal to 0 specifies the absence of deblocking filter control syntax elements in the PPS.

deblocking_filter_override_enabled_flag equal to 1 specifies the presence of pic_deblocking_filter_override_flag in the PHs referring to the PPS or slice_deblocking_filter_override_flag in the slice headers referring to the PPS. deblocking_filter_override_enabled_flag equal to 0 specifies the absence of pic_deblocking_filter_override_flag in PHs referring to the PPS or slice_deblocking_filter_override_flag in slice headers referring to the PPS. When not present, the value of deblocking_filter_override_enabled_flag is inferred to be equal to 0.

pps_deblocking_filter_disabled_flag equal to 1 specifies that the operation of deblocking filter is not applied for slices referring to the PPS in which slice_deblocking_filter_disabled_flag is not present. pps_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for slices referring to the PPS in which slice_deblocking_filter_disabled_flag is not present. When not present, the value of pps_deblocking_filter_disabled_flag is inferred to be equal to 0.

pps_beta_offset_div2 and pps_tc_offset_div2 specify the default deblocking parameter offsets for β and tC (divided by 2) that are applied for slices referring to the PPS, unless the default deblocking parameter offsets are overridden by the deblocking parameter offsets present in the slice headers of the slices referring to the PPS. The values of pps_beta_offset_div2 and pps_tc_offset_div2 shall both be in the range of −6 to 6, inclusive. When not present, the value of pps_beta_offset_div2 and pps_tc_offset_div2 are inferred to be equal to 0.

Các bảng sau đây thể hiện cú pháp ví dụ và các ngữ nghĩa của phần đầu ảnh theo phương án này.

[Bảng 5]

picture_header_rbsp( ) {	Descriptor
…	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
…	
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)

if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
...	
}	

[Bảng 6]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled__present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH.

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. The value of ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH.

ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. The value of ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

pic_sao_enabled_present_flag equal to 1 specifies that pic_sao_luma_flag and pic_sao_chroma_flag are present in the PH. pic_sao_enabled_present_flag equal to 0 specifies that pic_sao_luma_flag and pic_sao_chroma_flag are not present in the PH. When pic_sao_enabled_present_flag is not present, it is inferred to be equal to 0.

pic_sao_luma_enabled_flag equal to 1 specifies that SAO is enabled for the luma component in all slices associated with the PH; pic_sao_luma_enabled_flag equal to 0 specifies that SAO for the luma component may be disabled for one, or more, or all slices associated with the PH.

pic_sao_chroma_enabled_flag equal to 1 specifies that SAO is enabled for the chroma component in all slices associated with the PH; pic_sao_chroma_enabled_flag equal to 0 specifies that SAO for chroma component may be disabled for one, or more, or all slices associated with the PH.

pic_alf_enabled_present_flag equal to 1 specifies that pic_alf_enabled_flag, pic_num_alf_aps_ids_luma, pic_alf_aps_id_luma[i], pic_alf_chroma_idc, and pic_alf_aps_id_chroma are present in the PH.
pic_alf_enabled_present_flag equal to 0 specifies that pic_alf_enabled_flag, pic_num_alf_aps_ids_luma, pic_alf_aps_id_luma[i], pic_alf_chroma_idc, and pic_alf_aps_id_chroma are not present in the PH. When pic_alf_enabled_present_flag is not present, it is inferred to be equal to 0.

pic_alf_enabled_flag equal to 1 specifies that adaptive loop filter is enabled for all slices associated with the PH and may be applied to Y, Cb, or Cr colour component in the slices. pic_alf_enabled_flag equal to 0 specifies that adaptive loop filter may be disabled for one, or more, or all slices associated with the PH. When not present, pic_alf_enabled_flag is inferred to be equal to 0.

pic_num_alf_aps_ids_luma specifies the number of ALF APSs that the slices associated with the PH refers to.

pic_alf_aps_id_luma[i] specifies the adaptation_parameter_set_id of the i-th ALF APS that the luma component of the slices associated with the PH refers to.

The value of alf_luma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to pic_alf_aps_id_luma[i] shall be equal to 1.

pic_alf_chroma_idc equal to 0 specifies that the adaptive loop filter is not applied to Cb and Cr colour components.
pic_alf_chroma_idc equal to 1 indicates that the adaptive loop filter is applied to the Cb colour component.
pic_alf_chroma_idc equal to 2 indicates that the adaptive loop filter is applied to the Cr colour component.
pic_alf_chroma_idc equal to 3 indicates that the adaptive loop filter is applied to Cb and Cr colour components. When pic_alf_chroma_idc is not present, it is inferred to be equal to 0.

pic_alf_aps_id_chroma specifies the adaptation_parameter_set_id of the ALF APS that the chroma component of the slices associated with the PH refers to.

pic_deblocking_filter_override_present_flag equal to 1 specifies that pic_deblocking_filter_override_flag is present in the PH. pic_deblocking_filter_override_present_flag equal to 0 specifies that pic_deblocking_filter_override_flag is not present in the PH. When pic_deblocking_filter_override_present_flag is not present, it is inferred to be equal to 0.

pic_deblocking_filter_override_flag equal to 1 specifies that deblocking parameters are present in the PH.
pic_deblocking_filter_override_flag equal to 0 specifies that deblocking parameters are not present in the PH. When not present, the value of pic_pic_deblocking_filter_override_flag is inferred to be equal to 0.

pic_deblocking_filter_disabled_flag equal to 1 specifies that the operation of the deblocking filter is not applied for the slices associated with the PH. pic_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for the slices associated with the PH. When pic_deblocking_filter_disabled_flag is not present, it is inferred to be equal to pps_deblocking_filter_disabled_flag.

pic_beta_offset_div2 and pic_tc_offset_div2 specify the deblocking parameter offsets for β and tC (divided by 2) for the slices associated with the PH. The values of pic_beta_offset_div2 and pic_tc_offset_div2 shall both be in the range of −6 to 6, inclusive. When not present, the values of pic_beta_offset_div2 and pic_tc_offset_div2 are inferred to be equal to pps_beta_offset_div2 and pps_tc_offset_div2, respectively.

Các bảng sau đây thể hiện cú pháp ví dụ và các ngữ nghĩa của phần đầu lát theo phương án này.

[Bảng 7]

slice_header( ) {	**Descriptor**
…	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !pic_sao_enabled_present_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !pic_deblocking_filter_override_present_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
…	
}	

[Bảng 8]

cu_chroma_qp_offset_enabled_flag equal to 1 specifies that the cu_chroma_qp_offset_flag may be present in the transform unit and palette coding syntax. cu_chroma_qp_offset_enabled_flag equal to 0 specifies that the cu_chroma_qp_offset_flag is not present in the transform unit or palette coding syntax. When not present, the value of cu_chroma_qp_offset_enabled_flag is inferred to be equal to 0.

slice_sao_luma_flag equal to 1 specifies that SAO is enabled for the luma component in the current slice; slice_sao_luma_flag equal to 0 specifies that SAO is disabled for the luma component in the current slice. When slice_sao_luma_flag is not present, it is inferred to be equal to pic_sao_luma_enabled_flag.

slice_sao_chroma_flag equal to 1 specifies that SAO is enabled for the chroma component in the current slice; slice_sao_chroma_flag equal to 0 specifies that SAO is disabled for the chroma component in the current slice. When slice_sao_chroma_flag is not present, it is inferred to be equal to pic_sao_chroma_enabled_flag.

slice_alf_enabled_flag equal to 1 specifies that adaptive loop filter is enabled and may be applied to Y, Cb, or Cr colour component in a slice. slice_alf_enabled_flag equal to 0 specifies that adaptive loop filter is disabled for all colour components in a slice. When not present, the value of slice_alf_enabled_flag is inferred to be equal to pic_alf_enabled_flag.

slice_num_alf_aps_ids_luma specifies the number of ALF APSs that the slice refers to. When slice_alf_enabled_flag is equal to 1 and slice_num_alf_aps_ids_luma is not present, the value of slice_num_alf_aps_ids_luma is inferred to be equal to the value of pic_num_alf_aps_ids_luma.

slice_alf_aps_id_luma[i] specifies the adaptation_parameter_set_id of the i-th ALF APS that the luma component of the slice refers to. The TemporalId of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_luma[i] shall be less than or equal to the TemporalId of the coded slice NAL unit. When slice_alf_enabled_flag is equal to 1 and slice_alf_aps_id_luma[i] is not present, the value of slice_alf_aps_id_luma[i] is inferred to be equal to the value of pic_alf_aps_id_luma[i].

The value of alf_luma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_luma[i] shall be equal to 1.

slice_alf_chroma_idc equal to 0 specifies that the adaptive loop filter is not applied to Cb and Cr colour components. slice_alf_chroma_idc equal to 1 indicates that the adaptive loop filter is applied to the Cb colour component. slice_alf_chroma_idc equal to 2 indicates that the adaptive loop filter is applied to the Cr colour component. slice_alf_chroma_idc equal to 3 indicates that the adaptive loop filter is applied to Cb and Cr colour components. When slice_alf_chroma_idc is not present, it is inferred to be equal to pic_alf_chroma_idc.

slice_alf_aps_id_chroma specifies the adaptation_parameter_set_id of the ALF APS that the chroma component of the slice refers to. The TemporalId of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_chroma shall be less than or equal to the TemporalId of the coded slice NAL unit. When slice_alf_enabled_flag is equal to 1 and slice_alf_aps_id_chroma is not present, the value of slice_alf_aps_id_chroma is inferred to be equal to the value of pic_alf_aps_id_chroma.

The value of alf_chroma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_chroma shall be equal to 1.

slice_deblocking_filter_override_flag equal to 1 specifies that deblocking parameters are present in the slice header. slice_deblocking_filter_override_flag equal to 0 specifies that deblocking parameters are not present in the slice header. When not present, the value of slice_deblocking_filter_override_flag is inferred to be equal to pic_deblocking_filter_override_flag.

slice_deblocking_filter_disabled_flag equal to 1 specifies that the operation of the deblocking filter is not applied for the current slice. slice_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for the current slice. When slice_deblocking_filter_disabled_flag is not present, it is inferred to be equal to pic_deblocking_filter_disabled_flag.

slice_beta_offset_div2 and slice_tc_offset_div2 specify the deblocking parameter offsets for β and tC (divided by 2) for the current slice. The values of slice_beta_offset_div2 and slice_tc_offset_div2 shall both be in the range of −6 to 6, inclusive. When not present, the values of slice_beta_offset_div2 and slice_tc_offset_div2 are inferred to be equal to pic_beta_offset_div2 and pic_tc_offset_div2, respectively.

Sau đây, việc báo hiệu của thông tin về các biên ảo mà có thể được sử dụng trong việc lọc trong vòng lặp sẽ được mô tả.

Trong thiết kế hiện có, để bất hoạt các bộ lọc vòng lặp ngang qua các biên ảo, có hai lựa chọn, nghĩa là, lựa chọn i) trong đó cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) có thể được thiết đặt là 0, và cho mọi phần đầu ảnh, cờ biểu diễn các biên ảo PH (ph_loop_filter_across_virtual_boundaries_disabled_present_flag) có thể có mặt và

được thiết đặt là 0, và lựa chọn ii) trong đó cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) có thể được thiết đặt là 1, và thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_vertical_boudnaries) và thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_vertical_boudnaries) có thể được thiết đặt là 0.

Trong thiết kế hiện có, theo lựa chọn ii), cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) được thiết đặt là 1, và do đó bộ giải mã dự kiến việc báo hiệu trên các vị trí của các biên ảo, mà có thể gây ra vấn đề trong quy trình giải mã.

Các phương án được mô tả sau đây có thể đề xuất các giải pháp cho vấn đề được đề cập ở trên. Các phương án có thể được áp dụng một cách độc lập. Mặt khác, ít nhất hai phương án có thể được áp dụng kết hợp.

Trong một phương án của sáng chế, việc liệu các phần tử cú pháp để chỉ ra các biên ảo có được gồm trong SPS hay không có thể được điều khiển bởi (các) cờ. Ví dụ, số lượng của (các) cờ có thể là 2 (ví dụ, cờ được cho phép các biên ảo SPS, cờ biểu diễn các biên ảo SPS).

Trong một ví dụ theo phương án này, cờ được cho phép các biên ảo SPS có thể được gọi là sps_loop_filter_across_virtual_boundaries_disabled_flag (hoặc sps_virtual_boundaries_enabled_flag). Cờ được cho phép các biên ảo SPS có thể chỉ ra xem đặc tính để bất hoạt bộ lọc vòng lặp ngang qua các biên ảo có được cho phép hay không.

Trong một ví dụ theo phương án này, cờ biểu diễn các biên ảo SPS có thể được gọi là sps_loop_filter_across_virtual_boundaries_disabled_present_flag (hoặc sps_virtual_boundaries_present_flag). Cờ biểu diễn các biên ảo SPS có thể chỉ ra xem thông tin báo hiệu cho các biên ảo có được gồm trong SPS hoặc trong phần đầu ảnh (PH) hay không.

Trong một ví dụ theo phương án này, khi cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 0, thông tin báo hiệu để bất hoạt bộ lọc vòng lặp ngang qua các biên ảo có thể được gồm trong

PH.

Trong một ví dụ theo phương án này, khi thông tin về các vị trí của các biên ảo (ví dụ, các biên ảo dọc, các biên ảo ngang) được gồm trong SPS, có thể bị ràng buộc rằng tổng của số lượng của các biên ảo dọc và số lượng của các biên ảo ngang là lớn hơn 0.

Trong một ví dụ theo phương án này, (các) biến chỉ ra xem bộ lọc bị bất hoạt ở các biên ảo cho ảnh hiện tại có thể được dẫn xuất hay không. Ví dụ, (các) biến có thể gồm VirtualBoundariesDisabledFlag.

Như một trường hợp của ví dụ này, khi cờ cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, VirtualBoundariesDisabledFlag có thể là 1.

Như một trường hợp khác của ví dụ này, khi cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1, cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 0, và tổng của thông tin về số lượng của các biên ảo dọc (ví dụ, ph_num_ver_virtual_boundaries) và thông tin về số lượng của các biên ảo ngang (ví dụ, ph_num_hor_virtual_boundaries) là lớn hơn 0, VirtualBoundariesDisabledFlag có thể là 1.

Trong các trường hợp khác của ví dụ này, VirtualBoundariesDisabledFlag có thể là 0.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 9]

seq_parameter_set_rbsp( ) {	Descriptor
sps_decoding_parameter_set_id	u(4)
…	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 10]

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 11]

picture_header_rbsp( ) {	**Descriptor**
non_reference_picture_flag	u(1)
…	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag && !sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	ue(v)
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 12]

ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH. When ph_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.

The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    if( sps_loop_filter_across_virtual_boundaries_disabled_present_flag ||
            ( ph_num_ver_virtual_boundaries + ph_num_ver_virtual_boundaries > 0 ) )
        VirtualBoundariesDisabledFlag = 1
```

The parameter VirtualBoundariesNumVer is derived as follows:

```
VirtualBoundariesNumVer = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
        VirtualBoundariesNumVer =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_num_ver_virtual_boundaries : ph_num_ver_virtual_boundaries                (7-43)
```

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosX[ i ] =
(sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ]) * 8      (7-44)
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
The parameter VirtualBoundariesNumHor is derived as follows:

```
VirtualBoundariesNumHor = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesNumHor =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
            sps_num_hor_virtual_boundaries : ph_num_hor_virtual_boundaries            (7-45)
```

ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:

```
VirtualBoundariesPosY[ i ]
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosY[ i ] =
(sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_virtual_boundaries_pos_y[ i ] : ph_virtual_boundaries_pos_y[ i ]) * 8      (7-46)
```

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples.

Trong một phương án liên quan đến Bảng 9 đến Bảng 12, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua

luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (picture header, PH). SPS có thể gồm cờ được cho phép các biên ảo (sps_loop_filter_across_virtual_boundaries_disabled_flag). SPS có thể gồm cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag), dựa trên cờ được cho phép các biên ảo.

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, SPS có thể gồm cờ biểu diễn các biên ảo SPS. Dựa trên cờ được cho phép các biên ảo và cờ biểu diễn các biên ảo SPS, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]). Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 1, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS, thông tin về vị trí các biên ảo dọc SPS, thông tin về số lượng của các biên ảo ngang SPS, và thông tin về vị trí các biên ảo ngang SPS.

Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc SPS, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang SPS. Dựa trên cờ được cho phép các biên ảo và cờ biểu diễn các biên ảo SPS, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH (ph_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang PH (ph_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 0, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH, thông tin về vị trí các biên ảo dọc PH, thông tin về số lượng của các biên ảo ngang PH, và thông tin về vị trí biên ảo ngang PH. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc PH, và số lượng của các mẩu thông tin về vị trí các biên ảo

ngang PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang PH.

Trong một phương án khác của sáng chế, mỗi mẩu của thông tin phần đầu (các phần đầu ảnh) của các ảnh tham chiếu đến SPS có thể gồm cờ biểu diễn các biên ảo PH (ph_loop_filter_across_virtual_boundaries_disabled_present_flag hoặc ph_virtual_boundaries_present_flag). Phương án này cũng có thể được mô tả cùng với cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) và cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag).

Trong một ví dụ theo phương án này, khi giá trị của cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và giá trị của cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 0, mỗi mẩu thông tin (các phần đầu ảnh) của các ảnh tham chiếu đến SPS có thể gồm cờ biểu diễn các biên ảo PH (ph_loop_filter_across_virtual_boundaries_disalbed_present_flag hoặc ph_virtual_boundaries_present_flag).

Trong một ví dụ theo phương án này, khi thông tin về các vị trí của các biên ảo (ví dụ, các biên ảo dọc, các biên ảo ngang) được gồm trong SPS, tổng của số lượng của các biên ảo dọc và số lượng của các biên ảo ngang có thể bị ràng buộc là lớn hơn 0.

Trong một ví dụ theo phương án này, (các) biến chỉ ra xem bộ lọc bị bất hoạt ở các biên ảo có thể được dẫn xuất cho ảnh hiện tại hay không. Ví dụ, (các) biến có thể gồm VirtualBoundariesDisabledFlag.

Như một trường hợp của ví dụ này, khi cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, VirtualBoundariesDisabledFlag có thể là 1.

Như một trường hợp khác của ví dụ này, khi cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và cờ biểu diễn các biên ảo PH (ph_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, VirtualBoundariesDisabledFlag có thể là 1.

Trong các trường hợp khác của ví dụ này, VirtualBoundariesDisabledFlag có thể là 0.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 13]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
...	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 14]

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 15]

picture_header_rbsp( ) {	**Descriptor**
non_reference_picture_flag	u(1)
…	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag && !sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 16]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH. ph_loop_filter_across_virtual_boundaries_disabled__present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0.

The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag
        || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

Alternatively, the following constraint may be specified:
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 0, ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1
ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH. When ph_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
The parameter VirtualBoundariesNumVer is derived as follows:

```
VirtualBoundariesNumVer = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
        VirtualBoundariesNumVer =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
                sps_num_ver_virtual_boundaries : ph_num_ver_virtual_boundaries          (7-43)
```

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
      VirtualBoundariesPosX[ i ] =
(sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
              sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ]) * 8
```

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosX[ i ] =
(sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ]) * 8        (7-44)
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
The parameter VirtualBoundariesNumHor is derived as follows:

```
VirtualBoundariesNumHor = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesNumHor =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_num_hor_virtual_boundaries : ph_num_hor_virtual_boundaries        (7-45)
```

ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:

```
VirtualBoundariesPosY[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosY[ i ] =
(sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
        sps_virtual_boundaries_pos_y[ i ] : ph_virtual_boundaries_pos_y[ i ]) * 8        (7-46)
```

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.

Trong một phương án liên quan đến Bảng 13 đến Bảng 16, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (PH). SPS có thể gồm cờ được cho phép các biên ảo (sps_loop_filter_across_virtual_boundaries_disabled_flag). SPS có thể gồm cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag), dựa trên cờ được cho phép các biên ảo. Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, SPS có thể gồm cờ biểu diễn các biên ảo SPS. Dựa trên cờ được cho phép các biên ảo và cờ biểu diễn các biên ảo SPS, SPS có thể gồm thông tin về số lượng của các biên ảo dọc

SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 1, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS, thông tin về vị trí các biên ảo dọc SPS, thông tin về số lượng của các biên ảo ngang SPS, và thông tin về vị trí các biên ảo ngang SPS. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc SPS, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang SPS. Phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH, dựa trên cờ được cho phép các biên ảo và cờ biểu diễn các biên ảo SPS.

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 0, phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH. Dựa trên cờ biểu diễn các biên ảo PH, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH (ph_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang PH (ph_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH, thông tin về vị trí các biên ảo dọc PH, thông tin về số lượng của các biên ảo ngang PH, và thông tin về vị trí các biên ảo ngang PH. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc PH, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang PH.

Trong một phương án khác của sáng chế, việc liệu các phần tử cú pháp để chỉ ra các biên ảo có được gồm trong SPS hay không có thể được điều khiển bởi (các) cờ. Ví dụ, số lượng của (các) cờ có thể là 2 (ví dụ, cờ biểu diễn các biên ảo SPS, cờ biểu diễn các biên ảo PH SPS).

Trong một ví dụ theo phương án này, cờ biểu diễn các biên ảo SPS có thể được gọi là sps_loop_filter_across_virtual_boundaries_disabled_present_flag (hoặc sps_virtual_boundaries_present_flag). Cờ biểu diễn các biên ảo SPS có thể chỉ ra xem thông tin các biên ảo có được gồm trong SPS hay không.

Trong một ví dụ theo sáng chế, cờ biểu diễn các biên ảo PH SPS có thể được gọi là sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag. Cờ biểu diễn các biên ảo PH SPS có thể chỉ ra xem thông tin các biên ảo có được gồm trong phần đầu ảnh (PH) hay không.

Trong một ví dụ theo phương án này, nó có thể bị ràng buộc thêm rằng, khi cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, cờ biểu diễn các biên ảo PH SPS (sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag) là không có mặt và được suy ra là 0.

Trong một ví dụ theo phương án này, khi cờ biểu diễn các biên ảo PH SPS (sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, thông tin báo hiệu để bất hoạt bộ lọc vòng lặp ngang qua các biên ảo có thể được gồm trong PH.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 17]

seq_parameter_set_rbsp( ) {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
} else	
sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
...	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 18]

sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that virtual boundary locations for disabling in-loop filter operations are present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that virtual boundary locations for disabling in-loop filter operations are not present in the SPS . In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that virtual boundary locations for disabling in-loop filter operations may be present in the picture header of pictures referring to the SPS. sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that virtual boundary locations for disabling in-loop filter operations are not present in picture header of pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constrained to be equal to 0 and and sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag is constrained to be 1.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 19]

picture_header_rbsp( ) {	Descriptor
non_reference_picture_flag	u(1)
…	
if(sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 20]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH. ph_loop_filter_across_virtual_boundaries_disabled__present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0.
The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag ||
        sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ||
                ph_loop_filter_across_virtual_boundaries_disabled_present_flag          (7-42)
```

ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH. When ph_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0. . When sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, ph_num_ver_virtual_boundaries is constraint to be greater than 0.
The parameter VirtualBoundariesNumVer is derived as follows:

```
VirtualBoundariesNumVer = sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_num_ver_virtual_boundaries :
    (sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    ph_num_ver_virtual_boundaries : 0)                                          (7-43)
```

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_virtual_boundaries_pos_x[ i ] :
    (sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    ph_virtual_boundaries_pos_x[ i ]) * 8 : 0)                                  (7-44)
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.
The parameter VirtualBoundariesNumHor is derived as follows:

VirtualBoundariesNumHor = sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
sps_num_hor_virtual_boundaries :
(sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?
ph_num_hor_virtual_boundaries : 0) (7-45)

ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:

VirtualBoundariesPosY[i] = (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
sps_virtual_boundaries_pos_y[i] :
(sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?
ph_virtual_boundaries_pos_y[i]) * 8) : 0) (7-46)
The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples

Trong một phương án liên quan đến Bảng 17 đến Bảng 20, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (PH). SPS có thể gồm cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag). Dựa trên cờ biểu diễn các biên ảo SPS, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo SPS là 1, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS, thông tin về vị trí biên ảo dọc SPS, thông tin về số lượng của các biên ảo ngang SPS, và thông tin về vị trí biên ảo ngang SPS. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc SPS có thể được xác định dựa trên thông tin về các biên ảo dọc SPS, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang SPS có thể được xác định dựa trên số lượng của các biên ảo ngang SPS. SPS có thể gồm cờ biểu diễn các biên ảo PH SPS, dựa trên cờ biểu diễn các biên ảo SPS.

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo SPS là 0, SPS có thể gồm cờ biểu diễn các biên ảo PH SPS. Phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH, dựa trên cờ biểu diễn các biên ảo PH SPS. Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH SPS là 1, phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH. Dựa trên cờ biểu diễn các biên ảo PH, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH (ph_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang PH (ph_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH, thông tin về vị trí các biên ảo dọc PH, thông tin về số lượng của các biên ảo ngang PH, và thông tin về vị trí các biên ảo ngang PH. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc PH, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang PH.

Trong một phương án khác của sáng chế, khi việc làm mới giải mã dần (gradual decoding refresh, GDR) là sẵn có (nghĩa là, giá trị của gdr_enabled_flag là 1), dấu hiệu trong đó các bộ lọc vòng lặp được bất hoạt ở các biên ảo được cho phép, và thông tin các biên ảo có thể được báo hiệu trong phần đầu ảnh (có thể được gồm trong phần đầu ảnh).

Trong một phương án khác của sáng chế, khi chức năng của các bộ lọc vòng lặp bất hoạt ngang qua các biên ảo được cho phép, thông tin về việc báo hiệu của vị trí của các biên ảo có thể được gồm trong một hoặc nhiều tập thông số. Ví dụ, khi chức năng bất hoạt các bộ lọc vòng lặp ngang qua các biên ảo được cho phép, thông tin về vị trí của các biên ảo có thể được gồm trong SPS và phần đầu ảnh.

Trong phương án này, khi cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1 và thông tin báo hiệu về vị trí của các biên ảo được gồm trong một hoặc nhiều tập thông số, các mục sau đây có thể được áp dụng.

a) Thông tin báo hiệu về vị trí của các biên ảo có thể được gồm chỉ trong SPS, hoặc được gồm chỉ trong phần đầu ảnh, hoặc được gồm cả trong SPS và phần đầu ảnh.

b) VirtualBoundariesDisabledFlag cho mỗi ảnh có thể được dẫn xuất như sau đây.

- Khi sps_loop_filter_across_virtual_boundaries_disabled_flag là 0, VirtualBoundariesDisabledFlag có thể được thiết đặt là 0.

- Trong một trường hợp khác của ví dụ này, khi thông tin về vị trí của các biên ảo không được báo hiệu cả trong SPS và phần đầu ảnh được kết hợp với ảnh, VirtualBoundariesDisabledFlag có thể được thiết đặt là 0.

- Trong các trường hợp khác của ví dụ này (khi vị trí của các biên ảo được báo hiệu chỉ trong SPS hoặc chỉ trong phần đầu ảnh hoặc cả trong SPS và phần đầu ảnh), VirtualBoundariesDisabledFlag có thể được thiết đặt là 1.

c) Các biên ảo được áp dụng cho ảnh có thể gồm tập hợp các biên ảo được báo hiệu trong tập thông số mà được tham chiếu trực tiếp hoặc gián tiếp tới bởi ảnh. Ví dụ, các biên ảo có thể gồm các biên ảo (nếu có mặt) được báo hiệu trong SPS. Ví dụ, các biên ảo có thể gồm các biên ảo (nếu có mặt) được báo hiệu trong ảnh được kết hợp với ảnh.

d) Ràng buộc có thể được áp dụng sao cho số lượng tối đa của các biên ảo mỗi ảnh không vượt quá giá trị được xác định trước. Ví dụ, giá trị được xác định trước có thể là 8.

e) Có thể bị ràng buộc thêm rằng thông tin (nếu có mặt) ở vị trí của biên ảo được báo hiệu trong phần đầu ảnh sẽ không trùng với thông tin về các vị trí biên ảo được gồm trong tập thông số khác (ví dụ, SPS hoặc PPS).

- Mặt khác, đối với vị trí biên ảo bất kỳ được áp dụng cho ảnh hiện tại, vị trí biên ảo (ví dụ, cùng vị trí biên ảo được báo hiệu trong SPS và phần đầu ảnh được kết hợp với ảnh) có thể được gồm trong hai tập thông số khác nhau.

f) Có thể bị ràng buộc thêm rằng, khi cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) là 1, cờ biểu diễn các biên ảo PH SPS

(sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag) là không có mặt và được suy ra là 0.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 21]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
…	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 22]

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 23]

picture_header_rbsp( ) {	**Descriptor**
non_reference_picture_flag	u(1)
…	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong

cú pháp.

[Bảng 24]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH. ph_loop_filter_across_virtual_boundaries_disabled__present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0.

The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag
        || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

Alternatively, the following constraint may be specified:
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 0, ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1
ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH. When ph_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
The parameter VirtualBoundariesNumVer is derived as follows:

```
VirtualBoundariesNumVer =    sps_num_ver_virtual_boundaries +
ph_num_ver_virtual_boundaries                    (7-43)
```

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
The list VirtualBoundariesPosX[i], for I ranges from 0 to VirtualBoundariesNumVer − 1, inclusive, is derived as follows:

```
for( i = 0; i < VirtualBoundariesNumVer; i++) {
    if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
        VirtualBoundariesPosX[ i ] = ( i < sps_num_ver_virtual_boundaries ) ?
            sps_virtual_boundaries_pos_x[ i ] :
            ph_virtual_boundaries_pos_x[ i − sps_num_ver_virtual_boundaries ]
        VirtualBoundariesPosX[ i ]  *= 8
    } else
        VirtualBoundariesPosX[ i ] = 0
}
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
The parameter VirtualBoundariesNumHor is derived as follows:

VirtualBoundariesNumHor = sps_num_hor_virtual_boundaries + ph_num_hor_virtual_boundaries (7-45)

ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:
The list VirtualBoundariesPosY[i], for i ranges from 0 to VirtualBoundariesNumHor − 1, inclusive, is derived as follows:

```
for( i = 0; i < VirtualBoundariesNumHor; i++) {
    if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
        VirtualBoundariesPosY[ i ] = ( i < sps_num_hor_virtual_boundaries ) ?
            sps_virtual_boundaries_pos_y[ i ] :
            ph_virtual_boundaries_pos_y[ i − sps_num_hor_virtual_boundaries ]
        VirtualBoundariesPosY[ i ] *= 8
    } else
        VirtualBoundariesPosY[ i ] = 0
}
```

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.

Trong một phương án liên quan đến Bảng 21 đến Bảng 24, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (PH). SPS có thể gồm cờ được cho phép các biên ảo (sps_loop_filter_across_virtual_boundaries_disabled_flag). SPS có thể gồm cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag), dựa trên cờ được cho phép các biên ảo. Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, SPS có thể gồm cờ biểu diễn các biên ảo SPS. Dựa trên cờ được cho phép các biên ảo và cờ biểu diễn các biên ảo SPS, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 1, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS, thông tin về vị trí các biên ảo dọc SPS, thông tin về số lượng của các biên ảo ngang SPS, và thông tin về vị trí các biên ảo ngang SPS. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc SPS, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang SPS. Phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH, dựa trên cờ được cho phép các biên ảo.

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH. Dựa trên cờ biểu diễn các biên ảo PH, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH (ph_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang PH (ph_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y[i]). Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH, thông tin về vị trí các biên ảo dọc PH, thông tin về số lượng của các biên ảo ngang PH, và thông tin về vị trí các biên ảo ngang PH. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo dọc PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc PH, và số lượng của các mẩu thông tin về vị trí các biên ảo ngang PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang PH.

Trong một phương án khác của sáng chế, việc lọc vòng lặp có thể được thực hiện theo các phương án được đề cập ở trên bằng việc không giới hạn rằng tổng của số lượng của các biên ảo dọc và số lượng của các biên ảo ngang là lớn hơn 0.

Trong một phương án khác của sáng chế, thông tin về biên ảo có thể được báo hiệu trong cả SPS và PH. Trong một ví dụ của phương án này, khi giá trị của cờ được cho phép các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1, thông tin về số lượng của các biên ảo dọc, thông tin về số lượng của các biên ảo ngang, và/hoặc thông tin về vị trí các biên ảo có thể được gồm trong SPS. Ngoài ra, khi giá trị của cờ được cho phép các biên ảo SPS

(sps_loop_filter_across_virtual_boundaries_disabled_flag) là 1, thông tin về số lượng của các biên ảo dọc, thông tin về số lượng của các biên ảo ngang, và/hoặc thông tin về giá trị delta vị trí các biên ảo (giá trị delta của vị trí các biên ảo) có thể được gồm trong phần đầu ảnh. Giá trị delta của vị trí các biên ảo có thể là để chỉ sự chênh lệch giữa các vị trí của các biên ảo. Thông tin về dấu của vị trí các biên ảo cũng có thể được gồm trong phần đầu ảnh.

Theo một ví dụ của phương án này, để dẫn xuất các vị trí các biên ảo cho các ảnh tương ứng, nếu giá trị delta của vị trí các biên ảo là không có mặt trong phần đầu ảnh, thông tin về vị trí các biên ảo, được báo hiệu trong SPS, có thể được sử dụng cho việc lọc vòng lặp. Nếu giá trị delta của vị trí các biên ảo có mặt trong phần đầu ảnh, vị trí các biên ảo có thể được dẫn xuất dựa trên tổng của thông tin về vị trí các biên ảo, được báo hiệu trong SPS, và giá trị delta liên quan đến đó.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 25]

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_decoding_parameter_set_id	u(4)
…	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 26]

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 27]

picture_header_rbsp( ) {	Descriptor
non_reference_picture_flag	u(1)
…	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
for(i = 0; i < sps_num_ver_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_x_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_x_delta[i] > 0)	
ph_virtual_boundaries_pos_x_sign[i]	u(1)
}	
for(i = 0; i < sps_num_hor_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_y_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_y_delta[i] > 0)	
ph_virtual_boundaries_pos_y_sign[i]	u(1)
}	
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 28]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH. ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0.

The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag = (sps_num_ver_virtual_boundaries +
                                 sps_num_hor_virtual_boundaries > 0 ) ? 1 : 0
```

ph_virtual_boundaries_pos_x_delta[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
ph_virtual_boundaries_pos_x_sign[i] specifies the sign of the i-th virtual boundary, which specifies the location of the i th vertical virtual boundary in units of luma samples. When not present, the value of ph_virtual_boundaries_pos_x_sign[i] is inferred to be equal to 0.
The variable phVirtualBoundariesPosX[i] for i ranges from 0 to sps_num_ver_virtual_boundaries − 1, inclusive, is initialized as follows:

```
phVirtualBoundariesPosX[ i ]  = ph_virtual_boundaries_pos_x_delta[ i ] *
                                ( 1 − 2 * ph_virtual_boundaries_pos_x_sign[ i ])
```

The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
    if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )                (7-44)
        VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] +
                phVirtualBoundariesPosX[ i ] ) * 8
    else
        VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] )  * 8
}
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
ph_virtual_boundaries_pos_y_delta[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:

ph_virtual_boundaries_pos_y_sign [i] specifies the sign of the i-th virtual boundary, which specifies the location of the i th horizontal virtual boundary in units of luma samples. When not present, the value of ph_virtual_boundaries_pos_y_sign[i] is inferred to be equal to 0.
The variable phVirtualBoundariesPosY[i] for i ranges from 0 to sps_num_hor_virtual_boundaries – 1, inclusive, is initialized as follows:

```
phVirtualBoundariesPosY[ i ]  = ph_virtual_boundaries_pos_y_delta[ i ] *
                                ( 1 − 2 * ph_virtual_boundaries_pos_y_sign [ i ])

VirtualBoundariesPosY[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
    if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )                (7-46)
        VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] +
        phVirtualBoundariesPosY[ i ] ) * 8
    else
        VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] ) * 8
}
```

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries, sps_num_hor_virtual_boundaries, ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall not be greater than 8.

Trong một phương án liên quan đến Bảng 25 đến Bảng 28, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (PH). SPS có thể gồm cờ được cho phép các biên ảo (sps_loop_filter_across_virtual_boundaries_disabled_flag). Dựa trên cờ được cho phép các biên ảo, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]). Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, SPS có thể gồm thông tin về số lượng của các biên ảo ngang SPS, thông tin về vị trí các biên ảo ngang SPS, thông tin về số lượng của các biên ảo dọc SPS, và thông tin về vị trí các biên ảo dọc SPS.

Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo ngang SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang SPS, và số lượng của các mẩu thông tin về vị trí các biên ảo dọc SPS có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc SPS. Phần đầu ảnh có thể gồm cờ biểu diễn

các biên ảo PH, dựa trên cờ được cho phép các biên ảo. Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH. Dựa trên cờ biểu diễn các biên ảo PH, phần đầu ảnh có thể gồm thông tin về giá trị delta của vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_x_delta[i]), thông tin về dấu của vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_x_sign[i]), thông tin về giá trị delta của vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_y_delta[i]), và thông tin về dấu của vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_y_sign[i]).

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1, phần đầu ảnh có thể gồm thông tin về giá trị delta vị trí các biên ảo dọc PH, thông tin về dấu của vị trí các biên ảo dọc PH, thông tin về giá trị delta vị trí các biên ảo ngang PH, và thông tin dấu của vị trí các biên ảo ngang PH. Trong một ví dụ, số lượng của các mẩu thông tin về giá trị delta vị trí các biên ảo dọc PH và thông tin về số lượng của các mẩu thông tin về dấu của vị trí các biên ảo dọc PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc SPS, và số lượng của các mẩu thông tin về giá trị delta vị trí các biên ảo ngang PH và thông tin về số lượng của các mẩu thông tin dấu của vị trí các biên ảo ngang PH có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang SPS.

Trong một phương án khác của sáng chế, việc báo hiệu của thông tin về vị trí các biên ảo cho mỗi ảnh sẽ được mô tả. Trong một ví dụ, khi thông tin về vị trí các biên ảo được gồm trong SPS và thông tin về giá trị delta vị trí các biên ảo không được gồm trong phần đầu ảnh, thông tin về các biên ảo, được gồm trong SPS, có thể được sử dụng cho việc lọc vòng lặp. Khi thông tin về vị trí các biên ảo không được gồm trong SPS và thông tin về giá trị delta vị trí các biên ảo được gồm trong phần đầu ảnh, thông tin về các biên ảo, được gồm trong phần đầu ảnh, có thể được sử dụng cho việc lọc vòng lặp. Khi thông tin về vị trí các biên ảo được gồm trong SPS và thông tin về giá trị delta vị trí các biên ảo được gồm trong phần đầu ảnh, vị trí các biên ảo có thể được dẫn xuất dựa trên tổng của thông tin về vị trí các biên ảo, được tạo dấu trong SPS, và giá trị delta liên quan đến đó. Khi thông tin về vị trí các biên ảo không được gồm trong SPS và thông tin về giá trị delta vị trí các biên ảo không được gồm trong phần đầu ảnh, biên ảo có thể không được áp dụng cho ảnh.

Bảng dưới đây thể hiện cú pháp ví dụ của SPS theo phương án này.

[Bảng 29]

seq_parameter_set_rbsp( ) {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
...	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 30]

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0.
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.

Bảng dưới đây thể hiện cú pháp ví dụ của thông tin phần đầu (phần đầu ảnh) theo phương án này.

[Bảng 31]

picture_header_rbsp( ) {	**Descriptor**
non_reference_picture_flag	u(1)
…	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
if(!sps_num_ver_virtual_boundaries)	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_x_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_x_delta[i] > 0)	
ph_virtual_boundaries_pos_x_sign[i]	u(1)
}	
if(!sps_num_hor_virtual_boundaries)	
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_y_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_y_delta[i] > 0)	
ph_virtual_boundaries_pos_y_sign[i]	u(1)
}	
}	
}	
…	
}	

Bảng dưới đây thể hiện ngữ nghĩa ví dụ của các phần tử cú pháp được gồm trong cú pháp.

[Bảng 32]

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH. ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of ph_loop_filter_across_virtual_boundaries_disabled_present_flag is inferred to be equal to 0.

The parameter VirtualBoundariesDisabledFlag is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
sps_loop_filter_across_virtual_boundaries_disabled_present_flag
        || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

Alternatively, the following constraint may be specified:

When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 0, ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1

ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x_delta[i] syntax elements that are present in the PH. When ph_num_ver_virtual_boundaries is not present, it is inferred to be equal to sps_num_ver_virtual_boundaries.

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y_delta[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to sps_num_hot_virtual_boundaries.

ph_virtual_boundaries_pos_x_delta[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to Ceil(pic_width_in_luma_samples ÷ 8) − 1, inclusive.

ph_virtual_boundaries_pos_x_sign [i] specifies the sign of the i-th virtual boundary, which specifies the location of the i th vertical virtual boundary in units of luma samples. When not present, the value of ph_virtual_boundaries_pos_x_sign[i] is inferred to be equal to 0.

The variable phVirtualBoundariesPosX[i] for i ranges from 0 to sps_num_ver_virtual_boundaries – 1, inclusive, is initialized as follows:

```
phVirtualBoundariesPosX[ i ] = ph_virtual_boundaries_pos_x_delta[ i ] *
                        ( 1 − 2 * ph_virtual_boundaries_pos_x_sign[ i ])
```

The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
    if( sps_loop_filter_across_virtual_boundaries_disabled_present_flag ) {
        if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )        (7-44)
            VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] +
                                    phVirtualBoundariesPosC[ i ] ) * 8
        else
            VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] ) * 8
    } else if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag)
        VirtualBoundariesPosX[ i ] = phVirtualBoundariesPosX[ i ] * 8
}
```

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.
ph_virtual_boundaries_pos_y_delta[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) − 1, inclusive.
The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:
ph_virtual_boundaries_pos_y_sign [i] specifies the sign of the i-th virtual boundary, which specifies the location of the i th horizontal virtual boundary in units of luma samples. When not present, the value of ph_virtual_boundaries_pos_y_sign[i] is inferred to be equal to 0.
The variable phVirtualBoundariesPosY[i] for i ranges from 0 to sps_num_hor_virtual_boundaries − 1, inclusive, is initialized as follows:

```
phVirtualBoundariesPosY[ i ]  = ph_virtual_boundaries_pos_y_delta[ i ] *
                                ( 1 − 2 * ph_virtual_boundaries_pos_y_sign[ i ])

    VirtualBoundariesPosY[ i ] = 0
    if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
        if( sps_loop_filter_across_virtual_boundaries_disabled_present_flag )
            if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )          (7-46)
                VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] +
                                    phVirtualBoundariesPosY[ i ] ) * 8
            else
                VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] ) * 8
        else if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag)
            VirtualBoundariesPosY[ i ] = phVirtualBoundariesPosY[ i ] ) * 8
    }
```

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries, sps_num_hor_virtual_boundaries, ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall not be greater than 8.

Trong một phương án liên quan đến Bảng 29 đến Bảng 32, thông tin hình ảnh được thu nhận bởi thiết bị mã hóa và/hoặc thông tin hình ảnh được thu nhận thông qua luồng bit được nhận từ thiết bị mã hóa đến thiết bị giải mã có thể gồm tập thông số trình tự (SPS) và phần đầu ảnh (PH).

SPS có thể gồm cờ được cho phép các biên ảo (sps_loop_filter_across_virtual_boundaries_disabled_flag). SPS có thể gồm cờ biểu diễn các biên ảo SPS (sps_loop_filter_across_virtual_boundaries_disabled_present_flag), dựa trên cờ được cho phép các biên ảo. Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, SPS có thể gồm cờ biểu diễn các biên ảo SPS. Dựa trên cờ được cho phép các biên ảo và cờ

biểu diễn các biên ảo SPS, SPS có thể gồm thông tin về số lượng của các biên ảo dọc SPS (sps_num_ver_virtual_boundaries), thông tin về vị trí các biên ảo dọc SPS (sps_virtual_boundaries_pos_x[i]), thông tin về số lượng của các biên ảo ngang SPS (sps_num_hor_virtual_boundaries), và thông tin về vị trí các biên ảo ngang SPS (sps_virtual_boundaries_pos_y[i]).

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 1, SPS có thể gồm thông tin về số lượng của các biên ảo ngang, thông tin về vị trí các biên ảo ngang, thông tin về số lượng của các biên ảo dọc, và thông tin về vị trí các biên ảo dọc. Trong một ví dụ, số lượng của các mẩu thông tin về vị trí các biên ảo ngang có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang, và số lượng của các mẩu thông tin về vị trí các biên ảo dọc có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc. Phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH, dựa trên cờ được cho phép các biên ảo.

Ví dụ, khi giá trị của cờ được cho phép các biên ảo là 1, phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo PH. Phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH (ph_num_ver_virtual_boundaries), dựa trên cờ biểu diễn các biên ảo PH và thông tin về số lượng của các biên ảo dọc SPS. Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1 và giá trị của thông tin về số lượng của các biên ảo dọc SPS là 0, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc PH. Trong một ví dụ, dựa trên thông tin về số lượng của các biên ảo dọc PH, phần đầu ảnh có thể gồm thông tin về giá trị delta của vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x_delta[i]) và thông tin về dấu của vị trí các biên ảo dọc PH (ph_virtual_boundaries_pos_x_sign[i]). Trong một ví dụ, dựa trên thông tin về số lượng của các biên ảo dọc PH, số lượng của các mẩu thông tin về giá trị delta vị trí các biên ảo dọc PH và số lượng của các mẩu thông tin về dấu của vị trí các biên ảo dọc PH có thể được xác định. Phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo ngang PH (ph_num_hor_virtual_boundaries), dựa trên cờ biểu diễn các biên ảo PH và thông tin về số lượng của các biên ảo ngang SPS.

Ví dụ, khi giá trị của cờ biểu diễn các biên ảo PH là 1 và giá trị của thông tin về số lượng của các biên ảo ngang SPS là 0, phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo ngang PH. Trong một ví dụ, dựa trên thông tin về số lượng của các biên

ảo ngang PH, phần đầu ảnh có thể gồm thông tin về giá trị delta của vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y_delta[i]) và thông tin về dấu của vị trí các biên ảo ngang PH (ph_virtual_boundaries_pos_y_sign[i]). Trong một ví dụ, dựa trên thông tin về số lượng của các biên ảo ngang PH, số lượng của các mẩu thông tin về giá trị delta vị trí các biên ảo ngang PH và số lượng của các mẩu thông tin về dấu của vị trí các biên ảo ngang PH có thể được xác định.

Theo các phương án của sáng chế cùng với các bảng nêu trên, thông tin để thực hiện việc lọc trong vòng lặp ngang qua các biên ảo có thể được báo hiệu một cách hiệu quả. Ví dụ, việc lọc trong vòng lặp có thể được thực hiện dựa trên việc báo hiệu của thông tin liên quan đến liệu việc lọc trong vòng lặp có được cho phép ngang qua các biên ảo hay không.

Fig.7 và Fig.8 thể hiện một cách sơ lược ví dụ của phương pháp mã hóa viđeo/hình ảnh và các thành phần liên quan theo (các) phương án của sáng chế.

Phương pháp được bộc lộ trong Fig.7 có thể được thực hiện bởi thiết bị mã hóa được bộc lộ trong Fig.2 hoặc Fig.8. Cụ thể là, ví dụ, S700 và S710 của Fig.7 có thể được thực hiện bởi bộ xử lý phần dư 230 của thiết bị mã hóa của Fig.8, S720 của Fig.7 có thể được thực hiện bởi bộ lọc 260 của thiết bị mã hóa của Fig.8, và S730 của Fig.7 có thể được thực hiện bởi bộ mã hóa entrôpi 240 của thiết bị mã hóa của Fig.8. Ngoài ra, mặc dù không được thể hiện trên Fig.7, các mẫu dự đoán hoặc thông tin liên quan đến việc dự đoán có thể được dẫn xuất bởi bộ dự đoán 220 của thiết bị mã hóa của Fig.7, và luồng bit có thể được tạo ra từ thông tin phần dư hoặc thông tin liên quan đến việc dự đoán bởi bộ mã hóa entrôpi 240 của thiết bị mã hóa. Phương pháp được bộc lộ trong Fig.7 có thể gồm các phương án được đề cập ở trên trong sáng chế.

Tham khảo đến Fig.7, thiết bị mã hóa có thể dẫn xuất các mẫu phần dư (S700). Thiết bị mã hóa có thể dẫn xuất các mẫu phần dư cho khối hiện tại, và các mẫu phần dư cho khối hiện tại có thể được dẫn xuất dựa trên các mẫu gốc và các mẫu dự đoán của khối hiện tại. Cụ thể là, thiết bị mã hóa có thể dẫn xuất các mẫu dự đoán của các khối hiện tại, dựa trên chế độ dự đoán. Trong trường hợp này, các phương pháp dự đoán khác nhau được bộc lộ trong sáng chế, chẳng hạn như việc dự đoán liên hoặc việc dự đoán nội, có thể được áp dụng. Các mẫu phần dư có thể được dẫn xuất dựa trên các mẫu dự đoán và các mẫu gốc.

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi. Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi dựa trên quy trình biến đổi cho các mẫu phần dư. Ví dụ, quy trình biến đổi có thể gồm ít nhất một kỹ thuật trong số biến đổi côsin rời rạc (DCT), biến đổi sin rời rạc (DST), biến đổi dựa trên đồ thị (GBT), và biến đổi phi tuyến tùy theo điều kiện (CNT).

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi được lượng tử hóa. Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi được lượng tử hóa dựa trên quy trình lượng tử hóa cho các hệ số biến đổi. Các hệ số biến đổi được lượng tử hóa có thể có dạng vectơ 1 chiều dựa trên thứ tự quét hệ số.

Thiết bị mã hóa có thể tạo ra thông tin phần dư (S710). Thiết bị mã hóa có thể tạo ra thông tin phần dư, dựa trên các hệ số biến đổi. Thiết bị mã hóa có thể tạo ra thông tin phần dư chỉ ra các hệ số biến đổi được lượng tử hóa. Thông tin phần dư có thể được tạo ra thông qua các phương pháp mã hóa khác nhau như là Golomb hàm mũ, CAVLC, CABAC, hoặc tương tự.

Thiết bị mã hóa có thể tạo ra các mẫu được xây dựng lại. Thiết bị mã hóa có thể tạo ra các mẫu được xây dựng lại, dựa trên thông tin phần dư. Các mẫu được xây dựng lại có thể được tạo ra bằng việc bổ sung mẫu dự đoán và các mẫu phần dư dựa trên thông tin phần dư. Cụ thể là, thiết bị mã hóa có thể thực hiện việc dự đoán (việc dự đoán nội hoặc liên) trên khối hiện tại, và có thể tạo ra các mẫu được xây dựng lại dựa trên các mẫu gốc và các mẫu dự đoán được tạo ra từ việc dự đoán.

Các mẫu được xây dựng lại có thể gồm các mẫu độ chói được xây dựng lại và các mẫu sắc độ được xây dựng lại. Cụ thể là, các mẫu phần dư có thể gồm các mẫu độ chói phần dư và các mẫu sắc độ phần dư. Các mẫu độ chói phần dư có thể được tạo ra dựa trên các mẫu độ chói gốc và các mẫu độ chói dự đoán. Các mẫu sắc độ phần dư có thể được tạo ra dựa trên các mẫu sắc độ gốc và các mẫu sắc độ dự đoán. Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi cho các mẫu độ chói phần dư (các hệ số biến đổi độ chói) và/hoặc các hệ số biến đổi cho các mẫu sắc độ phần dư (các hệ số biến đổi sắc độ). Các hệ số biến đổi được lượng tử hóa có thể gồm các hệ số biến đổi độ chói được lượng tử hóa và/hoặc các hệ số biến đổi sắc độ được lượng tử hóa.

Thiết bị mã hóa có thể tạo ra thông tin liên quan đến việc lọc trong vòng lặp cho

các mẫu được xây dựng lại (S720). Thiết bị mã hóa có thể thực hiện quy trình lọc trong vòng lặp trên các mẫu được xây dựng lại, và có thể tạo ra thông tin liên quan đến việc lọc trong vòng lặp, dựa trên quy trình lọc trong vòng lặp. Ví dụ, thông tin liên quan đến việc lọc trong vòng lặp có thể gồm thông tin được đề cập ở trên về các biên ảo (cờ được cho phép các biên ảo SPS, cờ được cho phép các biên ảo phần đầu ảnh, cờ biểu diễn các biên ảo SPS, cờ biểu diễn các biên ảo phần đầu ảnh, thông tin về các vị trí của các biên ảo, v.v.).

Thiết bị mã hóa có thể mã hóa thông tin viđeo/hình ảnh (S730). Thông tin hình ảnh có thể gồm thông tin phần dư, thông tin liên quan đến việc dự đoán, và/hoặc thông tin liên quan đến việc lọc trong vòng lặp. Thông tin viđeo/hình ảnh được mã hóa có thể được xuất dưới dạng luồng bit. Luồng bit có thể được truyền tới thiết bị giải mã thông qua mạng hoặc phương tiện lưu trữ.

Thông tin hình ảnh/viđeo có thể gồm nhiều thông tin theo một phương án của sáng chế. Ví dụ, hình ảnh/viđeo có thể gồm thông tin được bộc lộ trong ít nhất một trong số các bảng từ 1 đến 32 ở trên.

Trong một phương án, thông tin liên quan đến việc lọc trong vòng lặp có thể gồm cờ được cho phép các biên ảo SPS liên quan đến việc liệu quy trình lọc trong vòng lặp có được thực hiện ngang qua các biên ảo hay không. Ví dụ, liệu việc báo hiệu của thông tin liên quan đến các biên ảo có mặt trong SPS hay thông tin phần đầu ảnh có thể được xác định dựa trên cờ được cho phép các biên ảo SPS. Cờ được cho phép các biên ảo SPS có thể chỉ ra liệu có thể bất hoạt quy trình lọc trong vòng lặp ngang qua các biên ảo hay không.

Trong một phương án, thông tin hình ảnh có thể gồm tập thông số trình tự (SPS). SPS có thể gồm cờ được cho phép các biên ảo SPS và cờ biểu diễn các biên ảo SPS. Ngoài ra, việc liệu thông tin về các vị trí của các biên ảo và thông tin về số lượng của các biên ảo có được gồm trong SPS hay không có thể được xác định dựa trên cờ biểu diễn các biên ảo SPS.

Trong một phương án, SPS có thể gồm thông tin về số lượng của các biên ảo dọc, dựa vào việc giá trị của cờ biểu diễn các biên ảo SPS là 1.

Trong một phương án, SPS có thể gồm thông tin về các vị trí của các biên ảo

dọc. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo dọc có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc.

Trong một phương án, SPS có thể gồm thông tin về số lượng của các biên ảo ngang, dựa vào việc giá trị của cờ biểu diễn các biên ảo SPS là 1.

Trong một phương án, SPS có thể gồm thông tin về các vị trí của các biên ảo ngang. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo ngang có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang.

Trong một phương án, thông tin hình ảnh có thể gồm thông tin phần đầu ảnh. Ngoài ra, thông tin phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo phần đầu ảnh, dựa vào việc giá trị của cờ được cho phép các biên ảo SPS là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 0.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc, dựa vào việc giá trị của cờ biểu diễn các biên ảo phần đầu ảnh là 1.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về các vị trí của các biên ảo dọc. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo dọc có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo ngang, dựa vào việc giá trị của cờ biểu diễn các biên ảo phần đầu ảnh là 1.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về các vị trí của các biên ảo ngang. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo ngang có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang.

Trong một phương án, tổng của số lượng của các biên ảo dọc và số lượng của các biên ảo ngang có thể là lớn hơn 0, dựa vào việc SPS gồm thông tin về các vị trí của các biên ảo dọc và thông tin về các vị trí của các biên ảo ngang.

Trong một phương án, thông tin liên quan đến việc lọc trong vòng lặp (và/hoặc thông tin liên quan đến các biên ảo) có thể còn gồm cờ biểu diễn các biên ảo SPS, cờ biểu diễn các biên ảo phần đầu ảnh, và cờ được cho phép làm mới giải mã dần (GDR). Ví dụ, dựa vào việc giá trị của cờ được cho phép GDR là 1, giá trị của cờ được cho phép

các biên ảo SPS (cờ được cho phép các biên ảo) có thể là 1, giá trị của cờ biểu diễn các biên ảo SPS có thể là 0, và giá trị của cờ biểu diễn các biên ảo phần đầu ảnh có thể là 1 (việc báo hiệu của thông tin của các biên ảo có thể có mặt trong phần đầu ảnh).

Fig.9 và Fig.10 thể hiện một cách sơ lược ví dụ của phương pháp giải mã viđeo/hình ảnh và các thành phần liên quan theo (các) phương án của sáng chế.

Phương pháp được bộc lộ trong Fig.9 có thể được thực hiện bởi thiết bị giải mã được bộc lộ trong Fig.3 hoặc Fig.10. Cụ thể là, ví dụ, S900 của Fig.9 có thể được thực hiện bởi bộ giải mã entrôpi 310 của thiết bị giải mã, S910 có thể được thực hiện bởi bộ xử lý phần dư 320 và/hoặc bộ cộng 340 của thiết bị giải mã, và S920 có thể được thực hiện bởi bộ lọc 350 của thiết bị giải mã. Phương pháp được bộc lộ trong Fig.9 có thể gồm các phương án được đề cập ở trên trong sáng chế.

Tham chiếu đến Fig.9, thiết bị giải mã có thể nhận/thu nhận viđeo/thông tin hình ảnh (S900). Thông tin viđeo/hình ảnh có thể gồm thông tin phần dư, thông tin liên quan đến việc dự đoán, và/hoặc thông tin liên quan đến việc lọc trong vòng lặp. Thiết bị giải mã có thể nhận/thu nhận thông tin hình ảnh/viđeo thông qua luồng bit.

Thông tin hình ảnh/viđeo có thể gồm nhiều thông tin theo một phương án của sáng chế. Ví dụ, hình ảnh/viđeo có thể gồm thông tin được bộc lộ trong ít nhất một trong số các bảng từ 1 đến 32 ở trên.

Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi được lượng tử hóa, dựa trên thông tin phần dư. Các hệ số biến đổi được lượng tử hóa có thể có dạng vectơ 1 chiều dựa trên thứ tự quét hệ số. Các hệ số biến đổi được lượng tử hóa có thể gồm các hệ số biến đổi độ chói được lượng tử hóa và/hoặc các hệ số biến đổi sắc độ được lượng tử hóa.

Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi. Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi, dựa trên quy trình khử lượng tử hóa cho các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi độ chói thông qua việc khử lượng tử hóa, dựa trên các hệ số biến đổi độ chói được lượng tử hóa. Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi sắc độ thông qua việc khử lượng tử hóa, dựa trên các hệ số biến đổi sắc độ được lượng tử hóa.

Thiết bị giải mã có thể tạo ra/dẫn xuất các mẫu phần dư. Thiết bị giải mã có thể

dẫn xuất các mẫu phần dư, dựa trên thủ tục biến đổi ngược cho các hệ số biến đổi. Thiết bị giải mã có thể dẫn xuất các mẫu độ chói phần dư thông qua quy trình biến đổi ngược, dựa trên các hệ số biến đổi độ chói. Thiết bị giải mã có thể dẫn xuất các mẫu sắc độ phần dư thông qua biến đổi ngược, dựa trên các hệ số biến đổi sắc độ.

Thiết bị giải mã có thể tạo ra/dẫn xuất các mẫu được xây dựng lại (S910). Ví dụ, thiết bị giải mã có thể tạo ra/dẫn xuất các mẫu độ chói được xây dựng lại và/hoặc các mẫu sắc độ được xây dựng lại. Thiết bị giải mã có thể tạo ra các mẫu độ chói được xây dựng lại và/hoặc các mẫu sắc độ được xây dựng lại, dựa trên thông tin phần dư. Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại dựa trên thông tin phần dư. Các mẫu được xây dựng lại có thể gồm các mẫu độ chói được xây dựng lại và/hoặc các mẫu sắc độ được xây dựng lại. Thành phần độ chói của các mẫu được xây dựng lại có thể tương ứng với các mẫu độ chói được xây dựng lại, và thành phần sắc độ của các mẫu được xây dựng lại có thể tương ứng với các mẫu sắc độ được xây dựng lại. Thiết bị giải mã có thể tạo ra các mẫu độ chói dự đoán và/hoặc các mẫu sắc độ dự đoán thông qua quy trình dự đoán. Thiết bị giải mã có thể tạo ra các mẫu độ chói được xây dựng lại dựa trên các mẫu độ chói dự đoán và các mẫu độ chói phần dư. Thiết bị giải mã có thể tạo ra các mẫu sắc độ xây dựng lại dựa trên các mẫu sắc độ dự đoán và các mẫu sắc độ phần dư.

Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại (được lọc) được cải biến (S920). Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại được cải biến bằng việc thực hiện quy trình lọc trong vòng lặp cho các mẫu được xây dựng lại của ảnh hiện tại. Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại được cải biến, dựa trên thông tin liên quan đến việc lọc trong vòng lặp (và/hoặc thông tin liên quan đến các biên ảo). Thiết bị giải mã có thể sử dụng quy trình giải khối, quy trình SAO, và/hoặc quy trình ALF để tạo ra các mẫu được xây dựng lại được cải biến.

Trong một phương án, thông tin hình ảnh có thể gồm SPS. SPS có thể gồm cờ được cho phép các biên ảo SPS. Ngoài ra, quy trình lọc trong vòng lặp có thể được thực hiện ngang qua các biên ảo, dựa trên cờ cho phép các biên ảo SPS (hoặc có thể được thực hiện ngang qua các biên ảo). Ví dụ, liệu việc báo hiệu của thông tin liên quan đến các biên ảo có mặt trong SPS hay thông tin phần đầu ảnh có thể được xác định dựa trên cờ được cho phép các biên ảo SPS. Cờ được cho phép các biên ảo SPS có thể chỉ ra liệu có thể bất hoạt quy trình lọc trong vòng lặp ngang qua các biên ảo hay không.

Trong một phương án, SPS có thể còn gồm cờ biểu diễn các biên ảo SPS. Việc liệu thông tin về các vị trí của các biên ảo và thông tin về số lượng của các biên ảo có được gồm trong SPS hay không có thể được xác định dựa trên cờ biểu diễn các biên ảo SPS.

Trong một phương án, SPS có thể gồm thông tin về số lượng của các biên ảo dọc, dựa vào việc giá trị của cờ biểu diễn các biên ảo SPS là 1.

Trong một phương án, SPS có thể gồm thông tin về các vị trí của các biên ảo dọc. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo dọc có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc.

Trong một phương án, SPS có thể gồm thông tin về số lượng của các biên ảo ngang, dựa vào việc giá trị của cờ biểu diễn các biên ảo SPS là 1.

Trong một phương án, SPS có thể gồm thông tin về các vị trí của các biên ảo ngang. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo ngang có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang.

Trong một phương án, thông tin hình ảnh có thể gồm thông tin phần đầu ảnh. Ngoài ra, thông tin phần đầu ảnh có thể gồm cờ biểu diễn các biên ảo phần đầu ảnh, dựa vào việc giá trị của cờ được cho phép các biên ảo SPS là 1 và giá trị của cờ biểu diễn các biên ảo SPS là 0.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo dọc, dựa vào việc giá trị của cờ biểu diễn các biên ảo phần đầu ảnh là 1.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về các vị trí của các biên ảo dọc. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo dọc có thể được xác định dựa trên thông tin về số lượng của các biên ảo dọc.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về số lượng của các biên ảo ngang, dựa vào việc giá trị của cờ biểu diễn các biên ảo phần đầu ảnh là 1.

Trong một phương án, thông tin phần đầu ảnh có thể gồm thông tin về các vị trí của các biên ảo ngang. Ngoài ra, số lượng của các mẩu thông tin về các vị trí của các biên ảo ngang có thể được xác định dựa trên thông tin về số lượng của các biên ảo ngang.

Trong một phương án, tổng của số lượng của các biên ảo dọc và số lượng của các biên ảo ngang có thể là lớn hơn 0, dựa vào việc SPS gồm thông tin về các vị trí của các biên ảo dọc và thông tin về các vị trí của các biên ảo ngang.

Trong một phương án, thông tin liên quan đến việc lọc trong vòng lặp (và/hoặc thông tin liên quan đến các biên ảo) có thể còn gồm cờ biểu diễn các biên ảo SPS, cờ biểu diễn các biên ảo phần đầu ảnh, và cờ được cho phép làm mới giải mã dần (GDR). Ví dụ, dựa vào việc giá trị của cờ được cho phép GDR là 1, giá trị của cờ được cho phép các biên ảo SPS (cờ được cho phép các biên ảo) có thể là 1, giá trị của cờ biểu diễn các biên ảo SPS có thể là 0, và giá trị của cờ biểu diễn các biên ảo phần đầu ảnh có thể là 1 (việc báo hiệu của thông tin của các biên ảo có thể có mặt trong phần đầu ảnh).

Trong sự có mặt của mẫu phần dư cho khối hiện tại, thiết bị giải mã có thể nhận thông tin phần dư cho khối hiện tại. Thông tin phần dư có thể gồm hệ số biến đổi cho các mẫu phần dư. Thiết bị giải mã có thể dẫn xuất các mẫu phần dư (hoặc mảng mẫu phần dư) cho khối hiện tại, dựa trên thông tin phần dư. Cụ thể là, thiết bị giải mã có thể dẫn xuất các hệ số biến đổi được lượng tử hóa, dựa trên thông tin phần dư. Các hệ số biến đổi được lượng tử hóa có thể có dạng vectơ 1 chiều dựa trên thứ tự quét hệ số. Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi, dựa trên quy trình khử lượng tử hóa cho các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể dẫn xuất các mẫu phần dư, dựa trên các hệ số biến đổi.

Thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại, dựa trên các mẫu dự đoán (nội) và các mẫu phần dư, và có thể dẫn xuất khối được xây dựng lại hoặc ảnh được xây dựng lại, dựa trên các mẫu được xây dựng lại. Cụ thể là, thiết bị giải mã có thể tạo ra các mẫu được xây dựng lại, dựa trên tổng giữa các mẫu dự đoán (nội) và các mẫu phần dư. Sau đó, như được mô tả ở trên, thiết bị giải mã có thể áp dụng tùy chọn quy trình lọc trong vòng lặp chẳng hạn như việc lọc giải khối và/hoặc quy trình SAO cho ảnh được xây dựng lại để cải thiện chất lượng hình ảnh chủ quan/khách quan.

Ví dụ, thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm tất cả hoặc các phần của các mẩu thông tin được mô tả ở trên (hoặc các phần tử cú pháp) bằng việc giải mã luồng bit hoặc thông tin được mã hóa. Ngoài ra, luồng bit hoặc thông tin được mã hóa có thể được lưu trữ trong phương tiện lưu trữ đọc được bởi máy tính, và có thể khiến phương pháp giải mã được mô tả ở trên được thực hiện.

Mặc dù các phương pháp đã được mô tả trên cơ sở lưu đồ trong đó các bước hoặc các khối được liệt kê theo trình tự trong các phương án được mô tả ở trên, các bước của sáng chế không bị giới hạn ở thứ tự cụ thể, và bước nhất định có thể được thực hiện trong bước khác hoặc theo thứ tự khác hoặc đồng thời với bước được mô tả ở trên. Ngoài ra, sẽ được hiểu bởi người có trình độ trung bình trong lĩnh vực rằng các bước của các lưu đồ là không bắt buộc, và bước khác có thể được gồm trong đó hoặc một hoặc nhiều bước trong lưu đồ có thể được xóa mà không tạo nên sự ảnh hưởng đến phạm vi của sáng chế.

Phương pháp được đề cập ở trên theo sáng chế có thể có dạng phần mềm, và thiết bị mã hóa và/hoặc thiết bị giải mã theo sáng chế có thể được gồm trong thiết bị để thực hiện xử lý hình ảnh, ví dụ, TV, máy tính, điện thoại thông minh, hộp thu tín hiệu truyền hình, thiết bị hiển thị, hoặc tương tự.

Khi các phương án của sáng chế được thực hiện bởi phần mềm, phương pháp được đề cập ở trên có thể được thực hiện bởi môđun (quy trình hoặc chức năng) mà thực hiện chức năng được đề cập ở trên. Môđun có thể được lưu trữ trong bộ nhớ và được thực thi bởi bộ xử lý. Bộ nhớ có thể được lắp đặt phía trong hoặc phía ngoài bộ xử lý và có thể được kết nối với bộ xử lý thông qua các phương tiện đã biết khác nhau. Bộ xử lý có thể gồm mạch tích hợp ứng dụng cụ thể (Application-Specific Integrated Circuit, ASIC), các bộ chip khác, mạch logic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ có thể gồm bộ nhớ chỉ đọc (Read-Only Memory, ROM), bộ nhớ truy cập ngẫu nhiên (Random Access Memory, RAM), bộ nhớ tác động nhanh, thẻ nhớ, phương tiện lưu trữ, và/hoặc thiết bị lưu trữ khác. Nói cách khác, các phương án theo sáng chế có thể được thực hiện và được thực thi trên bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc chip. Ví dụ, các đơn vị chức năng được minh họa trên các hình vẽ tương ứng có thể được thực hiện và được thực thi trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc chip. Trong trường hợp này, thông tin về việc thực hiện (ví dụ, thông tin về các lệnh) hoặc các thuật toán có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số.

Ngoài ra, thiết bị giải mã và thiết bị mã hóa mà cho nó (các) phương án của sáng chế được áp dụng có thể được gồm trong bộ thu phát phát rộng đa phương tiện, đầu cuối truyền thông di động, thiết bị viđeo điện ảnh tại nhà, thiết bị viđeo điện ảnh kỹ thuật số, camera giám sát, thiết bị trò chuyện viđeo, và thiết bị truyền thông thời gian thực chẳng

hạn như truyền thông viđeo, thiết bị phát luồng di động, phương tiện lưu trữ, máy quay viđeo, nhà cung cấp dịch vụ viđeo theo yêu cầu (video on demand, VoD), thiết bị cung cấp dịch vụ nội dung trên nền mạng viễn thông (Over The Top, OTT), thiết bị cung cấp dịch vụ tạo luồng Internet, thiết bị viđeo 3D, thiết bị thực tế ảo (Virtual Reality, VR), thiết bị thực tế tăng cường (Augment Reality, AR), thiết bị viđeo điện thoại hình ảnh, đầu cuối phương tiện (ví dụ, đầu cuối phương tiện (gồm phương tiện tự động), đầu cuối máy bay, hoặc đầu cuối tàu), và thiết bị viđeo y tế; và có thể được sử dụng để xử lý tín hiệu hoặc dữ liệu hình ảnh. Ví dụ, thiết bị viđeo OTT có thể gồm máy chơi trò chơi, máy đọc phát Bluray, TV được kết nối Internet, hệ thống rạp hát gia đình, điện thoại thông minh, PCT máy tính bảng, và đầu ghi viđeo kỹ thuật số (Digital Video Recorder, DVR).

Ngoài ra, phương pháp xử lý mà cho nó (các) phương án của sáng chế được áp dụng có thể được tạo ra dưới dạng chương trình được thực thi bởi máy tính và có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính. Dữ liệu đa phương tiện có cấu trúc dữ liệu theo (các) phương án của sáng chế cũng có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính. Phương tiện ghi đọc được bởi máy tính gồm tất cả các loại thiết bị lưu trữ và các thiết bị lưu trữ được phân phối trong đó dữ liệu đọc được bởi máy tính được lưu trữ. Phương tiện ghi đọc được bởi máy tính có thể gồm, ví dụ, đĩa Bluray (Bluray Disc, BD), bus nối tiếp vạn năng (universal serial bus, USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang. Phương tiện ghi đọc được bởi máy tính cũng gồm phương tiện được triển khai dưới dạng sóng mang (ví dụ, truyền qua Internet). Ngoài ra, luồng bit được tạo ra bởi phương pháp mã hóa có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính hoặc được truyền thông qua mạng truyền thông có dây hoặc không dây.

Ngoài ra, (các) phương án của sáng chế có thể được triển khai dưới dạng sản phẩm chương trình máy tính dựa trên mã chương trình, và mã chương trình có thể được thực thi trên máy tính theo (các) phương án của sáng chế. Mã chương trình có thể được lưu trữ trên vật mang đọc được bởi máy tính.

Fig.11 biểu diễn ví dụ của hệ thống phát luồng nội dung mà cho nó phương án của sáng chế có thể được áp dụng.

Tham khảo đến Fig.11, hệ thống phát luồng nội dung mà cho nó các phương án của sáng chế được áp dụng thường có thể gồm máy chủ mã hóa, máy chủ phát luồng,

máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị nhập đa phương tiện.

Máy chủ mã hóa thực hiện chức năng nén tới dữ liệu kỹ thuật số các nội dung được nhập từ các thiết bị đầu vào đa phương tiện, chẳng hạn như điện thoại thông minh, camera, máy quay viđeo và tương tự, để tạo ra luồng bit, và để truyền nó vào máy chủ phát luồng. Theo một ví dụ khác, trong trường hợp trong đó thiết bị đầu vào đa phương tiện, chẳng hạn như, điện thoại thông minh, camera, máy quay viđeo hoặc tương tự, trực tiếp tạo ra luồng bit, máy chủ mã hóa có thể được bỏ qua.

Luồng bit có thể được tạo ra bằng phương pháp mã hóa hoặc phương pháp tạo ra luồng bit mà cho nó các phương án của sáng chế được áp dụng. Và máy chủ phát luồng có thể lưu trữ tạm thời luồng bit trong quy trình truyền hoặc nhận luồng bit.

Máy chủ phát luồng truyền dữ liệu đa phương tiện tới thiết bị người dùng trên cơ sở của yêu cầu của người dùng thông qua máy chủ web, mà đóng vai trò là phương tiện mà thông tin cho người dùng về việc dịch vụ là gì. Khi người dùng yêu cầu dịch vụ mà người dùng muốn, máy chủ web truyền yêu cầu tới máy chủ phát luồng, và máy chủ phát luồng truyền dữ liệu đa phương tiện tới người dùng. Theo đó, hệ thống phát luồng nội dung có thể gồm máy chủ điều khiển tách biệt, và trong trường hợp này, máy chủ điều khiển đóng vai trò điều khiển các lệnh/các phản hồi giữa thiết bị tương ứng trong hệ thống phát luồng nội dung.

Máy chủ phát luồng có thể nhận các nội dung từ bộ lưu trữ phương tiện và/hoặc máy chủ mã hóa. Ví dụ, trong trường hợp các nội dung được nhận từ máy chủ mã hóa, các nội dung có thể được nhận trong thời gian thực. Trong trường hợp này, máy chủ phát luồng có thể lưu trữ luồng bit trong khoảng thời gian được xác định trước để cung cấp dịch vụ phát luồng trơn tru.

Ví dụ, thiết bị người dùng có thể gồm điện thoại di động, điện thoại thông minh, máy tính xách tay, thiết bị đầu cuối phát rộng kỹ thuật số, trợ lý kỹ thuật số cá nhân (personal digital assistant, PDA), máy phát đa phương tiện cầm tay (portable multimedia player, PMP), bộ điều hướng, PC dạng tấm, PCT máy tính bảng, máy tính ultrabook, thiết bị đeo được (ví dụ, đầu cuối kiểu đồng hồ (đồng hồ thông minh), đầu cuối kiểu kính (kính thông minh), bộ hiển thị gắn trên đầu (head mounted display, HMD)), TV kỹ

thuật số, máy tính để bàn, biển báo kỹ thuật số hoặc tương tự.

Mỗi trong số các máy chủ trong hệ thống phát luồng nội dung có thể được thao tác như máy chủ được phân phối, và trong trường hợp này, dữ liệu được nhận bởi mỗi máy chủ có thể được xử lý theo cách được phân phối.

Các điểm yêu cầu bảo hộ trong phần mô tả này có thể được kết hợp theo cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật trong các điểm yêu cầu bảo hộ phương pháp của phần mô tả này có thể được kết hợp để được thực hiện hoặc được triển khai trong thiết bị, và các dấu hiệu kỹ thuật trong các điểm yêu cầu bảo hộ thiết bị có thể được kết hợp để được thực hiện hoặc được triển khai trong phương pháp. Ngoài ra, các dấu hiệu kỹ thuật trong (các) điểm yêu cầu bảo hộ phương pháp và (các) điểm yêu cầu bảo hộ thiết bị có thể được kết hợp để được thực hiện hoặc được triển khai trong thiết bị. Ngoài ra, các dấu hiệu kỹ thuật trong (các) điểm yêu cầu bảo hộ phương pháp và (các) điểm yêu cầu bảo hộ thiết bị có thể được kết hợp để được thực hiện hoặc được triển khai trong phương pháp.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã, phương pháp này bao gồm các bước:

thu nhận thông tin hình ảnh gồm thông tin phần dư thông qua luồng bit;

tạo ra các mẫu được xây dựng lại của ảnh hiện tại, dựa trên thông tin phần dư; và

tạo ra các mẫu được xây dựng lại được cải biến, dựa trên quy trình lọc trong vòng lặp cho các mẫu được xây dựng lại,

trong đó thông tin hình ảnh gồm tập thông số trình tự (sequence parameter set, SPS),

trong đó SPS gồm cờ được cho phép các biên ảo SPS,

trong đó SPS ngoài ra còn gồm cờ biểu diễn các biên ảo SPS dựa trên giá trị của cờ được cho phép các biên ảo SPS, và

trong đó việc liệu quy trình lọc trong vòng lặp có được thực hiện ngang qua các biên ảo hay không được xác định, dựa trên cờ được cho phép các biên ảo SPS.

2. Phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa, phương pháp này bao gồm các bước:

tạo ra các mẫu phần dư cho khối hiện tại;

tạo ra thông tin phần dư, dựa trên các mẫu phần dư cho khối hiện tại;

tạo ra thông tin liên quan đến việc lọc trong vòng lặp cho các mẫu được xây dựng lại của ảnh hiện tại; và

mã hóa thông tin hình ảnh gồm thông tin phần dư và thông tin liên quan đến việc lọc trong vòng lặp,

trong đó thông tin hình ảnh gồm tập thông số trình tự (SPS),

trong đó SPS gồm cờ được cho phép các biên ảo SPS,

trong đó SPS ngoài ra còn gồm cờ biểu diễn các biên ảo SPS dựa trên giá trị của cờ được cho phép các biên ảo SPS, và

trong đó thông tin liên quan đến việc lọc trong vòng lặp gồm cờ được cho phép

các biên ảo SPS liên quan đến việc liệu quy trình lọc trong vòng lặp có được thực hiện ngang qua các biên ảo hay không.

3. Phương pháp truyền dữ liệu cho hình ảnh, phương pháp này bao gồm các bước:

thu nhận luồng bit cho hình ảnh, trong đó luồng bit được tạo ra dựa trên việc tạo ra các mẫu phần dư cho khối hiện tại, tạo ra thông tin phần dư, dựa trên các mẫu phần dư cho khối hiện tại, tạo ra thông tin liên quan đến việc lọc trong vòng lặp cho các mẫu được xây dựng lại của ảnh hiện tại, và mã hóa thông tin hình ảnh gồm thông tin phần dư và thông tin liên quan đến việc lọc trong vòng lặp; và

truyền dữ liệu bao gồm luồng bit,

trong đó thông tin hình ảnh gồm tập thông số trình tự (SPS),

trong đó SPS gồm cờ được cho phép các biên ảo SPS,

trong đó SPS ngoài ra còn gồm cờ biểu diễn các biên ảo SPS dựa trên giá trị của cờ được cho phép các biên ảo SPS, và

trong đó thông tin liên quan đến việc lọc trong vòng lặp gồm cờ được cho phép các biên ảo SPS liên quan đến việc liệu quy trình lọc trong vòng lặp có được thực hiện ngang qua các biên ảo hay không.

FIG. 1

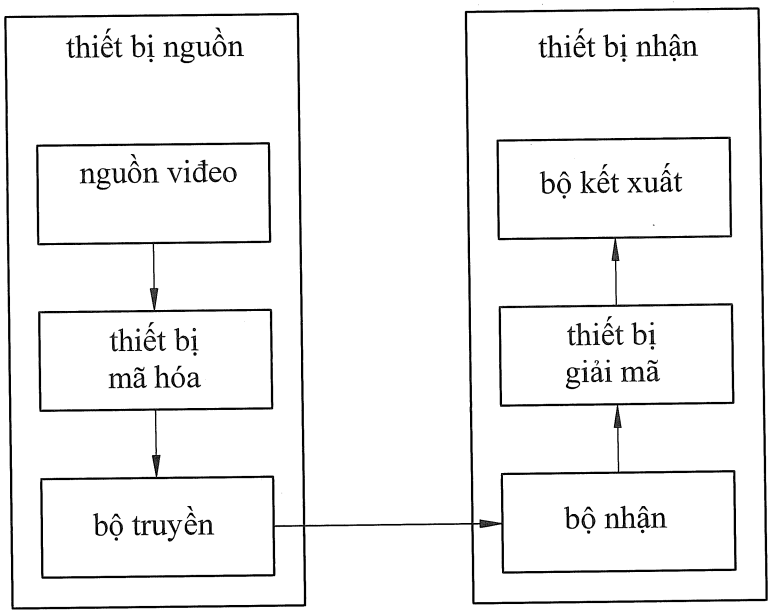

thiết bị nguồn
nguồn viđeo
thiết bị
mã hóa
bộ truyền
thiết bị nhận
bộ kết xuất
thiết bị
giải mã
bộ nhận

FIG. 2

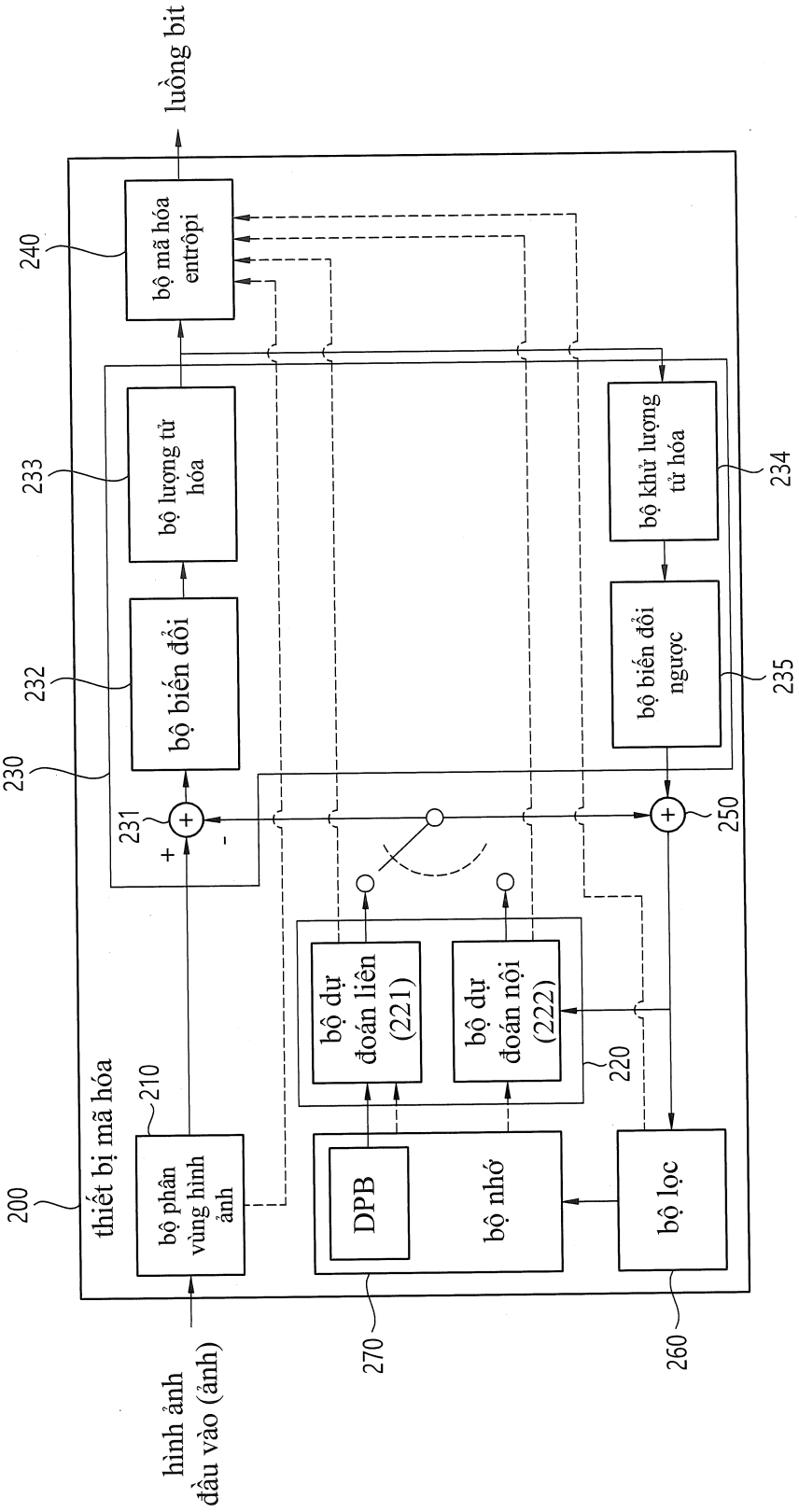

luồng bit
240
bộ mã hóa entrôpi
233
bộ lượng tử hóa
234
bộ khử lượng tử hóa
232
bộ biến đổi
235
bộ biến đổi ngược
230
231
+
-
250
bộ dự đoán liên (221)
bộ dự đoán nội (222)
220
200
thiết bị mã hóa
210
bộ phân vùng hình ảnh
DPB
bộ nhớ
bộ lọc
hình ảnh đầu vào (ảnh)
270
260

FIG. 3

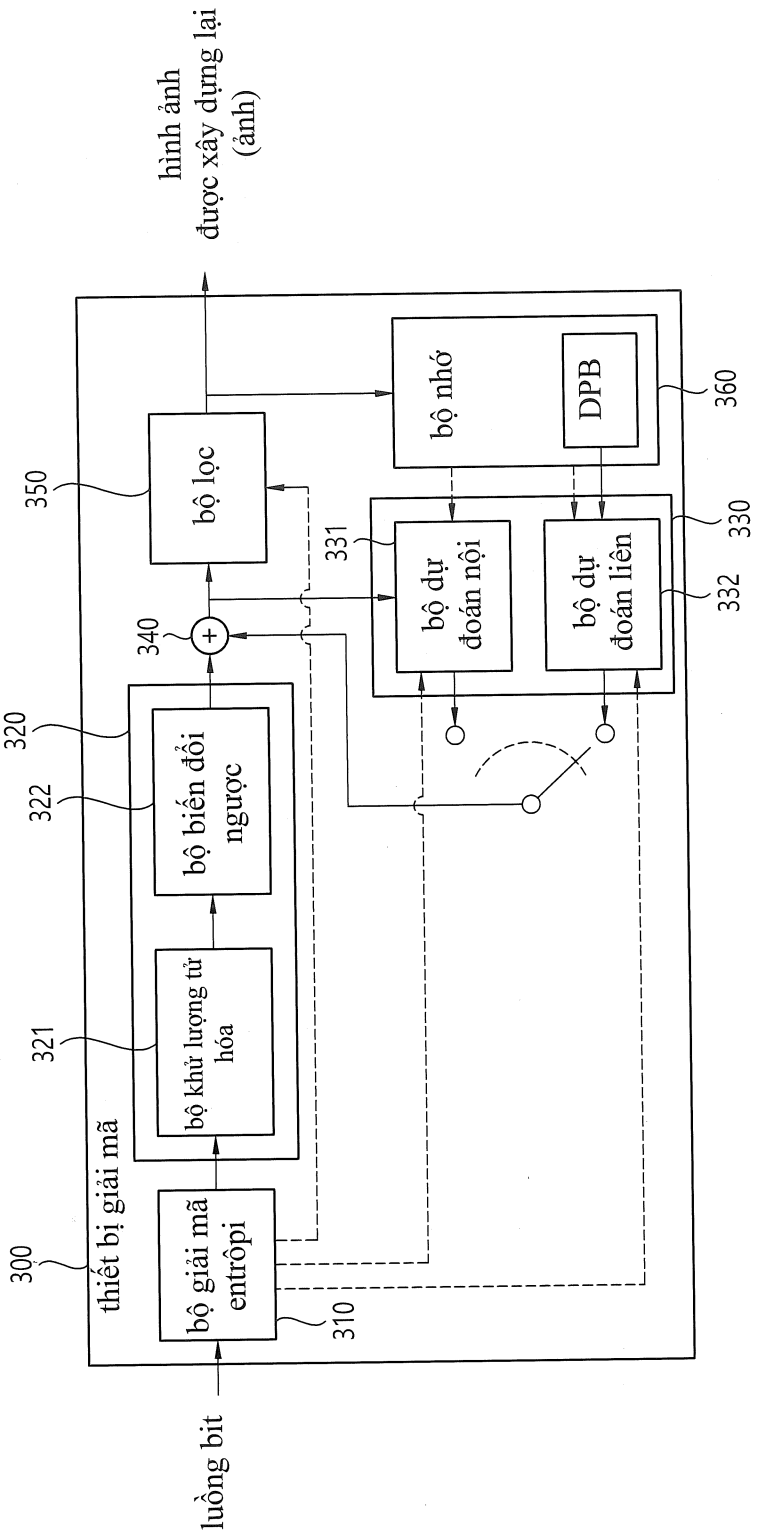

luồng bit
thiết bị giải mã
300
bộ giải mã entrôpi
310
bộ khử lượng tử hóa
321
bộ biến đổi ngược
322
320
340
bộ lọc
350
hình ảnh được xây dựng lại (ảnh)
bộ nhớ
DPB
360
bộ dự đoán nội
331
bộ dự đoán liên
332
330

FIG. 4

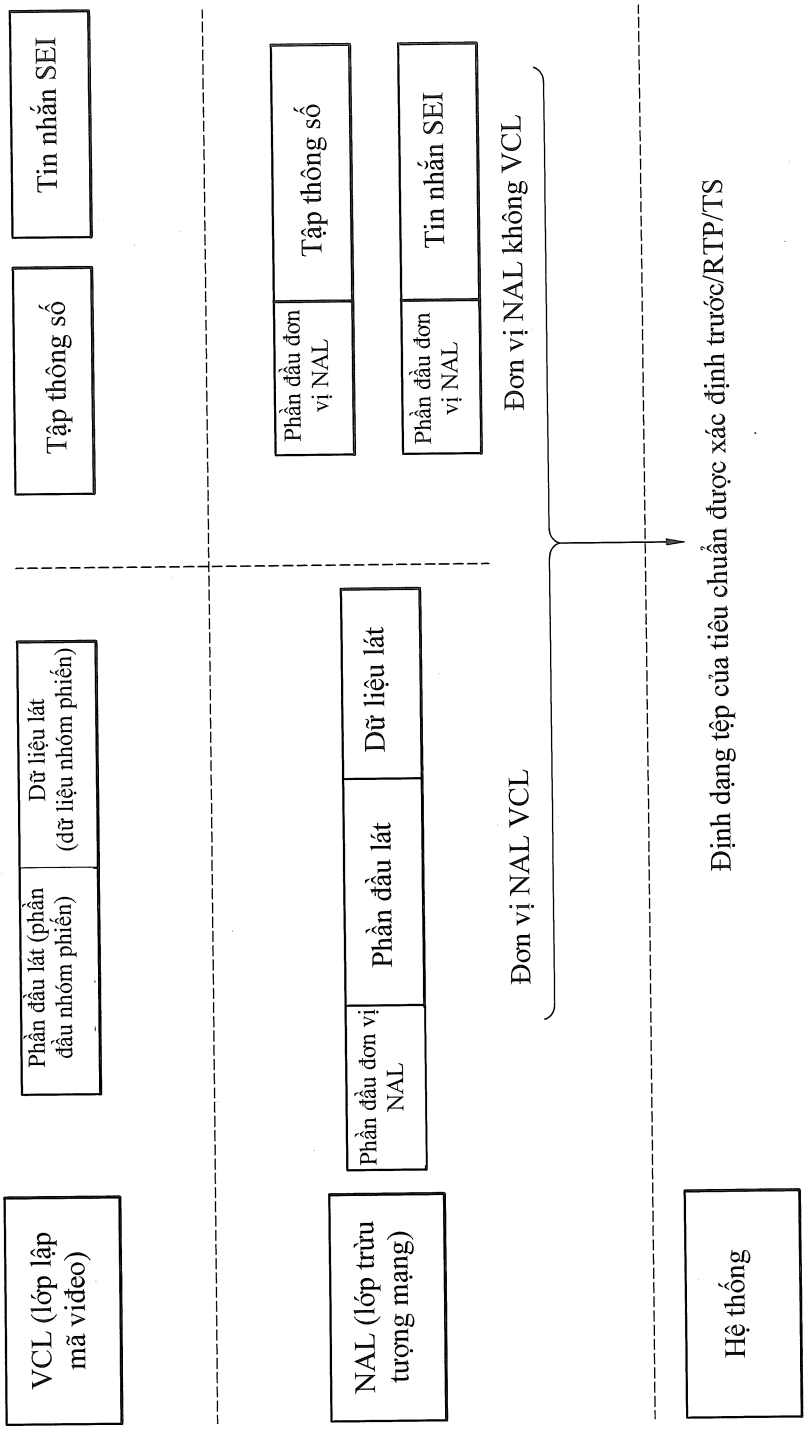

VCL (lớp lập mã video)
NAL (lớp trừu tượng mạng)
Hệ thống
Phần đầu lát (phần đầu nhóm phiến)
Dữ liệu lát (dữ liệu nhóm phiến)
Tập thông số
Tin nhắn SEI
Phần đầu đơn vị NAL
Phần đầu lát
Dữ liệu lát
Phần đầu đơn vị NAL
Tập thông số
Phần đầu đơn vị NAL
Tin nhắn SEI
Đơn vị NAL VCL
Đơn vị NAL không VCL
Định dạng tệp của tiêu chuẩn được xác định trước/RTP/TS

FIG. 5

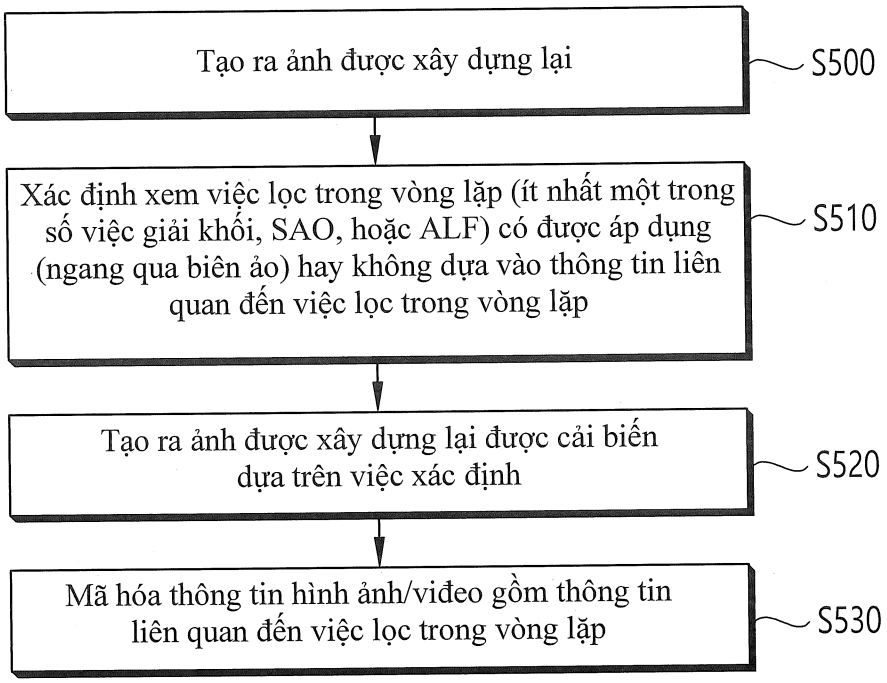

Tạo ra ảnh được xây dựng lại
S500
Xác định xem việc lọc trong vòng lặp (ít nhất một trong số việc giải khối, SAO, hoặc ALF) có được áp dụng (ngang qua biên ảo) hay không dựa vào thông tin liên quan đến việc lọc trong vòng lặp
S510
Tạo ra ảnh được xây dựng lại được cải biến dựa trên việc xác định
S520
Mã hóa thông tin hình ảnh/viđeo gồm thông tin liên quan đến việc lọc trong vòng lặp
S530

FIG. 6

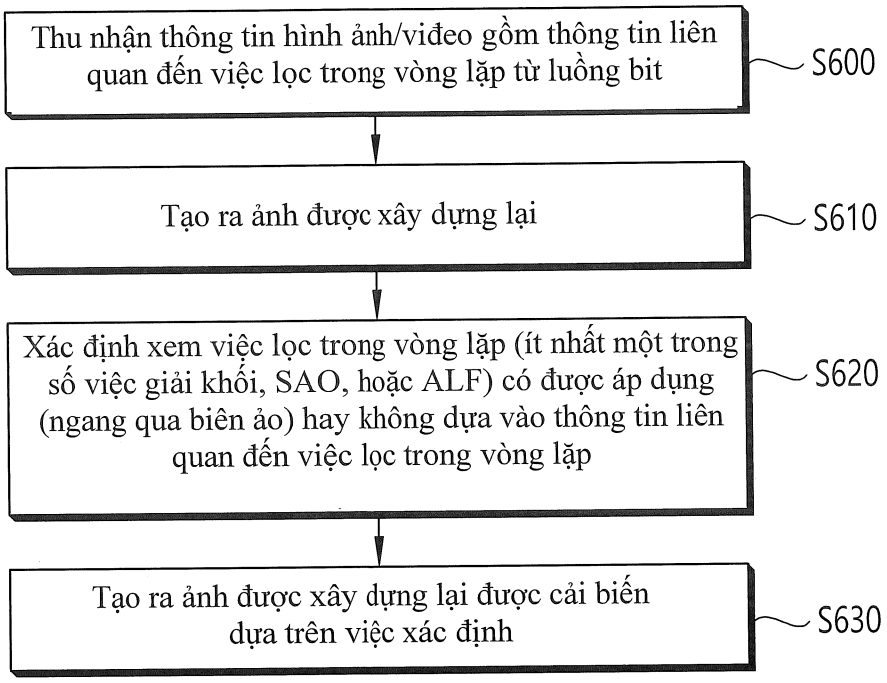

Thu nhận thông tin hình ảnh/viđeo gồm thông tin liên quan đến việc lọc trong vòng lặp từ luồng bit
S600
Tạo ra ảnh được xây dựng lại
S610
Xác định xem việc lọc trong vòng lặp (ít nhất một trong số việc giải khối, SAO, hoặc ALF) có được áp dụng (ngang qua biên ảo) hay không dựa vào thông tin liên quan đến việc lọc trong vòng lặp
S620
Tạo ra ảnh được xây dựng lại được cải biến dựa trên việc xác định
S630

FIG. 7

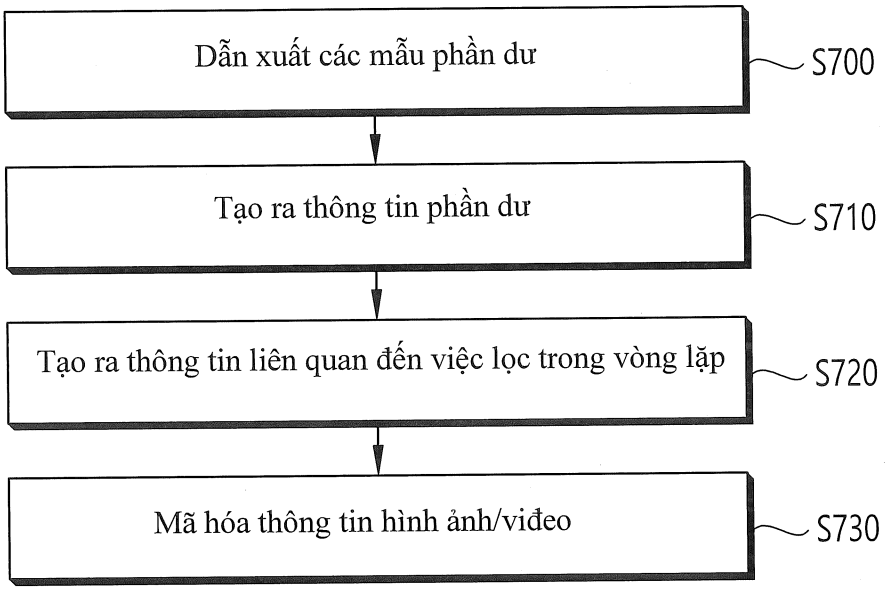

Dẫn xuất các mẫu phần dư
S700
Tạo ra thông tin phần dư
S710
Tạo ra thông tin liên quan đến việc lọc trong vòng lặp
S720
Mã hóa thông tin hình ảnh/viđeo
S730

FIG. 8

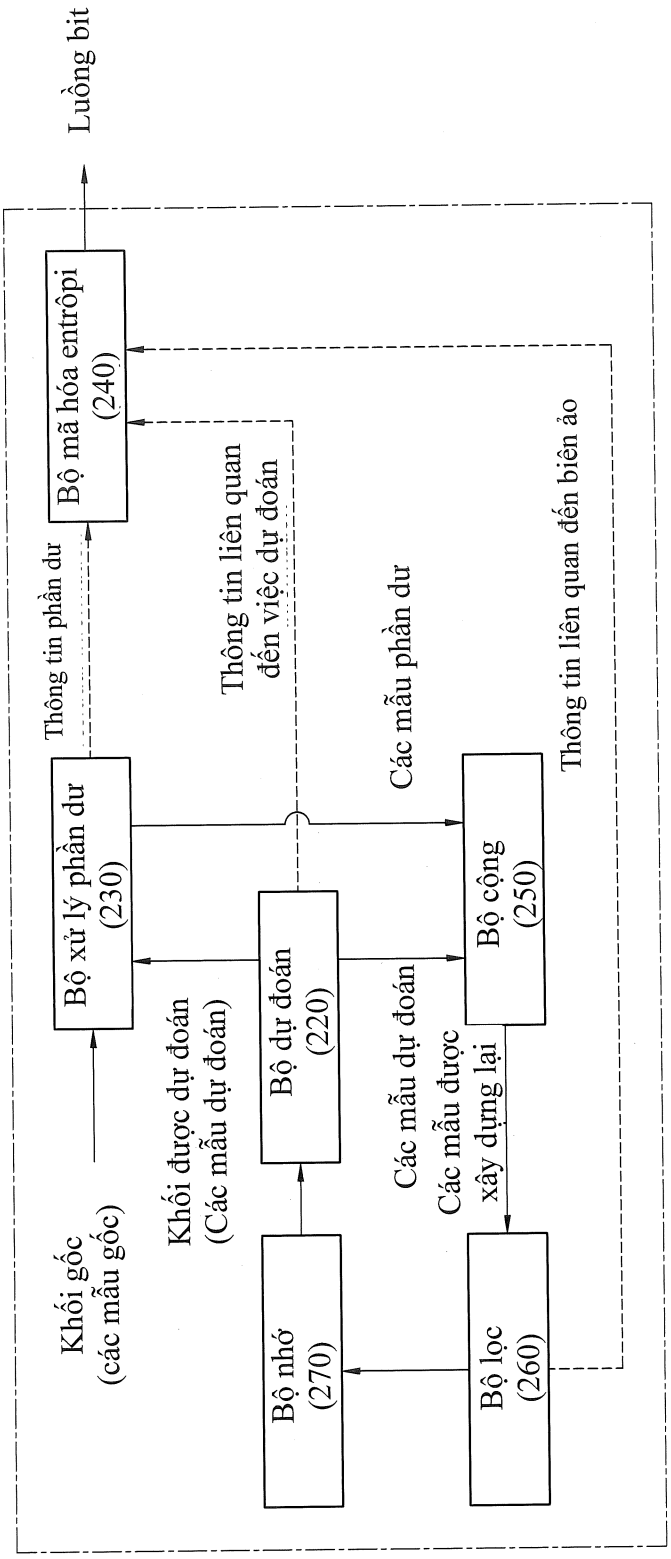

Luồng bit
Bộ mã hóa entrôpi (240)
Thông tin phần dư
Thông tin liên quan đến việc dự đoán
Các mẫu phần dư
Thông tin liên quan đến biên ảo
Bộ xử lý phần dư (230)
Bộ dự đoán (220)
Bộ cộng (250)
Khối gốc (các mẫu gốc)
Khối được dự đoán (Các mẫu dự đoán)
Các mẫu dự đoán
Các mẫu được xây dựng lại
Bộ nhớ (270)
Bộ lọc (260)
Thiết bị mã hóa (200)

FIG. 9

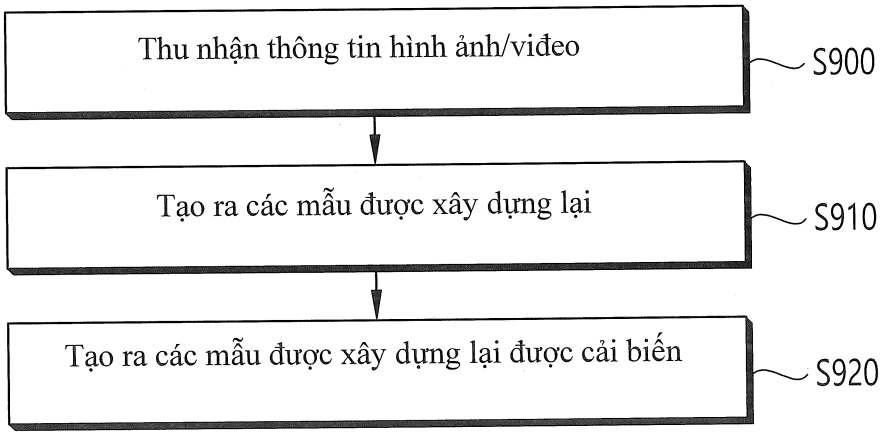

Thu nhận thông tin hình ảnh/viđeo
S900
Tạo ra các mẫu được xây dựng lại
S910
Tạo ra các mẫu được xây dựng lại được cải biến
S920

FIG. 10

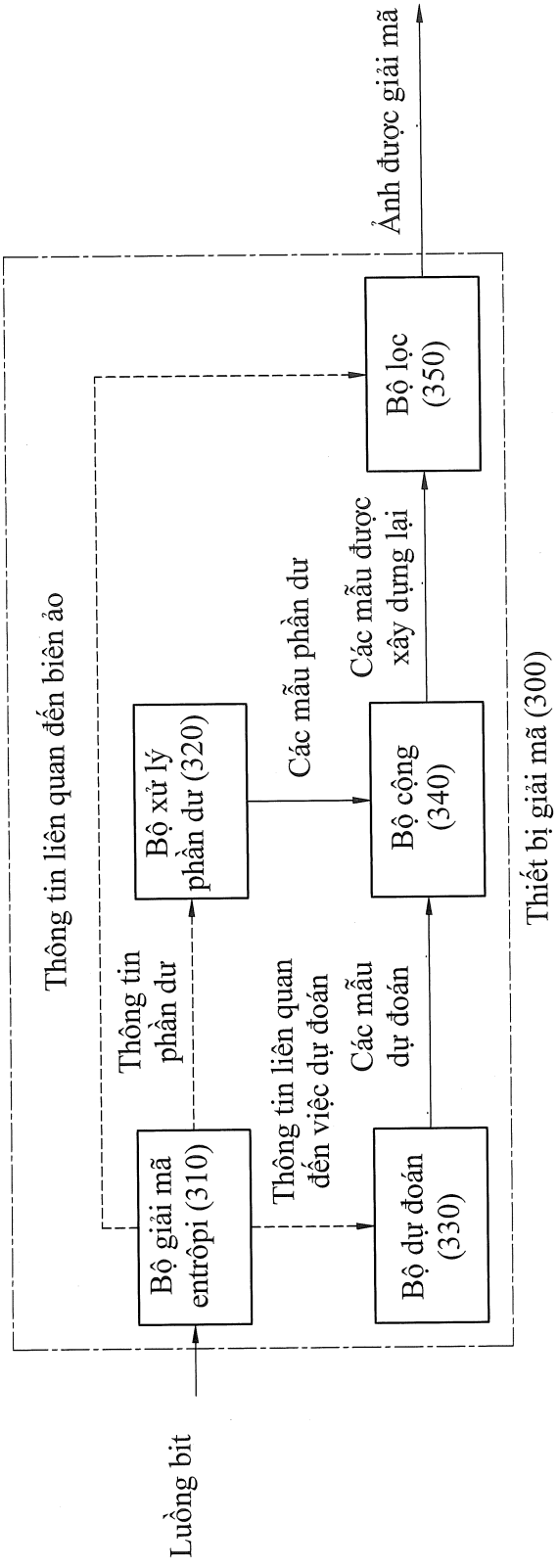
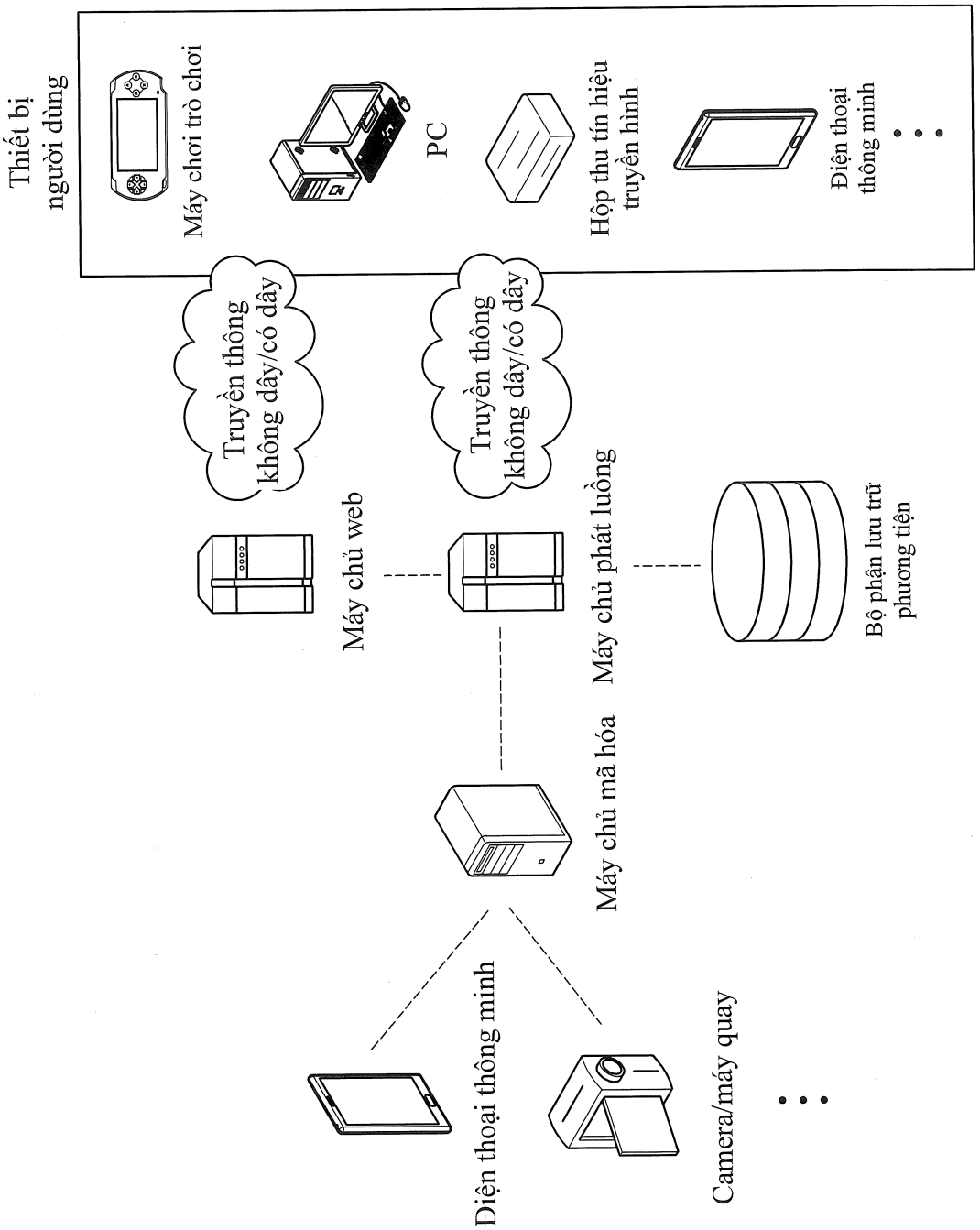

Luồng bit
Thông tin liên quan đến biên ảo
Bộ giải mã entrôpi (310)
Thông tin phần dư
Thông tin liên quan đến việc dự đoán
Các mẫu dự đoán
Bộ xử lý phần dư (320)
Các mẫu phần dư
Bộ dự đoán (330)
Bộ cộng (340)
Các mẫu được xây dựng lại
Bộ lọc (350)
Ảnh được giải mã
Thiết bị giải mã (300)

FIG. 11


Thiết bị người dùng
Máy chơi trò chơi
PC
Hộp thu tín hiệu truyền hình
Điện thoại thông minh
...
Truyền thông không dây/có dây
Truyền thông không dây/có dây
Máy chủ web
Máy chủ phát luồng
Bộ phận lưu trữ phương tiện
Máy chủ mã hóa
Điện thoại thông minh
Camera/máy quay
...