



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

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

(11)



1-0052147

(51)^{2020.01} H04N 19/103; H04N 19/124; H04N 19/186; H04N 19/136; H04N 19/157; H04N 19/176; H04N 19/12; H04N 19/132 (13) B

(21) 1-2022-01588

(22) 26/08/2020

(86) PCT/KR2020/011388 26/08/2020

(87) WO2021/040402 04/03/2021

(30) 62/891,948 26/08/2019 US

(45) 27/10/2025 451

(43) 25/07/2022 412A

(73) LG ELECTRONICS INC. (KR)

128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Republic of Korea

(72) ZHAO, Jie (US); PALURI, Seethal (IN); KIM, Seunghwan (KR).

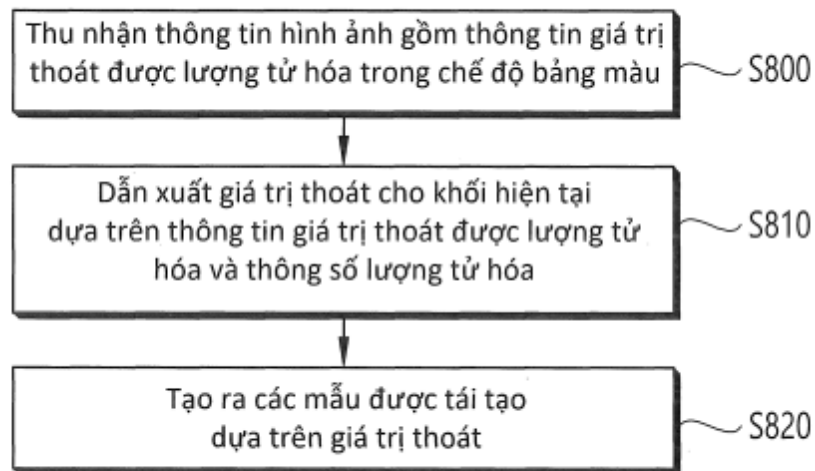
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ VÀ MÃ HÓA HÌNH ẢNH, VÀ PHƯƠNG PHÁP TRUYỀN LUỒNG BIT

(21) 1-2022-01588

(57) Sáng chế đề cập đến phương pháp giải mã và mã hóa hình ảnh, phương tiện lưu trữ đọc được bởi máy tính không chuyển tiếp. Theo sự bộc lộ của sáng chế, thông số lượng tử hóa cho giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể được dẫn xuất trên cơ sở của thông tin thông số lượng tử hóa tối thiểu về chế độ bỏ qua chuyển đổi để thực hiện quy trình định cỡ trên giá trị thoát được lượng tử hóa. Thông qua đó, lượng dữ liệu mà cần được phát tín hiệu cho tạo mã chế độ bảng màu có thể được giảm, và hiệu quả của việc tạo mã thoát trong chế độ bảng màu có thể được tăng cường.

FIG.8



Lĩnh vực kỹ thuật được đề cập

Sáng chế liên quan đến việc tạo mã video hoặc hình ảnh và, ví dụ, đến kỹ thuật tạo mã video hoặc hình ảnh dựa trên việc tạo mã bảng màu thoát.

Tình trạng kỹ thuật của sáng chế

Hiện tại, nhu cầu về hình ảnh/video độ phân giải cao, chất lượng cao như hình ảnh/video 4K, 8K hoặc độ phân giải siêu cao (Ultra High Definition, UHD) khác đang gia tăng trong nhiều lĩnh vực khác nhau. Vì độ phân giải hoặc chất lượng hình ảnh/video ngày càng trở nên cao hơn, nên lượng thông tin hoặc bit sẽ được truyền tương đối lớn hơn so với dữ liệu hình ảnh/video truyền thống. Do đó, nếu dữ liệu hình ảnh/video được truyền thông qua môi trường như đường truyền băng rộng có dây/không dây hoặc được lưu trữ trong phương tiện lưu trữ có tính kế thừa, thì chi phí truyền và lưu trữ có thể tăng một cách đáng kể.

Hơn nữa, sự quan tâm và nhu cầu hiện đang gia tăng với các nội dung thực tế ảo (Virtual Reality, VR) và thực tế nhân tạo (Artificial Reality, AR), và phương tiện nhập vai như toàn ảnh (hologram); và việc phát rộng của các hình ảnh/các video thể hiện các đặc tính hình ảnh/video khác với các đặc tính của các hình ảnh/video thực tế, như các hình ảnh/các video cho trò chơi, hiện cũng gia tăng.

Do đó, yêu cầu có kỹ thuật nén hình ảnh/video hiệu quả để nén và truyền, lưu trữ, hoặc phát lại một cách hiệu quả các hình ảnh/video có độ phân giải cao, chất lượng cao, thể hiện các đặc tính khác nhau như được mô tả ở trên.

Hơn thế nữa, có sự thảo luận về kỹ thuật tạo mã chế độ bảng màu để cải thiện hiệu quả tạo mã cho nội dung màn hình, như là video được tạo ra bởi máy tính mà chứa lượng văn bản và đồ họa đáng kể. Để áp dụng một cách hiệu quả kỹ thuật này, thì phương pháp tạo mã và phát tín hiệu thông tin có liên quan là cần thiết.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là để đề xuất phương pháp và thiết bị để nâng cao hiệu quả tạo mã video/hình ảnh.

Một mục đích khác của sáng chế là để đề xuất phương pháp và thiết bị để cải thiện hiệu quả trong việc tạo mã chế độ bảng màu.

Một mục đích khác nữa của sáng chế là để đề xuất phương pháp và thiết bị để tạo cấu hình và phát tín hiệu một cách hiệu quả các loại thông tin khác nhau được sử dụng trong việc tạo mã chế độ bảng màu.

Một mục đích khác này nữa của sáng chế là để đề xuất phương pháp và thiết bị để áp dụng một cách hiệu quả việc tạo mã thoát trong chế độ bảng màu.

Theo một phương án của sáng chế, thông số lượng tử hóa cho giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể được dẫn xuất dựa trên thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi. Hơn nữa, thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi có thể được phân tích cú pháp/được phát tín hiệu thông qua tập hợp thông số chuỗi (Sequence Parameter Set, SPS).

Theo một phương án của sáng chế, phạm vi của các giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể được giới hạn dựa trên độ sâu bit. Ví dụ, phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần độ chói có các giá trị giữa 0 và

$(1 \ll \text{BitDepthY}) - 1$, và phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần sắc độ có thể có các giá trị giữa 0 và $(1 \ll \text{BitDepthC}) - 1$.

Theo một phương án của sáng chế, thông tin kích cỡ mục nhập bảng màu để cấu thành bảng của bảng màu của chế độ bảng màu có thể được định rõ và được phát tín hiệu thông qua tập hợp thông số chuỗi (SPS). Hơn nữa, thông tin kích cỡ mục nhập bảng màu có thể là giá trị được thiết lập trước hoặc có thể được dẫn xuất dựa trên kích cỡ của đơn vị tạo mã.

Theo một phương án, sáng chế đề xuất phương pháp giải mã video/hình ảnh được thực hiện bởi thiết bị giải mã. Phương pháp giải mã video/hình ảnh có thể gồm phương pháp được bộc lộ trong các phương án của sáng chế.

Theo một phương án, sáng chế đề xuất thiết bị giải mã để thực hiện việc giải mã video/hình ảnh. Thiết bị giải mã có thể thực các phương pháp được bộc lộ trong các phương án của sáng chế.

Theo một phương án, sáng chế đề xuất phương pháp mã hóa video/hình ảnh được thực hiện bởi thiết bị mã hóa. Phương pháp mã hóa video/hình ảnh có thể gồm phương pháp được bộc lộ trong các phương án của sáng chế.

Theo một phương án, sáng chế đề xuất thiết bị mã hóa để thực hiện việc mã hóa video/hình ảnh. Thiết bị mã hóa có thể thực hiện phương pháp được bộc lộ trong các phương án của sáng chế.

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

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

Hiệu quả của sáng chế

Sáng chế có các hiệu quả khác nhau. Ví dụ, theo một phương án, sáng chế có thể cải thiện hiệu quả nén hình ảnh/video tổng thể. Bên cạnh đó, theo một phương án, sáng chế có thể cải thiện hiệu quả trong việc tạo mã chế độ bảng màu. Hơn nữa, theo một phương án, sáng chế có thể tạo cấu hình và phát tín hiệu một cách hiệu quả các loại thông tin khác nhau được sử dụng trong việc tạo mã chế độ bảng màu. Bên cạnh đó, theo một phương án, sáng chế có thể cải thiện độ chính xác và hiệu quả tạo mã cho các mẫu thoát bằng cách áp dụng một cách hiệu quả việc tạo mã thoát trong chế độ bảng màu.

Các hiệu quả có thể được thu nhận thông qua các ví dụ cụ thể của sáng chế không bị giới hạn ở các hiệu quả được liệt kê ở trên. Ví dụ, có thể có các hiệu quả kỹ thuật khác nhau mà người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng có thể hiểu hoặc dẫn xuất từ sáng chế. Theo đó, các hiệu quả cụ thể của sáng chế không bị giới hạn ở những hiệu quả được mô tả một cách rõ ràng trong sáng chế và có thể gồm các hiệu quả khác nhau mà có thể được hiểu hoặc được dẫn xuất từ các dấu hiệu kỹ thuật của sáng chế.

Mô tả vắn tắt các hình vẽ

Fig.1 thể hiện một cách sơ lược ví dụ về hệ thống tạo mã video/hình ảnh có thể áp dụng được vào các phương án của sáng chế.

Fig.2 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị mã hoá video/hình ảnh mà các phương án của sáng chế có thể áp dụng được vào đó.

Fig.3 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh mà các phương án của sáng chế có thể áp dụng được vào đó.

Fig.4 thể hiện ví dụ để mô tả cấu trúc cơ sở của việc tạo mã bảng màu.

Fig.5 thể hiện ví dụ để mô tả các phương pháp quét ngang qua theo chiều ngang và theo chiều dọc được sử dụng để tạo mã bản đồ chỉ số bảng màu.

Fig.6 là sơ đồ để mô tả ví dụ về phương pháp tạo mã dựa trên chế độ bảng màu.

Fig.7 thể hiện một cách sơ lược ví dụ về phương pháp mã hóa video/hình ảnh theo phương án (các phương án) của sáng chế.

Fig.8 thể hiện một cách sơ lược ví dụ về phương pháp mã hóa video/hình ảnh theo phương án (các phương án) của sáng chế.

Fig.9 thể hiện ví dụ về hệ thống phát luồng nội dung mà các phương án của sáng chế có thể áp dụng được vào đó.

Mô tả chi tiết sáng chế

Sáng chế có thể được sửa đổi theo các cách khác nhau và có thể có các phương án khác nhau, và các phương án cụ thể sẽ được minh họa trên các hình vẽ và được mô tả chi tiết. Tuy nhiên, điều này không nhằm giới hạn sáng chế ở các phương án cụ thể được bộc lộ ở đây. Các thuật ngữ được sử dụng chung trong bản mô tả này được sử dụng để mô tả phương án cụ thể và không nhằm giới hạn nguyên lý kỹ thuật của sáng chế. Sự trình bày dạng số ít gồm cả các sự trình bày dạng số nhiều trừ khi được thể hiện rõ ràng trong ngữ cảnh theo cách khác. Thuật ngữ, như là “gồm” hoặc “có” trong tài liệu này, nên được hiểu là để chỉ ra sự tồn tại của đặc tính, số, bước, hoạt động, phần tử, phần,

hoặc tổ hợp của chúng được mô tả trên bản mô tả này và không loại trừ sự tồn tại hoặc khả năng của việc bổ sung của một hoặc nhiều đặc tính, số, bước, hoạt động, phần tử, phần, hoặc tổ hợp của chúng.

Các phần tử trên các hình vẽ được mô tả trên sáng chế được minh họa một cách độc lập để tạo thuận tiện cho việc mô tả liên quan đến các chức năng đặc trưng khác nhau. Điều này không có nghĩa là mỗi phần tử trong số các phần tử được triển khai dưới dạng phần cứng tách biệt hoặc phần mềm tách biệt. Ví dụ, ít nhất hai phần tử có thể được tổ hợp để tạo thành phần tử đơn lẻ, hoặc phần tử đơn lẻ có thể được chia thành nhiều phần tử. Phương án mà trong đó các phần tử được tổ hợp và/hoặc được tách rời cũng được gồm trong phạm vi bảo hộ của sáng chế, trừ khi nó lệch khỏi bản chất của sáng chế này.

Các dấu hiệu kỹ thuật mà được mô tả một cách riêng lẻ trên một hình vẽ trong sáng chế có thể được triển khai một cách riêng lẻ hoặc đồng thời.

Ở dưới đây, các phương án của sáng chế sẽ được mô tả chi tiết hơn với sự tham khảo đến các hình vẽ kèm theo. Ở dưới đây, các số tham chiếu giống nhau được sử dụng cho các chi tiết giống nhau trong các hình vẽ, và các phần mô tả dư thừa của các chi tiết này sẽ được lược bỏ.

Fig.1 thể hiện một cách sơ lược ví dụ về hệ thống tạo mã video/hình ảnh có thể áp dụng được vào các phương án của sáng chế.

Tham khảo đến Fig.1, hệ thống tạo mã video/hình ảnh có thể gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị nhận). Thiết bị nguồn có thể truyền thông tin hoặc dữ liệu video/hình ảnh được mã hoá dưới dạng tệp tin hoặc phát luồng đến thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng.

Thiết bị nguồn có thể gồm nguồn video, 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 video/hình ảnh, và thiết bị giải mã có thể được gọi là thiết bị giải mã video/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, và bộ phận hiển thị này có thể được tạo cấu hình như thiết bị tách biệt hoặc thành phần ngoài.

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

Thiết bị mã hóa có thể mã hóa video/hình ảnh đầu vào. Thiết bị mã hoá có thể thực hiện một loạt thủ tục như là dự đoán, biến đổi, và lượng tử hoá cho hiệu quả nén và tạo mã. Dữ liệu được mã hóa (thông tin video/hình ảnh được mã hóa) có thể xuất ra trong dạng luồng bit.

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

qua định dạng tệp tin 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 nó đến thiết bị giải mã.

Thiết bị giải mã có thể giải mã video/hình ảnh bằng cách thực hiện một loạt thủ tục 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 video/hình ảnh được giải mã. Video/ảnh được kết xuất có thể được hiển thị thông qua đơn vị hiển thị.

Trong sáng chế, “A hoặc B” có thể có nghĩa là “chỉ A”, “chỉ B” hoặc “cả A và B”. Nói cách khác, trong sáng chế, “A hoặc B” có thể được diễn giải là để chỉ ra “A và/hoặc B”. Ví dụ, trong sáng chế, thuật ngữ “A, B hoặc C” có thể có nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “tổ hợp bất kỳ của A, B và C”.

Dấu gạch chéo “/” hoặc dấu phẩy được sử dụng trong sáng chế có nghĩa là “và/hoặc”. Ví dụ, “A/B” có nghĩa là “A và/hoặc B”. Theo đó, “A/B” có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ví dụ, “A, B, C” có thể có nghĩa là “A, B hoặc C”.

Trong sáng chế, “ít nhất một thành phần trong số A và B” có thể chỉ “chỉ A”, “chỉ B”, hoặc “cả A và B”. Hơn nữa, trong sáng chế, sự trình bày “ít nhất một thành phần trong số A hoặc B” hoặc “ít nhất một thành phần trong số A và/hoặc B” có thể được diễn giải như “ít nhất một thành phần trong số A và B”.

Hơn nữa, trong sáng chế, “ít nhất một thành phần trong số A, B và C” có thể chỉ “chỉ A”, “chỉ B”, “chỉ C”, hoặc “tổ hợp bất kỳ của A, B và C”. Hơn nữa, “ít nhất một thành phần trong số A, B hoặc C” hoặc “ít nhất một thành phần trong số A, B và/hoặc C” có thể có nghĩa là “ít nhất một thành phần trong số A, B và C”.

Hơn nữa, dấu ngoặc đơn được sử dụng trong sáng chế này có thể có nghĩa là “ví dụ”. Một cách cụ thể, trong trường hợp mà “việc dự đoán (việc dự đoán trong ảnh)” được trình bày, thì nó có thể được chỉ ra là “việc dự đoán trong ảnh” như được đề xuất như là ví dụ của “việc dự đoán”. Nói cách khác, thuật ngữ “việc dự đoán” trong sáng chế không bị hạn chế ở “việc dự đoán trong ảnh”, và nó có thể được chỉ ra là “việc dự đoán trong ảnh” được đề xuất như là ví dụ của “việc dự đoán”. Hơn nữa, ngay cả trong trường hợp mà “việc dự đoán (tức là, việc dự đoán trong ảnh)” được trình bày, thì nó có thể được chỉ ra là “việc dự đoán trong ảnh” như được đề xuất là ví dụ của “việc dự đoán”.

Sáng chế liên quan đến việc tạo mã video/hình ảnh. Ví dụ, phương pháp/phương án được bộc lộ trong sáng chế có thể được áp dụng cho phương pháp được bộc lộ theo tiêu chuẩn tạo mã video vạn năng (Versatile Video Coding, VVC). Hơn nữa, các phương pháp/các phương án được bộc lộ trong sáng chế có thể được áp dụng cho phương pháp được bộc lộ trong tiêu chuẩn tạo mã video thiết yếu (Essential Video Coding, EVC), tiêu chuẩn AOMedia video 1 (AOMedia Video 1, AV1), thế hệ thứ 2 của tiêu chuẩn tạo mã audio video (2nd Generation of Audio Video Coding Standard, AVS2), hoặc tiêu chuẩn tạo mã video/hình ảnh thế hệ tiếp theo (ví dụ, H.267 hoặc H.268, v.v.).

Sáng chế giới thiệu các phương án khác nhau của việc tạo mã video/hình ảnh, và các phương án có thể được thực hiện trong sự tổ hợp với nhau trừ khi được đề cập theo cách khác.

Trong sáng chế, video có thể nghĩa là tập hợp của một loạt các hình ảnh theo thời gian. Ảnh đề cập chung đến đơn vị biểu diễn một hình ảnh trong vùng thời gian cụ thể, và lát/phiến là đơn vị cấu thành một phần của ảnh trong khi tạo mã. Lát/phiến có thể gồm một hoặc nhiều đơn vị cây tạo mã (Coding Tree Unit, CTU). Một ảnh có thể gồm có một hoặc nhiều lát/phiến. Phiến là vùng hình chữ nhật của các CTU trong cột phiến

cụ thể và hàng phiến cụ thể trong ảnh. Cột phiến là vùng hình chữ nhật của các CTU có chiều cao bằng với chiều cao của ảnh và chiều rộng được chỉ rõ bởi các phần tử cú pháp trong tập hợp thông số ảnh. Hàng phiến là vùng hình chữ nhật của các CTU có chiều cao được chỉ rõ bởi các phần tử cú pháp trong tập hợp thông số ảnh và chiều rộng bằng với chiều rộng của ảnh. Việc quét phiến là việc sắp thứ tự theo trình tự cụ thể của các CTU phân vùng ảnh mà trong đó các CTU được sắp thứ tự liên tiếp trong việc quét mảnh CTU trong phiến, trong khi các phiến trong ảnh được sắp thứ tự liên tiếp trong việc quét mảnh của các phiến của ảnh. Lát gồm số nguyên của các phiến hoàn chỉnh hoặc số nguyên của các hàng CTU hoàn chỉnh liên tiếp trong phiến của ảnh, mà có thể được chứa theo cách dành riêng trong đơn vị NAL đơn lẻ.

Trong lúc đó, một ảnh có thể được phân chia thành hai hoặc nhiều ảnh con. Ảnh con có thể là vùng hình chữ nhật hoặc một hoặc nhiều lát nằm trong ảnh.

Điểm ảnh (pixel) hoặc pel có thể có nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc một 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. Theo cách thay thế, mẫu có thể có nghĩa là giá trị điểm ảnh trong miền không gian, hoặc có thể có nghĩa là hệ số biến đổi trong miền tần số khi giá trị điểm ảnh được biến đổi thành miền tần số.

Đơn vị có thể biểu diễn đơn vị cơ sở 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 và 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, khối $M \times N$ có thể gồm các mẫu (hoặc các mảng mẫu) hoặc tập (hoặc mảng) các hệ số biến đổi gồm M cột và N hàng.

Ngoài ra, 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/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 các hệ số biến đổi có thể được gọi là các hệ số hoặc các hệ số dư, hoặc vẫn có thể được gọi là các hệ số biến đổi vì tính đồng nhất của sự trình bày.

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 phát tín hiệu thông qua cú pháp tạo 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 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.

Fig.2 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị mã hoá video/hình ảnh mà các phương án của sáng chế có thể áp dụng được vào đó. Ở dưới đây, thiết bị mã hóa có thể gồm thiết bị mã hóa hình ảnh và/hoặc thiết bị mã hóa video.

Tham khảo đến Fig.2, thiết bị mã hóa 200 có thể gồm và được tạo cấu hình với bộ chia vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, và bộ mã hóa entropi 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể gồm bộ dự đoán

liên ảnh 221 và bộ dự đoán trong ảnh 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 tái tạo hoặc bộ tạo khối được tái tạo. Bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropi 240, bộ cộng 250, và bộ lọc 260 có thể được tạo cấu hình bởi ít nhất một thành phần phần cứng (ví dụ, bộ chip bộ mã hóa hoặc bộ xử lý) theo một phương án. Bên cạnh đó, 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 thành phần trong/ngoài.

Bộ phân vùng hình ảnh 210 có thể chia tách hình ảnh đầu vào (hoặc ảnh, khung) được nhập vào thiết bị mã hóa 200 thành một hoặc nhiều đơn vị xử lý. Theo một ví dụ, đơn vị xử lý có thể được gọi là đơn vị tạo mã (Coding Unit, CU). Trong trường hợp này, đơn vị tạo mã có thể có thể được chia tách theo cách đệ quy theo cấu trúc cây tứ phân cây nhị phân cây tam phân (Quad-Tree Binary-Tree Ternary-Tree, QTBTNT) từ đơn vị cây tạo mã (Coding Tree Unit, CTU) hoặc đơn vị tạo mã lớp nhất (Largest Coding Unit, LCU). Ví dụ, một đơn vị tạo mã có thể được chia tách thành nhiều đơn vị tạo mã có độ sâu sâu hơn dựa vào 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 trước tiên và cấu trúc cây nhị phân và/hoặc cấu trúc cây tam phân có thể được áp dụng sau. Theo cách thay thế, cấu trúc cây nhị phân có thể được áp dụng trước tiên. Thủ tục tạo mã theo sáng chế này có thể được thực hiện dựa vào đơn vị tạo mã cuối cùng mà không được chia tách thêm nữa. Trong trường hợp này, dựa trên hiệu quả tạo mã theo các đặc tính hình ảnh hoặc tương tự, thì đơn vị tạo mã tối đa có thể được sử dụng một cách trực tiếp như là đơn vị tạo mã cuối cùng, hoặc khi cần, đơn vị tạo mã có thể được chia tách một cách đệ quy thành các đơn vị tạo mã có độ sâu sâu hơn, sao cho đơn vị tạo mã có kích cỡ tối

ưu có thể được sử dụng như là đơn vị tạo mã cuối cùng. Ở đây, thủ tục tạo mã có thể gồm thủ tục như là việc dự đoán, biến đổi, và tái tạo 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 đơn vị trong số đơn vị dự đoán và đơn vị biến đổi có thể được chia tách và được phân vùng từ đơn vị tạo mã cuối cùng nêu trên. Đơn vị dự đoán có thể là đơn vị của mẫu dự đoán, và đơn vị biến đổi có thể là đơn vị để sinh ra hệ số biến đổi và/hoặc đơn vị để sinh ra tín hiệu dư từ hệ số biến đổi.

Đơn vị này có thể được sử dụng theo kiểu thay thế cho nhau với thuật ngữ như là như khối hoặc vùng trong một số trường hợp. Nói chung, khối $M \times N$ có thể biểu diễn các mẫu gồm có M cột và N hàng hoặc nhóm các hệ số biến đổi. Mẫu có thể biểu diễn chung cho điểm ảnh hoặc giá trị của điểm ảnh, và cũng có thể chỉ biểu diễn cho điểm ảnh/giá trị điểm ảnh của thành phần độ chói, và cũng có thể chỉ biểu diễn cho 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 mà tạo cấu hình một ảnh (hoặc hình ảnh).

Thiết bị mã hoá 200 có thể tạo ra tín hiệu dư (khối dư, mảng mẫu dư) bằng cách trừ đi tín hiệu được dự đoán (khối được dự đoán, mảng mẫu dự đoán), mà được xuất ra từ bộ dự đoán liên ảnh 221 hoặc bộ dự đoán trong ảnh 222, khỏi tín hiệu hình ảnh đầu vào (khối gốc, mảng mẫu gốc), và tín hiệu dư được tạo ra sẽ được truyền đến bộ biến đổi 232. Trong trường hợp này, như được minh họa, thì đơn vị để trừ đi tín hiệu được dự đoán (khối được dự đoán, mảng mẫu dự đoán) khỏi tín hiệu hình ảnh đầu vào (khối gốc, mảng mẫu gốc) trong bộ mã hoá 200 có thể được gọi là bộ trừ 231. Bộ dự đoán có thể thực hiện việc dự đoán đối với khối cần được xử lý (ở dưới đây được gọi là 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 liệu việc dự đoán trong ảnh được áp dụng hay việc dự đoán liên ảnh được áp dụng trong các đơn vị của khối hiện tại hoặc CU. Bộ dự đoán có thể tạo ra các thông tin khác nhau về việc dự đoán, như là thông tin về chế độ dự đoán, để truyền thông tin được tạo ra đến bộ mã hoá entropi 240, như sẽ được mô tả sau trong phần mô tả về mỗi chế độ dự đoán. Thông tin về việc dự đoán có thể được mã hoá bởi bộ mã hoá entropi 240 để được xuất ra trong dạng luồng bit.

Bộ dự đoán trong ảnh 222 có thể dự đoán khối hiện tại dựa vào các mẫu nằm 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 cách khối khối hiện tại theo chế độ dự đoán. Các chế độ dự đoán trong việc dự đoán trong ảnh có thể gồm các chế độ không có hướng và các 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 mịn của hướng dự đoán. Tuy nhiên, đây chỉ là minh hoạ, và các chế độ dự đoán có hướng với số lượng nhiều hoặc ít hơn so với số lượng ở trên cũng có thể được sử dụng theo việc thiết lập. Bộ dự đoán trong ảnh 222 cũng có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại nhờ sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên ảnh 221 có thể sinh ra khối được dự đoán của khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi véc tơ chuyển động trên ảnh tham chiếu. Lúc này, để giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên ảnh, thì thông tin chuyển động có thể được dự đoán theo các đơn vị là khối, khối con, hoặc mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm véc tơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin về hướng dự đoán

liên ảnh (dự đoán L0, dự đoán L1, dự đoán đôi, v.v.). Trong trường hợp dự đoán liên ảnh, thì khối lân cận có thể gồm khối lân cận về mặt không gian tồn tại trong ảnh hiện tại và khối lân cận về mặt thời gian tồn tại trong ảnh tham chiếu. Ảnh tham chiếu gồm khối tham chiếu và ảnh tham chiếu gồm khối lân cận về mặt thời gian này cũng có thể là giống nhau, và cũng có thể là khác nhau. Khối lân cận về mặt thời gian có thể được gọi với tên gọi như là khối tham chiếu được đặt cùng một chỗ, CU được đặt cùng một chỗ (collocated, CU, colCU), và 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 (co-located Picture, colPic). Ví dụ, bộ dự đoán liên ảnh 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 véctor chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại. Việc dự đoán liên ảnh 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, thì bộ dự đoán liên ảnh 221 có thể sử dụng thông tin chuyển động của khối lân cận làm thông tin chuyển động của khối hiện tại. Trong trường hợp chế độ bỏ qua, thì tín hiệu dư có thể không được truyền, không giống như chế độ hợp nhất. Chế độ dự đoán véctor chuyển động (Motion Vector Prediction, MVP) có thể chỉ ra véctor chuyển động của khối hiện tại bằng cách sử dụng véctor chuyển động của khối lân cận làm bộ dự đoán véctor chuyển động, và phát tín hiệu hiệu sự chênh lệch véctor chuyển động.

Bộ dự đoán 200 có thể tạo ra tín hiệu được dự đoán dựa trên các phương pháp dự đoán khác nhau, sẽ được mô tả sau. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc dự đoán trong ảnh hoặc dự đoán liên ảnh để dự đoán một khối, mà còn áp dụng một cách đồng thời việc dự đoán trong ảnh và việc dự đoán liên ảnh. Việc này có thể được gọi là việc dự đoán trong ảnh và liên ảnh được tổ hợp (Combined Inter and Intra Prediction, CIIP). Hơn nữa, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép khối trong ảnh

(Intra Block Copy, IBC) hoặc chế độ bảng màu (palette) để thực hiện việc dự đoán trên khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc tạo mã hình ảnh/video nội dung của trò chơi hoặc tương tự, như là, tạo mã nội dung màn (Screen Content Coding, SCC). IBC về cơ bản thực hiện việc dự đoán trong ảnh hiện tại, nhưng nó có thể được thực hiện theo cách tương tự như việc dự đoán liên ảnh ở chỗ nó dẫn xuất khối tham chiếu trong ảnh hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một kỹ thuật trong số các kỹ thuật dự đoán liên ảnh được mô tả trong tài liệu này. Chế độ bảng màu có thể được coi như là ví dụ của việc tạo mã trong ảnh hoặc việc dự đoán trong ảnh. Khi chế độ bảng màu được áp dụng, thì giá trị mẫu trong ảnh có thể được phát tín hiệu dựa trên thông tin về chỉ số bảng màu và bảng của bảng màu.

Tín hiệu được dự đoán mà được tạo ra thông qua bộ dự đoán (gồm bộ dự đoán liên ảnh 221 và/hoặc bộ dự đoán trong ảnh 222) có thể được sử dụng để tạo ra tín hiệu được tái tạo hoặc được sử dụng 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 cosin rời rạc (Discrete Cosine Transform, DCT), biến đổi sin rời rạc (Discrete Sine Transform, DST), biến đổi Karhunen-Loève (Karhunen-Loève Transform, KLT), biến đổi dựa trên đồ thị (Graph-Based Transform, GBT), hoặc biến đổi phi tuyến có điều kiện (Conditionally Non-linear Transform, CNT). Ở đây, khi thông tin quan hệ giữa các điểm ảnh được minh họa dưới dạng đồ thị, thì GBT có nghĩa là phép biến đổi thu nhận được từ đồ thị này. CNT có nghĩa là phép biến đổi mà giành được dựa trên tín hiệu được dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được tái tạo trước đó. Bên cạnh, quy trình biến đổi cũng có thể được áp dụng cho khối điểm ảnh có cùng kích cỡ là hình vuông, và cũng có thể được áp dụng cho khối có kích cỡ biến thiên không phải là hình vuông.

Bộ lượng tử hoá 233 có thể lượng tử hoá các hệ số biến đổi để truyền các hệ số biến đổi được lượng tử hoá đến bộ mã hoá entropi 240, và bộ mã hoá entropi 240 có thể mã hoá tín hiệu được lượng tử hoá (thông tin về các hệ số biến đổi được lượng tử hoá) thành tín hiệu được lượng tử hoá mà được mã hoá đến luồng bit. Thông tin về các hệ số biến đổi được lượng tử hoá có thể được gọi là thông tin phân dư. Bộ lượng tử hoá 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hoá có dạng khối trong dạng vectơ một chiều dựa trên thứ tự quét hệ số, và còn tạo ra thông tin về các hệ số biến đổi được lượng tử hoá dựa trên các hệ số biến đổi được lượng tử hoá của dạng vectơ một chiều. Bộ mã hoá entropi 240 có thể thực hiện các phương pháp mã hoá khác nhau, ví dụ, tạo mã Golomb hàm số mũ, tạo mã chiều dài biến thiên thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding, CAVLC), và tạo mã số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC). Bộ mã hoá entropi 240 cũng có thể mã hoá thông tin (ví dụ, các giá trị của các phần tử cú pháp, v.v.) cần thiết để tái tạo video/hình ảnh ngoài các hệ số biến đổi được lượng tử hoá, cùng nhau hoặc một cách tách biệt. Thông tin được mã hoá (ví dụ, thông tin video/hình ảnh được mã hoá) có thể được truyền hoặc được lưu trữ trong đơn vị của đơn vị lớp trừu tượng mạng (Network Abstraction Layer, NAL) trong dạng luồng bit. Thông tin video/hình ảnh có thể còn gồm thông tin về các tập hợp thông số khác nhau, như là tập hợp thông số thích ứng (Adaptation Parameter Set, APS), tập hợp thông số ảnh (Picture Parameter Set, PPS), tập hợp thông số chuỗi (Sequence Parameter Set, SPS), hoặc tập hợp thông số video (Video Parameter Set, VPS). Bên cạnh đó, thông tin video/hình ảnh có thể còn gồm thông tin ràng buộc chung. Thông tin được phát tín hiệu/được truyền và/hoặc các phần tử cú pháp được mô tả sau trong tài liệu này có thể được mã hóa thông qua thủ tục mã hóa nêu trên và vì thế được gồm trong luồng bit. Luồng bit có thể được truyền thông qua mạng, hoặc 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, hoặc tương tự, và 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, và SSD. Bộ truyền (không được minh họa) truyền tín hiệu được xuất ra từ bộ mã hóa entropi 240 và/hoặc bộ phận lưu trữ (không được minh họa) lưu trữ tín hiệu có thể được tạo cấu hình làm các phần tử trong/ngoài của thiết bị mã hóa 200, hoặc bộ truyền cũng có thể được gồm trong bộ mã hóa entropi 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 được dự đoán. Ví dụ, bộ khử lượng tử hóa 234 và bộ biến đổi ngược 235 áp dụng việc khử lượng tử và biến đổi ngược cho các hệ số biến đổi được lượng tử hóa, sao cho tín hiệu dư (khối dư hoặc các mẫu dư) có thể được tái tạo. Bộ cộng 250 cộng tín hiệu dư được tái tạo vào tín hiệu được dự đoán được xuất ra từ bộ dự đoán liên ảnh 221 hoặc bộ dự đoán trong ảnh 222 sao cho tín hiệu được tái tạo (ảnh được tái tạo, khối được tái tạo, mảng mẫu được tái tạo) có thể được tạo ra. Như trong trường hợp mà trong đó chế độ bỏ qua được áp dụng, nếu không có phần dư cho khối cần được xử lý, thì khối được dự đoán có thể được sử dụng làm khối được tái tạo. Bộ cộng 250 có thể được gọi là bộ tái tạo hoặc bộ tạo ra khối được tái tạo. Tín hiệu được tái tạo mà được tạo ra có thể được sử dụng cho việc dự đoán trong ảnh của khối tiếp theo cần được xử lý nằm trong ảnh hiện tại, và như được mô tả sau, cũng được sử dụng cho việc dự đoán liên ảnh của ảnh tiếp theo thông qua việc lọc.

Trong lúc đó, việc ánh xạ độ chói với việc định cỡ sắc độ (Luma Mapping with Chroma Scaling, LMCS) cũng có thể được áp dụng trong quy trình mã hóa và/hoặc tái tạo ảnh.

Bộ lọc 260 có thể áp dụng việc lọc cho tín hiệu được tái tạo, nhờ đó cải thiện các chất lượng hình ảnh chủ quan/khách quan. Ví dụ, bộ lọc 260 có thể áp dụng các phương

pháp lọc khác nhau cho ảnh được tái tạo để tạo ra ảnh được tái tạo mà được sửa đổi, và lưu trữ ảnh được tái tạo mà được sửa đổi trong bộ nhớ 270, một cách cụ thể, DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch tương thích mẫu, bộ lọc vòng lặp tương thích, bộ lọc song phương, và tương tự. Bộ lọc 260 có thể tạo ra các thông tin liên quan đến việc lọc khác nhau để truyền thông tin được tạo ra đến bộ mã hóa entropi 240, 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 entropi 240 để xuất ra trong dạng luồng bit.

Hình ảnh được tái tạo mà được sửa đổi sẽ được truyền đến bộ nhớ 270 có thể được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên ảnh 221. Nếu việc dự đoán liên ảnh được áp dụng bởi bộ dự đoán liên ảnh, thì thiết bị mã hoá có thể tránh được sự không khớp dự đoán giữa thiết bị mã hoá 200 và thiết bị giải mã, và còn cải thiện hiệu quả tạo mã.

DPB của bộ nhớ 270 có thể lưu trữ ảnh được tái tạo mà được sửa đổi để được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên ảnh 221. Bộ nhớ 270 có thể lưu trữ thông tin chuyển động của khối mà trong đó thông tin chuyển động trong ảnh hiện tại được dẫn xuất (hoặc được mã hoá) và/hoặc thông tin chuyển động của các khối trong ảnh được tái tạo trước đó. Thông tin chuyển động được lưu trữ này có thể được truyền đến bộ dự đoán liên ảnh 221 để được tận dụng làm thông tin chuyển động của khối lân cận về mặt không gian hoặc thông tin chuyển động của khối lân cận về mặt thời gian. Bộ nhớ 270 có thể lưu trữ các mẫu được tái tạo của các khối được tái tạo trong ảnh hiện tại, và truyền các mẫu được tái tạo này đến bộ dự đoán trong ảnh 222.

Fig.3 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh mà các phương án của sáng chế có thể áp dụng được vào đó. Ở dưới đây, thiết bị giải mã có thể gồm thiết bị giải mã hình ảnh và/hoặc thiết bị giải mã video.

Tham khảo đến Fig.3, thiết bị giải mã 300 có thể gồm và được tạo cấu hình với bộ giải mã entropi 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 ảnh 331 và bộ dự đoán trong ảnh 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 entropi 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, vốn được mô tả ở trên, có thể được tạo cấu hình bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip hoặc bộ xử lý bộ giải mã) theo phương án. Hơn nữa, bộ nhớ 360 có thể gồm bộ đệm ảnh được giải mã (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 360 như là thành phần trong/ngoài.

Khi luồng bit gồm thông tin video/hình ảnh được nhập vào, thì thiết bị giải mã 300 có thể tái tạo hình ảnh để đáp lại quy trình mà trong đó thông tin video/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 chia tách khối giành được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã nhờ sử dụng đơn vị xử lý được áp dụng cho thiết bị mã hóa. Do đó, đơn vị xử lý cho việc giải mã có thể, ví dụ, là đơn vị tạo mã, và đơn vị tạo mã này có thể chia tách theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây mã hóa hoặc đơn vị tạo mã tối đa. Một hoặc nhiều đơn vị biến đổi có thể được dẫn xuất từ đơn vị tạo mã. Bên cạnh đó, tín hiệu hình ảnh được tái tạo mà được giải mã và được xuất ra thông qua thiết bị giải mã 300 có thể được tái sản xuất thông qua thiết bị tái sản xuất.

Thiết bị giải mã 300 có thể nhận tín hiệu được xuất ra từ thiết bị mã hóa được minh họa trên Fig.2 trong dạng luồng bit, và tín hiệu nhận được có thể được giải mã thông qua bộ giải mã entropi 310. Ví dụ, bộ giải mã entropi 310 có thể để dẫn xuất thông tin (ví dụ, thông tin video/hình ảnh) cần thiết cho việc tái tạo hình ảnh (hoặc tái tạo ảnh) bằng cách phân tích cú pháp luồng bit. Thông tin video/hình ảnh có thể còn gồm thông tin về các tập hợp thông số khác nhau như là tập hợp thông số thích ứng (APS), tập hợp thông số ảnh (PPS), tập hợp thông số chuỗi (tSPS), và tập hợp thông số video (VPS). Bên cạnh đó, thông tin video/hình ảnh có thể còn gồm thông tin ràng buộc chung. Thiết bị giải mã có thể giải mã ảnh còn dựa trên thông tin về tập hợp thông số và/hoặc thông tin ràng buộc chung. Thông tin được phát tín hiệu/nhận được và/hoặc các phần tử cú pháp được mô tả sau trong tài liệu này có thể được giải mã thông qua thủ tục giải mã và giành được từ luồng bit. Ví dụ, bộ giải mã entropi 310 có thể giải mã thông tin trong luồng bit dựa trên phương pháp mã hóa như mã hóa Golomb hàm mũ, CAVLC, hoặc CABAC, và xuất ra giá trị của phần tử cú pháp cần thiết cho việc tái tạo hình ảnh, và các giá trị được lượng tử hóa của hệ số biến đổi liên quan đến phần dư. Một cách cụ thể hơn, phương pháp giải mã entropi CABAC có thể nhận ngán tương ứng với mỗi phần tử cú pháp từ luồng bit, xác định mô hình ngữ cảnh nhờ sử dụng thông tin phần tử cú pháp cần được giải mã và giải mã thông tin của khối lân cận và khối cần được giải mã hoặc thông tin của ký hiệu/ngán được giải mã trong giai đoạn trước đó, và tạo ra ký hiệu tương ứng với giá trị của mỗi phần tử cú pháp bằng cách dự đoán khả năng của việc tạo ra ngán theo mô hình ngữ cảnh được xác định để thực hiện việc giải mã số học của ngán. Lúc này, phương pháp giải mã entropi CABAC có thể xác định mô hình ngữ cảnh và sau đó cập nhật mô hình ngữ cảnh này nhờ sử dụng thông tin của ký hiệu/ngán được giải mã này cho mô hình ngữ cảnh của ký hiệu/ngán tiếp theo. Thông tin về việc dự đoán trong số các thông tin được giải mã bởi bộ giải mã entropi 310 có thể được cung

cấp cho bộ dự đoán (bộ dự đoán liên ảnh 332 và bộ dự đoán trong ảnh 331), và giá trị dư mà tại đó việc giải mã entropi được thực hiện bởi bộ giải mã entropi 310, nghĩa là, các hệ số biến đổi được lượng tử hoá và thông tin thông số liên quan, có thể được nhập vào bộ xử lý phần dư 320. Bộ xử lý phần dư 320 có thể dẫn xuất tín hiệu dư (khối dư, các mẫu dư, mảng mẫu dư). Bên cạnh đó, thông tin về việc lọc trong số thông tin được giải mã bởi bộ giải mã entropi 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 cũng có thể là thành phần của bộ giải mã entropi 310. Trong lúc đó, thiết bị giải mã theo tài liệu này có thể được gọi là thiết bị giải mã video/hình ảnh/ảnh, và thiết bị giải mã cũng có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin video/hình ảnh/ảnh) và bộ giải mã mẫu (bộ giải mã mẫu video/hình ảnh/ảnh). Bộ giải mã thông tin có thể gồm bộ giải mã entropi 310, và bộ giải mã mẫu có thể gồm ít nhất một thành phần trong số bộ khử lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ dự đoán liên ảnh 332, và bộ dự đoán trong ảnh 331.

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 bằng cách sử dụng thông số khử lượng tử hóa (ví dụ, thông tin kích cỡ bước lượng tử hóa) và giành được các hệ số biến đổi.

Bộ biến đổi ngược 322 biến đổi ngược các hệ số biến đổi để giành được tín hiệu dư (khối dư, mảng mẫu 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 liệu việc dự đoán trong ảnh được áp dụng hoặc việc dự đoán liên ảnh được áp dụng cho khối hiện tại dựa trên thông tin về việc dự đoán được xuất ra từ bộ giải mã entropi 310, và xác định chế độ dự đoán trong ảnh/liên ảnh cụ thể.

Bộ dự đoán có thể tạo ra tín hiệu được dự đoán dựa trên các phương pháp dự đoán khác nhau sẽ được mô tả sau. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc dự đoán trong ảnh hoặc việc dự đoán liên ảnh cho việc dự đoán của một khối, mà cũng có thể áp dụng một cách đồng thời việc dự đoán trong ảnh và việc dự đoán liên ảnh. Việc này có thể được gọi là việc dự đoán trong ảnh và liên ảnh được tổ hợp (CIIP). Hơn nữa, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép khối trong ảnh (IBC) hoặc chế độ bảng màu để thực hiện việc dự đoán trên khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc tạo mã hình ảnh/video nội dung của trò chơi hoặc tương tự, như là, tạo mã nội dung màn (SCC). IBC về cơ bản thực hiện việc dự đoán trong ảnh hiện tại, nhưng nó có thể được thực hiện theo cách tương tự như việc dự đoán liên ảnh ở chỗ nó dẫn xuất khối tham chiếu trong ảnh hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một kỹ thuật trong số các kỹ thuật dự đoán liên ảnh được mô tả trong sáng chế. Chế độ bảng màu có thể được coi như là ví dụ của việc tạo mã trong ảnh hoặc việc dự đoán trong ảnh. Khi chế độ bảng màu được áp dụng, thì thông tin về bảng của bảng màu và chỉ số bảng màu có thể được gồm trong thông tin video/hình ảnh và được phát tín hiệu.

Bộ dự đoán trong ảnh 331 có thể dự đoán khối hiện tại với sự tham khảo đến các mẫu nằm trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt lân cận với khối hiện tại theo chế độ dự đoán, hoặc cũng có thể được đặt cách khối khối hiện tại. Các chế độ dự đoán trong việc dự đoán trong ảnh có thể gồm các chế độ không có hướng và các

chế độ có hướng. Bộ dự đoán trong ảnh 331 cũng có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại nhờ sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên ảnh 332 có thể sinh ra khối được dự đoán của khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi véc tơ chuyển động trên ảnh tham chiếu. Lúc này, để giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên ảnh, thì thông tin chuyển động có thể được dự đoán trong các đơn vị của khối, khối con, hoặc mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm véc tơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin về hướng dự đoán liên ảnh (dự đoán L0, dự đoán L1, dự đoán đôi, v.v.). Trong trường hợp dự đoán liên ảnh, thì khối lân cận có thể gồm khối lân cận về mặt không gian tồn tại trong ảnh hiện tại và khối lân cận về mặt thời gian tồn tại trong ảnh tham chiếu. Ví dụ, bộ dự đoán liên ảnh 332 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận, và dẫn xuất véc tơ chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại dựa trên thông tin lựa chọn ứng viên nhận được. Việc dự đoán liên ảnh 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 ảnh của khối hiện tại.

Bộ cộng 340 có thể cộng tín hiệu dư giành được vào tín hiệu được dự đoán (khối được dự đoán, mảng mẫu dự đoán) mà được xuất ra từ bộ dự đoán (gồm bộ dự đoán liên ảnh 332 và/hoặc bộ dự đoán trong ảnh 331) để tạo ra tín hiệu được tái tạo (ảnh được tái tạo, khối được tái tạo, mảng mẫu được tái tạo). Như trong trường hợp chế độ bỏ qua được áp dụng, nếu không có phần dư nào cho khối cần được xử lý, thì khối được dự đoán có thể được sử dụng làm khối được tái tạo.

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

Trong lúc 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 áp dụng việc lọc cho tín hiệu được tái tạo, nhờ đó cải thiện các chất lượng hình ảnh chủ quan/khách quan. Ví dụ, bộ lọc 350 có thể áp dụng các phương pháp lọc khác nhau cho ảnh được tái tạo để tạo ra ảnh được tái tạo mà được sửa đổi, và truyền ảnh được tái tạo mà được sửa đổi đến bộ nhớ 360, một cách cụ thể, DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch thích ứng mẫu, lọc vòng thích ứng, lọc song phương, và tương tự.

Ảnh được tái tạo (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 ảnh 332. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối mà trong đó thông tin chuyển động trong ảnh hiện tại được dẫn xuất (được giải mã) và/hoặc thông tin chuyển động của các khối trong ảnh được tái tạo trước đó. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ dự đoán liên ảnh 260 để được tận dụng làm thông tin chuyển động của khối lân cận về mặt không gian hoặc thông tin chuyển động của khối lân cận về mặt thời gian. Bộ nhớ 360 có thể lưu trữ các mẫu được tái tạo của các khối được tái tạo nằm trong ảnh hiện tại và truyền các mẫu được tái tạo được lưu trữ đến bộ dự đoán trong ảnh 331.

Trong sáng chế, các phương án được mô tả trong bộ lọc 260, bộ dự đoán liên ảnh 221, và bộ dự đoán trong ảnh 222 của thiết bị mã hoá 200 có thể được áp dụng theo cách

tương tự cho bộ lọc 350, bộ dự đoán liên ảnh 332, và bộ dự đoán trong ảnh 331 của thiết bị giải mã 300.

Trong lúc đó, như được mô tả ở trên, khi thực hiện việc tạo mã video, thì việc dự đoán được thực hiện để nâng cao hiệu quả nén. Thông qua việc này, khối được dự đoán gồm các mẫu dựa đoán cho khối hiện tại như khối cần được tạo mã (tức là, khối mục tiêu tạo 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 thức giống như trong thiết bị mã hóa và thiết bị giải mã, và thiết bị mã hóa có thể phát tín hiệu thông tin (thông tin phần dư) về phần dư giữa khối gốc và khối được dự đoán chứ không phải giá trị mẫu gốc của khối gốc đến thiết bị giải mã, nhờ đó tăng hiệu quả tạo mã ảnh. Thiết bị giải mã có thể dẫn xuất khối dư gồm các mẫu dư dựa trên thông tin phần dư, cộng khối dư và khối được dự đoán để tạo ra các khối được tái tạo gồm các mẫu được tái tạo, và tạo ra hình ảnh được tái tạo gồm các khối được tái tạo.

Thông tin phần dư có thể được tạo ra thông qua thủ tục biến đổi và lượng tử hóa. Ví dụ, thiết bị mã hóa có thể dẫn xuất khối dư giữa khối gốc và khối được dự đoán, thực hiện thủ tục biến đổi trên các mẫu dư (mảng mẫu dư) được gồm trong khối dư để dẫn xuất các hệ số biến đổi, thực hiện thủ tục lượng tử hóa trên các hệ số biến đổi để dẫn xuất các hệ số biến đổi được lượng tử hóa, và phát tín 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, tham số lượng tử hóa, và tương tự. Thiết bị giải mã có thể thực hiện thủ tục khử lượng tử hóa/biến đổi ngược dựa trên thông tin phần dư và dẫn xuất các mẫu dư (hoặc các khối dư). Thiết bị giải mã có thể tạo ra ảnh được tái tạo dựa trên khối được dự đoán và khối dư. Ngoài ra, đối với sự tham chiếu cho việc dự đoán liên ảnh ảnh sau đó,

thì thiết bị mã hoá cũng có thể khử lượng tử hóa/biến đổi ngược các hệ số biến đổi được lượng tử hóa để dẫn xuất khối dư và tạo ra ảnh được tái tạo dựa trên đó.

Như được mô tả ở trên, ít nhất một hoạt động trong số việc lượng tử hóa/khử lượng tử hóa và/hoặc việc biến đổi/biến đổi ngược có thể được bỏ qua. Khi việc biến đổi/biến đổi ngược được bỏ qua, thì các hệ số biến đổi có thể được gọi là các hệ số hoặc các hệ số dư, hoặc có thể vẫn được gọi là các hệ số biến đổi để nhất quán trong việc biểu diễn. Việc liệu việc biến đổi/biến đổi ngược có được bỏ qua hay không có thể được phát tín hiệu dựa trên `transform_skip_flag`. Ví dụ, khi giá trị của `transform_skip_flag` là 1, thì nó có thể chỉ ra rằng việc biến đổi/việc biến đổi ngược được bỏ qua, vốn có thể được gọi là chế độ bỏ qua biến đổi.

Nói chung, trong việc tạo mã video/hình ảnh, tốc độ lượng tử hóa có thể được thay đổi, và tốc độ so sánh có thể được điều chỉnh nhờ sử dụng tốc độ lượng tử hóa được thay đổi. Từ góc độ triển khai, thông số lượng tử hóa (Quantization Parameter, QP) có thể được sử dụng thay cho tốc độ lượng tử hóa khi xem xét đến độ phức tạp. Ví dụ, các thông số lượng tử hóa có các giá trị nguyên từ 0 đến 63 có thể được sử dụng, và mỗi giá trị thông số lượng tử hóa có thể tương ứng với tốc độ lượng tử hóa thực tế. Ví dụ, thông số lượng tử hóa QP_Y cho thành phần độ chói (mẫu độ chói) và thông số lượng tử hóa QP_C cho thành phần sắc độ (mẫu sắc độ) có thể được thiết lập theo kiểu khác nhau.

Quy trình lượng tử hóa có thể lấy hệ số biến đổi C làm đầu vào, phân chia nó bởi tốc độ lượng tử hóa Q_{step} , và dựa trên việc này, thu nhận hệ số biến đổi được lượng tử hóa C' . Trong trường hợp này, khi xem xét đến độ phức tạp tính toán, thì tốc độ lượng tử hóa có thể được nhân với thang định cỡ để tạo thành số nguyên, và phép toán dịch chuyển có thể được thực hiện theo giá trị tương ứng với giá trị định cỡ. Thang định cỡ lượng tử hóa có thể được dẫn xuất dựa trên tích của tốc độ lượng tử hóa và giá trị định

cỡ. Nghĩa là, thang định cỡ lượng tử hóa có thể được dẫn xuất theo QP. Ví dụ, hệ số biến đổi được lượng tử hóa C' có thể được dẫn xuất bằng cách áp dụng thang định cỡ lượng tử hóa cho hệ số biến đổi C .

Quy trình khử lượng tử hóa là quy trình đảo ngược của quy trình lượng tử hóa và có thể thu nhận hệ số biến đổi được tái tạo C'' bằng cách nhân hệ số biến đổi được lượng tử hóa C' với tốc độ lượng tử hóa Q_{step} . Trong trường hợp này, thang định cỡ mức độ có thể được dẫn xuất theo thông số lượng tử hóa, và hệ số biến đổi được tái tạo C'' có thể được dẫn xuất bằng cách áp dụng thang định cỡ mức độ cho hệ số biến đổi được lượng tử hóa C' . Hệ số biến đổi được tái tạo C'' có thể khác một chút với hệ số biến đổi gốc C do sự mất mát trong quy trình biến đổi và/hoặc lượng tử hóa. Theo đó, trong thiết bị mã hóa, việc khử lượng tử hóa được thực hiện theo cùng cách thức như trong thiết bị giải mã.

Trong lúc đó, việc dự đoán có thể được thực hiện dựa trên việc tạo mã bảng màu. Việc tạo mã bảng màu là kỹ thuật hữu ích để biểu diễn các khối gồm số lượng nhỏ các giá trị màu sắc duy nhất. Thay vì áp dụng việc dự đoán và việc biến đổi cho các khối, thì chỉ số để chỉ ra giá trị màu sắc của mỗi mẫu được phát tín hiệu trong chế độ bảng màu. Chế độ bảng màu này là hữu ích để tiết kiệm không gian đệm bộ nhớ video. Khối có thể được tạo mã nhờ sử dụng chế độ bảng màu (ví dụ, `MODE_PLT`). Để giải mã khối được mã hóa theo cách thức này, thì bộ giải mã cần giải mã màu sắc và chỉ số bảng màu. Màu sắc của bảng màu có thể được biểu diễn bởi bảng của bảng màu và có thể được mã hóa bởi công cụ tạo mã bảng của bảng.

Fig.4 thể hiện ví dụ để mô tả cấu trúc cơ sở của việc tạo mã bảng màu.

Tham khảo đến Fig.4, hình ảnh 400 có thể được biểu diễn bởi biểu đồ 410. Ở đây, các giá trị màu sắc trội được ánh xạ theo cách điển hình đến chỉ số màu sắc 420 và hình ảnh có thể được tạo mã nhờ sử dụng bản đồ chỉ số màu sắc 430.

Việc tạo mã bảng màu có thể được gọi là chế độ bảng màu, chế độ tạo mã bảng màu (trong ảnh), hoặc tương tự. Khối hiện tại có thể được tái tạo theo việc tạo mã bảng màu hoặc chế độ bảng màu. Việc tạo mã bảng màu có thể được coi như là ví dụ về việc tạo mã trong ảnh hoặc có thể được coi như là một phương pháp trong số các phương pháp dự đoán trong ảnh. Tuy nhiên, theo cách tương tự với chế độ bỏ qua được mô tả ở trên, các giá trị dư bổ sung cho khối tương ứng có thể không được phát tín hiệu.

Ví dụ, chế độ bảng màu có thể được sử dụng để cải thiện hiệu quả tạo mã cho nội dung màn hình như là video được tạo ra bởi máy tính mà chứa lượng văn bản và đồ họa đáng kể. Nói chung, khu vực cục bộ của nội dung màn hình có một vài màu sắc được phân biệt bởi các cạnh sắc nét. Để tận dụng tính chất này, thì chế độ bảng màu có thể biểu diễn các mẫu của khối dựa trên các chỉ số chỉ ra các mục nhập màu sắc trong bảng của bảng màu.

Ví dụ, thông tin về bảng của bảng màu có thể được phát tín hiệu. Bảng của bảng màu có thể gồm giá trị chỉ số tương ứng với mỗi màu sắc. Dữ liệu dự đoán chỉ số bảng màu có thể được nhận, và bảng của bảng màu có thể gồm dữ liệu chỉ ra các giá trị chỉ số cho ít nhất một phần của bản đồ chỉ số bảng màu mà ánh xạ các điểm ảnh của dữ liệu video đến các chỉ số màu sắc của bảng của bảng màu. Dữ liệu dự đoán chỉ số bảng màu có thể gồm dữ liệu giá trị chạy mà liên kết các giá trị chỉ số cho ít nhất một phần của bản đồ chỉ số bảng màu với các giá trị chạy. Giá trị chạy có thể được liên kết với chỉ số màu sắc thoát. Bản đồ chỉ số bảng màu có thể được tạo ra từ dữ liệu dự đoán chỉ số bảng màu, ít nhất trong một phần, bằng cách xác định xem liệu có điều chỉnh giá trị chỉ số

của dữ liệu dự đoán chỉ số bảng màu dựa trên giá trị chỉ số cuối cùng hay không. Khởi hiện tại nằm trong ảnh có thể được tái tạo theo bản đồ chỉ số bảng màu.

Khi chế độ bảng màu được sử dụng, các giá trị điểm ảnh của CU có thể được biểu diễn bởi tập hợp của các giá trị màu sắc biểu diễn. Tập hợp như vậy có thể được gọi là bảng màu. Trong trường hợp của điểm ảnh có giá trị gần với giá trị màu sắc trong bảng màu, thì chỉ số bảng màu tương ứng với giá trị màu sắc trong bảng màu có thể được phát tín hiệu. Trong trường hợp của điểm ảnh có giá trị màu sắc khác với bảng màu, thì điểm ảnh có thể được biểu diễn bởi ký hiệu thoát và giá trị điểm ảnh được lượng tử hóa có thể được phát tín hiệu một cách trực tiếp. Trong sáng chế, điểm ảnh hoặc giá trị điểm ảnh có thể được gọi là mẫu hoặc giá trị mẫu.

Để giải mã khỏi được mã hóa trong chế độ bảng màu, thì bộ giải mã cần giải mã các màu sắc và các chỉ số bảng màu. Các màu sắc của bảng màu có thể được biểu diễn trong bảng của bảng màu và được mã hóa với công cụ tạo mã bảng của bảng. Cờ thoát có thể chỉ ra việc liệu ký hiệu thoát có mặt trong CU hiện tại hay không bằng cách được phát tín hiệu cho mỗi CU. Nếu ký hiệu thoát có mặt, thì bảng của bảng màu được tăng lên 1 và chỉ số cuối cùng có thể được chỉ định cho chế độ thoát. Bảng màu các chỉ số của tất cả các điểm ảnh trong CU có thể tạo thành bản đồ chỉ số bảng màu và có thể được mã hóa bởi công cụ tạo mã bản đồ chỉ số bảng màu.

Ví dụ, bộ dự đoán bảng màu có thể được duy trì để tạo mã bảng của bảng màu. Bộ dự đoán có thể được khởi tạo tại lúc bắt đầu của mỗi lát mà tại đó bộ dự đoán được thiết lập lại thành 0. Đối với mỗi mục nhập của bộ dự đoán bảng màu, thì cờ sử dụng lại có thể được phát tín hiệu để chỉ ra việc liệu nó có phải là một phần của bảng màu hiện tại hay không. Cờ sử dụng lại có thể được truyền nhờ sử dụng việc tạo mã thời gian chạy bằng 0. Sau đó, số lượng các mục nhập bảng màu mới có thể được phát tín hiệu nhờ sử

dụng việc tạo mã Golomb hàm mũ bậc 0. Cuối cùng, giá trị thành phần cho bảng màu mục nhập mới có thể được phát tín hiệu. Sau khi mã hóa CU hiện tại, thì bộ dự đoán bảng màu có thể được cập nhật nhờ sử dụng bảng màu hiện tại, và các mục nhập của bộ dự đoán bảng màu cũ mà không được sử dụng lại trong bảng màu hiện tại có thể được bổ sung vào phần cuối của bộ dự đoán bảng màu mới cho đến khi đạt được kích cỡ tối đa được chấp thuận (nhồi bảng màu).

Ví dụ, chỉ số có thể được tạo mã nhờ sử dụng các việc quét ngang qua theo chiều ngang và theo chiều dọc để tạo mã bản đồ chỉ số bảng màu. Thứ tự quét có thể được phát tín hiệu một cách rõ ràng từ luồng bit nhờ sử dụng thông tin cờ (ví dụ, `palette_transpose_flag`).

Fig.5 thể hiện ví dụ để mô tả các phương pháp quét ngang qua theo chiều ngang và theo chiều dọc được sử dụng để tạo mã bản đồ chỉ số bảng màu.

Fig.5(a) thể hiện ví dụ về việc tạo mã bản đồ chỉ số bảng màu nhờ sử dụng việc quét ngang qua theo chiều ngang và Fig.5(b) thể hiện ví dụ về việc tạo mã bản đồ chỉ số bảng màu nhờ sử dụng việc quét ngang qua theo chiều dọc.

Như được thể hiện trên Fig.5(a), khi việc quét theo chiều ngang được sử dụng, thì các mẫu trên hàng thứ nhất (hàng trên cùng) đến các mẫu trên hàng cuối cùng (hàng dưới cùng) trong khối hiện tại (tức là, CU hiện tại) có thể được quét theo hướng ngang để tạo mã các chỉ số bảng màu.

Như được thể hiện trên Fig.5(b), khi việc quét theo chiều dọc được sử dụng, các mẫu trên cột thứ nhất (cột ngoài cùng bên trái) đến các mẫu trên cột cuối cùng (cột ngoài cùng bên phải) trong khối hiện tại (tức là, CU hiện tại) có thể được quét theo hướng dọc để tạo mã các chỉ số bảng màu.

Chỉ số bảng màu có thể được tạo mã nhờ sử dụng hai chế độ mẫu bảng màu, ví dụ, chế độ “INDEX” và chế độ “COPY_ABOVE”. Chế độ bảng màu như vậy có thể được phát tín hiệu nhờ sử dụng cờ chỉ ra việc liệu chế độ là “INDEX” hay là “COPY_ABOVE”. Ở đây, cờ có thể được phát tín hiệu ngoại trừ hàng trên cùng khi việc quét theo chiều ngang được sử dụng, và cờ có thể được phát tín hiệu ngoại trừ cột thứ nhất A khi việc quét theo chiều dọc được sử dụng hoặc khi chế độ trước đó là chế độ “COPY_ABOVE”. Trong chế độ “COPY_ABOVE”, chỉ số bảng màu của mẫu trong hàng ở trên có thể được sao chép. Trong chế độ “INDEX”, chỉ số bảng màu có thể được phát tín hiệu một cách rõ ràng. Đối với cả chế độ “INDEX” và chế độ “COPY_ABOVE”, thì giá trị chạy chỉ ra số lượng các điểm ảnh được tạo mã nhờ sử dụng cùng chế độ có thể được phát tín hiệu.

Thứ tự mã hóa cho bản đồ chỉ số là như sau. Trước tiên, số lượng các giá trị chỉ số cho CU có thể được phát tín hiệu. Sau đó, các giá trị chỉ số thực tế cho toàn bộ CU có thể được phát tín hiệu nhờ sử dụng việc tạo mã nhị phân bị cắt cụt (Truncated Binary, TB). Cả số lượng các chỉ số và các giá trị chỉ số đều có thể được tạo mã trong chế độ đi vòng. Trong trường hợp này, các ngăn đi vòng liên quan đến chỉ số có thể được tạo nhóm với nhau. Tiếp theo, chế độ bảng màu (chế độ “INDEX” hoặc chế độ “COPY_ABOVE”) và chạy có thể được phát tín hiệu theo cách thức xen kẽ. Cuối cùng, các giá trị thoát thành phần tương ứng với các mẫu thoát cho toàn bộ CU có thể được tạo nhóm với nhau và được tạo mã trong chế độ đi vòng. Sau khi phát tín hiệu các giá trị chỉ số, thì phần tử cú pháp bổ sung last_run_type_flag có thể được phát tín hiệu. Phần tử cú pháp này không cần phát tín hiệu giá trị chạy tương ứng với lần chạy cuối cùng trong khối, cùng với số lượng các chỉ số.

Trong lúc đó, trong tiêu chuẩn VVC, thì cây kép có thể được cho phép cho I lát mà tách rời việc phân vùng đơn vị tạo mã cho độ chói và sắc độ. Việc tạo mã bảng màu (chế độ bảng màu) có thể được áp dụng cho độ chói (thành phần Y) và sắc độ (thành phần Cb và Cr) một cách riêng lẻ hoặc cùng nhau. Khi cây kép bị vô hiệu hóa thì việc tạo mã bảng màu (chế độ bảng màu) có thể được áp dụng cho độ chói (thành phần Y) và sắc độ (thành phần Cb và Cr) cùng nhau.

Fig.6 là sơ đồ để mô tả ví dụ về phương pháp tạo mã dựa trên chế độ bảng màu.

Tham khảo đến Fig.6, thiết bị giải mã có thể thu nhận thông tin bảng màu dựa trên luồng bit và/hoặc thông tin bảng màu trước đó (S600).

Trong một phương án, thiết bị giải mã có thể đi ngang qua thông tin chỉ số bảng màu, thông tin (thứ tự quét) hướng đi ngang qua, các mẫu trong CU từ luồng bit để nhận thông tin chế độ bảng màu cho mỗi vị trí mẫu và thông tin thời gian chạy của mỗi chế độ bảng màu.

Thiết bị giải mã có thể tạo cấu hình bảng màu dựa trên thông tin bảng màu (S610).

Trong phương án, thiết bị giải mã có thể tạo cấu hình bộ dự đoán bảng màu. Thông tin bảng màu được sử dụng trong khối trước đó có thể được lưu trữ cho CU bảng màu tiếp theo (tức là, CU được tạo mã trong chế độ bảng màu) sẽ được tạo ra sau, và điều này có thể được định rõ là mục nhập bộ dự đoán bảng màu. Bên cạnh đó, thiết bị giải mã có thể nhận thông tin mục nhập bảng màu mới và tạo cấu hình bảng màu cho CU hiện tại. Ví dụ, lúc nhận thông tin sử dụng lại bộ dự đoán bảng màu và thông tin mục nhập bảng màu mới cần được sử dụng trong CU hiện tại, thì thiết bị giải mã có thể tổ hợp hai mẫu thông tin mục nhập này để tạo thành một bảng màu biểu diễn CU hiện tại.

Thiết bị giải mã có thể dẫn xuất giá trị mẫu (giá trị dự đoán mẫu) trong khối hiện tại dựa trên bảng màu (S620).

Trong phương án, thiết bị giải mã có thể tạo cấu hình các mẫu từ thông tin bảng màu được thu nhận trong lúc đi ngang qua các mẫu trong CU theo hướng ngang hoặc hướng dọc dựa trên thông tin (thứ tự quét) hướng đi ngang qua. Nếu thông tin chế độ bảng màu chỉ ra chế độ COPY_ABOVE, thì mỗi giá trị mẫu trong CU có thể được dẫn xuất bằng cách sao chép thông tin chỉ số của vị trí mẫu bên trái trong việc quét theo chiều dọc và sao chép thông tin chỉ số của vị trí mẫu bên trên trong việc quét theo chiều ngang. Nghĩa là, bằng cách dẫn xuất giá trị màu sắc của mỗi mẫu từ bảng của bảng màu được tạo cấu hình dựa trên thông tin chỉ số của mỗi mẫu trong CU, thì các mẫu dự đoán trong CU có thể được dẫn xuất. Sau đó, thiết bị giải mã có thể tạo cấu hình lại mỗi thông tin mẫu trong CU nhờ sử dụng thông tin bảng màu và cập nhật bộ dự đoán bảng màu.

Trong lúc đó, việc tạo mã bảng màu được mô tả ở trên (chế độ bảng màu hoặc chế độ tạo mã bảng màu) có thể phát tín hiệu thông tin để chỉ ra việc liệu CU hiện tại có được tạo mã trong chế độ bảng màu hay không và tạo mã nó bằng cách áp dụng chế độ bảng màu cho nó.

Theo ví dụ, thông tin về việc liệu chế độ tạo mã bảng màu có khả dụng hay không có thể được phát tín hiệu thông qua tập hợp thông số chuỗi (SPS) như được thể hiện trong bảng 1 dưới đây.

[Bảng 1]

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>sps_decoding_parameter_set_id</code>	<code>u(4)</code>
<code>if(chroma_format_idc == 3)</code>	
<code>sps_palette_enabled_flag</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 1 có thể được thể hiện như được thể hiện trong bảng 2 dưới đây.

[Bảng 2]

sps_palette_enabled_flag equal to 1 specifies that <code>pred_mode_plt_flag</code> may be present in the coding unit syntax. <code>sps_palette_enabled_flag</code> equal to 0 specifies that <code>pred_mode_plt_flag</code> is not present in the coding unit syntax. When <code>sps_palette_enabled_flag</code> is not present, it is inferred to be equal to 0. Additionally, during the decoding of the coding unit (CU), a flag is enabled to indicate if the current CU is coded in the palette mode.

Tham khảo đến các bảng 1 và 2, phần tử cú pháp `sps_palette_enabled_flag` có thể được phân tích cú pháp/được phát tín hiệu trong SPS. Phần tử cú pháp `sps_palette_enabled_flag` có thể chỉ ra việc liệu chế độ tạo mã bảng màu có khả dụng hay không. Ví dụ, khi giá trị của `sps_palette_enabled_flag` là 1, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu là khả dụng, và trong trường hợp này, thông tin (ví dụ, `pred_mode_plt_flag`) chỉ ra việc liệu có áp dụng chế độ tạo mã bảng màu cho đơn vị tạo mã hiện tại trong cú pháp đơn vị tạo mã hay không có thể được phân tích cú pháp/được phát tín hiệu. Khi giá trị của `sps_palette_enabled_flag` là 0, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu là không khả dụng, và trong trường hợp này, thông tin (ví dụ, `pred_mode_plt_flag`) chỉ ra việc liệu có áp dụng chế độ tạo mã bảng màu cho đơn vị tạo

mã hiện tại trong cú pháp đơn vị tạo mã hay không có thể không được phân tích cú pháp/được phát tín hiệu.

Bên cạnh đó, ví dụ, thông tin về việc liệu có thực hiện tạo mã bằng cách áp dụng chế độ bảng màu hay không có thể được phát tín hiệu dựa trên thông tin về việc liệu chế độ tạo mã bảng màu có khả dụng hay không (ví dụ, `sps_palette_enabled_flag`), và thông tin có thể được phát tín hiệu thông qua cú pháp đơn vị tạo mã như được thể hiện trong bảng 3 dưới đây.

[Bảng 3]

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	Descriptor
<code>chType = treeType == DUAL_TREE_CHROMA ? 1 : 0</code>	
<code>if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {</code>	
<code>if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag)</code>	
<code>cu_skip_flag[x0][y0]</code>	ae(v)
<code>if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)</code>	
<code>pred_mode_flag</code>	ae(v)
<code>if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)</code>	
<code>pred_mode_ibc_flag</code>	ae(v)
<code>if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)</code>	
<code>pred_mode_plt_flag</code>	ae(v)
<code>}</code>	
<code>if(CuPredMode[chType][x0][y0] == MODE_INTRA CuPredMode[chType][x0][y0] == MODE_PLT) {</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>if(pred_mode_plt_flag) {</code>	
<code>if(treeType == DUAL_TREE_LUMA)</code>	
<code>palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)</code>	
<code>else /* SINGLE_TREE */</code>	
<code>palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)</code>	
<code>}</code>	
<code>...</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 3 có thể được thể hiện như được thể hiện trong bảng 4 dưới đây.

[Bảng 4]

pred_mode_plt_flag specifies the use of palette mode in the current coding unit. pred_mode_plt_flag equal to 1 indicates that palette mode is applied in the current coding unit. pred_mode_plt_flag equal to 0 indicates that palette mode is not applied in the current coding unit. When pred_mode_plt_flag is not present, it is inferred to be equal to 0. When pred_mode_plt_flag is equal to 1, the variable CuPredMode[x][y] is set to be equal to MODE_PLT for $x = x0..x0 + cbWidth - 1$ and $y = y0..y0 + cbHeight - 1$.

Tham khảo đến các bảng 3 và 4, phần tử cú pháp pred_mode_plt_flag có thể được phân tích cú pháp/được phát tín hiệu trong cú pháp đơn vị tạo mã. Phần tử cú pháp pred_mode_plt_flag có thể chỉ ra việc liệu chế độ bảng màu có được áp dụng cho đơn vị tạo mã hiện tại hay không. Ví dụ, khi giá trị của pred_mode_plt_flag là 1, thì nó có thể chỉ ra rằng chế độ bảng màu được áp dụng cho đơn vị tạo mã hiện tại, và nếu giá trị của pred_mode_plt_flag là 0, thì nó có thể chỉ ra rằng chế độ bảng màu không được áp dụng cho đơn vị tạo mã hiện tại.

Trong trường hợp này, pred_mode_plt_flag có thể được phân tích cú pháp/được phát tín hiệu dựa trên thông tin (ví dụ, sps_palette_enabled_flag) về việc liệu chế độ tạo mã bảng màu có khả dụng hay không. Ví dụ, khi giá trị của sps_palette_enabled_flag là 1 (tức là, khi chế độ tạo mã bảng màu là khả dụng), thì pred_mode_plt_flag có thể được phân tích cú pháp/được phát tín hiệu.

Bên cạnh đó, việc tạo mã có thể được thực hiện bằng cách áp dụng chế độ bảng màu cho đơn vị tạo mã hiện tại dựa trên pred_mode_plt_flag. Ví dụ, khi giá trị của pred_mode_plt_flag là 1, thì chế độ bảng màu có thể được áp dụng cho đơn vị tạo mã hiện tại để tạo ra mẫu được tái tạo bằng cách phân tích cú pháp/phát tín hiệu cú pháp palette_coding().

Theo ví dụ, bảng 5 dưới đây thể hiện cú pháp tạo mã bảng màu.

[Bảng 5]

palette_coding(x0, y0, cbWidth, cbHeight, startComp, numComps) {	Descriptor
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
for(predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[startComp] && !palettePredictionFinished && NumPredictedPaletteEntries[startComp] < palette_max_size; predictorEntryIdx++) {	
palette_predictor_run	ae(v)
if(palette_predictor_run != 1) {	
if(palette_predictor_run > 1)	
predictorEntryIdx += palette_predictor_run - 1	
PalettePredictorEntryReuseFlags[predictorEntryIdx] = 1	
NumPredictedPaletteEntries++	
} else	
palettePredictionFinished = 1	
}	
if(NumPredictedPaletteEntries < palette_max_size)	
num_signalled_palette_entries	ae(v)
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(i = 0; i < num_signalled_palette_entries; i++)	
new_palette_entries[cIdx][i]	ae(v)
if(CurrentPaletteSize[startComp] > 0)	
palette_escape_val_present_flag	ae(v)
if(MaxPaletteIndex > 0) {	
num_palette_indices_minus1	ae(v)
adjust = 0	
for(i = 0; i <= num_palette_indices_minus1; i++) {	
if(MaxPaletteIndex - adjust > 0) {	
palette_idx_idc	ae(v)
PaletteIndexIdc[i] = palette_idx_idc	
}	
adjust = 1	
}	
copy_above_indices_for_final_run_flag	ae(v)
palette_transpose_flag	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	
if(treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag) {	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag)	

cu_chroma_qp_offset_idx	ae(v)
}	
}	
remainingNumIndices = num_palette_indices_minus1 + 1	
PaletteScanPos = 0	
log2CbWidth = Log2(cbWidth)	
log2CbHeight = Log2(cbHeight)	
while(PaletteScanPos < cbWidth*cbHeight) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
if(PaletteScanPos > 0) {	
xcPrev =	
x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos - 1][0]	
ycPrev =	
y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos - 1][1]	
}	
PaletteRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1	
RunToEnd = 1	
CopyAboveIndicesFlag[xC][yC] = 0	
if(MaxPaletteIndex > 0)	
if((!palette_transpose_flag && yC > 0) (palette_transpose_flag && xC > 0))	
&& CopyAboveIndicesFlag[xcPrev][ycPrev] == 0)	
if(remainingNumIndices > 0 && PaletteScanPos < cbWidth*cbHeight - 1) {	
copy_above_palette_indices_flag	ae(v)
CopyAboveIndicesFlag[xC][yC] = copy_above_palette_indices_flag	
} else {	
if(PaletteScanPos == cbWidth * cbHeight - 1 && remainingNumIndices > 0)	
CopyAboveIndicesFlag[xC][yC] = 0	
else	
CopyAboveIndicesFlag[xC][yC] = 1	
}	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
currNumIndices = num_palette_indices_minus1 + 1 - remainingNumIndices	
PaletteIndexMap[xC][yC] = PaletteIndexIdc[currNumIndices]	
}	
if(MaxPaletteIndex > 0) {	
if(CopyAboveIndicesFlag[xC][yC] == 0)	
remainingNumIndices -= 1	
if(remainingNumIndices > 0 CopyAboveIndicesFlag[xC][yC] != copy_above_indices_for_final_run_flag) {	
PaletteMaxRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1 - remainingNumIndices - copy_above_indices_for_final_run_flag	
RunToEnd = 0	
if(PaletteMaxRunMinus1 > 0) {	
palette_run_prefix	ae(v)

if((palette_run_prefix > 1) && (PaletteMaxRunMinus1 != (1 << (palette_run_prefix - 1))))	
palette_run_suffix	ae(v)
}	
}	
}	
runPos = 0	
while (runPos <= PaletteRunMinus1) {	
xR = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yR = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
CopyAboveIndicesFlag[xR][yR] = 0	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xC][yC]	
} else {	
CopyAboveIndicesFlag[xR][yR] = 1	
if(!palette_transpose_flag)	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR - 1]	
else	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR - 1][yR]	
}	
runPos++	
PaletteScanPos ++	
}	
}	
if(palette_escape_val_present_flag) {	
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(sPos = 0; sPos < cbWidth* cbHeight; sPos++) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][1]	
if(PaletteIndexMap[cIdx][xC][yC] == MaxPaletteIndex) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
}	
}	
}	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 5 có thể được thể hiện như được thể hiện trong bảng 6 dưới đây.

[Bảng 6]

In the following semantics, the array indices x_0, y_0 specify the location (x_0, y_0) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture. The array indices x_C, y_C specify the location (x_C, y_C) of the sample relative to the top-left luma sample of the picture. The array index $startComp$ specifies the first colour component of the current palette table. $startComp$ equal to 0 indicates the Y component; $startComp$ equal to 1 indicates the Cb component; $startComp$ equal to 2 indicates the Cr component. $numComps$ specifies the number of colour components in the current palette table.

The predictor palette consists of palette entries from previous coding units that are used to predict the entries in the current palette.

The variable $PredictorPaletteSize[startComp]$ specifies the size of the predictor palette for the first colour component of the current palette table $startComp$. $PredictorPaletteSize$ is derived as specified in Section 1.4

The variable $PalettePredictorEntryReuseFlags[i]$ equal to 1 specifies that the i -th entry in the predictor palette is reused in the current palette. $PalettePredictorEntryReuseFlags[i]$ equal to 0 specifies that the i -th entry in the predictor palette is not an entry in the current palette. All elements of the array $PalettePredictorEntryReuseFlags[i]$ are initialized to 0.

palette_predictor_run is used to determine the number of zeros that precede a non-zero entry in the array $PalettePredictorEntryReuseFlags$.

It is a requirement of bitstream conformance that the value of $palette_predictor_run$ shall be in the range of 0 to ($PredictorPaletteSize - predictorEntryIdx$), inclusive, where $predictorEntryIdx$ corresponds to the current position in the array $PalettePredictorEntryReuseFlags$. The variable $NumPredictedPaletteEntries$ specifies the number of entries in the current palette that are reused from the predictor palette. The value of $NumPredictedPaletteEntries$ shall be in the range of 0 to $palette_max_size$, inclusive.

num_signalled_palette_entries specifies the number of entries in the current palette that are explicitly signalled for the first colour component of the current palette table $startComp$.

When $num_signalled_palette_entries$ is not present, it is inferred to be equal to 0.

The variable $CurrentPaletteSize[startComp]$ specifies the size of the current palette for the first colour component of the current palette table $startComp$ and is derived as follows:

```
CurrentPaletteSize[ startComp ] = NumPredictedPaletteEntries +
num_signalled_palette_entries
```

The value of CurrentPaletteSize[startComp] shall be in the range of 0 to palette_max_size, inclusive.

new_palette_entries[cIdx][i] specifies the value for the i-th signalled palette entry for the colour component cIdx.

The variable PredictorPaletteEntries[cIdx][i] specifies the i-th element in the predictor palette for the colour component cIdx.

The variable CurrentPaletteEntries[cIdx][i] specifies the i-th element in the current palette for the colour component cIdx and is derived as follows:

```
numPredictedPaletteEntries = 0
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
    if( PalettePredictorEntryReuseFlags[ i ] ) {
        for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
PredictorPaletteEntries[ cIdx ][ i ]
            numPredictedPaletteEntries++
    }

for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++)
    for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
        CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] =
new_palette_entries[ cIdx ][ i ]
```

palette_escape_val_present_flag equal to 1 specifies that the current coding unit contains at least one escape coded sample. **escape_val_present_flag** equal to 0 specifies that there are no escape coded samples in the current coding unit. When not present, the value of **palette_escape_val_present_flag** is inferred to be equal to 1.

The variable `MaxPaletteIndex` specifies the maximum possible value for a palette index for the current coding unit. The value of `MaxPaletteIndex` is set equal to `CurrentPaletteSize[ startComp ] - 1 + palette_escape_val_present_flag`.

num_palette_indices_minus1 plus 1 is the number of palette indices explicitly signalled or inferred for the current block.

When `num_palette_indices_minus1` is not present, it is inferred to be equal to 0.

palette_idx_idc is an indication of an index to the palette table, `CurrentPaletteEntries`. The value of `palette_idx_idc` shall be in the range of 0 to `MaxPaletteIndex`, inclusive, for the first index in the block and in the range of 0 to $(\text{MaxPaletteIndex} - 1)$, inclusive, for the remaining indices in the block.

When `palette_idx_idc` is not present, it is inferred to be equal to 0.

The variable `PaletteIndexIdc[ i ]` stores the *i*-th `palette_idx_idc` explicitly signalled or inferred. All elements of the array `PaletteIndexIdc[ i ]` are initialized to 0.

copy_above_indices_for_final_run_flag equal to 1 specifies that the palette indices of the last positions in the coding unit are copied from the palette indices in the row above if horizontal traverse scan is used or the palette indices in the left column if vertical traverse scan is used. **copy_above_indices_for_final_run_flag** equal to 0 specifies that the palette indices of the last positions in the coding unit are copied from `PaletteIndexIdc[ num_palette_indices_minus1 ]`.

When `copy_above_indices_for_final_run_flag` is not present, it is inferred to be equal to 0.

palette_transpose_flag equal to 1 specifies that vertical traverse scan is applied for scanning the indices for samples in the current coding unit. **palette_transpose_flag** equal to 0 specifies that horizontal traverse scan is applied for scanning the indices for samples in the current coding unit. When not present, the value of `palette_transpose_flag` is inferred to be equal to 0.

The array `TraverseScanOrder` specifies the scan order array for palette coding. `TraverseScanOrder` is assigned the horizontal scan order `HorTravScanOrder` if

palette_transpose_flag is equal to 0 and TraverseScanOrder is assigned the vertical scan order VerTravScanOrder if if palette_transpose_flag is equal to 1.

copy_above_palette_indices_flag equal to 1 specifies that the palette index is equal to the palette index at the same location in the row above if horizontal traverse scan is used or the same location in the left column if vertical traverse scan is used. copy_above_palette_indices_flag equal to 0 specifies that an indication of the palette index of the sample is coded in the bitstream or inferred.

The variable CopyAboveIndicesFlag[xC][yC] equal to 1 specifies that the palette index is copied from the palette index in the row above (horizontal scan) or left column (vertical scan). CopyAboveIndicesFlag[xC][yC] equal to 0 specifies that the palette index is explicitly coded in the bitstream or inferred. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture. The value of PaletteIndexMap[xC][yC] shall be in the range of 0 to (MaxPaletteIndex – 1), inclusive.

The variable PaletteIndexMap[xC][yC] specifies a palette index, which is an index to the array represented by CurrentPaletteEntries. The array indices xC, yC specify the location (xC, yC) of the sample relative to the top-left luma sample of the picture. The value of PaletteIndexMap[xC][yC] shall be in the range of 0 to MaxPaletteIndex, inclusive.

The variable adjustedRefPaletteIndex is derived as follows:

```
adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev =
    x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos – 1 ][ 0 ]
    ycPrev =
    y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos – 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] {
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC – 1 ]
```

```

        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
        }
    }

```

When CopyAboveIndicesFlag[xC][yC] is equal to 0, the variable CurrPaletteIndex is derived as follows:

```

    if( CurrPaletteIndex >= adjustedRefPaletteIndex )
        CurrPaletteIndex++

```

palette_run_prefix, when present, specifies the prefix part in the binarization of PaletteRunMinus1.

palette_run_suffix is used in the derivation of the variable PaletteRunMinus1. When not present, the value of palette_run_suffix is inferred to be equal to 0.

When RunToEnd is equal to 0, the variable PaletteRunMinus1 is derived as follows:

- If PaletteMaxRunMinus1 is equal to 0, PaletteRunMinus1 is set equal to 0.
- Otherwise (PaletteMaxRunMinus1 is greater than 0) the following applies:
 - If palette_run_prefix is less than 2, the following applies:

```

    PaletteRunMinus1 = palette_run_prefix

```

- Otherwise (palette_run_prefix is greater than or equal to 2), the following applies:

```

    PrefixOffset = 1 << ( palette_run_prefix - 1 )
    PaletteRunMinus1 = PrefixOffset + palette_run_suffix

```

The variable PaletteRunMinus1 is used as follows:

- If CopyAboveIndicesFlag[xC][yC] is equal to 0, PaletteRunMinus1 specifies the number of consecutive locations minus 1 with the same palette index.
- Otherwise if palette_transpose_flag equal to 0, PaletteRunMinus1 specifies the number of consecutive locations minus 1 with the same palette index as used in the corresponding position in the row above.

- Otherwise, `PaletteRunMinus1` specifies the number of consecutive locations minus 1 with the same palette index as used in the corresponding position in the left column.

When `RunToEnd` is equal to 0, the variable `PaletteMaxRunMinus1` represents the maximum possible value for `PaletteRunMinus1` and it is a requirement of bitstream conformance that the value of `PaletteMaxRunMinus1` shall be greater than or equal to 0.

`palette_escape_val` specifies the quantized escape coded sample value for a component.

The variable `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` specifies the escape value of a sample for which `PaletteIndexMap[ xC ][ yC ]` is equal to `MaxPaletteIndex` and `palette_escape_val_present_flag` is equal to 1. The array index `cIdx` specifies the colour component. The array indices `xC`, `yC` specify the location (`xC`, `yC`) of the sample relative to the top-left luma sample of the picture.

It is a requirement of bitstream conformance that `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` shall be in the range of 0 to $(1 \ll (\text{BitDepth}_Y + 1)) - 1$, inclusive, for `cIdx` equal to 0, and in the range of 0 to $(1 \ll (\text{BitDepth}_C + 1)) - 1$, inclusive, for `cIdx` not equal to 0.

Tham khảo đến các bảng 5 và 6, khi chế độ bảng màu được áp dụng cho khối hiện tại (tức là, đơn vị tạo mã hiện tại), thì cú pháp tạo mã bảng màu (ví dụ, `palette_coding()`) như trong bảng 5 có thể được phân tích cú pháp/được phát tín hiệu.

Ví dụ, bảng của bảng màu có thể được tạo cấu hình dựa trên thông tin mục nhập bảng màu. Thông tin mục nhập bảng màu có thể gồm các phần tử cú pháp như là `palette_predictor_run`, `num_signalled_palette_entries`, và `new_palette_entries`.

Bên cạnh đó, có thể tạo cấu hình bản đồ chỉ số bảng màu cho khối hiện tại dựa trên thông tin chỉ số bảng màu. Thông tin chỉ số bảng màu có thể gồm các phần tử cú pháp như là `num_palette_indices_minus1`, `palette_idx_idc`, `copy_above_indices_for_final_run_flag`, và `palette_transpose_flag`. Dựa trên thông tin chỉ số bảng màu như được mô tả ở trên, thì có thể dẫn xuất bảng màu các giá trị chỉ số (ví dụ, `PaletteIndexIdc`) cho các mẫu trong khối hiện tại trong khi đi ngang qua theo

hướng quét ngang qua (hướng dọc hoặc hướng ngang) để tạo cấu hình bản đồ chỉ số bảng màu (ví dụ, `PaletteIndexMap`).

Bên cạnh đó, có thể dẫn xuất giá trị mẫu cho bảng màu mục nhập trong bảng của bảng màu dựa trên bản đồ chỉ số bảng màu và tạo ra các mẫu được tái tạo của khối hiện tại dựa trên giá trị mẫu (tức là, giá trị màu sắc) được ánh xạ đến bảng màu mục nhập.

Khi mẫu có giá trị thoát trong khối hiện tại (tức là, khi giá trị của `palette_escape_val_present_flag` là 1), thì giá trị thoát cho khối hiện tại có thể được dẫn xuất dựa trên thông tin thoát. Thông tin thoát có thể gồm các phần tử cú pháp như là `palette_escape_val_present_flag` và `palette_escape_val`. Ví dụ, giá trị thoát cho mẫu thoát được tạo mã trong khối hiện tại có thể được dẫn xuất dựa trên thông tin giá trị thoát được lượng tử hóa (ví dụ, `palette_escape_val`). Các mẫu được tái tạo của khối hiện tại có thể được tạo ra dựa trên giá trị thoát.

Như được mô tả ở trên, thông tin (các phần tử cú pháp) trong bảng cú pháp được bọc lộ trong sáng chế có thể được gồm trong thông tin hình ảnh/video, được tạo cấu hình/được mã hóa theo kỹ thuật tạo mã (gồm việc tạo mã bảng màu) được thực hiện trong thiết bị mã hóa và được chuyển đến thiết bị giải mã trong dạng luồng bit. Thiết bị giải mã có thể phân tích cú pháp/giải mã thông tin (các phần tử cú pháp) trong bảng cú pháp. Thiết bị giải mã có thể thực hiện kỹ thuật tạo mã như là việc tạo mã bảng màu dựa trên thông tin được giải mã và có thể thực hiện thủ tục tái tạo (giải mã) khối/hình ảnh/video dựa trên đó. Ở dưới đây, sáng chế đề xuất bảng cú pháp và các phần tử cú pháp để tạo mã một cách hiệu quả khối/hình ảnh/video dựa trên việc tạo mã bảng màu.

Sáng chế đề xuất phương pháp tạo mã và phát tín hiệu một cách hiệu quả các giá trị thoát trong việc tạo mã chế độ bảng màu. Trong chế độ bảng màu, các giá trị thoát có thể được sử dụng để truyền theo cách bổ sung giá trị mẫu tương ứng cho mẫu có giá trị

khác với các giá trị của các mẫu lân cận trong khối. Vì các giá trị thoát như vậy là dữ liệu bổ sung, nên việc lượng tử hóa có thể được thực hiện để tiết kiệm các giá trị thoát. Bên cạnh đó, trong việc tạo mã thoát trong chế độ bảng màu, thì không có phép biến đổi nào được áp dụng và giá trị thoát được lượng tử hóa có thể được phát tín hiệu một cách trực tiếp. Điều này có thể được coi là tương tự với chế độ bỏ qua biến đổi mà trong đó không có phép biến đổi nào được áp dụng cho đơn vị tạo mã (CU).

Trong tiêu chuẩn VVC hiện tại, phạm vi đầy đủ của các giá trị thông số lượng tử hóa (QP) được áp dụng cho các giá trị thoát trong chế độ bảng màu. Tuy nhiên, sáng chế đề xuất phương pháp giới hạn phạm vi của các giá trị QP để ngăn chặn kích cỡ bước lượng tử hóa cho giá trị thoát tạo mã trong chế độ bảng màu không trở nên nhỏ hơn 1. Trong một phương án, cùng một sự ràng buộc là QP tối thiểu cho việc bỏ qua biến đổi có thể được áp dụng cho việc tạo mã giá trị thoát trong chế độ bảng màu. QP tối thiểu cho chế độ bảng màu có thể được cắt xén nhờ sử dụng QP tối thiểu cho việc bỏ qua biến đổi.

Theo ví dụ, thông tin về QP tối thiểu cho việc bỏ qua biến đổi có thể được phát tín hiệu thông qua tập hợp thông số chuỗi (SPS) như được thể hiện trong bảng 7 dưới đây.

[Bảng 7]

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>sps_decoding_parameter_set_id</code>	<code>u(4)</code>
<code>.....</code>	
<code>min_qp_prime_ts_minus4</code>	<code>ue(v)</code>
<code>....</code>	
<code>if(chroma_format_idc == 3)</code>	
<code>sps_palette_enabled_flag</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 7 có thể được thể hiện như được thể hiện trong bảng 8 dưới đây.

[Bảng 8]

min_qp_prime_ts_minus4 specifies the minimum allowed quantization parameter for transform skip mode as follows:

$$QpPrimeTsMin = 4 + \text{min_qp_prime_ts_minus4}$$

min_qp_prime_ts_minus4 is signalled in the SPS and is typically given by $6 * (\text{inputBitDepth} - 8)$.

Tham khảo đến các bảng 7 và 8, phần tử cú pháp **min_qp_prime_ts_minus4** có thể được phân tích cú pháp/được phát tín hiệu trong SPS. Phần tử cú pháp **min_qp_prime_ts_minus4** có thể chỉ ra thông số lượng tử hóa tối thiểu được chấp thuận cho chế độ bỏ qua biến đổi. Nói cách khác, giá trị thông số lượng tử hóa tối thiểu (ví dụ, **QpPrimeTsMin**) trong chế độ bỏ qua biến đổi có thể được dẫn xuất dựa trên phần tử cú pháp **min_qp_prime_ts_minus4**. Ví dụ, giá trị thông số lượng tử hóa tối thiểu (ví dụ, **QpPrimeTsMin**) có thể được dẫn xuất bằng cách cộng 4 vào giá trị của **min_qp_prime_ts_minus4**.

Như được mô tả ở trên, dựa trên phần tử cú pháp **min_qp_prime_ts_minus4** được phát tín hiệu thông qua SPS, thì QP cho các giá trị thoát trong chế độ bảng màu có thể được dẫn xuất như trong thuật toán được bộc lộ trong bảng 9 dưới đây. Nghĩa là, giá trị QP được sử dụng để tạo cấu hình lại giá trị thoát trong quy trình giải mã dựa trên chế độ bảng màu có thể được dẫn xuất như trong thuật toán được bộc lộ trong bảng 9 dưới đây.

[Bảng 9]

– Otherwise (bIsEscapeSample is equal to 1 and cu_transquant_bypass_flag is equal to 0), the following ordered steps apply: 1. The quantization parameter qP is derived as follows: – If cIdx is equal to 0, $qP = \text{Max}(QpPrimeTsMin, Qp'Y)$ – Otherwise, if cIdx is equal to 1, $qP = \text{Max}(QpPrimeTsMin, Qp'Cb)$ – Otherwise (cIdx is equal to 2), $qP = \text{Max}(QpPrimeTsMin, Qp'Cr)$ 2. The variables bitDepth is derived as follows: $\text{bitDepth} = (cIdx == 0) ? \text{BitDepth}_Y : \text{BitDepth}_C$ 3. The list levelScale[] is specified as levelScale[k] = { 40, 45, 51, 57, 64, 72 } with k = 0..5. 4. The following applies: $\text{tmpVal} = (\text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL] * \text{levelScale}[qP \% 6]) << (qP / 6) + 32) >> 6$ $\text{recSamples}[x][y] = \text{Clip3}(0, (1 << \text{bitDepth}) - 1, \text{tmpVal})$

Tham khảo đến bảng 9, khi giá trị thoát của chế độ bảng màu tồn tại, thì giá trị QP có thể được dẫn xuất. Nghĩa là, QP cho giá trị thoát của chế độ bảng màu có thể được dẫn xuất dựa trên giá trị thông số lượng tử hóa tối thiểu (ví dụ, QpPrimeTsMin) trong chế độ bỏ qua biến đổi được dẫn xuất dựa trên phần tử cú pháp min_qp_prime_ts_minus4 được mô tả ở trên. Ví dụ, như được thể hiện trong bảng 9, thì QP cho các giá trị thoát của chế độ bảng màu có thể được dẫn xuất làm giá trị lớn hơn nằm giữa QpPrimeTsMin và thông số lượng tử hóa Qp (Qp'Y cho thành phần độ chói và Qp'Cb hoặc Qp'Cr cho thành phần sắc độ). Sau đó, giá trị thoát có thể được dẫn xuất dựa trên QP cho các giá trị thoát của chế độ bảng màu để tái tạo các mẫu trong khối.

Bên cạnh đó, trong sáng chế, như được mô tả ở trên, khi phạm vi QP trong chế độ bảng màu được giới hạn là lớn hơn hoặc bằng giá trị thông số lượng tử hóa tối thiểu (ví dụ, QpPrimeTsMin) trong chế độ bỏ qua biến đổi, thì phạm vi của các giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể bị giới hạn. Theo phương án, phạm vi của các giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể được xác định dựa trên độ sâu bit và có thể được giới hạn sao cho nó không lớn hơn, ví dụ, $(1 \ll \text{BitDepth}) - 1$.

Ví dụ, giá trị thoát được lượng tử hóa trong chế độ bảng màu có thể được biểu diễn bởi phần tử cú pháp `palette_escape_val`. Phần tử cú pháp `palette_escape_val` có thể được phát tín hiệu thông qua cú pháp tạo mã bảng màu như được thể hiện trong bảng 10 dưới đây.

[Bảng 10]

<code>palette_coding(x0, y0, cbWidth, cbHeight, treeType) {</code>	Descriptor
<code>.....</code>	
<code>if(palette_escape_val_present_flag) {</code>	
<code>for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++) {</code>	
<code>for(sPos = minSubPos; sPos < maxSubPos; sPos++) {</code>	
<code>xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][0]</code>	
<code>yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][1]</code>	
<code>if(!(treeType == SINGLE_TREE && cIdx != 0 &&</code> <code>(xC % SubWidthC != 0 yC % SubHeightC != 0)) {</code>	
<code>if(PaletteIndexMap[cIdx][xC][yC] == MaxPaletteIndex) {</code>	
<code>palette_escape_val</code>	<code>ae(v)</code>
<code>PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 10 có thể được thể hiện như được thể hiện trong bảng 11 dưới đây.

[Bảng 11]

palette_escape_val specifies the quantized escape coded sample value for a component.

The variable `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` specifies the escape value of a sample for which `PaletteIndexMap[ xC ][ yC ]` is equal to `MaxPaletteIndex` and `palette_escape_val_present_flag` is equal to 1. The array index `cIdx` specifies the colour component. The array indices `xC`, `yC` specify the location (`xC`, `yC`) of the sample relative to the top-left luma sample of the picture.

It is a requirement of bitstream conformance that `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` shall be in the range of 0 to $(1 \ll (\text{BitDepth}_Y)) - 1$, inclusive, for `cIdx` equal to 0, and in the range of 0 to $(1 \ll (\text{BitDepth}_C)) - 1$, inclusive, for `cIdx` not equal to 0.

Tham khảo đến các bảng 10 và 11, phần tử cú pháp `palette_escape_val` có thể được phân tích cú pháp/được phát tín hiệu trong cú pháp tạo mã bảng màu. Phần tử cú pháp `palette_escape_val` có thể chỉ ra giá trị thoát được lượng tử hóa. Bên cạnh đó, như được thể hiện trong bảng 10, giá trị của phần tử cú pháp `palette_escape_val` có thể được thiết lập là `PaletteEscapeVal`, và `PaletteEscapeVal` có thể chỉ ra giá trị thoát của mẫu mà trong đó bản đồ chỉ số bảng màu (`PaletteIndexMap`) bằng chỉ số bảng màu tối đa (`MaxPaletteIndex`) và giá trị của `palette_escape_val_present_flag` là 1. Ở đây, trường hợp mà tại đó giá trị của `palette_escape_val_present_flag` là 1 có thể có nghĩa là ít nhất một mẫu thoát được tạo mã (giá trị thoát) được gồm trong CU hiện tại. Ví dụ, đối với thành phần độ chói, `PaletteEscapeVal` có thể được giới hạn ở phạm vi từ 0 đến $(1 \ll (\text{BitDepth}_Y)) - 1$. Đối với thành phần sắc độ, `PaletteEscapeVal` có thể được giới hạn ở phạm vi từ 0 đến $(1 \ll (\text{BitDepth}_C)) - 1$.

Bên cạnh đó, sáng chế đề xuất phương pháp định rõ kích cỡ bảng màu và phát tín hiệu kích cỡ bảng màu. Kích cỡ bảng màu có thể chỉ ra số lượng các mục nhập trong bảng của bảng màu (tức là, số lượng các chỉ số trong bảng của bảng màu). Theo phương

án, trong sáng chế, số lượng các mục nhập trong bảng màu có thể được chỉ ra bằng cách định rõ kích cỡ bảng màu theo một hoặc nhiều hằng số.

Theo ví dụ, kích cỡ bảng màu có thể được biểu diễn bởi phần tử cú pháp `palette_max_size`, và phần tử cú pháp `palette_max_size` có thể là giống nhau đối với toàn bộ chuỗi hoặc có thể khác nhau theo kích cỡ CU (tức là, số lượng các điểm ảnh trong CU). Ví dụ, kích cỡ bảng màu (`palette_max_size`) có thể chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu và có thể được định rõ là 31. Theo một ví dụ khác, kích cỡ bảng màu (`palette_max_size`) có thể chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu, và có thể được định rõ như được thể hiện trong bảng 12 dưới đây theo kích cỡ CU.

[Bảng 12]

```
if (cbWidth*cbHeight>=1024)
    palette_max_size is defined as 63
else if cbWidth*cbHeight>=256
    palette_max_size is defined as 31
else
    palette_max_size is defined as 15
```

Các kích cỡ bảng màu 63, 31, 15, v.v., và các kích cỡ CU 1024, 256, v.v. được bộc lộ trong bảng 12 chỉ được sử dụng làm các ví dụ và có thể được thay đổi thành các số khác.

Theo phương án, thông tin chỉ ra kích cỡ bảng màu (ví dụ, `palette_max_size`) có thể được phát tín hiệu thông qua SPS như được thể hiện trong bảng 13 dưới đây.

[Bảng 13]

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
.....	
if(chroma_format_idc == 3) {	
sps_palette_enabled_flag	u(1)
if (sps_palette_enabled_flag)	
palette_max_size	u(6)
}	
...	
}	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 13 có thể được thể hiện như được thể hiện trong bảng 14 dưới đây.

[Bảng 14]

palette_max_size specifies the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive.

Tham khảo đến các bảng 13 và 14, phần tử cú pháp **palette_max_size** syntax có thể được phân tích cú pháp/được phát tín hiệu trong SPS. Phần tử cú pháp **palette_max_size** có thể chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu và có thể được giới hạn ở phạm vi từ 1 đến 63.

Trong trường hợp này, phần tử cú pháp **palette_max_size** có thể được phân tích cú pháp/được phát tín hiệu dựa trên phần tử cú pháp **sps_palette_enabled_flag**, vốn là thông tin để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không. Ví dụ, khi giá trị của **sps_palette_enabled_flag** là 1 (tức là, khi nó chỉ ra rằng chế độ bảng màu được cho phép), thì phần tử cú pháp **palette_max_size** có thể được phân tích cú pháp/được phát tín hiệu.

Theo cách thay thế, theo phương án, thông tin chỉ ra kích cỡ bảng màu (ví dụ, `log2_palette_max_size`) có thể được phát tín hiệu thông qua SPS như được thể hiện trong bảng 15 dưới đây.

[Bảng 15]

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>sps_decoding_parameter_set_id</code>	<code>u(4)</code>
<code>.....</code>	
<code>if(chroma_format_idc == 3) {</code>	
<code>sps_palette_enabled_flag</code>	<code>u(1)</code>
<code>if(sps_palette_enabled_flag)</code>	
<code>log2_palette_max_size</code>	<code>u(3)</code>
<code>}</code>	
<code>...</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 15 có thể được thể hiện như được thể hiện trong bảng 16 dưới đây.

[Bảng 16]

`log2_palette_max_size` specifies the log2 of (`palette_max_size+1`). `palette_max_size=(1<<log2_palette_max_size)-1`. `palette_max_size` specifies the the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive.

Tham khảo đến các bảng 15 và 16, phần tử cú pháp `log2_palette_max_size` có thể được phân tích cú pháp/được phát tín hiệu trong SPS. Phần tử cú pháp `log2_palette_max_size` có thể chỉ ra giá trị log2 của kích cỡ bảng màu (tức là, `palette_max_size+1`). Theo đó, `palette_max_size` chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu có thể được dẫn xuất bằng cách tính $(1 \ll \log2_palette_max_size) - 1$ và có thể được giới hạn ở phạm vi từ 1 đến 63.

Trong trường hợp này, phần tử cú pháp `log2_palette_max_size` có thể được phân tích cú pháp/được phát tín hiệu dựa trên phần tử cú pháp `sps_palette_enabled_flag`, vốn là thông tin để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không. Ví dụ, khi giá trị của `sps_palette_enabled_flag` là 1 (tức là, khi nó chỉ ra rằng chế độ bảng màu chỉ ra được cho phép), thì phần tử cú pháp `log2_palette_max_size` có thể được phân tích cú pháp/được phát tín hiệu.

Theo cách thay thế, theo phương án, thông tin chỉ ra kích cỡ bảng màu (ví dụ, `log2_palette_CU_size_TH1`, `log2_palette_max_size_TH1`, `log2_palette_max_size_default`) có thể được phát tín hiệu thông qua SPS như được thể hiện trong bảng 17 dưới đây.

[Bảng 17]

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code> sps_decoding_parameter_set_id</code>	<code>u(4)</code>
<code> </code>	
<code> if(chroma_format_idc == 3) {</code>	
<code> sps_palette_enabled_flag</code>	<code>u(1)</code>
<code> if (sps_palette_enabled_flag) {</code>	
<code> log2_palette_CU_size_TH1</code>	<code>u(5)</code>
<code> log2_palette_max_size_TH1</code>	<code>u(3)</code>
<code> log2_palette_max_size_default</code>	<code>u(3)</code>
<code> }</code>	
<code> ...</code>	
<code>}</code>	

Các ngữ nghĩa của các phần tử cú pháp được gồm trong cú pháp của bảng 17 có thể được thể hiện như được thể hiện trong bảng 18 dưới đây.

[Bảng 18]

log2_Palette_CU_size_TH1 specifies the log2 of the size limit of palette_max_size_TH1. $\text{Palette_CU_size_TH1} = 1 \ll \text{log2_Palette_CU_size_TH1}$ log2_palette_max_size_TH1 specifies the log2 of (palette_max_size_TH1+1). $\text{palette_max_size_TH1} = (1 \ll \text{log2_palette_max_size_TH1}) - 1$. palette_max_size_TH1 specifies the maximum allowed index of the palette table for CU with size greater than $\text{Palette_CU_size_TH1}$, and shall be in the range of 1 to 63 inclusive. log2_palette_max_size_default specifies the log2 of (palette_max_size_default+1). $\text{palette_max_size_default} = (1 \ll \text{log2_palette_max_size_default}) - 1$. palette_max_size_default specifies the maximum allowed index of the palette table, and shall be in the range of 1 to 63 inclusive. Variable palette_max_size specifies the maximum allowed index of a palette table, and is derived as follows: If ($\text{cbWidth} * \text{cbHeight} \geq \text{Palette_CU_size_TH1}$) palette_max_size = palette_max_size_TH1 else palette_max_size = palette_max_size_default

Tham khảo đến các bảng 17 và 18, các phân tử cú pháp log2_palette_CU_size_TH1, log2_palette_max_size_TH1, và log2_palette_max_size_default có thể được phân tích cú pháp/được phát tín hiệu trong SPS.

Phân tử cú pháp log2_palette_CU_size_TH1 chỉ ra giá trị log2 của giới hạn kích cỡ của palette_max_size_TH1, và palette_max_size_TH1 có thể được dẫn xuất là $1 \ll \text{log2_Palette_CU_size_TH1}$.

Phân tử cú pháp log2_palette_max_size_TH1 chỉ ra giá trị log2 của (palette_max_size_TH1+1), và palette_max_size_TH1 có thể được dẫn xuất là $(1 \ll \text{log2_palette_max_size_TH1}) - 1$. palette_max_size_TH1 chỉ ra chỉ số có thể chấp thuận

được tối đa của bảng của bảng màu cho CU có kích cỡ lớn hơn `Palette_CU_size_TH1` và có thể được giới hạn nằm trong phạm vi từ 1 đến 63.

Phần tử cú pháp `log2_palette_max_size_default` chỉ ra giá trị `log2` của `(palette_max_size_default+1)`, và `palette_max_size_default` có thể được dẫn xuất là $(1 \ll \log2_palette_max_size_default) - 1$. `palette_max_size_default` chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu và có thể được giới hạn nằm trong phạm vi từ 1 đến 63.

Ở đây, các phần tử cú pháp `log2_palette_CU_size_TH1`, `log2_palette_max_size_TH1`, và `log2_palette_max_size_default` các phần tử cú pháp có thể được phân tích cú pháp/được phát tín hiệu dựa trên phần tử cú pháp `sps_palette_enabled_flag`, vốn là thông tin để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không. Ví dụ, khi giá trị của `sps_palette_enabled_flag` là 1 (tức là, khi nó chỉ ra rằng chế độ bảng màu được cho phép), thì các phần tử cú pháp `log2_palette_CU_size_TH1`, `log2_palette_max_size_TH1`, và `log2_palette_max_size_default` có thể được phân tích cú pháp/được phát tín hiệu.

Ngoài ra, một hoặc nhiều tập hợp của `palette_CU_size_TH` và `palette_max_size_TH` có thể được phát tín hiệu và được sử dụng để chỉ ra `palette_max_size`.

Các hình vẽ sau đây được tạo nên để diễn giải ví dụ cụ thể của sáng chế. Tên hoặc thuật ngữ cụ thể của các thiết bị cụ thể được minh họa trên các hình vẽ (ví dụ, tên của cú pháp/các phần tử cú pháp, v.v.) được giới thiệu theo cách làm ví dụ, và vì thế các dấu hiệu kỹ thuật của sáng chế không được giới hạn ở các tên cụ thể được sử dụng trên các hình vẽ dưới đây.

Fig.7 thể hiện một cách sơ lược ví dụ về phương pháp mã hóa video/hình ảnh theo phương án (các phương án) của sáng chế.

Phương pháp được minh họa trên Fig.7 có thể được thực hiện bởi thiết bị mã hóa 200 được thể hiện trên Fig.2. Một cách cụ thể, các bước S700 và S710 trên Fig.7 có thể được thực hiện bởi bộ dự đoán 220 được thể hiện trên Fig.2, và bước S720 trên Fig.7 có thể được thực hiện bởi bộ mã hóa entropi 240 được thể hiện trên Fig.2. Bên cạnh đó, phương pháp được minh họa trên Fig.7 có thể gồm các phương án được mô tả ở trên trong sáng chế. Theo đó, phần mô tả chi tiết về các phần dư thừa trong các phương án được mô tả ở trên và Fig.7 sẽ được lược bỏ hoặc được đơn giản hóa.

Tham khảo đến Fig.7, thiết bị mã hóa có thể dẫn xuất giá trị thoát cho khối hiện tại trong chế độ bảng màu (S700).

Theo phương án, thiết bị mã hóa có thể xác định chế độ dự đoán cho khối hiện tại và thực hiện việc dự đoán. Ví dụ, thiết bị mã hóa có thể xác định xem liệu sẽ thực hiện việc dự đoán liên ảnh hoặc việc dự đoán trong ảnh trên khối hiện tại. Theo cách thay thế, thiết bị mã hóa có thể xác định xem liệu có thực hiện việc dự đoán trên khối hiện tại hay không dựa trên chế độ CIIP, chế độ IBC, chế độ bảng màu, hoặc tương tự. Thiết bị mã hóa có thể xác định chế độ dự đoán dựa trên chi phí RD. Thiết bị mã hóa có thể dẫn xuất các mẫu dự đoán cho khối hiện tại bằng cách thực hiện việc dự đoán trong chế độ dự đoán được xác định. Hơn nữa, thiết bị mã hóa có thể tạo ra và mã hóa thông tin (ví dụ, thông tin chế độ dự đoán) liên quan đến việc dự đoán được áp dụng cho khối hiện tại.

Khi xác định được rằng chế độ bảng màu được cho phép và việc dự đoán dựa trên chế độ bảng màu được thực hiện trên khối hiện tại, thì thiết bị mã hóa có thể áp dụng tạo mã chế độ bảng màu được bọc lỏ trong các phương án được mô tả ở trên. Nghĩa là,

thiết bị mã hóa có thể dẫn xuất các mục nhập bảng màu, các chỉ số bảng màu, các giá trị thoát, và tương tự bằng cách áp dụng việc tạo mã chế độ bảng màu cho khối hiện tại.

Theo ví dụ, thiết bị mã hóa có thể tạo ra thông tin mục nhập bảng màu dựa trên các giá trị mẫu của khối hiện tại. Nghĩa là, thiết bị mã hóa có thể dẫn xuất các mục nhập bộ dự đoán bảng màu và thông tin sử dụng lại mục nhập bảng màu được sử dụng trong khối được tạo mã trong chế độ bảng màu trước đó để tạo cấu hình bảng của bảng màu và có thể dẫn xuất các mục nhập bảng màu cho khối hiện tại. Ví dụ, như được thể hiện trên các bảng 5 và 6, thiết bị mã hóa có thể dẫn xuất thông tin mục nhập bảng màu như là `palette_predictor_run`, `num_signalled_palette_entries`, và `new_palette_entries` được sử dụng để tạo cấu hình bảng của bảng màu.

Bên cạnh đó, thiết bị mã hóa có thể tạo ra thông tin chỉ số bảng màu cho khối hiện tại dựa trên thông tin mục nhập bảng màu. Nghĩa là, thiết bị mã hóa có thể dẫn xuất giá trị chỉ số bảng màu của mỗi mẫu trong khi đi ngang qua các mẫu của khối hiện tại theo hướng quét ngang qua (hướng dọc hoặc hướng ngang) và tạo cấu hình bản đồ chỉ số bảng màu. Ví dụ, như được thể hiện trên các bảng 5 và 6 ở trên, thiết bị mã hóa có thể dẫn xuất thông tin mục nhập bảng màu như là `palette_transpose_flag`, `palette_idx_idc`, `copy_above_indices_for_final_run_flag`, `num_palette_indices_minus1` được sử dụng để tạo cấu hình bản đồ chỉ số bảng màu.

Ở đây, bảng của bảng màu có thể gồm các giá trị màu sắc biểu diễn (các mục nhập bảng màu) cho các mẫu trong khối hiện tại và có thể gồm có bảng màu các giá trị chỉ số tương ứng với các giá trị màu sắc tương ứng. Nghĩa là, thiết bị mã hóa có thể dẫn xuất giá trị chỉ số bảng màu tương ứng với mục nhập (giá trị màu sắc) trong bảng của bảng màu cho mỗi mẫu trong khối hiện tại và phát tín hiệu nó đến thiết bị giải mã.

Thiết bị mã hóa có thể mã hóa thông tin hình ảnh gồm thông tin mục nhập bảng màu và thông tin chỉ số bảng màu, và phát tín hiệu nó đến thiết bị giải mã.

Bên cạnh đó, khi thực hiện việc dự đoán dựa trên chế độ bảng màu trên khối hiện tại, thì thiết bị mã hóa có thể dẫn xuất giá trị thoát cho khối hiện tại gồm ít nhất một mẫu thoát được tạo mã.

Như được mô tả ở trên, vì có hiệu quả để truyền theo cách bổ sung giá trị mẫu tương ứng cho mẫu có giá trị khác với các giá trị của các mẫu lân cận trong khối hiện tại trong chế độ bảng màu theo hiệu quả tạo mã, nên giá trị mẫu này có thể được phát tín hiệu là giá trị thoát. Trong trường hợp này, vì giá trị thoát là dữ liệu bổ sung, nên việc lượng tử hóa có thể được thực hiện để tiết kiệm nó. Bên cạnh đó, không có phép biến đổi nào được áp dụng cho giá trị thoát của chế độ bảng màu, và giá trị được lượng tử hóa có thể được phát tín hiệu một cách trực tiếp.

Thiết bị mã hóa có thể dẫn xuất giá trị thoát được lượng tử hóa dựa trên giá trị thoát và thông số lượng tử hóa (S710).

Theo phương án, thiết bị mã hóa có thể dẫn xuất giá trị thoát được lượng tử hóa bằng cách áp dụng thông số lượng tử hóa cho giá trị thoát cho giá trị thoát.

Ở đây, thông số lượng tử hóa có thể được dẫn xuất dựa trên thông tin thông số lượng tử hóa tối thiểu liên quan tới chế độ bỏ qua biến đổi. Ví dụ, thông số lượng tử hóa có thể được dẫn xuất dựa trên thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi được thể hiện trên các bảng 7 đến 9. Như được mô tả ở trên, vì không có phép biến đổi nào được áp dụng cho giá trị thoát của chế độ bảng màu, nên giá trị thoát có thể được lượng tử hóa dựa trên thông tin thông số lượng tử hóa tối thiểu được sử dụng trong chế độ bỏ qua biến đổi.

Theo ví dụ cụ thể, như được thể hiện trong bảng 9, trước tiên, thiết bị mã hóa có thể dẫn xuất giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) dựa trên thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi. Bên cạnh đó, thiết bị mã hóa có thể lựa chọn giá trị lớn hơn nằm giữa giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) và thông số lượng tử hóa `Qp` (`Qp'`Y cho thành phần độ chói và `Qp'`Cb hoặc `Qp'`Cr cho thành phần sắc độ) và sử dụng nó làm thông số lượng tử hóa trong chế độ bảng màu.

Nói cách khác, thông số lượng tử hóa trong chế độ bảng màu có thể có giá trị lớn hơn hoặc bằng giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) được dẫn xuất từ thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi.

Thiết bị mã hóa có thể dẫn xuất giá trị thoát được lượng tử hóa bằng cách áp dụng thông số lượng tử hóa trong chế độ bảng màu được dẫn xuất như được mô tả ở trên. Thiết bị mã hóa có thể tạo ra giá trị thoát được lượng tử hóa làm phần tử cú pháp `palette_escape_val` như được bộc lộ trên các bảng 5 và 6 và phát tín hiệu nó. Nghĩa là, thiết bị mã hóa có thể tạo ra và phát tín hiệu thông tin giá trị thoát được lượng tử hóa (ví dụ, `palette_escape_val`) đối với khối hiện tại gồm ít nhất một mẫu thoát được tạo mã. Hơn nữa, theo phương án, thiết bị mã hóa có thể tạo ra thông tin (ví dụ, `palette_escape_val_present_flag`) để chỉ ra rằng mẫu có giá trị thoát có mặt trong khối hiện tại và phát tín hiệu nó.

Theo một phương án, thiết bị mã hóa có thể giới hạn giá trị thoát được lượng tử hóa nằm trong phạm vi cụ thể. Vì các giá trị thoát có các đặc tính khác với các giá trị của các mẫu lân cận nên chúng được lượng tử hóa và được phát tín hiệu một cách trực tiếp. Tuy nhiên, lỗi do việc lượng tử hóa có thể xảy ra. Để làm giảm lỗi như vậy và để

tạo mã giá trị chính xác hơn, thì phạm vi của giá trị thoát được lượng tử hóa có thể được giới hạn dựa trên độ sâu bit.

Ví dụ, phạm vi của thông tin về giá trị thoát được lượng tử hóa có thể được xác định dựa trên độ sâu bit như được thể hiện trên các bảng 10 và 11 và có thể được giới hạn sao cho nó không lớn hơn $(1 \ll \text{BitDepth}) - 1$, ví dụ. Bên cạnh đó, độ sâu bit có thể gồm độ sâu bit BitDepthY cho thành phần độ chói và độ sâu bit BitDepthC cho thành phần sắc độ. Ở đây, phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần độ chói có thể có giá trị nằm giữa 0 và $(1 \ll \text{BitDepthY}) - 1$, và phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần sắc độ có thể có giá trị nằm giữa 0 và $(1 \ll \text{BitDepthC}) - 1$.

Hơn nữa, trong một phương án, thiết bị mã hóa có thể định rõ số lượng các mục nhập trong bảng của bảng màu (tức là, số lượng các chỉ số của bảng của bảng màu) và phát tín hiệu nó đến thiết bị giải mã. Nghĩa là, thiết bị mã hóa có thể xác định thông tin kích cỡ bảng màu liên quan tới chỉ số tối đa của bảng của bảng màu và phát tín hiệu nó. Thông tin kích cỡ bảng màu có thể là giá trị được thiết lập trước hoặc có thể được xác định dựa trên kích cỡ của đơn vị tạo mã.

Ví dụ, kích cỡ bảng màu có thể được biểu diễn là `palette_max_size` như được thể hiện trong bảng 12 có thể là giống nhau cho toàn bộ chuỗi hoặc có thể được xác định theo cách khác nhau theo kích cỡ CU (tức là, số lượng các điểm ảnh trong CU).

Ví dụ, kích cỡ bảng màu có thể được biểu diễn là `palette_max_size` như được thể hiện trên các bảng 13 và 14 và có thể được phát tín hiệu thông qua SPS. Trong trường hợp này, kích cỡ bảng màu (ví dụ, `palette_max_size`) có thể chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu và có thể được giới hạn nằm trong phạm vi từ 1 đến 63. Bên cạnh đó, kích cỡ bảng màu (ví dụ, `palette_max_size`) có thể được phát tín

hiệu dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không.

Hơn nữa, ví dụ, kích cỡ bảng màu có thể được biểu diễn là `log2_palette_max_size` như được thể hiện trên các bảng 15 và 16 và có thể được phát tín hiệu thông qua SPS. Trong trường hợp này, kích cỡ bảng màu (ví dụ, `log2_palette_max_size`) có thể chỉ ra giá trị `log2` của kích cỡ bảng màu (tức là, `palette_max_size+1`). Theo đó, `palette_max_size` chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu có thể được dẫn xuất bằng cách tính $(1 \ll \log2_palette_max_size) - 1$ và có thể được giới hạn nằm trong phạm vi từ 1 đến 63. Bên cạnh đó, kích cỡ bảng màu (ví dụ, `log2_palette_max_size`) có thể được phát tín hiệu dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không.

Hơn nữa, ví dụ, kích cỡ bảng màu có thể được dẫn xuất dựa trên `log2_palette_CU_size_TH1`, `log2_palette_max_size_TH1`, và `log2_palette_max_size_default` như được thể hiện trên các bảng 17 và 18 và có thể được phát tín hiệu thông qua SPS. Vì phương án cụ thể để dẫn xuất và phát tín hiệu kích cỡ bảng màu đã được mô tả ở trên trên các bảng 17 và 18, nên phần mô tả của nó sẽ được lược bỏ tại đây.

Thiết bị mã hóa có thể mã hóa thông tin hình ảnh (hoặc thông tin video) (S720). Ở đây, thông tin hình ảnh có thể gồm các loại thông tin khác nhau được sử dụng cho việc tạo mã chế độ bảng màu được mô tả ở trên.

Ví dụ, thiết bị mã hóa có thể tạo ra và mã hóa thông tin hình ảnh gồm thông tin về giá trị thoát được lượng tử hóa. Trong trường hợp này, giá trị thoát được lượng tử hóa có thể được tạo ra cho khối hiện tại gồm ít nhất một mẫu thoát được tạo mã.

Bên cạnh đó, thiết bị mã hóa có thể tạo ra và mã hóa thông tin hình ảnh gồm thông tin mục nhập bảng màu và thông tin chỉ số bảng màu.

Hơn nữa, thiết bị mã hóa có thể tạo ra và mã hóa thông tin hình ảnh gồm thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi. Trong trường hợp này, thông tin hình ảnh có thể gồm SPS, và SPS có thể gồm thông tin về thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi.

Bên cạnh đó, theo phương án, thiết bị mã hóa có thể xác định xem liệu có tạo mã khối hiện tại nhờ sử dụng chế độ bảng màu được mô tả ở trên dựa trên thông tin về việc liệu chế độ bảng màu có được cho phép hay không.

Ví dụ, như được thể hiện trên các bảng 1 đến 4, thiết bị mã hóa có thể xác định xem liệu chế độ bảng màu có được cho phép hay không, tạo ra thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không theo việc xác định và phát tín hiệu chế độ bảng màu thông qua SPS. Nếu giá trị của thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không là 1, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu được cho phép và, nếu giá trị của thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không là 0, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu không được cho phép.

Hơn nữa, ví dụ, thiết bị mã hóa có thể xác định xem liệu có tạo mã khối hiện tại hay không bằng cách áp dụng chế độ bảng màu cho nó dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ tạo mã bảng màu có được cho phép hay không. Ví dụ, như được thể hiện trên các bảng 1 đến 4, nếu giá trị của `sps_palette_enabled_flag` là 1 (vốn chỉ ra rằng chế độ tạo mã bảng màu được cho phép), thì thiết bị mã hóa có thể tạo ra và phát tín hiệu thông tin về việc liệu có tạo mã khối

hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không. Như được thể hiện trên các bảng 1 đến 4, thông tin về việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không có thể được biểu diễn bởi phần tử cú pháp `pred_mode_plt_flag`. Nếu giá trị của `pred_mode_plt_flag` là 1, thì nó có thể chỉ ra rằng chế độ bảng màu được áp dụng cho khối hiện tại và, nếu giá trị của `pred_mode_plt_flag` là 0, thì nó có thể chỉ ra rằng chế độ bảng màu không được áp dụng cho khối hiện tại. Bên cạnh đó, theo ví dụ, thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ tạo mã bảng màu có được cho phép hay không có thể được phát tín hiệu thông qua SPS, và thông tin (ví dụ, `pred_mode_plt_flag`) về việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không có thể được phát tín hiệu thông qua cú pháp đơn vị tạo mã. Hơn nữa, ví dụ, nếu giá trị của `pred_mode_plt_flag` là 1, thì chế độ bảng màu có thể được áp dụng cho khối hiện tại để tạo ra mẫu được tái tạo bằng cách phát tín hiệu cú pháp `palette_coding()` thông qua cú pháp đơn vị tạo mã.

Thiết bị mã hóa có thể tạo ra và mã hóa thông tin hình ảnh gồm thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không và thông tin (ví dụ, `pred_mode_plt_flag`) về việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không.

Bên cạnh đó, theo phương án, thiết bị mã hóa có thể tạo ra thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) cho chế độ bỏ qua biến đổi dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không và phát tín hiệu nó.

Thông tin hình ảnh gồm các loại thông tin khác nhau như được mô tả ở trên có thể được mã hóa và được xuất ra trong dạng luồng bit. Luồng bit có thể được truyền đến thiết bị giải mã qua mạng hoặc phương tiện lưu trữ (kỹ thuật số). Ở đây, mạng có thể

gồm mạng phát rộng và/hoặc mạng truyền thông, hoặc tương tự, và 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, và SSD.

Fig.8 thể hiện một cách sơ lược ví dụ về phương pháp mã hóa video/hình ảnh theo phương án (các phương án) của sáng chế.

Phương pháp được minh họa trên Fig.8 có thể được thực hiện bởi thiết bị giải mã 300 được minh họa trên Fig.3. Một cách cụ thể, bước S800 trên Fig.8 có thể được thực hiện bởi bộ giải mã entropi 310 được thể hiện trên Fig.3, và các bước S810 và S820 trên Fig.8 có thể được thực hiện bởi bộ dự đoán 330 được thể hiện trên Fig.3. Bên cạnh đó, phương pháp được minh họa trên Fig.8 có thể gồm các phương án được miêu tả ở trên trong sáng chế. Theo đó, phần mô tả chi tiết về các phần dư thừa trong các phương án được mô tả ở trên và Fig.8 sẽ được lược bỏ hoặc được đơn giản hóa.

Tham khảo đến Fig.8, thiết bị giải mã có thể nhận thông tin hình ảnh (hoặc thông tin video) từ luồng bit (S800).

Bộ giải mã entropi có thể phân tích cú pháp luồng bit để dẫn xuất thông tin (ví dụ, thông tin video/hình ảnh) cần thiết cho việc tái tạo hình ảnh (hoặc việc tái tạo ảnh). Trong trường hợp này, thông tin hình ảnh có thể gồm thông tin liên quan đến việc dự đoán (ví dụ, thông tin chế độ dự đoán). Bên cạnh đó, thông tin hình ảnh có thể gồm các loại thông tin khác nhau được sử dụng cho việc tạo mã chế độ bảng màu được mô tả ở trên. Ví dụ, thông tin hình ảnh gồm thông tin về việc liệu chế độ bảng màu có được cho phép hay không, thông tin về việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không, thông tin về giá trị thoát được lượng tử hóa, thông tin mục nhập bảng màu, thông tin chỉ số bảng màu, thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi. Nghĩa là, thông tin hình ảnh có thể gồm các loại thông tin khác

nhau cần thiết trong quy trình giải mã, và có thể được giải mã dựa trên phương pháp tạo mã, như là tạo mã Golomb hàm mũ, CAVLC, hoặc CABAC.

Theo phương án, thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm thông tin giá trị thoát được lượng tử hóa trong chế độ bảng màu từ luồng bit. Ví dụ, thông tin giá trị thoát được lượng tử hóa có thể là phần tử cú pháp `palette_escape_val` như được thể hiện trên các bảng 5 và 6. Trong trường hợp này, thông tin giá trị thoát được lượng tử hóa (ví dụ, `palette_escape_val`) có thể được thu nhận dựa trên thông tin (ví dụ, `palette_escape_val_present_flag`) chỉ ra việc liệu mẫu có giá trị thoát có mặt trong khối hiện tại hay không. Ví dụ, khi mẫu có giá trị thoát có mặt trong khối hiện tại (nghĩa là, khi giá trị của `palette_escape_val_present_flag` là 1), thì thiết bị giải mã có thể thu nhận thông tin giá trị thoát được lượng tử hóa (ví dụ, `palette_escape_val`) từ luồng bit. Nghĩa là, thiết bị giải mã có thể dẫn xuất giá trị thoát được lượng tử hóa bằng cách thu nhận thông tin giá trị thoát được lượng tử hóa cho khối hiện tại gồm ít nhất một mẫu thoát được tạo mã.

Thiết bị giải mã có thể dẫn xuất giá trị thoát cho khối hiện tại dựa trên thông tin giá trị thoát được lượng tử hóa và thông số lượng tử hóa (S810).

Theo phương án, thiết bị giải mã có thể dẫn xuất giá trị thoát bằng cách thực hiện việc khử lượng tử hóa (quy trình định cỡ) trên giá trị thoát được lượng tử hóa dựa trên thông số lượng tử hóa.

Ở đây, thông số lượng tử hóa có thể được dẫn xuất dựa trên thông tin thông số lượng tử hóa tối thiểu liên quan tới chế độ bỏ qua biến đổi. Ví dụ, thông số lượng tử hóa có thể được dẫn xuất dựa trên thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi được thể hiện trên các bảng 7 đến 9. Như được mô tả ở trên, vì không có phép biến đổi nào được áp dụng cho

giá trị thoát của chế độ bảng màu, nên giá trị thoát có thể được lượng tử hóa dựa trên thông tin thông số lượng tử hóa tối thiểu được sử dụng trong chế độ bỏ qua biến đổi. Ở đây, thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi có thể được phân tích cú pháp/được phát tín hiệu từ SPS.

Theo ví dụ cụ thể, như được thể hiện trong bảng 9, trước tiên, thiết bị giải mã có thể dẫn xuất giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) dựa trên thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi. Bên cạnh đó, thiết bị giải mã có thể lựa chọn giá trị lớn hơn nằm giữa giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) và thông số lượng tử hóa `Qp` (`Qp'`Y cho thành phần độ chói và `Qp'`Cb hoặc `Qp'`Cr cho thành phần sắc độ) và sử dụng nó làm thông số lượng tử hóa trong chế độ bảng màu.

Nói cách khác, thông số lượng tử hóa trong chế độ bảng màu có thể có giá trị lớn hơn hoặc bằng giá trị thông số lượng tử hóa tối thiểu (ví dụ, `QpPrimeTsMin`) được dẫn xuất từ thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) liên quan tới chế độ bỏ qua biến đổi.

Thiết bị giải mã có thể dẫn xuất giá trị thoát từ giá trị thoát được lượng tử hóa dựa trên thông số lượng tử hóa trong chế độ bảng màu được dẫn xuất như được mô tả ở trên.

Theo một phương án, thiết bị giải mã có thể giới hạn giá trị thoát được lượng tử hóa nằm trong phạm vi cụ thể. Vì các giá trị thoát có các đặc tính khác với các giá trị của các mẫu lân cận nên chúng được lượng tử hóa và được phát tín hiệu một cách trực tiếp. Tuy nhiên, lỗi do việc lượng tử hóa có thể xảy ra. Để làm giảm lỗi như vậy và để tạo mã giá trị chính xác hơn, thì phạm vi của giá trị thoát được lượng tử hóa có thể được giới hạn dựa trên độ sâu bit.

Ví dụ, phạm vi của thông tin về giá trị thoát được lượng tử hóa có thể được xác định dựa trên độ sâu bit như được thể hiện trên các bảng 10 và 11 và có thể được giới hạn sao cho nó không lớn hơn $(1 \ll \text{BitDepth}) - 1$, ví dụ. Bên cạnh đó, độ sâu bit có thể gồm độ sâu bit BitDepthY cho thành phần độ chói và độ sâu bit BitDepthC cho thành phần sắc độ. Ở đây, phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần độ chói có thể có giá trị nằm giữa 0 và $(1 \ll \text{BitDepthY}) - 1$, và phạm vi của thông tin giá trị thoát được lượng tử hóa cho thành phần sắc độ có thể có giá trị nằm giữa 0 và $(1 \ll \text{BitDepthC}) - 1$.

Hơn nữa, trong một phương án, thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm số lượng các mục nhập trong bảng của bảng màu (tức là, số lượng các chỉ số của bảng của bảng màu). Nghĩa là, thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm thông tin kích cỡ bảng màu liên quan tới chỉ số tối đa của bảng của bảng màu. Ở đây, thông tin kích cỡ bảng màu có thể là giá trị được thiết lập trước hoặc có thể được xác định dựa trên kích cỡ của đơn vị tạo mã.

Ví dụ, kích cỡ bảng màu có thể được biểu diễn là `palette_max_size` như được thể hiện trong bảng 12 có thể là giống nhau cho toàn bộ chuỗi hoặc có thể được xác định theo cách khác nhau theo kích cỡ CU (tức là, số lượng các điểm ảnh trong CU).

Ví dụ, kích cỡ bảng màu có thể được biểu diễn là `palette_max_size` như được thể hiện trên các bảng 13 và 14 và có thể được phân tích cú pháp/được phát tín hiệu thông qua SPS. Trong trường hợp này, kích cỡ bảng màu (ví dụ, `palette_max_size`) có thể chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu và có thể được giới hạn nằm trong phạm vi từ 1 đến 63. Bên cạnh đó, kích cỡ bảng màu (ví dụ, `palette_max_size`) có thể được phân tích cú pháp/được phát tín hiệu dựa trên thông tin (ví dụ,

sps_palette_enabled_flag) để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không.

Hơn nữa, ví dụ, kích cỡ bảng màu có thể được biểu diễn là log2_palette_max_size như được thể hiện trên các bảng 15 và 16 và có thể được phân tích cú pháp/được phát tín hiệu thông qua SPS. Trong trường hợp này, kích cỡ bảng màu (ví dụ, log2_palette_max_size) có thể chỉ ra giá trị log2 của kích cỡ bảng màu (tức là, palette_max_size+1). Theo đó, palette_max_size chỉ ra chỉ số có thể chấp thuận được tối đa của bảng của bảng màu có thể được dẫn xuất bằng cách tính $(1 \ll \log2_palette_max_size) - 1$ và có thể được giới hạn nằm trong phạm vi từ 1 đến 63. Bên cạnh đó, kích cỡ bảng màu (ví dụ, log2_palette_max_size) có thể được phân tích cú pháp/được phát tín hiệu dựa trên thông tin (ví dụ, sps_palette_enabled_flag) để chỉ ra việc liệu chế độ bảng màu có được cho phép hay không.

Hơn nữa, ví dụ, kích cỡ bảng màu có thể được dẫn xuất dựa trên log2_palette_CU_size_TH1, log2_palette_max_size_TH1, và log2_palette_max_size_default như được thể hiện trên các bảng 17 và 18 và có thể được phân tích cú pháp/được phát tín hiệu thông qua SPS. Vì phương án cụ thể để dẫn xuất và phân tích cú pháp/phát tín hiệu kích cỡ bảng màu đã được mô tả ở trên trên các bảng 17 và 18, nên phần mô tả của nó sẽ được lược bỏ tại đây.

Thiết bị giải mã có thể tạo ra các mẫu được tái tạo dựa trên giá trị thoát (S820).

Theo phương án, thiết bị giải mã có thể tạo ra các mẫu được tái tạo dựa trên giá trị thoát đối với khối hiện tại gồm ít nhất một mẫu thoát được tạo mã. Ví dụ, nếu có mẫu có giá trị thoát trong khối hiện tại (nghĩa là, khi giá trị của palette_escape_val_present_flag là 1), thì thiết bị giải mã có thể dẫn xuất giá trị thoát như được mô tả ở trên để tạo ra mẫu được tái tạo của mẫu thoát được tạo mã.

Bên cạnh đó, khi thực hiện việc dự đoán dựa trên chế độ bảng màu trên khối hiện tại (nghĩa là, khi chế độ bảng màu được áp dụng cho khối hiện tại), cho các mẫu khác với các mẫu thoát được tạo mã trong khối hiện tại, thì thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm thông tin mục nhập bảng màu và thông tin chỉ số bảng màu và tạo ra các mẫu được tái tạo dựa trên thông tin hình ảnh được thu nhận.

Theo ví dụ, thiết bị giải mã có thể tạo cấu hình bảng của bảng màu cho khối hiện tại dựa trên thông tin mục nhập bảng màu. Ví dụ, thông tin mục nhập bảng màu có thể gồm `palette_predictor_run`, `num_signalled_palette_entries`, `new_palette_entries`, và tương tự, như được thể hiện trên các bảng 5 và 6. Nghĩa là, thiết bị giải mã có thể dẫn xuất các mục nhập bộ dự đoán bảng màu và thông tin sử dụng lại mục nhập bảng màu được sử dụng trong khối được tạo mã trong chế độ bảng màu trước đó và dẫn xuất các mục nhập bảng màu cho khối hiện tại để tạo cấu hình bảng của bảng màu. Hơn nữa, thiết bị giải mã có thể tạo cấu hình bảng của bảng màu dựa trên các mục nhập bộ dự đoán bảng màu trước đó và các mục nhập bảng màu hiện tại.

Bên cạnh đó, thiết bị giải mã có thể tạo cấu hình bản đồ chỉ số bảng màu cho khối hiện tại dựa trên thông tin chỉ số bảng màu. Ví dụ, thông tin chỉ số bảng màu có thể gồm `palette_transpose_flag`, `palette_idx_idc`, `copy_above_indices_for_final_run_flag`, `num_palette_indices_minus1`, và tương tự, được sử dụng để tạo cấu hình bản đồ chỉ số bảng màu như được thể hiện trên các bảng 5 và 6. Nghĩa là, thiết bị giải mã có thể tạo cấu hình bản đồ chỉ số bảng màu (ví dụ, `PaletteIndexMap`) dựa trên thông tin (ví dụ, `palette_idx_idc`) chỉ ra giá trị chỉ số bảng màu của mỗi mẫu trong khi đi ngang qua các mẫu của khối hiện tại dựa trên thông tin (ví dụ, `palette_transpose_flag`) chỉ ra hướng quét ngang qua (hướng dọc hoặc hướng ngang).

Bên cạnh đó, thiết bị giải mã có thể dẫn xuất các giá trị mẫu cho các mục nhập bảng màu trong bảng của bảng màu dựa trên bản đồ chỉ số bảng màu. Thiết bị giải mã có thể tạo ra các mẫu được tái tạo dựa trên bản đồ chỉ số bảng màu và các giá trị mẫu cho các mục nhập bảng màu.

Ở đây, bảng của bảng màu có thể gồm các giá trị màu sắc biểu diễn (các mục nhập bảng màu) cho các mẫu trong khối hiện tại và có thể gồm có bảng màu các giá trị chỉ số tương ứng với các giá trị màu sắc tương ứng. Theo đó, thiết bị giải mã có thể dẫn xuất các giá trị mẫu (tức là, các giá trị màu sắc) của các mục nhập trong bảng của bảng màu tương ứng với các giá trị chỉ số của bản đồ chỉ số bảng màu và tạo ra chúng như là mẫu được tái tạo các giá trị của khối hiện tại.

Hơn nữa, theo phương án, thiết bị giải mã có thể xác định xem liệu có tạo mã khối hiện tại nhờ sử dụng chế độ bảng màu hay không dựa trên thông tin về việc liệu chế độ bảng màu có được cho phép hay không.

Ví dụ, như được thể hiện trên các bảng 1 đến 4, thiết bị giải mã có thể thu nhận thông tin gồm thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không và, dựa trên thông tin này, thì thu nhận thông tin mục nhập bảng màu, thông tin chỉ số bảng màu, thông tin giá trị thoát được lượng tử hóa, và tương tự từ luồng bit. Ví dụ, nếu giá trị của thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không là 1, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu được cho phép và, nếu giá trị của thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không là 0, thì nó có thể chỉ ra rằng chế độ tạo mã bảng màu không được cho phép.

Hơn nữa, ví dụ, thiết bị giải mã có thể thu nhận thông tin (ví dụ, `pred_mode_plt_flag`) chỉ ra việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ

bảng màu cho nó từ luồng bit hay không dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ bảng màu có được cho phép hay không. Ví dụ, như được thể hiện trên các bảng 1 đến 4, nếu giá trị của `sps_palette_enabled_flag` là 1 (vốn chỉ ra rằng chế độ tạo mã bảng màu được cho phép), thì thiết bị giải mã có thể còn thu nhận thông tin (ví dụ, `pred_mode_plt_flag`) về việc liệu có tạo mã khối hiện tại bằng cách áp dụng chế độ bảng màu cho nó hay không. Trong trường hợp này, nếu giá trị của `pred_mode_plt_flag` là 1, thì nó có thể chỉ ra rằng chế độ bảng màu được áp dụng cho khối hiện tại và, nếu giá trị của `pred_mode_plt_flag` là 0, thì nó có thể chỉ ra rằng chế độ bảng màu không được áp dụng cho khối hiện tại.

Hơn nữa, ví dụ, như được thể hiện trên các bảng 1 đến 4, nếu giá trị của `pred_mode_plt_flag` là 1, thì thiết bị giải mã có thể còn thu nhận cú pháp `palette_coding()` và dẫn xuất các mẫu được tái tạo bằng cách áp dụng chế độ bảng màu cho khối hiện tại dựa trên thông tin được gồm trong cú pháp `palette_coding()`. Ví dụ, cú pháp `palette_coding()` có thể gồm thông tin mục nhập bảng màu, thông tin chỉ số bảng màu, thông tin giá trị thoát được lượng tử hóa, và tương tự.

Bên cạnh đó, theo phương án, thiết bị giải mã có thể thu nhận thông tin thông số lượng tử hóa tối thiểu (ví dụ, `min_qp_prime_ts_minus4`) cho chế độ bỏ qua biến đổi dựa trên thông tin (ví dụ, `sps_palette_enabled_flag`) về việc liệu chế độ tạo mã bảng màu có được cho phép hay không để dẫn xuất giá trị thoát và tạo ra các mẫu được tái tạo.

Trong hệ thống làm ví dụ được mô tả ở trên, các phương pháp được mô tả theo lưu đồ bằng cách sử dụng các chuỗi các bước và các khối. Tuy nhiên, sáng chế này không bị hạn chế vào thứ tự cụ thể của các bước, và một số bước có thể được thực hiện với các bước khác nhau và trong thứ tự khác với các phần được mô tả ở trên hoặc được thực hiện theo cách đồng thời. Ngoài ra, người có hiểu biết trung bình trong lĩnh vực kỹ thuật

tương ứng sẽ hiểu rằng các bước được thể hiện trong lưu đồ là không loại trừ, các bước khác có thể còn được gồm, hoặc một hoặc nhiều bước của lưu đồ có thể được xóa mà không làm ảnh hưởng tới phạm vi kỹ thuật của tài liệu này.

Phương pháp theo tài liệu này có thể được triển khai trong dạng phần mềm, và thiết bị mã hóa và/hoặc thiết bị giải mã theo tài liệu này có thể được gồm trong thiết bị mà thực hiện việc xử lý hình ảnh, ví dụ, TV, máy tính, điện thoại thông minh, hộp giải mã tín hiệu truyền hình hoặc thiết bị hiển thị.

Khi các phương án của tài liệu này được triển khai bởi phần mềm, thì phương pháp được đề cập ở trên có thể được triển khai 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 bên trong hoặc bên ngoài bộ xử lý và có thể được kết nối với bộ xử lý qua các phương tiện đã biết khác nhau. Bộ xử lý có thể gồm mạch tích hợp chuyên dụng (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 (flash), thẻ nhớ, phương tiện lưu trữ, và/hoặc thiết bị lưu trữ khác. Nói cách khác, các phương án theo tài liệu này có thể được triển khai và được thực thi trên bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc con chip. Ví dụ, các khối chức năng được minh họa trên các hình tương ứng có thể được triển khai và được thực thi trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển, hoặc con chip. Trong trường hợp này, thông tin về cách triển khai (ví dụ, thông tin về các lệnh) hoặc các thuật toán có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số.

Bên cạnh đó, thiết bị giải mã và thiết bị mã hóa mà phương án (các phương án) của tài liệu này được áp dụng có thể được gồm trong bộ thu phát phát rộng đa phương

tiện, thiết bị đầu cuối truyền thông di động, thiết bị video chiếu phim tại nhà, thiết bị video chiếu phim kỹ thuật số, camera để giám sát, thiết bị thoại video, và thiết bị truyền thông thời gian thực như là như truyền thông video, thiết bị phát luồng di động, phương tiện lưu trữ, máy ghi hình cầm tay, thiết bị cung cấp dịch vụ video theo yêu cầu (Video On-Demand, VoD), thiết bị video trên nền mạng viễn thông (Over The Top, OTT), thiết bị cung cấp dịch vụ phát luồng Internet, thiết bị video ba chiều (Three-Dimensional, 3D), thiết bị thực tế ảo (Virtual Reality, VR), thiết bị thực tế tăng cường (Augmented Reality, AR), thiết bị gọi điện thoại video, thiết bị đầu cuối của các phương tiện vận tải (ví dụ, thiết bị đầu cuối của xe (gồm xe ô tô), thiết bị đầu cuối của máy bay, và thiết bị đầu cuối của tàu thủy), và thiết bị video y tế, và có thể được sử dụng để xử lý tín hiệu video hoặc tín hiệu dữ liệu. Ví dụ, thiết bị video OTT có thể gồm máy chơi trò chơi, đầu đĩa Bluray, TV được kết nối Internet, hệ thống rạp hát tại nhà, điện thoại thông minh, máy tính bảng, và bộ ghi video số (Digital Video Recorder, DVR).

Bên cạnh đó, phương pháp xử lý mà phương án (các phương án) của tài liệu này được áp dụng có thể được tạo ra trong 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 phương án (các phương án) của bản mô tả này 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 tán mà dữ liệu đọc được bởi máy tính được lưu trữ trong đó. Phương tiện ghi đọc được bởi máy tính này có thể gồm, ví dụ, đĩa Blu-ray (Blu-ray Disc, BD), thiết bị bus nối tiếp vạn năng (Universal Serial Bus, USB), bộ nhớ chỉ đọc (Read Only Memory, ROM), ROM lập trình được (Programmable ROM, PROM), ROM lập trình được và xoá được (Erasable Programmable ROM, EPROM), ROM lập trình được và xoá được bằng điện (Electrically Erasable Programmable ROM, EEPROM), bộ nhớ truy cập ngẫu nhiên

(Random Access Memory, RAM), đĩa CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang học. Bên cạnh đó, phương tiện ghi đọc được bởi máy tính gồm các phương tiện được triển khai trong dạng sóng mang (ví dụ, truyền thông qua Internet). Bên cạnh đó, luồng bit được tạo ra bởi phương pháp mã hóa này có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính hoặc được truyền thông qua mạng truyền thông có dây/không dây.

Hơn thế nữa, phương án (các phương án) của sáng chế có thể được triển khai dưới dạng sản phẩm chương trình máy tính bởi mã chương trình, và mã chương trình này có thể được thực thi trên máy tính theo phương án (các phương án) của sáng chế. Mã chương trình có thể được lưu trữ trên bộ mang đọc được bởi máy tính.

Fig.9 thể hiện ví dụ về hệ thống phát luồng nội dung mà các phương án được bộc lộ trong sáng chế có thể áp dụng được vào đó.

Tham khảo đến Fig.9, hệ thống phát luồng nội dung được áp dụng vào các phương án của sáng chế có thể gồm máy chủ mã hóa, máy chủ phát luồng, máy chủ web, bộ phận lưu trữ phương tiện, thiết bị người dùng, và thiết bị đầu vào đa phương tiện.

Máy chủ mã hóa có chức năng nén thành các dữ liệu kỹ thuật số, các nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện, như là điện thoại thông minh, camera, máy ghi hình cầm tay, và tương tự, để tạo ra luồng bit, và truyền nó đến máy chủ phát luồng. Theo một ví dụ khác, trong trường hợp mà tại đó thiết bị đầu vào đa phương tiện, như là điện thoại thông minh, camera, máy ghi hình cầm tay, hoặc tương tự, trực tiếp tạo ra luồng bit, thì máy chủ mã hóa có thể được lược bỏ.

Luồng bit có thể được tạo ra bởi phương pháp mã hóa hoặc phương pháp tạo ra luồng bit mà các phương án của tài liệu này được áp dụng vào đó. Và máy chủ phát luồng có thể tạm thời lưu trữ luồng bit trong quy trình truyền hoặc nhận luồng bit.

Máy chủ phát luồng truyền dữ liệu đa phương tiện đến thiết bị người dùng dựa trên cơ sở của yêu cầu của người dùng thông qua máy chủ web, có chức năng như trang thiết bị để thông báo cho người dùng biết có dịch vụ gì. Khi người dùng yêu cầu dịch vụ mà mình cần, thì máy chủ web truyền yêu cầu đó đến máy chủ phát luồng, và máy chủ phát luồng truyền dữ liệu đa phương tiện đến người dùng. Trong vấn đề này, hệ thống phát luồng nội dung có thể gồm máy chủ điều khiển tách biệt, và trong trường hợp này, máy chủ điều khiển có chức năng điều khiển các chỉ thị/các hồi đáp giữa các phần thiết bị tương ứng trong hệ thống phát luồng nội dung.

Máy chủ phát luồng có thể nhận các nội dung từ bộ phận lưu trữ phương tiện và/hoặc máy chủ mã hóa. Ví dụ, trong trường hợp các nội dung này được nhận từ máy chủ mã hóa, thì các nội dung này có thể nhận được trong thời gian thực. Trong trường hợp này, máy chủ phát luồng có thể lưu trữ luồng bit này trong một khoảng thời gian được xác định trước để cung cấp dịch vụ phát luồng một cách 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, máy trợ lý cá nhân kỹ thuật số (Personal Digital Assistant, PDA), máy chơi đa phương tiện di động (Portable Multimedia Player, PMP), thiết bị dẫn đường, máy tính bảng lai laptop, máy tính bảng, máy tính xách tay siêu mỏng nhẹ, thiết bị đeo được (ví dụ, thiết bị đầu cuối kiểu đồng hồ (đồng hồ thông minh), thiết bị đầu cuối kiểu kính mắt (kính thông minh), thiết bị hiển thị gắn trên đầu (Head Mounted Display, HMD)), tivi kỹ thuật số, máy tính để bàn, biển chỉ dẫn kỹ thuật số, hoặc tương tự.

Mỗi máy chủ trong số các máy chủ trong hệ thống phát luồng nội dung có thể được vận hành dưới dạng máy chủ được phân tán, và trong trường hợp này, dữ liệu được nhận bởi mỗi máy chủ có thể được xử lý theo cách được phân tán.

Các điểm yêu cầu bảo hộ được mô tả trong sáng chế có thể được tổ hợp theo các cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng phương pháp của sáng chế có thể được tổ hợp và được triển khai dưới dạng thiết bị, và các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng thiết bị của sáng chế có thể được tổ hợp và được triển khai dưới dạng phương pháp. Bên cạnh đó, các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng phương pháp của sáng chế và các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng thiết bị có thể được tổ hợp và được triển khai dưới dạng thiết bị, và các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng phương pháp và các dấu hiệu kỹ thuật của yêu cầu bảo hộ dạng thiết bị có thể được tổ hợp và được triển khai dưới dạng phương pháp.

YÊU CẦU BẢO HỘ

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

thu nhận thông tin hình ảnh gồm thông tin giá trị thoát được lượng tử hóa trong chế độ bảng màu từ luồng bit;

dẫn xuất giá trị thoát cho khối hiện tại dựa trên thông tin giá trị thoát được lượng tử hóa và thông số lượng tử hóa; và

tạo ra các mẫu được tái tạo dựa trên giá trị thoát,

trong đó đối với khối hiện tại gồm ít nhất một mẫu được tạo mã thoát, thông tin giá trị thoát được lượng tử hóa được thu nhận,

trong đó thông tin hình ảnh gồm thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi, trong đó thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi là phần tử cú pháp liên quan đến việc vãn xuất của giá trị thông số lượng tử hóa tối thiểu và được phát tín hiệu trong tập hợp thông số chuỗi (Sequence Parameter Set, SPS),

trong đó SPS gồm thông tin về việc liệu chế độ bảng màu có được cho phép hay không, và trong đó dựa trên giá trị của thông tin về việc liệu chế độ bảng màu có được cho phép hay không bằng 1, thì thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được phát tín hiệu trong SPS,

trong đó giá trị thông số lượng tử hóa tối thiểu được dẫn xuất bởi phương trình được xác định trước dựa trên thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được phát tín hiệu trong SPS, và trong đó phương trình được xác định trước

được đưa ra là 4 cộng với giá trị được chỉ rõ bởi thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được phát tín hiệu trong SPS, và

trong đó thông số lượng tử hóa được dẫn xuất dựa trên giá trị thông số lượng tử hóa tối thiểu được dẫn xuất, và giá trị thoát được dẫn xuất dựa trên thông số lượng tử hóa.

2. Phương pháp giải mã hình ảnh theo điểm 1, trong đó thông tin hình ảnh gồm thông tin kích cỡ bảng màu về chỉ số tối đa của bảng của bảng màu,

trong đó thông tin kích cỡ bảng màu là giá trị được thiết lập trước hoặc được xác định dựa trên kích cỡ của đơn vị tạo mã.

3. Phương pháp giải mã hình ảnh theo điểm 1, phương pháp còn bao gồm các bước:

tạo cấu hình bảng của bảng màu dựa trên thông tin mục nhập bảng màu;

tạo cấu hình bản đồ chỉ số bảng màu cho khối hiện tại dựa trên thông tin chỉ số bảng màu;

dẫn xuất các giá trị mẫu cho các mục nhập bảng màu trong bảng của bảng màu dựa trên bản đồ chỉ số bảng màu; và

tạo ra các mẫu được tái tạo dựa trên bản đồ chỉ số bảng màu và các giá trị mẫu cho các mục nhập bảng màu,

trong đó thông tin hình ảnh gồm thông tin mục nhập bảng màu và thông tin chỉ số bảng màu.

4. Phương pháp mã hoá hình ảnh được thực hiện bởi thiết bị mã hoá, phương pháp bao gồm các bước:

dẫn xuất giá trị thoát trong chế độ bảng màu cho khối hiện tại;

dẫn xuất giá trị thoát được lượng tử hóa dựa trên giá trị thoát và thông số lượng tử hóa; và

mã hóa thông tin hình ảnh gồm thông tin về giá trị thoát được lượng tử hóa,

trong đó thông tin về giá trị thoát được lượng tử hóa mà được tạo ra cho khối hiện tại gồm ít nhất một mẫu thoát được tạo mã,

trong đó thông tin hình ảnh gồm thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi, trong đó thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi là phần tử cú pháp liên quan đến việc vãn xuất của giá trị thông số lượng tử hóa tối thiểu và được phát tín hiệu trong tập hợp thông số chuỗi (SPS),

trong đó SPS gồm thông tin về việc liệu chế độ bảng màu có được cho phép hay không, và trong đó dựa trên giá trị của thông tin về việc liệu chế độ bảng màu có được cho phép hay không bằng 1, thì thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được phát tín hiệu trong SPS,

trong đó trong việc phát tín hiệu trong SPS, thì giá trị của thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được xác định dựa trên giá trị thông số lượng tử hóa tối thiểu được dẫn xuất bởi phương trình được xác định trước, và trong đó phương trình được xác định trước được đưa ra là 4 cộng với giá trị được chỉ rõ bởi thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi là giá trị thông số lượng tử hóa tối thiểu, và

trong đó thông số lượng tử hóa được dẫn xuất dựa trên giá trị thông số lượng tử hóa tối thiểu được dẫn xuất, và giá trị thoát được lượng tử hóa được dẫn xuất dựa trên thông số lượng tử hóa.

5. Phương pháp mã hoá hình ảnh theo điểm 4, trong đó thông tin hình ảnh gồm thông tin kích cỡ bảng mã liên quan tới chỉ số tối đa của bảng của bảng mã,

trong đó thông tin kích cỡ bảng mã là giá trị được thiết lập trước hoặc được xác định dựa trên kích cỡ của đơn vị tạo mã.

6. Phương pháp mã hóa hình ảnh theo điểm 4, phương pháp bao gồm các bước:

tạo ra thông tin mục nhập bảng mã dựa trên các giá trị mẫu của khối hiện tại;

tạo ra thông tin chỉ số bảng mã cho khối hiện tại dựa trên thông tin mục nhập bảng mã; và

mã hóa thông tin hình ảnh gồm thông tin mục nhập bảng mã và thông tin chỉ số bảng mã.

7. Phương pháp truyền luồng bit được tạo ra bởi phương pháp mã hóa, phương pháp mã hóa bao gồm các bước:

dẫn xuất giá trị thoát trong chế độ bảng mã cho khối hiện tại;

dẫn xuất giá trị thoát được lượng tử hóa dựa trên giá trị thoát và thông số lượng tử hóa; và

mã hóa thông tin hình ảnh gồm thông tin về giá trị thoát được lượng tử hóa,

trong đó thông tin về giá trị thoát được lượng tử hóa mà được tạo ra cho khối hiện tại gồm ít nhất một mẫu thoát được tạo mã,

trong đó thông tin hình ảnh gồm thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi, trong đó thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi là phần tử cú pháp liên quan đến việc vãn xuất của giá trị thông số lượng tử hóa tối thiểu và được phát tín hiệu trong tập hợp thông số chuỗi (SPS),

trong đó SPS gồm thông tin về việc liệu chế độ bảng màu có được cho phép hay không, và trong đó dựa trên giá trị của thông tin về việc liệu chế độ bảng màu có được cho phép hay không bằng 1, thì thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được phát tín hiệu trong SPS,

trong đó trong việc phát tín hiệu trong SPS, thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi được xác định dựa trên giá trị thông số lượng tử hóa tối thiểu được dẫn xuất bởi phương trình được xác định trước, và trong đó phương trình được xác định trước được đưa ra là 4 cộng với giá trị được chỉ rõ bởi thông tin thông số lượng tử hóa tối thiểu cho chế độ bỏ qua biến đổi là giá trị thông số lượng tử hóa tối thiểu, và

trong đó thông số lượng tử hóa được dẫn xuất dựa trên giá trị thông số lượng tử hóa tối thiểu được dẫn xuất, và giá trị thoát được lượng tử hóa được dẫn xuất dựa trên thông số lượng tử hóa.

FIG. 1

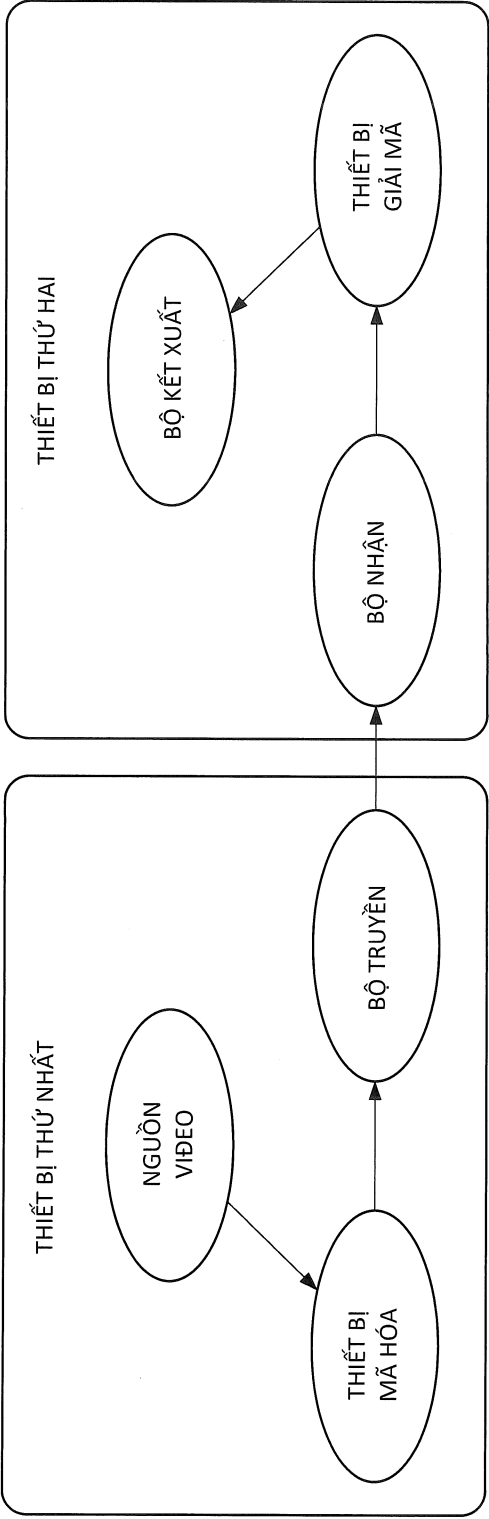


FIG. 3

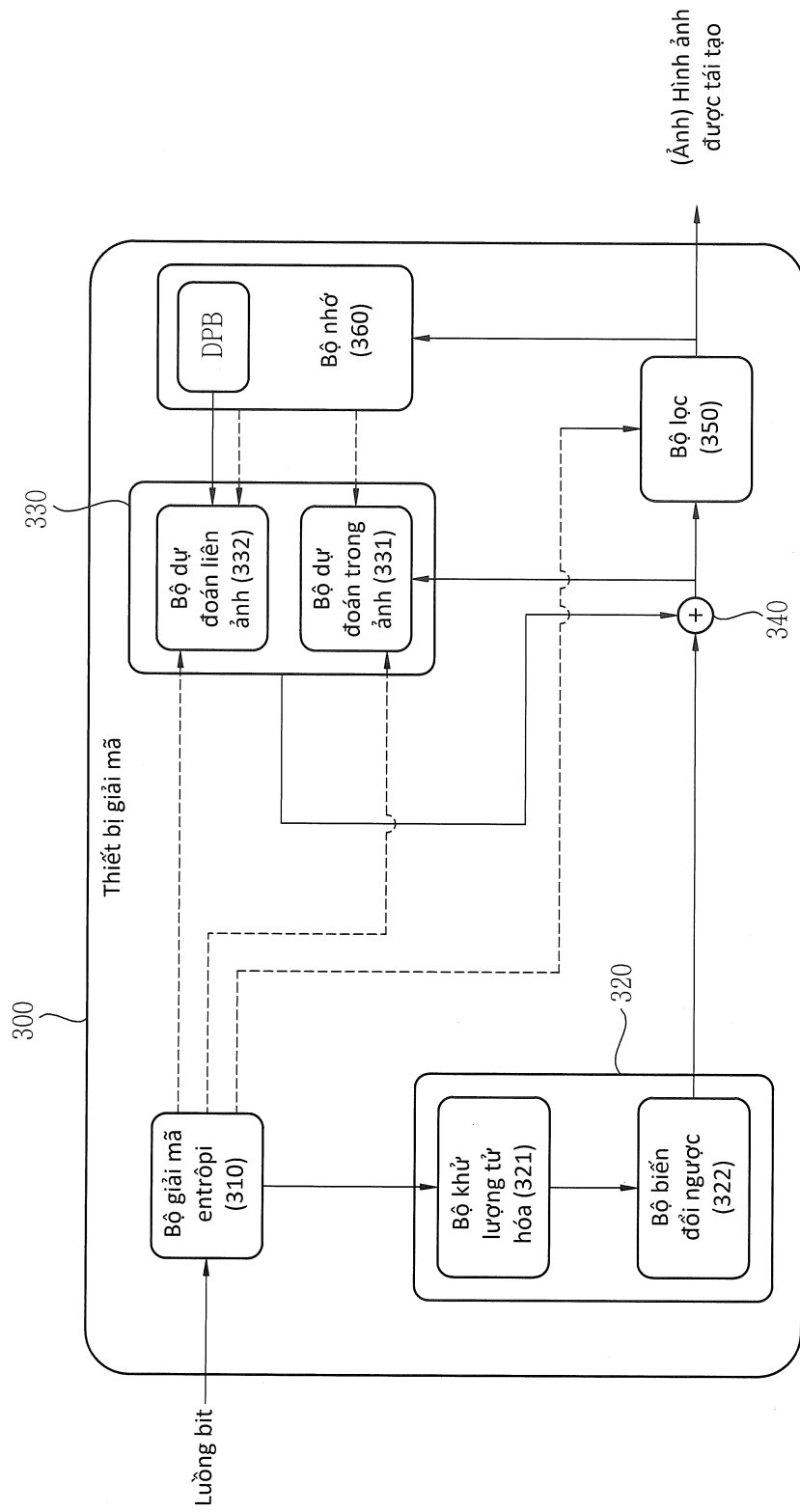


FIG. 4

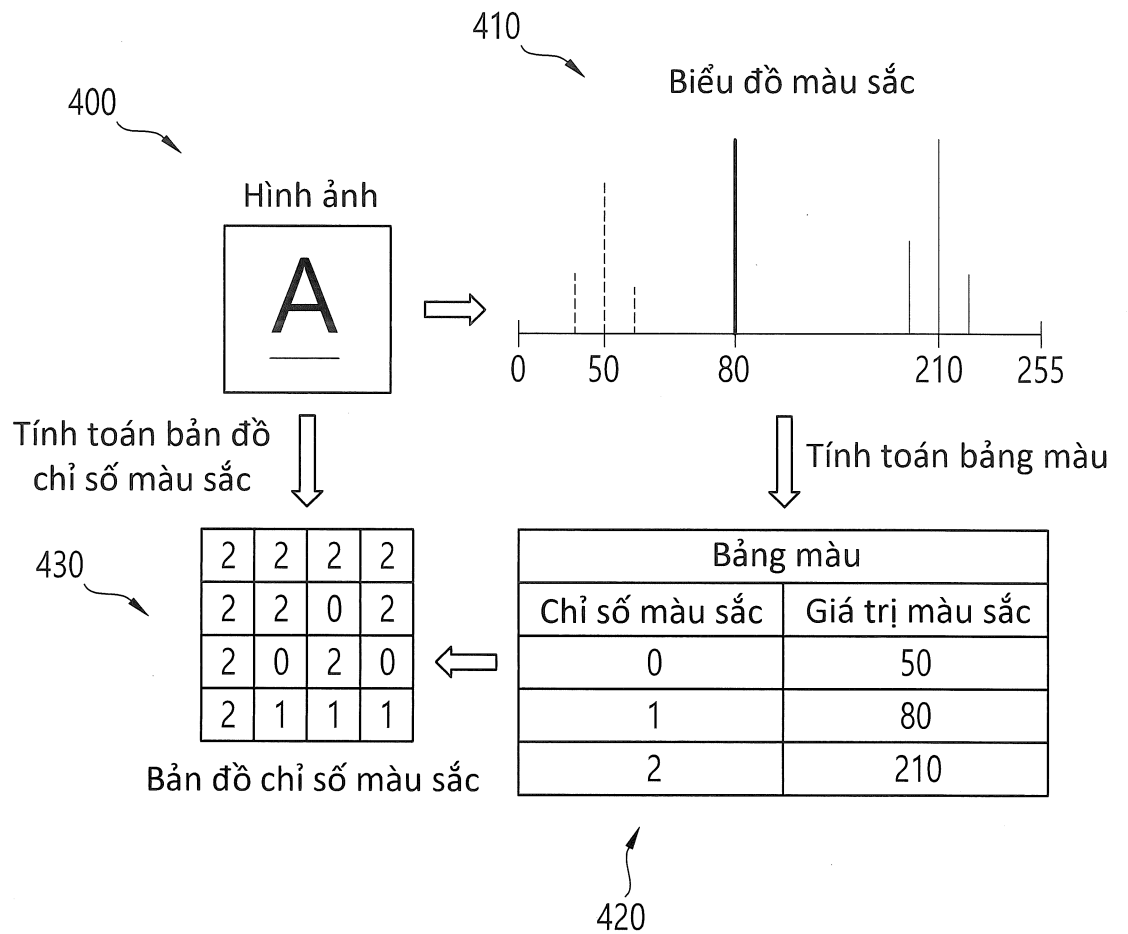


FIG. 5

quét ngang qua theo chiều ngang

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	0		1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

quét ngang qua theo chiều dọc

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	0		1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

FIG. 6

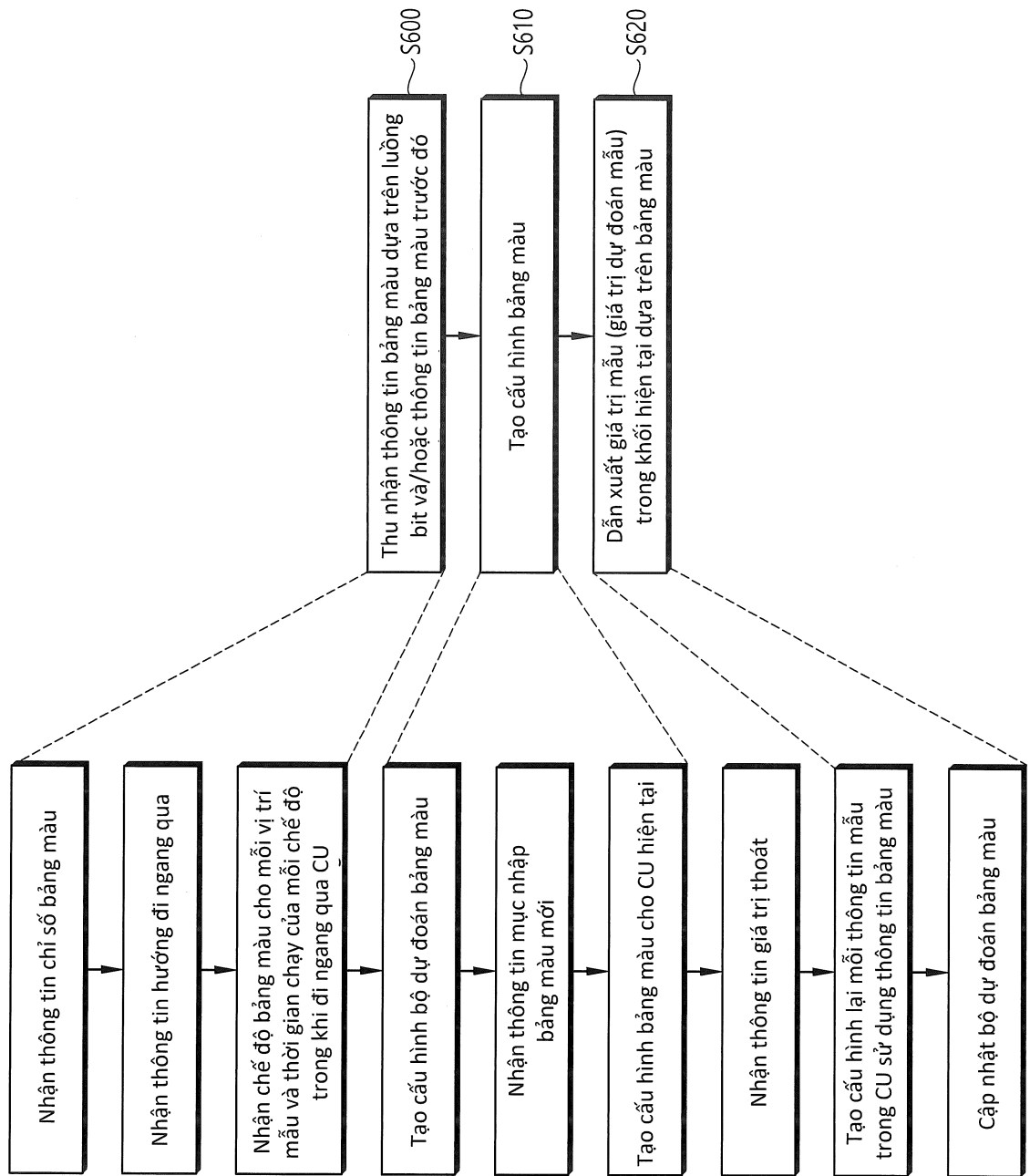


FIG. 7

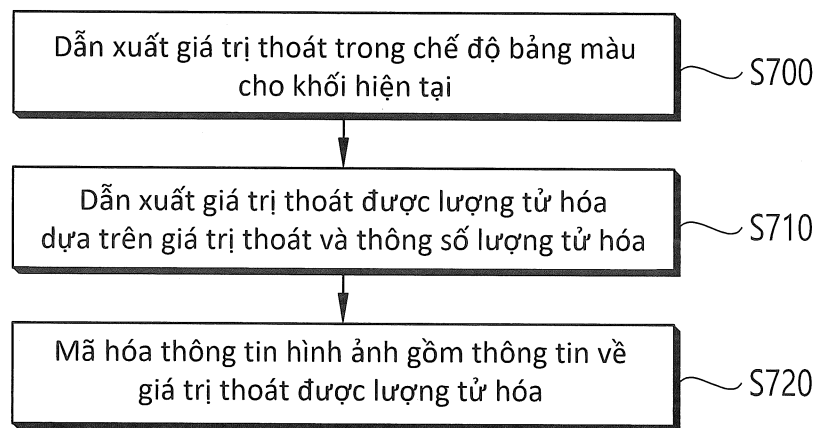


FIG. 8

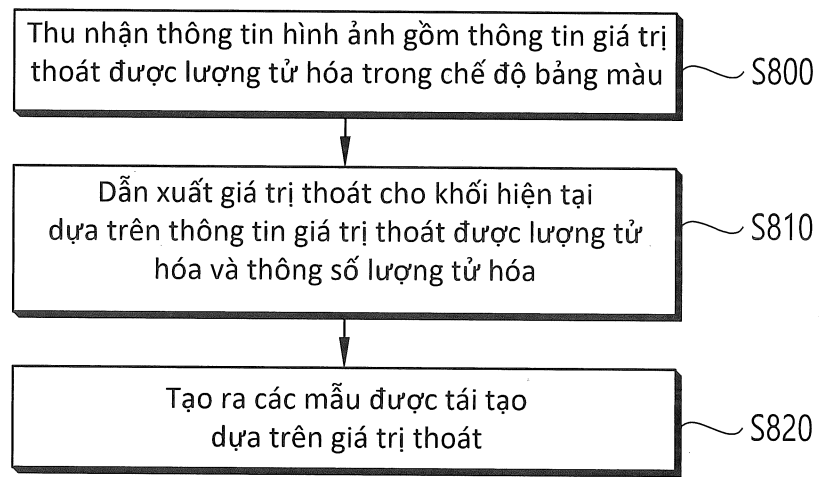


FIG. 9

